

BAB 4 REKAYASA KEBUTUHAN

Bab ini akan menjelaskan tentang proses yang dilakukan dalam rekayasa kebutuhan pembangunan Sistem Ujian Harian Siswa. Rekayasa kebutuhan dilakukan untuk menentukan kebutuhan-kebutuhan yang harus dimiliki sistem. Dalam proses rekayasa kebutuhan juga akan menjelaskan proses interaksi antara sistem dan entitas di luar sistem. Bab ini akan menjelaskan proses rekayasa kebutuhan sampai pemodelan kebutuhan.

4.1 Proses Rekayasa Kebutuhan

Proses rekayasa kebutuhan dilakukan untuk mengumpulkan kebutuhan-kebutuhan yang diperlukan dalam membangun Sistem Ujian Harian Siswa Berbasis *Web*. Tahapan dalam proses ini terdiri dari elisitasi kebutuhan atau analisis kebutuhan dan spesifikasi kebutuhan. Proses elisitasi kebutuhan dilakukan untuk mendapatkan kebutuhan sistem. Elisitasi kebutuhan dilakukan dengan cara melakukan identifikasi aktor yang akan berperan dalam sistem, hasil elisitasi kebutuhan, serta proses bisnis ujian harian yang selama ini dilaksanakan. Kemudian spesifikasi kebutuhan yang merupakan spesifikasi hasil dari elisitasi kebutuhan. Spesifikasi kebutuhan terdiri dari kebutuhan fungsional dan nonfungsional sistem yang kemudian akan dilakukan pengecekan kualitas berdasarkan ISO 25010. ISO 25010 digunakan sebagai acuan standar kualitas dari Sistem Ujian Harian Siswa.

4.1.1 Elisitasi Kebutuhan

1. Hasil Elisitasi Kebutuhan

Dalam wawancara, guru menjelaskan tentang ujian harian yang selama ini telah berjalan memiliki beberapa kelemahan seperti misalnya saat ujian pilihan ganda guru hanya bisa membuat variasi type soal paling banyak 3 variasi, sehingga dengan sedikitnya variasi type soal ini memudahkan siswa untuk saling berbagi jawaban dengan teman yang lainnya. Selain itu proses penilaian jawaban siswa, rekap nilai ujian hingga pengumuman nilai yang didapat siswa dianggap memakan waktu. Dengan berkembangnya teknologi maka solusi berupa sistem ujian harian berbasis *web* dapat digunakan untuk menyelesaikan permasalahan yang ada serta dengan didukungnya sumber daya jaringan internet yang ada di wilayah sekolah. Dengan adanya sistem ini maka proses ujian akan menjadi lebih mudah karena siswa dapat melakukan ujian dalam batas waktu yang telah ditentukan dan hasil ujiannya akan keluar saat itu juga sesaat setelah siswa selesai mengerjakan ujian serta guru sudah tidak perlu lagi melakukan penilaian manual pada jawaban siswa. Agar sistem dapat berjalan sesuai dengan yang diharapkan maka sistem yang dibangun harus bisa mengacak soal sebanyak mungkin, menilai jawaban siswa sesaat setelah selesai mengerjakan ujian dan saat ujian berlangsung koneksi internet selain untuk membuka *web* ujian harian akan diputus. Selain itu aktor yang dapat menggunakan sistem ini terdiri dari admin, guru dan siswa. Admin memiliki peran untuk mengelola data pengguna sistem yaitu admin, guru dan

siswa. Yang dimaksud mengelola disini adalah seperti melakukan perubahan, penambahan dan penghapusan data. Peran admin lainnya adalah menambahkan mata pelajaran dan kelas yang diampu oleh masing-masing guru agar guru dapat membuat ujian berdasarkan mata pelajaran dan kelas yang diampunya serta menambahkan siswa kedalam kelas yang ditempuhnya. Selain itu admin juga berperan menambahkan mata pelajaran dan kelas yang ada ke dalam sistem. Guru memiliki peran untuk mengelola bank soal dan membuat ujian harian untuk siswa yang diampunya. Selain itu guru juga dapat melihat hasil ujian siswa serta rekap hasil ujian siswa pada masing-masing mata pelajaran dan kelas yang diampunya. Siswa memiliki peran mengerjakan ujian harian sesuai dengan kelas yang ditempuhnya dan siswa dapat melihat hasil ujian dan rekap hasil ujian yang telah dikerjakannya.

2. Identifikasi Aktor

Dalam proses identifikasi aktor ini bertujuan untuk memberikan gambaran pengguna yang akan terlibat dalam penggunaan sistem. Pada Tabel 4.1 akan menjelaskan siapa saja aktor yang akan menggunakan sistem.

Tabel 4.1 Identifikasi aktor

Aktor	Deskripsi
Guru	Aktor yang dapat melakukan aktivitas <i>login</i> , <i>logout</i> , mengubah profilnya, mengelola ujian harian sesuai dengan kelas dan mata pelajaran yang diampunya, mengelola bank soal, melihat daftar nama siswa yang diampunya, melihat hasil ujian siswa dan melihat rekap hasil ujian siswa.
Siswa	Aktor yang dapat melakukan aktivitas <i>login</i> , <i>logout</i> , mengubah profilnya, mengerjakan ujian harian sesuai dengan kelas yang ditempuhnya, melihat hasil ujian harian dan melihat rekap hasil ujian yang telah dilaksanakannya
Admin	Aktor yang dapat melakukan aktivitas <i>login</i> , <i>logout</i> , mengelola data pengguna sistem (admin, guru, siswa), menambahkan mata pelajaran dan kelas yang diampu oleh masing-masing guru, menambahkan siswa kedalam kelas yang ditempuhnya, menambahkan mata pelajaran, menambahkan kelas

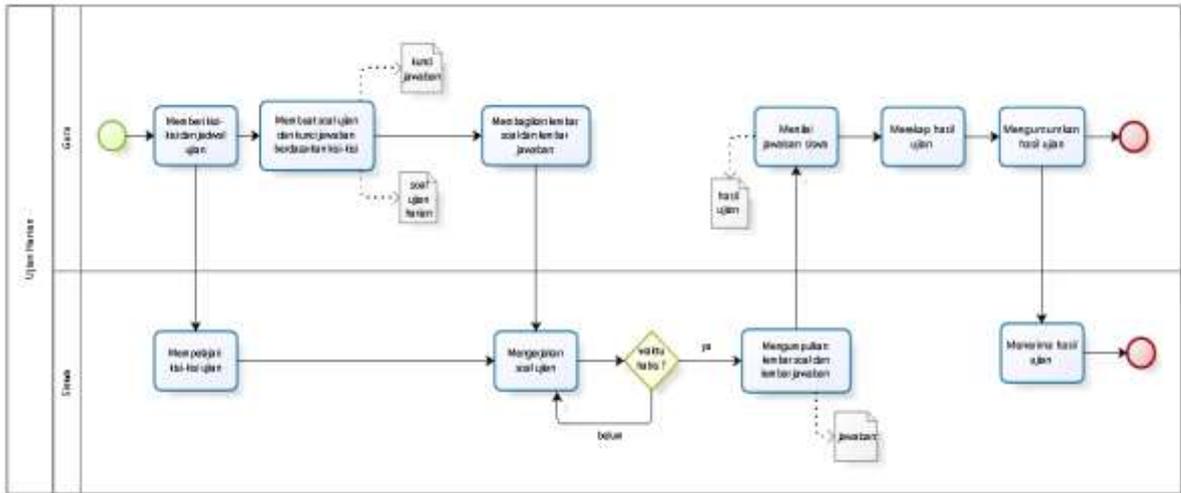
3. Analisis Proses Bisnis

Dalam pelaksanaan ujian harian di SMPN 3 Nglegok masih dilakukan secara manual yaitu dimulai dari guru memberikan kisi-kisi ujian harian kepada siswa dan menginformasikan kepada siswa waktu untuk melaksanakan ujian harian. Kemudian dari kisi-kisi tersebut guru membuat soal ujian. Kemudian ketika ujian, guru membagikan lembar soal dan jawaban dan siswa mengerjakan ujian sesuai waktu yang ditentukan. Saat ujian berakhir maka siswa harus menyerahkan lembar soal dan jawaban, namun apabila waktu masih tersisa maka siswa dapat mengoreksi jawaban kembali sampai waktu selesai. Setelah itu guru melakukan

koreksi jawaban siswa dan merekap nilai. Kemudian di pertemuan berikutnya guru akan memberitahu siswa nilai ujian harian yang telah didapatkan. Perhitungan nilai ujian harian pilihan ganda dapat dilihat pada Persamaan 4.1.

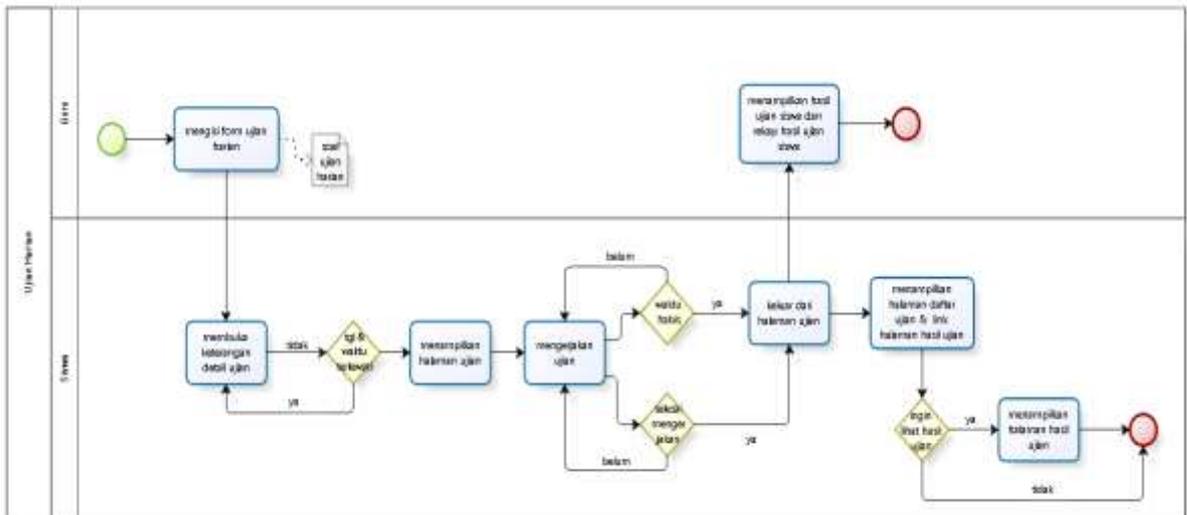
$$Nilai = \frac{\text{jumlah jawaban benar}}{\text{total soal}} \times 100\% \quad (4.1)$$

Proses bisnis ujian harian yang ada di SMPN 3 Nglegok dapat dilihat pada Gambar 4.1.



Gambar 4.1 Proses bisnis ujian harian manual

Dari gambaran proses bisnis Gambar 6.1 yang terlalu panjang maka dalam pembangunan sistem akan diminimalkan menjadi seperti pada Gambar 4.2.



Gambar 4.2 Proses bisnis sistem ujian harian

4.1.2 Daftar Kebutuhan Sistem

Dalam mendefinisikan kebutuhan sistem ini yang menjadi acuan adalah hasil dari proses elisitasi kebutuhan sebelumnya. Kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan nonfungsional. Seluruh kebutuhan fungsional yang didefinisikan dapat digolongkan kedalam karakteristik *functional suitability* pada ISO 25010. Kebutuhan fungsional akan diberi kode SUHS-F-XX, spesifikasi kebutuhan fungsional akan diberi kode SUHS-F-XX-Y dan kebutuhan nonfungsional diberi kode SUHS-NF-XX. SUHS merupakan singkatan dari Sistem Ujian Harian Siswa, F merupakan singkatan Fungsional, NF merupakan singkatan Nonfungsional, XX menunjukkan nomor dari definisi kebutuhan, dan Y menunjukkan nomor dari spesifikasi kebutuhan. Pada Tabel 4.2 dapat dilihat hasil definisi dan spesifikasi kebutuhan fungsional sistem dan Tabel 4.3 hasil kebutuhan nonfungsional sistem.

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem

Kode	Kebutuhan Fungsional	Use case	Aktor
SUHS-F-01	Sistem harus mampu menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan cara memasukkan <i>username</i> dan <i>password</i>	<i>Login</i>	Pengunjung
SUHS-F-02	Sistem harus mampu menyediakan fungsi <i>logout</i> agar pengguna yang sudah masuk ke dalam sistem dapat keluar dari sistem	<i>Logout</i>	Admin, Guru, Siswa
SUHS-F-03	Sistem harus mampu menampilkan profil siswa agar siswa dapat melihat profil pribadinya	Melihat profil siswa	Siswa
SUHS-F-03-1	Data profil yang ditampilkan meliputi nomor induk, <i>email</i> , <i>password</i> , nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon, kelas, tanggal masuk		
SUHS-F-04	Sistem harus mampu menampilkan profil guru agar guru dapat melihat profil pribadinya	Melihat profil guru	Guru

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

SUHS-F-04-1	Data profil yang ditampilkan meliputi nomor induk, <i>email</i> , <i>password</i> , nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon		
SUHS-F-05	Sistem harus mampu mengubah data yang ada pada profil siswa	Mengubah profil siswa	Siswa
SUHS-F-05-1	Sistem harus mampu menampilkan halaman untuk mengubah profil siswa		
SUHS-F-06	Sistem harus mampu mengubah data yang ada pada profil guru	Mengubah profil guru	Guru
SUHS-F-06-1	Sistem harus mampu menampilkan halaman untuk mengubah profil guru		
SUHS-F-07	Sistem harus mampu menambahkan admin baru kedalam sistem	Menambah admin	Admin
SUHS-F-07-1	Sistem harus mampu menyediakan halaman untuk menambahkan admin baru yang terdiri dari nomor induk, email, password, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon		
SUHS-F-08	Sistem harus mampu menambahkan guru baru kedalam sistem	Menambah guru	Admin
SUHS-F-08-1	Sistem harus mampu menyediakan halaman untuk menambahkan guru baru yang terdiri dari nomor induk, email, password, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon		
SUHS-F-09	Sistem harus mampu menambahkan siswa baru kedalam sistem	Menambah siswa	Admin

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

SUHS-F-09-1	Sistem harus mampu menyediakan halaman untuk menambahkan siswa baru yang terdiri dari nomor induk, email, password, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon, kelas, tanggal masuk		
SUHS-F-10	Sistem harus mampu mengubah data profil admin	Mengubah admin	Admin
SUHS-F-10-1	Sistem harus mampu menampilkan halaman untuk mengubah profil admin		
SUHS-F-11	Sistem harus mampu mengubah data profil guru	Mengubah guru	Admin
SUHS-F-11-1	Sistem harus mampu menampilkan halaman untuk mengubah profil guru		
SUHS-F-12	Sistem harus mampu mengubah data profil siswa	Mengubah siswa	Admin
SUHS-F-12-1	Sistem harus mampu menampilkan halaman untuk mengubah profil siswa		
SUHS-F-13	Sistem harus mampu menghapus data admin	Menghapus admin	Admin
SUHS-F-13-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus data admin		
SUHS-F-14	Sistem harus mampu menghapus data guru	Menghapus guru	Admin
SUHS-F-14-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus data guru		
SUHS-F-15	Sistem harus mampu menghapus data siswa	Menghapus siswa	Admin
SUHS-F-15-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus data siswa		

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

SUHS-F-16	Sistem harus mampu menambahkan mata pelajaran baru kedalam sistem	Menambah mata pelajaran	Admin
SUHS-F-16-1	Data mata pelajaran yang harus tersedia adalah nama mata pelajaran		
SUHS-F-17	Sistem harus mampu mengubah mata pelajaran	Mengubah mata pelajaran	Admin
SUHS-F-17-1	Data mata pelajaran yang harus tersedia adalah nama mata pelajaran		
SUHS-F-18	Sistem harus mampu menghapus mata pelajaran	Menghapus mata pelajaran	Admin
SUHS-F-18-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus mata pelajaran		
SUHS-F-19	Sistem harus mampu menambahkan kelas baru kedalam sistem	Menambah kelas	Admin
SUHS-F-19-1	Data kelas yang harus tersedia adalah nama kelas		
SUHS-F-20	Sistem harus mampu mengubah kelas	Mengubah kelas	Admin
SUHS-F-20-1	Data kelas yang harus tersedia adalah nama kelas		
SUHS-F-21	Sistem harus mampu menghapus kelas	Menghapus kelas	Admin
SUHS-F-21-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus kelas		
SUHS-F-22	Sistem harus mampu menambahkan guru kedalam mata pelajaran dan kelas yang diampunya	Memasukkan guru ke mata pelajaran dan kelas	Admin
SUHS-F-22-1	Data yang harus tersedia adalah nama guru, nama mata pelajaran dan nama kelas		

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

SUHS-F-23	Guru harus disediakan fasilitas untuk menambahkan soal dan jawaban baru ke dalam bank soal	Menambah soal dan jawaban	Guru
SUHS-F-23-1	Data yang harus tersedia meliputi mata pelajaran, level, soal, deskripsi soal, opsi jawaban dan jawaban yang benar		
SUHS-F-23-2	Pada saat memasukkan data, guru harus disediakan tombol simpan untu menyimpan soal dan jawaban		
SUHS-F-24	Sistem harus mampu menyediakan fasilitas untuk mengubah data soal dan jawaban	Mengubah soal dan jawaban	Guru
SUHS-F-24-1	Data yang harus tersedia meliputi mata pelajaran, level, soal, deskripsi soal, opsi jawaban dan jawaban yang benar		
SUHS-F-24-2	Pada saat memasukkan data, guru harus disediakan tombol simpan untu menyimpan perubahan soal dan jawaban		
SUHS-F-25	Sistem harus mampu menghapus soal dan jawaban	Menghapus soal dan jawaban	Guru
SUHS-F-25-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus soal dan jawaban		
SUHS-F-26	Sistem harus mampu menampilkan mata pelajaran dan kelas yang diampu oleh guru	Melihat mata pelajaran dan kelas	Guru
SUHS-F-26-1	Data yang harus tersedia yaitu nama mata pelajaran dan kelas yang diampu oleh guru		
SUHS-F-27	Guru harus disediakan fasilitas untuk membuat ujian harian	Membuat ujian harian	Guru
SUHS-F-27-1	Data yang harus tersedia meliputi nama ujian, mata pelajaran, deskripsi ujian, waktu mulai, waktu selesai, waktu mengerjakan ujian, berapa kali diperbolehkan		

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

	mengerjakan ujian, nilai minimal lulus, nilai benar, nilai salah, diijinkan melihat jawaban benar setelah ujian, kelas, soal		
SUHS-F-27-2	Pada saat memasukkan data, guru harus disediakan tombol simpan untu menyimpan ujian harian		
SUHS-F-28	Sistem harus mampu menyediakan fasilitas untuk mengubah data ujian harian	Mengubah ujian harian	Guru
SUHS-F-28-1	Data yang harus tersedia meliputi nama ujian, mata pelajaran, deskripsi ujian, waktu mulai, waktu selesai, waktu mengerjakan ujian, berapa kali diperbolehkan mengerjakan ujian, nilai minimal lulus, nilai benar, nilai salah, diijinkan melihat jawaban benar setelah ujian, kelas, soal		
SUHS-F-29	Sistem harus mampu menghapus ujian harian	Menghapus ujian harian	Guru
SUHS-F-29-1	Sistem harus mampu menyediakan tombol hapus untuk menghapus ujian harian		
SUHS-F-30	Sistem harus mampu menampilkan hasil ujian siswa pada tiap bab dalam mata pelajaran dan kelas yang diampu oleh guru	Melihat hasil ujian siswa	Guru
SUHS-F-30-1	Data yang harus tersedia yaitu nama siswa, status ujian lulus atau tidak, nilai		
SUHS-F-31	Sistem harus mampu menampilkan rekap hasil ujian siswa dalam mata pelajaran dan kelas yang diampu oleh guru	Melihat rekap hasil ujian siswa	Guru
SUHS-F-31-1	Data yang harus tersedia yaitu nama siswa, nilai pada masing-masing bab, rata-rata		

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

SUHS-F-32	Sistem harus mampu menampilkan daftar mata pelajaran yang memiliki ujian yang sedang ditempuh oleh siswa	Melihat daftar mata pelajaran	Siswa
SUHS-F-32-1	Data yang harus tersedia yaitu nama mata pelajaran		
SUHS-F-33	Sistem harus mampu menampilkan daftar ujian yang terdapat pada setiap mata pelajaran yang sedang ditempuh oleh siswa	Melihat daftar ujian	Siswa
SUHS-F-33-1	Data yang harus tersedia yaitu nama ujian, waktu mengerjakan, jumlah soal, diijinkan mengerjakan berapa kali		
SUHS-F-34	Siswa harus disediakan fasilitas untuk mengerjakan ujian harian	Mengerjakan ujian	Siswa
SUHS-F-34-1	Data yang harus tersedia meliputi nama bab, soal, jawaban		
SUHS-F-34-2	Pada saat mengerjakan ujian, siswa harus disediakan <i>timer</i> untuk mengetahui sisa waktu mengerjakan ujian, tombol hapus jawaban untuk menghapus jawaban, tombol kembali, tombol selanjutnya, tombol selesai untuk menyimpan ujian, <i>mapping</i> soal untuk melihat soal yang telah dikerjakan dan belum dikerjakan		
SUHS-F-35	Sistem harus mampu menampilkan hasil ujian siswa kepada siswa setelah mengerjakan ujian	Melihat hasil ujian	Siswa
SUHS-F-35-1	Data yang harus tersedia yaitu jumlah jawaban benar, total waktu mengerjakan, kapan waktu mengerjakan ujian, nilai, status ujian		

Tabel 4.2 Definisi dan spesifikasi kebutuhan fungsional sistem (lanjutan)

SUHS-F-36	Sistem harus mampu menampilkan rekap hasil ujian siswa dalam satu semester	Melihat rekap hasil ujian	Siswa
SUHS-F-36-1	Data yang harus tersedia yaitu nama siswa, nama ujian, status, nilai, hasil		

Tabel 4.3 Kebutuhan nonfungsional sistem

Kode	Kebutuhan Nonfungsional	Karakteristik ISO 25010
SUHS-NF-01	Sistem dapat melakukan perintah dari pengguna dengan batas waktu respon di bawah 10 detik	<i>Performance efficiency</i>
SUHS-NF-02	Sistem dapat dijalankan di <i>web browser</i> yang berbeda	<i>Compatibility</i>
SUHS-NF-03	Sistem mempunyai tampilan <i>user friendly</i> sehingga mudah digunakan	<i>Usability</i>
SUHS-NF-04	Sistem dapat berfungsi dengan baik walaupun pengguna sistem sewaktu-waktu membeludak	<i>Reliability</i>
SUHS-NF-05	Sistem dapat diakses kapanpun	
SUHS-NF-06	Sistem hanya dapat diakses oleh pengguna yang telah terdaftar	<i>Security</i>

Untuk memastikan kualitas dari Sistem Ujian Harian Siswa yang dibangun maka diambil standar kualitas ISO 25010 sebagai standar kualitas Sistem Ujian Harian Siswa. Dari spesifikasi kebutuhan, didapatkan kebutuhan fungsional Sistem Ujian Harian Siswa masuk kedalam karakteristik *functional suitability* sedangkan kebutuhan nonfungsional masuk dalam karakteristik *performance efficiency*, *compatibility*, *reliability*, *usability*, dan *security* yang ada pada standar ISO 25010. Standar ISO 25010 yang diambil disini didasarkan pada kebutuhan pengguna sistem.

4.2 Model Kebutuhan

Model kebutuhan digunakan untuk memudahkan proses pembacaan kebutuhan pada tahap perancangan. Dalam pemodelan kebutuhan ini menggunakan diagram *Unified Modelling Language (UML)* yang terdiri dari *use case diagram*, *use case scenario*, *sequence diagram*, dan *class diagram*.

4.2.1 Use Case Diagram

Use case diagram adalah diagram yang memodelkan perilaku sistem dari tampak luar. *Use case diagram* menggambarkan tentang aktor, *use case*, serta hubungan antara keduanya dalam sistem. Gambar 4.3 yang menjelaskan tentang diagram *use case* dari pembangunan sistem.

4.2.2 Use Case Scenario

Pada setiap *use case* yang dimodelkan akan dijelaskan secara detail dalam *use case scenario*. Dalam tabel *use case scenario* terdiri dari kode *use case*, nama *use case*, aktor yang menggunakan *use case*, tujuan serta deskripsi dari *use case*, *scenario* utama serta *scenario* alternatif yang digunakan ketika menggunakan *use case*. Berikut penjelasan spesifik tentang *use case diagram* dalam *use case scenario*.

1. Use Case Login

Tabel 4.4 Use case scenario login

Skenario Kasus pada Sistem	
Actor	Pengunjung
Objective	Sebagai proses identifikasi pengguna sistem, agar sistem mampu membedakan pengguna yang mengakses sistem sebagai admin, guru atau siswa.
Pre-Condition	Aktor mengakses halaman <i>Login</i>
Main Flow	1. Aktor memasukkan <i>username</i> dan <i>password</i> 2. Aktor mengklik tombol <i>login</i> 3. Sistem mengidentifikasi <i>username</i> dan <i>password</i>
Alternative Flow	1. Jika kolom <i>username</i> atau <i>password</i> kosong, maka sistem akan menampilkan peringatan " <i>Plis fill out this field</i> " 2. Jika <i>username</i> dan <i>password</i> salah, maka sistem akan menampilkan " <i>Email or password invalid or email address not verified</i> "
Post-Condition	<i>Login</i> berhasil

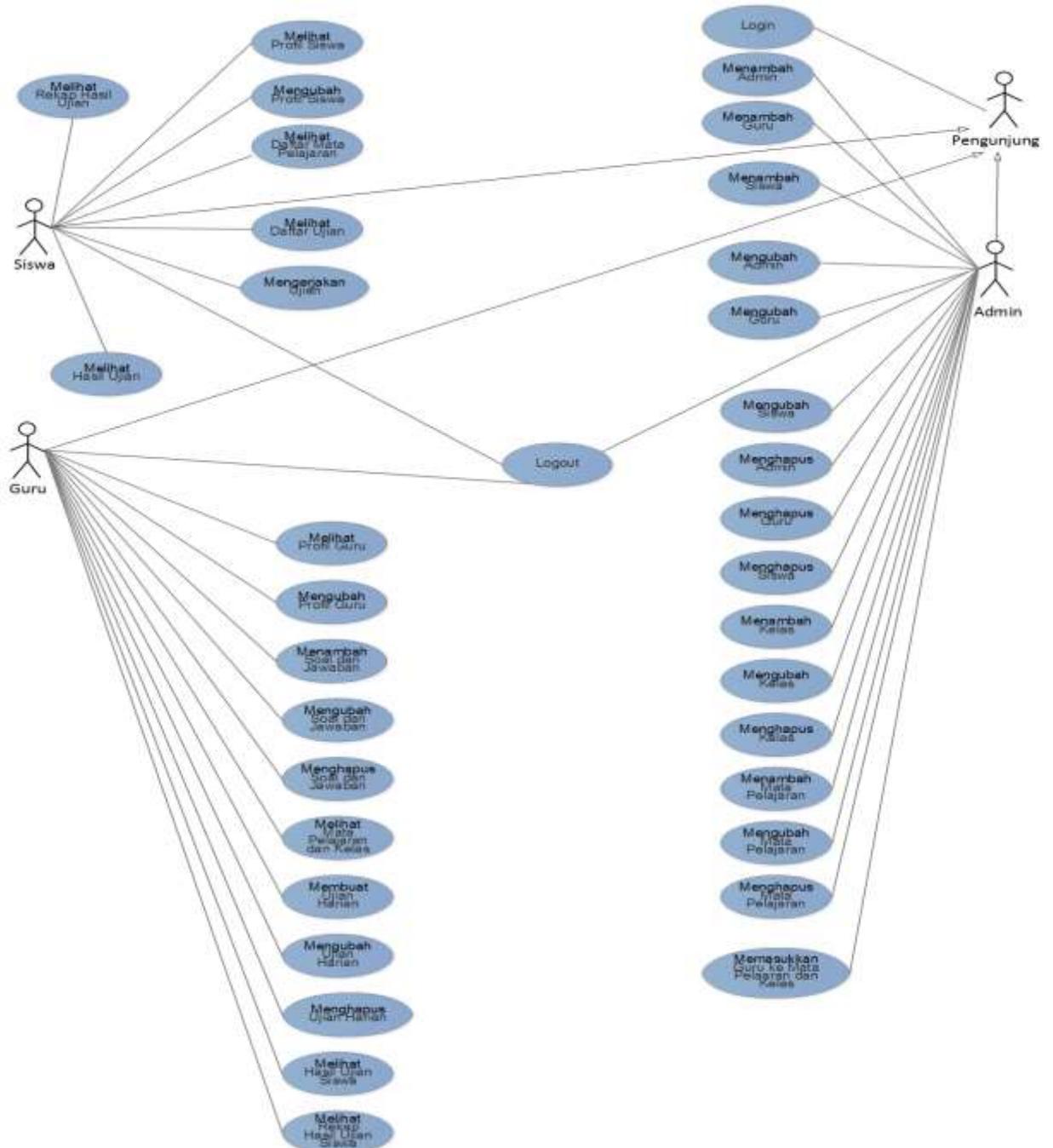
2. Use Case Logout

Tabel 4.5 Use case scenario logout

Skenario Kasus pada Sistem	
Actor	Admin, Guru, Siswa
Objective	Fungsi dapat keluar dari sistem

Tabel 4.5 Use case scenario logout (lanjutan)

Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan tombol " <i>logout</i> " 2. Sistem menampilkan halaman <i>Login</i>
Alternative Flow	-
Post-Condition	Aktor berhasil keluar dari sistem



Gambar 4.3 Use case diagram Sistem Ujian Harian Berbasis Web

3. Use Case Melihat Profil Siswa

Tabel 4.6 Use case scenario melihat profil siswa

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat melihat profil lengkap
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu “profil” 2. Sistem menampilkan profil siswa meliputi nomor induk, email, <i>password</i> , nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon, kelas, tanggal masuk
Alternative Flow	-
Post-Condition	Profil siswa berhasil ditampilkan

4. Use Case Melihat Profil Guru

Tabel 4.7 Use case scenario melihat profil guru

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat melihat profil lengkap
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu “profil” 2. Sistem menampilkan profil guru meliputi nomor induk, email, <i>password</i> , nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon
Alternative Flow	-
Post-Condition	Profil guru berhasil ditampilkan

5. Use Case Mengubah Profil Siswa

Tabel 4.8 Use case scenario mengubah profil siswa

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat memperbarui profilnya
Pre-Condition	Aktor dalam kondisi <i>login</i>

Tabel 4.8 Use case scenario mengubah profil siswa (lanjutan)

Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “profil” 2. Aktor mengubah nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, atau nomor telepon 3. Aktor menekan tombol simpan 4. Sistem menyimpan perubahan data
Alternative Flow	-
Post-Condition	Profil siswa berhasil diperbarui

6. Use Case Mengubah Profil Guru

Tabel 4.9 Use case scenario mengubah profil guru

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat memperbarui profilnya
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “profil” 2. Aktor mengubah nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, atau nomor telepon 3. Aktor menekan tombol simpan 4. Sistem menyimpan perubahan data
Alternative Flow	-
Post-Condition	Profil guru berhasil diperbarui

7. Use Case Menambah Admin

Tabel 4.10 Use case scenario menambah admin

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menambahkan admin baru kedalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “pengguna” 2. Aktor menekan submenu “admin”

Tabel 4.10 Use case scenario menambah admin (lanjutan)

	<p>3. Aktor menekan fitur “tambah admin”</p> <p>4. Aktor mengisi data admin baru berupa nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, dan nomor telepon</p> <p>5. Aktor menekan tombol simpan</p> <p>6. Sistem menyimpan data admin baru</p>
Alternative Flow	<p>1. Aktor menekan menu “pengguna”</p> <p>2. Aktor menekan submenu “admin”</p> <p>3. Aktor menekan fitur “tambah admin”</p> <p>4. Aktor mengisi data admin baru berupa nomor induk, email, <i>password</i>, nama depan</p> <p>5. Aktor menekan tombol simpan</p> <p>6. Sistem menyimpan data admin baru</p>
Post-Condition	Admin baru berhasil ditambahkan

8. Use Case Menambah Guru

Tabel 4.11 Use case scenario menambah guru

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menambahkan guru baru kedalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<p>1. Aktor menekan menu “pengguna”</p> <p>2. Aktor menekan submenu “guru”</p> <p>3. Aktor menekan fitur “tambah guru”</p> <p>4. Aktor mengisi data guru baru berupa nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, dan nomor telepon</p> <p>5. Aktor menekan tombol simpan</p> <p>6. Sistem menyimpan data guru baru</p>
Alternative Flow	<p>1. Aktor menekan menu “pengguna”</p> <p>2. Aktor menekan submenu “guru”</p> <p>3. Aktor menekan fitur “tambah guru”</p>

Tabel 4.11 Use case scenario menambah guru (lanjutan)

	<p>4. Aktor mengisi data guru baru berupa nomor induk, email, <i>password</i>, nama depan</p> <p>5. Aktor menekan tombol simpan</p> <p>6. Sistem menyimpan data guru baru</p>
Post-Condition	Guru baru berhasil ditambahkan

9. Use Case Menambah Siswa

Tabel 4.12 Use case scenario menambah siswa

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menambahkan siswa baru kedalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<p>1. Aktor menekan menu “pengguna”</p> <p>2. Aktor menekan submenu “siswa”</p> <p>3. Aktor menekan fitur “tambah siswa”</p> <p>4. Aktor mengisi data siswa baru berupa nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon, kelas dan tanggal masuk</p> <p>5. Aktor menekan tombol simpan</p> <p>6. Sistem menyimpan data siswa baru</p>
Alternative Flow	<p>1. Aktor menekan menu “pengguna”</p> <p>2. Aktor menekan submenu “siswa”</p> <p>3. Aktor menekan fitur “tambah siswa”</p> <p>4. Aktor mengisi data siswa baru berupa nomor induk, email, <i>password</i>, nama depan, kelas</p> <p>5. Aktor menekan tombol simpan</p> <p>6. Sistem menyimpan data siswa baru</p>
Post-Condition	Siswa baru berhasil ditambahkan

10. Use Case Mengubah Admin

Tabel 4.13 Use case scenario mengubah admin

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat memperbarui data admin yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor menekan menu "pengguna"2. Aktor menekan submenu "admin"3. Aktor menekan tombol "ubah data"4. Aktor memperbarui data admin berupa nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, atau nomor telepon5. Aktor menekan tombol simpan6. Sistem menyimpan perubahan data profil admin
Alternative Flow	-
Post-Condition	Data admin berhasil diperbarui

11. Use Case Mengubah Guru

Tabel 4.14 Use case scenario mengubah guru

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat memperbarui data guru yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor menekan menu "pengguna"2. Aktor menekan submenu "guru"3. Aktor menekan tombol "ubah data"4. Aktor memperbarui data admin berupa nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, atau nomor telepon5. Aktor menekan tombol simpan6. Sistem menyimpan perubahan data profil guru

Tabel 4.14 Use case scenario mengubah guru (lanjutan)

Alternative Flow	-
Post-Condition	Data guru berhasil diperbarui

12. Use Case Mengubah Siswa

Tabel 4.15 Use case scenario mengubah siswa

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat memperbarui data siswa yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “pengguna” 2. Aktor menekan submenu “siswa” 3. Aktor menekan tombol “ubah data” 4. Aktor memperbarui data siswa berupa nomor induk, email, <i>password</i>, nama depan, nama belakang, tanggal lahir, alamat, jenis kelamin, nomor telepon, kelas, dan tanggal masuk 5. Aktor menekan tombol simpan 6. Sistem menyimpan perubahan data profil siswa
Alternative Flow	-
Post-Condition	Data siswa berhasil diperbarui

13. Use Case Menghapus Admin

Tabel 4.16 Use case scenario menghapus admin

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menghapus akun admin yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “pengguna” 2. Aktor menekan submenu “admin” 3. Aktor menekan tombol “hapus”

Tabel 4.16 Use case scenario menghapus admin (lanjutan)

	4. Sistem menghapus data admin
Alternative Flow	-
Post-Condition	Akun admin berhasil dihapus

14. Use Case Menghapus Guru

Tabel 4.17 Use case scenario menghapus guru

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menghapus akun guru yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “pengguna” 2. Aktor menekan submenu “guru” 3. Aktor menekan tombol “hapus” 4. Sistem menghapus data guru
Alternative Flow	-
Post-Condition	Akun guru berhasil dihapus

15. Use Case Menghapus Siswa

Tabel 4.18 Use case scenario menghapus siswa

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menghapus akun siswa yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “pengguna” 2. Aktor menekan submenu “siswa” 3. Aktor menekan tombol “hapus” 4. Sistem menghapus data siswa
Alternative Flow	-
Post-Condition	Akun siswa berhasil dihapus

16. Use Case Menambah Mata Pelajaran

Tabel 4.19 Use case scenario menambah mata pelajaran

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menambahkan mata pelajaran baru kedalam dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor menekan menu "Daftar Mapel"2. Aktor memasukkan nama mata pelajaran baru kedalam text box "mata pelajaran"3. Aktor menekan tombol "tambah mata pelajaran"4. Sistem menyimpan mata pelajaran baru
Alternative Flow	-
Post-Condition	Mata pelajaran baru berhasil ditambahkan

17. Use Case Mengubah Mata Pelajaran

Tabel 4.20 Use case scenario mengubah mata pelajaran

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat mengganti nama mata pelajaran yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor menekan menu "Daftar Mapel"2. Aktor mengetikkan nama mata pelajaran baru di dalam nama mata pelajaran yang ingin dirubah3. Aktor menekan enter4. Sistem menyimpan perubahan nama mata pelajaran
Alternative Flow	-
Post-Condition	Mata pelajaran berhasil diubah

18. Use Case Menghapus Mata Pelajaran

Tabel 4.21 *Use case scenario* menghapus mata pelajaran

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menghapus mata pelajaran yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu "Daftar Mapel" 2. Aktor menekan tombol "X" pada mata pelajaran yang ingin dihapus 3. Sistem menampilkan peringatan "Apakah Anda yakin untuk menghapusnya?" 4. Aktor menekan "OK" 5. Sistem menghapus mata pelajaran
Alternative Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu "Daftar Mapel" 2. Aktor menekan tombol "X" pada mata pelajaran yang ingin dihapus 3. Sistem menampilkan peringatan "Apakah Anda yakin untuk menghapusnya?" 4. Aktor menekan "Cancel" 5. Sistem tidak menghapus mata pelajaran
Post-Condition	Mata pelajaran berhasil dihapus

19. Use Case Menambah Kelas

Tabel 4.22 *Use case scenario* menambah kelas

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menambahkan kelas baru kedalam dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu "Daftar Kelas" 2. Aktor memasukkan nama kelas baru kedalam text box "kelas" 3. Aktor menekan tombol "tambah kelas"

Tabel 4.22 Use case scenario menambah kelas (lanjutan)

	4. Sistem menyimpan kelas baru
Alternative Flow	-
Post-Condition	Kelas baru berhasil ditambahkan

20. Use Case Mengubah Kelas

Tabel 4.23 Use case scenario mengubah kelas

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat mengganti nama kelas yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu "Daftar Kelas" 2. Aktor mengetikkan nama kelas baru di dalam nama kelas yang ingin dirubah 3. Aktor menekan enter 4. Sistem menyimpan perubahan nama kelas
Alternative Flow	-
Post-Condition	Kelas berhasil diubah

21. Use Case Menghapus Kelas

Tabel 4.24 Use case scenario menghapus kelas

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat menghapus kelas yang ada dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu "Daftar Kelas" 2. Aktor menekan tombol "X" pada kelas yang ingin dihapus 3. Sistem menampilkan peringatan "Apakah Anda yakin untuk menghapusnya?" 4. Aktor menekan "OK" 5. Sistem menghapus kelas

Tabel 4.24 Use case scenario menghapus kelas (lanjutan)

Alternative Flow	<ol style="list-style-type: none"> 1. Aktor menekan menu “Daftar Kelas” 2. Aktor menekan tombol “X” pada kelas yang ingin dihapus 3. Sistem menampilkan peringatan “Apakah Anda yakin untuk menghapusnya?” 4. Aktor menekan “Cancel” 5. Sistem tidak menghapus kelas
Post-Condition	Kelas berhasil dihapus

22. Use Case Memasukkan Guru ke Mata Pelajaran dan Kelas

Tabel 4.25 Use case scenario memasukkan guru ke mata pelajaran dan kelas

Skenario Kasus pada Sistem	
Actor	Admin
Objective	Fungsi dapat memasukkan guru kedalam mata pelajaran dan kelas yang diampunya
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu “guru” 2. Aktor menekan tombol “tambah mata pelajaran & kelas” 3. Aktor memilih mata pelajaran dan kelas 4. Aktor menekan tombol simpan 5. Sistem menyimpan mata pelajaran dan kelas yang diampu oleh guru
Alternative Flow	<ol style="list-style-type: none"> 1. Aktor memilih submenu “guru” 2. Aktor menekan tombol “tambah mata pelajaran & kelas” 3. Aktor memilih mata pelajaran dan kelas yang sudah terpakai 4. Aktor menekan tombol simpan 5. Sistem menampilkan halaman pemberitahuan bahwa mata pelajaran dan kelas sudah terpakai
Post-Condition	Mata pelajaran dan kelas berhasil ditambahkan ke daftar mata pelajaran dan kelas yang diampu oleh guru

23. Use Case Menambah Soal dan Jawaban

Tabel 4.26 Use case scenario menambah soal dan jawaban

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat menambahkan soal beserta kunci jawaban ke dalam bank soal
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "bank soal" 2. Aktor menekan submenu "tambah soal" 3. Aktor memilih mata pelajaran, memilih level, memasukkan soal, memasukkan deskripsi soal, memasukkan 4 opsi jawaban dan menentukan jawaban yang benar 4. Aktor menekan tombol simpan 5. Sistem menyimpan soal dan jawaban baru
Alternative Flow	-
Post-Condition	Soal dan jawaban berhasil ditambahkan kedalam daftar bank soal

24. Use Case Mengubah Soal dan Jawaban

Tabel 4.27 Use case scenario mengubah soal dan jawaban

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat memperbaiki soal beserta kunci jawaban yang ada di dalam bank soal
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "bank soal" 2. Aktor menekan submenu "daftar soal" 3. Aktor menekan tombol berbentuk pensil 4. Aktor melakukan perubahan pada pilih mata pelajaran, pilih level, masukkan soal, masukkan deskripsi soal, 4 opsi jawaban atau menentukan jawaban yang benar 5. Aktor menekan tombol simpan 6. Sistem menyimpan perubahan soal dan jawaban

Tabel 4.27 *Use case scenario* mengubah soal dan jawaban (lanjutan)

Alternative Flow	-
Post-Condition	Soal dan jawaban berhasil diperbarui

25. Use Case Menghapus Soal dan Jawaban

Tabel 4.28 *Use case scenario* menghapus soal dan jawaban

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat menghapus soal beserta kunci jawaban yang ada dalam daftar bank soal
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "bank soal" 2. Aktor menekan submenu "tambah soal" 3. Aktor menekan tombol "X" pada soal yang ingin dihapus 4. Sistem menampilkan peringatan "Apakah Anda yakin untuk menghapusnya?" 5. Aktor menekan tombol "OK" 6. Sistem menghapus soal dan jawaban
Alternative Flow	<ol style="list-style-type: none"> 1. Aktor memilih menu "bank soal" 2. Aktor menekan submenu "tambah soal" 3. Aktor menekan tombol "X" pada soal yang ingin dihapus 4. Sistem menampilkan peringatan "Apakah Anda yakin untuk menghapusnya?" 5. Aktor menekan tombol "Cancel" 6. Sistem tidak menghapus soal dan jawaban
Post-Condition	Soal dan jawaban berhasil dihapus

26. Use Case Melihat Mata Pelajaran dan Kelas

Tabel 4.29 Use case scenario melihat mata pelajaran dan kelas

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat melihat mata pelajaran dan kelas yang diampunya
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu “mapel & kelas” 2. Sistem menampilkan mata pelajaran dan kelas
Alternative Flow	-
Post-Condition	Berhasil menampilkan mata pelajaran dan kelas yang diampu

27. Use Case Membuat Ujian Harian

Tabel 4.30 Use case scenario membuat ujian harian

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat membuat ujian harian
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor memilih menu “Ujian” 2. Aktor menekan submenu “tambah ujian” 3. Aktor memasukkan nama ujian, pilih mata pelajaran, dekripsi ujian, waktu mulai, waktu selesai, waktu mengerjakan ujian, diperbolehkan mengerjakan ujian berapa kali, nilai minimal yang harus diperoleh untuk dapat dinyatakan lulus, nilai benar, nilai salah, pilih ijin atau tidak melihat jawaban yang benar setelah melakukan submit ujian, pilih kelas yang diperbolehkan untuk melaksanakan ujian, memasukkan soal 4. Aktor menekan tombol simpan 5. Sistem menyimpan ujian harian baru
Alternative Flow	-
Post-Condition	Ujian harian berhasil ditambahkan

28. Use Case Mengubah Ujian Harian

Tabel 4.31 Use case scenario mengubah ujian harian

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat memperbarui ujian harian yang ada di dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor memilih menu "Ujian"2. Aktor menekan submenu "daftar ujian"3. Aktor menekan tombol pensil pada salah satu soal yang ada4. Aktor memperbarui nama ujian, pilih mata pelajaran, dekripsi ujian, waktu mulai, waktu selesai, waktu mengerjakan ujian, diperbolehkan mengerjakan ujian berapa kali, nilai minimal yang harus diperoleh untuk dapat dinyatakan lulus, nilai benar, nilai salah, pilih ijin atau tidak melihat jawaban yang benar setelah melakukan submit ujian, pilih kelas yang diperbolehkan untuk melaksanakan ujian, atau soal5. Aktor menekan tombol simpan6. Sistem meyimpan perubahan ujian harian
Alternative Flow	-
Post-Condition	Ujian harian berhasil diperbarui

29. Use Case Menghapus Ujian Harian

Tabel 4.32 Use case scenario menghapus ujian harian

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat menghapus ujian harian yang ada di dalam sistem
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<ol style="list-style-type: none">1. Aktor memilih menu "Ujian"2. Aktor menekan submenu "daftar ujian"3. Aktor menekan tombol "X" pada salah satu ujian yang ada

Tabel 4.32 Use case scenario menghapus ujian harian (lanjutan)

	<p>4. Sistem menampilkan peringatan “Apakah Anda yakin untuk menghapusnya?”</p> <p>5. Aktor menekan tombol “OK”</p> <p>6. Sistem menghapus ujian harian</p>
Alternative Flow	<p>1. Aktor memilih menu “Ujian”</p> <p>2. Aktor menekan submenu “daftar ujian”</p> <p>3. Aktor menekan tombol “X” pada salah satu ujian yang ada</p> <p>4. Sistem menampilkan peringatan “Apakah Anda yakin untuk menghapusnya?”</p> <p>5. Aktor menekan tombol “Cancel”</p> <p>6. Sistem tidak menghapus ujian harian</p>
Post-Condition	Ujian harian berhasil dihapus

30. Use Case Melihat Hasil Ujian Siswa

Tabel 4.33 Use case scenario melihat hasil ujian siswa

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat melihat hasil ujian siswa yang diampunya
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	<p>1. Aktor menekan menu “mapel & kelas”</p> <p>2. Aktor menekan submenu mapel&kelas yang diinginkan</p> <p>3. Aktor menekan tombol “lihat” pada ujian yang diinginkan</p> <p>4. Sistem menampilkan hasil ujian siswa pada bab yang dipilih</p>
Alternative Flow	-
Post-Condition	Hasil ujian siswa berhasil ditampilkan

31. Use Case Melihat Rekap Hasil Ujian Siswa

Tabel 4.34 Use case scenario melihat rekap hasil ujian siswa

Skenario Kasus pada Sistem	
Actor	Guru
Objective	Fungsi dapat rekap hasil ujian siswa pada mapel & kelas yang diampunya
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu “mapel & kelas” 2. Aktor menekan submenu mapel&kelas yang diinginkan 3. Sistem menampilkan rekap hasil ujian siswa dalam semester tersebut
Alternative Flow	-
Post-Condition	Rekap hasil ujian siswa berhasil ditampilkan

32. Use Case Melihat Daftar Mata Pelajaran

Tabel 4.35 Use case scenario melihat daftar mata pelajaran

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat melihat daftar mata pelajaran yang sudah terdapat ujian
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu “mata pelajaran” 2. Sistem menampilkan mata pelajaran yang memiliki ujian harian pada semester tersebut
Alternative Flow	-
Post-Condition	Daftar mata pelajaran yang terdapat ujian berhasil ditampilkan

33. Use Case Melihat Daftar Ujian

Tabel 4.36 Use case scenario melihat daftar ujian

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat melihat ujian dalam mata pelajaran yang ditempuhnya
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu "mata pelajaran" 2. Aktor memilih mata pelajaran yang diinginkan 3. Sistem menampilkan ujian harian yang ada pada mata pelajaran yang dipilih
Alternative Flow	-
Post-Condition	Daftar ujian dalam mata pelajaran yang dipilih berhasil ditampilkan

34. Use Case Mengerjakan Ujian

Tabel 4.37 Use case scenario mengerjakan ujian

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat mengerjakan soal yang ada dalam ujian yang dilaksanakan
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu "mata pelajaran" 2. Aktor memilih mata pelajaran 3. Aktor menekan tombol "kerjakan" pada ujian yang akan dikerjakan 4. Aktor menekan tombol "mulai" 5. Sistem menampilkan soal ujian 6. Aktor memilih jawaban 7. Aktor menekan tombol "selesai" 8. Sistem menampilkan peringatan "Apakah Anda yakin telah selesai mengerjakan ujian ini ?" 9. Aktor menekan tombol "selesai"

Tabel 4.37 Use case scenario mengerjakan ujian (lanjutan)

	<p>10. Sistem menyimpan hasil ujian siswa</p> <p>11. Sistem menampilkan <i>link</i> hasil ujian siswa pada halaman daftar ujian harian</p>
Alternative Flow	<p>1. Aktor menekan menu “mata pelajaran”</p> <p>2. Aktor memilih mata pelajaran</p> <p>3. Aktor menekan tombol “kerjakan” pada ujian yang akan dikerjakan</p> <p>4. Aktor menekan tombol “mulai”</p> <p>5. Sistem menampilkan soal ujian</p> <p>6. Aktor memilih jawaban</p> <p>7. Aktor menekan tombol “selesai”</p> <p>8. Sistem menampilkan peringatan “Apakah Anda yakin telah selesai mengerjakan ujian ini ?”</p> <p>9. Aktor menekan tombol “belum”</p> <p>10. Sistem akan kembali menampilkan halaman soal ujian</p>
Post-Condition	Berhasil menyimpan hasil ujian siswa

35. Use Case Melihat Hasil Ujian

Tabel 4.38 Use case scenario melihat hasil ujian

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat melihat hasil ujian
Pre-Condition	Aktor dalam kondisi selesai mengerjakan ujian
Main Flow	<p>1. Aktor menekan tombol “<i>click here</i>”</p> <p>2. Sistem menampilkan halaman hasil ujian</p>
Alternative Flow	-
Post-Condition	Hasil ujian berhasil ditampilkan

36. Use Case Melihat Rekap Hasil Ujian

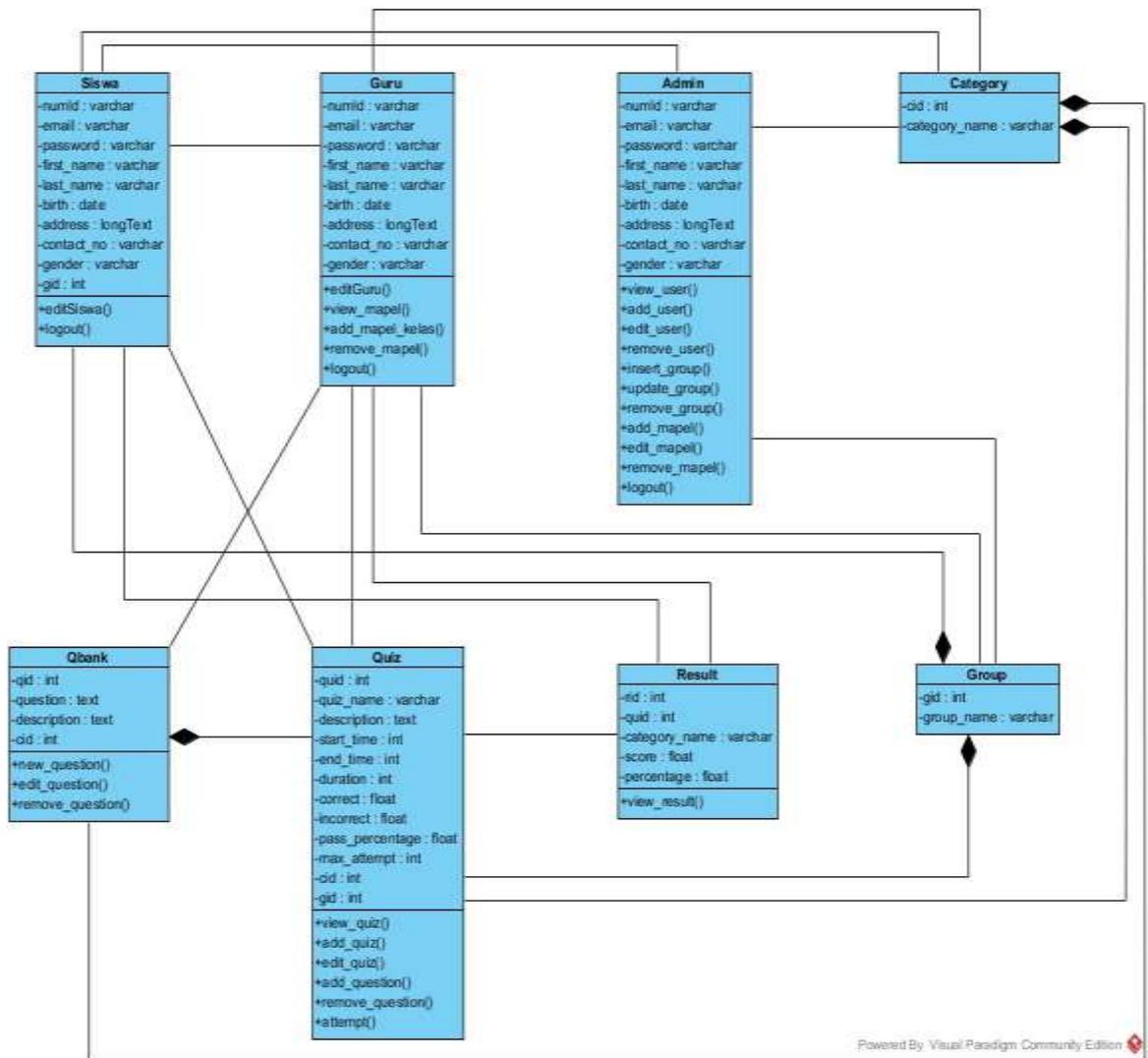
Tabel 4.39 Use case scenario melihat rekap hasil ujian

Skenario Kasus pada Sistem	
Actor	Siswa
Objective	Fungsi dapat melihat rekap hasil ujian yang telah dikerjakan selama ini
Pre-Condition	Aktor dalam kondisi <i>login</i>
Main Flow	1. Aktor menekan menu "rekap nilai" 2. Sistem menampilkan halaman rekap hasil ujian
Alternative Flow	-
Post-Condition	Rekap hasil ujian berhasil ditampilkan

4.2.3 Pemodelan Class Diagram Analisis

Class diagram analisis merupakan *class diagram* yang diperoleh dari hasil analisis kebutuhan fungsional sistem. Kebutuhan fungsional yang ada diklasifikasikan berdasarkan domain permasalahan yang sama untuk dijadikan *candidate class* pada *class diagram* analisis. *Class diagram* analisis nantinya akan dijadikan pedoman dalam pemodelan *class diagram* pada tahapan perancangan.

Dari kebutuhan fungsional yang telah didefinisikan sebelumnya didapatkan domain permasalahan yang memiliki kesamaan untuk dijadikan *candidate class*. *Candidate class* yang diperoleh meliputi *class* siswa, guru, admin, *qbank*, *quiz*, *group*, *category* dan *result*. Sebagian besar *class* dalam *class diagram* ini memiliki hubungan asosiasi namun untuk beberapa *class* terdapat hubungan komposisi seperti *class quiz* memiliki hubungan komposisi dengan *class qbank* yang menggambarkan bahwa *class quiz* bergantung dan tidak bisa berdiri sendiri tanpa adanya *class qbank*. Kemudian *class* siswa yang memiliki hubungan komposisi dengan *class group*, *class qbank* dengan *class category*, *class quiz* dengan *class group* dan *class quiz* dengan *class category*. Pemodelan *class diagram* analisis akan ditunjukkan pada Gambar 4.4.



Gambar 4.4 Pemodelan *class diagram* analisis