HALAMAN PENGESAHAN

PENGEMBANGAN SISTEM KOREKSI LEMBAR JAWAB KOMPUTER DAN ANALISIS HASIL UJIAN (SIKORAN) BERBASIS WEB MENGGUNAKAN METODE *GRAYSCALING-THRESHOLDING-CROPPING* (STUDI KASUS : *TRY OUT* SMPIT AL-GHOZALI JEMBER)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Afandi Ahmad Syaikhu
NIM : 125150400111014

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Desember 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
M. Chandra Saputra, S.Kom., M.Eng	Fajar Pradana, S.ST., M.Eng
NIK : 201609 860106 1 001	NIP : 19871121 201504 1 004

Mengetahui
Ketua Jurusan Sistem Informasi

Herman Tolle, Dr. Eng., S.T, M.T
NIP : 19740823 200012 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, 15 Desember 2016

Afandi Ahmad Syaikhu

NIM : 125150400111014

Semoga penelitian ini dapat menjadi saksi bagi seorang hamba yang mencoba menaati perintah Rabb-nya untuk saling berlomba dalam kebaikan

Semoga usaha ini mampu menjadi bukti bagi seorang pengikut yang mencoba menggenggam erat anjuran Sang Nabi shallallahu `alaihi wasallam tentang menjadi manusia yang bermanfaat bagi sesama

Semoga karya ini dapat merekahkan senyum di wajah orang tua dan keluarga besar penulis

Semoga artefak ini bisa menjadi salah satu tanda keberhasilan pembinaan yang telah dilakukan dibawah sebuah naungan bernama Ibaddurrahman

Semoga tulisan ini dapat menjadi salah satu petunjuk bahwa lembaga Unit Aktivitas Kerohanian Islam Universitas Brawijaya sanggup mengubah seseorang "from zero to hero"

Semoga perbuatan yang telah dirampungkan ini dapat kembali mengobarkan semangat teman-teman yang berhimpun di bawah panji Ikatan Mahasiswa Wringinagung

Semoga pekerjaan yang telah diselesaikan ini bisa membantu melecut gerilya perjuangan yang tengah dirintis oleh para pengajar SMPIT Al Ghozali

Semoga tindakan yang telah diambil ini mampu menginsprasi teman-teman yang berkumpul dalam sebuah lingkaran bernama PREDATOR untuk dapat melakukan tindakan serupa atau bahkan yang lebih baik lagi

KATA PENGANTAR

Segala puji bagi Allah *subhanahu wa ta`ala* atas segala rahmat, hidayah serta karunia yang telah Ia limpahkan sehingga laporan skripsi yang berjudul "Pengembangan Sistem Koreksi Lembar Jawab Komputer dan Analisis Hasil Ujian (SIKORAN) Berbasis Web Menggunakan Metode *Grayscaling-Thresholding-Cropping* (Studi Kasus : Try Out SMPIT Al-Ghozali Jember)" ini dapat terselesaikan.

Penulis menyadari bahwa bahwa penelitian ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada :

1. Bapak Mochamad Chandra Saputra, S.Kom, M.Eng dan Fajar Pradana, S.ST, M.Eng selaku dosen pembimbing yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan penelitian ini.
2. Sudiyanto, S.Pd selaku Kepala SMPIT Al Ghozali yang telah mengizinkan penulis untuk melakukan penelitian di lembaga pendidikan yang beliau pimpin.
3. Bapak Aditya Rahmadi, S.ST, M.TI selaku dosen penasihat akademik yang telah memberikan begitu banyak nasihat kepada penulis selama menempuh masa studi.
4. Orang tua dan keluarga besar penulis yang telah memberikan doa, semangat, nasihat dan dukungan moral lainnya demi terselesaikannya penelitian ini.
5. Seluruh civitas academica Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan begitu banyak bantuan dan dukungan selama penulis menempuh masa studi di Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan, sehingga kritik dan saran yang membangun sangat penulis butuhkan. Semoga penelitian ini dapat membawa manfaat bagi seluruh pihak yang menggunakannya.

Malang, 15 Desember 2016

Penulis

afandi.syaikhu@gmail.com

Repository Universitas Brawijaya

ABSTRAK

Sekolah Menengah Pertama Islam Terpadu Al-Ghozali merupakan sebuah lembaga pendidikan yang secara rutin melaksanakan tes tertulis dalam bentuk soal pilihan ganda kepada peserta didiknya. Tes tersebut membutuhkan lembar jawab komputer (LJK) sebagai media penyimpan jawaban. Selama ini, para pengajar mengoreksi LJK tersebut secara manual sehingga butuh waktu yang panjang dalam proses pengoreksian dan rekapitulasi nilai. Akibat panjangnya waktu yang dibutuhkan, proses penggalian informasi hanya dilaksanakan sampai sebatas mengetahui nilai yang diperoleh peserta didik. Padahal, beberapa aspek seperti peningkatan kualitas soal dan pemilihan metode pengajaran yang tepat dapat dilakukan apabila pengajar mampu melakukan analisis butir soal. Salah satu upaya untuk menyelesaikan permasalahan tersebut adalah mengembangkan sistem koreksi LJK dan analisis hasil ujian berbasis web. Metode yang digunakan untuk mendeteksi jawaban yang terdapat pada LJK adalah *grayscaling-thresholding-cropping*. Pengembangan sistem ini dimulai dengan proses analisis kebutuhan perangkat lunak yang menghasilkan 53 kebutuhan fungsional dan 2 kebutuhan non fungsional. Setelah itu, kebutuhan fungsional akan direpresentasikan dalam bentuk *use case diagram* dan setiap *use case* yang terbentuk akan dideskripsikan menggunakan *use case* skenario. Setelah proses analisis kebutuhan selesai, maka proses pengembangan sistem dilanjutkan pada tahap perancangan. Pada tahap ini dihasilkan beberapa artefak seperti *class diagram*, *sequence diagram*, *physical data model* dan desain antarmuka. Kemudian pengembangan sistem dilanjutkan pada tahap implementasi dan diakhiri dengan pengujian perangkat lunak. Setelah diuji menggunakan teknik *basis path testing* dan *validation testing*, hasil pengujian mengindikasikan bahwa setiap algoritma telah diterapkan secara akurat dan seluruh kebutuhan fungsional yang telah didefinisikan dapat diimplementasikan dengan sempurna.

Kata kunci: pengembangan perangkat lunak, sistem koreksi LJK, analisis butir soal, pengolahan citra, *grayscaling-thresholding-cropping*, sistem berbasis web

Repository Universitas Brawijaya

ABSTRACT

Sekolah Menengah Pertama Islam Terpadu Al-Ghozali *is an educational institution that regularly perform written multiple choice test for their students. Generally, the tests need bubble answer sheets to store the students' answer. All this time, teachers are forced to match those answers manually, which means that they need to spend a lot of time for finishing it. Due to the length of time required, the process of extracting the information from the conducted test can only result the students' scores. Actually, some aspects such as improving the quality of the questions and selecting the more appropriate teaching methods can be done if the teachers are able to analyze each item of the questions. One of the way to solve this problem is developing web-based system which able to automatically match the answers on the bubble answer sheet then analyze the result of the test. The method used to detect stored answers on the bubble answer sheet is grayscaling-thresholding-cropping. This development process begins with software requirement analysis that produces 53 functional requirements and 2 non-functional requirements. Afterward, the functional requirements are represented in a use case diagram and any use case formed will be described using use case scenario. After the software requirement analysis completed, the development process continued to the design phase. Some artifacts like class diagram, sequence diagrams, physical data model and interfaces design are produced at this phase. Then, the development process continued to the implementation phase and ended by testing the software. After tested using basis path testing and validation testing method, the result indicated that each algorithm has been applied accurately and every defined functional requirement was successfully implemented.*

***Keywords**: software development, digital mark reader, item analysis, image processing, grayscaling-thresholding-cropping, web-based system*

DAFTAR ISI

HALAMAN PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
HALAMAN PERSEMBAHAN iv
KATA PENGANTAR v
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xi
DAFTAR GAMBAR xv
DAFTAR LAMPIRAN xvii
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Manfaat 2
1.5 Batasan Masalah 3
1.6 Sistematika Pembahasan 3
BAB 2 LANDASAN KEPUSTAKAAN 4
2.1 Kajian Pustaka 4
2.2 Sistem 4
2.2.1 Definisi Sistem 4
2.2.2 Karakteristik Sistem 5
2.2.3 Definisi Subsistem 6
2.3 Analisis Butir Soal 6
2.3.1 Taraf Kesukaran Soal 7
2.4 *Software Process* 8
2.4.1 *The Waterfall Model* 8
2.5 *Unified Modeling Language* (UML) 10
2.5.1 *Use Case Diagram* 10
2.5.2 *Class Diagram* 11

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

2.5.3 *Sequence Diagram* 12
2.6 Pengujian Perangkat Lunak 13
2.6.1 *Black-box Testing* 13
2.6.2 *White-box Testing* 14
2.7 Pengolahan Citra 16
2.7.1 *Grayscaling* 17
2.7.2 *Thresholding* 17
2.7.3 *Cropping* 18
2.8 PHP Hypertext Preprocessor (PHP) 18
2.8.1 Penggunaan Variabel dalam PHP 19
2.8.2 Penggunaan Fungsi dalam PHP 19
2.9 Laravel Framework 21
BAB 3 METODOLOGI 23
3.1 Tempat dan Waktu Penelitian 24
3.2 Studi Literatur 24
3.3 Analisis Kebutuhan 24
3.3.1 Kebutuhan Fungsional 24
3.3.2 Kebutuhan Non Fungsional 24
3.4 Perancangan Sistem 25
3.5 Implementasi 25
3.6 Pengujian 25
3.7 Kesimpulan 25
BAB 4 analisis Kebutuhan 26
4.1 Penggalian dan Analisis Kebutuhan 26
4.2 Identifikasi Aktor 28
4.3 Spesifikasi Kebutuhan 28
4.3.1 Aturan Penomoran 28
4.3.2 Kebutuhan Fungsional 29
4.3.3 Kebutuhan Non Fungsional 35
4.4 Pemodelan Kebutuhan 36
4.4.1 *Use Case Diagram* 36
4.4.2 *Use Case Scenario* 37

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

BAB 5 Perancangan 81
5.1 Perancangan *Class Diagram* 81
5.2 Perancangan Interaksi Antar Objek 83
5.3 *Physical Data Model* 87
5.4 Perancangan Antarmuka 89
BAB 6 Implementasi 100
6.1 Spesifikasi Sistem 100
6.2 Implementasi Antarmuka 101
6.3 Implementasi Metode *Grayscaling-Thresholding-Cropping* 112
6.4 Implementasi Analisis Taraf Kesukaran Soal 116
BAB 7 Pengujian 122
7.1 Pengujian *White-Box* 122
7.2 Pengujian *Black-Box* 131
7.3 Uji Coba Perangkat Lunak 138
7.3.1 Pengujian Performa 138
7.3.2 Pengujian Akurasi 139
BAB 8 Penutup 140
8.1 Kesimpulan 140
8.2 Saran 141
DAFTAR PUSTAKA 142
LAMPIRAN A WAWANCARA 144
LAMPIRAN B VALIDASI KEBUTUHAN PERANGKAT LUNAK 145

DAFTAR TABEL

Tabel 2.1 Simbol-simbol dalam *use case diagram*..... 10
Tabel 2.2 Simbol-simbol dalam *class diagram*..... 12
Tabel 2.3 Simbol-simbol dalam *sequence diagram*..... 13
Tabel 2.4 Pendefinisian variabel dalam PHP..... 19
Tabel 2.5 Pendefinisian fungsi dalam PHP..... 20
Tabel 2.6 Penggunaan fungsi imagefilter()..... 20
Tabel 2.7 Penggunaan fungsi jpeg2wbmp()..... 21
Tabel 2.8 Penggunaan fungsi imagecrop()..... 21
Tabel 4.1 Kebutuhan sistem secara umum..... 27
Tabel 4.2 Aktor yang dapat berinteraksi dengan sistem..... 28
Tabel 4.3 Kebutuhan fungsional tamu..... 29
Tabel 4.4 Kebutuhan fungsional admin sistem..... 30
Tabel 4.5 Kebutuhan fungsional admin soal..... 32
Tabel 4.6 Kebutuhan fungsional korektor..... 34
Tabel 4.7 Kebutuhan non fungsional sistem..... 36
Tabel 4.8 Skenario login..... 37
Tabel 4.9 Skenario logout..... 38
Tabel 4.10 Skenario lihat mata pelajaran..... 39
Tabel 4.11 Skenario tambah mata pelajaran..... 39
Tabel 4.12 Skenario ubah mata pelajaran..... 41
Tabel 4.13 Skenario nonaktifkan mata pelajaran..... 42
Tabel 4.14 Skenario aktifkan mata pelajaran..... 43
Tabel 4.15 Skenario lihat materi ujian..... 44
Tabel 4.16 Skenario tambah materi ujian..... 44
Tabel 4.17 Skenario ubah materi ujian..... 45
Tabel 4.18 Skenario nonaktifkan materi ujian..... 46
Tabel 4.19 Skenario aktifkan materi ujian..... 47
Tabel 4.20 Skenario lihat kelas..... 48
Tabel 4.21 Skenario tambah kelas..... 49
Tabel 4.22 Skenario ubah kelas..... 49

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

Tabel 4.23 Skenario nonaktifkan kelas.....51
Tabel 4.24 Skenario aktifkan kelas.....51
Tabel 4.25 Skenario lihat identitas siswa.....52
Tabel 4.26 Skenario tambah siswa.....53
Tabel 4.27 Skenario ubah identitas siswa.....54
Tabel 4.28 Skenario nonaktifkan siswa.....55
Tabel 4.29 Skenario aktifkan siswa.....56
Tabel 4.30 Skenario lihat pengguna sistem.....56
Tabel 4.31 Skenario tambah pengguna sistem.....57
Tabel 4.32 Skenario ubah pengguna sistem.....58
Tabel 4.33 Skenario nonaktifkan pengguna.....59
Tabel 4.34 Skenario aktifkan pengguna.....60
Tabel 4.35 Skenario lihat diagram kemampuan siswa.....61
Tabel 4.36 Skenario simpan diagram kemampuan siswa.....62
Tabel 4.37 Skenario lihat diagram tingkat kesukaran soal.....63
Tabel 4.38 Skenario simpan diagram tingkat kesukaran soal.....63
Tabel 4.39 Skenario lihat diagram sebaran nilai.....64
Tabel 4.40 Skenario simpan diagram sebaran nilai.....65
Tabel 4.41 Skenario lihat grafik statistik nilai.....65
Tabel 4.42 Skenario simpan grafik statistik nilai.....66
Tabel 4.43 Skenario lihat hasil ujian kolektif.....67
Tabel 4.44 Skenario cetak hasil ujian kolektif.....67
Tabel 4.45 Skenario lihat hasil ujian per mapel.....68
Tabel 4.46 Skenario cetak hasil ujian per mapel.....69
Tabel 4.47 Skenario buat kunci jawaban.....69
Tabel 4.48 Skenario lihat kunci jawaban.....70
Tabel 4.49 Skenario ubah kunci jawaban.....71
Tabel 4.50 Skenario hapus kunci jawaban.....72
Tabel 4.51 Skenario koreksi jawaban manual.....72
Tabel 4.52 Tabel skenario lihat koreksi saya.....73
Tabel 4.53 Skenario ubah jawaban siswa.....74
Tabel 4.54 Skenario hapus jawaban siswa.....75

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

Tabel 4.55 Skenario koreksi LJK otomatis 76
Tabel 4.56 Skenario lihat lembar jawab komputer 77
Tabel 4.57 Skenario ubah hasil koreksi otomatis 77
Tabel 4.58 Skenario simpan hasil koreksi otomatis 78
Tabel 4.59 Skenario hapus hasil koreksi otomatis 79
Tabel 4.60 Skenario simpan semua hasil koreksi otomatis 79
Tabel 6.1 Spesifikasi perangkat keras 100
Tabel 6.2 Spesifikasi perangkat lunak 100
Tabel 6.3 *Pseudocode* metode *grayscaling-thresholding-cropping* 112
Tabel 6.4 Algoritma untuk menghitung nilai siswa 116
Tabel 6.5 *Pseudocode* implementasi analisis taraf kesukaran soal 117
Tabel 6.6 Hasil *try out* IPA untuk paket soal 01 117
Tabel 6.7 Jawaban siswa *upper group* untuk butir soal nomor 1 119
Tabel 6.8 Jawaban siswa *lower group* untuk butir soal nomor 1 120
Tabel 7.1 *Pseudocode* login 122
Tabel 7.2 *Test case* login 124
Tabel 7.3 *Pseudocode* buat kunci jawaban 125
Tabel 7.4 *Test case* buat kunci jawaban 127
Tabel 7.5 *Pseudocode* koreksi LJK otomatis 127
Tabel 7.6 *Test case* koreksi LJK otomatis 130
Tabel 7.7 Kasus uji login 131
Tabel 7.8 Kasus uji login : *alternative flow* 1a 132
Tabel 7.9 Kasus uji login : *alternative flow* 1b 132
Tabel 7.10 Kasus uji login : *alternative flow* 2a 133
Tabel 7.11 Kasus uji login : *alternative flow* 2b 133
Tabel 7.12 Kasus uji buat kunci jawaban 134
Tabel 7.13 Kasus uji buat kunci jawaban : *alternative flow* 10a 134
Tabel 7.14 Kasus uji koreksi LJK otomatis 135
Tabel 7.15 Kasus uji koreksi LJK otomatis : *alternative flow* 2a 136
Tabel 7.16 Kasus uji koreksi LJK otomatis : *alternative flow* 4a 136
Tabel 7.17 Kasus uji koreksi LJK otomatis : *alternative flow* 5a 137
Tabel 7.18 Kasus uji koreksi LJK otomatis : *alternative flow* 5b 137

Tabel 7.19 Spesifikasi perangkat komputer..... 138
Tabel 7.20 Waktu yang diperlukan untuk memproses 10 LJK..... 139
Tabel 7.21 Perbandingan hasil koreksi menggunakan SIKORAN dengan pengoreksian manual..... 139

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Model sistem menurut Mc Leod 5
Gambar 2.2 Gambaran subsistem dalam sistem 6
Gambar 2.3 *The waterfall model* 9
Gambar 2.4 *Flowchart* (a) dan *flow graph* (b) 15
Gambar 2.5 Citra yang telah mengalami proses *grayscaling* 17
Gambar 2.6 Citra yang telah mengalami proses *thresholding* 17
Gambar 2.7 Citra yang telah mengalami proses croping 18
Gambar 2.8 Tingkat kepopuleran *framework* PHP 22
Gambar 3.1 Diagram alir penelitian 23
Gambar 4.1 Format penomoran kebutuhan 29
Gambar 4.2 *Use case diagram* SIKORAN 37
Gambar 5.1 Perancangan *class diagram* 82
Gambar 5.2 *Sequence diagram* login 83
Gambar 5.3 *Sequence diagram* tambah mata pelajaran 84
Gambar 5.4 *Sequence diagram* buat kunci jawaban 85
Gambar 5.5 *Sequence diagram* lihat tingkat kesukaran soal 85
Gambar 5.6 *Sequence diagram* koreksi LJK otomatis 86
Gambar 5.7 Perancangan *database* sistem 88
Gambar 5.8 Perancangan halaman login 89
Gambar 5.9 Perancangan halaman mata pelajaran 90
Gambar 5.10 Perancangan halaman materi ujian 91
Gambar 5.11 Perancangan halaman kelas 92
Gambar 5.12 Perancangan halaman siswa 92
Gambar 5.13 Perancangan halaman pengguna 93
Gambar 5.14 Perancangan halaman analisis 94
Gambar 5.15 Perancangan halaman hasil ujian kolektif 94
Gambar 5.16 Perancangan halaman hasil ujian per mata pelajaran 95
Gambar 5.17 Perancangan halaman buat kunci jawaban 96
Gambar 5.18 Perancangan halaman lihat kunci jawaban 97
Gambar 5.19 Perancangan halaman koreksi manual 97

Gambar 5.20 Perancangan halaman koreksi otomatis.....98
Gambar 5.21 Perancangan halaman lihat hasil koreksi saya.....99
Gambar 6.1 Implementasi halaman login.....101
Gambar 6.2 Implementasi halaman mata pelajaran.....102
Gambar 6.3 Implementasi halaman materi ujian.....103
Gambar 6.4 Implementasi halaman kelas.....103
Gambar 6.5 Implementasi halaman siswa.....104
Gambar 6.6 Implementasi halaman pengguna.....104
Gambar 6.7 Implementasi halaman analisis.....105
Gambar 6.8 Implementasi halaman hasil ujian kolektif.....106
Gambar 6.9 Implementasi halaman hasil ujian per mata pelajaran.....107
Gambar 6.10 Implementasi halaman buat kunci jawaban.....108
Gambar 6.11 Implementasi halaman lihat kunci jawaban.....108
Gambar 6.12 Implementasi halaman koreksi manual.....109
Gambar 6.13 Implementasi halaman koreksi otomatis.....109
Gambar 6.14 Implementasi halaman lihat koreksi saya.....112
Gambar 6.15 Berkas LJK dalam warna asli.....113
Gambar 6.16 Berkas LJK dalam warna keabuan.....114
Gambar 6.17 Berkas LJK dalam warna monochrome.....115
Gambar 6.18 Hasil *cropping* pada salah satu pilihan jawaban.....116
Gambar 6.19 Diagram tingkat kesukaran soal IPA paket 01.....121
Gambar 7.1 *Flow graph* login.....123
Gambar 7.2 *Flow graph* buat kunci jawaban.....126
Gambar 7.3 *Flow graph* koreksi LJK otomatis.....129

DAFTAR LAMPIRAN

LAMPIRAN A WAWANCARA.....144
LAMPIRAN B VALIDASI KEBUTUHAN PERANGKAT LUNAK.....145

Pengembang tidak harus membuat sebuah aplikasi dengan beberapa macam bahasa pemrograman agar dapat diakses lintas *platform*. Aplikasi yang dikembangkan dalam *platform* web tidak perlu didistribusikan ke setiap *user*, cukup dengan menginstall aplikasi pada *server* maka setiap user yang terhubung dengan *server* tersebut dapat langsung mengakses aplikasi tanpa perlu melakukan proses installasi terlebih dahulu.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat dibuat rumusan masalah sebagai berikut :

a. Bagaimana mengembangkan sebuah sistem berbasis web yang mampu mengoreksi LJK dan menganalisis hasilnya ?
b. Bagaimana menerapkan metode *grayscaling-thresholding-cropping* untuk memperoleh jawaban yang terdapat dalam LJK ?
c. Bagaimana menganalisis hasil ujian menggunakan analisis taraf kesukaran soal berdasarkan nilai yang diperoleh seluruh peserta ujian ?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah menghasilkan sebuah sistem berbasis web yang dapat digunakan untuk mengoreksi LJK dan menganalisis hasil ujian yang diselenggarakan oleh SMPIT Al-Ghozali Jember.

1.4 Manfaat

Manfaat yang dapat diperoleh dari penelitian ini antara lain :

a. Bagi Peneliti
   1. Menambah pemahaman peneliti terkait penerapan metode MVC pada proses implementasi sistem koreksi LJK dan analisis hasil ujian.
   2. Menambah perbendaharaan kajian ilmiah seputar pengembangan sistem koreksi LJK.
b. Bagi Instansi SMPIT Al-Ghozali Jember
   1. Dapat membantu proses pengoreksian LJK dan analisis hasil ujian yang telah dilaksanakan.
   2. Dapat membantu merumuskan kebijakan yang berkaitan dengan proses pembelajaran berdasarkan informasi yang disajikan oleh perangkat lunak yang dihasilkan dari penelitian ini.

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

1.5 Batasan Masalah

Berdasarkan permasalahan yang diuraikan pada latar belakang, maka diperlukan batasan-batasan masalah agar penelitian yang dilakukan lebih terarah dan mudah dipahami. Batasan masalah dalam penelitian ini antara lain :

a. Sistem dibangun dalam *platform* web menggunakan pola *model-view-controller* (MVC).
b. Pengujian terhadap perangkat lunak dilakukan dengan metode pengujian *black-box* dan *white-box.*
c. Data LJK yang akan digunakan dalam penelitian ini diperoleh dari hasil penyelenggaraan *try out* SMPIT Al-Ghozali Jember.
d. Mesin *scanner* yang digunakan adalah Canon MX390.

1.6 Sistematika Pembahasan

Dalam penyusunan penelitian ini, sistematika pembahasan terdiri dari 8 bab. Uraian dari masing-masing bab tersebut antara lain :

BAB I Pendahuluan

Menjelaskan tentang latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan.

BAB II Landasan Kepustakaan

Menjabaran teori-teori yang akan digunakan dalam penelitian ini.

BAB III Metodologi Penelitian

Berisi pemaparan tentang metodologi penelitian yang akan digunakan dalam penyusunan penelitian ini.

BAB IV Analisis Kebutuhan

Spesifikasi kebutuhan perangkat lunak yang disesuaikan dengan kepentingan *stakeholder* akan dibahas pada bab ini.

BAB V Perancangan

Menguraikan perancangan perangkat lunak yang disesuaikan dengan poin-poin yang telah dirumuskan dalam analisis kebutuhan.

BAB VI Implementasi

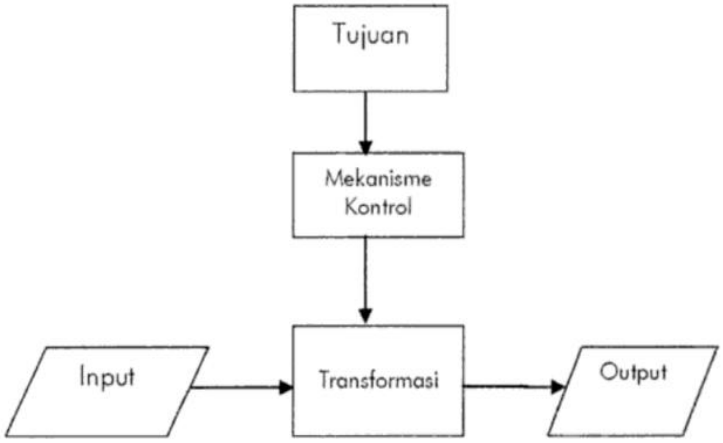
Hasil implementasi perangkat lunak akan dipaparkan dalam bab ini.

BAB VII Pengujian

Mendeskripsikan pengujian perangkat lunak menggunakan metode pengujian *black-box* dan *white-box*.

BAB VIII Penutup

Memberikan kesimpulan serta saran berdasarkan hasil penelitian.



Gambar 2.1 Model sistem menurut Mc Leod

Sumber : Mc Leod (1995, dalam Al Fatta, 2007)

2.2.2 Karakteristik Sistem

Karakteristik sistem yang dapat membedakan sebuah sistem dengan sistem yang lain, antara lain :

a. Batasan (*boundary*)

Batasan (*boundary*) merupakan penggambaran elemen atau unsur apa saja yang termasuk di dalam sistem dan elemen atau unsur apa yang termasuk di luar sistem.

b. Linkungan (*environment*)

Lingkungan (*environment*) adalah segala sesuatu di luar sistem. Lingkungan menyediakan asumsi, kendala dan input terhadap suatu sistem.

c. Masukan (*input*)

Setiap sumber daya (data, energi, peralatan, *dll*) yang berasal dari lingkungan (*environment*) kemudian dikonsumsi dan dimanipulasi oleh sistem disebut dengan masukan (*input*).

d. Keluaran (*output*)

Produk (informasi, laporan, dokumen, *dll*) yang disediakan untuk lingkungan setelah melalui proses-proses tertentu di dalam sistem ialah definisi dari keluaran (*output*).

e. Komponen (*component*)

Komponen (*component*) dapat didefinisikan sebagai kegiatan-kegiatan dalam sebuah sistem untuk mengubah masukan (*input*) menjadi keluaran (*output*). Komponen dapat berupa sub-sistem dari sebuah sistem.

f. Penghubung (*interface*)

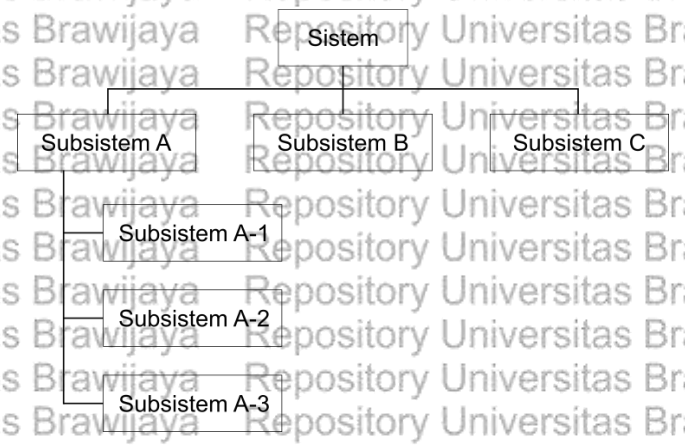
Tempat terjadinya interaksi antara komponen (*component*) atau sistem dengan lingkungannya (*environment*) adalah pengertian dari penghubung (*interface*).

g. Penyimpanan (*storage*)

Penyimpanan (storage) merupakan media yang digunakan untuk menyimpan energi, informasi, bahan baku, *dsb*.

2.2.3 Definisi Subsistem

Sebuah sistem yang kompleks tersusun dari beberapa subsistem. Subsistem dapat diartikan sebagai suatu sistem yang berada di dalam sebuah sistem yang lebih besar. Contoh, *automobile* adalah sebuah sistem yang dibentuk dari beberapa subsistem seperti subsistem suspensi, subsistem mesin, subsistem kelistrikan, subsistem keamanan, *dll*. Subsistem mesin juga dibangun di atas beberapa subsistem seperti subsistem karburator, subsistem generator, subsistem bahan bakar, *dst*. Ilustrasi subsistem dapat dilihat pada Gambar 2.2.



Gambar 2.2 Gambaran subsistem dalam sistem

2.3 Analisis Butir Soal

Proses belajar mengajar belum dapat dikatakan berhasil atau gagal sebelum pengajar melaksanakan evaluasi hasil belajar terhadap seluruh peserta didiknya (Sridadi, 2002). Pengajar seharusnya tidak berhenti pada tahap mengetahui hasil evaluasi belajar, melainkan harus melanjutkan sampai menyelesaikan tahap analisis hasil tersebut. Mengutip pendapat Thorndike dan Hagen (1997), Sridadi (2002) menerangkan jika hasil evaluasi belajar dapat dianalisis dengan baik, maka pengajar akan mampu mengetahui pada titik mana saja peserta didik gagal memahami materi yang diajarkan sehingga pengajar dapat menitikberatkan pembahasan pada materi-materi tertentu.

2.3.1 Taraf Kesukaran Soal

Menurut Purwanto (1994 dalam Sridadi, 2002) agar dapat mengetahui tingkat kesukaran setiap butir soal, langkah awal yang harus dilakukan adalah mengelompokkan peserta ujian ke dalam 3 kelompok sesuai dengan hasil ujiannya. Ketiga kelompok tersebut, antara lain :

a. *Upper Group*

Upper group diambil dari 25% - 27% dari jumlah peserta didik dengan nilai tertinggi. Kelompok ini disebut juga sebagai kelompok siswa pandai.

b. *Lower Group*

Lower group adalah 25% - 27 % dari jumlah peserta didik dengan nilai terendah. Kelompok ini dapat disebut dengan kelompok siswa kurang pandai.

c. *Middle group*

Middle group adalah kelompok peserta didik yang tidak masuk ke dalam *upper group* maupun *lower group*.

Kelompok yang akan dilibatkan dalam analisis butir soal adalah *upper group* dan *lower group*. Agar taraf kesukaran tiap butir soal dapat diketahui, maka diterapkanlah persamaan seperti yang tertera pada Persamaan 2.1.

$$TK = \frac{U + L}{T} \tag{2.1}$$

Berdasarkan Persamaan 2.1, maka indeks taraf kesukaran (TK) dapat ditemukan dengan cara menghitung jumlah siswa dari *upper group* yang menjawab butir soal dengan benar (U) kemudian menambahkannya dengan jumlah siswa dari *lower group* yang dapat menjawab butir soal dengan benar (L) lalu membagi hasilnya dengan total siswa yang termasuk dalam *upper group* dan *lower group* (T).

Besaran indeks taraf kesukaran berkisar antara 0,00 sampai 1,00. Jika indeks taraf kesukaran soal tidak lebih besar dari 0,30, maka soal tersebut termasuk dalam kategori sukar. Apabila indeks taraf kesukaran soal lebih besar dari 0,30 namun tidak lebih besar dari 0,70, maka soal tersebut termasuk dalam kategori sedang. Soal akan dikategorikan mudah apabila indeks taraf kesukarannya melebihi angka 0,70 (Direktorat Peminaan SMA, 2010).

2.4 *Software Process*

Software process adalah serangkaian kegiatan untuk memproduksi sebuah perangkat lunak dimana ada keterkaitan antara satu kegiatan dengan kegiatan yang lain (Somerville, 2011). Ada 4 aktifitas utama yang harus ada dalam *software process*, antara lain :

a. *Software specification*

Fungsionalitas perangkat lunak serta batasan-batasan operasinya harus didefinisikan dengan jelas.

b. *Software design & implementation*

Untuk memenuhi spesifikasi yang telah didefinisikan, maka perangkat lunak harus dibangun.

c. *Software validation*

Perangkat lunak harus menyelesaikan tahap validasi agar dapat diketahui apakah perangkat lunak yuang telah dikembangkan telah sesuai dengan apa yang diinginkan pelanggan.

d. *Software evolution*

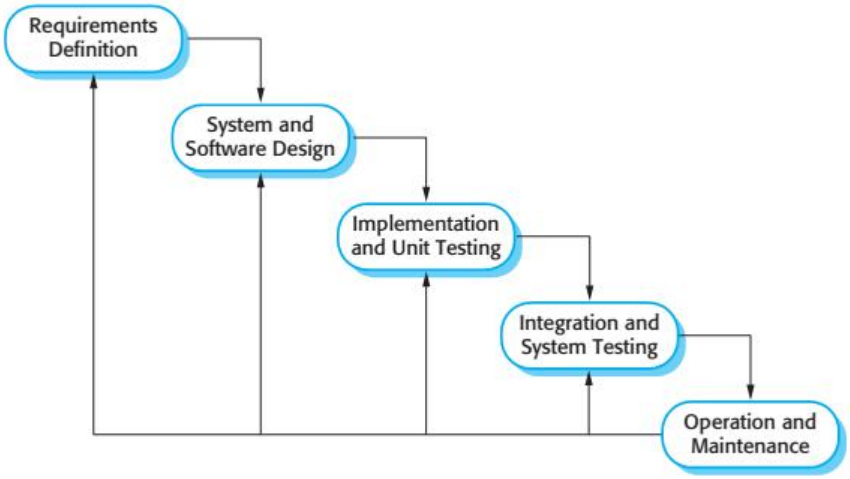
Perangkat lunak harus mampu berevolusi agar dapat memenuhi keinginan pelanggan yang senantiasa berubah.

Software process memiliki berbagai macam jenis yang berbeda, namun tidak dapat disebutkan jenis apa yang terbaik. Setiap organisasi telah mengembangkan dan memutakhirkan software process agar dapat memaksimalkan potensi yang dimiliki anggotanya serta agar dapat menyesuaikan dengan karakteristik perangkat lunak yang sedang dikembangkan oleh organisasi tersebut.

2.4.1 *The Waterfall Model*

Software process model adalah representasi *software process* yang telah disederhanakan. Setiap process model menggambarkan sebuah proses dari sudut pandang tertentu dan hanya akan memberikan potongan-potongan informasi sesuai dengan proses yang sedang dijalankan (Somerville, 2011).

Waterfall model merupakan salah satu representasi dari *software process model*. *Waterfall model* termasuk sebagai salah satu representasi *software process model* karena telah memiliki 4 aktivitas utama dalam *software process* seperti *specification, development, validation & evolution*. Keempat aktivitas tersebut dituangkan dalam beberapa fase yang berbeda seperti *requirements specification, software design, implementation, testing*, dsb.



Gambar 2.3 *The waterfall model*

Sumber : Somerville (2011)

Waterfall model bukan sebuah deskripsi utuh dari *software process*, melainkan sebuah pendekatan yang digunakan untuk menggambarkan proses pengembangan perangkat lunak.

Tahapan-tahapan pengembangan perangkat lunak yang terdapat dalam *waterfall model* antara lain :

a. *Requirements analysis & definition*

Fungsionalitas, batasan serta tujuan dari sistem ditetapkan dengan jalan berkonsultasi dengan pengguna sistem. Kemudian hal-hal tersebut didefinisikan dengan rinci agar siap disajikan sebagai spesifikasi sistem.

b. *System & software design*

Perancangan perangkat lunak melibatkan proses identifikasi dan penggambaran abtraksi perangkat lunak serta relasi-relasi yang ada di dalamnya.

c. *Implementation & unit testing*

Dalam tahap ini, desain dari perangkat lunak yang telah dibuat mulai direalisasikan menjadi beberapa unit program yang terpisah. *Unit testing* dilakukan rerhadap setiap unit program yang telah selesai dibuat, agar masing-masing unit program sesuai dengan peruntukannya.

d. *Integration & system testing*

Setiap unit program yang ada disatukan menjadi sebuah program yang utuh. Setelah itu, dilaksanakanlah *system testing* agar dapat diketahui bahwa *software requirement* telah terpenuhi.

e. *Operation & maintenance*

Walaupun tidak terlalu dibutuhkan, umumnya tahap ini adalah tahaan terpanjang dalam proses pengembangan perangkat lunak. Proses *maintenance* meliputi perbaikan terhadap kesalahan-kesalahan yang tidak terdeteksi pada proses sebelumnya, peningkatan kualitas layanan sistem juga diperkenankan dalam tahapan ini.

Setiap tahapan yang terdapat dalam *waterfall model* menghasilkan dokumentasi. Hal ini membuat proses yang berjalan terlihat sangat transparan, sehingga manajer proyek dapat memantau kemajuan yang terjadi dalam proses pengembangan perangkat lunak.

2.5 *Unified Modeling Language* (UML)

UML adalah sebuah bahasa yang menjadi standar dalam merancang dan mendokumentasikan sebuah sistem atau perangkat lunak (Khasanah, 2015). UML menyediakan 9 (sembilan) standar diagram yang dapat digunakan untuk menggambarkan sebuah sistem dari berbagai sudut pandang. Diagram tersebut antara lain : diagram kelas, diagram objek, *use case diagram*, *sequence diagram*, *collaboration diagram*, *state chart diagram*, activity diagram, *component diagram*, *deployment diagram*.

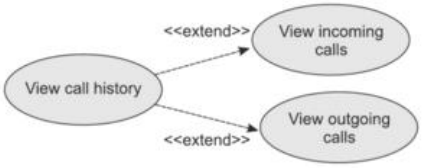
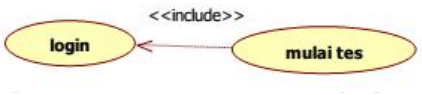
2.5.1 *Use Case Diagram*

Rumbaugh, Jacobson dan Booch (2005) mendefinisikan *use case diagram* sebagai sebuah diagram yang menggambarkan hubungan antara aktor dan *use case* dalam sebuah sistem. *Use case* diagram memiliki notasi-notasi yang mewakili fungsi-fungsi tertentu seperti yang digambarkan pada Tabel 2.1.

Tabel 2.1 Simbol-simbol dalam *use case diagram*

Simbol	Nama	Deskripsi
	Aktor	Aktor merepresentasikan peran-peran yang dapat diambil oleh user untuk dapat berinteraksi dengan sistem. Aktor tidak selalu berwujud manusia, aktor juga dapat berbentuk sistem lain.
	Use case	*Use case* merupakan gambaran dari aksi-aksi yang dapat dilakukan oleh aktor
	Relationship	Relationship menggambarkan hubungan khusus antar elemen yang memiliki makna tertentu

Tabel 2.1 (Lanjutan)

Simbol	Nama	Deskripsi
<<extend>>	*Extend Relationship*	*Extend relationship* merupakan relasi yang menggambarkan hubungan subuah use case dengan *use case* tambahan lain yang bersifat *optional*. Contoh :  *Use case "view incoming calls"* dan *"view outgoing calls"* adalah *optional use case* dari *"view call history"*.
<<include>>	*Include Relationship*	*Include relationship* merupakan relasi yang mengharuskan sebuah *use case* dijalankan terlebih dahulu agar dapat menjalankan *use case* lain. Contoh :  Use case "mulai tes" hanya dapat dijalankan setelah use case "login" telah dijalankan.

Sumber : Diadaptasi dari Rumbaugh, Jacobson & Booch (2005) dan Rosa (2013, dalam Khasanah, 2015)

2.5.2 *Class Diagram*

Menurut Rosa (2011, dalam Fadhihansah, 2014) *class diagram* adalah diagram yang menggambarkan struktur sebuah sistem berdasarkan pendefinisian kelas-kelas yang digunakan untuk membangun sistem tersebut. Kelas memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang menyusun kelas tersebut, sementara metode adalah fungsi-fungsi yang dapat dijalankan dalam kelas. Notasi-notasi yang terdapat dalam *class diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol-simbol dalam *class diagram*

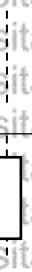
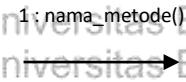
Simbol	Nama	Deskripsi
NamaKelas +atribut +operasi()	Kelas	Himpunan dari objek yang saling berbagi atribut dan metode yang sama.
nama_interface	*Interface*	Memiliki konsep yang sama dengan *interface* dalam pemrograman berorientasi objek.
	Association	Menggambarkan relasi antar kelas dengan makna yang umum.
	Directed Association	Menggambarkan relasi antar kelas dengan makna "kelas yang satu digunakan oleh kelas yang lain".
	Generalization	Menggambarkan relasi antar kelas dengan makna "generalisasi-spesialisasi (umum-khusus)".
	Dependency	Menggambarkan relasi antar kelas dengan makna "kebergantungan sebuah kelas dengan kelas yang lain".

Sumber : Diadaptasi dari Rosa (2011, dalam Fadhihansah, 2014)

2.5.3 *Sequence Diagram*

Sequence diagram adalah diagram yang menggambarkan perilaku objek dengan mendeskripsikan masa hidup objek tersebut serta transaksi *message* antar objek. Sebelum menggambar *sequence diagram*, objek yang terlibat serta metode-metode yang dimiliki *class* yang direpresentasikan menjadi objek tersebut harus diketahui terlebih dahulu (Rosa, 2011 dalam Fadhihansah, 2014). Notasi-notasi yang bisa digunakan dalam proses penyusunan *sequence diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol dalam *sequence diagram*

Simbol	Nama	Deskripsi
	Aktor	Himpunan dari objek yang saling berbagi atribut dan metode yang sama.
	*Lifeline*	Menyatakan rentan waktu dimana objek dapat hidup di dalamnya.
	Waktu aktif	Menyatakan jika sebuah objek dalam kondisi aktif dan tengah melakukan transaksi.
	*Call*	Menggambarkan jika sebuah objek memanggil operasi atau metode tertentu, baik dari objek lain atau objek itu sendiri.
	return	Menggambarkan sebuah *return* dari suatu objek setelah menjalankan operasi tertentu.

Sumber : Diadaptasi dari Rosa (2011, dalam Fadhihansah, 2014)

2.6 Pengujian Perangkat Lunak

Tujuan dari pengujian perangkat lunak adalah menemukan *error* yang terdapat dalam perangkat lunak tersebut. Teknik pengujian yang baik adalah sebuah teknik yang mampu memberikan tingkat probabilitas yang tinggi untuk menemukan sebuah *error* (Pressman, 2010).

Setiap produk yang dihasilkan dari proses rekayasa dapat diuji menggunakan salah satu dari dua metode pengujian, yaitu :

1. Pengujian yang dilakukan untuk mengetahui apakah suatu fungsi tertentu telah dibangun sesuai dengan peruntukannya. Pengujian ini mengambil sudut pandang eksternal. *Black-box testing* merupakan representasi dari pengujian ini.
2. Pengujian yang berfungsi untuk memastikan bahwa setiap komponen penyusun produk sudah berjalan dengan baik dan saling mendukung antara satu sama lain. Pengujian ini dilaksanakan menggunakan sudut pandang internal dan dikenal dengan istilah *white-box testing*.

2.6.1 *Black-box Testing*

Black-box testing adalah teknik pengujian perangkat lunak yang dapat diterapkan tanpa harus mengetahui *source code* dari perangkat lunak yang akan diuji. Teknik pengujian ini dapat dilakukan pada sebuah

perangkat lunak secara keseluruhan atau salah satu unit (fungsi) yang terdapat dalam perangkat lunak tersebut. Jika pengujian dilakukan terhadap sebuah perangkat lunak secara keseluruhan, maka perangkat lunak tersebut dianggap sebagai sebuah *black-box.* Sebuah unit yang ada dalam perangkat lunak juga dapat disebut sebagai *black-box* apabila pengujian dilakukan pada level unit.

Pressman (2010) berpendapat bahwa *black-box testing* mempunyai kemampuan yang tinggi untuk menemukan *error* pada beberapa kategori, seperti :

a) Fungsi yang hilang atau salah
b) Kesalahan pada *interface*
c) Kesalahan pada struktur data atau akses *database*
d) Kesalahan kinerja
e) Kesalahan inisialisasi dan terminasi

Black-box testing cenderung dilaksanakan pada tahap akhir pengeujian sebuah perangkat lunak, berbeda dengan *white-box testing* yang dilaksanakan pada tahap awal pengujian sebuah perangkat lunak.

2.6.1.1 Validation Testing

Validation testing dilaksanakan ketika sebuah perangkat lunak telah tersusun sempurna menjadi satu kesatuan yang utuh. Kegiatan pengujian berfokus pada aksi-aksi yang dapat dilakukan oleh *user* serta *output* yang dapat dikenali oleh *user*.

Validation testing mempunyai definisi yang sangat luas. Sama seperti strategi pengujian lainnya, *validation testing* juga berusaha menemukan *error* pada perangkat lunak. *Validation testing* lebih menitik beratkan fokus pengujian pada kesesuaian perangkat lunak yang telah dibangun dengan kebutuhan dari *user*.

2.6.2 *White-box Testing*

White-box testing merupakan teknik pengujian perangkat lunak yang dapat diterapkan jika penguji telah mengetahui *source code* perangkat lunak yang akan diuji. Umumnya pengujian ini dilakukan oleh penulis *source code* atau tim penguji dari pengembang sebuah perangkat lunak.

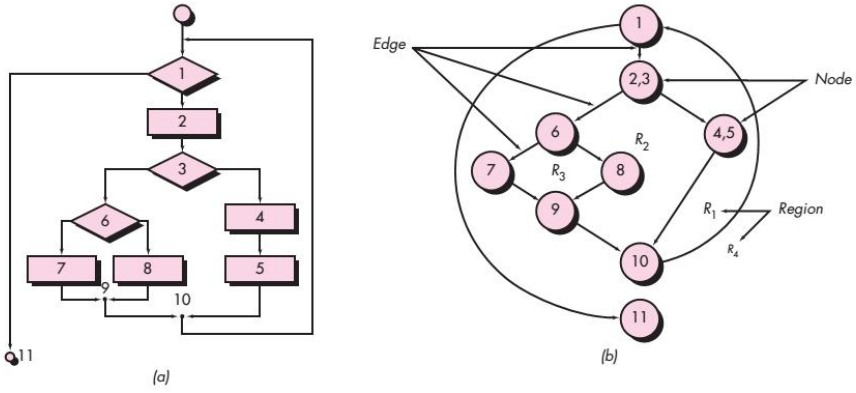
2.6.2.1 Basis Path Testing

Basis path testing adalah teknik pengujian yang dikenalkan oleh Tom McCabe pada tahun 1980-an. Teknik pengujian ini digunakan untuk menganalisis *flow graph* suatu program. Dari *flow graph* tersebut dapat dihitung nilai *cyclomatic complexity* (V(G)) sebuah program. Setelah mendapat nilai V(G) maka *software tester* dapat menentukan

independent path yang selanjutnya akan digunakan untuk membuat *test case*.

a) *Flow Graph*

Flow graph adalah notasi sederhana yang berfungsi sebagai representasi *control flow* dari sebuah program.



Gambar 2.4 *Flowchart* (a) dan *flow graph* (b)

Sumber : Pressman (2010)

Gambar 2.4a merupakan *flowchart* yang menggambarkan aliran prosedur yang terdapat dalam sebuah sistem tertentu. Gambar 2.4b adalah representasi *flow graph* dari Gambar 2.4a. Mengacu pada Gambar 2.4b, setiap lingkaran yang terdapat di dalamnya dapat disebut dengan *node*. Setiap *node* dapat mewakili satu atau beberapa *procedural statement*. Sedangkan setiap *node* yang mengandung *condition* disebut dengan *predicate node*. Sementara tanda panah yang menghubungkan satu *node* dengan *node* yang lain disebut dengan *edge* atau *link*. *Edge* menggambarkan arah aliran yang disesuaikan dengan arah aliran yang terdapat pada *flowchart*. Wilayah yang dibatasi oleh *edge* dan *node* disebut dengan *region*. Wilayah yang terdapat di luar *flow graph* dihitung sebagai satu *region*.

b) *Cyclomatic Complexity*

Dalam *basis path testing*, *cyclomatic complexity* dapat digunakan untuk menghitung jumlah *independent path* yang dimiliki sebuah program. *Cyclomatic complexity* (V(G)) dapat dihitung dengan cara :

1) V(G) = jumlah *region* dalam *flow graph*
2) V(G) = E - N + 2
   :: E = jumlah *edge* dalam *flow graph*
   :: N = jumlah *node* dalam *flow graph*

3) V(G) = P + 1

:: P = jumlah *predicate note* dalam *flow graph*

Nilai V(G) dari *flow graph* yang terdapat pada Gambar 2.4b adalah :

1) *Flow graph* mempunyai 4 *region*
2) V(G) = 11 *edge* - 9 *node* + 2
   = 4
3) V(G) = 3 *predicate node* + 1
   = 4

c) *Independent Path*

Independent path adalah setiap jalur dalam perangkat lunak yang menghubungkan antara *node* awal dengan *node* akhir dan minimal melewati satu edge baru dengan alur yang belum pernah dilalui sebelumnya (Khasanah, 2015). *Independent path* yang dapat diperoleh dari *flow graph* yang terdapat pada Gambar 2.4b, antara lain :

Path 1 : 1-11

Path 2 : 1-2-3-4-5-10-1-11

Path 3 : 1-2-3-6-8-9-10-1-11

Path 4 : 1-2-3-6-7-9-10-1-11

Setelah *independent path* diketahui, maka i*ndependent path* akan diimplementasikan dalam bentuk *test case.*

d) *Test case*

Khasanah (2015) menjelaskan jika *test case* dibuat untuk menguji apakah alur logika yang telah dibuat sesuai dengan *independent path*.

2.7 Pengolahan Citra

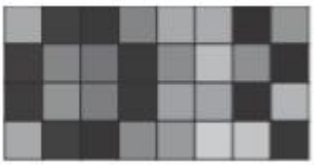
Septian (2009, dalam Fata 2014) menyatakan bahwa pengolahan citra merupakan sebuah kegiatan untuk melakukan manipulasi pada kualitas citra agar lebih mudah untuk ditafsirkan, baik oleh manusia maupun mesin. Beberapa elemen dasar dari citra yang dapat dimanipulasi antara lain : warna, kecerahan (*brightness*), kontras, *dsb*. Salah satu teknik pengolahan citra adalah *grayscaling*, *thresholding* dan *cropping*.

2.7.1 *Grayscaling*

Menurut Achmad dan Firdausy (2013, dalam Fata 2014) *grayscaling* merupakan kegiatan untuk memetakan tingkat keabuan suatu *pixel* (satuan terkecil penyusun citra digital). Skala keabuan dapat diketahui dengan menggunakan persamaan seperti yang terdapat pada Persamaan 2.2.

$$K_o = \frac{R_i + G_i + B_i}{3} \quad (2.2)$$

Nilai keabuan (K_o) sebuah *pixel* dapat dicari dengan menghitung nilai rata-rata dari warna merah (R_i), warna hijau (G_i) dan warna biru (B_i) yang menyusun *pixel* tersebut. Contoh citra yang telah diubah menjadi keabuan dapat dilihat pada Gambar 2.5.

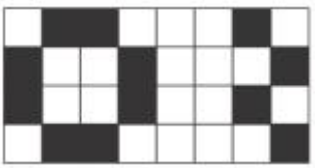


Gambar 2.5 Citra yang telah mengalami proses *grayscaling*

Setelah menjalani proses *grayscaling*, hanya akan ada warna putih, hitam dan warna abu-abu pada citra.

2.7.2 *Thresholding*

Operasi *thresholding* dapat digunakan untuk mengubah citra keabuan menjadi citra *monochrome* yang hanya terdiri dari dua nilai, yaitu hitam dan putih. Langkah awal operasi ini dimulai dengan menentukan ambang batas, setelah itu melakukan pengecekan terhadap masing-masing *pixel* penyusun citra. Jika nilai keabuan suatu *pixel* lebih rendah dari ambang batas yang ditentukan, maka *pixel* tersebut akan diubah menjadi warna putih. Begitu pula sebaliknya. Contoh citra yang telah diubah menjadi citra *monochrome* dapat dilihat pada Gambar 2.6.

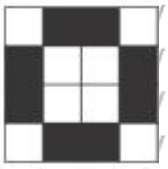


Gambar 2.6 Citra yang telah mengalami proses *thresholding*

Hanya ada dua warna yang akan dikenali pada citra *monochrome*, yaitu warna hitam dan putih.

2.7.3 *Cropping*

Cropping adalah proses untuk memotong dan mengambil sebagian dari sebuah citra sehingga diperoleh citra baru yang berdimensi lebih kecil. Contoh citra yang telah mengalami cropping dapat dilihat pada Gambar 2.8. Citra yang terdapat pada Gambar 2.7 adalah hasil cropping dari citra yang semula terdiri dari 32 pixel (dapat dilihat pada Gambar 2.6).



Gambar 2.7 Citra yang telah mengalami proses croping

2.8 PHP Hypertext Preprocessor (PHP)

Vaswani (2009) menjelaskan bahwa bahasa pemrograman PHP pertama kali dikembangkan oleh Rasmus Lerdorf pada tahun 1994. Versi PHP tersebut dikenal dengan nama Personal Home Page/Form Interpreted (PHP/FI). Setelah itu, Pada tahun 1997 Rasmus memperkenalkan PHP/FI 2.0 yang memiliki kemampuan jauh lebih baik dibanding versi PHP sebelumnya. Pengembangan bahasa pemrograman PHP diambil alih oleh Andi Gutmans and Zeev Suraski, kemudian kedua pengembang tersebut mengganti akronim PHP/FI menjadi PHP (PHP Hypertext Preprocessor) dan memperkenalkan PHP 3.0 pada tahun 1997 pula.

PHP 4.0 adalah generasi bahasa pemrograman PHP yang mulai mendukung konsep pemrograman berbasis objek. Versi PHP ini mulai diperkenalkan tahun 1999. Saat ini, versi PHP paling mutakhir adalah PHP 7.0. PHP 7.0 mulai diperkenalkan secara resmi pada pertengahan 2016.

PHP memiliki beberapa kelebihan yang tidak dimiliki bahasa pemrograman berbasis web pada umumnya (Vaswani, 2009). Kelebihan-kelebihan tersebut antara lain :

1. *Performance*

   Proses pengeksekusian *script* yang ditulis menggunakan bahasa pemrograman PHP dapat berlangsung lebih cepat dibanding *script* yang ditulis menggunakan bahasa pemrograman berbasis web lainnya seperti JSP, ASP.NET maupun PEARL. Teknologi yang dikembangkan pada tubuh PHP didesain sedemikian rupa sehingga mampu memaksimalkan penggunaan memori dan berdampak pada meningkatnya kemampuan untuk pengeksekusian *script*.

2. *Open Source*

   PHP adalah sebuah proyek *open source* yang dikembangkan oleh begitu banyak relawan dari berbagai penjuru dunia, sehingga para pengembang tidak perlu mengeluarkan biaya untuk dapat menggunakan PHP.

3. *Ease of Use*

Syntax-syntax yang digunakan dalam PHP sangat jelas dan mudah dipahami. PHP juga menyediakan dokumentasi yang lengkap untuk lebih dari 5000 fungsi yang tertanam di dalamnya, sehingga para pengembang yang baru menggunakan bahasa PHP dapat menghemat jam belajarnya. Faktor inilah yang membuat PHP menjadi salah satu bahasa pemrograman yang paling populer di dunia pemrograman web.

4. *Third-party Application Support*

PHP mampu mendukung lebih dari 15 jenis *database engine* yang berbeda. Selain itu PHP juga dapat membaca dan membuat *file* dalam format gambar seperti GIF, PNG, JPG, *dll*. PHP juga memberikan ijin kepada para pengembang untuk menggunakan dan membuat *library* baru sesuai kebutuhan mereka.

2.8.1 Penggunaan Variabel dalam PHP

Variabel adalah sebuah wadah yang digunakan untuk menampung nilai-nilai tertentu, dapat berupa angka ataupun yang selainnya. PHP memiliki aturan khusus untuk mendefinisikan nama sebuah variabel. Pendefinisian nama sebuah variabel harus diawali dengan simbol dolar ($). Contoh pendefinisian variabel pada PHP dapat dilihat pada Tabel 2.4.

Tabel 2.4 Pendefinisian variabel dalam PHP

No	*Syntax*
1.	`<?php`
2.	`  $nama = "Budi";`
3.	`  $umur = 12;`
4.	`?>`

Dari *syntax-syntax* yang terdapat pada Tabel 2.4, dapat disimpulkan bahwa variabel $nama yang terdapat pada *syntax* nomor 2 menyimpan nilai "Budi" dan angka 12 adalah nilai yang tersimpan dalam variabel $umur.

2.8.2 Penggunaan Fungsi dalam PHP

Fungsi merupakan salah satu jalan keluar yang dapat digunakan untuk mengatasi pemakaian *syntax* secara berulang-ulang. Pendefinisian fungsi cukup dilakukan sekali saja, namun penggunaannya dapat dilakukan barulang-ulang sesuai dengan kebutuhan. Proses pengujian program juga dapat berlangsung lebih mudah apabila program dibagi ke dalam beberapa fungsi. Contoh pendefinisian fungsi dapat dilihat pada Tabel 2.5

Tabel 2.5 Pendefinisian fungsi dalam PHP

No	*Syntax*
1.	<?php
2.	function ucapkanSalam(){
3.	echo "Hello World :D";
4.	}
5.	
6.	ucapkanSalam();
7.	?>

Dari Tabel 2.5 dapat diketahui bahwa *syntax* nomor 2 digunakan untuk mendefinisikan nama fungsi, sementara *syntax* nomor 3 adalah prosedur yang harus dikerjakan oleh program ketika fungsi telah dipanggil dan *syntax* nomor 4 adalah batas akhir dari pendefinisian fungsi ucapkanSalam(). Fungsi ucapkanSalam() dapat dipanggil menggunakan *syntax* seperti yang terdapat pada baris kode nomor 7.

Selain dapat membuat fungsi baru sesuai dengan kebutuhan, PHP juga menyediakan *built-in function* yang sewaktu-waktu dapat langsung diakses oleh para pengembang tanpa perlu mendefinisikan fungsi terlebih dahulu. *Built-in function* yang disediakan sangat beranekaragam, mulai dari fungsi untuk mendapatkan informasi tentang tanggal, jenis tipe data sebuah variabel sampai dengan fungsi untuk pengolahan citra.

2.8.2.1 Fungsi untuk *Grayscaling*

PHP menyediakan fungsi dengan nama imagefilter() yang dapat digunakan untuk memanipulasi sebuah citra digital. Butuh dua parameter untuk memanggil fungsi ini. Parameter pertama diisi dengan citra yang akan dimanipulasi dan parameter kedua adalah jenis manipulasi yang ingin diterapkan. Contoh penggunaan fungsi imagefilter() dapat dilihat pada Tabel 2.6.

Tabel 2.6 Penggunaan fungsi imagefilter()

No	*Syntax*
1.	<?php
2.	$im = imagecreatefromjpeg("gambar.jpg");
3.	imagefilter($im, IMG_FILTER_GRAYSCALE);
4.	imagedestroy($im);
5.	?>

Jenis pemanipulasian citra yang harus dimasukkan pada fungsi imagefilter() agar dapat mengubah citra menjadi keabuan adalah IMG_FILTER_GRAYSCALE, seperti yang ditunjukkan pada syntax nomor 3.

2.8.2.2 Fungsi untuk *Thresholding*

Salah satu fungsi dalam PHP yang dapat digunakan untuk memanipulasi citra menjadi citra dengan warna *monochrome* (hitam dan putih) adalah jpeg2wbmp(). Parameter yang dibutuhkan fungsi tersebut agar dapat bekerja adalah nama dari citra yang ingin dimanipulasi, nama yang diinginkan dari citra yang telah dimanipulasi, dimensi tinggi citra, dimensi lebar, ambang batas *threshold* yang diinginkan. Contoh pemanggilan fungsi jpeg2wbmp() dapat dilihat pada Tabel 2.7.

Tabel 2.7 Penggunaan fungsi jpeg2wbmp()

No	*Syntax*
1.	`<?php`
2.	`$im = "gambar_old.jpg";`
3.	`$im_dim = getimagesize($im);`
4.	`imagefilter($im, IMG_FILTER_GRAYSCALE);`
5.	`jpeg2wbmp($im, 'gambar.wbmp', $im_dim[1], $im_dim[0], 5);`
6.	`?>`

2.8.2.3 Fungsi untuk *Cropping*

Fungsi imagecrop() adalah *built-in function* dalam PHP yang dapat dapat digunakan untuk memotong dan mengambil sebagian dari citra yang diinginkan. Dibutuhkan 2 parameter untuk memanggil fungsi tersebut. Parameter-parameter tersebut adalah nama dari citra yang akan dipotong dan sebuah array yang berisi koordinat dan besaran dimensi yang akan dipotong. Contoh pemanggilan fungsi imagecrop() dapat dilihat pada Tabel 2.8.

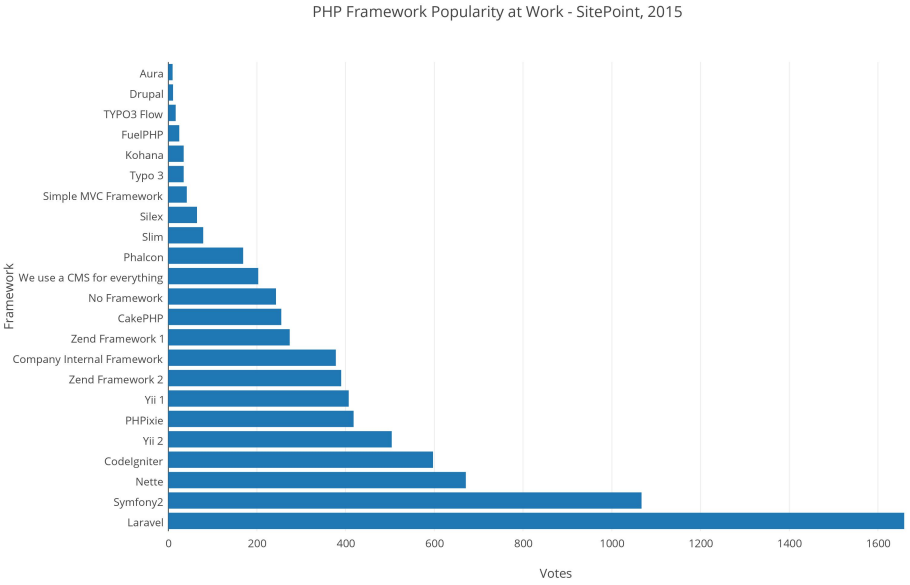
Tabel 2.8 Penggunaan fungsi imagecrop()

No	*Syntax*
1.	`<?php`
2.	`$im = imagecreatefromjpeg("gambar.jpg");`
3.	`$to_crop_array = [ 'x' => 5, 'y' => 5, 'width' => 25, 'height' => 25 ];`
4.	`imagecrop($im, $to_crop_array);`
5.	`?>`

2.9 Laravel Framework

Laravel adalah sebuah *framework* yang dibangun oleh Taylor Otwell sejak tahun 2012 dengan memanfaatkan bahasa pemrograman PHP (*Hypertext Preprocessor*).

Berdasarkan survey yang dilakukan oleh SitePoint (sitepoint.com) pada tahun 2015, Laravel menempati urutan pertama sebagai framework php paling populer di dunia. Hasil survey tersebut dapat dilihat pada Gambar 2.8.



Gambar 2.8 Tingkat kepopuleran *framework* PHP

Sumber : SitePoint.com

McCool (2012) menyebutkan beberapa kelebihan Laravel dibandingkan dengan framework PHP lainnya, antara lain:

a. Ekspresif

Laravel adalah sebuah *framework* yang ekspresif, artinya seorang *programmer* yang baru menggunakan framework tersebut langsung dapat mengetahui fungsi dari suatu *syntax* saat pertama kali melihatnya.

b. Sederhana

Salah satu faktor yang membuat Laravel menjadi *framework* yang sederhana adalah adanya *Eloquent*. *Eloquent* memungkinkan kita untuk membuat (*create*), memperbaharui (*update*) dan menghapus (*delete*) data dalam *database* tanpa harus menulis *query* secara utuh. Contoh :

```
$siswaKelasTiga = siswa::where('kelas','=','3')->get();
```

Syntax tersebut mempunyai arti jika kita akan mengambil seluruh data dari siswa kelas 3 yang tersimpan dalam tabel 'siswa'.

c. Dokumentasi sangat baik

Pengembang Laravel berkomitmen untuk menyediakan dokumentasi yang lengkap pada setiap perilisan Laravel versi terbaru.

3.1 Tempat dan Waktu Penelitian

Penelitian ini akan dilaksanakan di Sekolah Menengah Pertama Islam Terpadu (SMPIT) Al-Ghozali Jember selama bulan Maret 2016 sampai dengan Juni 2016.

3.2 Studi Literatur

Studi literatur dilaksanakan dengan tujuan mempelajari literatur-literatur dari bidang ilmu tertentu guna mendukung proses penelitian ini. Teori-teori yang dipelajari antara lain :

a. Perangkat lunak pengoreksi LJK
b. Sistem
c. *Software process*
d. *Unified modeling language*
e. Metode *grayscaling-thresholding-cropping*
f. Laravel *framework*
g. Pengujian perangkat lunak

Sumber referensi yang digunakan sebagai pendukung dan penunjang penelitian ini antara lain dari buku, *e-book*, jurnal penelitian, laporan skripsi.

3.3 Analisis Kebutuhan

Analisis kebutuhan merupakan salah satu tahap yang paling penting dalam pengembangan perangkat lunak. Kesalahan dalam tahap ini dapat berakibat fatal pada tahapan-tahapan selanjutnya. Spesifikasi kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang diperlukan untuk membangun sebuah sistem sesuai dengan keperluan *stakeholder*. Artefak yang dihasilkan pada tahap ini adalah *use case diagram*, dimana setiap *use case* yang tergambar didalamnya akan dijabarkan menggunakan *use case scenario*.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional terdiri berisi proses, fungsi dan fitur-fitur yang disesuaikan dengan kebutuhan *stakeholder*.

3.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional berisi kebutuhan-kebutuhan yang berada di luar kebutuhan fungsional. Kebutuhan non fungsional dapat meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.4 Perancangan Sistem

Setelah kebutuhan perangkat lunak telah selesai didefinisikan pada tahap analisis kebutuhan, maka dibuatlah rancangan perangkat lunak atau sistem yang akan dibangun dalam tahap ini.

Perancangan sistem dikerjakan menggunakan *unified modeling language* (UML). Diagram yang akan dipakai untuk memodelkan sistem ini, antara lain :

a. *Sequence diagram*
b. *Class diagram*

3.5 Implementasi

Dalam tahap ini, hasil dari analisa kebutuhan dan perancangan sistem akan diimplementasikan menggunakan bahasa pemrograman php dan dikembangkan menggunakan metode *model view controller* (MVC) dengan memanfaatkan *framework* Laravel.

3.6 Pengujian

Pada tahap ini akan dilakukan proses pengujian terhadap perangkat lunak. Pengujian terhadap perangkat lunak ini dilaksanakan menggunakan metode *black-box testing* dan *white-box testing*. Tahap pengujian ini bertujuan untuk mengetahui apakah perangkat lunak yang dikembangkan telah sesuai dengan kebutuhan yang didefinisikan di awal.

3.7 Kesimpulan

Penarikan kesimpulan diperlukan untuk mengetahui hasil atau pengetahuan baru apa yang dapat diperoleh dari penelitian ini. Kesimpulan juga harus mampu menjawab rumusan masalah yang telah dipaparkan di awal penelitian.

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

Sistem juga harus menyediakan fungsi untuk mencetak daftar hasil ujian, agar transparansi hasil ujian dapat diketahui oleh pihak-pihak terkait.

d. Menampilkan analisis hasil ujian

Sistem harus mampu menampilkan analisis hasil ujian. Analisis hasil ujian direpresentasikan dalam diagram dan grafik. Analisis hasil ujian terdiri dari gambaran kemampuan siswa dalam menjawab soal, tingkat kesukaran soal, sebaran nilai siswa dan statistik nilai siswa.

Sistem juga harus memiliki fitur untuk mengubah diagram atau grafik yang ditampilkan kedalam format gambar kemudian menyimpannya dalam *storage* pengguna, jika pengguna menginginkannya.

e. Mengelola pengguna sistem

Sistem hanya dapat diakses oleh pengguna-pengguna yang telah didaftarkan dalam sistem, oleh karena itu sistem harus menyediakan fitur untuk mengelola pengguna sistem.

Sistem harus mampu menyediakan fitur untuk menambahkan pengguna baru, mengubah data pengguna dan menghapus pengguna dari sistem. Pengguna yang memiliki otoritas untuk melaksanakan tugas-tugas tersebut adalah pengguna yang memiliki peran sebagai admin sistem.

Dari keterangan-keterangan di atas, maka dapat disusun kebutuhan-kebutuhan yang harus ada dalam sistem. Secara umum, kebutuhan yang harus ada pada sistem dapat dilihat pada Tabel 4.1.

Tabel 4.1 Kebutuhan sistem secara umum

No	Tipe Kebutuhan	Kebutuhan
1.	Fungsional	Sistem harus menyediakan fasilitas untuk mengelola kunci jawaban
2.	Fungsional	Sistem harus menyediakan fasilitas untuk mengelola jawaban peserta ujian
3.	Fungsional	Sistem harus menyediakan fasilitas untuk menampilkan hasil ujian
4.	Fungsional	Sistem harus menyediakan fitur untuk mencetak hasil ujian
5.	Fungsional	Sistem harus menyediakan fasilitas untuk menampilkan analisis hasil ujian
6.	Fungsional	Sistem harus menyediakan fitur untuk menyimpan analisis hasil ujian

Tabel 4.1 (Lanjutan)

No	Tipe Kebutuhan	Kebutuhan
7.	Fungsional	Sistem harus menyediakan fasilitas untuk mengelola pengguna sistem
8.	Fungsional	Sistem harus menyediakan fasilitas untuk melakukan login
9.	Fungsional	Sistem harus menyediakan fasilitas untuk melakukan logout
10.	Non Fungsional	*Password* pengguna diamankan menggunakan enkripsi *bcrypt*
11.	Non Fungsional	Sistem harus dapat dijalankan pada berbagai aplikasi *web browser*

4.2 Identifikasi Aktor

Pada tahan ini dilakukan identifikasi terhadap aktor-aktor yang dapat berinteraksi dengan sistem yang akan dibangun. Terdapat 4 aktor yang dapat berinteraksi dengan sistem, antara lain tamu, admin sistem, admin soal dan korektor. Peran tiap aktor akan dijabarkan dalam Tabel 4.2.

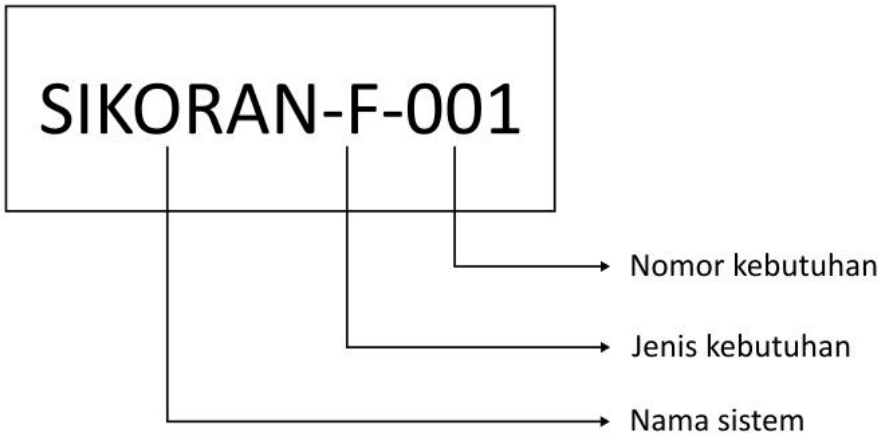
Tabel 4.2 Aktor yang dapat berinteraksi dengan sistem

No	Nama Aktor	Deskripsi
1.	Tamu	Tamu adalah aktor yang dapat melihat halaman awal sistem dan melakukan login
2.	Admin sistem	Admin sistem adalah aktor yang mempunyai wewenang untuk mendaftarkan dan mengelola data pengguna sistem serta data-data pendukung lain yang sangat dibutuhkan oleh sistem
3.	Admin soal	Admin soal adalah aktor yang ditugaskan untuk mengelola kunci jawaban, mencetak hasil ujian serta melihat analisis hasil ujian
4.	Korektor	Korektor adalah aktor yang bertugas untuk memasukkan dan mengelola jawaban peserta ujian

4.3 Spesifikasi Kebutuhan

4.3.1 Aturan Penomoran

Nomor kebutuhan perangkat lunak ditulis menggunakan format tertentu demi menjaga konsistensi penulisan. Contoh format penulisan yang digunakan dalam penelitian dapat dilihat pada Gambar 4.1.



Gambar 4.1 Format penomoran kebutuhan

Format penomoran kebutuhan yang digunakan dalam penelitian ini tersusun atas 3 bagian seperti yang tertera pada Gambar 4.1. Nama sistem yang digunakan dalam format penomoran adalah SIKORAN yang merupakan akronim dari sistem koreksi LJK dan analisis hasil ujian. Jenis kebutuhan dapat diisi dengan huruf F untuk mewakili kebutuhan fungsional atau huruf NF untuk mewakili kebutuhan non fungsional. Nomor kebutuhan diisi dengan nomor urut dan dimulai dari angka 1.

4.3.2 Kebutuhan Fungsional

Deskripsi kebutuhan fungsional mengacu pada fasilitas-fasilitas yang harus disediakan oleh sistem yang telah dijelaskan pada sub bab Penggalian dan Analisis Kebutuhan. Daftar kebutuhan fungsional akan disajikan dalam bentuk tabel dan dikelompokkan berdasarkan aktor.

A. Tamu

Daftar kebutuhan fungsional untuk aktor tamu dapat dilihat pada Tabel 4.3.

Tabel 4.3 Kebutuhan fungsional tamu

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIKORAN-F-001	Login	Tamu harus melakukan login agar dapat mengakses fitur-fitur yang telah disediakan oleh sistem

B. Admin Sistem

Daftar kebutuhan fungsional untuk aktor admin sistem dapat dilihat pada Tabel 4.4.

Tabel 4.4 Kebutuhan fungsional admin sistem

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIKORAN-F-002	Logout	Admin sistem dapat melakukan logout jika ia berkeinginan untuk keluar dari sistem
2.	SIKORAN-F-003	Lihat mata pelajaran	Admin sistem dapat melihat daftar mata pelajaran yang telah tersimpan dalam sistem
3.	SIKORAN-F-004	Tambah mata pelajaran	Admin sistem dapat menambahkan mata pelajaran baru ke dalam sistem
4.	SIKORAN-F-005	Ubah mata pelajaran	Admin sistem dapat mengubah data mata pelajaran yang tersimpan dalam sistem
5.	SIKORAN-F-006	Nonaktifkan mata pelajaran	Admin sistem mengubah status mata pelajaran menjadi nonaktif
6.	SIKORAN-F-007	Aktifkan mata pelajaran	Admin sistem dapat mengubah status mata pelajaran menjadi aktif
7.	SIKORAN-F-008	Lihat materi ujian	Admin sistem dapat melihat daftar materi ujian yang telah tersimpan dalam sistem
8.	SIKORAN-F-009	Tambah materi ujian	Admin sistem dapat menambahkan materi ujian baru ke dalam sistem
9.	SIKORAN-F-010	Ubah materi ujian	Admin sistem dapat mengubah data materi ujian yang tersimpan dalam sistem
10.	SIKORAN-F-011	Nonaktifkan materi ujian	Admin sistem dapat mengubah status materi ujian menjadi nonaktif
11.	SIKORAN-F-012	Aktifkan materi ujian	Admin sistem dapat mengubah status materi ujian menjadi aktif

Tabel 4.4 (Lanjutan)

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
12.	SIKORAN-F-013	Lihat kelas	Admin sistem dapat melihat daftar kelas yang telah tersimpan dalam sistem
13.	SIKORAN-F-014	Tambah kelas	Admin sistem dapat menambahkan kelas baru ke dalam sistem
14.	SIKORAN-F-015	Ubah kelas	Admin sistem dapat mengubah data kelas yang tersimpan dalam sistem
15.	SIKORAN-F-016	Nonaktifkan kelas	Admin sistem dapat mengubah status kelas menjadi nonaktif
16.	SIKORAN-F-017	Aktifkan kelas	Admin sistem dapat mengubah status kelas menjadi aktif
17.	SIKORAN-F-018	Lihat identitas siswa	Admin sistem dapat melihat daftar siswa yang telah tersimpan dalam sistem
18.	SIKORAN-F-019	Tambah siswa	Admin sistem dapat menambahkan identitas siswa baru ke dalam sistem
19.	SIKORAN-F-020	Ubah identitas siswa	Admin sistem dapat mengubah identitas siswa yang tersimpan dalam sistem
20.	SIKORAN-F-021	Nonaktifkan siswa	Admin sistem dapat mengubah status siswa menjadi nonaktif
21.	SIKORAN-F-022	Aktifkan siswa	Admin sistem dapat mengubah status siswa menjadi aktif
22.	SIKORAN-F-023	Lihat pengguna sistem	Admin sistem dapat melihat daftar pengguna sistem yang telah tersimpan dalam sistem
23.	SIKORAN-F-024	Tambah pengguna sistem	Admin sistem dapat menambahkan pengguna baru ke dalam sistem

Tabel 4.4 (Lanjutan)

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
24.	SIKORAN-F-025	Ubah pengguna sistem	Admin sistem dapat mengubah data pengguna yang tersimpan dalam sistem
25.	SIKORAN-F-026	Nonaktifkan pengguna sistem	Admin sistem dapat mengubah status pengguna sistem menjadi nonaktif
26.	SIKORAN-F-027	Aktifkan pengguna sistem	Admin sistem dapat mengubah status pengguna sistem menjadi aktif

C. Admin soal

Daftar kebutuhan fungsional untuk aktor admin soal dapat dilihat pada Tabel 4.5.

Tabel 4.5 Kebutuhan fungsional admin soal

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIKORAN-F-002	Logout	Admin soal dapat melakukan logout jika ia berkeinginan untuk keluar dari sistem
2.	SIKORAN-F-028	Lihat diagram kemampuan siswa	Admin soal dapat melihat jumlah siswa yang mampu menjawab soal dengan benar untuk setiap soal yang diujikan dalam mata pelajaran tertentu
3.	SIKORAN-F-029	Simpan diagram kemampuan siswa	Admin soal dapat menyiman diagram kemampuan siswa soal dalam format gambar
4.	SIKORAN-F-030	Lihat diagram tingkat kesukaran soal	Admin soal dapat melihat diagram tingkat kesukaran setiap soal yang diujikan dalam mata pelajaran tertentu
5.	SIKORAN-F-031	Simpan diagram tigkat kesukaran soal	Admin soal dapat menyimpan diagram tingkat kesukaran soal dalam format gambar

Tabel 4.5 (Lanjutan)

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
6.	SIKORAN-F-032	Lihat diagram sebaran nilai	Admin soal dapat melihat tingkat kesukaran setiap soal yang diujikan dalam mata pelajarn tertentu
7.	SIKORAN-F-033	Simpan diagram sebaran nilai	Admin soal dapat menyimpan diagram sebaran nilai siswa dalam format gambar
8.	SIKORAN-F-034	Lihat grafik statistik nilai	Admin soal dapat melihat statistik dan perkembangan nilai siswa dari satu periode ke periode ujian berikutnya
9.	SIKORAN-F-035	Simpan grafik statistik nilai	Admin soal dapat menyimpan grafik statistik nilai siswa dalam format gambar
10.	SIKORAN-F-036	Lihat hasil ujian kolektif	Admin soal dapat melihat nilai seluruh siswa untuk seluruh mata pelajaran yang diujikan dalam periode ujian tententu
11.	SIKORAN-F-037	Cetak hasil ujian kolektif	Admin soal dapat mencetak nilai seluruh siswa untuk seluruh mata pelajaran yang diujikan dalam periode ujian tententu
12.	SIKORAN-F-038	Lihat hasil ujian per mapel	Admin soal dapat melihat nilai seluruh siswa untuk satu mata pelajaran tertentu dalam periode ujian tertentu
13.	SIKORAN-F-039	Cetak hasil ujian per mapel	Admin soal dapat mencetak nilai seluruh siswa untuk satu mata pelajaran tertentu dalam periode ujian tertentu

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

Tabel 4.5 (Lanjutan)

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
14.	SIKORAN-F-040	Buat kunci jawaban	Admin soal dapat menambahkan kunci jawaban baru ke dalam sistem
15.	SIKORAN-F-041	Lihat kunci jawaban	Admin soal dapat melihat kunci jawaban yang telah tersimpan dalam sistem
16.	SIKORAN-F-042	Ubah kunci jawaban	Admin soal dapat mengubah kunci jawaban yang tersimpan dalam sistem
17.	SIKORAN-F-043	Hapus kunci jawaban	Admin soal dapat menghapus kunci jawaban dari sistem

D. Korektor

Daftar kebutuhan fungsional untuk aktor korektor dapat dilihat pada Tabel 4.6.

Tabel 4.6 Kebutuhan fungsional korektor

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
1.	SIKORAN-F-002	Logout	Korektor dapat melakukan logout jika ia berkeinginan untuk keluar dari sistem
2.	SIKORAN-F-038	Lihat hasil ujian per mapel	Korektor dapat melihat nilai seluruh siswa untuk satu mata pelajaran tertentu dalam periode ujian tertentu
3.	SIKORAN-F-044	Koreksi jawaban manual	Korektor dapat memasukkan jawaban siswa ke dalam sistem dengan cara memilih pilihan jawaban sesuai dengan jawaban yang tertera pada LJK siswa
4.	SIKORAN-F-045	Lihat koreksi saya	Korektor dapat melihat daftar siswa yang ia koreksi jawabannya

Tabel 4.6 (Lanjutan)

No.	Kode Kebutuhan	Nama Fungsi	Deskripsi
5.	SIKORAN-F-046	Ubah jawaban siswa	Korektor dapat mengubah jawaban siswa yang tersimpan dalam sistem
6.	SIKORAN-F-047	Hapus jawaban siswa	Korektor dapat menghapus jawaban siswa dari sistem
7.	SIKORAN-F-048	Koreksi LJK otomatis	Korektor dapat memasukkan gambar LJK siswa yang sebelumnya telah dipindai dengan mesin pemindai Canon MX390
8.	SIKORAN-F-049	Lihat lembar jawab komputer	Korektor dapat melihat gambar dari LJK yang sudah dikoreksi
9.	SIKORAN-F-050	Ubah hasil koreksi otomatis	Korektor dapat mengubah data yang diperoleh dari proses pengoreksian secara otomatis
10.	SIKORAN-F-051	Simpan hasil koreksi otomatis	Korektor dapat menyimpan hasil koreksi otomatis satu per satu ke dalam database sistem
11.	SIKORAN-F-052	Hapus hasil koreksi otomatis	Korektor dapat menghapus hasil koreksi otomatis dari database sistem
12.	SIKORAN-F-053	Simpan semua hasil koreksi otomatis	Korektor dapat menyimpan seluruh hasil koreksi otomatis ke dalam sistem

4.3.3 Kebutuhan Non Fungsional

Penentuan kebutuhan non fungsional didasarkan pada hasil penggalian dan analisis kebutuhan yang dipaparkan dalam sub bab 4.1. Daftar kebutuhan non fungsional dapat dilihat pada Tabel 4.7.

Tabel 4.7 Kebutuhan non fungsional sistem

No.	Kode Kebutuhan	Parameter	Deskripsi
1.	SIKORAN-NF-001	*Security*	Sistem akan menggunakan enkripsi *bcrypt* untuk keamanan *password* pengguna sistem
2.	SIKORAN-NF-002	*Compatibility*	Sistem dapat dijalankan dalam berbagai jenis aplikasi web *browser*

4.4 Pemodelan Kebutuhan

Pemodelan kebutuhan dilaksanakan dalam beberapa tahap. Pertama, kebutuhan akan digambarkan menggunakan *use case diagram*. Setelah itu, masing-masing *use case* yang terdapat pada *use case diagram* akan dideskripsikan secara detail menggunakan *use case scenario*.

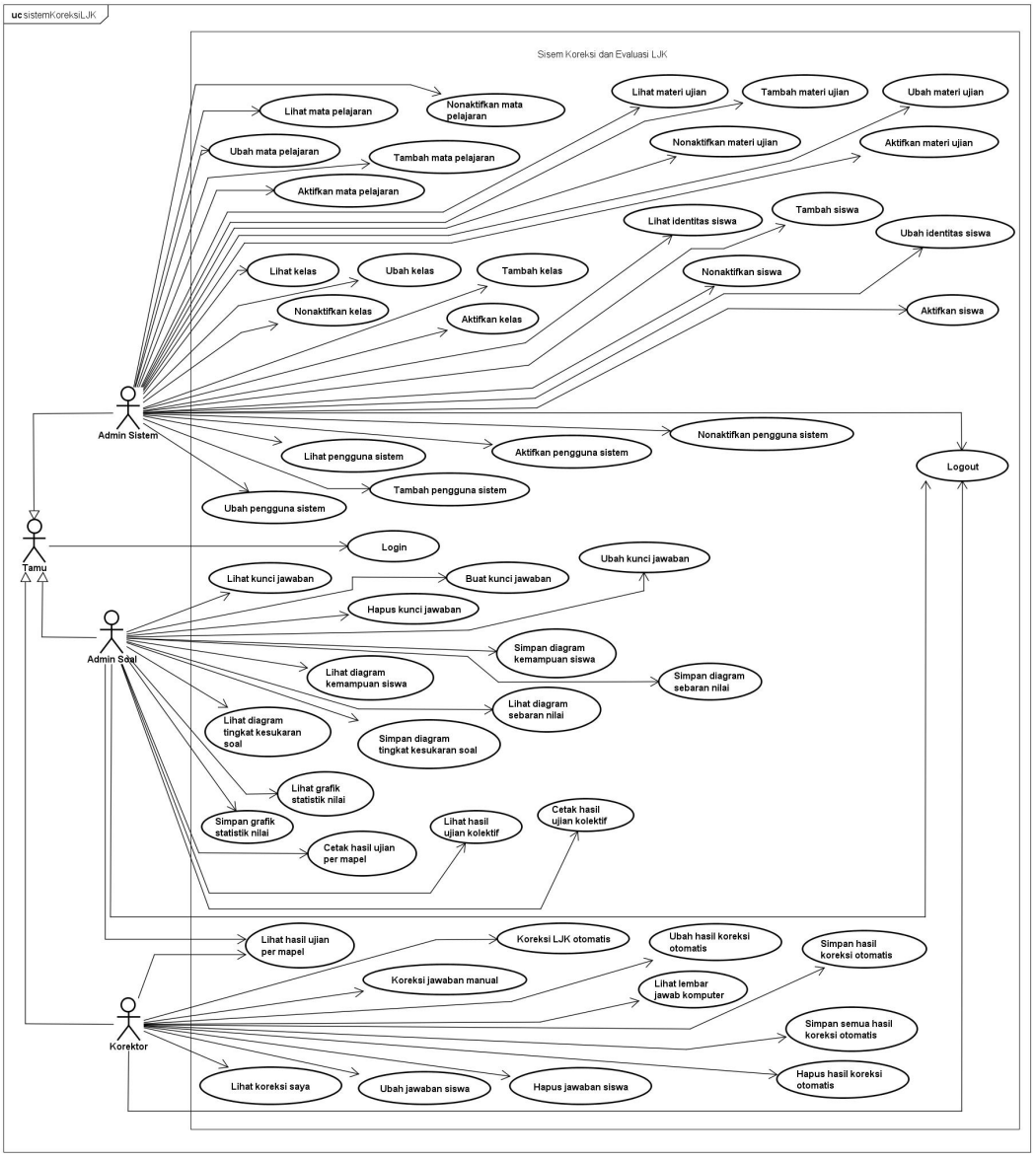
4.4.1 *Use Case Diagram*

Gambar 4.2 menggambarkan kebutuhan fungsional dari SIKORAN. Terdapat 4 aktor yang akan berperan dalam sistem ini, antara lain tamu, admin sistem, admin soal dan korektor. Dalam diagram tersebut juga digambarkan adanya 53 aktivitas yang dilakukan di dalam sistem. Setiap aktor hanya dapat lakukan aktivitas-aktivitas tertentu sesuai yang telah dijelaskan pada sub bab 4.3.2.

Aktor admin sistem dapat menjalankan 26 *use case* mulai dari lihat mata pelajaran, tambah mata pelajaran, ubah mata pelajaran, nonaktifkan mata pelajaran, aktifkan mata pelajaran, *dst* sampai dengan logout.

Aktor admin soal dapat menjalankan 17 *use case* mulai dari buat kunci jawaban, lihat kunci jawaban, ubah kunci jawaban, hapus kunci jawaban, lihat hasil ujian per mapel, cetak hasil ujian per mapel, lihat diagam tingkat kesukaran soal, *dst* sampai pada *use case* logout.

Terdapat 12 *use case* yang dapat dijalankan oleh aktor korektor. *Use case-use case* tersebut dimulai dari koreksi LJK otomatis, lihat lembar jawab komputer, ubah hasil koreksi otomatis, *dsb* sampai pada *logout*.



Gambar 4.2 *Use case diagram* SIKORAN

4.4.2 *Use Case Scenario*

Use case scenario dari setiap kebutuhan fungsional yang telah digambarkan pada Gambar 4.2 akan dideskripsikan dalam bentuk tabel. Tabel *use case scenario* akan berisi beberapa keterangan seperti nama *use case*, deskripsi singkat *use case*, aktor yang berperan, *pre* dan *post condition* serta *main* dan *alternative flow*.

Tabel 4.8 Skenario login

Login (SIKORAN-F-001)	
Aktor	Tamu

Tabel 4.8 (Lanjutan)

Login (SIKORAN-F-001)	
Deskripsi	Tamu ingin masuk ke dalam sistem agar dapat mengakses fitur-fitur yang telah disediakan untuknya
Pre condition	1. Tamu memiliki *username* dan *password* yang masih berlaku
Main flow	1. Tamu memasukkan *username* 2. Tamu memasukkan *password* 3. Tamu memilih fungsi untuk melakukan login 4. Sistem menampilkan fitur-fitur yang dapat diakses aktor sesuai dengan perannya dalam sistem
Alternative flow	1a. Tamu tidak memasukkan *username* 1a1. Sistem memerintahkan tamu agar memasukkan *username* 1b. *Username* tidak terdaftar dalam sistem 1b1. Sistem menampilkan pemberitahuan bahwa *username* tidak terdaftar dalam sistem 2a. Tamu tidak memasukkan *password* 2a1. Sistem memerintahkan tamu agar memasukkan *password* 2b. *Password* yang dimasukkan tidak sesuai 2b1. Sistem menampilkan pemberitahuan bahwa *password* yang dimasukkan salah
Post condition	1. *Session* aktor diaktifkan

Tabel 4.9 Skenario logout

Logout (SIKORAN-F-002)	
Aktor	Admin sistem, admin soal atau korektor
Deskripsi	Aktor ingin keluar dari sistem karena aktor telah menyelesaikan aktivitasnya dalam sistem
Pre condition	1. Aktor telah menyelesaikan aktivitasnya dalam sistem

Tabel 4.9 (Lanjutan)

Logout (SIKORAN-F-002)	
Main flow	1. Aktor memilih menu untuk melakukan logout 2. Sistem mengeluarkan aktor dari sistem
Alternative flow	-
Post condition	1. *Session* aktor terapus dari sistem

Tabel 4.10 Skenario lihat mata pelajaran

Lihat Mata Pelajaran (SIKORAN-F-003)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin melihat seluruh daftar pelajaran yang tersimpan dalam sistem
Pre condition	-
Main flow	1. Admin sistem memilih fungsi untuk menampilkan daftar mata pelajaran 2. Sistem menampilkan daftar mata pelajaran yang tersimpan dalam sistem
Alternative flow	2a. Sistem tidak dapat menampilkan daftar mata pelajaran 2a1. Sistem menampilkan pesan bahwa tidak ada data yang bisa ditampilkan
Post condition	1. Aktor mengetahui daftar mata pelajaran yang tersimpan dalam sistem

Tabel 4.11 Skenario tambah mata pelajaran

Tambah Mata Pelajaran (SIKORAN-F-004)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin menambahkan mata pelajaran baru ke dalam sistem
Pre condition	1. Admin sistem memiliki nama dan alias mata pelajaran yang akan ditambahkan dalam sistem

Tabel 4.11 (Lanjutan)

Tambah Mata Pelajaran (SIKORAN-F-004)	
Main flow	1. Admin sistem memasukkan nama mata pelajaran 2. Admin sistem memasukkan akronim atau singkatan dari mata pelajaran yang terdiri dari 3 karakter 3. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 4. Sistem menyimpan data mata pelajaran 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan seluruh daftar mata pelajaran yang tersimpan dalam sistem
Alternative flow	1a. Admin sistem tidak memasukkan nama mata pelajaran 1a1. Sistem menampilkan perintah untuk memasukkan nama mata pelajaran 1b. Nama mata pelajaran yang dimasukkan sudah ada dalam sistem 1b1. Sistem memberitahukan bahwa nama mata pelajaran sudah digunakan 2a. Admin sistem tidak memasukkan akronim atau singkatan dari mata pelajaran 2a1. Sistem menampilkan perintah untuk memasukkan akronim atau singkatan dari mata pelajaran 2b. Akronim atau singkatan mata pelajaran yang dimasukkan sudah ada dalam sistem 2b1. Sistem memberitahukan bahwa akronim atau singkatan mata pelajaran sudah digunakan 2c. Akronim atau singkatan mata pelajaran yang dimasukkan lebih dari 3 karakter 2c1. Sistem memberitahukan bahwa panjang akronim atau singkatan yang dimasukkan melebihi 3 karakter
Post condition	1. Data mata pelajaran tersimpan dalam sistem

Tabel 4.12 Skenario ubah mata pelajaran

Ubah Mata Pelajaran (SIKORAN-F-005)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah salah satu mata pelajaran yang telah tersimpan dalam sistem
Pre condition	1. Admin sistem mengetahui mata pelajaran yang akan diubah 2. Mata pelajaran berstatus aktif 3. Admin sistem memiliki data mata pelajaran terbaru yang akan digunakan untuk mengubah mata pelajaran terpilih 4. Admin sistem telah menyelesaikan *use case* Lihat Mata Pelajaran (SIKORAN-F-003)
Main flow	1. Admin sistem memilih fungsi untuk mengubah data dari satu mata pelajaran yang terdapat dalam daftar 2. Sistem menampilkan data mata pelajaran yang dapat diubah 3. Admin sistem mengganti sebagian atau seluruh data mata pelajaran dengan data terbaru 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data mata pelajaran 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan daftar mata pelajaran yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem tidak memasukkan nama mata pelajaran 3a1. Sistem menampilkan perintah untuk memasukkan nama mata pelajaran 3b. Nama mata pelajaran yang dimasukkan sudah ada dalam sistem 3b1. Sistem memberitahukan bahwa nama mata pelajaran sudah digunakan

Repository Universitas Brawijaya
REPOSITORY.UB.AC.ID
UNIVERSITAS BRAWIJAYA

Tabel 4.12 (Lanjutan)

	Ubah Mata Pelajaran (SIKORAN-F-005)
	3c. Admin sistem tidak memasukkan akronim atau singkatan dari mata pelajaran 3c1. Sistem menampilkan perintah untuk memasukkan akronim atau singkatan dari mata pelajaran 3d. Akronim atau singkatan mata pelajaran yang dimasukkan sudah ada dalam sistem 3d1. Sistem memberitahukan bahwa akronim atau singkatan mata pelajaran sudah digunakan 3e. Akronim atau singkatan mata pelajaran yang dimasukkan lebih dari 3 karakter 3e1. Sistem memberitahukan bahwa panjang akronim atau singkatan yang dimasukkan melebihi 3 karakter
Post condition	1. Data mata pelajaran berhasil diubah

Tabel 4.13 Skenario nonaktifkan mata pelajaran

	Nonaktifkan Mata Pelajaran (SIKORAN-F-006)
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu mata pelajaran yang semula aktif menjadi nonaktif
Pre condition	1. Admin sistem mengetahui mata pelajaran yang akan dinonaktifkan 2. Mata pelajaran berstatus aktif 3. Admin sistem telah menyelesaikan *use case* Lihat Mata Pelajaran (SIKORAN-F-003)
Main flow	1. Admin sistem memilih fungsi untuk menonaktifkan salah satu mata pelajaran yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan aktor untuk menonaktifkan mata pelajaran

Tabel 4.13 (Lanjutan)

Nonaktifkan Mata Pelajaran (SIKORAN-F-006)	
	3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk menonaktifkan mata pelajaran 4. Sistem mengubah status mata pelajaran menjadi nonaktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan seluruh daftar mata pelajaran yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak menonaktifkan mata pelajaran
Post condition	1. Status mata pelajaran berubah menjadi nonaktif

Tabel 4.14 Skenario aktifkan mata pelajaran

Aktifkan Mata Pelajaran (SIKORAN-F-007)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu mata pelajaran yang semula nonaktif menjadi aktif
Pre condition	1. Admin sistem mengetahui mata pelajaran yang akan diaktifkan 2. Mata pelajaran berstatus nonaktif 3. Admin sistem telah menyelesaikan *use case* Lihat Mata Pelajaran (SIKORAN-F-003)
Main flow	1. Admin sistem memilih fungsi untuk mengaktifkan salah satu mata pelajaran yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan aktor untuk mengaktifkan mata pelajaran 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk mengaktifkan mata pelajaran 4. Sistem mengubah status mata pelajaran menjadi aktif 5. Sistem menampilkan pesan sukses

Tabel 4.14 (Lanjutan)

Aktifkan Mata Pelajaran (SIKORAN-F-007)	
	6. Sistem menampilkan seluruh daftar mata pelajaran yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak mengaktifkan mata pelajaran
Post condition	1. Status mata pelajaran berubah menjadi nonaktif

Tabel 4.15 Skenario lihat materi ujian

Lihat Materi Ujian (SIKORAN-F-008)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin melihat seluruh daftar materi ujian yang tersimpan dalam sistem
Pre condition	-
Main flow	1. Admin sistem memilih fungsi untuk menampilkan daftar materi ujian 2. Sistem menampilkan daftar materi ujian yang tersimpan dalam sistem
Alternative flow	2a. Sistem tidak dapat menampilkan daftar materi ujian 2a1. Sistem menampilkan pesan bahwa tidak ada data yang bisa ditampilkan
Post condition	1. Aktor mengetahui daftar materi ujian yang tersimpan dalam sistem

Tabel 4.16 Skenario tambah materi ujian

Tambah Materi Ujian (SIKORAN-F-009)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin menambahkan materi ujian baru ke dalam sistem
Pre condition	1. Admin sistem memiliki data materi ujian yang akan ditambahkan ke dalam sistem

Tabel 4.16 (Lanjutan)

	Tambah Materi Ujian (SIKORAN-F-009)
Main flow	1. Sistem menampilkan pilihan mata pelajaran 2. Admin sistem memilih salah satu mata pelajaran 3. Admin sistem memasukkan nama materi ujian 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data materi ujian 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan seluruh daftar materi ujian yang tersimpan dalam sistem
Alternative flow	2a. Admin sistem tidak memilih satupun mata pelajaran 2a1. Sistem menampilkan perintah untuk memilih mata pelajaran 3a. Admin sistem tidak memasukkan nama materi ujian 3a1. Sistem menampilkan perintah untuk memasukkan nama materi ujian 3b. Nama materi ujian yang dimasukkan sudah ada dalam sistem 3b1. Sistem memberitahukan bahwa nama materi ujian sudah digunakan
Post condition	1. Data materi ujian tersimpan dalam sistem

Tabel 4.17 Skenario ubah materi ujian

	Ubah Materi Ujian (SIKORAN-F-010)
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah data salah satu materi ujian yang tersimpan dalam sistem
Pre condition	1. Admin sistem mengetahui materi ujian yang akan diubah 2. Materi ujian berstatus aktif

Tabel 4.17 (Lanjutan)

Ubah Materi Ujian (SIKORAN-F-010)	
	3. Admin sistem memiliki data materi ujian terbaru yang akan digunakan untuk mengubah materi ujian terpilih 4. Admin sistem telah menyelesaikan *use case* Lihat Materi Ujian (SIKORAN-F-008)
Main flow	1. Admin sistem memilih fungsi untuk mengubah data dari salah satu materi ujian yang terdapat dalam daftar 2. Sistem menampilkan data materi ujian yang dapat diubah 3. Admin sistem mengganti sebagian atau seluruh data materi ujian dengan data terbaru 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data materi ujian 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan daftar materi ujian yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem tidak memasukkan nama materi ujian 3a1. Sistem menampilkan perintah untuk memasukkan nama materi ujian 3b. Nama materi ujian yang dimasukkan sudah ada dalam sistem 3b1. Sistem memberitahukan bahwa nama materi ujian sudah digunakan
Post condition	1. Data materi ujian terbaru tersimpan dalam sistem

Tabel 4.18 Skenario nonaktifkan materi ujian

Nonaktifkan Materi Ujian (SIKORAN-F-011)	
Aktor	Admin sistem

Tabel 4.18 (Lanjutan)

Nonaktifkan Materi Ujian (SIKORAN-F-011)	
Deskripsi	Admin sistem ingin mengubah status salah satu materi ujian yang semula aktif menjadi nonaktif
Pre condition	1. Admin sistem mengetahui data materi ujian yang akan dinonaktifkan 2. Materi ujian berstatus aktif 3. Admin sistem telah menyelesaikan *use case* Lihat Materi Ujian (SIKORAN-F-008)
Main flow	1. Admin sistem memilih fungsi untuk menonaktifkan salah satu materi ujian yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk menonaktifkan materi ujian 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk menonaktifkan materi ujian 4. Sistem mengubah status materi ujian menjadi nonaktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan seluruh daftar materi ujian yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak menonaktifkan materi ujian
Post condition	1. Status materi ujian berubah menjadi nonaktif

Tabel 4.19 Skenario aktifkan materi ujian

Aktifkan Materi Ujian (SIKORAN-F-012)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu materi ujian yang semula nonaktif menjadi aktif
Pre condition	1. Admin sistem mengetahui data materi ujian yang akan dinonaktifkan 2. Materi ujian berstatus aktif

Tabel 4.19 (Lanjutan)

	Aktifkan Materi Ujian (SIKORAN-F-012)
	3. Admin sistem telah menyelesaikan *use case* Lihat Materi Ujian (SIKORAN-F-008)
Main flow	1. Admin sistem memilih fungsi untuk mengaktifkan salah satu materi ujian yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk mengaktifkan materi ujian 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk mengaktifkan materi ujian 4. Sistem mengubah status materi ujian menjadi aktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan seluruh daftar materi ujian yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak mengaktifkan materi ujian
Post condition	1. Status materi ujian berubah menjadi aktif

Tabel 4.20 Skenario lihat kelas

	Lihat Kelas (SIKORAN-F-013)
Aktor	Admin sistem
Deskripsi	Admin sistem ingin melihat daftar kelas yang tersimpan dalam sistem
Pre condition	-
Main flow	1. Admin sistem memilih fungsi untuk menampilkan daftar kelas 2. Sistem menampilkan daftar kelas yang tersimpan dalam sistem
Alternative flow	2a. Sistem tidak dapat menampilkan daftar kelas 2a1. Sistem menampilkan pesan bahwa tidak ada data yang dapat ditampilkan
Post condition	1. Admin sistem mengetahui daftar kelas yang tersimpan dalam sistem

Tabel 4.21 Skenario tambah kelas

Tambah Kelas (SIKORAN-F-014)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin menambahkan kelas baru ke dalam sistem
Pre condition	1. Admin sistem memiliki data kelas yang akan ditambahkan ke dalam sistem
Main flow	1. Sistem menampilkan pilihan tingkatan kelas 2. Admin sistem memilih salah satu tingkat kelas 3. Admin sistem memasukkan nama kelas 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data kelas 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan daftar kelas yang tersimpan dalam sistem
Alternative flow	2a. Admin sistem tidak memilih satupun tingkat kelas 2a1. Sistem menampilkan perintah untuk memilih tingkat kelas 3a. Admin sistem tidak memasukkan nama kelas 3a1. Sistem menampilkan perintah untuk memasukkan nama kelas 3b. Nama yang dimasukkan sudah ada dalam sistem 3b1. Sistem memberitahukan bahwa nama kelas sudah digunakan
Post condition	1. Data kelas tersimpan dalam sistem

Tabel 4.22 Skenario ubah kelas

Ubah Kelas (SIKORAN-F-015)	
Aktor	Admin sistem

Tabel 4.22 (Lanjutan)

Ubah Kelas (SIKORAN-F-015)	
Deskripsi	Admin sistem ingin mengubah data salah satu kelas yang tersimpan dalam sistem
Pre condition	1. Admin sistem mengetahui kelas yang akan diubah 2. Kelas berstatus aktif 3. Admin sistem memiliki data kelas terbaru yang akan digunakan untuk mengubah kelas terpilih 4. Admin sistem telah menyelesaikan *use case* Lihat Kelas (SIKORAN-F-013)
Main flow	1. Admin sistem memilih fungsi untuk mengubah data dari salah satu kelas yang terdapat dalam daftar 2. Sistem menampilkan data kelas yang dapat diubah 3. Admin sistem mengganti sebagian atau seluruh data kelas dengan data yang baru 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data kelas 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan daftar kelas yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem tidak memasukkan nama kelas 3a1. Sistem menampilkan perintah untuk memasukkan nama kelas 3b. Nama yang dimasukkan sudah ada dalam sistem 3b1. Sistem memberitahukan bahwa nama kelas sudah digunakan
Post condition	1. Data kelas terbaru tersimpan dalam sistem

Tabel 4.23 Skenario nonaktifkan kelas

Nonaktifkan Kelas (SIKORAN-F-016)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu kelas yang semula aktif menjadi nonaktif
Pre condition	1. Admin sistem mengetahui data kelas yang akan dinonaktifkan 2. Kelas berstatus aktif 3. Admin sistem telah menyelesaikan *use case* Lihat Kelas (SIKORAN-F-013)
Main flow	1. Admin sistem memilih fungsi untuk menonaktifkan salah satu kelas yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk menonaktifkan kelas 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk menonaktifkan kelas 4. Sistem mengubah status kelas menjadi nonaktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan daftar kelas yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak menonaktifkan kelas
Post condition	1. Status kelas berubah menjadi nonaktif

Tabel 4.24 Skenario aktifkan kelas

Aktifkan Kelas (SIKORAN-F-017)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu kelas yang semula nonaktif menjadi aktif
Pre condition	1. Admin sistem mengetahui data kelas yang akan diaktifkan 2. Kelas berstatus nonaktif

Tabel 4.24 (Lanjutan)

Aktifkan Kelas (SIKORAN-F-017)	
	3. Admin sistem telah menyelesaikan *use case* Lihat Kelas (SIKORAN-F-013)
Main flow	1. Admin sistem memilih fungsi untuk mengaktifkan salah satu kelas yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk mengaktifkan kelas 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk mengaktifkan kelas 4. Sistem mengubah status kelas menjadi aktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan daftar kelas yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak mengaktifkan kelas
Post condition	1. Status kelas berubah menjadi aktif

Tabel 4.25 Skenario lihat identitas siswa

Lihat Identitas Siswa (SIKORAN-F-018)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin melihat identitas semua siswa yang tersimpan dalam sistem
Pre condition	-
Main flow	1. Admin sistem memilih fungsi untuk menampilkan daftar siswa 2. Sistem menampilkan daftar siswa yang tersimpan dalam sistem
Alternative flow	2a. Sistem tidak dapat menampilkan daftar siswa 2a1. Sistem menampilkan pesan bahwa tidak ada data yang dapat ditampilkan
Post condition	1. Admin sistem mengetahui daftar siswa yang tersimpan dalam sistem

Tabel 4.26 Skenario tambah siswa

Tambah Siswa (SIKORAN-F-019)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin menambahkan siswa baru ke dalam sistem
Pre condition	1. Admin sistem memiliki data siswa yang akan ditambahkan ke dalam sistem
Main flow	1. Admin sistem memasukkan nomor induk siswa 2. Admin sistem memasukkan nama siswa 3. Admin sistem memasukkan kelas yang ditempati oleh siswa 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data siswa 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan daftar siswa yang tersimpan dalam sistem
Alternative flow	1a. Admin sistem tidak memasukkan nomor induk siswa 1a1. Sistem menampilkan perintah untuk memasukkan nomor induk siswa 1b. Nomor induk yang dimasukkan sudah ada dalam sistem 1b1. Sistem memberitahukan bahwa nomor induk sudah digunakan 2a. Admin sistem tidak memasukkan nama siswa 2a1. Sistem menampilkan perintah untuk memasukkan nama siswa 3a. Admin sistem tidak memasukkan kelas yang ditempati oleh siswa 3a1. Sistem menampilkan perintah untuk memasukkan kelas yang ditempati siswa
Post condition	1. Data siswa tersimpan dalam sistem

Tabel 4.27 Skenario ubah identitas siswa

Ubah Identitas Siswa (SIKORAN-F-020)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah identitas salah satu siswa yang telah tersimpan dalam sistem
Pre condition	1. Admin sistem mengetahui identitas siswa yang akan diubah 2. Siswa berstatus aktif 3. Admin sistem memiliki data identitas siswa terbaru yang akan digunakan untuk mengubah identitas siswa terpilih 4. Admin sistem telah menyelesaikan *use case* Lihat Identitas Siswa (SIKORAN-F-018)
Main flow	1. Admin sistem memilih fungsi untuk mengubah identitas salah satu siswa yang terdapat dalam daftar 2. Sistem menampilkan data identitas siswa yang dapat diubah 3. Admin sistem mengganti sebagian atau seluruh data siswa dengan data yang baru 4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 5. Sistem menyimpan data siswa 6. Sistem menampilkan pesan sukses 7. Sistem menampilkan daftar seluruh siswa yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem tidak memasukkan nomor induk siswa 3a1. Sistem menampilkan perintah untuk memasukkan nomor induk siswa 3b. Nomor induk yang dimasukkan sudah ada dalam sistem 3b1. Sistem memberitahukan bahwa nomor induk sudah digunakan

Tabel 4.27 (Lanjutan)

Ubah Identitas Siswa (SIKORAN-F-020)	
	3c. Admin sistem tidak memasukkan nama siswa 3c1. Sistem menampilkan perintah untuk memasukkan nama siswa
Post condition	1. Data identitas siswa tersimpan dalam sistem

Tabel 4.28 Skenario nonaktifkan siswa

Nonaktifkan Siswa (SIKORAN-F-021)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu siswa yang semula aktif menjadi nonaktif
Pre condition	1. Admin sistem mengetahui siswa yang akan dinonaktifkan 2. Siswa berstatus aktif 3. Admin sistem telah menyelesaikan *use case* Lihat Identitas Siswa (SIKORAN-F-018)
Main flow	1. Admin sistem memilih fungsi untuk menonaktifkan salah satu siswa yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk menonaktifkan siswa 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk menonaktifkan siswa 4. Sistem mengubah status siswa menjadi nonaktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan daftar siswa yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak menonaktifkan siswa
Post condition	1. Satus siswa berubah menjadi nonaktif

Tabel 4.29 Skenario aktifkan siswa

Aktifkan Siswa (SIKORAN-F-022)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu siswa yang semula nonaktif menjadi aktif
Pre condition	1. Admin sistem mengetahui siswa yang akan diaktifkan 2. Siswa berstatus nonaktif 3. Admin sistem telah menyelesaikan *use case* Lihat Identitas Siswa (SIKORAN-F-018)
Main flow	1. Admin sistem memilih fungsi untuk mengaktifkan salah satu siswa yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk mengaktifkan siswa 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk mengaktifkan siswa 4. Sistem mengubah status siswa menjadi aktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan daftar siswa yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak mengaktifkan siswa
Post condition	1. Satus siswa berubah menjadi aktif

Tabel 4.30 Skenario lihat pengguna sistem

Lihat Pengguna Sistem (SIKORAN-F-023)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin melihat daftar pengguna yang tersimpan dalam sistem
Pre condition	-
Main flow	1. Admin sistem memilih fungsi untuk menampilkan daftar pengguna sistem

Tabel 4.30 (Lanjutan)

Lihat Pengguna Sistem (SIKORAN-F-023)	
	2. Sistem menampilkan daftar pengguna yang tersimpan dalam sistem
Alternative flow	2a. Sistem tidak dapat menampilkan daftar pengguna 2a1. Sistem menampilkan pesan bahwa tidak ada data yang dapat ditampilkan
Post condition	1. Admin sistem mengetahui daftar pengguna yang tersimpan dalam sistem

Tabel 4.31 Skenario tambah pengguna sistem

Tambah Pengguna Sistem (SIKORAN-F-024)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin menambahkan pengguna baru ke dalam sistem
Pre condition	1. Admin sistem memiliki data pengguna yang akan ditambahkan ke dalam sistem
Main flow	1. Admin sistem memasukkan nama pengguna 2. Admin sistem memasukkan peran pengguna dalam sistem 3. Admin sistem memasukkan *username* yang dapat digunakan pengguna untuk login 4. Admin sistem memasukkan *password* yang dapat digunakan pengguna untuk login 5. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan 6. Sistem menyimpan data pengguna 7. Sistem menampilkan pesan sukses 8. Sistem menampilkan daftar pengguna yang tersimpan dalam sistem
Alternative flow	1a. Aktor tidak memasukkan nama pengguna 1a1. Sistem menampilkan perintah untuk memasukkan nama pengguna

Tabel 4.31 (Lanjutan)

Tambah Pengguna Sistem (SIKORAN-F-024)	
	2a. Admin sistem tidak memasukkan peran pengguna
	2a1. Sistem menampilkan perintah untuk memasukkan peran pengguna
	3a. Admin sistem tidak memasukkan *username* yang dapat digunakan pengguna untuk login
	3a1. Sistem menampilkan perintah untuk memasukkan *username*
	3b. *Username* yang dimasukkan sudah digunakan pengguna lain
	3b1. Sistem memberitahukan bahwa *username* sudah digunakan pengguna lain
	4a. Admin sistem tidak memasukkan *password* yang dapat digunakan pengguna untuk login
	4a1. Sistem menampilkan perintah untuk memasukkan *password*
Post condition	1. Data pengguna tersimpan dalam sistem

Tabel 4.32 Skenario ubah pengguna sistem

Ubah Pengguna Sistem (SIKORAN-F-025)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah data salah satu pengguna yang telah tersimpan dalam sistem
Pre condition	1. Admin sistem mengetahui pengguna yang akan diubah 2. Pengguna berstatus aktif 3. Admin sistem memiliki data pengguna terbaru yang akan digunakan untuk mengubah data saat ini 4. Admin sistem telah menyelesaikan *use case* Lihat Pengguna Sistem (SIKORAN-F-023)
Main flow	1. Aktor memilih fungsi untuk mengubah salah satu pengguna yang terdapat dalam daftar

Tabel 4.32 (Lanjutan)

Ubah Pengguna Sistem (SIKORAN-F-025)	
	2. Sistem menampilkan data pengguna yang dapat diubah
	3. Admin sistem mengganti sebagian atau seluruh data pengguna dengan data yang baru
	4. Admin sistem memilih fungsi untuk menyimpan data yang telah dimasukkan
	5. Sistem menyimpan data pengguna
	6. Sistem menampilkan pesan sukses
	7. Sistem menampilkan daftar pengguna yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem tidak memasukkan nama pengguna
	3a1. Sistem menampilkan perintah untuk memasukkan nama pengguna
	3b. Admin sistem tidak memasukkan *username* yang dapat digunakan pengguna untuk login
	3b1. Sistem menampilkan perintah untuk memasukkan *username*
	3c. *Username* yang dimasukkan sudah digunakan pengguna lain
	3c1. Sistem memberitahukan bahwa *username* sudah digunakan pengguna lain
	3d. Admin sistem tidak memasukkan *password* yang dapat digunakan pengguna untuk login
	3d1. Sistem menampilkan perintah untuk memasukkan *password*
Post condition	1. Data pengguna tersimpan dalam sistem

Tabel 4.33 Skenario nonaktifkan pengguna

Nonaktifkan Pengguna (SIKORAN-F-026)	
Aktor	Admin sistem

Tabel 4.33 (Lanjutan)

Nonaktifkan Pengguna (SIKORAN-F-026)	
Deskripsi	Admin sistem ingin mengubah status salah satu pengguna yang semula akif menjadi nonaktif
Pre condition	1. Admin sistem mengetahui pengguna yang akan dinonaktifkan 2. Pengguna berstatus aktif 3. Admin sistem telah menyelesaikan *use case* Lihat Pengguna Sistem (SIKORAN-F-023)
Main flow	1. Admin sistem memilih fungsi untuk menonaktifkan salah satu pengguna yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk menonaktifkan pengguna 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk menonaktifkan pengguna 4. Sistem mengubah status pengguna menjadi nonaktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan daftar pengguna yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak menonaktifkan pengguna
Post condition	1. Status pengguna berubah menjadi nonaktif

Tabel 4.34 Skenario aktifkan pengguna

Aktifkan Pengguna (SIKORAN-F-027)	
Aktor	Admin sistem
Deskripsi	Admin sistem ingin mengubah status salah satu pengguna yang semula nonakif menjadi aktif
Pre condition	1. Admin sistem mengetahui pengguna yang akan diaktifkan 2. Pengguna berstatus nonaktif 3. Admin sistem telah menyelesaikan *use case* Lihat Pengguna Sistem (SIKORAN-F-023)

Tabel 4.34 (Lanjutan)

Aktifkan Pengguna (SIKORAN-F-027)	
Main flow	1. Admin sistem memilih fungsi untuk mengaktifkan salah satu pengguna yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan admin sistem untuk mengaktifkan pengguna 3. Admin sistem memilih fungsi yang mewakili keyakinannya untuk mengaktifkan pengguna 4. Sistem mengubah status pengguna menjadi aktif 5. Sistem menampilkan pesan sukses 6. Sistem menampilkan daftar pengguna yang tersimpan dalam sistem
Alternative flow	3a. Admin sistem memilih fungsi untuk tidak mengaktifkan pengguna
Post condition	1. Status pengguna berubah menjadi aktif

Tabel 4.35 Skenario lihat diagram kemampuan siswa

Lihat Diagram Kemampuan Siswa (SIKORAN-F-028)	
Aktor	Admin soal
Deskripsi	Admin soal ingin melihat jumlah siswa yang mampu menjawab soal dengan benar untuk setiap soal yang diujikan dalam mata ujian tertentu
Pre condition	1. Jawaban siswa telah dimasukkan dalam sistem
Main flow	1. Admin soal memilih fungsi untuk menampilkan diagram kemampuan siswa 2. Sistem menampilkan paket-paket soal yang telah diujikan 3. Admin soal memilih paket soal yang ingin dilihat 4. Sistem menampilkan kelas-kelas yang mengikuti ujian dengan paket soal yang dipilih admin soal 5. Admin soal memilih kelas

Tabel 4.35 (Lanjutan)

Lihat Diagram Kemampuan Siswa (SIKORAN-F-028)	
	6. Sistem menampilkan diagram yang menggambarkan kemampuan seluruh siswa yang berada dalam kelas terpilih dalam menjawab soal yang terdapat dalam paket soal yang telah terpilih
Alternative flow	-
Post condition	1. Admin soal mengetahui kemampuan siswa dalam menjawab soal

Tabel 4.36 Skenario simpan diagram kemampuan siswa

Simpan Diagram Kemampuan Siswa (SIKORAN-F-029)	
Aktor	Admin soal
Deskripsi	Admin soal telah melihat diagram kemampuan siswa dalam menjawab soal, setelah itu admin soal ingin menyimpan diagram tersebut ke dalam *storage*
Pre condition	1. Admin soal menyelesaikan *use case* Lihat Diagram Kemampuan Siswa (SIKORAN-F-028)
Main flow	1. Admin soal memilih fungsi untuk menyimpan diagram 2. Admin soal memilih lokasi untuk menyimpan diagram 3. Admin soal memilih fungsi untuk menyimpan gambar ke dalam lokasi penyimpanan yang telah ditentukan
Alternative flow	3a. Admin soal memilih fungsi untuk membatalkan penyimpanan gambar pada lokasi yang telah ditentukan
Post condition	1. Gambar diagram kemampuan siswa dalam menjawab soal telah tersimpan dalam *storage*

Tabel 4.37 Skenario lihat diagram tingkat kesukaran soal

Lihat Diagram Tingkat Kesukaran Soal (SIKORAN-F-030)	
Aktor	Admin soal
Deskripsi	Admin soal ingin melihat tingkat kesukaran untuk setiap soal yang diujikan dalam mata ujian tertentu
Pre condition	1. Jawaban siswa telah dimasukkan dalam sistem
Main flow	1. Admin soal memilih fungsi untuk menampilkan diagram tingkat kesukaran soal 2. Sistem menampilkan paket-paket soal yang telah diujikan 3. Admin soal memilih paket soal yang ingin dilihat 4. Sistem menampilkan kelas-kelas yang mengikuti ujian dengan paket soal yang dipilih admin soal 5. Admin soal memilih kelas 6. Sistem menampilkan diagram yang menggambarkan tingkat kesukaran soal
Alternative flow	-
Post condition	1. Admin soal mengetahui tingkat kesukaran soal yang telah diujikan

Tabel 4.38 Skenario simpan diagram tingkat kesukaran soal

Simpan Diagram Tingkat Kesukaran Soal (SIKORAN-F-031)	
Aktor	Admin soal
Deskripsi	Admin soal telah melihat diagram tingkat kesukaran soal. Setelah itu admin soal ingin menyimpan diagram tersebut ke dalam *storage*
Pre condition	1. Admin soal menyelesaikan *use case* Lihat Diagram Tingkat Kesukaran Soal (SIKORAN-F-030)
Main flow	1. Admin soal memilih fungsi untuk menyimpan diagram

Tabel 4.38 (Lanjutan)

Simpan Diagram Tingkat Kesukaran Soal (SIKORAN-F-031)	
	2. Admin soal memilih lokasi untuk menyimpan diagram 3. Admin soal memilih fungsi untuk menyimpan diagram ke dalam lokasi penyimpanan yang telah ditentukan
Alternative flow	3a. Admin soal memilih fungsi untuk membatalkan penyimpanan gambar pada lokasi yang telah ditentukan
Post condition	1. Gambar diagram tingkat kesukaran soal tersimpan dalam *storage*

Tabel 4.39 Skenario lihat diagram sebaran nilai

Lihat Diagram Sebaran Nilai (SIKORAN-F-032)	
Aktor	Admin soal
Deskripsi	Admin soal ingin melihat sebaran nilai siswa dalam paket soal tertentu
Pre condition	1. Jawaban siswa telah dimasukkan dalam sistem
Main flow	1. Admin soal memilih fungsi untuk menampilkan diagram sebaran nilai 2. Sistem menampilkan paket-paket soal yang telah diujikan 3. Admin soal memilih paket soal yang ingin dilihat 4. Sistem menampilkan kelas-kelas yang mengikuti ujian dengan paket soal yang dipilih admin soal 5. Admin soal memilih kelas 6. Sistem menampilkan diagram yang menggambarkan sebaran nilai siswa
Alternative flow	-
Post condition	1. Admin soal mengetahui sebaran nilai siswa untuk paket soal yang telah dipilih

Tabel 4.40 Skenario simpan diagram sebaran nilai

Simpan Diagram Sebaran Nilai (SIKORAN-F-033)	
Aktor	Admin soal
Deskripsi	Admin soal telah melihat diagram sebaran nilai. Setelah itu admin soal ingin menyimpan diagram tersebut ke dalam *storage*
Pre condition	1. Admin soal menyelesaikan *use case* Lihat Diagram Sebaran Nilai (SIKORAN-F-032)
Main flow	1. Admin soal memilih fungsi untuk menyimpan diagram 2. Admin soal memilih lokasi untuk menyimpan diagram 3. Admin soal memilih fungsi untuk menyimpan diagram ke dalam lokasi penyimpanan yang telah ditentukan
Alternative flow	3a. Admin soal memilih fungsi untuk membatalkan penyimpanan gambar pada lokasi yang telah ditentukan
Post condition	1. Gambar diagram sebaran nilai tersimpan dalam *storage*

Tabel 4.41 Skenario lihat grafik statistik nilai

Lihat Grafik Statistik Nilai (SIKORAN-F-034)	
Aktor	Admin soal
Deskripsi	Admin soal ingin melihat grafik statistik nilai siswa dalam mata ujian tertentu
Pre condition	1. Jawaban siswa telah dimasukkan dalam sistem
Main flow	1. Admin soal memilih fungsi untuk menampilkan grafik statistik nilai siswa 2. Sistem menampilkan tahun ajaran dilaksanakannya ujian 3. Admin soal memilih tahun ajaran 4. Sistem menampilkan mata pelajaran yang telah diujikan

Tabel 4.41 (Lanjutan)

Lihat Grafik Statistik Nilai (SIKORAN-F-034)	
	5. Admin soal memilih mata pelajaran 6. Sistem menampilkan kelas yang telah mengikuti ujian 7. Admin soal memilih kelas 8. Sistem menampilkan grafik yang menggambarkan statistik nilai siswa
Alternative flow	-
Post condition	1. Admin soal mengetahui statistik nilai siswa untuk mata ujian tertentu

Tabel 4.42 Skenario simpan grafik statistik nilai

Simpan Grafik Statistik Nilai (SIKORAN-F-035)	
Aktor	Admin soal
Deskripsi	Admin soal telah melihat grafik statistik nilai. Setelah itu admin soal ingin menyimpan grafik tersebut ke dalam *storage*
Pre condition	1. Admin soal menyelesaikan *use case* Lihat Grafik Statistik Nilai (SIKORAN-F-034)
Main flow	1. Admin soal memilih fungsi untuk menyimpan grafik 2. Admin soal memilih lokasi untuk menyimpan grafik 3. Admin soal memilih fungsi untuk menyimpan grafik ke dalam lokasi penyimpananyang telah ditentukan
Alternative flow	3a. Admin soal memilih fungsi untuk membatalkan penyimpanan grafik pada lokasi yang telah ditentukan
Post condition	1. Gambar grafik statistik nilai tersimpan dalam *storage*

Tabel 4.43 Skenario lihat hasil ujian kolektif

Lihat Hasil Ujian Kolektif (SIKORAN-F-036)	
Aktor	Admin soal
Deskripsi	Admin soal ingin melihat daftar nilai dari seluruh peserta ujian dalam satu periode ujian tertentu
Pre condition	1. Jawaban siswa telah dimasukkan dalam sistem
Main flow	1. Admin soal memilih fungsi untuk menampilkan hasil ujian secara kolektif 2. Sistem menampilkan periode-periode dilaksanakannya ujian 3. Admin soal memilih periode ujian 4. Sistem menampilkan daftar kelas yang mengikuti ujian pada periode yang dipilih admin soal 5. Admin Soal memilih kelas 6. Sistem menampilkan nilai dari semua peserta ujian untuk seluruh mata pelajaran yang diujikan dalam periode tersebut
Alternative flow	-
Post condition	1. Admin soal mengetahui daftar nilai dari seluruh peserta ujian dalam satu periode ujian tertentu

Tabel 4.44 Skenario cetak hasil ujian kolektif

Cetak Hasil Ujian Kolektif (SIKORAN-F-037)	
Aktor	Admin soal
Deskripsi	Admin soal telah melihat hasil ujian kolektif. Setelah itu admin soal ingin mencetak hasil ujian tersebut
Pre condition	1. Admin soal menyelesaikan *use case* Lihat Hasil Ujian Kolektif (SIKORAN-F-036)
Main flow	1. Admin soal memilih fungsi untuk menampilkan pengaturan pencetakan

Tabel 4.44 (Lanjutan)

Cetak Hasil Ujian Kolektif (SIKORAN-F-037)	
	2. Sistem menampilkan fungsi pengaturan hasil cetakan agar sesuai dengan keinginan aktor
	3. Admin soal mengatur *printer* yang akan digunakan, jenis kertas, batas kertas, *layout* atau halaman yang ingin dicetak
	4. Admin soal memilih fungsi untuk mencetak hasil ujian
	5. Sistem mencetak hasil ujian
Alternative flow	4a. Admin soal memilih fungsi untuk membatalkan pencetakan hasil ujian
Post condition	1. Hasil ujian kolektif tercetak sesuai pengaturan yang dilakukan admin soal

Tabel 4.45 Skenario lihat hasil ujian per mapel

Lihat Hasil Ujian Per Mapel (SIKORAN-F-038)	
Aktor	Admin soal atau korektor
Deskripsi	Aktor ingin melihat daftar nilai dari seluruh peserta ujian untuk satu mata pelajaran tertentu dalam periode ujian tertentu
Pre condition	1. Jawaban siswa telah dimasukkan dalam sistem
Main flow	1. Aktor memilih fungsi untuk menampilkan hasil ujian per mata pelajaran
	2. Sistem menampilkan mata pelajaran beserta periode ujian mata pelajaran tersebut
	3. Aktor memilih mata pelajaran
	4. Sistem menampilkan daftar kelas yang mengikuti mata pelajaran yang dipilih aktor
	5. Aktor memilih kelas
	6. Sistem menampilkan nilai dari semua siswa untuk mata pelajaran terpilih
Alternative flow	-
Post condition	1. Aktor mengetahui daftar nilai dari peserta ujian untuk satu mata pelajaran tertentu

Tabel 4.46 Skenario cetak hasil ujian per mapel

Cetak Hasil Ujian Per Mapel (SIKORAN-F-039)	
Aktor	Admin soal
Deskripsi	Admin soal telah melihat hasil ujian per mata pelajaran. Setelah itu admin soal ingin mencetak hasil ujian tersebut
Pre condition	1. Admin soal menyelesaikan *use case* Lihat Hasil Ujian Per Mapel (SIKORAN-F-038)
Main flow	1. Admin soal memilih fungsi untuk menampilkan pengaturan pencetakan 2. Sistem menampilkan fungsi pengaturan hasil cetakan agar sesuai keinginan aktor 3. Admin soal mengatur *printer* yang akan digunakan, jenis kertas, batas kertas, *layout* atau halaman yang ingin dicetak 4. Admin soal memilih fungsi untuk mencetak hasil ujian 5. Sistem mencetak hasil ujian
Alternative flow	4a. Admin soal memilih fungsi untuk membatalkan pencetakan hasil ujian
Post condition	1. Hasil ujian untuk mata pelajaran yang telah dipilih tercetak sesuai pengaturan yang dilakukan admin soal

Tabel 4.47 Skenario buat kunci jawaban

Buat Kunci Jawaban (SIKORAN-F-040)	
Aktor	Admin soal
Deskripsi	Admin soal ingin membuat kunci jawaban dan menyimpannya ke dalam sistem
Pre condition	1. Admin soal memiliki data kunci jawaban yang akan ditambahkan ke dalam sistem
Main flow	1. Admin soal memilih fungsi untuk memulai proses pembuatan kunci jawaban 2. Sistem menamilkan fitur untuk membuat kunci jawaban

Tabel 4.47 (Lanjutan)

Buat Kunci Jawaban (SIKORAN-F-040)	
	3. Admin soal memasukkan tahun ajaran
	4. Admin soal memasukkan periode ujian
	5. Admin soal memasukkan mata pelajaran
	6. Admin soal memasukkan kode soal
	7. Admin soal memasukkan jumlah soal
	8. Admin soal memasukkan jumlah pilihan jawaban
	9. Admin soal memasukkan jumlah baris nomor soal yang terdapat pada LJK
	10. Sistem menampilkan soal beserta pilihan jawaban dan pilihan materi ujian sesuai data yang dimasukkan admin soal
	11. Admin soal memilih materi ujian dan jawaban yang paling tepat untuk tiap nomor soal
	12. Admin soal memilih fungsi untuk menyimpan kunci jawaban
	13. Sistem menyimpan kunci jawaban
	14. Sistem menampilkan pesan sukses
Alternative flow	10a. Kunci jawaban untuk mata pelajaran dengan paket soal yang dimasukkan sudah pernah dibuat 10a1. Sistem memberitahukan bahwa kunci jawaban untuk paket soal tersebut sudah tersedia
Post condition	1. Kunci jawaban tersimpan dalam sistem

Tabel 4.48 Skenario lihat kunci jawaban

Lihat Kunci Jawaban (SIKORAN-F-041)	
Aktor	Admin soal
Deskripsi	Admin soal ingin melihat kunci jawaban yang tersimpan dalam sistem
Pre condition	-

Tabel 4.48 (Lanjutan)

Lihat Kunci Jawaban (SIKORAN-F-041)	
Main flow	1. Admin soal memilih fungsi untuk menampilkan daftar kunci jawaban 2. Sistem hanya menampilkan pilihan daftar kunci jawaban yang pernah dibuat oleh aktor 3. Admin soal memilih kunci jawaban yang dikehendaki 4. Sistem menampilkan kunci jawaban
Alternative flow	-
Post condition	1. Aktor mengetahui kunci jawaban yang dikehendaki

Tabel 4.49 Skenario ubah kunci jawaban

Ubah Kunci Jawaban (SIKORAN-F-042)	
Aktor	Admin soal
Deskripsi	Admin soal ingin mengubah salah satu kunci jawaban yang telah tersimpan dalam sistem
Pre condition	1. Admin soal mengetahui kunci jawaban yang akan diubah 2. Admin soal memiliki data kunci jawaban terbaru yang akan digunakan untuk mengubah kunci jawaban sebelumnya 3. Admin soal telah menyelesaikan *use case* Lihat Kunci Jawaban (SIKORAN-F-041)
Main flow	1. Admin soal memilih fungsi untuk mengubah kunci jawaban 2. Sistem mengaktifkan pilihan jawaban dan pilihan materi ujian agar dapat diubah aktor 3. Admin soal mengubah pilihan jawaban atau materi ujian sesuai data terbaru yang dimiliki admin soal 4. Admin soal memilih fungsi untuk menyimpan kunci jawaban 5. Sistem menyimpan kunci jawaban

Tabel 4.49 (Lanjutan)

Ubah Kunci Jawaban (SIKORAN-F-042)	
	6. Sistem menampilkan pesan sukses
Alternative flow	4a. Admin soal memilih fungsi untuk membatalkan perubahan pada kunci jawaban
Post condition	1. Kunci jawaban tersimpan dalam sistem

Tabel 4.50 Skenario hapus kunci jawaban

Hapus Kunci Jawaban (SIKORAN-F-043)	
Aktor	Admin soal
Deskripsi	Admin soal ingin menghapus salah satu kunci jawaban yang telah tersimpan dalam sistem
Pre condition	1. Admin soal mengetahui kunci jawaban yang akan dihapus 2. Admin soal telah menyelesaikan *use case* Lihat Kunci Jawaban (SIKORAN-F-041)
Main flow	1. Admin soal memilih fungsi untuk menghapus kunci jawaban 2. Sistem menanyakan kesungguhan admin soal untuk menghapus kunci jawaban 3. Admin soal memilih fungsi yang mewakili keyakinannya untuk menghapus kunci jawaban 4. Sistem menghapus kunci jawaban 5. Sistem menampilkan pesan sukses
Alternative flow	3a. Admin soal memilih fungsi untuk tidak menghapus kunci jawaban
Post condition	1. Kunci jawaban terhapus dari sistem

Tabel 4.51 Skenario koreksi jawaban manual

Koreksi Jawaban Manual (SIKORAN-F-044)	
Aktor	Korektor

Tabel 4.51 (Lanjutan)

Koreksi Jawaban Manual (SIKORAN-F-044)	
Deskripsi	Korektor memasukkan jawaban siswa kedalam sistem dengan cara memilih jawaban satu persatu sesuai dengan jawaban peserta yang terdapat dalam LJK
Pre condition	1. Korektor memiliki LJK siswa
Main flow	1. Korektor memilih fungsi untuk menampilkan fitur koreksi jawaban secara manual 2. Sistem menampilkan pilihan mata pelajaran 3. Korektor memilih mata pelajaran 4. Sistem menampilkan paket soal sesuai mata pelajaran yang dipilih 5. Korektor memilih paket soal 6. Sistem menampilkan daftar nama siswa 7. Korektor memilih nama siswa sesuai dengan LJK yang sedang dikoreksi 8. Sistem menampilkan nomor induk dan kelas dari siswa yang dipilih korektor 9. Sistem menampilkan nomor soal dan pilihan jawaban sesuai dengan paket soal yang telah dipilih 10. Korektor memilih jawaban sesuai dengan jawaban siswa yang tertera pada LJK 11. Korektor memilih fungsi untuk menyimpan jawaban siswa 12. Sistem menyimpan jawaban yang telah dimasukkan 13. Sistem menampilkan pesan sukses
Alternative flow	-
Post condition	1. Jawaban siswa tersimpan dalam sistem

Tabel 4.52 Tabel skenario lihat koreksi saya

Lihat Koreksi Saya (SIKORAN-F-045)	
Aktor	Korektor

Tabel 4.52 (Lanjutan)

Lihat Koreksi Saya (SIKORAN-F-045)	
Deskripsi	Korektor ingin melihat daftar siswa yang pernah ia koreksi jawabannya
Pre condition	-
Main flow	1. Korektor memilih fungsi untuk menampilkan daftar siswa yang pernah ia koreksi 2. Sistem menampilkan paket soal yang pernah dikoreksi korektor 3. Korektor memilih paket soal yang dikehendaki 4. Sistem menampilkan daftar siswa
Alternative flow	-
Post condition	1. Korektor mengetahui daftar siswa yang pernah ia koreksi

Tabel 4.53 Skenario ubah jawaban siswa

Ubah Jawaban Siswa (SIKORAN-F-046)	
Aktor	Korektor
Deskripsi	Korektor ingin mengubah jawaban salah satu siswa yang tersimpan dalam sistem
Pre condition	1. Korektor telah menyelesaikan *use case* Lihat Koreksi Saya (SIKORAN-F-046) 2. Korektor mengetahui data siswa yang akan diubah jawabannya 3. Korektor memiliki jawaban siswa terbaru yang akan digunakan untuk mengubah jawaban yang tersimpan dalam sistem
Main flow	1. Korektor memilih fungsi untuk mengubah jawaban salah satu siswa yang terdapat dalam daftar 2. Sistem menampilkan jawaban siswa 3. Korektor mengubah jawaban siswa sesuai dengan data terbaru 4. Korektor memilih fungsi untuk menyimpan jawaban siswa

Tabel 4.53 (Lanjutan)

Ubah Jawaban Siswa (SIKORAN-F-046)	
	5. Sistem menyimpan jawaban siswa
	6. Sistem menampilkan pesan sukses
Alternative flow	4a. Korektor memilih fungsi untuk membatalkan perubahan pada jawaban siswa
Post condition	1. Jawaban siswa tersimpan dalam sistem

Tabel 4.54 Skenario hapus jawaban siswa

Hapus Jawaban Siswa (SIKORAN-F-047)	
Aktor	Korektor
Deskripsi	Korektor ingin menghapus jawaban salah satu siswa dari sistem
Pre condition	1. Korektor telah menyelesaikan *use case* Lihat Koreksi Saya (SIKORAN-F-046) 2. Korektor mengetahui data siswa yang akan dihapus jawabannya
Main flow	1. Korektor memilih fungsi untuk menghapus jawaban salah satu siswa yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan korektor untuk menghapus jawaban siswa 3. Korektor memilih fungsi yang mewakili keyakinannya untuk menghapus jawaban siswa 4. Sistem menghapus jawaban siswa 5. Sistem menampilkan pesan sukses
Alternative flow	3a. Korektor memilih fungsi untuk tidak menghapus jawaban siswa
Post condition	1. Jawaban siswa terhapus dari sistem

Tabel 4.55 Skenario koreksi LJK otomatis

Koreksi LJK Otomatis (SIKORAN-F-048)	
Aktor	Korektor
Deskripsi	Korektor memasukkan jawaban siswa kedalam sistem dengan cara memasukkan berkas jawaban siswa yang telah dipindai menggunakan mesin pemindai Canon MX390
Pre condition	1. Korektor memiliki berkas LJK siswa yang telah dipindai
Main flow	1. Sistem menampilkan pilihan mata pelajaran 2. Korektor memilih mata pelajaran 3. Sistem menampilkan paket soal sesuai mata pelajaran yang dipilih 4. Korektor memilih paket soal 5. Korektor memilih berkas LJK yang akan dikoreksi 6. Korektor memilih fungsi untuk memulai pengoreksian berkas LJK 7. Sistem menampilkan pesan sukses 8. Sistem menampilkan hasil pengoreksian
Alternative flow	2a. Korektor tidak memilih mata pelajaran 2a1. Sistem menampilkan perintah untuk memilih mata pelajaran 4a. Korektor tidak memilih paket soal 4a1. Sistem menampilkan perintah untuk memilih paket soal 5a. Korektor tidak memilih satupun berkas untuk dikoreksi 5a1. Sistem menampilkan perintah untuk memilih berkas 5b. Berkas yang dipilih bukan gambar 5b1. Sistem menampilkan pemberitahuan bahwa berkas yang dipilih bukan gambar
Post condition	1. Korektor mengetahui hasil pengoreksian

Tabel 4.56 Skenario lihat lembar jawab komputer

Lihat Lembar Jawab Komputer (SIKORAN-F-049)	
Aktor	Korektor
Deskripsi	Korektor ingin melihat citra dari salah satu LJK yang telah dikoreksi menggunakan SIKORAN
Pre condition	1. Korektor telah menyelesaikan *use case* Koreksi LJK Otomatis (SIKORAN-F-048)
Main flow	1. Sistem menampilkan daftar LJK yang telah dikoreksi menggunakan SIKORAN 2. Korektor memilih fungsi untuk menampilkan citra pada salah satu LJK yang terdapat dalam daftar 3. Sistem menampilkan citra LJK yang dipilih oleh korektor
Alternative flow	-
Post condition	1. Korektor mengetahui citra dari LJK yang ia pilih

Tabel 4.57 Skenario ubah hasil koreksi otomatis

Ubah Hasil Koreksi Otomatis (SIKORAN-F-050)	
Aktor	Korektor
Deskripsi	Korektor ingin mengubah jawaban atau nomor peserta pada salah satu LJK yang telah dikoreksi
Pre condition	1. Korektor telah menyelesaikan *use case* Koreksi LJK Otomatis (SIKORAN-F-048) 2. Korektor mengetahui LJK yang akan diubah 3. Korektor memiliki jawaban atau nomor peserta baru yang akan digunakan untuk mengubah jawaban atau nomor peserta yang telah tersimpan dalam sistem
Main flow	1. Korektor memilih fungsi untuk mengubah hasil koreksi pada salah satu LJK yang terdapat dalam daftar 2. Sistem menampilkan jawaban dan nomor peserta ujian

Tabel 4.57 (Lanjutan)

Ubah Hasil Koreksi Otomatis (SIKORAN-F-050)	
	3. Korektor mengubah jawaban atau nomor peserta ujian sesuai dengan data terbaru 4. Korektor memilih fungsi untuk menyimpan perubahan yang dilakukan 5. Sistem menyimpan jawaban dan nomor peserta ujian 6. Sistem menampilkan pesan sukses
Alternative flow	4a. Korektor memilih fungsi untuk membatalkan perubahan pada hasil koreksi otomatis
Post condition	1. Jawaban dan nomor peserta tersimpan dalam sistem

Tabel 4.58 Skenario simpan hasil koreksi otomatis

Simpan Hasil koreksi Otomatis (SIKORAN-F-051)	
Aktor	Korektor
Deskripsi	Korektor ingin menyimpan hasil koreksi salah satu LJK agar dapat diolah dan ditampilkan ke aktor lainnya
Pre condition	1. Korektor telah menyelesaikan *use case* Koreksi LJK Otomatis (SIKORAN-F-048) 2. Korektor mengetahui hasil koreksi dari LJK yang akan disimpan
Main flow	1. Korektor memilih fungsi untuk menyimpan hasil koreksi pada salah satu LJK yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan korektor untuk menyimpan hasil koreksi dari LJK yang dipilih 3. Korektor memilih fungsi yang mewakili keyakinannya untuk menyimpan hasil koreksi 4. Sistem menyimpan hasil koreksi 5. Sistem menampilkan pesan sukses

Tabel 4.58 (Lanjutan)

Simpan Hasil koreksi Otomatis (SIKORAN-F-051)	
Alternative flow	3a. Korektor memilih fungsi untuk tidak menyimpan hasil koreksi
Post condition	1. Hasil koreksi tersimpan dalam sistem

Tabel 4.59 Skenario hapus hasil koreksi otomatis

Hapus Hasil koreksi Otomatis (SIKORAN-F-052)	
Aktor	Korektor
Deskripsi	Korektor ingin menghapus hasil koreksi salah satu LJK
Pre condition	1. Korektor telah menyelesaikan *use case* Koreksi LJK Otomatis (SIKORAN-F-048) 2. Korektor mengetahui hasil koreksi dari LJK yang akan dihapus
Main flow	1. Korektor memilih fungsi untuk menghapus hasil koreksi pada salah satu LJK yang terdapat dalam daftar 2. Sistem menanyakan kesungguhan korektor untuk menghapus hasil koreksi dari LJK yang dipilih 3. Korektor memilih fungsi yang mewakili keyakinannya untuk menghapus hasil koreksi 4. Sistem menghapus hasil koreksi 5. Sistem menampilkan pesan sukses
Alternative flow	3a. Korektor memilih fungsi untuk tidak menghapus hasil koreksi
Post condition	1. Hasil koreksi terhapus dari sistem

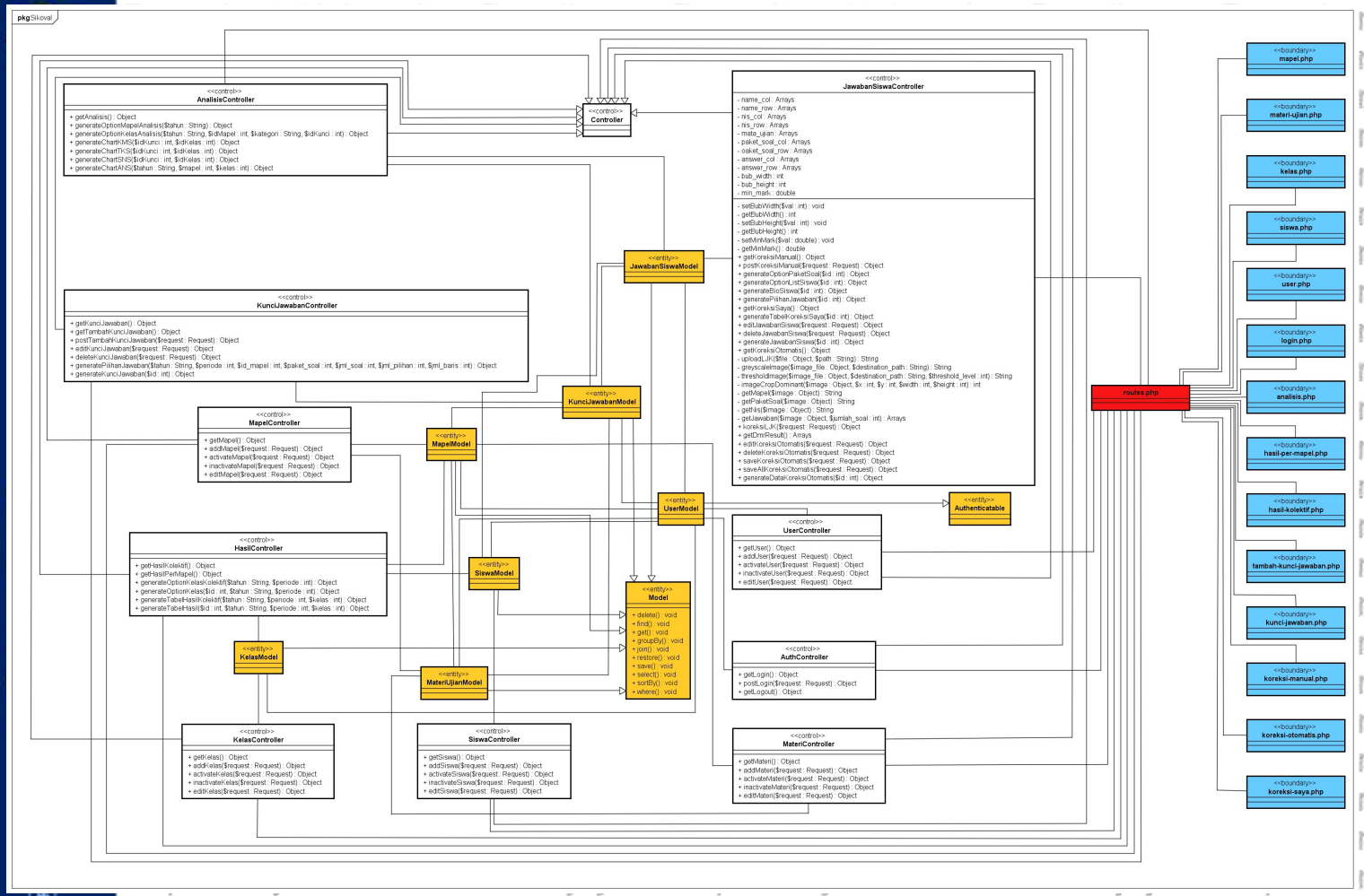
Tabel 4.60 Skenario simpan semua hasil koreksi otomatis

Simpan Semua Hasil koreksi Otomatis (SIKORAN-F-053)	
Aktor	Korektor
Deskripsi	Korektor ingin menyimpan seluruh hasil koreksi LJK agar dapat diolah dan ditampilkan ke aktor lainnya

Repository Universitas Brawijaya

Tabel 4.60 (Lanjutan)

Simpan Semua Hasil koreksi Otomatis (SIKORAN-F-053)	
Pre condition	1. Korektor telah menyelesaikan *use case* Koreksi LJK Otomatis (SIKORAN-F-048)
Main flow	6. Korektor memilih fungsi untuk menyimpan seluruh hasil koreksi terdapat dalam daftar 7. Sistem memberitahukan jumlah LJK yang tidak akan disimpan karena kesalahan pengarsiran dan menanyakan kesungguhan korektor untuk menyimpan seluruh hasil koreksi 8. Korektor memilih fungsi yang mewakili keyakinannya untuk menyimpan seluruh hasil koreksi 9. Sistem menyimpan seluruh hasil koreksi 10. Sistem menampilkan pesan sukses
Alternative flow	3a. Korektor memilih fungsi untuk tidak menyimpan hasil koreksi
Post condition	1. Seluruh hasil koreksi tersimpan dalam sistem



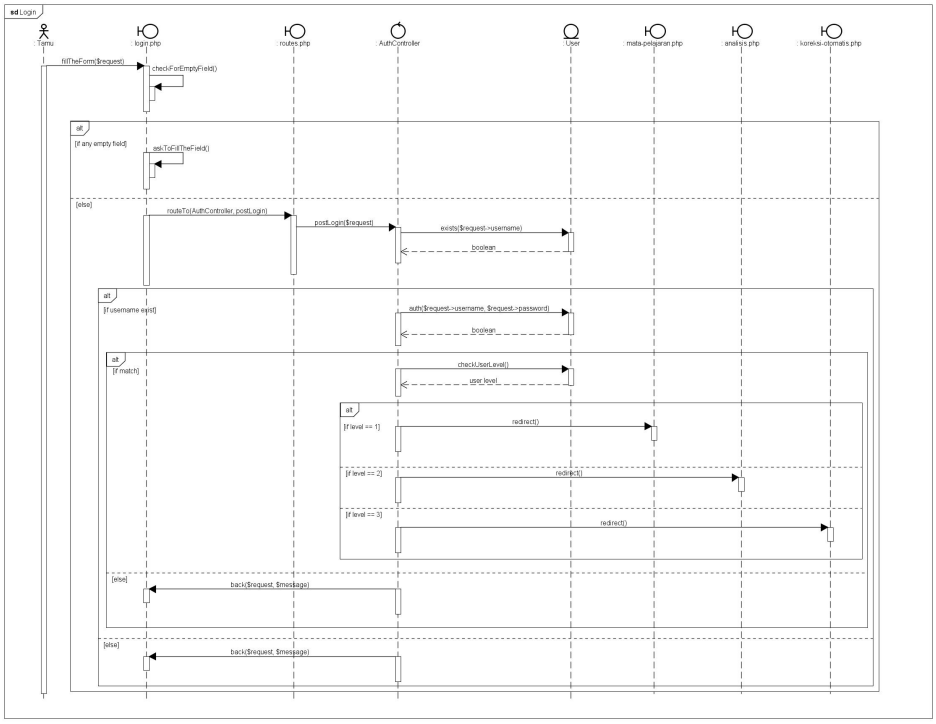
Gambar 5.1 Perancangan *class diagram*

5.2 Perancangan Interaksi Antar Objek

Interaksi yang terjadi antar objek dalam kurun waktu tertentu akan digambarkan menggunakan *sequence diagram*. Tidak semua kebutuhan fungsional yang telah didefinisikan akan digambarkan dalam *sequence diagram,* melainkan hanya beberapa kebutuhan fungsional yang dinilai dapat mewakili fitur inti sistem saja yang akan digambarkan.

a. *Sequence Diagram* Login

Gambar 5.2 menggambarkan interaksi antar objek yang terjadi ketika fitur login dijalankan.



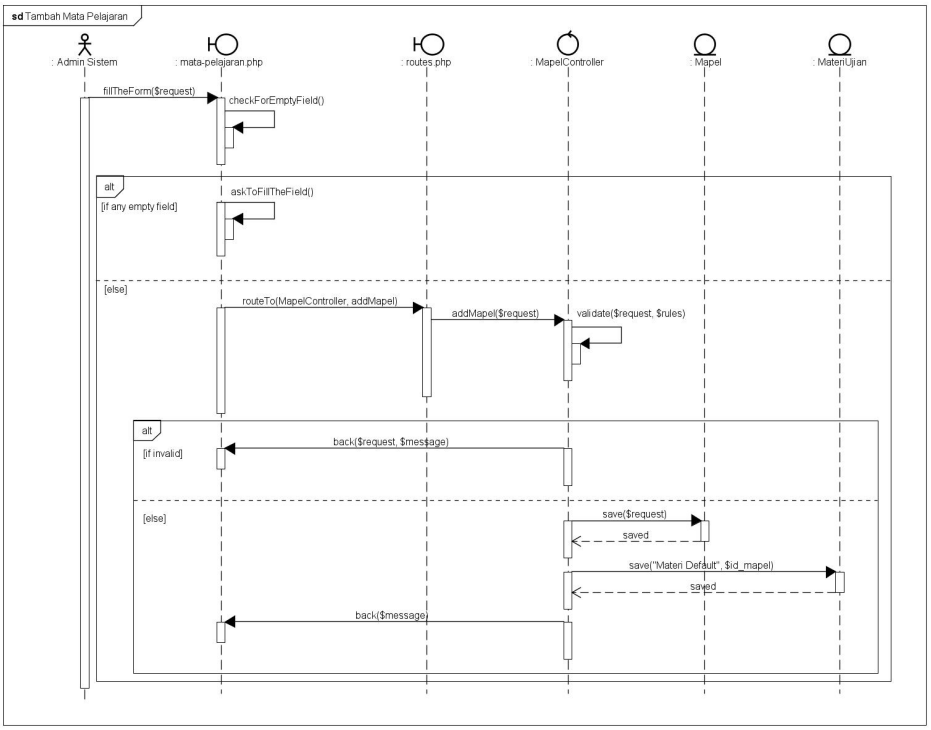
Gambar 5.2 *Sequence diagram* login

Setelah aktor tamu memasukkan *username* dan *password* pada halaman login, maka objek Route akan bertugas menunjukkan *path* yang seharusnya dilalui dan *controller* yang akan menangani pesan yang dikirimkan tamu melalui *view*. Jika *username* dan *password* yang dimasukkan sesuai maka tamu akan diarahkan ke halaman tujuan.

Apabila *username* dan *password* yang dimasukkan oleh aktor tidak sesuai atau belum terdaftar dalam sistem, maka aktor akan dikembalikan ke halaman login.

b. *Sequence Diagram* Tambah Mata Pelajaran

Gambar 5.3 menggambarkan interaksi yang terjadi saat menambahkan mata pelajaran baru.



Gambar 5.3 *Sequence diagram* tambah mata pelajaran

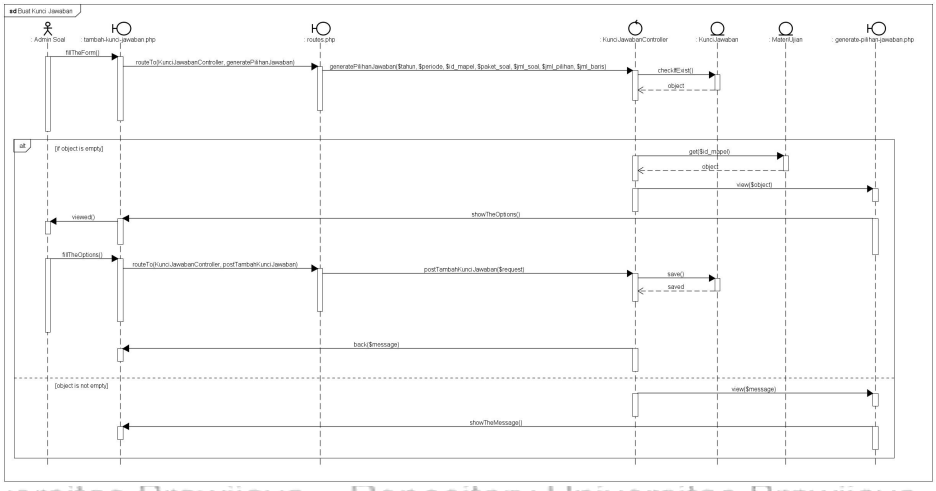
Setelah admin sistem mengisi data mata pelajaran, maka MapelController akan melakukan validasi terhadap data tersebut. Jika data yang dimasukkan sesuai dengan ketentuan yang dibuat, maka data akan disimpan dan akan muncul pesan sukses ketika admin sistem diarahkan ke halaman mata pelajaran. Namun jika data yang dimasukkan tidak sesuai dengan ketentuan yang telah dibuat, maka data tidak akan disimpan dan admin sistem akan mendapat informasi letak kesalahan pada data yang dimasukkan.

c. *Sequence Diagram* Buat Kunci Jawaban

Sequence diagram buat kunci jawaban dapat dilihat pada Gambar 5.4. Setelah admin soal selesai mengisi *form* untuk membuat kunci jawaban baru, maka obek Route akan mengarahkan masukan admin soal ke KunciJawabanController untuk memastikan apakah kunci jawaban untuk paket soal yang dimasukkan admin soal sudah ada atau belum. Jika kunci jawaban untuk soal tersebut sudah ada, maka admin soal akan menerima pemberitahuan jika kunci jawaban sudah tersedia. Akan tetapi, jika kunci jawaban untuk paket soal tersebut belum tersedia, KunciJawabanController akan mengakses generate-pilihan-jawaban untuk menampilkan pilihan jawaban yang jumlahnya sesuai dengan masukan dari admin soal.

Setelah pilihan jawaban ditampilkan, admin soal dipersilahkan untuk membuat kunci jawaban sesuai dengan paket soal yang dimasukkan.

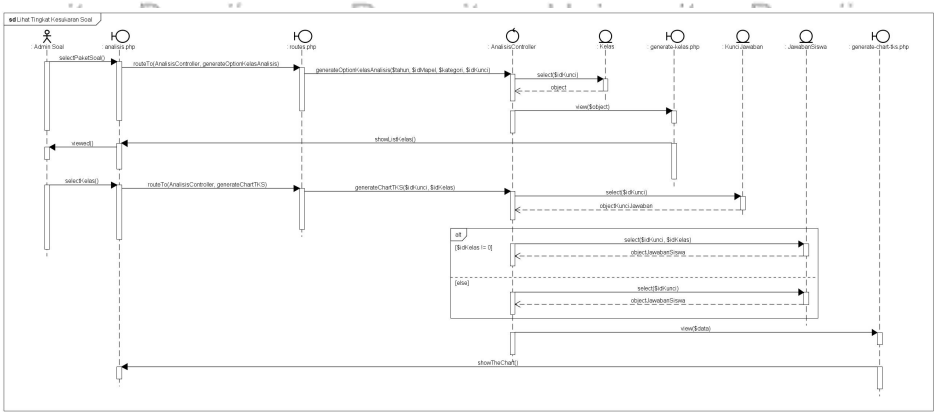
Kunci jawaban akan disimpan dalam sistem dan admin soal akan menerima pemberitahuan bahwa kunci jawaban berhasil disimpan.



Gambar 5.4 *Sequence diagram* buat kunci jawaban

d. *Sequence Diagram* Lihat Tingkat Kesukaran Soal

Sequence diagram lihat tingkat kesukaran soal dapat dilihat pada Gambar 5.5.

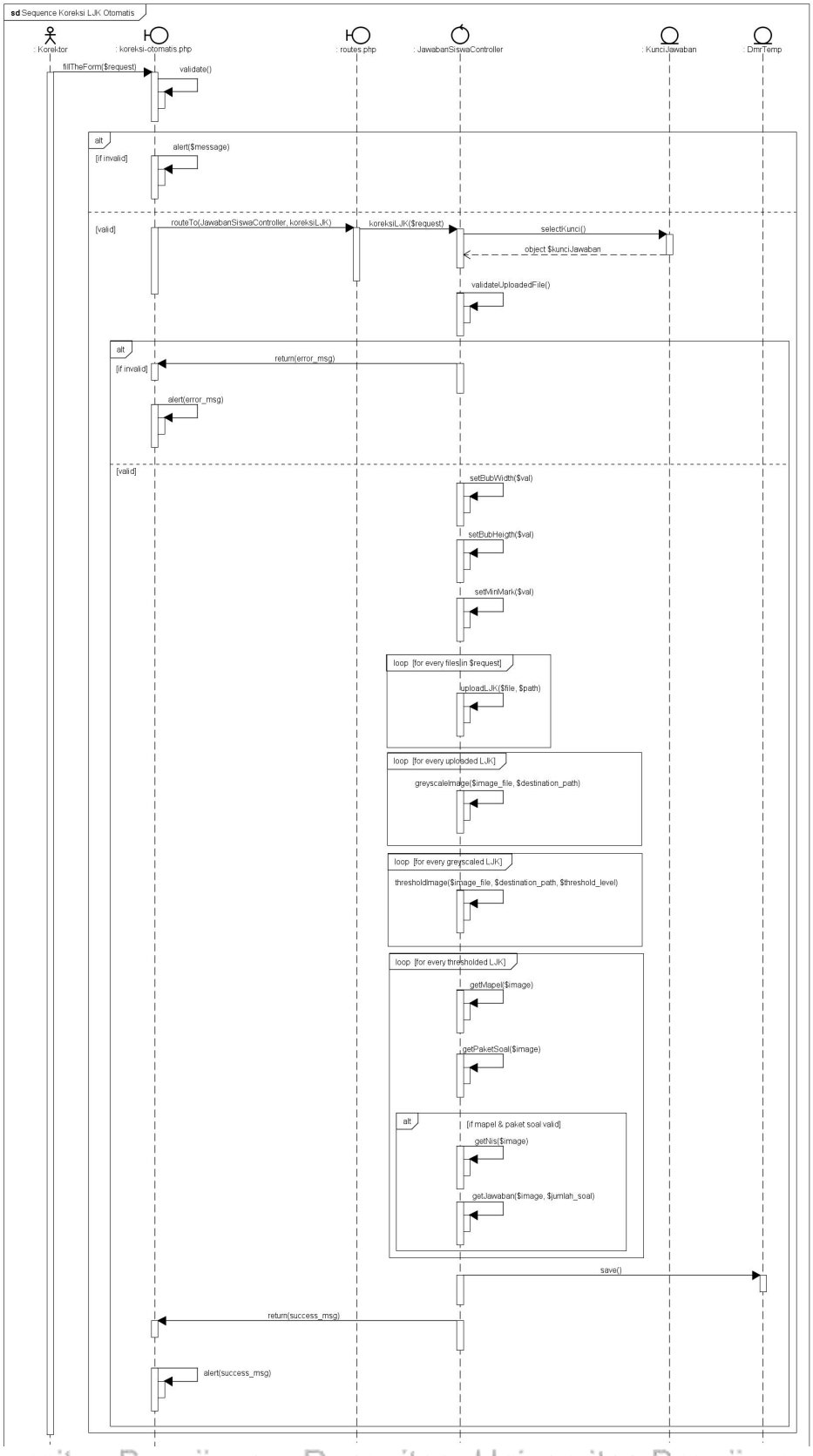


Gambar 5.5 *Sequence diagram* lihat tingkat kesukaran soal

Pada halaman analisis akan ditampilkan paket soal yang telah diujikan. Setelah admin soal memilih salah satu paket soal, maka objek Route akan meneruskan pilihan admin soal ke AnalisisController. Kemudian AnalisisController akan memeriksa, kelas apa saja yang telah mengikuti ujian dengan paket soal tersebut. Setelah pilihan kelas sudah ditampilkan pada admin soal, maka admin soal boleh memilih kelas mana yang ingin dilihat. Setelah admin soal memilih kelas yang diinginkan, maka diagram analisis tingkat kesukaran soal akan ditampilkan.

e. *Sequence Diagram* Koreksi LJK Otomatis

Sequence diagram koreksi LJK otomatis akan diperlihatkan dalam Gambar 5.6.



Gambar 5.6 *Sequence diagram* koreksi LJK otomatis

Fungsi validate() pada koreksi LJK otomatis digunakan untuk memastikan apakah *form* untuk koreksi LJK otomatis telah dilengkapi atau belum. Jika belum dilengkapi, maka korektor akan mendapatkan perintah untuk melengkapi *form* tersebut. Proses akan dilanjutkan apabila *form* sudah terisi lengkap.

Fungsi validateUploadedFile() yang terdapat pada *class* jawabanSiswaController akan digunakan untuk memastikan bahwa berkas-berkas yang dipilih korektor adalah gambar. Jika ada salah satu berkas yang bukan gambar, maka sistem akan memperingatan korektor. Setelah dipastikan bahwa seluruh berkas yang dipilih adalah gambar, maka proses akan dilanjutkan pada tahap pengunggahan gambar ke server.

Setiap gambar yang berhasil diunggah akan dicitrakan menjadi keabu-abuan menggunakan fungsi greyscaleImage(). Kemudian, gambar akan mendapatkan perlakuan berupa *thresholding* dengan menggunakan fungsi thresholdImage(). Setelah gambar LJK mendapat perlakuan *thresholding*, maka langkah selanjutnya adalah mengambil beberapa data seperti mata pelajaran, paket soal, nomor ujian dan jawaban siswa. Setelah itu, data-data tersebut akan disimpan dalam *database* dan korektor akan mendapat pemberitahuan bahwa proses koreksi telah selesai.

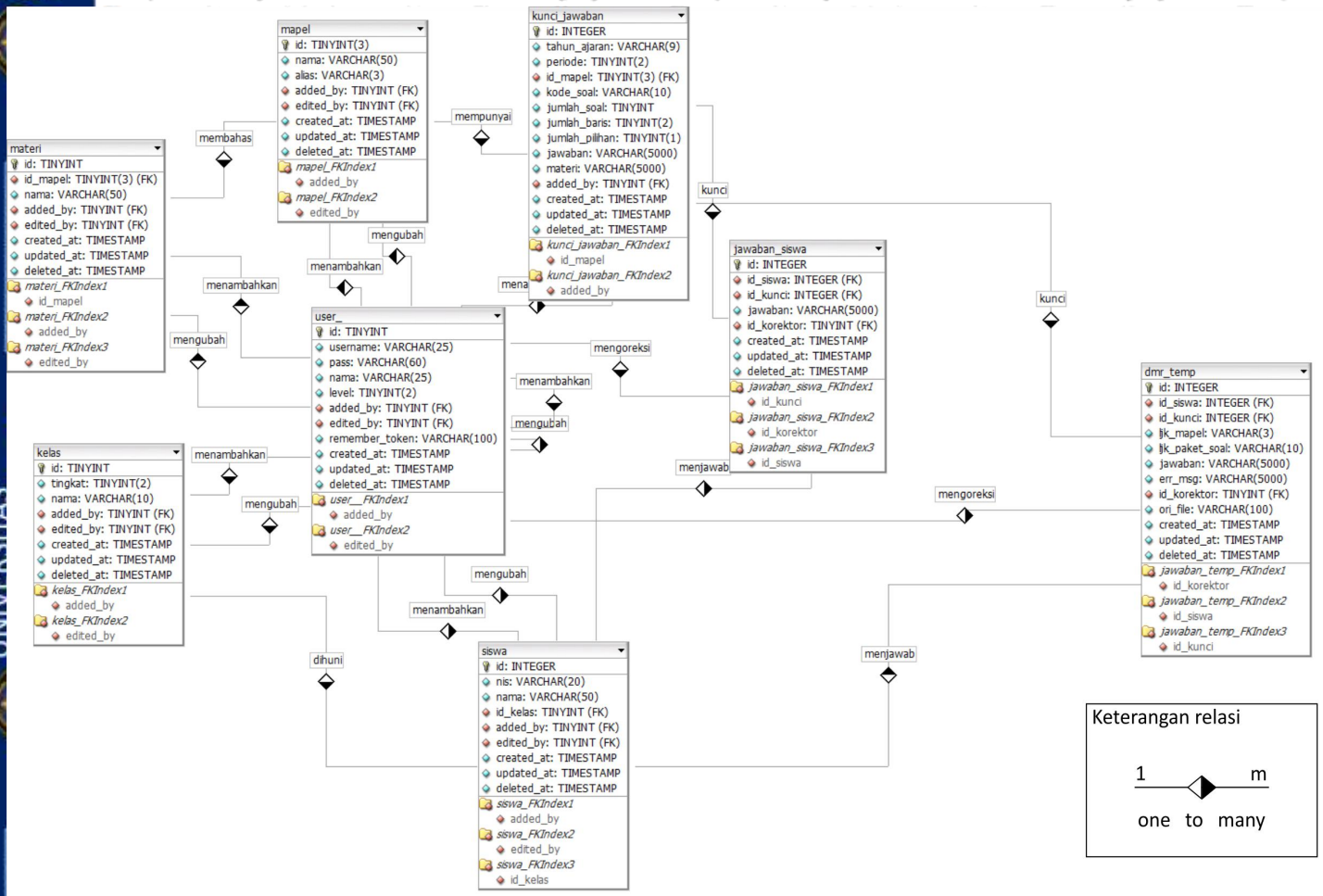
5.3 *Physical Data Model*

Gambar 5.7 merupakan gambaran tabel-tabel serta relasi antar tabel yang akan digunakan sebagai acuan untuk membangun *database* sistem yang dikembangkan dalam penelitian ini.

Tabel-tabel yang akan dibuat guna mendukung kinerja sistem, antara lain : mapel, materi, kelas, user, siswa, kunci_jawaban, jawaban_siswa, dmr_temp.

Tabel mapel akan digunakan untuk menyimpan data-data yang berkaitan dengan mata pelajaran seperti nama mata pelajaran, singkatan dari pelajaran tersebut, orang yang menambahkan mata pelajaran tersebut kedalam sistem, *dll*. Setiap mata pelajaran akan diberikan id yang dapat membedakan antara mata pelajaran satu dengan lainnya. Tabel-tabel lain seperti materi, kelas, user, siswa dan kunci jawaban juga mempunyai fungsi yang identik dengan tabel mapel.

Tabel dmr_temp digunakan untuk menyimpan hasil pembacaan citra LJK. Setelah proses pembacaan citra LJK selesai, maka data yang ada pada tabel dmr_temp akan ditampilkan pada pengguna. Pengguna memiliki pilihan untuk menyimpan, mengubah atau menghapus hasil pembacaan citra LJK. Jika pengguna memutuskan untuk menyimpan hasil tersebut, maka data-data tersebut akan dipindahkan ke tabel jawaban_siswa. Data-data yang terdapat pada tabel jawaban_siswa inilah yang akan digunakan untuk menghitung nilai siswa, tingkat kesukaran soal, *dll*.



Gambar 5.7 Perancangan *database* sistem

5.4 Perancangan Antarmuka

Perancangan antarmuka sangat dibutuhkan oleh pengembang untuk membantu proses implementasi antarmuka sistem. Perancangan antarmuka digambarkan sesuai dengan menu-menu terdapat dalam sistem. Berikut perancangan antarmuka dari sistem yang akan dibangun dalam penelitian ini.

a. Halaman Login

Perancangan antarmuka halaman login dapat dilihat pada Gambar 5.8. Halaman ini memiliki fungsi yang sama seperti pintu gerbang. Jika tamu tidak terdaftar dalam sistem, maka tamu tidak akan diijinkan untuk mengakses sistem. Tamu yang ingin masuk ke sistem harus mengisi *username* dan *password*.

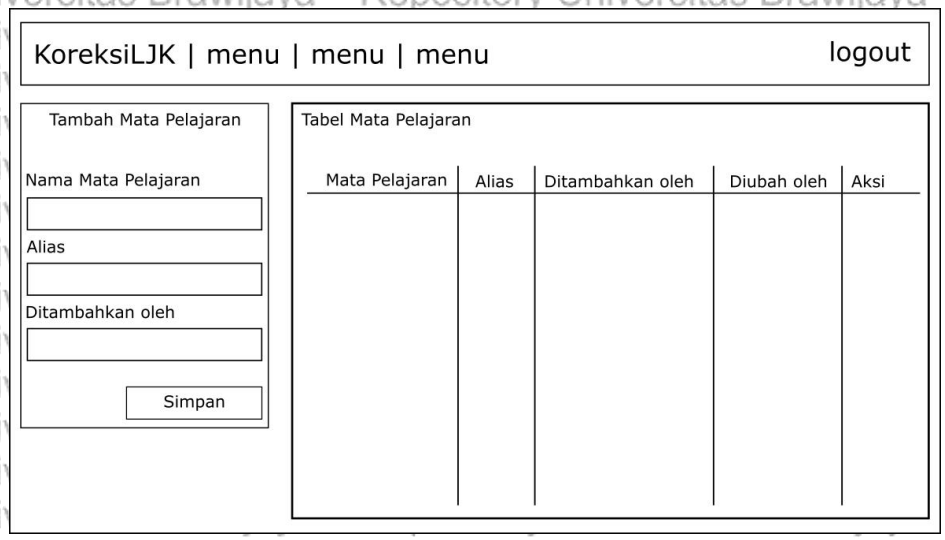
Gambar 5.8 Perancangan halaman login

Form login mempunyai dua kolom yang harus diisi, yaitu kolom *username* yang akan digunakan pengguna untuk memasukkan *username* yang telah mereka daftarkan pada sistem dan kolom *password* yang akan digunakan untuk memasukkan *password* mereka.

Tombol login berfungsi untuk mengirimkan perintah pada sistem untuk memastikan apakah *username* dan *password* yang dimasukkan pengguna telah terdaftar dalam sistem atau tidak.

b. Halaman Mata Pelajaran

Halaman mata pelajaran hanya dapat diakses oleh pengguna yang memiliki peran sebagai admin sistem. Halaman mata pelajaran dibagi menjadi tiga bagian utama seperti yang tergambar pada Gambar 5.9.



Gambar 5.9 Perancangan halaman mata pelajaran

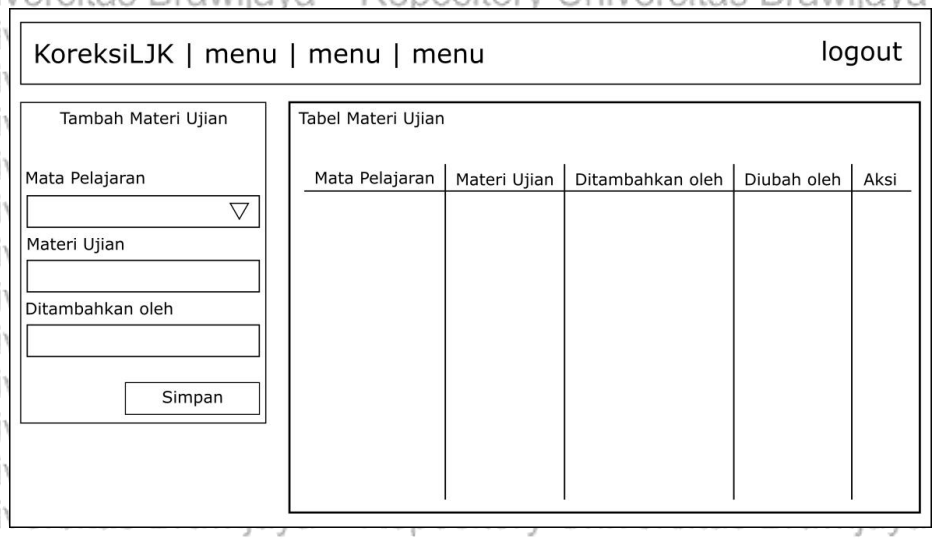
Bagian *header* berisi nama sistem dan menu-menu yang dapat diakses oleh admin sistem serta menu untuk *logout*. Bagian tambah mata pelajaran berisi *form* yang akan digunakan untuk menambahkan data mata pelajaran baru. Sementara bagian tabel mata pelajaran akan berfungsi untuk menampilkan daftar mata pelajaran yang tersimpan dalam sistem.

Kolom ditambahkan oleh yang berada di *form* tambah mata pelajaran akan otomatis terisi dengan nama pengguna yang sedang login saat itu dan kolom ini tidak dapat diubah. Kolom nama mata pelajaran dapat diisi dengan nama mata pelajaran yang akan ditambahkan dalam sistem. Kolom alias hanya dapat diisi dengan singkatan atau akronim dari nama mata pelajaran yang panjangnya tidak boleh lebih dari tiga karakter.

Tombol simpan adalah tombol yang berfungsi untuk memberikan instruksi pada sistem agar melakukan validasi terhadap masukan yang diberikan pengguna, jika masukan yang diberikan pengguna dinyatakan valid maka sistem akan menyimpannya.

c. Halaman Materi Ujian

Sama seperti halaman mata pelajaran, halaman materi ujian hanya dapat diakses oleh pengguna yang memunyai peran sebagai admin sistem. Halamannya pun juga terdiri dari tiga bagian utama yaitu *header* yang berisi menu-menu yang dapat diakses oleh admin sistem, *form* tambah mata pelajaran yang dapat digunakan untuk menambahkan data materi ujian baru serta tabel materi ujian yang bertugas menampilkan daftar materi ujian yang tersimpan dalam sistem. Perancangan antarmuka halaman materi ujian dapat dilihat pada Gambar 5.10.



Gambar 5.10 Perancangan halaman materi ujian

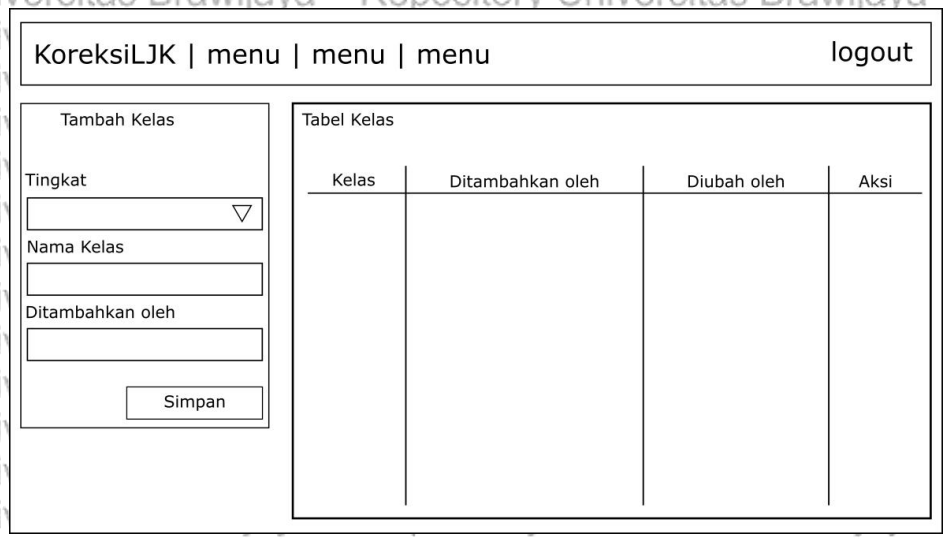
Pada form tambah materi ujian, *field input* yang digunakan untuk mengisi kolom mata pelajaran menggunakan *style dropdown*. Pilihan mata pelajaran yang akan ditampilkan dalam kolom tersebut sesuai dengan daftar mata pelajaran yang tersimpan dalam *database* sistem. Kolom materi ujian dapat diisi dengan nama materi ujian yang diajarkan dalam mata pelajaran yang telah dipilih. Secara otomatis, kolom ditambahkan oleh akan terisi dengan nama pengguna yang sedang login saat itu dan data yang tertulis di dalamnya tidak dapat diubah.

Fungsi dari tombol simpan yang ada dalam halaman ini sama dengan tombol simpan yang ada dalam halaman sebelumnya, yaitu untuk memastikan apakah masukan yang diberikan user telah valid atau belum. Jika masukan yang diberikan telah dinyatakan valid, maka sistem akan menyimpannya.

d. Halaman Kelas

Pengguna yang dapat mengakses halaman ini adalah pengguna yang mempunyai peran sebagai admin sistem. Perancangan antarmuka untuk halaman kelas dapat dilihat pada Gambar 5.11.

Pilihan yang akan ditampilkan dalam kolom tingkat yang berada pada *form* tambah kelas akan disesuaikan dengan tingkatan kelas pada diterapkan pada sekolah menengah tingkat pertama, yaitu tingkat 7 (tujuh), 8 (delapan) dan 9 (sembilan). Kolom nama kelas dapat diisi dengan abjad yang digunakan untuk menamai kelas tersebut, seperti A, B, C, D, *dst*. Kolom ditambahkan oleh adalah kolom yang tidak dapat diubah dan kolom ini secara otomatis akan terisi dengan nama pengguna yang sedang login. Masukan yang telah diberikan dalam *form* tambah kelas akan divalidasi oleh sistem ketika pengguna menekan tombol simpan. Jika masukan yang diberikan valid, maka data tersebut akan disimpan dalam *database* sistem.

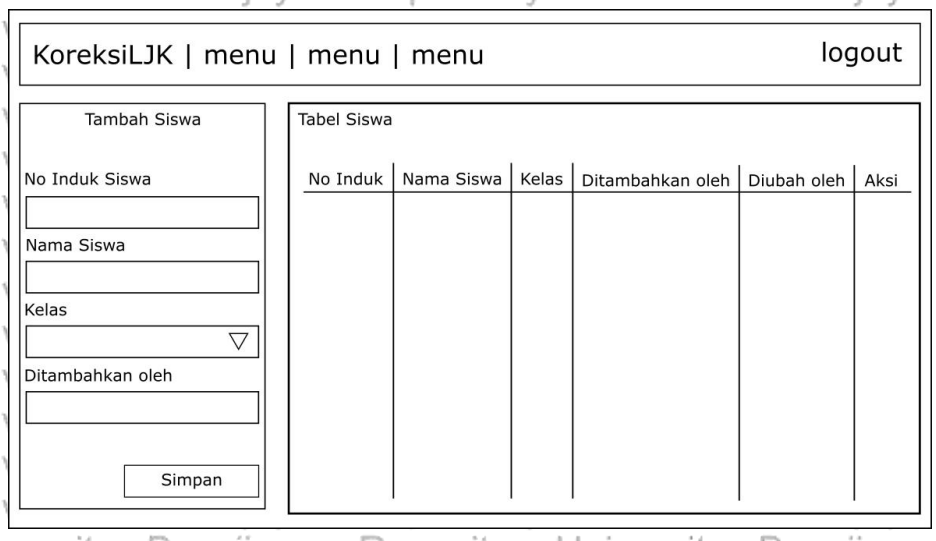


Gambar 5.11 Perancangan halaman kelas

Tabel kelas akan bertugas untuk menampilkan daftar kelas yang tersimpan dalam sistem.

e. Halaman Siswa

Hanya pengguna dengan peran sebagai admin sistem yang dapat mengakses halaman ini. Perancangan antarmuka untuk halaman siswa dapat dilihat pada Gambar 5.12.

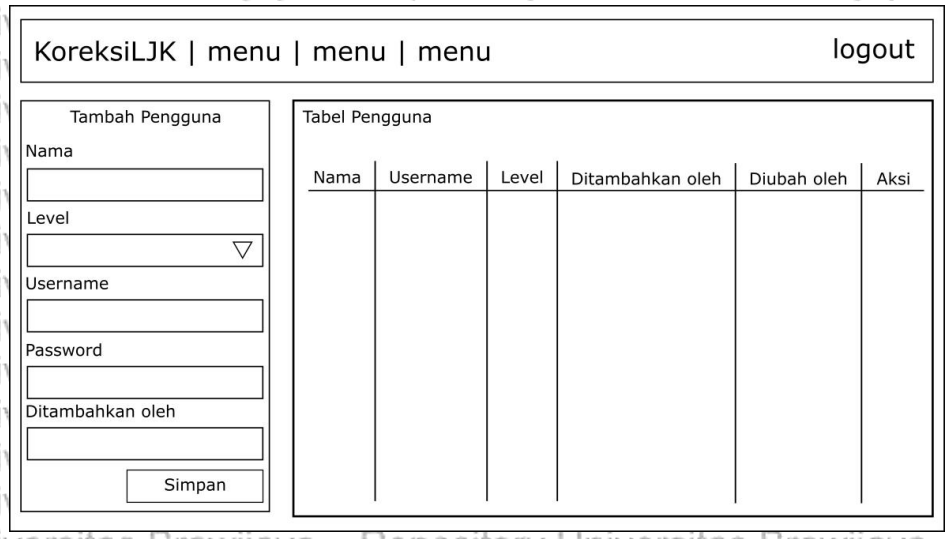


Gambar 5.12 Perancangan halaman siswa

Kolom nomor induk yang terdapat pada *form* tambah siswa dapat diisi dengan nomor induk yang dimiliki oleh siswa. Kolom nama siswa dapat diisi dengan nama lengkap dari siswa yang akan ditambahkan dalam sistem. Pilihan kelas yang akan ditampilkan akan disesuaikan dengan daftar kelas yang tersimpan dalam *database* sistem. Kegunaan kolom ditambahkan oleh dan tombol simpan masih sama seperti yang dijelaskan pada perancangan antarmuka sebelumnya.

f. Halaman Pengguna

Halaman ini hanya dapat diakses oleh pengguna yang berperan sebagai admin sistem. Gambar 5.13 memberikan gambaran rancangan antarmuka yang akan diterapkan untuk membangun halaman pengguna.



Gambar 5.13 Perancangan halaman pengguna

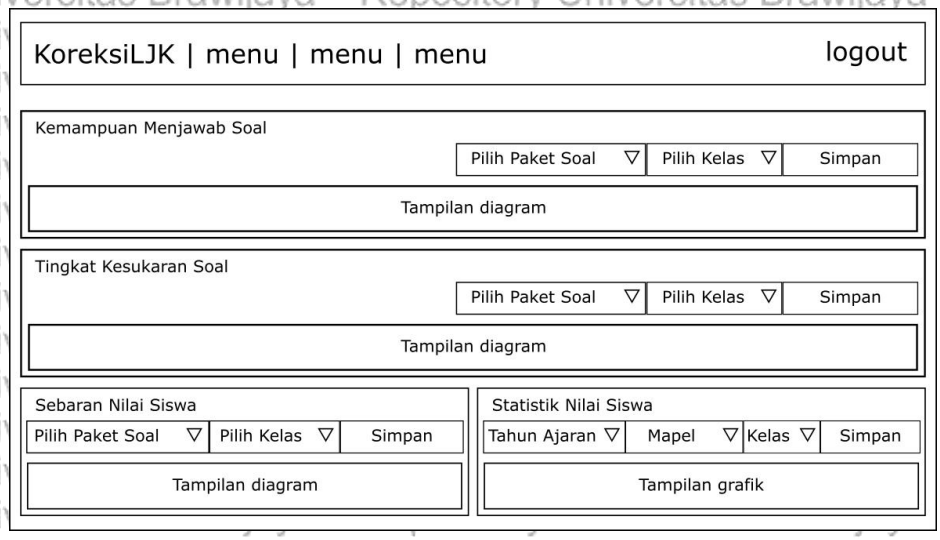
Pilihan level yang ada dalam *form* tambah pengguna akan diisi dengan pilihan admin sistem, admin soal dan korektor. Kolom nama harus diisi dengan nama pengguna yang akan didaftarkan dalam sistem. Kolom *username* dan *password* diisi dengan *username* dan *password* yang akan digunakan pengguna untuk masuk dalam sistem. Kolom ditambahkan oleh akan secara otomatis terisi dengan nama pengguna yang sedang login dan data yang terdapat dalam kolom ini tidak dapat diubah.

Tombol simpan yang terdapat dalam *form* tambah pengguna berfungsi untuk melakukan validasi terhadap masukan yang diberikan pengguna. Jika masukan yang diberikan telah dinyatakan valid oleh sistem, maka masukan tadi akan disimpan dalam *database* sistem.

g. Halaman Analisis

Halaman analisis disediakan khusus hanya untuk pengguna yang memiliki peran sebagai admin soal. Halaman ini akan menampilkan diagram-diagram dan grafik yang menrepresentasikan hasil analisis yang dilakukan terhadap jawaban peserta ujian. Perancangan antarmuka halaman analisis dapat dilihat pada Gambar 5.14.

Setiap fitur analisis dilengkapi dengan tombol simpan yang berfungsi untuk mengonversi diagram atau grafik dalam bentuk gambar, kemudian menyimpannya dalam *storage* pengguna.

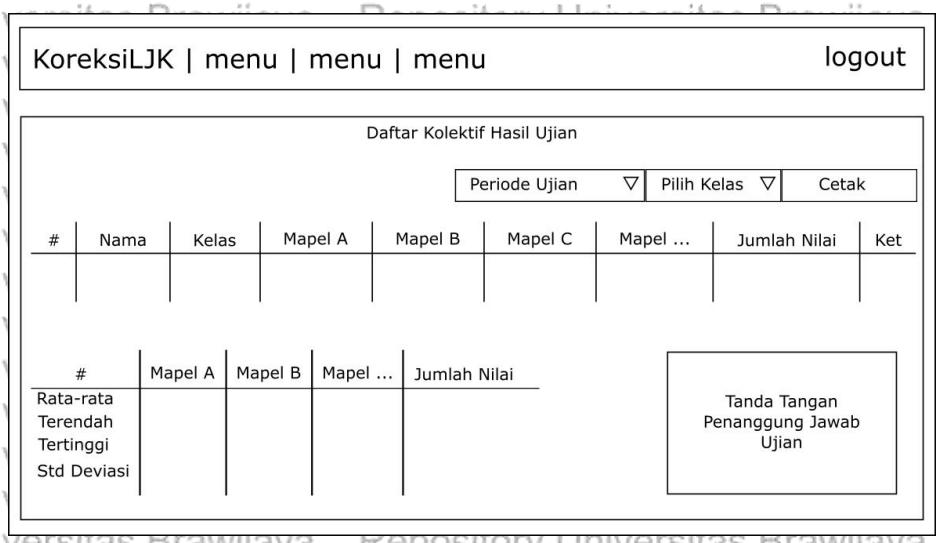


Gambar 5.14 Perancangan halaman analisis

Diagram atau grafik pada tiap-tiap fitur analisis tidak akan ditampilkan sebelum piliham yang ada pada masing-masing fitur diisi dengan benar. *Dropdown* pilih paket soal berisi paket-paket soal yang telah diujikan. *Dropdown* pilih kelas berisi kelas-kelas yang tersimpan dalam sistem. *Dropdown* tahun ajaran berisi tahun ajaran dilaksanakannya ujian.

h. Halaman Hasil Ujian Kolektif

Halaman hasil ujian kolektif dapat diakses oleh setiap pengguna yang memiliki peran sebagai admin soal. Halaman ini akan menampilkan nilai dan statistik untuk seluruh mata pelajaran yang diikuti oleh seluruh siswa dalam periode ujian tertentu. Perancangan antarmuka halaman ini dapat dilihat pada Gambar 5.15.

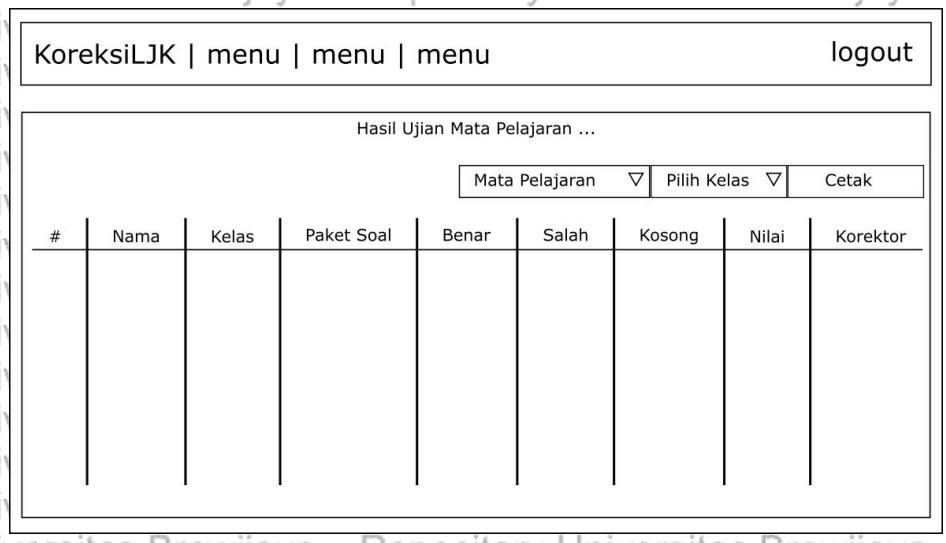


Gambar 5.15 Perancangan halaman hasil ujian kolektif

Halaman ini dilengkapi dengan menu untuk mencetak daftar hasil ujian yang ditampilkan. *Dropdown* periode ujian berisi periode-periode dilaksanakannya ujian. *Dropdown* pilih kelas berisi kelas-kelas yang telah mengikuti ujian.

i. Halaman Hasil Ujian Per Mata Pelajaran

Halaman ini digunakan untuk menampilkan hasil ujian untuk mata pelajaran tertentu pada periode ujian tertentu sesuai dengan pilihan yang ditentukan. Perancangan antarmuka halaman ini dapat dilihat pada Gambar 5.16.



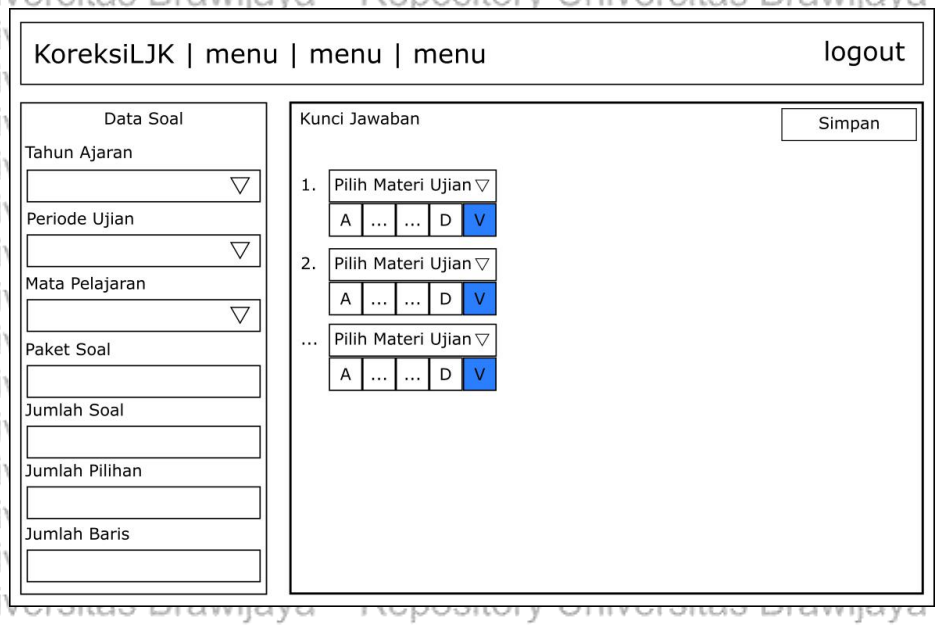
Gambar 5.16 Perancangan halaman hasil ujian per mata pelajaran

Halaman ini dapat diakses oleh pengguna yang mempunyai peran sebagai admin soal atau korektor. Jika admin soal mengakses halaman ini, maka fitur untuk mencetak hasil ujian akan diaktifkan dan penggunapun dapat mencetak hasil ujian. Sementara jika korektor yang mengakses halaman ini, maka fitur untuk mencetak hasil akan ditiadakan. Artinya, ketika korektor mengakses halaman ini, maka ia hanya dapat melihat hasil ujian saja tanpa bisa mencetaknya.

Dropdown mata pelajaran berisi pilihan mata pelajaran yang telah diujikan, sedangkan *dropdown* pilih kelas berisi kelas-kelas yang telah mengikuti ujian untuk mata pelajaran tersebut.

j. Halaman Buat Kunci Jawaban

Perancangan antarmuka halaman buat kunci jawaban dapat dilihat pada Gambar 5.17. Halaman ini hanya dapat diakses oleh pengguna yang memiliki peran sebagai admin soal.



Gambar 5.17 Perancangan halaman buat kunci jawaban

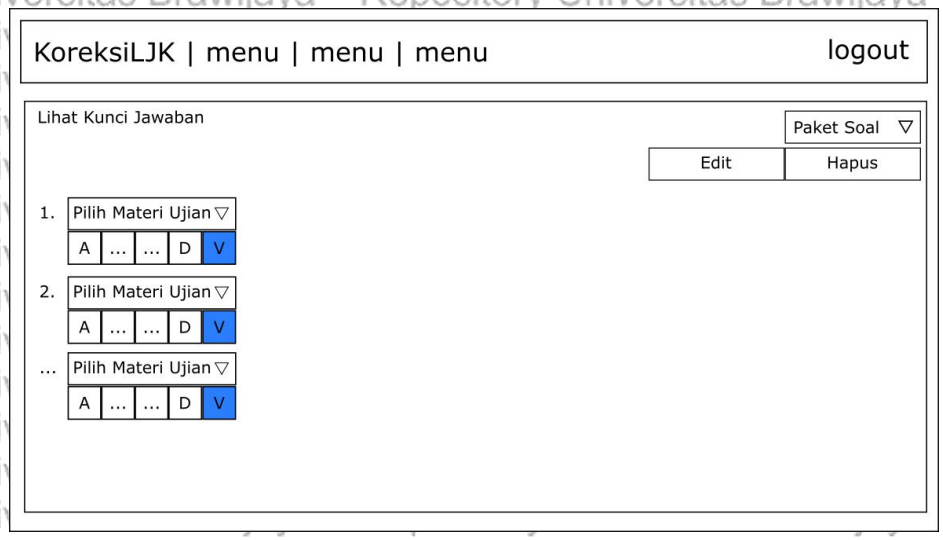
Bagian kunci jawaban akan tetap kosong dan tidak menampilkan apapun sampai semua data yang ada pada *form* data soal terisi dengan benar. Setelah semua kolom pada *form* data soal terisi dengan benar, maka pada bagian kunci jawaban akan ditampilkan tombol-tombol pilihan kunci jawaban sebanyak masukan yang diberikan pada form data soal.

Dropdown tahun ajaran berisi pilihan tahun ajaran dilaksanakannya ujian. *Dropdown* periode ujian berisi pilihan periode dilaksanakannya ujian. *Dropdown* mata pelajaran berisi pilihan mata pelajaran. Kolom paket soal, jumlah soal dan jumlah pilihan dapat diisi sesuai dengan informasi yang tertera pada soal ujian. Kolom jumlah baris diisi sesuai dengan jumlah baris yang terdapat pada blok jawaban di LJK siswa.

k. Halaman Lihat Kunci Jawaban

Halaman lihat kunci jawaban hanya dapat diakses oleh pengguna yang memiliki peran sebagai admin soal. Perancangan antarmuka halaman lihat kunci jawaban dapat dilihat pada Gambar 5.18.

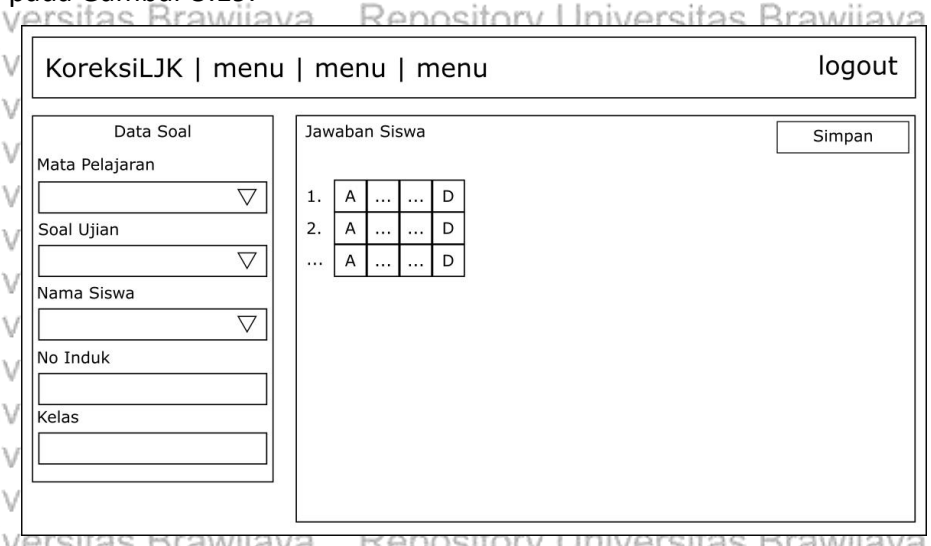
Kunci jawaban tidak akan ditampilkan sebelum pengguna memilih salah satu paket soal yang terdapat dalam *dropdown* paket soal. Kunci jawaban tidak dapat diubah selama pengguna tidak menekan tombol edit. Tombol hapus berfungsi untuk menghapus kunci jawaban dari sistem.



Gambar 5.18 Perancangan halaman lihat kunci jawaban

I. Halaman Koreksi Manual

Halaman ini akan digunakan oleh korektor untuk memasukkan jawaban siswa kedalam sistem dengan cara manual, yaitu memilih jawaban satu per satu sesuai dengan jawaban siswa yang tertera pada LJK. Perancangan antarmuka halaman koreksi manual dapat dlihat pada Gambar 5.19.



Gambar 5.19 Perancangan halaman koreksi manual

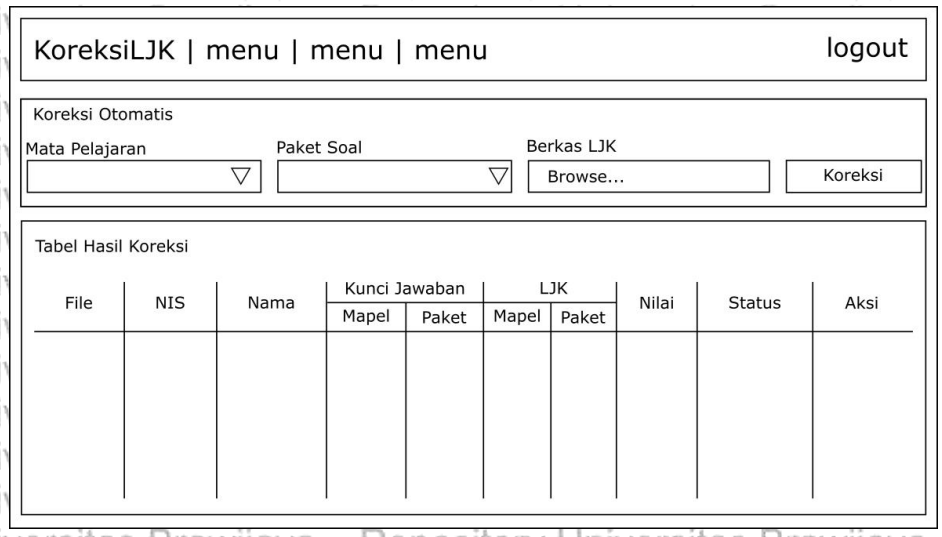
Pilihan jawaban yang terdapat dalam blok jawaban siswa tidak akan tampil sebelum *form* data soal terisi dengan benar.

Dropdown mata pelajaran berisi pilihan mata pelajaran yang telah dibuatkan kunci jawaban. *Dropdown* soal ujian berisi pilihan paket soal yang kunci jawabannya telah disediakan dalam sistem. *Dropdown* nama berisi pilihan nama siswa yang mengikuti ujian. Secara otomatis, nomor induk dan kelas siswa yang terpilih dalam *dropdown* nama akan ditampilkan pada kolom no induk dan kolom kelas. Data yang

ditampilkan dalam kolom-kolom tersebut tidak dapat diubah. Kolom-kolom tersebut dibuat agar korektor dapat memmastikan bahwa data yang ia pilih sesuai dengan yang tertera pada LJK siswa.

m.Halaman Koreksi Otomatis

Perancangan antarmuka halaman koreksi otomatis dapat dilihat pada gambar Gambar 5.20. Halaman ini dapat digunakan oleh korektor untuk memasukkan LJK siswa yang telah dipindai menggunakan mesin pemindai Canon MX390.

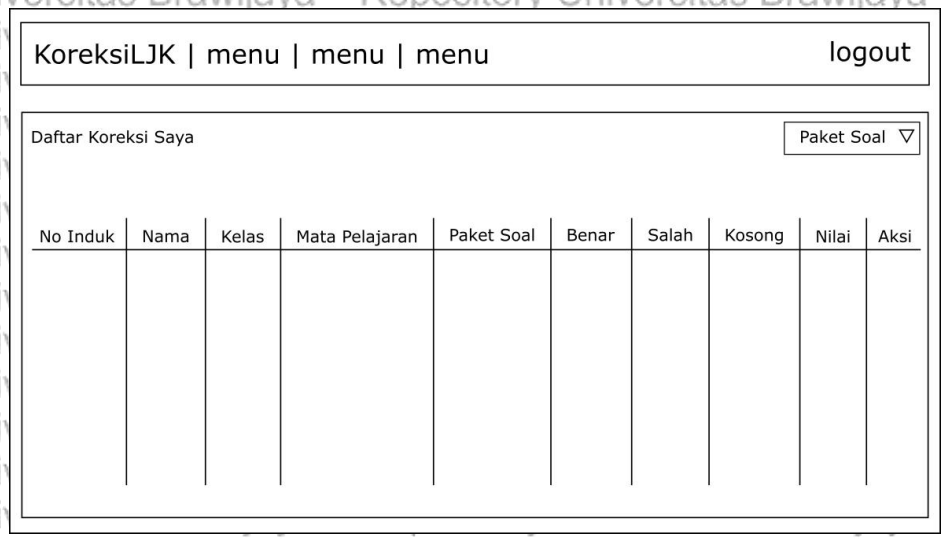


Gambar 5.20 Perancangan halaman koreksi otomatis

Dropdown mata pelajaran dan paket soal digunakan untuk memilih mata pelajaran dan paket soal sesuai dengan informasi yang tertera pada LJK yang akan dikoreksi. Kolom berkas LJK digunakan untuk memilih berkas-berkas LJK yang akan dikoreksi. Setelah tombol koreksi ditekan, maka hasil pindaian akan ditampilkan pada tabel hasil koreksi. Dalam kolom aksi akan terdapat 4 tombol, yaitu lihat LJK, ubah, simpan dan hapus. Tombol lihat LJK digunakan untuk menampilkan gambar LJK yang telah dikoreksi, korektor dapat memilih menu ubah jika korektor merasa masih ada kesalahan dalam proses koreksi yang dilakukan. Jika hasil koreksi dirasa sudah benar, maka korektor dapat menyimpan hasil koreksi dengan cara memilih menu simpan. Tombol hapus dapat digunakan untuk menghapus data hasil koreksi dari sistem, jika data sudah tidak diperlukan lagi.

n. Halaman Lihat Koreksi Saya

Halaman lihat koreksi saya disediakan khusus untuk pengguna dengan peran sebagai korektor. Perancangan antarmuka halaman ini dapat dilihat pada Gambar 5.21.



Gambar 5.21 Perancangan halaman lihat hasil koreksi saya

Setelah korektor memilih salah satu paket soal yang terdapat dalam *dropdown* paket soal, sistem akan menampilkan daftar siswa yang dikoreksi olehnya. Jika korektor merasa ada kesalahan ia lakukan ketika memasukkan jawaban siswa, maka pengguna tersebut dapat memilih menu edit yang berada pada kolom aksi.