

# Rancang Bangun Aplikasi *Backend* Sistem Ujian Online Terpadu

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Herdian Adi Winarno

NIM:115060800111018



PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2015

## PENGESAHAN

RANCANG BANGUN APLIKASI *BACKEND* SISTEM UJIAN ONLINE TERPADU

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Herdian Adi Winarno  
NIM:115060800111018

Skripsi ini telah diuji dan dinyatakan lulus pada  
25 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Agi Putra Kharisma, S.T., M.T.  
NIK:2013048604301001

Dr. Eng Herman Tolle, S.T., M.T.  
NIP:197408232000121001

Mengetahui  
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T  
NIP: 19670801 199203 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 8 Januari 2016



Herdian Adi Winarno

NIM: 115060800111059

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang selalu memberikan rahmat, hidayah dan berkah-Nya sehingga penulis dapat menyelesaikan Proposal Skripsi dengan judul “Rancang Bangun Sistem Ujian Online Terpadu”. Proposal skripsi ini diajukan sebagai salah satu syarat untuk mendapatkan gelar sarjana S-1 di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Keberadaan skripsi ini tidak lepas dari bimbingan dan bantuan berbagai pihak, untuk itu penulis menyampaikan rasa terima kasih sebesar-besarnya kepada:

1. Agi Putra Kharisma, S.T., M.T. selaku Dosen Pembimbing I yang memberikan bimbingan, arahan, dan motivasi sehingga skripsi ini dapat terselesaikan;
2. Herman Tolle, Dr. Eng., S.T, M.T. selaku Dosen Pembimbing II yang turut memberikan bimbingan, dan arahan untuk kesempurnaan penulisan skripsi ini;
3. Ir. Sutrisno, M.T., selaku Ketua Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, dan Drs. Marji, M.T. selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya;
4. Kedua orang tua dan keluarga penulis yang senantiasa memberikan doa dan semangat hingga telah banyak membantu lancarnya penulisan dalam skripsi ini;
5. Seluruh dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, atas dukungan dan kerjasamanya;
6. Teman-teman Program Studi Teknik Informatika Universitas Brawijaya angkatan 2011 untuk bantuan dan semangatnya; dan
7. Semua pihak yang tidak dapat disebutkan satu-persatu.

Malang, 15 Januari 2016

Herdian Adi Winarno  
emailnyaherdian@gmail.com

## ABSTRAK

Ujian pada umumnya masih dilaksanakan secara tradisional. Peserta mengerjakan ujian di kertas dan penghitungan nilai dilakukan secara manual atau menggunakan *digital mark reader*. Ujian tradisional memiliki beberapa kekurangan, seperti: membutuhkan lebih banyak *resource*, rentan terhadap kesalahan manusia maupun teknis, rentan terhadap kecurangan dan waktu proses koreksi jawaban yang relatif lama. Oleh karena itu diperlukan sistem ujian online yang dapat memberikan solusi terhadap kekurangan tersebut. Seiring perkembangan teknologi, ujian online pun telah banyak dikembangkan. Meskipun demikian, masih terdapat kekurangan dari sistem ujian online sebelumnya. Seperti: kurang fleksibel, tidak memiliki *record* pengerjaan ujian peserta, tidak dapat menyertakan gambar dalam soal dan tidak memiliki *timer*. Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi yang dapat mengakomodasi kekurangan-kekurangan dari ujian tradisional maupun ujian online yang sudah pernah dikembangkan sebelumnya dengan menyediakan *web service* untuk berbagai *client* yang terdiri dari berbagai platform dan perangkat. Fitur-fitur yang diutamakan pada aplikasi ini adalah pengoreksian ujian, manajemen soal dan pengacakan soal. Berdasarkan hasil pengujian validasi, *security* dan *reliability* menunjukkan bahwa aplikasi *backend* sistem ujian online terpadu telah sesuai dengan kebutuhan dan berhasil diimplementasikan.

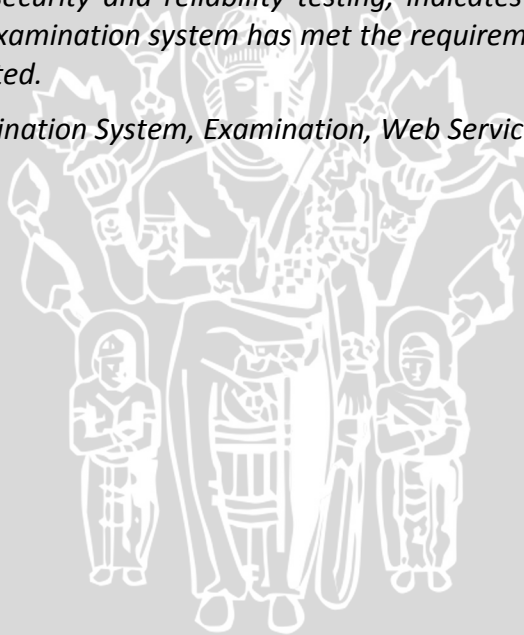
Kata kunci: Sistem Ujian Online, Ujian, *Web Service*.



## ABSTRACT

*Examination in general is still conducted traditionally. Examinee takes a test on paper and value calculation is performed manually or using digital mark reader. Traditional examination has some disadvantages, such as: requires more resources, prone to both human and technical error, prone to cheating and correction process time is relatively long. Therefore online examination system that can provide a solution to those disadvantages is required. Along with the development of technology, online examination has also been developed. Nevertheless, there are some shortages of the previous online examination system. Such as: lack of flexibility, inability to save examinee's records, inability to attach image in question and does not have a timer. This research aims to design and implement an application that can accommodate the disadvantages of traditional examination and online examination systems that have been developed earlier by providing a web service for various clients which consists of a wide range of platforms and devices. The main features of this application is exam correction, question management and question shuffling. Based on the results of validation, security and reliability testing, indicates that the backend application of online examination system has met the requirements and has been successfully implemented.*

**Keyword :**Online Examination System, Examination, Web Service.



## DAFTAR ISI

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| PENGESAHAN .....                                                          | 2  |
| PERNYATAAN ORISINALITAS .....                                             | 3  |
| ABSTRAK.....                                                              | 5  |
| DAFTAR ISI .....                                                          | 7  |
| BAB 1 PENDAHULUAN.....                                                    | 11 |
| 1.1 Latar belakang.....                                                   | 11 |
| 1.2 Rumusan masalah .....                                                 | 12 |
| 1.3 Tujuan .....                                                          | 12 |
| 1.4 Manfaat.....                                                          | 12 |
| 1.5 Batasan masalah .....                                                 | 12 |
| 1.6 Sistematika pembahasan .....                                          | 13 |
| BAB 2 TINJAUAN PUSTAKA.....                                               | 14 |
| 2.1 Ujian .....                                                           | 14 |
| 2.1.1 Jenis Soal Ujian.....                                               | 14 |
| 2.1.2 <i>Security Ujian Online</i> .....                                  | 14 |
| 2.1.3 <i>Software Requirement Patterns</i> untuk Sistem Ujian Online..... | 15 |
| 2.2 Arsitektur <i>Client-Server</i> .....                                 | 18 |
| 2.2.1 <i>Web Service</i> .....                                            | 18 |
| 2.2.2 CodeIgniter.....                                                    | 21 |
| 2.3 Algoritma Pengacakan Fisher-Yates .....                               | 21 |
| 2.4 UML.....                                                              | 23 |
| 2.4.1 <i>Use Case Diagram</i> .....                                       | 24 |
| 2.4.2 <i>Activity Diagram</i> .....                                       | 24 |
| 2.4.3 <i>Sequence Diagram</i> .....                                       | 25 |
| 2.4.4 <i>Class Diagram</i> .....                                          | 25 |
| BAB 3 METODOLOGI Penelitian .....                                         | 27 |
| 3.1 Studi Literatur .....                                                 | 27 |
| 3.2 Analisis Kebutuhan .....                                              | 28 |
| 3.3 Perancangan Sistem.....                                               | 28 |
| 3.4 Implementasi .....                                                    | 28 |
| 3.5 Pengujian dan Analisis .....                                          | 29 |

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| 3.6 Pengambilan Kesimpulan dan Saran .....                      | 29  |
| BAB 4 Analisis dan perancangan .....                            | 30  |
| 4.1 Analisis Kebutuhan .....                                    | 30  |
| 4.1.1 Gambaran Umum Aplikasi .....                              | 31  |
| 4.1.2 Identifikasi Aktor .....                                  | 31  |
| 4.1.3 Analisis Kebutuhan Fungsional .....                       | 31  |
| 4.1.4 Analisis Kebutuhan Non Fungsional .....                   | 59  |
| 4.2 Perancangan Aplikasi .....                                  | 59  |
| 4.2.1 Perancangan Arsitektur Sistem .....                       | 60  |
| 4.2.2 Perancangan <i>Activity Diagram</i> .....                 | 66  |
| 4.2.3 Perancangan <i>Sequence Diagram</i> .....                 | 87  |
| 4.2.4 Perancangan Class Diagram .....                           | 108 |
| 4.2.5 Perancangan Basis Data .....                              | 109 |
| 4.2.6 Perancangan Antarmuka .....                               | 110 |
| BAB 5 Implementasi .....                                        | 118 |
| 5.1 Spesifikasi Sistem .....                                    | 118 |
| 5.1.1 Spesifikasi Perangkat Keras .....                         | 118 |
| 5.1.2 Spesifikasi Perangkat Lunak .....                         | 118 |
| 5.2 Batasan Implementasi .....                                  | 119 |
| 5.3 Implementasi <i>Database</i> .....                          | 119 |
| 5.4 Implementasi Kode Program .....                             | 123 |
| 5.4.1 Algoritma Pengiriman soal kepada <i>client</i> .....      | 123 |
| 5.4.2 Algoritma Koreksi Jawaban .....                           | 127 |
| 5.4.3 Algoritma Pengacakan Soal dan Pilihan Jawaban .....       | 128 |
| 5.4.4 Algoritma Untuk Menerima Jawaban dari <i>Client</i> ..... | 129 |
| 5.5 Implementasi Antarmuka .....                                | 132 |
| 5.5.1 Halaman Login Administrator .....                         | 132 |
| 5.5.2 Halaman Registrasi Administrator .....                    | 133 |
| 5.5.3 Halaman Lihat Daftar Ujian .....                          | 133 |
| 5.5.4 Halaman Buat Ujian .....                                  | 134 |
| 5.5.5 Halaman Edit Ujian .....                                  | 134 |
| 5.5.6 Halaman Lihat Detail Ujian .....                          | 134 |

|                                                         |     |
|---------------------------------------------------------|-----|
| 5.5.7 Halaman Tambah Peserta Ujian.....                 | 135 |
| Halaman Tampilkan Hasil Ujian .....                     | 135 |
| Halaman Lihat Daftar Soal.....                          | 136 |
| Halaman Buat Soal .....                                 | 136 |
| Halaman Edit Soal .....                                 | 137 |
| Halaman Tampilkan Profil Pengguna .....                 | 137 |
| Halaman Lihat Request Ujian.....                        | 138 |
| BAB 6 Pengujian dan analisis .....                      | 139 |
| 6.1 Pengujian Fungsional .....                          | 139 |
| 6.1.1 Kasus Uji .....                                   | 139 |
| 6.1.2 Hasil Pengujian Validasi.....                     | 148 |
| 6.2 Pengujian Non Fungsional .....                      | 158 |
| 6.2.1 Pengujian <i>Security</i> .....                   | 158 |
| 6.2.2 Pengujian <i>Reliability</i> .....                | 162 |
| 6.3 Analisis Hasil Pengujian.....                       | 167 |
| 6.3.1 Analisis Hasil Pengujian Validasi .....           | 167 |
| 6.3.2 Analisis Hasil Pengujian <i>Security</i> .....    | 167 |
| 6.3.3 Analisis Hasil Pengujian <i>Reliability</i> ..... | 168 |
| BAB 7 penutup .....                                     | 169 |
| 7.1 Kesimpulan.....                                     | 169 |
| 7.2 Saran .....                                         | 169 |
| DAFTAR PUSTAKA.....                                     | 170 |



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Ujian adalah suatu cara yang lazim digunakan untuk mengukur tingkat kemampuan seseorang. Mayoritas ujian pada saat ini masih dilakukan secara tradisional, yakni peserta ujian datang ke tempat ujian dan menjawab pertanyaan pada lembar jawaban dengan pensil ataupun pena. Kemudian lembar jawaban dikoreksi secara manual atau menggunakan *digital mark reader* (alat periksa digital).

Sistem ujian tradisional memiliki banyak kekurangan. Kekurangan yang pertama adalah sistem ujian tradisional membutuhkan lebih banyak *resource* seperti kertas, tinta dan biaya pengiriman soal. Kekurangan yang kedua adalah sistem ujian tradisional rentan terhadap berbagai macam kesalahan. Baik kesalahan teknis (*technical error*) maupun kesalahan manusia (*human error*). Kesalahan teknis meliputi LJK yang kotor atau rusak. Sedangkan kesalahan manusia meliputi kesalahan dalam mengisi LJK (Pengisian kode kelas, Kode MK), kesalahan koreksi, maupun kurang tebal dalam menandai jawaban (Pratondo, 2010). Kekurangan yang ketiga adalah proses koreksi sistem ujian tradisional membutuhkan *resource* dan waktu yang dibutuhkan relatif lama.

Dengan pesatnya perkembangan teknologi informasi, sistem ujian tradisional dapat dikembangkan lebih lanjut. Sistem yang baru harus dapat memberikan solusi terhadap kekurangan-kekurangan sistem ujian tradisional. Penelitian ini membahas tentang sistem ujian *online* terpadu sebagai alternatif dari sistem ujian tradisional.

Sistem ujian *online* terpadu memanfaatkan teknologi informasi dalam prosesnya. Peserta dapat mengikuti ujian dimana saja selama masih terhubung dengan internet. Sistem ujian *online* ini lebih hemat *resource* dibandingkan dengan sistem ujian tradisional karena sistem ujian *online* tidak menggunakan kertas, tinta dan bebas biaya pengiriman soal. Meskipun demikian, ujian online juga memiliki kekurangan. Salah satunya adalah rentan terhadap kecurangan. Sehingga dibutuhkan sebuah sistem yang tidak hanya handal tapi juga dapat meminimalisir kecurangan tersebut. Misalkan dengan cara pengacakan soal atau mekanisme lainnya.

Sistem ujian *online* sudah pernah dikembangkan sebelumnya. Namun masih banyak kekurangan yang harus dibenahi. Kekurangan sistem ujian *online* sebelumnya seperti tidak memiliki *record* hasil pengerjaan *user* (Pramudita, 2013), tidak dapat menyertakan gambar dalam soal (Pramudita, 2013) dan tidak memiliki *timer* dalam pengerjaan soal (Sukmawati, 2014).

Ujian *online* terdiri dari server dan *client*. Server menangani akses ke *database*, logika bisnis dari sistem dan pengiriman data ke *client*. Sedangkan *client* bertugas untuk berinteraksi dengan peserta ujian. *Client* dari sistem ujian *online* umumnya berupa aplikasi *desktop* atau berbasis web dan dijalankan menggunakan *browser* dari sebuah *personal computer*. Seiring dengan banyaknya pengguna *mobile device* di seluruh dunia yang mencapai lebih dari

2,23 miliar (E-marketer, 2015), sudah saatnya untuk menggunakan *mobile device* sebagai *client* dari sistem ujian *online* terpadu. Penggunaan *mobile device* sebagai alternatif *client* dari *personal computer* dapat meningkatkan fleksibilitas karena *mobile device* dapat dengan mudah dibawa kemana saja dibandingkan dengan *personal computer*. Penelitian ini membahas aplikasi *backend* atau server dari sistem ujian online yang dapat menyediakan *service* bagi aplikasi *client* yang dibangun dengan berbagai macam platform dan bahasa pemrograman untuk pelaksanaan ujian.

Dengan adanya sistem ujian online terpadu, diharapkan sistem ini dapat membantu instansi maupun organisasi dalam menyelenggarakan ujian online yang lebih hemat *resource*, lebih praktis dalam pengelolaan datanya dan lebih cepat dalam melakukan koreksi jawaban.

## 1.2 Rumusan masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka rumusan masalah dapat disusun sebagai berikut :

1. Bagaimana rancangan dan implementasi aplikasi *backend* ujian *online* terpadu yang dapat menyediakan *service* untuk mengirimkan soal dan menerima jawaban kepada *client* dengan *secure* dan *reliable*?
2. Bagaimana rancangan sistem ujian *online* terpadu yang dapat melakukan pengacakan soal, manajemen soal dan koreksi jawaban secara terpadu ?
3. Bagaimana tingkat keamanan dan *reliability* dari aplikasi *backend* sistem ujian online terpadu?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah untuk merancang dan membuat aplikasi *backend* sistem ujian *online* yang dapat menyediakan layanan untuk mengirimkan soal-soal kepada *client*, melakukan pengacakan soal dan koreksi jawaban secara terpadu sehingga organisasi atau instansi dapat terbantu dalam penyelenggaraan ujian *online*.

## 1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini adalah :

1. Suatu instansi atau organisasi dapat terbantu dalam menyelenggarakan ujian *online* secara terpadu.
2. Suatu instansi atau organisasi dapat melakukan koreksi jawaban secara terpadu dan cepat.

## 1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal :

1. Jenis soal yang dapat diakomodasi oleh sistem berupa *multiple choice*, *true or false* dan jawaban singkat.

## 1.6 Sistematika pembahasan

Penyusunan skripsi ini menggunakan kerangka pembahasan yang tersusun sebagai berikut :

### **Bab I            Pendahuluan**

Bab ini menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

### **Bab II            Tinjauan Pustaka**

Bab ini berisi kajian pustaka, referensi atau sumber-sumber yang berhubungan dengan permasalahan dalam skripsi baik mengenai software maupun hardware serta teori yang diperlukan dalam perancangan aplikasi efek gitar dengan kendali pergerakan menggunakan akselerometer

### **Bab III          Metode Penelitian**

Bab ini menjelaskan bagaimana metodologi untuk perancangan dan pembuatan aplikasi efek gitar dengan kendali pergerakan menggunakan akselerometer.

### **Bab IV          Analisis dan Perancangan**

Bab ini menjelaskan analisis dan perancangan aplikasi efek gitar dengan kendali pergerakan menggunakan akselerometer yang dapat menjawab permasalahan yang telah diuraikan pada rumusan masalah.

### **Bab V            Implementasi**

Bab ini berisi tentang implementasi aplikasi yang dibuat, meliputi penjelasan mengenai aplikasi efek gitar dengan kendali pergerakan menggunakan akselerometer.

### **Bab VI          Analisis dan Pengujian**

Bab ini berisi tentang teknik atau metode pengujian yang dilakukan pada aplikasi efek gitar dengan kendali pergerakan menggunakan akselerometer serta melakukan analisa terhadap hasil pengujian.

### **Bab VII        Penutup**

Bab ini menjelaskan kesimpulan yang dapat diambil dari penelitian skripsi disertai saran yang dapat dijadikan masukan untuk penelitian lebih lanjut.

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Ujian

Ujian adalah salah satu sarana untuk mengevaluasi proses belajar. Dalam dunia pendidikan, ujian dimaksudkan untuk mengukur taraf pencapaian suatu tujuan pengajaran oleh siswa atau mahasiswa sebagai peserta didik, sehingga siswa atau mahasiswa dapat mengetahui tingkat kemampuannya dalam memahami bidang studi yang ditempuh. Bila ternyata hasilnya belum maksimal, maka proses belajar harus ditingkatkan baik kualitas maupun kuantitas.

Tipe ujian dapat dibagi menjadi dua kategori. Ujian tradisional dan ujian *online*. Ujian tradisional adalah ujian yang masih menggunakan kertas. Peserta ujian harus datang di suatu tempat pada waktu tertentu untuk melaksanakan ujian. Sedangkan ujian *online* adalah ujian yang memanfaatkan teknologi dalam prosesnya. Peserta tidak lagi menggunakan kertas untuk mengerjakan ujian namun menggunakan alat seperti *personal computer*, *smartphone*, laptop dll. Mayoritas ujian online masih menggunakan *personal computer* dalam proses pengerjaannya. Selain itu, Peserta tidak harus hadir ke suatu tempat untuk melaksanakan ujian. Ujian dapat dilaksanakan dimanapun selama terdapat koneksi internet.

#### 2.1.1 Jenis Soal Ujian

Ujian, baik tradisional maupun online memiliki jenis soal yang beragam. Jenis soal diantaranya : *Multiple Choice*, *True or False*, Jawaban singkat, Essai, Oral, Praktek, *Open book* dan pemecahan masalah (Learnline, 2015). Jenis soal pada sistem ujian online terpadu dibatasi pada jenis soal *multiple choice*, *true or false*, dan jawaban singkat.

Jenis soal *Multiple choice* atau pilihan ganda adalah jenis soal yang menyediakan pilihan bagi peserta ujian. Pilihan jawaban umumnya terdiri dari empat pilihan yang mana hanya tersedia satu jawaban yang benar. Peserta harus memilih satu jawaban yang paling tepat dari pilihan-pilihan yang tersedia. Jenis soal *True or False* adalah jenis soal yang hanya menyediakan dua pilihan, benar atau salah. Soal berupa pernyataan dimana peserta harus memutuskan apakah pernyataan tersebut benar atau salah. Sedangkan jenis soal jawaban singkat adalah jenis soal yang tidak menyediakan pilihan seperti dua jenis soal sebelumnya. Peserta ujian harus menjawab pertanyaan dengan satu atau lebih kata yang tepat.

#### 2.1.2 Security Ujian Online

Permasalahan yang sering muncul dalam ujian online adalah keamanan. Keamanan dalam hal ini mencakup keaslian jawaban, kerahasiaan jawaban dan tidak adanya kecurangan yang terjadi. Yang dimaksud dengan keaslian jawaban adalah jawaban yang dikirimkan oleh peserta ujian, harus sama dengan yang diterima oleh server. Tidak ada pihak ketiga yang dapat mengubah jawaban dari peserta ujian. Sama halnya dengan kerahasiaan jawaban, tidak ada pihak yang

dapat melihat jawaban dari peserta ujian. Hanya peserta dan Administrator yang dapat melihat jawaban ujian. Jika kedua hal tersebut dapat dilakukan, maka sistem dikatakan aman. Sedangkan kecurangan adalah aktivitas yang membuat peserta menjawab pertanyaan tidak dengan kemampuannya sendiri. Hal ini menyebabkan hasil ujian tidak valid dan tidak dapat dijadikan tolak ukur untuk peserta ujian. Sistem ujian online yang aman, adalah sistem ujian online yang dapat mengatasi atau mengurangi kecurangan yang dapat terjadi.

Salah satu cara untuk mengatasi masalah keaslian dan kerahasiaan jawaban peserta ujian, adalah dengan menggunakan metode algoritma penyandian informasi, atau yang lebih dikenal dengan nama kriptografi. Dalam kriptografi terdapat metode yang cukup penting dalam pengamanan informasi, yaitu menjaga kerahasiaan data dengan enkripsi (*encryption*). Pada penelitian ini, metode enkripsi belum dibahas. Namun Aplikasi *Backend* Sistem Ujian Online menerapkan penggunaan token untuk setiap *request* ke aplikasi. Penggunaan token digunakan agar informasi ujian tidak dapat diakses oleh sembarang orang, namun hanya bisa diakses oleh pihak-pihak yang memiliki hak.

Kemudian menyangkut masalah kecurangan, terdapat cara untuk mengatasi hal tersebut. Salah satunya adalah dengan menerapkan kontrol prosedur yang akan meminimalisir (tidak menghilangkan) kecurangan (Cluskeyr, 2015). Kontrol prosedur yang pertama adalah melaksanakan ujian dalam satu waktu atau secara bersamaan. Dengan demikian, maka tidak akan ada peserta ujian yang akan memberikan jawabannya kepada peserta ujian lain yang akan mengikuti ujian tersebut setelahnya. Namun hal ini memiliki kekurangan, peserta ujian masih dapat bekerja sama dan saling berkomunikasi. Kontrol prosedur yang kedua adalah dengan memberikan batas waktu tertentu untuk masing-masing pertanyaan. Setiap pertanyaan memiliki waktu penyelesaian sendiri sehingga peserta ujian diharapkan fokus pada ujiannya masing-masing. Namun kontrol prosedur ini juga memiliki kekurangan. Peserta yang telah menjawab soal masih dapat memberikan jawabannya kepada peserta lain jika terdapat sisa waktu dalam menjawab soal. Kontrol prosedur yang ketiga adalah dengan mengacak soal dan pilihan sehingga masing-masing peserta ujian tidak mendapatkan urutan soal dan pilihan yang sama. Hal ini akan menyulitkan peserta ujian untuk melakukan kecurangan.

Penelitian ini menerapkan kontrol prosedur yang ketiga untuk mengatasi masalah kecurangan dalam ujian online. Soal dan pilihan akan diacak sehingga masing-masing peserta ujian tidak memiliki urutan soal yang sama. Selain itu, waktu pengerjaan ujian juga dibatasi sehingga peserta akan fokus untuk menyelesaikan ujiannya sendiri.

### **2.1.3 Software Requirement Patterns untuk Sistem Ujian Online**

Penggunaan ulang merupakan aktivitas penting dari proses pengembangan perangkat lunak yang bisa memberikan keuntungan terhadap pengembangan perangkat lunak yang dikembangkan. Penggunaan ulang bisa diterapkan dalam berbagai cara. Salah satunya adalah penggunaan ulang suatu

kebutuhan perangkat lunak. Penggunaan ulang kebutuhan perangkat lunak bisa membantu *requirement engineer* dalam proses elisitasi, analisis, validasi dan dokumentasi kebutuhan perangkat lunak (Sangeeta, 2015).

Dengan mengobservasi dan melakukan penelitian terhadap beberapa perangkat lunak yang memiliki domain yang sama, dalam hal ini adalah sistem ujian online, maka dapat dibentuk suatu pola kebutuhan perangkat lunak. Pola kebutuhan perangkat lunak sendiri merupakan salah satu jenis artifak perangkat lunak yang bisa digunakan dalam pengambilan kebutuhan yang juga dapat berpengaruh positif terhadap aktivitas lain seperti analisis, dokumentasi dan validasi (Sangeeta, 2015).

Tabel 2.1 merupakan daftar pola kebutuhan fungsional dan non-fungsional perangkat lunak dari sistem ujian online terpadu (Sangeeta, 2015).

**Tabel 2.1 Software Requirement Patterns untuk Sistem Ujian Online**

| Nama Pola Kebutuhan                       | Deskripsi                                                                                                                         |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <i>User Operation</i>                     | Untuk menspesifikasikan kemudahan antarmuka dan operasi pengguna dalam menggunakan sistem.                                        |
| <i>User Registration</i>                  | Untuk menspesifikasikan bagaimana pengguna baru terdaftar dan informasi disimpan untuk autentikasi.                               |
| <i>User Authentication</i>                | Untuk menspesifikasikan bahwa seseorang harus login dan terdaftar ke dalam sistem sebelum dapat melakukan aktivitas dalam sistem. |
| <i>User Authorization</i>                 | Untuk menspesifikasikan bahwa sekumpulan pengguna berwenang atau tidak dalam mengakses atau melihat suatu aktivitas.              |
| <i>System Operation</i>                   | Untuk menspesifikasikan platform yang digunakan untuk membangun atau menjalankan sistem.                                          |
| <i>Information Storage</i>                | Untuk menspesifikasikan bagaimana sebuah informasi atau konten direpresentasikan dan disimpan.                                    |
| <i>Presentation</i>                       | Untuk mendefinisikan elemen visual dan layout antarmuka dari registrasi OES dan website ujian.                                    |
| <i>Information Content Representation</i> | Untuk mendefinisikan skema dalam merepresentasikan konten informasi seperti text, gambar, dll.                                    |

|                               |                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Formula Calculation</i>    | Untuk mendefinisikan formula yang digunakan untuk perhitungan data.                                                                                    |
| <i>Navigation</i>             | Untuk menspesifikasikan bagaimana navigasi pada sistem.                                                                                                |
| <i>Data Lifespan</i>          | Untuk menspesifikasikan berapa lama suatu jenis informasi disimpan secara aktif dalam sistem setelah setelah dihapus atau disimpan dalam data history. |
| <i>Data History</i>           | Untuk menspesifikasikan data mana yang akan dipindah pada waktu tertentu dari penyimpanan sistem yang aktif ke <i>history record</i> .                 |
| <i>Entity Definition</i>      | Untuk mendefinisikan entitas dimana informasi disimpan dan jangka hidupnya.                                                                            |
| <i>Inquiry</i>                | Untuk mendefinisikan fungsi tampilan layar yang menampilkan informasi tertentu kepada pengguna.                                                        |
| <i>Report</i>                 | Untuk mendefinisikan laporan yang menampilkan informasi tertentu kepada pengguna.                                                                      |
| <i>Response Time</i>          | Untuk menampilkan berapa lama waktu yang dibutuhkan bagi sistem untuk merespon suatu <i>request</i> .                                                  |
| <i>Throughput, Efficiency</i> | Untuk menspesifikasikan laju sistem dalam melakukan proses input atau output.                                                                          |
| <i>Dynamic Capacity</i>       | Untuk menspesifikasikan kuantitas dari jenis entitas yang harus bisa dilakukan oleh sistem secara bersamaan.                                           |
| <i>Static Capacity</i>        | Untuk menspesifikasikan kuantitas dari suatu jenis entitas yang harus bisa disimpan secara permanen oleh sistem.                                       |
| <i>Error Notification</i>     | Untuk tampilan jika terjadi kesalahan                                                                                                                  |
| <i>Availability</i>           | Untuk mendefnisikan kapan sistem tersedia bagi pengguna.                                                                                               |
| <i>Expandability</i>          | Untuk menspesifikasikan cara bagi sistem untuk bisa dikembangkan nanti untuk mengakomodasi pertumbuhan volume.                                         |
| <i>Installability</i>         | Untuk menspesifikasikan seberapa mudah melakukan proses instalasi atau <i>upgrade</i>                                                                  |

|                                      |                                                                                                                      |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|                                      | sistem.                                                                                                              |
| <i>Approval</i>                      | Untuk menspesifikasikan bahwa aksi tertentu harus disetujui oleh pihak berwenang lainnya sebelum dapat dilaksanakan. |
| <i>Fee</i>                           | Untuk menspesifikasikan biaya yang harus dihitung, dilaporkan atau dipungut oleh sistem.                             |
| <i>Transaction</i>                   | Untuk mendefinisikan jenis <i>event</i> dalam masa hidup suatu entitas atau fungsi untuk masuk ke dalam transaksi.   |
| <i>Interface Interaction</i>         | Untuk mendefinisikan jenis antarmuka eksternal yang akan berhubungan dengan sistem lain.                             |
| <i>Customizability</i>               | Adaptasi terhadap kebutuhan masing-masing pengguna.                                                                  |
| <i>Concurrence with requirements</i> | Kesesuaian dengan dokumen kebutuhan perangkat lunak.                                                                 |
| <i>Security</i>                      | Melawan ancaman, korupsi dan pembajakan.                                                                             |

Pada penelitian ini, tidak semua pola kebutuhan diterapkan. Rancang bangun Aplikasi *Backend* Sistem Ujian Online Terpadu fokus pada penerapan pola kebutuhan perangkat lunak fungsional saja.

## 2.2 Arsitektur *Client-Server*

Arsitektur *client server* merupakan model konektivitas pada jaringan yang membedakan fungsi computer sebagai *client* dan *server*. Prinsip kerja dari arsitektur ini sederhana. *Server* sebagai penyedia layanan akan menunggu permintaan dari *client* (*request*) dan memprosesnya, kemudian *server* akan memberikan hasil kepada *client*. *Server* dan *client* terkoneksi melalui suatu jaringan seperti Internet.

### 2.2.1 *Web Service*

*Web Service* adalah sebuah unit logika aplikasi yang menyediakan data dan layanan kepada aplikasi lainnya. Aplikasi lain mengakses *web service* melalui web protokol dan data format seperti HTTP, XML dan SOAP, tanpa harus mengetahui bagaimana *web service* diimplementasikan (Bouguettaya, 2009).

*Web Service* dibuat untuk menjalankan tugas spesifik yang juga dapat melibatkan *web service* lainnya. Dengan berinteraksi dengan *web service* lainnya,

sebuah sistem, produk maupun proses yang inovatif dapat tercipta (Bouguettaya, 2009).

*Web Service* secara arsitektural dibagi menjadi tiga kategori, yaitu : REST, RPC-style dan REST-RPC hybrid (Richardson, 2007).

### 2.2.1.1 REST

REST adalah singkatan dari *Representational State Transfer*. REST merupakan sebuah konsep tentang bagaimana data dapat ditransfer secara elegan dan memanfaatkan sebagian besar fitur dari HTTP (Lorna, 2013). Yang dimaksud transfer data yang elegan adalah pengiriman data kepada *client* yang memiliki akses terhadap resource yang lebih mudah untuk ditemukan dan lebih terorganisir. REST menggunakan fitur HTTP seperti POST, GET, PUT dan DELETE untuk melakukan fungsi *Create, Read, Update, Delete* (CRUD) pada *resource* dari suatu sistem. Arsitektur REST memetakan masing-masing *resource* pada URL yang berbeda sehingga lebih mudah diakses oleh sistem lain dengan menggunakan HTTP. Penelitian ini menggunakan REST karena REST lebih mudah untuk diimplementasikan, lebih mudah dipelajari dan tidak membutuhkan layer pertukaran data tambahan.

Pada arsitektur REST, terdapat *REST Maturity Model*. *REST Maturity Model* adalah sebuah pengklasifikasian *Web Service* yang dibuat oleh Leonard Richardson. *REST Maturity Model* berfungsi untuk memperjelas batasan-batasan antara “REST” yang sebenarnya dan bukan. Lebih jelasnya, *REST Maturity Model* digunakan sebagai tolok ukur seberapa jauh suatu teknologi menerapkan konsep *information resource*, HTTP sebagai protokol aplikasi dan Hypermedia sebagai media kontrol (Sletten, 2010).

*REST Maturity Model* memiliki beberapa level. Semakin tinggi level, maka konsep REST yang diterapkan juga semakin tinggi. Tabel 2.1 adalah tabel dari *REST Maturity Model*.

**Tabel 2.1. Tabel Level REST Maturity Model**

| Level | Keterangan                                                                                                                                                                                                                                |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0     | Pada level ini adalah penerapan RPC. Tidak ada <i>resource</i> informasi. HTTP digunakan sebagai <i>transport protocol</i> dan tidak ada konsep dari hypermedia. Format untuk pertukaran data dapat berupa XML-SOAP, JSON, dan lain-lain. |
| 1     | Penggunaan URI untuk masing-masing <i>resource</i> telah digunakan. Namun pada level ini masih menggunakan satu metode HTTP untuk semua aktivitas. Seperti POST ataupun GET.                                                              |
| 2     | Pada level ini, sama dengan pada level 1. Namun metode yang digunakan untuk setiap aktivitas berbeda. HTTP diterapkan sebagai protokol aplikasi.                                                                                          |
| 3     | Level ini menerapkan konsep REST secara keseluruhan. HTTP digunakan                                                                                                                                                                       |

|                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sebagai protokol aplikasi dan sama halnya dengan level 2, seluruh metode dari HTTP digunakan untuk interaksi antara client dan server. Selain itu, Hypermedia telah digunakan sebagai media pertukaran data. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 2.2.1.2 RPC

Web Service yang menggunakan arsitektur RPC (*Remote Procedure Calls*) adalah *service* yang menggunakan pemanggilan *procedure* secara jarak jauh. RPC tetap menggunakan protokol HTTP namun hanya menggunakan fitur POST untuk melakukan *request* maupun memberikan *response*. *Service* yang berbasis RPC biasanya hanya memiliki satu *endpoint* atau semua *request* dari client akan dikirim ke URL yang sama (Mitchell, 2010). Protokol lain yang digunakan RPC selain HTTP adalah SOAP. SOAP atau *Simple Object Access Protocol* adalah sebuah protokol komunikasi yang memungkinkan sebuah sistem berkomunikasi dengan sistem yang lain melalui internet. SOAP juga merupakan sebuah format dalam untuk pengiriman pesan antara *client* dan *server* (w3school, 2015).

#### 2.2.1.3 REST-RPC Hybrid

*Web Service* yang digolongkan sebagai REST-RPC Hybrid adalah kombinasi dari REST dan RPC. Suatu web service digolongkan ke dalam kategori ini jika konsep maupun teknologi yang digunakan dari kedua arsitektur digunakan bersama. Keberadaan dari arsitektur hybrid ini memunculkan banyak kebingungan (Richardson, 2007). Contoh nyata dari arsitektur hybrid adalah Flickr Web API. Flickr Web API menggunakan GET untuk memodifikasi dan menghapus data atau *resource* dari sistem dimana hal ini seharusnya bukan fungsi dari GET. Jika Flickr Web API adalah RESTful, maka seharusnya proses modifikasi maupun penghapusan data menggunakan PUT dan DELETE. Flickr Web API digolongkan sebagai RESTful ketika *client* mengambil *resource* dari sistem menggunakan GET dan dikatakan sebagai RPC-style ketika *client* memodifikasi atau menghapus *resource* dari sistem (Richardson, 2007).

#### 2.2.1.4 JSON

*Javascript Object Notation* atau yang biasa dikenal dengan JSON, adalah format teks yang memfasilitasi pertukaran data antar aplikasi. JSON dapat digunakan untuk semua bahasa pemrograman. JSON menggunakan format sebagai berikut (json.org, 2014):

1. Objek (*Object*)  
Objek (*object*) merupakan sepasang nama / nilai yang tidakurut. Objek dimulai dengan kurung kurawal buka ({) dan diakhiri dengan kurung kurawal tutup (}). Setiap nama diganti dengan titik dua (:) dan setiap pasangan nama / nilai dipisahkan dengan koma (,).
2. Larik (*Array*)

Larik (*array*) adalah kumpulan nilai yang terurutkan. Larik dimulai dengan kurung kotak buka ([) dan diakhiri dengan kurung kotak tutup (]). Setiap nilai dipisahkan oleh koma (,).

3. Nilai (*Value*)

Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true* atau *false* atau *null*, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun betingkat.

4. String

String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan kutip ganda. Di dalam string dapat digunakan *backslashes* (\) untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string.

5. Angka

Angka sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.

### 2.2.2 CodeIgniter

CodeIgniter (CI) adalah sebuah framework untuk aplikasi berbasis web yang menggunakan bahasa pemrograman PHP. CodeIgniter menyediakan berbagai macam library untuk membantu pengembang aplikasi dalam membuat aplikasi. Dengan menggunakan CodeIgniter, seorang pengembang aplikasi tidak perlu mengembangkan aplikasi dari awal (*from scratch*) dan dapat langsung fokus pada pengerjaan fitur-fitur fungsional dari aplikasi yang ingin dibangun.

CodeIgniter menggunakan arsitektur MVC atau *Model View Controller*, dimana komponen terbagi menjadi tiga, yaitu model sebagai abstraksi database, views sebagai template dari halaman web dan controller sebagai tempat berjalannya proses aplikasi.

Terdapat banyak fitur yang disediakan oleh CodeIgniter. Seperti database class yang support dengan banyak platform, Ringan digunakan, Dukungan Active Record, Form dan validasi data, Keamanan dan XSS Filtering, Manajemen sesi, Enkripsi Data, Benchmarking, Sejumlah library dan class untuk mengakomodasi kebutuhan sistem, dan masih banyak lagi (ellislab, 2015).

Aplikasi Backend Sistem Ujian Online pada penelitian ini dibangun menggunakan CodeIgniter agar proses pengembangan aplikasi dapat berjalan lebih cepat dan aplikasi yang dihasilkan akan lebih baik dan *reliable*.

### 2.3 Algoritma Pengacakan Fisher-Yates

Algoritma Fisher-Yates merupakan algoritma pengacakan yang menghasilkan permutasi acak dari suatu himpunan yang terhingga. Keuntungan dari algoritma ini adalah setiap permutasi yang dihasilkan akan memiliki kemungkinan yang hampir sama rata (Susantol, 2013). Dengan demikian, algoritma ini cocok digunakan dalam sistem pengacakan soal ujian online. Karena tujuan dari pengacakan soal pada ujian online adalah supaya setiap peserta ujian tidak memiliki pola soal yang sama.

Metode dasar pada algoritma ini adalah sebagai berikut(Susantol, 2013) :

1. Tuliskan angka dari 1 sampai N.
2. Pilih sebuah angka acak K diantara 1 sampai dengan jumlah angka yang belum dicoret.
3. Dihitung dari bawah, coret angka K yang belum dicoret, dan tuliskan angka tersebut di lain tempat.
4. Ulangi langkah 2 dan 3 sampai semua angka sudah tercoret.
5. Urutan angka yang dituliskan pada langkah 3 adalah permutasi acak dari angka awal.

Terdapat dua versi dari algoritma Fisher-Yates, yaitu versi asli dan versi modern. Metode diatas adalah versi asli dari algoritma Fisher-Yates. sedangkan algoritma Fisher-Yates versi modern diperkenalkan oleh Richard Durstenfeld dan kemudian dipopulerkan oleh Donald E. Knuth dalam bukunya *The Art of Computer Programming* sebagai "Algorithm P".

Pada algoritma Fisher-Yates versi modern, angka yang terpilih tidak dicoret namun ditukar dengan angka pada elemen terakhir dari deret angka yang belum terpilih. Tabel 2.2 merupakan contoh pengacakan menggunakan algoritma Fisher-Yates versi modern. *Range* adalah jumlah angka yang belum terpilih, *roll* adalah angka acak yang terpilih berdasarkan *range*. *Scratch* adalah deret angka yang belum terpilih, dan *result* adalah hasil pengacakan yang didapatkan.

**Tabel 2.2 Contoh Pengacakan Algoritma Fisher-Yates versi modern.**

| Range | Roll | Scratch     | Result      |
|-------|------|-------------|-------------|
|       |      | 1,2,3,4,5,6 |             |
| 1-6   | 3    | 1,2,6,4,5   | 3           |
| 1-5   | 4    | 1,2,6,5     | 4,3         |
| 1-4   | 1    | 5,2,6       | 1,4,3       |
| 1-3   | 3    | 5,2         | 6,1,4,3     |
| 1-2   | 1    | 2,          | 5,6,1,4,3   |
| 1     | 1    |             | 2,5,6,1,4,3 |

Penelitian ini menggunakan algoritma Fisher-Yates versi modern. Hal ini karena algoritma versi modern memiliki keunggulan dibandingkan versi aslinya. Yang pertama, algoritma Fisher-Yates versi modern tidak mencoret angka yang terpilih namun menukarnya. Jika diimplementasikan, algoritma versi modern hanya memerlukan satu array untuk menyimpan deret angka. Sedangkan pada versi asli, diperlukan dua array untuk menyimpan deret angka yang belum terpilih dan deret yang sudah teracak. Hal ini tentu akan lebih menghemat memori jika terdapat banyak angka yang akan diacak.

Keunggulan lainnya adalah dari segi kompleksitas. Pada algoritma Fisher-Yates versi asli, langkah 3 adalah menghitung kembali jumlah elemen array dari akhir. Langkah ini kurang efisien. Sedangkan pada versi modernnya, jumlah elemen array tidak perlu dihitung kembali karena array yang digunakan adalah array awal yang kemudian hanya ditukar nilai-nilainya. Hal ini mengurangi kompleksitas algoritma Fisher-Yates dari yang semula  $O(n^2)$  menjadi  $O(n)$  (Wikipedia, 2015).

## 2.4 UML

Unified Modeling Language atau UML adalah salah satu alat yang telah menjadi standar dalam perancangan perangkat lunak. Dengan menggunakan UML, sebuah sistem dapat dimodelkan menjadi diagram-diagram yang merepresentasikan sistem tersebut. Model inilah yang kemudian menjadi *blueprint* untuk tahap implementasi perangkat lunak dan dokumentasi.

Secara umum penggunaan UML dibagi menjadi tiga. UML sebagai sketsa, *blueprint* dan bahasa pemrograman (Hamilton, 2006). UML sebagai sketsa adalah penggunaan UML untuk memodelkan kunci-kunci penting dari suatu sistem. Umumnya sistem yang dimodelkan adalah sistem yang skalanya kecil dan menengah. UML sebagai *blueprint* adalah penggunaan UML untuk memodelkan sebuah sistem yang relatif besar dan tidak mentolerir kesalahan. Seperti pada sistem rumah sakit dan sistem pabrik. Sedangkan UML sebagai bahasa pemrograman adalah penggunaan UML untuk memodelkan sebuah sistem yang kemudian diteruskan ke proses *deployment*. Hal ini berguna untuk membuat sistem yang berada pada berbagai macam platform. Namun penggunaan UML sebagai bahasa pemrograman masih dalam proses pengembangan dan masih membutuhkan penelitian lebih lanjut.

Berikut adalah kelebihan UML (Hamilton, 2006) :

- 1) UML adalah bahasa formal.  
Tiap elemen dari UML memiliki makna tersendiri sehingga tidak akan terjadi kesalah pahaman.
- 2) UML komprehensif  
UML dapat digunakan untuk memodelkan seluruh aspek-aspek penting dari sebuah sistem
- 3) UML memiliki skalabilitas yang tinggi  
UML dapat digunakan untuk memodelkan mulai dari sistem yang sederhana hingga sistem yang sangat besar.
- 4) UML memiliki banyak referensi  
UML telah dikembangkan lebih dari 15 tahun dan memiliki banyak sumber atau referensi yang dapat dijadikan bahan pembelajaran.
- 5) UML adalah standar

UML adalah standar untuk pemodelan yang dikontrol oleh open standart group yang memiliki kontribusi aktif dari berbagai macam vendor dan akademisi dari seluruh dunia sehingga transformabilitas dan interoperabilitas UML terjamin.

### 2.4.1 Use Case Diagram

*Use case diagram* adalah diagram grafis yang merepresentasikan interaksi antara *actor* dengan sistem dalam lingkungannya. Use case memodelkan fungsionalitas yang ada pada sistem. Use case hanya menggambarkan apa yang seharusnya dapat dilakukan oleh sistem atau kebutuhan fungsional, bukan menggambarkan apa yang seharusnya tidak dilakukan oleh sistem atau kebutuhan nonfungsional (Hamilton, 2006).

*Use case diagram* terdiri dari *actor*, *use case*, garis penghubung dan batasan sistem atau sub-sistem. *Actor* menggambarkan *stakeholder* yang berinteraksi dengan sistem. *Use case diagram* menggambarkan aktifitas yang dapat dilakukan oleh sistem. Sedangkan garis penghubung adalah garis yang menghubungkan *actor* dan *use case*. Dan batasan sistem atau sub-sistem menggambarkan apa yang ada di dalam dan diluar sistem atau sub-sistem. Batasan sistem atau sub-sistem diwakili dengan sebuah kotak persegi.

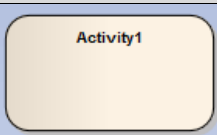
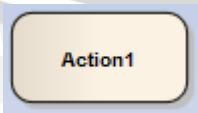
### 2.4.2 Activity Diagram

*Activity Diagram* adalah salah satu diagram UML yang digunakan untuk memodelkan alur dari proses pada sebuah sistem. *Activity diagram* menampilkan aksi-aksi yang saling berhubungan yang mewakili suatu proses yang terjadi dalam suatu sistem yang dimodelkan (Hamilton, 2006).

*Activity diagram* baik digunakan untuk memodelkan proses bisnis. Selain itu, *activity diagram* menggunakan simbol-simbol yang sama dengan flowchart sehingga diagram ini cocok digunakan untuk menggambarkan suatu proses kepada *audience* yang luas (Hamilton, 2006).

Tabel 2.3 merupakan tabel simbol yang digunakan dalam *Activity diagram*:

**Tabel 2.3 Tabel Simbol Activity Diagram**

| NO | Notasi                                                                              | Keterangan                                                                                                             |
|----|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| 1  |  | Notasi untuk memulai dan mengakhiri suatu activity diagram                                                             |
| 2  |  | Activity adalah proses yang dimodelkan.                                                                                |
| 3  |  | Action adalah bagian dari activity atau langkah-langkah yang membentuk suatu activity                                  |
| 4  |  | Decision adalah percabangan dalam activity diagram untuk mengeksekusi urutan action tertentu berdasarkan suatu kondisi |

|   |                                                                                   |                                                                                 |
|---|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| 5 |  | Fork digunakan untuk menunjukkan action-action yang dieksekusi secara bersamaan |
|---|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------|

### 2.4.3 Sequence Diagram

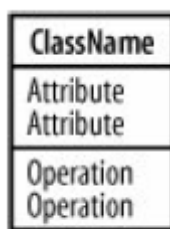
Sequence Diagram Sequence diagram menggambarkan interaksi antara aktor dengan objek serta interaksi antar objek didalam sistem(termasuk pengguna, display dan sebagainya), berupa *message* yang digambarkan sesuai dengan urutan waktu terjadinya interaksi (IBM, 2015). Sequence diagram terdiri atas dimensi vertikal(waktu) dan dimensi horizontal(objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Terdapat beberapa notasi pada sequence diagram seperti *lifeline*, *message*, *guards*, *alternative*, *option* dan *loop*. *Lifeline* mewakili obyek yang ikut berinteraksi dalam sistem. Lifeliness digambarkan sebagai kotak yang memiliki nama di dalamnya dengan garis putus-putus ke bawah. Message digambarkan dengan garis yang memiliki anah panah di ujungnya. Message/nama *method* ditempatkan diatas garis. Message yang dikirim mewakili operasi/ method yang diimplementasikan oleh obyek yang dituju. *Guard* digunakan ketika suatu kondisi harus dipenuhi sebelum sebuah message dapat dikirim. *Guard* diwakili dengan notasi [kondisi]. Alternatif pada sequence diagram digunakan jika terdapat dua atau lebih alternatif *sequence* yang harus dieksekusi berdasarkan kondisi tertentu. Sama halnya dengan percabangan pada pemrograman. Sedangkan *option*, hampir sama dengan *guard* namun *option* melibatkan sequence yang mengandung lebih dari 2 pengiriman message. Yang terakhir adalah *loop*. Loop adalah perulangan yang terjadi pada sequence ketika memenuhi kondisi tertentu.

### 2.4.4 Class Diagram

*Class* adalah sebuah cetak biru dari suatu objek. Sebuah objek dimodelkan dengan melakukan abstraksi. Abstraksi adalah proses menghilangkan informasi yang tidak penting dari sebuah objek dan hanya menggunakan informasi yang relevan dan cukup untuk mewakili suatu objek (Hamilton, 2006). Hasil dari abstraksi suatu objek adalah class. Sedangkan *Class Diagram* adalah sebuah diagram yang menggambarkan class yang dibutuhkan oleh suatu sistem serta relasi antar class tersebut (Hamilton, 2006).

Sebuah Class dalam Class diagram dimodelkan dengan kotak yang berisi Nama kelas, Attribut dan Behavior. Gambar 2.1 menunjukkan gambar dari sebuah class



**Gambar 2.1.** Contoh Sebuah Class

Dalam sebuah class diagram, class yang ada didalamnya saling berhubungan. Ada beberapa macam jenis relasi atau hubungan pada class diagram. Gambar 2.2 menunjukkan jenis-jenis relasi pada class diagram.



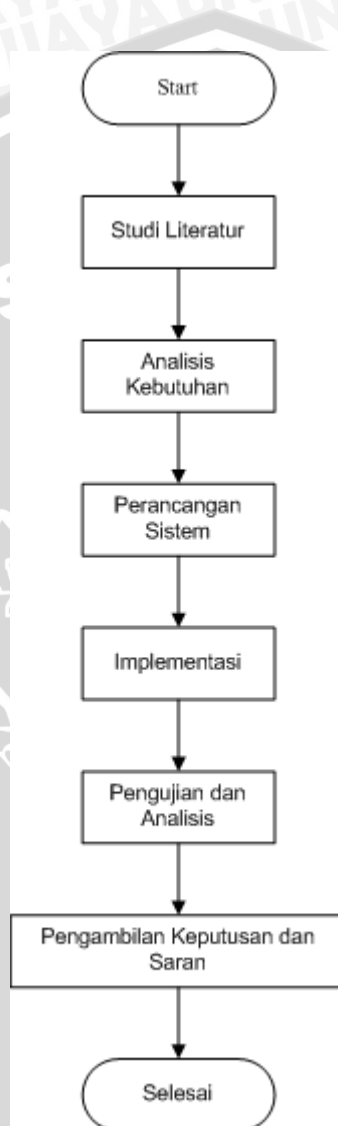
**Gambar 2.2** Jenis garis dalam class diagram

Dependensi adalah relasi dari *class* yang digunakan ketika suatu class membutuhkan class yang lain ketika ingin menggunakan objeknya (Hamilton, 2006). Asosiasi adalah relasi dari class yang mengandung referensi kepada objek dari kelas lain dalam bentuk atribut (Hamilton, 2006). Agregasi adalah versi yang lebih kuat dari asosiasi. Relasi ini berarti bahwa suatu class memiliki dan dapat berbagi objek dari *class* yang lain (Hamilton, 2006). Komposisi adalah versi yang lebih kuat dari agregasi. Suatu class dapat mengandung class lainnya dan tidak dapat dishare kepada bagian lain dari sistem. Komposisi berguna untuk memodelkan bagian internal yang membuat suatu class. Generalisasi atau *inheritance* adalah relasi terkuat untuk menunjukkan bahwa suatu class adalah bagian dari class yang lain. Dalam relasi ini, terdapat *superclass* dan *subclass* dimana *subclass* akan mewarisi atau memiliki atribut dan *method/operasi* dasar dari *superclass*.

Dalam class diagram, suatu class dapat mewakili banyak objek. Dalam diagram, hal ini ditunjukkan dengan kardinalitas. Kardinalitas dapat diterapkan dengan menggunakan relasi asosiasi dengan menampilkan jumlah kardinalitasnya. Terdapat tiga jenis kardinalitas yaitu one to one (1:1), one to many (1 :) dan many to many (\*:\*). Jika class A dan B memiliki kardinalitas one to one, maka class A hanya boleh berhubungan dengan satu objek B. Jika class A dan B memiliki kardinalitas one to many, maka setiap anggota A dapat berhubungan dengan banyak objek B namun tidak sebaliknya. Sedangkan jika class A dan B memiliki kardinalitas many to many, maka setiap anggota A dapat berhubungan dengan banyak objek B dan juga sebaliknya.

## BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan tentang langkah-langkah yang digunakan dalam penelitian. Metodologi penelitian ini terdiri dari enam tahap. Runtutan pengerjaan penelitian dapat dilihat pada diagram alir berikut.



**Gambar 3.1 Diagram alir penelitian**

### 3.1 Studi Literatur

Studi literature berisi dasar teori yang digunakan sebagai sumber acuan untuk penulisan skripsi. Studi literatur dilakukan dengan membaca buku, jurnal, paper dan artikel-artikel di internet. Teori dan pustaka yang berkaitan dengan tugas akhir ini meliputi :

1. Ujian
  - Jenis Soal Ujian
  - *Security Ujian Online*
2. Arsitektur Client Server
  - Web Service
  - REST
  - RPC
  - JSON
3. CodeIgniter
4. Algoritma Pengacakan Fisher-Yates
5. UML
  - Use Case Diagram
  - Activity Diagram
  - Sequence Diagram
  - Class Diagram

### 3.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk memperoleh kebutuhan-kebutuhan dari aplikasi yang akan dibangun. Tahapan ini dilakukan dengan mengidentifikasi semua kebutuhan (*requirement*) sistem dan semua stakeholder yang terlibat di dalamnya. Semua kebutuhan yang diperoleh dari proses ini harus dapat diakomodasi oleh sistem.

Proses analisis kebutuhan dilakukan dengan cara studi literatur terhadap *software requirement patterns* untuk sistem ujian online. Kemudian tahapan dilanjutkan dengan menjabarkan daftar kebutuhan fungsional dan non fungsional dari *software requirement patterns* yang harus ada pada sistem ujian online terpadu. Tahapan terakhir adalah memodelkan daftar kebutuhan pada sistem ujian online menjadi sebuah diagram *use case*.

### 3.3 Perancangan Sistem

Setelah semua kebutuhan aplikasi diperoleh dari proses analisis kebutuhan, tahapan perancangan sistem dapat dilakukan. Sistem ujian online terpadu dirancang menggunakan *Unified Modeling Language* atau UML. Perancangan sistem dilakukan dengan merancang alur atau aktivitas yang dilakukan pengguna yang kemudian dimodelkan dalam bentuk *class diagram*, *activity diagram*. Setelah semua alur telah dimodelkan, tahapan selanjutnya adalah merancang antarmuka dari sistem ujian online terpadu.

### 3.4 Implementasi

Tahapan ini adalah proses merealisasikan model yang telah dirancang pada proses sebelumnya, menjadi sebuah aplikasi yang dapat memenuhi kebutuhan atau *requirement* dari aplikasi. Proses implementasi dimulai dengan menjabarkan spesifikasi *environment* dari sistem yang akan dibangun. Spesifikasi ini terdiri dari spesifikasi *hardware* dan spesifikasi *software* yang dibutuhkan agar aplikasi backend sistem ujian online terpadu dapat berjalan dengan normal.

Kemudian tahapan dilanjutkan ke pembuatan *database* berdasarkan rancangan yang telah dibuat pada proses perancangan sistem. Implementasi proses bisnis sistem ujian online dilakukan setelah proses implementasi *database*. Tahapan terakhir dari proses implementasi adalah implementasi *user interface* berdasarkan perancangan sebelumnya. Aplikasi backend sistem ujian online ini diimplementasikan menggunakan IDE phpDesigner 8.

### 3.5 Pengujian dan Analisis

Pengujian dilakukan untuk menguji apakah perangkat lunak telah mampu bekerja sesuai dengan spesifikasi kebutuhan yang telah dijabarkan sebelumnya. Pada penelitian ini, proses pengujian dilakukan dengan melakukan pengujian validasi. Pengujian validasi menggunakan metode *blackbox* dalam penerapannya. Metode *blackbox* ini menekankan pada kebutuhan fungsional dari sistem. Dengan menggunakan metode *blackbox*, dapat diketahui sistem telah mampu memenuhi *requirement* yang telah dispesifikasikan atau tidak. Setelah itu dilakukan analisis terhadap sistem untuk mengetahui hasil dari pengujian.

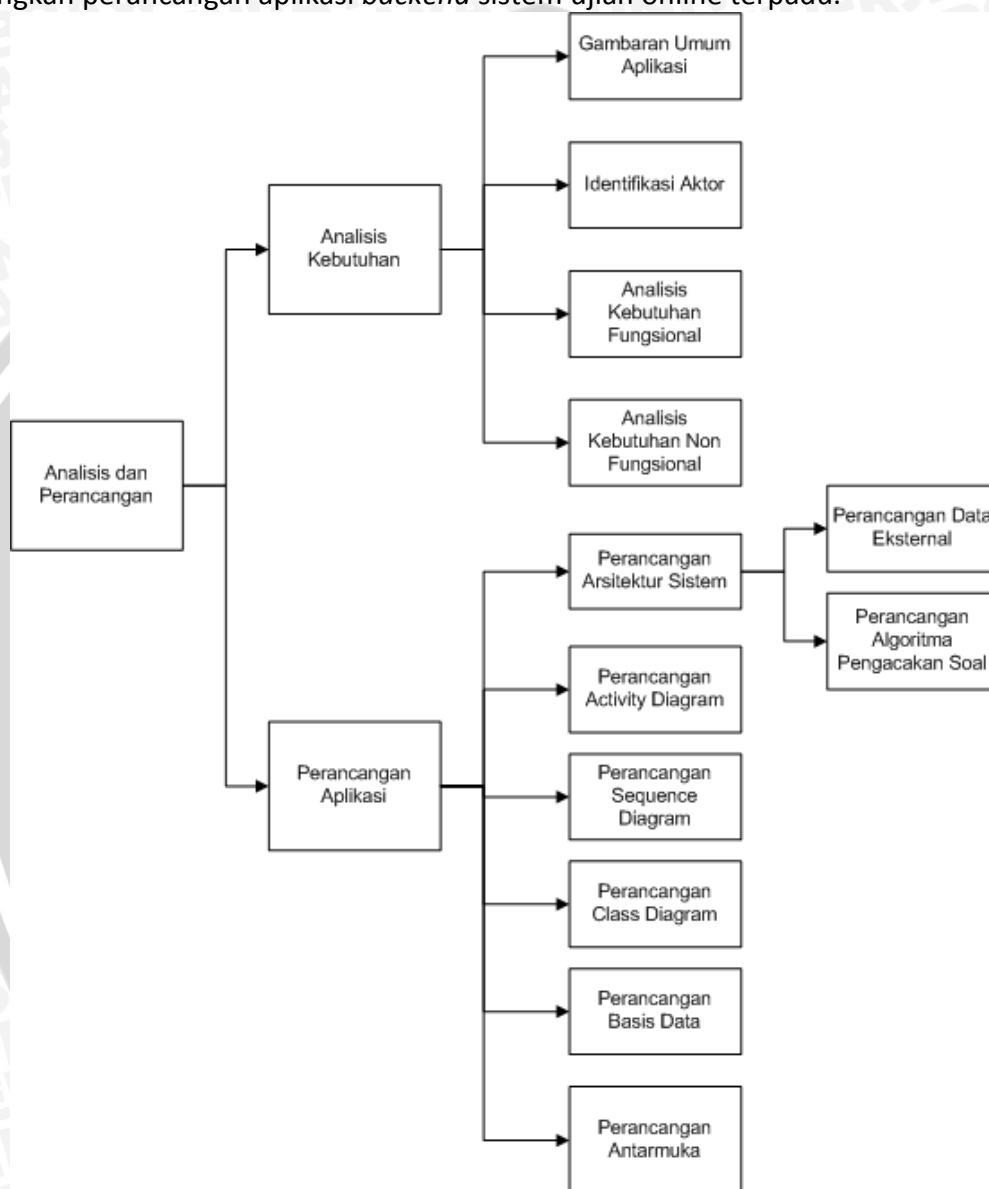
### 3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah semua tahapan metodologi penelitian telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis dari aplikasi dan dijabarkan apakah sistem dapat menyelesaikan permasalahan-permasalahan yang telah dirumuskan.

Penulisan saran bertujuan agar kekurangan-kekurangan dari sistem dapat diperbaiki dan memberikan pertimbangan untuk penelitian selanjutnya.

## BAB 4 ANALISIS DAN PERANCANGAN

Bab ini membahas perancangan dan implementasi aplikasi *backend* sistem ujian online terpadu. Tahap perancangan dibagi menjadi dua tahap, yaitu analisis kebutuhan dan perancangan aplikasi. Gambar 4.1 merupakan langkah-langkah perancangan aplikasi *backend* sistem ujian online terpadu.



Gambar 4.1 Diagram pohon analisis dan perancangan

### 4.1 Analisis Kebutuhan

Tahap analisis kebutuhan ini adalah tahapan untuk mengidentifikasi seluruh kebutuhan aplikasi *backend* sistem ujian online terpadu. Analisis kebutuhan dibagi menjadi empat tahap, yaitu gambaran umum aplikasi, identifikasi aktor yang terlibat dalam sistem, analisis kebutuhan fungsional dan analisis kebutuhan non fungsional.

#### 4.1.1 Gambaran Umum Aplikasi

Aplikasi *backend* sistem ujian online terpadu merupakan aplikasi yang bertujuan untuk membantu instansi ataupun organisasi untuk melaksanakan ujian online. Aplikasi ini terdiri dari website dan *web service*. Website digunakan oleh Administrator untuk melakukan manajemen ujian dan soal-soal. Sedangkan *web service* berfungsi untuk menyediakan layanan kepada aplikasi *client*.

Fitur-fitur yang difokuskan pada aplikasi ini adalah pengacakan soal, manajemen soal dan koreksi jawaban. Dengan adanya pengacakan soal, diharapkan tingkat kecurangan dalam ujian online dapat dikurangi. Koreksi jawaban yang terpadu bertujuan agar sistem dapat dengan otomatis memeriksa jawaban peserta ujian tanpa harus melibatkan manusia untuk melakukan koreksi jawaban. Selain itu, fitur manajemen soal memberikan hak kepada Administrator untuk membuat soal, menghapus dan mengubah soal dengan mudah karena disertai dengan antarmuka yang nyaman bagi pengguna.

Secara garis besar, aplikasi *backend* sistem ujian online terpadu menangani penyimpanan data ke dalam *database* dan pengiriman layanan yang dibutuhkan oleh *client*. Layanan ini berupa permintaan *client* untuk mendapatkan, mengubah, mengirim *resource* seperti data soal-soal dan lain sebagainya.

#### 4.1.2 Identifikasi Aktor

Tahap ini merupakan proses untuk identifikasi aktor-aktor yang berinteraksi dengan aplikasi. Tabel 4.1 merupakan tabel aktor yang berinteraksi dengan aplikasi.

**Tabel 4.1** Tabel Aktor dari Aplikasi *Backend* Sistem Ujian Online Terpadu

| Aktor         | Deskripsi                                                                                                                                                                                                                                                                                                                              |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Peserta ujian | Peserta ujian adalah pengguna yang berinteraksi dengan aplikasi untuk melakukan ujian online. Peserta ujian menggunakan aplikasi <i>client</i> berupa aplikasi pada perangkat bergerak yang melakukan interaksi dengan aplikasi <i>backend</i> sistem ujian online terpadu dengan melakukan <i>request</i> data ataupun mengirim data. |
| Administrator | Administrator adalah pengguna yang menyelenggarakan ujian online. Administrator dapat mengadakan ujian, melakukan manajemen soal, mengundang peserta ujian untuk melakukan ujian, dan lain-lain. Administrator menggunakan browser untuk mengakses aplikasi <i>backend</i> sistem ujian online terpadu.                                |

#### 4.1.3 Analisis Kebutuhan Fungsional

Tahap ini merupakan identifikasi kebutuhan fungsional dari aplikasi. Kebutuhan-kebutuhan dari sistem harus dirancang dengan baik agar aplikasi

dapat memenuhi kebutuhan pengguna. Analisis kebutuhan fungsional dari aplikasi ini dibagi menjadi dua berdasarkan aktor yang terlibat, yaitu analisis kebutuhan fungsional Administrator dan analisis kebutuhan fungsional peserta ujian. Kebutuhan Fungsional sendiri adalah layanan yang harus disediakan sistem, bagaimana sistem harus bereaksi terhadap input tertentu dan bagaimana sistem harus bereaksi dalam kondisi tertentu (Sommerville, 2004).

#### 4.1.3.1 Analisis Kebutuhan Fungsional Berdasarkan Pola Kebutuhan Perangkat Lunak

Penelitian ini menggunakan pola kebutuhan perangkat lunak sebagai alat bantu dalam proses rancang bangun Aplikasi *Backend* Sistem Ujian Online Terpadu. Oleh karena itu, perlu dilakukan validasi kebutuhan perangkat lunak yang ada pada sistem dengan pola kebutuhan perangkat lunak yang dijadikan acuan. Tabel 4.4 adalah tabel validasi pola kebutuhan perangkat lunak terhadap kebutuhan perangkat lunak yang ada pada sistem yang akan dibangun.

**Tabel 4.4 Tabel Pola Kebutuhan Perangkat Lunak**

| Nama Pola Kebutuhan Perangkat Lunak       | Daftar Kebutuhan yang Ada Pada Sistem                                                                                               |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <i>User Registration</i>                  | Registrasi Peserta dan Administrator.                                                                                               |
| <i>User Authentication</i>                | Login Peserta dan Administrator.                                                                                                    |
| <i>User Authorization</i>                 | Aktivitas yang dapat dilakukan oleh peserta dan Administrator dibatasi sesuai dengan haknya.                                        |
| <i>System Operation</i>                   | Sistem menyediakan layanan bagi aplikasi <i>client</i> dari berbagai macam platform dan bahasa pemrograman.                         |
| <i>Information Storage</i>                | Seluruh data dari sistem ujian online disimpan dalam DBMS MySQL.                                                                    |
| <i>Information Content Representation</i> | Pengiriman layanan dari aplikasi <i>backend</i> berbentuk JSON.                                                                     |
| <i>Formula Calculation</i>                | Perhitungan skor ujian.                                                                                                             |
| <i>Navigation</i>                         | Navigasi antar website dilakukan dengan melakukan klik pada menu dan tombol yang tersedia.                                          |
| <i>Data History</i>                       | Data yang tidak aktif seperti ujian yang telah dilakukan akan dipindah pada penyimpanan tertentu.                                   |
| <i>Entity Definition</i>                  | Data yang digunakan dalam sistem harus dapat dijadikan acuan untuk melakukan aktivitas. Seperti login, request soal, dan lain-lain. |
| <i>Report</i>                             | Laporan hasil ujian bagi peserta dan bagi                                                                                           |

|                           |                                                                                    |
|---------------------------|------------------------------------------------------------------------------------|
|                           | Administrator.                                                                     |
| <i>Error Notification</i> | Pesan error dikirimkan ketika peserta melakukan <i>request</i> yang tidak valid.   |
| <i>Approval</i>           | Pendaftaran ujian oleh peserta harus disetujui terlebih dahulu oleh Administrator. |

Dari Tabel 4.4, maka dapat didapatkan kebutuhan-kebutuhan fungsional yang dibagi berdasarkan aktor yang kemudian akan dijabarkan dan dimodelkan dalam diagram *use case*.

#### 4.1.3.2 Analisis Kebutuhan Fungsional *Administrator*

Tabel 4.5 merupakan daftar kebutuhan fungsional dari aplikasi *backend* sistem ujian online terpadu ditinjau dari kebutuhan Administrator.

**Tabel 4.5 Daftar kebutuhan fungsional Administrator**

| Nomor SRS | Kebutuhan                                                                                                                              | Use Case                 |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| SRS_1_01  | Aplikasi harus menyediakan fasilitas login bagi Administrator                                                                          | Login_Administrator      |
| SRS_1_02  | Aplikasi harus menyediakan fasilitas untuk pendaftaran Administrator                                                                   | Registrasi_Administrator |
| SRS_1_03  | Aplikasi harus menyediakan fitur untuk melihat daftar ujian yang telah dibuat oleh Administrator                                       | Lihat_daftar_ujian       |
| SRS_1_04  | Aplikasi harus menyediakan fitur untuk membuat ujian baru. Ujian harus memiliki pilihan apakah menggunakan pengacakan soal atau tidak. | Buat_ujian               |
| SRS_1_05  | Aplikasi harus menyediakan fitur untuk menghapus ujian                                                                                 | Hapus_ujian              |
| SRS_1_06  | Aplikasi harus menyediakan fitur untuk mengubah ujian                                                                                  | Edit_ujian               |
| SRS_1_07  | Aplikasi harus menyediakan fitur untuk menampilkan detail ujian                                                                        | Lihat_detail_ujian       |
| SRS_1_08  | Aplikasi harus menyediakan fitur untuk menambahkan                                                                                     | Tambah_peserta_ujian     |

|          |                                                                                                            |                           |
|----------|------------------------------------------------------------------------------------------------------------|---------------------------|
|          | peserta ke dalam ujian                                                                                     |                           |
| SRS_1_09 | Aplikasi harus menyediakan fitur untuk menampilkan hasil ujian                                             | Tampilkan_hasil_ujian     |
| SRS_1_10 | Aplikasi harus menyediakan fitur untuk melihat daftar soal dalam suatu ujian                               | Lihat_daftar_soal         |
| SRS_1_11 | Aplikasi harus menyediakan fitur untuk membuat soal ujian.                                                 | Buat_soal                 |
| SRS_1_12 | Aplikasi harus menyediakan fitur untuk menghapus soal ujian                                                | Hapus_soal                |
| SRS_1_13 | Aplikasi harus menyediakan fitur untuk mengubah soal ujian                                                 | Edit_soal                 |
| SRS_1_14 | Aplikasi harus menyediakan fitur untuk menampilkan profil peserta ujian                                    | Tampilkan_profil_pengguna |
| SRS_1_15 | Aplikasi harus menyediakan fitur untuk menampilkan daftar request peserta yang mendaftar untuk suatu ujian | Lihat_request_ujian       |
| SRS_1_16 | Aplikasi harus menyediakan fitur untuk menyetujui request pendaftaran peserta terhadap suatu ujian         | Approve_request_ujian     |
| SRS_1_17 | Aplikasi harus menyediakan fitur untuk menolak request pendaftaran peserta terhadap suatu ujian            | Reject_request_ujian      |

#### 4.1.3.3 Analisis Kebutuhan Fungsional Peserta Ujian

Tabel 4.3 merupakan daftar kebutuhan fungsional dari aplikasi *backend* sistem ujian online terpadu ditinjau dari kebutuhan Administrator.

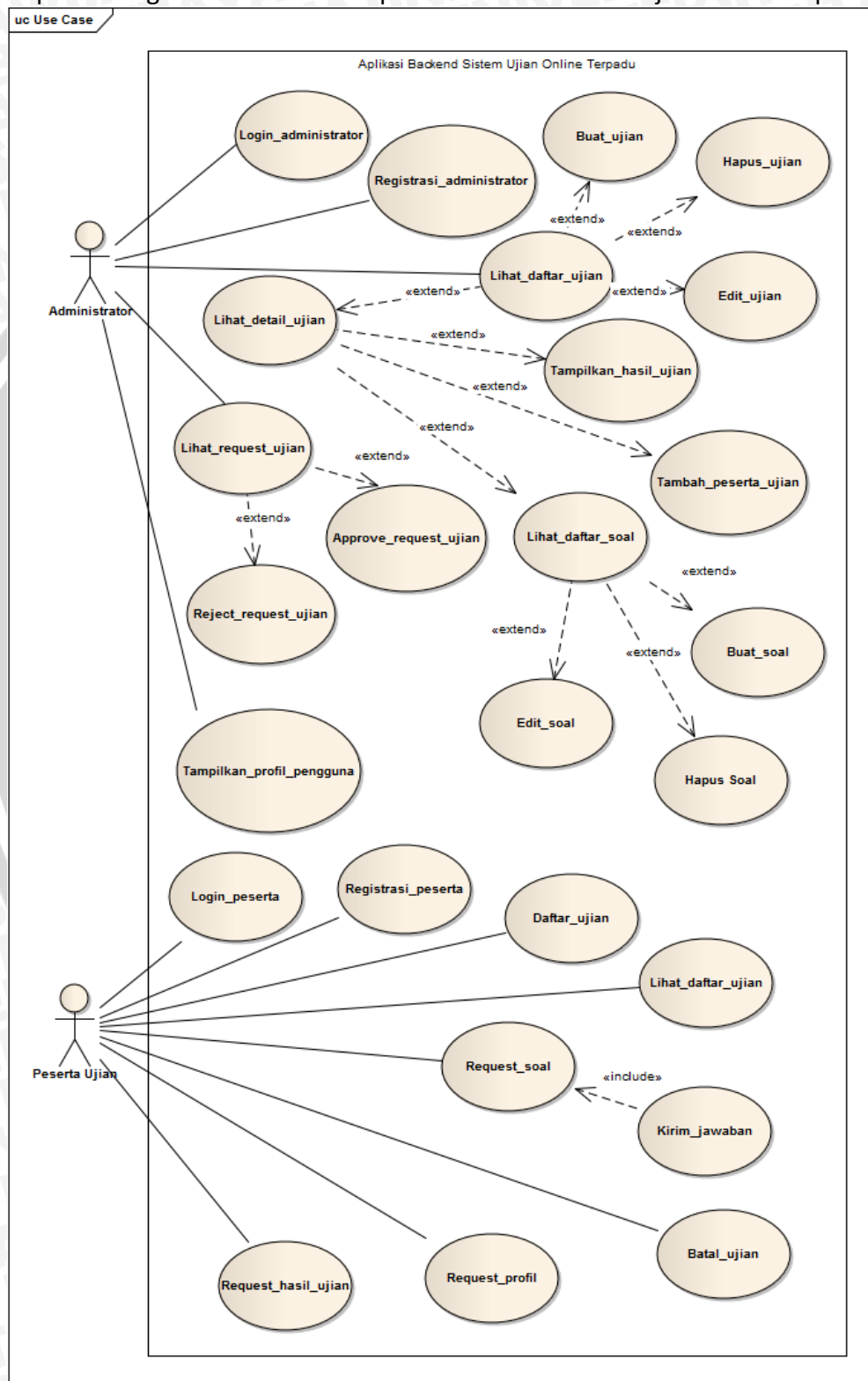
**Tabel 4.3 Daftar kebutuhan fungsional peserta ujian**

| Nomor SRS | Kebutuhan                                               | Use Case      |
|-----------|---------------------------------------------------------|---------------|
| SRS_2_01  | Aplikasi harus menyediakan fasilitas login bagi peserta | Login_peserta |

|          |                                                                                                                                                                                                                                                         |                            |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
|          | ujian                                                                                                                                                                                                                                                   |                            |
| SRS_2_02 | Aplikasi harus menyediakan fasilitas bagi peserta ujian untuk mendaftar ke dalam sistem                                                                                                                                                                 | Registrasi_peserta         |
| SRS_2_03 | Aplikasi harus menyediakan fitur untuk mendaftar ke dalam suatu ujian atau <i>enroll</i> ke dalam ujian.                                                                                                                                                | Daftar_ujian               |
| SRS_2_04 | Aplikasi harus menyediakan fitur untuk mengirimkan data ujian yang telah dilakukan maupun belum dilakukan. Data ujian yang ditampilkan hanya data ujian yang mana peserta telah resmi menjadi peserta ujiannya atau telah disetujui oleh Administrator. | Lihat_daftar_ujian_peserta |
| SRS_2_05 | Aplikasi harus menyediakan fitur untuk mengirim soal ke client. Soal yang dikirimkan bisa dalam kondisi teracak atau tidak tergantung pada pengaturan ujian.                                                                                            | Request_soal               |
| SRS_2_06 | Aplikasi harus menyediakan fitur untuk menerima jawaban dari client sesuai dengan jumlah soal dan skema soal yang telah dikirimkan sebelumnya.                                                                                                          | Kirim_jawaban              |
| SRS_2_07 | Aplikasi harus menyediakan fitur untuk mengirimkan data hasil ujian peserta                                                                                                                                                                             | Request_hasil_ujian        |
| SRS_2_08 | Aplikasi harus menyediakan fitur bagi peserta ujian untuk mengundurkan diri dari ujian yang belum berlangsung.                                                                                                                                          | Batal_ujian                |
| SRS_2_09 | Aplikasi harus menyediakan fitur untuk mengirimkan data peserta ujian.                                                                                                                                                                                  | Request_profil             |

#### 4.1.3.4 Diagram Use Case

Diagram use case adalah diagram yang menggambarkan kebutuhan dari aplikasi dan aktor yang berinteraksi dengan aplikasi. Tiap kebutuhan dari aplikasi diwakili oleh satu *use case* yang berinteraksi dengan aktor tertentu. Gambar 4.2 merupakan diagram use case dari aplikasi *backend* sistem ujian online terpadu.



Gambar 4.2 Use Case Diagram

#### 4.1.3.5 Skenario Use Case

Skenario *use case* berisi penjelasan rinci dari tiap-tiap *use case* yang terdapat pada Gambar 4.2. Pada skenario *use case*, dijabarkan tentang nama *use case*, aktor yang terlibat, tujuan, deskripsi, langkah-langkah *use case*, kondisi awal yang harus dipenuhi (*pre-condition*) dan kondisi akhir yang diharapkan (*post condition*).

Tabel 4.4 Skenario Use Case Login Administrator

| Nama Use Case                                                                                                  | Login_Administrator                                                                                                            |
|----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| Nomor Use Case                                                                                                 | SRS_1_01                                                                                                                       |
| Tujuan                                                                                                         | Masuk ke dalam aplikasi                                                                                                        |
| Deskripsi                                                                                                      | Use case ini menjelaskan bagaimana Administrator dapat masuk ke dalam sistem menggunakan <i>username</i> dan <i>password</i> . |
| Aktor                                                                                                          | Administrator                                                                                                                  |
| Pre-Condition                                                                                                  | Administrator membuka browser dan mengakses halaman utama dari aplikasi                                                        |
| Post-Condition                                                                                                 | Administrator masuk ke dalam sistem dan halaman utama Administrator ditampilkan                                                |
| Skenario Utama                                                                                                 |                                                                                                                                |
| Aksi dari Aktor                                                                                                | Respon dari Sistem                                                                                                             |
| 6. User mengisi <i>username</i> dan <i>password</i> pada kolom yang telah disediakan dan menekan tombol login. | 7. Aplikasi melakukan validasi <i>username</i> dan <i>password</i> di database.                                                |
|                                                                                                                | 8. Aplikasi membawa Administrator ke halaman utama Administrator.                                                              |
| Skenario Alternatif 1 : Jika ada salah satu kolom yang belum diisi                                             |                                                                                                                                |
|                                                                                                                | 9. Aplikasi menampilkan pesan untuk mengisi kolom yang kosong                                                                  |
| Skenario Alternatif 2 : Jika <i>username</i> atau <i>password</i> salah                                        |                                                                                                                                |

|  |                                                                                              |
|--|----------------------------------------------------------------------------------------------|
|  | 10. Aplikasi menampilkan pesan <i>error</i> bahwa <i>username</i> atau <i>password</i> salah |
|--|----------------------------------------------------------------------------------------------|

Tabel 4.4 Skenario *Use Case* Registrasi\_Administrator

| Nama <i>Use Case</i>                                                                                     | Registrasi_Administrator                                                                                         |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Nomor <i>Use Case</i>                                                                                    | SRS_1_02                                                                                                         |
| Tujuan                                                                                                   | Mendaftar untuk mendapatkan akun dari sistem                                                                     |
| Deskripsi                                                                                                | <i>Use case</i> ini menjelaskan bagaimana Administrator yang belum memiliki akun untuk mendaftar ke dalam sistem |
| Aktor                                                                                                    | Administrator                                                                                                    |
| <i>Pre-Condition</i>                                                                                     | Administrator membuka browser dan mengakses halaman utama dari aplikasi                                          |
| <i>Post-Condition</i>                                                                                    | Administrator mendapatkan akun yang dapat login ke dalam sistem                                                  |
| Skenario Utama                                                                                           |                                                                                                                  |
| Aksi dari Aktor                                                                                          | Respon dari Sistem                                                                                               |
| 1. Administrator menekan tombol registrasi.                                                              | 2. Menampilkan halaman registrasi .                                                                              |
| 3. Mengisi kolom <i>username</i> , nama, email dan <i>password</i> dan kemudian menekan tombol register. | 4. Memberikan konfirmasi bahwa akun telah sukses dibuat dan membawa user ke halaman login Administrator.         |
| Skenario Alternatif 1 : Jika ada salah satu kolom yang belum diisi                                       |                                                                                                                  |
|                                                                                                          | 5. Menampilkan pesan untuk mengisi kolom yang belum terisi.                                                      |
| Skenario Alternatif 2 : Jika terdapat pengisian kolom yang salah                                         |                                                                                                                  |
|                                                                                                          | 1. Menampilkan pesan error bahwa kolom yang bersangkutan tidak terisi dengan benar.                              |

Tabel 4.5 Skenario *Use Case* Lihat\_daftar\_ujian

| Nama <i>Use Case</i> | Lihat_daftar_ujian |
|----------------------|--------------------|
|----------------------|--------------------|

|                                       |                                                                                                                                                                 |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nomor Use Case</b>                 | SRS_1_03                                                                                                                                                        |
| <b>Tujuan</b>                         | Melihat daftar ujian yang telah dibuat oleh Administrator                                                                                                       |
| <b>Deskripsi</b>                      | <i>Use case</i> ini menjelaskan bagaimana Administrator melihat daftar ujian yang telah dibuatnya. Baik yang telah dilaksanakan maupun yang belum dilaksanakan. |
| <b>Aktor</b>                          | Administrator                                                                                                                                                   |
| <b>Pre-Condition</b>                  | Administrator berhasil login ke dalam sistem                                                                                                                    |
| <b>Post-Condition</b>                 | Sistem menampilkan daftar ujian                                                                                                                                 |
| <b>Skenario Utama</b>                 |                                                                                                                                                                 |
| <b>Aksi dari Aktor</b>                | <b>Respon dari Sistem</b>                                                                                                                                       |
| 1. Menekan tombol lihat daftar ujian. | 2. Menampilkan daftar ujian yang telah dibuat oleh Administrator.                                                                                               |

**Tabel 4.6 Skenario Use Case Buat\_ujian**

|                                                                                                                      |                                                                                                                     |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>                                                                                                 | <b>Lihat_daftar_ujian</b>                                                                                           |
| <b>Nomor Use Case</b>                                                                                                | SRS_1_04                                                                                                            |
| <b>Tujuan</b>                                                                                                        | Membuat ujian baru                                                                                                  |
| <b>Deskripsi</b>                                                                                                     | <i>Use case</i> ini menjelaskan bagaimana Administrator membuat ujian baru                                          |
| <b>Aktor</b>                                                                                                         | Administrator                                                                                                       |
| <b>Pre-Condition</b>                                                                                                 | Administrator login ke dalam sistem dan berada pada halaman lihat daftar ujian                                      |
| <b>Post-Condition</b>                                                                                                | Sistem menampilkan daftar ujian                                                                                     |
| <b>Skenario Utama</b>                                                                                                |                                                                                                                     |
| <b>Aksi dari Aktor</b>                                                                                               | <b>Respon dari Sistem</b>                                                                                           |
| 1. Menekan tombol tambah ujian.                                                                                      | 2. Menampilkan form untuk pembuatan ujian baru.                                                                     |
| 3. Mengisi kolom yang tersedia. Seperti kolom nama ujian, deskripsi, durasi pengerjaan, tanggal pelaksanaan, tanggal | 4. Ujian berhasil dibuat dan disimpan dalam basis data, kemudian Administrator dibawa ke halaman lihat daftar soal. |

|                                                                                                            |                                                                                    |
|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| pembuatan, deskripsi, passing grade, enrollment key, dll. Kemudian Administrator menekan tombol buat ujian |                                                                                    |
| <b>Skenario Alternatif 1 : Terdapat kolom yang belum terisi</b>                                            |                                                                                    |
|                                                                                                            | 5. Menampilkan pesan bahwa kolom belum terisi                                      |
| <b>Skenario Alternatif 2 : Terdapat kolom pengisian yang salah</b>                                         |                                                                                    |
|                                                                                                            | 6. Menampilkan pesan error bahwa kolom yang bersangkutan tidak terisi dengan benar |

**Tabel 4.7 Skenario Use Case Hapus\_ujian**

| Nama Use Case                                           | Hapus_ujian                                                                                                                                                        |
|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Use Case                                          | SRS_1_05                                                                                                                                                           |
| Tujuan                                                  | Menghapus ujian                                                                                                                                                    |
| Deskripsi                                               | Use case ini menjelaskan bagaimana Administrator menghapus ujian yang telah dibuat.                                                                                |
| Aktor                                                   | Administrator                                                                                                                                                      |
| Pre-Condition                                           | Administrator login ke dalam sistem dan berada pada halaman lihat daftar ujian                                                                                     |
| Post-Condition                                          | Ujian berhasil dihapus                                                                                                                                             |
| <b>Skenario Utama</b>                                   |                                                                                                                                                                    |
| Aksi dari Aktor                                         | Respon dari Sistem                                                                                                                                                 |
| 1. Menghapus tombol ujian pada ujian yang ingin dihapus | 2. Menghapus ujian dari <i>database</i> dan memberikan konfirmasi bahwa ujian telah berhasil dihapus. Administrator kemudian dibawa ke halaman lihat daftar ujian. |

**Tabel 4.8 Skenario Use Case Edit Ujian**

| Nama Use Case  | Edit_ujian                         |
|----------------|------------------------------------|
| Nomor Use Case | SRS_1_06                           |
| Tujuan         | Mengubah attribute dari ujian yang |

|                                                                        |                                                                                                                                                              |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                        | telah dibuat.                                                                                                                                                |
| <b>Deskripsi</b>                                                       | <i>Use case</i> ini menjelaskan bagaimana Administrator mengubah atribut ujian yang telah dibuat. Seperti mengubah nama, tanggal pelaksanaan, dan lain-lain. |
| <b>Aktor</b>                                                           | Administrator                                                                                                                                                |
| <b>Pre-Condition</b>                                                   | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat daftar ujian                                                                      |
| <b>Post-Condition</b>                                                  | Sistem mengupdate ujian sesuai dengan perubahan yang telah dibuat oleh Administrator.                                                                        |
| <b>Skenario Utama</b>                                                  |                                                                                                                                                              |
| <b>Aksi dari Aktor</b>                                                 | <b>Respon dari Sistem</b>                                                                                                                                    |
| 1. Menekan tombol edit ujian pada ujian yang ingin diubah.             | 2. Menampilkan atribut dari ujian dalam bentuk form.                                                                                                         |
| 3. Mengubah atribut yang ingin diubah. Kemudian menekan tombol simpan. | 4. Menyimpan perubahan atribut ujian ke dalam <i>database</i> .                                                                                              |
| <b>Skenario Alternatif 1 : Perubahan kolom tidak benar</b>             |                                                                                                                                                              |
|                                                                        | 5. Menampilkan pesan <i>error</i> bahwa perubahan yang dilakukan pada kolom yang bersangkutan tidak benar.                                                   |

**Tabel 4.9 Skenario *Use Case* Lihat\_detail\_ujian**

|                              |                                                                                                                                                              |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nama <i>Use Case</i></b>  | <b>Lihat_detail_ujian</b>                                                                                                                                    |
| <b>Nomor <i>Use Case</i></b> | SRS_1_07                                                                                                                                                     |
| <b>Tujuan</b>                | Melihat detail dari suatu ujian                                                                                                                              |
| <b>Deskripsi</b>             | <i>Use case</i> ini menjelaskan bagaimana Administrator melihat detail ujian yang telah dibuat. Baik yang telah dilaksanakan maupun yang belum dilaksanakan. |
| <b>Aktor</b>                 | Administrator                                                                                                                                                |
| <b>Pre-Condition</b>         | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat                                                                                   |

|                                                                     |                                                                                                                                                      |
|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                     | daftar ujian                                                                                                                                         |
| <b>Post-Condition</b>                                               | Sistem menampilkan detail ujian yang berisi informasi tentang ujian dan daftar peserta yang telah terdaftar ke dalam ujian tersebut.                 |
| <b>Skenario Utama</b>                                               |                                                                                                                                                      |
| <b>Aksi dari Aktor</b>                                              | <b>Respon dari Sistem</b>                                                                                                                            |
| 1. Menekan tombol lihat detail ujian pada ujian yang ingin dilihat. | 2. Membawa Administrator ke halaman detail ujian yang berisi informasi tentang ujian dan daftar peserta yang telah terdaftar ke dalam ujian tersebut |

**Tabel 4.10 Skenario Use Case Tambah\_peserta\_ujian**

|                                                                                                    |                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>                                                                               | <b>Tambah_peserta_ujian</b>                                                                                                                                      |
| <b>Nomor Use Case</b>                                                                              | SRS_1_08                                                                                                                                                         |
| <b>Tujuan</b>                                                                                      | Menambahkan peserta ke dalam suatu ujian yang belum dilaksanakan.                                                                                                |
| <b>Deskripsi</b>                                                                                   | Use case ini menjelaskan bagaimana Administrator menambahkan peserta ujian ke dalam ujian yang belum dilaksanakan.                                               |
| <b>Aktor</b>                                                                                       | Administrator                                                                                                                                                    |
| <b>Pre-Condition</b>                                                                               | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat detail ujian                                                                          |
| <b>Post-Condition</b>                                                                              | Peserta berhasil ditambahkan ke dalam ujian.                                                                                                                     |
| <b>Skenario Utama</b>                                                                              |                                                                                                                                                                  |
| <b>Aksi dari Aktor</b>                                                                             | <b>Respon dari Sistem</b>                                                                                                                                        |
| 1. Menekan tombol tambah peserta ujian                                                             | 2. Membawa Administrator ke halaman yang berisi form untuk menambahkan peserta. Terdapat kolom email yang berfungsi untuk menambahkan peserta berdasarkan email. |
| 3. Mengisi kolom username dengan username dari satu ataupun lebih peserta dengan pemisah koma (,). | 4. Memasukkan peserta ke dalam ujian dengan email yang telah dimasukkan oleh Administrator.                                                                      |

|                                                                                                |                                                                                                                   |
|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Kemudian Administrator menekan tombol tambah.                                                  | Setelah ditambahkan, peserta terdaftar sebagai peserta ujian yang sah.                                            |
| <b>Skenario Alternatif 1 : Terdapat kesalahan dalam pengisian kolom username peserta ujian</b> |                                                                                                                   |
|                                                                                                | 5. Menampilkan pesan bahwa telah terjadi kesalahan. Misalkan pemisah antar email menggunakan karakter selain (,). |

**Tabel 4.11 Skenario Use Case Tampilkan\_hasil\_ujian**

| Nama Use Case                                           | Tampilkan_hasil_ujian                                                                                                                      |
|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Use Case                                          | SRS_1_09                                                                                                                                   |
| Tujuan                                                  | Menampilkan hasil ujian yang telah dilaksanakan.                                                                                           |
| Deskripsi                                               | Use case ini menjelaskan bagaimana Administrator menampilkan hasil ujian yang telah dilaksanakan.                                          |
| Aktor                                                   | Administrator                                                                                                                              |
| Pre-Condition                                           | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat detail ujian                                                    |
| Post-Condition                                          | Sistem menampilkan hasil ujian yang terdiri dari analisis jawaban benar dan daftar peserta yang lulus dan tidak lulus dalam bentuk grafik. |
| <b>Skenario Utama</b>                                   |                                                                                                                                            |
| Aksi dari Aktor                                         | Respon dari Sistem                                                                                                                         |
| 1. Menekan tombol tampilkan hasil ujian.                | 2. Membawa Administrator ke halaman yang menampilkan analisis jawaban dan daftar peserta yang lulus dan tidak lulus dalam bentuk grafik.   |
| <b>Skenario Alternatif 1 : Ujian belum dilaksanakan</b> |                                                                                                                                            |
|                                                         | 3. Menampilkan pesan bahwa ujian belum dilaksanakan sehingga tidak ada hasil untuk ditampilkan.                                            |

Tabel 4.12 Skenario *Use Case* Lihat\_daftar\_soal

|                                      |                                                                                                             |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>                 | <b>Lihat_daftar_soal</b>                                                                                    |
| <b>Nomor Use Case</b>                | SRS_1_10                                                                                                    |
| <b>Tujuan</b>                        | Menampilkan daftar soal yang berada dalam suatu ujian.                                                      |
| <b>Deskripsi</b>                     | <i>Use case</i> ini menjelaskan bagaimana Administrator menampilkan daftar soal yang digunakan untuk ujian. |
| <b>Aktor</b>                         | Administrator                                                                                               |
| <b>Pre-Condition</b>                 | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat detail ujian.                    |
| <b>Post-Condition</b>                | Sistem menampilkan daftar soal yang ada dalam ujian tersebut.                                               |
| <b>Skenario Utama</b>                |                                                                                                             |
| <b>Aksi dari Aktor</b>               | <b>Respon dari Sistem</b>                                                                                   |
| 1. Menekan tombol lihat daftar soal. | 2. Membawa Administrator ke halaman yang menampilkan daftar soal dalam ujian tersebut.                      |

Tabel 4.13 Skenario *Use Case* Buat\_soal

|                               |                                                                                                          |
|-------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>          | <b>Buat_soal</b>                                                                                         |
| <b>Nomor Use Case</b>         | SRS_1_11                                                                                                 |
| <b>Tujuan</b>                 | Membuat soal baru                                                                                        |
| <b>Deskripsi</b>              | <i>Use case</i> ini menjelaskan bagaimana Administrator membuat soal baru dalam suatu ujian.             |
| <b>Aktor</b>                  | Administrator                                                                                            |
| <b>Pre-Condition</b>          | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat daftar soal dari suatu ujian. |
| <b>Post-Condition</b>         | Sistem berhasil menyimpan soal ke dalam <i>database</i> .                                                |
| <b>Skenario Utama</b>         |                                                                                                          |
| <b>Aksi dari Aktor</b>        | <b>Respon dari Sistem</b>                                                                                |
| 1. Menekan tombol tambah soal | 2. Membawa Administrator ke                                                                              |

|                                                                                                                                                                   |                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
|                                                                                                                                                                   | halaman yang berisi form untuk pembuatan soal baru.                                       |
| 3. Administrator mengisi kolom-kolom pada form. Seperti kolom pertanyaan, level kesulitan, jawaban, dan jenis soal. Kemudian Administrator menekan tombol tambah. | 4. Menyimpan data soal ke dalam database.                                                 |
| <b>Skenario Alternatif 1 : Jika kolom yang tersedia tidak diisi seluruhnya</b>                                                                                    |                                                                                           |
|                                                                                                                                                                   | 5. Menampilkan pesan bahwa terdapat kolom yang belum diisi                                |
| <b>Skenario Alternatif 2 : Jika terdapat kesalahan dalam pengisian kolom</b>                                                                                      |                                                                                           |
|                                                                                                                                                                   | 6. Menampilkan pesan <i>error</i> bahwa kolom yang bersangkutan tidak diisi dengan benar. |

**Tabel 4.14 Skenario Use Case Hapus\_soal**

|                                                            |                                                                                                          |
|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>                                       | <b>Lihat_daftar_soal</b>                                                                                 |
| <b>Nomor Use Case</b>                                      | SRS_1_12                                                                                                 |
| <b>Tujuan</b>                                              | Menghapus soal dalam suatu ujian.                                                                        |
| <b>Deskripsi</b>                                           | Use case ini menjelaskan bagaimana Administrator menghapus soal dalam suatu ujian.                       |
| <b>Aktor</b>                                               | Administrator                                                                                            |
| <b>Pre-Condition</b>                                       | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat daftar soal dari suatu ujian. |
| <b>Post-Condition</b>                                      | Sistem berhasil menghapus soal ujian.                                                                    |
| <b>Skenario Utama</b>                                      |                                                                                                          |
| <b>Aksi dari Aktor</b>                                     | <b>Respon dari Sistem</b>                                                                                |
| 1. Menekan tombol hapus soal pada soal yang ingin dihapus. | 2. Menghapus soal dari <i>database</i> dan membawa user ke halaman tampilkan daftar soal.                |

**Tabel 4.15 Skenario Use Case Edit\_soal**

|                       |                          |
|-----------------------|--------------------------|
| <b>Nama Use Case</b>  | <b>Lihat_daftar_soal</b> |
| <b>Nomor Use Case</b> | SRS_1_13                 |

|                                                                                      |                                                                                                          |
|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>Tujuan</b>                                                                        | Mengubah soal yang telah dibuat sebelumnya.                                                              |
| <b>Deskripsi</b>                                                                     | <i>Use case</i> ini menjelaskan bagaimana Administrator mengubah soal yang telah dibuat.                 |
| <b>Aktor</b>                                                                         | Administrator                                                                                            |
| <b>Pre-Condition</b>                                                                 | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat daftar soal dari suatu ujian. |
| <b>Post-Condition</b>                                                                | Sistem mengubah attribut soal sesuai dengan perubahan yang ditentukan oleh Administrator.                |
| <b>Skenario Utama</b>                                                                |                                                                                                          |
| <b>Aksi dari Aktor</b>                                                               | <b>Respon dari Sistem</b>                                                                                |
| 1. Menekan tombol edit soal pada soal yang ingin diubah.                             | 2. Membawa Administrator ke halaman yang menampilkan form berisi attribut soal.                          |
| 3. Mengubah attribut soal sesuai kebutuhan dan menekan tombol simpan.                | 4. Menyimpan soal yang telah diubah ke dalam <i>database</i> .                                           |
| <b>Skenario Alternatif 1 : Jika terdapat kesalahan pada pengubahan attribut soal</b> |                                                                                                          |
|                                                                                      | 5. Menampilkan pesan <i>error</i> bahwa telah terjadi kesalahan pada pengubahan kolom yang bersangkutan. |

**Tabel 4.16 Skenario Use Case Tampilkan\_profil\_pengguna**

|                       |                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>  | <b>Tampilkan_profil_pengguna</b>                                                                             |
| <b>Nomor Use Case</b> | SRS_1_14                                                                                                     |
| <b>Tujuan</b>         | Menampilkan profil peserta ujian ataupun Administrator.                                                      |
| <b>Deskripsi</b>      | <i>Use case</i> ini menjelaskan bagaimana Administrator menampilkan profil peserta ujian atau Administrator. |
| <b>Aktor</b>          | Administrator                                                                                                |
| <b>Pre-Condition</b>  | Administrator berada pada halaman lihat daftar peserta ujian.                                                |
| <b>Post-Condition</b> | Sistem menampilkan profil peserta                                                                            |

|                                       | ujian ataupun Administrator.                                               |
|---------------------------------------|----------------------------------------------------------------------------|
| Skenario Utama                        |                                                                            |
| Aksi dari Aktor                       | Respon dari Sistem                                                         |
| 1. Melakukan klik pada gambar profil. | 2. Membawa Administrator ke halaman yang menampilkan profil peserta ujian. |

Tabel 4.17 Skenario Use Case Lihat\_request\_ujian

| Nama Use Case                                     | Tampilkan_profil_pengguna                                                                                                                                                                                                          |
|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Use Case                                    | SRS_1_15                                                                                                                                                                                                                           |
| Tujuan                                            | Menampilkan daftar request peserta yang ingin mendaftar ke dalam suatu ujian.                                                                                                                                                      |
| Deskripsi                                         | Use case ini menjelaskan bagaimana Administrator menampilkan daftar request peserta yang ingin mendaftar ke dalam suatu ujian. Request yang dapat dilihat hanyalah request terhadap ujian yang dibuat oleh Administrator tersebut. |
| Aktor                                             | Administrator                                                                                                                                                                                                                      |
| Pre-Condition                                     | Administrator berhasil login ke dalam sistem                                                                                                                                                                                       |
| Post-Condition                                    | Sistem menampilkan daftar request peserta ujian                                                                                                                                                                                    |
| Skenario Utama                                    |                                                                                                                                                                                                                                    |
| Aksi dari Aktor                                   | Respon dari Sistem                                                                                                                                                                                                                 |
| 1. Administrator memilih menu lihat request ujian | 2. Aplikasi mengambil data request ujian dari database                                                                                                                                                                             |
|                                                   | 3. Aplikasi menampilkan halaman yang berisi data request ujian dari database.                                                                                                                                                      |

Tabel 4.18 Skenario Use Case Approve\_request\_ujian

| Nama Use Case  | Tampilkan_profil_pengguna             |
|----------------|---------------------------------------|
| Nomor Use Case | SRS_1_16                              |
| Tujuan         | Menyetujui request peserta yang ingin |

|                                                                          |                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                          | mengikuti suatu ujian.                                                                                                                                                                                                   |
| <b>Deskripsi</b>                                                         | <i>Use case</i> ini menjelaskan bagaimana Administrator menyetujui request peserta yang ingin mengikuti suatu ujian. Request yang dapat dilihat hanyalah request terhadap ujian yang dibuat oleh Administrator tersebut. |
| <b>Aktor</b>                                                             | Administrator                                                                                                                                                                                                            |
| <b>Pre-Condition</b>                                                     | Administrator berhasil login ke dalam sistem dan berada pada halaman lihat request ujian.                                                                                                                                |
| <b>Post-Condition</b>                                                    | Sistem menampilkan daftar request peserta ujian                                                                                                                                                                          |
| <b>Skenario Utama</b>                                                    |                                                                                                                                                                                                                          |
| <b>Aksi dari Aktor</b>                                                   | <b>Respon dari Sistem</b>                                                                                                                                                                                                |
| 1. Administrator menekan tombol approve pada request yang akan disetujui | 2. Aplikasi memasukkan peserta sebagai peserta ujian yang telah disetujui.                                                                                                                                               |
|                                                                          | 3. Aplikasi menampilkan halaman lihat request ujian.                                                                                                                                                                     |

**Tabel 4.19 Skenario *Use Case* Reject\_request\_ujian**

|                              |                                                                                                                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Nama <i>Use Case</i></b>  | <b>Tampilkan_profil_pengguna</b>                                                                                                                                                                                      |
| <b>Nomor <i>Use Case</i></b> | SRS_1_17                                                                                                                                                                                                              |
| <b>Tujuan</b>                | Menolak request peserta yang ingin mengikuti suatu ujian.                                                                                                                                                             |
| <b>Deskripsi</b>             | <i>Use case</i> ini menjelaskan bagaimana Administrator menolak request peserta yang ingin mengikuti suatu ujian. Request yang dapat dilihat hanyalah request terhadap ujian yang dibuat oleh Administrator tersebut. |
| <b>Aktor</b>                 | Administrator                                                                                                                                                                                                         |
| <b>Pre-Condition</b>         | Administrator berada pada halaman lihat request ujian.                                                                                                                                                                |
| <b>Post-Condition</b>        | Sistem menampilkan daftar request peserta ujian                                                                                                                                                                       |

| Skenario Utama                                                          |                                                      |
|-------------------------------------------------------------------------|------------------------------------------------------|
| Aksi dari Aktor                                                         | Respon dari Sistem                                   |
| 1. Administrator menekan tombol Reject pada request yang akan disetujui | 2. Aplikasi menyimpan request menolak                |
|                                                                         | 3. Aplikasi menampilkan halaman lihat request ujian. |

Tabel 4.20 Skenario *Use Case* Login\_peserta

| Nama Use Case                                                                                                                                                                                                              | Login_peserta                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Use Case                                                                                                                                                                                                             | SRS_2_01                                                                                                                                                                                                   |
| Tujuan                                                                                                                                                                                                                     | Untuk masuk ke dalam aplikasi                                                                                                                                                                              |
| Deskripsi                                                                                                                                                                                                                  | Use case ini menjelaskan bagaimana peserta masuk ke dalam sistem.                                                                                                                                          |
| Aktor                                                                                                                                                                                                                      | Peserta                                                                                                                                                                                                    |
| Pre-Condition                                                                                                                                                                                                              | Peserta membuka aplikasi <i>client</i> dan membuka halaman login.                                                                                                                                          |
| Post-Condition                                                                                                                                                                                                             | Peserta masuk ke dalam sistem dan dapat mengakses layanan yang disediakan oleh sistem.                                                                                                                     |
| Skenario Utama                                                                                                                                                                                                             |                                                                                                                                                                                                            |
| Aksi dari Aktor                                                                                                                                                                                                            | Respon dari Sistem                                                                                                                                                                                         |
| 1. Peserta mengisi kolom <i>username</i> dan <i>password</i> kemudian menekan tombol login pada aplikasi <i>client</i> . Aplikasi <i>client</i> kemudian akan melakukan <i>request</i> untuk login ke <i>web service</i> . | 2. Aplikasi memvalidasi <i>username</i> dan <i>password</i> peserta ujian.                                                                                                                                 |
|                                                                                                                                                                                                                            | 3. Aplikasi mengirimkan token yang berfungsi sebagai kunci untuk mendapatkan layanan dari <i>web service</i> disertai data ujian yang telah diikuti oleh peserta. Baik yang sudah terlaksana maupun belum. |
| Skenario Alternatif 1 : Jika ada salah satu kolom yang kosong                                                                                                                                                              |                                                                                                                                                                                                            |
|                                                                                                                                                                                                                            | 4. Aplikasi mengirimkan pesan error bahwa terdapat kolom yang kosong                                                                                                                                       |

| Skenario Alternatif 2 : Jika <i>username</i> atau <i>password</i> salah |                                                                                             |
|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
|                                                                         | 5. Aplikasi mengirimkan pesan error bahwa <i>username</i> atau <i>password</i> tidak valid. |

Tabel 4.21 Skenario *Use Case* Registrasi\_peserta

| Nama <i>Use Case</i>                                                                                                                                              | Registrasi_peserta                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Nomor <i>Use Case</i>                                                                                                                                             | SRS_2_02                                                                                                     |
| Tujuan                                                                                                                                                            | Untuk mendapatkan akun untuk mengakses aplikasi.                                                             |
| Deskripsi                                                                                                                                                         | <i>Use case</i> ini menjelaskan bagaimana peserta mendaftar untuk mendapatkan akun untuk mengakses aplikasi. |
| Aktor                                                                                                                                                             | Peserta                                                                                                      |
| <i>Pre-Condition</i>                                                                                                                                              | Peserta membuka aplikasi <i>client</i> dan membuka halaman registrasi.                                       |
| <i>Post-Condition</i>                                                                                                                                             | Peserta terdaftar ke dalam sistem.                                                                           |
| Skenario Utama                                                                                                                                                    |                                                                                                              |
| Aksi dari Aktor                                                                                                                                                   | Respon dari Sistem                                                                                           |
| 1. Peserta mengakses halaman registrasi, mengisi kolom-kolom yang tersedia ( <i>username</i> , <i>password</i> , nama, email) kemudian menekan tombol registrasi. | 2. Menyimpan data ke dalam <i>database</i> dan mengirimkan konfirmasi bahwa peserta telah terdaftar.         |
| Skenario Alternatif 1 : Jika parameter tidak lengkap                                                                                                              |                                                                                                              |
|                                                                                                                                                                   | 3. Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                     |
| Skenario Alternatif 2 : Jika parameter tidak valid                                                                                                                |                                                                                                              |
|                                                                                                                                                                   | 4. Aplikasi mengirimkan pesan <i>error</i> bahwa telah terjadi kesalahan pada parameter yang dikirimkan      |

Tabel 4.22 Skenario *Use Case* Daftar\_ujian

| Nama <i>Use Case</i>  | Daftar_ujian                          |
|-----------------------|---------------------------------------|
| Nomor <i>Use Case</i> | SRS_2_03                              |
| Tujuan                | Untuk mendaftar ke dalam suatu ujian. |

|                                                                                                                                          |                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Deskripsi</b>                                                                                                                         | <i>Use case</i> ini menjelaskan bagaimana peserta mendaftar ke dalam suatu ujian.                         |
| <b>Aktor</b>                                                                                                                             | Peserta                                                                                                   |
| <b>Pre-Condition</b>                                                                                                                     | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> .                               |
| <b>Post-Condition</b>                                                                                                                    | Peserta terdaftar ke dalam suatu ujian.                                                                   |
| <b>Skenario Utama</b>                                                                                                                    |                                                                                                           |
| <b>Aksi dari Aktor</b>                                                                                                                   | <b>Respon dari Sistem</b>                                                                                 |
| 1. Peserta memasukkan kode <i>enroll</i> dan aplikasi <i>client</i> mengirimkan kode <i>enroll</i> beserta token ke <i>web service</i> . | 2. Aplikasi memvalidasi token peserta.                                                                    |
|                                                                                                                                          | 3. Aplikasi melakukan pencarian kode <i>enroll</i> pada <i>database</i> .                                 |
|                                                                                                                                          | 4. Aplikasi berhasil mendaftarkan peserta ke dalam suatu ujian.                                           |
| <b>Skenario Alternatif 1 : Jika parameter tidak lengkap</b>                                                                              |                                                                                                           |
|                                                                                                                                          | 5. Aplikasi mengirimkan pesan error bahwa parameter tidak lengkap                                         |
| <b>Skenario Alternatif 2 : Jika token tidak valid</b>                                                                                    |                                                                                                           |
|                                                                                                                                          | 6. Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.                                      |
| <b>Skenario Alternatif 3 : Jika kode enroll tidak valid</b>                                                                              |                                                                                                           |
|                                                                                                                                          | 7. Aplikasi mengirimkan pesan <i>error</i> bahwa tidak terdapat ujian dengan kode <i>enroll</i> tersebut. |
| <b>Skenario Alternatif 4 : Jika telah mengirimkan permohonan ujian</b>                                                                   |                                                                                                           |
|                                                                                                                                          | 8. Aplikasi mengirimkan pesan <i>error</i> bahwa permohonan ujian telah dikirimkan                        |

Tabel 4.23 Skenario *Use Case* Lihat\_daftar\_ujian\_peserta

|                       |                                   |
|-----------------------|-----------------------------------|
| <b>Nama Use Case</b>  | <b>Lihat_daftar_ujian_peserta</b> |
| <b>Nomor Use Case</b> | SRS_2_04                          |
| <b>Tujuan</b>         | Untuk mendapatkan daftar ujian.   |

|                                                                                                                                     |                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Deskripsi</b>                                                                                                                    | <i>Use case</i> ini menjelaskan bagaimana peserta mendapatkan daftar ujian yang mana peserta ujian telah terdaftar. Daftar ini termasuk ujian yang telah terlaksana maupun yang belum terlaksana. |
| <b>Aktor</b>                                                                                                                        | Peserta                                                                                                                                                                                           |
| <b>Pre-Condition</b>                                                                                                                | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> .                                                                                                                       |
| <b>Post-Condition</b>                                                                                                               | Peserta mendapatkan daftar seluruh ujian yang telah didaftarkan.                                                                                                                                  |
| <b>Skenario Utama</b>                                                                                                               |                                                                                                                                                                                                   |
| <b>Aksi dari Aktor</b>                                                                                                              | <b>Respon dari Sistem</b>                                                                                                                                                                         |
| 1. Peserta menekan tombol lihat daftar ujian dan aplikasi <i>client</i> melakukan request data daftar ujian ke <i>web service</i> . | 2. Aplikasi memvalidasi token peserta.                                                                                                                                                            |
|                                                                                                                                     | 3. Aplikasi mengirimkan daftar ujian yang diminta oleh peserta.                                                                                                                                   |
| <b>Skenario Alternatif 1 : Jika parameter tidak lengkap</b>                                                                         |                                                                                                                                                                                                   |
|                                                                                                                                     | 4. Aplikasi mengirimkan pesan error bahwa parameter tidak lengkap                                                                                                                                 |
| <b>Skenario Alternatif 2 : Jika token tidak valid</b>                                                                               |                                                                                                                                                                                                   |
|                                                                                                                                     | 5. Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.                                                                                                                              |

Tabel 4.24 Skenario *Use Case* Request\_soal

|                       |                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Nama Use Case</b>  | <b>Request_soal</b>                                                                                                     |
| <b>Nomor Use Case</b> | SRS_2_05                                                                                                                |
| <b>Tujuan</b>         | Untuk melakukan <i>request soal</i> dan memulai ujian.                                                                  |
| <b>Deskripsi</b>      | <i>Use case</i> ini menjelaskan bagaimana peserta mendapatkan soal untuk melaksanakan suatu ujian.                      |
| <b>Aktor</b>          | Peserta                                                                                                                 |
| <b>Pre-Condition</b>  | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> . Selain itu, ujian harus sedang berada dalam |

|                                                                                          |                                                                                                                                                          |
|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                          | status sedang dilaksanakan.                                                                                                                              |
| <b>Post-Condition</b>                                                                    | Peserta mendapatkan data soal dan memulai ujian.                                                                                                         |
| <b>Skenario Utama</b>                                                                    |                                                                                                                                                          |
| <b>Aksi dari Aktor</b>                                                                   | <b>Respon dari Sistem</b>                                                                                                                                |
| 1. Peserta menekan tombol mulai ujian pada ujian yang berada dalam daftar ujian peserta. | 2. Aplikasi memvalidasi <i>token</i> peserta.                                                                                                            |
|                                                                                          | 3. Aplikasi melakukan pengecekan apakah ujian bisa dilaksanakan atau tidak. Baik berdasarkan waktu pelaksanaan ataupun hak akses peserta terhadap ujian. |
|                                                                                          | 4. Aplikasi memilih soal dan melakukan pengacakan soal dari <i>database</i> .                                                                            |
|                                                                                          | 5. Aplikasi menyimpan pemetaan soal dan jawaban dalam <i>database</i> .                                                                                  |
|                                                                                          | 6. Aplikasi mengirimkan data soal dan atribut ujian lainnya (contoh: waktu pengerjaan) kepada aplikasi <i>client</i> .                                   |
| <b>Skenario Alternatif 1 : Jika parameter tidak lengkap</b>                              |                                                                                                                                                          |
|                                                                                          | 7. Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                                                 |
| <b>Skenario Alternatif 2 : Jika token tidak valid</b>                                    |                                                                                                                                                          |
|                                                                                          | 8. Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                                                              |
| <b>Skenario Alternatif 3 : Jika id_ujian tidak valid</b>                                 |                                                                                                                                                          |
|                                                                                          | 9. Aplikasi mengirimkan pesan bahwa id_ujian tidak valid                                                                                                 |
| <b>Skenario Alternatif 4 : Jika ujian sedang tidak dapat dilakukan</b>                   |                                                                                                                                                          |
|                                                                                          | 10. Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tidak dapat dilaksanakan.                                                                        |
| <b>Skenario Alternatif 5 : Jika peserta bukan termasuk peserta ujian tersebut</b>        |                                                                                                                                                          |
|                                                                                          | 11. Aplikasi mengirimkan pesan bahwa peserta tidak memiliki hak untuk mengikuti ujian tersebut.                                                          |

**Skenario Alternatif 6 : Jika peserta telah me-request soal**

|  |                                                                                                                    |
|--|--------------------------------------------------------------------------------------------------------------------|
|  | 12. Aplikasi mengirimkan pesan <i>error</i> bahwa peserta telah me-request soal dan tidak dapat melakukannya lagi. |
|--|--------------------------------------------------------------------------------------------------------------------|

**Tabel 4.25 Skenario Use Case Kirim\_Jawaban**

| Nama Use Case                                                                                                                                            | Kirim_jawaban                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Nomor Use Case                                                                                                                                           | SRS_2_06                                                                                                                  |
| Tujuan                                                                                                                                                   | Untuk mengirimkan jawaban ujian ke <i>web service</i> .                                                                   |
| Deskripsi                                                                                                                                                | Use case ini menjelaskan bagaimana peserta mengirimkan jawaban ujian yang telah dikerjakan ke <i>web service</i> .        |
| Aktor                                                                                                                                                    | Peserta                                                                                                                   |
| Pre-Condition                                                                                                                                            | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> . Peserta juga telah sedang melaksanakan ujian. |
| Post-Condition                                                                                                                                           | Peserta berhasil mengirimkan jawaban ke <i>web service</i> .                                                              |
| Skenario Utama                                                                                                                                           |                                                                                                                           |
| Aksi dari Aktor                                                                                                                                          | Respon dari Sistem                                                                                                        |
| 1. Setelah menjawab semua soal, peserta menekan tombol submit. Aplikasi <i>client</i> mengirimkan data jawaban dalam format JSON ke <i>web service</i> . | 2. Aplikasi memvalidasi token peserta.                                                                                    |
|                                                                                                                                                          | 3. Aplikasi melakukan pengecekan validitas ujian.                                                                         |
|                                                                                                                                                          | 4. Aplikasi melakukan pengecekan skema jawaban yang dikirimkan oleh aplikasi <i>client</i> .                              |
|                                                                                                                                                          | 5. Aplikasi melakukan koreksi jawaban berdasarkan pemetaan soal dan jawaban yang tersimpan dalam <i>database</i> .        |
|                                                                                                                                                          | 6. Aplikasi menyimpan data hasil ujian peserta ke dalam <i>database</i> .                                                 |

| Skenario Alternatif 1 : Jika parameter tidak lengkap                      |                                                                                                                                  |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
|                                                                           | 7. Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                         |
| Skenario Alternatif 2 : Jika token tidak valid                            |                                                                                                                                  |
|                                                                           | 8. Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                                      |
| Skenario Alternatif 3 : Jika id_ujian yang dikirimkan tidak valid         |                                                                                                                                  |
|                                                                           | 9. Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                                                      |
| Skenario Alternatif 4 : Jika skema jawaban tidak valid                    |                                                                                                                                  |
|                                                                           | 10. Aplikasi mengirimkan pesan <i>error</i> bahwa skema ujian tidak valid.                                                       |
| Skenario Alternatif 5 : Jika pengguna belum melakukan <i>request</i> soal |                                                                                                                                  |
|                                                                           | 11. Aplikasi mengirimkan pesan <i>error</i> bahwa peserta belum memulai ujian                                                    |
| Skenario Alternatif 6 : Jika pengguna telah mengirimkan jawaban           |                                                                                                                                  |
|                                                                           | 12. Aplikasi mengirimkan pesan <i>error</i> bahwa peserta sudah pernah mengirimkan jawaban dan tidak bisa mengirimkannya kembali |
| Skenario Alternatif 7 : Jika waktu pengerjaan telah habis                 |                                                                                                                                  |
|                                                                           | 13. Aplikasi mengirimkan pesan <i>error</i> bahwa waktu pengerjaan telah habis dan peserta tidak dapat mengumpulkan jawaban      |

Tabel 4.26 Skenario *Use Case* Request\_hasil\_ujian

| Nama Use Case  | Request_hasil_ujian                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------|
| Nomor Use Case | SRS_2_07                                                                                             |
| Tujuan         | Untuk mendapatkan data hasil ujian.                                                                  |
| Deskripsi      | <i>Use case</i> ini menjelaskan bagaimana peserta mendapatkan data hasil ujian yang telah dilakukan. |
| Aktor          | Peserta                                                                                              |
| Pre-Condition  | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> . Peserta                  |

|                                                                                                                                                                                                                                  |                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|                                                                                                                                                                                                                                  | juga telah melaksanakan ujian yang akan dilihat hasilnya.                   |
| <b>Post-Condition</b>                                                                                                                                                                                                            | Peserta berhasil mendapatkan data hasil ujian.                              |
| <b>Skenario Utama</b>                                                                                                                                                                                                            |                                                                             |
| <b>Aksi dari Aktor</b>                                                                                                                                                                                                           | <b>Respon dari Sistem</b>                                                   |
| 1. Setelah mengirimkan jawaban suatu ujian, maka peserta dapat melakukan <i>request</i> data hasil ujian. Aplikasi <i>client</i> melakukan <i>request</i> ke <i>web service</i> dengan mengirimkan parameter token dan id ujian. | 2. Aplikasi memvalidasi token peserta.                                      |
|                                                                                                                                                                                                                                  | 3. Aplikasi melakukan pengecekan validitas ujian.                           |
|                                                                                                                                                                                                                                  | 4. Aplikasi melakukan pengecekan apakah peserta telah melaksanakan ujian    |
|                                                                                                                                                                                                                                  | 5. Aplikasi mengirimkan data hasil ujian kepada aplikasi <i>client</i> .    |
| <b>Skenario Alternatif 1 : Jika parameter tidak lengkap</b>                                                                                                                                                                      |                                                                             |
|                                                                                                                                                                                                                                  | 6. Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap    |
| <b>Skenario Alternatif 2 : Jika token tidak valid</b>                                                                                                                                                                            |                                                                             |
|                                                                                                                                                                                                                                  | 7. Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid. |
| <b>Skenario Alternatif 3 : Jika ujian yang dikirimkan tidak valid</b>                                                                                                                                                            |                                                                             |
|                                                                                                                                                                                                                                  | 8. Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid |

Tabel 4.27 Skenario *Use Case* Batal\_ujian

|                       |                                                             |
|-----------------------|-------------------------------------------------------------|
| <b>Nama Use Case</b>  | <b>Batal_ujian</b>                                          |
| <b>Nomor Use Case</b> | SRS_2_08                                                    |
| <b>Tujuan</b>         | Untuk mengundurkan diri dari ujian yang belum dilaksanakan. |

|                                                                                                                                                                                                                                   |                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Deskripsi</b>                                                                                                                                                                                                                  | <i>Use case</i> ini menjelaskan bagaimana peserta mengundurkan diri dari ujian yang belum dilaksanakan.                                                                             |
| <b>Aktor</b>                                                                                                                                                                                                                      | Peserta                                                                                                                                                                             |
| <b>Pre-Condition</b>                                                                                                                                                                                                              | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> . Peserta juga telah terdaftar sebagai peserta ujian yang akan dibatalkan namun ujian belum dilaksanakan. |
| <b>Post-Condition</b>                                                                                                                                                                                                             | Peserta berhasil mengundurkan diri dari ujian yang belum dilaksanakan.                                                                                                              |
| <b>Skenario Utama</b>                                                                                                                                                                                                             |                                                                                                                                                                                     |
| <b>Aksi dari Aktor</b>                                                                                                                                                                                                            | <b>Respon dari Sistem</b>                                                                                                                                                           |
| 1. Peserta menekan tombol batal ujian pada aplikasi <i>client</i> . Aplikasi <i>client</i> kemudian akan melakukan <i>request</i> pembatalan ujian ke <i>web service</i> dengan mengirimkan parameter token dan <i>id_ujian</i> . | 2. Aplikasi memvalidasi token peserta.                                                                                                                                              |
|                                                                                                                                                                                                                                   | 3. Aplikasi melakukan pengecekan validitas ujian.                                                                                                                                   |
|                                                                                                                                                                                                                                   | 4. Aplikasi melakukan pengecekan apakah peserta telah terdaftar dalam ujian tersebut.                                                                                               |
|                                                                                                                                                                                                                                   | 5. Aplikasi menghapus peserta dari daftar peserta ujian.                                                                                                                            |
| <b>Skenario Alternatif 1 : Jika parameter tidak lengkap</b>                                                                                                                                                                       |                                                                                                                                                                                     |
|                                                                                                                                                                                                                                   | 6. Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                                                                            |
| <b>Skenario Alternatif 2 : Jika token tidak valid</b>                                                                                                                                                                             |                                                                                                                                                                                     |
|                                                                                                                                                                                                                                   | 7. Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.                                                                                                                |
| <b>Skenario Alternatif 3 : Jika ujian yang dikirimkan tidak valid</b>                                                                                                                                                             |                                                                                                                                                                                     |
|                                                                                                                                                                                                                                   | 8. Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                                                                                                         |
| <b>Skenario Alternatif 4 : Jika peserta bukan termasuk peserta dalam ujian yang akan dibatalkan</b>                                                                                                                               |                                                                                                                                                                                     |
|                                                                                                                                                                                                                                   | 9. Aplikasi mengirimkan pesan <i>error</i> bahwa peserta bukan ujian sehingga tidak perlu                                                                                           |

|  |                    |
|--|--------------------|
|  | mengundurkan diri. |
|--|--------------------|

Tabel 4.28 Skenario *Use Case* Request\_profil

| Nama <i>Use Case</i>                                                                                                                                                                                                            | Request_profil                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Nomor <i>Use Case</i>                                                                                                                                                                                                           | SRS_2_09                                                                                                                              |
| Tujuan                                                                                                                                                                                                                          | Untuk mendapatkan profil pengguna.                                                                                                    |
| Deskripsi                                                                                                                                                                                                                       | <i>Use case</i> ini menjelaskan bagaimana peserta mendapatkan profil pengguna. Baik profil Administrator maupun profil peserta ujian. |
| Aktor                                                                                                                                                                                                                           | Peserta                                                                                                                               |
| <i>Pre-Condition</i>                                                                                                                                                                                                            | Peserta telah berhasil ke dalam sistem dan mendapatkan akses <i>token</i> .                                                           |
| <i>Post-Condition</i>                                                                                                                                                                                                           | Peserta berhasil mendapatkan data pengguna.                                                                                           |
| Skenario Utama                                                                                                                                                                                                                  |                                                                                                                                       |
| Aksi dari Aktor                                                                                                                                                                                                                 | Respon dari Sistem                                                                                                                    |
| 1. Peserta melakukan tap pada gambar profil pengguna pada aplikasi <i>client</i> . Aplikasi <i>client</i> kemudian akan melakukan <i>request</i> data ke <i>web service</i> dengan mengirimkan parameter token dan id pengguna. | 2. Aplikasi memvalidasi token peserta.                                                                                                |
|                                                                                                                                                                                                                                 | 3. Aplikasi melakukan pengecekan validitas id pengguna.                                                                               |
|                                                                                                                                                                                                                                 | 4. Aplikasi mengirimkan data profil pengguna pada aplikasi <i>client</i> .                                                            |
| Skenario Alternatif 1 : Jika parameter tidak lengkap                                                                                                                                                                            |                                                                                                                                       |
|                                                                                                                                                                                                                                 | 5. Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap.                                                             |
| Skenario Alternatif 2 : Jika token tidak valid                                                                                                                                                                                  |                                                                                                                                       |
|                                                                                                                                                                                                                                 | 6. Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                                           |

#### 4.1.4 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan kebutuhan aplikasi yang tidak diminta oleh pengguna namun dibutuhkan agar aplikasi dapat berjalan dengan baik dalam memenuhi kebutuhan pengguna. Kebutuhan Non-fungsional sendiri adalah batasan pada layanan atau fungsi yang ditawarkan oleh sistem seperti batasan waktu, batasan pada proses pembangunan, standar, dan lain-lain.

Analisis kebutuhan non fungsional dari aplikasi *backend* sistem ujian online terpadu dijelaskan pada Tabel 4.3 berikut ini.

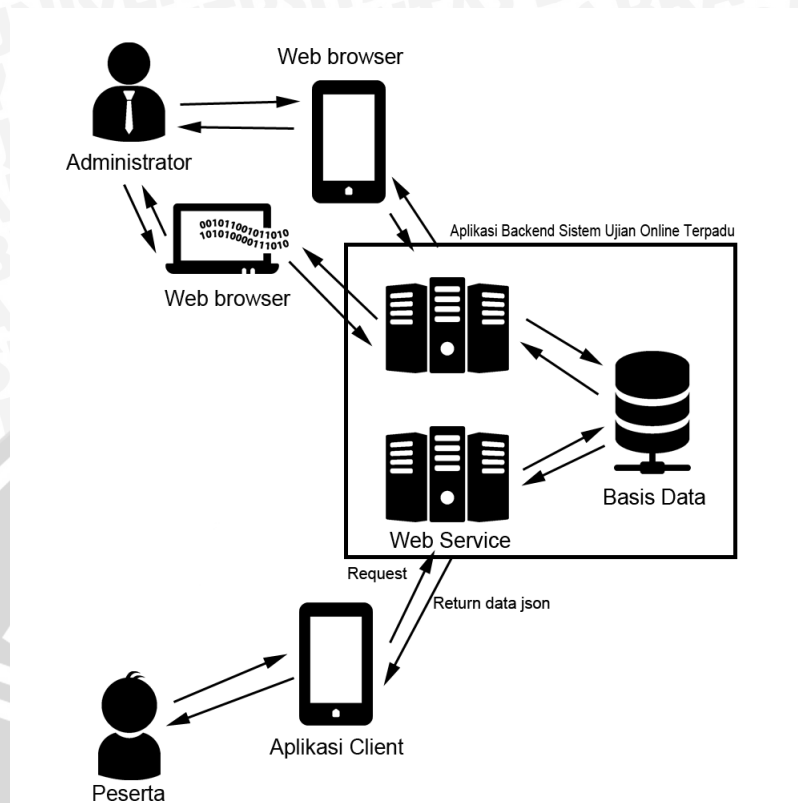
**Tabel 4.3 Spesifikasi kebutuhan non fungsional**

| Parameter          | Deskripsi Kebutuhan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Security</i>    | Aplikasi bisa diakses oleh pengguna, baik administrator maupun peserta ujian berdasarkan batas hak masing-masing melalui fitur login. Selain itu, pada <i>web service</i> , perlu diterapkan penggunaan token. Token dimiliki oleh masing-masing peserta secara unik. Untuk melakukan <i>request</i> , aplikasi <i>client</i> harus mengirimkan token yang valid. Token terdiri dari 32 karakter yang dihasilkan dari penggabungan <i>string username</i> , 1 angka acak antara 1 sampai 10 dan <i>password</i> yang kemudian dienkripsi menggunakan algoritma md5. Token akan diganti setiap 5 jam untuk menghindari kecurangan. Selain menggunakan token, ip address juga diperiksa ketika <i>client</i> melakukan <i>request</i> . |
| <i>Reliability</i> | Aplikasi menerapkan mekanisme pengiriman soal sekaligus ketika peserta melakukan <i>request</i> untuk memulai ujian. Hal ini berfungsi agar ujian bisa dilaksanakan tanpa harus selalu terhubung dengan internet selama ujian berlangsung. Dengan demikian, aplikasi harus dapat melayani <i>concurrent user</i> yang mengakses <i>service</i> secara bersamaan.                                                                                                                                                                                                                                                                                                                                                                      |

#### 4.2 Perancangan Aplikasi

Tahap perancangan aplikasi *backend* sistem ujian online terpadu terdiri dari beberapa tahap, yaitu perancangan arsitektur sistem, perancangan *activity diagram*, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, perancangan data eksternal dan perancangan navigasi serta perancangan antarmuka.

#### 4.2.1 Perancangan Arsitektur Sistem



**Gambar 4.3 Arsitektur Sistem Ujian Online Terpadu**

Perancangan arsitektur sistem dapat diilustrasikan seperti pada gambar 4.3. Terdapat dua aktor dalam sistem, yaitu Administrator dan peserta ujian. Administrator mengakses aplikasi melalui browser. Browser berfungsi sebagai *client* yang melakukan *request* ke server. Server kemudian akan merespon permintaan dari *client* sesuai dengan apa yang diminta oleh *client*. Sedangkan peserta ujian mengakses sistem melalui aplikasi perangkat bergerak yang berfungsi sebagai *client*. Namun aplikasi *client* tidak dibatasi hanya pada aplikasi perangkat bergerak. Aplikasi *backend* dirancang agar dapat menyediakan layanan untuk berbagai aplikasi yang dibangun dengan *platform* yang berbeda-beda. Peserta ujian berinteraksi dengan aplikasi *client* dan aplikasi *client* kemudian mengirimkan permintaan kepada *web service* dari aplikasi *backend* sistem ujian online terpadu. Data yang dikirimkan oleh *web service* direpresentasikan dalam bentuk JSON yang kemudian diolah lagi pada aplikasi *client* dan ditampilkan kepada peserta.

##### 4.2.1.1 Perancangan Data Eksternal

Perancangan data eksternal membahas perancangan input tertentu yang kemudian menghasilkan output tertentu. Pada Aplikasi Backend Sistem Ujian Online Terpadu, Aplikasi backend menyediakan *web service* bagi aplikasi *client* untuk melakukan *request resource*. Aplikasi melakukan *request resource* yang diinginkan ke URL dari masing-masing *resource* dengan mengirimkan parameter menggunakan HTTP *method* tertentu (POST,GET,DELETE,PUT). Pertukaran data

menggunakan *middleware* JSON. JSON dipilih karena lebih ringan dibandingkan dengan XML (Kudi, 2015). Tabel 4.4 merupakan tabel *service* yang disediakan oleh Aplikasi Backend Sistem Ujian Online Terpadu.

**Tabel 4.4 Service pada Aplikasi Backend Sistem Ujian Online Terpadu**

| Nomor SRS | Nama Service               | Parameter                                                                    | HTTP Method | Return Data Format | Request URL                                                                                 |
|-----------|----------------------------|------------------------------------------------------------------------------|-------------|--------------------|---------------------------------------------------------------------------------------------|
| SRS_2_01  | Login                      | - username<br>- password                                                     | POST        | JSON               | http://localhost/oes/api/login/                                                             |
| SRS_2_02  | Register                   | - username<br>- nama<br>- email<br>- password<br>- tanggal_lahir<br>- gender | POST        | JSON               | http://localhost/oes/api/register/                                                          |
| SRS_2_03  | Daftar_ujian               | - token<br>- kode_enroll                                                     | POST        | JSON               | http://localhost/oes/api/daftar_ujian/                                                      |
| SRS_2_04  | Lihat_daftar_ujian_peserta | - token                                                                      | GET         | JSON               | http://localhost/oes/api/lihat_daftar_ujian/token/<br>667a4c7b1ef981ccc01f4637c7b01842      |
| SRS_2_05  | Request_soal               | - token<br>- id_ujian                                                        | GET         | JSON               | http://localhost/oes/api/mulai_ujian/token/<br>667a4c7b1ef981ccc01f4637c7b01842/id_ujian/42 |
| SRS_2_06  | Kirim_jawaban              | - token<br>- id_ujian<br>- jawaban                                           | POST        | JSON               | http://localhost/oes/api/kirim_jawaban                                                      |
| SRS_2_07  | Request_hasil_ujian        | - token<br>- id_ujian                                                        | GET         | JSON               | http://localhost/oes/api/hasil_ujian/token/667a4c7b1ef981ccc01f4637c7b01842/id/42           |

|          |                |                       |      |      |                                                                        |
|----------|----------------|-----------------------|------|------|------------------------------------------------------------------------|
| SRS_2_08 | Batal_ujian    | - token<br>- id_ujian | POST | JSON | http://localhost/oes/api/batal_ujian                                   |
| SRS_2_09 | Request_profil | - token               | GET  | JSON | http://localhost/oes/api/profil/token/667a4c7b1ef981ccc01f4637c7b01842 |

Setelah aplikasi *client* melakukan *request* dengan mengirimkan parameter yang benar ke URL dari *resource* yang diinginkan, aplikasi akan memberikan respon dengan mengirimkan *resource* dalam bentuk JSON. Tabel 4.5 merupakan tabel respon masing-masing *service* dalam bentuk JSON.

**Tabel 4.5 Respon masing-masing *service***

| Nama Service | Potongan Data JSON                                                                                                     |
|--------------|------------------------------------------------------------------------------------------------------------------------|
| Login        | <pre> 1 { 2   "status": true, 3   "message": "login sukses", 4   "token": "667a4c7b1ef981ccc01f4637c7b01842" 5 }</pre> |
| Register     | <pre> 1 { 2   "status": true, 3   "message": "registrasi sukses" 4 }</pre>                                             |
| Daftar_ujian | <pre> 1 { 2   "status": true, 3   "message": "permohonan pendaftaran ujian berhasil dikirimkan" 4 }</pre>              |

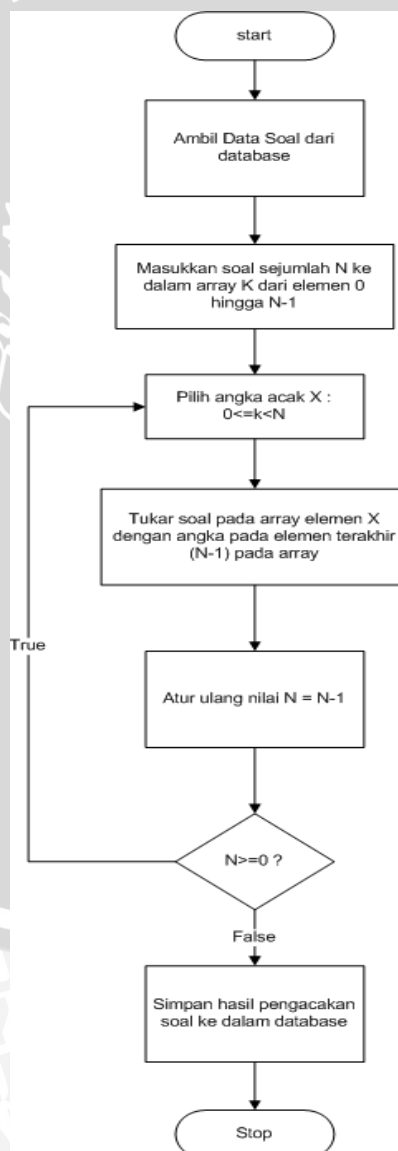
|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Lihat_daftar_ujian_peserta</p> | <pre> 1  { 2      "status": true, 3      "message": "request sukses", 4      "data": [{ 5          "id_ujian": "41", 6          "nama_ujian": "Quiz Matematika 1", 7          "waktu_mulai_pelaksanaan": "2015-11-21 01:39:00", 8          "waktu_akhir_pelaksanaan": "2015-11-23 02:00:00", 9          "username_creator": "jack0falltrades", 10         "status": "sudah melaksanakan" 11     }, { 12         "id_ujian": "44", 13         "nama_ujian": "Ujian Psikotes Test1", 14         "waktu_mulai_pelaksanaan": "2015-12-08 13:37:00", 15         "waktu_akhir_pelaksanaan": "2015-12-09 01:00:00", 16         "username_creator": "jack0falltrades", 17         "status": "belum melaksanakan" 18     }] 19 } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <p>Request_soal</p>               | <pre> 1  { 2      "status": true, 3      "message": "Pesan sukses", 4      "data_ujian": { 5          "id": "44", 6          "username_creator": "jack0falltrades", 7          "grade_max": "100", 8          "enrollment_key": "psikotest", 9          "deskripsi": "Ujian percobaan untuk psikotest", 10         "passing_grade": "50", 11         "nama_ujian": "Ujian Psikotes Test1", 12         "durasi": "30", 13         "waktu_pembuatan": "2015-12-08 13:36:21", 14         "waktu_mulai_pelaksanaan": "2015-12-08 13:37:00", 15         "waktu_akhir_pelaksanaan": "2015-12-09 01:00:00", 16         "jumlah_soal_minimal": "3", 17         "status": "1", 18         "is_shuffle": "1" 19     }, 20     "data_soal": [{ 21         "id": "62", 22         "nama_soal": "soal 2", 23         "id_ujian": "44", 24         "tipe_soal": "0", 25         "pertanyaan": "&lt;p&gt;&lt;span style='font-family: arial; font-size: 12px; text-align: justify; background-color: #f0f0f0; padding: 5px;'&gt;Di sini ada perempatan jalan yang saling tegak lurus. Salah satu jalan itu menuju ke kota Semarang utara, maka jalan yang ke kiri dan belakang saya adalah jalan yang menuju ke tempat lain. Jalan yang saya menuju ke Laut Jawa. Jalan manakah yang menuju Semarang ?&lt;/span&gt;&lt;/p&gt;", 26         "level_kesulitan": "sedang", 27         "gambar": "", 28         "pilihan": [{ 29             "id": "12", 30             "teks": "Jawaban a, b, dan c salah", 31             "gambar": "" 32         }, { 33             "id": "10", 34             "teks": "Ke Timur", 35             "gambar": "" 36         }, { 37             "id": "11", 38             "teks": "Ke Barat", 39             "gambar": "" 40         }, { 41             "id": "9", 42             "teks": "Ke utara", 43             "gambar": "" 44         }] 45     }], 46 } </pre> |
| <p>Kirim_jawaban</p>              | <p>Skema jawaban yang diterima oleh server (Input dari aplikasi client)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <pre> 1 [{ 2   "nomor": 1, 3   "jawaban": "Jawaban a, b dan c salah" 4 }, { 5   "nomor": 2, 6   "jawaban": "Ke Timur" 7 }, { 8   "nomor": 3, 9   "jawaban": "86" 10 }] </pre> <p>Keterangan :</p> <ul style="list-style-type: none"> <li>- nomor adalah nomor soal</li> <li>- jawaban adalah string atau teks dari pilihan jawaban ataupun jawaban yang dimasukkan peserta ujian</li> </ul> <p>Response dari server :</p> <pre> 1 { 2   "status": true, 3   "message": "Jawaban berhasil dikirim" 4 } </pre> |
| Request_hasil_ujian | <pre> 1 { 2   "status": true, 3   "message": "sukses", 4   "data": { 5     "id": "42", 6     "nama_ujian": "Ujian psikotes", 7     "passing_grade": "20", 8     "grade_max": "100", 9     "waktu_ujian": "2015-11-22 15:56:01", 10    "skor": "33.3333", 11    "waktu_submit": "2015-11-22 15:56:04", 12    "status": "lulus" 13  } 14 } </pre>                                                                                                                                                              |
| Batal_ujian         | <pre> 1 { 2   "status": true, 3   "message": "pembatalan ujian berhasil" 4 } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                          |

|                |                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Request_profil | <pre> 1      { 2          "username": "herdian", 3          "nama": "herdian adi winarno", 4          "email": "emailnyaherdian@mail.com", 5          "gambar": "", 6          "tanggal_lahir": "1993-02-13", 7          "gender": "L" 8      } </pre> |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

#### 4.2.1.2 Perancangan Algoritma Pengacakan Soal

Algoritma yang digunakan untuk mengacak soal pada Aplikasi *Backend* Sistem Ujian Online Terpadu menggunakan Algoritma Fisher-Yates versi modern. Gambar 4.2 menunjukkan diagram alur dari algoritma Fisher-Yates.

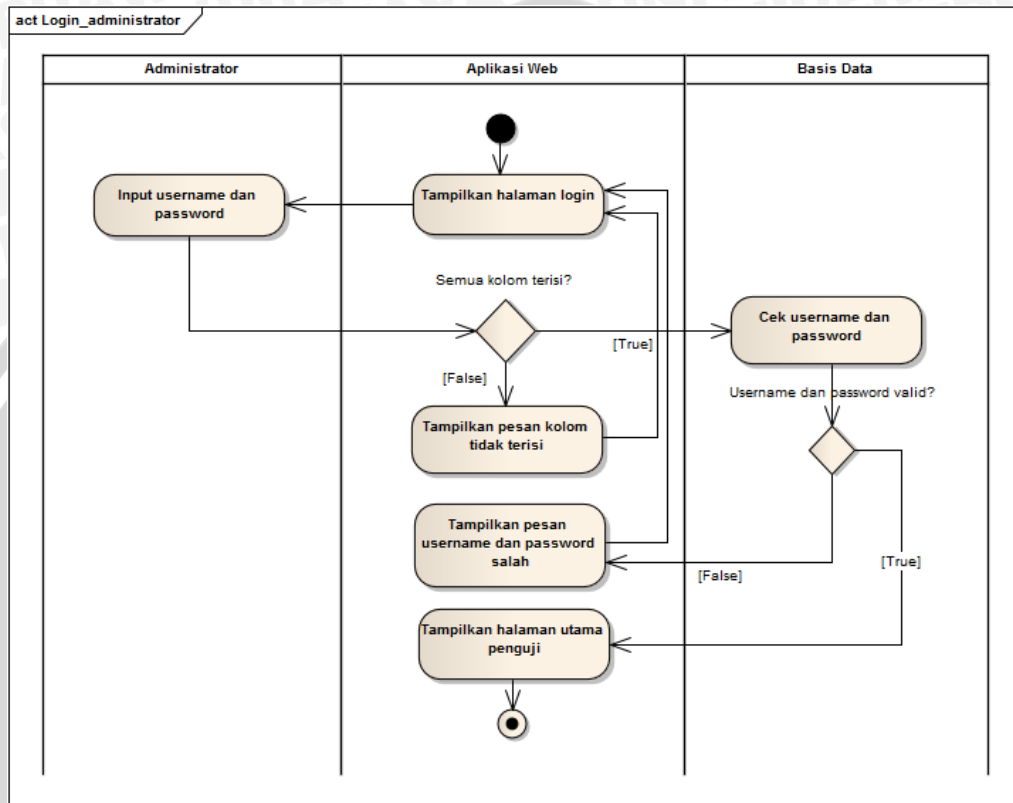


Gambar 4.2 Diagram Alur Algoritma Pengacakan Fisher-Yates

#### 4.2.2 Perancangan Activity Diagram

Perancangan *Activity Diagram* merupakan pemodelan aktifitas dan interaksi antara pengguna dan sistem berdasarkan pada skenario *use case*. Tahapan tiap skenario *use case* dijabarkan ke dalam grafis diagram alur.

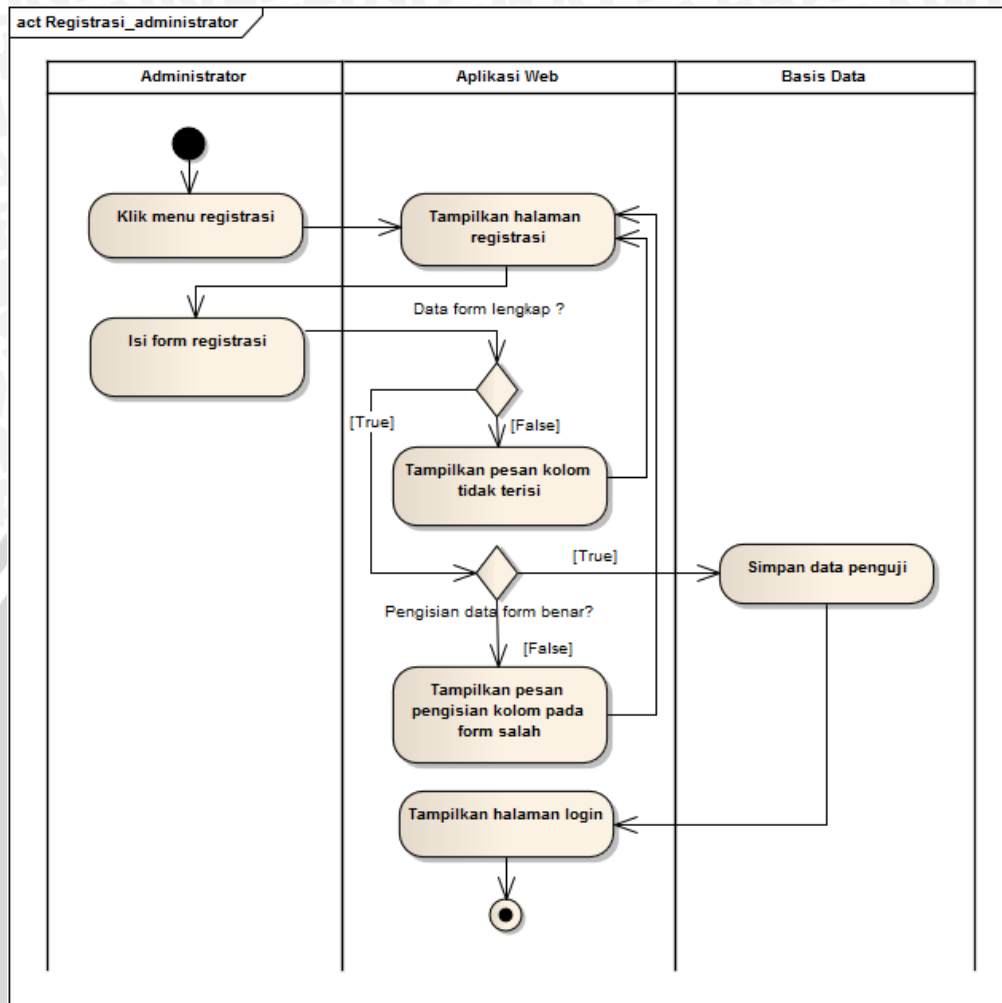
##### 4.2.2.1 Activity Diagram Login Administrator



**Gambar 4.3 Activity Diagram login Administrator**

Gambar 4.3 menunjukkan interaksi antara sistem dengan Administrator. Untuk dapat login ke dalam aplikasi, Administrator harus memasukkan *username* dan *passssord* yang valid. Aplikasi akan memeriksa *username* dan *password* Administrator saat Administrator mencoba untuk *login*. Jika keduanya valid, maka Administrator dibawa ke halaman utama Administrator.

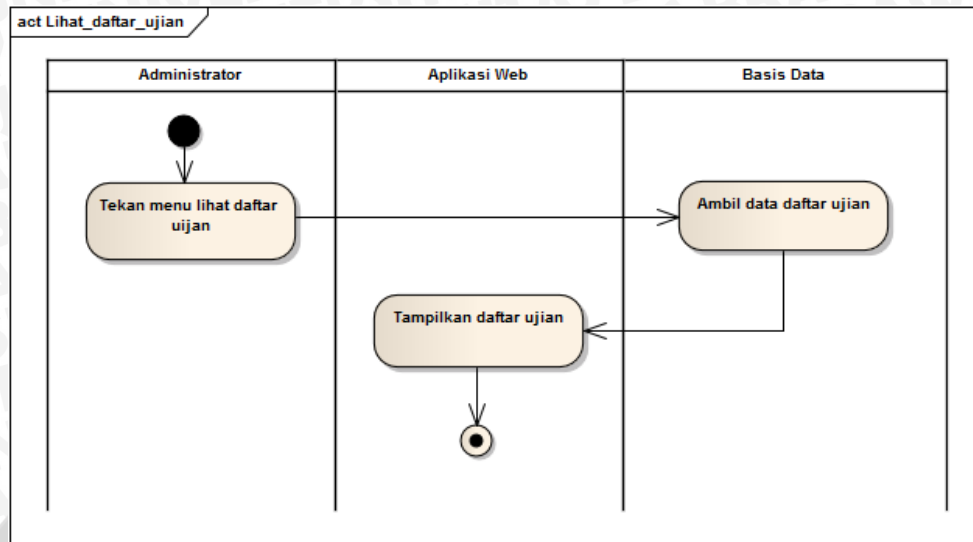
#### 4.2.2.2 Activity Diagram Registrasi Administrator



**Gambar 4.4 Activity Diagram Registrasi Administrator**

Gambar 4.4 menunjukkan interaksi antara sistem dan Administrator ketika Administrator melakukan registrasi ke dalam sistem. Fitur registrasi ini bertujuan agar pengguna yang belum memiliki akun Administrator, dapat membuat akun yang bisa login ke dalam sistem. Pertama setelah mendapatkan form registrasi, calon Administrator harus mengisi form registrasi tersebut dengan benar. Jika semua data telah terisi dengan benar, maka akun Administrator akan dibuat dan pengguna yang mendaftar dapat login ke dalam sistem menggunakan akun tersebut.

#### 4.2.2.3 Activity Diagram Lihat Daftar Ujian

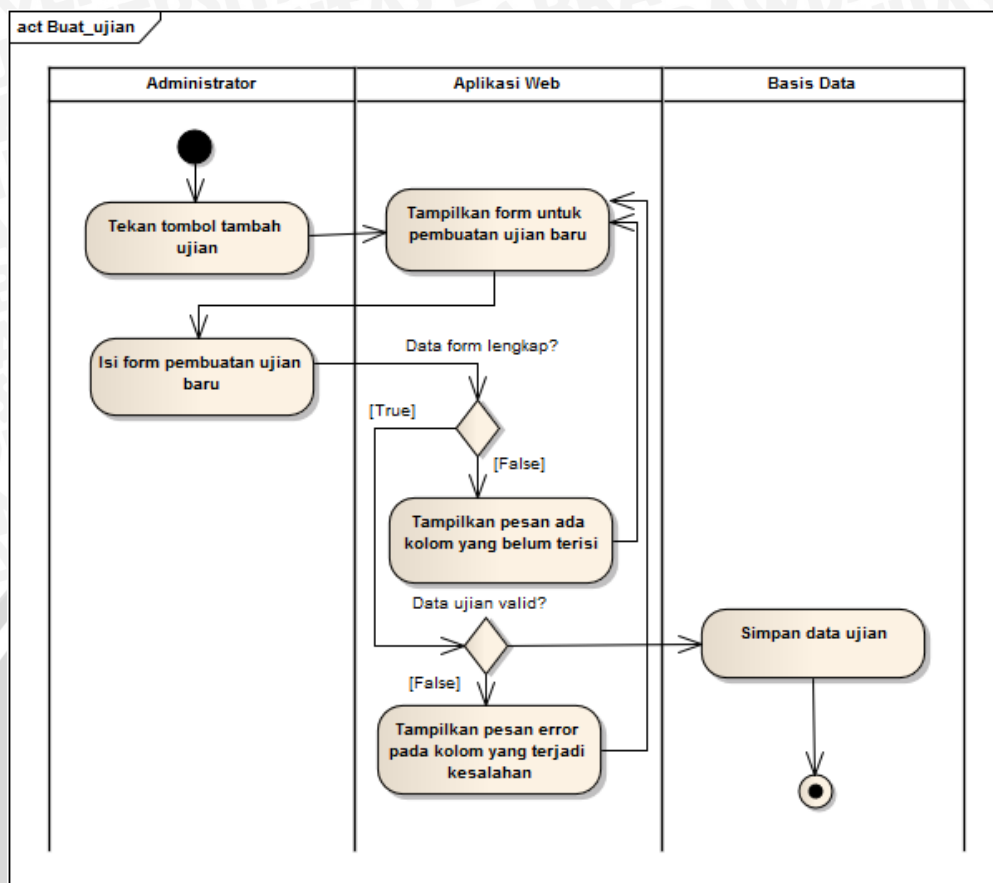


**Gambar 4.5 Activity Diagram Lihat Daftar Ujian**

Gambar 4.5 menggambarkan interaksi antara sistem dengan Administrator ketika Administrator akan melihat daftar ujian yang telah dibuatnya. Untuk melihat daftar ujian tersebut, Administrator perlu menekan tombol lihat daftar ujian. Setelah itu data ujian dari *database* akan diambil dan ditampilkan. Daftar ujian yang ditampilkan hanyalah ujian yang dibuat oleh user. Semua ujian yang dibuat oleh user akan ditampilkan, baik ujian yang telah dilaksanakan maupun yang belum dilaksanakan.

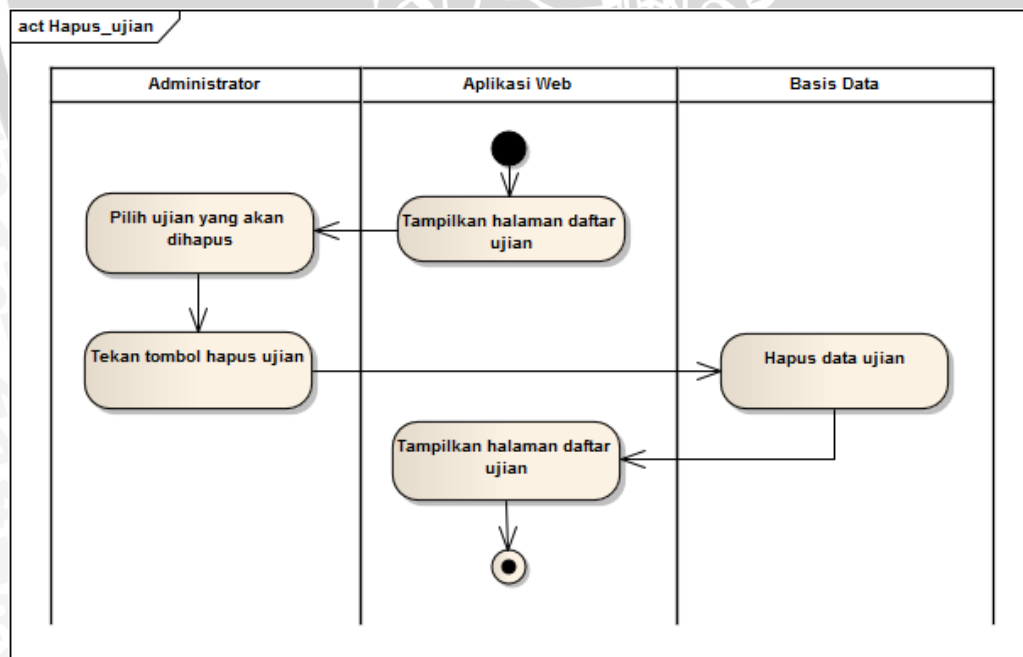
#### 4.2.2.4 Activity Diagram Buat Ujian

Gambar 4.6 merupakan interaksi antara sistem dan Administrator ketika Administrator akan membuat ujian baru. Administrator akan diberikan form pembuatan ujian baru oleh sistem yang harus diisi oleh Administrator. Setelah semuanya terisi dan benar, maka proses pembuatan ujian berhasil. Sebagai catatan, ketika selesai dibuat, ujian akan berstatus belum lengkap. Hal ini dikarenakan Administrator belum menyiapkan soal yang lengkap untuk ujian. Jika soal telah memasukkan soal-soal dan mencapai jumlah minimal soal yang diperlukan untuk melaksanakan ujian, maka status ujian akan berubah menjadi lengkap dan dapat dilaksanakan. Sebagai contoh, jika minimal soal yang diperlukan untuk melakukan ujian adalah 25, maka Administrator harus memasukkan minimal 25 soal ke dalam ujian tersebut. Baru setelah itu status ujian akan berubah menjadi lengkap.



Gambar 4.6 Buat Ujian

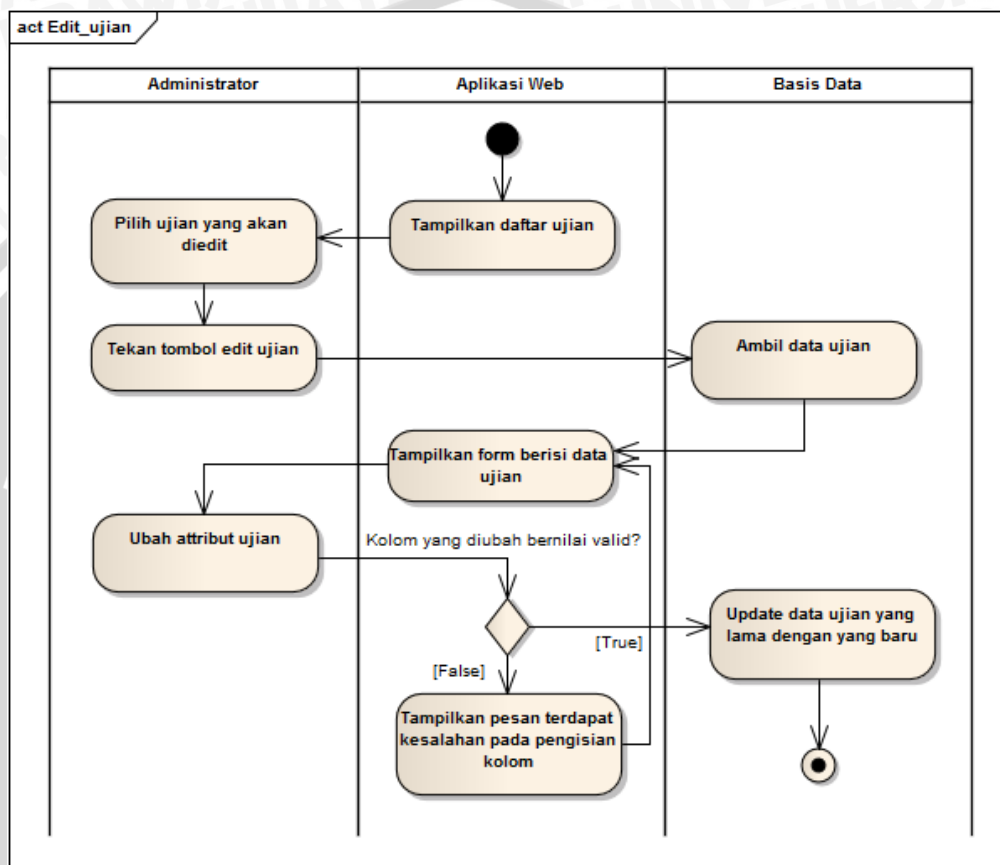
#### 4.2.2.5 Activity Diagram Hapus Ujian



Gambar 4.7 Activity Diagram Hapus Ujian

Gambar 4.7 merupakan gambaran interaksi antara sistem dengan Administrator ketika Administrator melakukan penghapusan ujian. Proses penghapusan ujian sangat singkat. Setelah melihat daftar ujian yang dibuat Administrator, Administrator kemudian memilih ujian yang akan dihapus dan menekan tombol hapus ujian. Setelah itu ujian akan dihapus dari *database*.

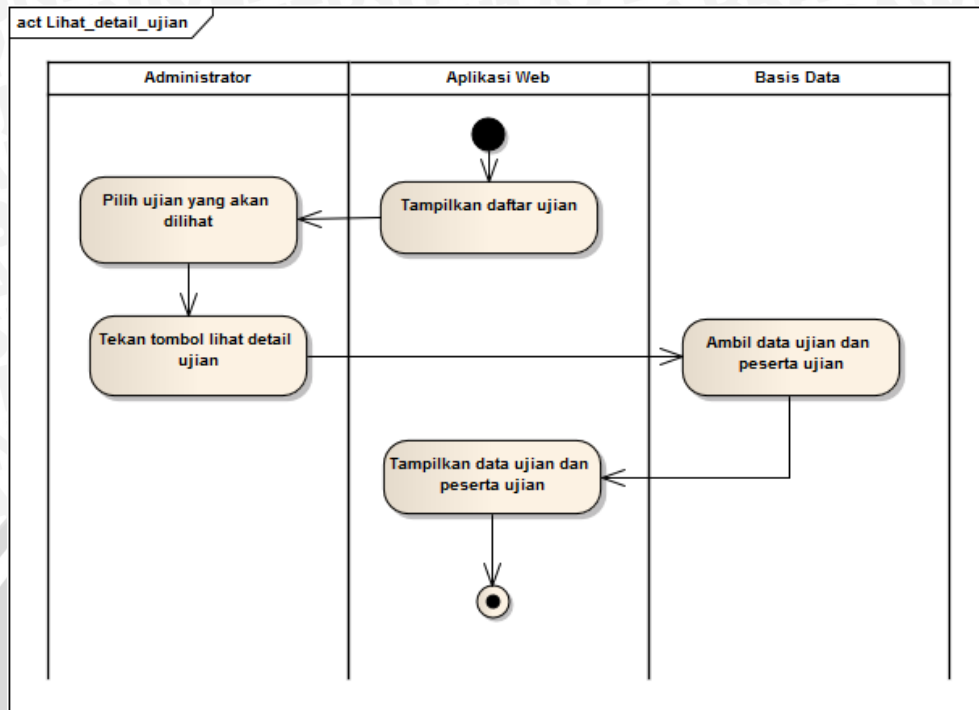
#### 4.2.2.6 Activity Diagram Edit Ujian



**Gambar 4.8** Activity Diagram Edit Ujian

Gambar 4.8 merupakan interaksi antara sistem dan Administrator ketika Administrator melakukan edit ujian. Setelah Administrator memilih ujian yang akan diedit, sistem akan menampilkan form berisi data ujian sebelumnya. Kemudian Administrator mengubah data pada form sesuai dengan perubahan yang ingin dilakukan. Sistem akan memeriksa apakah perubahan yang dilakukan valid atau tidak. Jika perubahan yang dilakukan valid, maka data ujian yang lama akan diganti dengan data ujian yang baru.

#### 4.2.2.7 Activity Diagram Lihat Detail Ujian

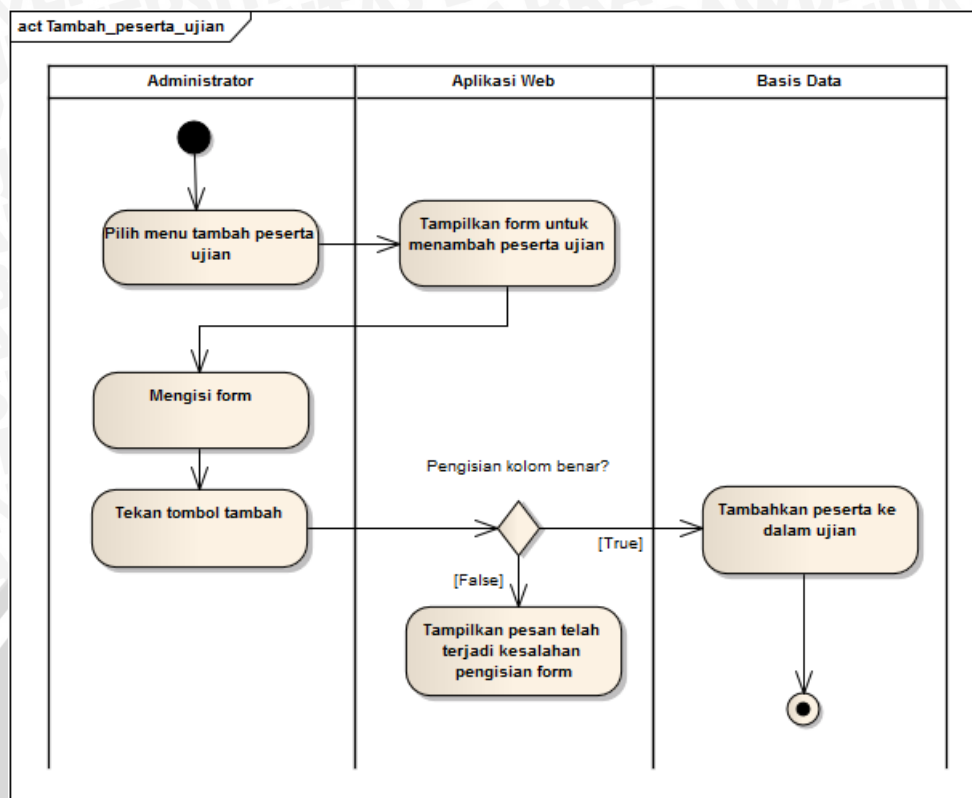


**Gambar 4.9 Activity Diagram Lihat Detail Ujian**

Gambar 4.9 merupakan interaksi antara sistem dengan Administrator ketika Administrator akan melihat detail ujian. Ketika Administrator memilih lihat detail ujian, maka sistem akan menampilkan data detail suatu ujian seperti data atribut ujian dan peserta ujian. Untuk melihat detail ujian, Administrator hanya cukup menekan tombol lihat detail ujian pada ujian yang ingin dilihat. Setelah itu sistem akan menampilkan data ujian tersebut kepada Administrator.

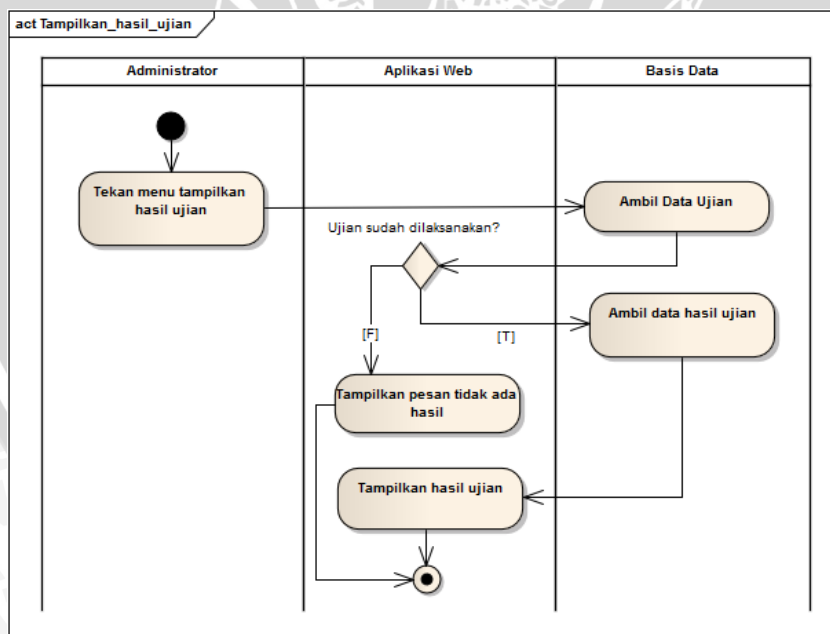
#### 4.2.2.8 Activity Diagram Tambah Peserta Ujian

Gambar 4.10 menunjukkan interaksi antara sistem dengan Administrator ketika Administrator akan menambahkan peserta dalam suatu ujian. Selain peserta yang dapat mengajukan diri untuk menjadi peserta suatu ujian, terdapat juga fitur bagi Administrator untuk menambahkan peserta ujian secara manual. Untuk melakukan hal ini, Administrator hanya perlu menekan tombol tambah peserta ujian dan mengisi form penambahan peserta ujian. Peserta ditambahkan dengan menggunakan username. Administrator dapat menambahkan banyak peserta sekaligus dengan menggunakan pemisah koma (,) ketika mengisi form penambahan peserta ujian. Setelah Administrator mengisi form, sistem akan memeriksa pesan. Jika tidak terdapat kesalahan, maka data peserta akan ditambahkan sebagai peserta resmi suatu ujian.



Gambar 4.10 Activity Diagram Tambah Peserta Ujian

#### 4.2.2.9 Activity Diagram Tampilkan Hasil Ujian

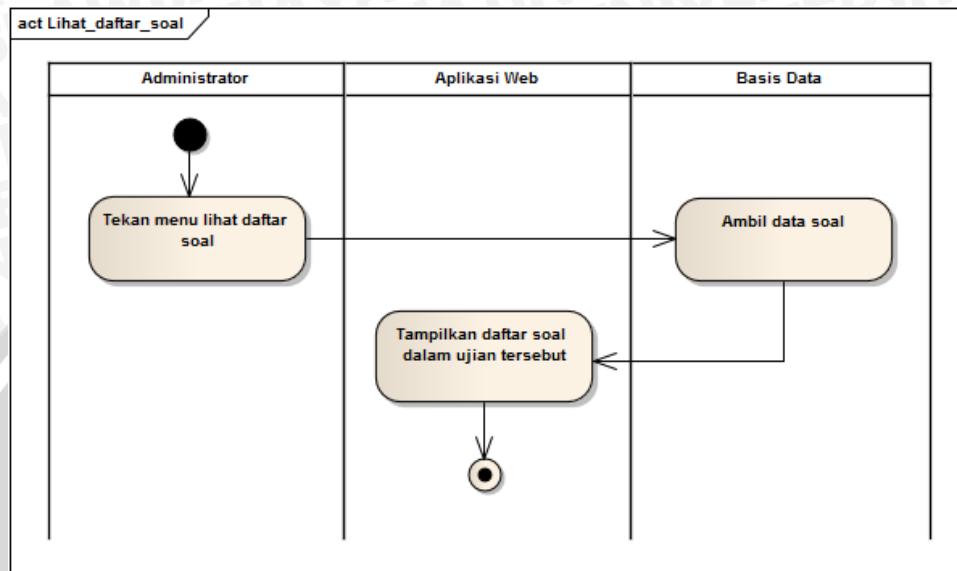


Gambar 4.11 Activity Diagram Tampilkan Hasil Ujian

Gambar 4.11 menggambarkan interaksi antara sistem dan Administrator ketika Administrator ingin menampilkan data hasil ujian. Data hasil ujian yang ditampilkan berupa data record hasil ujian peserta yang pernah mengikuti ujian,

data peserta yang lulus, grafik peserta yang lulus dan tidak, jumlah benar dan salah ketika peserta menjawab suatu soal, dan lain-lain. Untuk menampilkan hasil ujian, Administrator cukup memilih menu tampilkan hasil ujian dan sistem akan menampilkan data hasil ujian.

#### 4.2.2.10 Activity Diagram Lihat Daftar Soal

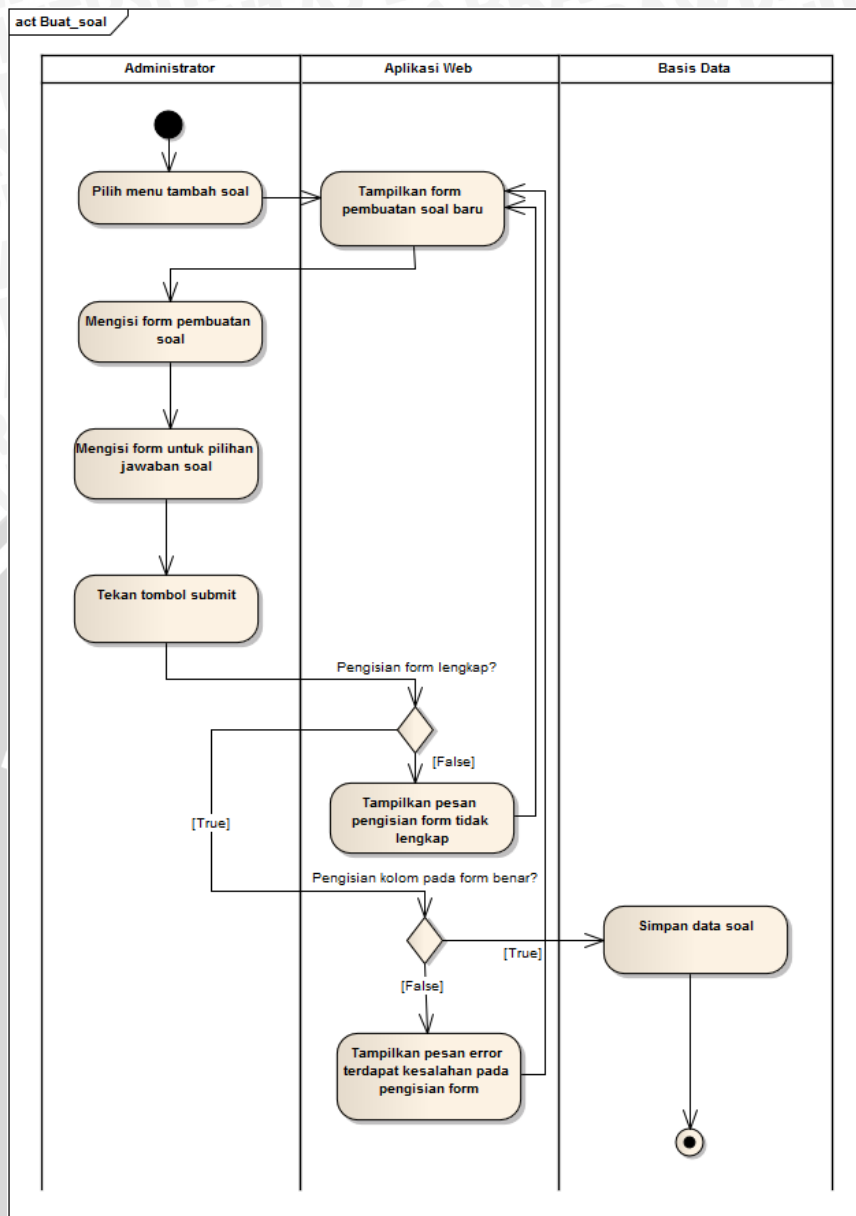


**Gambar 4.12 Activity Diagram Lihat Daftar Soal**

Gambar 4.12 menunjukkan interaksi antara sistem dan Administrator ketika Administrator ingin melihat daftar soal dalam suatu ujian. Untuk melihat soal yang ada pada suatu ujian, Administrator hanya perlu menekan menu lihat daftar soal. Sistem kemudian akan menampilkan daftar soal yang ada dalam suatu ujian.

#### 4.2.2.11 Activity Diagram Buat Soal

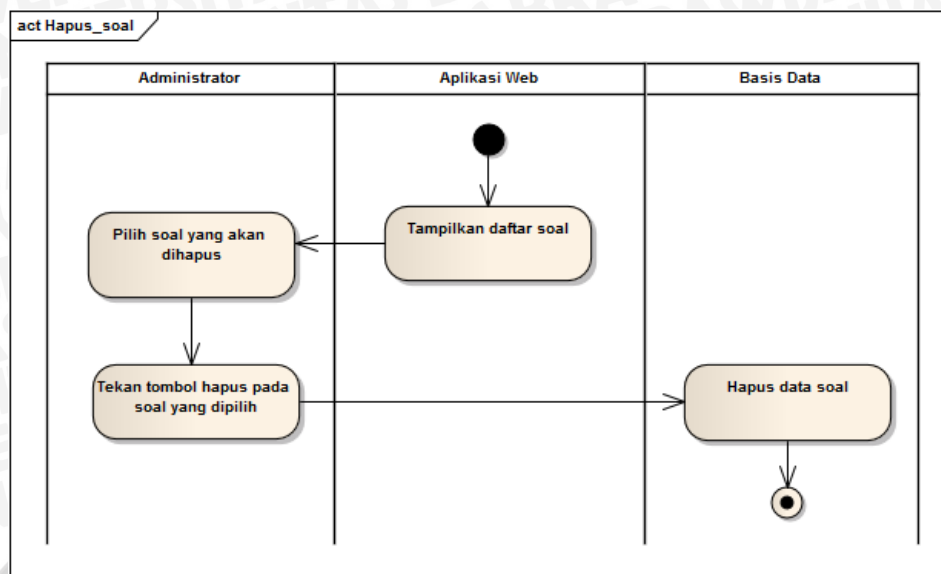
Gambar 4.13 menunjukkan interaksi antara sistem dan Administrator ketika Administrator akan membuat soal baru dalam suatu ujian.



Gambar 4.13 Activity Diagram Buat Soal

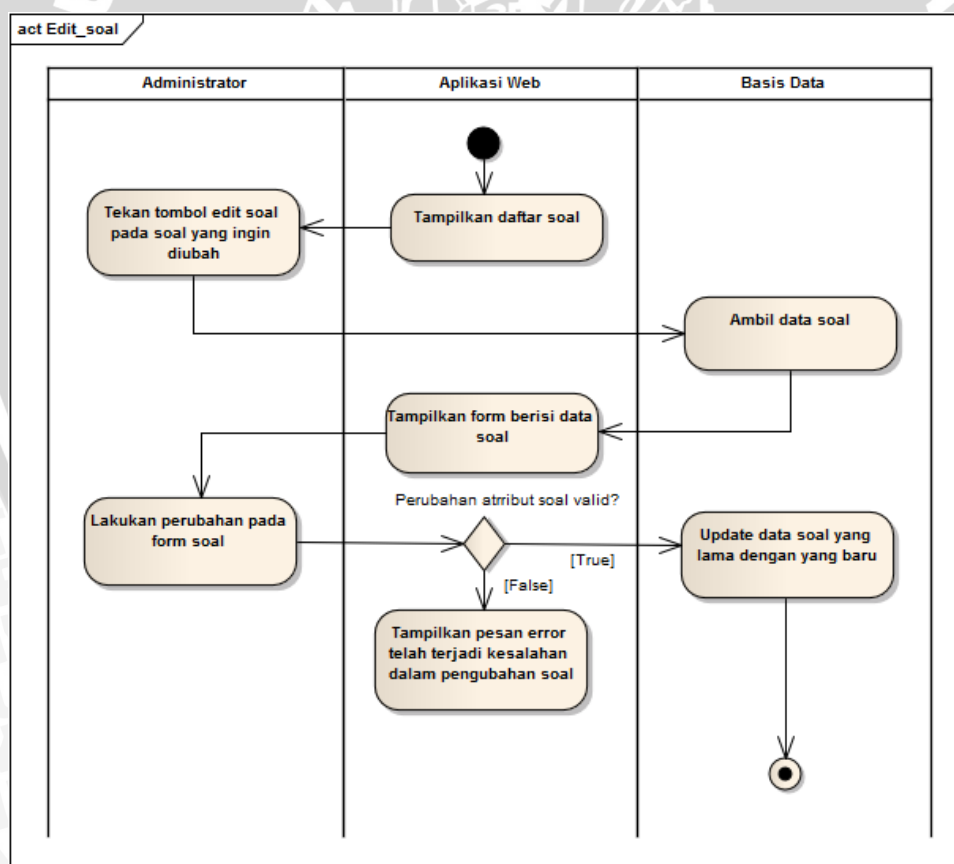
#### 4.2.2.12 Activity Diagram Hapus Soal

Gambar 4.14 menunjukkan interaksi antara sistem dan Administrator ketika Administrator akan menghapus soal dalam suatu ujian. Untuk menghapus soal, pertama Administrator harus memilih soal yang akan dihapus pada halaman lihat daftar soal. Setelah itu Administrator cukup menekan tombol hapus soal pada soal yang dipilih. Soal kemudian akan dihapus dari *database*.



Gambar 4.14 Activity Diagram Hapus Soal

#### 4.2.2.13 Activity Diagram Edit Soal

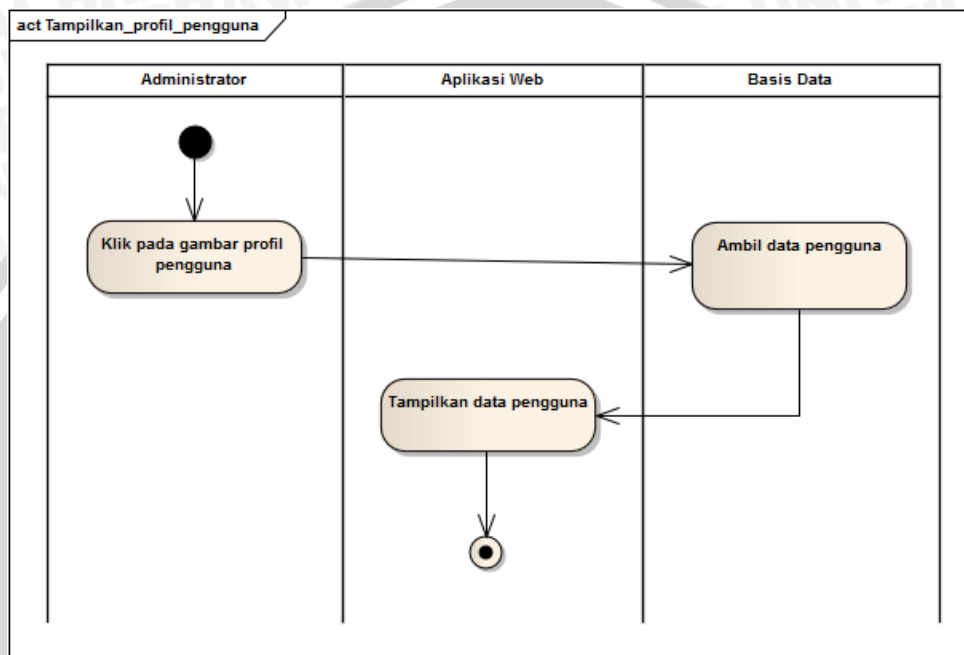


Gambar 4.15 Activity Diagram Edit Soal

Gambar 4.15 menunjukkan interaksi antara sistem dan Administrator ketika Administrator akan melakukan edit soal. Pada halaman lihat daftar soal,

Administrator harus menekan tombol edit soal pada salah satu soal yang akan di-*edit*. Setelah itu, sistem akan menampilkan form berisi data soal yang lama. Administrator kemudian mengubah data soal yang lama dengan baru dan menekan tombol simpan. Sebelum disimpan dalam *database*, data yang baru akan diperiksa terlebih dahulu oleh sistem. Jika semua data valid, maka data soal yang lama akan diganti dengan data soal yang baru.

#### 4.2.2.14 Activity Diagram Tampilkan Profil Pengguna

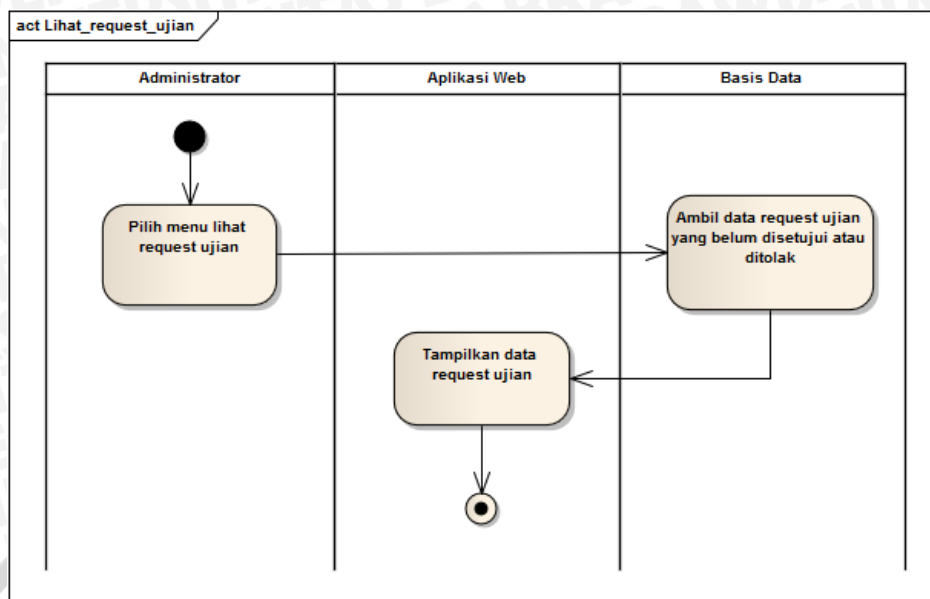


**Gambar 4.18 Activity Diagram Tampilkan Profil Pengguna**

Gambar 4.18 menunjukkan interaksi antara sistem dan Administrator ketika Administrator ingin menampilkan profil peserta. Untuk melihat profil pengguna, maka Administrator hanya perlu melakukan klik pada gambar profil pengguna dan sistem kemudian akan menampilkan profil pengguna tersebut.

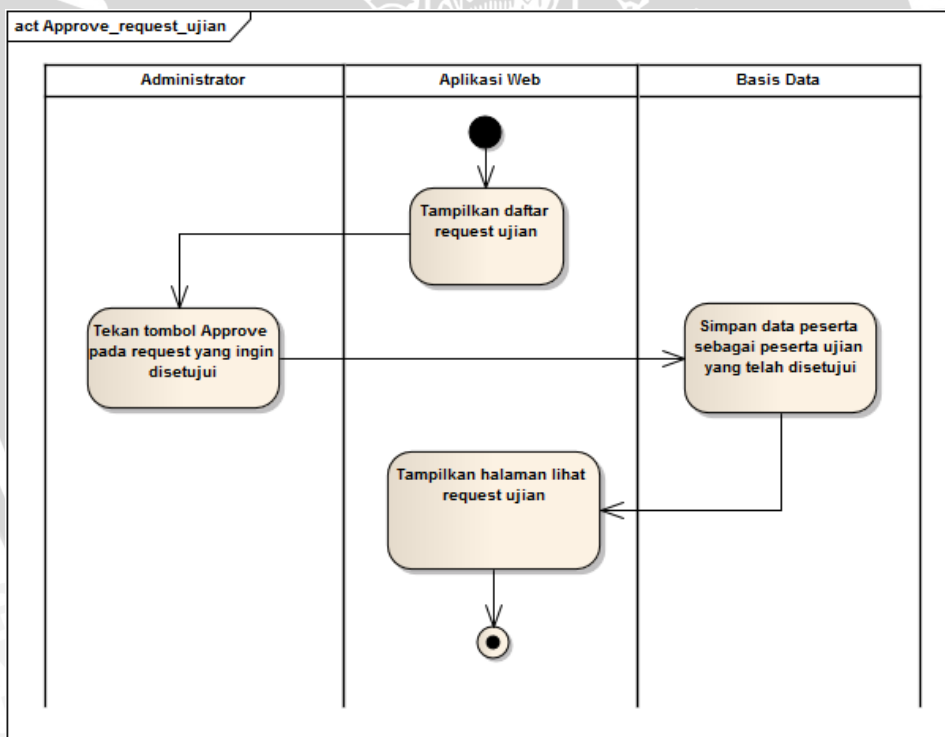
#### 4.2.2.15 Activity Diagram Lihat Request Ujian

Gambar 4.19 merupakan interaksi antara Administrator dan sistem ketika Administrator ingin melihat request ujian. Request ujian dilakukan oleh peserta ujian yang ingin mendaftar menjadi peserta suatu ujian. Administrator kemudian dapat menyetujui atau menolak request dari peserta ujian. Request ujian yang dapat dilihat hanyalah request ujian terhadap ujian yang dibuat oleh Administrator tersebut. Untuk melihat request ujian, Administrator hanya perlu memilih menu lihat request ujian dan sistem kemudian akan menampilkan daftar request ujian.



Gambar 4.19 Lihat Request Ujian

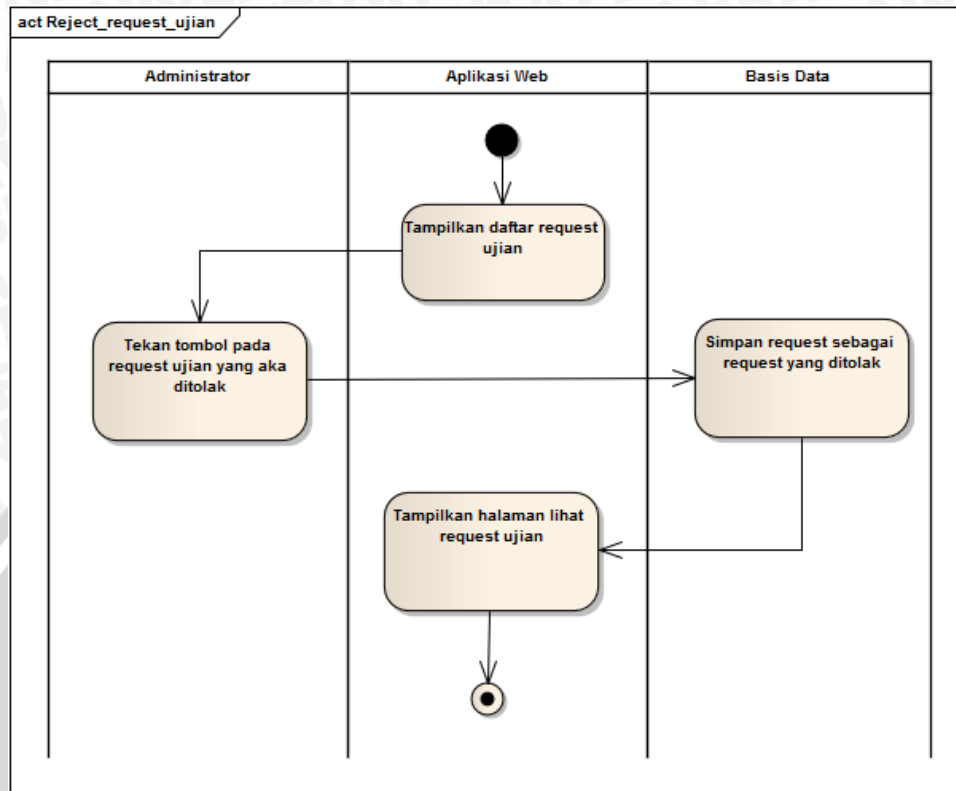
#### 4.2.2.16 Activity Diagram Approve Request Ujian



Gambar 4.20 Activity Diagram Approve Request Ujian

Gambar 4.20 menunjukkan interaksi antara sistem dan Administrator ketika Administrator akan menyetujui request dari peserta. Setelah disetujui, maka peserta akan terdaftar sebagai peserta resmi suatu ujian dan dapat melaksanakan ujian pada waktu yang telah ditentukan. Administrator menyetujui peserta dengan menekan tombol approve pada request yang ingin disetujui.

#### 4.2.2.17 Activity Diagram Reject Request Ujian

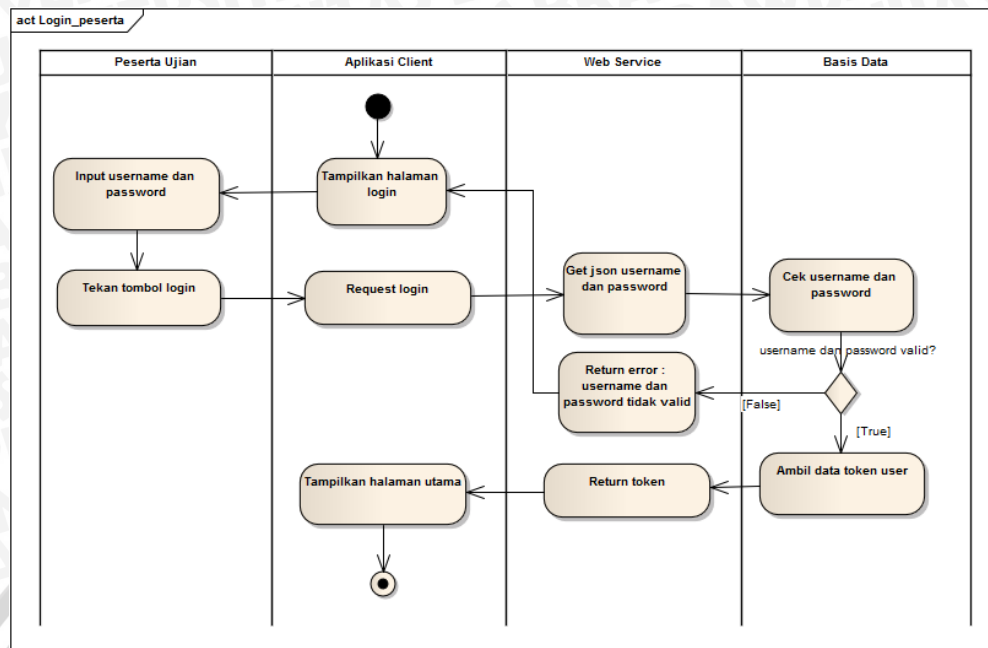


**Gambar 4.21 Activity Diagram Reject Request Ujian**

Gambar 4.21 menunjukkan interaksi antara sistem dan Administrator ketika Administrator akan menolak request ujian dari peserta. Fitur ini berguna agar Administrator dapat menolak peserta ilegal yang ingin mengikuti ujian. Administrator perlu menekan tombol tolak request pada request peserta yang akan ditolak.

#### 4.2.2.18 Activity Diagram Login Peserta

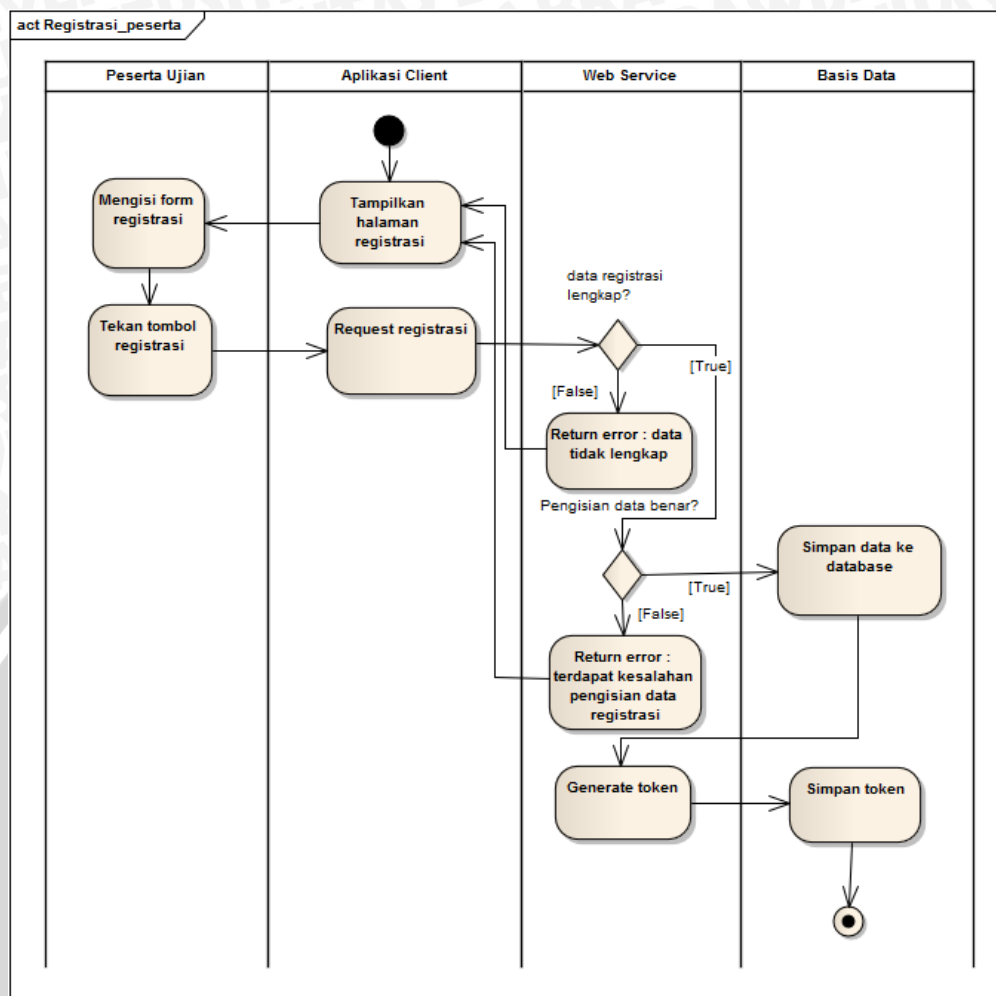
Gambar 4.22 menunjukkan interaksi antara sistem (*web service*) dan peserta ujian ketika peserta ujian akan login ke dalam sistem. Untuk login ke dalam sistem, peserta ujian harus memasukkan *username* dan *password* yang valid. Jika keduanya valid, maka *web service* akan mengembalikan token yang dapat digunakan untuk melakukan *request* lainnya.



Gambar 4.22 Activity Diagram Login Peserta

#### 4.2.2.19 Activity Diagram Registrasi Peserta

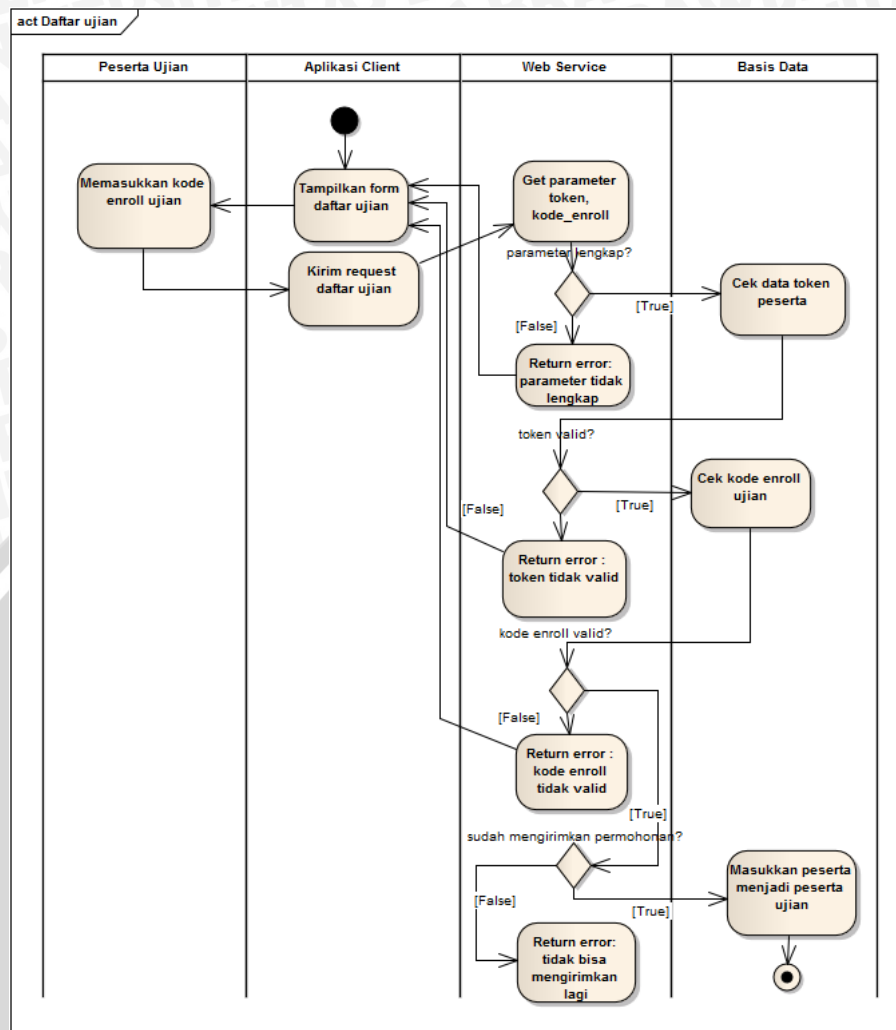
Gambar 4.23 menunjukkan interaksi antara sistem (*web service*) dan peserta ujian ketika peserta ujian melakukan registrasi. Untuk melakukan registrasi, peserta ujian harus mengisi data yang valid, seperti *username*, nama, email, *password*, dan lain-lain. Jika semua data valid, maka peserta akan mendapatkan akun untuk login ke dalam sistem.



Gambar 4.23 Activity Diagram Registrasi Peserta

#### 4.2.2.20 Activity Diagram Daftar Ujian

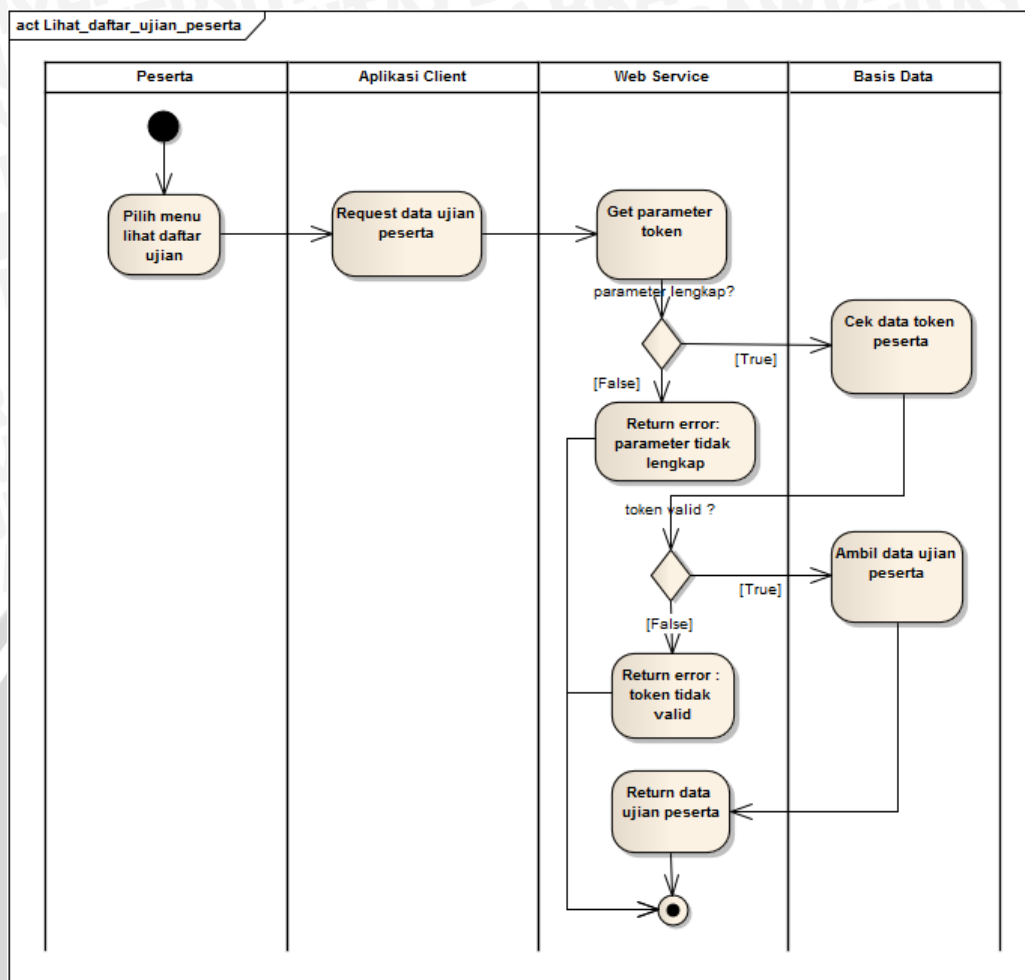
Gambar 4.24 menunjukkan interaksi antara sistem (*web service*) dan peserta ujian ketika peserta ujian ingin mendaftar menjadi peserta resmi suatu ujian. Untuk mendaftar ke dalam suatu ujian, peserta harus memasukkan kode *enroll* dari suatu ujian. Sistem akan memeriksa apakah kode tersebut benar. Jika benar, maka *request* peserta akan dikirim kepada Administrator untuk diverifikasi. Peserta ujian akan menjadi peserta resmi suatu ujian setelah disetujui oleh Administrator yang bersangkutan.



Gambar 4.24 Activity Diagram Daftar Ujian

#### 4.2.2.21 Activity Diagram Lihat Daftar Ujian Peserta

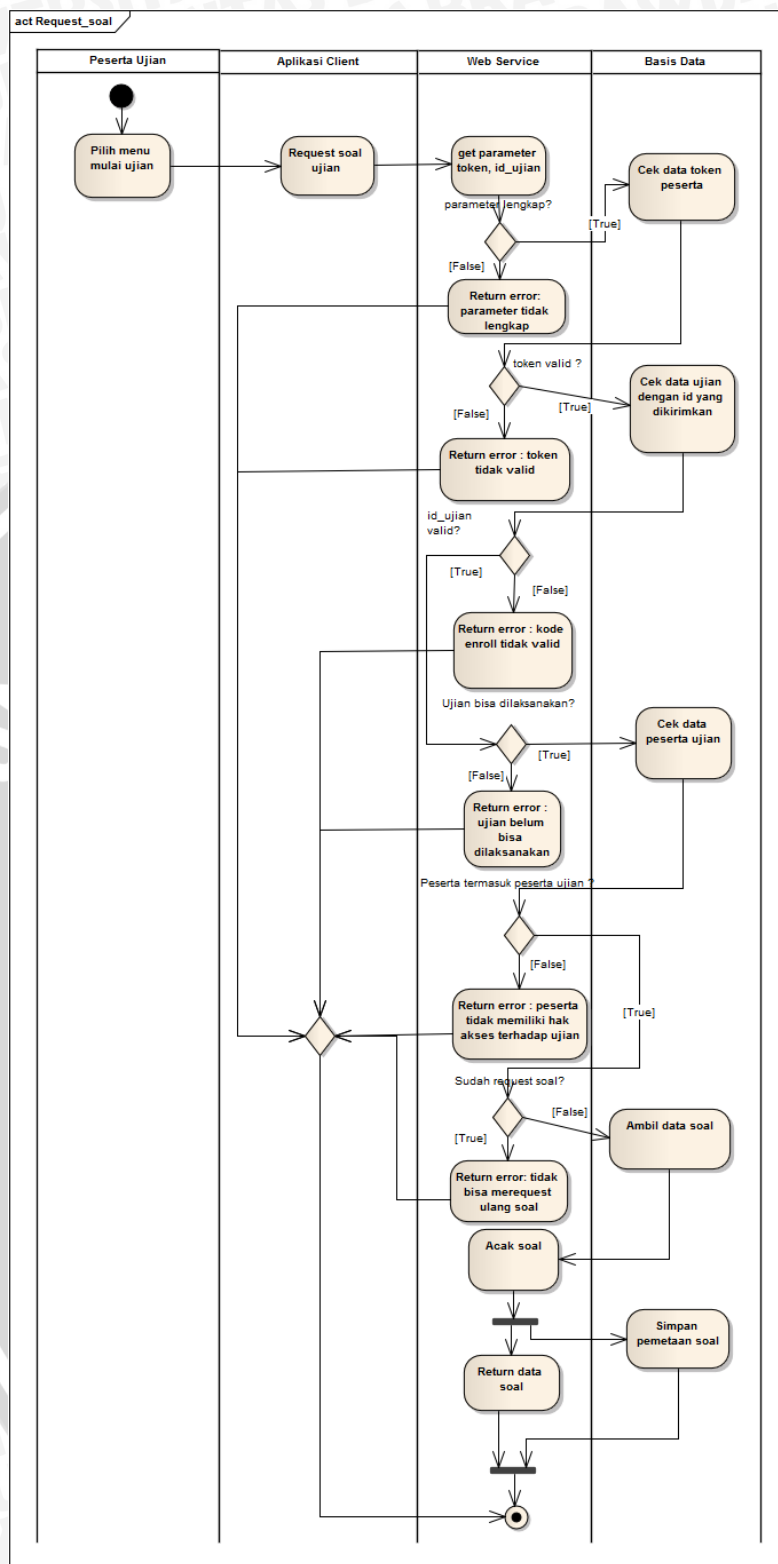
Gambar 4.25 menunjukkan interaksi antara sistem (web service) dan peserta ujian ketika peserta ujian ingin melihat daftar ujian yang diikutinya. Fitur ini hanya menampilkan daftar ujian yang mana peserta telah resmi menjadi peserta ujian dalam ujian tersebut. Ujian yang belum terlaksana maupun sudah terlaksana ditampilkan. Untuk melihat daftar ujian, peserta ujian perlu melakukan *request* ke *web service* menggunakan token yang valid.



Gambar 4.25 Activity Diagram Lihat Daftar Ujian Peserta

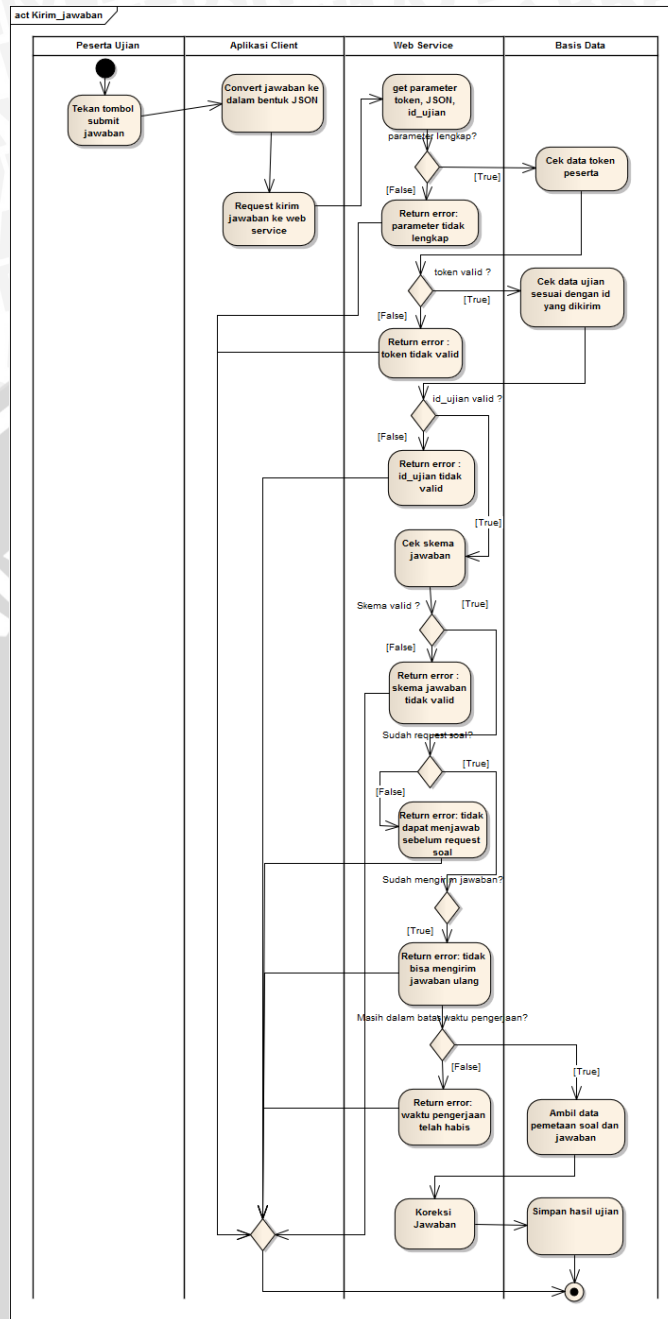
#### 4.2.2.22 Activity Diagram Request Soal

Gambar 4.26 menunjukkan interaksi antara sistem (web service) dan peserta ujian ketika peserta ujian ingin melaksanakan suatu ujian. Ketika ujian dimulai, aplikasi *client* akan melakukan request soal ke *web service*. Sistem akan memeriksa apakah token, hak akses dan ujian yang dikirimkan valid. Jika semuanya valid, sistem akan menyimpan pemetaan soal dan mengirimkan data soal kepada aplikasi *client*.



Gambar 4.26 Activity Diagram Request Soal

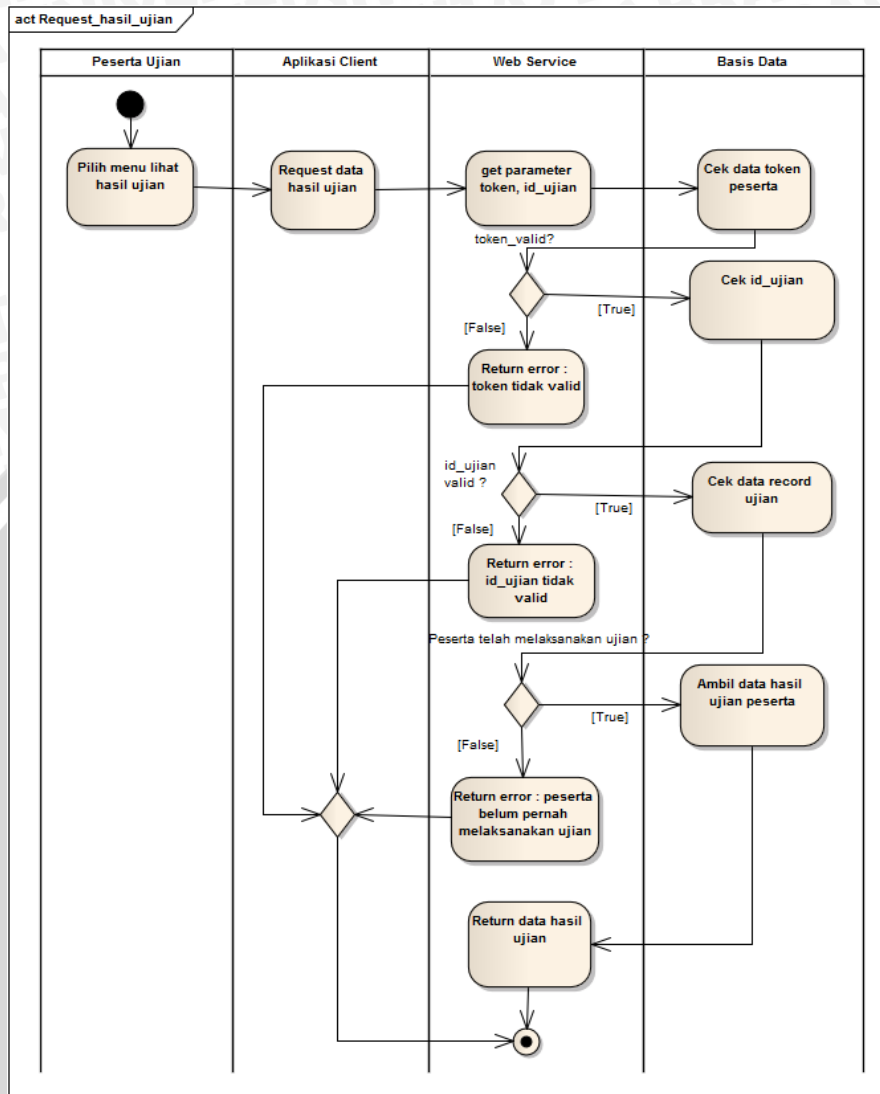
#### 4.2.2.23 Activity Diagram Kirim Jawaban



**Gambar 4.27 Activity Diagram Kirim Jawaban**

Gambar 4.27 menunjukkan interaksi antara sistem (web service) dan peserta ujian ketika peserta mengirimkan jawaban kepada sistem. Setelah melakukan request soal, maka peserta perlu untuk mengirimkan jawabannya kembali. Jawaban yang dikirimkan oleh peserta dari aplikasi *client* harus dalam bentuk JSON. Sistem akan melakukan beberapa pemeriksaan. Seperti pemeriksaan token, validitas ujian dan skema jawaban. Jika semuanya valid, maka sistem akan mengoreksi jawaban peserta.

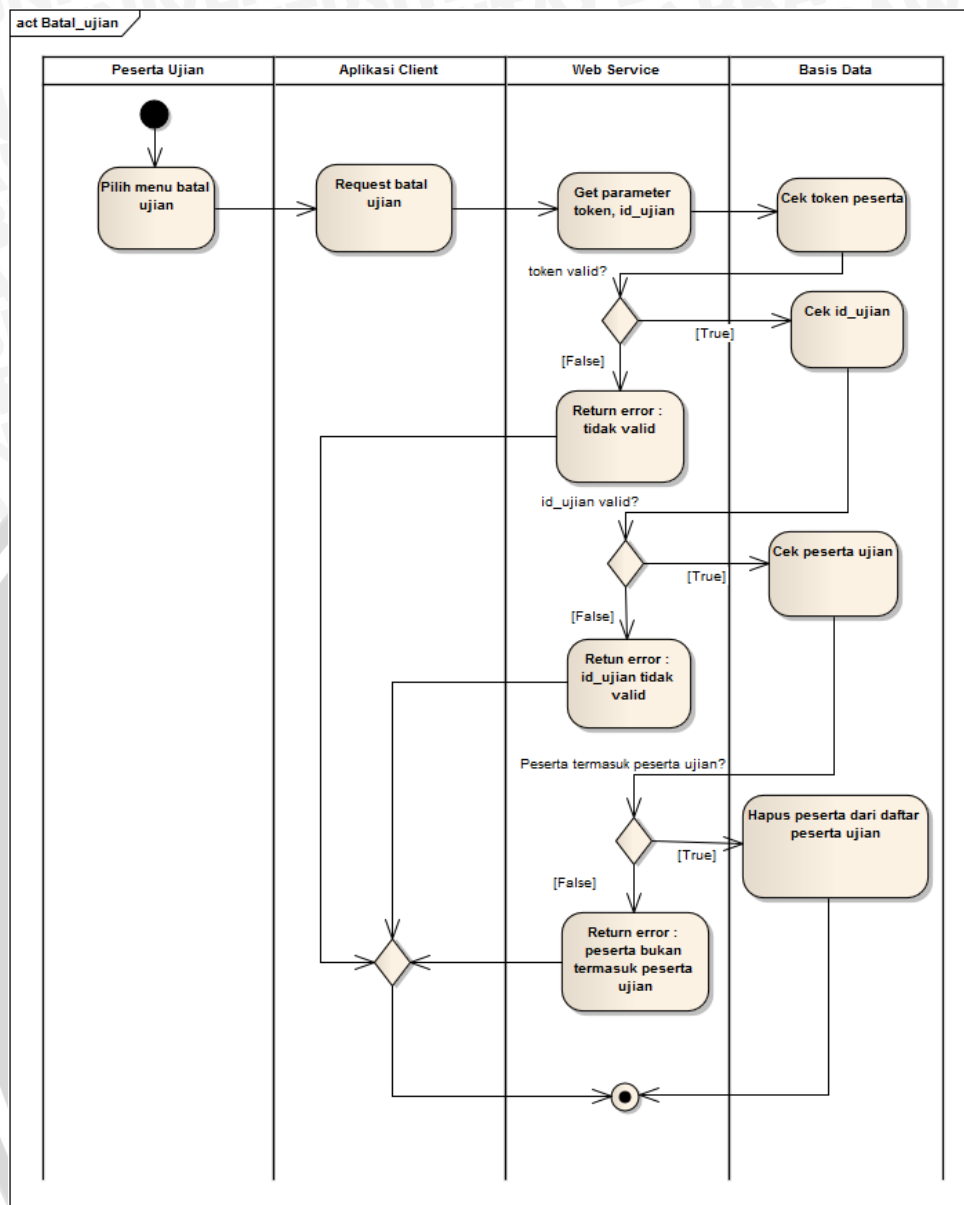
#### 4.2.2.24 Activity Diagram Request Hasil Ujian



**Gambar 4.28 Activity Diagram Request Hasil Ujian**

Gambar 4.28 menunjukkan interaksi antara sistem (web service) dan peserta ujian ketika peserta ujian ingin melihat hasil ujian yang telah dilaksanakan. Sistem akan melakukan pemeriksaan terhadap request dari aplikasi *client*. Seperti token, validitas ujian dan apakah peserta telah benar-benar melaksanakan ujian. Jika semuanya valid, maka *web service* akan mengirimkan data hasil ujian peserta ke aplikasi *client*.

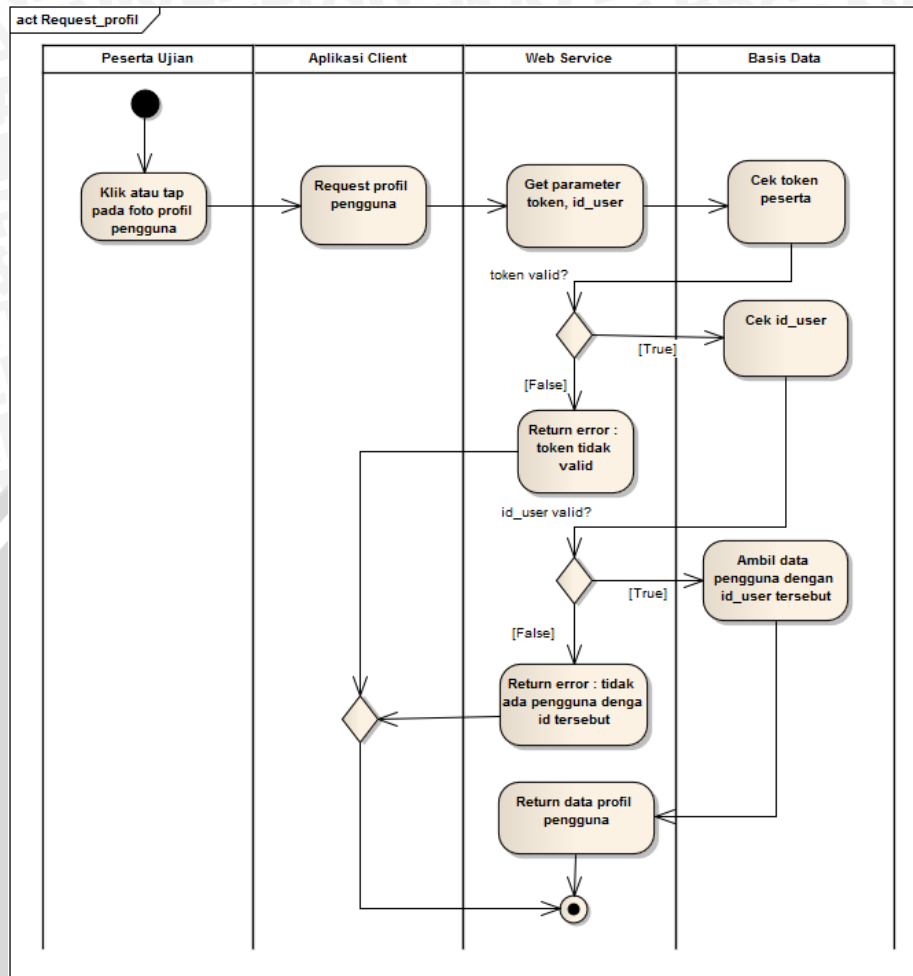
#### 4.2.2.25 Activity Diagram Batal Ujian



**Gambar 4.30 Activity Diagram Batal Ujian**

Gambar 4.30 menunjukkan interaksi antara sistem (*web service*) dan peserta ujian ketika peserta ujian ingin mengundurkan diri dari suatu ujian. Fitur ini hanya dapat berfungsi jika peserta telah resmi menjadi peserta dalam suatu ujian. Yang dimaksud resmi adalah peserta ujian telah disetujui sebagai peserta ujian dalam suatu ujian oleh Administrator yang bersangkutan. Untuk mengundurkan diri dari suatu ujian, aplikasi *client* harus mengirimkan data token dan *id\_ujian*. Sistem akan memeriksa apakah token dan ujian valid serta apakah peserta yang mengundurkan diri telah terdaftar sebagai peserta ujian. Jika semuanya valid, sistem akan menghapus peserta dari daftar peserta ujian yang bersangkutan.

#### 4.2.2.26 Activity Diagram Request Profil



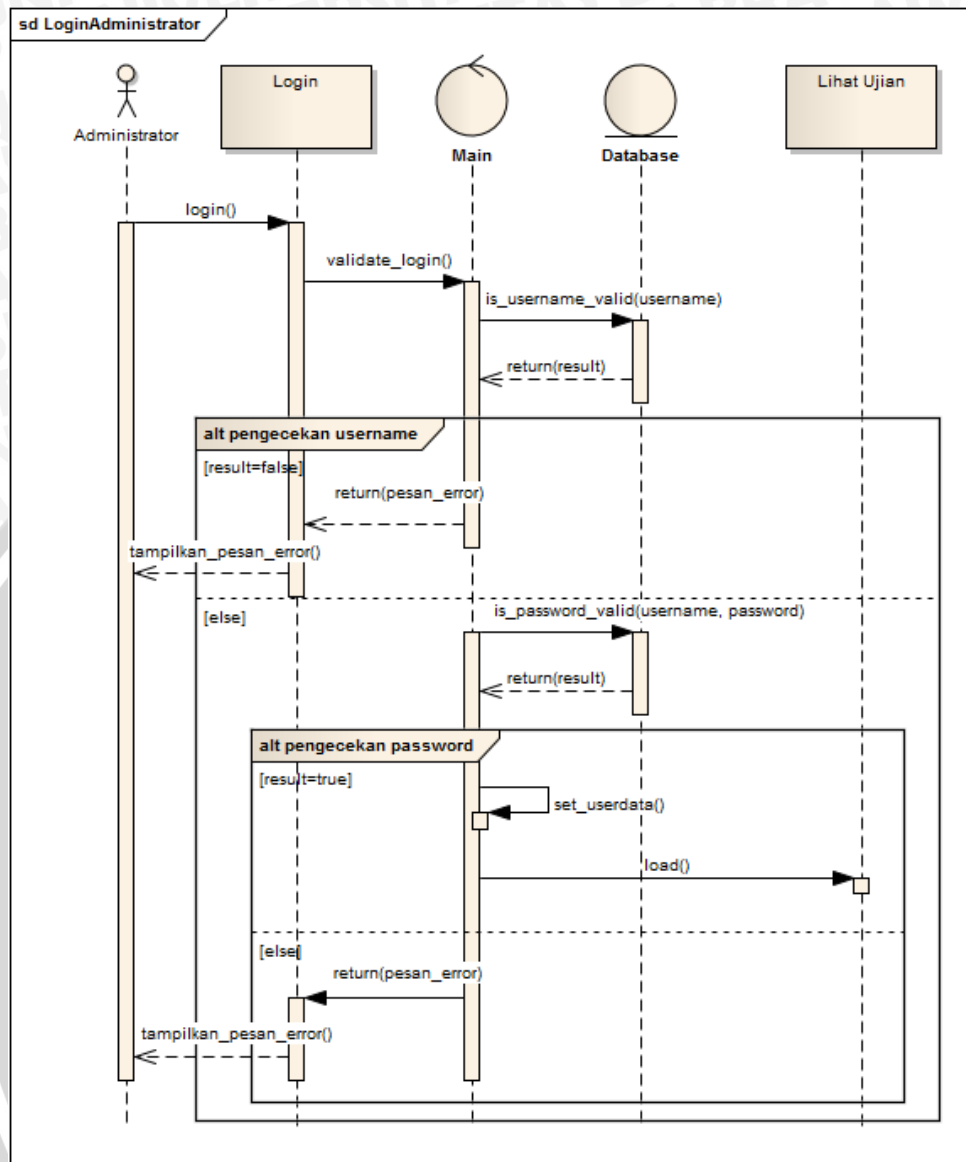
**Gambar 4.31** Activity Diagram Request Profil

Gambar 4.31 menunjukkan interaksi antara sistem (*web service*) dan peserta ujian ketika peserta ujian ingin melihat profil pengguna. Yang dimaksud dengan pengguna dalam hal ini adalah peserta ujian dan Administrator. Untuk melihat profil pengguna, maka aplikasi *client* harus mengirimkan *request* pada *web service* dengan mengirimkan token dan id pengguna. Sistem akan memeriksa token dan validitas dari id yang dikirimkan. Jika semuanya valid, maka *web service* akan mengirimkan data pengguna kepada peserta ujian.

#### 4.2.3 Perancangan *Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antara aktor dengan objek serta interaksi antar objek didalam sistem (termasuk pengguna, display dan sebagainya). *Sequence diagram* dari aplikasi dimodelkan pada gambar berikut.

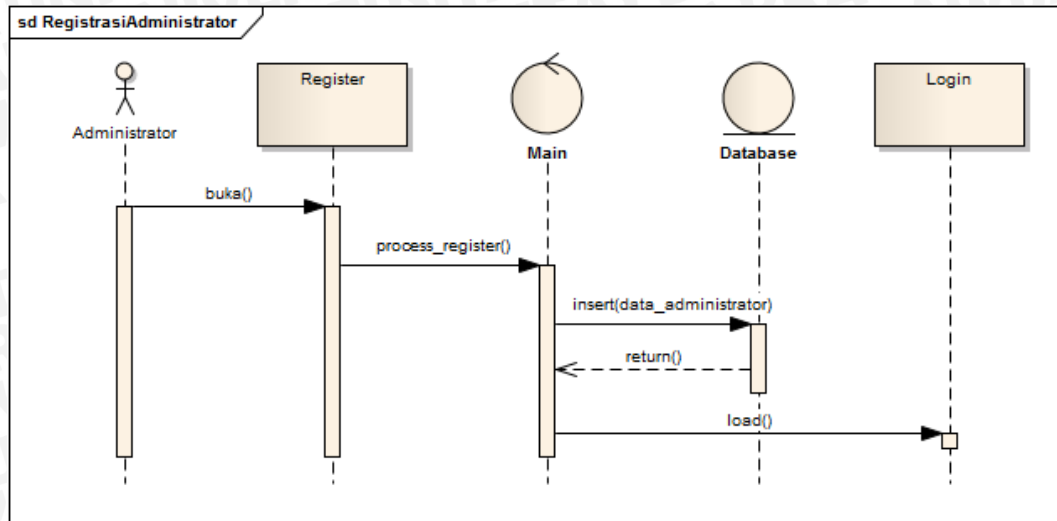
#### 4.2.3.1 Sequence Diagram Login Administrator



**Gambar 4.32 Sequence Diagram Login Administrator**

Gambar 4.32 adalah *sequence diagram* ketika *administrator* akan masuk ke dalam aplikasi. *Administrator* akan mengakses halaman login dari aplikasi. Setelah itu data *username* dan *password* dari *administrator* akan dicocokkan dengan data pada *database*. Pertama *username* akan diperiksa. Kemudian jika *username* valid, *username* dan *password* akan diperiksa kembali. Jika keduanya valid, *administrator* akan dibawa ke halaman lihat ujian.

#### 4.2.3.2 Sequence Diagram Registrasi Administrator

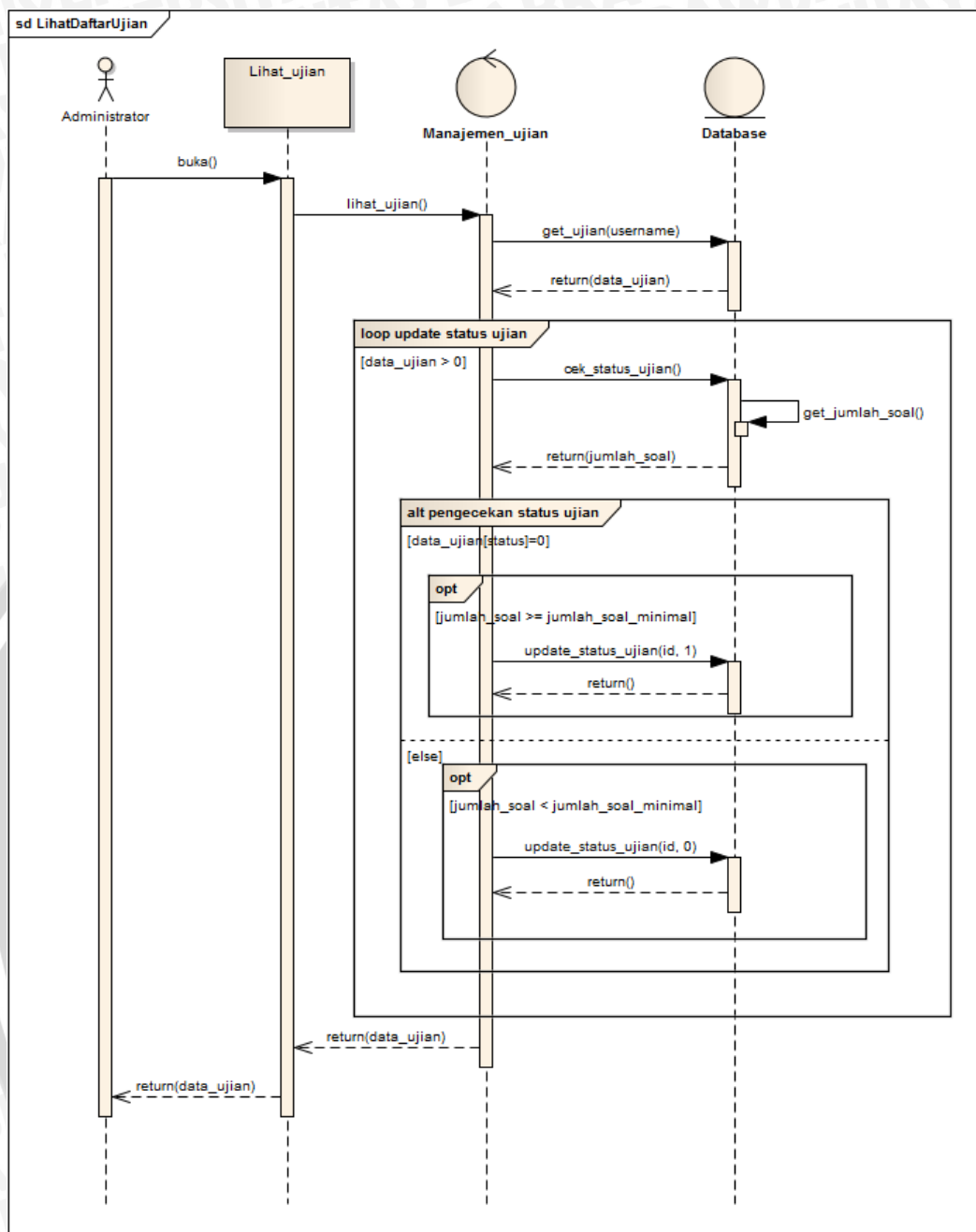


**Gambar 4.33 Sequence Diagram Registrasi Administrator**

Gambar 4.33 adalah *sequence diagram* ketika *administrator* akan mendaftar ke dalam sistem. Pertama *administrator* akan mengakses halaman registrasi dan mengisi form. Kemudian data akan diproses oleh *controller* Main dan data disimpan ke dalam *database*. *Administrator* kemudian akan dibawa ke halaman login untuk masuk ke dalam sistem menggunakan *username* dan *password* yang telah terdaftar.

#### 4.2.3.3 Sequence Diagram Lihat Daftar Ujian

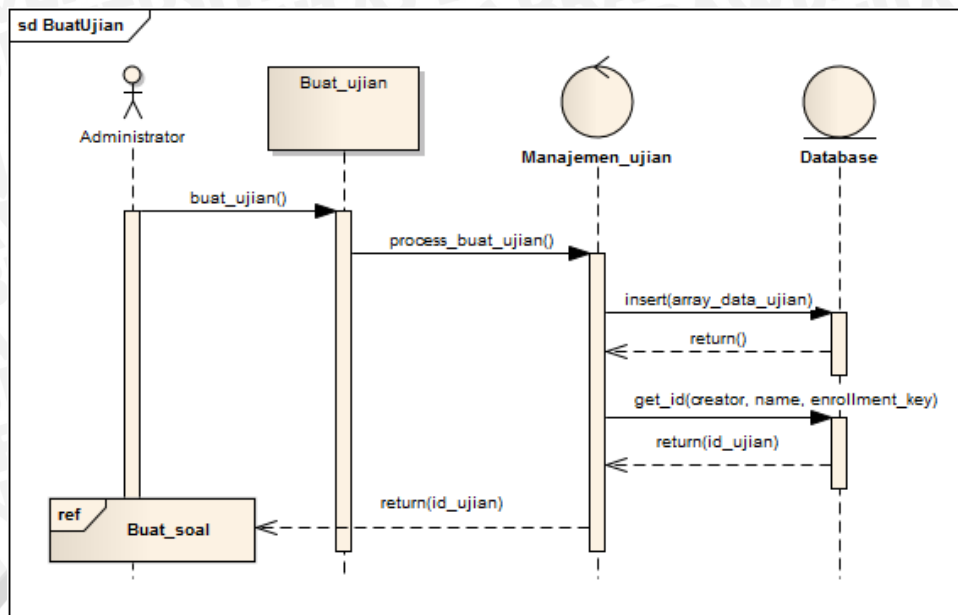
Gambar 4.34 merupakan *sequence diagram* ketika *administrator* akan melihat daftar ujian yang dikelolanya. Pertama *administrator* membuka halaman lihat ujian. *Controller* kemudian menjalankan fungsi *lihat\_ujian()* yang kemudian mengambil data dari *database*. Pada proses ini setiap ujian akan diperiksa kembali statusnya apakah telah lengkap atau belum lengkap. Jika jumlah soal pada suatu ujian telah memenuhi jumlah soal minimal untuk pelaksanaan ujian, maka status ujian akan berganti menjadi 'lengkap', begitu pula sebaliknya. Setelah pengecekan status, data ujian akan ditampilkan ke *administrator*.



Gambar 4.34 Sequence Diagram Lihat Daftar Ujian

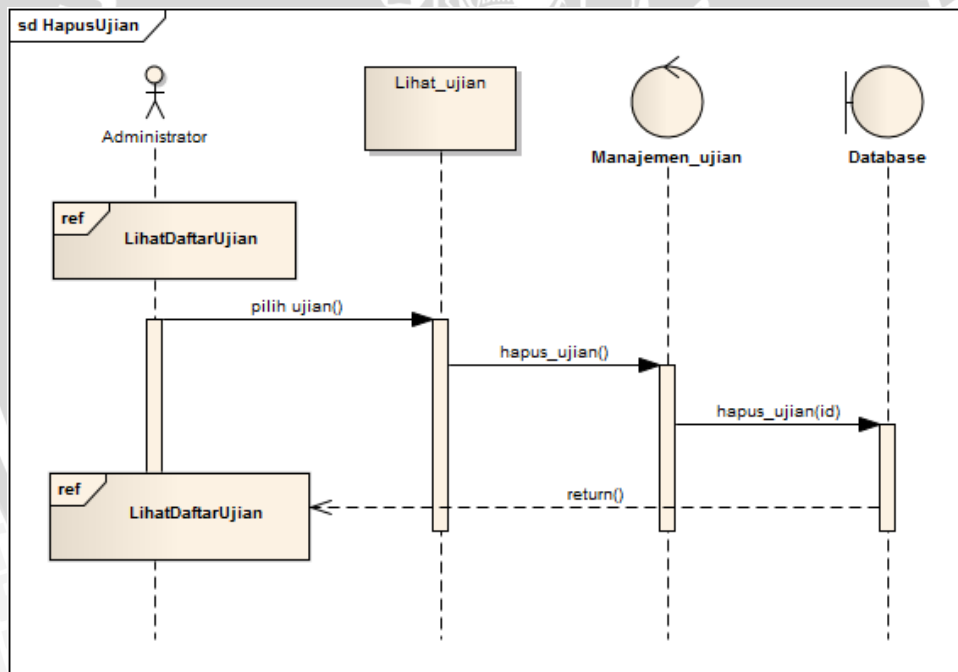
#### 4.2.3.4 Sequence Diagram Buat Ujian

Gambar 4.35 adalah *sequence diagram* ketika *administrator* akan membuat ujian baru. Setelah membuka halaman buat ujian dan mengisi form dengan lengkap dan benar, fungsi `process_buat_ujian()` akan dieksekusi dan data akan dimasukkan ke dalam *database*. Setelah itu *administrator* akan langsung dibawa ke halaman buat soal untuk membuat soal-soal dalam ujian.



Gambar 4.35 Sequence Diagram Buat Ujian

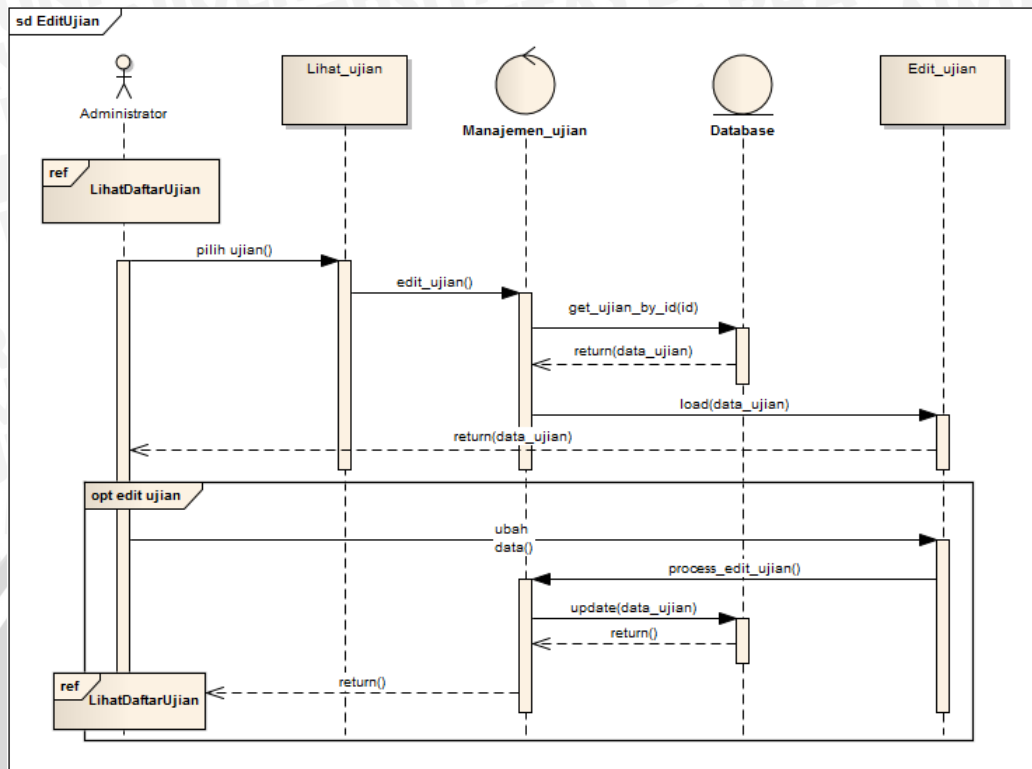
#### 4.2.3.5 Sequence Diagram Hapus Ujian



Gambar 4.36 Sequence Diagram Hapus Ujian

Gambar 4.36 merupakan *sequence diagram* ketika *administrator* ingin menghapus suatu ujian. Setelah melihat daftar ujian yang ada, *administrator* kemudian memilih ujian dan menekan tombol hapus pada halaman lihat ujian. Fungsi `hapus_ujian()` kemudian akan dijalankan dan data ujian dihapus dari *database*.

#### 4.2.3.6 Sequence Diagram Edit Ujian

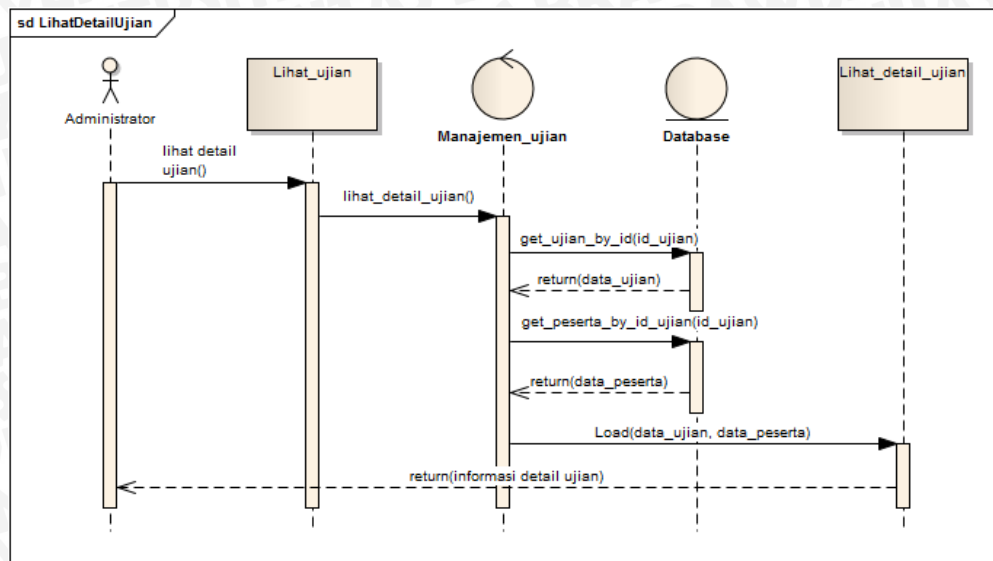


Gambar 4.37 Sequence Diagram Edit Ujian

Gambar 4.37 adalah *sequence diagram* ketika *administrator* mengubah atau mengedit ujian. Setelah melihat daftar ujian di halaman lihat ujian, *administrator* memilih ujian yang akan diedit. Fungsi `edit_ujian()` kemudian dieksekusi, data ujian akan diambil dari database dan dimuat ke halaman edit ujian. Pada halaman edit ujian, *administrator* akan mengedit ujian jika diperlukan. Jika terjadi perubahan, data yang baru akan mengganti data yang lama dalam *database*.

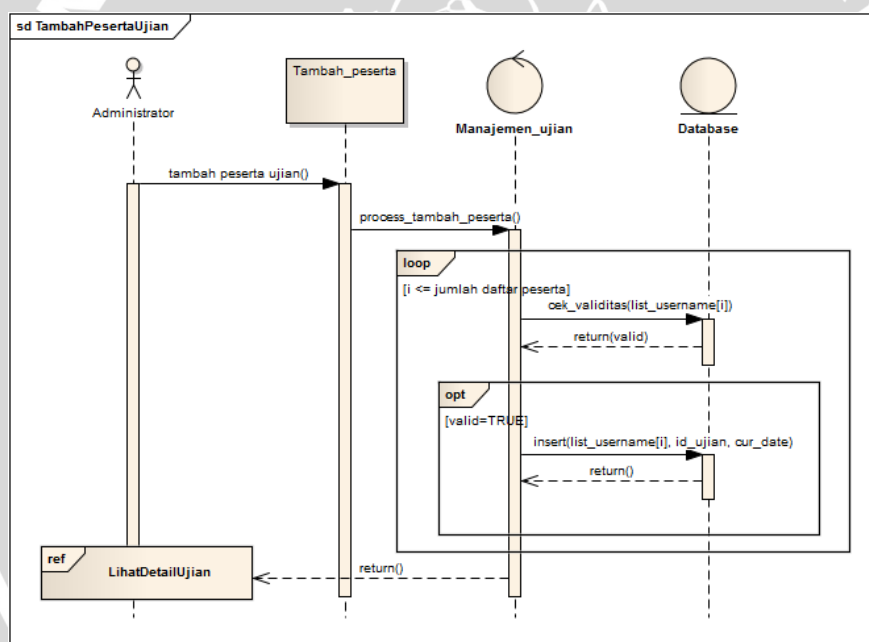
#### 4.2.3.7 Sequence Diagram Lihat Detil Ujian

Gambar 4.38 adalah *sequence diagram* ketika *administrator* akan melihat informasi detil dari suatu ujian. Setelah menekan tombol lihat detail ujian pada halaman lihat ujian, fungsi `lihat_detail_ujian()` pada *controller* Manajemen\_ujian akan dijalankan. Data ujian dan data peserta kemudian diambil dari *database* dan ditampilkan ke halaman lihat detail ujian.



Gambar 4.38 Sequence Diagram Lihat Detil Ujian

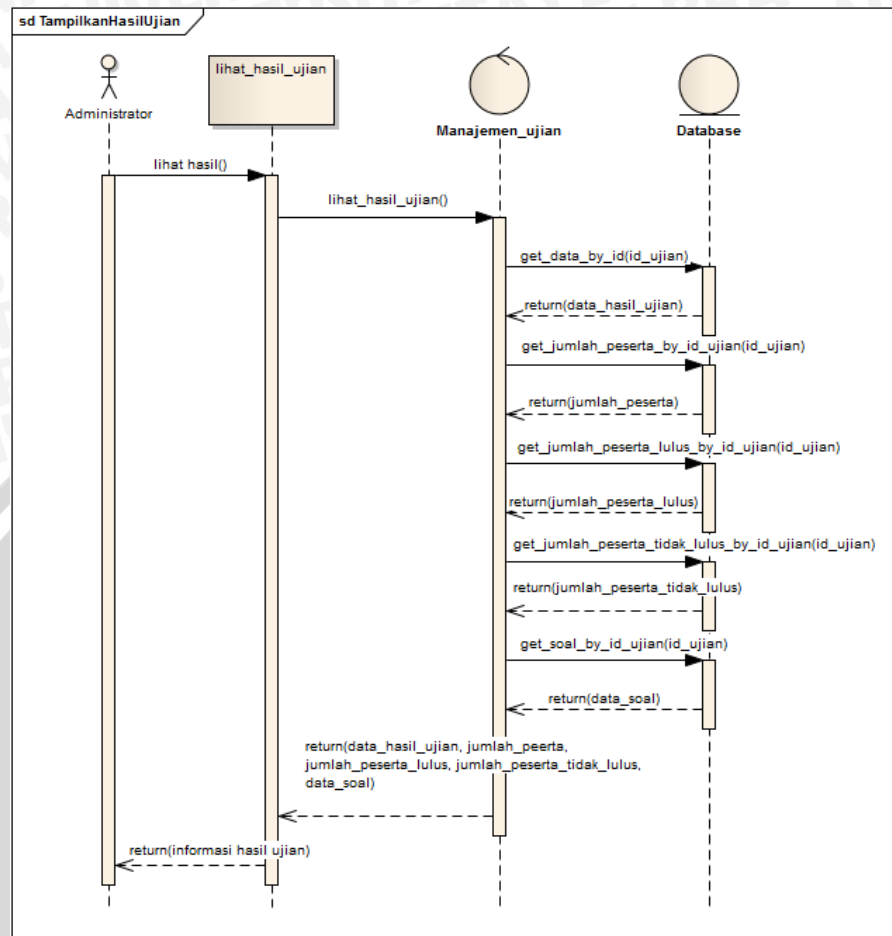
#### 4.2.3.8 Sequence Diagram Tambah Peserta Ujian



Gambar 4.39 Sequence Diagram Tambah Peserta Ujian

Gambar 4.38 merupakan *sequence diagram* ketika *administrator* akan memasukkan peserta ke dalam suatu ujian. Setelah *administrator* membuka halaman tambah peserta dan mengisi *username* peserta yang akan ditambahkan, fungsi `process_tambah_peserta()` akan dieksekusi. Setiap *username* akan diperiksa. Jika *username* valid, peserta dengan *username* tersebut akan dimasukkan ke dalam daftar peserta ujian. *Administrator* kemudian akan dibawa ke halaman lihat detail ujian.

#### 4.2.3.9 Sequence Diagram Tampilkan Hasil Ujian

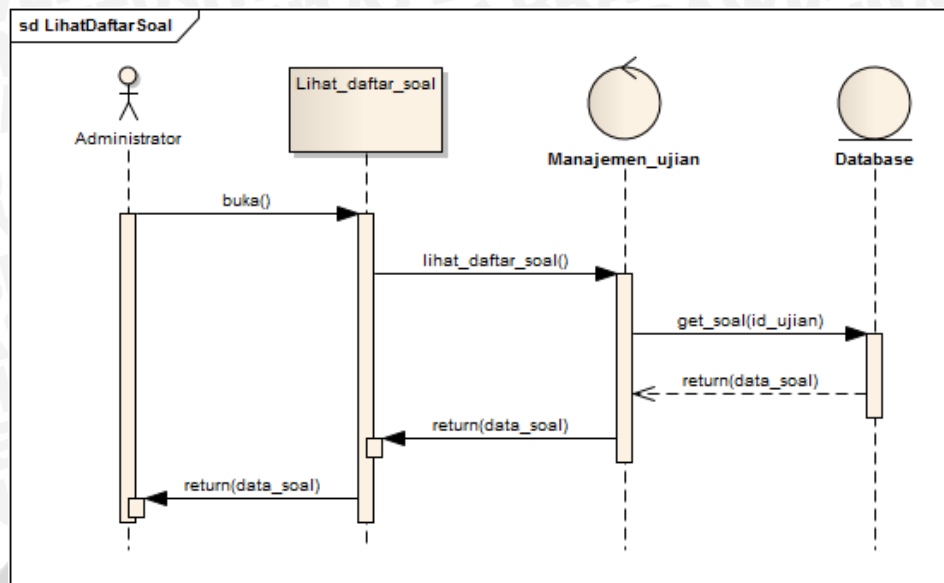


**Gambar 4.40 Sequence Diagram Tampilkan Hasil Ujian**

Gambar 4.40 merupakan *sequence diagram* ketika *administrator* akan menampilkan hasil ujian. Setelah memilih ujian dan menekan tombol lihat hasil, fungsi *lihat\_hasil\_ujian()* akan dieksekusi. Data yang diperlukan seperti data hasil ujian, data peserta dan data soal akan dimuat dan diolah kemudian ditampilkan kembali ke *administrator*.

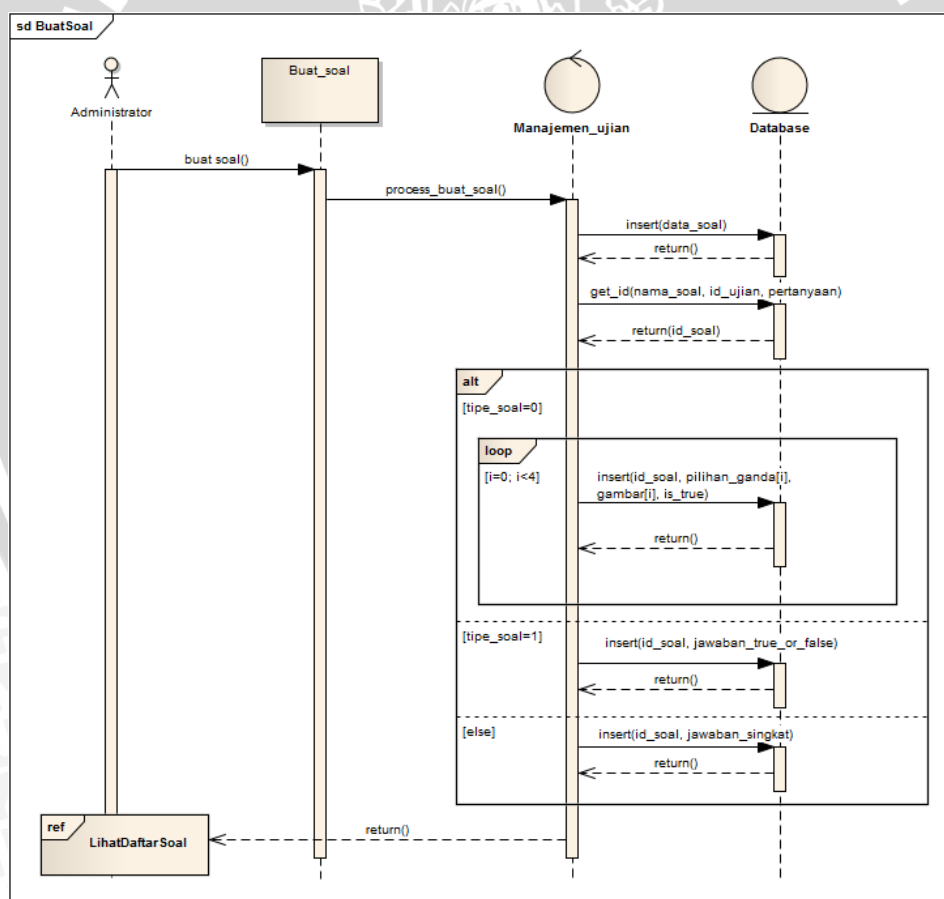
#### 4.2.3.10 Sequence Diagram Lihat Daftar Soal

Gambar 4.41 merupakan *sequence diagram* ketika *administrator* akan melihat daftar soal dalam suatu ujian. Setelah membuka halaman lihat daftar soal, data soal pada *database* akan diambil dan ditampilkan kembali ke *administrator*.



4.41 Gambar Sequence Diagram Lihat Daftar Soal

#### 4.2.3.11 Sequence Diagram Buat Soal

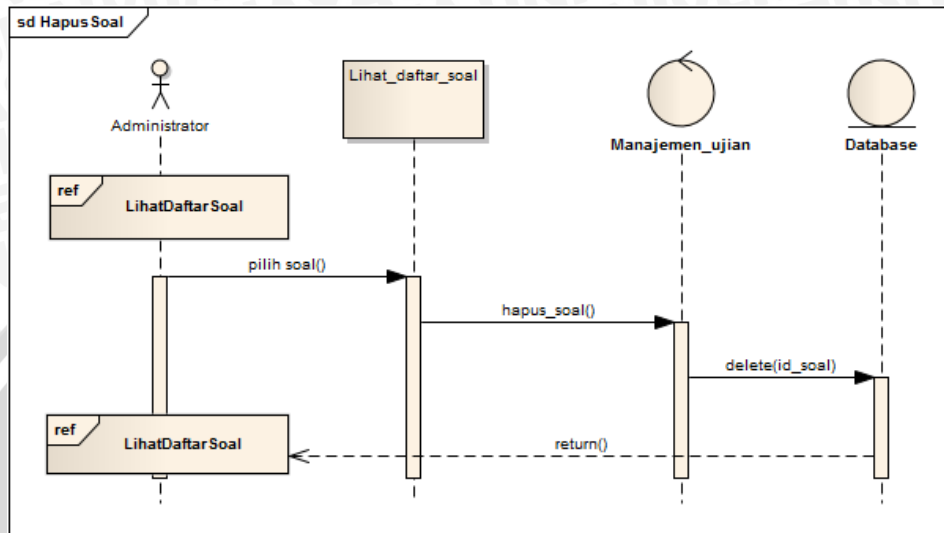


Gambar 4.42 Gambar Sequence Diagram Buat Soal

Gambar 4.42 adalah *sequence diagram* ketika *administrator* akan membuat soal baru dalam suatu ujian. Setelah menekan tombol buat soal,

halaman *buat\_soal* akan dimuat dan *administrator* harus melengkapi form. Fungsi *process\_buat\_soal()* pada *controller* Manajemen\_ujian akan dieksekusi dan data akan dimasukkan ke dalam *database*. Jawaban dari soal akan disimpan ke tabel berbeda dalam *database* berdasarkan jenis soal.

#### 4.2.3.12 Sequence Diagram Hapus Soal

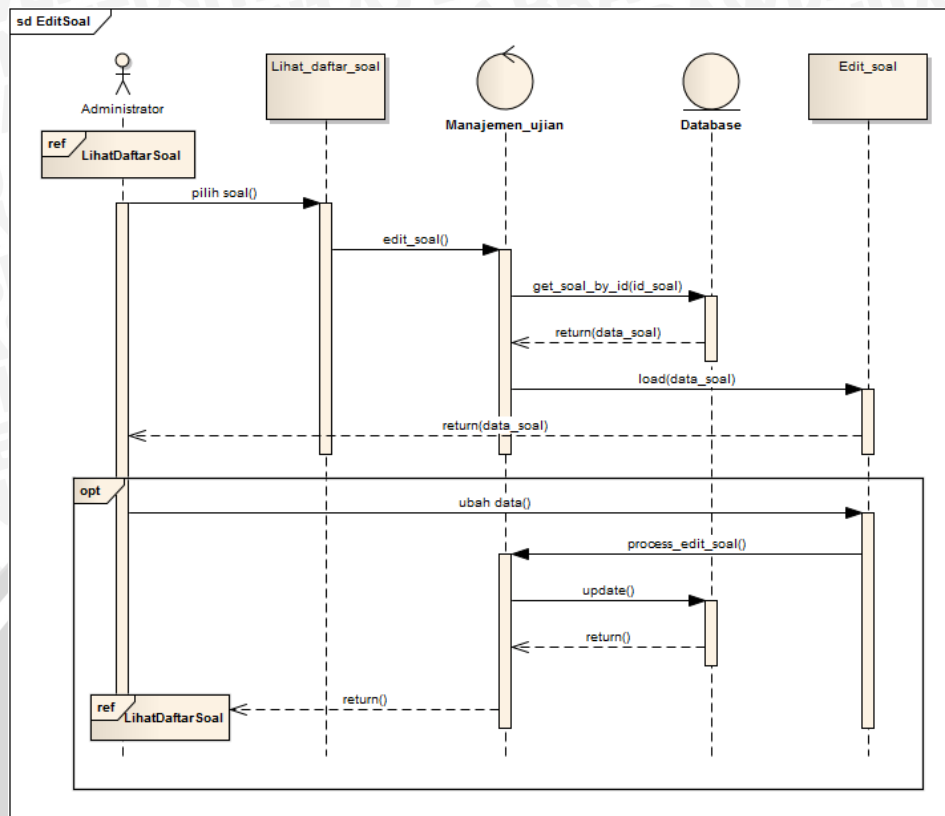


Gambar 4.43 Sequence Diagram Hapus Soal

Gambar 4.43 adalah *sequence diagram* ketika *administrator* akan menghapus soal dari suatu ujian. Setelah melihat daftar soal, *administrator* memilih soal dengan menekan tombol hapus. Fungsi *hapus\_soal()* kemudian akan dieksekusi dan data soal pada *database* akan dihapus. *Administrator* kemudian akan dibawa kembali ke halaman lihat daftar soal.

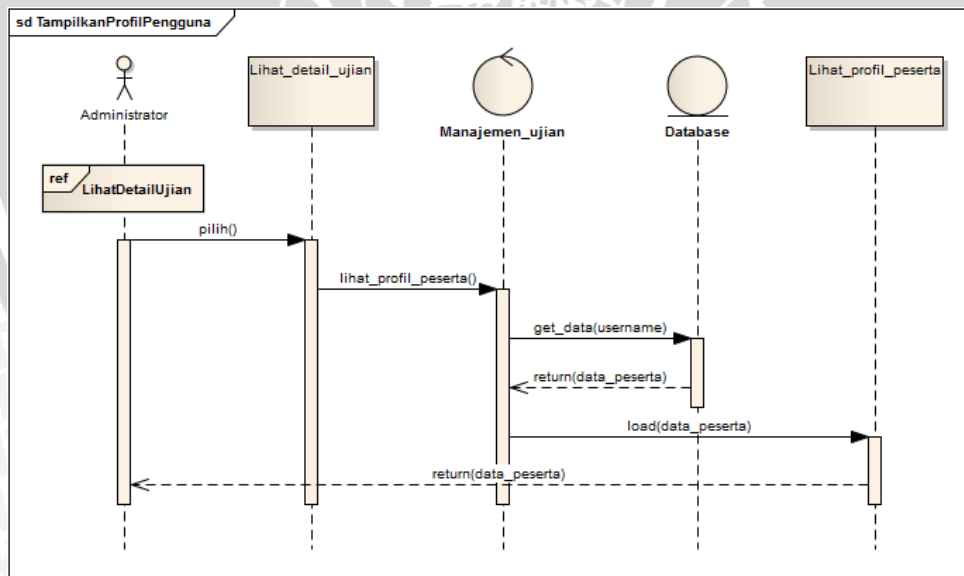
#### 4.2.3.13 Sequence Diagram Edit Soal

Gambar 4.44 adalah *sequence diagram* ketika *administrator* akan mengubah atau mengedit soal. Setelah melihat daftar soal dan memilih soal untuk diedit, fungsi *edit\_soal()* akan dieksekusi. Data soal diambil dari *database* dan dimuat ke halaman edit soal untuk ditampilkan ke *administrator*. Jika diperlukan, *administrator* dapat mengubah data soal. Jika terjadi perubahan, data soal yang baru akan disimpan ke dalam *database* menggantikan data yang lama.



Gambar 4.44 Sequence Diagram Edit Soal

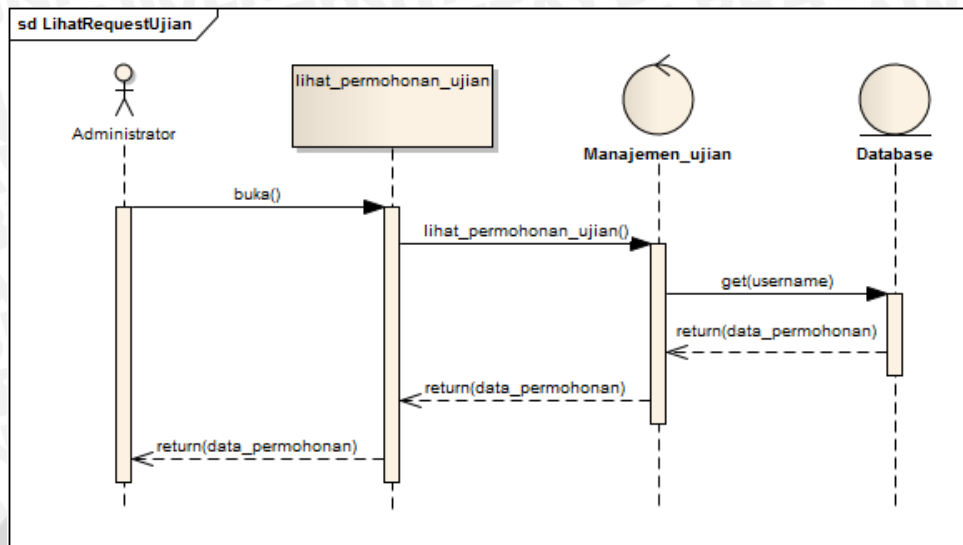
#### 4.2.3.14 Sequence Diagram Tampilkan Profil Pengguna



Gambar 4.45 Sequence Diagram Tampilkan Profil Pengguna

Gambar 4.45 adalah *sequence diagram* ketika *administrator* akan menampilkan profil pengguna atau peserta. Pada halaman lihat detail ujian, *administrator* memilih pengguna yang akan ditampilkan profilnya. Setelah itu fungsi *lihat\_profil\_peserta()* akan dieksekusi dan data peserta akan diambil dari *database* dan ditampilkan kembali ke *administrator*.

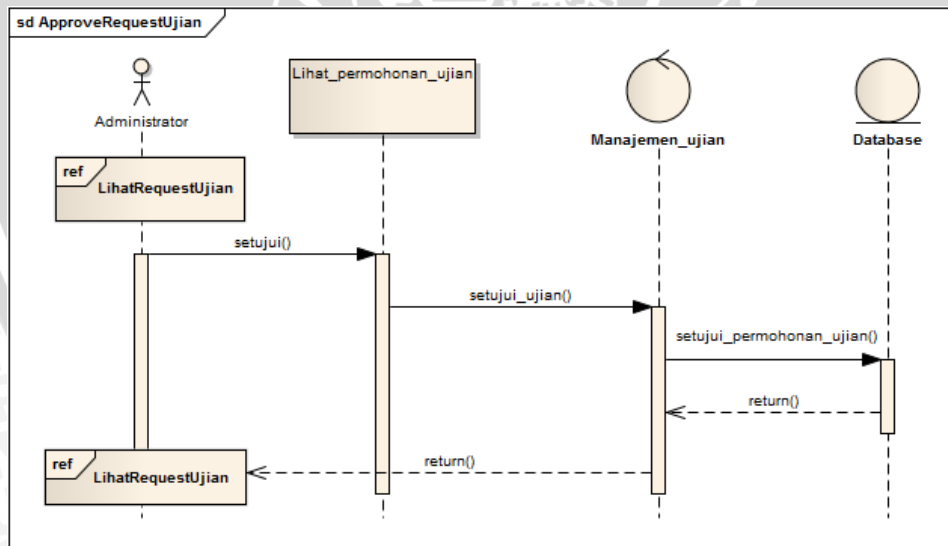
#### 4.2.3.15 Sequence Diagram Lihat Request Ujian



**Gambar 4.46 Sequence Diagram Lihat Request Ujian**

Gambar 4.46 adalah *sequence diagram* ketika *administrator* akan melihat permohonan ujian yang dikirimkan oleh peserta. *Administrator* membuka halaman lihat permohonan ujian dan fungsi *lihat\_permohonan\_ujian()* akan dijalankan. Data diambil dari *database* dan ditampilkan kembali ke *administrator* pada halaman lihat permohonan ujian.

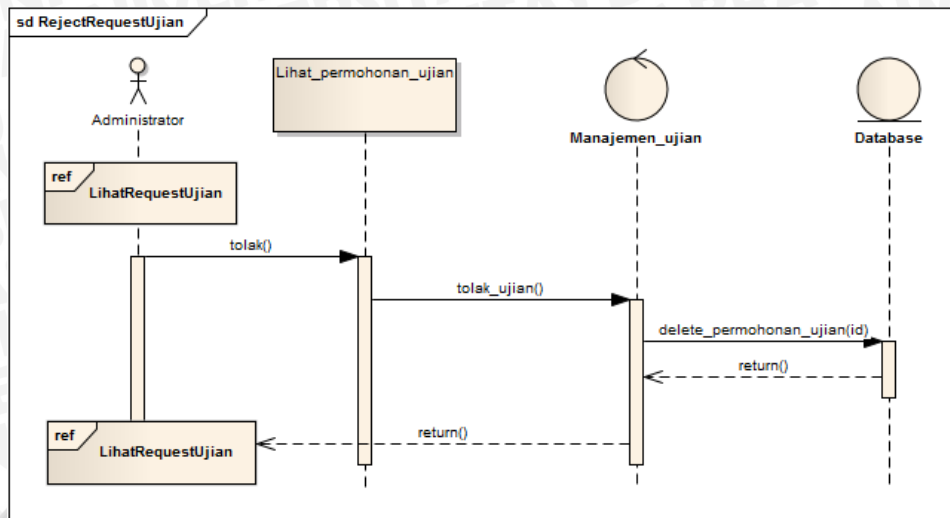
#### 4.2.3.16 Sequence Diagram Approve Request Ujian



**Gambar 4.47 Sequence Diagram Approve Request Ujian**

Gambar 4.47 merupakan *sequence diagram* ketika *administrator* akan menyetujui permohonan ujian dari peserta. Setelah melihat daftar permohonan, pada halaman lihat permohonan ujian *administrator* menekan tombol setuju. Kemudian fungsi *setujui\_ujian()* pada *controller* manajemen\_ujian akan dieksekusi dan data disimpan dalam *database*.

#### 4.2.3.17 Sequence Diagram Reject Request Ujian

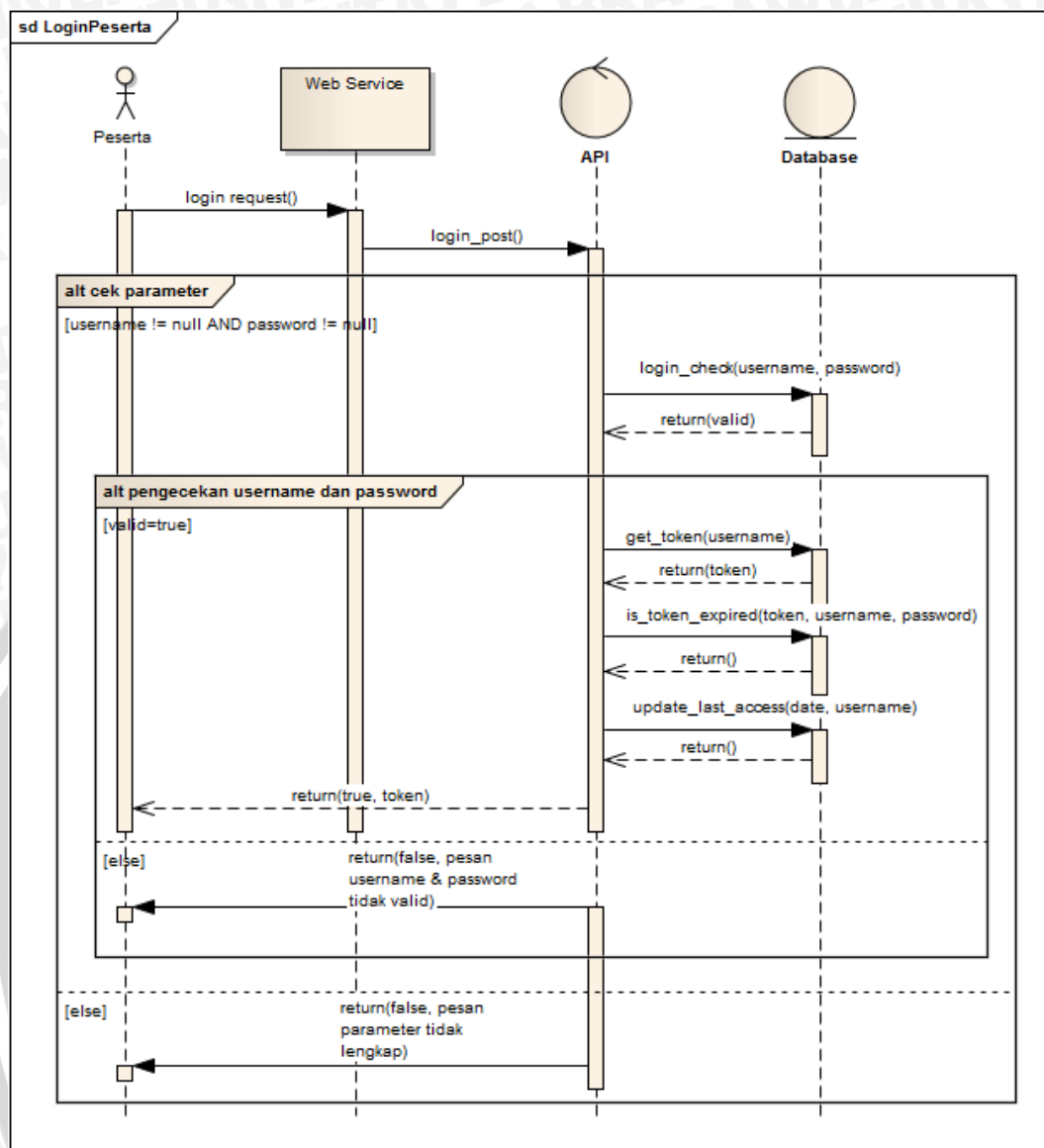


**Gambar 4.48 Sequence Diagram Reject Request Ujian**

Gambar 4.48 adalah *sequence diagram* ketika *administrator* akan menolak permohonan ujian dari peserta. Setelah melihat daftar permohonan, pada halaman lihat permohonan ujian *administrator* menekan tombol tolak. Kemudian fungsi `tolak_ujian()` pada *controller* manajemen\_ujian akan dieksekusi dan data disimpan dalam *database*.

#### 4.2.3.18 Sequence Diagram Login Peserta

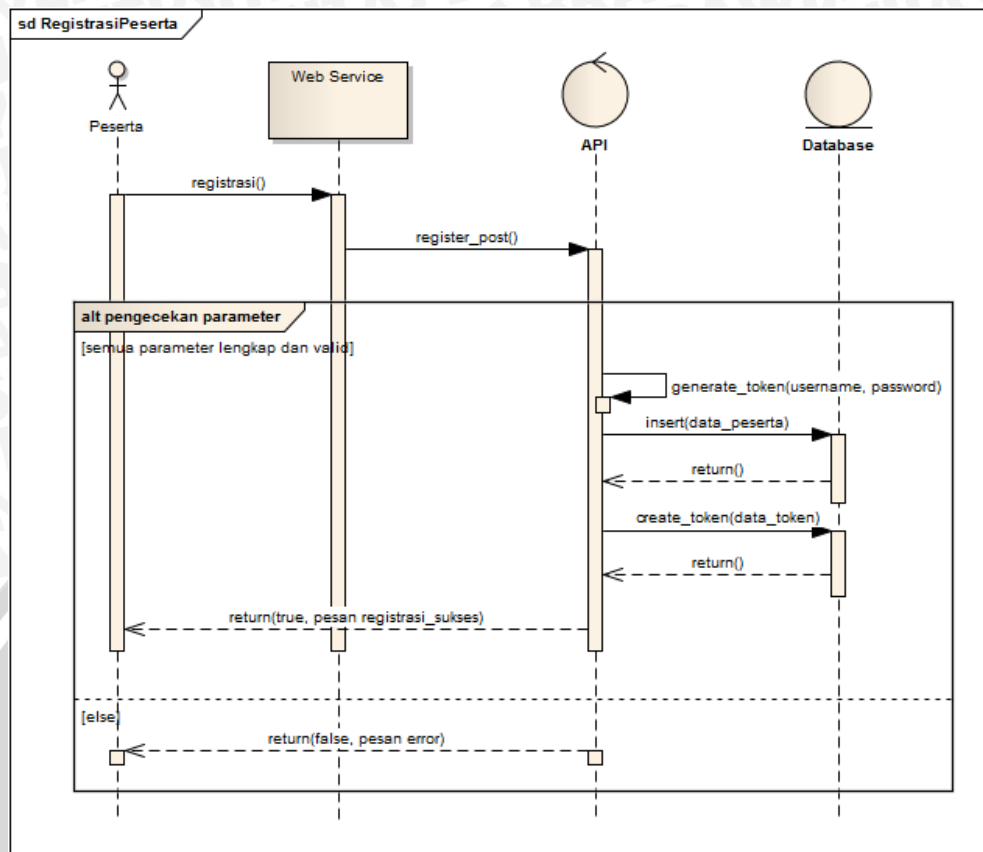
Gambar 4.49 adalah *sequence diagram* ketika peserta akan login ke dalam sistem. Peserta menggunakan aplikasi *client* melakukan *request* ke *web service* dengan mengirimkan *username* dan *password*. Jika keduanya lengkap dan valid, data token dari *database* akan diambil dan fungsi `is_token_expired()` akan memeriksa apakah token telah kadaluarsa atau belum. Jika iya, maka token yang baru akan dibuat dan disimpan dalam *database*. Data token kemudian akan dikirimkan kembali ke peserta ujian dalam bentuk JSON.



Gambar 4.49 Sequence Diagram Login Peserta

#### 4.2.3.19 Sequence Diagram Registrasi Peserta

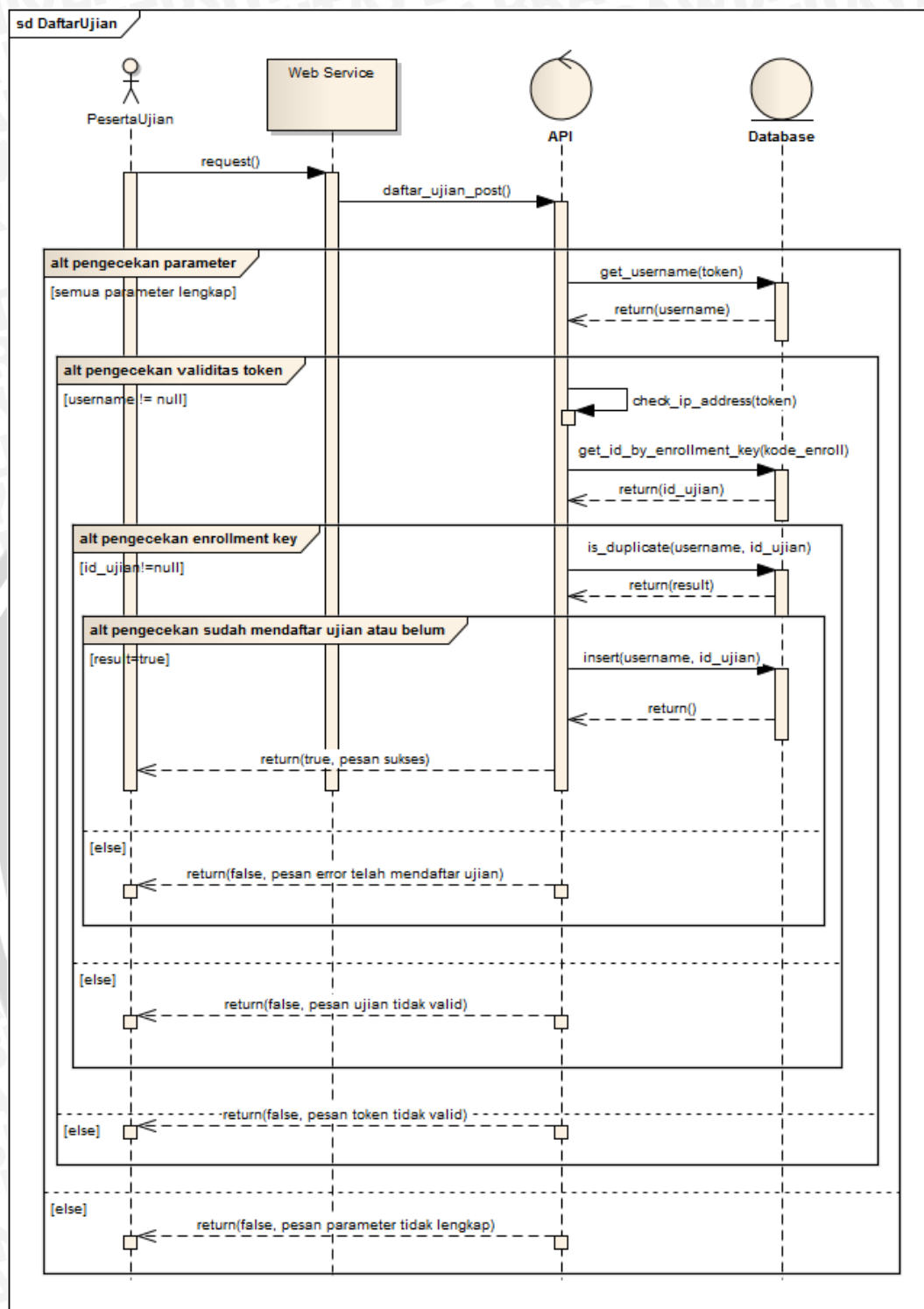
Gambar 4.50 adalah *sequence diagram* ketika peserta akan mendaftar ke dalam sistem. Peserta melakukan *request* ke *web service*. *Controller API* kemudian akan mengeksekusi fungsi *register\_post()*. Parameter dari *request* akan diperiksa. Jika lengkap, maka fungsi *generate\_token()* yang berfungsi untuk membuat token baru akan dijalankan. Token kemudian disimpan dalam database dan dikembalikan ke peserta dalam bentuk JSON.



Gambar 4.50 Sequence Diagram Registrasi Peserta

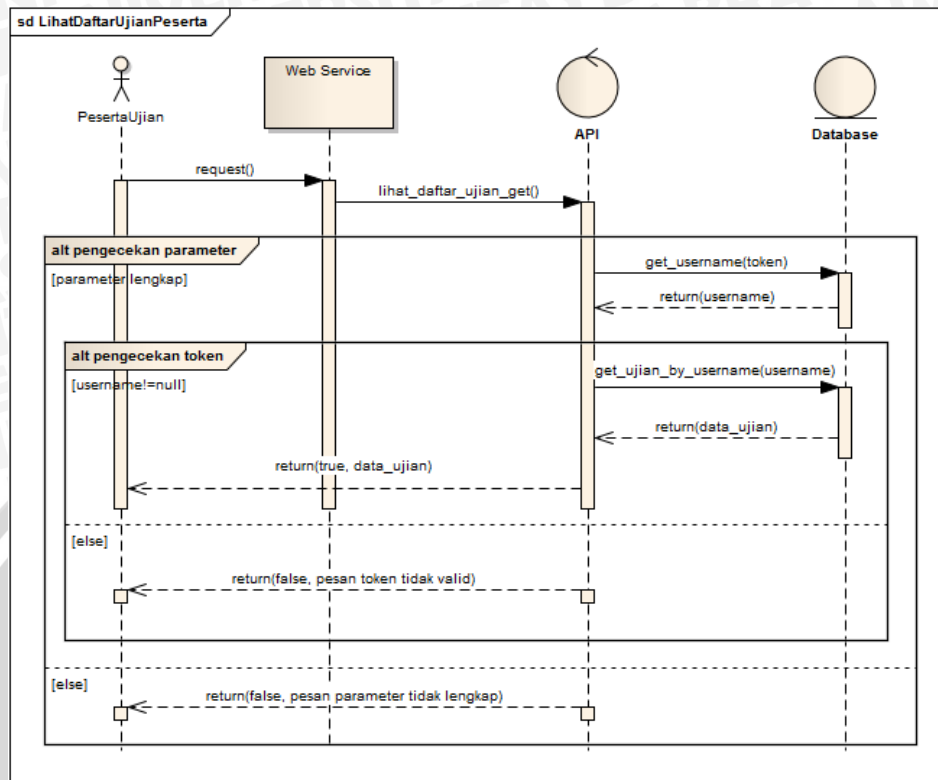
#### 4.2.3.20 Sequence Diagram Daftar Ujian

Gambar 4.51 adalah *sequence diagram* ketika peserta akan mendaftar ke dalam suatu ujian. Peserta mengirimkan *request* ke *web service* dengan parameter yang dibutuhkan. Fungsi *daftar\_ujian\_post()* kemudian akan dieksekusi dan dilakukan pengecekan parameter, pengecekan validitas token, pengecekan validitas *enrollment key* dan pengecekan apakah peserta telah mengajukan permohonan sebelumnya atau belum. Jika semua terpenuhi, maka fungsi *insert()* akan dieksekusi untuk menyimpan data ke dalam *database*. Data dan pesan sukses kemudian akan dikembalikan ke peserta dalam bentuk JSON.



Gambar 4.51 Sequence Diagram Daftar Ujian

#### 4.2.3.21 Sequence Diagram Lihat Daftar Ujian Peserta

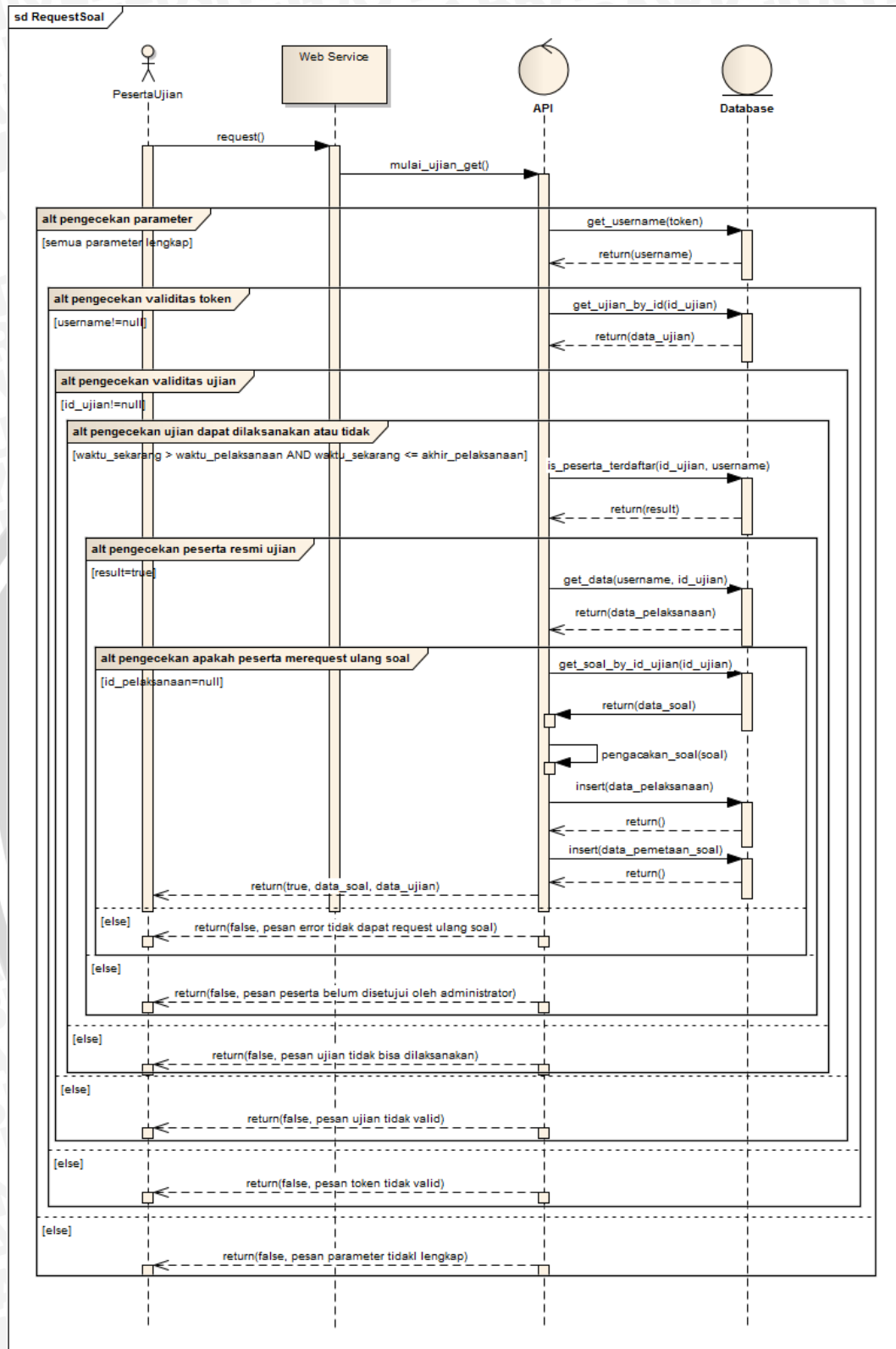


**Gambar 4.52 Sequence Diagram Lihat Daftar Ujian Peserta**

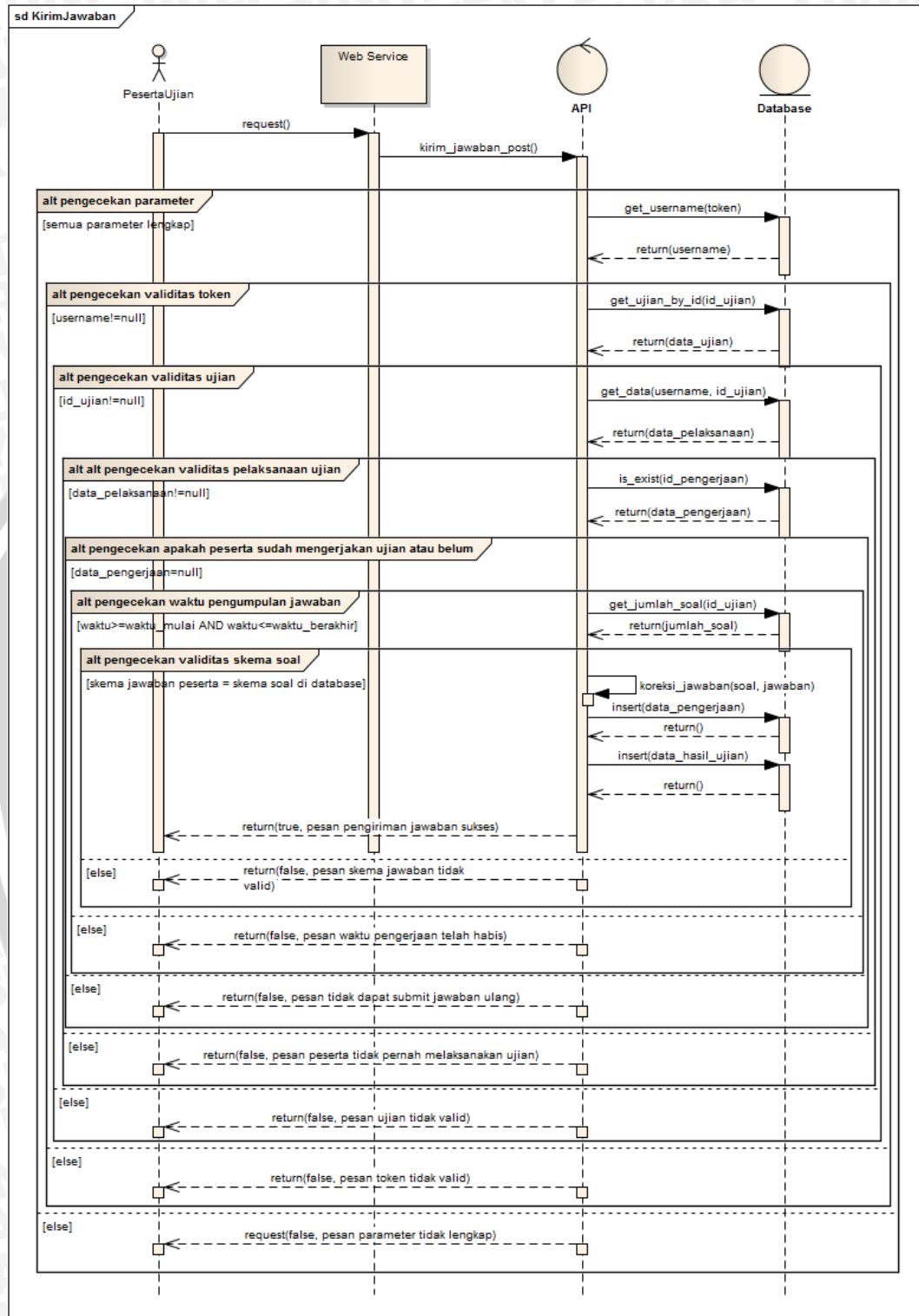
Gambar 4.52 adalah *sequence diagram* ketika peserta akan melihat daftar ujian yang diikuti. Peserta melakukan *request* ke *web service*. *Controller API* kemudian akan mengeksekusi fungsi *lihat\_daftar\_ujian\_get()*. Kelengkapan parameter dan validitas token dari *request* peserta diperiksa. Jika keduanya terpenuhi, data ujian akan dikembalikan dalam bentuk JSON.

#### 4.2.3.22 Sequence Diagram Request Soal

Gambar 4.53 adalah *sequence diagram* ketika peserta akan memulai suatu ujian. Pada proses ini, peserta melakukan *request* soal ke *web service*. Fungsi *mulai\_ujian\_get()* akan dieksekusi dan pengecekan parameter akan dilakukan. Kemudian dilakukan pengecekan validitas token, pengecekan validitas ujian, pengecekan apakah ujian dapat dilaksanakan, pengecekan apakah peserta adalah peserta ujian yang terdaftar dan pengecekan apakah peserta telah melakukan *request* soal sebelumnya. Jika semua kondisi terpenuhi, fungsi *get\_soal\_by\_id\_ujian()* akan mengambil data dari *database* dan kemudian diacak oleh fungsi *pengacakan\_soal()*. Sebelum respon dikirim, data pelaksanaan dan data pemetaan soal disimpan dalam *database*. Barulah kemudian respon berisi data soal dan data ujian dikirimkan kembali ke peserta dalam bentuk JSON.



#### 4.2.3.23 Sequence Diagram Kirim Jawaban

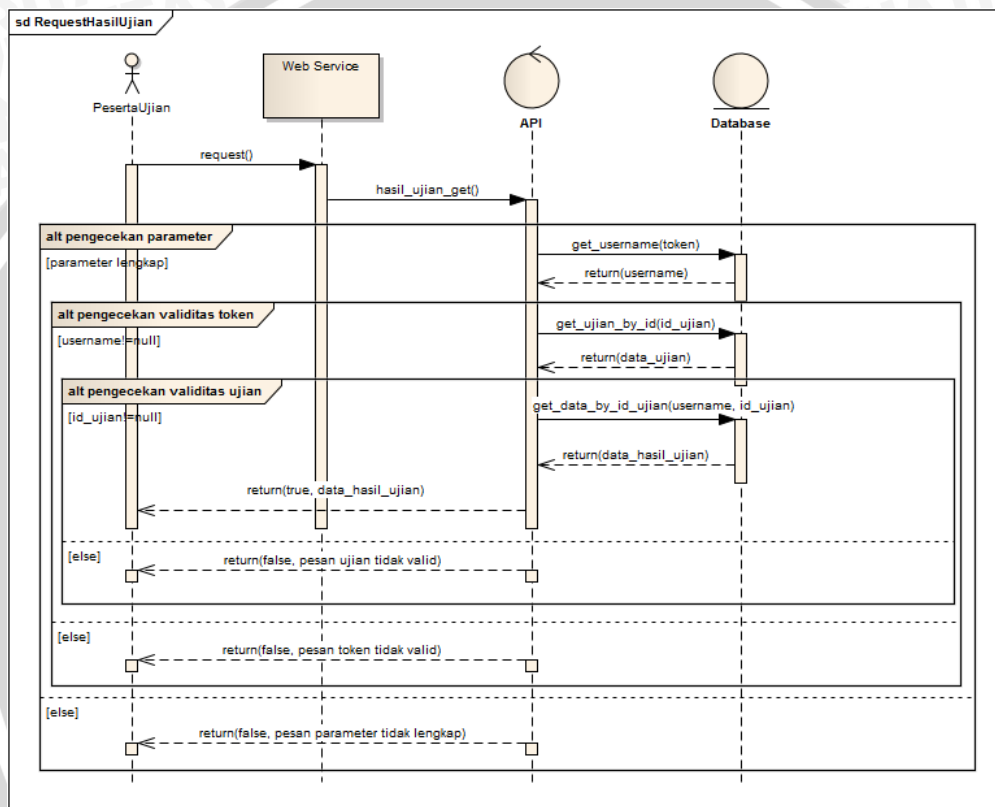


**Gambar 4.54 Sequence Diagram Kirim Jawaban**

Gambar 4.54 adalah *sequence diagram* ketika peserta melakukan pengiriman jawaban ke *web service*. *Controller API* akan mengeksekusi fungsi `kirim_jawaban_post()`. Kemudian akan terjadi beberapa kali pengecekan yaitu

pengecekan parameter, pengecekan validitas token, pengecekan validitas ujian, pengecekan validitas waktu pelaksanaan ujian, pengecekan apakah peserta telah mengumpulkan jawaban sebelumnya, pengecekan waktu pengumpulan jawaban dan pengecekan skema jawaban dari peserta. Jika semua kondisi terpenuhi, maka fungsi `koreksi_jawaban()` yang berfungsi untuk mengoreksi jawaban peserta akan dieksekusi. Setelah itu data pengerjaan dan hasil disimpan ke dalam database. Respon kemudian akan dikirimkan dalam bentuk JSON yang berisi pesan bahwa pengiriman jawaban telah berhasil.

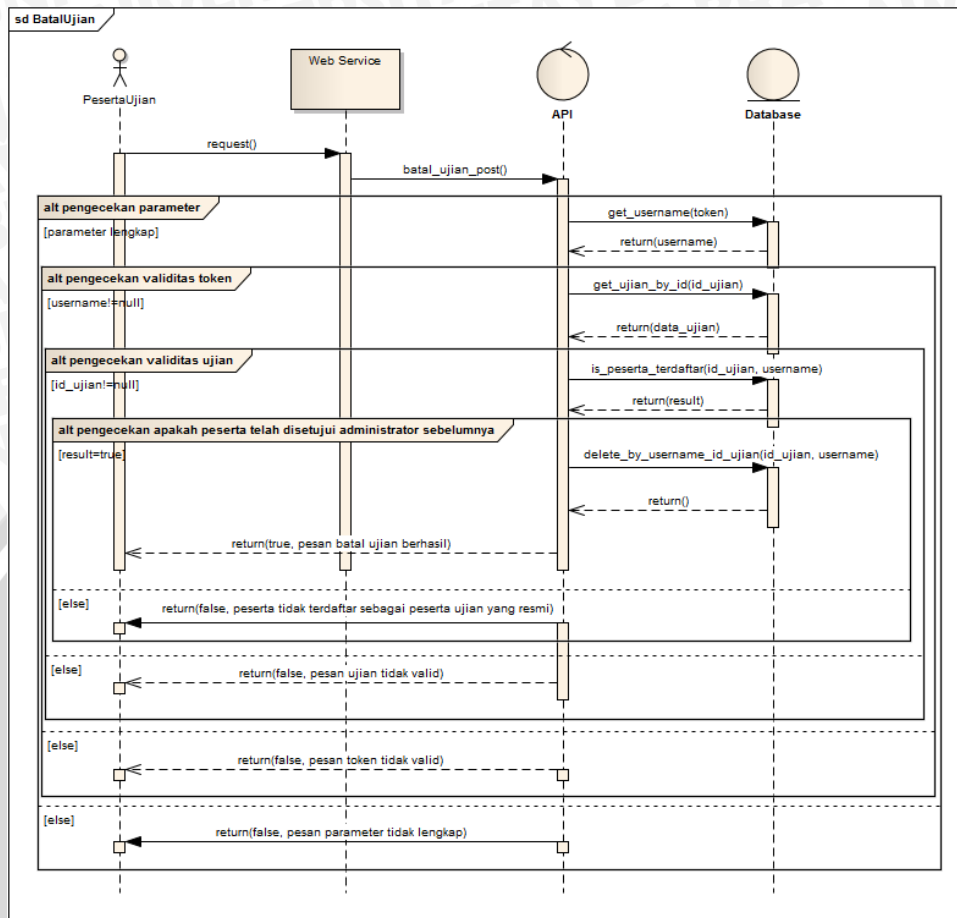
#### 4.2.3.24 Sequence Diagram Request Hasil Ujian



**Gambar 4.55 Sequence Diagram Request Hasil Ujian**

Gambar 4.55 adalah *sequence diagram* ketika peserta ingin melihat hasil ujian. Peserta akan melakukan *request* ke *web service*. Fungsi `hasil_ujian_get()` dijalankan dan akan terjadi pengecekan parameter, pengecekan validitas token dan pengecekan validitas ujian. Jika semua kondisi terpenuhi, maka fungsi `get_data_by_id_ujian()` akan mengambil data dari *database* dan akan dikembalikan ke peserta dalam format JSON.

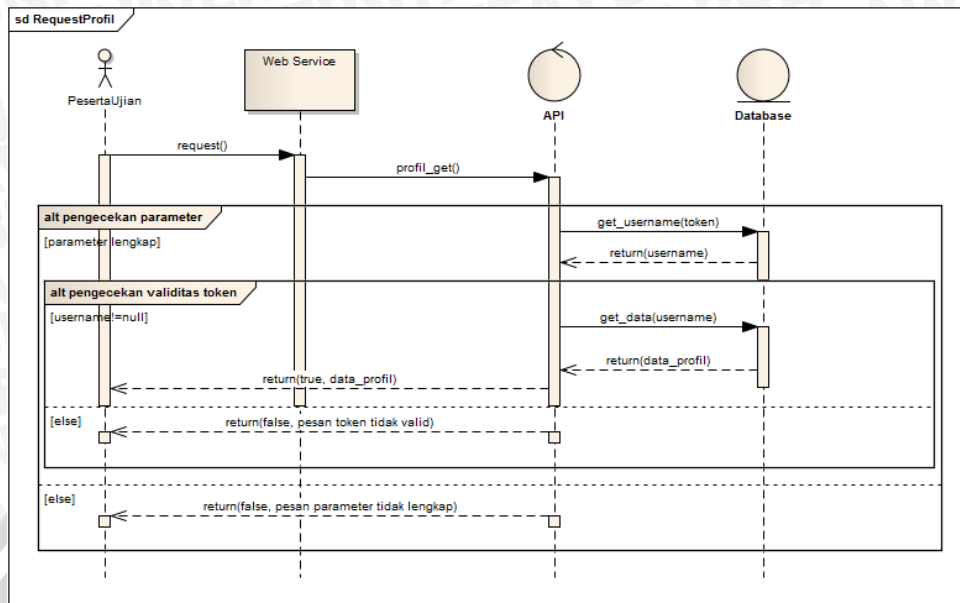
#### 4.2.3.25 Sequence Diagram Batal Ujian



**Gambar 4.56 Sequence Diagram Batal Ujian**

Gambar 4.56 adalah *sequence diagram* ketika peserta akan membatalkan ujian. Pertama peserta akan melakukan *request* ke *web service* yang kemudian akan menjalankan fungsi *batal\_ujian\_post()*. Kemudian terjadi pemeriksaan parameter, validitas token, validitas ujian dan pengecekan apakah peserta benar-benar telah menjadi peserta resmi ujian yang akan dibatalkan. Jika semua kondisi terpenuhi, maka fungsi *delete\_by\_username\_id\_ujian()* akan menghapus nama peserta dari daftar peserta ujian di *database*.

#### 4.2.3.26 Sequence Diagram Request Profil

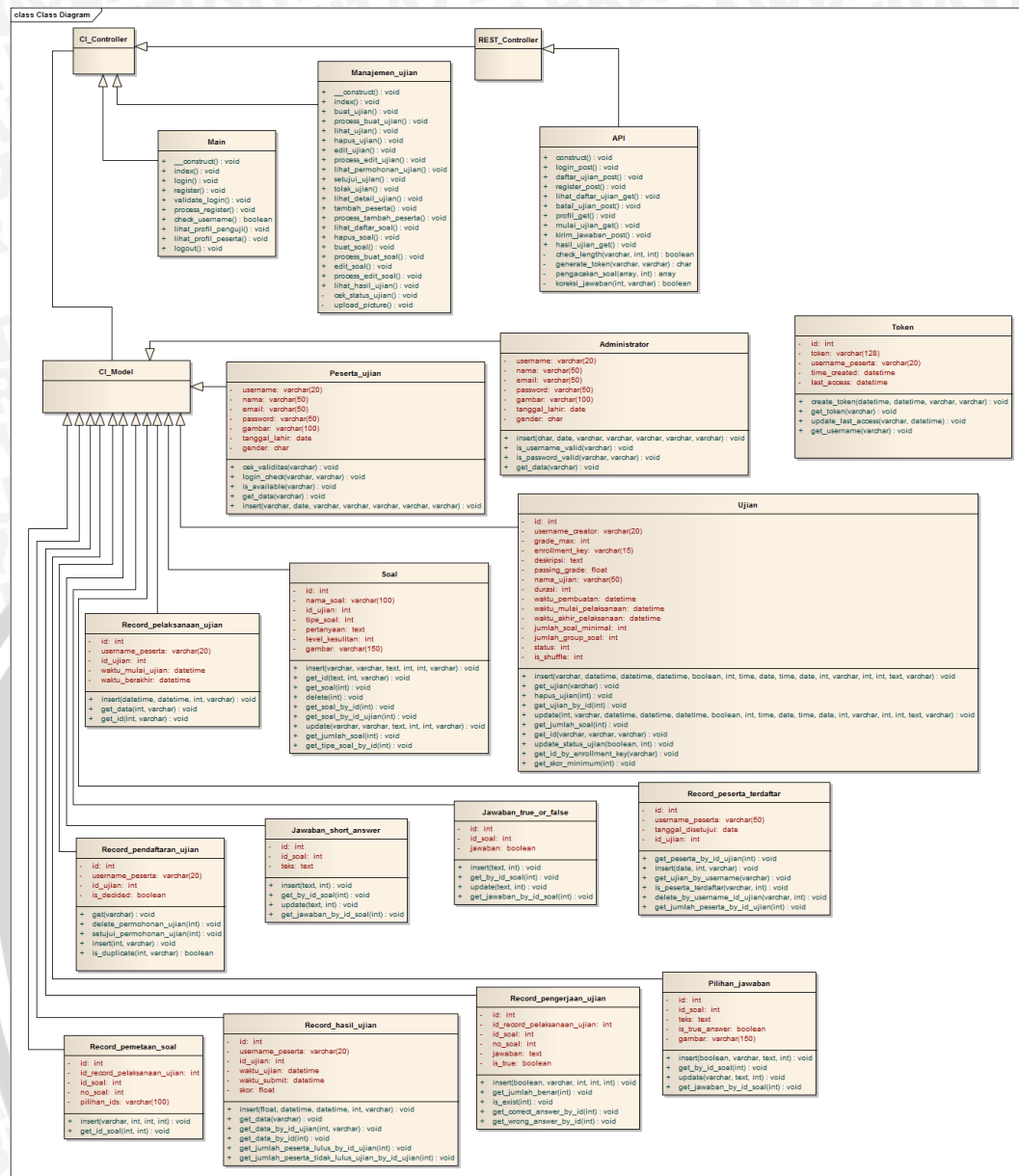


**Gambar 4.57 Sequence Diagram Request Profil**

Gambar 4.57 adalah *sequence diagram* ketika peserta akan melihat profil dirinya. Peserta akan melakukan *request* ke *web service* menggunakan parameter token. Fungsi *profil\_get()* akan dijalankan dan kelengkapan token dan validitas token akan diperiksa. Jika semua kondisi terpenuhi, maka fungsi *get\_data()* akan mengambil data peserta dari *database* dan *controller* akan mengirimkannya kembali ke peserta dalam bentuk JSON.

#### 4.2.4 Perancangan Class Diagram

*Class diagram* adalah diagram yang digunakan untuk menampilkan kelas-kelas yang ada pada perangkat lunak. Diagram kelas juga menunjukkan hubungan antar kelas dalam sistem. Gambar 4.32 merupakan gambar *class diagram* pada Aplikasi *Backend Sistem Ujian Online Terpadu*.



Gambar 4.32 Class Diagram

#### 4.2.5 Perancangan Basis Data

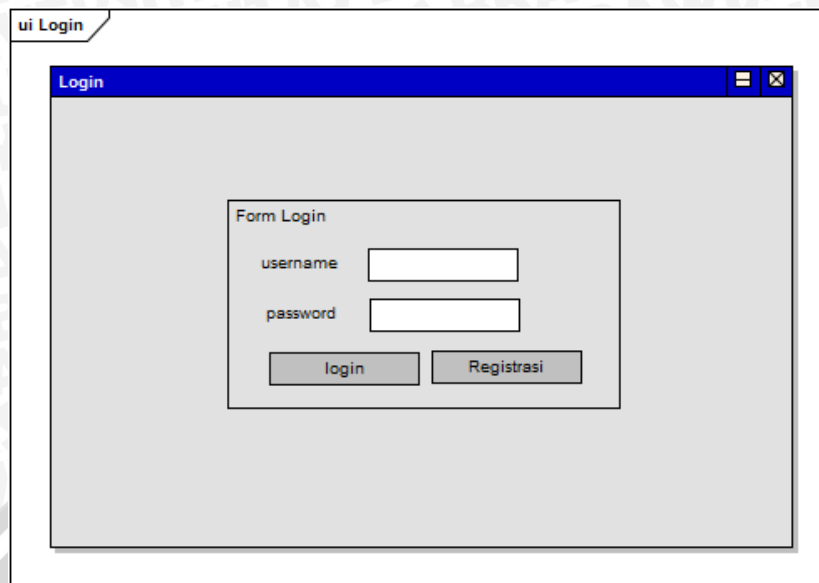
Basis data digunakan untuk menyimpan data-data ujian online terpadu seperti data peserta, data soal dan lain-lain. Oleh karena itu, perancangan yang baik sangat diperlukan agar aplikasi dapat berjalan dengan baik. Diagram ini merepresentasikan struktur sistem yang terdiri dari entitas-entitas yang saling berelasi.

Gambar 4.33 merupakan gambar dari pemodelan basis data aplikasi *backend* sistem ujian online terpadu yang menjelaskan tentang relasi antar entitas dalam sistem.



Bagian ini menjelaskan mengenai bagaimana rancangan antarmuka aplikasi *backend* sistem ujian online terpadu. Aplikasi *Backend* Sistem Ujian Online Terpadu secara garis besar dibagi menjadi dua bagian, yaitu aplikasi berbasis web yang digunakan oleh Administrator dan *web service* yang digunakan untuk menangani *request* dari peserta. Antarmuka yang dirancang pada bagian ini hanya aplikasi berbasis web yang digunakan oleh Administrator.

Login merupakan salah satu antarmuka yang berfungsi sebagai halaman login Administrator ke dalam sistem. Untuk login, Administrator harus memasukkan username dan password. Gambar 4.34 adalah antarmuka halaman login

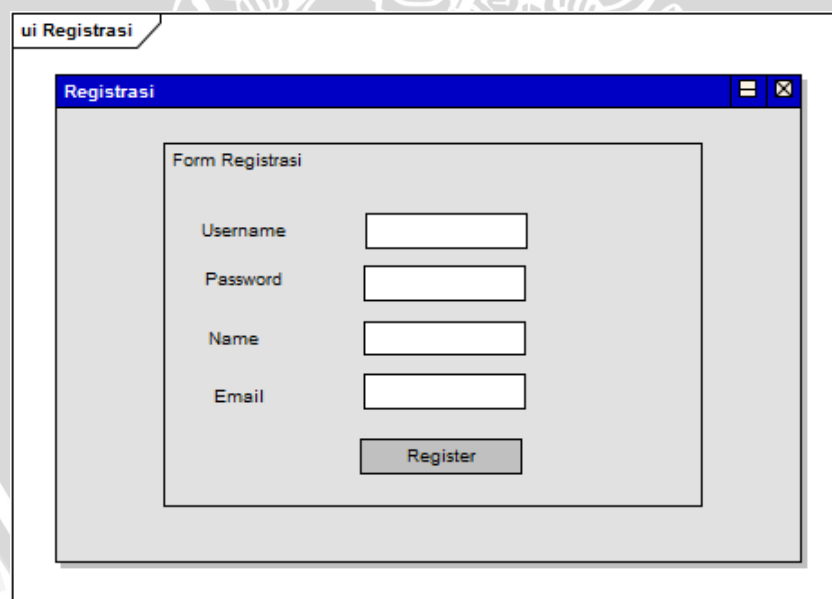


The screenshot shows a web application window titled "Login". Inside the window, there is a section titled "Form Login". This section contains two text input fields, one labeled "username" and one labeled "password". Below these fields are two buttons: "login" and "Registrasi". The window has a standard title bar with a close button (X) and a maximize button (square icon).

**Gambar 4.34 Antarmuka Halaman Login**

**b. Halaman Registrasi**

Halaman registrasi merupakan halaman bagi pengguna yang ingin mendaftar menjadi Administrator. Gambar 4.35 merupakan gambar antarmuka halaman registrasi.

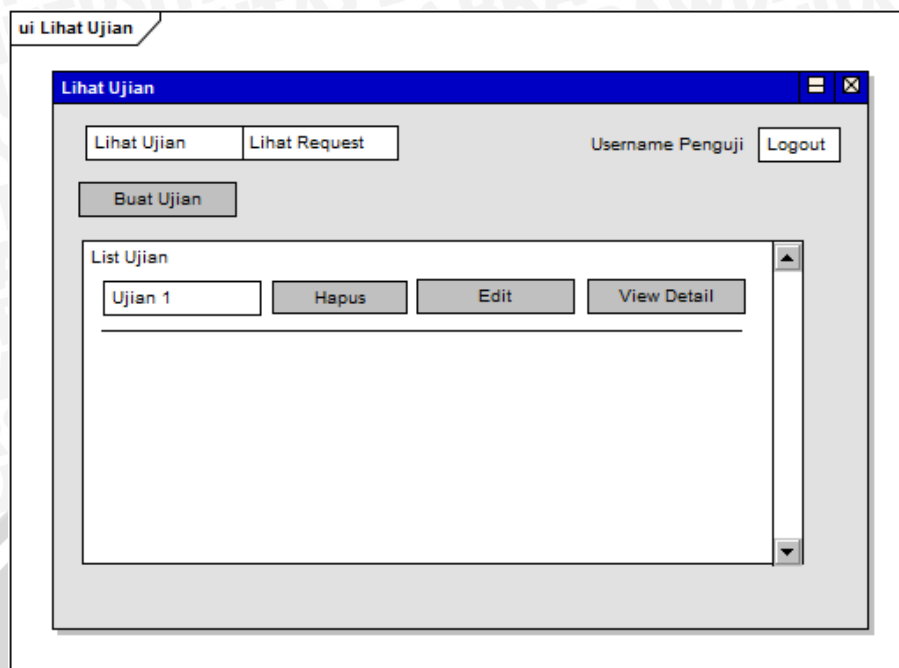


The screenshot shows a web application window titled "Registrasi". Inside the window, there is a section titled "Form Registrasi". This section contains four text input fields, labeled "Username", "Password", "Name", and "Email". Below these fields is a single button labeled "Register". The window has a standard title bar with a close button (X) and a maximize button (square icon).

**Gambar 4.35 Antarmuka Halaman Registrasi**

**c. Halaman Lihat Ujian**

Halaman lihat ujian adalah halaman yang berfungsi untuk menampilkan daftar ujian yang dikelola oleh Administrator. Gambar 4.36 merupakan gambar rancangan antarmuka halaman lihat ujian.



**Gambar 4.36 Antarmuka Halaman Lihat Ujian**

d. Halaman Buat Ujian

Halaman buat ujian adalah halaman untuk membuat ujian baru. Terdapat form di dalam halaman ini yang mana harus diisi oleh Administrator untuk pembuatan ujian baru. Gambar 4.37 merupakan gambar antarmuka halaman buat ujian.

**Gambar 4.37 Antarmuka Halaman Buat Ujian**

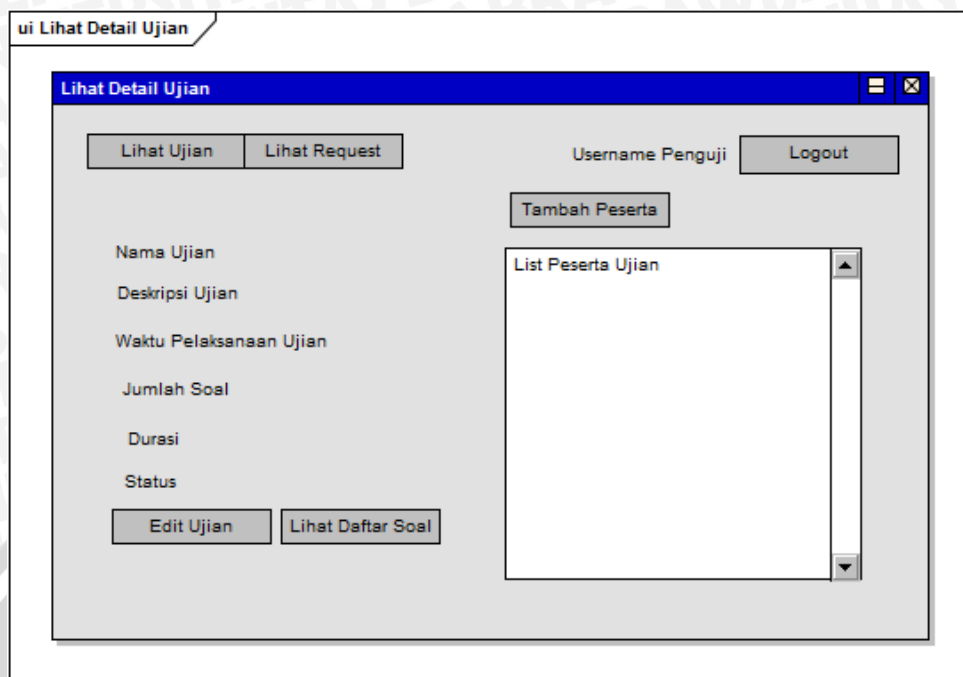
e. Halaman Edit Ujian

Halaman edit ujian adalah halaman untuk melakukan proses edit pada ujian yang telah dibuat. Gambar 4.38 merupakan antarmuka halaman edit ujian.

**Gambar 4.38 Antarmuka Halaman Edit Ujian**

f. Halaman Lihat Detail Ujian

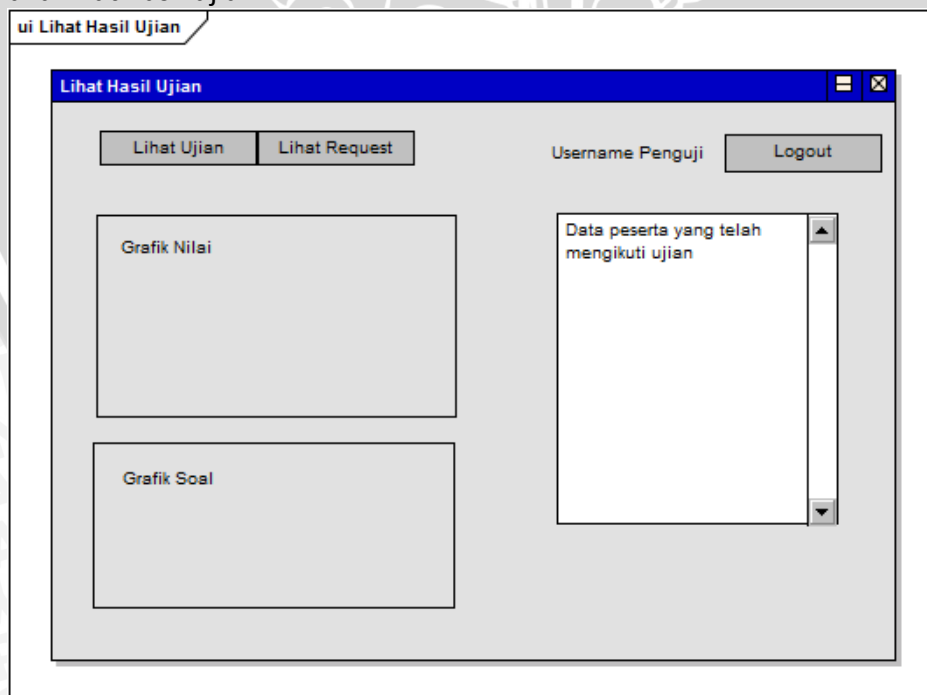
Halaman lihat detail ujian adalah halaman yang menampilkan detail dari suatu ujian. Halaman ini menampilkan data attribut soal dan daftar peserta yang mengikuti ujian. Gambar 4.39 adalah gambar antarmuka halaman lihat detail ujian.



**Gambar 4.39 Antarmuka Halaman Lihat Detail Ujian**

g. Halaman Lihat Hasil Ujian

Halaman lihat hasil ujian adalah halaman yang menampilkan hasil ujian. Data-data yang ditampilkan adalah data peserta yang telah mengikuti ujian beserta nilainya, grafik nilai, serta grafik soal. Gambar 4.41 merupakan gambar antarmuka lihat hasil ujian.



**Gambar 4.40 Antarmuka Halaman Lihat Hasil Ujian**

#### h. Halaman Tambah Peserta Ujian

Halaman tambah peserta ujian adalah halaman bagi Administrator untuk menambahkan peserta ke dalam suatu ujian. Gambar 4.41 merupakan gambar antarmuka halaman tambah peserta ujian.

**Gambar 4.41 Antarmuka Halaman Tambah Peserta Ujian**

#### i. Halaman Lihat Daftar Soal

Halaman lihat daftar soal adalah halaman yang menampilkan daftar soal yang ada dalam suatu ujian. Gambar 4.42 merupakan gambar antarmuka halaman lihat daftar soal.

**Gambar 4.42 Antarmuka Halaman Lihat Daftar Soal**

j. Halaman Buat Soal

Halaman buat soal adalah halaman yang berfungsi untuk pembuatan soal baru. Administrator harus mengisi form dengan lengkap untuk membuat soal yang baru. Gambar 4.43 merupakan gambar antarmuka halaman buat soal.

**Gambar 4.43 Antarmuka Halaman Buat Soal**

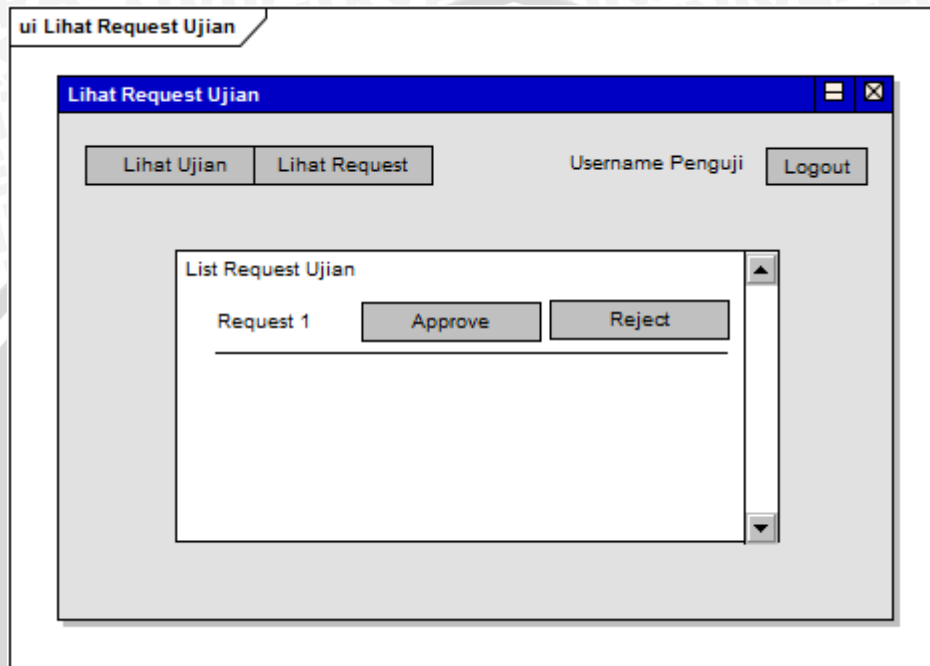
k. Halaman Edit Soal

Halaman edit soal adalah halaman yang berfungsi untuk melakukan edit soal. Soal yang telah dibuat dapat dirubah pada halaman ini. Gambar 4.44 adalah gambar antarmuka halaman edit soal.

**Gambar 4.44 Antarmuka Halaman Edit Soal**

#### I. Halaman Lihat *Request* Ujian

Halaman lihat *request* ujian adalah halaman yang menampilkan daftar *request* ujian yang dibuat oleh peserta. Administrator kemudian dapat melihat *request-request* yang ada dan dapat memutuskan untuk menyetujui permohonan peserta atau menolaknya melalui halaman ini. Gambar 4.47 merupakan gambar antarmuka halaman lihat *request* ujian.



Gambar 4.45 Antarmuka Halaman Lihat *Request* Ujian

## BAB 5 IMPLEMENTASI

Bab ini akan membahas mengenai tahapan implementasi perangkat lunak Aplikasi Backend Sistem Ujian Online Terpadu. Pembahasan terdiri dari spesifikasi sistem, batasan implementasi, implementasi *database*, implementasi kode program dan implementasi antarmuka.

### 5.1 Spesifikasi Sistem

Proses implementasi Aplikasi Backend Sistem Ujian Online Terpadu berasal dari analisis kebutuhan dan perancangan. Proses implementasi diterapkan pada lingkungan aplikasi yang terdiri dari perangkat keras dan perangkat lunak.

#### 5.1.1 Spesifikasi Perangkat Keras

Proses pengembangan aplikasi ini menggunakan perangkat komputer dengan spesifikasi *processor*, *memory*, dan *display* yang akan dijelaskan pada tabel 5.1.

**Tabel 5.1 Spesifikasi Perangkat Lunak**

| Nama Komponen       | Spesifikasi                                                |
|---------------------|------------------------------------------------------------|
| <i>System Model</i> | Asus K43SV                                                 |
| <i>Processor</i>    | Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz (4 CPUs), ~2.1GHz |
| <i>Memory</i>       | 4GB RAM                                                    |
| <i>Display</i>      | NVIDIA GeForce GT 540M                                     |

#### 5.1.2 Spesifikasi Perangkat Lunak

Proses pengembangan aplikasi ini menggunakan perangkat lunak dengan spesifikasi yang akan dijelaskan pada tabel 5.2

**Tabel 5.2 Spesifikasi Perangkat Lunak**

|                                |                              |
|--------------------------------|------------------------------|
| <i>Operating System</i>        | Windows 8.1 Pro 64-bit       |
| <i>Programming Language</i>    | PHP, Javascript, HTML        |
| <i>Editor Tools</i>            | Sublime Text 1.4             |
| <i>Framework</i>               | CodeIgniter 3.0.1, Bootstrap |
| <i>Programming Environment</i> | XAMPP                        |

## 5.2 Batasan Implementasi

Pada implementasi Aplikasi Backend Sistem Ujian Online Terpadu terdapat batasan-batasan implementasi yaitu sebagai berikut :

1. Proses pengacakan soal menggunakan algoritma Fisher-Yates.
2. Proses pertukaran data dari aplikasi ke *client* menggunakan format JSON.

## 5.3 Implementasi Database

Implementasi basis data dilakukan menggunakan Database Management System MySQL. Terdapat beberapa tabel yang diperlukan untuk menyimpan data pada aplikasi. Berikut merupakan gambar dan spesifikasi tabel-tabel pada database yang diimplementasikan.

| #                           | Name                    | Type         | Collation         | Attributes | Null | Default | Extra          |
|-----------------------------|-------------------------|--------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1  | <u>id</u>               | int(11)      |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2  | username_creator        | varchar(20)  | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3  | grade_max               | int(11)      |                   |            | No   | 100     |                |
| <input type="checkbox"/> 4  | enrollment_key          | varchar(15)  | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 5  | deskripsi               | varchar(100) | latin1_swedish_ci |            | Yes  | NULL    |                |
| <input type="checkbox"/> 6  | passing_grade           | float        |                   |            | No   | None    |                |
| <input type="checkbox"/> 7  | nama_ujian              | varchar(50)  | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 8  | durasi                  | int(11)      |                   |            | No   | None    |                |
| <input type="checkbox"/> 9  | waktu_pembuatan         | datetime     |                   |            | No   | None    |                |
| <input type="checkbox"/> 10 | waktu_mulai_pelaksanaan | datetime     |                   |            | No   | None    |                |
| <input type="checkbox"/> 11 | waktu_akhir_pelaksanaan | datetime     |                   |            | No   | None    |                |
| <input type="checkbox"/> 12 | jumlah_soal_minimal     | int(11)      |                   |            | No   | None    |                |
| <input type="checkbox"/> 13 | jumlah_group_soal       | int(11)      |                   |            | No   | 0       |                |
| <input type="checkbox"/> 14 | status                  | int(11)      |                   |            | No   | 0       |                |
| <input type="checkbox"/> 15 | is_shuffle              | int(11)      |                   |            | No   | None    |                |

Gambar 5.1 Tabel Ujian

Gambar 5.1 menjelaskan kolom-kolom yang ada pada tabel ujian. Tabel ujian digunakan untuk menyimpan data ujian pada aplikasi.

| #                          | Name             | Type         | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|------------------|--------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>        | int(11)      |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | token            | varchar(128) | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3 | username_peserta | varchar(20)  | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 4 | time_created     | datetime     |                   |            | No   | None    |                |
| <input type="checkbox"/> 5 | last_access      | datetime     |                   |            | No   | None    |                |

Gambar 5.2 Tabel Token

Gambar 5.2 menjelaskan kolom-kolom yang ada pada tabel token. Tabel token digunakan untuk menyimpan data token yang diperlukan oleh aplikasi *client* untuk melakukan *request* ke *web service*.

| #                          | Name            | Type         | Collation         | Attributes | Null | Default | Extra |
|----------------------------|-----------------|--------------|-------------------|------------|------|---------|-------|
| <input type="checkbox"/> 1 | <u>username</u> | varchar(20)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 2 | nama            | varchar(50)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 3 | email           | varchar(50)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 4 | password        | varchar(50)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 5 | gambar          | varchar(100) | latin1_swedish_ci |            | Yes  | NULL    |       |
| <input type="checkbox"/> 6 | tanggal_lahir   | date         |                   |            | No   | None    |       |
| <input type="checkbox"/> 7 | gender          | varchar(1)   | latin1_swedish_ci |            | No   | None    |       |

**Gambar 5.3 Tabel Administrator**

Gambar 5.3 menjelaskan kolom-kolom yang ada pada tabel Administrator. Tabel Administrator digunakan untuk menyimpan data Administrator.

| #                          | Name            | Type         | Collation         | Attributes | Null | Default | Extra |
|----------------------------|-----------------|--------------|-------------------|------------|------|---------|-------|
| <input type="checkbox"/> 1 | <u>username</u> | varchar(20)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 2 | nama            | varchar(50)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 3 | email           | varchar(50)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 4 | password        | varchar(50)  | latin1_swedish_ci |            | No   | None    |       |
| <input type="checkbox"/> 5 | gambar          | varchar(100) | latin1_swedish_ci |            | Yes  | NULL    |       |
| <input type="checkbox"/> 6 | tanggal_lahir   | date         |                   |            | No   | None    |       |
| <input type="checkbox"/> 7 | gender          | varchar(1)   | latin1_swedish_ci |            | No   | None    |       |

**Gambar 5.4 Tabel Peserta\_ujian**

Gambar 5.4 menjelaskan kolom-kolom yang ada pada tabel peserta ujian. Tabel peserta ujian digunakan untuk menyimpan data peserta ujian.

| #                          | Name            | Type         | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|-----------------|--------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>       | int(11)      |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | nama_soal       | varchar(100) | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3 | id_ujian        | int(11)      |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | tipe_soal       | int(11)      |                   |            | No   | 0       |                |
| <input type="checkbox"/> 5 | pertanyaan      | text         | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 6 | level_kesulitan | varchar(10)  | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 7 | gambar          | varchar(150) | latin1_swedish_ci |            | Yes  | NULL    |                |

**Gambar 5.5 Tabel Soal**

Gambar 5.5 menjelaskan kolom-kolom yang ada pada tabel soal. Tabel soal digunakan untuk menyimpan data soal-soal pada aplikasi.

| #                          | Name                  | Type         | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|-----------------------|--------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>             | int(11)      |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | <u>id_soal</u>        | int(11)      |                   |            | No   | None    |                |
| <input type="checkbox"/> 3 | <u>teks</u>           | text         | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 4 | <u>gambar</u>         | varchar(150) | latin1_swedish_ci |            | Yes  | NULL    |                |
| <input type="checkbox"/> 5 | <u>is_true_answer</u> | tinyint(1)   |                   |            | No   | 0       |                |

**Gambar 5.6 Tabel Pilihan\_jawaban**

Gambar 5.6 menjelaskan kolom-kolom yang ada pada tabel pilihan jawaban. Tabel ini digunakan untuk menyimpan pilihan jawaban untuk soal-soal berjenis pilihan ganda.

| #                          | Name           | Type       | Collation | Attributes | Null | Default | Extra          |
|----------------------------|----------------|------------|-----------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>      | int(11)    |           |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | <u>id_soal</u> | int(11)    |           |            | No   | None    |                |
| <input type="checkbox"/> 3 | <u>jawaban</u> | tinyint(1) |           |            | No   | None    |                |

**Gambar 5.7 Tabel Jawaban\_true\_or\_false**

Gambar 5.7 menjelaskan tentang kolom-kolom pada tabel jawaban\_true\_or\_false. Tabel ini digunakan untuk menyimpan data jawaban soal yang berjenis true or false.

| #                          | Name           | Type    | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|----------------|---------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>      | int(11) |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | <u>id_soal</u> | int(11) |                   |            | No   | None    |                |
| <input type="checkbox"/> 3 | <u>teks</u>    | text    | latin1_swedish_ci |            | No   | None    |                |

**Gambar 5.8 Tabel Jawaban\_short\_answer**

Gambar 5.8 menjelaskan tentang kolom-kolom pada tabel jawaban\_short\_answer. Tabel ini digunakan untuk menyimpan data jawaban soal yang berjenis *short answer* atau jawaban singkat.

| #                          | Name                     | Type        | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|--------------------------|-------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>                | int(11)     |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | <u>username_peserta</u>  | varchar(50) | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3 | <u>tanggal_disetujui</u> | date        |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | <u>id_ujian</u>          | int(11)     |                   |            | No   | None    |                |

**Gambar 5.9 Tabel Record\_peserta\_terdaftar**

Gambar 5.9 menjelaskan kolom-kolom pada tabel record\_peserta\_terdaftar. Tabel ini digunakan untuk menyimpan data peserta yang terdaftar dalam ujian.

| #                          | Name             | Type        | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|------------------|-------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>        | int(11)     |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | username_peserta | varchar(20) | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3 | id_ujian         | int(11)     |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | is_decided       | tinyint(1)  |                   |            | No   | 0       |                |

**Gambar 5.10 Tabel Record\_pendaftaran\_ujian**

Gambar 5.10 menjelaskan kolom-kolom yang ada pada tabel record\_pendaftaran\_ujian. Tabel ini digunakan untuk menyimpan data permohonan peserta untuk menjadi peserta suatu ujian.

| #                          | Name              | Type        | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|-------------------|-------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>         | int(11)     |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | username_peserta  | varchar(20) | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3 | id_ujian          | int(11)     |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | waktu_mulai_ujian | datetime    |                   |            | No   | None    |                |
| <input type="checkbox"/> 5 | waktu_berakhir    | datetime    |                   |            | No   | None    |                |

**Gambar 5.11 Tabel Record\_pelaksanaan\_ujian**

Gambar 5.11 menjelaskan kolom-kolom yang ada pada tabel record\_pelaksanaan\_ujian. Tabel ini menyimpan data pelaksanaan ujian oleh peserta.

| #                          | Name                        | Type    | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|-----------------------------|---------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>                   | int(11) |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | id_record_pelaksanaan_ujian | int(11) |                   |            | No   | None    |                |
| <input type="checkbox"/> 3 | id_soal                     | int(11) |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | no_soal                     | int(11) |                   |            | No   | None    |                |
| <input type="checkbox"/> 5 | pilihan_ids                 | text    | latin1_swedish_ci |            | Yes  | NULL    |                |

**Gambar 5.12 Tabel Record\_pemetaan\_soal**

Gambar 5.12 menjelaskan kolom-kolom yang ada pada tabel record\_pemetaan\_soal. Tabel ini digunakan untuk menyimpan skema soal yang telah diacak dan dikirimkan ke peserta ujian.

| #                          | Name                        | Type       | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|-----------------------------|------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | <u>id</u>                   | int(11)    |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | id_record_pelaksanaan_ujian | int(11)    |                   |            | No   | None    |                |
| <input type="checkbox"/> 3 | id_soal                     | int(11)    |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | no_soal                     | int(11)    |                   |            | No   | None    |                |
| <input type="checkbox"/> 5 | jawaban                     | text       | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 6 | is_true                     | tinyint(1) |                   |            | No   | None    |                |

**Gambar 5.13 Tabel Record\_pengerjaan\_ujian**

Gambar 5.13 menjelaskan kolom-kolom yang ada pada tabel record\_pengerjaan\_ujian. Tabel ini digunakan untuk menyimpan data

pengerjaan soal oleh peserta seperti jawaban yang dipilih atau dijawab oleh peserta.

| #                          | Name             | Type        | Collation         | Attributes | Null | Default | Extra          |
|----------------------------|------------------|-------------|-------------------|------------|------|---------|----------------|
| <input type="checkbox"/> 1 | id               | int(11)     |                   |            | No   | None    | AUTO_INCREMENT |
| <input type="checkbox"/> 2 | username_peserta | varchar(20) | latin1_swedish_ci |            | No   | None    |                |
| <input type="checkbox"/> 3 | id_ujian         | int(11)     |                   |            | No   | None    |                |
| <input type="checkbox"/> 4 | waktu_ujian      | datetime    |                   |            | No   | None    |                |
| <input type="checkbox"/> 5 | waktu_submit     | datetime    |                   |            | Yes  | NULL    |                |
| <input type="checkbox"/> 6 | skor             | float       |                   |            | Yes  | 0       |                |

**Gambar 5.14 Tabel Record\_hasil\_ujian**

Gambar 5.14 menjelaskan kolom-kolom yang ada pada tabel record\_hasil\_ujian. Tabel ini digunakan untuk menyimpan hasil ujian yang telah dikoreksi oleh sistem.

## 5.4 Implementasi Kode Program

Pada Aplikasi *Backend* Sistem Ujian Online Terpadu ini terdapat banyak *method* ataupun fungsi yang diterapkan. Pada laporan skripsi ini hanya dibahas algoritma dari beberapa fungsi inti saja.

### 5.4.1 Algoritma Pengiriman soal kepada *client*

```

1 public function mulai_ujian_get(){
2     $token=$this->get('token');
3     $id_ujian=$this->get('id_ujian');
4     //cek parameter lengkap atau tidak
5     if(!empty($token) && !empty($id_ujian)){
6         //cek token valid atau tidak
7         $data_token=$this->token->get_username($token);
8         if(!empty($data_token->username_peserta)){
9             //cek id ujian valid atau tidak
10            $this->load->model('ujian','',TRUE);
11            $data_ujian=$this->ujian->get_ujian_by_id($id_ujian);
12            if(!empty($data_ujian->id)){
13                //cek apakah ujian bisa dilaksanakan
14                date_default_timezone_set("UTC");
15                $cur_time=date('Y-m-d H:i:s', time());
16                $cur_time=strtotime($cur_time);
17                $durasi=$data_ujian->durasi;
18                $mulai_pelaksanaan=strtotime($data_ujian->
19                >waktu_mulai_pelaksanaan);
20                $akhir_pelaksanaan=strtotime("-$durasi
21                minutes",strtotime($data_ujian->waktu_akhir_pelaksanaan));
22
23                if($cur_time>=$mulai_pelaksanaan &&
24                $cur_time<=$akhir_pelaksanaan){
25                    //cek apakah peserta terdaftar sebagai peserta ujian
26                    $this->load-
27                >model('record_peserta_terdaftar','',TRUE);
28                    $peserta_valid=$this->record_peserta_terdaftar-
29                >is_peserta_terdaftar($id_ujian,$data_token->username_peserta);
30                    if($peserta_valid==TRUE){
31
32                        //jika peserta sedang mengerjakan ujian namun
33                merequest soal kembali.

```

```

34         $this->load-
35 >model('record_pelaksanaan_ujian','',TRUE);
36         $data_pelaksanaan=$this-
37 >record_pelaksanaan_ujian->get_data($data_token->username_peserta,$id_ujian);
38         if(empty($data_pelaksanaan->id)){
39             //-----SUKSES-----
40
41             //ambil data soal
42             $this->load->model('soal','',TRUE);
43             $this->load-
44 >model('record_pemetaan_soal','',TRUE);
45             $this->load-
46 >model('pilihan_jawaban','',TRUE);
47             $this->load-
48 >model('jawaban_true_or_false','',TRUE);
49             $this->load-
50 >model('jawaban_short_answer','',TRUE);
51             $raw=$this->soal-
52 >get_soal_by_id_ujian($id_ujian);
53
54             //persiapan data
55             $count=0;
56             foreach($raw->result_array() as $data_soal){
57
58                 if($data_soal['tipe_soal']==0){
59                     $soal[$count]=$data_soal;
60                     $data_jawaban=$this->pilihan_jawaban-
61 >get_by_id_soal($data_soal['id']);
62                     $i=0;
63                     foreach($data_jawaban->result() as
64 $pilihan_jawaban){
65
66                     $soal[$count]['pilihan'][$i]['id']=$pilihan_jawaban->id;
67                     $soal[$count]['pilihan'][$i]['teks']=$pilihan_jawaban->teks;
68                     $soal[$count]['pilihan'][$i]['gambar']=$pilihan_jawaban->gambar;
69                     $i++;
70                     }
71                     $soal[$count]['pilihan']=$this-
72 >pengacakan_soal($soal[$count]['pilihan'],3);
73                 }
74                 else{
75                     $soal[$count]=$data_soal;
76                 }
77
78                 $count++;
79             }
80
81             //mengacak soal
82             if($data_ujian->is_shuffle==1){
83                 $soal=$this->pengacakan_soal($soal,$count-1);
84             }
85
86             //simpan record waktu mulai ujian ke database
87             date_default_timezone_set("UTC");
88             $cur_time=date('Y-m-d H:i:s', time());
89             $waktu_berakhir=strtotime("+$durasi
90 minutes",strtotime($cur_time));
91             $waktu_berakhir=date('Y-m-d H:i:s',

```

```

99     $waktu_berakhir);
100
101         $this->record_pelaksanaan_ujian-
102 >insert($data_token->username_peserta,$id_ujian,$cur_time,$waktu_berakhir);
103         $id_pelaksanaan=$this-
104 >record_pelaksanaan_ujian->get_id($data_token->username_peserta,$id_ujian);
105
106         //simpan pemetaan soal ke database
107
108         $count=count($soal);
109         for($i=0;$i<$count;$i++){
110             if($soal[$i]['tipe_soal']==0){
111
112 $array=array($soal[$i]['pilihan'][0]['id'],$soal[$i]['pilihan'][1]['id'],$soa
113 l[$i]['pilihan'][2]['id'],$soal[$i]['pilihan'][3]['id']);
114                 $array=implode(":",$array);
115                 $this->record_pemetaan_soal-
116 >insert($id_pelaksanaan->id,$soal[$i]['id'],$i+1,$array);
117             }
118             else{
119                 $this->record_pemetaan_soal-
120 >insert($id_pelaksanaan->id,$soal[$i]['id'],$i+1,'');
121             }
122             //echo $id_pelaksanaan->id."<br/>";
123         }
124
125         //kirim data ujian dan soal
126
127         $this->response([
128             'status' => TRUE,
129             'message' => 'Pesanan sukses',
130             'data_ujian' => $data_ujian,
131             'data_soal' => $soal
132         ], REST_Controller::HTTP_OK); //
133
134         //-----END SUKSES-----
135
136     }
137     else{
138         $this->response([
139             'status' => FALSE,
140             'message' => 'peserta sedang atau
141 telah melaksanakan ujian dan tidak bisa melakukan request ulang'
142         ], REST_Controller::HTTP_FORBIDDEN); //
143     }
144
145     }
146     else{
147         $this->response([
148             'status' => FALSE,
149             'message' => 'peserta tidak terdaftar
150 sebagai peserta ujian'
151         ], REST_Controller::HTTP_FORBIDDEN); //
152     }
153
154     }
155     else{
156         $this->response([
157             'status' => FALSE,
158             'message' => 'ujian belum siap atau telah
159 dilaksanakan'
160         ], REST_Controller::HTTP_FORBIDDEN); //
161     }
162
163     }
164     else{
165         $this->response([

```

```

164         'status' => FALSE,
165         'message' => 'parameter id ujian tidak valid'
166     ], REST_Controller::HTTP_NOT_FOUND); //
167     }
168 }
169 else{
170     $this->response([
171         'status' => FALSE,
172         'message' => 'token tidak valid'
173     ], REST_Controller::HTTP_FORBIDDEN); //
174 }
175 }
176 else{
177     $this->response([
178         'status' => FALSE,
179         'message' => 'parameter tidak lengkap'
180     ], REST_Controller::HTTP_BAD_REQUEST); //
181 }
182 }
183 }
184 }

```

**Gambar 5.15 Algoritma Pengiriman Soal**

Pada gambar 5.15, ditampilkan implementasi kode program algoritma pengiriman soal. Algoritma ini dimulai ketika *client* melakukan *request* untuk memulai suatu ujian. Berikut beberapa penjelasan mengenai kode program pada gambar 5.15:

1. Baris 1 merupakan nama fungsi yang berada pada kelas api yang bertugas untuk menangani setiap *service*.
2. Baris 2-3 merupakan inisialisasi variabel token dan id\_ujian. Kedua variabel ini digunakan untuk menyimpan parameter yang dikirimkan oleh *client*.
3. Baris 4-5 merupakan pengecekan apakah parameter yang dikirimkan lengkap atau tidak.
4. Baris 6-8 merupakan pengecekan apakah token valid atau tidak
5. Baris 9-12 merupakan pengecekan apakah ujian valid atau tidak.
6. Baris 13-24 merupakan pengecekan apakah ujian bisa dilaksanakan atau tidak. Ujian bisa dilaksanakan apabila minimum soal untuk ujian telah terpenuhi dan berada dalam waktu pengerjaan ujian.
7. Baris 25-31 merupakan pengecekan apakah peserta merupakan peserta yang telah disetujui oleh Administrator.
8. Baris 32-38 merupakan pengecekan apakah peserta *request* data kembali. Pengecekan ini dilakukan agar peserta tidak dapat melakukan kecurangan dengan melakukan *request* soal berulang kali.
9. Baris 39-53 merupakan pengambilan data soal dan data lainnya yang diperlukan dari *database*.
10. Baris 54-84 merupakan penyiapan data, data yang telah diambil dari *database* disusun sedemikian rupa ke dalam suatu array yang nanti akan dikirimkan ke *client*.
11. Baris 85-90 merupakan pengacakan soal. Data yang telah disusun tadi dikirimkan ke dalam fungsi *pengacakan\_soal* dan kemudian dikembalikan dalam keadaan telah teracak. Pengacakan hanya berjalan jika attribute *is\_shuffle* bernilai 1.
12. Baris 93-104 merupakan proses penyimpanan waktu pelaksanaan ujian ke tabel *record\_pelaksanaan\_ujian* di *database*.

13. Baris 105-123 merupakan proses penyimpanan pemetaan soal atau soal yang telah diacak ke dalam tabel `record_pemetaan_soal` di *database*. Data ini perlu diolah kembali agar sesuai dengan attribute pada tabel `record_pemetaan_soal`.
14. Baris 124-133 merupakan pengiriman data ujian dan data soal yang telah diolah kepada *client*.
15. Baris 136-184 merupakan berbagai pengecualian jika syarat-syarat untuk melakukan *request* tidak terpenuhi. Sebagai contoh, jika token tidak valid maka server akan merespon dengan memberikan pesan error : token tidak valid.

#### 5.4.2 Algoritma Koreksi Jawaban

```

1 private function koreksi_jawaban($id_soal,$jawaban){
2     $jawaban=ltrim($jawaban);
3     $jawaban=rtrim($jawaban);
4     $this->load->model('soal','',TRUE);
5     $tipe_soal=$this->soal->get_tipe_soal_by_id($id_soal);
6
7     if($tipe_soal==0){
8         $this->load->model('pilihan_jawaban','',TRUE);
9         $data_jawaban=$this->pilihan_jawaban-
10 >get_jawaban_by_id_soal($id_soal);
11         if(strcasecmp($data_jawaban->teks,$jawaban)==0)
12 return 1;
13         else return 0;
14     }
15     elseif($tipe_soal==1){
16         $this->load-
17 >model('jawaban_true_or_false','',TRUE);
18         $data_jawaban=$this->jawaban_true_or_false-
19 >get_jawaban_by_id_soal($id_soal);
20         if($data_jawaban==$jawaban) return 1;
21         else return 0;
22     }
23     else{
24         $this->load-
25 >model('jawaban_short_answer','',TRUE);
26         $data_jawaban=$this->jawaban_short_answer-
27 >get_jawaban_by_id_soal($id_soal);
28         if(strcasecmp($data_jawaban,$jawaban)==0) return
29 1;
30         else return 0;
31     }
32 }
33
34
35
36
37
38

```

**Gambar 5.16 Algoritma Koreksi Jawaban**

Pada gambar 5.16, ditampilkan implementasi kode program algoritma untuk koreksi jawaban Berikut beberapa penjelasan mengenai kode program pada gambar 5.16:

1. Baris 1 merupakan nama fungsi yang berada pada kelas api yang bertujuan untuk menangani setiap *service*.

2. Baris 2-3 merupakan filter agar data jawaban yang dikirimkan dari *client* terbebas dari spasi.
3. Baris 4-6 merupakan pengambilan tipe soal dari database
4. Baris 8 merupakan pengecekan apakah tipe soal yang dikoreksi merupakan tipe soal pilihan ganda. Jika iya, maka *statement* di dalamnya akan dijalankan.
5. Baris 9-11 merupakan pengambilan jawaban yang benar dari *database*
6. Baris 12-17 merupakan pengecekan jawaban peserta dengan jawaban yang benar di *database*. Jika benar, maka fungsi pengacakan soal akan mengembalikan data yang bernilai 1 dan jika salah bernilai 0.
7. Baris 18 merupakan pengecekan apakah tipe soal yang dikoreksi merupakan tipe soal true or false. Jika iya, maka *statement* di dalamnya akan dijalankan.
8. Baris 19-22 merupakan pengambilan jawaban yang benar dari *database*
9. Baris 23-25 merupakan pengecekan jawaban peserta dengan jawaban yang benar. Jika benar maka fungsi akan mengembalikan data yang bernilai 1, sebaliknya jika salah maka data kembalian akan bernilai 0.
10. Baris 26 merupakan pengecekan apakah tipe soal yang dikoreksi merupakan jenis soal selain pilihan ganda dan true or false, yaitu tipe soal short answer. Jika iya, maka *statement* di dalamnya akan dijalankan.
11. Baris 27-31 merupakan pengambilan jawaban yang benar dari database
12. Baris 32-38 merupakan koreksi jawaban peserta dengan jawaban yang benar. Jika benar maka fungsi akan mengembalikan data yang bernilai 1, sebaliknya jika salah maka data kembalian akan bernilai 0.

#### 5.4.3 Algoritma Pengacakan Soal dan Pilihan Jawaban

```

1  private function pengacakan_soal($array,$jumlah){
2
3
4      for($jumlah=$jumlah;$jumlah>=0;$jumlah--){
5          //pilih angka acak
6          $rand=rand(0,$jumlah);
7          //tukar array pada element[rand] dengan elemen
8          terakhir N-1 pada array
9          $temp=$array[$rand];
10         $array[$rand]=$array[$jumlah];
11         $array[$jumlah]=$temp;
12
13     }
14
15     return($array);
16
17 }
18
19

```

**Gambar 5.17 Algoritma Fisher-Yates untuk Pengacakan Soal**

Pada gambar 5.17, ditampilkan implementasi kode program algoritma untuk pengacakan soal. Algoritma ini merupakan penerapan dari algoritma Fisher-Yates versi modern. Berikut beberapa penjelasan mengenai kode program pada gambar 5.17:

1. Baris 1 merupakan nama fungsi yang berada di kelas api. Terdapat 2 parameter yang dibutuhkan, yaitu array dan jumlah. Array merupakan array yang akan diacak dan jumlah merupakan jumlah indeks array.

2. Baris 4 merupakan pengulangan untuk mengacak array. Selama jumlah masih lebih besar atau sama dengan 0, maka pengulangan akan tetap berjalan.
3. Baris 6-7 merupakan pemilihan angka acak dengan batas bawah 0 dan batas atas menggunakan nilai jumlah.
4. Baris 10-12 merupakan pertukaran nilai dari array. Nilai dari elemen array diacak dan ditukar menggunakan variable bantu \$temp.
5. Baris 17 merupakan pengembalian data array yang telah diacak.

#### 5.4.4 Algoritma Untuk Menerima Jawaban dari *Client*

```

1 public function kirim_jawaban_post(){
2     $token=$this->post('token');
3     $id_ujian=$this->post('id_ujian');
4     $jawaban=$this->post('jawaban');
5
6     //cek parameter lengkap atau tidak
7     if(!empty($token) && !empty($id_ujian) &&
8     !empty($jawaban)){
9         $jawaban=json_decode($jawaban,TRUE);
10        //cek token valid atau tidak
11        $data_token=$this->token->get_username($token);
12        if(!empty($data_token->username_peserta)){
13            //cek id ujian valid atau tidak
14            $this->load->model('ujian','',TRUE);
15            $data_ujian=$this->ujian-
16            >get_ujian_by_id($id_ujian);
17            if(!empty($data_ujian->id)){
18                //cek pelaksanaan ujian ada atau tidak
19                $this->load-
20                >model('record_pelaksanaan_ujian','',TRUE);
21                $data_pelaksanaan=$this-
22                >record_pelaksanaan_ujian->get_data($data_token-
23                >username_peserta,$id_ujian);
24                if(!empty($data_pelaksanaan->id)){
25                    //cek sudah mengumpulkan jawaban atau tidak
26                    $this->load-
27                    >model('record_pengerjaan_ujian','',TRUE);
28                    $valid=$this->record_pengerjaan_ujian-
29                    >is_exist($data_pelaksanaan->id);
30                    if($valid==TRUE){
31                        //cek waktu pengumpulan melewati batas waktu atau tidak
32                        date_default_timezone_set("UTC");
33                        $cur_time=time();
34
35                        $waktu_mulai=strtotime($data_pelaksanaan->waktu_mulai_ujian);
36
37                        $waktu_berakhir=strtotime($data_pelaksanaan->waktu_berakhir);
38                        if($cur_time>=$waktu_mulai && $cur_time<=$waktu_berakhir){
39
40                            //cek apakah skema benar dan jumlah soal yang dikirimkan sama
41                            $valid=true;
42                            $count=count($jawaban);
43                            //jumlah soal di database
44                            $this->load-
45                            >model('soal','',TRUE);
46                            $jumlah_soal=$this->soal-
47                            >get_jumlah_soal($id_ujian);
48
49                            for($i=0;$i<$count;$i++){
50
51                                if(array_key_exists('nomor', $jawaban[$i])==false ||
52                                array_key_exists('jawaban', $jawaban[$i])==false){
53                                    $valid=false;

```

```

54     }
55     }
56
57     if($valid==true AND
58     $count==$jumlah_soal){
59         //simpan jawaban peserta ke
60         database
61         $this->load-
62         >model('record_pemetaan_soal','',TRUE);
63         $this->load-
64         >model('record_pengerjaan_ujian','',TRUE);
65         $this->load-
66         >model('record_hasil_ujian','',TRUE);
67         for($i=0;$i<$count;$i++){
68             //ambil data dari pemetaan soal
69             $id_soal=$this-
70             >record_pemetaan_soal->get_id_soal($data_pelaksanaan-
71             >id,$jawaban[$i]['nomor']);
72             //koreksi jawaban peserta
73             $sis_true=$this-
74             >koreksi_jawaban($id_soal,$jawaban[$i]['jawaban']);
75             //simpan pengerjaan ujian peserta ke database
76             $this-
77             >record_pengerjaan_ujian->insert($data_pelaksanaan-
78             >id,$id_soal,$jawaban[$i]['nomor'],$jawaban[$i]['jawaban'],$sis_true
79             );
80         }
81         //perhitungan skor
82         $skor=$data_ujian-
83         >grade_max/$data_ujian->jumlah_soal_minimal;
84
85         $jumlah_jawaban_benar=$this->record_pengerjaan_ujian-
86         >get_jumlah_benar($data_pelaksanaan->id);
87
88         $hasil=$skor*$jumlah_jawaban_benar;
89         //simpan data ke tabel hasil ujian
90         $this-
91         >record_hasil_ujian->insert($data_token-
92         >username_peserta,$id_ujian,$data_pelaksanaan-
93         >waktu_mulai_ujian,date('Y-m-d H:i:s', $cur_time),$hasil);
94
95         $this->response([
96             'status' => TRUE,
97             'message' =>
98             'Jawaban berhasil dikirim'
99         ],
100         REST_Controller::HTTP_OK); //
101     }
102     else{
103         $this->response([
104             'status' => FALSE,
105             'message' => 'Skema jawaban
106             tidak valid'
107         ],
108         REST_Controller::HTTP_BAD_REQUEST); //
109     }
110 }
111 else{
112     $this->response([
113         'status' => FALSE,
114         'message' => 'Waktu
115         pengerjaan telah habis, tidak bisa mengumpulkan jawaban'
116     ],
117

```

```

119 REST_Controller::HTTP_FORBIDDEN); //
120     }
121     }
122     else{
123         $this->response([
124             'status' => FALSE,
125             'message' => 'Pegguna telah
126 mengirimkan jawaban, tidak dapat mengirim ulang'
127         ], REST_Controller::HTTP_FORBIDDEN); //
128     }
129     }
130     else{
131         $this->response([
132             'status' => FALSE,
133             'message' => 'Pegguna belum
134 melaksanakan ujian'
135         ], REST_Controller::HTTP_FORBIDDEN); //
136     }
137     }
138     else{
139         $this->response([
140             'status' => FALSE,
141             'message' => 'parameter id ujian
142 tidak valid'
143         ], REST_Controller::HTTP_NOT_FOUND); //
144     }
145     }
146     else{
147         $this->response([
148             'status' => FALSE,
149             'message' => 'token tidak valid'
150         ], REST_Controller::HTTP_FORBIDDEN); //
151     }
152     }
153     else{
154         $this->response([
155             'status' => FALSE,
156             'message' => 'parameter tidak lengkap'
157         ], REST_Controller::HTTP_BAD_REQUEST); //
158     }
159     }
160
161

```

**Gambar 5.18 Algoritma untuk menerima jawaban dari *client*.**

Pada gambar 5.18, ditampilkan implementasi kode program algoritma untuk menerima jawaban peserta. Berikut beberapa penjelasan mengenai kode program pada gambar 5.17:

1. Baris 1 merupakan nama fungsi yang berada pada class api.
2. Baris 2-4 merupakan inisialisasi variable yang dikirimkan oleh *client*.
3. Baris 6-8 merupakan pengecekan apakah parameter yang dikirimkan *client*.
4. Baris 9 merupakan decode jawaban dari client yang sebelumnya berbentuk JSON
5. Baris 10-12 merupakan pengecekan apakah token yang dikirimkan valid atau tidak.
6. Baris 13-17 merupakan pengecekan apakah id ujian yang dikirimkan valid atau tidak.
7. 18-24 merupakan pengecekan apakah sudah terdapat data pelaksanaan ujian atau tidak. Jika telah ada, maka klien tidak dapat melakukan ujian ulang.
8. Baris 25-30 merupakan pengecekan apakah klien sudah pernah mengumpulkan jawaban sebelumnya. Pengecekan ini bertujuan agar klien tidak dapat melakukan kecurangan dengan mengirimkan kembali jawabannya.

9. Baris 31-39 merupakan pengecekan apakah waktu pengumpulan telah melewati batas waktu pengerjaan yang seharusnya. Jika telah melewati, maka klien tidak dapat mengumpulkan jawabannya.
10. Baris 40-47 merupakan pengambilan data yang dibutuhkan dari *database* untuk pengecekan skema jawaban yang dikirimkan oleh peserta.
11. Baris 48-58 merupakan pengecekan apakah skema jawaban yang dikirimkan oleh peserta valid dan jumlah soal yang dikirimkan sama. Jika tidak, maka jawaban tidak diterima oleh aplikasi.
12. Baris 59- 71 merupakan pengambilan data pemetaan soal dari *database*.
13. Baris 72-74 merupakan pemanggilan fungsi untuk mengoreksi jawaban.
14. Baris 75-80 merupakan proses penyimpanan data pengerjaan ujian oleh peserta ke *database*.
15. Baris 81-88 merupakan perhitungan skor dari jawaban yang benar. Perhitungan skor dihitung dengan rumus :  $(\text{skor\_maksimum}/\text{jumlah\_soal\_minimum}) * \text{jumlah jawaban benar}$ .
16. Baris 89-94 merupakan penyimpanan hasil ujian ke dalam tabel *record\_hasil\_ujian* di *database*.
17. Baris 95-101 merupakan pengiriman respon dari aplikasi kembali ke *client*.
18. Baris 102-161 merupakan pengecualian jika *request* yang dilakukan oleh client tidak valid. Pemberitahuan kesalahan yang terjadi akan dikirimkan kembali ke *client*.

## 5.5 Implementasi Antarmuka

Bab ini menjelaskan tampilan antarmuka dari Aplikasi *Backend* Sistem Ujian Online Terpadu. Fungsi-fungsi yang memiliki antarmuka adalah fungsi yang memenuhi kebutuhan Administrator saja. Hal ini dikarenakan untuk peserta ujian, aplikasi *backend* hanya menyediakan *service* untuk pertukaran data. Berikut merupakan implementasi antarmuka dari aplikasi.

### 5.5.1 Halaman Login Administrator

Administrator harus melakukan login agar bisa masuk ke dalam aplikasi. Di dalam halaman login, terdapat username dan password yang diisi. Jika keduanya valid, maka Administrator dapat login. Gambar 5.19 merupakan gambar implementasi halaman login Administrator.

Gambar 5.19 Halaman Login Administrator

### 5.5.2 Halaman Registrasi Administrator

Halaman registrasi Administrator adalah halaman dimana Administrator yang belum memiliki akun mendaftar agar dapat login ke dalam aplikasi. Gambar 5.20 merupakan gambar implementasi halaman registrasi Administrator.

Gambar 5.19 Halaman Registrasi Administrator

### 5.5.3 Halaman Lihat Daftar Ujian

Halaman lihat daftar ujian menampilkan daftar ujian yang dikelola oleh Administrator yang bersangkutan. Terdapat tiga aksi yang dapat dilakukan pada halaman ini. Yaitu lihat detail, edit dan hapus. Untuk ujian yang sedang berlangsung atau telah dilaksanakan, maka fitur edit tidak dapat dilakukan. Sedangkan untuk fitur hapus hanya tidak berlaku untuk ujian yang sedang berlangsung. Gambar 5.20 merupakan gambar implementasi halaman lihat daftar ujian.

| SISTEM UJIAN ONLINE TERPADU                                                            |                          |                |               |                     |                     |                        |                                            |
|----------------------------------------------------------------------------------------|--------------------------|----------------|---------------|---------------------|---------------------|------------------------|--------------------------------------------|
| <div> <div> <div>Manajemen Ujian</div> <div>Lihat Permohonan Ujian</div> </div> </div> |                          |                |               |                     |                     |                        |                                            |
| > Lihat Ujian                                                                          |                          |                |               |                     |                     |                        |                                            |
| ID                                                                                     | Nama                     | Enrollment Key | Status        | Pelaksanaan         | Akhir Pelaksanaan   | View                   | Edit                                       |
| 41                                                                                     | Dream Theater Quiz       | dreams         | Lengkap       | 21-11-2015 08:39:00 | 23-11-2015 09:00:00 | <a href="#">Detail</a> | <a href="#">Hapus</a>                      |
| 43                                                                                     | Ujian Interview bank BRI | BRIbrawijaya   | Belum Lengkap | 09-12-2015 07:00:00 | 10-12-2015 12:00:00 | <a href="#">Detail</a> | <a href="#">Edit</a> <a href="#">Hapus</a> |
| 44                                                                                     | Ujian Psikotes Test1     | psikotest      | Lengkap       | 08-12-2015 20:37:00 | 09-12-2015 08:00:00 | <a href="#">Detail</a> | <a href="#">Hapus</a>                      |

Gambar 5.20 Halaman Lihat\_daftar\_ujian

#### 5.5.4 Halaman Buat Ujian

Halaman buat ujian berfungsi untuk membuat ujian baru. Terdapat form yang harus diisi oleh Administrator dengan benar. Gambar 5.21 merupakan gambar implementasi halaman buat ujian.

The screenshot displays the 'Buat Ujian' (Create Exam) interface. It includes a sidebar with a user profile and a main form area. The form contains the following fields:

- Nama ujian:** Text input field.
- Deskripsi ujian:** Text area for description.
- Skor Maksimal:** Number input field (set to 100).
- Skor Lolos Minimal:** Number input field.
- Enrollment key:** Text input field.
- Durasi ujian:** Text input field.
- Tanggal dan jam pelaksanaan ujian:** Date and time picker (set to 12/09/2015 07:00 AM).
- Tanggal dan jam ujian berakhir:** Date and time picker.

A calendar widget on the right side of the form shows the date December 10, 2015, is selected.

Gambar 5.21 Halaman Buat Ujian

#### 5.5.5 Halaman Edit Ujian

Halaman edit ujian merupakan halaman untuk mengubah attribute ujian yang telah dibuat sebelumnya. Gambar 5.22 merupakan gambar implementasi halaman edit ujian.

The screenshot displays the 'Edit Ujian' (Edit Exam) interface. It includes a sidebar with a user profile and a main form area. The form contains the following fields:

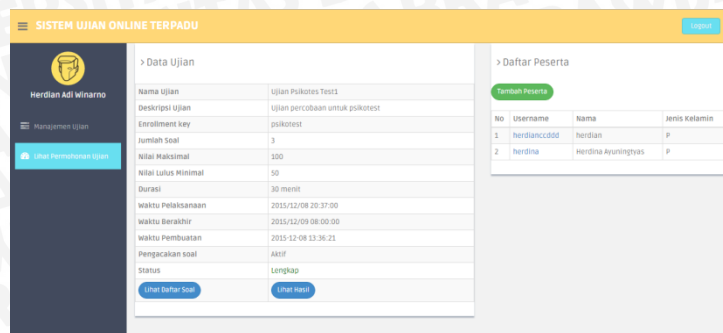
- Nama ujian:** Text input field (set to 'Ujian Interview Bank BRI').
- Deskripsi ujian:** Text area for description (set to 'Ujian untuk rekrutmen pegawai bank BRI').
- Skor Maksimal:** Number input field (set to 100).
- Skor Lolos Minimal:** Number input field (set to 70).
- Enrollment key:** Text input field (set to 'BRI04wajaya').
- Durasi Ujian:** Text input field (set to 120).
- Tanggal dan jam pelaksanaan ujian:** Date and time picker (set to 12/09/2015 07:00 AM).
- Tanggal dan jam ujian berakhir:** Date and time picker.

A calendar widget on the right side of the form shows the date December 10, 2015, is selected.

Gambar 5.22 Halaman Edit Ujian

#### 5.5.6 Halaman Lihat Detail Ujian

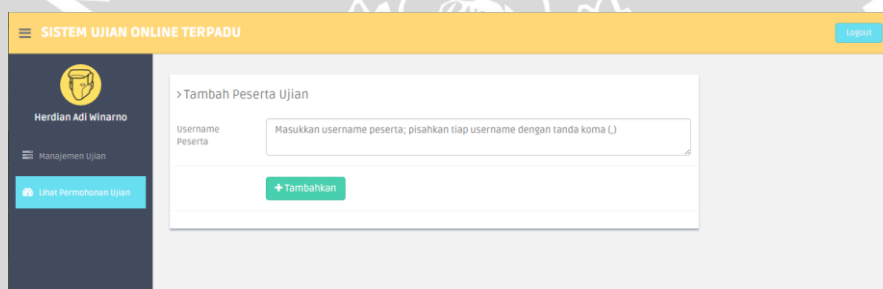
Halaman lihat detail ujian menampilkan rincian informasi dari suatu ujian. Terdapat empat tombol aksi di halaman ini. Yaitu lihat daftar soal, lihat hasil, edit dan tambah peserta. Tombol edit tidak akan ditampilkan jika ujian sedang berlangsung atau telah dilaksanakan. Gambar 5.23 merupakan gambar implementasi halaman lihat detail ujian.



Gambar 5.23 Halaman Lihat Detail Ujian

### 5.5.7 Halaman Tambah Peserta Ujian

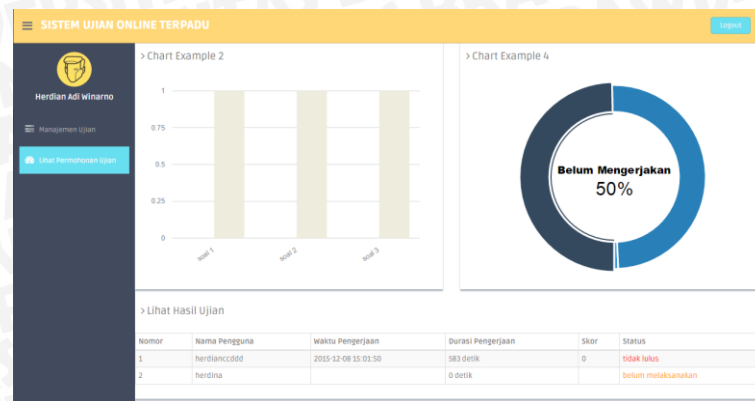
Halaman tambah peserta ujian merupakan halaman untuk memasukkan peserta ke dalam suatu ujian. Administrator dapat memasukkan banyak username peserta sekaligus dengan pemisah koma. Gambar 5.24 merupakan gambar implementasi halaman tambah peserta ujian.



Gambar 5.24 Halaman Tambah Peserta Ujian

### Halaman Tampilkan Hasil Ujian

Halaman tampilkan hasil ujian merupakan halaman yang menampilkan hasil dari suatu ujian. Terdapat 3 informasi yang ditampilkan yaitu persentase peserta yang lulus, tidak lulus dan belum mengerjakan, kemudian informasi benar dan salah dari tiap-tiap soal, dan informasi rinci mengenai hasil ujian dari masing-masing peserta. Gambar 5.25 merupakan gambar implementasi halaman tampilkan hasil ujian.



Gambar 5.25 Halaman Tampilkan Hasil Ujian

### Halaman Lihat Daftar Soal

Halaman lihat daftar soal merupakan halaman yang menampilkan daftar soal dalam suatu ujian. Terdapat 3 tombol aksi pada halaman ini, yaitu buat soal baru, edit soal dan hapus soal. Gambar 5.26 merupakan gambar implementasi halaman lihat daftar soal.

| ID | Nama Soal | Tipe Soal       | Pertanyaan                                                                                                                                                                                                                                                                                                   | Level Kesulitan | Edit | Hapus |
|----|-----------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|-------|
| 61 | soal 1    | Pilihan Ganda   | ABOLUSI                                                                                                                                                                                                                                                                                                      | sedang          | Edit | Hapus |
| 62 | soal 2    | Pilihan Ganda   | Di sini ada perempatan jalan yang saling tegak lurus. Salah satu jalan itu menuju ke kota Semarang. Jika saya menghadap utara, maka jalan yang ke kiri dan belakang saya adalah jalan yang menuju ke tempat lain. Jalan yang tepat di hadapan saya menuju ke Laut Jawa. Jalan manakah yang menuju Semarang ? | sedang          | Edit | Hapus |
| 63 | soal 3    | Jawaban Singkat | 1000 : 500 + 98 - 90 = ?                                                                                                                                                                                                                                                                                     | mudah           | Edit | Hapus |

Gambar 5.26 Halaman Lihat Daftar Soal

### Halaman Buat Soal

Halaman buat soal merupakan halaman untuk membuat soal baru. Terdapat form yang harus diisi oleh Administrator. Gambar 5.27 merupakan gambar implementasi halaman buat soal.

**Gambar 5.27 Halaman Buat Soal**

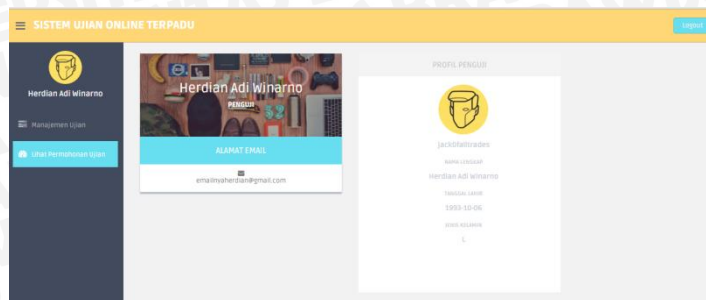
### Halaman Edit Soal

Halaman Edit Soal merupakan halaman untuk mengubah soal yang telah dibuat sebelumnya. Gambar 5.28 merupakan gambar implementasi halaman edit soal.

**Gambar 5.28 Halaman Edit Soal**

### Halaman Tampilkan Profil Pengguna

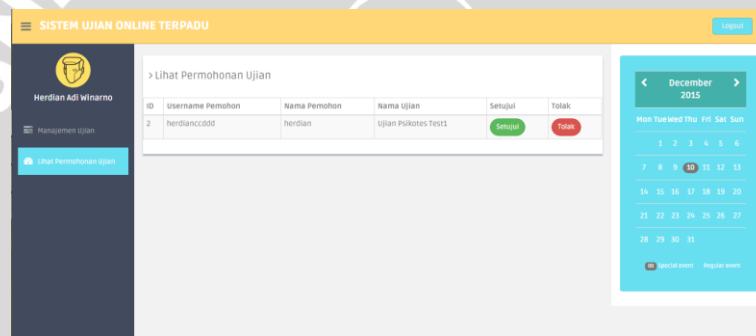
Halaman tampilkan profil pengguna merupakan halaman yang menampilkan informasi mengenai pengguna baik Administrator maupun peserta ujian. Gambar 5.29 merupakan implementasi halaman tampilkan profil pengguna.



Gambar 5.29 Halaman Tampilan Profil Pengguna

### Halaman Lihat *Request* Ujian

Halaman lihat *request* ujian merupakan halaman yang menampilkan permohonan untuk mengikuti ujian dari peserta. Terdapat 2 tombol aksi pada halaman ini, yaitu setuju dan tolak. Gambar 5.30 merupakan gambar implementasi halaman lihat *request* ujian.

Gambar 5.30 Halaman Lihat *Request* Ujian

## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap Aplikasi *Backend* Sistem Ujian Online Terpadu. Terdapat dua jenis pengujian pada bab ini, yaitu pengujian fungsional dan pengujian non-fungsional. Pengujian fungsional adalah pengujian validasi yang menggunakan teknik pengujian *black-box* (*black-box testing*). Sedangkan pengujian non-fungsional terdiri dari dua pengujian. Yaitu pengujian *security* dan pengujian *reliability*. Pengujian *security* menguji apakah aplikasi hanya bisa diakses oleh pihak yang memiliki hak saja. Pengujian *security* dilakukan dengan menguji fitur login dan menguji penggunaan token pada *web service*. Pengujian *reliability* menguji apakah ujian telah dapat mengirimkan jawaban secara sekaligus dan menerima jawaban sekaligus dari *client*. Pengujian *reliability* dilakukan dengan melihat respon *server* terhadap berbagai kemungkinan kesalahan dan gangguan saat pengiriman soal dan jawaban.

### 6.1 Pengujian Fungsional

Pengujian fungsional pada penelitian ini menggunakan pengujian validasi. Pengujian ini dilakukan untuk mengetahui kesesuaian antara daftar-daftar kebutuhan yang merupakan hasil dari analisis kebutuhan dengan kinerja sistem.

#### 6.1.1 Kasus Uji

Pada setiap kebutuhan sistem, dilakukan proses pengujian kasus yang akan ditunjukkan pada Tabel-tabel berikut.

**Tabel 6.1 Kasus Uji Login Administrator**

|                       |                                                                                                            |
|-----------------------|------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_01                                                                                                   |
| Nama Kasus Uji        | Login Administrator                                                                                        |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_01)                                                                             |
| Tujuan Pengujian      | Memastikan Administrator dapat masuk ke dalam sistem dengan <i>username</i> dan <i>password</i> yang valid |
| Data Masukan          | - <i>Username</i><br>- <i>Password</i>                                                                     |
| Prosedur Pengujian    | Mengisi data masukan di halaman login dan menekan tombol login                                             |
| Hasil yang Diharapkan | Administrator berhasil login dan dibawa ke halaman lihat daftar soal                                       |

**Tabel 6.2 Kasus Uji Registrasi Administrator**

|                  |                                                   |
|------------------|---------------------------------------------------|
| Nomor Kasus Uji  | SRS_1_02                                          |
| Nama Kasus Uji   | Registrasi Administrator                          |
| Objek Uji        | Kebutuhan Fungsional(SRS_1_02)                    |
| Tujuan Pengujian | Memastikan Administrator dapat terdaftar ke dalam |

|                       |                                                                                                                                                                     |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | sistem                                                                                                                                                              |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Username</li> <li>- Password</li> <li>- Nama</li> <li>- Email</li> <li>- Tanggal Lahir</li> <li>- Jenis Kelamin</li> </ul> |
| Prosedur Pengujian    | Mengisi data masukan di halaman registrasi dan menekan tombol daftar                                                                                                |
| Hasil yang Diharapkan | Administrator berhasil terdaftar ke dalam sistem.                                                                                                                   |

**Tabel 6.3 Kasus Uji Lihat Daftar Ujian**

|                       |                                                                                                            |
|-----------------------|------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_03                                                                                                   |
| Nama Kasus Uji        | Lihat Daftar Ujian                                                                                         |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_03)                                                                             |
| Tujuan Pengujian      | Memastikan data daftar ujian yang dikelola penguji dapat ditampilkan                                       |
| Data Masukan          | -                                                                                                          |
| Prosedur Pengujian    | Menekan menu lihat daftar ujian                                                                            |
| Hasil yang Diharapkan | Administrator dibawa ke halaman lihat daftar soal yang menampilkan data daftar soal yang dikelola penguji. |

**Tabel 6.4 Kasus Uji Buat Ujian**

|                       |                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_04                                                                                                                                                                                                                                                                                                                                                       |
| Nama Kasus Uji        | Buat Ujian                                                                                                                                                                                                                                                                                                                                                     |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_04)                                                                                                                                                                                                                                                                                                                                 |
| Tujuan Pengujian      | Memastikan Administrator dapat membuat ujian baru                                                                                                                                                                                                                                                                                                              |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Nama ujian</li> <li>- Deskripsi ujian</li> <li>- Skor maksimal</li> <li>- Skor lulus minimal</li> <li>- Enrollment key</li> <li>- Durasi ujian</li> <li>- Tanggal dan jam pelaksanaan ujian</li> <li>- Tanggal dan jam berakhirnya ujian</li> <li>- Jumlah soal minimal</li> <li>- Pilihan pengacakan soal</li> </ul> |
| Prosedur Pengujian    | Mengisi data masukan di halaman buat ujian dan menekan tombol submit                                                                                                                                                                                                                                                                                           |
| Hasil yang Diharapkan | Administrator berhasil membuat ujian dan data ujian disimpan di <i>database</i>                                                                                                                                                                                                                                                                                |

**Tabel 6.5 Kasus Uji Hapus Ujian**

|                       |                                                                    |
|-----------------------|--------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_05                                                           |
| Nama Kasus Uji        | Hapus Ujian                                                        |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_05)                                     |
| Tujuan Pengujian      | Memastikan Administrator dapat menghapus ujian                     |
| Data Masukan          | -                                                                  |
| Prosedur Pengujian    | Menekan tombol hapus pada ujian yang ingin dihapus.                |
| Hasil yang Diharapkan | Penghapusan berhasil dan data ujian dihapus dari <i>database</i> . |

**Tabel 6.6 Kasus Uji Edit Ujian**

|                       |                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_06                                                                                                                                                                                                                                                                                                                                                       |
| Nama Kasus Uji        | Edit Ujian                                                                                                                                                                                                                                                                                                                                                     |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_06)                                                                                                                                                                                                                                                                                                                                 |
| Tujuan Pengujian      | Memastikan Administrator dapat melakukan perubahan pada ujian yang telah ada                                                                                                                                                                                                                                                                                   |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Nama ujian</li> <li>- Deskripsi ujian</li> <li>- Skor maksimal</li> <li>- Skor lulus minimal</li> <li>- Enrollment key</li> <li>- Durasi ujian</li> <li>- Tanggal dan jam pelaksanaan ujian</li> <li>- Tanggal dan jam berakhirnya ujian</li> <li>- Jumlah soal minimal</li> <li>- Pilihan pengacakan soal</li> </ul> |
| Prosedur Pengujian    | Mengubah data pada halaman edit ujian dan menekan tombol submit.                                                                                                                                                                                                                                                                                               |
| Hasil yang Diharapkan | Data ujian yang lama berhasil diganti dengan data ujian yang baru.                                                                                                                                                                                                                                                                                             |

**Tabel 6.7 Kasus Uji Lihat Detail Ujian**

|                    |                                                                     |
|--------------------|---------------------------------------------------------------------|
| Nomor Kasus Uji    | SRS_1_07                                                            |
| Nama Kasus Uji     | Lihat Detail Ujian                                                  |
| Objek Uji          | Kebutuhan Fungsional(SRS_1_07)                                      |
| Tujuan Pengujian   | Memastikan sitem dapat menampilkan data ujian secara rinci.         |
| Data Masukan       | -                                                                   |
| Prosedur Pengujian | Menekan tombol detail pada ujian yang ingin ditampilkan rinciannya. |

|                       |                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------|
| Hasil yang Diharapkan | Sistem berhasil menampilkan data ujian secara rinci pada halaman lihat detail ujian. |
|-----------------------|--------------------------------------------------------------------------------------|

**Tabel 6.8 Kasus Uji Tambah Peserta Ujian**

|                       |                                                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_08                                                                                                  |
| Nama Kasus Uji        | Tambah Peserta Ujian                                                                                      |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_08)                                                                            |
| Tujuan Pengujian      | Memastikan Administrator dapat menambahkan peserta ujian menjadi peserta yang terdaftar dari suatu ujian. |
| Data Masukan          | - Username peserta ujian                                                                                  |
| Prosedur Pengujian    | Memasukkan username peserta dan menekan tombol tambah peserta                                             |
| Hasil yang Diharapkan | Username yang valid akan ditambahkan sebagai peserta terdaftar dari suatu ujian.                          |

**Tabel 6.9 Kasus Uji Tampilkan Hasil Ujian**

|                       |                                                                          |
|-----------------------|--------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_09                                                                 |
| Nama Kasus Uji        | Tampilkan Hasil Ujian                                                    |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_09)                                           |
| Tujuan Pengujian      | Memastikan Administrator dapat menampilkan hasil dari suatu ujian        |
| Data Masukan          | -                                                                        |
| Prosedur Pengujian    | Menekan tombol lihat hasil ujian pada ujian yang ingin dilihat hasilnya. |
| Hasil yang Diharapkan | Data hasil ujian berhasil ditampilkan.                                   |

**Tabel 6.10 Kasus Uji Lihat Daftar Soal**

|                       |                                                                                |
|-----------------------|--------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_10                                                                       |
| Nama Kasus Uji        | Lihat Daftar Soal                                                              |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_10)                                                 |
| Tujuan Pengujian      | Memastikan sistem dapat menampilkan daftar soal yang ada pada suatu ujian.     |
| Data Masukan          | -                                                                              |
| Prosedur Pengujian    | Menekan tombol lihat daftar soal pada ujian yang ingin dilihat daftar soalnya. |
| Hasil yang Diharapkan | Sistem berhasil menampilkan daftar soal dari suatu ujian.                      |

**Tabel 6.11 Kasus Uji Buat Soal**

|                       |                                                                                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_11                                                                                                                                                                                                                    |
| Nama Kasus Uji        | Buat Soal                                                                                                                                                                                                                   |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_11)                                                                                                                                                                                              |
| Tujuan Pengujian      | Memastikan Administrator dapat membuat soal baru                                                                                                                                                                            |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Nama soal</li> <li>- Tipe soal</li> <li>- Pertanyaan</li> <li>- Level Kesulitan</li> <li>- Gambar (opsional)</li> <li>- Pilihan Jawaban (opsional tergantung tipe soal)</li> </ul> |
| Prosedur Pengujian    | Mengisi data masukan pada halaman buat soal dan menekan tombol submit                                                                                                                                                       |
| Hasil yang Diharapkan | Penambahan soal berhasil dan data soal tersimpan dalam <i>database</i> .                                                                                                                                                    |

**Tabel 6.12 Kasus Uji Hapus Soal**

|                       |                                                                   |
|-----------------------|-------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_12                                                          |
| Nama Kasus Uji        | Hapus Soal                                                        |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_12)                                    |
| Tujuan Pengujian      | Memastikan Administrator dapat menghapus soal                     |
| Data Masukan          | -                                                                 |
| Prosedur Pengujian    | Menekan tombol hapus pada soal yang ingin dihapus.                |
| Hasil yang Diharapkan | Penghapusan berhasil dan data soal dihapus dari <i>database</i> . |

**Tabel 6.13 Kasus Uji Edit Soal**

|                       |                                                                                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_13                                                                                                                                                                                                                    |
| Nama Kasus Uji        | Edit Soal                                                                                                                                                                                                                   |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_13)                                                                                                                                                                                              |
| Tujuan Pengujian      | Memastikan Administrator dapat melakukan perubahan terhadap soal yang telah ada                                                                                                                                             |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Nama soal</li> <li>- Tipe soal</li> <li>- Pertanyaan</li> <li>- Level Kesulitan</li> <li>- Gambar (opsional)</li> <li>- Pilihan Jawaban (opsional tergantung tipe soal)</li> </ul> |
| Prosedur Pengujian    | Mengubah data pada halaman edit soal dan menekan tombol submit                                                                                                                                                              |
| Hasil yang Diharapkan | Data soal yang lama berhasil diganti dengan data yang baru pada <i>database</i> .                                                                                                                                           |

**Tabel 6.14 Kasus Uji Tampilkan Profil Pengguna**

|                       |                                                          |
|-----------------------|----------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_14                                                 |
| Nama Kasus Uji        | Tampilkan Profil Pengguna                                |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_14)                           |
| Tujuan Pengujian      | Memastikan sistem dapat menampilkan data profil pengguna |
| Data Masukan          | -                                                        |
| Prosedur Pengujian    | Menekan akun avatar atau link pada daftar peserta ujian  |
| Hasil yang Diharapkan | Sistem berhasil menampilkan profil pengguna.             |

**Tabel 6.15 Kasus Uji Lihat *Request* Ujian**

|                       |                                                                                             |
|-----------------------|---------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_15                                                                                    |
| Nama Kasus Uji        | Lihat Request Ujian                                                                         |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_15)                                                              |
| Tujuan Pengujian      | Memastikan sistem dapat menampilkan <i>request</i> ujian yang dikirimkan oleh peserta ujian |
| Data Masukan          | -                                                                                           |
| Prosedur Pengujian    | Menekan menu lihat request ujian                                                            |
| Hasil yang Diharapkan | Sistem berhasil menampilkan data <i>request</i> ujian yang dikirimkan oleh peserta ujian    |

**Tabel 6.16 Kasus Uji *Approve Request* Ujian**

|                       |                                                                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_1_16                                                                                                                             |
| Nama Kasus Uji        | Approve <i>Request</i> Ujian                                                                                                         |
| Objek Uji             | Kebutuhan Fungsional(SRS_1_16)                                                                                                       |
| Tujuan Pengujian      | Memastikan permohonan ujian yang dikirimkan peserta dapat disetujui dan peserta terdaftar sebagai peserta ujian yang telah terdaftar |
| Data Masukan          | -                                                                                                                                    |
| Prosedur Pengujian    | Menekan tombol setuju pada permohonan yang terpilih di halaman lihat <i>request</i> ujian                                            |
| Hasil yang Diharapkan | Peserta berhasil terdaftar sebagai peserta ujian yang sah.                                                                           |

**Tabel 6.17 Kasus Uji *Reject Request* Ujian**

|                 |                             |
|-----------------|-----------------------------|
| Nomor Kasus Uji | SRS_1_17                    |
| Nama Kasus Uji  | <i>Reject Request</i> Ujian |

|                       |                                                                                          |
|-----------------------|------------------------------------------------------------------------------------------|
| Objek Uji             | Kebutuhan Fungsional(SRS_1_17)                                                           |
| Tujuan Pengujian      | Memastikan permohonan ujian yang dikirimkan peserta dapat ditolak oleh Administrator     |
| Data Masukan          | -                                                                                        |
| Prosedur Pengujian    | Menekan tombol tolak pada permohonan yang terpilih di halaman lihat <i>request</i> ujian |
| Hasil yang Diharapkan | Permohonan ujian dari peserta berhasil dihapus                                           |

**Tabel 6.18 Kasus Uji Login Peserta**

|                       |                                                                                                                                                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_01                                                                                                                                                                                                  |
| Nama Kasus Uji        | Login Peserta                                                                                                                                                                                             |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_01)                                                                                                                                                                            |
| Tujuan Pengujian      | Memastikan Peserta Ujian dapat masuk ke dalam sistem                                                                                                                                                      |
| Data Masukan          | <ul style="list-style-type: none"> <li>- <i>Username</i></li> <li>- <i>Password</i></li> </ul>                                                                                                            |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/login">http://localhost/oes/api/login</a> menggunakan HTTP method POST |
| Hasil yang Diharapkan | Sistem mengirimkan token sebagai tanda bahwa peserta berhasil login.                                                                                                                                      |

**Tabel 6.19 Kasus Uji Registrasi Peserta**

|                       |                                                                                                                                                                                                                 |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_02                                                                                                                                                                                                        |
| Nama Kasus Uji        | Registrasi Peserta                                                                                                                                                                                              |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_02)                                                                                                                                                                                  |
| Tujuan Pengujian      | Memastikan pengguna yang belum terdaftar dapat terdaftar sebagai peserta ujian                                                                                                                                  |
| Data Masukan          | <ul style="list-style-type: none"> <li>- <i>Username</i></li> <li>- <i>Password</i></li> <li>- Nama</li> <li>- Email</li> <li>- Tanggal lahir</li> <li>- Jenis kelamin</li> </ul>                               |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/register">http://localhost/oes/api/register</a> menggunakan HTTP method POST |
| Hasil yang Diharapkan | Registrasi berhasil dan data peserta ujian tersimpan di <i>database</i> dan sistem memberikan konfirmasi bahwa proses registrasi berhasil.                                                                      |

**Tabel 6.20 Kasus Uji Daftar Ujian**

|                       |                                                                                                                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_03                                                                                                                                                                                                                |
| Nama Kasus Uji        | Daftar Ujian                                                                                                                                                                                                            |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_03)                                                                                                                                                                                          |
| Tujuan Pengujian      | Memastikan peserta ujian dapat mengajukan permohonan untuk mengikuti suatu ujian ke Administrator                                                                                                                       |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Token</li> <li>- Kode enroll</li> </ul>                                                                                                                                        |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/daftar_ujian">http://localhost/oes/api/daftar_ujian</a> menggunakan HTTP method POST |
| Hasil yang Diharapkan | Permohonan ujian berhasil disimpan ke database dan sistem memberikan konfirmasi bahwa permohonan telah berhasil dikirim                                                                                                 |

**Tabel 6.21 Kasus Uji Lihat Daftar Ujian Peserta**

|                       |                                                                                                                                                                                                                                    |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_04                                                                                                                                                                                                                           |
| Nama Kasus Uji        | Lihat Daftar Ujian Peserta                                                                                                                                                                                                         |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_04)                                                                                                                                                                                                     |
| Tujuan Pengujian      | Memastikan sistem dapat memberikan data ujian yang diikuti oleh peserta                                                                                                                                                            |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Token</li> </ul>                                                                                                                                                                          |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/lihat_daftar_ujian">http://localhost/oes/api/lihat_daftar_ujian</a> menggunakan HTTP method GET |
| Hasil yang Diharapkan | Sistem berhasil mengirimkan data ujian yang mana peserta telah terdaftar.                                                                                                                                                          |

**Tabel 6.22 Kasus Uji *Request* Soal**

|                    |                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------|
| Nomor Kasus Uji    | SRS_2_05                                                                                     |
| Nama Kasus Uji     | <i>Request</i> Soal                                                                          |
| Objek Uji          | Kebutuhan Fungsional(SRS_2_05)                                                               |
| Tujuan Pengujian   | Memastikan sistem berhasil mengirimkan soal dan peserta dapat memulai ujian                  |
| Data Masukan       | <ul style="list-style-type: none"> <li>- Token</li> <li>- Id_ujian</li> </ul>                |
| Prosedur Pengujian | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke |

|                       |                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------|
|                       | URL <a href="http://localhost/oes/api/mulai_ujian">http://localhost/oes/api/mulai_ujian</a> menggunakan HTTP method GET |
| Hasil yang Diharapkan | Sistem berhasil mengirimkan data ujian dan data soal dan memulai pelaksanaan ujian bagi peserta tersebut.               |

**Tabel 6.23 Kasus Uji Kirim Jawaban**

|                       |                                                                                                                                                                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_06                                                                                                                                                                                                                  |
| Nama Kasus Uji        | Kirim Jawaban                                                                                                                                                                                                             |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_06)                                                                                                                                                                                            |
| Tujuan Pengujian      | Memastikan peserta dapat mengirimkan jawaban ujiannya melalui <i>web service</i>                                                                                                                                          |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Token</li> <li>- Id_ujian</li> <li>- Jawaban</li> </ul>                                                                                                                          |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/kirim_jawaban">http://localhost/oes/api/kirim_jawaban</a> menggunakan HTTP method POST |
| Hasil yang Diharapkan | Jawaban berhasil disimpan di <i>database</i> dan dikelola. Sistem kemudian memberikan konfirmasi bahwa pengiriman jawaban berhasil.                                                                                       |

**Tabel 6.24 Kasus Uji Request Hasil Ujian**

|                       |                                                                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_07                                                                                                                                                                                                             |
| Nama Kasus Uji        | Request Hasil Ujian                                                                                                                                                                                                  |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_07)                                                                                                                                                                                       |
| Tujuan Pengujian      | Memastikan sistem dapat mengirimkan data hasil ujian yang diminta peserta                                                                                                                                            |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Token</li> <li>- Id_ujian</li> </ul>                                                                                                                                        |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/hasil_ujian">http://localhost/oes/api/hasil_ujian</a> menggunakan HTTP method GET |
| Hasil yang Diharapkan | Sistem berhasil mengirimkan data hasil ujian yang diminta oleh peserta                                                                                                                                               |

**Tabel 6.25 Kasus Uji Batal Ujian**

|                 |                                |
|-----------------|--------------------------------|
| Nomor Kasus Uji | SRS_2_08                       |
| Nama Kasus Uji  | Batal Ujian                    |
| Objek Uji       | Kebutuhan Fungsional(SRS_2_08) |

|                       |                                                                                                                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tujuan Pengujian      | Memastikan peserta ujian dapat melakukan pembatalan ujian                                                                                                                                                             |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Token</li> <li>- Id_ujian</li> </ul>                                                                                                                                         |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/batal_ujian">http://localhost/oes/api/batal_ujian</a> menggunakan HTTP method POST |
| Hasil yang Diharapkan | Pembatalan ujian berhasil dan peserta berhasil dihapus dari daftar peserta ujian yang bersangkutan.                                                                                                                   |

**Tabel 6.26 Kasus Uji *Request* Profil**

|                       |                                                                                                                                                                                                            |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nomor Kasus Uji       | SRS_2_09                                                                                                                                                                                                   |
| Nama Kasus Uji        | <i>Request</i> Profil                                                                                                                                                                                      |
| Objek Uji             | Kebutuhan Fungsional(SRS_2_09)                                                                                                                                                                             |
| Tujuan Pengujian      | Memastikan sistem dapat mengirimkan data peserta ujian                                                                                                                                                     |
| Data Masukan          | <ul style="list-style-type: none"> <li>- Token</li> </ul>                                                                                                                                                  |
| Prosedur Pengujian    | Melakukan <i>request</i> ke web service dengan mengirimkan data masukan sebagai parameter ke URL <a href="http://localhost/oes/api/profil">http://localhost/oes/api/profil</a> menggunakan HTTP method GET |
| Hasil yang Diharapkan | Sistem berhasil mengirimkan data profil peserta                                                                                                                                                            |

### 6.1.2 Hasil Pengujian Validasi

Hasil pengujian validasi menggunakan teknik pengujian *black-box* dijabarkan pada Tabel 6.26.

**Tabel 6.26 Hasil Pengujian Validasi**

| No | Nama Kasus Uji      | Skenario              | Hasil Yang Diharapkan                                                | Hasil Yang Diperoleh                                                 | Validitas |
|----|---------------------|-----------------------|----------------------------------------------------------------------|----------------------------------------------------------------------|-----------|
| 1  | Login Administrator | Skenario Utama        | Administrator berhasil login dan dibawa ke halaman lihat daftar soal | Administrator berhasil login dan dibawa ke halaman lihat daftar soal | Valid     |
|    |                     | Skenario Alternatif 1 | Aplikasi menampilkan pesan untuk mengisi kolom yang kosong           | Aplikasi menampilkan pesan untuk mengisi kolom yang kosong           | Valid     |

|   |                          |                       |                                                                                                            |                                                                                                            |       |
|---|--------------------------|-----------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|-------|
|   |                          | Skenario Alternatif 2 | Aplikasi menampilkan pesan <i>error</i> bahwa <i>username</i> dan <i>password</i> salah                    | Aplikasi menampilkan pesan <i>error</i> bahwa <i>username</i> dan <i>password</i> salah                    | Valid |
| 2 | Registrasi Administrator | Skenario Utama        | Administrator berhasil terdaftar ke dalam sistem.                                                          | Administrator berhasil terdaftar ke dalam sistem.                                                          | Valid |
|   |                          | Skenario Alternatif 1 | Aplikasi menampilkan pesan untuk mengisi kolom yang kosong                                                 | Aplikasi menampilkan pesan untuk mengisi kolom yang kosong                                                 | Valid |
|   |                          | Skenario Alternatif 2 | Aplikasi menampilkan pesan <i>error</i> tentang kesalahan pengisian kolom                                  | Aplikasi menampilkan pesan <i>error</i> tentang kesalahan pengisian kolom                                  | Valid |
| 3 | Lihat Daftar Ujian       | Skenario Utama        | Administrator dibawa ke halaman lihat daftar soal yang menampilkan data daftar soal yang dikelola penguji. | Administrator dibawa ke halaman lihat daftar soal yang menampilkan data daftar soal yang dikelola penguji. | Valid |
| 4 | Buat Ujian               | Skenario Utama        | Administrator berhasil membuat ujian dan data ujian disimpan di <i>database</i>                            | Administrator berhasil membuat ujian dan data ujian disimpan di <i>database</i>                            | Valid |
|   |                          | Skenario Alternatif 1 | Aplikasi menampilkan bahwa kolom belum terisi                                                              | Aplikasi menampilkan bahwa kolom belum terisi                                                              | Valid |
|   |                          | Skenario Alternatif 2 | Aplikasi menampilkan bahwa kolom tidak terisi dengan benar                                                 | Aplikasi menampilkan bahwa kolom tidak terisi dengan benar                                                 | Valid |

|   |                       |                       |                                                                                       |                                                                                       |       |
|---|-----------------------|-----------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------|
| 5 | Hapus Ujian           | Skenario Utama        | Penghapusan berhasil dan data ujian dihapus dari <i>database</i> .                    | Penghapusan berhasil dan data ujian dihapus dari <i>database</i> .                    | Valid |
| 6 | Edit Ujian            | Skenario Utama        | Data ujian yang lama berhasil diganti dengan data ujian yang baru.                    | Data ujian yang lama berhasil diganti dengan data ujian yang baru.                    | Valid |
|   |                       | Skenario Alternatif 1 | Aplikasi menampilkan pesan error bahwa pengubahan kolom tidak benar                   | Aplikasi menampilkan pesan error bahwa pengubahan kolom tidak benar                   | Valid |
| 7 | Lihat Detail Ujian    | Skenario Utama        | Sistem berhasil menampilkan data ujian secara rinci pada halaman lihat detail ujian.  | Sistem berhasil menampilkan data ujian secara rinci pada halaman lihat detail ujian.  | Valid |
| 8 | Tambah Peserta Ujian  | Skenario Utama        | Username yang valid akan ditambahkan sebagai peserta terdaftar dari suatu ujian.      | Username yang valid akan ditambahkan sebagai peserta terdaftar dari suatu ujian.      | Valid |
|   |                       | Skenario Alternatif 1 | Aplikasi menampilkan pesan bahwa telah terjadi kesalahan dalam pengisian nama peserta | Aplikasi menampilkan pesan bahwa telah terjadi kesalahan dalam pengisian nama peserta | Valid |
| 9 | Tampilkan Hasil Ujian | Skenario Utama        | Data hasil ujian berhasil ditampilkan.                                                | Data hasil ujian berhasil ditampilkan.                                                | Valid |
|   |                       | Skenario Alternatif 1 | Aplikasi menampilkan pesan bahwa ujian belum dilaksanakan sehingga tidak              | Aplikasi menampilkan pesan bahwa ujian belum dilaksanakan sehingga tidak              | Valid |

|    |                            |                       |                                                                                                 |                                                                                                 |       |
|----|----------------------------|-----------------------|-------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-------|
|    |                            |                       | ada hasil untuk ditampilkan                                                                     | ada hasil untuk ditampilkan                                                                     |       |
| 10 | Lihat Daftar Soal          | Skenario Utama        | Sistem berhasil menampilkan daftar soal dari suatu ujian.                                       | Sistem berhasil menampilkan daftar soal dari suatu ujian.                                       | Valid |
| 11 | Buat Soal                  | Skenario Utama        | Penambahan soal berhasil dan data soal tersimpan dalam <i>database</i> .                        | Penambahan soal berhasil dan data soal tersimpan dalam <i>database</i> .                        | Valid |
|    |                            | Skenario Alternatif 1 | Aplikasi menampilkan pesan bahwa ada kolom yang belum terisi                                    | Aplikasi menampilkan pesan bahwa ada kolom yang belum terisi                                    | Valid |
|    |                            | Skenario Alternatif 2 | Aplikasi menampilkan pesan <i>error</i> tentang kesalahan pengisian                             | Aplikasi menampilkan pesan <i>error</i> tentang kesalahan pengisian                             | Valid |
| 12 | Hapus Soal                 | Skenario Utama        | Penghapusan berhasil dan data soal dihapus dari <i>database</i> .                               | Penghapusan berhasil dan data soal dihapus dari <i>database</i> .                               | Valid |
| 13 | Edit Soal                  | Skenario Utama        | Data soal yang lama berhasil diganti dengan data yang baru pada <i>database</i> .               | Data soal yang lama berhasil diganti dengan data yang baru pada <i>database</i> .               | Valid |
|    |                            | Skenario Alternatif 1 | Aplikasi menampilkan pesan <i>error</i> bahwa telah terjadi kesalahan ketika mengubah isi kolom | Aplikasi menampilkan pesan <i>error</i> bahwa telah terjadi kesalahan ketika mengubah isi kolom | Valid |
| 14 | Tampilkan Profil Pengguna  | Skenario Utama        | Sistem berhasil menampilkan profil pengguna.                                                    | Sistem berhasil menampilkan profil pengguna.                                                    | Valid |
| 15 | Lihat <i>Request</i> Ujian | Skenario Utama        | Sistem berhasil menampilkan                                                                     | Sistem berhasil menampilkan                                                                     | Valid |

|    |                              |                       |                                                                                                                                            |                                                                                                                                            |       |
|----|------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|-------|
|    |                              |                       | data <i>request</i> ujian yang dikirimkan oleh peserta ujian                                                                               | data <i>request</i> ujian yang dikirimkan oleh peserta ujian                                                                               |       |
| 16 | <i>Approve Request</i> Ujian | Skenario Utama        | Peserta berhasil terdaftar sebagai peserta ujian yang sah.                                                                                 | Peserta berhasil terdaftar sebagai peserta ujian yang sah.                                                                                 | Valid |
| 17 | <i>Reject Request</i> Ujian  | Skenario Utama        | Permohonan ujian dari peserta berhasil dihapus                                                                                             | Permohonan ujian dari peserta berhasil dihapus                                                                                             | Valid |
| 18 | Login Peserta                | Skenario Utama        | Sistem mengirimkan respon token sebagai tanda bahwa peserta berhasil login.                                                                | Sistem mengirimkan respon token sebagai tanda bahwa peserta berhasil login.                                                                | Valid |
|    |                              | Skenario Alternatif 1 | Aplikasi menampilkan pesan bahwa terdapat kolom yang belum terisi                                                                          | Aplikasi menampilkan pesan bahwa terdapat kolom yang belum terisi                                                                          | Valid |
|    |                              | Skenario Alternatif 2 | Aplikasi menampilkan pesan <i>error</i> bahwa <i>username</i> atau <i>password</i> tidak valid                                             | Aplikasi menampilkan pesan <i>error</i> bahwa <i>username</i> atau <i>password</i> tidak valid                                             | Valid |
| 19 | Registrasi Peserta           | Skenario Utama        | Registrasi berhasil dan data peserta ujian tersimpan di <i>database</i> dan sistem memberikan konfirmasi bahwa proses registrasi berhasil. | Registrasi berhasil dan data peserta ujian tersimpan di <i>database</i> dan sistem memberikan konfirmasi bahwa proses registrasi berhasil. | Valid |
|    |                              | Skenario Alternatif 1 | Aplikasi mengirimkan respon bahwa                                                                                                          | Aplikasi mengirimkan respon bahwa                                                                                                          | Valid |

|    |              |                       |                                                                                                                         |                                                                                                                         |       |
|----|--------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-------|
|    |              |                       | parameter tidak lengkap dan data tidak tersimpan                                                                        | parameter tidak lengkap dan data tidak tersimpan                                                                        |       |
|    |              | Skenario Alternatif 2 | Aplikasi mengirimkan respon bahwa nilai parameter tidak valid                                                           | Aplikasi mengirimkan respon bahwa nilai parameter tidak valid                                                           | Valid |
| 20 | Daftar Ujian | Skenario Utama        | Permohonan ujian berhasil disimpan ke database dan sistem memberikan konfirmasi bahwa permohonan telah berhasil dikirim | Permohonan ujian berhasil disimpan ke database dan sistem memberikan konfirmasi bahwa permohonan telah berhasil dikirim | Valid |
|    |              | Skenario Alternatif 1 | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                   | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                   | Valid |
|    |              | Skenario Alternatif 2 | Aplikasi mengirimkan pesan <i>error</i> bahwa token tidak valid                                                         | Aplikasi mengirimkan pesan <i>error</i> bahwa token tidak valid                                                         | Valid |
|    |              | Skenario Alternatif 3 | Aplikasi mengirimkan pesan <i>error</i> bahwa tidak kode enroll tidak valid                                             | Aplikasi mengirimkan pesan <i>error</i> bahwa tidak kode enroll tidak valid                                             | Valid |
|    |              | Skenario Alternatif 4 | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta telah mengirim permohonan dan tidak dapat mengirim                | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta telah mengirim permohonan dan tidak dapat mengirim                | Valid |

|    |                            |                       |                                                                                                           |                                                                                                           |       |
|----|----------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|-------|
|    |                            |                       | kembali                                                                                                   | kembali                                                                                                   |       |
| 21 | Lihat Daftar Ujian Peserta | Skenario Utama        | Sistem berhasil mengirimkan data ujian yang mana peserta telah terdaftar.                                 | Sistem berhasil mengirimkan data ujian yang mana peserta telah terdaftar.                                 | Valid |
|    |                            | Skenario Alternatif 1 | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                     | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                     | Valid |
|    |                            | Skenario Alternatif 2 | Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.                                         | Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.                                         | Valid |
| 22 | Request Soal               | Skenario Utama        | Sistem berhasil mengirimkan data ujian dan data soal dan memulai pelaksanaan ujian bagi peserta tersebut. | Sistem berhasil mengirimkan data ujian dan data soal dan memulai pelaksanaan ujian bagi peserta tersebut. | Valid |
|    |                            | Skenario Alternatif 1 | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                     | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                     | Valid |
|    |                            | Skenario Alternatif 2 | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                  | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                  | Valid |
|    |                            | Skenario Alternatif 3 | Aplikasi mengirimkan pesan bahwa id_ujian tidak valid.                                                    | Aplikasi mengirimkan pesan bahwa id_ujian tidak valid.                                                    | Valid |
|    |                            | Skenario              | Aplikasi                                                                                                  | Aplikasi                                                                                                  | Valid |

|    |               |                       |                                                                                                                                     |                                                                                                                                     |       |
|----|---------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------|
|    |               | Alternatif 4          | mengirimkan pesan <i>error</i> bahwa ujian tidak dapat dilaksanakan.                                                                | mengirimkan pesan <i>error</i> bahwa ujian tidak dapat dilaksanakan.                                                                |       |
|    |               | Skenario Alternatif 5 | Aplikasi mengirimkan pesan bahwa peserta tidak memiliki hak untuk mengikuti ujian tersebut.                                         | Aplikasi mengirimkan pesan bahwa peserta tidak memiliki hak untuk mengikuti ujian tersebut.                                         | Valid |
|    |               | Skenario Alternatif 6 | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta telah me- <i>request</i> soal dan tidak dapat melakukannya lagi.              | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta telah me- <i>request</i> soal dan tidak dapat melakukannya lagi.              | Valid |
| 23 | Kirim Jawaban | Skenario Utama        | Jawaban berhasil disimpan di <i>database</i> dan dikelola. Sistem kemudian memberikan konfirmasi bahwa pengiriman jawaban berhasil. | Jawaban berhasil disimpan di <i>database</i> dan dikelola. Sistem kemudian memberikan konfirmasi bahwa pengiriman jawaban berhasil. | Valid |
|    |               | Skenario Alternatif 1 | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                               | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                                                               | Valid |
|    |               | Skenario Alternatif 2 | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                                            | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                                                            | Valid |

|    |                     |                       |                                                                                                                              |                                                                                                                              |       |
|----|---------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|-------|
|    |                     | Skenario Alternatif 3 | Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                                                     | Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                                                     | Valid |
|    |                     | Skenario Alternatif 4 | Aplikasi mengirimkan pesan <i>error</i> bahwa skema jawaban tidak valid                                                      | Aplikasi mengirimkan pesan <i>error</i> bahwa skema jawaban tidak valid                                                      | Valid |
|    |                     | Skenario Alternatif 5 | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta belum memulai ujian                                                    | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta belum memulai ujian                                                    | Valid |
|    |                     | Skenario Alternatif 6 | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta sudah pernah mengirimkan jawaban dan tidak bisa mengirimkannya kembali | Aplikasi mengirimkan pesan <i>error</i> bahwa peserta sudah pernah mengirimkan jawaban dan tidak bisa mengirimkannya kembali | Valid |
|    |                     | Skenario Alternatif 7 | Aplikasi mengirimkan pesan <i>error</i> bahwa waktu pengerjaan telah habis dan peserta tidak dapat mengumpulkan jawaban      | Aplikasi mengirimkan pesan <i>error</i> bahwa waktu pengerjaan telah habis dan peserta tidak dapat mengumpulkan jawaban      | Valid |
| 24 | Request Hasil Ujian | Skenario Utama        | Sistem berhasil mengirimkan data hasil ujian yang diminta oleh peserta                                                       | Sistem berhasil mengirimkan data hasil ujian yang diminta oleh peserta                                                       | Valid |
|    |                     | Skenario Alternatif 1 | Aplikasi mengirimkan                                                                                                         | Aplikasi mengirimkan                                                                                                         | Valid |

|    |             |                       |                                                                                                     |                                                                                                     |       |
|----|-------------|-----------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-------|
| 25 | Batal Ujian |                       | pesan <i>error</i> bahwa parameter tidak lengkap                                                    | pesan <i>error</i> bahwa parameter tidak lengkap                                                    |       |
|    |             | Skenario Alternatif 2 | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                            | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                            | Valid |
|    |             | Skenario Alternatif 3 | Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                            | Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                            | Valid |
|    |             | Skenario Utama        | Pembatalan ujian berhasil dan peserta berhasil dihapus dari daftar peserta ujian yang bersangkutan. | Pembatalan ujian berhasil dan peserta berhasil dihapus dari daftar peserta ujian yang bersangkutan. | Valid |
|    |             | Skenario Alternatif 1 | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                               | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap                               | Valid |
|    |             | Skenario Alternatif 2 | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                            | Aplikasi mengirimkan pesan <i>error</i> bahwa token peserta tidak valid.                            | Valid |
|    |             | Skenario Alternatif 3 | Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                            | Aplikasi mengirimkan pesan <i>error</i> bahwa ujian tersebut tidak valid                            | Valid |
|    |             | Skenario Alternatif 4 | Aplikasi mengirimkan pesan <i>error</i>                                                             | Aplikasi mengirimkan pesan <i>error</i>                                                             | Valid |

|    |                       |                       |                                                                        |                                                                        |       |
|----|-----------------------|-----------------------|------------------------------------------------------------------------|------------------------------------------------------------------------|-------|
|    |                       |                       | bahwa peserta bukan ujian sehingga tidak perlu mengundurkan diri.      | bahwa peserta bukan ujian sehingga tidak perlu mengundurkan diri.      |       |
| 26 | <i>Request Profil</i> | Skenario Utama        | Sistem berhasil mengirimkan data profil peserta                        | Sistem berhasil mengirimkan data profil peserta                        | Valid |
|    |                       | Skenario Alternatif 1 | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap. | Aplikasi mengirimkan pesan <i>error</i> bahwa parameter tidak lengkap. | Valid |
|    |                       | Skenario Alternatif 2 | Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.      | Aplikasi mengirimkan pesan error bahwa token peserta tidak valid.      | Valid |

## 6.2 Pengujian Non Fungsional

Pengujian non fungsional bertujuan untuk mengetahui apakah aplikasi yang dikembangkan telah memenuhi kebutuhan non fungsional yang ada pada sistem. Pada penelitian ini, kebutuhan non fungsional dari aplikasi adalah *security* dan *reliability*.

### 6.2.1 Pengujian *Security*

Pengujian *security* menguji aspek keamanan dari sistem. Berdasarkan kebutuhan ini, aplikasi hanya mengizinkan pihak-pihak tertentu yang bisa masuk ke dalam sistem. Administrator maupun peserta ujian harus menggunakan *username* dan *password* yang valid untuk masuk ke dalam sistem. Selain itu, untuk setiap *request* ke *web service*, harus menggunakan token yang unik untuk masing-masing peserta ujian. Pengujian ini dilakukan dengan cara mensimulasikan setiap skenario pada proses login Administrator dan Peserta ujian dan proses *request* aplikasi *client* ke *web service*. Pengujian ini ditujukan untuk menguji keamanan dasar dari aplikasi. Pengujian keamanan menggunakan teknik *sql injection* juga dilakukan untuk menguji keamanan aplikasi. Pengujian ini dilakukan dengan mencoba memasukkan karakter-karakter ilegal seperti (') ke dalam url maupun form-form yang berada di dalam Aplikasi.

Tabel 6.27 menjabarkan skenario yang akan diujikan pada pengujian non fungsional.

**Tabel 6.27 Skenario Pengujian Non Fungsional**

| No | Nama Kasus Uji         | Skenario                                                                                                               |
|----|------------------------|------------------------------------------------------------------------------------------------------------------------|
| 1. | Login Administrator    | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang valid.                                            |
|    |                        | - Mencoba login menggunakan <i>username</i> yang tidak valid dan <i>password</i> yang valid.                           |
|    |                        | - Mencoba login menggunakan <i>username</i> yang valid dan <i>password</i> yang tidak valid.                           |
|    |                        | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang tidak valid.                                      |
| 2  | Login Peserta          | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang valid.                                            |
|    |                        | - Mencoba login menggunakan <i>username</i> yang tidak valid dan <i>password</i> yang valid.                           |
|    |                        | - Mencoba login menggunakan <i>username</i> yang valid dan <i>password</i> yang tidak valid.                           |
|    |                        | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang tidak valid.                                      |
| 3  | Request ke Web Service | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token dan ip address yang valid                        |
|    |                        | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token yang tidak valid dan ip address yang tidak valid |
|    |                        | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token yang tidak valid dan ip address yang valid       |
|    |                        | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token yang valid dan ip address yang tidak valid       |

Setelah mensimulasikan skenario pada Tabel 6.27, maka didapatkan hasil pengujian *security*. Tabel 6.28 menjabarkan hasil pengujian *security* Aplikasi Backend Sistem Ujian Online Terpadu.

**Tabel 6.28** Hasil Pengujian *Security*

| Nama Kasus Uji      | Skenario                                                                                     | Hasil yang diharapkan                         | Hasil yang diperoleh                          | Validitas |
|---------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------|-----------|
| Login Administrator | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang valid.                  | Pengguna berhasil login sebagai Administrator | Pengguna berhasil login sebagai Administrator | Valid     |
|                     | - Mencoba login menggunakan <i>username</i> yang tidak valid dan <i>password</i> yang valid. | Pengguna tidak dapat login                    | Pengguna tidak dapat login                    | Valid     |
|                     | - Mencoba login menggunakan <i>username</i> yang valid dan <i>password</i> yang tidak valid. | Pengguna tidak dapat login                    | Pengguna tidak dapat login                    | Valid     |
|                     | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang tidak valid.            | Pengguna tidak dapat login                    | Pengguna tidak dapat login                    | Valid     |
| Login Peserta       | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang valid.                  | Pengguna berhasil login sebagai peserta ujian | Pengguna berhasil login sebagai peserta ujian | Valid     |

|                        |                                                                                              |                                                                                                                 |                                                                                                                 |       |
|------------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-------|
|                        | - Mencoba login menggunakan <i>username</i> yang tidak valid dan <i>password</i> yang valid. | Pengguna tidak dapat login                                                                                      | Pengguna tidak dapat login                                                                                      | Valid |
|                        | - Mencoba login menggunakan <i>username</i> yang valid dan <i>password</i> yang tidak valid. | Pengguna tidak dapat login                                                                                      | Pengguna tidak dapat login                                                                                      | Valid |
|                        | - Mencoba login menggunakan <i>username</i> dan <i>password</i> yang tidak valid.            | Pengguna tidak dapat login                                                                                      | Pengguna tidak dapat login                                                                                      | Valid |
| Request ke Web Service | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token yang valid             | Server mengembalikan respon dengan status <i>true</i>                                                           | Server mengembalikan respon dengan status <i>true</i>                                                           | Valid |
|                        | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token yang tidak valid.      | Server merespon dengan status <i>false</i> dan <i>resource</i> yang diminta oleh <i>client</i> tidak dikirimkan | Server merespon dengan status <i>false</i> dan <i>resource</i> yang diminta oleh <i>client</i> tidak dikirimkan | Valid |
|                        | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan                              | Server merespon dengan status <i>false</i>                                                                      | Server merespon dengan status <i>false</i>                                                                      | Valid |

|  |                                                                                                                  |                                                                                                                 |                                                                                                                 |       |
|--|------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|-------|
|  | token yang tidak valid dan ip address yang valid                                                                 | dan <i>resource</i> yang diminta oleh <i>client</i> tidak dikirimkan                                            | dan <i>resource</i> yang diminta oleh <i>client</i> tidak dikirimkan                                            |       |
|  | - Mencoba melakukan <i>request</i> ke <i>web service</i> dengan token yang valid dan ip address yang tidak valid | Server merespon dengan status <i>false</i> dan <i>resource</i> yang diminta oleh <i>client</i> tidak dikirimkan | Server merespon dengan status <i>false</i> dan <i>resource</i> yang diminta oleh <i>client</i> tidak dikirimkan | Valid |

### 6.2.2 Pengujian *Reliability*

Pengujian *reliability* menguji kebutuhan non fungsional dari aplikasi. Pada penelitian ini, aplikasi harus mampu mengirimkan soal dan menerima jawaban dari peserta secara sekaligus (*concurrency*). Pengujian ini dilakukan dengan melakukan pengukuran waktu dan jumlah *request* yang berhasil dilayani pada level *concurrency* yang berbeda-beda dalam waktu 10 detik. Pengujian dilakukan hingga sistem tidak dapat lagi merespon *request*. 100 *concurrent user* akan terus ditambahkan selama sistem masih dapat merespon. Untuk melakukan hal tersebut, pengujian menggunakan ab tools atau *apache bench tools*. Ab adalah alat yang sering digunakan untuk menguji kehandalan dari suatu aplikasi berdasarkan waktu respon dan jumlah *client* yang bisa dilayani. Tabel 6.29 adalah skenario pengujian *reliability* menggunakan ab tools.

**Tabel 6.29 Skenario Pengujian *Reliability***

| No Skenario | <i>Concurrent User</i> | Jumlah <i>Request</i> | Batas Waktu Pengujian (detik) |
|-------------|------------------------|-----------------------|-------------------------------|
| 1           | 1                      | 50.000                | 10                            |
| 2           | 100                    | 50.000                | 10                            |
| 3           | 200                    | 50.000                | 10                            |
| 4           | 300                    | 50.000                | 10                            |
| 5           | 400                    | 50.000                | 10                            |
| 6           | 500                    | 50.000                | 10                            |
| 7           | 600                    | 50.000                | 10                            |

*Concurrent user* adalah jumlah pengguna yang melakukan *request* secara bersamaan dalam suatu waktu. Jumlah *request* adalah jumlah *request* yang dilakukan. Batas waktu pengujian digunakan agar pengujian tidak membutuhkan waktu yang sangat lama.

Pengujian dilakukan di *localhost*. Spesifikasi dari *server* dijelaskan pada Tabel 6.30.

**Tabel 6.30 Spesifikasi server**

| Nama Komponen     | Spesifikasi                                                |
|-------------------|------------------------------------------------------------|
| <i>Processor</i>  | Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz (4 CPUs), ~2.1GHz |
| <i>Memory</i>     | 4GB RAM                                                    |
| <i>Web Server</i> | Apache                                                     |

Proses pengujian menggunakan skenario 1 sampai 7 ditunjukkan pada Gambar 6.2 hingga 6.8.

```

Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 10 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      57 bytes

Concurrency Level:    1
Time taken for tests:  11.002 seconds
Complete requests:    10
Failed requests:      0
Write errors:         0
Non-2xx responses:    0
Total transferred:    5470 bytes
HTML transferred:     570 bytes
Requests per second:  0.91 [#/sec] (mean)
Time per request:     1100.165 [ms] (mean)
Time per request:     1100.165 [ms] (mean, across all concurrent requests)
Transfer rate:        0.49 [Kbytes/sec] received

Connection Times (ms)
      min    mean[+/-sdl]  median    max
Connect:    0         0      0.0         0
Processing: 1078      1100    16.9    1094    1125
Waiting:    1078      1100    16.9    1094    1125
Total:      1078      1100    16.9    1094    1125

Percentage of the requests served within a certain time (ms)
 50%    1094
 66%    1110
 75%    1110
 80%    1125
 90%    1125
 95%    1125
 98%    1125
 99%    1125
100%    1125 (longest request)
  
```

**Gambar 6.2 Pengujian menggunakan skenario 1**

```

Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 345 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      57 bytes

Concurrency Level:    100
Time taken for tests:  10.007 seconds
Complete requests:    345
Failed requests:       0
Write errors:         0
Non-2xx responses:    345
Total transferred:    188715 bytes
HTML transferred:     17665 bytes
Requests per second:  34.47 [#/sec] (mean)
Time per request:     2900.679 [ms] (mean)
Time per request:     29.007 [ms] (mean, across all concurrent requests)
Transfer rate:        18.42 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0        0   2.4      31
Processing: 1125    2486  520.3   2474   4283
Waiting:    1125    2486  520.4   2474   4283
Total:      1125    2486  520.4   2474   4283

Percentage of the requests served within a certain time (ms)
 50%    2474
 66%    2583
 75%    2704
 80%    2829
 90%    3141
 95%    3376
 98%    3611
 99%    3954
100%    4283 (longest request)

```

Gambar 6.3 Pengujian menggunakan skenario 2

```

Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 303 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      57 bytes

Concurrency Level:    200
Time taken for tests:  10.155 seconds
Complete requests:    303
Failed requests:       0
Write errors:         0
Non-2xx responses:    303
Total transferred:    165741 bytes
HTML transferred:     17271 bytes
Requests per second:  29.84 [#/sec] (mean)
Time per request:     6703.152 [ms] (mean)
Time per request:     33.516 [ms] (mean, across all concurrent requests)
Transfer rate:        15.94 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0        0   2.0      16
Processing: 1156    4636  1719.0   5142   7593
Waiting:    1156    4636  1719.0   5142   7593
Total:      1156    4636  1718.9   5142   7593

Percentage of the requests served within a certain time (ms)
 50%    5142
 66%    5408
 75%    5970
 80%    6048
 90%    6508
 95%    6961
 98%    7381
 99%    7428
100%    7593 (longest request)

```

Gambar 6.4 Pengujian menggunakan skenario 3

```
Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 298 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      57 bytes

Concurrency Level:    300
Time taken for tests:  10.152 seconds
Complete requests:    298
Failed requests:       0
Write errors:          0
Non-2xx responses:    298
Total transferred:    163006 bytes
HTML transferred:     16986 bytes
Requests per second:  29.35 [#/sec] (mean)
Time per request:     10220.163 [ms] (mean)
Time per request:     34.067 [ms] (mean, across all concurrent requests)
Transfer rate:        15.68 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median   max
Connect:    0      0  4.1      0    63
Processing: 1203  5032 2250.6   5675  8902
Waiting:    1203  5031 2250.9   5675  8902
Total:      1203  5032 2250.5   5675  8902

Percentage of the requests served within a certain time (ms)
 50%    5675
 66%    6379
 75%    6769
 90%    7413
 90%    8051
 95%    8394
 98%    8519
 99%    8621
100%    8902 (longest request)
```

Gambar 6.5 Pengujian menggunakan skenario 4

```
Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 295 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      57 bytes

Concurrency Level:    400
Time taken for tests:  12.237 seconds
Complete requests:    295
Failed requests:       5
  (Connect: 5, Receive: 0, Length: 0, Exceptions: 0)
Write errors:          0
Non-2xx responses:    295
Total transferred:    161365 bytes
HTML transferred:     16815 bytes
Requests per second:  24.11 [#/sec] (mean)
Time per request:     16592.823 [ms] (mean)
Time per request:     41.482 [ms] (mean, across all concurrent requests)
Transfer rate:        12.88 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median   max
Connect:    0      0  2.2      0    16
Processing: 4354  8498 2159.4   8167 12097
Waiting:    4244  6441 1917.0   8120  8167
Total:      4354  8499 2159.2   8167 12097

Percentage of the requests served within a certain time (ms)
 50%    8167
 66%    9329
 75%   10409
 80%   10409
 90%   12081
 95%   12081
 98%   12097
 99%   12097
100%   12097 (longest request)
```

Gambar 6.6 Pengujian menggunakan skenario 5

```
Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 275 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      57 bytes

Concurrency Level:    500
Time taken for tests:  14.323 seconds
Complete requests:     275
Failed requests:       6
  (Connect: 6, Receive: 0, Length: 0, Exceptions: 0)
Write errors:          0
Non-2xx responses:     275
Total transferred:     150425 bytes
HTML transferred:      15675 bytes
Requests per second:   19.20 [#/sec] (mean)
Time per request:      26042.320 [ms] (mean)
Time per request:      52.085 [ms] (mean, across all concurrent requests)
Transfer rate:         10.26 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-std] median  max
Connect:    0      1   3.6      0    31
Processing: 7938 10899 1991.6 11403 14120
Waiting:    7782 7867  58.6   7860  8001
Total:      7938 10900 1991.6 11403 14120

Percentage of the requests served within a certain time (ms)
 50% 11403
 66% 11419
 75% 13011
 80% 13011
 90% 13011
 95% 14120
 98% 14120
 99% 14120
100% 14120 (longest request)
```

Gambar 6.7 Pengujian menggunakan skenario 6

```
Administrator: Command Prompt

Benchmarking localhost (be patient)
Finished 0 requests

Server Software:      Apache/2.4.3
Server Hostname:      localhost
Server Port:          80

Document Path:        /oes/api/mulai_ujian/token/667a4c7b1ef981ccc01f4637c7b01
Document Length:      0 bytes

Concurrency Level:    600
Time taken for tests:  10.834 seconds
Complete requests:     0
Failed requests:       5
  (Connect: 5, Receive: 0, Length: 0, Exceptions: 0)
Write errors:          0
Non-2xx responses:     388
Total transferred:     212236 bytes
HTML transferred:      22116 bytes

C:\xampp\apache\bin>
```

Gambar 6.8 Pengujian menggunakan skenario 7

Setelah melakukan pengujian menggunakan skenario pada Tabel 6.29, didapatkan hasil waktu yang dibutuhkan untuk masing-masing skenario. Tabel 6.31 adalah hasil dari pengujian *reliability*.

Tabel 6.31 Hasil Pengujian *Reliability*

| No Skenario | Concurrent User | Time Per Request (mean) | Request Per Second | Completed Request | Failed Request |
|-------------|-----------------|-------------------------|--------------------|-------------------|----------------|
| 1           | 1               | 1101 ms                 | 0,91               | 10                | 0              |
| 2           | 100             | 2900 ms                 | 34,47              | 345               | 0              |
| 3           | 200             | 6703 ms                 | 29,84              | 303               | 0              |
| 4           | 300             | 10220 ms                | 29,35              | 298               | 0              |
| 5           | 400             | 16592 ms                | 24,11              | 295               | 5              |
| 6           | 500             | 26042 ms                | 19,20              | 275               | 6              |
| 7           | 600             | -                       | -                  | -                 | 5              |

Pada Tabel 6.30, *Time per request* adalah waktu yang dibutuhkan untuk memenuhi satu *request*. *Request Per Second* adalah jumlah *request* yang dilayani setiap detik. *Complete Request* adalah jumlah total *request* yang berhasil dilayani dalam batas waktu pengujian (10 detik), Sedangkan *Failed Request* adalah jumlah total *request* yang gagal dilayani dalam batas waktu pengujian (10 detik).

### 6.3 Analisis Hasil Pengujian

Proses analisis terhadap hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi *backend* sistem ujian online terpadu yang telah dilakukan. Proses analisis mengacu pada hasil pengujian yang telah didapatkan. Analisis dilakukan di setiap tahap pengujian yang meliputi analisis hasil pengujian validasi, analisis pengujian *security* dan analisis hasil pengujian *reliability*.

#### 6.3.1 Analisis Hasil Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian fungsi yang diuji dengan hasil perancangan sistem. Berdasarkan hal tersebut, dapat disimpulkan bahwa aplikasi *backend* sistem ujian online terpadu telah memenuhi seluruh kebutuhan fungsional aplikasi yang telah dirancang pada tahap analisis dan perancangan.

#### 6.3.2 Analisis Hasil Pengujian *Security*

Proses analisis terhadap hasil pengujian *security* dilakukan dengan melihat apakah aspek keamanan pada aplikasi telah terpenuhi berdasarkan kebutuhan non fungsional sistem. Berdasarkan hal tersebut, dapat diambil kesimpulan bahwa aplikasi *backend* sistem ujian online terpadu hanya dapat diakses oleh pihak-pihak yang memiliki wewenang dan hanya dapat mengirimkan layanan untuk pihak dengan hak akses saja. Aplikasi juga telah aman dari serangan *sql injection*.

### 6.3.3 Analisis Hasil Pengujian *Reliability*

Proses analisis terhadap hasil pengujian *reliability* dilakukan dengan mengamati hasil pengujian *reliability* yang telah dilakukan sebelumnya. Berdasarkan hasil pengujian, dapat diketahui bahwa waktu rata-rata dalam proses *request* soal adalah 1 detik dan jumlah maksimal *request* dari *client* yang bisa dilayani secara bersamaan pada satu waktu adalah 500 *request*. Waktu respon berbanding lurus dengan banyaknya jumlah *request*. Semakin banyak *request*, semakin lama pula waktu respon dari *server*. Kecepatan proses dari sistem bergantung pada berbagai faktor. Seperti spesifikasi dan konfigurasi perangkat keras *server*, jaringan, desain *database*, arsitektur sistem, dan lain-lain.



## BAB 7 PENUTUP

### 7.1 Kesimpulan

Berdasarkan hasil analisis perancangan, implementasi dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan pengujian validasi, seluruh kebutuhan fungsional aplikasi yang telah dirancang, termasuk fungsi pengacakan soal, manajemen soal dan koreksi jawaban secara terpadu, telah berhasil diimplementasikan.
2. Berdasarkan hasil pengujian validasi, *security* dan *reliability*, aplikasi telah berhasil menyediakan layanan bagi *client* secara aman dan *reliable*.
3. Berdasarkan pengujian *reliability*, diketahui bahwa jumlah maksimal *concurrent user* yang dapat dilayani oleh aplikasi dalam satu waktu adalah 500 dengan waktu rata-rata 1 detik.
4. Berdasarkan pengujian *security*, aplikasi telah dikatakan aman dari pihak-pihak yang tidak bertanggung jawab yang mencoba mengakses sistem dengan menerapkan pengamanan dengan *password*, *token* dan *ip address*.

### 7.2 Saran

1. Diperlukan penelitian lebih lanjut untuk *client* dari aplikasi *backend* sistem ujian online terpadu pada perangkat *mobile*.
2. Diperlukan pengembangan lebih lanjut agar sistem dapat mengakomodasi jenis soal yang lebih beragam.
3. Diperlukan pengembangan lebih lanjut agar sistem dapat memasukkan audio maupun video ke dalam soal.

## DAFTAR PUSTAKA

- Pramudita, Hafiz Ridha. 2013. *Perancangan Aplikasi Mobile Learning Pada Lembaga Bimbingan Belajar Newtonsix Educenter Berbasis Android*.
- Sukmawati, Muliani. 2014. *Rancang Bangun Aplikasi Simulasi TOEFL Berbasis Android*.
- Pratondo, Agus. Idestio, Barsyah Dwi. 2010. *Pembangunan Aplikasi Ujian Akhir Semester Online Untuk Mengukur Pencapaian Kompetensi Peserta Didik Studi Kasus : Politeknik Telkom*.
- E-marketer, <http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536>. [Diakses 21 April 2015].
- Cluskey Jr, G.R. Ehlen, Craig R, Raiborn, Mitchell H. *Thwarting online exam cheating without proctor supervision*. Diunduh pada tanggal 06-07-2015.
- Yu, Qi. Bouguettaya, Athman. *Foundations for Efficient Web Service Selection*. New York: Springer, 2009. Diunduh pada tanggal 26-5-2015.
- Richardson, Leonard. Ruby, Sam. *RESTful Web Services*. USA : O'Reilly Media, 2007. Diunduh pada tanggal 25-5-2015.
- Kudi, Pooja. Dhattrak, Tejaswini. Daware, Kavita. Manekar, Amitkumar. 2015. *Survey On: Theory Online Examination with Short Text Matching*.
- Hamilton, Kim. Miles, Russell. *Learning UML 2.0*. USA : O'Reilly Media, 2006. Diunduh pada tanggal 28-10-2014.
- w3school, [http://www.w3schools.com/webservices/ws\\_soap\\_intro.asp](http://www.w3schools.com/webservices/ws_soap_intro.asp). [Diakses 3 Juni 2015].
- JSON.org. *Introducing JSON*. <http://json.org/>. [Diakses 14 Desember 2014]
- Ellislab. *User guide*. <https://ellislab.com/codeigniter/user-guide/overview/features.html>. [Diakses 25 Juni 2015].
- IBM. <http://www.ibm.com/developerworks/rational/library/3101.html>. [Diakses 23 Juni 2015].
- Learnline. <http://learnline.cdu.edu.au/studyskills/studyskills/differentexams.html>. [Diakses 23 Juni 2015].
- Sletten, Brian. 2010. *REST:Foundations of RESTful Architecture*. DZone, Inc.
- Mitchell, Lorna J., 2013, *PHP Web Service*, O'Reilly Media, Inc., Sebastopol.
- Susanto, Antony. Henky, Honggo. 2013, *Perancangan Ujian Online pada STMIK GI MDP Berbasis Web*, STMIK GI MDP.
- Wikipedia. Fisher-Yates Shuffle. [http://en.wikipedia.org/wiki/Fisher-Yates\\_shuffle](http://en.wikipedia.org/wiki/Fisher-Yates_shuffle). [Diakses 10 September 2015].

(Sangeeta, 2015) Sangeeta Srivastava. 2013. *A Repository of Software Requirement Patterns for Online Examination System*. Delhi University, Delhi, India.

(Sommerville, 2004) Ian Sommerville. 2004. *Software Engineering* 7<sup>th</sup> edition. prechelt@inf.fu-berlin.de

