

MOBIK – Aplikasi Kuis Generator Berbasis Android

TUGAS AKHIR

Oleh :

Agus Supriyadi 3311111027

Edi Purnomo 3311011003

Disusun untuk memenuhi syarat kelulusan Program Diploma III



PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2013

LEMBAR PENGESAHAN

Batam, 2013

Pembimbing,

Mir'atul Khusna Mufida, S.ST.

NIK. 109057

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Agus Supriyadi

Nama : 3311111027

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

MOBIK – Aplikasi Kuis Generator Berbasis Android

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 9 Juli 2013

Agus Supriyadi
NIM : 3311111027

LEMBAR PERNYATAAN

Dengan ini, saya:

NIM : Edi Purnomo

Nama : 3311011003

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

MOBIK – Aplikasi Kuis Generator Berbasis Android

disusun dengan:

1. Tidak melakukan plagiat terhadap naskah karya orang lain.
2. Tidak melakukan pemalsuan data.
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa izin pemilik.

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 9 Juli 2013

Edi Purnomo
NIM : 3311011003

HALAMAN PERSEMBAHAN

Untuk Orangtua yang telah membesarkan dan meberikan restunya kepada kami,

Untuk seluruh teman-teman kami di Politeknik Negeri Batam,

Untuk seluruh anggota keluarga tercinta di Rumah.

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadirat Allah SWT atas segala rahmat, berkah, karunia dan ridho-Nya sehingga kami dapat menyelesaikan laporan Tugas Akhir yang berjudul “MOBIK – Aplikasi Kuis Generator berbasis Android”.

Dalam kesempatan ini, kami ingin menyampaikan ucapan ribuan terima kasih kepada pihak-pihak yang telah membantu kami dalam menyelesaikan laporan ini, baik dalam memberikan semangat, dukungan, bimbingan, waktu dan biaya, antara lain:

1. Orang tua dan keluarga yang telah memberikan dukungan baik moral maupun materi serta ribuan kata maaf jika dari dulu sampai sekarang sering mengecewakan dan menyia-nyiakan kesempatan yang telah diberikan.
2. Bapak Ari Wibowo selaku Koordinator Tugas Akhir.
3. Ibu Mir'atul Khusna Mufida S.ST. selaku Pembimbing Tugas Akhir ini. Terima kasih atas bimbingan dan motivasi yang ibu berikan dalam penyelesaian laporan ini dan kami juga mohon maaf apabila terdapat kesalahan selama proses bimbingan.
4. Buat sahabat dan teman-teman yang tidak dapat penyusun sebutkan satu per satu yang telah membantu dalam menyelesaikan laporan ini. Terima kasih atas diskusi-diskusi serta kritik dan saran yang telah diberikan.

Penyusun juga mengucapkan terima kasih kepada pihak-pihak lainnya yang sumbernya berupa artikel, tulisan, dan buku yang telah penyusun jadikan referensi guna penyusunan laporan ini, semoga dapat terus berkarya, guna menghasilkan tulisan-tulisan yang mengacu terwujudnya generasi masa depan yang lebih baik.

Penyusun berharap, semoga informasi yang ada dalam laporan ini dapat berguna bagi penyusun khususnya dan bagi para pembaca pada umumnya.

Pepatah mengatakan “Tak ada gading yang tak retak”, penyusun menyadari bahwa laporan ini masih jauh dari sempurna, jikalau banyak kekurangan dan kesalahan disana-sini, kami selaku penyusun laporan sebelumnya memohon maaf.

Penyusun setulus hati menerima kritik dan saran membangun, guna penyempurnaan laporan ini.

Batam, 9 Juli 2013

Penyusun

ABSTRAK

MOBIK – Aplikasi Kuis Generator Berbasis Android

Pengujian kemampuan mahasiswa dalam memahami suatu mata kuliah yang dilakukan oleh dosen adalah hal yang biasa dilakukan dalam proses belajar-mengajar. Dan suatu kuis dalam suatu mata kuliah merupakan rutinitas belajar-mengajar yang kerap dilakukan untuk mengukur tingkat pemahaman mahasiswa dalam mata kuliah yang diajarkan. Pada umumnya kuis dilakukan dengan cara manual (menggunakan lembar kertas soal+lembar kertas jawaban), di sini dosen membagikan soal, memeriksa, menilai, dan mengumumkan hasil kuis mahasiswa secara manual.

Hal ini tentu memakan banyak waktu, tahapan proses, dan *effort* yang dilakukan dosen dan mahasiswa. Untuk mengatasi masalah tersebut, kami mencoba membuat aplikasi *kuis mobile* yang terkomputerisasi. Model proses perangkat lunak yang digunakan yaitu: model proses sekuensial linear atau *Waterfall*. Model ini mengusulkan pendekatan kepada perkembangan perangkat lunak sistematis dan sekuensial dimulai dari tingkat dan kemajuan sistem pada seluruh tahap analisis, desain, kode, dan pengujian.

Perangkat lunak yang dihasilkan adalah Aplikasi MOBIK (= *Mobile*+*Kuis*) yang merupakan aplikasi berbasis *Client-Server*. MOBIK *Client* dibangun dengan sistem Android, menggunakan bahasa pemrograman Java. Perangkat lunak IDE (*Integrated Development Environment*) yang digunakan adalah “App Inventor”. Sedangkan MOBIK *Server* dibangun dengan sistem Web dengan menggunakan *Apache*, bahasa pemrograman yang digunakan adalah PHP, JavaScript, dan MySQL sebagai pendukung basis datanya. Pengujian terhadap perangkat lunak dilakukan dengan metode *blackbox*. Aplikasi ini diharapkan dapat membantu dosen dalam penilaian kuis, dan laporan kinerja mahasiswa berdasarkan nilai yang diperoleh.

Kata Kunci: MOBIK, *Mobile*, Kuis, *Android*, *App Inventor*, MySQL.

ABSTRACT

MOBIK – Quiz Generator – Android Based Application

Testing the ability of students to understand a course conducted by a lecturer is common practice in the teaching-learning process. And a quiz in a course is a teaching and learning routines that are often done to measure the level of understanding of students in the course being taught. In general quiz done by hand (using sheets of papers with the question sheets and the answer sheets), here teachers share questions, examine, assess and announce the results of the quiz to students manually.

It is certainly time consuming stage of the process, and the effort made by lecturer and students. To overcome these problems, we try to make mobile applications a ‘computerized quiz’. Software process models used are: linear or sequential process models Waterfall. This model proposes an approach to software development systematic and sequential start of the level and progress of the system at all stages of analysis, design, code, and testing.

The resulting software is MOBIK Application(=MOBIle+Kuis) which is a Client-Server based applications. MOBIK Client built with the Android system, using the Java programming language. Software IDE (Integrated Development Environment) which is used is “App Inventor”. While MOBIK Server built with Web systems using Apache, programming languages used are PHP, JavaScript, and MySQL as its database support. Testing of the software is done by blackbox methods. This application is expected to assist faculty in assessment quizzes, and report student performance based on the values obtained.

Key words: MOBIK, Mobile, Kuis, Android, App Inventor, MySQL.

DAFTAR ISI

| | | |
|--------|--|----|
| Bab I | Pendahuluan..... | 1 |
| I.1 | Latar Belakang..... | 1 |
| I.2 | Rumusan Masalah..... | 2 |
| I.3 | Batasan Masalah | 3 |
| I.4 | Tujuan | 4 |
| I.5 | Sistematika Penulisan | 4 |
| Bab II | Landasan Teori | 6 |
| II.1 | Rekayasa Perangkat Lunak Berbasis <i>Object Oriented</i> | 6 |
| II.1.1 | Spesifikasi Kebutuhan dan Diagram Use Case | 6 |
| II.1.2 | Diagram Robustness | 8 |
| II.1.3 | Diagram Sequence | 10 |
| II.1.4 | Diagram Kelas Analisis | 13 |
| II.2 | WEB | 16 |
| II.3 | HTML..... | 16 |
| II.4 | PHP..... | 17 |
| II.5 | MySQL | 18 |
| II.6 | JavaScript..... | 18 |
| II.7 | jQuery | 19 |
| II.8 | JSON..... | 19 |
| II.9 | Android..... | 20 |
| II.9.1 | Sejarah Singkat Sistem Operasi Android | 20 |
| II.9.2 | Bahasa Pemrograman Java di Android..... | 22 |
| II.9.3 | Arsitektur Android..... | 23 |
| II.9.4 | Struktur Aplikasi Android | 24 |
| II.9.5 | Versi Android | 24 |
| II.9.6 | Pengembangan Android | 26 |
| II.10 | App Inventor..... | 27 |
| II.11 | Gambaran Umum Pada Pengembang Sebelumnya | 29 |
| II.12 | Rumus Cara Penilaian dan Cara Pengacuannya | 32 |

| | | |
|---------|---|-----|
| II.12.1 | Pengolahan Nilai..... | 32 |
| II.12.2 | Skala (Skala 1-100) | 32 |
| II.12.3 | Rangking/peringkat | 33 |
| Bab III | Analisis dan Perancangan..... | 34 |
| III.1 | Analisis Aplikasi MOBIK | 34 |
| III.1.1 | Deskripsi Detail Aplikasi..... | 34 |
| III.1.2 | Spesifikasi Sistem..... | 35 |
| III.1.3 | Analisis Karakteristik Pengguna | 35 |
| III.1.4 | Analisis Kebutuhan Hardware..... | 36 |
| III.1.5 | Analisis Kebutuhan Perangkat Komunikasi | 36 |
| III.1.6 | Analisis Kebutuhan Software | 36 |
| III.1.7 | Analisis Kebutuhan Fungsional..... | 37 |
| III.2 | Perancangan Aplikasi MOBIK..... | 39 |
| III.2.1 | Diagram Use Case | 39 |
| III.2.2 | Skenario Use Case | 40 |
| III.3 | Analisis Kelas | 49 |
| III.3.1 | Diagram Interaksi | 51 |
| III.3.2 | Diagram Kelas | 101 |
| III.3.3 | Rancangan Kelas Rinci..... | 101 |
| III.3.4 | Perancangan Basis Data..... | 107 |
| III.3.5 | Perancangan Tampilan | 110 |
| Bab IV | Implementasi dan Pengujian..... | 127 |
| IV.1 | Implementasi | 127 |
| IV.1.1 | Fungsi | 127 |
| IV.1.2 | Database..... | 128 |
| IV.1.3 | Implementasi Antarmuka | 133 |
| IV.1.4 | Implementasi Laporan/Report | 147 |
| IV.2 | Pengujian/Testing | 150 |
| Bab V | Kesimpulan dan Saran | 201 |
| V.1 | Kesimpulan..... | 201 |
| V.2 | Saran | 201 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar II-1 Simbol-simbol pada diagram <i>Robustness</i> | 9 |
| Gambar II-2 Contoh Notasi Diagram Sequence | 11 |
| Gambar II-3 Contoh Diagram Kelas | 14 |
| Gambar II-4 Mekanisme pemanggilan aplikasi web bertipe HTML | 16 |
| Gambar II-5 Arsitektur Android | 23 |
| Gambar II-6 Diagram App Inventor | 28 |
| Gambar III-1 Detail Aplikasi | 34 |
| Gambar III-2 Diagram <i>Use Case</i> | 39 |
| Gambar III-3 Analisis Kelas | 49 |
| Gambar III-4 Robustness Diagram – <i>Login</i> | 51 |
| Gambar III-5 Sequence Diagram – <i>Login</i> | 52 |
| Gambar III-6 Robustness Diagram – Logout | 53 |
| Gambar III-7 Sequence Diagram – Logout | 54 |
| Gambar III-8 Robustness Diagram – Ganti Kata Kunci | 54 |
| Gambar III-9 Sequence Diagram – Ganti Kata Kunci | 55 |
| Gambar III-10 Robustness Diagram – Tambah User | 57 |
| Gambar III-11 Robustness Diagram – Ubah User | 58 |
| Gambar III-12 Robustness Diagram – Hapus User | 59 |
| Gambar III-13 Sequence Diagram – Tambah User | 60 |
| Gambar III-14 Sequence Diagram – Ubah User | 61 |
| Gambar III-15 Sequence Diagram – Hapus User | 62 |
| Gambar III-16 Robustness Diagram – Tambah Prodi | 63 |
| Gambar III-17 Robustness Diagram – Ubah Prodi | 64 |
| Gambar III-18 Robustness Diagram – Hapus Prodi | 65 |
| Gambar III-19 Sequence Diagram – Tambah Prodi | 65 |
| Gambar III-20 Sequence Diagram – Ubah Prodi | 66 |
| Gambar III-21 Sequence Diagram – Hapus Prodi | 67 |
| Gambar III-22 Robustness Diagram – Tambah Makul | 68 |
| Gambar III-23 Robustness Diagram – Ubah Makul | 69 |

| | |
|---|----|
| Gambar III-24 Robustness Diagram – Hapus Makul..... | 70 |
| Gambar III-25 Sequence Diagram – Tambah Makul..... | 70 |
| Gambar III-26 Sequence Diagram – Ubah Makul | 71 |
| Gambar III-27 Sequence Diagram – Hapus Makul | 72 |
| Gambar III-28 Robustness Diagram – Tambah Level Pengguna | 73 |
| Gambar III-29 Robustness Diagram – Ubah Level Pengguna..... | 74 |
| Gambar III-30 Robustness Diagram – Hapus Level Pengguna | 75 |
| Gambar III-31 Sequence Diagram – Tambah Level Pengguna | 75 |
| Gambar III-32 Sequence Diagram – Ubah Level Pengguna..... | 76 |
| Gambar III-33 Sequence Diagram – Hapus Level Pengguna | 77 |
| Gambar III-34 Robustness Diagram – Tambah Soal | 78 |
| Gambar III-35 Robustness Diagram – Ubah Soal..... | 79 |
| Gambar III-36 Robustness Diagram – Hapus Soal | 80 |
| Gambar III-37 Sequence Diagram – Tambah Bank Soal | 81 |
| Gambar III-38 Sequence Diagram – Ubah Bank Soal | 82 |
| Gambar III-39 Sequence Diagram – Hapus Bank Soal | 83 |
| Gambar III-40 Robustness Diagram – Setup Quiz | 84 |
| Gambar III-41 Sequence Diagram – Setup Quiz | 85 |
| Gambar III-42 Robustness Diagram – Peserta Kuis | 86 |
| Gambar III-43 Sequence Diagram – Peserta Kuis | 87 |
| Gambar III-44 Robustness Diagram – Cetak Lembar Jawaban..... | 88 |
| Gambar III-45 Sequence Diagram – Cetak Lembar Jawaban per Matakuliah..... | 89 |
| Gambar III-46 Sequence Diagram – Cetak Lembar Jawaban per Mahasiswa..... | 90 |
| Gambar III-47 Robustness Diagram – Cetak Rekap Nilai..... | 91 |
| Gambar III-48 Sequence Diagram – Cetak Rekap Nilai per Lembar Soal | 92 |
| Gambar III-49 Sequence Diagram – Cetak Rekap Nilai per Mata Kuliah | 93 |
| Gambar III-50 Robustness Diagram – Login Kuis | 94 |
| Gambar III-51 Robustness Diagram – Mengikuti Kuis | 95 |
| Gambar III-52 Sequence Diagram – Login Kuis | 96 |
| Gambar III-53 Sequence Diagram – Mengikuti Kuis | 97 |
| Gambar III-54 - Robustness Diagram – Review Kuis | 98 |

| | |
|---|-----|
| Gambar III-55 Sequence Diagram – Review Kuis | 99 |
| Gambar III-56 Robustness Diagram – Review Nilai | 100 |
| Gambar III-57 Sequence Diagram – Review Nilai | 100 |
| Gambar III-58 Diagram Kelas Mobik..... | 101 |
| Gambar III-59 Tampilan Layar Login (Android) | 110 |
| Gambar III-60 Tampilan Layar Login (<i>Web</i>)..... | 111 |
| Gambar III-61 Tampilan Layar Ganti Kata Kunci (Android)..... | 112 |
| Gambar III-62 Tampilan Layar Ganti Kata Kunci (<i>Web</i>)..... | 113 |
| Gambar III-63 Tampilan Layar Menu Admin (<i>Web</i>)..... | 115 |
| Gambar III-64 Tampilan Layar Sub Menu Admin (<i>Web</i>)..... | 116 |
| Gambar III-65 Tampilan Layar Menu Dosen (<i>Web</i>)..... | 117 |
| Gambar III-66 Tampilan Layar Sub Menu Dosen (<i>Web</i>)..... | 118 |
| Gambar III-67 Tampilan Layar Menu Mahasiswa (Android)..... | 120 |
| Gambar III-68 Tampilan Layar Mengikuti Kuis (Android)..... | 122 |
| Gambar III-69 Tampilan Layar Kuis (Android) | 123 |
| Gambar III-70 Tampilan Layar Review Kuis (Android) | 125 |
| Gambar III-71 Tampilan Layar Review Nilai (Android)..... | 126 |
| Gambar IV-1 Antarmuka Login (login.php)..... | 134 |
| Gambar IV-2 Antarmuka Ganti Kata Kunci (gantipass.php) | 134 |
| Gambar IV-3 Antarmuka Halaman Utama (index.php)..... | 135 |
| Gambar IV-4 Antarmuka Halaman Administrasi - Utama | 135 |
| Gambar IV-5 Antarmuka Halaman Administrasi – Form | 136 |
| Gambar IV-6 Antarmuka Halaman Dosen..... | 136 |
| Gambar IV-7 Antarmuka Halaman Dosen - Form..... | 137 |
| Gambar IV-8 Antarmuka Halaman Dosen - Nilai | 137 |
| Gambar IV-9 Antarmuka Halaman Mahasiswa..... | 138 |
| Gambar IV-10 Antarmuka Halaman Profil..... | 138 |
| Gambar IV-11 Mobik icon pada Android..... | 139 |
| Gambar IV-12 Antarmuka Halaman Login – Android..... | 140 |
| Gambar IV-13 Antarmuka Halaman Ganti Kata Kunci – Android | 141 |
| Gambar IV-14 Antarmuka Halaman Utama – Android..... | 142 |

| | |
|---|-----|
| Gambar IV-15 Antarmuka Halaman Ikut Kuis – Android..... | 143 |
| Gambar IV-16 Antarmuka Halaman Kuis – Android..... | 144 |
| Gambar IV-17 Antarmuka Halaman Kirim Jawaban – Android | 145 |
| Gambar IV-18 Antarmuka Halaman <i>Review</i> Nilai – Android..... | 146 |
| Gambar IV-19 Antarmuka Halaman <i>Review</i> Kuis – Android | 147 |
| Gambar IV-20 Laporan Lembar Soal - Soal..... | 148 |
| Gambar IV-21 Laporan PDF Lembar Soal – Kunci Jawaban..... | 148 |
| Gambar IV-22 Laporan PDF Lembar Jawaban/Review Kuis..... | 149 |
| Gambar IV-23 Laporan Excel Nilai..... | 149 |

DAFTAR TABEL

| | |
|--|-----|
| Tabel 1 Notasi Diagram Use Case | 7 |
| Tabel 2 Notasi <i>Robustness Diagram</i> | 10 |
| Tabel 3 Notasi Sequence Diagram..... | 12 |
| Tabel 4 Notasi Diagram Kelas | 15 |
| Tabel 5 Aktor Pengguna Mobik..... | 35 |
| Tabel 6 Kebutuhan Fungsi | 37 |
| Tabel 7 Spesifikasi Analisis Kelas..... | 50 |
| Tabel 8 Daftar tabel dalam basis data yang digunakan aplikasi Mobik..... | 107 |
| Tabel 9 Lokasi fungsi dalam file..... | 127 |
| Tabel 10 Struktur tabel BankSoal | 128 |
| Tabel 11 Struktur tabel HasilKuis..... | 129 |
| Tabel 12 Struktur tabel JenisMK | 129 |
| Tabel 13 Struktur tabel LembarSoal | 129 |
| Tabel 14 Struktur tabel LevelPengguna..... | 130 |
| Tabel 15 Struktur tabel Makul | 130 |
| Tabel 16 Struktur tabel PenggunaMakul..... | 130 |
| Tabel 17 Struktur tabel PenggunaMobik | 130 |
| Tabel 18 Struktur tabel PesertaKuis..... | 131 |
| Tabel 19 Struktur tabel Prodi | 131 |
| Tabel 20 Struktur tabel ProfilPengguna..... | 131 |
| Tabel 21 Struktur tabel vCekHasilJawaban | 131 |
| Tabel 22 Struktur tabel vNilai..... | 132 |
| Tabel 23 Struktur tabel vpm..... | 132 |
| Tabel 24 Struktur tabel vStatusJawaban | 132 |
| Tabel 25 Struktur tabel vStatusPeserta | 132 |
| Tabel 26 Struktur tabel vCekHasilJawaban | 133 |

Bab I Pendahuluan

Seiring dengan berkembangnya teknologi telepon genggam (*handphone*) dan perangkat bergerak (*gadget mobile*), maka kami berusaha untuk dapat menambahkan satu fungsi penggunaan pada perangkat tersebut. Diharapkan dapat dimanfaatkan dalam dunia pendidikan dengan memanfaatkan teknologi informasi dan jaringan komputer yang saling terhubung.

I.1 Latar Belakang

Dalam era digitalisasi saat ini, perkembangan teknologi komunikasi sungguh sangat pesat. Bahkan teknologi komunikasi berperan penting dalam kehidupan manusia. *Handphone* (telepon genggam) yang pada awal ditemukannya hanya untuk telepon dan kirim pesan, namun sekarang *handphone* sudah sangat canggih. Di mana yang dulunya hanya bisa dilakukan komputer sekarang dapat dilakukan oleh *handphone*, yang kita sebut dengan julukan *smartphone* atau telepon pintar.

Perusahaan-perusahaan *Operating System* (OS) besar seperti Apple, BlackBerry, Nokia, Windows dan Google belomba-lomba menciptakan OS yang canggih untuk kebutuhan *smartphone* tersebut. Salah satunya adalah OS Android yang diciptakan serta dikembangkan oleh Google dengan berbasis Linux.

Teknologi Android akhir-akhir ini sudah menjamur dikalangan remaja, baik yang masih sekolah/kuliah maupun yang sudah bekerja. Fenomena ini disebabkan oleh murahnya teknologi Android, sehingga semua kalangan mampu untuk membelinya. Android disukai oleh pengguna karena sifatnya yang terbuka / *open source code* sehingga bisa diubah-ubah sesuai keinginan pengguna, serta tampilannya yang tidak kalah menarik. Bahkan lebih menarik daripada OS yang lain, misal iOS, OS BB, Windows Mobile serta Symbian.

Android bisa dimanfaatkan dalam berbagai bidang, salah satu contoh penggunaannya bisa dipergunakan dalam bidang pendidikan.

Biasanya (dengan cara manual) pada saat ujian dosen memberikan soal di atas kertas dan mahasiswa mengerjakannya di kertas juga. Tentu ini sangat menguras tenaga dan waktu pada saat melakukan pemeriksaan terhadap lembar jawaban dan penilaian. Dosen memeriksa satu persatu lembar jawaban tersebut yang jumlahnya puluhan bahkan hingga ratusan lembar, ini menjadi hal yang sangat membosankan dan bisa saja terjadi kekeliruan dalam pengecekan disebabkan oleh faktor kelelahan. Juga dibutuhkan waktu yang tidak sedikit dalam proses pemeriksaan dan penilaian.

Untuk mengatasi hal-hal tersebut kita bisa memanfaatkan teknologi Android dengan membuat sebuah aplikasi *mobile quiz* yang diberi nama “Mobik” (*Mobile Kuis*). Aplikasi Mobik akan di rancang berbasis aplikasi *client-server*. Semua bank soal dan bank jawaban disimpan di sisi *server*. Sedangkan pada sisi *client*, hanya bisa menerima soal dan mengirimkan jawaban ke *server*.

Dengan sistem ini diharapkan mahasiswa bisa melakukan kuis secara *mobile* dengan menggunakan perangkat Android yang disediakan oleh kampus atau yang dimiliki oleh mahasiswa yang telah terpasang aplikasi Mobik. Aplikasi Mobik diharapkan melakukan pemeriksaan jawaban dan menampilkan nilai kuis mahasiswa secara *real time*, sesaat setelah jawaban dikirimkan atau setelah waktu kuis yang ditentukan terlampaui.

I.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas dapat dirumuskan berbagai permasalahan.

1. Bagaimana membangun suatu aplikasi Android dengan berbasis *client-server*.
2. Bagaimana menambahkan satu aplikasi yang berfungsi untuk melakukan kuis pada *smartphone* untuk proses belajar-mengajar?
3. Bagaimana membuat autotentifikasi pengguna, sehingga setiap pengguna dapat mengakses data berdasarkan kode pengguna yang mereka miliki?

4. Bagaimana memberikan kumpulan pertanyaan kuis kepada mahasiswa dengan mengacak susunan soal pertanyaan dan pilihan jawaban?
5. Bagaimana mengirimkan jawaban kuis dari *client* ke *server*?
6. Bagaimana mereview pertanyaan dan jawaban yang telah diberikan?
7. Bagaimana mereview nilai dengan cepat dan tepat?

I.3 Batasan Masalah

Dikarenakan luasnya cakupan masalah di atas, maka dalam penyusunan dan penulisan proposal tugas akhir ini kami memberikan batasan masalah yang akan dibahas. Batasan masalahnya diharapkan aplikasi yang akan dibuat ini dapat menangani beberapa hal di bawah ini:

1. Aplikasi mobik dapat melakukan autotentifikasi untuk setiap pengguna.
2. Aplikasi mobik menampilkan soal untuk setiap *client* dengan tidak memiliki urutan yang sama (*random*), dan dapat memberikan jumlah soal yang sama kepada setiap *client*.
3. Setelah *client* mengirimkan jawaban-jawaban untuk di nilai, *Aplikasi Server* melakukan pemeriksaan jawaban serta memberikan penilaian secara cepat dan tepat.
4. Menampilkan nilai kepada pengguna di sisi *client*.
5. Pengguna dapat melakukan review terhadap soal dan jawaban yang telah dikerjakan.

Aplikasi ini tidak menangani beberapa hal di bawah ini:

1. Kuis dalam bentuk essay.
2. Memperbaiki atau mengatur kuat lemah sinyal *Wi-fi*.

I.4 Tujuan

Tujuan dari Penelitian dari Tugas Akhir ini adalah:

1. Membangun Aplikasi Mobik, berbasis *client-server*.
2. Dapat menambahkan satu aplikasi tambahan pada perangkat *smartphone Android* yang dapat digunakan untuk proses belajar-mengajar.
3. Aplikasi Mobik dapat melakukan autotentifikasi untuk setiap pengguna.
4. Aplikasi Mobik dapat menyimpan kumpulan soal dan mengirimkan soal-soal yang telah dipilih untuk pengguna yang telah ditentukan.
5. Pengguna di sisi *client* dapat menerima soal mengirimkan jawaban ke *server*.
6. Pengguna dapat me-*review* ulang pertanyaan dan jawaban yang telah di kirim ke *server*.
7. Pengguna dapat melihat nilai untuk setiap kumpulan soal kuis yang telah diberikan.

I.5 Sistematika Penulisan

Sistematika penulisan laporan Tugas Akhir MOBIK adalah sebagai berikut:

BAB I: Pendahuluan

Berisi tentang latar belakang pembuatan tugas akhir, tujuan yang akan dicapai, pembahasan, metodologi dan sistematika pembelajaran.

BAB II: Landasan Teori

Meliputi uraian dari teori-teori yang akan digunakan sebagai dasar pada perencanaan dan pembuatan tugas akhir.

BAB III: Analisis dan Perancangan

Berisi tentang analisis dan perancangan suatu program simulasi.

BAB IV: Implementasi dan pengujian

Membahas tentang implementasi dan pengujian dari perangkat lunak yang telah dibuat serta analisisnya.

BAB V: Kesimpulan dan Saran

Berisi tentang kesimpulan dan saran yang didapatkan selama proses perencanaan dari sistem serta rencana pengembangan dari tugas akhir dimasa yang akan datang.

Daftar Pustaka

Berisi tentang referensi-referensi yang telah digunakan sebagai landasan selama pembuatan tugas akhir.

Bab II Landasan Teori

Pada bab ini berisi tentang teori-teori yang berhubungan dengan Mobile kuis, ujian on-line, program berorientasi objek dan PHP berdasarkan penelitian-penelitian yang sudah ada, serta penelitian yang telah dilakukan.

II.1 Rekayasa Perangkat Lunak Berbasis *Object Oriented*

Dalam aplikasi-aplikasi bisnis, metodologi-metodologi pengembangan aplikasi yang menggunakan bahasa pemrograman terstruktur (*structure programming language*), DFD dan ERD kurang dapat beradaptasi dengan kebutuhan dan harapan pengguna (*user's need and expectations*). Dengan alasan itu, para pakar dibidang perancangan perangkat lunak pada sekitar tahun 1980-1990 mulai bekerja dengan bahasa pemrograman yang berorientasi objek atau lebih dikenal dengan OOP (*Object Oriented Programming*) seperti C++ dan Java. Dengan demikian diperlukan metodologi dan perkakas-perkakas (*tools*) yang lebih sesuai. Dalam hal ini, UML (*Unified Modelling Language*) yang merupakan metodologi kolaborasi antara metode-metode Booch, OMT (*Object Modelling Technique*), serta OOSE (*Object Oriented Software Engineering*) dan beberapa metode lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek (OOP).

II.1.1 Spesifikasi Kebutuhan dan Diagram Use Case

Sebuah sistem/perangkat lunak pada umumnya dikembangkan berdasarkan analisis kebutuhan (*requirements analysis*). Analisis kebutuhan ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analis dan perancang sistem/perangkat lunak untuk berusaha tahu secara pasti mengenai hal-hal yang menjadi kebutuhan dan pengguna sehingga kelak aplikasi yang dibuat memang akan digunakan oleh pengguna (*user*) serta akan memuaskan kebutuhan dan harapannya.

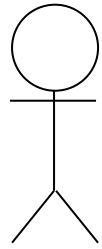
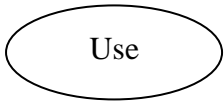
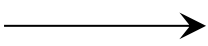
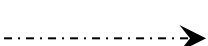

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *Use Case diagram* yang merupakan deskripsi peringkat tinggi bagaimana perangkat lunak atau aplikasi akan digunakan oleh penggunanya. Selanjutnya *Use Case diagram* tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan (*design*), untuk mencari (mencoba menemukan) kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (*testing*).

Use Case diagram menggambarkan fungsi fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

II.1.1.1 Notasi Diagram Use Case

Use case merupakan teknik berdasarkan skenario yang mendeskripsikan model system berorientasi objek, yang mengidentifikasi aktor yang terlibat dalam interaksi dan nama dari tipe interaksi tersebut.

Tabel 1 Notasi Diagram Use Case

| Simbol | Nama | Keterangan |
|---|-----------------------------------|---|
|  | <i>Actor</i> | <i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> . |
|  | <i>Use Case</i> | Aksi yang akan dilakukan <i>actor</i> . |
|  | <i>Generalization Dependensi</i> | Generalisasi dari masing-masing <i>use case</i> . |
|  | <i>Extends/Include Dependensi</i> | Hubungan antara dua <i>use case</i> . |
|  | <i>Association</i> | Menghubungkan antara <i>actor</i> dengan <i>use case</i> . |
| Sistem | <i>System</i> | Lingkungan kerja dari |

| Simbol | | Nama | Keterangan |
|--------|--|------|-----------------------------------|
| | | | perangkat lunak yang akan dibuat. |

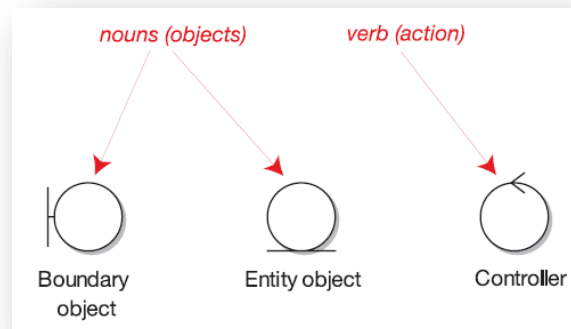
Cara penggunaan masing-masing dependensi: (Fowler, 1999)

1. *Generalization* digunakan ketika menggambarkan sebuah variasi pada sebuah *behaviour* dan ingin digambarkan secara sederhana (*casually*).
2. *Include* digunakan apabila terdapat repetisi pada dua atau lebih *use case* yang terpisah.
3. *Extend* digunakan saat menggambarkan variasi pada *behaviour* dan diharapkan ada bentuk-bentuk pengendalian yang lebih.

II.1.2 Diagram Robustness

Menurut Rosenberg dan Stephens (2007), diagram *Robustness* menjembatani tahap analisis dan desain. Jika tahap analisis (*Use Cases*) dianggap sebagai “apa” dan desain sebagai “bagaimana”, maka analisis *Robustness* merupakan desain sistem yang sangat awal. Pada tahap ini dibuat asumsi awal tentang desain.

Jadi, tahap ini dapat menjadi bagian dari tahap analisis dan dapat menjadi bagian dari tahap desain. Diagram *Robustness* merupakan representasi bergambar dari perilaku (*behavior*) yang dideskripsikan oleh *Use Cases*. Diagram *Robustness* menunjukkan perilaku dari kelas-kelas dan perilaku dari perangkat lunak. Pada diagram ini tidak digambarkan kelas mana yang bertanggung jawab terhadap perilaku tertentu. Walaupun demikian, diagram *Robustness* dapat dibaca seperti diagram aktivitas (*activity diagram*) atau sebagai sebuah *flowchart* dalam arti suatu objek “berbicara” dengan objek lainnya. Simbol-simbol yang terlibat dalam diagram *Robustness* ditunjukkan oleh gambar di bawah ini :



Gambar II-1 Simbol-simbol pada diagram *Robustness*¹

Boundary object adalah antarmuka antara sistem dengan segala sesuatu di luar sistem. Contohnya adalah layar atau halaman web.

Entity object merupakan kelas-kelas dari domain model. *Controller* adalah penghubung antara *boundary* dan *entity object*. Aturan-aturan dalam menggambar diagram *Robustness* adalah (Rosenberg & Stephens 2007):

1. Objek (*noun*) dapat berbicara kepada *controller* (*verb*) dan sebaliknya.
2. Objek tidak dapat berbicara kepada objek lainnya.

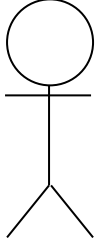
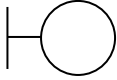


Controller dapat berbicara kepada *controller* lainnya.

II.1.2.1 Notasi Robustness Diagram

Ide dasarnya adalah bahwa untuk menganalisa langkah-langkah dari *use case* untuk memvalidasi logika bisnis di dalamnya dan untuk memastikan bahwa terminology yang konsisten dengan kasus penggunaan lainnya yang sebelumnya dianalisis. Penggunaan lainnya adalah untuk mengidentifikasi objek potensial atau tanggung jawab objek untuk mendukung logika yang disebut dalam *use case*, efektif sebagai jembatan untuk antar diagram lainnya.

¹ Use Case Driven Object Modeling with UML, Theory and Practice – Resenberg, halaman 103

Tabel 2 Notasi *Robustness Diagram*

| Simbol | Nama | Keterangan |
|---|-----------------|---|
|  | <i>Actor</i> | <i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan input atau memberikan output, maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> . |
|  | <i>Boundary</i> | <i>Boundary</i> terletak di antara sistem dengan dunia sekelilingnya. Semua form, laporan-laporan, antar muka ke perangkat keras seperti printer atau scanner dan antar muka ke sistem lainnya adalah termasuk dalam kategori ini. |
|  | <i>Control</i> | <i>Control</i> berhubungan dengan fungsionalitas seperti pemanfaatan sumber daya, pemrosesan terdistribusi, atau penanganan kesalahan. |
|  | <i>Entity</i> | <i>Entity</i> digunakan menangani informasi yang mungkin akan disimpan secara permanen. <i>Entity</i> bisa juga merupakan sebuah tabel pada struktur basis data. |

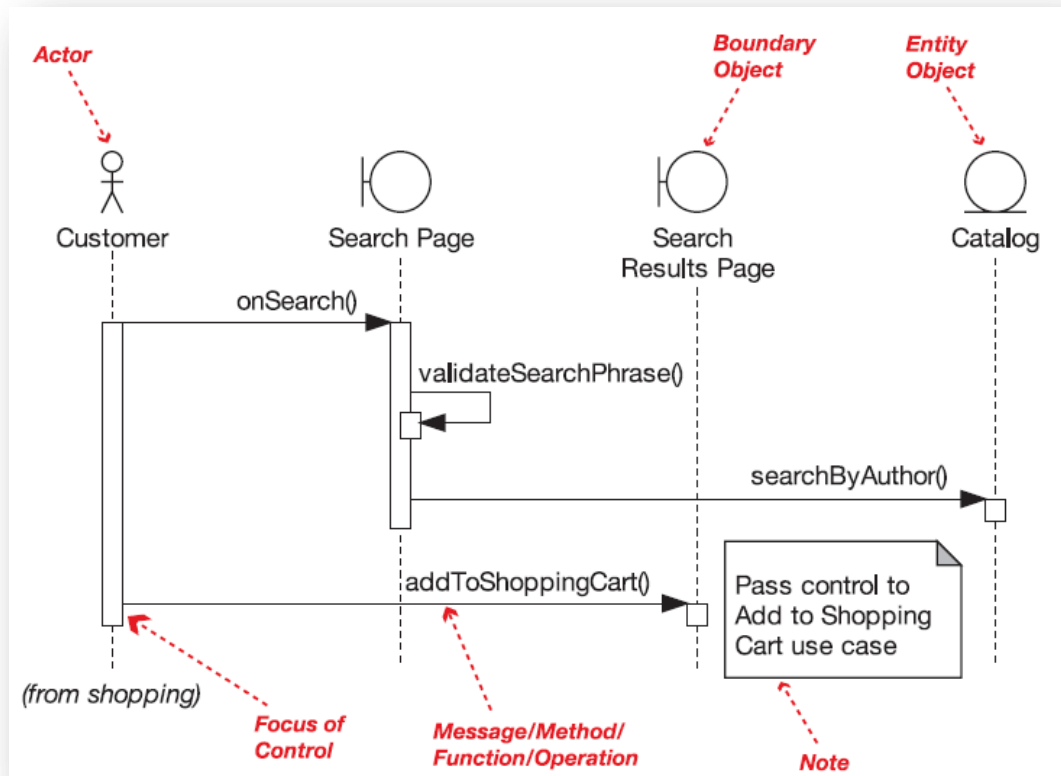
II.1.3 Diagram Sequence

Diagram *Sequence* adalah sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah sistem. Interaksi tersebut berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram terdiri dari dimensi horizontal (objek-objek) dan dimensi vertikal (waktu).

Karakteristik diagram *Sequence* :

- a) Menggambarkan alur kejadian sebuah aktivitas.

- b) Lebih detail dalam menggambarkan aliran data, termasuk data atau *behaviour* yang dikirimkan atau diterima.
- c) Kurang mampu menjelaskan detail dari sebuah algoritma (*loop*, *branching*).



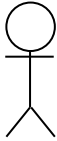
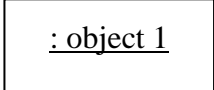


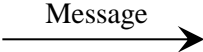
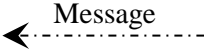

Gambar II-2 Contoh Notasi Diagram Sequence²

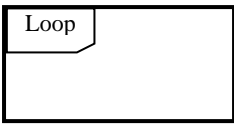
II.1.3.1 Notasi Sequence Diagram

Diagram *Sequence* atau disebut juga diagram interaksi. Jelas bahwa diagram ini digunakan untuk menggambarkan beberapa jenis interaksi antara unsur-unsur yang berbeda dalam model. Jadi interaksi ini adalah bagian dari perilaku dinamis dari sistem.

² Use Case Driven Object Modeling with UML, Theory and Practice – Resenberg, halaman 186

Tabel 3 Notasi Sequence Diagram

| Simbol | Nama | Keterangan |
|---|----------------------------|--|
|  | <i>Actor</i> | <i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. |
|  | <i>Object</i> (Partisipan) | <i>Object</i> atau biasa juga disebut partisipan merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma. |
|  | <i>Lifeline</i> | <i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> . |
|  | <i>Activation</i> | <i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah objek yang akan melakukan sebuah aksi. |
|  | <i>Message</i> | <i>Message</i> , digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> . |
|  | <i>Return Message</i> | <i>Return Message</i> , digambarkan dengan anak panah horizontal dan garis putus-putus antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi balik antara <i>object-object</i> . |
|  | <i>Self-Message</i> | <i>Self-message</i> atau panggilan mandiri mengindikasikan komunikasi kembali ke dalam sebuah <i>object</i> itu sendiri. |

| Simbol | Nama | Keterangan |
|---|-------------|---|
|  | <i>Loop</i> | Operator <i>loop</i> adalah <i>fragmen</i> yang dapat mengeksekusi berulang kali dan penjaga menunjukkan dasar iterasi. |

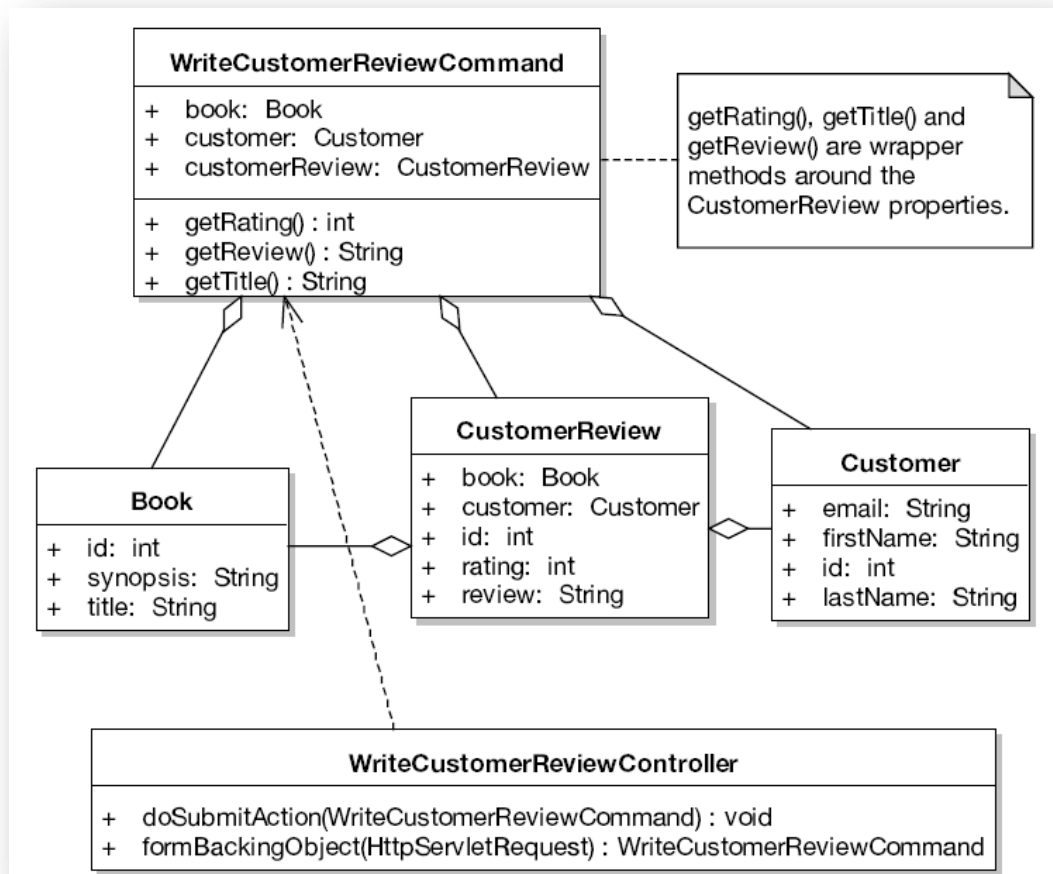
II.1.4 Diagram Kelas Analisis

Diagram *class* sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain.

Komponen-komponen diagram kelas analisis antara lain :

- a) Objek adalah abstraksi dari sebuah entitas nyata atau tidak nyata yang informasinya harus diingat atau disimpan.
- b) *Class* adalah deskripsi lebih dari satu atau lebih objek dengan sejumlah atribut dan layanan yang sama termasuk deskripsi tentang cara membuat objek dari kelas tersebut.
- c) Atribut adalah *variable* data, yang dapat memberikan informasi keadaan di mana tiap objek dari suatu kelas mempunya nilai tersendiri.
- d) Metoda adalah prosedur atau fungsi yang menjadi perilaku kelas dan objek dan menjadi tanggung jawab objek tersebut.



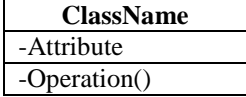
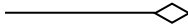
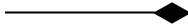
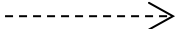
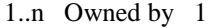
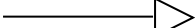
Gambar II-3 Contoh Diagram Kelas³

II.1.4.1 Notasi Diagram Kelas

Diagram Kelas merupakan sebuah statik diagram. Ini merepresentasikan tampilan yang tetap dari sebuah aplikasi. Diagram kelas tidak hanya untuk memvisualisasikan, menggambarkan dan mendokumentasikan berbagai aspek dari sistem, tetapi juga untuk membangun kode *excuteble* dari aplikasi perangkat lunak.

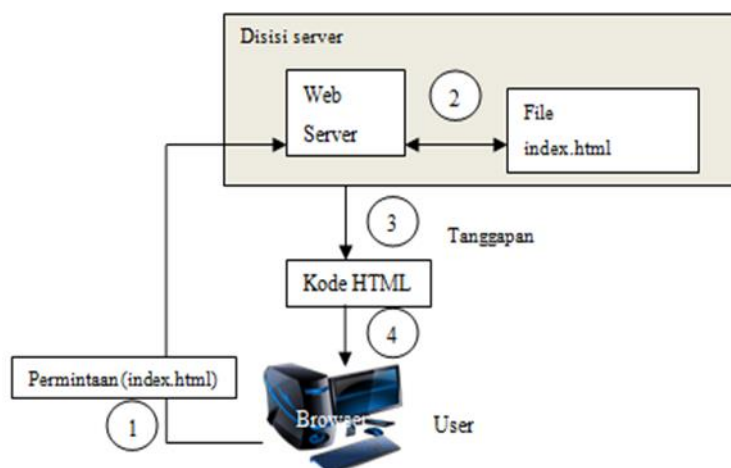
³ Use Case Driven Object Modeling with UML, Theory and Practice – Resenberg, halaman 248

Tabel 4 Notasi Diagram Kelas

| Simbol | Nama | Keterangan |
|---|---------------------------------------|--|
|  | <i>Class</i> | <i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi objek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan method-method/operasi dari sebuah <i>class</i> . |
|  | <i>The Aggregation (containing)</i> | <i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi “mempunyai sebuah” atau “bagian dari”. Sebuah <i>aggregation</i> digambarkan sebagai sebuah garis dengan sebuah jajaran genjang yang tidak berisi/tidak solid. |
|  | <i>The Aggregation (composition)</i> | Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid. |
|  | <i>The Aggregation (dependency)</i> | Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik. |
|  | <i>The Relationship (association)</i> | Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> , dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> (Contoh: One-to-one, one-to-many, many-to-many). |
|  | <i>Generalization</i> | Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi objek. Sebuah <i>generalization</i> dilambangkan dengan sebuah panah dengan kepala panah yang tidak solid yang mengarah ke kelas “parent”-nya/induknya. |

II.2 WEB

Secara terminologi, web adalah kumpulan dari halaman-halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web (WWW)* di dalam internet. Sebuah halaman web biasanya berupa dokumen yang ditulis dalam format *HTML (Hyper Text Markup Language)*, yang selalu bisa diakses melalui *HTTP*, yaitu sebuah protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*.



Gambar II-4 Mekanisme pemanggilan aplikasi web bertipe HTML

Pada gambar di atas langkah pertama menunjukkan saat user melakukan permintaan terhadap sebuah halaman web. Langkah kedua menggambarkan sewaktu web server mengambil file. Langkah ketiga menunjukkan saat web server mengirimkan kode *HTML* kepada user yang meminta. Sedangkan langkah keempat menggambarkan ketika browser melakukan penerjemahan kode *HTML* ke dalam bentuk tampilan pada layar.

II.3 HTML

HTML adalah kependekan dari *Hyper Text Markup Language*. Fasilitas *hypertext* merupakan metode yang menautkan (*link*) satu dokumen ke dokumen lain melalui suatu teks. *HTML* merupakan halaman yang berada pada suatu situs internet atau web. Jadi, suatu situs terdiri atas beberapa halaman *HTML*, atau

webpage. Semakin menarik halaman *web*-nya, semakin sering *website* itu dikunjungi. Misalnya dengan menampilkan gambar-gambar yang menarik, suara-suara, animasi huruf atau multimedia.

Sebenarnya, *HTML* merupakan kumpulan kode-kode yang ditampilkan dalam format tag untuk menampilkan dalam bentuk *web*. Biasanya *hyperlink* ada pada halaman *web* yang jika diklik oleh pengguna, *browser* akan menampilkan dokumen yang tertaut dengan *link* tersebut. (Lia, 2006).

II.4 PHP

PHP *Hypertext Preprocessor* atau *Personal Home Page Hyper Text Processor* yang biasa di sebut PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. PHP banyak dipakai untuk memrogram situs web dinamis.

PHP adalah bahasa *scripting* yang menyatu dengan *HTML* dan dijalankan pada *serverside*. Artinya semua sintaks yang diberikan akan sepenuhnya dijalankan pada server, sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. Ketika seorang pengguna internet membuka suatu situs yang menggunakan fasilitas *server side scripting* PHP, maka terlebih dahulu server bersangkutan akan memproses semua perintah PHP di server lalu mengirimkan hasilnya dalam format *HTML* ke *web server* pengguna internet tadi. Sehingga kode asli yang di tulis PHP tidak terlihat di browser pengguna.

PHP tidak bisa bekerja dan digunakan secara mandiri, diperlukan beberapa perangkat lunak tambahan yang harus di *install*. Adapun perangkat lunak pendukung yang harus di *install* antara lain :

1. *Apache Server*

Merupakan web server yang digunakan oleh PHP, berfungsi menampilkan hasil prose script PHP ke komputer browser dalam bentuk tag *HTML*.

2. *MySQL*

Merupakan database server yang paling sering digunakan dalam pemrograman PHP.

Berfungsi menyimpan data dalam database dan memanipulasi data-data yang diperlukan.

3. *PHPMysqlAdmin*

Merupakan kakas untuk pengelolaan database yang berbasis web. *PHPMysqlAdmin* bukan merupakan suatu keharusan, manipulasi data yang bisa juga digantikan dengan kakas yang lain, misalnya *MySQL* console (berbasis teks). Namun dengan *PHPMysqlAdmin* pengelolaan atau manipulasi database menjadi lebih mudah.

II.5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (bahasa inggris : *database management system*) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

II.6 JavaScript

Javascript adalah bahasa *script* (bahas pemrograman yang dapat memegang kontrol aplikasi) yang berbasis pada bahasa pemrograman java, namun *javascript* bukanlah bagian teknologi *java* dari *sun*. *Java* merupakan bahasa pemrograman berorientasi objek murni sedangkan *javascript* digunakan secara prosedural. Cara kerja *javascript* adalah mengakses elemen pada *HTML* dan membuat aksi jika elemen-elemen itu mengalami perubahan, misalnya berubahnya warna halaman web begitu sebuah tombol di klik. *Javascript* membuat sebuah halaman *web* menjadi lebih dinamis. *Javascript* yang digunakan pada halaman *web* merupakan *client side scripting* yang berarti bahwa *web browser* mengidentifikasi dan menjalankan *script* program yang disisipkan dalam dokumen *web* (yang diterima oleh *server*), dan mungkin memperbaharui tampilan halaman di komputer pemakai (*user*) tanpa mengirimkan permintaan (*request*) baru kepada *web server*. Karena *javascript* pada halaman *web* adalah *client side scripting* maka

kode *javascript* yang ditulis satu file dengan dokumen *HTML* akan ditampilkan ketika sebuah halaman *web* dilihat *source code*-nya.

Javascript merupakan bahasa yang *case sensitive* seperti halnya bahasa pemrograman *java* yaitu membedakan penulisan dengan huruf kecil dan huruf besar memiliki arti yang berbeda. (Shalahuddin, 2010).

II.7 jQuery

jQuery adalah *multi-browser JavaScript library* yang dirancang untuk menyederhanakan *script* sisi klien dari HTML. Dirilis pada bulan Januari 2006 di BarCamp NYC oleh John Resig. Saat ini dikembangkan oleh tim pengembang yang dipimpin oleh Dave Methvin. Digunakan oleh lebih dari 65% dari 10.000 situs yang paling dikunjungi, jQuery adalah JavaScript *library* yang paling populer yang digunakan saat ini.

jQuery adalah, *free open source software*, berlisensi di bawah Lisensi MIT. sintaks jQuery ini dirancang untuk memudahkan untuk menavigasi dokumen, pilih DOM elemen, membuat animasi, menangani *event*, dan mengembangkan aplikasi Ajax. jQuery juga menyediakan kemampuan bagi pengembang untuk membuat *plug-in* di atas perpustakaan JavaScript. Hal ini memungkinkan pengembang untuk membuat abstraksi untuk interaksi tingkat rendah dan animasi, efek canggih dan tingkat tinggi, widget dengan tema. Pendekatan modular ke perpustakaan jQuery memungkinkan penciptaan kuat halaman web dinamis dan aplikasi web.

Himpunan fitur inti jQuery - pilihan elemen DOM, traversal dan manipulasi -, diaktifkan oleh mesin pemilihnya (bernama "Sizzle" dari v1.3), menciptakan "gaya pemrograman" baru, algoritma dan DOM-data-struktur, dan dipengaruhi arsitektur lain kerangka Javascript seperti YUI v3 dan Dojo .

II.8 JSON

JSON (dilafalkan "Jason"), singkatan dari *JavaScript Object Notation* (bahasa Indonesia: notasi objek JavaScript), adalah suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan terbaca-manusia serta

digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. Aplikasi utamanya adalah pada pemrograman aplikasi web AJAX dengan berperan sebagai alternatif terhadap penggunaan tradisional format XML.

Walaupun JSON didasarkan pada subset bahasa pemrograman JavaScript (secara spesifik, edisi ketiga standar ECMA-262, Desember 1999) dan umumnya digunakan dengan bahasa tersebut, JSON dianggap sebagai format data yang tak tergantung pada suatu bahasa. Kode untuk pengolahan dan pembuatan data JSON telah tersedia untuk banyak jenis bahasa pemrograman. Situs json.org menyediakan daftar komprehensif pengikatan JSON yang tersedia, disusun menurut bahasa.

Format JSON dispesifikasikan di RFC 4627 oleh Douglas Crockford. Tipe media Internet resmi JSON adalah `application/json` sedangkan ekstensi berkasnya adalah “.json”.

II.9 Android

II.9.1 Sejarah Singkat Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel (*smartphone*). Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorium dari 34 perusahaan peranti keras, piranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis perdana Android (5 November 2007), Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *Open Source* pada perangkat *mobile*. Dilain pihak, Google merilis kode-kode Android di bawah

lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusi tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). Sekitar bulan September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasi. Telepon seluler ini diproduksi oleh HTC Cooration dan tersedia di pasaran pada 5 Januari 2008.

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Co dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana merek Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6.

Sejak Android dirilis telah dilakukan berbagai pembaharuan berupa perbaikan *bug* dan penambahan fitur baru. Pada masa saat ini, kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android. Vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus, dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi Android. Banyak vendor yang memproduksi *smartphone* berbasis Android karena Android adalah sistem operasi yang *Open Source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android juga menjadi pesaing utama dari Apple pada sistem operasi Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan di atas adalah karena Android merupakan *platform* yang sangat lengkap baik dari sistem operasinya, aplikasi dan

tools pengembangan, *market* aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia.

II.9.2 Bahasa Pemrograman Java di Android

Java merupakan perangkat lunak produksi **Sun Microsystem Inc.** untuk pemrograman beberapa tujuan (*multi puose*), *multi platform* dapat berjalan di beberapa sistem operasi, mudah dipelajari dan *powerfull*. Aplikasi-aplikasi yang dapat dibuat dengan java meliputi *web programming* (pemrograman web), *desktop programming* (pemrograman desktop), *mobile programming* (pemrograman *handphone*).

Perangkat lunak java sintaknya (tulisanannya) mirip dengan bahasa pemrograman C karena bahasa Java memakai bahasa pemrograman C, tetapi bahasa Java menyempurnakan kekurangan bahasa pemrograman C. Saat pertama kali rilis, Java disebut JDK (*Java Development Kit*), hingga JDK Versi 2 atau dikenal dengan Java 2, dibagi menjadi 3 edisi, yaitu J2SE (*Java 2 Standard Edition*), J2EE (*Java 2 Enterise Edition*) dan J2ME (*Java 2 Micro Edition*).

J2SE merupakan edisi atau teknologi untuk pemrograman *desktop* atau aplikasi layar (*console*). J2SE juga merupakan perangkat lunak dasar yang harus dipasang (*install*) sebelum memakai J2EE dan J2ME. J2EE merupakan edisi atau teknologi untuk pemrograman *enterise*, seperti pemrograman *database*, JSP, Beans dan lain-lainnya. J2ME merupakan edisi atau teknologi untuk pemrograman *mobile/handphone* atau peralatan kecil (*small device*). Setelah Java teasang, maka di dalam sistem komputer terdapat JVM (*Java Virtual Machine*). Di dalam JVM tersebut terdapat JRE (*Java Runtime Environment*). Program yang diketik dengan bahasa Java memiliki ekstensi **.java**, akan menghasilkan *file .class* jika dikompilasi. *File .class* dapat dijalankan dengan memanfaatkan JRE.

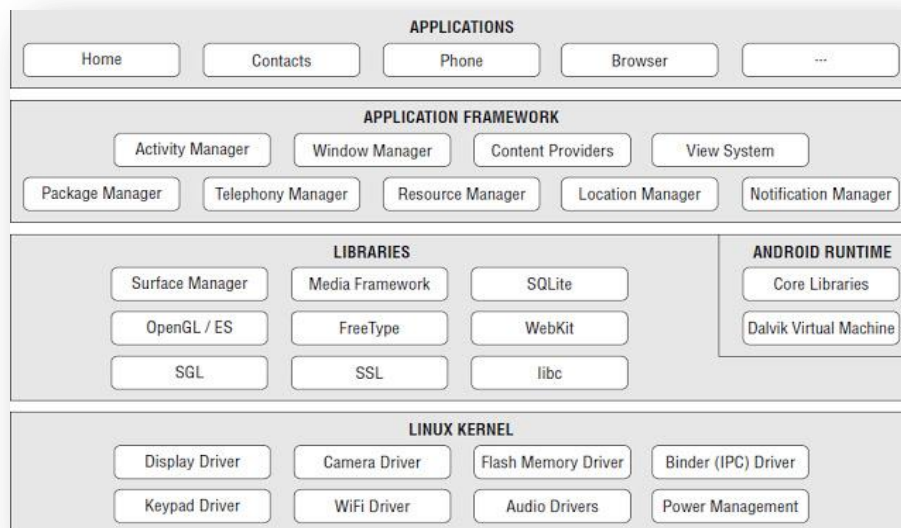
Android hanya menyediakan lingkungan *runtime* sebagai *intereter*, di mana kode sumber yang telah *dcompile* dengan *compiler* Java akan dioptimasi

oleh Delvik, yaitu sebuah *virtual machine* yang memang dibuat khusus untuk menjalankan kode-kode program yang dibuat dengan bahasa pemrograman Java. Kemudian oleh *dex tools* (merupakan bagian dari DVM) mengubah Java *Class* yang telah *dcompile* oleh *compiler* Java ke lingkungan *native* yang berbentuk *.dex format* (Dalvik executable), yang teroptimasi untuk lingkungan perangkat keras dengan komputasi yang rendah.

II.9.3 Arsitektur Android

Secara garis besar arsitektur Android terdiri atas *Applications*, *Applications Frameworks*, *Libraries*, *Android Run Time*, dan *Linux Kernel*.

- a. *Applications* merupakan *layer* (lapis), di mana *user* berhubungan dengan aplikasi saja.



Gambar II-5 Arsitektur Android⁴

- b. *Applications Frameworks* merupakan *Open Development Platform* yang ditawarkan Android untuk dapat dikembangkan guna membangun aplikasi. Pengembang memiliki akses penuh menuju *API Frameworks* seperti yang

⁴ ANDROID – Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, halaman 9.

dilakukan oleh aplikasi kategori inti. Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.

- c. *Libraries* merupakan *layer*, di mana fitur-fitur Android berada.
- d. *Android Run Time* merupakan *layer* yang membuat aplikasi Android dapat dijalankan, di mana dalam prosesnya menggunakan implementasi Linux.
- e. *Linux Kernel* merupakan *layer* inti dari sistem operasi Android berada.

II.9.4 Struktur Aplikasi Android

Struktur aplikasi Android atau Fundamental Aplikasi ditulis dalam bahasa pemrograman Java. Kode Java ini dikompilasi bersama dengan *file resource* yang dibutuhkan oleh aplikasi, di mana prosesnya *dipackage* oleh *tools* yang disebut "*apt-tools*" ke dalam paket Android, sehingga menghasilkan *file* dengan ekstensi **.apk**. *File .apk* ini yang disebut dengan aplikasi, dan nantinya dapat dijalankan pada *device/peralatan mobile*.

Ada 4 komponen pada aplikasi Android, antara lain :

- a. *Activities* merupakan komponen untuk menyajikan *user Tampilan* (tampilan program) kepada pengguna.
- b. *Service* merupakan komponen yang tidak memiliki *user Tampilan* (tampilan program), tetapi *service* berjalan secara *backgrounds*.
- c. *Broadcast Receiver* merupakan komponen yang berfungsi menerima dan beraksi untuk menyimpan notifikasi.
- d. *Contact Provider* merupakan komponen untuk membuat aplikasi data secara spesifik sehingga dapat digunakan oleh aplikasi lain.

II.9.5 Versi Android

Banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versinya, fitur yang digunakan pun semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi Android

adalah HTC Dream yang dirilis pada tanggal 22 Oktober 2008. Beberapa uraian versi Android antara lain :

1. **Android Versi 1.1** pertama kali dirilis oleh Google pada tanggal 9 Maret 2009. Versi ini memiliki fitur-fitur : jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail dan pemberitahuan *e-mail*.
2. **Android Versi 1.5** (Cupcake) merupakan versi yang dirilis oleh Google pada pertengahan Mei 2009. Fitur-fitur yang dimiliki versi ini adalah merekam dan menonton video dengan modus kamera, *upload* video ke Youtube dan gambar Picasa langsung dari telepon, *bluetooth* A2DP, terhubung langsung ke headset *bluetooth*, animasi layar, *keyboard* pada layar yang disesuaikan sistem.
3. **Android Versi 1.6** (Donut) merupakan versi yang dirilis pada September 2009 dengan fitur-fitur : proses pencarian yang lebih baik dibandingkan versi sebelumnya, penggunaan baterai indikator dan *control applet* VPN, galeri memilih foto yang dihapus, kamera, *camcorder*, CDMA/EVDo, 802.1x, Gestures, *text-to-speech engine*, dan *dial contact*.
4. **Android Versi 2.0/2.1** (Eclair) merupakan versi yang dirilis Google pada tanggal 3 Desember 2009, memiliki fitur-fitur : mengoptimalkan *hardware* versi sebelumnya, Google Maps 3.1.2, perubahan *user Tampilan*, *browser* baru, HTML5, daftar kontak baru, *flash* untuk kamera 3,2MP, *digital zoom*, *bluetooth* 2.1.
5. **Android Versi 2.2** (Froyo : Frozen Yoghurt) merupakan versi yang dirilis oleh Google pada Mei 2010. Versi pertama (Rev 1) memiliki fitur-fitur : dapat menghapus komponen, DVM dioptimalkan, *graphic* 2D dan 3D, *SQLite*, Media Audio dan Video berbagai *format*, GSM, *bluetooth*, EDGE, 3G, WiFi, *camera*, GPS (*Global Positioning System*), kompas, *accelerometer* (tergantung *hardware*).
6. **Android Versi 2.3** (Gingerbread) merupakan versi yang dirilis oleh Google pada Desember 2010, memiliki fitur-fitur : *Sip-based* VoIP, NFC (*Near Field*

Communications), *Gyroscope* dan *Sensor*, *Multiple Cameras Support*, *Mixable Audio Effects* dan *Download Manager*.

7. **Android Versi 3.0/Versi 3.1** (HoneyComb) merupakan versi yang dirilis oleh Google khusus untuk PC Tablet. Beberapa *smartphone* tidak dapat menggunakan karena versi ini memang banyak dipakai bukan untuk *voice call*. Memiliki fitur-fitur : aksesoris API terbuka, *USB host API*, *keyboard external* dan perangkat penunjuk, *joystick* dan *gamepads*, WiFi, perluasan daftar terbaru Apps, layar *home widget resizeable*.
8. **Android Versi 4.0** (ICS: Ice Cream Sandwich) diumumkan pada tanggal 19 Oktober 2011. Android versi 4.0 ini membawa fitur Honeycomb untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, teadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline* dan berbagai informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.
9. **Android Versi 4.1 – 4.2** (Jelly Bean) diluncurkan pada acara Google I/O yang membawa sejumlah keunggulan dan fitur baru. Penambahan baru tersebut diantaranya meningkatkan *input keyboard*, *desain* baru fitur pencarian, *user Tampilan* yang baru dan pencarian melalui *voice search* yang lebih cepat. Selain itu terdapat pembaharuan pada *Google Now*. *Google Now* memberikan informasi yang tepat pada waktu yang tepat juga. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas ataupun hasil pertandingan olahraga. Sistem operasi Android Jelly Bean 4.1 muncul pertama kali dalam produk tablet Asus, yaitu Google Nexus 7.

II.9.6 Pengembangan Android

Android adalah sistem operasi *mobile* berbasis Linux. Awalnya Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel pintar/*smartphone*. Beberapa hal yang harus kita kerjakan sebelum memulai untuk membuat aplikasi Android sebagai berikut:

1. Java SE

Install Java SE versi 1.5 atau 1.6 ke atas untuk dapat mengkompilasi Android.

2. Andoird SDK

Android SDK adalah tools API (*Application Programing Tampilans*) Yang di perlukan untuk mengembangkan aplikasi pada *platform* Android.

3. App Inventor

App Inventor Merupakan IDE yang akan kita gunakan untuk coding aplikasi Android.

II.10 App Inventor

App Inventor adalah sebuah tool untuk membuat aplikasi android. *App Inventor* berbasis visual block programming, jadi kita bisa membuat aplikasi tanpa satupun kode. Mengapa disebut *visual block programming?*, karena kita akan melihat, menggunakan, menyusun dan drag-drops “blok” yang merupakan simbol-simbol perintah dan fungsi *–event handler* tertentu dalam membuat aplikasi, dan secara sederhana kita bisa menyebutnya tanpa menuliskan kode program *–coding less*.

Kita tidak harus memiliki *basic programmer*, mengerti kode-kode atau berkecimpung dalam dunia teknologi informasi untuk membuat aplikasi dengan *App Inventor*. Bahkan *App Inventor* tidak hanya untuk membuat aplikasi, karena bisa digunakan untuk mengasah logika anda, seperti halnya menyusun sebuah puzzle. Untuk *programmer* tentu ada opsi-opsi *advance* untuk membuatnya sesuai dengan level kita.



Gambar II-6 Diagram App Inventor

Framework visual programming ini terkait dengan bahasa pemrograman *Scratch* dari MIT (Massachusetts Institute of Technology), yang secara spesifik merupakan implementasi dari *Open Block* yang didistribusikan oleh *MIT Scheller Teacher Education Program* yang diambil dari riset yang dilakukan oleh *Ricarose Roque*. *App Inventor* menggunakan *Kawa Language Framework* dan *Kawa's dialect* – yg di develop oleh *Per Bothner* dan di distribusikan sebagai bagian dari GNU Operating System oleh *Free Software Foundation* sebagai *Compiler* yang mentranslate *visual block programming* untuk diimplementasikan pada platform Android. Tentu disini anda tinggal menikmati kemudahan yang ada.

App Inventor Saat ini bisa di akses melalui <http://beta.appinventor.mit.edu/> versi beta dari MIT, harus online ketika membuat aplikasi.

II.11 Gambaran Umum Pada Pengembang Sebelumnya

Di Politeknik Negeri Batam, sudah banyak Mahasiswa yang membahas Tugas Akhirnya tentang sistem ujian online, diantaranya Muhammad Tarmizi (2005) Aplikasi Tes Online, Karmila (2008) Aplikasi Tes UMPB Berbasis Komputer, Muhammad Irfan Fahmi (2008) Aplikasi Simulasi Ujian SPMB, Januardi dan Jonathan RK (2010) Aplikasi Ujian Online, Lies Handayani & Ferri Martin (2011) Pengembangan Sistem Ujian Online Berbasis Objek.

Deskripsi umum pada Aplikasi Tes Online yang dibuat oleh Muhammad Tarmizi (2005) terdiri dari:

1. Aplikasi mampu menampilkan soal tes secara acak untuk tiap *user*.
2. Aplikasi mampu menyimpan hasil tes peserta.
3. Aplikasi mampu menghitung nilai tes peserta.

Dari deskripsi umum Aplikasi Tes Online yang dibuat oleh Muhammad Tarmizi masih memiliki kekurangan, seperti:

1. Pemilihan level pada soal yang masih statis.
2. Pengaturan soal ujian yang belum teratur menurut level.
3. Penyajian nilai ujian yang masih statis.

Deskripsi umum pada Aplikasi Tes UMPB Berbasis Komputer yang dibuat oleh Karmila (2008) terdiri dari:

1. Aplikasi dapat menampilkan soal-soal ujian secara acak.
2. Aplikasi dapat menentukan jumlah soal yang akan dikerjakan oleh peserta.
3. Aplikasi dapat menghitung nilai dari hasil tes UMPB secara cepat.
4. Aplikasi dapat melihat peserta-peserta yang diterima dan ditolak pada program studi tertentu berdasarkan hasil tes.

5. Aplikasi dapat menampilkan report/laporan hasil tes UMPB.
6. Aplikasi dapat melakukan pengelolaan data soal yang terdiri dari insert, update dan delete soal.
7. Aplikasi dapat melakukan penambahan mata pelajaran.
8. Aplikasi mampu mengelola data soal bertipe suara dan gambar.
9. Aplikasi dapat melakukan pengelolaan data peserta yang terdiri dari insert, update dan delete peserta ujian.
10. Aplikasi dapat melakukan pencarian *user* berdasarkan nama, *id_user*, dan program studi.

Dari deskripsi umum Aplikasi Tes UMPB Berbasis Komputer yang dibuat oleh Karmila masih memiliki kekurangan, seperti:

1. Tidak memiliki bank soal.
2. Penyediaan level masih statis.
3. Pengaturan nilai ujian yang masih statis.

Deskripsi umum pada Aplikasi Simulasi Ujian SPMB yang dibuat oleh Muhammad Irfan Fahmi (2008) terdiri dari:

1. Aplikasi dapat berjalan secara client-server.
2. Aplikasi dapat menampilkan soal secara acak.
3. Aplikasi dapat memberikan laporan kelulusan.

Dari deskripsi umum Aplikasi Simulasi Ujian SPMB yang dibuat oleh Muhammad Irfan Fahmi masih memiliki kekurangan, seperti:

1. Aplikasi tidak dapat memberikan informasi peringkat.
2. Aplikasi tidak dapat memberikan pembahasan jawaban.

Deskripsi umum pada Aplikasi Ujian Online yang dibuat oleh Januardi dan Jonathan RK (2010) terdiri dari:

1. Aplikasi dapat membuat mata pelajaran sesuai kebutuhan.

2. Aplikasi dapat memberikan informasi hasil jawaban dan kunci jawaban dari soal-soal yang telah dikerjakan.
3. Aplikasi dapat membuat level dan nilai yang dapat diubah sesuai kebutuhan.
4. Aplikasi dapat melakukan pengaturan soal yang akan ditampilkan.
5. Aplikasi dapat memberikan informasi peringkat.
6. Aplikasi dapat membuat aturan waktu pelaksanaan ujian.
7. Aplikasi dapat menampilkan soal ujian dengan pembagian level yang ditentukan.

Dari deskripsi umum Aplikasi Ujian Online yang dibuat oleh Januardi dan Jonathan RK masih memiliki kekurangan, seperti:

1. Tidak dapat membuat random pada pilihan jawaban.
2. Tidak dapat menangani registrasi peserta ujian yang dilakukan secara online.
3. Tidak menangani tempat pemberian checklist pada soal ujian yang digunakan sebagai tanda jika peserta ujian merasa ragu-ragu pada jawaban soal tersebut.

Deskripsi umum pada Aplikasi Pengembangan Sistem Ujian Online Berbasis Objek (2011) oleh Lies Handayani & Ferri Martin terdiri dari:

1. Aplikasi mampu mengelola informasi profil organisasi.
2. Aplikasi mampu mengelola data *user*.
3. Aplikasi mampu mengelola program studi.
4. Aplikasi mampu mengelola konfigurasi profil ujian berupa pengaturan subjek beserta tingkat kesulitannya.
5. Aplikasi mampu mengelola konfigurasi profil ujian berupa pengaturan kelulusan berdasarkan nilai (UN atau sertifikasi) atau kuota.
6. Aplikasi mampu membuat aturan distribusi.
7. Aplikasi mampu membuat aturan nilai.
8. Aplikasi mampu membuat soal dan jawaban serta kunci jawaban.
9. Aplikasi mampu menangani registrasi peserta ujian.
10. Aplikasi menyediakan fasilitas konfigurasi pelaksanaan ujian berupa soal siap uji.

11. Aplikasi mampu melakukan pengelolaan pelaksanaan ujian berupa pengaturan waktu ujian.
12. Aplikasi menyediakan hak akses soal dan menyimpan jawaban peserta.
13. Aplikasi mampu melakukan kalkulasi nilai peserta.
14. Aplikasi mampu menampilkan informasi kelulusan peserta ujian sesuai konfigurasi yang telah ditentukan.
15. Aplikasi mampu menampilkan informasi peringkat peserta ujian.

Dari deskripsi umum Aplikasi Pengembangan Sistem Ujian Online Berbasis Objek oleh Lies Handayani & Ferri Martin masih memiliki kekurangan, seperti:

1. Aplikasi tidak mendukung perangkat *mobile (android)*.
2. Aplikasi tidak dapat mencetak lembar soal dan jawaban peserta ujian.
3. Aplikasi tidak dapat membuat cetakan rekap nilai untuk dosen.

II.12 Rumus Cara Penilaian dan Cara Pengacuannya

II.12.1 Pengolahan Nilai

Proses penilaian adalah suatu proses membandingkan skor yang diperoleh tiap mahasiswa dengan acuan yang dipakai penilaian aturan patokan atau penilaian aturan normal (PAP atau PAN), yang hasilnya berbentuk nilai dengan skala 0 – 100.

II.12.2 Skala (Skala 1-100)

Memang sebaiknya bahwa angka itu merupakan bilangan bulat. Dengan menggunakan skala 1-10 maka bilangan bulat yang ada masih menunjukkan penilaian yang agak kasar. Ada sebenarnya hasil prestasi yang berada diantara kedua angka bulat itu. Untuk itulah maka dengan menggunakan skala 1-100, dimungkinkan melakukan penilaian yang lebih halus karena terdapat 100 bilangan bulat. Nilai 5,5 dan 6,4 dalam skala 1-10 yang biasanya dibulatkan menjadi 6, dalam skala 1-100 ini boleh dituliskan dengan 55 dan 64.

$$Nilai = \frac{\sum \text{Nilai soal yang benar}}{\sum \text{Nilai Total Soal}} \times 100$$

Persamaan 1

Misalnya si A mendapat mengerjakan soal dengan benar sebanyak 100 soal dari 150 soal yang ada, maka nilai yang dia dapat adalah:

$$Nilai = \frac{100}{150} \times 100$$

$$Nilai = 66$$

II.12.3 Rangking/peringkat

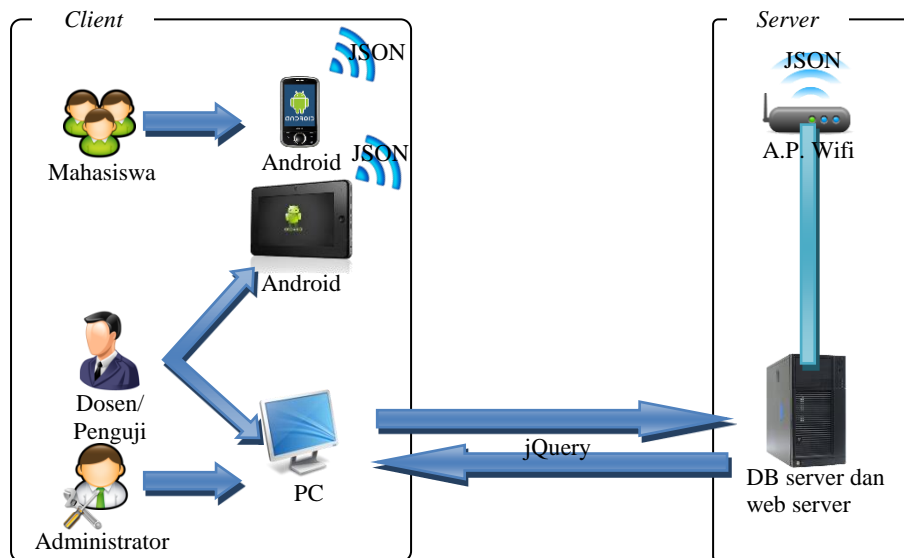
Metode ini merupakan pendekatan penyekalaan komparatif yaitu dengan menanyakan kepada responden rangking (kesatu, kedua dan seterusnya) teknik ini relatif lebih cepat dan lebih mudah dipahami responden.

Bab III Analisis dan Perancangan

Pada tugas akhir ini akan dirancang sebuah aplikasi Mobik untuk mahasiswa Politeknik Negeri Batam Jurusan Informatika. Aplikasi ini menyesuaikan dengan standar pelaksanaan dan penilaian kuis yang ada di Politeknik Negeri Batam. Mobik dirancang berbasis *client-server* dimana masing-masing peserta kuis akan menggunakan Android sebagai *client* dan akan mengakses *Access Point Wifi* yang tersambung ke *server*.

III.1 Analisis Aplikasi MOBIK

III.1.1 Deskripsi Detail Aplikasi



Gambar III-1 Detail Aplikasi

Sistem kuis yang di bangun dalam tugas akhir ini memiliki dua bagian, yaitu *client* dan *server*. Perangkat di sisi *Client* (Mahasiswa) berfungsi untuk sarana pengerjaan kuis dan pengecekan nilai. Perangkat di sisi *Client* (Dosen) berfungsi untuk sarana pembuatan soal dan jawaban, pencetakan lembar soal, pencetakan rekap nilai. Sedangkan *server* berguna sebagai media penyimpanan data berupa bank soal, kunci jawaban dan generator soal acak yang didistribusikan ke masing-masing *client* (Mahasiswa). Dan Administrator bertugas untuk

mengatur hak akses *user* sebagai pengguna aplikasi Mobik. *Client* terdiri dari dosen dan mahasiswa. Dosen dapat menggunakan perangkat berbasis Android dan/atau PC, sedangkan mahasiswa menggunakan perangkat berbasis Android, Administrator menggunakan perangkat PC di sisi *server*. Gambar III-1 menunjukkan detail dari perangkat sistem untuk aplikasi Mobik.

III.1.2 Spesifikasi Sistem

Mobik dibangun dengan spesifikasi kebutuhan-kebutuhan di bawah ini:

1. *User (Administrator, Dosen, dan mahasiswa)* bisa melakukan *Login* dengan NIM dan *Password* masing-masing.
2. *User (Administrator, Dosen, dan mahasiswa)* bisa merubah *password*.
3. *Administrator*, dapat membuat dan mengatur hak akses dan detail user.
4. *Administrator*, dapat mengatur prodi, makul, dan level pengguna.
5. Dosen membuat soal dan jawaban.
6. Dosen mengatur soal dan peserta yang akan mengikuti kuis.
7. Dosen bisa mencetak lembar soal dan jawaban.
8. Dosen bisa mencetak rekap nilai.
9. Mahasiswa bisa mengikuti kuis jika memang sudah terdaftar pada sistem kuis Mobik. Dan Mahasiswa dapat mengirim jawaban ke server.
10. Mahasiswa dapat melihat nilai kuis.
11. Mahasiswa dapat melihat peringkat nilai.

III.1.3 Analisis Karakteristik Pengguna

Karakteristik dari pengguna aplikasi Mobik ini terbagi atas 3 bagian, seperti yang terlihat pada tabel 5, di bawah ini:

Tabel 5 Aktor Pengguna Mobik

| Aktor | Definisi/Karakteristik |
|---------------|--|
| Administrator | Administrator bertugas untuk mengontrol dan <i>maintenance</i> server aplikasi Mobik. |
| Dosen | Dosen mempunyai hak akses untuk manage soal dan mahasiswa. Dosen juga bisa menentukan bobot nilai dari masing-masing soal. |
| Mahasiswa | Mahasiswa mempunyai hak akses untuk melakukan kuis, mereview kuis, melihat hasil kuis dan mengubah <i>Password</i> . |

III.1.4 Analisis Kebutuhan Hardware

Pada aplikasi Mobik, kebutuhan *hardware* dibagi dalam beberapa bagian, yaitu pada sisi *server*, *client*, dan jaringan *wireless*.

Spesifikasi minimum pada *server*:

1. Processor : Intel Core 2 Duo atau setara.
2. Memori : 2 GB.
3. HDD sistem : 100 GB.
4. HDD db : 400 GB.
5. LAN : 100/1000 Mbps.

Spesifikasi minimum pada *client PC*:

1. Processor : Intel Core 2 Duo atau setara.
2. Memori : 2 GB.
3. HDD sistem : 20 GB.
4. HDD db : 20 GB.
5. LAN : 100/1000 Mbps.

Spesifikasi minimum pada *client Android*:

1. Processor : ARMv7
2. Memori : 128KB.
3. LAN : *wireless*

III.1.5 Analisis Kebutuhan Perangkat Komunikasi

Secara umum aplikasi Mobik berkomunikasi menggunakan jaringan *wireless* (wifi). Berikut spesifikasi minimum pada jaringan *wireless*:

1. *Wireless* standar : IEEE 802.11n, IEEE 802.11g, IEEE 802.11b.
2. *Wireless security* : 64/128/152-bit WEP / WPA / WPA2, WPA-PSK / WPA2-PSK

III.1.6 Analisis Kebutuhan Software

Kebutuhan *software* pada aplikasi ini dibagi menjadi dua bagian, yaitu pada *server* dan *client*. *Software-software* ini digunakan dalam pembuatan serta

pengembangan pada sisi *server* dan juga untuk menjalankan aplikasi Mobik pada sisi *client*.

Kebutuhan minimum *software* pada sisi *server* dan untuk pengembangan:

1. *Operating System* : Windows XP Service Pack 3.
2. PHP Server : XAMPP 1.7.7
3. Editor : Notepad ++, App Inventor
4. Android Simulator : Android SDK

Kebutuhan minimum *software* pada sisi *client Android*:

1. *Operating System* : Windows XP Service Pack 3.
2. *Software* : Web Browser (Mozilla FireFox, Internet Explorer)

Kebutuhan minimum *software* pada sisi *client Android*:

1. *Operating System* : Android 4.0.2 (ICS)

III.1.7 Analisis Kebutuhan Fungsional

Aplikasi Mobik dibagi dalam beberapa fungsi seperti yang tertera dalam tabel 6, di bawah ini:

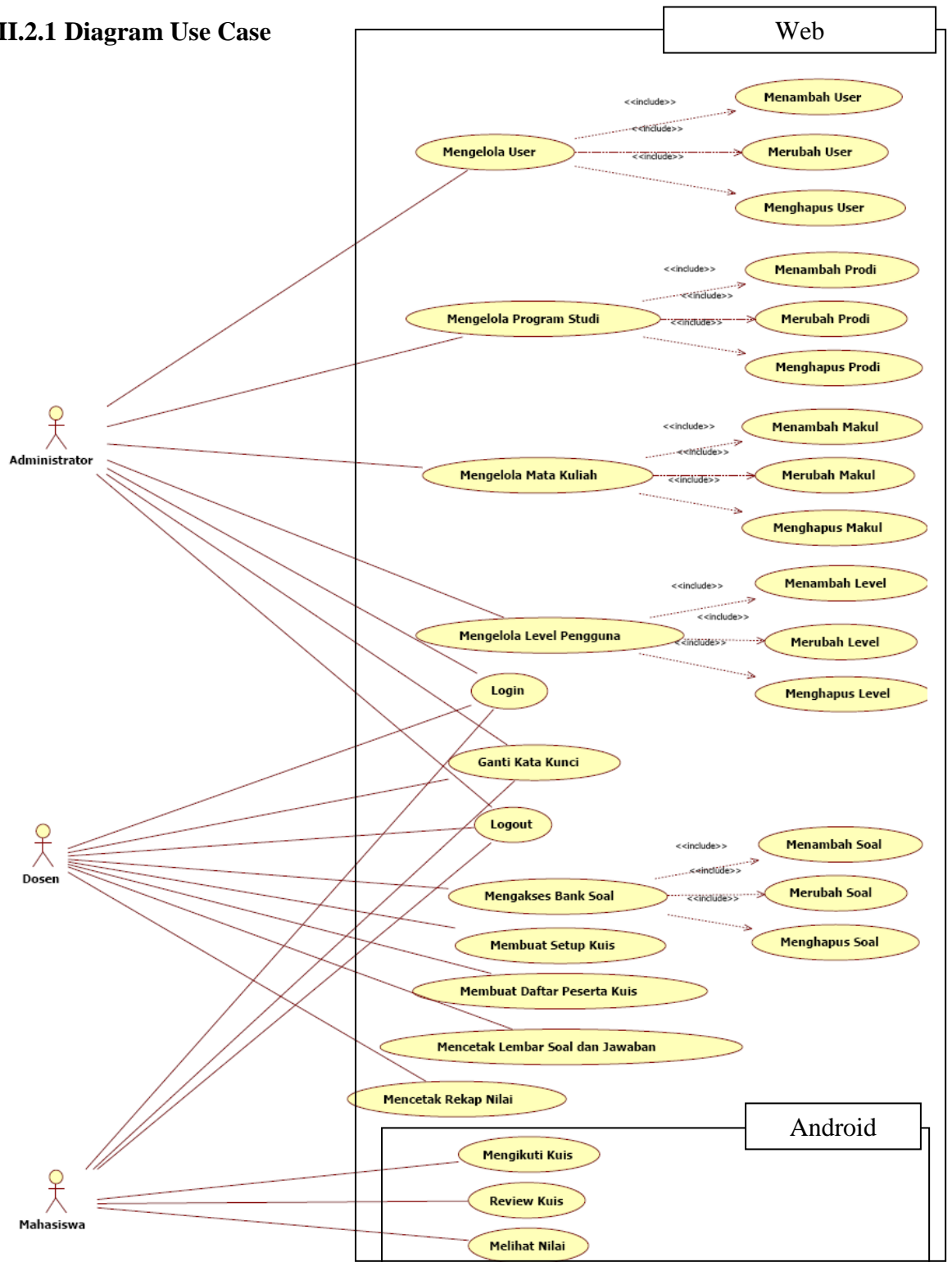
Tabel 6 Kebutuhan Fungsi

| Fungsi ID | Keterangan | Tipe Antarmuka | Pengguna |
|-----------|---|----------------|----------|
| FMOBIK-01 | Pembuatan <i>Login</i> untuk otentifikasi akses ke dalam aplikasi MOBIK. | Android / Web | Semua |
| FMOBIK-02 | Menejemen data <i>user</i> pemakai aplikasi MOBIK (mahasiswa/dosen/administrator), meliputi <i>Password</i> dan level pemakai aplikasi MOBIK. Pada Menejemen data pemakai ditampilkan nama-nama pemakai yang sudah tersimpan. | Web | Admin |
| FMOBIK-03 | Menejemen program studi yang tersedia. | Web | Admin |
| FMOBIK-04 | Menejemen mata kuliah yang tersedia. | Web | Admin |
| FMOBIK-05 | Menejemen <i>level user</i> /tingkat pengguna yang tersedia. | Web | Admin |
| FMOBIK-06 | <i>User</i> dapat mengubah <i>Password</i> nya masing-masing | Android/Web | Semua |
| FMOBIK-07 | Menejemen data bank soal, dosen dapat melakukan penambahan, perubahan, penghapusan, dan pencarian soal untuk mata kuliah dosen terkait. Pada Menejemen bank soal ditampilkan juga soal-soal yang sudah ada. | Web | Dosen |
| FMOBIK-08 | Dosen membuat setup kuis: Menejemen kuis, Daftar mahasiswa peserta kuis (mahasiswa). Waktu mulai dan lama kuis. | Web | Dosen |

| Fungsi ID | Keterangan | Tipe Antarmuka | Pengguna |
|-----------|--|----------------|-----------|
| | Jenis soal kuis (soal bebas/pilihan). Kode lembar soal. Katakunci untuk mengikuti kuis. Mengubah lama waktu kuis. Setelah setup kuis dikonfirmasi, maka aplikasi akan membuat dan mengacak soal dan mengacak pilihan jawaban untuk tiap peserta kuis ke dalam database. Sistem akan mencatat soal-soal yang akan diberikan kepada <i>client</i> dan merekap semua susunan soal dan susunan pilihan yang telah diacak oleh sistem untuk digunakan sebagai bahan pemeriksaan untuk proses penilaian. | | |
| FMOBIK-09 | <i>User</i> (mahasiswa) meminta soal dari server dan server mengambil soal yang telah dipilih/ditandai berdasarkan nama pengguna peserta kuis. | Android | Mahasiswa |
| FMOBIK-10 | <i>User</i> (mahasiswa) dapat melihat 1 soal dalam 1 layar, dapat mengerjakan soal secara acak, dan dapat memajukan/memundurkan 1 soal di layar <i>client</i> . | Android | Mahasiswa |
| FMOBIK-11 | <i>User</i> (mahasiswa) mengirim jawaban secara manual dengan menekan tombol kirim/submit di akhir soal yang diberikan. | Android | Mahasiswa |
| FMOBIK-12 | <i>Client</i> (mahasiswa) akan mengirim jawaban secara otomatis jika waktu yang telah diberikan telah habis. | Android | Mahasiswa |
| FMOBIK-13 | <i>Server</i> akan memeriksa jawaban ketika status jawaban telah berubah menjadi "terkirim/untuk di nilai" dan memberikan nilai untuk setiap soal dan memberikan nilai secara keseluruhan kepada peserta (mahasiswa) kuis. | - | Sistem |
| FMOBIK-14 | <i>User</i> (dosen) dapat mencetak soal/jawaban. (untuk rekap/bukti tertulis jika diperlukan) | Web | Dosen |
| FMOBIK-15 | <i>User</i> (dosen) dapat melihat/mencetak rekap nilai mahasiswa. (untuk rekap/bukti tertulis jika diperlukan) | Web | Dosen |
| FMOBIK-16 | <i>User</i> (mahasiswa) dapat melihat nilai. | Android | Mahasiswa |
| FMOBIK-17 | Semua <i>user</i> dapat melakukan LogOut | Android / Web | Semua |

III.2 Perancangan Aplikasi MOBIK

III.2.1 Diagram Use Case



Gambar III-2 Diagram Use Case

III.2.2 Skenario Use Case

Gambar III-2 menggambarkan interaksi 3 aktor dengan sistem. Aplikasi ini memiliki tiga aktor, yaitu Admin, Dosen dan Mahasiswa. Setiap aktor yang akan menggunakan aplikasi ini harus melakukan *Login* (masuk) terlebih dahulu, dan apabila ingin mengakhiri penggunaan aplikasi ini harus melakukan *Logout* (keluar). Dan setiap aktor dapat melakukan perubahan *Password* masing-masing.

Spesifikasi interaksi antara admin dengan sistem adalah sebagai berikut:

1. Mengelola daftar *User*
2. Mengelola program studi
3. Mengelola mata kuliah
4. Mengelola tingkat/nama pengguna

Spesifikasi interaksi antara dosen dengan sistem adalah sebagai berikut:

1. Mengakses bank soal/jawaban kuis
2. Membuat setup kuis
3. Mencetak lembar soal dan jawaban
4. Mencetak rekap nilai

Spesifikasi interaksi Mahasiswa dengan sistem adalah sebagai berikut:

1. Mengikuti kuis
2. Menampilkan hasil kuis (*review*)
3. Melihat nilai

III.2.2.1 Skenario Use Case Login

Aktor : Semua *User* (Admin, Dosen, dan Mahasiswa)

Kondisi Awal : *User* memasukkan data berupa nama pengguna dan kata sandi kemudian menekan tombol [Masuk]

Skenario Normal :

1. Aplikasi akan mengecek apakah data tersebut *valid* atau tidak.
2. Aplikasi akan menampilkan menu sesuai dengan hak akses *user*.

Skenario Alternatif :

Jika nama *user/kata kunci* salah atau tidak terdaftar:

1. Aplikasi akan menampilkan pesan kesalahan.

Kondisi Akhir : *User* tercatat masuk ke dalam sistem secara *online* dan menampilkan hak akses dan menu masing-masing *user*.

III.2.2.2 Skenario Use Case Logout

Aktor : Semua *User* (Admin, Dosen, dan Mahasiswa)

Kondisi Awal : *User* menekan/memilih tombol [Keluar]

Skenario :

1. Aplikasi akan mengeluarkan *user* dari sistem

Kondisi Akhir : *User* sudah tidak lagi tercatat menggunakan sistem secara *online*. Tampilan kembali ke halaman Login

III.2.2.3 Skenario Use Case Ganti Kata kunci

Aktor : Semua *User* (Admin, Dosen, dan Mahasiswa)

Kondisi Awal : *User* memasukkan data berupa nama *user* dan kata sandi dan menekan tombol [Ganti Kata kunci] di layar menu *login*.

Skenario Normal :

1. Aplikasi akan mengecek apakah data nama *user* dan kata sandi tersebut valid atau tidak.
2. Aplikasi akan meminta masukkan berupa *kata kunci* baru.
3. Aplikasi akan meminta konfirmasi masukkan *kata kunci* baru.
4. Aplikasi akan menyimpan *kata kunci* baru.

Skenario Alternatif :

Jika nam *user/kata kunci* salah atau tidak terdaftar:

1. Aplikasi akan menampilkan pesan kesalahan.

Kondisi Akhir : *User* tercatat masuk ke dalam sistem secara online dan menampilkan hak akses dan menu masing-masing *user*.

III.2.2.4 Skenario *Use Case* Mengelola Pengguna

Aktor : Admin

Kondisi Awal : Admin menekan tombol menu [*Manage*]

Skenario Normal :

Admin dapat menentukan hak akses setiap *user* (admin, dosen, dan mahasiswa). Selain itu, admin juga dapat menambah dan mengubah *user* serta dapat mengubah kata sandi pengguna Mobik.

Kondisi Akhir :

Informasi daftar *user* yang menggunakan aplikasi sesuai dengan hak aksesnya masing-masing.

III.2.2.5 Skenario *Use Case* Mengelola Program Studi

Aktor : Admin

Kondisi Awal : Admin memilih tombol [Program Studi]

Skenario Normal :

Admin dapat menambah serta mengubah daftar/list program studi yang akan ditampilkan pada saat peserta melakukan pendaftaran dimana peserta dapat memilih program studi tersebut.

Kondisi Akhir :

Daftar program studi yang akan ditampilkan dosen pada saat membuat soal kuis.

III.2.2.6 Skenario *Use Case* Mengelola Mata Kuliah

Aktor : Admin

Kondisi Awal : Admin memilih tombol [Mata Kuliah]

Skenario Normal :

Admin dapat menambah serta mengubah daftar mata kuliah yang akan ditampilkan pada saat peserta melakukan pendaftaran dimana peserta dapat memilih mata kuliah tersebut.

Kondisi Akhir :

Daftar mata kuliah yang akan ditampilkan dosen pada saat membuat soal kuis.

III.2.2.7 Skenario *Use Case* Mengelola Level Pengguna

Aktor : Admin

Kondisi Awal : Admin memilih tombol [Level Pengguna]

Skenario Normal :

Admin dapat menambah serta mengubah daftar level pengguna yang akan digunakan untuk menentukan akses dari masing-masing *user*.

Kondisi Akhir :

Daftar level pengguna yang akan ditampilkan pada saat pendaftaran pengguna.

III.2.2.8 Skenario *Use Case* Mengakses Bank Soal

Aktor : Dosen

Kondisi Awal : Dosen menekan tombol [Bank Soal]

Skenario Normal :

Dosen memilih prodi dan mata kuliah untuk membuat/mengubah/menghapus soal yang akan di atur untuk disimpan pada bank soal.

Dosen dapat membuat/mengubah/menghapus:

1. Soal
2. Pilihan jawaban
3. Jawaban

4. Lampiran gambar soal
5. Lampiran gambar jawaban
6. Bobot nilai

Kondisi Akhir :

Sistem akan menyimpan soal dan jawaban ke dalam basis data dalam bank data soal dan menyimpan gambar lampiran dengan kode unik sesuai dengan kode soal.

III.2.2.9 Skenario *Use Case* Membuat Setup Kuis

Aktor : Dosen

Kondisi Awal : Dosen menekan tombol [Setup Kuis]

Skenario Normal :

Dosen dapat membuat/mempersiapkan:

1. Memilih program studi dan mata kuliah untuk soal yang akan dikuiskan
2. Menentukan jenis soal (dari bank data soal)
3. Waktu Kuis (tanggal & jam kuis)
4. Lama Kuis
5. Peserta Kuis
6. Kata kunci mengikuti kuis

Kondisi Akhir :

Sistem akan membuat Kode lembar soal untuk setup kuis tersebut yang akan diberikan kepada mahasiswa untuk mengambil lembar soal di bank data soal saat pelaksanaan kuis.

III.2.2.10 Skenario *Use Case* Membuat Daftar Peserta Kuis

Aktor : Dosen

Kondisi Awal : Dosen menekan tombol [Daftar Peserta Kuis]

Skenario Normal :

Dosen dapat membuat/mempersiapkan:

1. Dosen memasukkan kode lembar jawaban
2. Secara default peserta kuis diambil dari dari daftar mahasiswa untuk prodi dan mata kuliah tersebut untuk di seleksi/di pilih ulang.
3. Dosen dapat merubah daftar peserta kuis dan menyimpan daftar peserta kuis tersebut.

Kondisi Akhir :

Sistem akan menyimpan daftar peserta kuis untuk kode lembar soal tersebut dan hanya akan memberikan lembar soal kepada mahasiswa yang telah dipilih.

III.2.2.11 Skenario *Use Case* Mencetak Lembar Soal dan Jawaban

Aktor : Dosen

Kondisi Awal : Dosen menekan tombol [Cetak Lembar Soal dan Jawaban]

Skenario Normal :

Dosen dapat memilih kode lembar soal yang akan di cetak dengan memilih:

1. Semua lembar jawaban dari mahasiswa yang mengikuti kuis
2. Hanya lembar jawaban untuk mahasiswa tertentu saja

Kondisi Akhir :

Sistem akan mencetak lembar soal dan jawaban beserta nilai dalam format PDF untuk semua mahasiswa yang mengikuti kuis atau mahasiswa tertentu sesuai pilihan.

III.2.2.12 Skenario *Use Case* Mencetak Rekap Nilai

Aktor : Dosen

Kondisi Awal : Dosen menekan tombol [Cetak Rekap Nilai]

Skenario Normal :
Dosen dapat memilih kode lembar soal yang akan di cetak daftar rekap nilai untuk semua lembar jawaban dari mahasiswa yang mengikuti kuis.

Kondisi Akhir :
Sistem akan mencetak rekap nilai dalam format PDF untuk semua mahasiswa peserta kuis.

III.2.2.13 Skenario *Use Case* Mengikuti Kuis

Aktor : Mahasiswa

Kondisi Awal : Mahasiswa menekan tombol [Mulai Kuis]

Skenario Normal :

1. Mahasiswa akan diminta memasukkan kode lembar jawaban.
2. Mahasiswa akan diminta memasukkan kata kunci untuk lembar jawaban.
3. Setelah kode lembar jawaban dan kata kuncinya cocok, maka system akan menampilkan soal-soal untuk diberikan kepada mahasiswa tersebut.

Skenario Alternatif :
Jika mahasiswa tidak dapat memasukkan kode lembar jawaban dan kata kunci untuk lembar soal tersebut, maka mahasiswa tersebut tidak akan dapat mengikuti kuis.

Kondisi Akhir :
Mahasiswa dapat mengikuti kuis.

III.2.2.14 Skenario *Use Case* Review Kuis

Aktor : Mahasiswa

Kondisi Awal : Mahasiswa menekan tombol [Review Kuis]

Skenario Normal :

1. Mahasiswa akan diminta memasukkan kode lembar jawaban.
2. Mahasiswa akan diminta memasukkan kata kunci untuk lembar jawaban.
3. Setelah kode lembar jawaban dan kata kuncinya cocok, maka system akan menampilkan soal-soal untuk diberikan kepada mahasiswa tersebut. Jika mahasiswa belum pernah mengikuti kuis tersebut maka system tidak akan menampilkan soal-soal dan jawaban kuis.

Skenario Alternatif :

Jika mahasiswa tidak dapat memasukkan kode lembar jawaban dan kata kunci untuk lembar soal tersebut, maka mahasiswa tersebut tidak akan dapat melihat kuis yang telah diberikan.

Kondisi Akhir :

Mahasiswa dapat melihat soal-soal dan jawaban kuis yang telah dibuatnya.

III.2.2.15 Skenario Use Case Melihat Nilai

Aktor : Mahasiswa

Kondisi Awal : Mahasiswa menekan tombol [Review Nilai]

Skenario Normal :

1. Mahasiswa akan diminta memasukkan kode lembar jawaban.
2. Mahasiswa akan diminta memasukkan kata kunci untuk lembar jawaban.
3. Setelah kode lembar jawaban dan kata kuncinya cocok, maka system akan menampilkan soal-soal untuk diberikan kepada mahasiswa tersebut. Jika

mahasiswa belum pernah mengikuti kuis tersebut maka system tidak akan menampilkan nilai dan ranking pada soal tersebut.

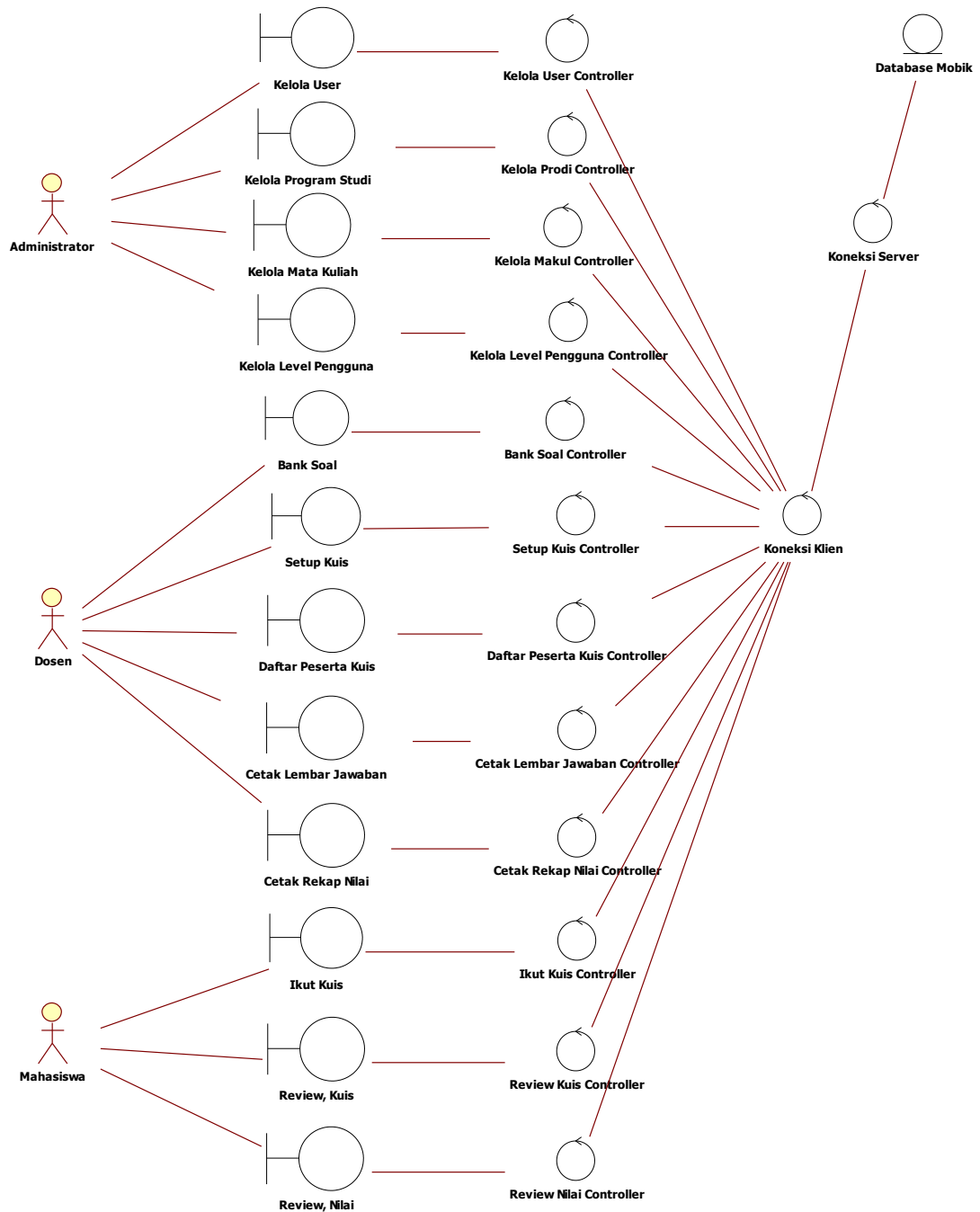
Skenario Alternatif :

Jika mahasiswa tidak dapat memasukkan kode lembar jawaban dan kata kunci untuk lembar soal tersebut, maka mahasiswa tersebut tidak akan dapat melihat nilai kuis yang telah diberikan.

Kondisi Akhir :

Mahasiswa dapat melihat soal-soal dan jawaban kuis yang telah dibuatnya.

III.3 Analisis Kelas



Gambar III-3 Analisis Kelas

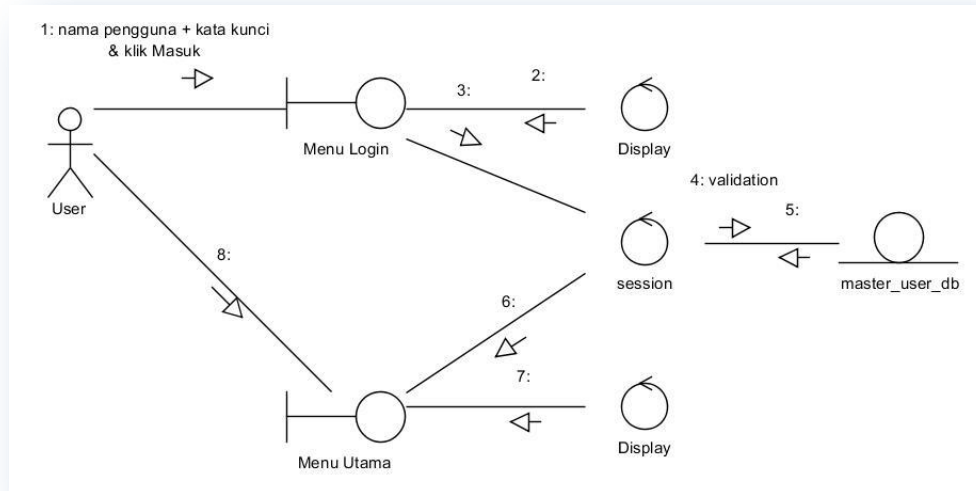
Gambar III-3 menjelaskan tentang analisis kelas pada aplikasi mobik ini. Adapun rincian dari kelas-kelas pada gambar tersebut dijelaskan pada tabel 7, sebagai berikut:

Tabel 7 Spesifikasi Analisis Kelas

| Jenis Kelas | Nama Kelas | Deskripsi |
|--|--|---|
| Kelas <i>Boundary</i> | Kelola <i>User</i> | Menyimpan informasi data <i>User</i> (pengguna Mobik): Administrator, Dosen, dan Mahasiswa. |
| | Kelola Program Studi | Menyimpan informasi data Program Studi. |
| | Kelola Mata Kuliah | Menyimpan informasi data Mata Kuliah. |
| | Kelola <i>Level</i> Pengguna | Menyimpan informasi data Tingkat Pengguna. |
| | Bank Soal | Menyimpan informasi data Soal-soal kuis. |
| | Setup Kuis | Menyimpan informasi data program studi, mata kuliah, jenis soal, waktu kuis, lama kuis, peserta kuis, kode lembar soal, dan kata kunci untuk mengikuti kuis. |
| | Daftar Peserta Kuis | Menyimpan informasi data peserta kuis untuk lembar soal yang telah ditentukan pada setup kuis. |
| | Cetak Lembar Jawaban | Sistem akan mencetak lembar soal berdasarkan mahasiswa atau matakuliah. |
| | Cetak Rekap Nilai | Sistem akan mencetak rekap nilai untuk lembar jawaban mahasiswa atau mata kuliah. |
| | Ikut Kuis | Mahasiswa dapat mengikuti kuis dan mengirim jawaban, dan sistem akan mengacak susunan soal dan susunan pilihan jawaban ketika mahasiswa mulai mengikuti kuis dengan mengisi kode lembar soal dan kata kunci untuk mengikuti kuis. |
| | <i>Review</i> , Kuis | Mahasiswa dapat meninjau ulang soal dan jawabannya. |
| | <i>Review</i> , Nilai | Mahasiswa dapat melihat nilai pada daftar nilai. |
| | Kelas <i>Controller</i> | <i>User Controller</i> |
| Kelola Program Studi <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Kelola Mata Kuliah <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Kelola <i>Level</i> Pengguna <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Bank Soal <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Setup Kuis <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Daftar Peserta Kuis <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Cetak Lembar Jawaban <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Cetak Rekap Nilai <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Ikut Kuis <i>Controller</i> | | Mengatur akses data dari dan ke database |
| <i>Review</i> Kuis <i>Controller</i> | | Mengatur akses data dari dan ke database |
| <i>Review</i> Nilai <i>Controller</i> | | Mengatur akses data dari dan ke database |
| Koneksi Klien | | Mengatur akses dari Klien ke database Mobik |
| Koneksi Server | Mengatur akses dari Server ke database Mobik | |
| Kelas <i>Entity</i> | Database Mobik | Menyimpan dan mengatur informasi data ke dalam tabel-tabel di database Mobik. |

III.3.1 Diagram Interaksi

III.3.1.1 Diagram Interaksi *Use Case Login*



Gambar III-4 Robustness Diagram – *Login*

Penjelasan proses *login*. Setiap *user* (administrator, dosen, dan mahasiswa).

1: memasukkan nama pengguna dan kata kunci, kemudian klik tombol Masuk pada layar menu Login.

2: sistem menampilkan menu Login.

3: sistem membuat suatu *session* untuk pengguna yang melakukan *login*.

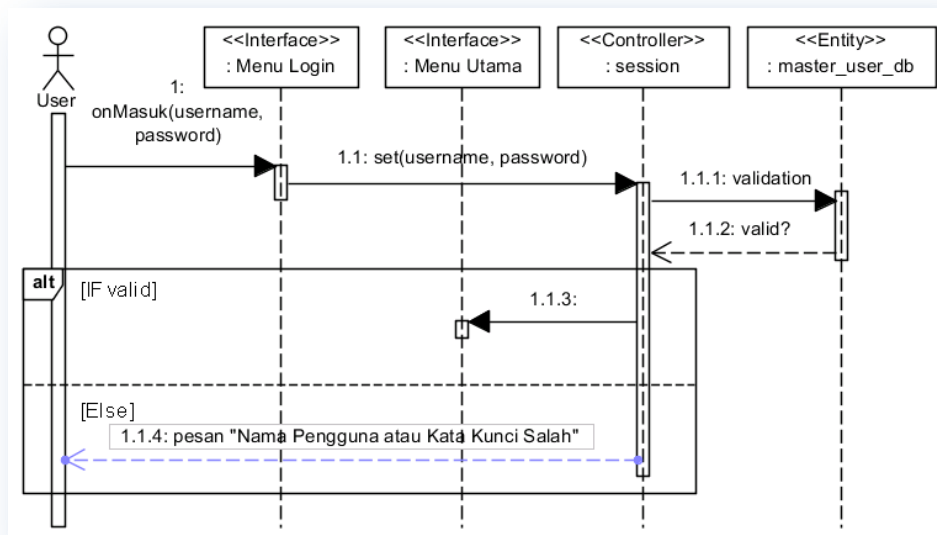
4: sistem memvalidasi nama pengguna dan kata kunci yang diterima dari user dengan data pada *database* *master_user_db*.

5: sistem memberikan hasil validasi untuk *session* tersebut.

6: jika user valid maka sistem akan menampilkan menu Utama sesuai dengan tingkat pengguna.

7: sistem menampilkan menu Utama.

8: menu Utama bisa di akses oleh pengguna mobik.



Gambar III-5 Sequence Diagram – Login

Penjelasan proses *login*. Setiap *user* (administrator, dosen, dan mahasiswa).

1: ketika nama pengguna dan kata kunci di kirim dari menu Login melalui method *OnMasuk()*.

1.1: maka sistem akan men-set *session* dengan nama pengguna dan kata kunci.

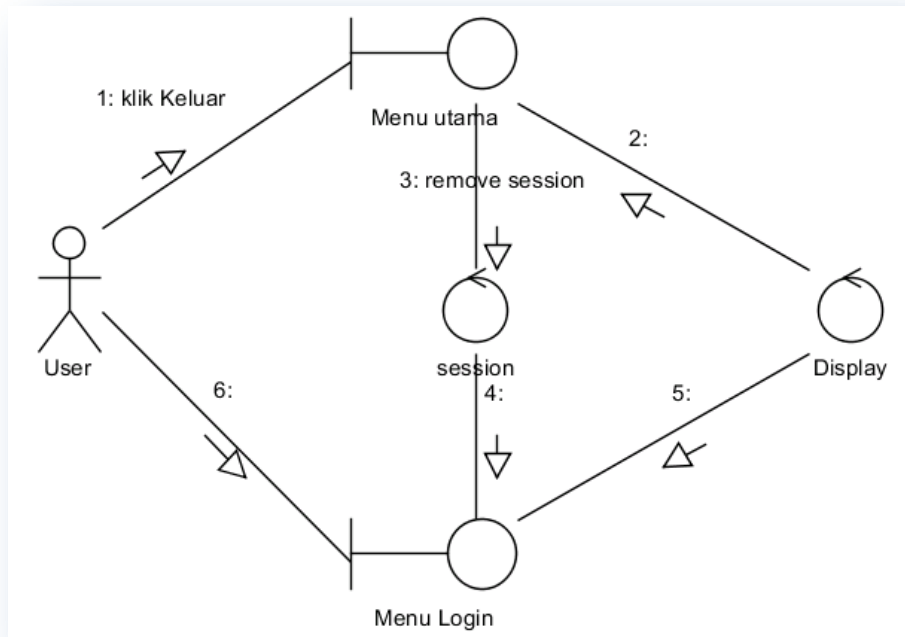
1.1.1: sistem akan melakukan validasi dengan mencocokkan data di *database* *master_user_db*.

1.1.2: kembalikan nilai hasil validasi untuk *session* tersebut.

1.1.3: jika hasilnya valid, maka sistem akan menampilkan menu Utama.

1.1.4: jika hasilnya tidak valid, maka sistem akan menampilkan pesan ke pengguna dengan menampilkan “Nama pengguna atau kata kunci salah.”

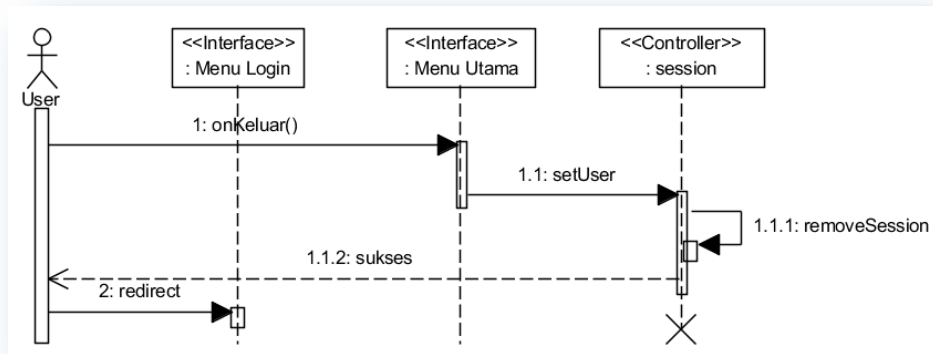
III.3.1.2 Diagram Interaksi Use Case Logout



Gambar III-6 Robustness Diagram – Logout

Penjelasan proses *logout*. Setiap *user* (administrator, dosen, dan mahasiswa).

- 1: klik tombol Keluar pada menu Utama
- 2: sistem menampilkan menu Utama.
- 3: menghapus *session* pengguna pada *memory*.
- 4: kembali ke menu Login.
- 5: sistem menampilkan menu Login.
- 6: tampilan kembali ke menu Login.



Gambar III-7 Sequence Diagram – Logout

Penjelasan proses *logout*. Setiap *user* (administrator, dosen, dan mahasiswa).

1: klik tombol Keluar.

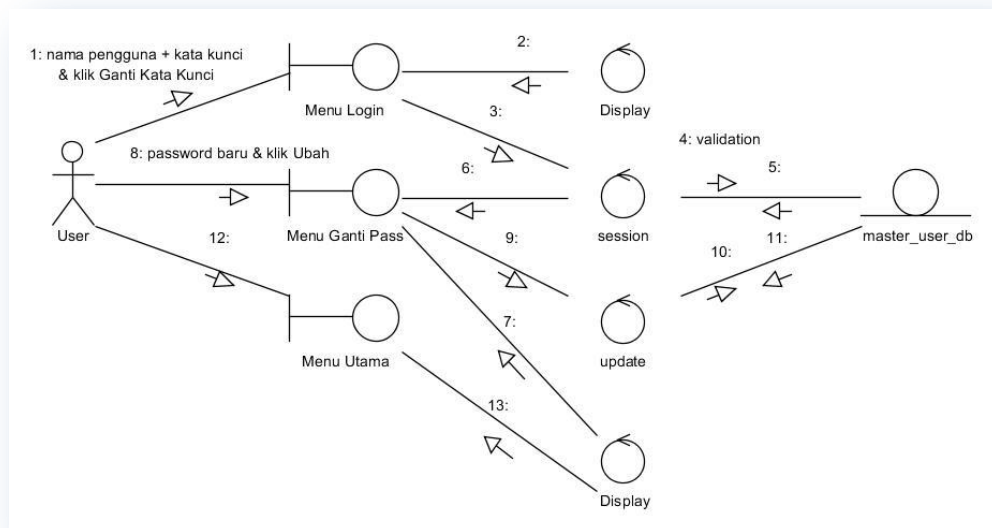
1.1: sistem men-set *user*.

1.1.1: sistem melakukan penghapusan pada *memory*.

1.1.2: pesan “Sukses”.

2: *user* kembali ke menu Login.

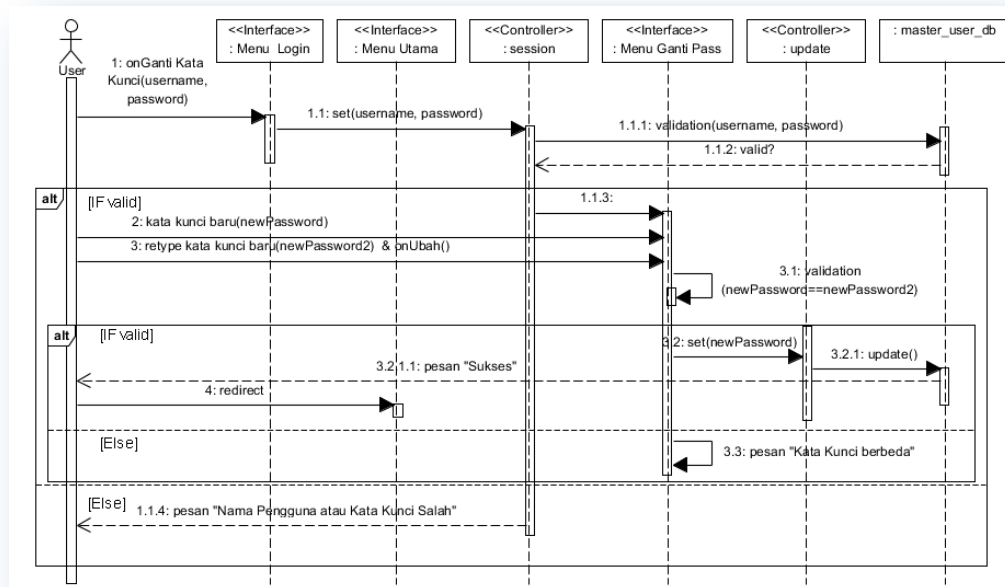
III.3.1.3 Diagram Interaksi Use Case Ganti Kata Kunci



Gambar III-8 Robustness Diagram – Ganti Kata Kunci

Penjelasan proses ganti kata kunci. Setiap *user* (administrator, dosen, dan mahasiswa).

- 1: memasukkan nama pengguna dan kata kunci, kemudian klik tombol Masuk pada layar menu Login.
- 2: sistem menampilkan menu Login.
- 3: sistem membuat suatu *session* untuk pengguna yang melakukan *login*.
- 4: sistem memvalidasi nama pengguna dan kata kunci yang diterima dari user dengan data pada *database* *master_user_db*.
- 5: sistem memberikan hasil validasi untuk *session* tersebut.
- 6: jika user valid maka sistem akan menampilkan menu Ganti password.
- 7: sistem menampilkan menu Ganti password.
- 8: memasukkan kata kunci atau *password* baru dan klik tombol Ubah.
- 9, 10, 11: sistem melakukan *update* ke *database* *master_user_db*.
- 12: mengakses menu Utama.
- 13: sistem menampilkan menu Utama.



Gambar III-9 Sequence Diagram – Ganti Kata Kunci

Penjelasan proses ganti kata kunci. Setiap *user* (administrator, dosen, dan mahasiswa)

1: ketika nama pengguna dan kata kunci di kirim dari menu Login melalui method OnMasuk().

1.1: maka sistem akan men-set *session* dengan nama pengguna dan kata kunci.

1.1.1: sistem akan melakukan validasi dengan mencocokkan data di *database* master_user_db.

1.1.2: kembalikan nilai hasil validasi untuk *session* tersebut.

Jika valid

1.1.3: sistem akan menampilkan menu Ganti *password*.

2: memasukkan kata kunci baru.

3: memasukkan kembali kata kunci baru dan klik tombol Ubah.

3.1: sistem melakukan validasi kata kunci baru.

Jika valid

3.2: men-set kata kunci baru.

3.2.1: melakukan *update* ke *database* master_user_db.

3.2.1.1: menampilkan pesan “Sukses”.

4: mengakses menu Utama.

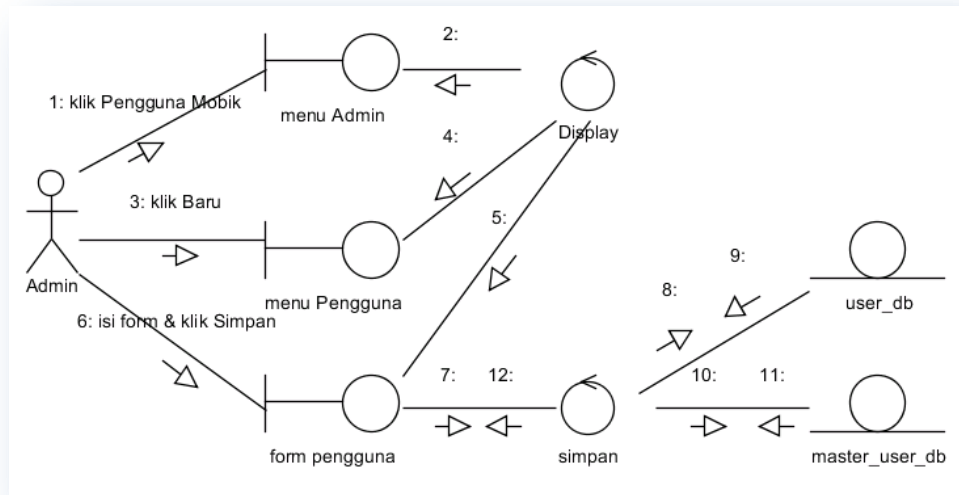
Jika tidak

3.3: menampilkan pesan “Kata kunci berbeda”.

Jika tidak

1.1.4: menampilkan pesan “Nama pengguna atau kata kunci salah.”

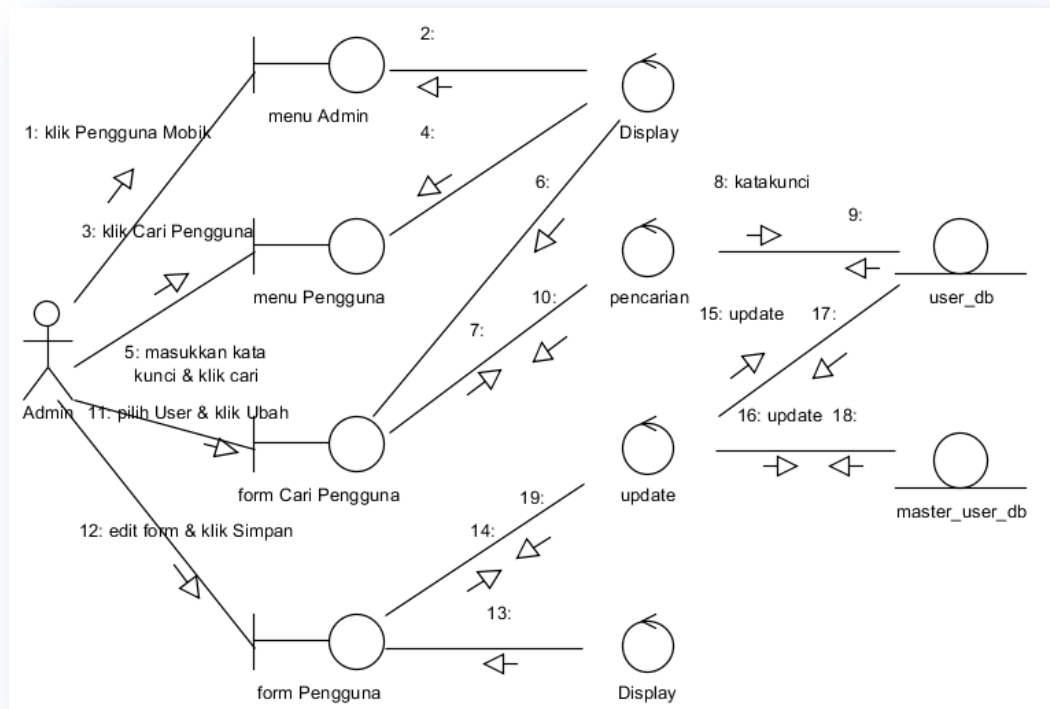
III.3.1.4 Diagram Interaksi Use Case Kelola User



Gambar III-10 Robustness Diagram – Tambah User

Penjelasan proses tambah *user* atau pemakai oleh Admin.

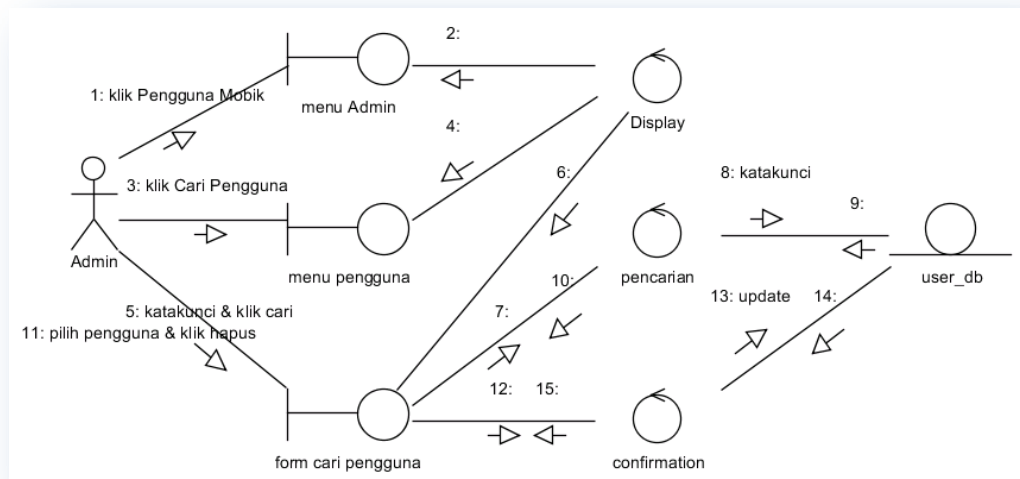
- 1: klik tombol Pengguna Mobik di menu Admin.
- 2: menampilkan menu Admin.
- 3: klik tombol Baru di menu Pengguna.
- 4: menampilkan menu Pengguna.
- 5: mengisi form dan klik tombol Simpan di form pengguna.
- 6: menampilkan form pengguna.
- 7-12: proses simpan ke *database* *user_db* dan *master_user_db*.



Gambar III-11 Robustness Diagram – Ubah User

Penjelasan proses ubah *user* atau pemakai oleh Admin.

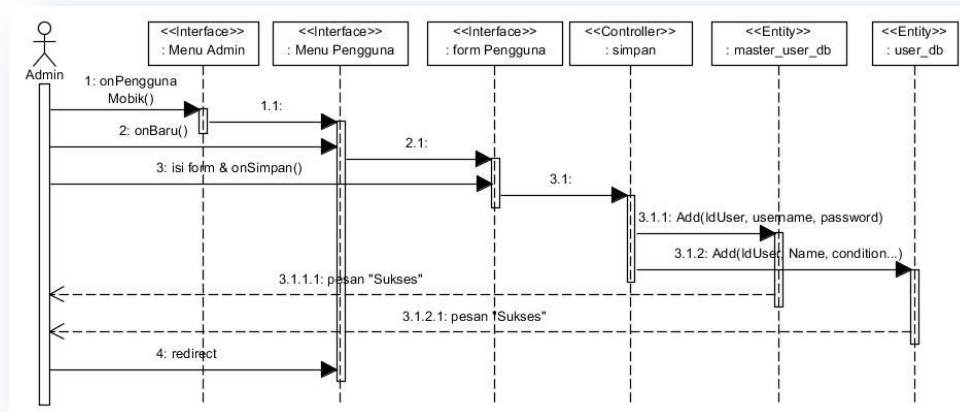
- 1: klik tombol Pengguna Mobik di menu Admin.
- 2: menampilkan menu Admin.
- 3: klik tombol Cari Pengguna di menu Pengguna.
- 4: menampilkan menu Pengguna.
- 5: memasukkan kata kunci dan klik tombol Cari.
- 6: menampilkan form Cari Pengguna.
- 7-10: proses pencarian ke *database* user_db sesuai kata kunci.
- 11: memilih *user*/pengguna yang akan diubah dan klik tombol Ubah.
- 12: ubah isi form dan klik tombol Simpan di form Pengguna.
- 13: menampilkan form Pengguna.
- 14-19: proses penyimpanan data ke ke *database* user_db dan master_user_db.



Gambar III-12 Robustness Diagram – Hapus User

Penjelasan proses hapus *user* atau pemakai oleh Admin.

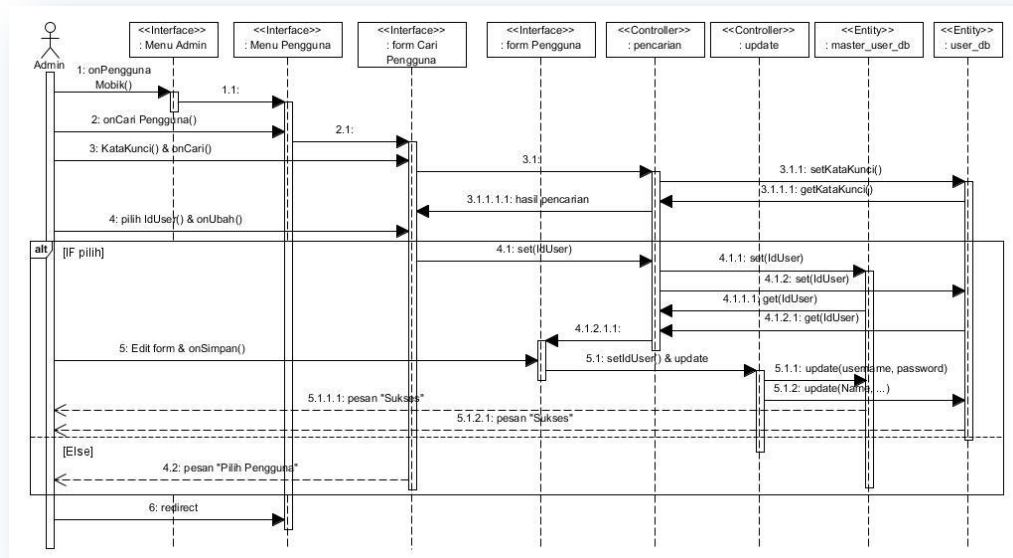
- 1: klik tombol *Pegguna Mobik* di menu Admin.
- 2: menampilkan menu Admin.
- 3: klik tombol *Cari Pengguna* di menu *Pengguna*.
- 4: menampilkan menu *Pengguna*.
- 5: memasukkan kata kunci dan klik tombol *Cari*.
- 6: menampilkan form *Cari Pengguna*.
- 7-10: proses pencarian ke *database user_db* sesuai kata kunci.
- 11: memilih *user/pengguna* yang akan dihapus dan klik tombol *Hapus*.
- 12-15: proses penghapusan dan konfirmasi *user/pengguna* di *database user_db* dengan mengubah nilai *condition = 0*.



Gambar III-13 Sequence Diagram – Tambah User

Penjelasan proses tambah *user* atau pemakai oleh Admin.

- 1: klik tombol Pengguna Mobik di menu Admin.
- 1.1: menampilkan menu Pengguna.
- 2: klik tombol Baru.
- 2.1: menampilkan form Pengguna.
- 3: pengisian form untuk *user* baru dan klik tombol Simpan.
- 3.1: set data untuk di simpan.
- 3.1.1: penambahan data idUser, username, password ke *database* master_user_db.
- 3.1.2: penambahan data idUser, Name, Prodi, Add, BoD, Condition ke *database* user_db.
- 3.1.1.1: pesan “Sukses”.
- 3.1.1.2: pesan “Sukses”.
- 4: akses langsung ke menu Pengguna.



Gambar III-14 Sequence Diagram – Ubah User

Penjelasan proses ubah *user* atau pemakai oleh Admin.

1: klik tombol Pengguna Mobik di menu Admin.

1.1: menampilkan menu Pengguna.

2: klik tombol Cari Pengguna.

2.1: menampilkan form Cari Pengguna.

3: memasukkan kata kunci dan klik tombol Cari.

3.1: melakukan pencarian.

3.1.1: set kata kunci ke *database* user_db.

3.1.1.1: get kata kunci dari *database* user_db.

3.1.1.1.1: menampilkan hasil pencarian pada form Cari Pengguna.

4. memilih *user* yang akan diubah dan klik tombol Ubah.

Jika pilih

4.1: set IdUser.

4.1.1: set IdUser ke *database* master_user_db.

4.1.2: set IdUser ke *database* user_db.

4.1.1.1: get IdUser dari *database* master_user_db.

4.1.2.1: set IdUser dari *database* user_db.

4.1.2.1.1: tampilkan pada form Pengguna.

5: ubah isi form dan klik tombol Simpan.

5.1: set IdUser.

5.1.1: *update* ke *database* master_user_db.

5.1.2: *update* ke *database* user_db.

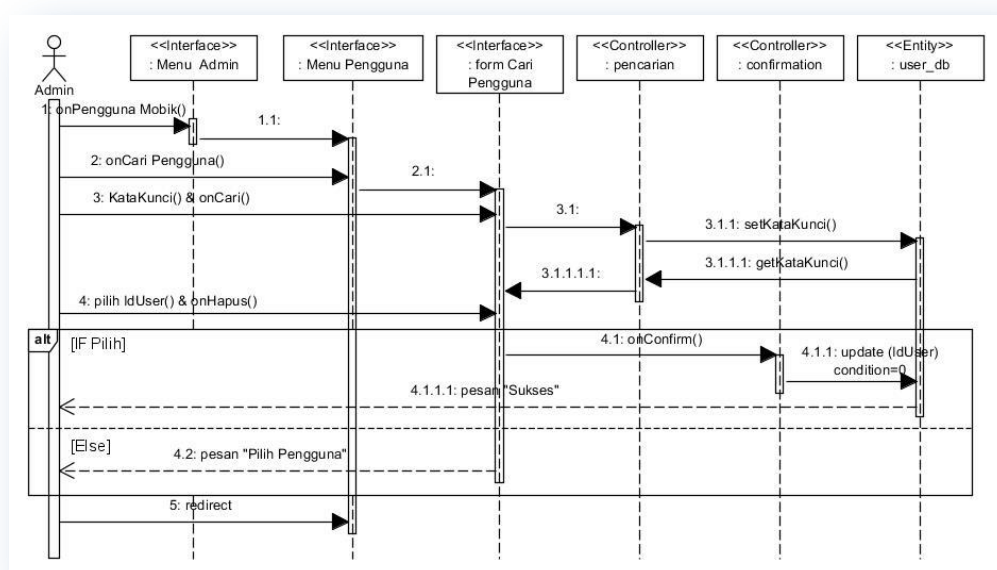
5.1.1.1: pesan “Sukses”.

5.1.1.2: pesan “Sukses”.

Jika tidak

4.2: pesan “Pilih Pengguna”.

6: akses langsung ke menu Pengguna.



Gambar III-15 Sequence Diagram – Hapus User

Penjelasan proses ubah *user* atau pemakai oleh Admin.

1: klik tombol Pengguna Mobik di menu Admin.

1.1: menampilkan menu Pengguna.

2: klik tombol Cari Pengguna.

2.1: menampilkan form Cari Pengguna.

3: memasukkan kata kunci dan klik tombol Cari.

3.1: melakukan pencarian.

3.1.1: set kata kunci ke *database* user_db.

3.1.1.1: get kata kunci dari *database* user_db.

3.1.1.1.1: menampilkan hasil pencarian pada form Cari Pengguna.

4. memilih *user* yang akan diubah dan klik tombol Hapus.

Jika pilih

4.1: klik tombol konfirmasi.

4.1.1: *update* IdUser condition = 0 pada *database* user_db.

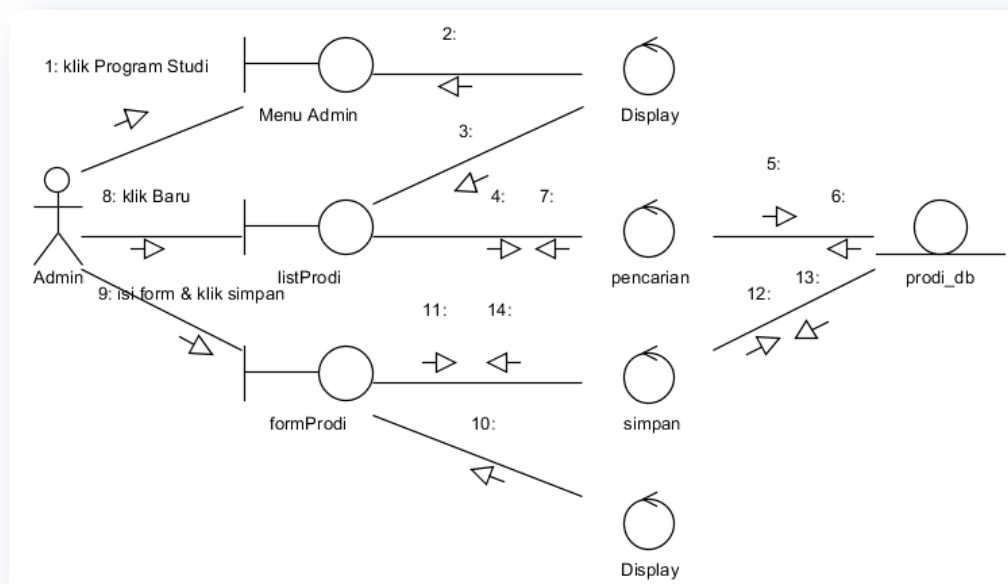
4.1.1.1: pesan “Sukses”.

Jika tidak

4.2: pesan “Pilih Pengguna”.

5: akses langsung ke menu Pengguna.

III.3.1.5 Diagram Interaksi Use Case Kelola Program Studi



Gambar III-16 Robustness Diagram – Tambah Prodi

Penjelasan proses menambah prodi oleh Admin.

1: klik tombol Program Studi pada menu Admin.

2: menampilkan menu Admin.

3: menampilkan listProdi.

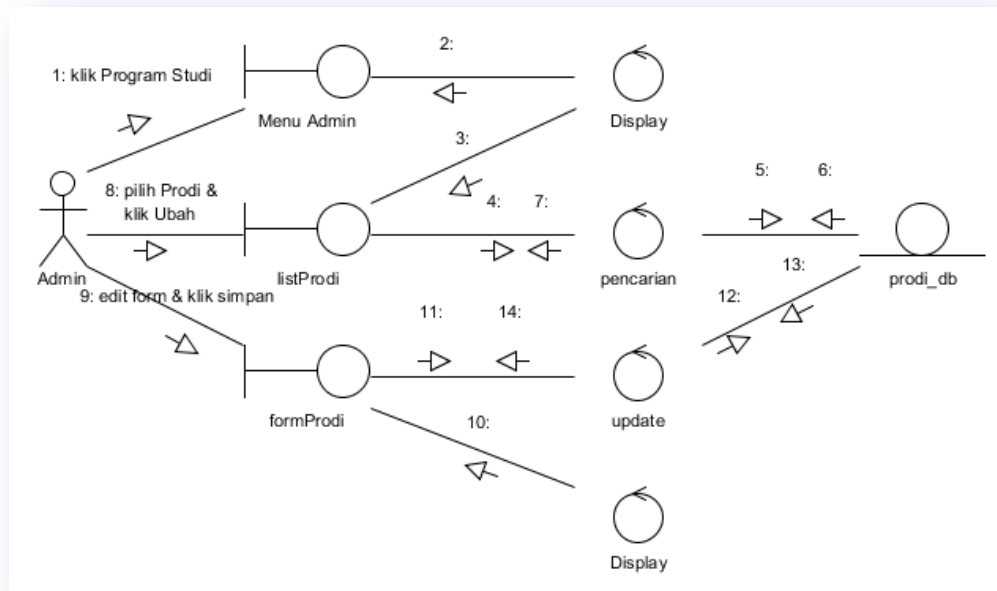
4-7: pencarian prodi dan menampilkan di listProdi.

8: klik tombol Baru.

9: isi form dan klik tombol Simpan.

10: menampilkan formProdi.

11-14: proses penyimpanan ke *database* prodi_db.



Gambar III-17 Robustness Diagram – Ubah Prodi

Penjelasan proses ubah prodi oleh Admin.

1: klik tombol Program Studi pada menu Admin.

2: menampilkan menu Admin.

3: menampilkan listProdi.

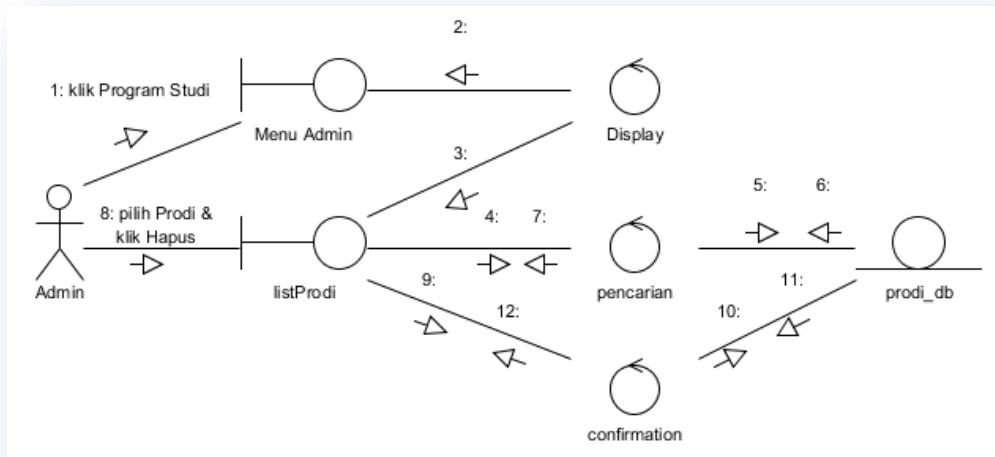
4-7: pencarian prodi dan menampilkan di listProdi.

8: klik tombol Baru.

9: ubah isi form dan klik tombol Simpan.

10: menampilkan formProdi.

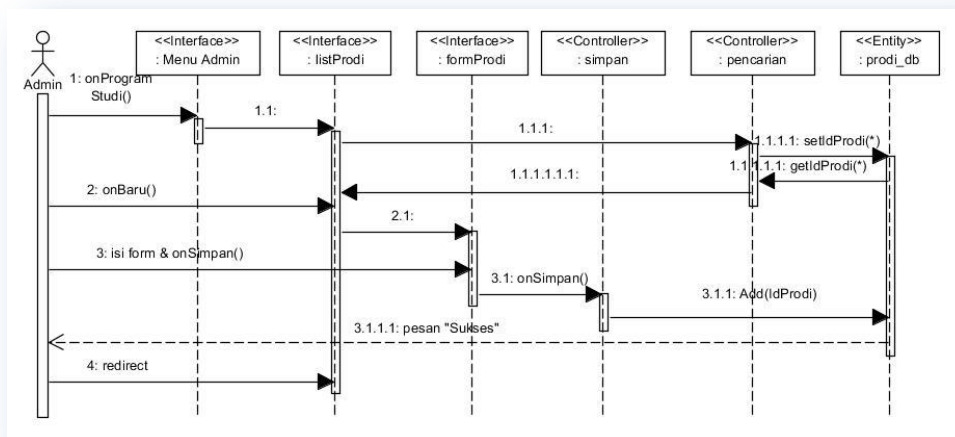
11-14: proses *update* ke *database* prodi_db.



Gambar III-18 Robustness Diagram – Hapus Prodi

Penjelasan proses hapus prodi oleh Admin.

- 1: klik tombol Program Studi pada menu Admin.
- 2: menampilkan menu Admin.
- 3: menampilkan listProdi.
- 4-7: pencarian prodi dan menampilkan di listProdi.
- 8: pilih Prodi dan klik tombol Hapus.
- 9-12: proses *update* condition=0 di *database* prodi_db.

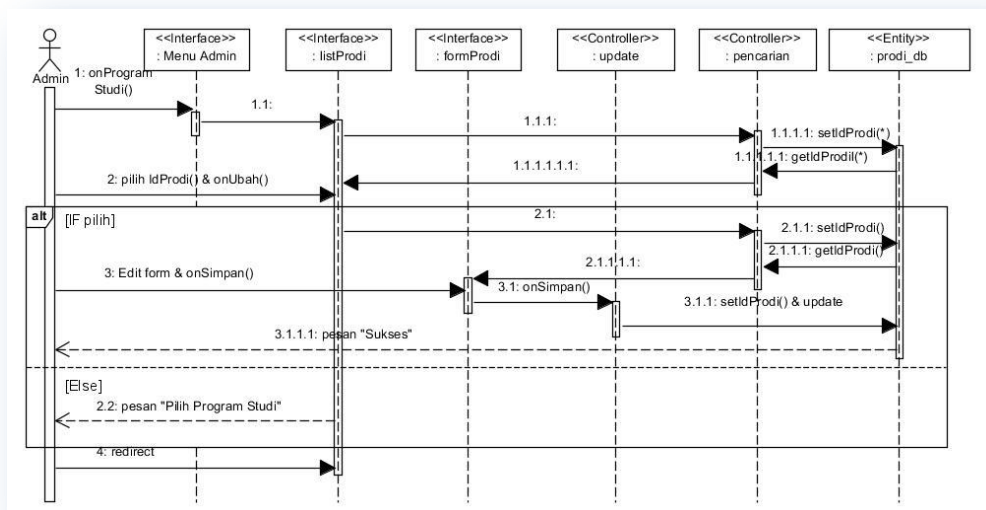


Gambar III-19 Sequence Diagram – Tambah Prodi

Penjelasan proses menambah prodi oleh Admin.

- 1: klik tombol Program Studi pada menu Admin.

- 1.1: menampilkan listProdi.
- 1.1.1: melakukan pencarian.
- 1.1.1.1: set IdProdi ke *database* prodi_db.
- 1.1.1.1.1: get IdProdi dari *database* prodi_db.
- 1.1.1.1.1.1: menampilkan listProdi dari *database* prodi_db.
- 2: klik tombol Baru.
- 2.1: menampilkan formProdi.
- 3: pengisian form dan klik tombol Simpan.
- 3.1: proses penyimpanan.
- 3.1.1: menambahkan ke *database* prodi_db.
- 3.1.1.1: pesan “Sukses”.
- 4: akses langsung ke listProdi.



Gambar III-20 Sequence Diagram – Ubah Prodi

Penjelasan proses ubah prodi oleh Admin.

- 1: klik tombol Program Studi pada menu Admin.
- 1.1: menampilkan listProdi.
- 1.1.1: melakukan pencarian.
- 1.1.1.1: set IdProdi ke *database* prodi_db.
- 1.1.1.1.1: get IdProdi dari *database* prodi_db.
- 1.1.1.1.1.1: menampilkan listProdi dari *database* prodi_db.

2: pilih IdProdi dan klik tombol Ubah.

Jika pilih

2.1: melakukan pencarian.

2.1.1: set IdProdi ke *database* prodi_db.

2.1.1.1: get IdProdi dari *database* prodi_db.

2.1.1.1.1: menampilkan hasil ke formProdi dari *database* prodi_db.

3: ubah isi form dan klik tombol Simpan.

3.1: melakukan penyimpanan dengan update data.

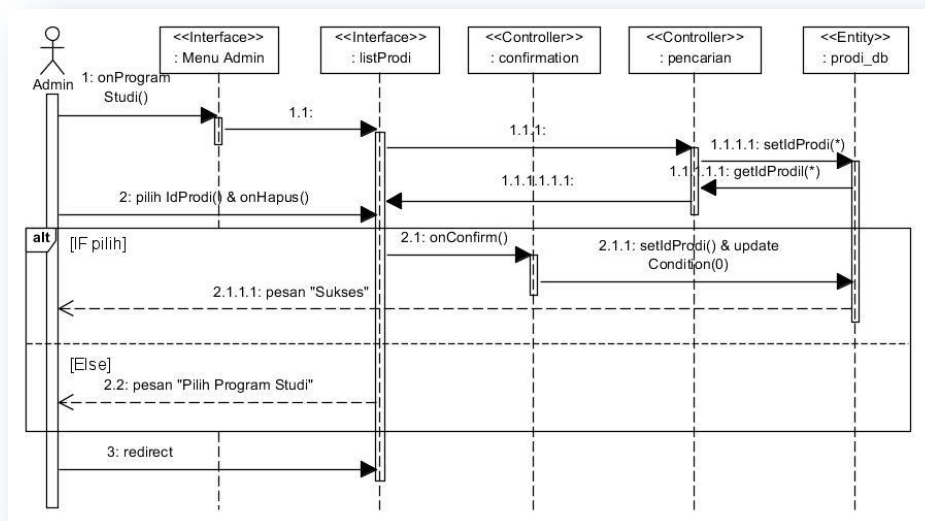
3.1.1: set IdProdi dan update *database* prodi_db.

3.1.1.1: pesan "Sukses".

Jika tidak

2.2: pesan "Pilih Program Studi".

4: akses langsung ke listProdi.



Gambar III-21 Sequence Diagram – Hapus Prodi

Penjelasan proses hapus prodi oleh Admin.

1: klik tombol Program Studi pada menu Admin.

1.1: menampilkan listProdi.

1.1.1: melakukan pencarian.

1.1.1.1: set IdProdi ke *database* prodi_db.

1.1.1.1.1: get IdProdi dari *database* prodi_db.

1.1.1.1.1: menampilkan listProdi dari *database* prodi_db.

2: pilih IdProdi dan klik tombol Hapus.

Jika pilih

2.1: melakukan konfirmasi.

2.1.1: set IdProdi dan *update condition=0* pada *database* prodi_db.

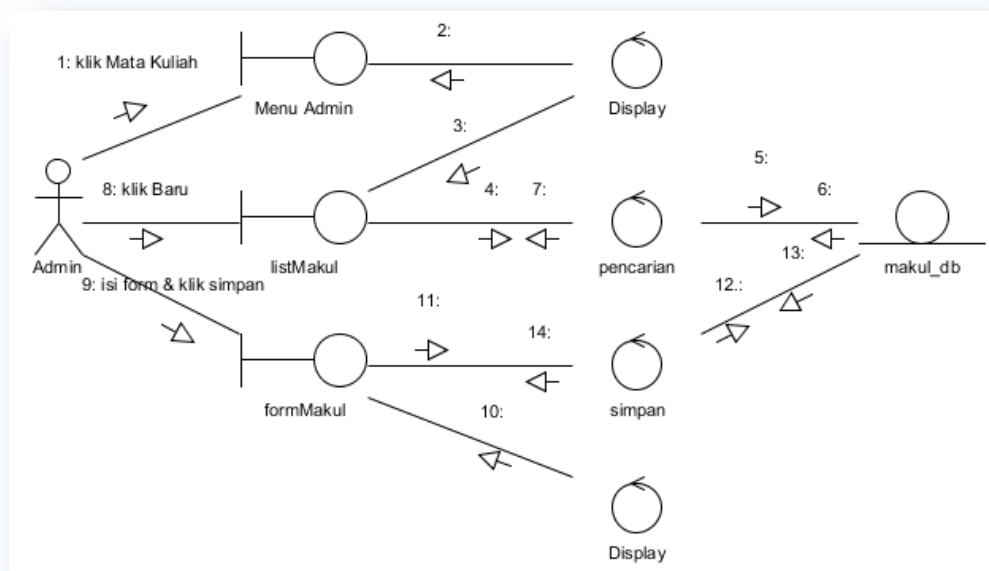
2.1.1.1: pesan “Sukses”.

Jika tidak

2.2: pesan “Pilih Program Studi”.

3: akses langsung ke listProdi.

III.3.1.6 Diagram Interaksi Use Case Kelola Mata Kuliah



Gambar III-22 Robustness Diagram – Tambah Makul

Penjelasan proses menambah makul oleh Admin.

1: klik tombol Mata Kuliah pada menu Admin.

2: menampilkan menu Admin.

3: menampilkan listMakul.

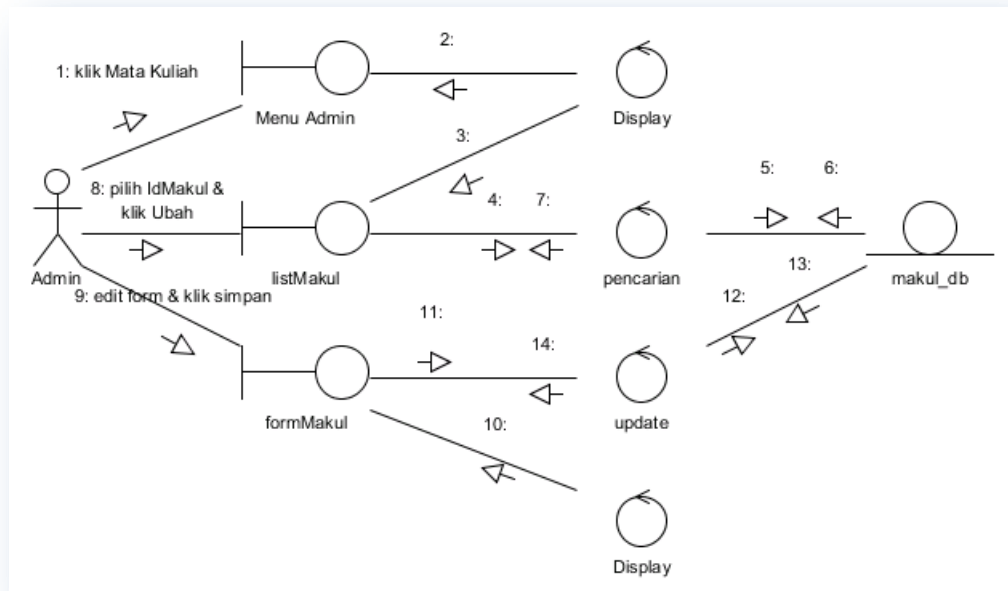
4-7: pencarian makul dan menampilkan di listMakul.

8: klik tombol Baru.

9: isi form dan klik tombol Simpan.

10: menampilkan formMakul.

11-14: proses penyimpanan ke *database* makul_db.



Gambar III-23 Robustness Diagram – Ubah Makul

Penjelasan proses ubah makul oleh Admin.

1: klik tombol Mata Kuliah pada menu Admin.

2: menampilkan menu Admin.

3: menampilkan listMakul.

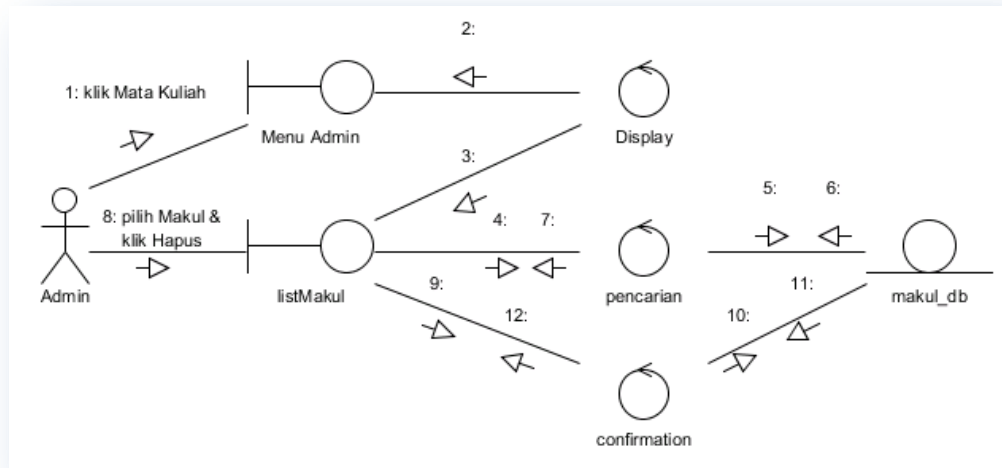
4-7: pencarian makul dan menampilkan di listMakul.

8: klik tombol Baru.

9: ubah isi form dan klik tombol Simpan.

10: menampilkan formMakul.

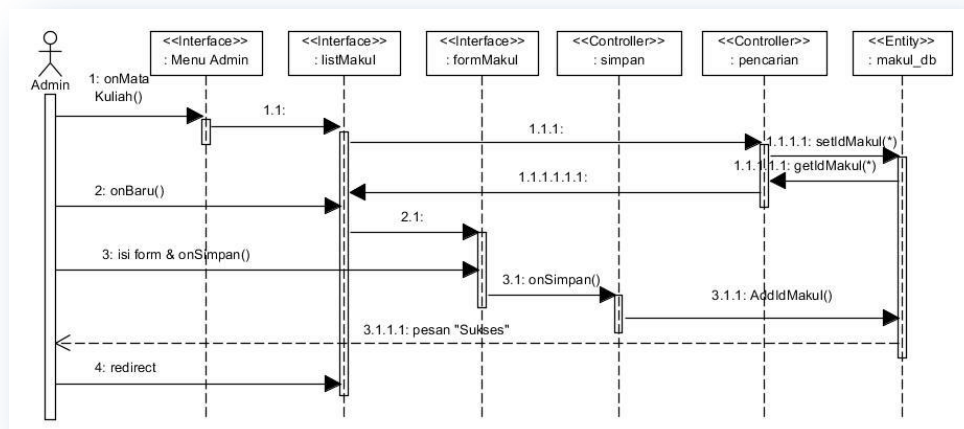
11-14: proses *update* ke *database* makul_db.



Gambar III-24 Robustness Diagram – Hapus Makul

Penjelasan proses hapus makul oleh Admin.

- 1: klik tombol Mata Kuliah pada menu Admin.
- 2: menampilkan menu Admin.
- 3: menampilkan listMakul.
- 4-7: pencarian makul dan menampilkan di listMakul.
- 8: pilih Makul dan klik tombol Hapus.
- 9-12: proses *update* condition=0 di *database* makul_db.

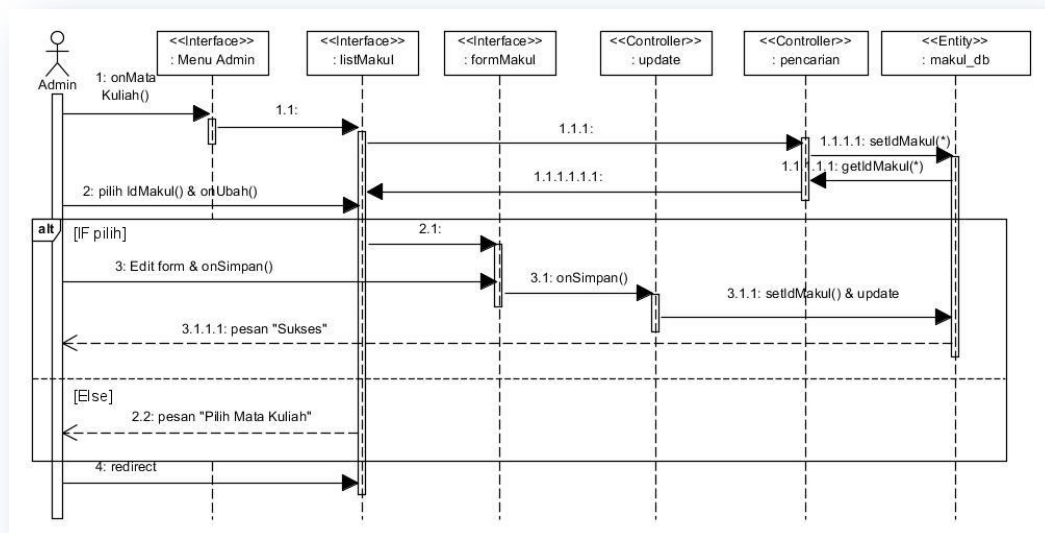


Gambar III-25 Sequence Diagram – Tambah Makul

Penjelasan proses menambah makul oleh Admin.

- 1: klik tombol Mata Kuliah pada menu Admin.

- 1.1: menampilkan listMakul.
- 1.1.1: melakukan pencarian.
- 1.1.1.1: set IdMakul ke *database* makul_db.
- 1.1.1.1.1: get IdMakul dari *database* makul_db.
- 1.1.1.1.1.1: menampilkan listMakul dari *database* makul_db.
- 2: klik tombol Baru.
- 2.1: menampilkan formMakul.
- 3: pengisian form dan klik tombol Simpan.
- 3.1: proses penyimpanan.
- 3.1.1: menambahkan ke *database* makul_db.
- 3.1.1.1: pesan “Sukses”.
- 4: akses langsung ke listMakul.



Gambar III-26 Sequence Diagram – Ubah Makul

Penjelasan proses ubah makul oleh Admin.

- 1: klik tombol Mata Kuliah pada menu Admin.
- 1.1: menampilkan listMakul.
- 1.1.1: melakukan pencarian.
- 1.1.1.1: set IdMakul ke *database* makul_db.
- 1.1.1.1.1: get IdMakul dari *database* makul_db.
- 1.1.1.1.1.1: menampilkan listMakul dari *database* makul_db.

2: pilih IdMakul dan klik tombol Ubah.

Jika pilih

2.1: melakukan pencarian.

2.1.1: set IdMakul ke *database* makul_db.

2.1.1.1: get IdMakul dari *database* makul_db.

2.1.1.1.1: menampilkan hasil ke formMakul dari *database* makul_db.

3: ubah isi form dan klik tombol Simpan.

3.1: melakukan penyimpanan dengan update data.

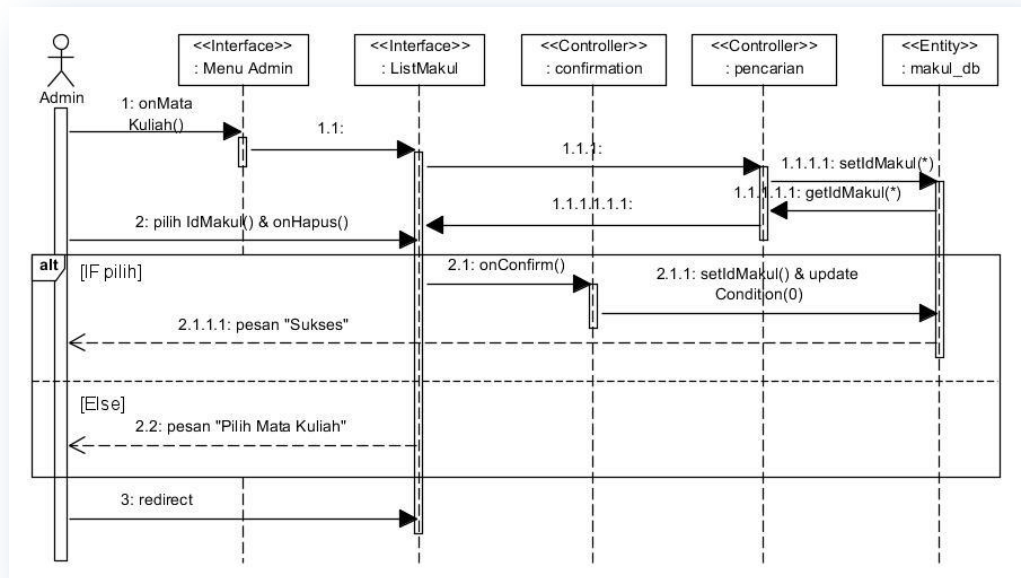
3.1.1: set IdMakul dan update *database* makul_db.

3.1.1.1: pesan “Sukses”.

Jika tidak

2.2: pesan “Pilih Mata Kuliah”.

4: akses langsung ke listMakul.



Gambar III-27 Sequence Diagram – Hapus Makul

Penjelasan proses hapus makul oleh Admin.

1: klik tombol Mata Kuliah pada menu Admin.

1.1: menampilkan listMakul.

1.1.1: melakukan pencarian.

1.1.1.1: set IdMakul ke *database* makul_db.

1.1.1.1.1: get IdMakul dari *database* makul_db.

1.1.1.1.1.1: menampilkan listMakul dari *database* makul_db.

2: pilih IdMakul dan klik tombol Hapus.

Jika pilih

2.1: melakukan konfirmasi.

2.1.1: set IdMakul dan *update condition=0* pada *database* makul_db.

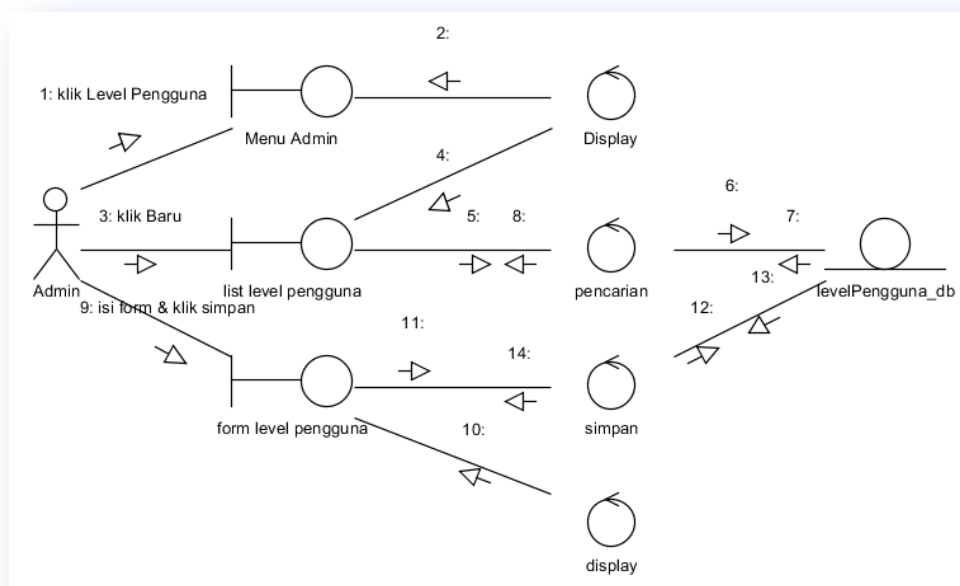
2.1.1.1: pesan “Sukses”.

Jika tidak

2.2: pesan “Pilih Mata Kuliah”.

3: akses langsung ke listMakul.

III.3.1.7 Diagram Interaksi Use Case Kelola Level Pengguna



Gambar III-28 Robustness Diagram – Tambah Level Pengguna

Penjelasan proses menambah Level Pengguna oleh Admin.

1: klik tombol Program Studi pada menu Admin.

2: menampilkan menu Admin.

3: menampilkan list Level Pengguna.

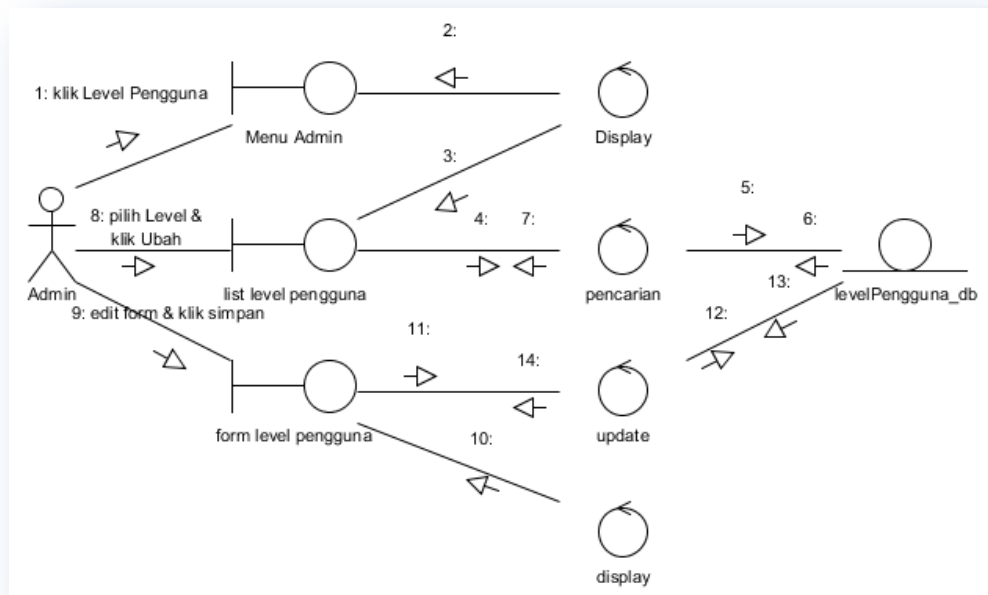
4-7: pencarian Level Pengguna dan menampilkan di list Level Pengguna.

8: klik tombol Baru.

9: isi form dan klik tombol Simpan.

10: menampilkan form Level Pengguna.

11-14: proses penyimpanan ke *database* levelPengguna_db.



Gambar III-29 Robustness Diagram – Ubah Level Pengguna

Penjelasan proses ubah Level Pengguna oleh Admin.

1: klik tombol Level Pengguna pada menu Admin.

2: menampilkan menu Admin.

3: menampilkan list Level Pengguna.

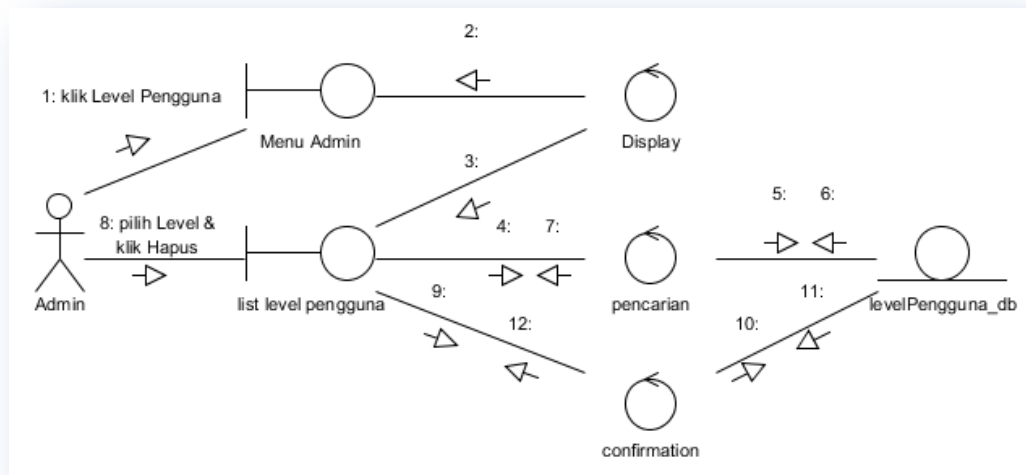
4-7: pencarian Level Pengguna dan menampilkan di list Level Pengguna.

8: klik tombol Baru.

9: ubah isi form dan klik tombol Simpan.

10: menampilkan form Level Pengguna.

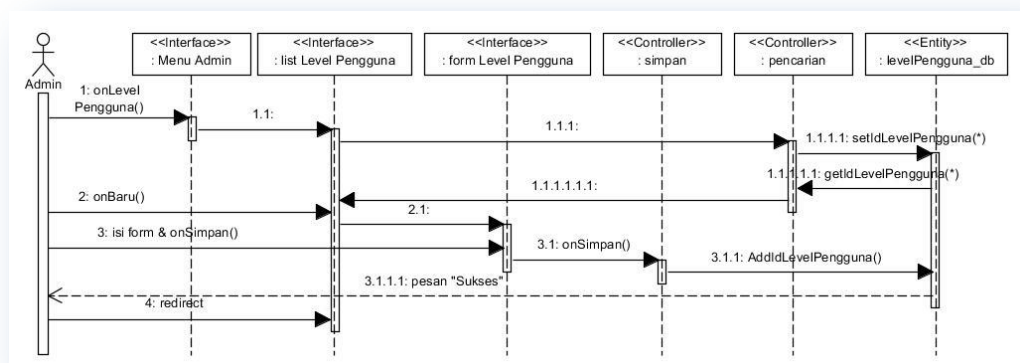
11-14: proses *update* ke *database* levelPengguna_db.



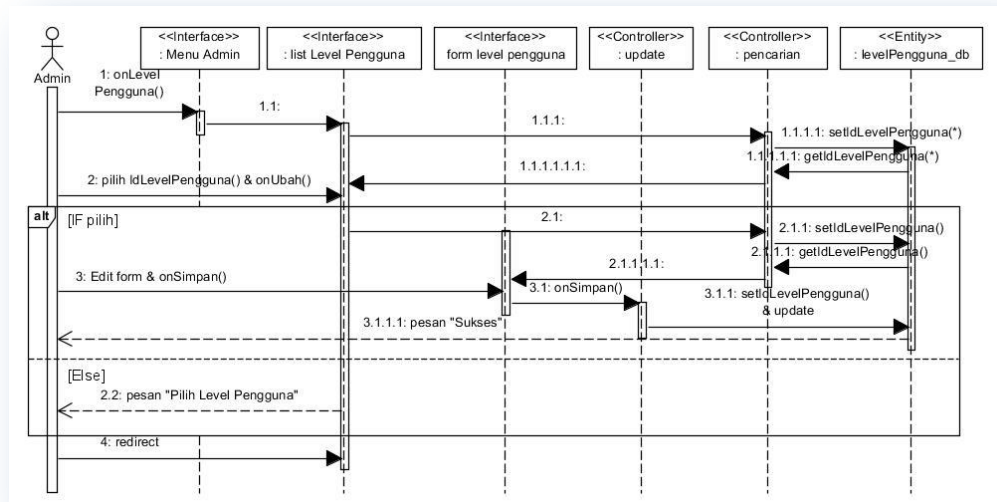
Gambar III-30 Robustness Diagram – Hapus Level Pengguna

Penjelasan proses hapus Level Pengguna oleh Admin.

- 1: klik tombol Level Pengguna pada menu Admin.
- 2: menampilkan menu Admin.
- 3: menampilkan list Level Pengguna.
- 4-7: pencarian Level Pengguna dan menampilkan di list Level Pengguna.
- 8: pilih Level Pengguna dan klik tombol Hapus.
- 9-12: proses *update condition=0* di *database levelPengguna_db*.



Gambar III-31 Sequence Diagram – Tambah Level Pengguna



Gambar III-32 Sequence Diagram – Ubah Level Pengguna

Penjelasan proses ubah level pengguna oleh Admin.

1: klik tombol Level Pengguna pada menu Admin.

1.1: menampilkan list Level Pengguna.

1.1.1: melakukan pencarian.

1.1.1.1: set IdLevelPengguna ke *database* levelPengguna_db.

1.1.1.1.1: get IdLevelPengguna dari *database* levelPengguna_db.

1.1.1.1.1.1: menampilkan list Level Pengguna dari *database* levelPengguna_db.

2: pilih IdLevelPengguna dan klik tombol Ubah.

Jika pilih

2.1: melakukan pencarian.

2.1.1: set IdLevelPengguna ke *database* levelPengguna_db.

2.1.1.1: get IdLevelPengguna dari *database* levelPengguna_db.

2.1.1.1.1: menampilkan hasil ke form Level Pengguna dari *database* levelPengguna_db.

3: ubah isi form dan klik tombol Simpan.

3.1: melakukan penyimpanan dengan update data.

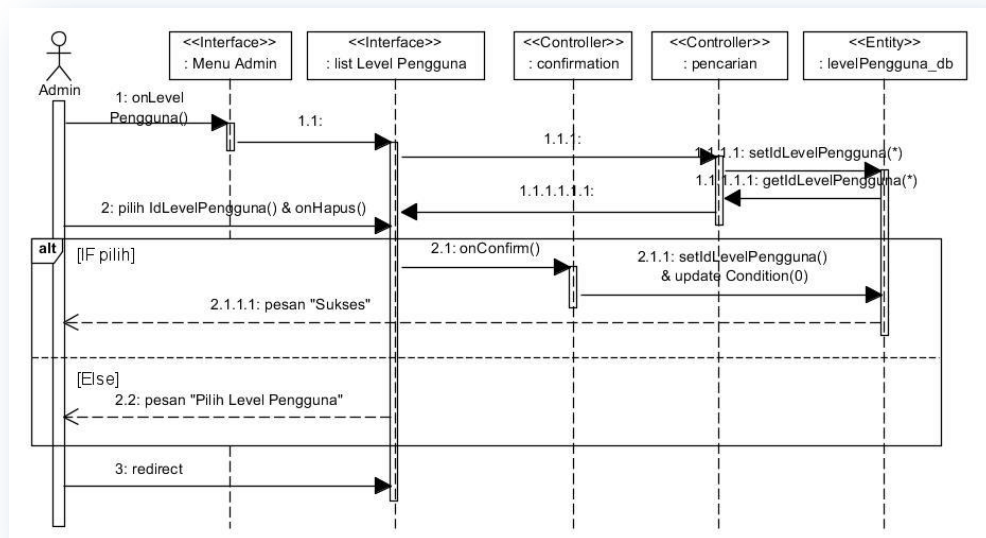
3.1.1: set IdLevelPengguna dan update *database* levelPengguna_db.

3.1.1.1: pesan “Sukses”.

Jika tidak

2.2: pesan “Pilih Level Pengguna”.

4: akses langsung ke list Level Pengguna.



Gambar III-33 Sequence Diagram – Hapus Level Pengguna

Penjelasan proses hapus level pengguna oleh Admin.

1: klik tombol Level Pengguna pada menu Admin.

1.1: menampilkan list Level Pengguna.

1.1.1: melakukan pencarian.

1.1.1.1: set IdLevelPengguna ke *database* levelPengguna_db.

1.1.1.1.1: get IdLevelPengguna dari *database* levelPengguna_db.

1.1.1.1.1.1: menampilkan list Level Pengguna dari *database* levelPengguna_db.

2: pilih IdLevelPengguna dan klik tombol Hapus.

Jika pilih

2.1: melakukan konfirmasi.

2.1.1: set IdLevelPengguna dan *update condition=0* pada *database* levelPengguna_db.

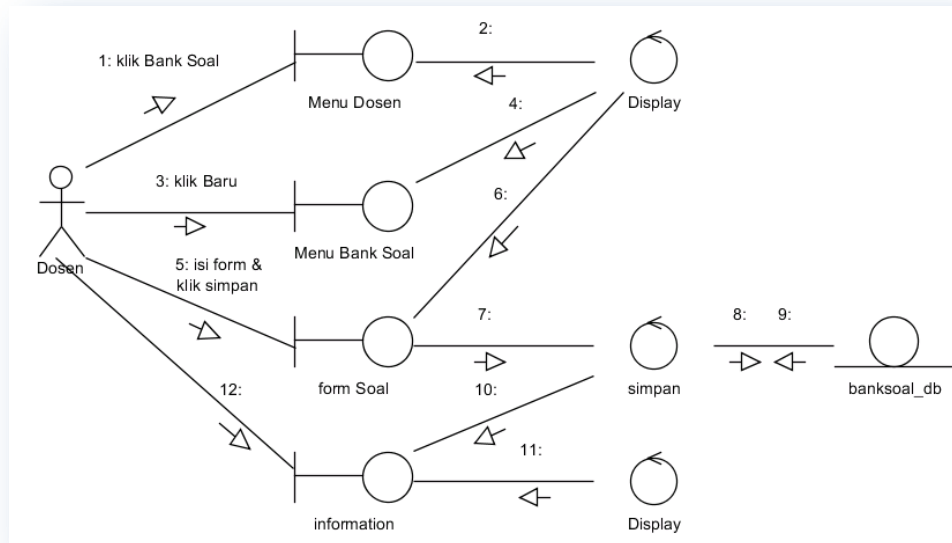
2.1.1.1: pesan “Sukses”.

Jika tidak

2.2: pesan “Pilih Level Pengguna”.

3: akses langsung ke list Level Pengguna.

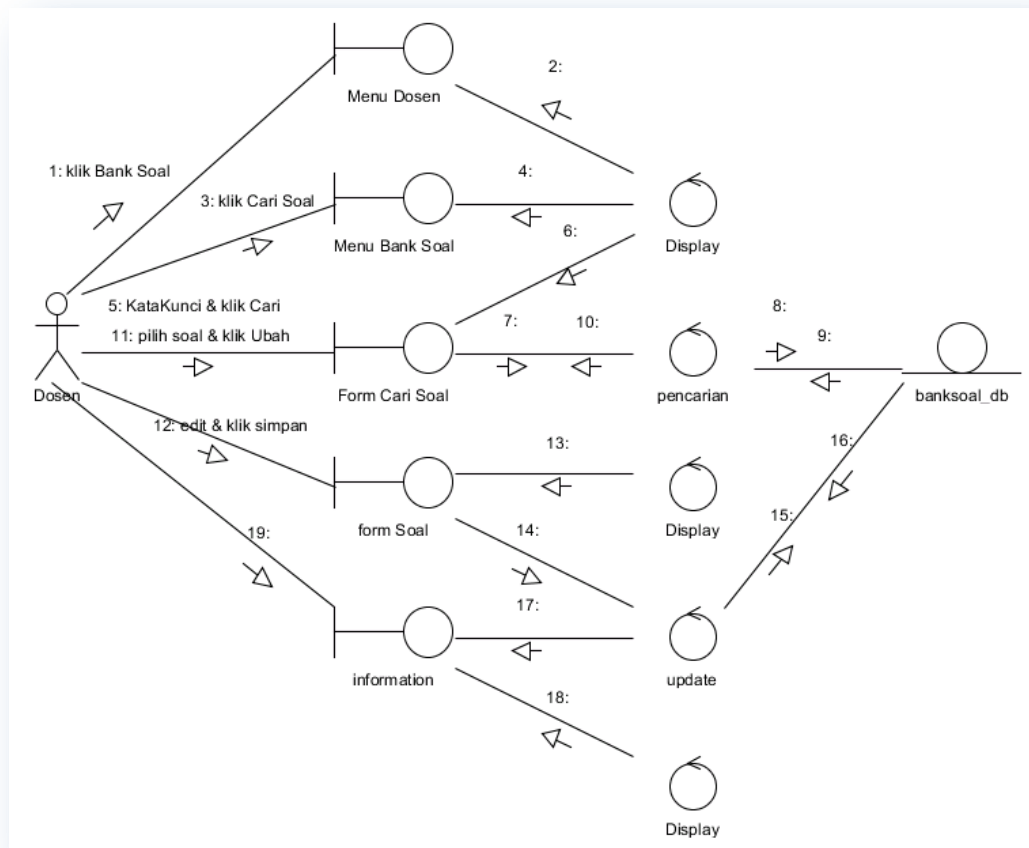
III.3.1.8 Diagram Interaksi Use Case Bank Soal



Gambar III-34 Robustness Diagram – Tambah Soal

Penjelasan proses menambah bank soal oleh Dosen.

- 1: klik tombol Bank Soal pada Menu Dosen.
- 2: menampilkan Menu Dosen.
- 3: klik tombol Baru pada Menu Bank Soal.
- 4: menampilkan Menu Bank Soal.
- 5: isi form dan klik tombol Simpan pada form Soal.
- 6: menampilkan form Soal.
- 7-10: proses penyimpanan data ke *database* banksoal_db.
- 11: menampilkan informasi.
- 12: Dosen melihat informasi sukses.



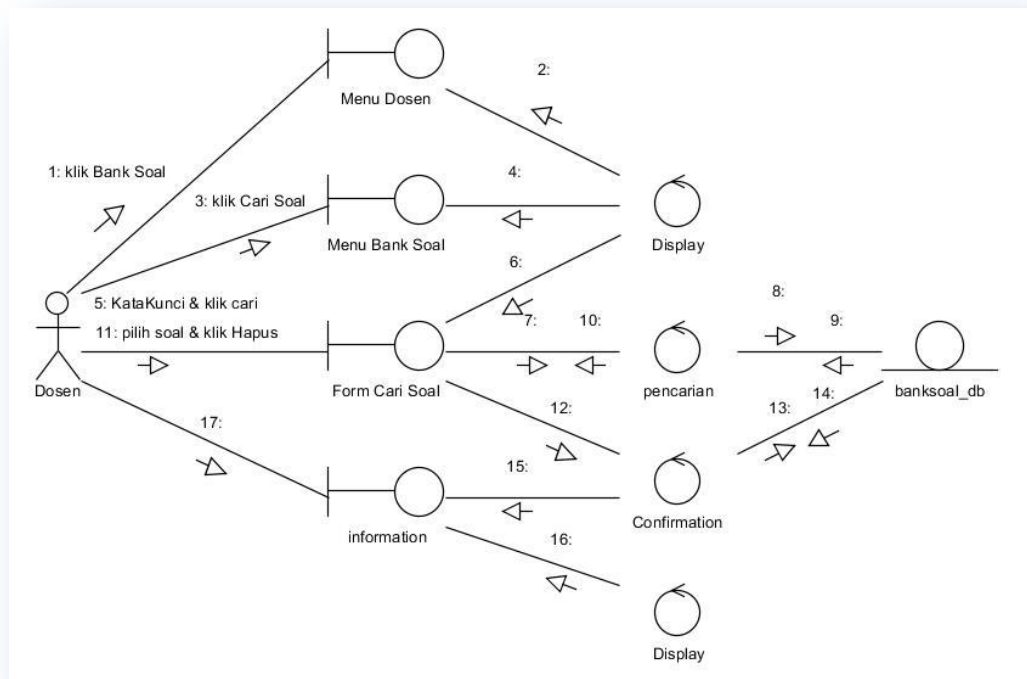
Gambar III-35 Robustness Diagram – Ubah Soal

Penjelasan proses mengubah bank soal oleh Dosen.

- 1: klik tombol Bank Soal pada Menu Dosen.
- 2: menampilkan Menu Dosen.
- 3: klik tombol Cari Soal pada Menu Bank Soal.
- 4: menampilkan Menu Bank Soal.
- 5: masukkan kata kunci dan klik tombol Cari pada Form Cari Soal.
- 6: menampilkan Form Cari Soal.
- 7-9: melakukan pencarian ke *database* banksoal_db.
- 10: menampilkan hasil pencarian ke Form Cari Soal.
- 11: memilih soal dan klik tombol Ubah.
- 12: ubah isi form dan klik tombol Simpan pada form Soal.
- 13: menampilkan Form Soal.
- 14-17: melakukan *update* ke *database* banksoal_db.

18: menampilkan informasi.

19: Dosen melihat informasi sukses.



Gambar III-36 Robustness Diagram – Hapus Soal

Penjelasan proses menghapus bank soal oleh Dosen.

1: klik tombol Bank Soal pada Menu Dosen.

2: menampilkan Menu Dosen.

3: klik tombol Cari Soal pada Menu Bank Soal.

4: menampilkan Menu Bank Soal.

5: masukkan kata kunci dan klik tombol Cari pada Form Cari Soal.

6: menampilkan Form Cari Soal.

7-9: melakukan pencarian ke *database* banksoal_db.

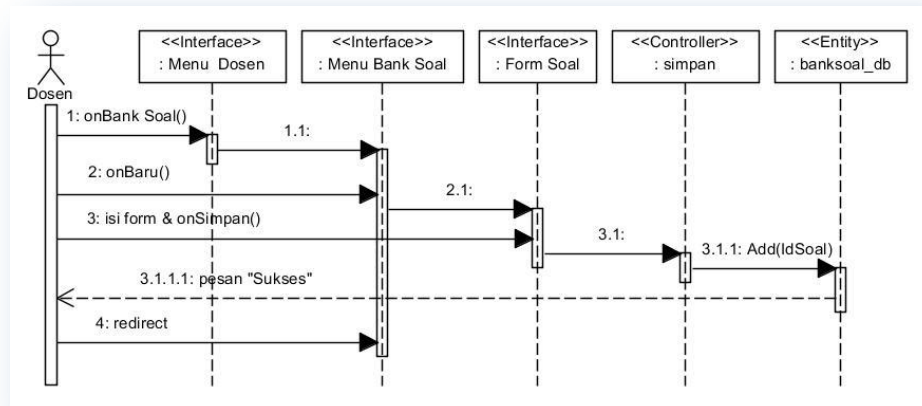
10: menampilkan hasil pencarian ke Form Cari Soal.

11: memilih soal dan klik tombol Hapus.

12-15: proses penghapusan di *database* banksoal_db.

16: menampilkan informasi.

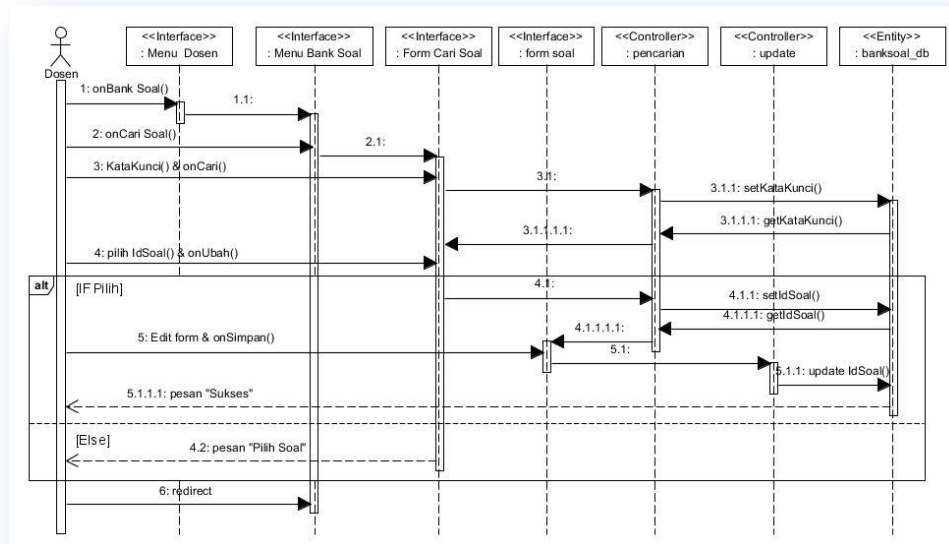
19: Dosen melihat informasi sukses.



Gambar III-37 Sequence Diagram – Tambah Bank Soal

Penjelasan proses penambahan bank soal oleh Dosen.

- 1: klik tombol Bank Soal.
- 1.1: menampilkan Menu Bank Soal.
- 2: klik tombol Baru.
- 2.1: menampilkan Form Soal.
- 3: isi form dan klik tombol Simpan.
- 3.1: proses penyimpanan soal.
- 3.1.1: Menambahkan soal ke *database* banksoal_db.
- 3.1.1.1: pesan “Sukses”.
- 4: akses langsung ke Menu Bank Soal.



Gambar III-38 Sequence Diagram – Ubah Bank Soal

Penjelasan proses perubahan bank soal oleh Dosen.

1: klik tombol Bank Soal pada Menu Dosen.

1.1: menampilkan Menu Bank Soal.

2: klik tombol Cari Soal.

2.1: menampilkan Form Cari Soal.

3: masukkan kata kunci dan klik tombol Cari.

3.1: melakukan pencarian.

3.1.1: set kata kunci ke *database* banksoal_db.

3.1.1.1: get kata kunci dari *database* banksoal_db.

3.1.1.1.1: menampilkan hasil ke Form Cari Soal.

4: pilih IdSoal dan klik tombol Ubah.

Jika pilih

4.1: melakukan pencarian.

4.1.1: set IdSoal ke *database* banksoal_db.

4.1.1.1: get IdSoal dari *database* banksoal_db.

4.1.1.1.1: menampilkan hasil ke Form Soal.

5: ubah isi form dan klik tombol Simpan.

5.1: melakukan proses *update*.

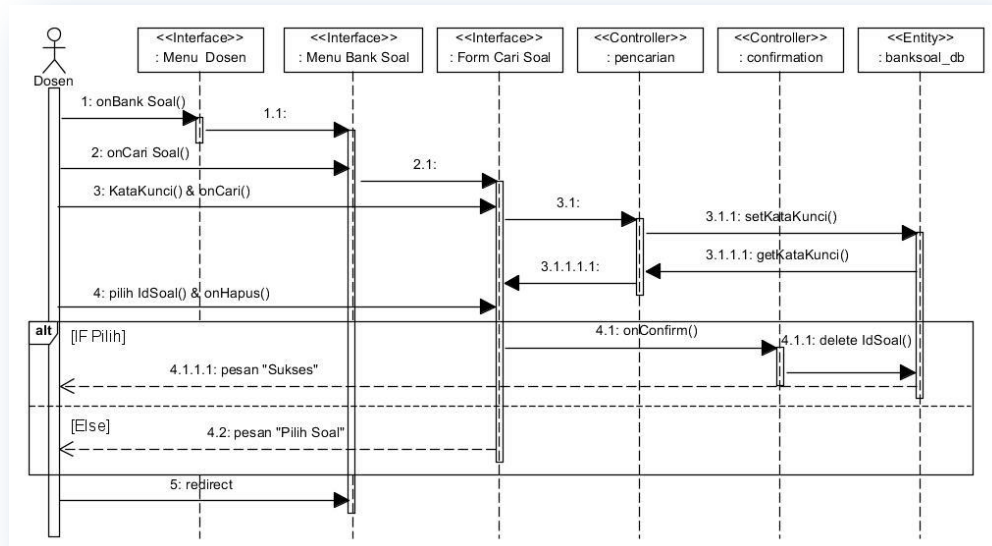
5.1.1: *update* data di *database* banksoal_db.

5.1.1.1: pesan “Sukses”.

Jika tidak

4.2: pesan “Pilih Level Pengguna”.

6: akses langsung ke Menu Bank Soal.



Gambar III-39 Sequence Diagram – Hapus Bank Soal

Penjelasan proses penghapusan bank soal oleh Dosen.

1: klik tombol Bank Soal pada Menu Dosen.

1.1: menampilkan Menu Bank Soal.

2: klik tombol Cari Soal.

2.1: menampilkan Form Cari Soal.

3: masukkan kata kunci dan klik tombol Cari.

3.1: melakukan pencarian.

3.1.1: set kata kunci ke *database* banksoal_db.

3.1.1.1: get kata kunci dari *database* banksoal_db.

3.1.1.1.1: menampilkan hasil ke Form Cari Soal.

4: pilih IdSoal dan klik tombol Hapus.

Jika pilih

4.1: melakukan konfirmasi.

4.1.1: hapus IdSoal di *database* banksoal_db.

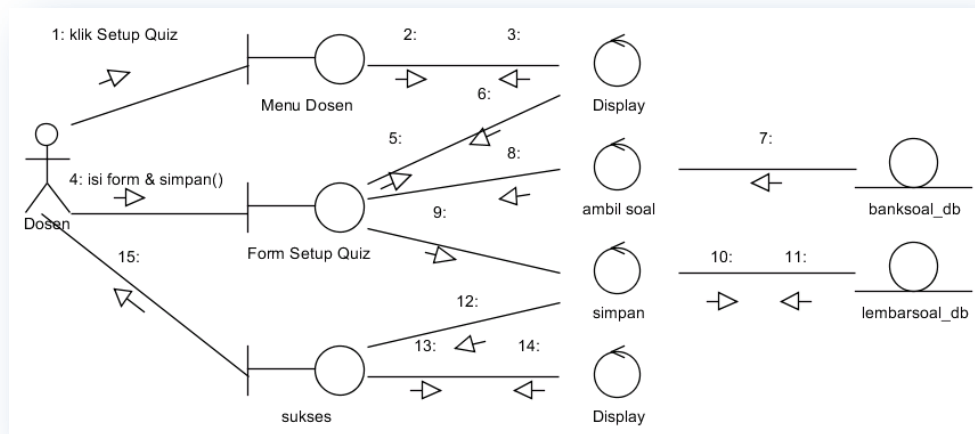
4.1.1.1: pesan “Sukses”.

Jika tidak

4.2: pesan “Pilih Level Pengguna”.

5: akses langsung ke Menu Bank Soal.

III.3.1.9 Diagram Interaksi Use Case Setup Kuis



Gambar III-40 Robustness Diagram – Setup Quiz

Penjelasan proses setup kuis oleh Dosen.

1: klik tombol Setup Quiz pada Menu Dosen.

2-3: menampilkan Menu Dosen.

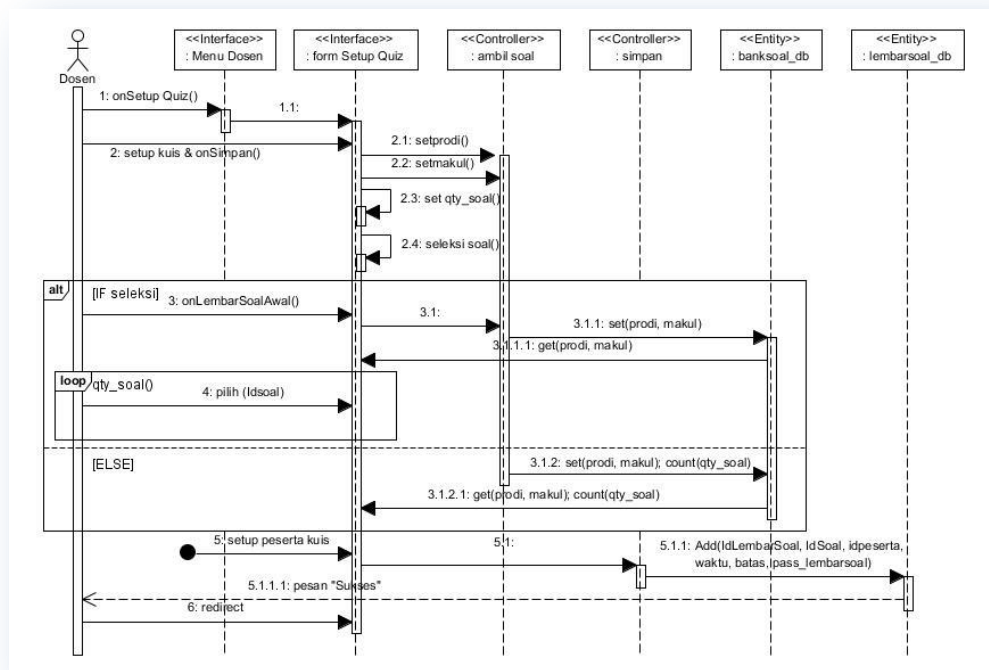
4: isi form dan klik tombol Simpan pada form Setup Quiz.

5-6: menampilkan form Setup Quiz.

7-8: mengambil list soal di *database* banksoal_db.

9-12: melakukan proses penyimpanan ke *database* lembarsoal_db.

13-15: pesan sukses.



Gambar III-41 Sequence Diagram – Setup Quiz

Penjelasan proses setup kuis oleh Dosen.

1: klik tombol Setup Quiz pada Menu Dosen.

1.1: menampilkan form Setup Quiz.

2: setup kuis dan klik tombol Simpan.

2.1: set prodi soal.

2.2: set makul soal.

2.3: set jumlah soal.

2.4: seleksi soal.

Jika seleksi

3: klik tombol Lembar Soal Awal.

3.1: melakukan pengambilan soal.

3.1.1: set prodi dan makul pada *database* banksoal_db.

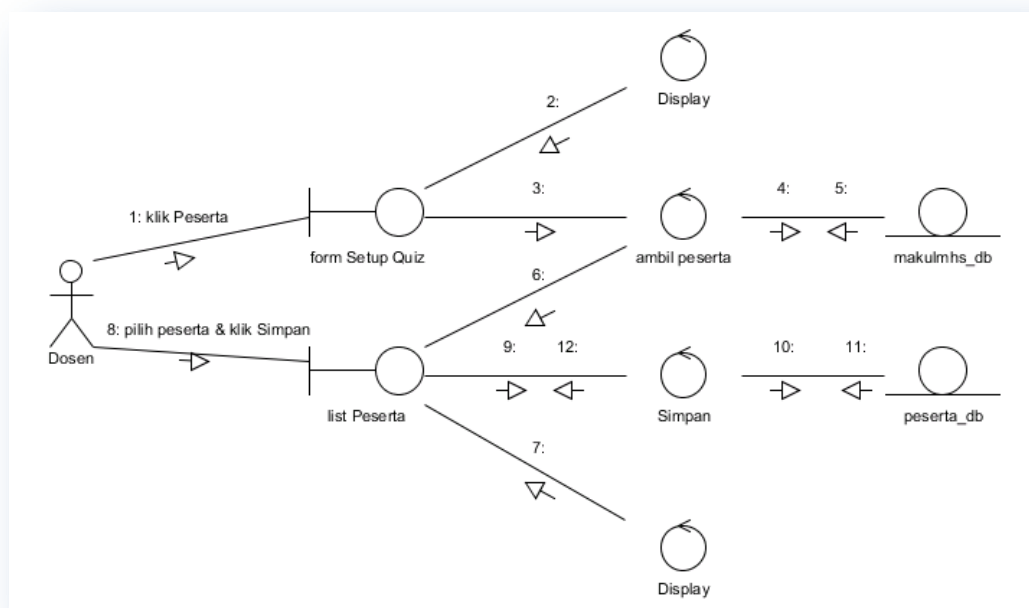
3.1.1.1: get prodi dan makul dan ditampilkan ke form Setup Quiz.

4: pilih soal sesuai jumlah soal (2.3:).

Jika tidak

- 3.1.2: set prodi, makul, dan jumlah soal pada *database* banksoal_db.
- 3.1.2.1: get prodi, makul dan jumlah soal ditampilkan ke form Setup Quiz.
- 5: melakukan setup peserta kuis.
- 5.1: melakukan penyimpanan lembar soal.
- 5.1.1: penambahan lembar soal pada *database* lembarsoal_db.
- 5.1.1.1: pesan “Sukses”.
- 6: akses langsung ke form Setup Quiz.

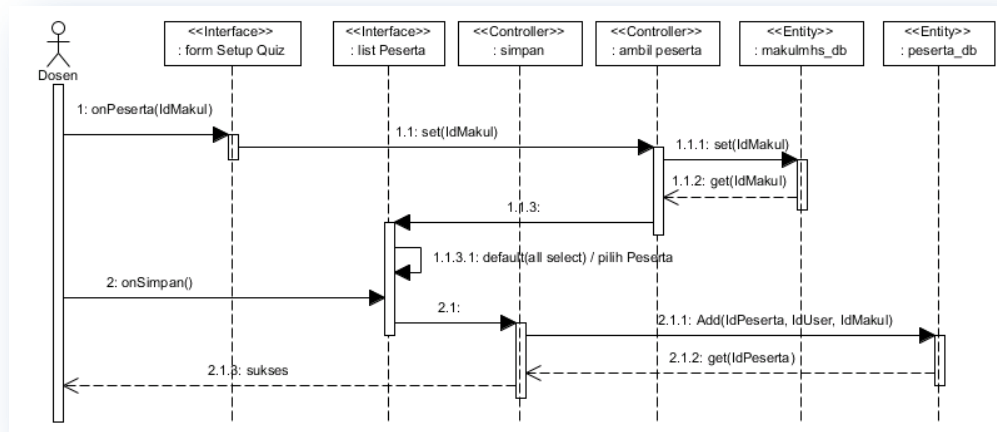
III.3.1.10 Diagram Interaksi Use Case Daftar Peserta Kuis



Gambar III-42 Robustness Diagram – Peserta Kuis

Penjelasan prose setup peserta kuis oleh Dosen.

- 1: klik tombol Peserta pada form Setup Quiz
- 2: menampilkan form Setup Quiz.
- 3-5: mengambil data peserta sesuai makulnya di *database* makulmhs_db.
- 6: menampilkan list mahasiswa ke list Peserta.
- 7: menampilkan list Peserta.
- 8: pilih peserta dan klik tombol Simpan.
- 9-12: proses penyimpanan peserta kuis ke *database* peserta_db.



Gambar III-43 Sequence Diagram – Peserta Kuis

Penjelasan prose setup peserta kuis oleh Dosen.

1: klik Peserta pada form Setup Quiz.

1.1: proses pengambilan data peserta.

1.1.1: set IdMakul pada *database* makulmhs_db.

1.1.2: get IdMakul dari *database* makulmhs_db.

1.1.3: menampilkan hasil pencarian ke list Peserta.

1.1.3.1: pilih peserta.

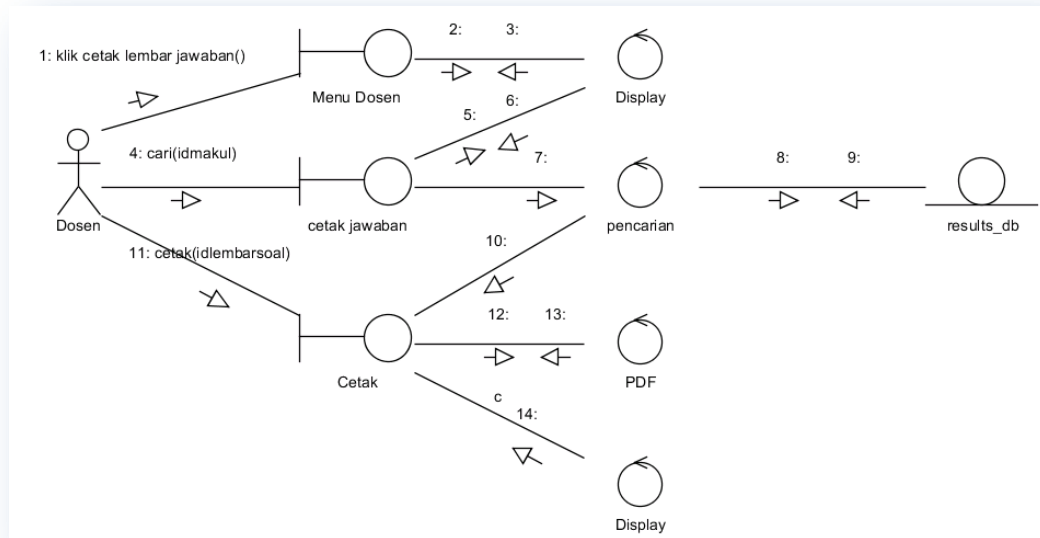
2: klik tombol Simpan.

2.1: proses penyimpanan.

2.1.1: menambahkan peserta kuis ke *database* peserta_db.

2.1.2-2.1.3: pesan “Sukses”.

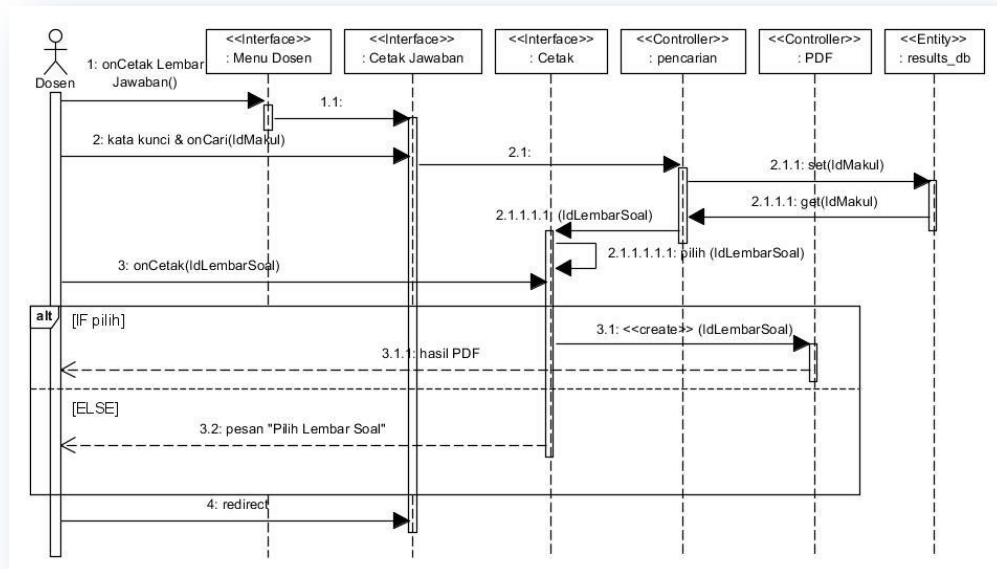
III.3.1.11 Diagram Interaksi Use Case Cetak Lembar Soal dan Jawaban



Gambar III-44 Robustness Diagram – Cetak Lembar Jawaban

Penjelasan proses cetak lembar jawaban oleh Dosen.

1. klik tombol Cetak Lembar Jawaban pada Menu Dosen.
- 2-3: menampilkan Menu Dosen.
- 4: masukkan kata kunci dan klik tombol Cari.
- 5-6: menampilkan Cetak Jawaban.
- 7-9: pencarian kata kunci di *database* results_db.
- 10: menampilkan hasil pencarian.
- 11: pilih IdLemabarSoal dan klik tombol Cetak.
- 12-13: melakukan pencetakan ke PDF.
- 14: menampilkan menu Cetak.



Gambar III-45 Sequence Diagram – Cetak Lembar Jawaban per Mata kuliah

Penjelasan proses cetak lembar jawaban per mata kuliah oleh Dosen.

1: klik tombol Cetak Lembar Jawaban pada Menu Dosen.

1.1: menampilkan Cetak Jawaban.

2: isi form dan klik tombol Cari.

2.1: melakukan pencarian.

2.1.1: set IdMakul ke *database* results_db.

2.1.1: get IdMakul dari *database* results_db.

2.1.1.1: menampilkan hasil ke menu Cetak.

2.1.1.1.1: pilih IdLembarSoal.

3: klik tombol Cetak.

Jika pilih

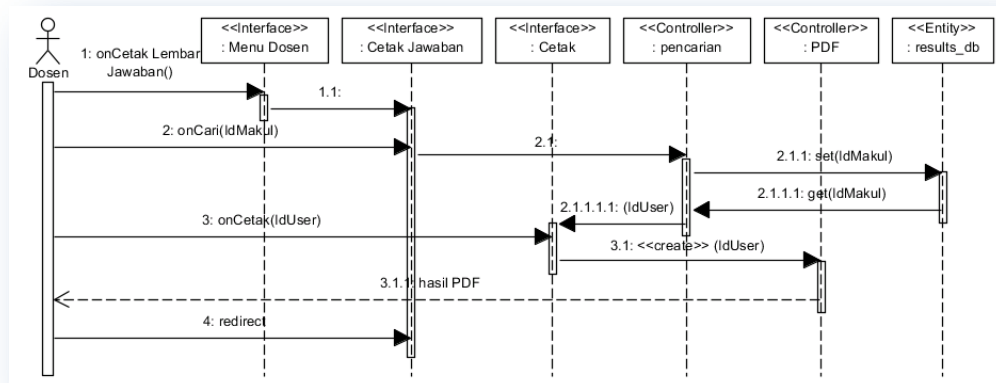
3.1: proses pembuatan file PDF.

3.1.1: hasil cetakan PDF.

Jika tidak

3.2: pesan “Pilih Lembar Soal”.

4: akses langsung ke Cetak Jawaban.



Gambar III-46 Sequence Diagram – Cetak Lembar Jawaban per Mahasiswa

Penjelasan proses cetak lembar jawaban per mahasiswa oleh Dosen.

1: klik tombol Cetak Lembar Jawaban pada Menu Dosen.

1.1: menampilkan Cetak Jawaban.

2: isi form dan klik tombol Cari.

2.1: melakukan pencarian.

2.1.1: set IdMakul ke *database* results_db.

2.1.1: get IdMakul dari *database* results_db.

2.1.1.1: menampilkan hasil ke menu Cetak.

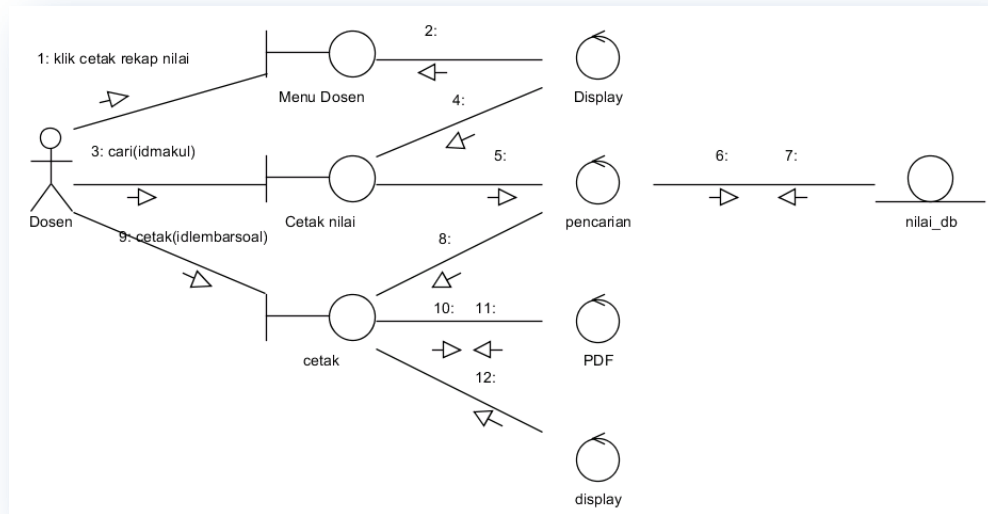
3: pilih IdUser dan klik tombol Cetak.

3.1: proses pembuatan file PDF.

3.1.1: hasil cetakan PDF.

4: akses langsung ke Cetak Jawaban.

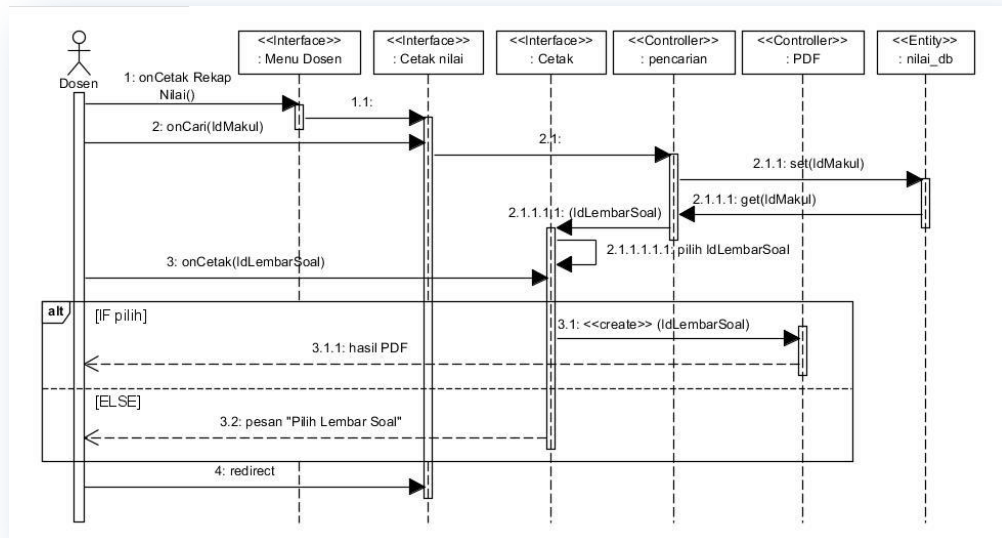
III.3.1.12 Diagram Interaksi Use Case Cetak Rekap Nilai



Gambar III-47 Robustness Diagram – Cetak Rekap Nilai

Penjelasan proses cetak rekap nilai per lembar jawaban oleh Dosen.

1. klik tombol Cetak rekap nilai pada Menu Dosen.
2. menampilkan Menu Dosen.
3. masukkan kata kunci dan klik tombol Cari.
4. menampilkan Cetak Jawaban.
- 5-7: pencarian kata kunci di *database* nilai_db.
- 8: menampilkan hasil pencarian.
- 9: pilih IdLemabarSoal dan klik tombol Cetak.
- 10-11: melakukan pencetakan ke PDF.
- 12: menampilkan menu Cetak.



Gambar III-48 Sequence Diagram – Cetak Rekap Nilai per Lembar Soal

Penjelasan proses cetak rekap nilai per lembar soal oleh Dosen.

1: klik tombol Cetak Rekap Nilai pada Menu Dosen.

1.1: menampilkan Cetak Nilai.

2: isi form dan klik tombol Cari.

2.1: melakukan pencarian.

2.1.1: set IdMakul ke *database* nilai_db.

2.1.1: get IdMakul dari *database* nilai_db.

2.1.1.1: menampilkan hasil ke menu Cetak.

2.1.1.1.1: pilih IdLembarSoal.

3: klik tombol Cetak.

Jika pilih

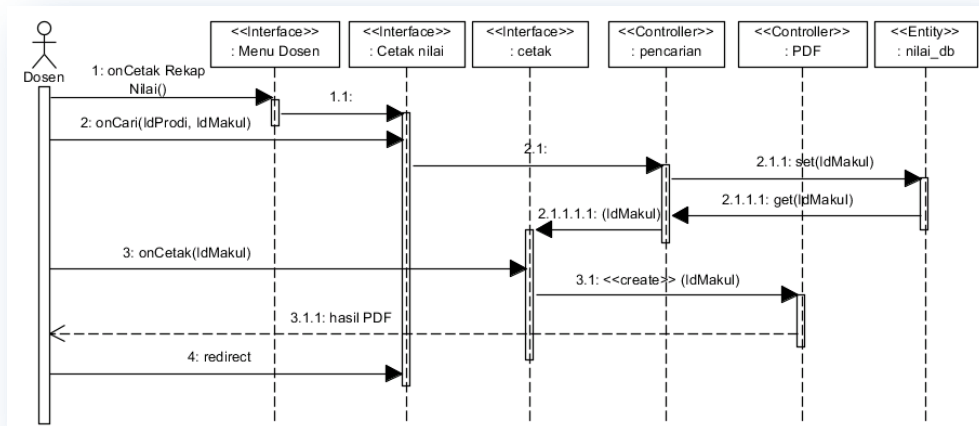
3.1: proses pembuatan file PDF.

3.1.1: hasil cetakan PDF.

Jika tidak

3.2: pesan “Pilih Lembar Soal”.

4: akses langsung ke Cetak Nilai.



Gambar III-49 Sequence Diagram – Cetak Rekap Nilai per Mata Kuliah

Penjelasan proses cetak rekap nilai per mata kuliah oleh Dosen.

1: klik tombol Cetak Rekap Nilai pada Menu Dosen.

1.1: menampilkan Cetak Jawaban.

2: isi form dan klik tombol Cari.

2.1: melakukan pencarian.

2.1.1: set IdMakul ke *database* nilai_db.

2.1.1: get IdMakul dari *database* nilai_db.

2.1.1.1: menampilkan hasil ke menu Cetak.

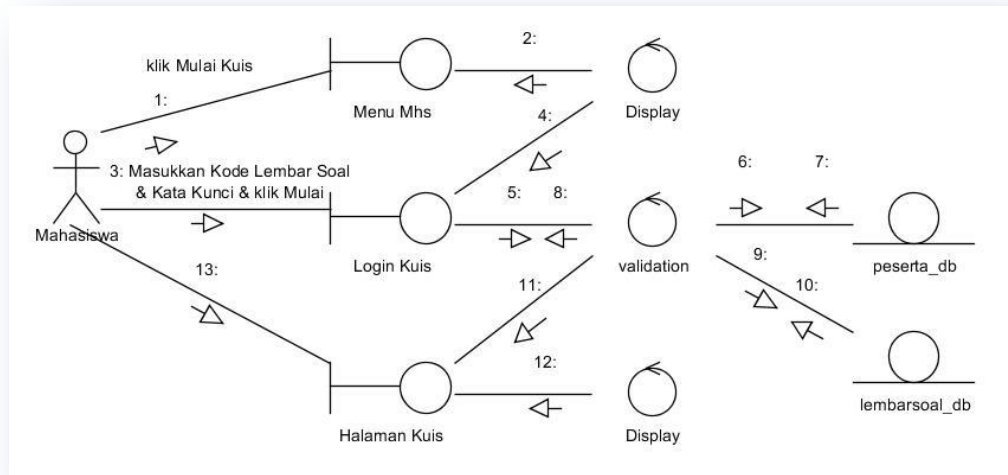
3: pilih IdMakul dan klik tombol Cetak.

3.1: proses pembuatan file PDF.

3.1.1: hasil cetakan PDF.

4: akses langsung ke Cetak Nilai.

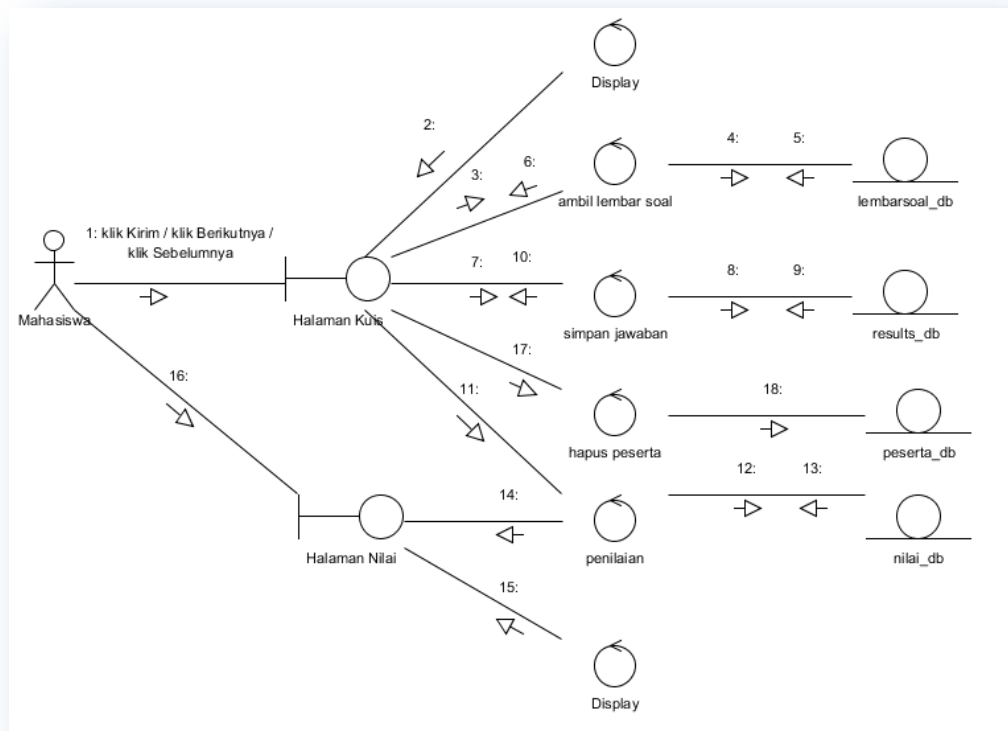
III.3.1.13 Diagram Interaksi Use Case Mengikuti Kuis



Gambar III-50 Robustness Diagram – Login Kuis

Penjelasan proses mengikuti kuis – Login Kuis oleh Mahasiswa.

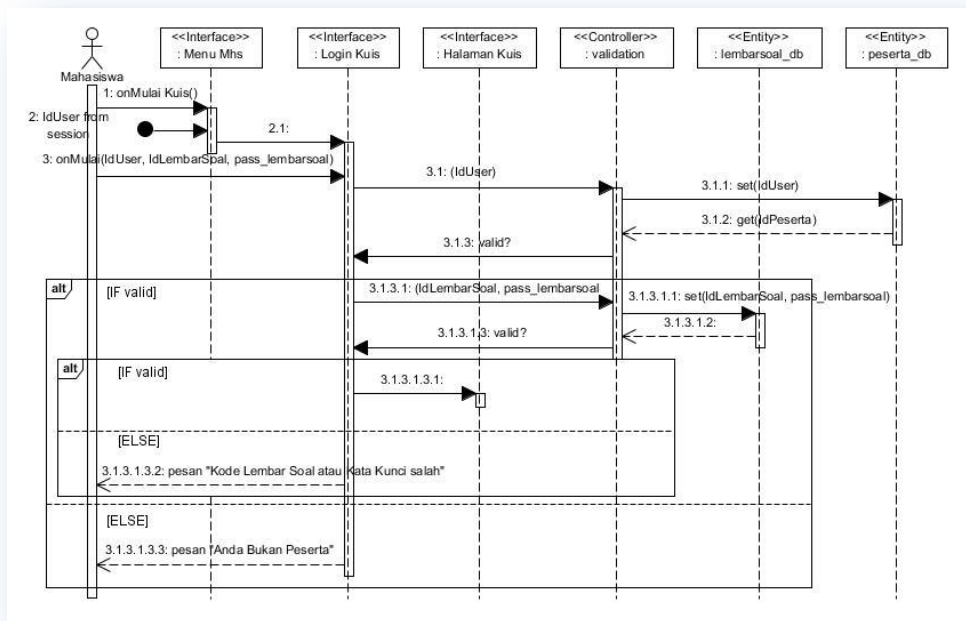
- 1: klik tombol Mulai Kuis pada Menu Mhs.
- 2: menampilkan Menu Mhs.
- 3: masukkan Kode Lembar Soal dan Kata Kunci dan klik tombol Mulai.
- 4: menampilkan Login Kuis.
- 5-10: validasi ke *database* peserta_db dan lembarsoal_db.
- 11: menampilkan soal di Halaman Kuis.
- 12: menampilkan Halaman Kuis.
- 13: memulai kuis.



Gambar III-51 Robustness Diagram – Mengikuti Kuis

Penjelasan proses mengikuti kuis oleh Mahasiswa.

- 1: klik tombol Kirim/Berikutnya/Sebelumnya pada Halaman Kuis.
- 2: menampilkan Halaman Kuis.
- 3-6: mengambil lembar soal dari *database* lembarsoal_db.
- 7-10: menyimpan jawaban ke *database* results_db.
- 11-13: proses penilain dan disimpan di *database* nilai_db.
- 14: menampilkan penilain ke Halaman Nilai.
- 15: menampilkan Halaman Nilai.
- 16: melihat nilai hasil kuis.



Gambar III-52 Sequence Diagram – Login Kuis

Penjelasan proses mengikuti kuis – Login Kuis oleh Mahasiswa.

1: klik tombol Mulai Kuis.

2: mengambil *session* IdUser.

3: masukkan IdLembarSoal dan kata kunci dan klik tombol Mulai.

3.1: melakukan validasi peserta kuis.

3.1.1: set IdUser ke *database* peserta_db.

3.1.2: get IdUser data *database* peserta_db.

3.1.3: hasil validasi.

Jika valid

3.1.3.1: melakukan validasi IdLembarSoal dan kata kunci.

3.1.3.1.1: set IdLembarSoal dan pass_lembarsoal ke *database* lembarsoal_db.

3.1.3.1.2: get IdLembarSoal dan pass_lembarsoal dari *database* lembarsoal_db.

3.1.3.1.3: hasil validasi.

Jika valid

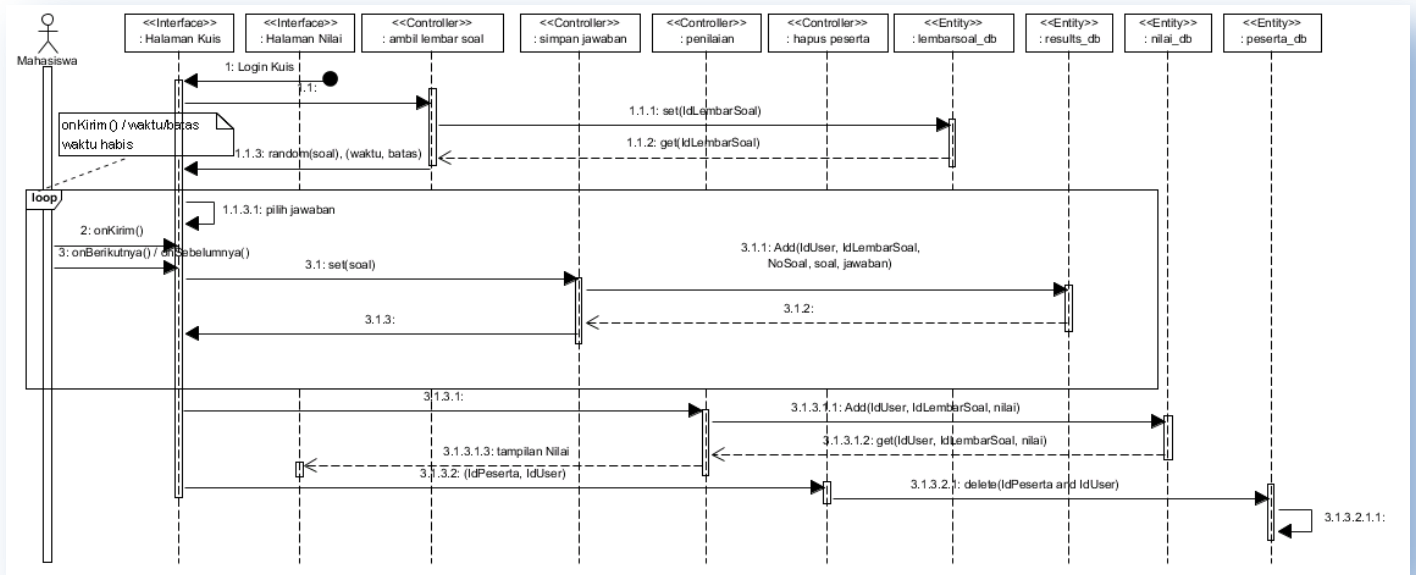
3.1.3.1.3.1: menampilkan Halaman Kuis.

Jika tidak

3.1.3.1.3.2: pesan “Kode Lembar Soal atau Kata Kunci salah”.

Jika tidak

3.1.3.1.3.3: pesan “Anda Bukan Peserta”.



Gambar III-53 Sequence Diagram – Mengikuti Kuis

Penjelasan proses mengikuti kuis oleh Mahasiswa.

1: melakukan login kuis.

1.1: proses pengambilan lembar soal.

1.1.1: set IdLembarSoal ke *database* lembarsoal_db.

1.1.2: get IdLembarSoal dari *database* lembarsoal_db.

1.1.3: menampilkan *random* kuis dan *timer*.

Looping sampai klik tombol Kirim/waktu habis.

1.1.3.1: memilih jawaban.

2-3: klik tombol Kirim/Berikutnya/Sebelumnya.

3.1: proses penyimpanan hasil kuis.

3.1.1: menambahkan hasil kuis ke *database* results_db.

3.1.2-3.1.3: melakukan perulangan sesuai jumlah kuis.

Waktu habis

3.1.3.1: proses penilain kuis.

3.1.3.1.1: menambahkan hasil kuis ke *database* results_db.

3.1.3.1.2: get hasil kuis ke *database* results_db.

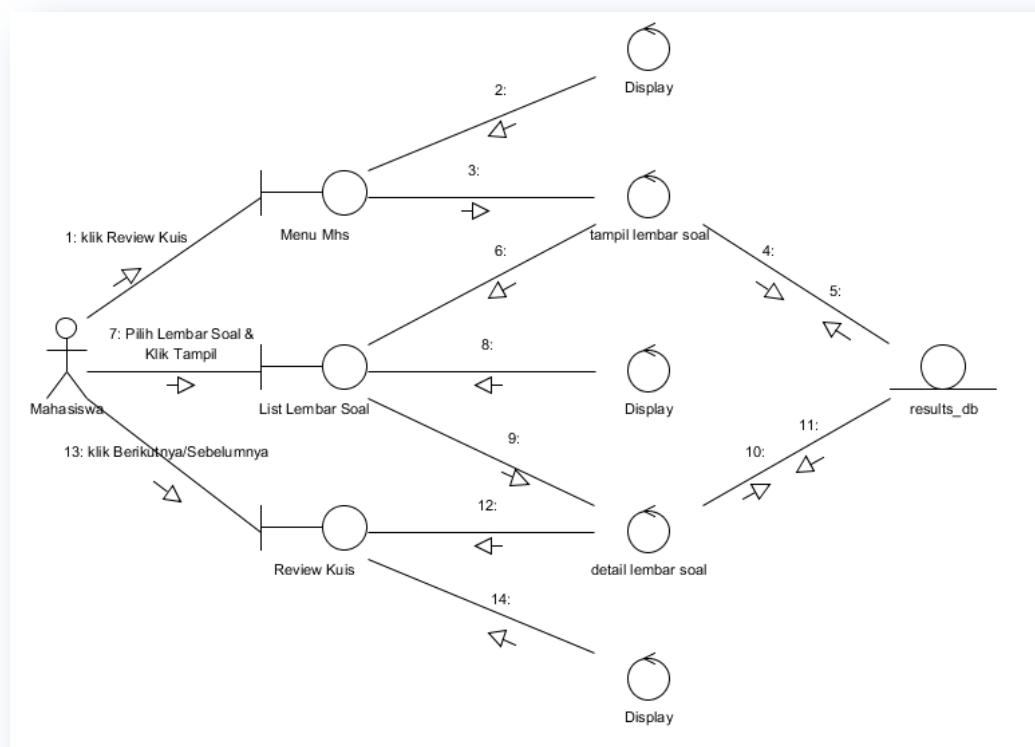
3.1.3.1.3: menampilkan nilai hasil kuis.

3.1.3.2: melakukan penghapusan peserta kuis.

3.1.3.2.1: penghapusan peserta kuis di *database* peserta_db.

3.1.3.2.1.1: hapus data.

III.3.1.14 Diagram Interaksi Use Case Review Kuis



Gambar III-54 - Robustness Diagram – Review Kuis

Penjelasan proses *review* kuis oleh Mahasiswa.

1: klik tombol Reviess Kuis pada Menu Mhs.

2: menampilkan Menu Mhs.

3-5: proses pengambilan data lembar soal pada *database* results_db.

6: menampilkan hasil ke list lembar soal.

7: pilih lembar soal dan klik tombol Tampil.

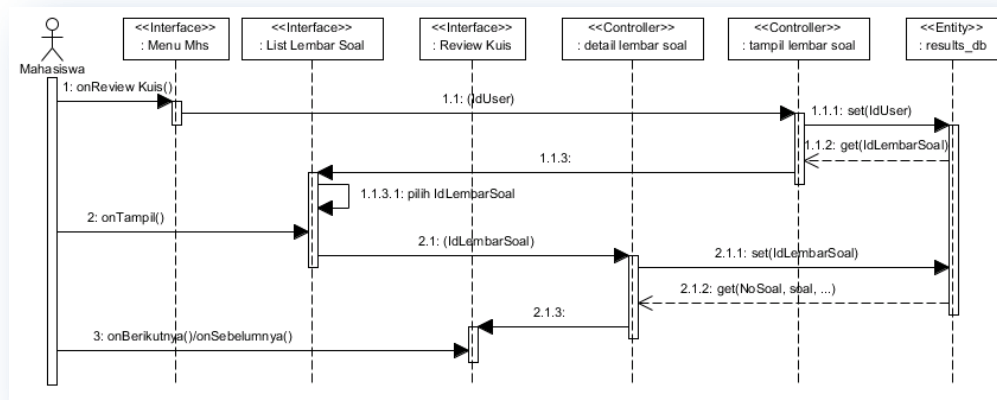
8: menampilkan List Lembar Soal.

9: proses pemngambilan detail lembar soal pada *database* results_db.

12: menampilkan hasil ke Review Kuis.

13: menampilkan Review Kuis.

14: klik tombol Berikutnya/Sebelumnya untuk melihat kuis.



Gambar III-55 Sequence Diagram – Review Kuis

Penjelasan proses *review* kuis oleh Mahasiswa.

1: klik tombol Review Kuis.

1.1: proses menampilkan lembar soal sesuai IdUser.

1.1.1: set IdUser ke *database* results_db.

1.1.2: get IdLembarSoal dari *database* results_db.

1.1.3: menampilkan list lembar soal.

1.1.3.1: pilih IdLembarSoal.

2: klik tombol Tampil.

2.1: proses menampilkan detail lembar kuis.

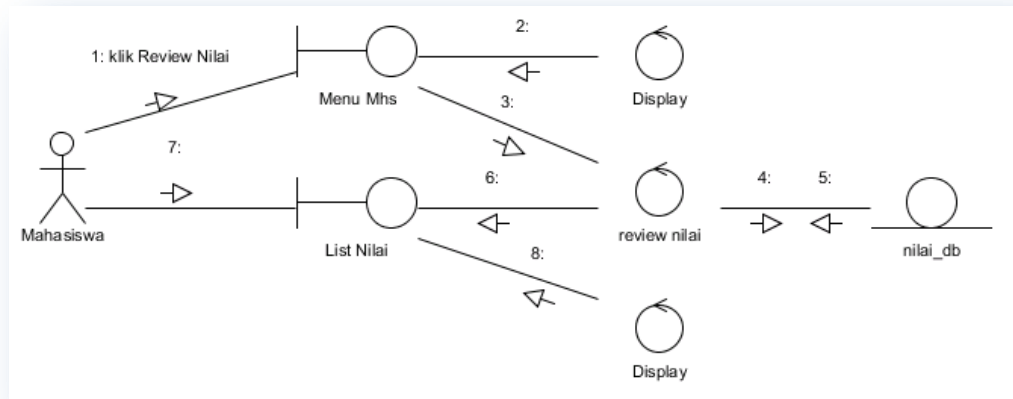
2.1.1: set IdLembarSoal ke *database* results_db.

2.1.2: get IdLembarSoal dari *database* results_db.

2.1.3: menampilkan detail hasil kuis.

3: klik tombol Berikutnya/Sebelumnya untuk melihat kuis.

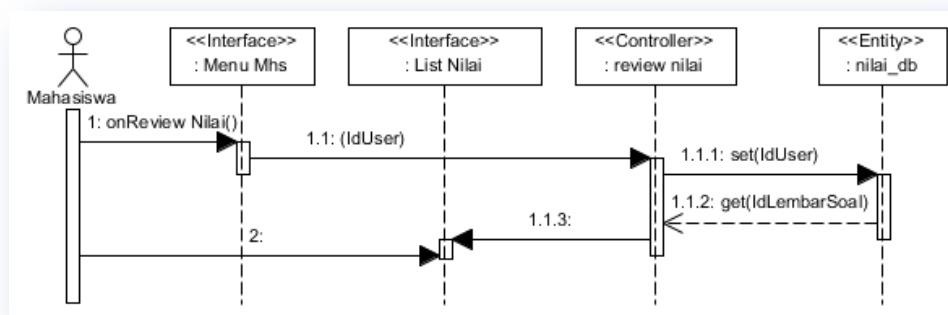
III.3.1.15 Diagram Interkasi *Use Case Review Nilai*



Gambar III-56 Robustness Diagram – Review Nilai

Penjelasan proses *review* nilai oleh Mahasiswa.

- 1: klik tombol Review Nilai pada Menu Mhs.
- 2: menampilkan Menu Mhs.
- 3-5: proses *review* nilai dari *database* nilai_db.
- 6: menampilkan hasil ke List Nilai.
- 7: melihat hasil nilai.
- 8: menampilkan List Nilai.



Gambar III-57 Sequence Diagram – Review Nilai

Penjelasan proses *review* nilai oleh Mahasiswa.

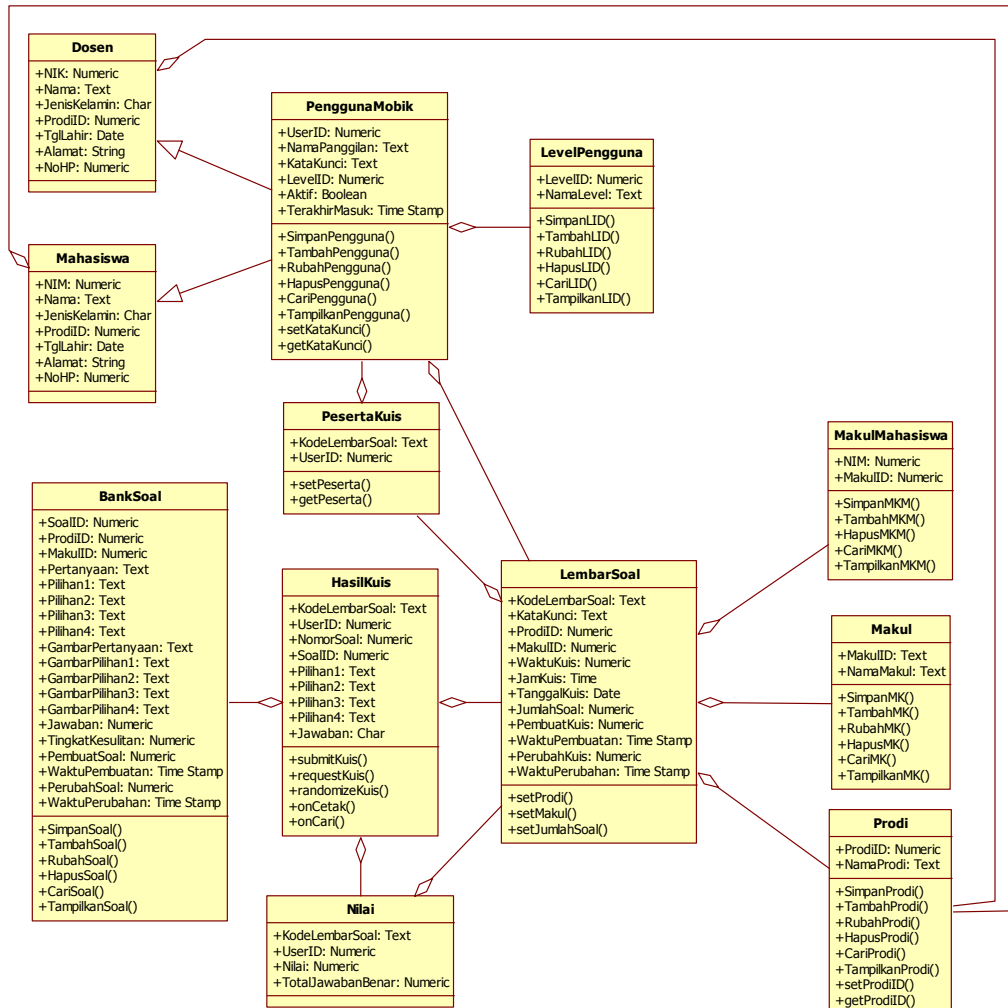
- 1: klik tombol Review Nilai pada Menu Mhs.
- 1.1: proses *review* nilai sesuai IdUser.
- 1.1.1: set IdUser ke *database* nilai_db.

1.1.2: get IdUser dari *database* nilai_db.

1.1.3: menampilkan hasil ke List Nilai.

2: melihat hasil nilai.

III.3.2 Diagram Kelas



Gambar III-58 Diagram Kelas Mobik

III.3.3 Rancangan Kelas Rinci

Rancangan rinci setiap kelas disajikan pada tabel basis data

III.3.3.1 Kelas Kelola Pengguna

-

| PenggunaMobik |
|---|
| +UserID: Numeric +NamaPanggilan: Text +KataKunci: Text +LevelID: Numeric +Aktif: Boolean +TerakhirMasuk: Time Stamp |
| +SimpanPengguna() +TambahPengguna() +RubahPengguna() +HapusPengguna() +CariPengguna() +TampilkanPengguna() +setKataKunci() +getKataKunci() |

III.3.3.2 Kelas Kelola Program Studi

-

| Prodi |
|---|
| +ProdiID: Numeric +NamaProdi: Text |
| +SimpanProdi() +TambahProdi() +RubahProdi() +HapusProdi() +CariProdi() +TampilkanProdi() +setProdiID() +getProdiID() |

III.3.3.3 Kelas Kelola Mata Kuliah

-

| Makul |
|---|
| +MakulID: Text +NamaMakul: Text |
| +SimpanMK() +TambahMK() +RubahMK() +HapusMK() +CariMK() +TampilkanMK() |

III.3.3.4 Kelas Kelola Level Pengguna

-

| LevelPengguna |
|---|
| +LevelID: Numeric +NamaLevel: Text |
| +SimpanLID() +TambahLID() +RubahLID() +HapusLID() +CariLID() +TampilkanLID() |

III.3.3.5 Kelas Bank Soal

-

| BankSoal |
|--|
| +SoalID: Numeric +ProdiID: Numeric +MakuID: Numeric +Pertanyaan: Text +Pilihan1: Text +Pilihan2: Text +Pilihan3: Text +Pilihan4: Text +GambarPertanyaan: Text +GambarPilihan1: Text +GambarPilihan2: Text +GambarPilihan3: Text +GambarPilihan4: Text +Jawaban: Numeric +TingkatKesulitan: Numeric +PembuatSoal: Numeric +WaktuPembuatan: Time Stamp +PerubahSoal: Numeric +WaktuPerubahan: Time Stamp |
| +SimpanSoal() +TambahSoal() +RubahSoal() +HapusSoal() +CariSoal() +TampilkanSoal() |

III.3.3.6 Kelas Setup Kuis

-

| LembarSoal |
|--|
| +KodeLembarSoal: Text +KataKunci: Text +ProdiID: Numeric +MakulID: Numeric +WaktuKuis: Numeric +JamKuis: Time +TanggalKuis: Date +JumlahSoal: Numeric +PembuatKuis: Numeric +WaktuPembuatan: Time Stamp +PerubahKuis: Numeric +WaktuPerubahan: Time Stamp +setProdi() +setMakul() +setJumlahSoal() |

III.3.3.7 Kelas Daftar Peserta Kuis

-

| PesertaKuis |
|---|
| +KodeLembarSoal: Text +UserID: Numeric +setPeserta() +getPeserta() |

III.3.3.8 Kelas Cetak Lembar Soal dan Jawaban

-

| HasilKuis |
|---|
| +KodeLembarSoal: Text +NIM: Numeric +NomorSoal: Numeric +SoalID: Numeric +Pilihan1: Text +Pilihan2: Text +Pilihan3: Text +Pilihan4: Text +Jawaban: Char +submitKuis() +requestKuis() +randomizeKuis() +onCetak() +onCari() |

III.3.3.9 Kelas Cetak Rekap Nilai

-

| Nilai |
|--|
| +KodeLembarSoal: Text +NIM: Numeric +Nilai: Numeric +TotalJawabanBenar: Numeric |

III.3.3.10 Kelas Mengikuti Kuis

-

| HasilKuis |
|---|
| +KodeLembarSoal: Text +NIM: Numeric +NomorSoal: Numeric +SoalID: Numeric +Pilihan1: Text +Pilihan2: Text +Pilihan3: Text +Pilihan4: Text +Jawaban: Char +submitKuis() +requestKuis() +randomizeKuis() +onCetak() +onCari() |

III.3.3.11 Kelas Review Kuis

-

| HasilKuis |
|---|
| +KodeLembarSoal: Text +NIM: Numeric +NomorSoal: Numeric +SoalID: Numeric +Pilihan1: Text +Pilihan2: Text +Pilihan3: Text +Pilihan4: Text +Jawaban: Char +submitKuis() +requestKuis() +randomizeKuis() +onCetak() +onCari() |

III.3.3.12 Kelas Review Nilai

-

| Nilai |
|--|
| +KodeLembarSoal: Text +NIM: Numeric +Nilai: Numeric +TotalJawabanBenar: Numeric |

III.3.3.13 Kelas Koneksi Klien Android

-

| Klien Android |
|---|
| +login() +session() +logout() +tampilkan() +rubah() +hapus() +simpan() +set() +get() +cari() +konfirmasi() +AmbilSoal() +AmbilPeserta() +HapusPeserta() +PDF() +SimpanJawaban() +AmbilLembarSoal() +DetailLembarSoal() +ReviewNilai() |

III.3.3.14 Kelas Koneksi Klien PC

-

| Klien PC |
|--|
| +login() +session() +tampilkan() +rubah() +hapus() +simpan() +konfirmasi() +AmbilPeserta() +HapusPeserta() +run() +stop() +set() +PDF() +setHttpResult() +logout() |

III.3.3.15 Kelas Koneksi Server

-

| Server PC |
|--|
| +login() +logout() +RandomizeSoal() +set() +get() +tambah() +hapus() +rubah() +tampilkan() |

III.3.4 Perancangan Basis Data

Dalam tabel 8, di bawah ini adalah daftar tabel basis data yang digunakan dalam aplikasi Mobik.

Tabel 8 Daftar tabel dalam basis data yang digunakan aplikasi Mobik

| No. | Nama Tabel | Nama Kolom | Jenis | Panjang | Primary Key | Index |
|-----|---------------|------------|---------|---------|-------------|-------|
| 1 | LevelPengguna | LevelID | Numeric | 1 | Ya | |
| | | NamaLevel | Text | 15 | | |
| 2 | Prodi | ProdiID | Numeric | 2 | Ya | |
| | | NamaProdi | Text | 40 | | |

| No. | Nama Tabel | Nama Kolom | Jenis | Panjang | Primary Key | Index |
|-----|----------------|------------------|------------|---------|-------------|-------|
| 3 | Makul | MakulID | Text | 10 | Ya | |
| | | NamaMakul | Text | 80 | | |
| 4 | MakulMahasiswa | NIM | Numeric | 10 | Ya | |
| | | MakulID | Text | 10 | | |
| 5 | PenggunaMobik | UserID | Numeric | 10 | Ya | |
| | | NamaPanggilan | Text | 30 | | |
| | | KataKunci | Text | 16 | | |
| | | LevelID | Numeric | 1 | | |
| | | Aktif | Boolean | 1 | | |
| | | TerakhirMasuk | Time | | | |
| 6 | Dosen | NIK | Numeric | 10 | Ya | |
| | | Nama | Text | 30 | | |
| | | JenisKelamin | Char | 1 | | |
| | | ProdiID | Numeric | 2 | | |
| | | TglLahir | Date | | | |
| | | Alamat | Text | 80 | | |
| | | NoHP | Numeric | 15 | | |
| 7 | Mahasiswa | NIM | Numeric | 10 | Ya | |
| | | Nama | Text | 30 | | |
| | | JenisKelamin | Char | 1 | | |
| | | ProdiID | Numeric | 2 | | |
| | | TglLahir | Date | | | |
| | | Alamat | Text | 80 | | |
| | | NoHP | Numeric | 15 | | |
| 8 | BankSoal | SoalID | Numeric | 10 | Ya | |
| | | ProdiID | Numeric | 2 | | Ya |
| | | MakulID | Text | 10 | | Ya |
| | | Pertanyaan | Text | 255 | | |
| | | Pilihan1 | Text | 80 | | |
| | | Pilihan2 | Text | 80 | | |
| | | Pilihan3 | Text | 80 | | |
| | | Pilihan4 | Text | 80 | | |
| | | GambarPertanyaan | Text | 80 | | |
| | | GambarPilihan1 | Text | 80 | | |
| | | GambarPilihan2 | Text | 80 | | |
| | | GambarPilihan3 | Text | 80 | | |
| | | GambarPilihan4 | Text | 80 | | |
| | | Jawaban | Numeric | 1 | | |
| | | TingkatKesulitan | Numeric | 1 | | Ya |
| | | PembuatSoal | Numeric | 10 | | |
| | | WaktuPembuatan | Time Stamp | | | |
| | | PerubahSoal | Numeric | 10 | | |
| | | WaktuPerubahan | Time Stamp | | | |
| | | | | | | |
| | | | | | | |

| No. | Nama Tabel | Nama Kolom | Jenis | Panjang | Primary Key | Index |
|---------|-------------|-------------------|------------|---------|-------------|-------|
| 9 | LembarSoal | KodeLembarSoal | Text | 12 | Ya | |
| | | KataKunci | Text | 16 | | |
| | | ProdiID | Numeric | 2 | | Ya |
| | | MakulID | Text | 10 | | Ya |
| | | WaktuKuis | Numeric | 3 | | |
| | | JamKuis | Time | | | |
| | | TglKuis | Date | | | |
| | | JumlahSoal | Numeric | 3 | | |
| | | PembuatKuis | Numeric | 10 | | |
| | | WaktuPembuatan | Time Stamp | | | |
| | | PerubahKuis | Numeric | 10 | | |
| | | WaktuPerubahan | Time Stamp | | | |
| 10 | PesertaKuis | KodeLembarSoal | Text | 12 | Ya | |
| | | UserID | Numeric | 10 | Ya | |
| 11 | HasilKuis | KodeLembarSoal | Text | 12 | Ya | |
| | | UserID | Numeric | 10 | Ya | |
| | | NomorSoal | Numeric | 3 | | |
| | | SoalID | Numeric | 10 | | |
| | | Pilihan1 | Text | 80 | | |
| | | Pilihan2 | Text | 80 | | |
| | | Pilihan3 | Text | 80 | | |
| | | Pilihan4 | Text | 80 | | |
| Jawaban | Numeric | 1 | | | | |
| 12 | Nilai | KodeLembarSoal | Text | 12 | Ya | |
| | | UserID | Numeric | 10 | Ya | |
| | | TotalJawabanBenar | Numeric | 3 | | |
| | | Nilai | Numeric | 3 | | |

III.3.5 Perancangan Tampilan

III.3.5.1 Layar Login

III.3.5.1.1 Rancangan Tampilan Android



Gambar III-59 Tampilan Layar Login (Android)

Gambar III-59, pada layar *Mobile* Android, ini merupakan tampilan layar login untuk semua Pengguna. Pengguna memasukkan nama user dan kata kunci yang sesuai, jika berhasil maka pengguna akan dibawa ke menu utama sesuai dengan level pengguna. Jika nama pengguna atau kata kunci tidak sesuai, maka akan muncul pesan dialog “Nama pengguna atau kata kunci salah”.

III.3.5.1.2 Rancangan Tampilan Web

The image shows a web login interface. At the top left, there is a label 'Logo'. Below it is a rounded rectangular container for the login form. Inside this container, the word 'Login' is displayed at the top left. Below 'Login', there are two input fields. The first is labeled 'Kode Pengguna:' and the second is labeled 'Kata Sandi:'. Below the second input field, there is a checkbox with the label 'Ganti Kata Sandi'. At the bottom right of the container, there is a button labeled 'MASUK'.

Gambar III-60 Tampilan Layar Login (Web)

III.3.5.1.3 Algoritma/Query

Nama Kelas: Login.class

Nama operasi: Login

Algoritma:

// input nama pengguna dan kata kunci

// melakukan koneksi ke server

Koneksi("Login", newstring[] (nama_pengguna, kata_kunci))

// hasil pengembalian

Hasil=nama_user, kata_kunci, level_pengguna

Simpan(PenggunaMobik.aktif='Y')

// Jika tombol masuk di tekan dan level_pengguna=1 (admin)

Tampilkan Menu Admin

// Jika tombol masuk di tekan dan Jika level_pengguna=2 (dosen)

Tampilkan Menu Dosen

// Jika tombol masuk di tekan dan Jika level_pengguna=3 (mahasiswa)

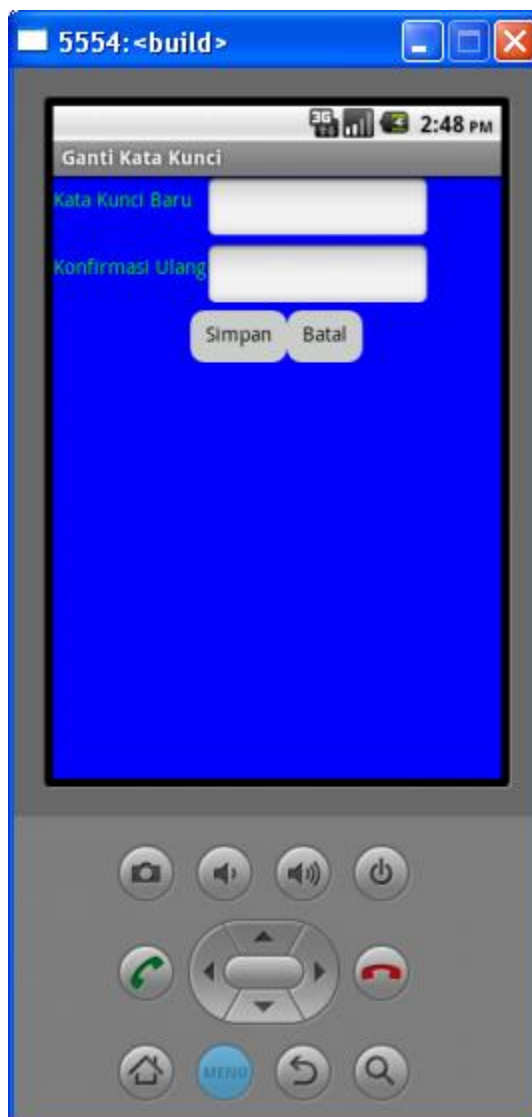
Tampilkan Menu Mahasiswa

// Jika tombol ganti kata kunci di tekan dan login berhasil

Tampilkan Menu Ganti Kata Kunci

III.3.5.2 Layar Ganti Kata Kunci

III.3.5.2.1 Rancangan Tampilan Android



Gambar III-61 Tampilan Layar Ganti Kata Kunci (Android)

Gambar III-61, pada layar *Mobile* Android, ini merupakan tampilan layar login untuk semua Pengguna. Pengguna memasukkan kata kunci dan konfirmasi ulang kata kunci yang sesuai, jika berhasil maka kata kunci untuk nama pengguna yang telah berhasil di bawa ke layar ganti kata kunci akan di ganti dengan kata kunci baru yang dimasukkan di sini. Jika kata kunci dan konfirmasi ulang kata kunci tidak sesuai, maka akan muncul pesan dialog “Kata kunci tidak cocok, silahkan ulangi lagi”.

III.3.5.2.2 Rancangan Tampilan Web

The image shows a web interface for changing a password. At the top left is the 'Logo' and at the top right is the user name 'Nama Pengguna' with a 'Keluar X' link. Below this is a navigation bar with 'HOME' selected. The main content area is titled 'Ganti Kata Sandi' and contains two input fields: 'Kata Sandi Baru:' and 'Ketik Ulang Kata Sandi:'. At the bottom of the form are two buttons: 'GANTI' and 'BATAL'.

Gambar III-62 Tampilan Layar Ganti Kata Kunci (Web)

III.3.5.2.3 Algoritma/Query

Nama Kelas: GantiKataKunci.class

Nama operasi: GantiKataKunci

Algoritma:

// input kata kunci dan konfirmasi ulang kata kunci

// dan tombol simpan di tekan

// lakukan pemeriksaan kata kunci yang baru dimasukkan

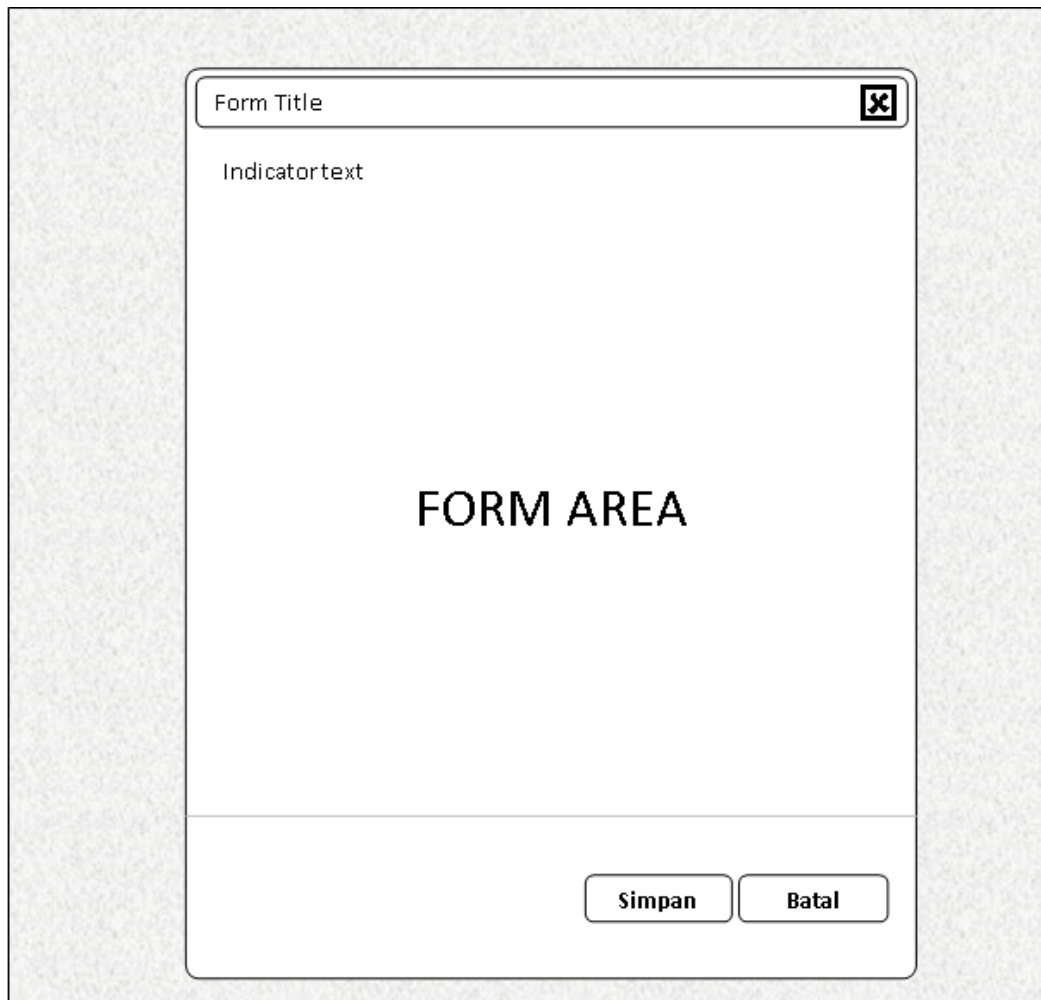
```
PeriksaKatakunci(kata_kunci, konfirmasi_kata_kunci): boolean
// hasil pengembalian
KatakunciBaruOK=Ya/Tidak
// Jika KataKunciBaruOK=Ya
Rubah(userID, konfirmasi_kata_kunci)
Tampilkan Menu Utama sesuai dengan level pengguna
// jika tombol Batal di tekan
Tampilkan layar login
```

III.3.5.3 Menu Admin

III.3.5.3.1 Rancangan Tampilan Web

The wireframe shows a web page layout for an admin menu. At the top left is the **Logo**, and at the top right is the user information **Nama Pengguna | Keluar X**. Below the logo is a **HOME** tab. The main content area is divided into a left sidebar and a right main panel. The sidebar contains two menu items: **SUB MENU 1 +** and **SUB MENU 2 +**. The main panel has a **MENU TITLE** section with two dropdown menus: **Prodi:** with the placeholder **--Pilih Prodi--** and **Makul:** with the placeholder **--Pilih Makul--**. Below these are two buttons: **CARI** and **TAMPIL SEMUA**. A **Tambah Baru** button is located below the search buttons. At the bottom of the main panel is a large rounded rectangle labeled **TABLE RESULT**.

Gambar III-63 Tampilan Layar Menu Admin (Web)



Gambar III-64 Tampilan Layar Sub Menu Admin (Web)

III.3.5.3.2 Algoritma/Query

Nama Kelas: MenuAdmin.class

Nama operasi: MenuAdmin

Algoritma:

```
// jika tombol [Pengguna Mobik] ditekan  
Tampilkan Layar Kelola Pengguna Mobik  
// jika tombol [Program Studi] ditekan  
Tampilkan Layar Kelola Program Studi  
// jika tombol [Mata Kuliah] ditekan  
Tampilkan Layar Kelola Mata Kuliah  
// jika tombol [Level Pengguna] ditekan
```

Tampilkan Layar Kelola Level Pengguna

// jika tombol [Keluar] ditekan

// melakukan koneksi ke server

Koneksi("Logout", newstring[] (nama_pengguna))

// hasil pengembalian

Simpan(PenggunaMobik.aktif='T')

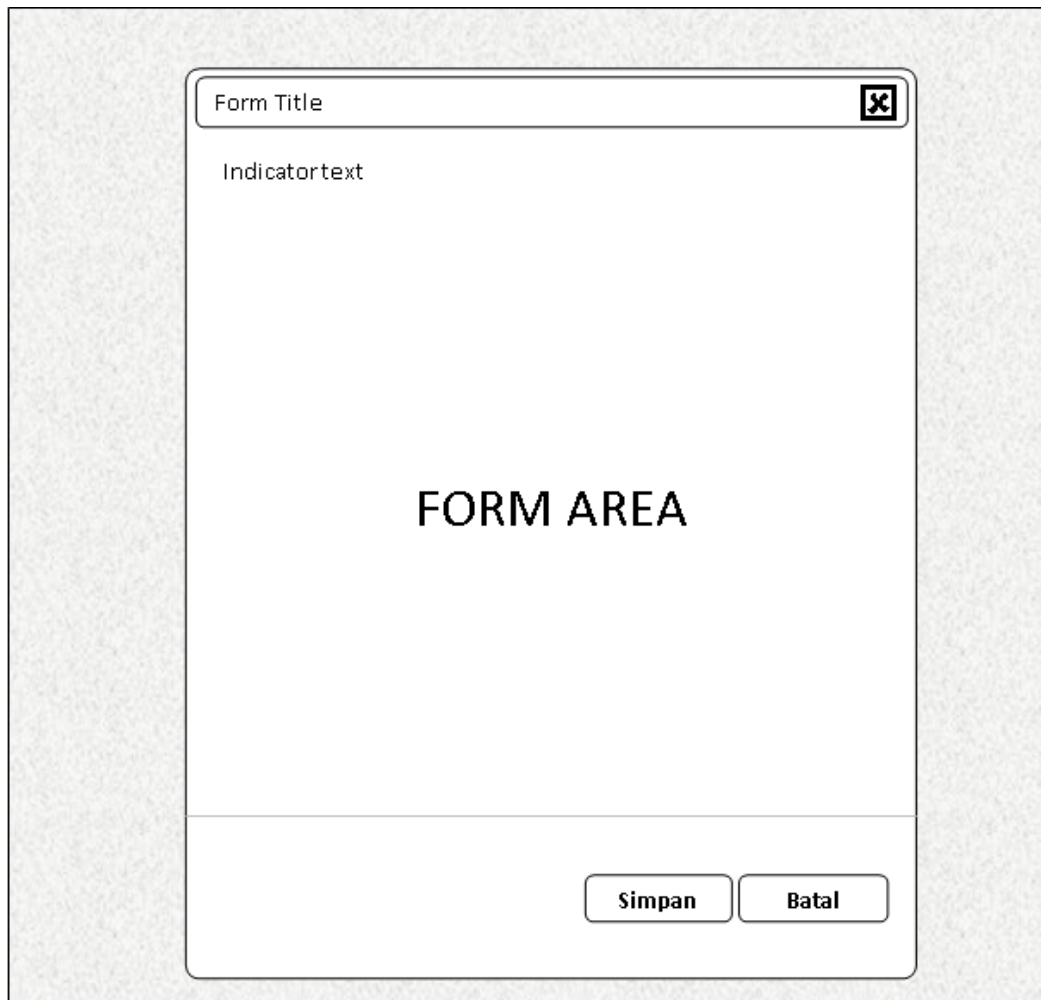
// Aplikasi Mobik di tutup

III.3.5.4 Menu Dosen

III.3.5.4.1 Rancangan Tampilan Web

The wireframe shows a web interface for a lecturer's menu. At the top left is the 'Logo' and at the top right is the user information 'Nama Pengguna | Keluar X'. Below this is a 'HOME' tab. The main content area is titled 'MENU TITLE' and contains two dropdown menus for 'Prodi' and 'Makul', followed by 'CARI' and 'TAMPIL SEMUA' buttons. A 'Tambah Baru' button is located below the search options. The bottom section is labeled 'TABLE RESULT'.

Gambar III-65 Tampilan Layar Menu Dosen (Web)



Gambar III-66 Tampilan Layar Sub Menu Dosen (Web)

III.3.5.4.2 Algoritma/Query

Nama Kelas: MenuDosen.class

Nama operasi: MenuDosen

Algoritma:

// jika tombol [Bank Soal] ditekan

Tampilkan Layar Akses Bank Soal

// jika tombol [Setup Kuis] ditekan

Tampilkan Layar Setup Kuis

// jika tombol [Daftar Peserta] ditekan

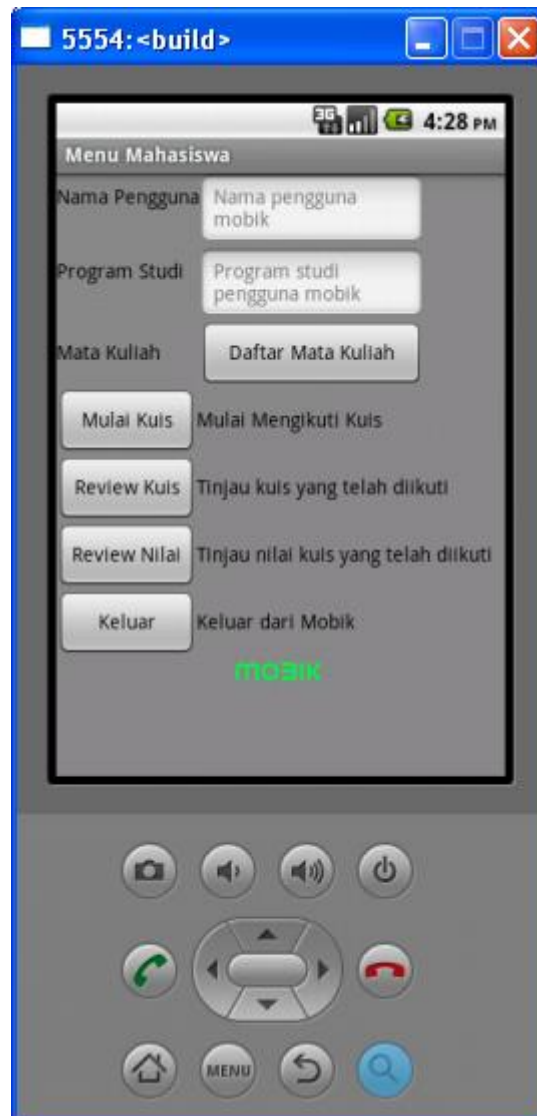
Tampilkan Layar Daftar Peserta

// jika tombol [Cetak Jawaban] ditekan

```
Tampilkan Layar Cetak Lembar Jawaban
// jika tombol [Cetak Nilai] ditekan
Tampilkan Layar Cetak Rekap Nilai
// jika tombol [Keluar] ditekan
// melakukan koneksi ke server
Koneksi("Logout", newstring[] (nama_pengguna))
// hasil pengembalian
Simpan(PenggunaMobik.aktif='T')
// Aplikasi Mobik di tutup
```

III.3.5.5 Menu Mahasiswa

III.3.5.5.1 Rancangan Tampilan Android



Gambar III-67 Tampilan Layar Menu Mahasiswa (Android)

Gambar III-67, pada layar *Mobile* Android, ini merupakan tampilan layar Menu Utama Mahasiswa untuk level pengguna = Mahasiswa.

III.3.5.5.2 Algoritma/Query

Nama Kelas: MenuMahasiswa.class

Nama operasi: MenuMahasiswa

Algoritma:

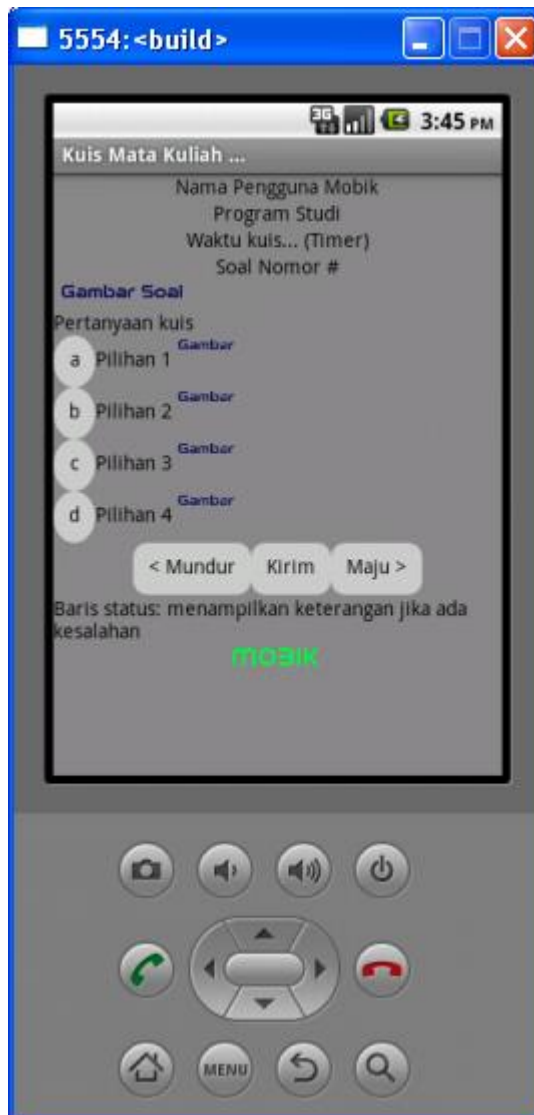
```
// jika tombol [Mulai Kuis] ditekan
Tampilkan Layar Pilih Kuis
// jika tombol [Review Kuis] ditekan
Tampilkan Layar Review Kuis
// jika tombol [Review Nilai] ditekan
Tampilkan Layar Review Nilai
// jika tombol [Keluar] ditekan
// melakukan koneksi ke server
Koneksi("Logout", newstring[] (nama_pengguna))
// hasil pengembalian
Simpan(PenggunaMobik.aktif='T')
// Aplikasi Mobik di tutup
```

III.3.5.6 Mengikuti Kuis

III.3.5.6.1 Rancangan Tampilan Android



Gambar III-68 Tampilan Layar Mengikuti Kuis (Android)



Gambar III-69 Tampilan Layar Kuis (Android)

Gambar III-68 dan Gambar III-69, pada layar *Mobile* Android, ini merupakan tampilan layar Mengikuti Kuis untuk level pengguna = Mahasiswa.

III.3.5.6.2 Algoritma/Query

Nama Kelas: MobikKuis.class

Nama operasi: MobikKuis

Algoritma:

// input kode lembar soal & kata kunci, jika kode lembar soal dan kata kunci benar

Cek(KodeLembarSoal, KataKunciLembarSoal)

Tampilkan Layar Kuis

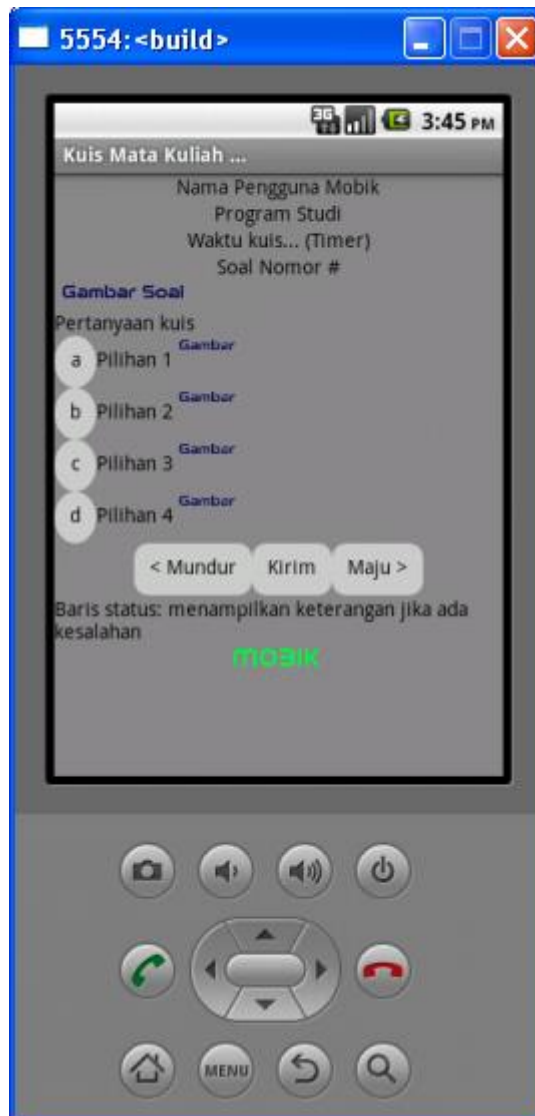
```

// jika kode lembar soal dan kata kunci salah,
PesanKesalahan("Kode Lembar soal atau kata kunci salah")
// jika tombol [Ambil Soal] di tekan maka sistem akan menampilkan soal untuk
pengguna Mahasiswa.
// jika tombol [Batal Ambil Soal] di tekan maka sistem akan kembali ke layar
menu Mahasiswa.
// melakukan koneksi ke server dan mengambil soal
AmbilSoal(KodeSoal, userID, makulID, prodiID)
Tampilkan Layar Kuis()
// jika tombol pilihan A ditekan
simpanJawaban (KodeSoal, UserID, SoalNo, Jawaban=1)
// jika tombol pilihan B ditekan
simpanJawaban (KodeSoal, UserID, SoalNo, Jawaban=2)
// jika tombol pilihan C ditekan
simpanJawaban (KodeSoal, UserID, SoalNo, Jawaban=3)
// jika tombol pilihan D ditekan
simpanJawaban(KodeSoal, UserID, SoalNo, Jawaban=4)
// jika tombol [Maju] ditekan
SoalNo+=1
Tampilkan(KodeSoal, UserID, SoalNo)
// jika tombol [Mundur] ditekan
SoalNo-=1
Tampilkan(KodeSoal, UserID, SoalNo)
// jika tombol [Kirim] ditekan
SubmitJawaban(KodeSoal, UserID)
// Kembali ke Menu
Tampilkan MenuMahasiswa()

```

III.3.5.7 Review Kuis

III.3.5.7.1 Rancangan Tampilan Android



Gambar III-70 Tampilan Layar Review Kuis (Android)

Gambar III-70, pada layar Mobile Android, ini merupakan tampilan layar Review Kuis untuk level pengguna = Mahasiswa.

III.3.5.7.2 Algoritma/Query

Nama Kelas: ReviewKuis.class

Nama operasi: ReviewKuis

Algoritma:

// tampilkan kuis yang dipilih pengguna

III.3.5.8 Review Nilai

III.3.5.8.1 Rancangan Tampilan Android



Gambar III-71 Tampilan Layar Review Nilai (Android)

Gambar III-71, pada layar *Mobile* Android, ini merupakan tampilan layar Review Nilai untuk level pengguna = Mahasiswa.

III.3.5.8.2 Algoritma/Query

Nama Kelas: ReviewNilai.class

Nama operasi: ReviewNilai

Algoritma:

// tampilkan daftar nilai pengguna

Bab IV Implementasi dan Pengujian

Setelah sistem dianalisis dan didesain secara rinci, maka akan menuju tahap implementasi. Implementasi merupakan tahap meletakkan sistem sehingga siap untuk dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan, sehingga pengguna dapat memberikan masukan kepada pembangun sistem. Untuk mengetahui apakah aplikasi tersebut dapat melakukan fungsi sesuai dengan perencanaan, maka diperlukan pengujian-pengujian.

IV.1 Implementasi

IV.1.1 Fungsi

Fungsi-fungsi yang telah dirancang, kemudian diimplementasikan ke bentuk file-file PHP. Berikut fungsi-fungsi dan file hasil implementasinya.

Tabel 9 Lokasi fungsi dalam file

| Fungsi ID | Keterangan | File |
|-----------|--|---|
| FMOBIK-01 | Pembuatan <i>Login</i> untuk otentifikasi akses ke dalam aplikasi MOBIK. | login.php, login-check.php |
| FMOBIK-02 | Menejemen data <i>user</i> pemakai aplikasi MOBIK (mahasiswa/dosen/administrator), meliputi <i>Password</i> dan level pemakai aplikasi MOBIK. Pada Menejemen data pemakai ditampilkan nama-nama pemakai yang sudah tersimpan. | user-mg.php, user-mg-tb.php, user-page.php |
| FMOBIK-03 | Menejemen program studi yang tersedia. | prodi-mg.php, prodi-mg-tb.php, prodi-page.php |
| FMOBIK-04 | Menejemen mata kuliah yang tersedia. | makul-mg.php, makul-mg-tb.php, makul-page.php |
| FMOBIK-05 | Menejemen <i>level user</i> / tingkat pengguna yang tersedia. | user-mg-tb.php |
| FMOBIK-06 | <i>User</i> dapat mengubah <i>Password</i> nya masing-masing | gantipass.php, Webdb.php |
| FMOBIK-07 | Menejemen data bank soal, dosen dapat melakukan penambahan, perubahan, penghapusan, dan pencarian soal untuk mata kuliah dosen terkait. Pada Menejemen bank soal ditampilkan juga soal-soal yang sudah ada. | bsoal-mg.php, bsoal-mg-tb.php, bsoal-page.php |
| FMOBIK-08 | Dosen membuat setup kuis: Menejemen kuis, Daftar mahasiswa peserta kuis (mahasiswa). Waktu mulai dan lama kuis. Jenis soal kuis (soal bebas/pilihan). Kode lembar soal. Katakunci untuk mengikuti kuis. Mengubah lama waktu kuis. Setelah setup kuis dikonfirmasi, maka aplikasi akan membuat dan mengacak soal dan | kuis-mg.php, kuis-mg-tb.php, kuis-page.php |

| Fungsi ID | Keterangan | File |
|------------------|--|---|
| | mengacak pilihan jawaban untuk tiap peserta kuis ke dalam database. Sistem akan mencatat soal-soal yang akan diberikan kepada <i>client</i> dan merekap semua susunan soal dan susunan pilihan yang telah diacak oleh sistem untuk digunakan sebagai bahan pemeriksaan untuk proses penilaian. | |
| FMOBIK-09 | <i>User</i> (mahasiswa) meminta soal dari server dan server mengambil soal yang telah dipilih/ditandai berdasarkan nama pengguna peserta kuis. | Webdb.php |
| FMOBIK-10 | <i>User</i> (mahasiswa) dapat melihat 1 soal dalam 1 layar, dapat mengerjakan soal secara acak, dan dapat memajukan/memundurkan 1 soal di layar <i>client</i> . | Webdb.php |
| FMOBIK-11 | <i>User</i> (mahasiswa) mengirim jawaban secara manual dengan menekan tombol kirim/submit di akhir soal yang diberikan. | Webdb.php |
| FMOBIK-12 | <i>Client</i> (mahasiswa) akan mengirim jawaban secara otomatis jika waktu yang telah diberikan telah habis. | Webdb.php |
| FMOBIK-13 | <i>Server</i> akan memeriksa jawaban ketika status jawaban telah berubah menjadi “terkirim/untuk di nilai” dan memberikan nilai untuk setiap soal dan memberikan nilai secara keseluruhan kepada peserta (mahasiswa) kuis. | Mysql <i>view query</i> |
| FMOBIK-14 | <i>User</i> (dosen) dapat mencetak soal/jawaban. (untuk rekap/bukti tertulis jika diperlukan) | lembarjawaban-mg.php, lembarjawaban-mg-tb.php, lembarjawaban-page.php |
| FMOBIK-15 | <i>User</i> (dosen) dapat melihat/mencetak rekap nilai mahasiswa. (untuk rekap/bukti tertulis jika diperlukan) | Lembarjawaban-report.php |
| FMOBIK-16 | <i>User</i> (mahasiswa) dapat melihat nilai dan peringkat. | reviewjawaban-mg.php, reviewjawaban-mg-tb.php, reviewjawaban-page.php, Webdb.php |
| FMOBIK-17 | Semua <i>user</i> dapat melakukan LogOut | function.php, Webdb.php |

IV.1.2 Database

Database MySQL yang telah diimplementasikan untuk sistem adalah sebagai berikut:

Tabel 10 Struktur tabel BankSoal

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|-------------------|--------------------|-------------|----------------|
| <i>SoalID</i> | <i>varchar(10)</i> | <i>No</i> | 0 |
| <i>ProdiID</i> | <i>char(2)</i> | <i>No</i> | |
| <i>MakulID</i> | <i>int(4)</i> | <i>No</i> | |
| <i>Pertanyaan</i> | <i>longtext</i> | <i>No</i> | |

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|------------------|--------------------|-------------|---------------------|
| Pilihan1 | <i>varchar(80)</i> | <i>No</i> | |
| Pilihan2 | <i>varchar(80)</i> | <i>No</i> | |
| Pilihan3 | <i>varchar(80)</i> | <i>No</i> | |
| Pilihan4 | <i>varchar(80)</i> | <i>No</i> | |
| Jawaban | <i>int(1)</i> | <i>No</i> | |
| TingkatKesulitan | <i>int(1)</i> | <i>No</i> | |
| PembuatSoal | <i>varchar(10)</i> | <i>No</i> | |
| PerubahSoal | <i>varchar(10)</i> | <i>No</i> | |
| WaktuPerubahan | <i>timestamp</i> | <i>No</i> | CURRENT_TIMESTAMP |
| WaktuPembuatan | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |

Tabel 11 Struktur tabel HasilKuis

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|----------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| NomorSoal | <i>int(3)</i> | <i>No</i> | |
| SoalID | <i>varchar(10)</i> | <i>No</i> | |
| Pilihan1 | <i>int(1)</i> | <i>No</i> | |
| Pilihan2 | <i>int(1)</i> | <i>No</i> | |
| Pilihan3 | <i>int(1)</i> | <i>No</i> | |
| Pilihan4 | <i>int(1)</i> | <i>No</i> | |
| Jawaban | <i>int(1)</i> | <i>No</i> | |

Tabel 12 Struktur tabel JenisMK

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|---------------|--------------------|-------------|----------------|
| <i>Jenis</i> | <i>varchar(4)</i> | <i>No</i> | |
| Keterangan | <i>varchar(30)</i> | <i>Yes</i> | NULL |

Tabel 13 Struktur tabel LembarSoal

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|------------------|------------------|-------------|---------------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| KataKunci | <i>text</i> | <i>No</i> | |
| ProdiID | <i>char(2)</i> | <i>No</i> | |
| MakulID | <i>int(4)</i> | <i>No</i> | |
| WaktuKuis | <i>int(3)</i> | <i>No</i> | |
| TanggalMulaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| TanggalAkhirKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |

| | | | |
|----------------|--------------------|-----------|---------------------|
| JumlahSoal | <i>int(3)</i> | <i>No</i> | |
| PembuatKuis | <i>varchar(10)</i> | <i>No</i> | |
| WaktuPembuatan | <i>timestamp</i> | <i>No</i> | CURRENT_TIMESTAMP |
| PerubahKuis | <i>varchar(10)</i> | <i>No</i> | |
| WaktuPerubahan | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |

Tabel 14 Struktur tabel LevelPengguna

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|---------------|-------------|----------------|
| LevelID | <i>int(1)</i> | <i>No</i> | |
| NamaLevel | <i>text</i> | <i>No</i> | |

Tabel 15 Struktur tabel Makul

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|-------------------|-------------|----------------|
| MakulID | <i>int(4)</i> | <i>No</i> | |
| NamaMakul | <i>text</i> | <i>No</i> | |
| ProdiID | <i>char(2)</i> | <i>No</i> | |
| Jenis | <i>varchar(4)</i> | <i>No</i> | |
| SKS | <i>tinyint(1)</i> | <i>No</i> | |
| Semester | <i>tinyint(1)</i> | <i>No</i> | |

Tabel 16 Struktur tabel PenggunaMakul

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|---------------|--------------------|-------------|----------------|
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| MakulID | <i>int(4)</i> | <i>No</i> | |

Tabel 17 Struktur tabel PenggunaMobik

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|-------------------|
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| Nama | <i>text</i> | <i>No</i> | |
| KataKunci | <i>text</i> | <i>No</i> | |
| LevelID | <i>int(1)</i> | <i>No</i> | |
| Aktif | <i>tinyint(1)</i> | <i>No</i> | |
| OnLine | <i>tinyint(1)</i> | <i>No</i> | |
| TerakhirMasuk | <i>datetime</i> | <i>No</i> | |
| TerakhirKeluar | <i>datetime</i> | <i>No</i> | |
| AlamatIP | <i>varchar(15)</i> | <i>No</i> | |
| TerakhirUpdate | <i>timestamp</i> | <i>No</i> | CURRENT_TIMESTAMP |

Tabel 18 Struktur tabel PesertaKuis

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|---------------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| JamMulaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| JamSelesaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| status | <i>tinyint(1)</i> | <i>No</i> | |
| PerubahanData | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| DiubahOleh | <i>varchar(10)</i> | <i>No</i> | |

Tabel 19 Struktur tabel Prodi

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|----------------|-------------|----------------|
| ProdiID | <i>char(4)</i> | <i>No</i> | |
| NamaProdi | <i>text</i> | <i>No</i> | |

Tabel 20 Struktur tabel ProfilPengguna

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|---------------------|
| NI | <i>varchar(10)</i> | <i>No</i> | |
| Nama | <i>text</i> | <i>No</i> | |
| JenisKelamin | <i>text</i> | <i>No</i> | |
| ProdiID | <i>char(2)</i> | <i>No</i> | |
| TempatLahir | <i>varchar(20)</i> | <i>No</i> | |
| TglLahir | <i>date</i> | <i>No</i> | |
| Alamat | <i>longtext</i> | <i>No</i> | |
| NoHP | <i>text</i> | <i>No</i> | |
| TahunMasuk | <i>varchar(4)</i> | <i>No</i> | |
| Kategori | <i>text</i> | <i>No</i> | |
| Kelas | <i>text</i> | <i>No</i> | |
| TerakhirUpdate | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |

Tabel 21 Struktur tabel vCekHasilJawaban

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|----------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| NomorSoal | <i>int(3)</i> | <i>No</i> | |
| SoalID | <i>varchar(10)</i> | <i>No</i> | |
| JawabanPeserta | <i>int(1)</i> | <i>No</i> | |
| KunciJawaban | <i>int(1)</i> | <i>No</i> | |
| StatusJawaban | <i>varchar(5)</i> | <i>Yes</i> | NULL |

Tabel 22 Struktur tabel vNilai

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|------------------------|----------------------|-------------|----------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| JumlahSoal | <i>bigint(21)</i> | <i>No</i> | 0 |
| SoalTerjawab | <i>bigint(21)</i> | <i>No</i> | 0 |
| JawabanBenar | <i>decimal(23,0)</i> | <i>Yes</i> | NULL |
| TotalBobotNilaiSoal | <i>decimal(32,0)</i> | <i>Yes</i> | NULL |
| TotalBobotNilaiPeserta | <i>decimal(33,0)</i> | <i>Yes</i> | NULL |
| Nilai | <i>decimal(40,4)</i> | <i>Yes</i> | NULL |

Tabel 23 Struktur tabel vpm

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|---------------|--------------------|-------------|----------------|
| UserID | <i>varchar(10)</i> | <i>Yes</i> | NULL |
| MakulID | <i>int(4)</i> | <i>Yes</i> | NULL |
| NamaMakul | <i>text</i> | <i>No</i> | |
| ProdiID | <i>char(2)</i> | <i>No</i> | |

Tabel 24 Struktur tabel vStatusJawaban

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|---------------------|--------------------|-------------|---------------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| JamMulaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| JamSelesaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| WaktuKuis | <i>int(3)</i> | <i>No</i> | |
| WaktuKuisBerakhir | <i>datetime</i> | <i>Yes</i> | NULL |
| WaktuServerSekarang | <i>datetime</i> | <i>No</i> | 0000-00-00 00:00:00 |
| JumlahSoal | <i>bigint(21)</i> | <i>No</i> | 0 |
| SoalTerjawab | <i>bigint(21)</i> | <i>No</i> | 0 |
| status | <i>tinyint(1)</i> | <i>No</i> | |

Tabel 25 Struktur tabel vStatusPeserta

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|---------------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| JamMulaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| JamSelesaiKuis | <i>timestamp</i> | <i>No</i> | 0000-00-00 00:00:00 |
| WaktuKuis | <i>int(3)</i> | <i>No</i> | |

| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|---------------------|-------------------|-------------|---------------------|
| WaktuKuisBerakhir | <i>datetime</i> | <i>Yes</i> | NULL |
| WaktuServerSekarang | <i>datetime</i> | <i>No</i> | 0000-00-00 00:00:00 |
| status | <i>tinyint(1)</i> | <i>No</i> | |

Tabel 26 Struktur tabel vCekHasilJawaban

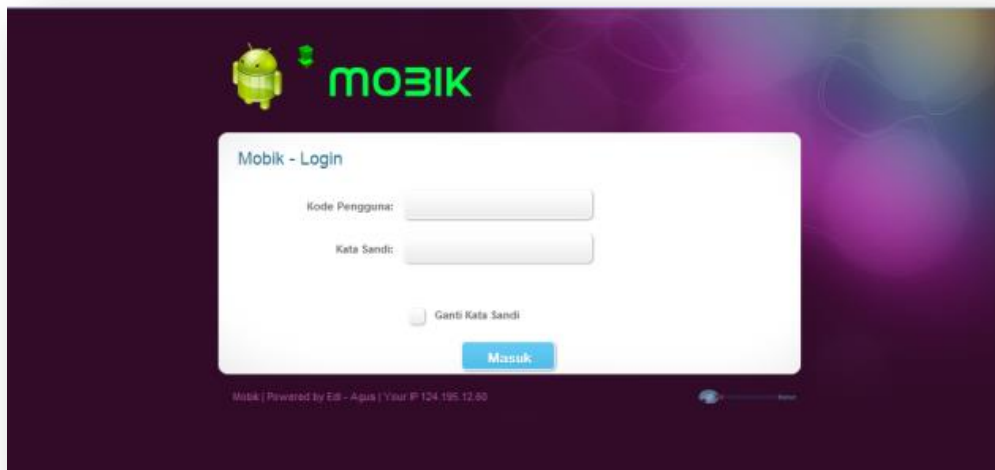
| <i>Column</i> | <i>Type</i> | <i>Null</i> | <i>Default</i> |
|----------------|--------------------|-------------|----------------|
| KodeLembarSoal | <i>text</i> | <i>No</i> | |
| UserID | <i>varchar(10)</i> | <i>No</i> | |
| NomorSoal | <i>int(3)</i> | <i>No</i> | |
| SoalID | <i>varchar(10)</i> | <i>No</i> | |
| JawabanPeserta | <i>int(1)</i> | <i>No</i> | |
| KunciJawaban | <i>int(1)</i> | <i>No</i> | |
| StatusJawaban | <i>varchar(5)</i> | <i>Yes</i> | NULL |

IV.1.3 Implementasi Antarmuka

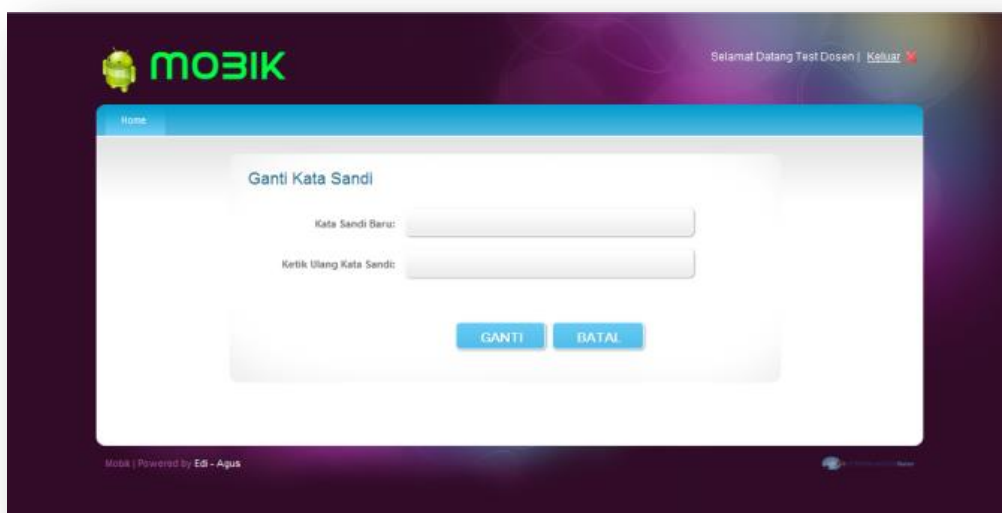
Implementasi antarmuka dilakukan dengan setiap halaman aplikasi yang dibuat dan pengkodeannya dalam bentuk file program. Berikut ini adalah implementasi antarmuka yang telah dibuat dalam dua format:

IV.1.3.1 Antarmuka pada *web page*

Antarmuka ini digunakan untuk *mensetup* kuis atau pra-kuis dan juga dilengkapi untuk pengecekan nilai hasil kuis. Untuk membuka halaman web ini ketik pada *address bar* `http://alamatweb/Mobik/` (alamatweb tergantung lokasi server). Berikut beberapa antarmuka tersebut:



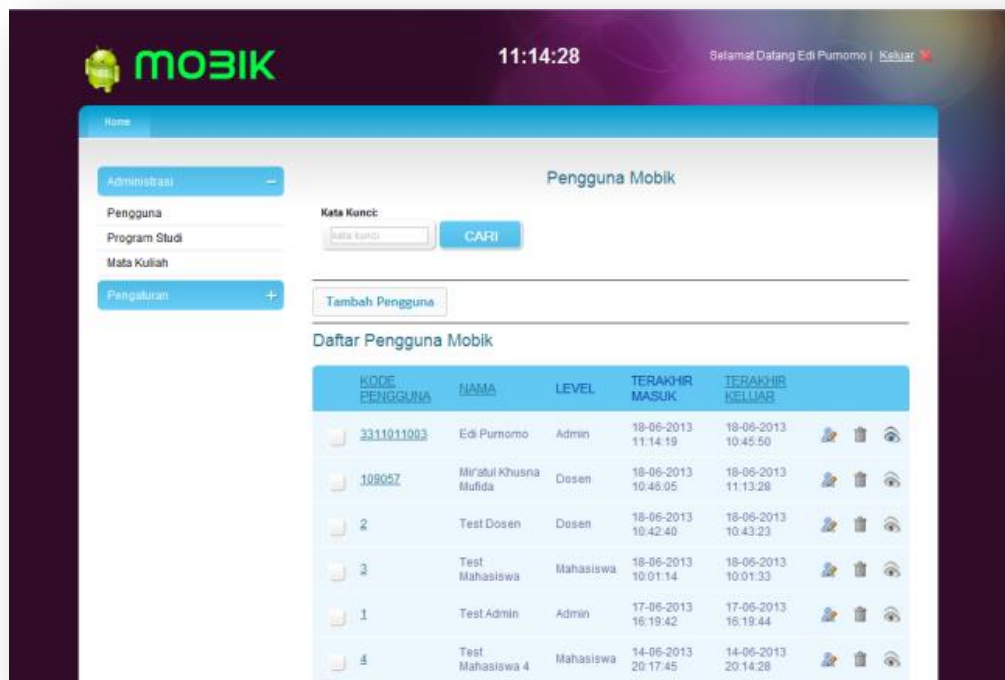
Gambar IV-1 Antarmuka Login (login.php)



Gambar IV-2 Antarmuka Ganti Kata Kunci (gantipass.php)



Gambar IV-3 Antarmuka Halaman Utama (index.php)



Gambar IV-4 Antarmuka Halaman Administrasi - Utama

Perubahan Pengguna Baru

Semua form isian ini dibutuhkan.

Pilih Prodi: --Pilih Prodi--

Kode Pengguna:

Nama:

Tanggal Lahir:

Jenis Kelamin:

Level:

Angkatan: 2009

Kategori:

Kelas:

Gambar IV-5 Antarmuka Halaman Administrasi – Form

MOBIK 10:46:15 Selamat Datang Miratul Khusna Mukida | Kobar

Home

Dosen

Bank Soal

Kuis

Lembar Jawaban

Pengaturan

Pilih Prodi: --Pilih Prodi--

Pilih Matakuliah: --Pilih Matakuliah--

Daftar Bank Soal

| Kode Soal | Soal | | | | Babot |
|------------|--|------------|------------|-------------------|-------|
| | Pilihan 1 | Pilihan 2 | Pilihan 3 | Pilihan 4 | |
| 1001000001 | $((p \Rightarrow q) \wedge p) \Rightarrow q$ benar jika, (1) p benar dan q benar, (2) p benar dan q salah, (3) p salah dan q benar, (4) p salah dan q salah. Pernyataan yang benar adalah: | | | | 3 |
| | a. 1, 2, dan 3 | b. 1 dan 3 | c. 2 dan 4 | d. 1, 2, 3, dan 4 | |
| 1001000002 | Nilai x yang menyebabkan pernyataan "Jika $x^2 + x = 6$ maka $x^2 + 3x < 9$ " bernilai salah adalah... | | | | 3 |
| | a. -3 | b. -2 | c. 1 | d. 2 | |

Gambar IV-6 Antarmuka Halaman Dosen

Bank Soal

Penambahan Bank Soal ✕

Semua form isian ini dibutuhkan.

Kode Prodi: IF
 Kode Makul: 1001
 Nama Makul: MATEMATIKA
 Kode Soal: 100100033

Text Soal:

Pilihan:

a.

b.

c.

d.

Bobot: Mudah Sedang Sulit

Gambar IV-7 Antarmuka Halaman Dosen - Form

11:02:32 Selamat Datang M.Faisal Khairina Mulda

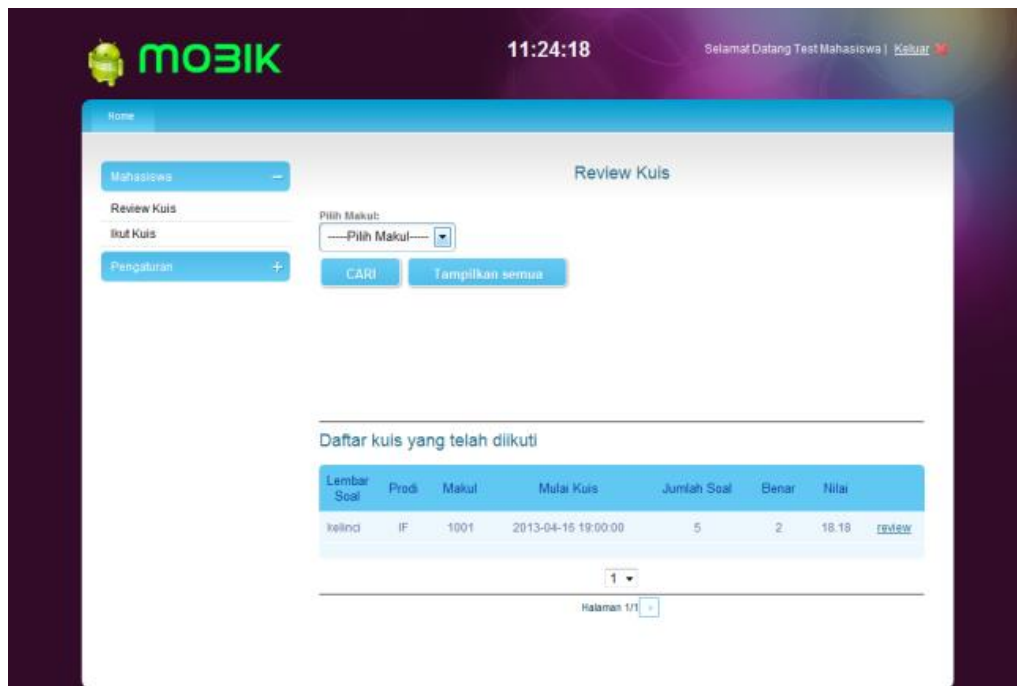
Peserta Kuis KELINCI ✕

show 10 entries Search:

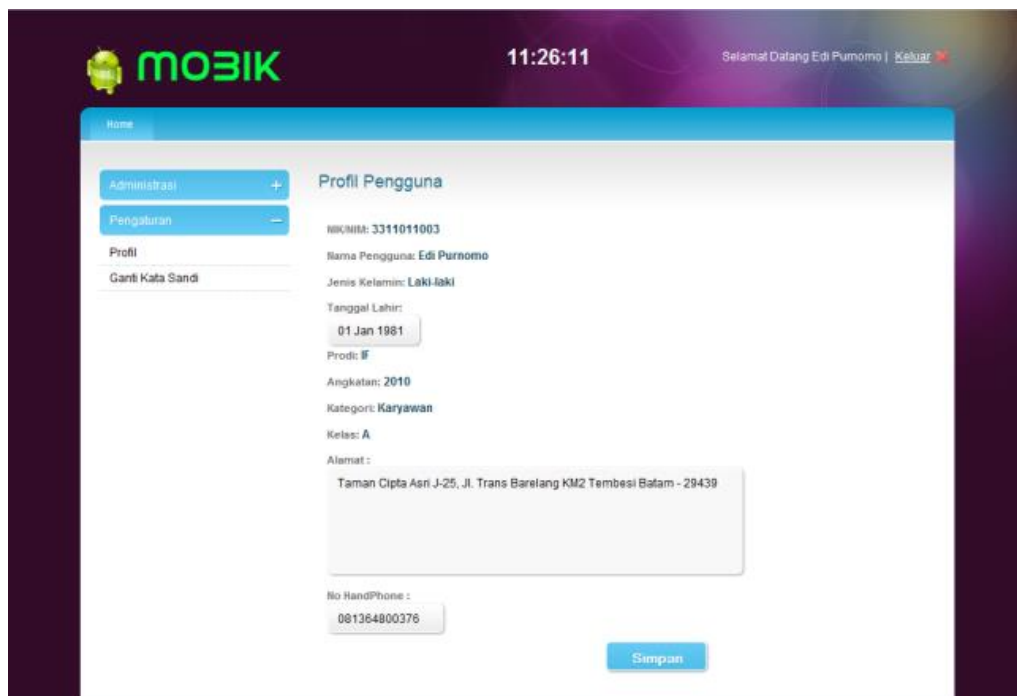
| | Kode Peserta | Nama | Kategori | Kelas | Nilai | Status |
|-----------------------|--------------|------------------|----------|-------|-------|-------------|
| cetak | 3 | Test Mahasiswa | KARYAWAN | B | 18.18 | selesai |
| cetak | 4 | Test Mahasiswa 4 | KARYAWAN | B | 0.00 | selesai |
| cetak | 123456 | ROBI | REGULER | A | - | belum mulai |

Showing 1 to 3 of 3 entries 🔍

Gambar IV-8 Antarmuka Halaman Dosen - Nilai



Gambar IV-9 Antarmuka Halaman Mahasiswa



Gambar IV-10 Antarmuka Halaman Profil

IV.1.3.2 Antarmuka pada Android

Antarmuka ini digunakan untuk mahasiswa untuk melakukan/mengerjakan kuis secara *mobile*. Mahasiswa harus melakukan *download* aplikasi Mobik.apk yang berada pada halaman login versi *web page* yang dibuka melalui perangkat Android, kemudian klik gambar panah hijau (lihat: Gambar IV-1). Setelah *download* komplit, *install* aplikasi Mobik pada perangkat Android tersebut. Untuk menjalankan Mobik klik pada *icon* Mobik (lihat: Gambar IV-11). Berikut beberapa antarmuka tersebut:



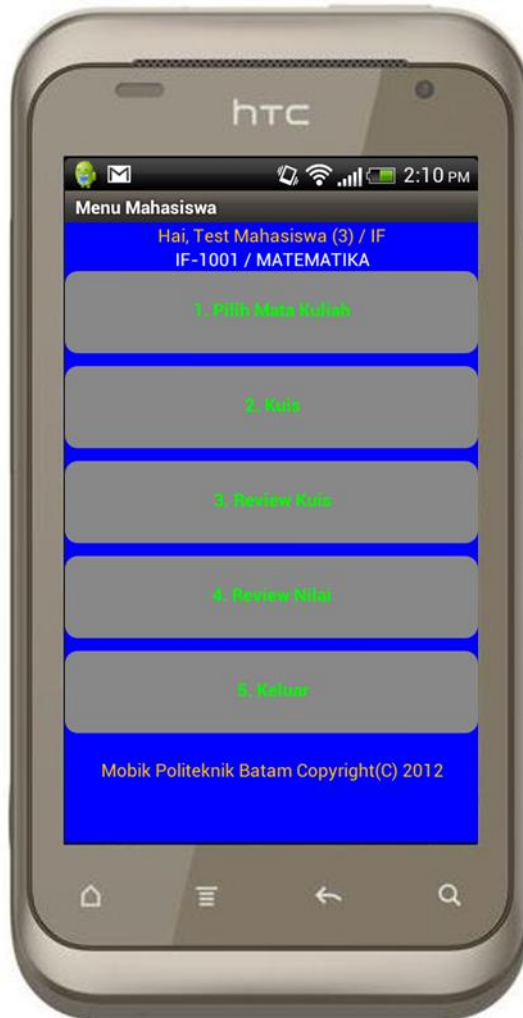
Gambar IV-11 Mobik icon pada Android



Gambar IV-12 Antarmuka Halaman Login – Android



Gambar IV-13 Antarmuka Halaman Ganti Kata Kunci – Android



Gambar IV-14 Antarmuka Halaman Utama – Android



Gambar IV-15 Antarmuka Halaman Ikut Kuis – Android



Gambar IV-16 Antarmuka Halaman Kuis – Android



Gambar IV-17 Antarmuka Halaman Kirim Jawaban – Android



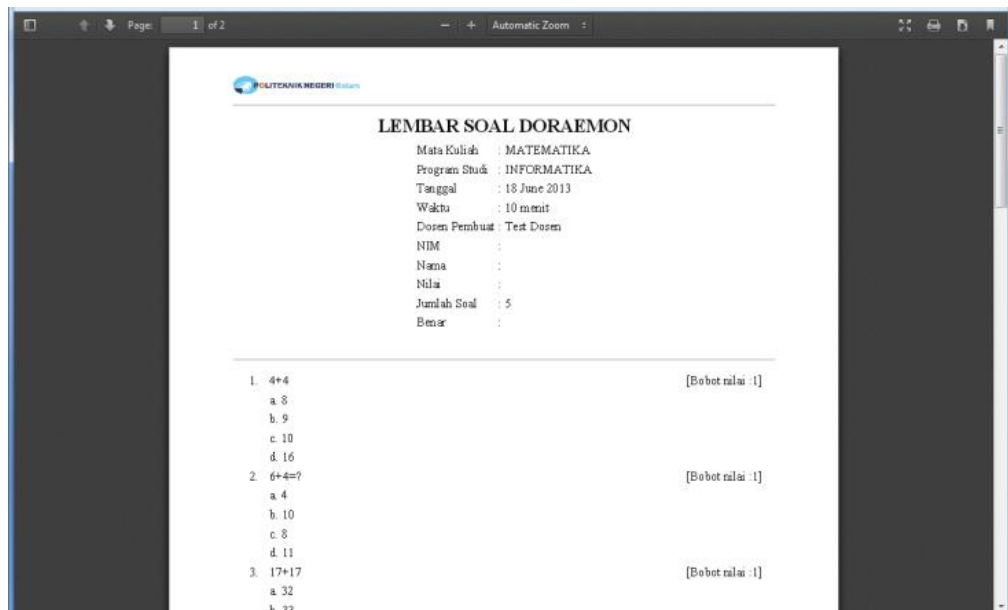
Gambar IV-18 Antarmuka Halaman *Review* Nilai – Android



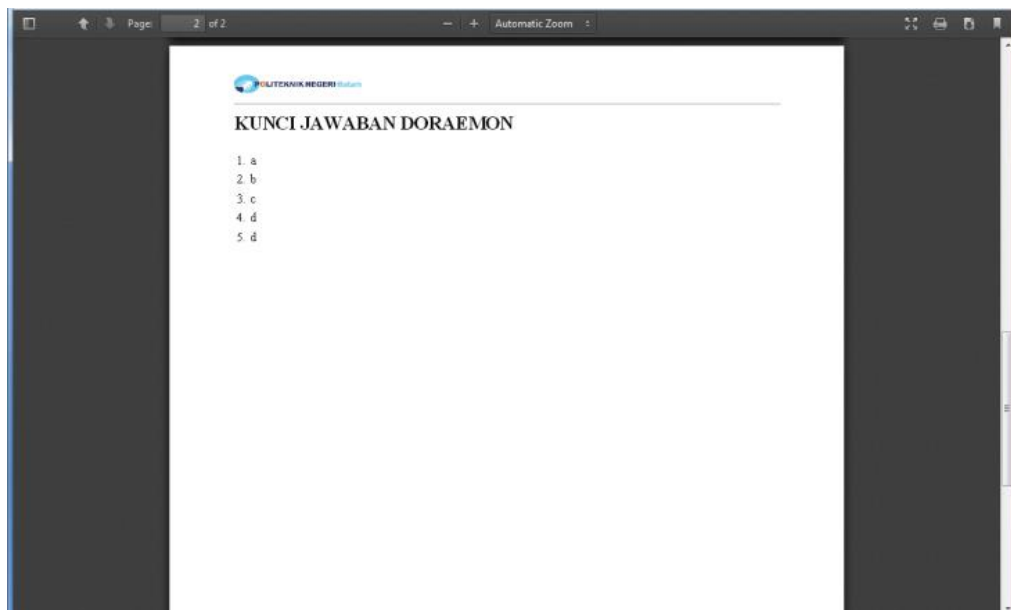
Gambar IV-19 Antarmuka Halaman *Review Kuis* – Android

IV.1.4 Implementasi Laporan/Report

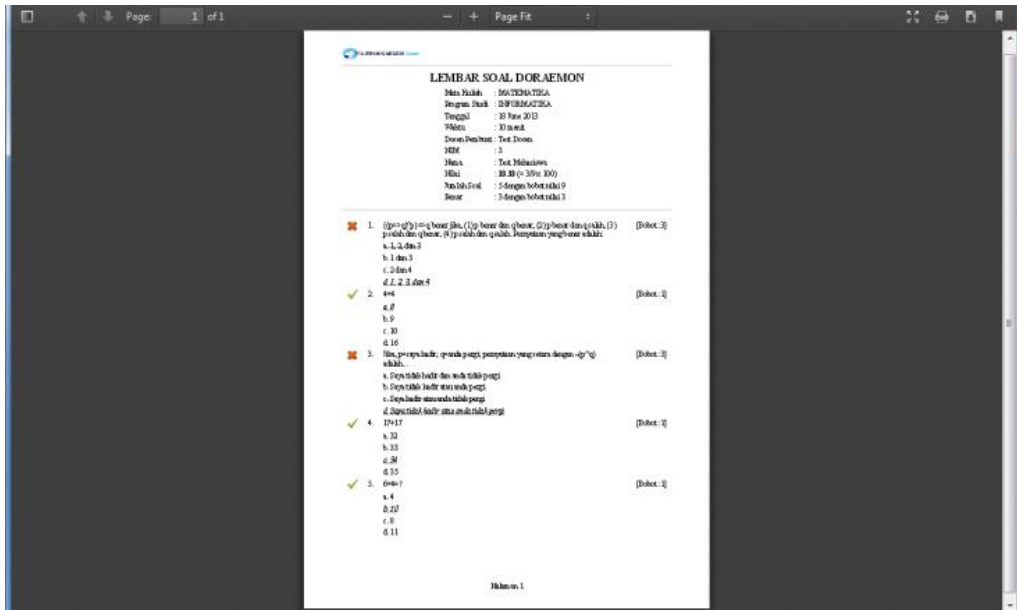
Laporan dibuat dalam dua format, yaitu format PDF dan format Excel (khusus untuk nilai). Laporan melingkupi: lembar soal, lembar jawaban (*review kuis*) dan nilai. Untuk membuat laporan harus melalui *desktop PC*. Berikut bentuk laporan yang telah dibuat:



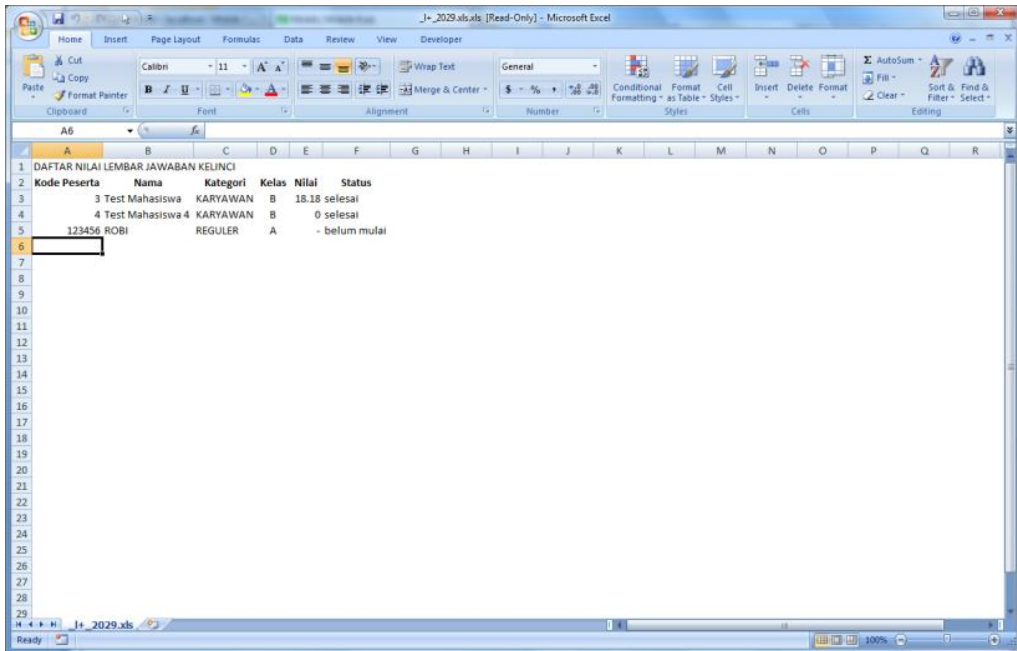
Gambar IV-20 Laporan Lembar Soal - Soal



Gambar IV-21 Laporan PDF Lembar Soal – Kunci Jawaban



Gambar IV-22 Laporan PDF Lembar Jawaban/Review Kuis



Gambar IV-23 Laporan Excel Nilai

IV.2 Pengujian/Testing

Strategi *testing* yang kami lakukan menggunakan strategi *Black Box*. *Black Box testing* merupakan strategi *testing* di mana hanya memperhatikan atau memfokuskan kepada faktor fungsionalitas dan spesifikasi perangkat lunak.

Berbeda dengan *white box*, *black box testing* tidak membutuhkan pengetahuan mengenai alur internal (*internal path*), struktur atau implementasi dari *software under test* (SUT).

Karena itu uji coba black box memungkinkan untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional program.

Pengujian-pengujian pada bagian ini telah di uji oleh Dosen pembimbing kami, yaitu Ibu Mir'atul Khusna Mufida, S.ST.

IV.2.1.1 Black Box Testing pada Web Page

Berikut ini strategi-strategi *black box testing* pada *web page*:

IV.2.1.1.1 FMOBIK-01 Login

Pengujian login berlaku untuk semua pengguna.

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengosongkan semua isian data login, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: - Kata sandi: - | Sistem akan menolak akses login dan menampilkan pesan "Masukkan kode pengguna Anda." | ✓ | |
| 2. | Hanya menginputkan data Kode Pengguna 1 dan mengosongkan data kata sandi, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: 1 Kata sandi: - | Sistem akan menolak akses login dan menampilkan pesan "Kode pengguna/kata sandi salah." | ✓ | |
| 3. | Hanya menginputkan data kata sandi dan mengosongkan data Kode Pengguna, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: - Kata sandi:test | Sistem akan menolak akses login dan menampilkan pesan "Masukkan kode pengguna Anda." | ✓ | |
| 4. | Menginputkan dengan kondisi salah satu data benar dan satu lagi salah, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: mimin (salah). Kata sandi:test (benar). | Sistem akan menolak akses login dan menampilkan pesan "Kode pengguna/kata sandi salah." | ✓ | |
| 5. | Menginputkan sintaks <i>SQL Injection</i> pada kedua data login, lalu mengklik tombol 'Masuk'. | Kode Pengguna: OR '1'='1 Kata sandi: OR '1'='1 | Sistem akan menolak akses login dan menampilkan pesan "Kode pengguna/kata sandi salah." | ✓ | |
| 6. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 1 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama. | ✓ | |

IV.2.1.1.2 FMOBIK-02 Menejemen Pengguna

Pengujian pada menejemen pengguna ada beberapa tahap, yaitu: pengujian untuk penambahan pengguna baru, ubah pengguna, dan penghapusan pengguna.

IV.2.1.1.2.1 Penambahan Pengguna Baru

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 1 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian administrasi. | ✓ | |
| 2. | Mengklik menu Pungguna pada menu Administrasi. | | Keluar daftar Pengguna Mobik | ✓ | |
| 3. | Mengklik tombol 'Tambah Pengguna' | | Keluar form Penambahan Pengguna Baru | ✓ | |
| 4. | Membiarkan semua form tidak terisi, lalu mengklik tombol 'Simpan' | Pilih Prodi: --Pilih Prodi-- Kode Pengguna: - Nama: - Tanggal Lahir: - Jenis Kelamin: - Level: - Angkatan: 2009 | Muncul indikator 'Pilih Prodi...!' dan Pilih Prodi tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | Kategori: - Kelas:- | | | |
| 5. | Memilih hanya Prodi saja dan form yang lain tidak terisi, atau mengisi Kode Pengguna kurang dari 6 digit, lalu mengklik tombol 'Simpan'. | Pilih Prodi: IF (benar) Kode Pengguna: -/ 3311 (salah) Nama: - Tanggal Lahir: - Jenis Kelamin: - Level: - Angkatan: 2009 Kategori: - Kelas:- | Muncul indikator 'Panjang kode pengguna harus antara 6 dan 10.' dan Kode Pengguna tersorot. | ✓ | |
| 6. | Membiarkan kosong form di bawah Kode Pengguna atau mengisi Nama kurang dari 3 digit, lalu klik 'Simpan' | Pilih Prodi: IF Kode Pengguna: 3311011007 (benar) Nama: -/Yo (salah) Tanggal Lahir: - Jenis Kelamin: - Level: - Angkatan: 2009 Kategori: - Kelas:- | Muncul indikator 'Panjang nama harus antara 3 dan 16.' dan Nama tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 7. | Membiarkan kosong form di bawah Nama, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 Nama: Yoga (benar) Tanggal Lahir: - Jenis Kelamin: - Level: - Angkatan: 2009 Kategori: - Kelas:- | Muncul indikator 'Isi tanggal lahir...!' dan Tanggal Lahir tersorot. | ✓ | |
| 8 | Mengklik form Tanggal Lahir, lalu pilih tanggal. | Tanggal Lahir: 1985-06-18 | Menampilkan daftar tanggal. Lalu menampilkannya pada form. | ✓ | |
| 9. | Membiarkan kosong form di bawah Tanggal Lahir, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 Nama: Yoga (benar) Tanggal Lahir: 1985-06-18 (benar) Jenis Kelamin: - Level: - Angkatan: 2009 Kategori: - Kelas:- | Muncul indikator 'Pilih jenis kelamin...!' | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 10. | Memilih Jenis Kelamin dan membiarkan kosong form di bawah Jenis Kelamin, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 Nama: Yoga (benar) Tanggal Lahir: 1985-06-18 (benar) Jenis Kelamin: Laki-Laki Level: - Angkatan: 2009 Kategori: - Kelas:- | Muncul indikator 'Pilih level pengguna...!' | ✓ | |
| 11. | Memilih Level dan membiarkan kosong form di bawah Level, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 Nama: Yoga (benar) Tanggal Lahir: 1985-06-18 (benar) Jenis Kelamin: Laki-Laki Level: Mahasiswa Angkatan: 2009 Kategori: - Kelas:- | Muncul indikator 'Pilih kategori...!' | ✓ | |
| 12. | Memilih Kategori dan membiarkan kosong form Kelas, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 | Muncul indikator 'Panjang nama harus antara 1 dan 3.' dan Kelas | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | Nama: Yoga (benar) Tanggal Lahir: 1985-06-18 Jenis Kelamin: Laki-Laki Level: Mahasiswa Angkatan: 2009 Kategori: Karyawan Kelas:- | tersorot. | | |
| 13. | Mengisi semua form, namun Kode Pengguna menggunakan selain angka saja, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 331101100a atau 3311@11007 (salah) Nama: Yoga (benar) Tanggal Lahir: 1985-06-18 Jenis Kelamin: Laki-Laki Level: Mahasiswa Angkatan: 2009 Kategori: Karyawan Kelas: A | Muncul indikator 'Kode Pengguna terdiri dari angka 0-9.' dan Kode Pengguna tersorot. | ✓ | |
| 14. | Mengisi semua form, namun Nama menggunakan selain huruf saja, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 (benar) Nama: Yog@ atau Yog4 (salah) | Muncul indikator 'Nama terdiri dari huruf saja.' dan Nama tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | Tanggal Lahir: 1985-06-18 Jenis Kelamin: Laki-Laki Level: Mahasiswa Angkatan: 2009 Kategori: Karyawan Kelas: A | | | |
| 15. | Mengisi semua form, namun Kelas menggunakan selain <i>symbol</i> , lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 (benar) Nama: Yoga Tanggal Lahir: 1985-06-18 Jenis Kelamin: Laki-Laki Level: Mahasiswa Angkatan: 2009 Kategori: Karyawan Kelas: #A | Muncul indikator 'Kelas terdiri dari a-z, 0-9.' dan Kelas tersorot. | ✓ | |
| 16. | Mengisi semua form dengan benar, lalu klik 'Simpan' | Pilih Prodi: IF (benar) Kode Pengguna: 3311011007 (benar) Nama: Yoga Tanggal Lahir: 1985-06-18 Jenis Kelamin: Laki-Laki | Muncul pop-up "Data telah disimpan" dan form akan dikosongkan serta muncul indikator 'Semua form isian ini dibutuhkan.' | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | Level: Mahasiswa Angkatan: 2009 Kategori: Karyawan Kelas: A | | | |
| 17. | Mengklik tombol 'Batal' atau tanda silang. | | Kembali ke Daftar Pengguna Baru dan terdapat data yang telah ditambah jika memang ada. | ✓ | |

IV.2.1.1.2.2 Ubah/Edit Pengguna

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---------------------------------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik <i>icon</i> gambar pensil pada tabel Daftar Pengguna Mobik. | | Sistem menampilkan form perubahan data pengguna beserta isinya. | ✓ | |
| 2. | Mengganti Kode Pengguna dengan yang sudah ada, lalu klik 'Simpan' | Kode Pengguna: 3311011006 (sudah ada) | Muncul indikator 'Kode Pengguna sudah ada.' | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 3. | Mengganti sebagian atau seluruh form, lalu klik 'Simpan' | Kode Pengguna: 3311011008 (benar) Nama: Sugeng Prayoga (ganti) Pilih Prodi: IF Angkatan: 2009 Level: Mahasiswa Kategori: Reguler (ganti) Kelas: B (ganti) | Muncul pop-up "Data telah disimpan" dan kembali ke halaman pengguna. Kode Pengguna dan Nama berubah sesuai yang telah di ubah. | ✓ | |

IV.2.1.1.2.3 Hapus Pengguna

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik satu <i>icon</i> gambar tong sampah pada data Kode Pengguna 1 di tabel Daftar Pengguna Mobik. | | Sistem menampilkan pertanyaan "Anda Yakin?" | ✓ | |
| 2. | Mengklik tombol Ya atau Tidak | Ya atau Tidak | Jika Ya; akan menampilkan <i>error</i> 'Tidak ada yang dihapus, Kode Pengguna '1' adalah Anda Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman pengguna. Jika Tidak; tidak terjadi apa-apa. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|-------------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 3. | Mengklik satu <i>icon</i> gambar tong sampah pada data Kode Pengguna 3311011008 di tabel Daftar Pengguna Mobik. | | Sistem menampilkan pertanyaan “Anda Yakin?” | ✓ | |
| 4. | Mengklik tombol Ya atau Tidak | Ya atau Tidak | Jika Ya; akan menampilkan <i>valid</i> ‘User ‘3311011008’ Sudah terhapus Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman pengguna. Jika Tidak; tidak terjadi apa-apa. | ✓ | |
| 5 | Klik tombol ‘Hapus’ | | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |
| 6. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>error</i> : Tidak ada data yang dihapus. Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman pengguna. Jika Cancel; kembali ke halaman pengguna. | ✓ | |
| 7. | Pilih beberapa pengguna yang akan dihapus lalu klik tombol ‘Hapus’ | Pilih 2 pengguna. | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--------------------------------|----------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 8. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>valid: 2</i> pengguna sudah dihapus Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman pengguna. Jika Cancel; kembali ke halaman pengguna. | ✓ | |

IV.2.1.1.3 FMOBIK-03 Menejemen Program Studi

Pengujian pada menejemen program studi (prodi) juga ada beberapa tahap, yaitu: pengujian untuk penambahan prodi baru, ubah prodi, dan penghapusan prodi.

IV.2.1.1.3.1 Penambahan Prodi Baru

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--------------------------------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 1 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian administrasi. | ✓ | |
| 2. | Mengklik menu Program Studi pada menu Administrasi. | | Keluar daftar Program Studi | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 3. | Mengklik tombol 'Tambah Prodi' | | Keluar form Penambahan Prodi Baru | ✓ | |
| 4. | Membiarkan semua form tidak terisi, lalu mengklik tombol 'Simpan' | Kode Prodi: - Nama Prodi: | Muncul indikator 'Panjang kode prodi harus antara 2 dan 4.' dan Kode Prodi tersorot. | ✓ | |
| 5. | Mengisi Kode Prodi kurang dari 2 digit atau lebih dari 4 digit, lalu mengklik tombol 'Simpan'. | Kode Prodi: A (salah) Nama Prodi: | Muncul indikator 'Panjang kode prodi harus antara 2 dan 4.' dan Kode Prodi tersorot. | ✓ | |
| 6. | Mengisi Kode Prodi dengan selain huruf, atau kombinasi huruf, angka dan <i>symbol</i> , lalu mengklik tombol 'Simpan'. | Kode Prodi: A1 / A1# (salah) Nama Prodi: | Muncul indikator 'Kode prodi terdiri dari A-Z' dan Kode Prodi tersorot. | ✓ | |
| 7. | Mengisi Kode Prodi dengan benar, lalu mengklik tombol 'Simpan'. | Kode Prodi: ELK (benar) Nama Prodi: | Muncul indikator 'Panjang nama prodi harus antara 3 dan 30.' dan Nama Prodi tersorot. | ✓ | |
| 8. | Mengisi Kode Prodi dengan benar dan mengisi Nama Prodi kurang dari 3 digit atau lebih dari 30 digit, lalu mengklik tombol 'Simpan'. | Kode Prodi: ELK (benar) Nama Prodi: El (salah) | Muncul indikator 'Panjang nama prodi harus antara 3 dan 30.' dan Nama Prodi tersorot. | ✓ | |
| 9. | Mengisi Kode Prodi dengan benar dan mengisi Nama Prodi selain huruf dan | Kode Prodi: ELK (benar) Nama Prodi: @\$% / E2\$ / 123 (salah) | Muncul indikator 'Nama prodi terdiri dari a-z, 0-9.' dan Nama Prodi tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | angka atau kombinasi huruf, angka dan <i>symbol</i> , atau angka saja, lalu mengklik tombol 'Simpan'. | | | | |
| 10. | Mengisi Kode Prodi dengan benar dan mengisi Nama Prodi dengan benar, lalu mengklik tombol 'Simpan'. | Kode Prodi: ELK (benar) Nama Prodi: Elektronika (benar) | Muncul pop-up "Data telah disimpan" dan form akan dikosongkan serta muncul indikator 'Semua form isian ini dibutuhkan.' | ✓ | |
| 11. | Mengklik tombol 'Batal' atau tanda silang. | | Kembali ke Program Studi dan terdapat data yang telah ditambah jika memang ada. | ✓ | |

IV.2.1.1.3.2 Ubah/Edit Prodi

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik <i>icon</i> gambar pensil pada tabel Daftar Program Studi. | | Sistem menampilkan form perubahan data prodi beserta isinya. | ✓ | |
| 2. | Ubah data, lalu klik 'Simpan' | Kode Prodi: EL (benar) Nama Prodi: Elektro (benar) | Muncul <i>valid</i> indikator 'Prodi EL Sudah diubah. Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman prodi. | ✓ | |

IV.2.1.1.3.3 Hapus Prodi

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|----------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik satu <i>icon</i> gambar tong sampah pada data Kode Prodi EL di tabel Daftar Program Studi. | | Sistem menampilkan pertanyaan “Anda Yakin?” | ✓ | |
| 2. | Mengklik tombol Ya atau Tidak | Ya atau Tidak | Jika Ya; akan menampilkan <i>valid</i> ‘Prodi ‘EL’ Sudah terhapus Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman prodi. Jika Tidak; tidak terjadi apa-apa. | ✓ | |
| 3. | Klik tombol ‘Hapus’ | | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |
| 4. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>error</i> : Tidak ada data yang dihapus. Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman prodi. Jika <i>Cancel</i> ; kembali ke halaman prodi. | ✓ | |
| 5. | Pilih beberapa prodi yang akan dihapus lalu klik tombol ‘Hapus’ | Pilih 2 prodi. | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |
| 6. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>valid</i> : 2 | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--------------------|-----------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | | prodi sudah dihapus <u>Klik</u> di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman prodi. Jika Cancel; kembali ke halaman prodi. | | |

IV.2.1.1.4 FMOBIK-04 Menejemen Mata Kuliah

Pengujian pada menejemen mata kuliah (makul) juga ada beberapa tahap, yaitu: pengujian untuk penambahan Makul baru, ubah Makul, dan penghapusan Makul.

IV.2.1.1.4.1 Penambahan Makul Baru

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--------------------------------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 1 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian administrasi. | ✓ | |
| 2. | Mengklik menu Mata Kuliah pada menu Administrasi. | | Keluar daftar Mata Kuliah | ✓ | |
| 3. | Mengklik tombol 'Tambah Makul' | | Keluar form Penambahan Makul Baru | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 4. | Membiarkan semua form tidak terisi, lalu mengklik tombol 'Simpan' | Pilih Prodi: --Pilih Prodi-- Kode Makul: Nama Makul: Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | Muncul indikator 'Pilih Prodi...!' dan Pilih Prodi tersorot. | ✓ | |
| 5. | Memilih hanya Prodi saja dan form yang lain tidak terisi, atau mengisi Kode Makul kurang dari 2 digit, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 1 (salah) Nama Makul: Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | Muncul indikator 'Panjang Kode makul harus antara 2 dan 5.' dan Kode Makul tersorot. | ✓ | |
| 6. | Memilih prodi serta mengisi Kode Makul dengan selain angka, atau kombinasi huruf, angka dan <i>symbol</i> , lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: A1/A#/@\$/AB (salah) Nama Makul: Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | Muncul indikator 'Kode Makul terdiri dari 0-9' dan Kode Makul tersorot. | ✓ | |
| 7. | Memilih prodi serta mengisi Kode Makul yang sudah ada, lalu | Pilih Prodi: ELK (benar) Kode Makul: 1001 (salah) | Muncul indikator 'Kode makul sudah ada.' dan Kode Makul tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | mengklik tombol 'Simpan'. | Nama Makul: Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | | | |
| 8. | Memilih prodi serta mengisi Kode Makul dengan benar, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | Muncul indikator 'Panjang nama Makul harus antara 3 dan 30.' dan Nama Makul tersorot. | ✓ | |
| 9. | Memilih prodi serta mengisi Kode Makul dengan benar dan mengisi Nama Makul kurang dari 3 digit atau lebih dari 30 digit, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Ar (salah) Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | Muncul indikator 'Panjang nama Makul harus antara 3 dan 30.' dan Nama Makul tersorot. | ✓ | |
| 10. | Memilih prodi serta mengisi Kode Makul dengan benar dan mengisi Nama Makul selain huruf dan angka atau kombinasi huruf, angka dan <i>symbol</i> , | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Ar@/123/#@\$ Jenis Makul: --Pilih Jenis Makul-- | Muncul indikator 'Nama Makul terdiri dari a-z, 0-9.' dan Nama Makul tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| | atau angka saja, lalu mengklik tombol 'Simpan'. | Jumlah SKS: Semester: | | | |
| 11. | Memilih prodi serta mengisi Kode Makul dengan benar dan mengisi Nama Makul dengan benar, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Arus Lemah (benar) Jenis Makul: --Pilih Jenis Makul-- Jumlah SKS: Semester: | Muncul indikator 'Pilih jenis makul...!' dan Jenis Makul tersorot. | ✓ | |
| 12. | Memilih prodi serta mengisi Kode Makul dengan benar dan mengisi Nama Makul dengan benar juga memilih Jenis Makul, atau mengisi SKS selain angka, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Arus Lemah (benar) Jenis Makul: EL (benar) Jumlah SKS: a@ (salah) Semester: | Muncul indikator 'SKS terdiri dari 0-9.' Dan SKS tersorot. | ✓ | |
| 13. | Memilih prodi serta mengisi Kode Makul dengan benar, mengisi Nama Makul dengan benar, memilih Jenis Makul, mengisi SKS dengan benar atau mengisi semester selain angka, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Arus Lemah (benar) Jenis Makul: EL (benar) Jumlah SKS: 2 (benar) Semester: A\$ (salah) | Muncul indikator 'Semester terdiri dari 0-9.' Dan Semester tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 14. | Mengisi seluruh form dengan benar, lalu mengklik tombol 'Simpan'. | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Arus Lemah (benar) Jenis Makul: EL (benar) Jumlah SKS: 2 (benar) Semester: 4 (benar) | Muncul pop-up "Data telah disimpan" dan form akan dikosongkan serta muncul indikator 'Semua form isian ini dibutuhkan.' | ✓ | |
| 15. | Mengklik tombol 'Batal' atau tanda silang. | | Kembali ke Mata Kuliah dan terdapat data yang telah ditambah jika memang ada. | ✓ | |

IV.2.1.1.4.2 Ubah/Edit Makul

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik <i>icon</i> gambar pensil pada tabel Daftar Mata Kuliah. | | Sistem menampilkan form perubahan data Makul beserta isinya. | ✓ | |
| 2. | Ubah data, lalu klik 'Simpan' | Pilih Prodi: ELK (benar) Kode Makul: 3001 (benar) Nama Makul: Arus Lemah 2 (ganti) Jenis Makul: EL (benar) Jumlah SKS: 2 (benar) Semester: 4 (benar) | Muncul pop-up "Data telah disimpan" dan halaman akan kembali ke halaman Makul. | ✓ | |

IV.2.1.1.4.3 Hapus Makul

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|----------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik satu <i>icon</i> gambar tong sampah pada data Kode Makul 3001 di tabel Daftar Mata Kuliah. | | Sistem menampilkan pertanyaan “Anda Yakin?” | ✓ | |
| 2. | Mengklik tombol Ya atau Tidak | Ya atau Tidak | Jika Ya; akan menampilkan <i>valid</i> ‘Makul ‘3001’ Sudah terhapus Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman Makul. Jika Tidak; tidak terjadi apa-apa. | ✓ | |
| 3. | Klik tombol ‘Hapus’ | | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |
| 4. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>error</i> : Tidak ada data yang dihapus. Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman Makul. Jika <i>Cancel</i> ; kembali ke halaman Makul. | ✓ | |
| 5. | Pilih beberapa Makul yang akan dihapus lalu klik tombol ‘Hapus’ | Pilih 2 Makul. | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--------------------------------|----------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 6. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>valid: 2</i> Makul sudah dihapus Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Makul. Jika Cancel; kembali ke halaman Makul. | ✓ | |

IV.2.1.1.5 FMOBIK-05 Level User

Pengujian untuk *level user* terdapat pada pengujian penambahan pengguna.

IV.2.1.1.6 FMOBIK-06 Ganti Kata Kunci

Pada pengujian ganti kata kunci berlaku untuk semua user.

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mencentang “Ganti Kata Kunci” lalu mengklik tombol ‘Masuk’. | Kode Pengguna: 1 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman ganti kata kunci. | ✓ | |
| 2. | Mengosongkan semua isian data ganti kata kunci, lalu langsung mengklik tombol ‘Ganti’. | Kata Sandi Baru: - Ketik Ulang Kata Sandi: - | Tombol ‘Ganti’ tidak berfungsi. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 3. | Hanya mengisi data Kata Sandi Baru 123456 dan mengosongkan data Ketik Ulang Kata Sandi, lalu langsung mengklik tombol 'Ganti'. | Kata Sandi Baru: 123456 Ketik Ulang Kata Sandi: - | Tombol 'Ganti' tidak berfungsi. | ✓ | |
| 4. | Hanya mengisi data Ketik Ulang Kata Sandi dan mengosongkan data Kata Sandi Baru, lalu langsung mengklik tombol 'Ganti'. | Kata Sandi Baru: - Ketik Ulang Kata Sandi: 123456 | Tombol 'Ganti' tidak berfungsi. Dan keluar tanda silang. | ✓ | |
| 5. | Menginputkan Ketik Ulang Kata Sandi tidak sesuai dengan Kata Sandi Baru. | Kata Sandi Baru: 123456 (benar) Ketik Ulang Kata Sandi: 1234 (salah) | Tombol 'Ganti' tidak berfungsi. Dan keluar tanda silang. | ✓ | |
| 6. | Menginputkan Kata Sandi Baru tidak sesuai dengan Ketik Ulang Kata Sandi. | Kata Sandi Baru: 1234 (salah) Ketik Ulang Kata Sandi: 123456 (benar) | Tombol 'Ganti' tidak berfungsi. Dan keluar tanda silang. | ✓ | |
| 7. | Menginputkan Ketik Ulang Kata Sandi sesuai dengan Kata Sandi Baru. Lalu mengklik tombol 'Ganti'. | Kata Sandi Baru: 123456 (benar) Ketik Ulang Kata Sandi: 123456 (benar) | Muncul tanda centang. Dan sistem menerima akses ganti kata kunci dan kemudian langsung menampilkan halaman utama. | ✓ | |
| 8. | Mengklik tombol 'Batal' | Kata Sandi Baru: 123456 (benar) Ketik Ulang Kata Sandi: 123456 (benar) | Menampilkan halaman utama. | ✓ | |

IV.2.1.1.7 FMOBIK-07 Menejemen Bank Soal

Pengujian pada menejemen bank soal ada beberapa tahap, yaitu: pengujian untuk penambahan soal baru, ubah soal, dan penghapusan soal.

IV.2.1.1.7.1 Penambahan Soal Baru

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 2 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian dosen. | ✓ | |
| 2. | Mengklik menu Bank Soal pada menu Dosen | | Keluar daftar seluruh bank soal | ✓ | |
| 3. | Mengklik tombol 'Cari' | Pilih Prodi: --Pilih Prodi-- Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 4. | Memilih Prodi lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 5. | Memilih Prodi dan Makul lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: 1001 | Muncul data soal untuk makul 1001 dan tombol 'Tambah Soal' serta 'Import Soal' aktif. | ✓ | |
| 6. | Mengklik tombol 'Tambah Soal' | | Muncul form untuk menambahkan soal baru dengan kode soal otomatis. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 7. | Membiarkan seluruh form kosong lalu mengklik tombol 'Simpan' | Text Soal: - Pilihan: <input type="radio"/> a. <input type="radio"/> b. <input type="radio"/> c. <input type="radio"/> d. Bobot: <input type="radio"/> mudah <input type="radio"/> sedang <input type="radio"/> sulit | Muncul indikator 'Text soal harus diisi.' dan Text soal tersorot. | ✓ | |
| 8. | Mengisi Text Soal, lalu mengklik tombol 'Simpan'. | Text Soal: $9+6=?$ Pilihan: <input type="radio"/> a. <input type="radio"/> b. <input type="radio"/> c. <input type="radio"/> d. Bobot: <input type="radio"/> mudah <input type="radio"/> sedang <input type="radio"/> sulit | Muncul indikator 'Pilihan a harus diisi.' dan Pilihan a tersorot. | ✓ | |
| 9. | Ulangi langkah 8 untuk pilihan-pilihan dibawahnya. | | Muncul indikator 'Pilihan a/b/c/d harus diisi.' dan Pilihan a/b/c/d tersorot. | ✓ | |
| 10. | Mengisi Text Soal dan mengisi seluruh pilihan, lalu mengklik tombol 'Simpan'. | Text Soal: $9+6=?$ Pilihan: <input type="radio"/> a. 3 <input type="radio"/> b. 5 <input type="radio"/> c. 15 | Muncul indikator 'Pilih jawaban...!' | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | <p><input type="radio"/>d. 54</p> <p>Bobot: <input type="radio"/>mudah <input type="radio"/>sedang <input type="radio"/>sulit</p> | | | |
| 11. | Mengisi Text Soal dan mengisi seluruh pilihan dan memilih salah satu jawaban, lalu mengklik tombol 'Simpan'. | <p>Text Soal: $9+6=?$</p> <p>Pilihan: <input type="radio"/>a. 3</p> <p><input type="radio"/>b. 5</p> <p><input checked="" type="radio"/>c. 15</p> <p><input type="radio"/>d. 54</p> <p>Bobot:</p> <p><input type="radio"/>mudah <input type="radio"/>sedang <input type="radio"/>sulit</p> | Muncul indikator 'Pilih bobot soal...!' | ✓ | |
| 12. | Mengisi Text Soal dan mengisi seluruh pilihan, memilih salah satu jawaban, dan memilih bobot, lalu mengklik tombol 'Simpan'. | <p>Text Soal: $9+6=?$</p> <p>Pilihan: <input type="radio"/>a. 3</p> <p><input type="radio"/>b. 5</p> <p><input checked="" type="radio"/>c. 15</p> <p><input type="radio"/>d. 54</p> <p>Bobot:</p> <p><input checked="" type="radio"/>mudah <input type="radio"/>sedang <input type="radio"/>sulit</p> | Muncul pop-up "Data telah disimpan" dan form akan dikosongkan serta muncul indikator 'Semua form isian ini dibutuhkan.' dan kode soal otomatis bertambah. | ✓ | |
| 13. | Mengklik tombol 'Batal' atau tanda silang. | | Kembali ke Mata Kuliah dan terdapat data yang telah ditambah jika memang ada. | ✓ | |
| 14. | Mengklik tombol 'Import Soal' | | Muncul form untuk import soal. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|----------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 15. | Mengklik tombol 'Browse' atau klik form. | | Muncul explore window untuk memilih <i>file</i> yang akan di import. | ✓ | |
| 16. | Memilih <i>file</i> yang bukan berextension CSV, lalu klik tombol 'Import' | File: test.txt | Akan menampilkan <i>error</i> : Bukan format file csv. Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Bank Soal. | ✓ | |
| 17. | Memilih <i>file</i> CSV yang sesuai dengan format yang ditampilkan pada form, lalu klik tombol 'Import' | File: test.csv | Akan menampilkan <i>valid</i> : 'jumlah' soal berhasil di-import Klik di sini untuk kembali.' Dalam 5 detik halaman akan kembali ke halaman Bank Soal yang sudah ada soal baru. Kode soal otomatis. | ✓ | |

IV.2.1.1.7.2 Ubah/Edit Soal

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik <i>icon</i> gambar pensil pada tabel Daftar Bank Soal. | | Sistem menampilkan form perubahan data Soal beserta isinya. | ✓ | |
| 2. | Ubah data, lalu klik 'Simpan' | Text Soal: 9-6=? Pilihan: <input type="radio"/> a.3 | Muncul pop-up "Data telah disimpan" dan halaman akan kembali ke halaman Soal. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--------------------|---|-----------------------|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | Ob.5 Oc.15 Od.54 Bobot: <input checked="" type="radio"/> mudah <input type="radio"/> sedang <input type="radio"/> sulit | | | |

IV.2.1.1.7.3 Hapus Soal

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik satu <i>icon</i> gambar tong sampah pada data Kode Soal 10010000033 di tabel Daftar Bank Soal. | | Sistem menampilkan pertanyaan “Anda Yakin?” | ✓ | |
| 2. | Mengklik tombol Ya atau Tidak | Ya atau Tidak | Jika Ya; akan menampilkan <i>valid</i> ‘Soal Sudah terhapus Klik di sini untuk kembali’. Dalam 5 detik halaman akan kembali ke halaman Soal. Jika Tidak; tidak terjadi apa-apa. | ✓ | |
| 3. | Klik tombol ‘Hapus’ | | Muncul ‘Anda yakin ingin menghapus?’ | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|----------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 4. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>error</i> : Tidak ada data yang dihapus. Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Soal. Jika <i>Cancel</i> ; kembali ke halaman Soal. | ✓ | |
| 5. | Pilih beberapa Soal yang akan dihapus lalu klik tombol 'Hapus' | Pilih 2 Soal. | Muncul 'Anda yakin ingin menghapus?' | ✓ | |
| 6. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>valid</i> : 2 Soal sudah dihapus Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Soal. Jika <i>Cancel</i> ; kembali ke halaman Soal. | ✓ | |

IV.2.1.1.8 FMOBIK-08 Setup Kuis

Pengujian pada *setup* kuis ada beberapa tahap, yaitu: pengujian untuk penambahan kuis baru, ubah kuis, dan penghapusan kuis.

IV.2.1.1.8.1 Penambahan Soal Baru

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 2 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian dosen. | ✓ | |
| 2. | Mengklik menu Kuis pada menu Dosen | | Keluar daftar seluruh kuis | ✓ | |
| 3. | Mengklik tombol 'Cari' | Pilih Prodi: --Pilih Prodi-- Pilih Makul: --Pilih Makul-- Pilih Kategori: -- Pilih Kategori-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 4. | Memilih Prodi lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: --Pilih Makul-- Pilih Kategori: -- Pilih Kategori-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 5. | Memilih Prodi dan Makul lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: 1001 Pilih Kategori: -- Pilih Kategori-- | Muncul data lembar soal untuk makul 1001 dan tombol 'Tambah Lembar Soal' aktif. | ✓ | |
| 6. | Mengklik tombol 'Tambah Lembar Soal' | | Muncul form untuk menambahkan lembar soal/kuis baru. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 7. | Membiarkan seluruh form kosong atau mengisi Kode Lembar Soal kurang dari 6 digit atau lebih dari 20 digit, lalu mengklik tombol 'Simpan' | Kode Lembar Soal: - / 123 (salah) Kata Sandi: - Jumlah Soal: - Lama Kuis: - | Muncul indikator 'Panjang kode lembar soal harus antara 6 dan 20.' dan Kode Lembar Soal tersorot. | ✓ | |
| 8. | Mengisi Kode Lembar Soal dengan yang sudah ada, kemudian klik di luar form, lalu mengklik tombol 'Simpan' | Kode Lembar Soal: doraemon (salah) Kata Sandi: - Jumlah Soal: - Lama Kuis: - | Muncul indikator 'Kode lembar soal sudah ada.' dan Kode Lembar Soal tersorot. | ✓ | |
| 9. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto', lalu mengklik tombol 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: - Jumlah Soal: - Lama Kuis: - | Muncul indikator 'Panjang kata sandi lembar soal harus antara 6 dan 20.' dan Kata Sandi tersorot. | ✓ | |
| 10. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Membiarkan form di bawahnya kosong atau mengisi Kata Sandi kurang dari 6 digit atau lebih dari 20 digit lalu mengklik tombol 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: - /123 (salah) Jumlah Soal: - Lama Kuis: - | Muncul indikator 'Panjang kata sandi lembar soal harus antara 6 dan 20.' dan Kata Sandi tersorot. | ✓ | |
| 11. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Mengisi Kata Sandi dengan | Kode Lembar Soal: polbat (benar) Kata Sandi: - | Muncul indikator 'Panjang jumlah soal harus antara 1 dan 3.' dan Jumlah Soal | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | benar. Membiarkan form di bawahnya kosong, lalu mengklik tombol 'Simpan' | Jumlah Soal: - Lama Kuis: - | tersorot. Serta tombol Pilih Soal tidak aktif. | | |
| 12. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Mengisi Kata Sandi dengan benar dan Jumlah Soal dengan selain angka. Membiarkan form di bawahnya kosong, lalu mengklik tombol 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: polbat (benar) Jumlah Soal: a (salah) Lama Kuis: - | Muncul indikator 'Jumlah soal terdiri dari angka 0-9.' dan Jumlah Soal tersorot. | ✓ | |
| 13. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Mengisi Kata Sandi dan Jumlah Soal dengan benar. Membiarkan form di bawahnya kosong, lalu mengklik tombol 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: polbat (benar) Jumlah Soal: 5 (benar) Lama Kuis: - | Muncul indikator 'Jumlah soal yang dipilih kurang: 0 dari 5.' dan Jumlah Soal tersorot. | ✓ | |
| 14. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Mengisi Kata Sandi dan Jumlah Soal dengan benar. Membiarkan form di bawahnya kosong, lalu mengklik tombol 'Pilih Soal' | Kode Lembar Soal: polbat (benar) Kata Sandi: polbat (benar) Jumlah Soal: 5 (benar) Lama Kuis: - | Muncul daftar soal sesuai makul yang dipilih. | ✓ | |
| 15. | Memilih soal kurang dari jumlah soal, lalu klik 'Simpan' pada daftar Soal, lalu klik 'Simpan'. | Pilih beberapa soal kurang dari 5. | Daftar soal tertutup. Muncul indikator 'Jumlah soal yang dipilih kurang: 2 dari 5.' dan Jumlah Soal tersorot. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 16. | Ulangi langkah no. 14, pilih soal sesuai jumlah soal, lalu klik 'Simpan' pada daftar Soal, lalu klik 'Simpan'. | Pilih 5 soal. | Muncul indikator 'Panjang lama kuis harus antara 1 dan 3.' dan Lama Kuis tersorot. | ✓ | |
| 17. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Mengisi Kata Sandi dan Jumlah Soal dengan benar, mengisi Lama Kuis selain angka, lalu mengklik tombol 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: polbat (benar) Jumlah Soal: 5 (benar) Lama Kuis: A/@/# (salah) | Muncul indikator 'Lama kuis terdiri dari angka 0-9.' dan Lama Kuis tersorot. | ✓ | |
| 18. | Mengisi Kode Lembar Soal dengan benar atau klik 'Auto'. Mengisi Kata Sandi, Jumlah Soal, mengisi dan Lama Kuis dengan benar, set tanggal mulai dan akhir kuis, pilih peserta dengan mengklik Peserta Kuis, lalu mengklik tombol 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: polbat (benar) Jumlah Soal: 5 (benar) Lama Kuis: 30 (benar) Tanggal mulai kuis: auto Tanggal mulai kuis: auto | Muncul pop-up "Data telah disimpan" dan form akan dikosongkan serta muncul indikator 'Semua form isian ini dibutuhkan.' | ✓ | |
| 19. | Mengklik tombol 'Batal' atau tanda silang. | | Kembali ke daftar Kuis dan terdapat data yang telah ditambah jika memang ada. | ✓ | |

IV.2.1.1.8.2 Ubah/Edit Soal

Ada beberapa prosedur yang dilakukan oleh sistem, yaitu: Jika kuis tersebut telah ada yang mengerjakan maka yang di ubah oleh sistem hanya sebatas Kata Sandi, Lama Kuis, Tanggal Mulai Kuis, Tanggal Akhir Kuis, menambah peserta kuis dan mengurangi peserta kuis yang belum melaksanakan kuis.

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik <i>icon</i> gambar pensil pada tabel Daftar Kuis. | | Sistem menampilkan form perubahan data lembar soal beserta isinya. | ✓ | |
| 2. | Ubah data, lalu klik 'Simpan' | Kode Lembar Soal: polbat (benar) Kata Sandi: polbat (benar) Jumlah Soal: 5 (benar) Lama Kuis: 50 (ganti) Tanggal mulai kuis: auto Tanggal mulai kuis: auto | Muncul pop-up "Data telah disimpan" dan halaman akan kembali ke halaman kuis. | ✓ | |

IV.2.1.1.8.3 Hapus Soal

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengklik satu <i>icon</i> gambar tong sampah pada data Kode Lembar Soal polbat di tabel Daftar Kuis. | | Sistem menampilkan pertanyaan "Anda Yakin?" | ✓ | |
| 2. | Mengklik tombol Ya atau Tidak | Ya atau Tidak | Jika Ya; akan menampilkan <i>valid</i> 'Soal Sudah terhapus Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Kuis. Jika Tidak; tidak terjadi apa-apa. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|----------------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 3. | Klik tombol 'Hapus' | | Muncul 'Anda yakin ingin menghapus?' | ✓ | |
| 4. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>error</i> : Tidak ada data yang dihapus. Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Kuis. Jika <i>Cancel</i> ; kembali ke halaman Kuis. | ✓ | |
| 5. | Pilih beberapa lembar soal yang akan dihapus lalu klik tombol 'Hapus' | Pilih 2 lembar soal. | Muncul 'Anda yakin ingin menghapus?' | ✓ | |
| 6. | Mengklik tombol OK atau Cancel | OK atau Cancel | Jika OK; akan menampilkan <i>valid</i> : 2 lembar soal sudah dihapus Klik di sini untuk kembali'. Dalam 5 detik halaman akan kembali ke halaman Kuis. Jika <i>Cancel</i> ; kembali ke halaman Kuis. | ✓ | |

IV.2.1.1.9 FMOBIK-13 Pemberian Nilai

Pengujian pada nilai hanya menampilkan hasil kuis dan nilai karena telah diproses oleh sistem.

IV.2.1.1.9.1 Melihat Nilai

| No | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 2 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian dosen. | ✓ | |
| 2. | Mengklik menu Lembar Jawaban pada menu Dosen | | Keluar daftar seluruh kuis | ✓ | |
| 3. | Mengklik tombol 'Cari' | Pilih Prodi: --Pilih Prodi-- Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 4. | Memilih Prodi lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 5. | Memilih Prodi dan Makul lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: 1001 (benar) | Muncul daftar lembar soal untuk makul 1001. | ✓ | |
| 6. | Mengklik kode lembar jawaban. | Klik doraemon | Muncul Peserta kuis Doraemon dan nilai masing-masing peserta. | ✓ | |
| 7. | Mengklik tombol 'Export ke Excel' | | Muncul <i>download</i> . | ✓ | |
| 8. | Mengklik tanda silang. | | Kembali ke daftar lembar jawaban. | ✓ | |

IV.2.1.1.10 FMOBIK-14 Mencetak Soal

| No | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 2 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian dosen. | ✓ | |
| 2. | Mengklik menu Kuis pada menu Dosen | | Keluar daftar seluruh lembar soal | ✓ | |
| 3. | Mengklik tombol 'Cari' | Pilih Prodi: --Pilih Prodi-- Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 4. | Memilih Prodi lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 5. | Memilih Prodi dan Makul lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: 1001 (benar) | Muncul daftar lembar soal untuk makul 1001. | ✓ | |
| 6. | Mengklik kode lembar jawaban. | Klik doraemon | Akan membuka Soal dan kunci jawaban dalam bentuk PDF format pada window baru. | ✓ | |

IV.2.1.1.11 FMOBIK-15 Mencetak Jawaban

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 2 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian dosen. | ✓ | |
| 2. | Mengklik menu Lembar Jawaban pada menu Dosen | | Keluar daftar seluruh kuis | ✓ | |
| 3. | Mengklik tombol 'Cari' | Pilih Prodi: --Pilih Prodi-- Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 4. | Memilih Prodi lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih pilih prodi dan makul.' | ✓ | |
| 5. | Memilih Prodi dan Makul lalu mengklik tombol 'Cari' | Pilih Prodi: IF (benar) Pilih Makul: 1001 (benar) | Muncul daftar lembar soal untuk makul 1001. | ✓ | |
| 6. | Mengklik kode lembar jawaban. | Klik doraemon | Muncul Peserta kuis Doraemon dan nilai masing-masing peserta. | ✓ | |
| 7. | Mengklik 'cetak' | | Akan membuka lembar jawaban satu peserta saja dalam bentuk PDF format pada window baru. | ✓ | |
| 19. | Mengklik tanda silang. | | Kembali ke daftar lembar jawaban. | ✓ | |
| 20. | Mengklik 'cetak semua' pada daftar lembar jawaban. | Klik 'cetak semua' pada baris lembar jawaban 'doraemon' | Akan membuka lembar jawaban seluruh peserta kuis doraemon dalam bentuk PDF format pada window baru. | ✓ | |

IV.2.1.1.12 FMOBIK-16 Mahasiswa Melihat Nilai

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 3 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian mahasiswa. | ✓ | |
| 2. | Mengklik menu Review Kuis pada menu Mahasiswa | | Keluar daftar kuis dari seluruh makul beserta nilainya. | ✓ | |
| 3. | Mengklik tombol 'Cari' | Pilih Makul: --Pilih Makul-- | Muncul indikator 'Silahkan pilih makul.' | ✓ | |
| 4. | Memilih Makul lalu mengklik tombol 'Cari' | Pilih Makul: 1001 (benar) | Muncul daftar lembar soal untuk makul 1001. | ✓ | |
| 5. | Klik Review | Klik 'cetak semua' pada baris lembar jawaban 'doraemon' | Akan membuka lembar jawaban kuis doraemon dalam bentuk PDF format pada window baru. | ✓ | |

IV.2.1.1.13 FMOBIK-17 LogOut

Pengujian logout berlaku untuk semua pengguna.

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--------------------------------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 3 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman utama bagian mahasiswa. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--------------------|-----------|---------------------------|-----------------|-------|
| | | | | Berhasil | Gagal |
| 2. | Mengklik keluar. | | Kembali ke halaman Login. | ✓ | |

IV.2.1.2 Black Box Testing pada Android

Berikut ini strategi-strategi *black box testing* pada *Android*:

IV.2.1.2.1 FMOBIK-01 Testing [Login]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengosongkan semua isian data login, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: - Kata kunci: - | Sistem tidak akan mengizinkan pengguna untuk dapat menekan tombol login. | ✓ | |
| 2. | Hanya menginputkan data Kode Pengguna dan mengosongkan data kata sandi, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: 3 Kata kunci: - | Sistem akan menolak akses login dan menampilkan pesan "Nama pengguna/kata kunci salah." | ✓ | |
| 3. | Hanya menginputkan data kata kunci dan mengosongkan data Kode Pengguna, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: - Kata kunci : test | Sistem tidak akan mengizinkan pengguna untuk dapat menekan tombol login. | ✓ | |
| 4. | Menginputkan dengan kondisi salah satu data benar dan satu lagi salah, lalu langsung mengklik tombol 'Masuk'. | Kode Pengguna: mimin (salah). Kata sandi: test (benar). | Sistem akan menolak akses login dan menampilkan pesan "Nama pengguna/kata kunci salah." | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 5. | Menginputkan sintaks <i>SQL Injection</i> pada kedua data login, lalu mengklik tombol 'Masuk'. | Kode Pengguna: ' OR '1'='1 Kata sandi: ' OR '1'='1 | Sistem akan menolak akses login dan menampilkan pesan "Nama pengguna/kata kunci salah." | ✓ | |
| 6. | Menginputkan data login yang benar, lalu mengklik tombol 'Masuk'. | Kode Pengguna: 3 Kata sandi: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman menu utama. | ✓ | |
| 7. | Menekan tombol 'Keluar' | | Menampilkan jendela dialog untuk mengkonfirmasi apakah sudah yakin akan keluar dari aplikasi | ✓ | |
| 8. | Menekan tombol 'Keluar' | Memilih tombol 'Tidak' | Kembali ke layar login | ✓ | |
| 9. | Menekan tombol 'Keluar' | Memilih tombol 'Ya' | Keluar dari aplikasi Mobik | ✓ | |

IV.2.1.2.2 FMOBIK-06 Testing [Ganti Kata Kunci]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|-----------------------------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Mengosongkan semua isian data login, lalu langsung mengklik tombol 'Ganti Kata Kunci' | Kode Pengguna: - Kata kunci: - | Tombol 'Ganti Kata Kunci' tidak akan ditampilkan. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 2. | Menginputkan data login yang benar, lalu menekan tombol “Ganti Kata Kunci” | Kode Pengguna: 3 Kata kunci: test | Sistem menerima akses login dan kemudian langsung menampilkan halaman ganti kata kunci. | ✓ | |
| 3. | Mengosongkan semua isian data ganti kata kunci, lalu langsung mengklik tombol ‘Simpan’. | Kata kunci Baru: - Konfirmasi Ulang: - | Tombol ‘Simpan’ tidak ditampilkan. Label status menampilkan pesan ‘Masukkan kata kunci baru’ | ✓ | |
| 4. | Hanya mengisi data Kata Kunci Baru 123456 dan mengosongkan data Konfirmasi Ulang, lalu langsung mengklik tombol ‘Simpan’. | Kata kunci Baru: 123456 Konfirmasi Ulang: - | Tombol ‘Simpan’ tidak ditampilkan. Label status menampilkan ‘Maaf, 'Kata Kunci' baru anda belum sama!’ | ✓ | |
| 5. | Hanya mengisi data Konfirmasi Ulang dan mengosongkan data Kata Kunci Baru, lalu langsung mengklik tombol ‘Simpan’. | Kata Kunci Baru: - Konfirmasi Ulang: 123456 | Tombol ‘Simpan’ tidak ditampilkan. Label status menampilkan ‘Maaf, 'Kata Kunci' baru anda belum sama!’ | ✓ | |
| 6. | Menginputkan Konfirmasi Ulang tidak sesuai dengan Kata Kunci Baru. | Kata Kunci Baru: 123456 (benar) Konfirmasi Ulang: 1234 (salah) | Tombol ‘Simpan’ tidak ditampilkan. Label status menampilkan ‘Maaf, 'Kata Kunci' baru anda belum sama!’ | ✓ | |
| 7. | Menginputkan Kata Kunci Baru tidak sesuai dengan Ketik Konfirmasi Ulang. | Kata Kunci Baru: 1234 (salah) Konfirmasi Ulang: 123456 (benar) | Tombol ‘Simpan’ tidak ditampilkan. Label status menampilkan ‘Maaf, 'Kata Kunci' baru anda belum sama!’ | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 8. | Menginputkan Konfirmasi Ulang sesuai dengan Kata Kunci Baru. Lalu mengklik tombol 'Simpan'. | Kata Sandi Baru: 123456 (benar) Ketik Ulang Kata Sandi: 123456 (benar) | Tombol 'Simpan' akan ditampilkan. Label status menampilkan 'Kata Kunci' baru anda cocOK! Keluar pesan: Ganti Kata Kunci: Sukses | ✓ | |
| 9. | Mengklik tombol 'Batal' | | Menampilkan halaman utama. | ✓ | |
| 10. | Mengklik tombol 'Back' pada android button | | Kembali ke layar login | ✓ | |

IV.2.1.2.3 FMOBIK-10 - FMOBIK-12 Testing [Menu Utama Mahasiswa]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|------------------------|--------------------------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Setelah login berhasil | | Sistem Aplikasi akan menampilkan daftar menu berupa tombol: <ol style="list-style-type: none"> 1. Pilih Matakuliah 2. Kuis 3. Review Kuis 4. Review Nilai 5. Keluar | ✓ | |
| 2. | Setelah login berhasil | Tidak memilih matakuliah | Sistem Aplikasi akan menampilkan daftar menu berupa tombol: <ol style="list-style-type: none"> 1. Pilih Matakuliah (dapat dipilih) | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|------------------------|-------------------------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | | 2. Kuis (tidak dapat dipilih) 3. Review Kuis (tidak dapat dipilih) 4. Review Nilai (tidak dapat dipilih) 5. Keluar (dapat dipilih) | | |
| 3. | Setelah login berhasil | Memilih salah satu matakuliah | Sistem Aplikasi akan menampilkan daftar menu berupa tombol: 1. Pilih Matakuliah (dapat dipilih) 2. Kuis (dapat dipilih) 3. Review Kuis (dapat dipilih) 4. Review Nilai (dapat dipilih) 5. Keluar (dapat dipilih) | ✓ | |

IV.2.1.2.3.1 Testing [1. Pilih Mata Kuliah]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menekan tombol '1.Pilih Mata Kuliah' pada menu utama mahasiswa | | Sistem Aplikasi akan menampilkan daftar mata kuliah yang tersedia untuk mahasiswa yang sedang online | ✓ | |
| 2. | Menekan tombol '1.Pilih Mata Kuliah' pada menu utama mahasiswa | Menekan tombol 'Back' pada tombol android | Sistem Aplikasi tidak akan mengambil pilihan yang tersedia | ✓ | |
| 3. | Menekan tombol '1.Pilih Mata | Memilih salah satu pilihan dari | Sistem Aplikasi akan mengambil pilihan yang di | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|-----------------------------------|----------------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | Kuliah' pada menu utama mahasiswa | daftar pilihan | pilih pengguna, dan menampilkan matakuliah yang terpilih dibawah judul menu. Dan tombol-tombol berikut akan dapat dipilih: 2. Kuis 3. Review Kuis 4. Review Nilai | | |

IV.2.1.2.3.2 Testing [2. Kuis]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menekan tombol '2.Kuis' pada menu utama mahasiswa | | Sistem Aplikasi akan menampilkan formulir pengisian untuk mengisi Kode Lembar Soal dan Kata Kunci Lembar Soal | ✓ | |
| 2. | Menekan tombol 'Mulai Kuis' tanpa mengisi formulir isian | Kode Lembar Soal: - Kata Kunci Lembar Soal: - | Sistem Aplikasi tidak akan mengijinkan tombol Mulai Kuis untuk di akses. | ✓ | |
| 3. | Menekan tombol 'Batal' tanpa mengisi formulir isian | Kode Lembar Soal: - Kata Kunci Lembar Soal: - | Sistem Aplikasi akan kembali ke menu utama Mahasiswa. | ✓ | |
| 4. | Menekan tombol 'Mulai Kuis' dan menekan tombol "Daftar Lembar Soal" | Kode Lembar Soal: - Kata Kunci Lembar Soal: - | Sistem Aplikasi akan menampilkan daftar lembar soal yang tersedia untuk mahasiswa | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | Menekan tombol “Daftar Lembar Soal” | tersebut. | | |
| 5. | Menekan tombol ‘Mulai Kuis’ dan memilih pilihan dalam “Daftar Lembar Soal” | Kode Lembar Soal: doraemon Kata Kunci Lembar Soal: 123456 Menekan tombol “Daftar Lembar Soal” dan Memilih salah satu pilihan yang tersedia | Sistem Aplikasi akan menampilkan pilihan pada isian Kode Lembar Soal. | ✓ | |
| 6. | Menekan tombol ‘Mulai Kuis’ dengan hanya mengisi Kode Lembar Soal saja | Kode Lembar Soal: doraemon Kata Kunci Lembar Soal: - | Sistem Aplikasi akan menampilkan pesan : “Terjadi kesalahan!!! Periksa kembali...” | ✓ | |
| 7. | Menekan tombol ‘Mulai Kuis’ dengan hanya mengisi Kata Kunci Kode Lembar Soal saja | Kode Lembar Soal: - Kata Kunci Lembar Soal: 123456 | Sistem Aplikasi akan menampilkan pesan : “Terjadi kesalahan!!! Periksa kembali...” | ✓ | |
| 8. | Menekan tombol ‘Mulai Kuis’ dengan hanya mengisi Kode Lembar Soal dan Kata Kunci Kode Lembar Soal yang salah | Kode Lembar Soal: xyzabc Kata Kunci Lembar Soal: 123456 | Sistem Aplikasi akan menampilkan pesan : “Terjadi kesalahan!!! Periksa kembali...” | ✓ | |
| 9. | Menekan tombol ‘Mulai Kuis’ dengan | Kode Lembar Soal: doraemon | Sistem Aplikasi akan mengambil soal dan | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | mengisi Kode Lembar Soal dan Kata Kunci Kode Lembar Soal yang benar | Kata Kunci Lembar Soal: 123456 Timer akan menunjukkan sisa waktu kuis dan menghitung mundur waktu pengerjaan kuis dalam detik. | memeriksa apakah waktu kuis masih diijinkan untuk melakukan kuis atau tidak. Jika masih diijinkan untuk melakukan pengisian kuis, maka mahasiswa tersebut akan disediakan pilihan untuk dapat mengisi pilihan dengan judul pada layar “Selamat Mengerjakan Kuis” dan status kuis berubah menjadi “1”. Jika sudah tidak diijinkan untuk melakukan pengisian kuis, karena waktu kuis telah terlampaui, maka mahasiswa tersebut hanya dapat mereview kuis saja dengan menampilkan pada layar judul “Review Kuis” | | |
| 10. | Menekan tombol ‘Mulai Kuis’ dengan mengisi Kode Lembar Soal dan Kata Kunci Kode Lembar Soal yang benar. Dan menjawab pada pilihan soal yang tersedia | Kode Lembar Soal: doraemon Kata Kunci Lembar Soal: 123456 Memilih tombol a,b,c, atau d | Sistem Aplikasi akan menyimpan jawaban mahasiswa pada soal terpilih dan mengupdate pilihan tersebut pada database di sisi server. Dan akan melingkari tombol jawaban yang telah dipilih. Jika jawaban yang telah di lingkari di tekan kembali, maka pilihan jawaban akan menjadi null. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|--|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 11. | Menekan tombol 'Mulai Kuis' dengan mengisi Kode Lembar Soal dan Kata Kunci Kode Lembar Soal yang benar. Dan melakukan navigasi, maju ke soal berikutnya atau kembali ke soal sebelumnya. | Kode Lembar Soal: doraemon Kata Kunci Lembar Soal: 123456 Memilih tombol "<Sebelumnya" atau "Berikutnya" | Sistem Aplikasi akan menampilkan soal dan pilihan jawaban untuk soal berikutnya jika tombol "Berikutnya>" ditekan dan akan menampilkan soal sebelumnya jika tombol "<Sebelumnya" di tekan. Jika salah satu tombol ditekan dan ditahan agak lama, maka sistem akan membawa mahasiswa ke soal paling awal untuk tombol "<Sebelumnya" atau membawa mahasiswa ke soal paling akhir untuk tombol "Berikutnya>" | ✓ | |
| 12. | Menekan tombol 'Mulai Kuis' dengan mengisi Kode Lembar Soal dan Kata Kunci Kode Lembar Soal yang benar. Dan menekan tombol "Kirim dan Nilai" | Kode Lembar Soal: doraemon Kata Kunci Lembar Soal: 123456 Memilih tombol "Kirim dan Nilai" | Sistem Aplikasi akan mengirimkan jawaban dan merubah status menjadi "2" untuk menandakan bahwa mahasiswa telah mengirimkan jawaban untuk di nilai. | ✓ | |
| 13. | Menekan tombol 'Mulai Kuis' dengan mengisi Kode Lembar Soal dan Kata Kunci Kode Lembar Soal yang benar. Perubahan Status Hasil Kuis | Kode Lembar Soal: doraemon Kata Kunci Lembar Soal: 123456 Timer menunjukkan angka 0 | Sistem Aplikasi akan merubah status kuis menjadi "Review Kuis" setelah timer untuk kuis tersebut memiliki sisa waktu pengerjaan 0 detik. | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--------------------|-----------|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| | | | Jika pada saat pengerjaan kuis, mahasiswa tersebut kehabisan waktu kuis (penghitung mundur=0 detik) maka status hasil kuis akan berubah menjadi “3” jika kuis sudah di jawab sebagian, atau status kuis akan berubah menjadi “4” jika kuis sudah di jawab semua. Dan jika pada waktu pengambilan soal mahasiswa tersebut diluar masa berlaku kuis maka status kuis menjadi “5” | | |

IV.2.1.2.3.3 Testing [3. Review Kuis]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menekan tombol ‘3.Review Kuis’ pada menu utama mahasiswa | | Sistem Aplikasi akan menampilkan formulir pengisian untuk mengisi Kode Lembar Soal yang akan di review | ✓ | |
| 2. | Menekan tombol ‘3.Review Kuis’ pada menu utama mahasiswa Dan menekan tombol ‘Back’ pada tombol android | Menekan tombol ‘Back’ pada tombol android | Sistem Aplikasi kembali ke menu utama mahasiswa | ✓ | |

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|--|---|-----------------|-------|
| | | | | Berhasil | Gagal |
| 3. | Menekan tombol '3.Review Kuis' pada menu utama mahasiswa Dan menekan tombol 'Batal' pada layar | Menekan tombol 'Batal' pada layar | Sistem Aplikasi kembali ke menu utama mahasiswa | ✓ | |
| 4. | Menekan tombol '3.Review Kuis' pada menu utama mahasiswa dan menekan tombol "Kode Lembar Soal" | Menekan tombol 'Kode Lembar Soal' pada layar | Sistem Aplikasi akan menampilkan daftar lembar soal untuk kuis-kuis yang telah diikuti oleh mahasiswa tersebut. | ✓ | |
| 5. | Menekan tombol '3.Review Kuis' pada menu utama mahasiswa dan menekan tombol "Kode Lembar Soal" Dan memilih kode lembar soal yang tersedia | Memilih lembar soal pada daftar pilihan yang tersedia pada layar | Sistem Aplikasi akan menampilkan daftar lembar soal terpilih pada formulir isian memasukkan kode lembar soal. | ✓ | |
| 6. | Menekan tombol '3.Review Kuis' pada menu utama mahasiswa dan memasukkan kode lembar soal yang benar | Kode lembar soal: doraemon | Sistem Aplikasi akan mengambil soal dapat mereview kuis dengan menampilkan pada layar judul "Review Kuis" | ✓ | |
| 7. | Menekan tombol '3.Review Kuis' pada menu utama mahasiswa dan memasukkan kode lembar soal yang salah | Kode lembar soal: dorayaki | Sistem Aplikasi akan menampilkan pesan kesalahan. Kode lembar soal dorayaki tidak ditemukan | ✓ | |

IV.2.1.2.3.4 Testing [4. Review Nilai]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|--|---|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menekan tombol '4.Review Nilai' pada menu utama mahasiswa | | Sistem Aplikasi akan menampilkan formulir Daftar Nilai Kuis dengan menampilkan No urut, Kode Lembar Soal dan Nilai yang di raih oleh mahasiswa yang sedang <i>online</i> | ✓ | |
| 2. | Menekan tombol '4.Review Nilai' pada menu utama mahasiswa Dan menekan tombol 'Kembali ke Menu' pada layar | Menekan tombol 'Kembali ke Menu' pada layar | Sistem Aplikasi kembali ke menu utama mahasiswa | ✓ | |
| 3. | Menekan tombol '4.Review Nilai' pada menu utama mahasiswa Dan menekan tombol 'Back' pada tombol android | Menekan tombol 'Back' pada tombol android | Sistem Aplikasi kembali ke menu utama mahasiswa | ✓ | |

IV.2.1.2.4 FMOBIK-18 Testing [5. Keluar]

| No. | Skenario pengujian | Test case | Hasil yang diharapkan | Hasil pengujian | |
|-----|---|-----------|--|-----------------|-------|
| | | | | Berhasil | Gagal |
| 1. | Menekan tombol '5.Keluar' pada menu utama mahasiswa | | Sistem Aplikasi akan kembali ke menu login | ✓ | |

Bab V Kesimpulan dan Saran

V.1 Kesimpulan

1. Aplikasi Mobik yang berbasis *client-server* telah di bangun sesuai dengan perancangan.
2. Aplikasi Mobik dapat berfungsi sebagai aplikasi tambahan pada *smartphone Android* yang dapat digunakan untuk proses belajar-mengajar.
3. Aplikasi Mobik dapat melakukan autotentifikasi pada pengguna.
4. Aplikasi Mobik dapat menyimpan kumpulan soal dan mengirimkan soal-soal yang telah dipilih untuk pengguna yang telah ditentukan.
5. Pengguna Mobik di sisi *client* dapat menerima soal dari *server*, dan mengirimkan jawaban ke *server*.
6. Pengguna Mobik dapat melakukan *review* ulang untuk setiap soal dan jawaban yang telah dikirimkan ke *server*.
7. Pengguna Mobik di sisi *client* dapat melihat nilai untuk setiap lembar soal yang telah dikerjakan.
8. Aplikasi Server dapat mencetak lembar jawaban untuk setiap mahasiswa yang mengikuti kuis.
9. Aplikasi Server dapat menerima masukan soal dengan menggunakan fasilitas import yang telah disediakan.

V.2 Saran

Berdasarkan kesimpulan yang telah diperoleh dalam tugas akhir ini, diajukan beberapa saran yang perlu dipertimbangkan sebagai berikut:

1. Untuk menunjang aplikasi ini, maka dari pihak kampus menyediakan beberapa perangkat android yang sudah terinstall aplikasi Mobik.
2. Untuk memperkuat sinyal serta memisahkan koneksi *wifi* dari pemakaian secara umum, agar bisa dipakai lebih dari 20 pengguna android secara bersamaan dengan kualitas kecepatan yang terjaga.

DAFTAR PUSTAKA

1. Betts, M., *The Almanac: Hot Tech*, ComputerWorld 52, 17 November 2003
2. Berkhin, P., *Survey of Clustering Data Mining Techniques*, Accrue Software, 1045 Forest Knoll Dr., San Jose, 2002
3. <http://www.stanford.edu/class/ee368/Android/Tutorial-3-Server-Client-Communication-for-Android.pdf>, di akses pada tanggal 26 Desember 2012.
4. Fowler, M., Scott, K.; 1999; *UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language*; Addison Wesley.
5. <http://www.agilemodeling.com>, di akses pada tanggal 26 Desember 2012.
6. Sudjana, Nana. *Penilaian Hasil Proses Belajar Mengajar*, cetakan ketiga. Bandung: PT Remaja RosdaKarya. 1991.
7. Android Developer Guide: <http://developer.android.com/about/index.html>, di akses pada tanggal 26 Desember 2012.
8. Rosenberg, Doug and Stephens, Matt. *Use Case Driven Object Modeling with UML: Theory and Practice*. APress. 2007.
9. Safaat, Nazruddin. *ANDROID – Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika. 2011.
10. Lia, K. *Mahir dan terampil berkomputer*. Jakarta: Grafindo Media Pratama, 2006.
11. <http://web.mit.edu/newsoffice/2011/center-mobile-learning.html>, di akses pada tanggal 26 Desember 2012.
12. <http://appinventor.mit.edu/explore/content/what-app-inventor.html>, di akses pada tanggal 26 Desember 2012.
13. <http://jQuery.com>, diakses pada tanggal 10 Januari 2013.
14. <http://www.datatables.net>, di akses pada tanggal 21 Januari 2013.
15. www.mpdf1.com/mpdf, di akses pada tanggal 23 Januari 2013.