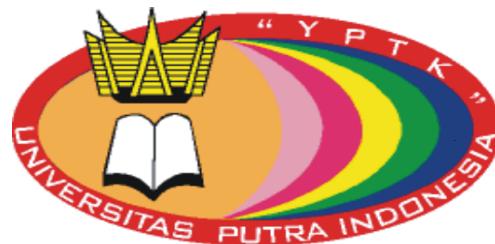


**PERANCANGAN COMPUTER ASSISTED TEST (CAT)
MENGGUNAKAN BAHASA PEMOGRAMAN NODE.JS DAN
DATABASE POSTGRESQL BERBASIS WEB/MOBILE PADA
UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG**

SKRIPSI

*Untuk Memenuhi Sebagian Persyaratan
Mencapai Gelar Sarjana Komputer
Sistem Informasi Strata I*



**Dosen Pembimbing I :
Siflia Andini ,M.Kom**

**Diajukan Oleh:
EGO DAFMA DASA
14101152610565**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG
PADANG
2017**

Daftar Isi

Daftar Isi.....	1
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Hipotesa.....	3
1.4 Batasan Masalah.....	3
1.5 Tujuan Penelitian.....	4
1.6 Manfaat Penelitian.....	4
1.7 Metodologi Penelitian.....	5
1.7.1 Penelitian Lapangan.....	5
1.7.2 Penelitian Perpustakaan.....	5
1.7.3 Penelitian Laboratorium.....	5
1.8 Tinjauan Umum.....	6
1.8.1 Sejarah Berdirinya Universitas Putra Indonesia “YPTK” Padang.....	6
1.8.2 Struktur Organisasi.....	7
1.8.3 Pembagian Tugas Dan Fungsi.....	9
BAB II.....	18
LANDASAN TEORI.....	18
2.1 Konsep Dasar Sistem.....	18
2.1.1 Pengertian Sistem.....	18
2.1.2 Karakteristik Sistem.....	18
2.1.3 Klasifikasi Sistem.....	20
2.1.4 Daur Hidup Sistem.....	22
2.2 Konsep Informasi.....	24
2.2.1 Pengertian Data dan Informasi.....	24
2.2.2 Atribut Informasi.....	25
2.2.3 Siklus Data dan Informasi.....	27
2.3 Konsep Dasar Sistem Informasi.....	28
2.3.1 Pengertian Sistem Informasi.....	28
2.3.2 Komponen Sistem Informasi.....	29
2.3.3 Perencanaan Sistem Informasi.....	32
2.4 Computer Assisted Test (CAT).....	32
2.5 Konsep Basis Data (Database).....	33
2.5.1 Sistem Basis Data.....	34
2.5.2 Hierarki Data.....	34
2.5.3 Tujuan Pengembangan Basis Data.....	36
2.5.4 Keuntungan Pengembangan Basis Data.....	36
2.5.5 PostgreSQL.....	37
2.6 Bahasa Pemrograman Javascript.....	38
2.6.1 NodeJS.....	39
2.7 Unified Modelling Language (UML).....	39

2.7.1 <i>Use Case Diagram</i>	40
2.7.2 <i>Activity Diagram</i>	42
2.7.3 <i>Sequence Diagram</i>	43
2.7.4 <i>Class Diagram</i>	44
BAB III.....	47
ANALISA PERANCANGAN SISTEM.....	47
3.1 Analisa Data.....	47
3.2 Analisa Proses.....	48
3.3 Perancangan UML.....	48
3.3.1 <i>Use Case Diagram</i>	48
3.3.1.1 Defenisi Aktor.....	48
3.3.1.2 <i>Use Case</i>	49
3.3.1.3 Skenario <i>Use Case</i>	50
3.3.2 <i>Class Diagram</i>	66
3.3.3 <i>Sequence Diagram</i>	66
3.3.4 <i>Activity Diagram</i>	66
3.4 Desain Terinci.....	66
3.4.1 Desain Ouput.....	66
3.4.2 Desain Input.....	66
3.4.3 Desain <i>File</i>	66

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penerapan sistem komputerisasi didalam kehidupan sehari-hari disaat sekarang sangat banyak sekali. Dimulai dari pengolahan data dan informasi, penjualan berbasis internet dll.

Selain itu dalam bidang pendidikan juga terdapat penerapan dari sistem komputerisasi seperti perpustakaan digital dan pembelajaran digital dimana komputer atau perangkat smartphone digunakan sebagai media pengakses informasi pembelajaran.

Tak hanya itu saja, sekarang sudah berkembang aplikasi komputer yang bernama “Computer Assisted Test” dimana kegiatan tes atau ujian atau semacamnya dilakukan berbasis komputer. Hal tersebut memungkinkan peserta melakukan pengisian jawaban dan pembacaan soal serta penggerjaannya dilakukan pada komputer.

Setiap tahun ajaran baru, terdapat ribuan calon mahasiswa yang mendaftar untuk menjadi mahasiswa Universitas Putra Indonesia “YPTK” Padang. Namun sebelum menjadi mahasiswa, calon mahasiswa terlebih dahulu harus mengikuti semacam tes dengan mata pelajaran seperti pada ujian UN. Jika aplikasi *Computer Assisted Test* diterapkan pada tes penerimaan mahasiswa baru atau diterapkan pada ujian lain seperti ujian mid semester dan akhir semester, maka pihak kampus bisa menghemat banyak biaya karena penggunaan kertas yang

berkurang serta ketepatan komputer dalam mengolah data menjadikan informasi seperti hasil ujian dari peserta dapat dilihat langsung setelah ujian berakhir. Dengan demikian hal tersebut akan menghemat waktu dosen dalam memeriksa ujian mahasiswa.

Berdasarkan uraian diatas, maka penulis tertarik untuk menerapkan aplikasi *Computer Assisted Test* dalam bentuk skripsi dengan judul **“PERANCANGAN COMPUTER ASSISTED TEST (CAT) MENGGUNAKAN BAHASA PEMOGRAMAN NODE.JS BERBASIS WEB/MOBILE PADA UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG”**

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah yang diungkapkan diatas, penulis mengemukakan beberapa masalah yang ditemukan pada Universitas Putra Indonesia “YPTK” Padang sebagai berikut :

1. Apakah ujian yang berlangsung pada Universitas Putra Indonesia “YPTK” Padang masih menerapkan sistem tradisional dimana jawaban peserta ditulis atau ditandai pada kertas?
2. Apakah ujian masih bersifat manual meskipun hasil ujian dari peserta ujian sudah disimpan pada Sistem Informasi Universitas Putra Indonesia “YPTK” Padang?

3. Bagaimanakah kesalahan dalam pemeriksaan hasil ujian atau tes secara manual masih menemukan kesalahan walau dalam angka yang sangat kecil?

1.3 Hipotesa

Berdasarkan perumusan masalah diatas, diperoleh hipotesa yaitu:

1. Dengan merancang dan diterapkannya *Computer Assisted Test*, maka kesalahan peserta dalam melaksanakan ujian atau kesalahan teknis lainnya bisa diminimalkan.
2. Dengan adanya *Computer Assisted Test*, hasil tes atau ujian peserta dapat langsung dilihat hasilnya sehingga dapat mempercepat pengecekan hasil ujian.
3. Dengan menggunakan kemampuan bahasa pemrograman NodeJS yaitu *non blocking*, diharapkan lebih handal dalam menangani *computer assisted test* ketimbang bahasa pemrograman server side lainnya. Dan didukung dengan database *postgresql* yang terbilang lebih stabil ketimbang *mysql*.

1.4 Batasan Masalah

Agar pembahasan lebih terarah, maka perlu diberikan batasan-batasan terhadap topik diatas. Dalam pembahasan ini, lebih berfokus kepada perancangan aplikasi *Computer Assisted Test* menggunakan *NodeJS* dan database *PostgreSQL*

yang nantinya bisa diterapkan pada jenis tes atau ujian tertentu yang biasanya diadakan oleh pihak Universitas Putra Indonesia “YPTK” Padang.

1.5 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah :

1. Melihat apakah *Computer Assisted Test* mampu mempermudah mahasiswa dan peserta ujian dalam melaksakan tes.
2. Melihat sejauh mana *Computer Assisted Test* dapat diterapkan pada Universitas Putra Indonesia “YPTK” Padang.
3. Merancang sebuah aplikasi *Computer Assisted Test* yang dapat diterapkan pada Universitas Putra Indonesia “YPTK” Padang.

1.6 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini, yaitu :

1. Bagi pihak Universitas Putra Indonesia “YPTK” Padang
Diharapkan dengan dilakukannya penelitian ini, pihak Universitas Putra Indonesia “YPTK” Padang dapat dengan lebih mudah dalam melangsungkan sebuah tes atau ujian dan juga lebih hemat waktu dan biaya.
2. Bagi pengembangan Ilmu pengetahuan
Dapat memberikan sumbangan yang besar terhadap ilmu pengetahuan khususnya ilmu teknologi dan informasi serta sebagai perbandingan

penelitian yang satu dengan yang lainnya sehingga dapat memilih mana yang seharusnya diterapkan.

1.7 Metodologi Penelitian

Pada penelitian ini, Penulis berusaha mencari data-data yang sesuai dengan topik permasalahan. Untuk itu dilakukan beberapa metode antara lain :

1.7.1 Penelitian Lapangan

Untuk mengetahui kekurangan yang ada pada pelaksanaan tes atau ujian pada Universitas Putra Indonesia “YPTK” Padang dengan cara pengamatan langsung dan wawancara terhadap pelaksana serta peserta tes.

1.7.2 Penelitian Perpustakaan

Disain penelitian akan memakai disain analisis yaitu dengan mengumpulkan data dan teori yang mendukung penelitian dengan mempelajari literatur-literatur yang akan digunakan sebagai masukan sekaligus sebagai pemecahan masalah yang ada.

1.7.3 Penelitian Laboratorium

Pada penelitian ini digunakan sebuah Laptop yang mana hardware dan software yang terpasang pada Laptop tersebut adalah dengan spesifikasi sebagai berikut:

1. Hardware
 - a. Processor AMD A8-7410
 - b. Memori RAM 4 Gb
 - c. Media penyimpanan hardisk 1 Tb
 - d. Printer canon MP287
2. Software
 - a. Sistem Operasi Ubuntu 16.04 32bit
 - b. NodeJS versi 6.11.3
 - c. Node Package Manager (NPM) versi 5.4.2
 - d. Text editor Geany versi 1.29
 - e. Postgresql versi 9.5
 - f. Software pendukung lainnya

1.8 Tinjauan Umum

1.8.1 Sejarah Berdirinya Universitas Putra Indonesia “YPTK” Padang

Universitas ini didirikan oleh H. Herman Nawas beserta istrinya Dr. Hj. Zerni Melmusni, M.M, Ak. Melalui yayasananya yaitu Yayasan Pendidikan Tinggi Komputer Padang "(YPTK Padang)" pada tahun 1985. Pada awal berdirinya, universitas ini masih berbentuk sekolah tinggi yang terdiri dari STMIK, AMIK, STIE dan AAMPK.

1.8.2 Struktur Organisasi

Merupakan suatu cara bagi organisasi untuk mencapai tujuan dengan tenaga kerja orang lain. Struktur organisasi menggambarkan kekuasaan dan tanggung jawab yang digambarkan dengan garis vertikal dan horizontal. Garis vertikal menggambarkan hubungan antara atasan dan bawahan serta garis horizontal menggambarkan hubungan antar bagian.

Struktur organisasi disetiap perusahaan berbeda-beda tergantung besar kecilnya sebuah perusahaan.

Adapun bentuk struktur organisasi yang umum dikenal adalah :

1. Organisasi Garis

Dalam organisasi ini tugas perencanaan, pengorganisasian, dan pengawasan berada pada suatu tangan serta garis wewenang langsung dari pimpinan kepada bawahan dan bawahan hanya mengenal satu pimpinan sebagai sumber kewenangan yang memberikan perintah atau instruksi.

2. Organisasi Garis dan Staff

Pada organisasi ini terdapat unit secara langsung ikut serta melaksanakan tercapainya tugas pokok atau tujuan organisasi. Staff hanya memberikan kontribusinya dalam hal tidak langsung tapi dengan menyediakan bantuan di bidang kepegawaian, keuangan, material dan bantuan lainnya.

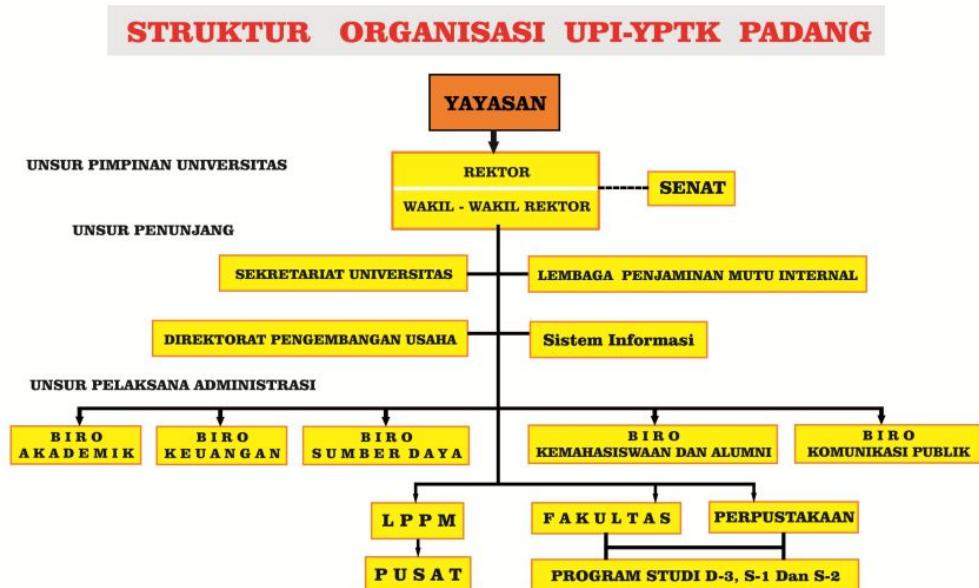
3. Organisasi Fungsi

Organisasi ini umumnya terdapat pada organisasi niaga yang disusun atas dasar kegiatan tiap-tiap fungsi sesuai dengan kepentingan perusahaan.

4. Organisasi Panitia

Panitia adalah sekelompok orang yang ditunjuk untuk melaksanakan kegiatan khusus yang tidak dapat diselesaikan oleh seseorang atau banyak orang. Pada umumnya organisasi ini dibentuk dalam jangka waktu terbatas dan setelah itu organisasi dibubarkan

Adapun struktur organisasi dari Universitas Putra Indonesia "YPTK" Padang adalah :



Gambar 1.1 Struktur Organisasi Universitas Putra Indonesia "YPTK" Padang

(Sumber : Website Resmi Universitas UPI YPTK)

1.8.3 Pembagian Tugas Dan Fungsi

Berdasarkan struktur organisasi tersebut, maka berikut adalah masing-masing komponen dari struktur tersebut :

1. Rektor

- a. Memimpin Universitas Putra Indonesia “YPTK” Padang dan membina civitas akademika agar berdaya guna dan berhasil guna.
- b. Menentukan kebijaksanaan pelaksanaan penyelenggaraan pendidikan dan pengajaran, penelitian dan pengabdian pada masyarakat di lingkungan Universitas Putra Indonesia “YPTK” yang secara fungsional menjadi tanggung jawabnya sesuai dengan kebijaksanaan umum pemerintahan, Menteri Pendidikan dan Kebudayaan serta kebijaksanaan teknis Direktur Jenderal Pendidikan Tinggi dan Yayasan Universitas Putra Indonesia “YPTK”.
- c. Membina dan melaksanakan kerjasama dengan instansi, badan swasta dan masyarakat untuk memecahkan persoalan yang timbul terutama yang menyangkut tanggung jawabnya.

2. Wakil Rektor I (Bidang Akademik dan Kerjasama)

Bertugas membantu rektor dalam memimpin pelaksanaan pendidikan dan pengajaran, penelitian, pengabdian kepada masyarakat serta program kerjasama dengan instansi pemerintah, swasta, dan masyarakat yang meliputi perencanaan pelaksanaan dan pengembangan pendidikan

dan pengajaran, pembinaan tenaga pengajar dan tenaga peneliti, persiapan program pendidikan baru pada berbagai tingkat maupun bidang, penyusunan program bagi usaha pengembangan daya penalaran mahasiswa, perencanaan dan pelaksanaan kerjasama pendidikan dan pengajaran, serta penelitian dengan lembaga di dalam maupun di luar negeri, pengolahan data yang menyangkut pendidikan dan pengajaran, penelitian dan pengabdian pada masyarakat serta pelaksanaan kegiatan di bidang pengabdian pada masyarakat dalam rangka turut membantu memecahkan masalah yang dihadapi oleh masyarakat dalam pembangunan.

3. Wakil Rektor II (Bidang Administrasi Umum)

Bertugas membantu rektor dalam memimpin pelaksanaan kegiatan bidang keuangan, administrasi umum dan manajemen sumber daya yang meliputi perencanaan dan pengelolaan anggaran, pembinaan kepegawaian dan kesejahteraan, pengelolaan perlengkapan, pengurusan kerumahtanggaan dan pemeliharaan ketertiban, pengurusan ketatausahaan, penyelenggaraan hubungan masyarakat serta pengolahan data yang menyangkut bidang administrasi umum.

4. Wakil Rektor III (Bidang Kemahasiswaan)

Bertugas membantu rektor dalam memimpin pelaksanaan kegiatan bidang pembinaan, pengembangan, dan pelayanan mahasiswa yang meliputi pelaksanaan pembinaan mahasiswa oleh staf pengajar dalam

pengembangan sikap dan orientasi serta kegiatan mahasiswa antara lain dalam seni budaya dan olah raga sebagai bagian dari pembinaan civitas akademika yang merupakan tugas pendidikan tinggi pada umumnya, pelaksanaan usaha kesejahteraan serta usaha bimbingan dan penyuluhan bagi mahasiswa, pelaksanaan usaha pengembangan daya penalaran mahasiswa yang diprogram oleh Wakil Rektor I, bekerjasama dengan pihak lain dalam setiap usaha kemahasiswaan, pengabdian pada masyarakat dan usaha penunjangnya, menciptakan iklim pendidikan yang baik dalam kampus dan membantu pelaksanaan pemeliharaan kesatuan bangsa berdasarkan Pancasila dan Undang-Undang Dasar 1945 serta pelaksanaan kegiatan kemahasiswaan dalam rangka usaha pembangunan yang tetap dilandasi dan nilai-nilai dan tanggung jawab yang bersifat akademik.

5. Biro Administrasi Akademik dan Kemahasiswaan

Biro Administrasi Akademik dan kemahasiswaan adalah unsur Wakil pimpinan di bidang administrasi akademik dan kemahasiswaan yang berada dan bertanggung jawab langsung kepada rektor. Biro Administrasi Akademik dan Kemahasiswaan bertugas memberikan pelayanan teknis dan administrasi di bidang akademik dan kemahasiswaan di lingkungan Universitas Putra Indonesia “YPTK” Padang meliputi :

- a. Menyusun rencana dan program kerja biro sebagai pedoman pelaksanaan tugas
- b. Menelaah peraturan perundang-undangan di bidang akademik, kemahasiswaan, perencanaan dan sistem informasi
- c. Merumuskan saran alternatif di bidang administrasi akademik, kemahasiswaan, perencanaan, dan sistem informasi sebagai bahan penyusunan kebijakan
- d. Menyusun kalender akademik sebagai pedoman kegiatan akademik
- e. Menyusun naskah kerjasama dengan instansi pemerintah dan swasta sebagai bahan pengembangan Tri Dharma Perguruan Tinggi
- f. Melaksanakan penawaran dan seleksi calon penerima beasiswa berdasarkan ketentuan yang berlaku
- g. Melaksanakan pembinaan dan pengembangan administrasi akademik, perencanaan dan sistem informasi
- h. Memantau dan mengevaluasi pelaksanaan kegiatan administrasi akademik, kemahasiswaan, perencanaan, dan sistem informasi untuk mengetahui perkembangannya dan sebagai bahan untuk penyusunan kebijakan.
- i. Melaksanakan kegiatan administrasi penerimaan mahasiswa baru
- j. Melakukan koordinasi, validasi, penggabungan dan pelaporan Evaluasi Program Studi Berdasarkan Evaluasi Diri (EPSBED)

k. Menyusun laporan biro sesuai dengan hasil yang dicapai sebagai pertanggungjawaban pelaksanaan tugas

l. Melaksanakan tugas lain yang diberikan atasan

6. Biro Administrasi Umum dan Keuangan (BAUK)

Biro Administrasi Umum dan Keuangan adalah unsur Wakil pimpinan di bidang administrasi umum, keuangan, dan sumberdaya yang berada di bawah dan bertanggung jawab langsung kepada rektor. Biro Administrasi Umum dan Keuangan mempunyai tugas melaksanakan penyusunan rencana dan program manajemen kepegawaian, manajemen keuangan dan manajemen sumberdaya fisik dan perlengkapan, kerumahtanggaan, dan ketatausahaan di lingkungan Universitas UPI “YPTK” Padang. Dalam melaksanakan tugas tersebut Biro Administrasi Umum dan Keuangan menyelenggarakan fungsi :

a. Penyusunan rencana dan fungsi organisasi

b. Pelaksanaan administrasi dan manajemen pegawai dan dosen

c. Pelaksanaan manajemen keuangan dan akuntansi

d. Perlaksanaan administrasi dan sumberdaya fisik

e. Pelaksanaan kerumahtanggaan

f. Pelaksanaan administrasi ketatausahaan

Biro Administrasi Umum dan Keuangan dipimpin oleh seorang kepala yang sekaligus bertugas sebagai pejabat penanggung jawab keuangan

universitas sebagaimana tertera dalam fungsi BAUK dan sesuai dengan PP nomor 23 Tahun 2005.

7. Fakultas

Fakultas terdiri dari unsur-unsur seperti yang diatur pada Peraturan Pemerintah No.60 tahun 1999 tentang Pendidikan Tinggi sebagai berikut :

a. Dekan

Dekan adalah pimpinan fakultas yang bertugas memimpin penyelenggaraan pendidikan, penelitian, dan pengabdian pada masyarakat, membina tenaga kependidikan, mahasiswa, tenaga administrasi dan administrasi fakultas. Dekan bertanggung jawab kepada rektor.

b. Wakil Dekan

Wakil dekan bertugas membantu pelaksanaan tugas sehari-hari dekan. Wakil dekan terdiri atas Wakil Dekan Bidang Akademik, Wakil Dekan Bidang Administrasi Umum, dan Wakil Dekan Bidang Kemahasiswaan. Wakil Dekan bertanggung jawab kepada Dekan.

c. Senat Fakultas

Senat Fakultas merupakan badan normatif dan perwakilan tertinggi di lingkungan fakultas yang memiliki wewenang untuk menjabarkan kebijakan dan peraturan universitas untuk fakultas yang bersangkutan.

d. Jurusan

Jurusan merupakan unit pelaksana akademik yang melaksanakan pendidikan akademik. Jurusan terdiri dari unsur pimpinan (Ketua Jurusan dan Sekretaris Jurusan), unsur pelaksana akademik (para dosen), serta laboratorium/studio. Ketua Jurusan bertanggung jawab kepada dekan fakultas yang membawahinya.

e. Kelompok Dosen

Kelompok dosen adalah tenaga pengajar di lingkungan fakultas yang berada di bawah dan bertanggung jawab pada atasan masing-masing di lingkungan fakultas yang bersangkutan. Kelompok pengajar terdiri dari tenaga pengajar tetap dan tenaga pengajar luar biasa, serta dosen PNS yang diperbantukan.

f. Bagian Tata Usaha Fakultas

Bagian tata usaha fakultas adalah unit pelaksana teknis dan administratif di lingkungan fakultas yang berada di bawah dekan. Bagian tata usaha fakultas dipimpin oleh seorang kepala bagian yang bertanggung jawab langsung pada dekan.

8. Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM)

Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) adalah pelaksana pembinaan dan pengembangan ilmu pengetahuan, teknologi dan seni, serta pengabdian pada masyarakat dengan berpedoman pada ketentuan yang ditetapkan oleh pimpinan Universitas Putra Indonesia

“YPTK” Padang. Dalam melaksanakan tugas tersebut Lembaga Penelitian dan Pengabdian kepada Masyarakat menyelenggarakan fungsi :

- a. Perumusan kebijakan di bidang penelitian dan pengabdian masyarakat
- b. Pelaksanaan penelitian lingkup ilmu pengetahuan, teknologi, dan/atau seni yang dimaksudkan untuk menunjang pengembangan konsepsi pembangunan nasional, wilayah, dan/atau daerah
- c. Pelaksanaan penelitian untuk pengembangan sistem pendidikan dan institusi Universitas Putra Indonesia “YPTK” Padang
- d. Pelaksanaan tugas administrasi lembaga penelitian
- e. Pelaksanaan pengembangan pola dan konsepsi pembangunan nasional, wilayah dan/atau daerah berdasar nilai-nilai lokal, ilmu pengetahuan, teknologi, dan/atau seni
- f. Pengamalan hasil-hasil penelitian lingkup ilmu pengetahuan, teknologi dan/atau seni
- g. Peningkatan relevansi program pengabdian sesuai kebutuhan masyarakat
- h. Pelaksanaan pemberian bantuan keahlian kepada masyarakat dalam melaksanakan pembangunan

9. Perpustakaan

Perpustakaan adalah unit pelaksana teknis di bidang perpustakaan di lingkungan Universitas Putra Indonesia “YPTK” Padang yang berada di bawah dan bertanggung jawab langsung kepada rektor dan pembinaannya dilakukan oleh Wakil Rektor Bidang Akademik dan Kerjasama. Perpustakaan dipimpin oleh kepala yang ditunjuk di antara pustakawan atau dosen. Perpustakaan mempunyai tugas melaksanakan pelayanan perpustakaan untuk keperluan pendidikan, penelitian, dan pengabdian kepada masyarakat. Dalam melaksanakan tugasnya, Perpustakaan Universitas Putra Indoensia “YPTK” Padang melaksanakan fungsi :

- a. Penyusunan dan perumusan konsep kebijakan dan perencanaan program perpustakaan
- b. Pengembangan kepustakaan dan pustakawan
- c. Pengadaan, pelayanan, dan pemeliharaan bahan pustaka; dan
- d. Pelaksanaan kerjasama antar perpustakaan perguruan tinggi dan/atau badan lain di dalam/luar negeri

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Istilah “sistem” masih populer. Terminologi ini digunakan untuk mendeskripsikan banyak hal. Usaha yang dilakukan pada masa lampau dalam pemrosesan data terfokus pada pengembangan mesin yang menjalankan operasi secara efisien. Kemudian, penemuan *punched cord* menegaskan bahwa pengkonversian data menjadi informasi adalah suatu proses. Selain itu, pengembangan komputer digital berikut teknologi yang menyertainya juga meningkatkan kepopuleran penggunaan terminologi “sistem” (Tata Sutabri 2016: 9).

2.1.1 Pengertian Sistem

Sistem adalah sekumpulan komponen yang saling berhubungan satu sama lainnya dan bekerja sama untuk mencapai suatu tujuan (Eddy Prahasta 2014: 78).

2.1.2 Karakteristik Sistem

Konsep sebuah sistem yang sangat sederhana terdiri atas input, proses, dan output, sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran (Tata Sutabri 2016: 10).

Selain itu, sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik sistem adalah sebagai berikut (Tata Sutabri 2016: 10) :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi artinya saling bekerja sama membentuk satu kesatuan.

2. Batasan Sistem

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.

3. Lingkungan Luar Sistem

Bentuk apa pun yang ada di luar ruang lingkup atau batasan sistem yang memengaruhi operasi sistem tersebut disebut lingkungan luar sistem.

4. Penghubung Sistem

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (Input)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (maintenance input) dan sinyal (signal input).

6. Keluaran Sistem (Output)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain.

7. Pengolah Sistem (Proses)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (Objective)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan. (Tata Sutabri 2016: 11)

2.1.3 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut. Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandangan, di antaranya (Tata Sutabri 2016: 11):

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, seperti sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin, yang disebut *human machine system*.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilistik.

4. Sistem Terbuka dan Tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak berpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya (Tata Sutabri 2016: 11-14).

2.1.4 Daur Hidup Sistem

Siklus hidup sistem (system life cycle) adalah proses evolusioner yang diikuti dalam menerapkan sistem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang erat mengikuti langkah-langkah pendekatan sistem karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Siklus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem. Berikut adalah daur hidup sistem :

1. Mengenali Adanya Kebutuhan

Sebelum segala sesuatunya terjadi, timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan dari organisasi dan volume yang meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektivitasnya.

2. Pembangunan Sistem

Suatu proses atau seperangkat prosedur yang harus diikuti untuk menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk dapat memenuhi kebutuhan tersebut.

3. Pemasangan Sistem

Setelah tahap pembangunan sistem selesai. Sistem kemudian akan dioperasikan. Pemasangan sistem merupakan tahap yang penting pula dalam daur hidup sistem. Peralihan dari tahap pembangunan menuju tahap operasional terjadi pemasangan sistem yang sebenarnya, yang merupakan langkah akhir dari suatu pembangunan sistem.

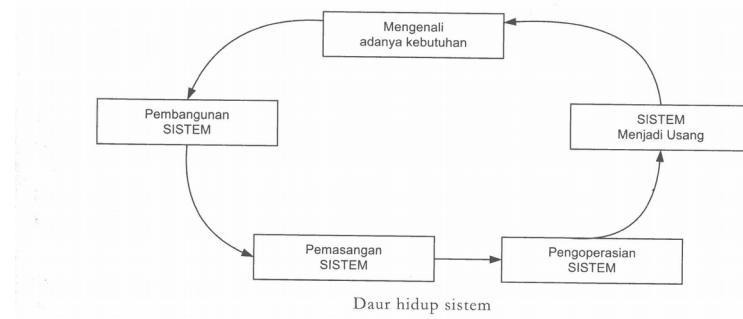
4. Pengoperasian Sistem

Program-program komputer dan prosedur—prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi ditunjang oleh sistem informasi tadi. Ia selalu mengalami perubahan-perubahan itu karena pertumbuhan kegiatan bisnis, perubahan peraturan, dan kebijaksanaan ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbarui.

5. Sistem Menjadi Usang

Kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan-perbaikan pada sistem yang berjalan. Tiba-tiba saatnya secara ekonomis dan teknis sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya (Tata Sutabri 2016: 14-15).

Berikut adalah gambar dari daur hidup sistem :



Gambar 2.1: Daur Hidup Sistem

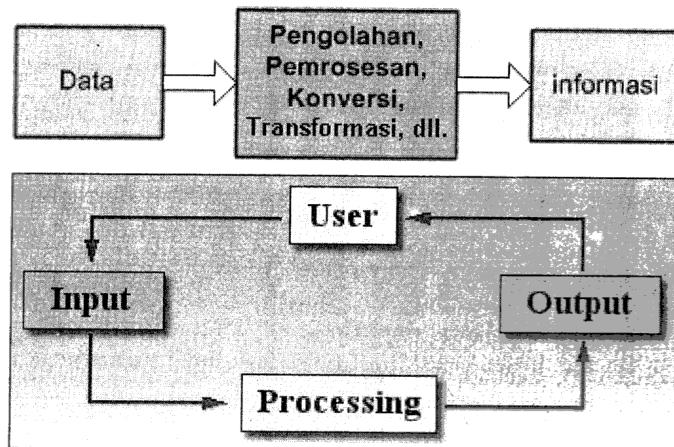
(Sumber Buku *Sistem Informasi Manajemen* Karangan Tata Sutabri 2016)

2.2 Konsep Informasi

Analisis mengenai sistem informasi pada umumnya dimulai dengan deskripsi mengenai “data” dan “informasi. Informasi berasal dari data maka dari itu keduanya tidak dapat dipisahkan (Eddy Prahasta 2014: 69).

2.2.1 Pengertian Data dan Informasi

Data merupakan bahasa, *mathematical*, dan atau simbol pengganti lain yang disepakati secara umum dalam menggambarkan suatu objek, manusia, peristiwa, aktivitas, konsep, atau objek penting lainnya. Singkatnya, data merupakan suatu kenyataan apa adanya. Sedangkan informasi adalah data yang telah ditempatkan pada konteks yang penuh arti oleh penerimanya (Eddy Prahasta 2010: 69-70).

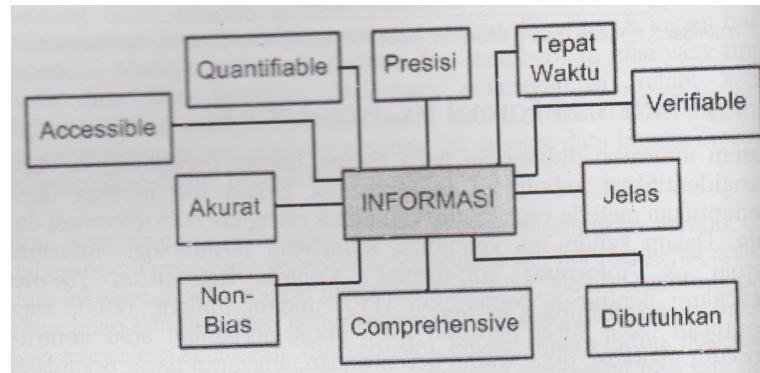


Gambar 2.2: Hubungan Antara Data dan Informasi

(Sumber Buku *Sistem Informasi Geografis* Karangan Eddy Prahasta 2014)

2.2.2 Atribut Informasi

Banyak atribut atau kualitas yang berkaitan dengan konsep informasi dapat membantu perancang dalam mengidentifikasi kebutuhan informasi. Sebagai contoh, gambar berikut memberikan ilustrasi mengenai beberapa atribut informasi (Eddy Prahasta 2014: 72).



Gambar 2.3: Atribut-Atribut Informasi

(Sumber : Buku Sistem Informasi Geografis Karangan Eddy Prahasta 2014 hal 72)

Keterangan Gambar :

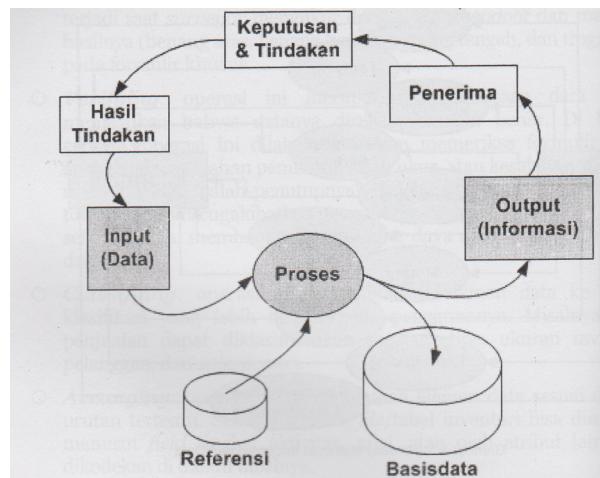
- Akurat: tingkat kebebasan informasi dari kesalahan.
- Quantifiable: tingkat kemampuan menyatakan informasi dalam bentuk numerik.
- Presisi: ukuran detil untuk menyediakan infomasi.
- Tepat waktu: penerimaan informasi masih dalam toleransi waktu yang dibutuhkan oleh penerima; tidak kadaluarsa.
- Jelas: tingkat kebebasan informasi dari keraguan.
- Dibutuhkan: tingkat relevansi informasi dengan kebutuhan pengguna.
- Verifiable: tingkat kesamaan nilai sebagai hasil pengujian informasi yang sama oleh berbagai pengguna.
- Accessible: tingkat kemudahan dan kecepatan memperoleh informasi.

- Non-bias: tingkat perubahan untuk memodifikasi informasi dengan tujuan mempengaruhi para penerimanya.
- Comprehensive: tingkat kelengkapan informasi.

Pada saat mengidentifikasi kebutuhan informasi, sedapat mungkin kebutuhan tersebut dideskripsikan dalam terminologi atribut informasi. Analisis yang benar akan menunjukkan keterkaitan erat antara kebutuhan informasi beserta attributnya pada perancangan sistem informasi. Jadi, yang diperlukan adalah penyediaan informasi yang benar pada waktu yang tepat (Eddy Prahasta 2014: 73).

2.2.3 Siklus Data dan Informasi

Aliran fakta simbol, string alfabetis/kumpulan angka rekaman kejadian disebut sebagai “data”. Pada diagram, aliran ini masuk ke proses. Oleh karena itu, data disebut masukan/ input. Setelah aliran ini diproses (hingga menghasilkan keluaran dalam bentuk yang dimengerti), bentuk ini disebut “informasi”. Oleh karena itu, dipandang dari sisi pemroses data dikenal sebagai input, sementara informasi disebut sebagai output (Eddy Prahasta 2014: 76).



Gambar 2.4: Siklus Data dan Informasi

(Sumber : Buku Sistem Informasi Geografis Karangan Eddy Prahasta 2014)

Di sisi pemrosesan, fenomena input-proses-output bersifat relatif. Suatu proses bisa menerima input hasil olahan (output) proses lainnya. Hal ini nampak pada sekelompok manusia dengan perannya sebagai pemroses. Pihak mana yang melakukan proses pertama, dan pihak lain mana) melakukan proses berikutnya. Tidak jarang, sekelompok pemroses tidak hanya berkemampuan melakukan pemrosesan yang sekarang menjadi tugasnya, mereka juga mampu melakukan pemrosesan yang sebenarnya menjadi tugas kelompok lain pada saat ini. Selain itu, seiring dengan perubahan waktu & proses belajar, suatu kelompok juga bisa melakukan proses lainnya. Oleh karena itu, kemampuan ganda & daya-adaptasi manusia bisa “mengaburkan” pengertian data & informasi (Eddy Prahasta 2014: 76).

2.3 Konsep Dasar Sistem Informasi

Didalam sebuah perusahaan biasanya terdapat seorang manajer. Seorang manajer yang dulunya mengambil kuliah sistem informasi manajemen sudah pasti bisa menjadi seorang manajer praktisi. Jika seandainya manajer tersebut tidak mengambil kuliah sistem informasi manajemen dan tidak mampu menggunakan komputer untuk kebutuhan perusahaannya, maka manajer tersebut menjadi tidak berguna dan tak bermanfaat lagi sebagai pengambil keputusan (Tata Sutabri 2016: 40).

2.3.1 Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Tata Sutabri 2016: 40).

2.3.2 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan yang saling bekerjasama.

1. Blok Masukan

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan.

2. Blok Model

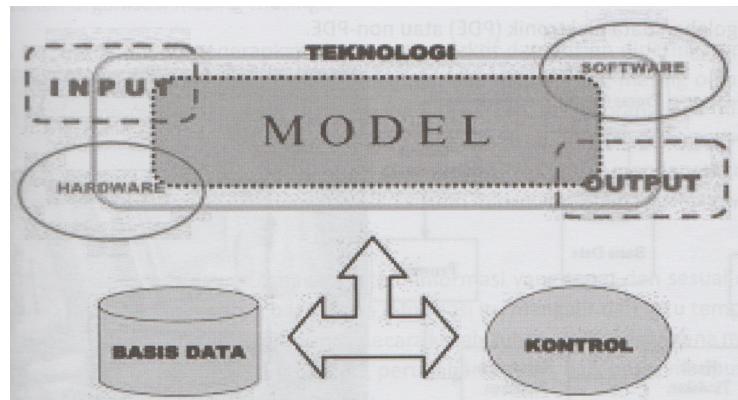
Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

Produk dari sistem informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

Teknologi merupakan *toolbox* dalam sistem informasi. Teknologi didunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian yaitu teknisi, perangkat lunak dan perangkat keras.



Gambar 2.5: Komponen Sistem Informasi

(Sumber : Buku Sistem Informasi Manajemen Karangan Tata Sutabri 2016)

5. Blok Basis Data

Basis data (database) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (database management system)

6. Blok Kendali

Banyak hal yang dapat merusak sistem informasi seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidakefisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah

ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Tata Sutabri 2016: 41-42).

Berdasarkan komponen fisik penyusunnya, komponen sistem informasi terdiri atas :

1. Perangkat Keras

Meliputi perangkat-perangkat yang digunakan oleh sistem komputer untuk masukan dan keluaran, *memory*, *modem*, pengolah dan periferal lainnya.

2. Perangkat Lunak

Berupa program-program komputer yang meliputi sistem operasi dan program-program aplikasi.

3. Berkas Basis Data

Merupakan sekumpulan data dalam basis data yang disimpan dengan cara-cara tertentu sehingga dapat digunakan kembali dengan mudah dan cepat.

4. Prosedur

Meliputi prosedur pengoperasian untuk sistem informasi, manual dan dokumen-dokumen yang memuat aturan-aturan yang berhubungan dengan sistem informasi dan lainnya.

5. Manusia

Manusia yang terlibat dalam suatu sistem informasi meliputi operator, *programmer*, *system analyst*, manajer sistem informasi, manajer pada

tingkat operasional, manajer pada tingkat manajerial, manajer pada tingkat strategis, teknisi, administrator basis data serta individu lain yang terlibat didalamnya (Tata Sutabri 2016: 43).

2.3.3 Perencanaan Sistem Informasi

Pada umumnya, setiap terjadi perencanaan atau perubahan pada sistem, baik besar maupun kecil, selalu akan melalui tingkatan-tingkatan sebagai berikut (Tata Sutabri 2016: 43) :

- Tingkat I : Ide, mengetahui perlu adanya perubahan.
- Tingkat II : Desain, merancang cara pemecahannya.
- Tingkat III : Pelaksanaan, menerapkan desain kedalam sistem.
- Tingkat IV : Kontrol, memeriksa tingkat pelaksanaan dijalankan sesuai dengan desain
- Tingkat V : Evaluasi, memeriksa apakah perubahan yang terjadi sesuai dengan tujuan semula.
- Tingkat VI : Tindak lanjut, melaksanakan perubahan sesuai dengan hasil evaluasi yang ada. (Tata Sutabri 2014: 43)

2.4 Computer Assisted Test (CAT)

Computer Assisted Test didefinisikan sebagai suatu metode ujian dengan menggunakan alat bantu komputer yang digunakan untuk mendapatkan standar minimal kompetensi dasar maupun standar kompetensi kepegawaian (BKN,

2014). Adapun tahapan proses dalam perancangan sistem CAT diawali dengan penelitian dan pengumpulan data, kemudian perencanaan, pembuatan prototipe, pelaksanaan uji coba, dan diikuti perbaikan dan pengembangan. (I Ketut Buana dan Made Gede Wirakusuma 2015: 5)

Adapun tujuan dari penggunaan Computer Assisted Test (CAT) adalah (Khusnul Khotimah 2016: 54) :

1. Mempercepat proses pemeriksaan dan laporan hasil ujian
2. Menciptakan standarisasi hasil ujian secara nasional
3. Menetapkan standar nilai.

Menurut BKN (2014), CAT memiliki prinsip sebagai berikut :

1. Sistem CAT dirancang semudah mungkin agar pengguna dapat dengan mudah memahaminya.
2. Pengoperasian yang sangat mudah. Yaitu dengan menggerakan *mouse* untuk memilih soal, menampilkan soal dan menjawab soal.
3. Soal ditampilkan secara acak pada setiap peserta.
4. Pemeriksaan hasil tes langsung dilakukan oleh aplikasi CAT secara otomatis (I Ketut Buana dan Made Gede Wirakusuma 2015: 5).

2.5 Konsep Basis Data (Database)

Menurut pendapat Eddy Sutanta, istilah basis data dapat dipahami sebagai suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data

(Kalaupun ada maka kerangkapan data tersebut harus seminimal mungkin dan terkontrol), data disimpan dengan cara-cara tertentu sehingga mudah digunakan; data dapat digunakan oleh satu atau lebih program-program aplikasi secara optimal; data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya; data disimpan sedemikian rupa sehingga proses penambahan, pengambilan, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol (2011: 29).

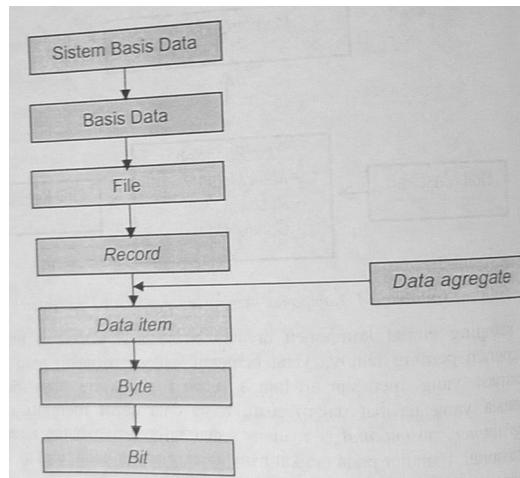
2.5.1 Sistem Basis Data

Menurut Eddy Sutanta, sistem basis data dapat diartikan sebagai sekumpulan basis data dalam suatu sistem yang mungkin tidak ada hubungan satu sama lain, tetapi secara keseluruhan mempunyai hubungan sebagai sebuah sistem dengan didukung oleh komponen lainnya (2011: 32).

2.5.2 Hierarki Data

Berikut hierarki data yang dimulai dari paling kompleks hingga yang paling sederhana :

1. Sistem Basis Data, merupakan sekumpulan basis data dengan para pemakai yang menggunakan basis data tersebut secara bersama-sama.
2. Basis Data, merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar-*record* dan rincian data terhadap obyek tertentu.



Gambar 2.6: Hierarki Data

(Sumber Buku *Basis Data dalam Tinjauan Konseptual* Karangan Eddy Sutanta 2011)

3. File, merupakan sekumpulan *record* sejenis secara relasi yang tersimpan dalam media penyimpanan sekunder.
4. *Record*, merupakan sekumpulan *field/attribut/data item* yang saling berhubungan terhadap obyek tertentu.
5. *Field*, merupakan unit terkecil yang disebut data, yaitu sekumpulan *byte* yang mempunyai makna.
6. *Data Aggregate*, merupakan sekumpulan data *field/item/attribut* dengan ciri khas tertentu dan diberi nama.
7. *Byte*, merupakan bagian terkecil yang dialamatkan pada memori.
8. *Bit*, merupakan sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1 (Eddy Sutanto 2011: 35-37).

2.5.3 Tujuan Pengembangan Basis Data

Berikut adalah tujuan utama dari pengembangan basis data :

1. Data-data dalam basis data dapat digunakan oleh banyak pengguna.
2. Menjaga investasi intelektual.
3. Penekanan biaya.
4. Menghilangkan pengembangan sistem ganda.
5. Kinerja untuk memenuhi kebutuhan informasi.
6. Kejelasan data yang disimpan.
7. Kemudahan pemakaian.
8. Fleksibilitas penggunaan.
9. Memenuhi kebutuhan data yang tidak terantisipasi.
10. Perubahan yang muda.
11. Akurasi dan konsistensi.
12. Privasi.
13. Keamanan.
14. Ketersediaan (Eddy Sutanta 2011: 42-44).

2.5.4 Keuntungan Pengembangan Basis Data

Basis data dikembangkan untuk mengatasi permasalahan-permasalahan pada saat pengolahan data. Basis data yang dikembangkan secara benar akan memberikan keuntungan, diantaranya :

1. Kerangkapan data dapat diminimalkan.
2. Inkonsistensi data yang dapat dihindari.
3. Data dapat digunakan secara bersama.
4. Standarisasi data dapat dilakukan.
5. Pembatasan untuk keamanan data dapat diterapkan.
6. Integritas data dapat dipelihara.
7. Perbedaan kebutuhan data dapat diseimbangkan (Eddy Sutanto 2011: 49).

2.5.5 PostgreSQL

Postgresql adalah *ORDBMS (Object Relational Database Management System)* yang paling maju didunia. *Postgresql* merupakan perangkat lunak *open source*. *Postgresql* dikembangkan oleh *PostgreSQL Global Development Group* yang terdiri dari segelintir relawan yang dipekerjakan dan diawasi oleh perusahaan seperti *Red Hat* dan *Enterprise DB*.

PostgreSQL tersedia untuk hampir semua sistem operasi seperti: *Linux* (semua distribusi terbaru), *Windows*, *UNIX*, *Mac OSX*, *FreeBSD*, *OpenBSD*, *Solaris*, dan lainnya. Sistem seperti *Unix* bekerja pada semua mayoritas arsitektur seperti: *x86*, *x86-64*, *IA64*, *PowerPC*, *Sparc*, *Alpha*, *ARM*, *MIPS*, *PA-RISC*, *VAX*, *M32R*.

MySQL dan *PostgreSQL* keduanya bersaing kuat di bidang *database* relasional karena keduanya memiliki fungsi yang canggih dan juga kinerja dan

kecepatan yang sebanding. Yang terpenting kedua *database* tersebut bersifat *open source* (Sukhdeep Kaur 2016: 285).

Sebagai *ORDBMS* (*Object Relational Database Management System*) yang ada saat ini, *PostgreSQL* memiliki berbagai macam kemampuan yang dimiliki oleh database komersil umum lainnya, seperti dukungan akan perintah-perintah *SQL*, dimana dengan menggunakan perintah-perintah *SQL* memungkinkan database administrator lebih mudah berinteraksi dengan database *PostgreSQL*, baik dalam manipulasi data seperti *insert*, *update*, ataupun *delete*. Selain bersifat *open source*, keunggulan *database PostgreSQL* ini dapat mendukung sebagai media penyimpanan pada banyak bahasa pemrograman yang ada, baik itu pemrograman berbasis *desktop* seperti : *Java*, *Gambas*, dll ataupun pemrograman berbasis web, seperti : *Phyton*, *PHP*, *Java*, *Server Pages*, *Perl*, dll (Mochamad Alfan Rosid 2016: 33).

2.6 Bahasa Pemrograman Javascript

Javascript adalah bahasa yang digunakan untuk membuat program yang digunakan agar dokumen *HTML* yang ditampilkan dalam *browser* menjadi lebih interaktif. *Javascript* memberikan beberapa fungsionalitas kedalam halaman web, sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antarmuka web (Betha Sidik 2011: 1).

2.6.1 NodeJS

NodeJS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Aplikasi ini ditulis dalam bahasa *JavaScript*, menggunakan basis *event* dan *asynchronous I/O*. Tidak seperti kebanyakan bahasa *JavaScript* yang dijalankan pada peramban, *NodeJS* dieksekusi sebagai aplikasi server. Aplikasi ini terdiri dari *V8 JavaScript Engine* buatan *Google* dan beberapa modul bawaan yang terintegrasi (Muhammad Iqbal C. R., Muhammad Husni dan Hudan Studiawan: 2012).

2.7 Unified Modelling Language (UML)

Bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang (Rosa A.S & M. Shalahuddin: 2015).

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural, kemudian juga ada *State*

Transition Diagram (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata) (Rosa A.S & M. Shalahuddin: 2015).

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasi, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa A.S & M. Shalahuddin: 2015).

2.7.1 Use Case Diagram

Menurut Rosa A.S dan M. Shalahuddin (2015), *use case* merupakan pemodelan yang digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakannya.

Ada dua hal utama pada *use case* yaitu pendefinisian apa yang akan disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah tabel simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1: Simbol pada diagram use case

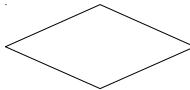
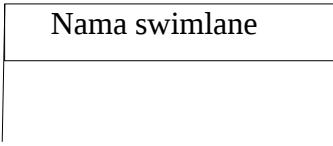
Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawali di awal frase nama <i>use case</i> .
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
Generalisasi 	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan <i>include</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

(Sumber Buku Rekayasa Perangkat Lunak Karangan Rosa A.S dan M. Shalahudin 2015)

2.7.2 Activity Diagram

Diagram aktivitas menggambarkan aliran kerja dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2: Simbol-simbol pada activity diagram

Simbol	Deskripsi
Status awal	 Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
aktivitas	 Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan	 Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
penggabungan	 Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir	 Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane	 Memisahkan organisasi bisnis yang bertanggung jawab terhadap yang terjadi.

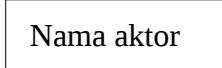
(Sumber Buku Rekayasa Perangkat Lunak Karangan Rosa A.S dan M. Shalahudin 2015)

2.7.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu. Banyaknya sekuen diagram minimal sama dengan banyaknya *use case* (Rosa A.S dan M. Shalahudin: 2015).

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2.3: Simbol-simbol pada diagram sekuen

Simbol	Deskripsi
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
Garis hidup 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek

1 : nama_mendote()	lain atau dirinya sendiri.
Pesan tipe send 1 : masukan	Menyatakan bahwa suatu objek mengirimkan data ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

(Sumber Buku Rekayasa Perangkat Lunak Karangan Rosa A.S dan M. Shalahudin 2015)

2.7.4 Class Diagram

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Dan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa A.S dan M. Shalahudin: 2015).

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

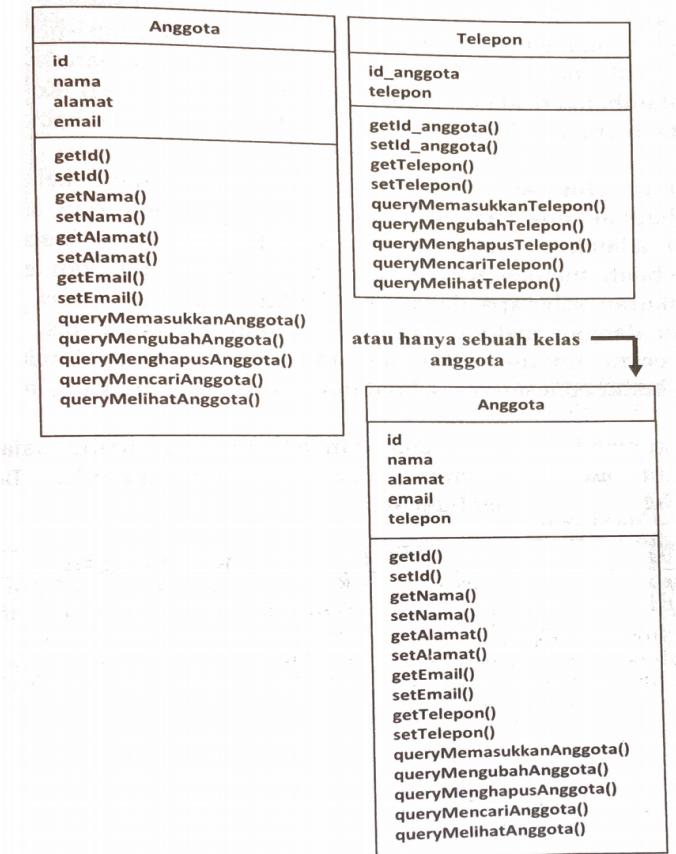
3. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalue pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggungjawabkan atau tetap ada di dalam perangcangan kelas (Rosa A.S dan M. Shalahudin: 2015).



Gambar 2.7: Contoh kelas diagram

(Sumber Buku Rekayasa Perangkat Lunak Karangan Rosa A.S dan M. Shalahudin 2015)

BAB III

ANALISA PERANCANGAN SISTEM

3.1 Analisa Data

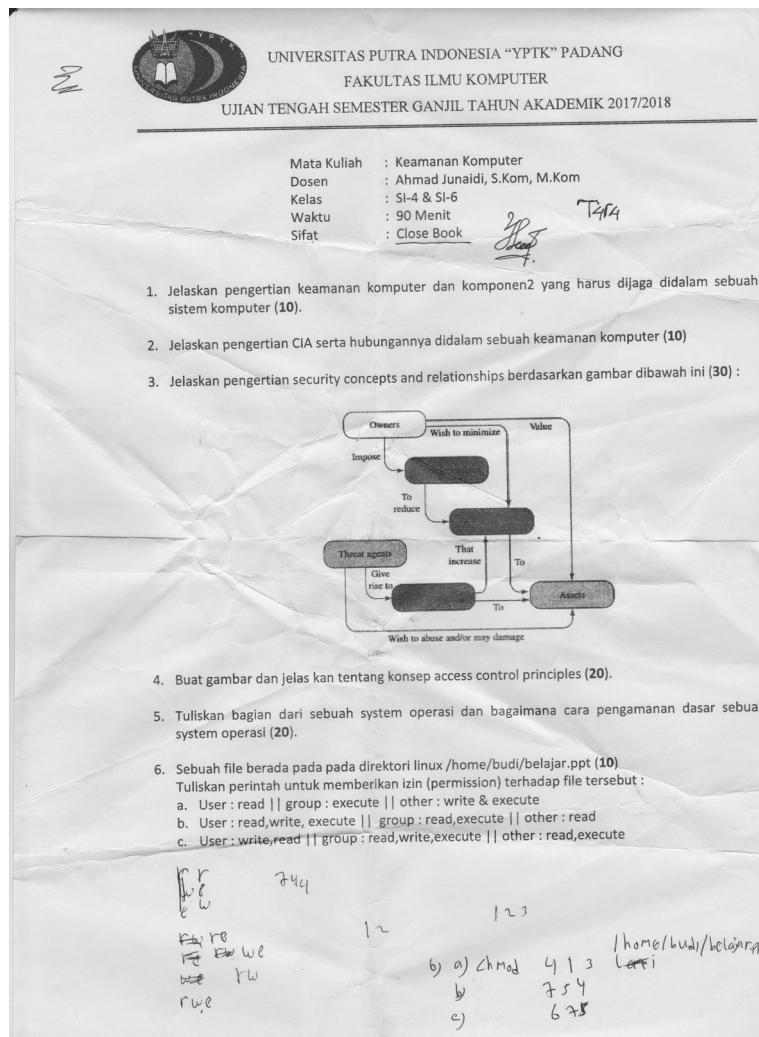
Pada tahap ini, dilakukan evaluasi kinerja sistem lama, identifikasi masalah.

Tahap analisa data merupakan tahap yang paling penting dalam, rancangan sistem serta langkah-langkah yang dibutuhkan untuk mengembangkan sistem baru. Pada penelitian, digunakan bahasa pemrograman *javascript* dengan bantuan perangkat lunak *nodejs* untuk menciptakan aplikasi *computer assisted test* dan *database postgresql* untuk menyimpan *data* sistem.

Pada penelitian ini dilakukan analisa terhadap lembar jawaban dan lembar soal yang biasanya digunakan pada saat ujian. Karena kedua hal ini adalah hal yang paling penting didalam pelaksanaan sebuah ujian.

3.1.1 Data Soal

Pada Universitas Putra Indonesia “YPTK” Padang, soal ujian dibuat oleh dosen menggunakan aplikasi pengolah kata seperti *Microsoft Word* atau sejenisnya. Dan nantinya akan dicetak serta diperbanyak dan dibagikan kepada peserta ujian. Berikut adalah gambar dari salah satu lembar soal ujian :



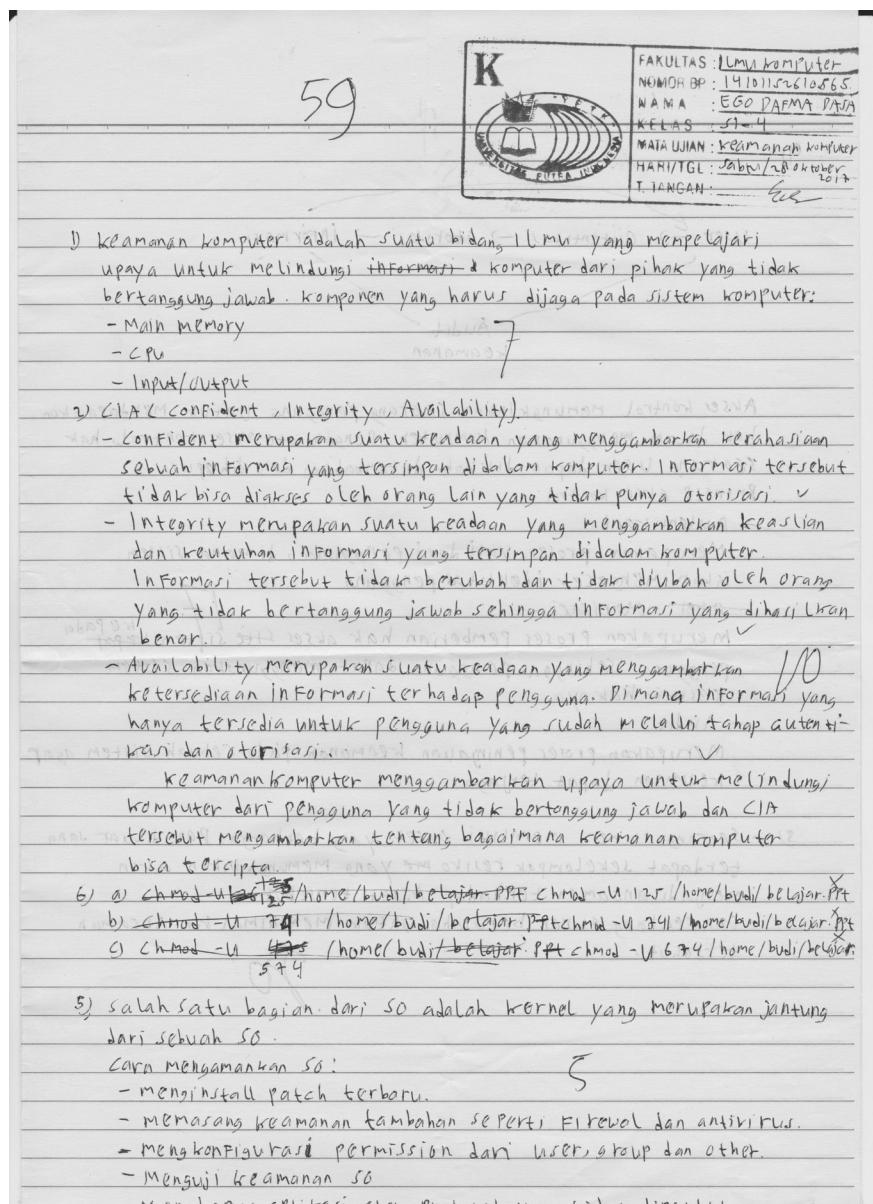
Gambar 3.1: Lembar Soal Ujian

Banyak kertas soal ujian biasanya disediakan sebanyak mahasiswa yang mengikuti ujian atau dilebihkan sedikit untuk berjaga-jaga. Dan dibutuhkan biaya besaran tertentu untuk mencetak soal.

3.1.2 Data Lembar Jawaban

Selain lembar soal, lembar jawaban juga disediakan oleh penyelenggara ujian dan biasanya menggunakan kertas *double folio*. Dibagian atas kanan kertas

sendiri sudah dibuat semacam hasil stempel yang nantinya digunakan untuk mengisi biodata mahasiswa. Hal tersebut bertujuan agar semua lembar jawaban seragam sehingga tidak menyulitkan dosen saat melakukan pemeriksaan. Berikut adalah gambar lembar jawaban ujian :



Gambar 3.2: Lembar Jawaban Ujian

Agar memudahkan pemeriksaan, dosen secara langsung menuliskan nilai bobot yang didapat pada setiap soal langsung pada lembar jawaban agar mudah untuk ditotalkan bobotnya nanti. Dan juga nilai akhir dituliskan pada lembar jawaban sehingga jika lembar jawaban dikembalikan kepada mahasiswa, mahasiswa bisa tahu nilai akhir yang didapat.

3.2 Analisa Proses

Perancangan sistem ini dibuat untuk mengetahui tentang apa saja yang akan dibuat pada sistem selanjutnya serta memberikan gambaran yang jelas tentang sistem yang akan dikembangkan. Sesuai dengan metode pendekatan yang akan digunakan adalah pendekatan *object oriented*, maka model yang digunakan untuk menggambarkan seluruh proses dan objek adalah dengan menggunakan UML.

3.2.1 Analisa Proses Sistem Yang Sedang Berjalan

Berikut adalah gambaran sistem pengadaan ujian pada Universitas Putra Indonesia “YPTK” Padang yang sedang berjalan :

1. Dosen membuat soal untuk pelaksanaan ujian pada aplikasi *office* seperti *Microsoft word* atau sejenisnya.
2. Kemudian dosen memberikan soal tersebut kepada panitia pelaksana ujian dan nantinya soal tersebut dicetak berdasarkan jumlah mahasiswa yang ikut ujian.

3. Lembar jawaban berupa kertas doublefolio juga disediakan bagi peserta ujian.
4. Mahasiswa datang ujian sesuai dengan jadwal yang telah ditentukan.
5. Pada saat ujian, mahasiswa diwajibkan untuk menunjukkan kartu ujian.
6. Mahasiswa mengerjakan soal dan mengumpulkan lembar jawaban jika ujian telah selesai dilaksanakan.
7. Dosen menerima lembar jawaban dari panitia ujian dan melakukan pemeriksaan terhadap lembar jawaban tersebut dengan menuliskan nilai ujian langsung pada lembar jawaban mahasiswa.

3.2.2 Desain Sistem Baru

Berdasarkan hasil analisa proses sistem yang sedang berjalan, berikut adalah rencana rancangan sistem baru yang akan dikembangkan :

1. Bagi dosen yang ingin menggunakan aplikasi *CAT*, terlebih dahulu harus menambahkan kuliah yang ingin diujangkan.
2. Kemudian dosen menambahkan daftar mahasiswa yang mengikuti kuliah tersebut.
3. Dan dosen sudah bisa menambahkan ujian baru sesuai dengan kuliah yang telah ditambahkan tadi. Ujian dapat berupa UAS,UTS dan lain-lain.
4. Mahasiswa kemudian *login* ke aplikasi *CAT* dan memilih ujian yang akan diikuti.

5. Mahasiswa menjawab pertanyaan yang muncul dilayar komputer. Jika pertanyaan berupa pilihan ganda, maka mahasiswa tinggal mengklik pilihan yang diyakini benar. Jika berupa essai, mahasiswa mengetikkan jawabannya pada layar dan mengklik tombol simpan.
6. Halaman ujian otomatis tertutup jika waktu ujian sudah habis dan semua jawaban mahasiswa akan disimpan otomatis.
7. Dosen dapat melihat hasil ujian secara langsung jika tipe soalnya pilihan ganda dan jika tipe soal essai, maka dosen harus memeriksa jawaban mahasiswa satu per satu secara langsung pada aplikasi *CAT* tersebut.

3.3 Perancangan UML

UML merupakan pemodelan yang digunakan untuk menggambarkan cara kerja sistem sebuah perangkat lunak menggunakan *use case diagram, class diagram, sequence diagram, dan activity diagram*.

Semua diagram *UML* pada penelitian ini dibuat menggunakan aplikasi *open source* bernama *umlet*. Berikut adalah penjelasan dari perancangan sistem baru menggunakan *UML*.

3.3.1 Use Case Diagram

Use case diagram adalah abstraksi dari interaksi antara sistem dengan aktor atau pengguna sistem. *Use case diagram* menggambarkan interaksi apa saja yang dapat dilakukan seorang pengguna sistem terhadap sistem.

3.3.1.1 Defenisi Aktor

Aktor merupakan pengguna sistem. Berikut adalah tabel yang menunjukkan aktor pada aplikasi *computer assisted test* pada penelitian ini :

Tabel 3.1: Definisi Aktor

No	Aktor	Deskripsi
1	Admin	Aktor yang mengelola sistem.
2	Dosen	Aktor yang akan memproses hasil ujian serta melihat hasil akhir ujian mahasiswa.
3	Mahasiswa	Aktor yang akan mengikuti ujian.

3.3.1.2 Daftar Use Case

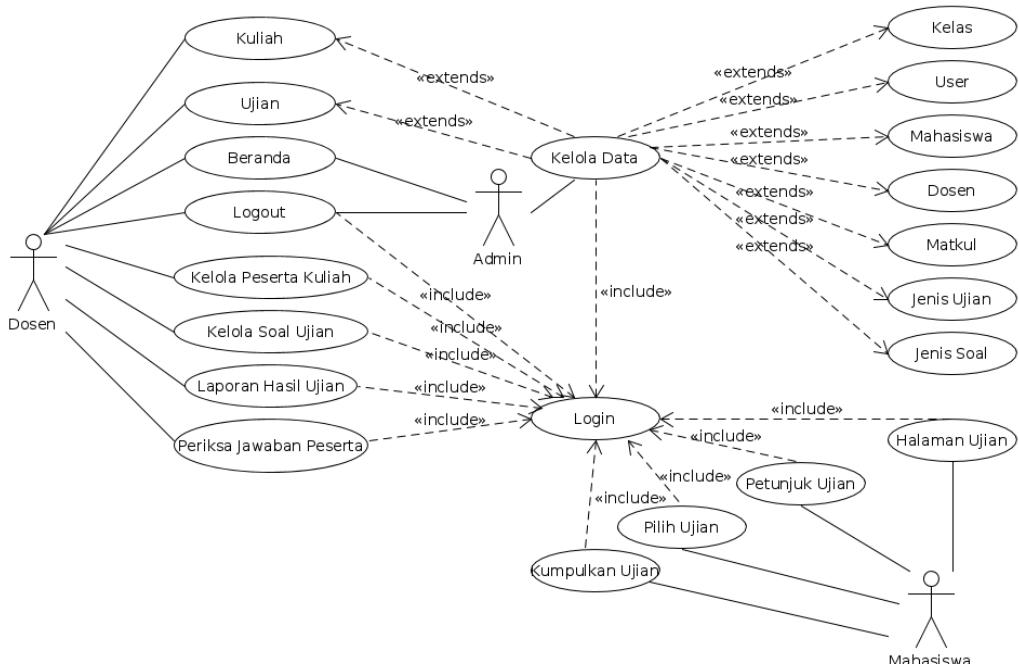
Use case menggambarkan tentang apa yang biasa dilakukan oleh seorang aktor terhadap sebuah sistem. Berikut adalah daftar *use case* beserta aktor yang terdapat pada penelitian ini :

1. Beranda : Admin, Dosen
2. Kelola User : Admin
3. Kelola Mahasiswa : Admin
4. Kelola Dosen : Admin
5. Kelola Matkul : Admin
6. Kelola Jenis Soal : Admin
7. Kelola Jenis Ujian : Admin
8. Kelola Kelas : Admin
9. Kelola Ujian : Admin, Dosen

10. Kelola Soal Ujian : Dosen
11. Kelola Peserta Kuliah : Dosen
12. Laporan Hasil Ujian : Dosen
13. Periksa Jawaban Peserta : Dosen
14. Login :Admin, Dosen, Mahasiswa
15. Petunjuk Ujian : Mahasiswa
16. Halaman Ujian : Mahasiswa
17. Kumpulkan Ujian : Mahasiswa
18. Pilih Ujian : Mahasiswa
19. Logout :Admin, Dosen
20. Kelola Kuliah : Admin, Dosen

Berdasarkan daftar *use case* diatas, berikut adalah gambar dari *use case*

diagramnya :



Gambar 3.3: Use Case Diagram

3.3.1.3 Skenario Use Case

Aktifitas detail seorang aktor terhadap sistem akan digambarkan pada tabel berikut :

1. Beranda (Admin, Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase beranda yang akan dirancang.

Tabel 3.2 Skenario Usecase Kelola User (Admin, Dosen)

Identifikasi	
No	1
Nama	Beranda
Tujuan	Aktor melihat beranda
Deskripsi	Aktor mengakses beranda
Tipe :-	
Aktor : Admin, Dosen	
Skenario Utama	
Kondisi awal : Login	
Aksi aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem menampilkan halaman beranda
Aktor melihat halaman beranda	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem gagal menampilkan halaman beranda

Entri username dan password kembali	Sistem cek username dan password
	Sistem menampilkan halaman beranda
Aktor melihat halaman beranda	
Kondisi Akhir : Aktor berhasil melihat halaman beranda	

2. Kelola User (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola user yang akan dirancang.

Tabel 3.3 Skenario Usecase Kelola User (Admin)

Identifikasi	
No	2
Nama	Kelola user
Tujuan	Aktor mengatur data user
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data user
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data user(simpan,tambah,hapus,edit,cari)	Validasi data user dan sistem menyimpan data kelola user
Submit/Ok	Informasi kelola data user
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data user(simpan,tambah,hapus,edit) kembali	Validasi data user kelola user

Submit/Ok kembali	Informasi kelola data user
Kondisi Akhir : Aktor berhasil mengelola data user	

3. Kelola Mahasiswa (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola mahasiswa yang akan dirancang.

Tabel 3.4 Skenario Usecase Kelola Mahasiswa (Admin)

Identifikasi	
No	3
Nama	Kelola mahasiswa
Tujuan	Aktor mengatur data mahasiswa
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data mahasiswa
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data mahasiswa(simpan,tambah,hapus,edit,caril)	Validasi data mahasiswa dan sistem menyimpan data kelola mahasiswa
Submit/Ok	Informasi kelola data mahasiswa
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data mahasiswa(simpan,tambah,hapus,edit) kembali	Validasi data mahasiswa kelola mahasiswa
Submit/Ok kembali	Informasi kelola data mahasiswa
Kondisi Akhir : Aktor berhasil mengelola data mahasiswa	

4. Kelola Dosen (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola dosen yang akan dirancang.

Tabel 3.5 Skenario Usecase Kelola Dosen (Admin)

Identifikasi	
No	4
Nama	Kelola dosen
Tujuan	Aktor mengatur data dosen
Deskripsi	Aktor membaca, mengedit, membuat dan menghapus data dosen
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data dosen(simpan,tambah,hapus,edit,cari)	Validasi data dosen dan sistem menyimpan data kelola dosen
Submit/Ok	Informasi kelola data dosen
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data dosen(simpan,tambah,hapus,edit) kembali	Validasi data dosen kelola dosen
Submit/Ok kembali	Informasi kelola data dosen
Kondisi Akhir : Aktor berhasil mengelola data dosen	

5. Kelola Matkul (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola dosen yang akan dirancang.

Tabel 3.6 Skenario Usecase Kelola Matkul (Admin)

Identifikasi	
No	5
Nama	Kelola matkul
Tujuan	Aktor mengatur data matkul
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data matkul
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data matkul(simpan,tambah,hapus,edit,cari)	Validasi data matkul dan sistem menyimpan data kelola matkul
Submit/Ok	Informasi kelola data matkul
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data matkul(simpan,tambah,hapus,edit) kembali	Validasi data matkul kelola matkul
Submit/Ok kembali	Informasi kelola data matkul
Kondisi Akhir : Aktor berhasil mengelola data matkul	

6. Kelola Jenis Soal (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase jenis soal yang akan dirancang.

Tabel 3.7 Skenario Usecase Kelola Jenis Soal (Admin)

Identifikasi	
No	6
Nama	Kelola jenis soal
Tujuan	Aktor mengatur data jenis soal
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data jenis soal
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data jenis soal(simpan,tambah,hapus,edit,cari)	Validasi data jenis soal dan sistem menyimpan data kelola jenis soal
Submit/Ok	Informasi kelola data jenis soal
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data jenis soal(simpan,tambah,hapus,edit) kembali	Validasi data jenis soal kelola jenis soal
Submit/Ok kembali	Informasi kelola data jenis soal
Kondisi Akhir : Aktor berhasil mengelola data jenis soal	

7. Kelola Jenis Ujian (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola jenis ujian yang akan dirancang.

Tabel 3.8 Skenario Usecase Kelola Jenis Ujian (Admin)

Identifikasi	
No	7
Nama	Kelola jenis ujian
Tujuan	Aktor mengatur data jenis ujian
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data jenis ujian
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data jenis ujian(simpan,tambah,hapus,edit,cari)	Validasi data jenis ujian dan sistem menyimpan data kelola jenis ujian
Submit/Ok	Informasi kelola data jenis ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data jenis ujian(simpan,tambah,hapus,edit) kembali	Validasi data jenis ujian kelola jenis ujian
Submit/Ok kembali	Informasi kelola data jenis ujian
Kondisi Akhir : Aktor berhasil mengelola data jenis ujian	

8. Kelola Kelas (Admin)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola kelas yang akan dirancang.

Tabel 3.9 Skenario Usecase Kelola Kelas (Admin)

Identifikasi	
No	8
Nama	Kelola Kelas
Tujuan	Aktor mengatur data Kelas
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data Kelas
Tipe :-	
Aktor : Admin	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data Kelas(simpan,tambah,hapus,edit,cari)	Validasi data Kelas dan sistem menyimpan data kelola Kelas
Submit/Ok	Informasi kelola data Kelas
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data Kelas(simpan,tambah,hapus,edit) kembali	Validasi data Kelas kelola Kelas
Submit/Ok kembali	Informasi kelola data Kelas
Kondisi Akhir : Aktor berhasil mengelola data Kelas	

9. Kelola Ujian (Admin, Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola ujian yang akan dirancang.

Tabel 3.10 Skenario Usecase Kelola Ujian (Dosen)

Identifikasi	
No	9
Nama	Kelola ujian
Tujuan	Aktor mengatur data ujian
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data ujian
Tipe :-	
Aktor : Admin, Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data ujian(simpan,tambah,hapus,edit,cari)	Validasi data ujian dan sistem menyimpan data kelola ujian
Submit/Ok	Informasi kelola data ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data ujian(simpan,tambah,hapus,edit) kembali	Validasi data ujian kelola ujian
Submit/Ok kembali	Informasi kelola data ujian
Kondisi Akhir : Aktor berhasil mengelola data ujian	

10. Kelola Soal Ujian (Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola soal ujian yang akan dirancang.

Tabel 3.11 Skenario Usecase Kelola Soal Ujian (Dosen)

Identifikasi	
No	10
Nama	Kelola soal ujian
Tujuan	Aktor mengatur data soal ujian
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data soal ujian
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data soal ujian(simpan,tambah,hapus,edit,cari)	Validasi data soal ujian dan sistem menyimpan data kelola soal ujian
Submit/Ok	Informasi kelola data soal ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data soal ujian(simpan,tambah,hapus,edit) kembali	Validasi data soal ujian kelola soal ujian
Submit/Ok kembali	Informasi kelola data soal ujian
Kondisi Akhir : Aktor berhasil mengelola data soal ujian	

11. Kelola Peserta Kuliah (Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola peserta kuliah yang akan dirancang.

Tabel 3.12 Skenario Usecase Kelola Peserta Kuliah (Dosen)

Identifikasi	
No	11
Nama	Kelola peserta kuliah
Tujuan	Aktor mengatur data peserta kuliah
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus data peserta kuliah
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data peserta kuliah(simpan,tambah,hapus,cari)	Validasi data peserta kuliah dan sistem menyimpan data kelola peserta kuliah
Submit/Ok	Informasi kelola data peserta kuliah
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola data peserta kuliah(simpan,tambah,hapus,edit) kembali	Validasi data peserta kuliah kelola peserta kuliah
Submit/Ok kembali	Informasi kelola data peserta kuliah
Kondisi Akhir : Aktor berhasil mengelola data peserta kuliah	

12. Laporan Hasil Ujian (Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase laporan hasil ujian yang akan dirancang.

Tabel 3.13: Skenario Use Case Laporan Hasil Ujian (Dosen)

Identifikasi	
No	12
Nama	Laporan Hasil Ujian
Tujuan	Aktor mengetahui hasil ujian mahasiswa
Deskripsi	Aktor membaca laporan hasil ujian mahasiswa
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Klik menu laporan hasil ujian	Sistem menampilkan laporan hasil ujian
Aktor mengentri kata kunci untuk pencarian hasil ujian mahasiswa	Sistem menampilkan hasil pencarian hasil ujian
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik menu laporan hasil ujian	Sistem tidak menampilkan laporan ujian
Klik kembali laporan hasil ujian	Sistem menampilkan laporan hasil ujian
Kondisi Akhir : Aktor berhasil melihat laporan hasil ujian	

13. Periksa Jawaban Peserta (Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase periksa jawaban peserta yang akan dirancang.

Tabel 3.14: Skenario Use Case Periksa Jawaban Peserta (Dosen)

Identifikasi	
No	13
Nama	Periksa Jawaban
Tujuan	Aktor memeriksa jawaban ujian peserta
Deskripsi	Aktor membaca dan memberikan nilai bobot paa setiap jawaban peserta
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman Kelola Ujian	
Aksi aktor	Reaksi Sistem
Klik menu peserta ujian	Sistem menampilkan daftar peserta ujian
Klik tombol periksa jawaban	Sistem menampilkan halaman periksa jawaban beserta jawaban peserta
Aktor memberikan nilai bobot disetiap jawaban peserta dan mengklik tombol simpan	Sistem menyimpan nilai bobot
Klik tombol simpan hasil periksa	Sistem menyimpan hasil pemeriksaan jawaban peserta
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik menu peserta ujian	Sistem tidak menampilkan peserta ujian

Klik kembali menu peserta ujian	Sistem menampilkan peserta ujian
Klik menu periksa jawaban	Sistem menampilkan halaman periksa jawaban
Klik simpan hasil periksa	Sistem tidak berhasil menyimpan hasil periksa
Klik kemmbali simpan hasil periksa	Sistem berhasil menyimpan hasil periksa
Kondisi Akhir : Aktor berhasil memeriksa jawaban peserta	

14. Login (Admin, Dosen, Mahasiswa)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase login yang akan dirancang.

Tabel 3.15: Skenario Use Case Login (Admin, Dosen, Mahasiswa)

Identifikasi	
No	14
Nama	Login
Tujuan	Aktor masuk kesistem
Deskripsi	Aktor mengentri informasi yang dibutuhkan untuk masuk kesistem
Tipe :-	
Aktor : Admin, Dosen, Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman login	
Aksi aktor	
Entri username dan password serta klik tombol login	Sistem mengecek username dan password
	Sistem menemukan username dan password yang cocok
	Sistem mengarahkan aktor kedalam

	sistem
Aktor berada didalam sistem	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem tidak berhasil mengecek username dan password
Entri kembali username dan password	Sistem berhasil mengecek username dan password
Kondisi Akhir : Aktor berhasil login	

15. Petunjuk Ujian (Mahasiswa)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase petunjuk ujian yang akan dirancang.

Tabel 3.16: Skenario Use Case Periksa Petunjuk Ujian (Mahasiswa)

Identifikasi	
No	15
Nama	Petunjuk Ujian
Tujuan	Aktor mengetahui petunjuk ujian
Deskripsi	Aktor membaca petunjuk ujian
Tipe :-	
Aktor : Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman login	
Aksi aktor	Reaksi Sistem
Login dengan username dan password	Sistem membawa aktor kehalaman petunjuk ujian
	Sistem menampilkan halaman

	petunjuk ujian
Aktor membaca petunjuk ujian	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Login dengan username dan password	Sistem membawa aktor kehalaman petunjuk ujian
	Sistem gagal menampilkan petunjuk ujian
Login kembali	Sistem berhasil menampilkan petunjuk ujian
Kondisi Akhir : Aktor berhasil mengetahui petunjuk ujian	

16. Halaman Ujian (Mahasiswa)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase halaman ujian yang akan dirancang.

Tabel 3.17: Skenario Use Case Periksa Jawaban Peserta (Mahasiswa)

Identifikasi	
No	16
Nama	Halaman Ujian
Tujuan	Aktor mengikuti ujian
Deskripsi	Aktor menjawab pertanyaan yang diujikan
Tipe :-	
Aktor : Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman petunjuk ujian	
Aksi aktor	Reaksi Sistem
Klik tombol masuk kehalaman ujian	Sistem menampilkan halaman ujian

Aktor membaca pertanyaan dan memilih atau mengetik jawaban dan mengklik tombol simpan jawaban	Sistem menyimpan jawaban peserta
Klik tombol kumpulkan ujian	Sistem menyimpan semua jawaban peserta
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor membaca dan menjawab serta mengklik simpan jawaban	Sistem gagal menyimpan jawaban
Aktor klik kembali simpan jawaban	Sistem berhasil menyimpan jawaban
Aktor klik kumpulkan ujian	Sistem tidak berhasil mengumpulkan ujian
Klik kembali kumpulkan ujian	Sistem berhasil mengumpulkan ujian
Kondisi Akhir : Ujian peserta terlaksana	

17. Kumpulkan Ujian (Mahasiswa)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase kumpulkan ujian yang akan dirancang.

Tabel 3.18: Skenario Use Case Periksa Kumpulkan Ujian Mahasiswa)

Identifikasi	
No	17
Nama	Kumpulkan Ujian
Tujuan	Aktor mengumpulkan ujian
Deskripsi	Aktor mengumpulkan ujian setelah selesai ujian
Tipe :-	
Aktor : Dosen	
Skenario Utama	

Kondisi awal : Halaman ujian	
Aksi aktor	Reaksi Sistem
Klik tombol kumpulkan ujian	Sistem mengumpulkan jawaban peserta dan menyimpannya
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik tombol kumpulkan ujian	Sistem gagal mengumpulkan jawaban peserta dan menyimpannya
Klik tombol kumpulkan ujian kembali	Sistem mengumpulkan jawaban peserta dan menyimpannya
Kondisi Akhir : Aktor berhasil logout	

18. Pilih Ujian (Mahasiswa)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase pilih ujian yang akan dirancang.

Tabel 3.19: Skenario Use Case Periksa Jawaban Peserta (Admin, Dosem)

Identifikasi	
No	18
Nama	Pilih ujian
Tujuan	Aktor memilih ujian
Deskripsi	Aktor memilih ujian yang akan diikuti
Tipe :-	
Aktor : Mahasiswa	
Skenario Utama	
Kondisi awal : Halaman login	
Aksi aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password

	Sistem menampilkan halaman pilih ujian
Aktor memilih ujian	
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Entri username dan password	Sistem cek username dan password
	Sistem gagal menampilkan halaman pilih ujian
Entri username dan password kembali	Sistem cek username dan password
	Sistem menampilkan halaman pilih ujian
Aktor memilih ujian	
Kondisi Akhir : Aktor berhasil memilih ujian	

19. Logout (Admin, Dosen)

Pada tabel dibawah ini dapat dilihat rincian skenario usecase logout yang akan dirancang.

Tabel 3.20: Skenario Use Case Periksa Jawaban Peserta (Admin, Dosem)

Identifikasi	
No	19
Nama	Logout
Tujuan	Aktor keluar dari sistem
Deskripsi	Aktor keluar dari sistem setelah menggunakan sistem
Tipe :-	
Aktor : Admin, Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	

Aksi aktor	Reaksi Sistem
Klik tombol logout	Sistem membawa aktor keluar sistem
	Sistem menampilkan halaman login
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Klik tombol logout	Sistem gagal membawa aktor keluar sistem
Klik tombol logout kemmbali	Sistem berhasil membawa aktor keluar
Kondisi Akhir : Aktor berhasil logout	

20. Kelola Kuliah (Admin, Dosen)

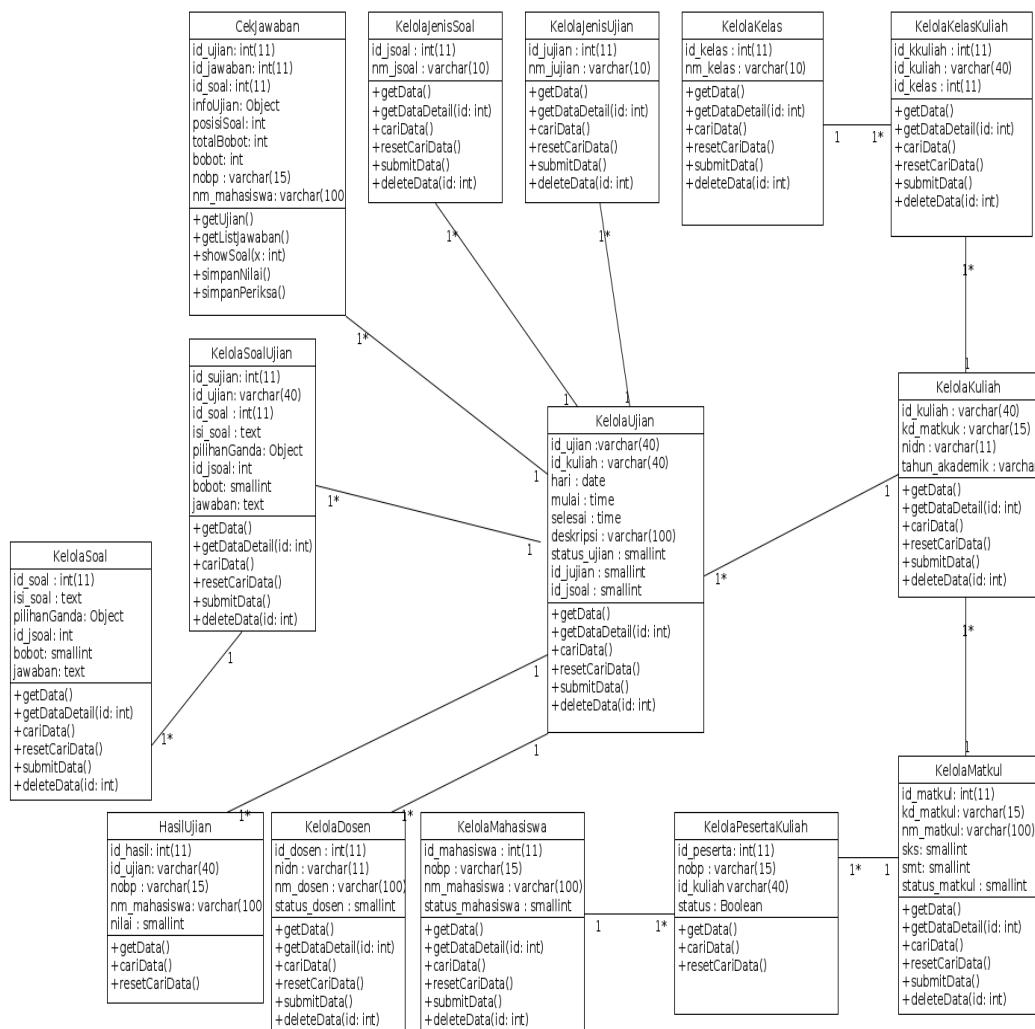
Pada tabel dibawah ini dapat dilihat rincian skenario usecase kelola kuliah yang akan dirancang.

Tabel 3.21 Skenario Usecase Kelola Kuliah (Admin, Dosen)

Identifikasi	
No	20
Nama	Kelolakuliah
Tujuan	Aktor mengatur datakuliah
Deskripsi	Aktor membaca,mengedit,membuat dan menghapus datakuliah
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola datakuliah(simpan,tambah,hapus,cari)	Validasi datakuliah dan sistem menyimpan data kelolakuliah
Submit/Ok	Informasi kelola datakuliah
Skenario Gagal	
Aksi Aktor	Reaksi Sistem
Aktor mengelola datakuliah(simpan,tambah,hapus,edit) kembali	Validasi datakuliah kelolakuliah
Submit/Ok kembali	Informasi kelola datakuliah
Kondisi Akhir : Aktor berhasil mengelola data kuliah	

3.3.2 Class Diagram

Class diagram struktur *class* serta deskripsi *class* dan *object* serta hubungan antar data-data yang terdapat pada basis data. Berikut adalah gambaran *class* diagram yang akan diterapkan nantinya :

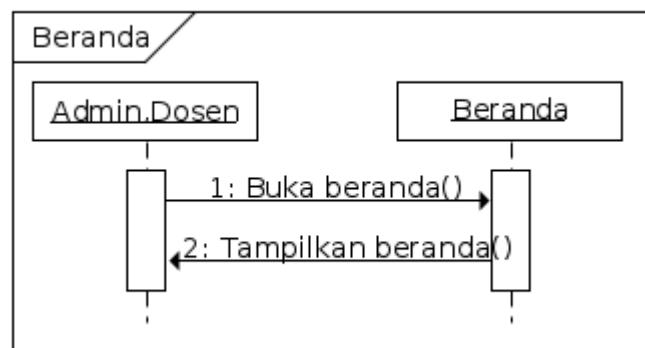


Gambar 3.4: Class Diagram

3.3.3 Sequence Diagram

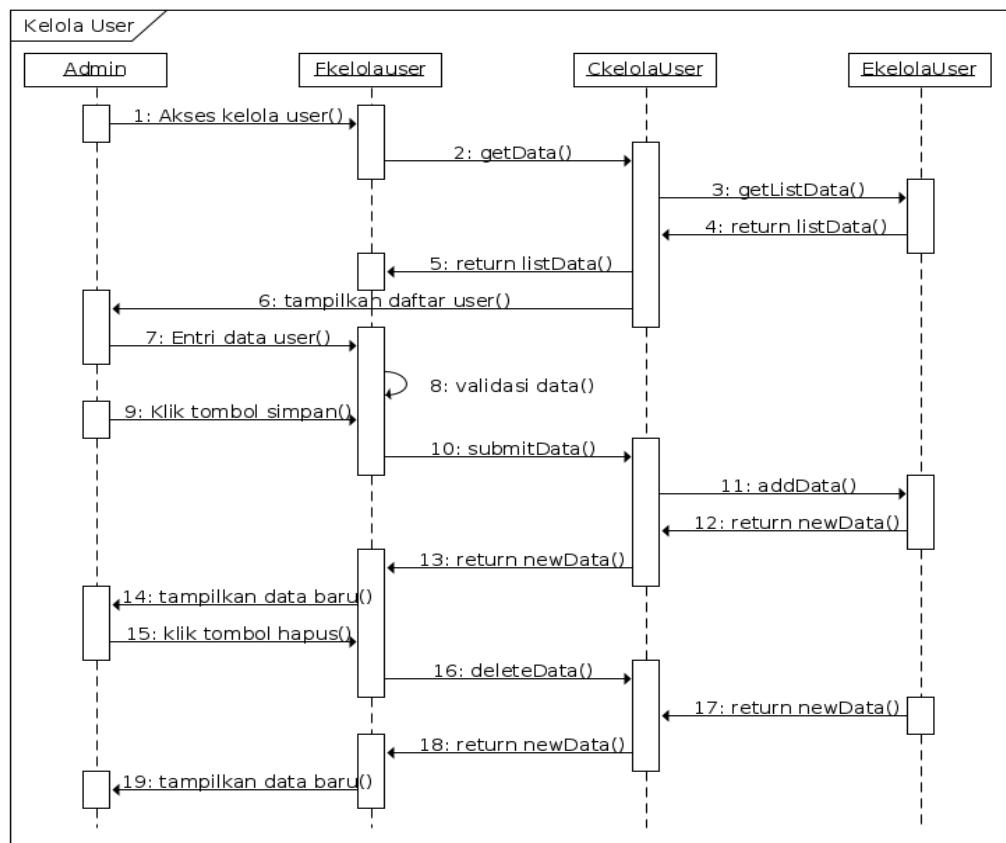
Sequence diagram menampilkan setiap eksekusi dari sebuah skenario *use case*. Interaksi digambarkan pada *sequence* diagram menggunakan garis hidup yang berjalan secara vertikal kebawah halaman dan urutan pesan dengan membaca ke arah bawah halaman (Andy Prasetyo Utomo, 2013). Banyaknya *sequence* diagram biasanya hampir sama dengan banyaknya *use case*. Berikut ini adalah gambar kumpulan *sequence* diagram pada aplikasi *computer assisted test* yang dikembangkan:

1. Beranda



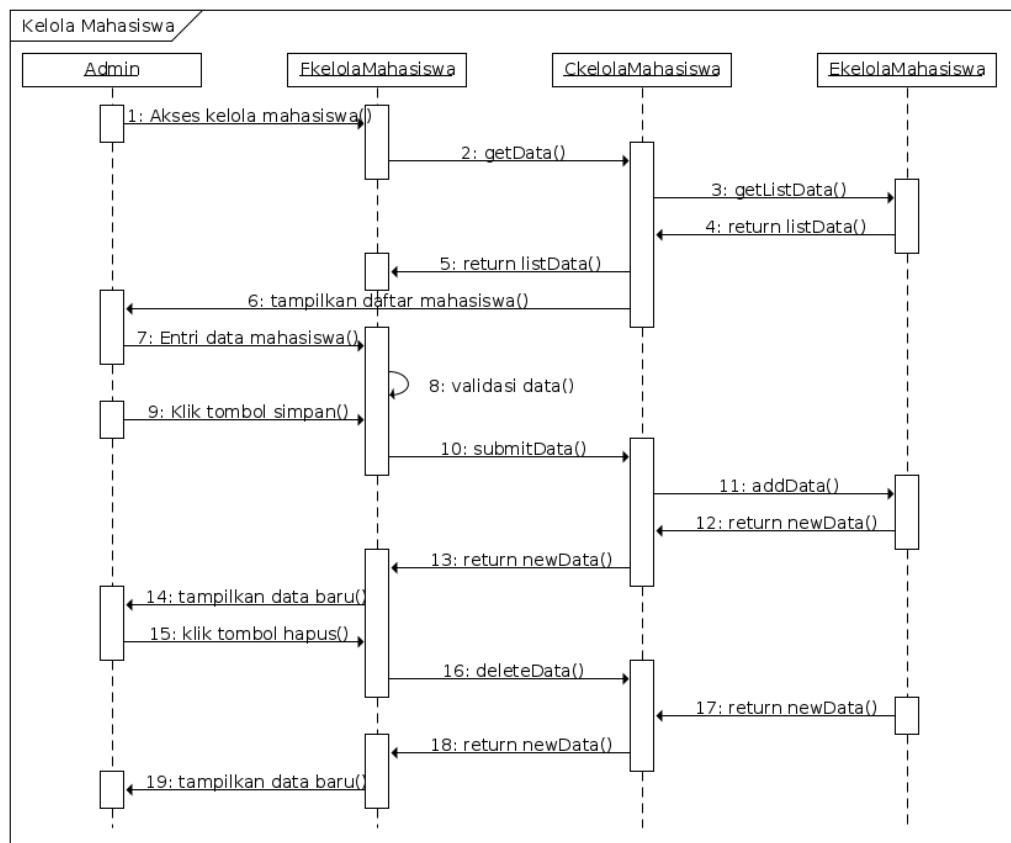
Gambar 3.5: Sequence Diagram Beranda

2. Kelola User



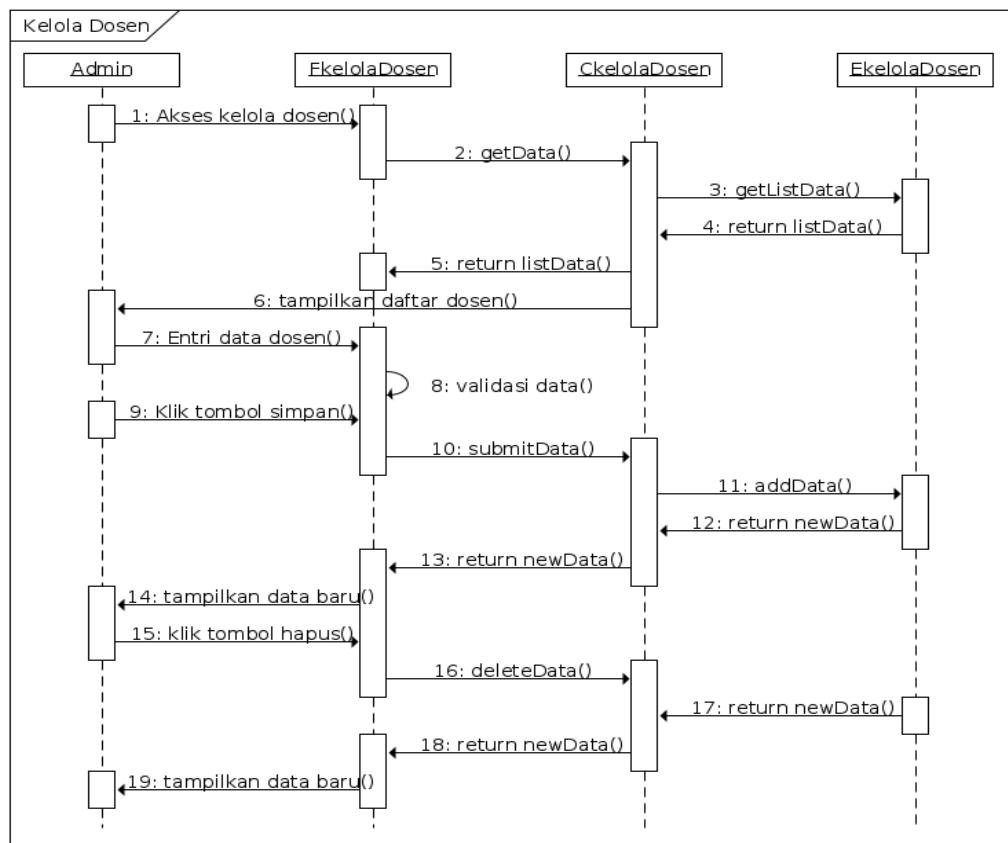
Gambar 3.6: *Sequence Diagram Kelola User*

3. Kelola Mahasiswa



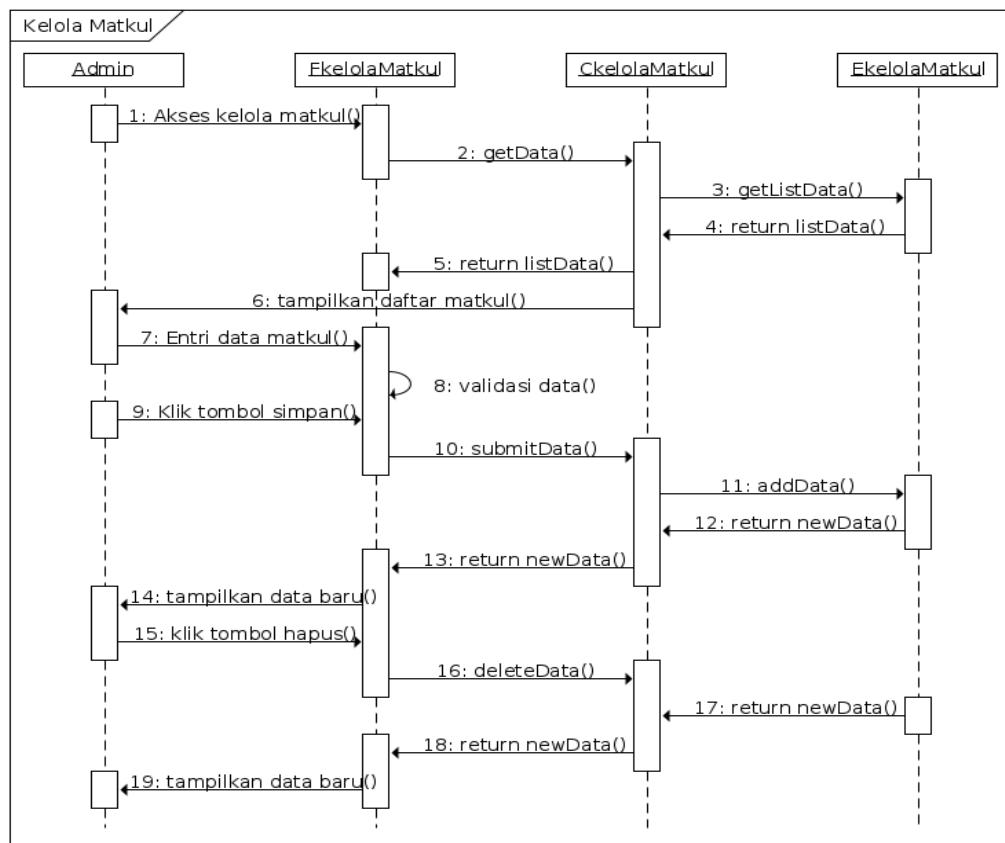
Gambar 3.7: Sequence Diagram Kelola Mahasiswa

4. Kelola Dosen



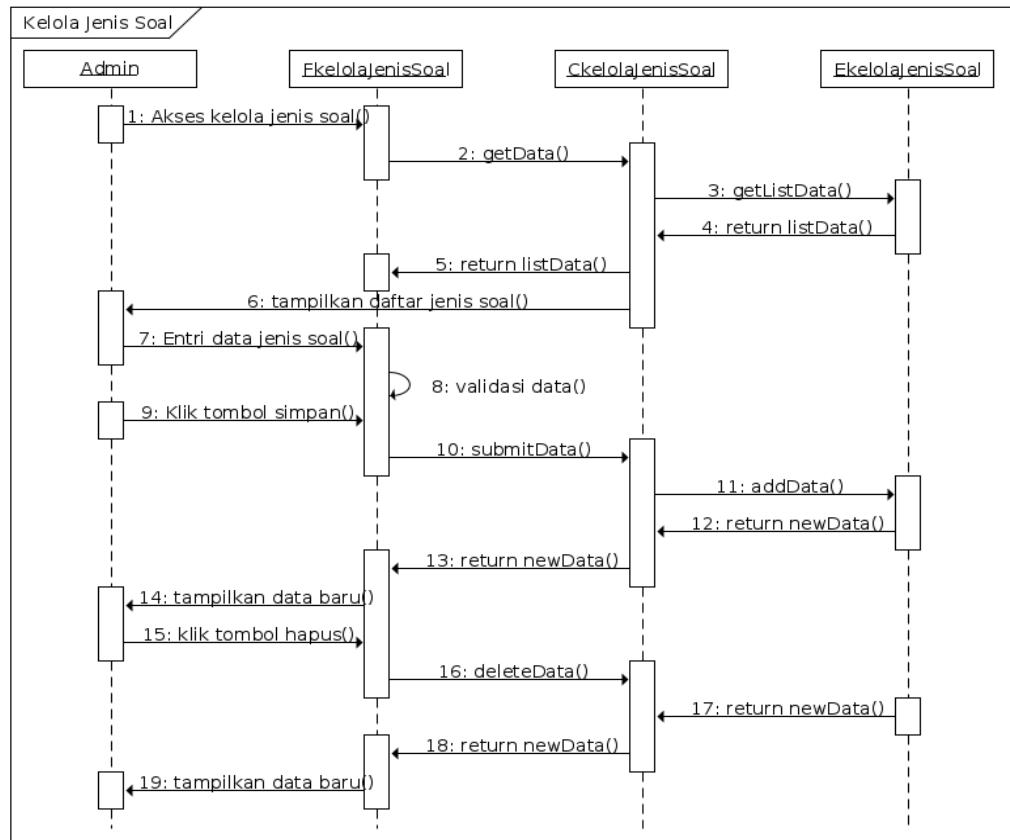
Gambar 3.8: Sequence Diagram Kelola Dosen

5. Kelola Matkul



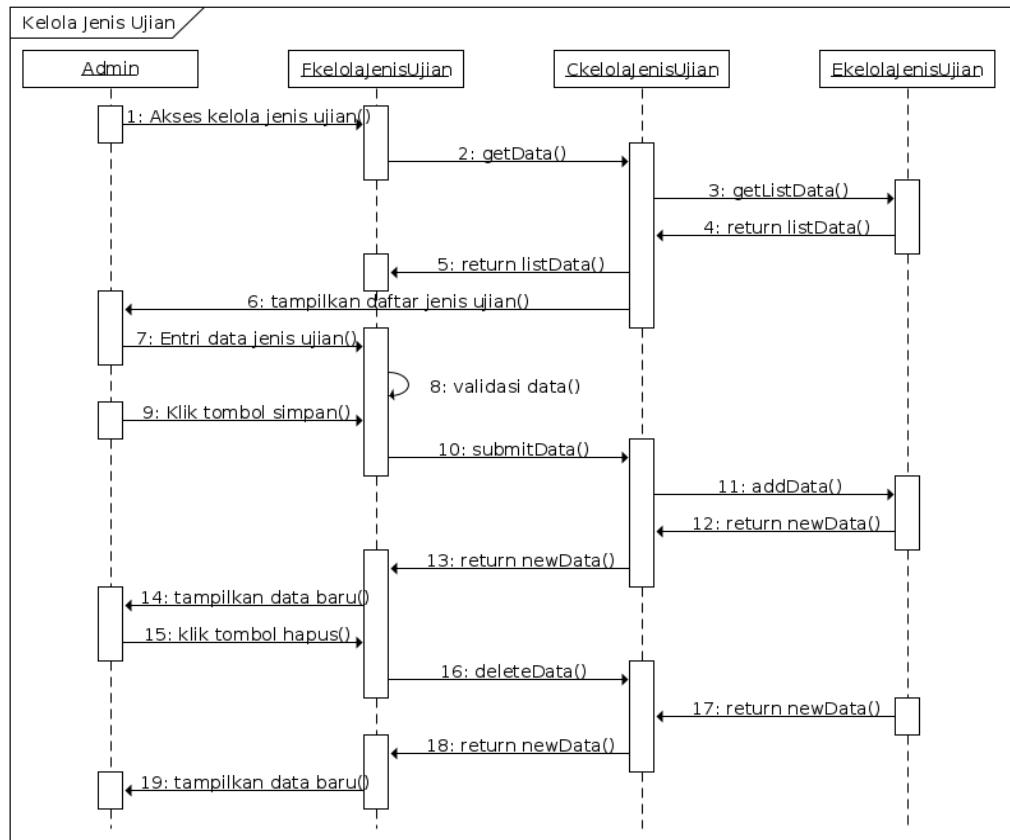
Gambar 3.9: Sequence Diagram Kelola Matkul

6. Kelola Jenis Soal



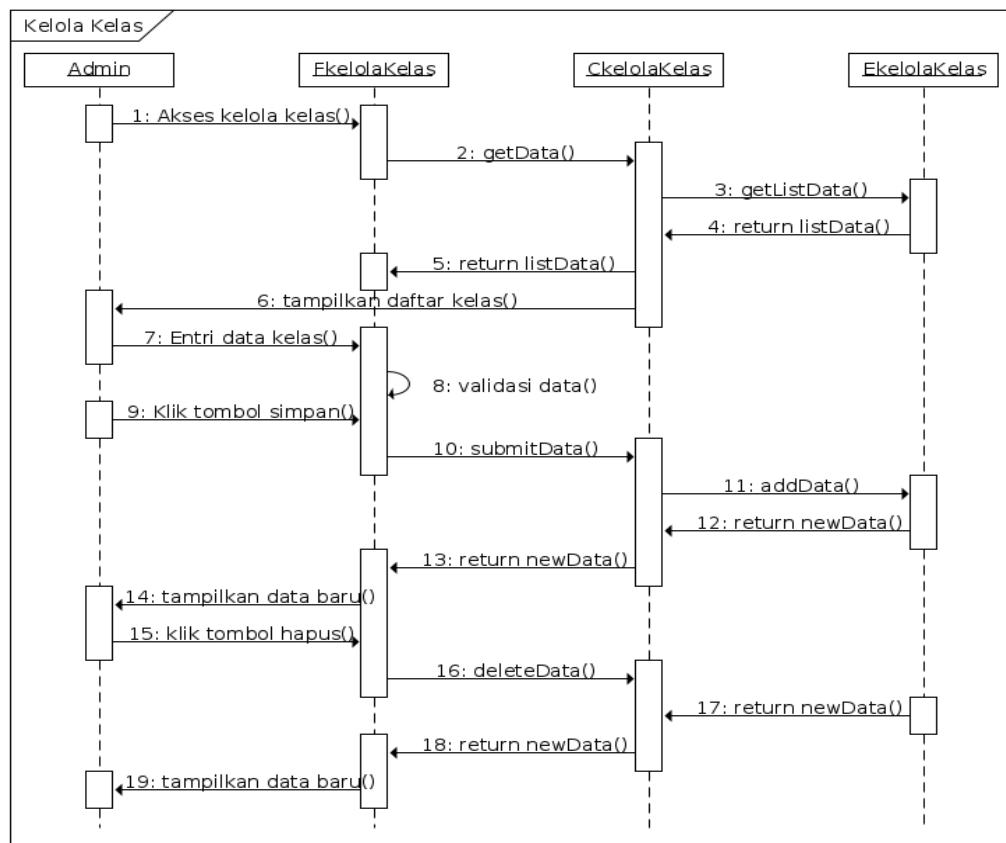
Gambar 3.10: Sequence Diagram Kelola Jenis Soal

7. Kelola Jenis Ujian



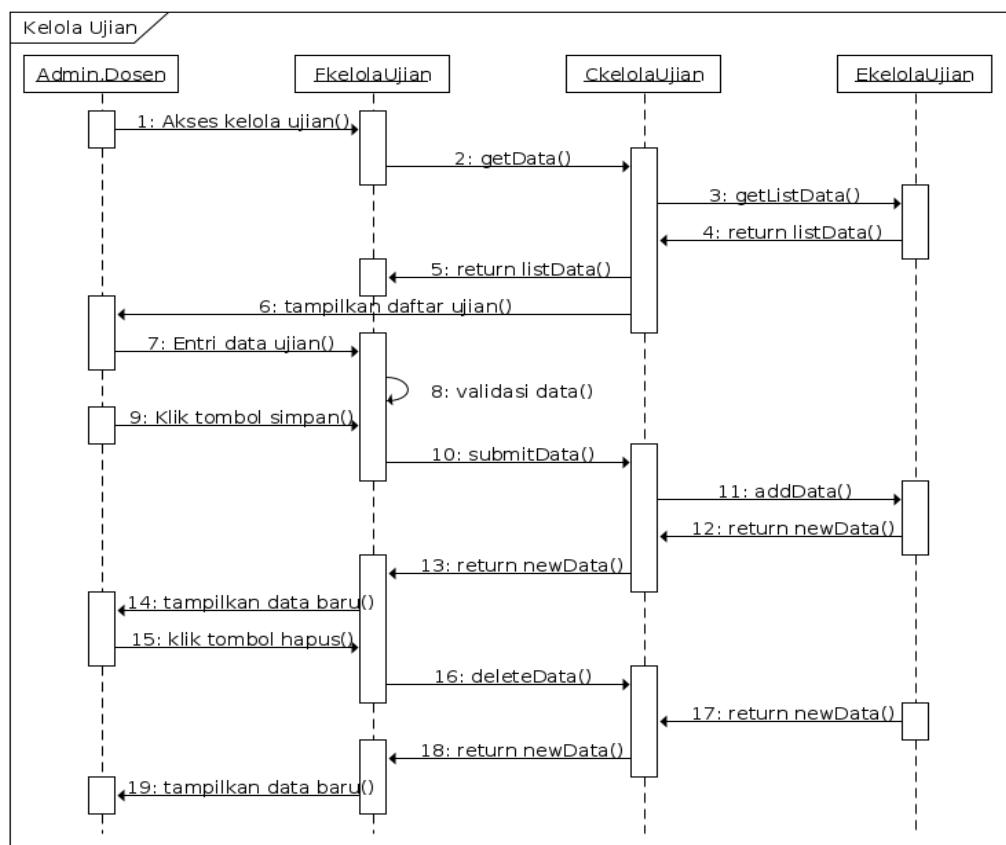
Gambar 3.11: Sequence Diagram Kelola Jenis Ujian

8. Kelola Kelas



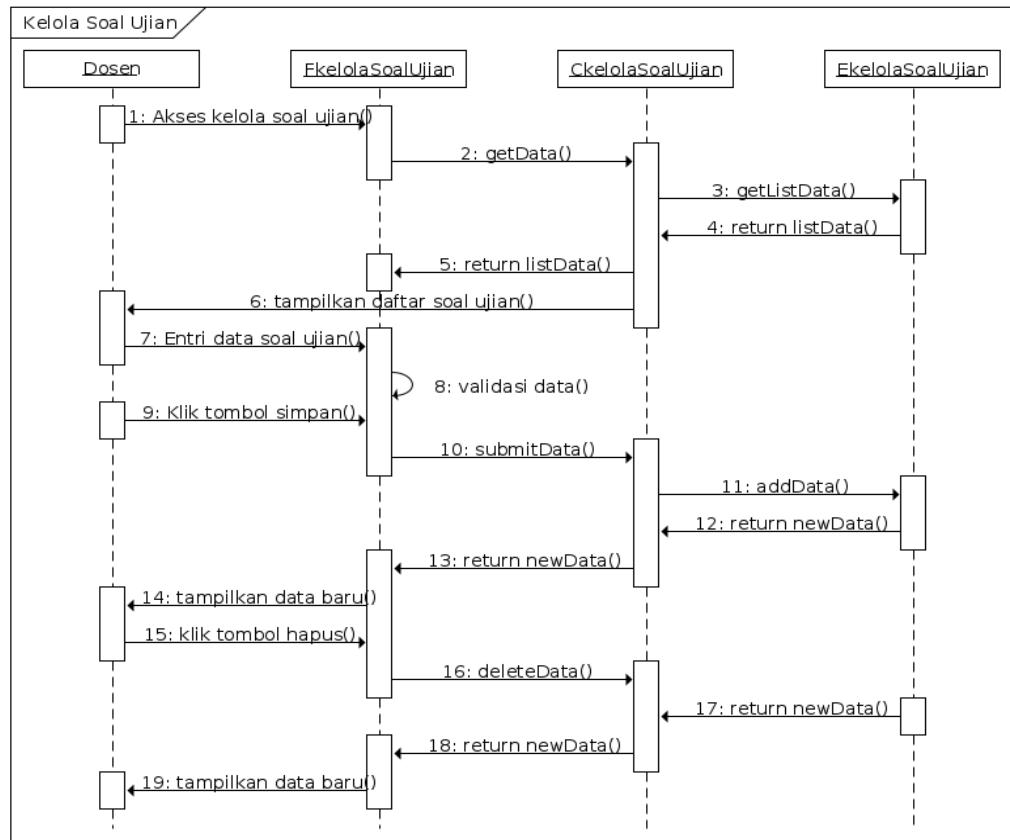
Gambar 3.12: Sequence Diagram Kelola Kelas

9. Kelola Ujian



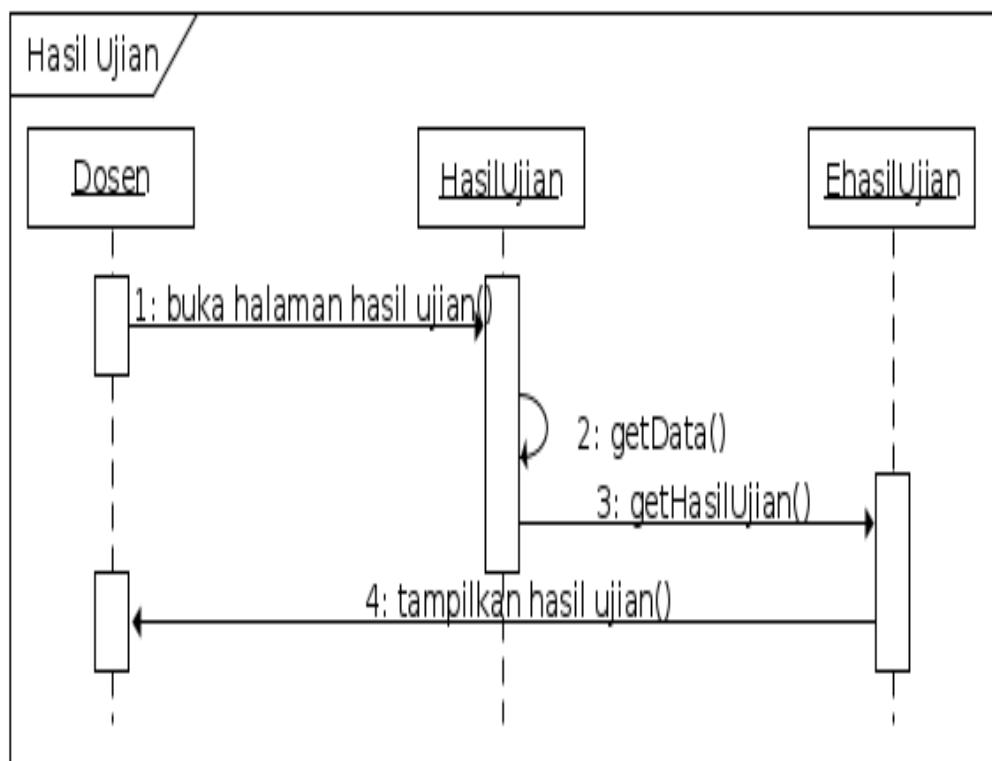
Gambar 3.13: Sequence Diagram Kelola Ujian

10. Kelola Soal Ujian



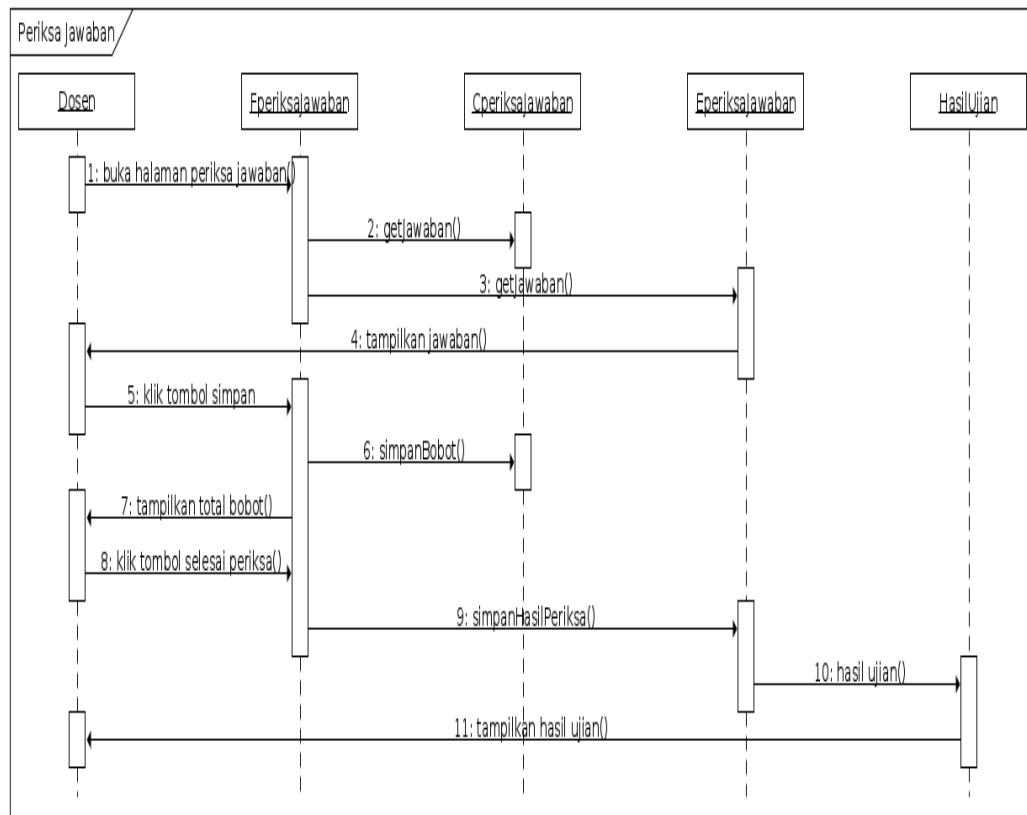
Gambar 3.14: *Sequence Diagram Kelola Soal Ujian*

11. Laporan Hasil Ujian



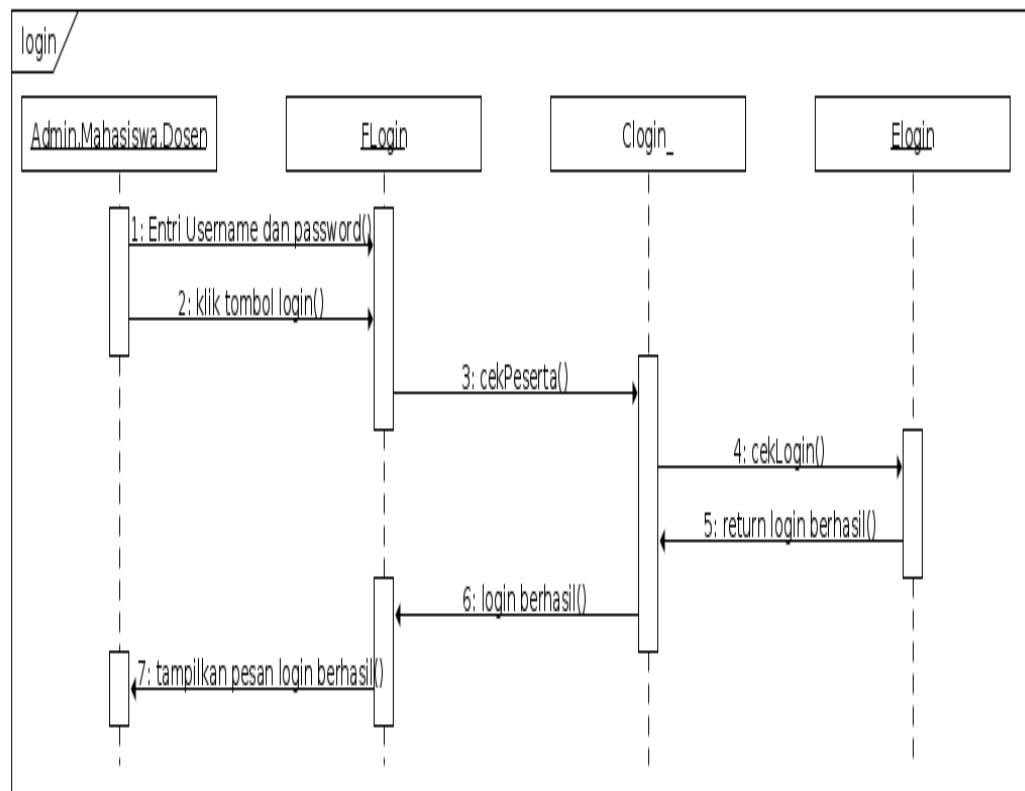
Gambar 3.15: Sequence Diagram Laporan Hasil Ujian

12. Periksa Jawaban Peserta



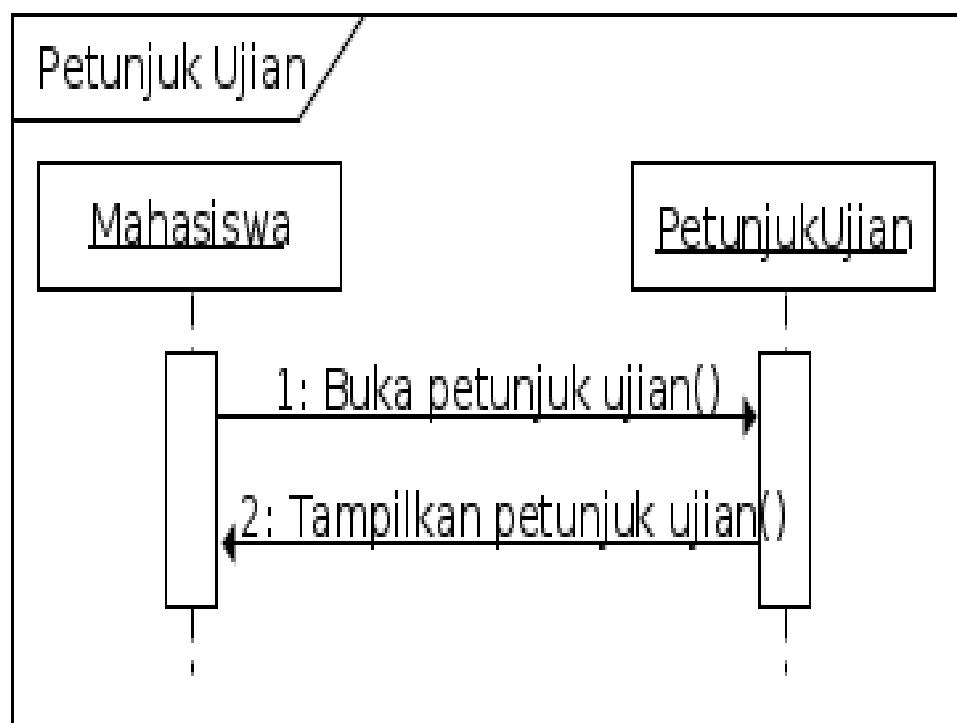
Gambar 3.16: Sequence Diagram Periksa Jawaban Peserta

13. Login



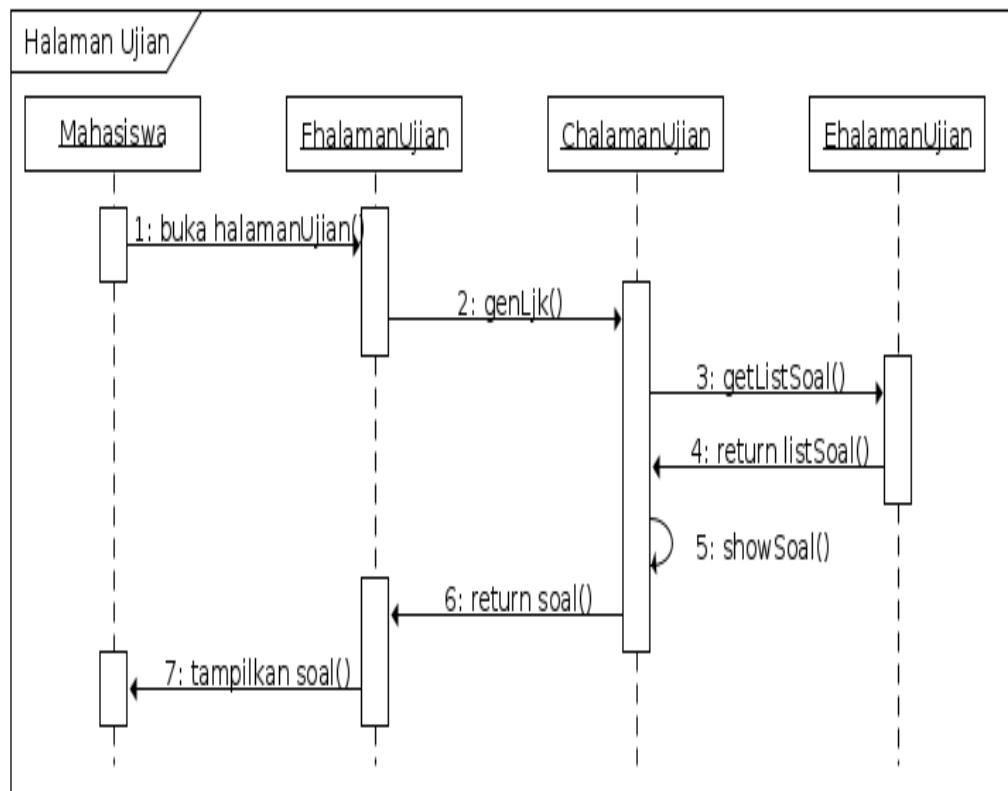
Gambar 3.17: Sequence Diagram Login

14. Petunjuk Ujian



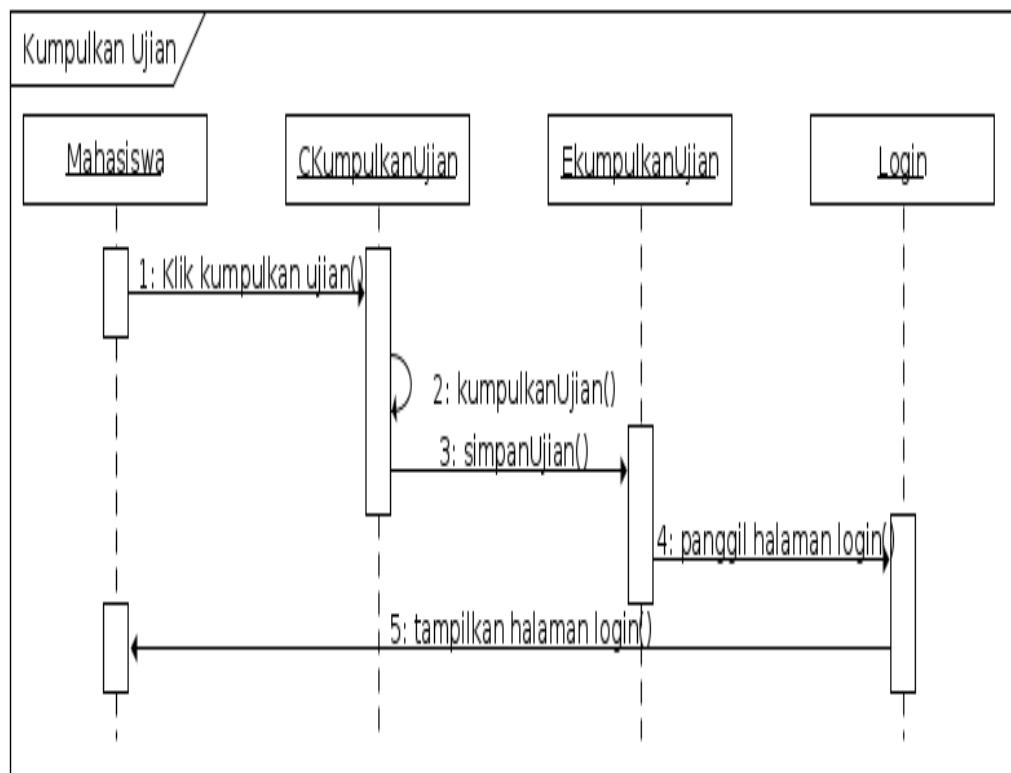
Gambar 3.18: Sequence Diagram Petunjuk Ujian

15. Halaman Ujian



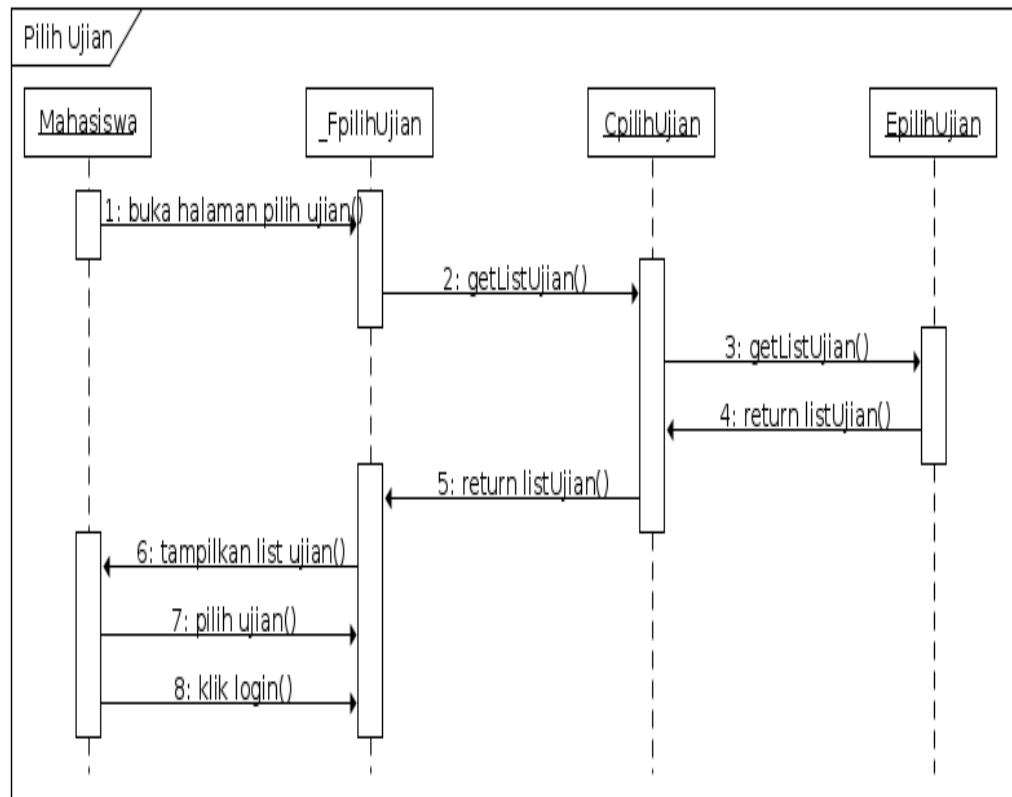
Gambar 3.19: Sequence Diagram Halaman Ujian

16. Kumpulkan Ujian



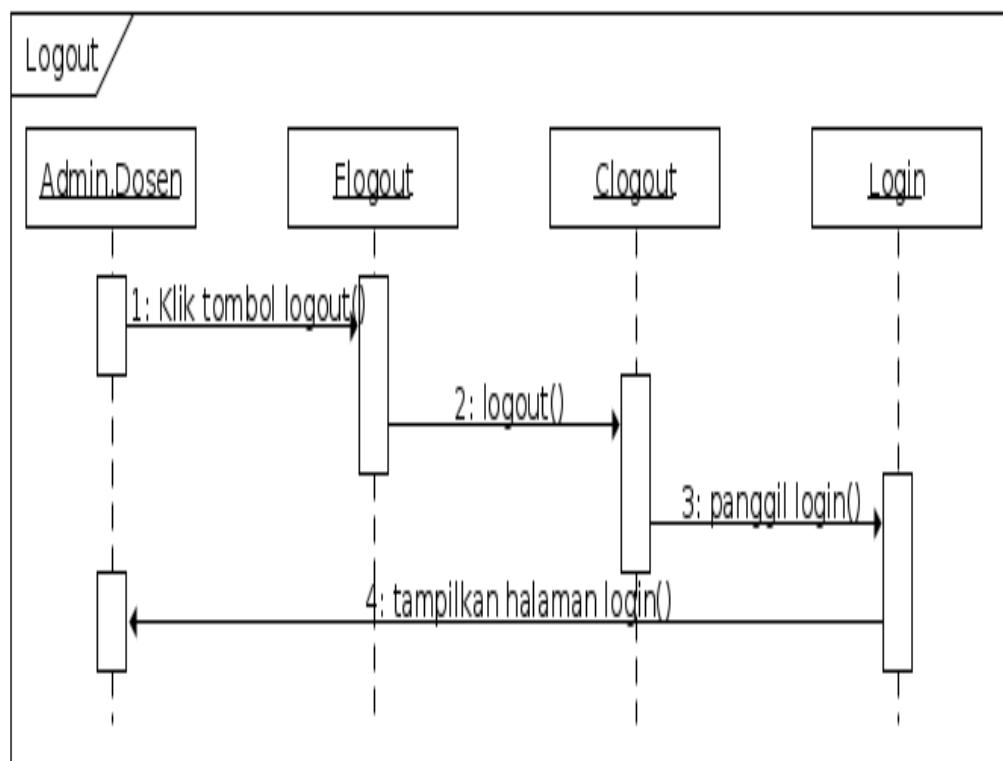
Gambar 3.20: Sequence Diagram Kumpulkan Ujian

17. Pilih Ujian



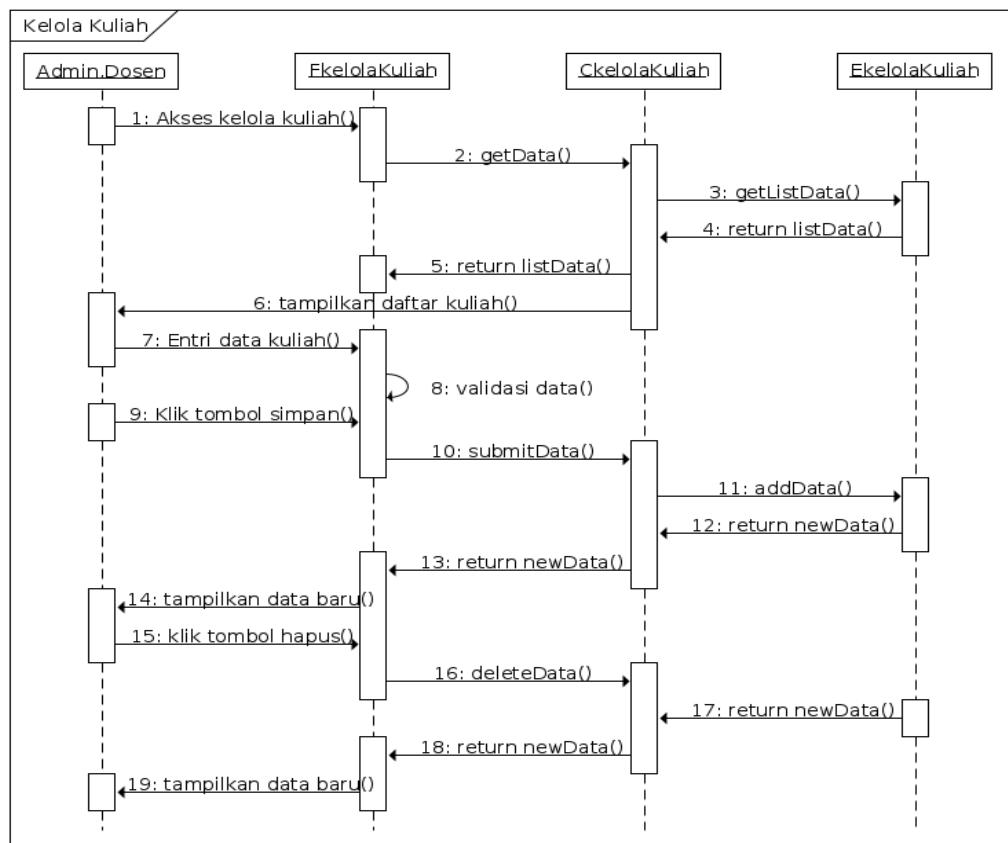
Gambar 3.21: Sequence Diagram Pilih Ujian

18. Logout



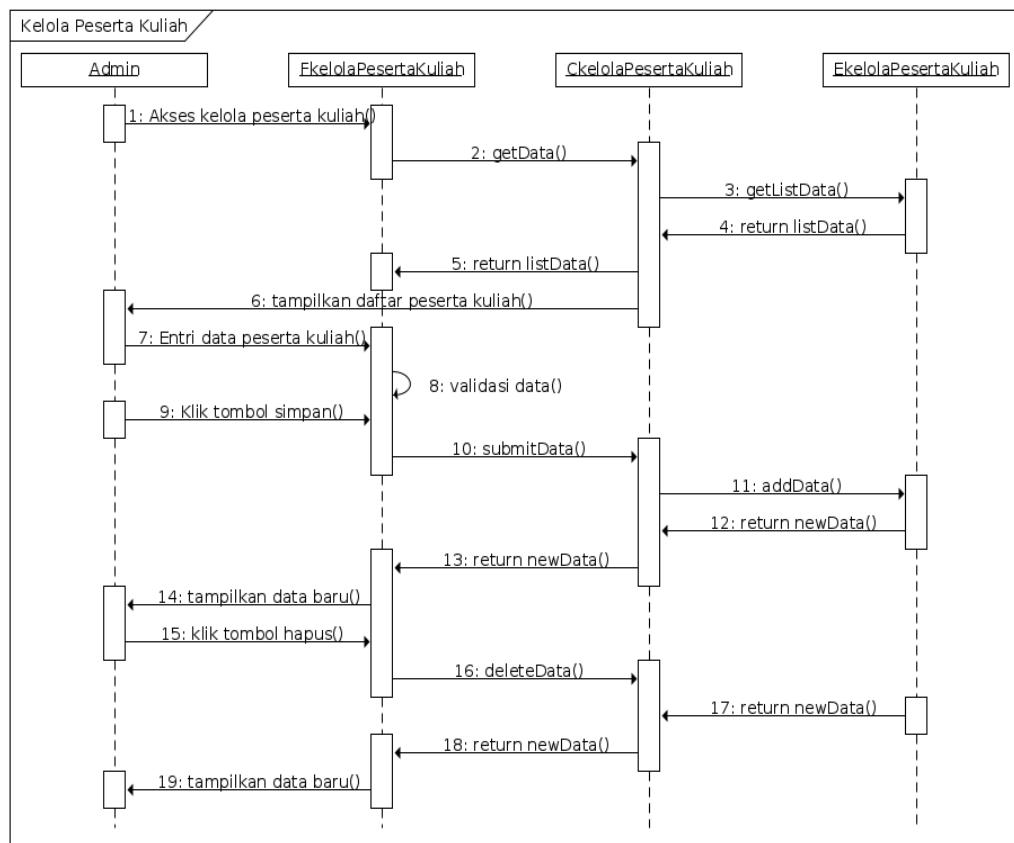
Gambar 3.22: *Sequence Diagram Logout*

19. Kelola Kuliah



Gambar 3.23: Sequence Diagram Kelola Kuliah

20. Kelola Peserta Kuliah



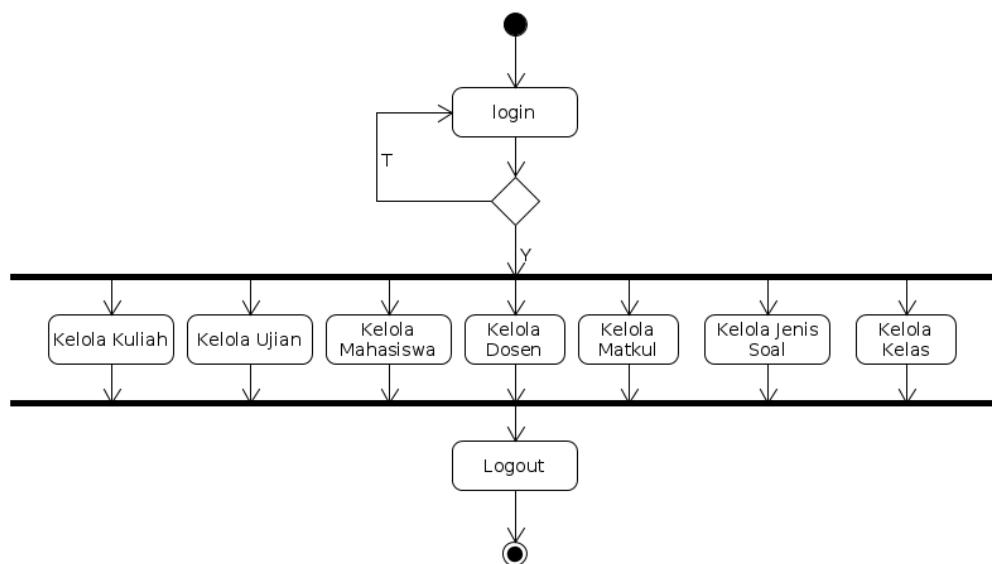
Gambar 3.24: Sequence Diagram Kelola Peserta Kuliah

3.3.4 Activity Diagram

Sesuai dengan namanya, *activity diagram* menggambarkan aktivitas dari seorang aktor terhadap sebuah sistem. Aktifitas tersebut biasanya berupa *use case* yang sudah dibahas sebelumnya. Berikut adalah *activity diagram* dari setiap aktor :

1. *Activity Diagram Admin*

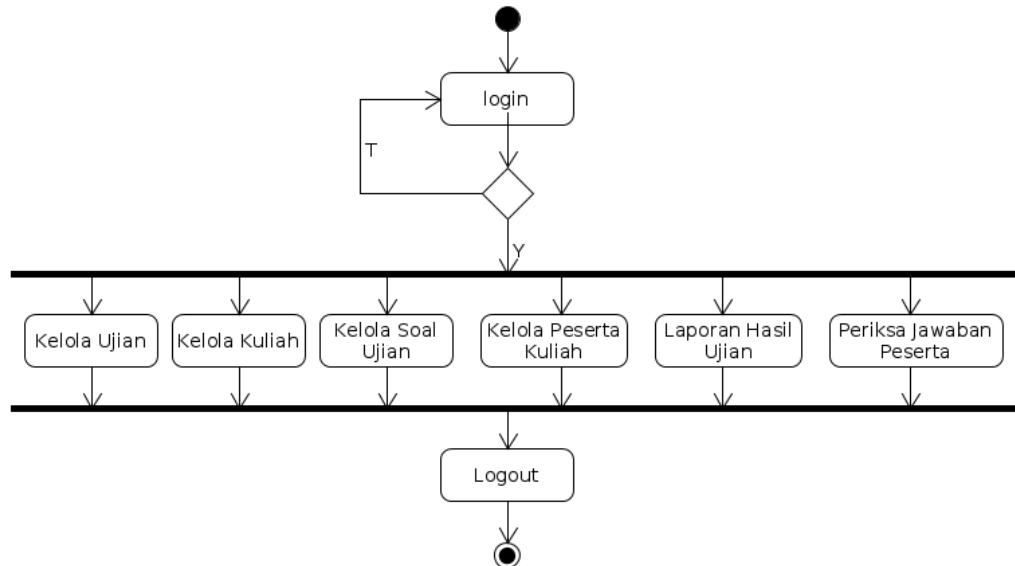
Seorang admin biasanya mengelola data utama dari sebuah sistem. Data utama tersebut nantinya akan digunakan oleh aktor lainnya. Berikut adalah gambaran *activity diagram* seorang admin :



Gambar 3.25: *Activity Diagram Admin*

2. *Activity Diagram Dosen*

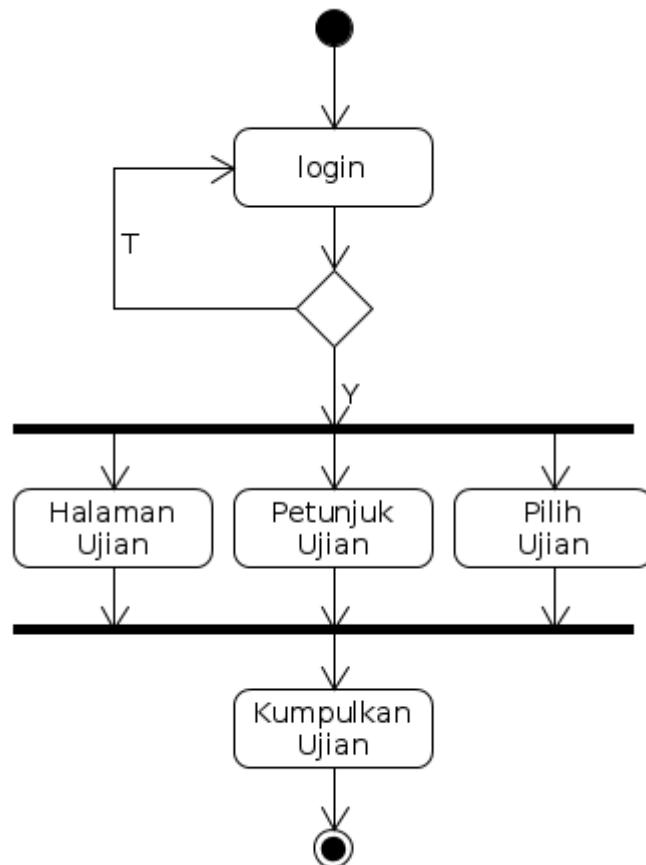
Pada penelitian ini, dosen nantinya bertugas mengurus segala sesuatu yang dibutuhkan untuk mengadakan sebuah ujian. Seorang dosen harus mendaftarkan kuliah yang akan diujikan terlebih dahulu pada sistem dan kemudian soal, peserta serta hasil ujian akan ditentukan oleh dosen. Berikut adalah gambaran *activity diagram* seorang dosen :



Gambar 3.26: Activity Diagram Dosen

3. Activity Diagram Mahasiswa

Mahasiswa merupakan aktor yang akan mengikuti ujian yang telah dibuat oleh seorang dosen. Berikut adalah gambaran *activity diagram* seorang mahasiswa



Gambar 3.27: Activity Diagram Mahasiswa

3.4 Desain Terinci

Desain terinci merupakan gambaran lebih lanjut dari sebuah sistem. Desain terinci terdiri atas desain *input*, desain *output* dan desain *file*.

3.4.1 Desain Ouput

Desain *output* merupakan desain antarmuka aplikasi yang dikhususkan untuk menampilkan informasi dari sistem. Berikut adalah desain *output* dari aplikasi *computer assisted test* berupa halaman web yang akan dikembangkan :

1. Desain Halaman Kelola Ujian

Halaman ini menampilkan daftar ujian yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola ujian :

Header						
Menu	Kelola Ujian					
	Tambah Data		Refresh		Cari	
No	Nama Matkul	Kelas	Waktu Ujian	TA	Aksi	
1	Z	Z	Z	Z	Z	Z

Gambar 3.28: Desain Halaman Kelola Ujian

2. Desain Halaman Kelola Mahasiswa

Halaman ini menampilkan daftar mahasiswa yang terdaftar. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola ujian :

Header				
Menu	Kelola Mahasiswa			
	<input type="button" value="Tambah Data"/>	<input type="button" value="Refresh"/>	<input type="text"/>	<input type="button" value="Cari"/> <input type="button" value="X"/>
No	Nama	NOBP	Status	Aksi
Z	Z	Z	Z	Z

Gambar 3.29: Desain Halaman Kelola Mahasiswa

3. Desain Halaman Kelola Dosen

Halaman ini menampilkan daftar dosen yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola dosen :

Header				
Menu	Kelola Dosen			
	<input type="button" value="Tambah Data"/>	<input type="button" value="Refresh"/>	<input type="text"/>	<input type="button" value="Cari"/> <input type="button" value="X"/>
No	NIDN	Nama Dosen	Aksi	

Gambar 3.30: Desain Halaman Kelola Dosen

4. Desain Halaman Kelola Matkul

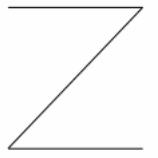
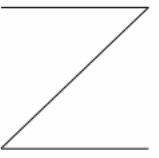
Halaman ini menampilkan daftar matkul yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola matkul :

Header				
Menu	Kelola Matkul			
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>		<input type="text"/>	<input type="button" value="Cari"/> <input type="button" value="X"/>
No	Kode Matkul	Nama Matkul	Aksi	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Gambar 3.31: Desain Halaman Kelola Matkul

5. Desain Halaman Kelola Kelas

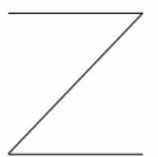
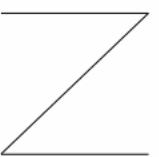
Halaman ini menampilkan daftar kelas yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola kelas :

Header		
Menu	Kelola Kelas	
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="X"/>	
No	Nama Kelas	Aksi
		

Gambar 3.32: Desain Halaman Kelola Kelas

6. Desain Halaman Kelola Jenis Soal

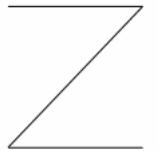
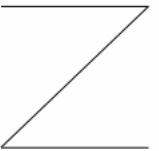
Halaman ini menampilkan daftar kelas yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola kelas :

Header		
Menu	Kelola Jenis Soal	
<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="X"/>		
No	Nama Jenis Ujian	Aksi
		

Gambar 3.33: Desain Halaman Kelola Jenis Soal

7. Desain Halaman Kelola Jenis Ujian

Halaman ini menampilkan daftar jenis ujian yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola jenis ujian :

Header		
Menu	Kelola Jenis Ujian	
<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="X"/>		
No	Nama Jenis Ujian	Aksi
		

Gambar 3.34: Desain Halaman Kelola Jenis Ujian

8. Desain Halaman Kelola Soal Ujian

Halaman ini menampilkan daftar soal ujian yang telah dibuat. Pengguna dapat melakukan aksi seperti menambah data baru, mengedit data dan menghapus data melalui halaman ini. Berikut adalah gambar desain halaman kelola soal ujian :

Header				
Menu	Detail Ujian			
	Nama Matkul Kelas Hari Jenis Ujian		Nama Dosen Tahun Akademik Waktu Jenis Soal	
<input type="button" value="Soal Ujian"/>		<input type="button" value="Hasil Ujian"/>		
<input type="button" value="Tambah Data"/>		<input type="button" value="Refresh"/>	<input type="text"/>	<input type="button" value="Cari"/>
<input type="button" value="X"/>				
No	Isi Soal	Jawaban	Bobot Soal	Aksi
Z	Z	Z	Z	Z

Gambar 3.35: Desain Halaman Kelola Soal Ujian

9. Desain Halaman Hasil Ujian

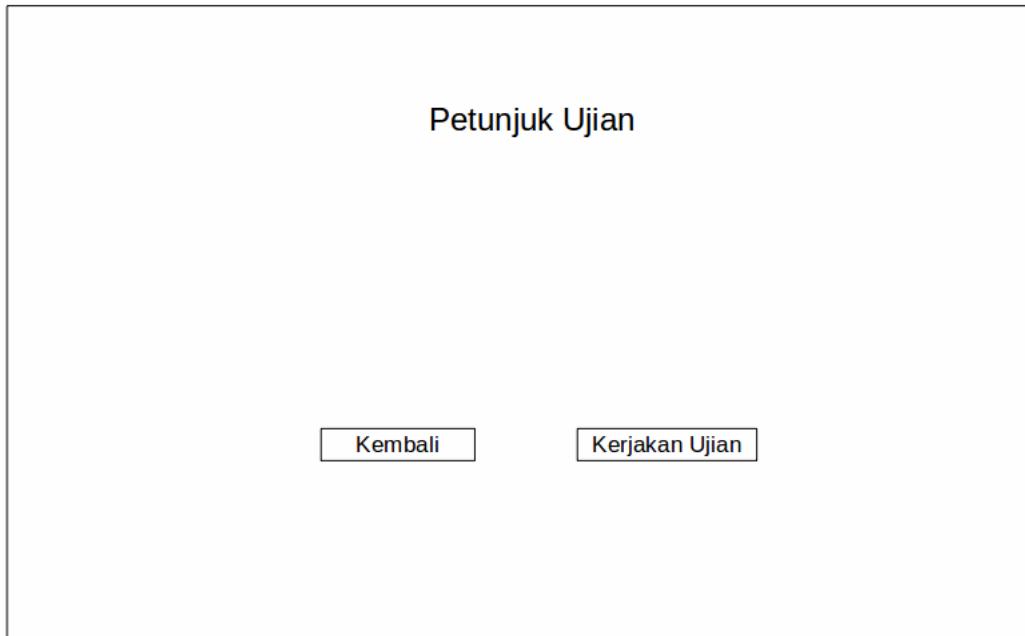
Halaman ini menampilkan daftar hasil ujian dari peserta yang ditampilkan per ujian. Berikut adalah gambar desain halaman hasil ujian :

Header																	
Menu	Detail Ujian																
	Nama Matkul Kelas Hari Jenis Ujian		Nama Dosen Tahun Akademik Waktu Jenis Soal														
	<input type="button" value="Soal Ujian"/>		<input type="button" value="Hasil Ujian"/>														
	<div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <input type="button" value="Refresh"/> </div> <div style="flex: 1;"> <input type="text"/> </div> <div style="flex: 1;"> <input type="button" value="Cari"/> </div> <div style="flex: 1;"> <input type="button" value="X"/> </div> </div>																
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>NOBP</th> <th>Nama Mahasiswa</th> <th>Kelas</th> <th>Nilai</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>Z</td> <td>Z</td> <td>Z</td> <td>Z</td> <td>Z</td> <td>Z</td> </tr> </tbody> </table>					No	NOBP	Nama Mahasiswa	Kelas	Nilai	Aksi	Z	Z	Z	Z	Z	Z
No	NOBP	Nama Mahasiswa	Kelas	Nilai	Aksi												
Z	Z	Z	Z	Z	Z												

Gambar 3.36: Desain Halaman Hasil Ujian

10. Desain Halaman Petunjuk Ujian

Halaman ini menampilkan tentang cara penggunaan aplikasi *CAT* yang sedang dikembangkan. Berikut adalah gambar desain halaman petunjuk ujian :



Gambar 3.37: Desain Halaman Petunjuk Ujian

11. Desain Halaman Kelola Kuliah

Halaman ini menampilkan seluruh kuliah yang diampu semua dosen. Berikut adalah desain halaman kelola kuliah

Header				
Menu	Kelola Kuliah			
		<input type="text"/> Tambah Data <input type="button" value="Refresh"/>		<input type="text"/> Cari <input type="button" value="X"/>
No	Nama Matkul	Kelas	Aksi	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Gambar 3.38: Desain Halaman Kelola Kuliah

12. Desain Halaman Kelola Peserta Kuliah

Halaman ini menampilkan seluruh peserta kuliah yang ada di sebuah kuliah.

Berikut adalah desain halaman kelola kuliah

Header														
Menu	Detail Kuliah													
	<p>Daftar Peserta Kuliah</p> <p><input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="text"/> <input type="button" value="Cari"/> <input type="button" value="X"/></p> <table border="1"><thead><tr><th>No</th><th>NOBP</th><th>Nama Mahasiswa</th><th>Kelas</th><th>Aksi</th></tr></thead><tbody><tr><td>Z</td><td>Z</td><td>Z</td><td>Z</td><td>Z</td></tr></tbody></table>				No	NOBP	Nama Mahasiswa	Kelas	Aksi	Z	Z	Z	Z	Z
No	NOBP	Nama Mahasiswa	Kelas	Aksi										
Z	Z	Z	Z	Z										

Gambar 3.39: Desain Halaman Kelola Peserta Kuliah

13. Desain Halaman Kelola User

Halaman ini menampilkan seluruh user yang dapat masuk ke sistem. Berikut adalah desain halaman kelola user :

Header				
Menu	Daftar User			
<input type="button" value="Tambah Data"/>		<input type="button" value="Refresh"/>	<input type="text"/> <input type="button" value="Cari"/> <input type="button" value="X"/>	
No	username	Jenis User	Status user	Aksi
/	/	/	/	/

Gambar 3.40: Desain Halaman Kelola user

3.4.2 Desain Input

Desain *input* menggambarkan tentang cara aplikasi menerima masukan dari pengguna. Berikut adalah desain *input* yang akan dikembangkan :

1. Desain Halaman Input Ujian

Halaman ini digunakan untuk melakukan entri data ujian baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman ujian :

Header	
Menu	Tambah Data
	Pilih Kuliah <input type="text"/> Jenis Ujian <input type="text"/> Jenis Soal Ujian <input type="radio"/> Pilihan Ganda <input type="radio"/> Essai Waktu Ujian <input type="text"/> Jam Mulai <input type="text"/> Jam Berakhir <input type="text"/> Keterangan Ujian <input type="text"/> Simpan Batal Reset

Gambar 3.41: Desain Halaman Input Ujian

2. Desain Halaman Input Dosen

Halaman ini digunakan untuk melakukan entri data dosen baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman dosen :

Header	
Menu	<p>Tambah Data</p> <p>NIDN</p> <input type="text"/> <p>Nama Dosen</p> <input type="text"/> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Reset"/></p>

Gambar 3.42: Desain Halaman Input Dosen

3. Desain Halaman Input Mahasiswa

Halaman ini digunakan untuk melakukan entri data mahasiswa baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman mahasiswa :

Header	
Menu	<p>Tambah Data</p> <p>Nama Mahasiswa <input type="text"/></p> <p>NOBP <input type="text"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Reset"/></p>

Gambar 3.43: Desain Halaman Input Mahasiswa

4. Desain Halaman Input Matkul

Halaman ini digunakan untuk melakukan entri data matkul baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman matkul :

Header	
Menu	<p>Tambah Data</p> <p>Nama Matkul <input type="text"/></p> <p>SKS <input type="text"/></p> <p>Semester <input type="text"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Reset"/></p>

Gambar 3.44: Desain Halaman Input Matkul

5. Desain Halaman Input Kelas

Halaman ini digunakan untuk melakukan entri data kelas baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman kelas :

Header	
Menu	<p>Tambah Data</p> <p>Nama Kelas</p> <input type="text"/> <p>Simpan Batal Reset</p>

Gambar 3.45: Desain Halaman Input Kelas

6. Desain Halaman Input Jenis Soal

Halaman ini digunakan untuk melakukan entri data jenis soal baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman jenis soal :

Header	
Menu	<p>Tambah Data</p> <p>Nama Tipe Soal</p> <input type="text"/> <p>Simpan Batal Reset</p>

Gambar 3.46: Desain Halaman Input Jenis Soal

7. Desain Halaman Input Jenis Ujian

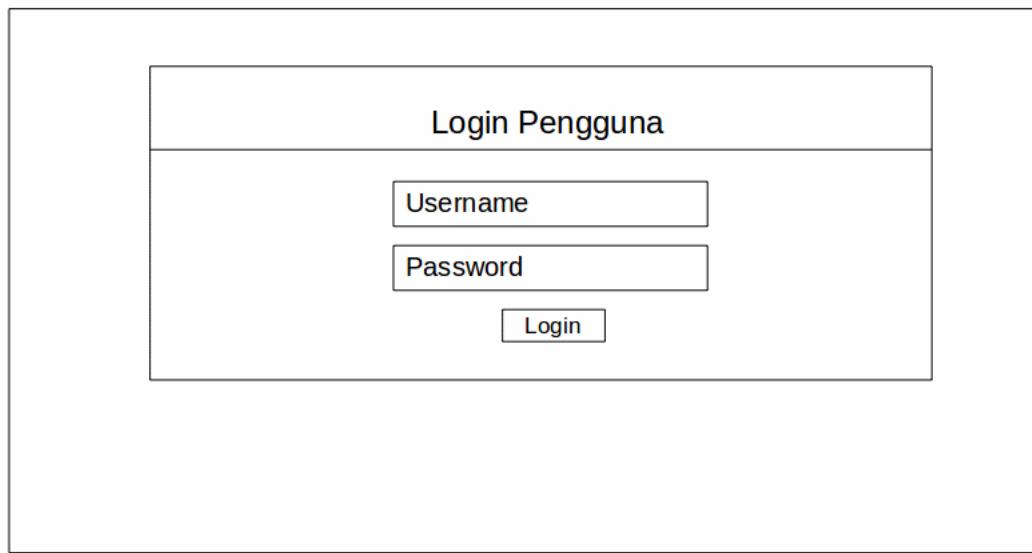
Halaman ini digunakan untuk melakukan entri data jenis ujian baru. Sebelum disimpan ke basis data, data yang dimasukkan dicek terlebih dahulu apakah sudah sesuai dengan kriteria yang ditentukan atau belum. Berikut adalah gambar desain halaman jenis ujian :

Header	
Menu	<p>Tambah Data</p> <p>Nama Jenis Ujian</p> <input type="text"/> <p>Simpan Batal Reset</p>

Gambar 3.47: Desain Halaman Input Jenis Ujian

8. Desain Halaman Login

Halaman ini digunakan bagi pengguna yang ingin masuk ke sistem. Berikut adalah desain halaman login :



The diagram shows a wireframe of a login page. At the top, a header bar contains the text "Login Pengguna". Below the header are two rectangular input fields: the top one is labeled "Username" and the bottom one is labeled "Password". At the bottom of the page is a single rectangular button labeled "Login". The entire form is enclosed within a large rectangular frame.

Gambar 3.48: Desain Halaman Login

9. Desain Halaman Ujian

Halaman ini digunakan oleh mahasiswa yang ingin melaksanakan ujian. Mahasiswa dapat menjawab jawaban secara langsung pada aplikasi dan jawaban akan tersimpan kedalam sistem. Berikut adalah desain halaman ujian :

Header											
Pilih Soal <input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> Info Ujian Nama Ujian Hari Waktu <input type="button" value="Kumpulkan Ujian"/>	<table border="1"> <tr> <td>Dikerjakan</td> <td>Total Soal</td> </tr> <tr> <td>Belum Dikerjakan</td> <td>Sisa Waktu</td> </tr> </table> <p>1. Isi soal ujian</p> <table border="1"> <tr> <td colspan="2">Jawaban Peserta</td> </tr> <tr> <td colspan="2">Simpan Jawaban</td> </tr> <tr> <td style="text-align: center;"><--></td> <td style="text-align: center;">--></td> </tr> </table>	Dikerjakan	Total Soal	Belum Dikerjakan	Sisa Waktu	Jawaban Peserta		Simpan Jawaban		<-->	-->
Dikerjakan	Total Soal										
Belum Dikerjakan	Sisa Waktu										
Jawaban Peserta											
Simpan Jawaban											
<-->	-->										

Gambar 3.49: Desain Halaman Ujian

10. Desain Halaman Pilih Ujian

Halaman ini digunakan oleh mahasiswa pada saat ingin mengikuti ujian.

Mahasiswa diminta untuk memilih ujian yang akan diikuti. Berikut adalah desain halaman pilih ujian :

Pilih Ujian
<input type="text"/>
<input type="button" value="Login"/>

Gambar 3.50: Desain Halaman Pilih Ujian

11. Desain Halaman Periksa Jawaban

Halaman ini digunakan oleh dosen untuk memeriksa jawaban mahasiswa dengan tipe soal essai. Berikut adalah desain halaman periksa jawaban:

Header		
Menu	Nilai total	Soal
	Pilih Soal	Jawaban Dosen
	Nilai Bobot	Jawaban Peserta
	<input type="text"/> <input type="button" value="Simpan"/>	<input type="button" value="Simpan Hasil Periksa"/>

Gambar 3.51 : Desain Halaman Periksa Ujian

12. Desain Halaman Input User

Halaman ini digunakan untuk menambahkan pengguna sistem baru. Berikut adalah desain halaman kelola user :

Header	
Menu	
	<p>Tambah Data</p> <p>Username</p> <input type="text"/> <p>Password</p> <input type="text"/> <p>Jenis user</p> <input type="text"/> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Reset"/></p>

Gambar 3.52: Desain Halaman Input user

13. Desain Halaman Input Kuliah

Halaman ini digunakan untuk menambahkan kuliah baru seiring dengan tahun akademik baru. Berikut adalah desain halaman kelola kuliah :

Header	
Menu	<p>Tambah Data</p> <p>Pilih Matkul</p> <input type="text"/> <p>Pilih Kelas</p> <input type="text"/> <p>Tahun Akademik</p> <input type="text"/> <p><input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Reset"/></p>

Gambar 3.53: Desain Halaman Input Kuliah

14. Desain Halaman Input Peserta Kuliah

Halaman ini digunakan untuk menambahkan mahasiswa kedalam sebuah kuliah. Berikut adalah desain halaman kelola peserta kuliah :

Header	
Menu	Tambah Data Pilih Kelas <input type="text"/> Pilih Mahasiswa <input type="text"/> <input type="button" value="Simpan"/> <input type="button" value="Batal"/> <input type="button" value="Reset"/>

Gambar 3.54: Desain Halaman Input Peserta Kuliah

3.4.3 Desain File

File merupakan tabel yang digunakan pada sebuah basis data. Berikut adalah daftar tabel yang digunakan oleh aplikasi yang sedang dikembangkan :

1. File status

File ini digunakan untuk menampung keterangan dari status yang digunakan pada file lain. Berikut adalah desain file status :

Tabel 3.22: Desain File Status

Nama Tabel	:	tbstatus	
Primary Key	:	id_status	
Nama Kolom	Tipe Data	Panjang Data	Deskripsi

id_status	integer	11	Id soal ujian
status	varchar	20	Nama status

2. File kelas kuliah

File ini digunakan untuk menampung daftar kelas yang berada pada sebuah kuliah. Berikut adalah tabel file kelas kuliah :

Tabel 3.23: Desain File Kelas Kuliah

Nama Tabel : tbkelas_kuliah
Primary Key : id_kkuliah

Nama Kolom	Tipe Data	Panjang Data	Deskripsi
id_kkuliah	integer	11	Id kelas kuliah
id_kuliah	varchar	40	Id kuliah
id_kelas	integer	11	Id kelas

3. File kuliah

File ini digunakan untuk menampung semua daftar kuliah. Berikut adalah desain file kuliah :

Tabel 3.24: Desain File Kuliah

Nama Tabel : tbkuliah
Primary Key : id_kuliah

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_kuliah	varchar	40	Id kuliah
nidn	varchar	11	NIDN Dosen
kd_matkul	varchar	15	Kode matkul
tahun_akademik	varchar	5	Tahun akademik

4. File peserta kuliah

File ini digunakan untuk menampung daftar mahasiswa yang mengikuti sebuah kuliah. Berikut adalah desain file peserta kuliah :

Tabel 3.25: Desain File Peserta Kuliah

Nama Tabel : tbpeserta_kuliah

Primary Key : id_peserta

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_peserta	varchar	60	Id peserta
nobp	varchar	15	NOBP mahasiswa
id_kuliah	varchar	40	Id kuliah
status_peserta	smallint	3	Status peserta
id_kelas	integer	11	Id kelas

5. File hasil ujian

File ini digunakan untuk menampung daftar mahasiswa serta nilai ujian mereka. Berikut adalah desain file hasil ujian :

Tabel 3.26: Desain File Hasil Ujian

Nama Tabel : tbhasil_ujian

Primary Key : id_hasil

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_hasil	integer	11	Id hasil
id_ujian	varchar	40	Id ujian
nobp	varchar	15	NOBP Mahasiswa
nilai	smallint	3	Nilai peserta

6. File jawaban

File ini digunakan untuk menampung jawaban ujian peserta yang bertipe soal essai. Berikut adalah desain file jawaban :

Tabel 3.27: Desain File Jawaban

Nama Tabel : tbjawaban
Primary Key : id_jawaban

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_jawaban	integer	11	Id jawaban
id_ujian	varchar	60	Id ujian
nobp	varchar	15	NOBP mahasiswa
jawaban	text	-	Jawaban peserta
id_soal	integer	11	Id soal

7. File soal

File ini digunakan untuk menampung soal ujian. Berikut adalah desain file soal :

Tabel 3.28: Desain File Soal

Nama Tabel : tbsoal
Primary Key : id_soal

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_soal	integer	11	Id soal
isi_soal	text	-	Isi soal
pilihanGanda	JSON	-	Isi pilihan ganda
id_jsoal	smallint	3	Id jenis soal
bobot	smallint	3	Bobot soal
jawaban	text	-	Jawaban soal

8. File kelas

File ini digunakan untuk menampung daftar nama kelas yang bisa digunakan pada kuliah. Berikut adalah desain file kelas :

Tabel 3.29: Desain File Kelas

Nama Tabel : tbkelas

Primary Key : id_kelas

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_kelas	integer	11	Id kelas
nm_kelas	varchar	10	Nama kelas

9. File jenis soal

File ini digunakan untuk menampung nama jenis soal. Berikut adalah desain file jenis soal :

Tabel 3.30: Desain File Jenis Soal

Nama Tabel : tbjenis_soal

Primary Key : id_jsoal

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_jsoal	integer	11	Id jsoal
nm_jsoal	varchar	20	Nama jenis soal

10. File jenis ujian

File ini digunakan untuk menampung nama jenis ujian. Berikut adalah desain file jenis ujian :

Tabel 3.31: Desain File Jenis Ujian

Nama Tabel : tbjenis_ujian

Primary Key : id_jujian

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_jujian	integer	11	Id jujian
nm_jujian	varchar	10	Nama jenis ujian

11. File jenis user

File ini digunakan untuk menampung nama jenis user. Berikut adalah desain file jenis user :

Tabel 3.32: Desain File Jenis User

Nama Tabel : tbjenis_user
Primary Key : id_juser

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_juser	integer	11	Id juser
nm_juser	varchar	20	Nama jenis user

12. File user

File ini digunakan untuk menampung daftar user. Berikut adalah desain file user :

Tabel 3.33: Desain File User

Nama Tabel : tbuser
Primary Key : id_user

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_user	integer	11	Id user
username	varchar	30	Username
password	text	-	Kata sandi
id_juser	smallint	3	Id jenis user
status_user	smallint	3	Status user

13. File ujian

File ini digunakan untuk menampung daftar ujian. Berikut adalah desain file ujian :

Tabel 3.34: Desain File Ujian

Nama Tabel : tbujian
Primary Key : id_ujian

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_ujian	varchar	40	Id ujian
hari	date	-	Hari ujian
mulai	time	-	Mulai ujian
selesai	time	-	Selesai ujian
deskripsi	varchar	100	Deskripsi ujian
status_ujian	smallint	3	Status ujian
id_jujian	smallint	3	Id jenis ujian
id_jsoal	smallint	3	Id jenis soal
id_kuliah	varchar	40	Id kuliah

14. File mahasiswa

File ini digunakan untuk menampung seluruh data mahasiswa. Berikut adalah desain file mahasiswa :

Tabel 3.35: Desain File Mahasiswa

Nama Tabel : tbmahasiswa
Primary Key : id_mahasiswa

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_mahasiswa	integer	11	Id mahasiswa
nobp	varchar	15	NOBP
nm_mahasiswa	varchar	100	Nama mahasiswa
status_mahasiswa	smallint	3	Status mahasiswa

15. File dosen

File ini digunakan untuk menampung seluruh data dosen. Berikut adalah desain file dosen :

Tabel 3.36: Desain File Dosen

Nama Tabel : tb_dosen

Primary Key : id_dosen

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_dosen	integer	11	Id dosen
nidn	varchar	15	NIDN
nm_dosen	varchar	100	Nama dosen
status_dosen	smallint	3	Status dosen

16. File matkul

File ini digunakan untuk menampung seluruh data matkul. Berikut adalah desain file matkul :

Tabel 3.37: Desain File Matkul

Nama Tabel : tb_matkul

Primary Key : id_matkul

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_matkul	integer	11	Id matkul
kd_matkul	varchar	15	Kode matkul
nm_matkul	varchar	100	Nama matkul
sks	smallint	3	Banyak SKS
smt	smallint	3	Semester
status_matkul	smallint	3	Status matkul

17. File soal ujian

File ini digunakan untuk menampung data soal dan ujian serta kaitannya.

Berikut adalah desain file soal ujian :

Tabel 3.38: Desain File Soal Ujian

Nama Tabel : tbsoal_ujian

Primary Key : id_sujian

Nama Kolom	Tipe Data	Panjang data	Deskripsi
id_sujian	integer	11	Id soal ujian
id_ujian	varchar	40	Id ujian
id_soal	integer	11	Id soal

BAB IV

IMPLEMENTASI

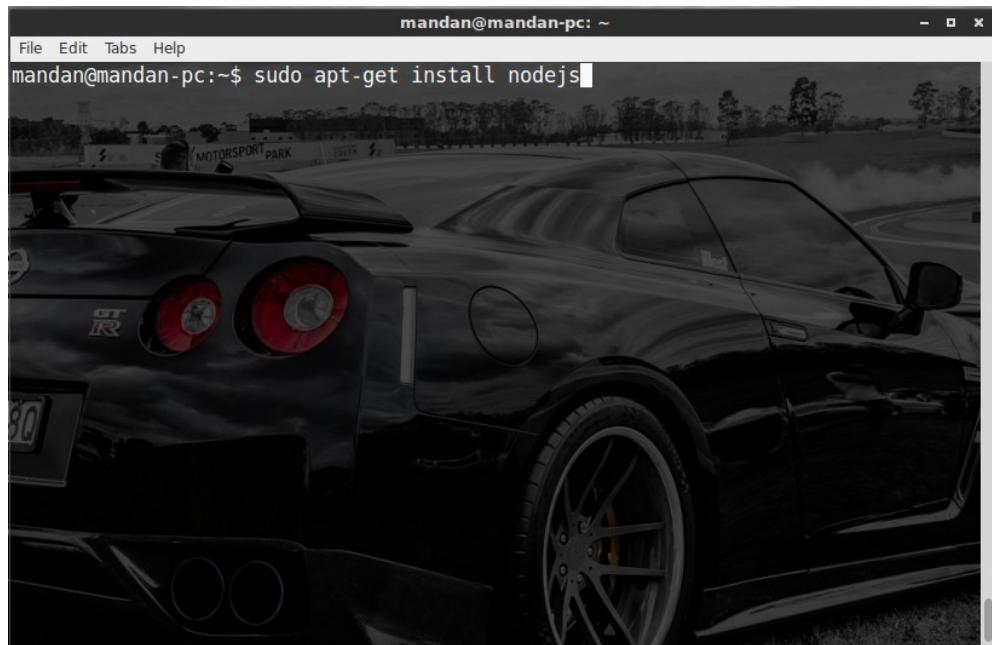
4.1 Pemasangan

Pada sistem ini, digunakan bahasa pemrograman *javascript* dengan *nodejs*, basis data *postgresql* dan *text editor* bernama *geany*. Semua aplikasi tersebut dijalankan pada sistem operasi *ubuntu 16.04*. Berikut akan dijelaskan cara pemasangan aplikasi-aplikasi tersebut :

4.1.1 Pemasangan *NodeJS*

NodeJS merupakan perangkat lunak yang menjadikan bahasa pemrograman *javascript* bisa berjalan pada sisi *server* layaknya *PHP*. Berikut adalah langkah pemasangan *NodeJS* :

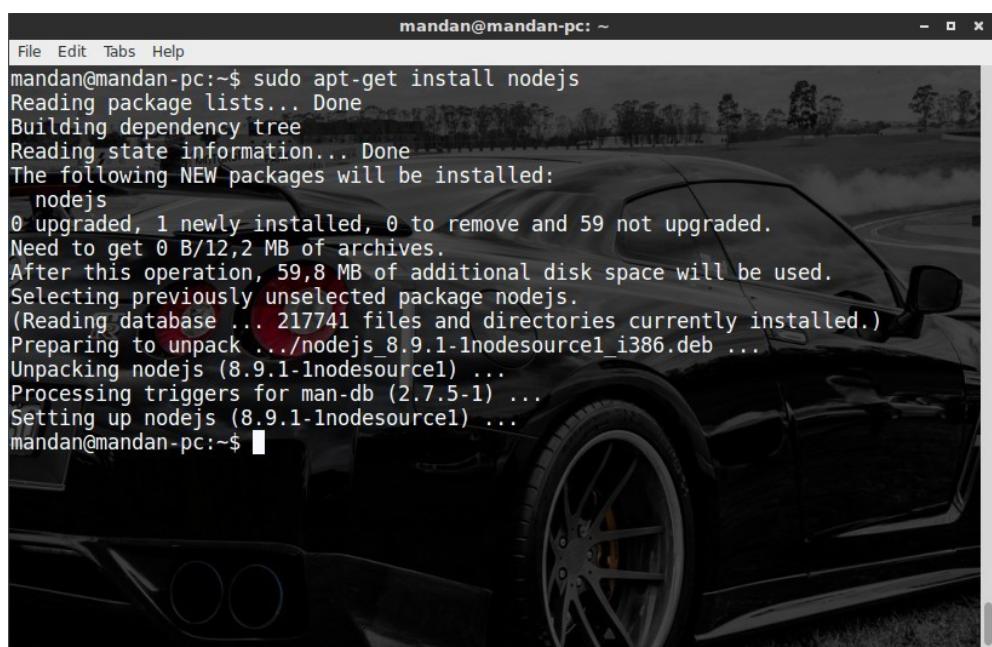
1. Pada monitor, buka *terminal* dengan cara menekan tombol “*ctrl + alt + t*”. Jika *terminal* terbuka, ketik “*sudo apt-get install nodejs*”. Berikut detail gambar dari langkah pertama :



Gambar 4.1: Tampilan awal terminal

2. Kemudian tekan “enter” dan proses pemasangan akan berlangsung.

Berikut detail gambar proses pemasangan :



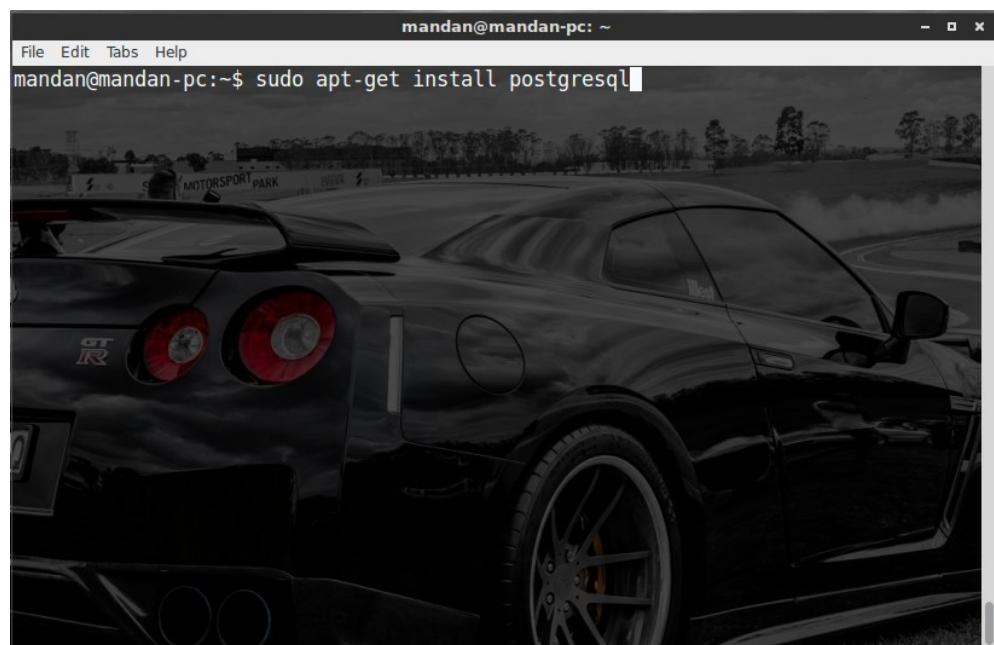
Gambar 4.2: Tampilan proses pemasangan NodeJS

4.1.2 Pemasangan *Postgresql*

Postgresql merupakan perangkat lunak yang bertugas mengatur basis data.

Berikut adalah langkah pemasangan *postgresql* :

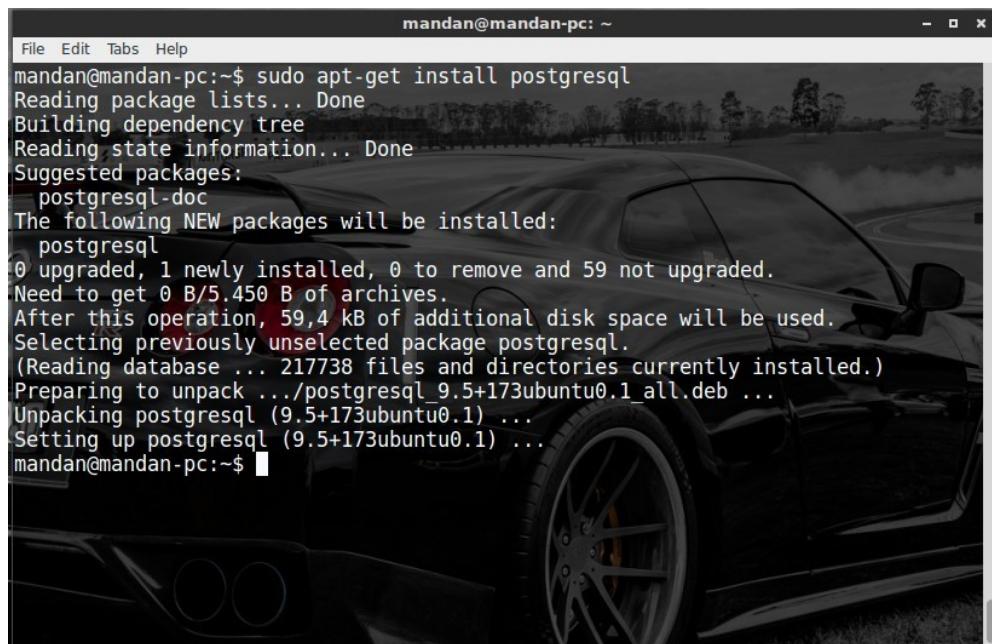
1. Buka terminal dan ketik “*sudo apt-get install postgresql*”. Berikut adalah gambar detail langkah pertama :



Gambar 4.3: Langkah pertama pemasangan *postgresql*

2. Kemudian tekan “*enter*” dan proses pemasangan akan berlangsung.

Berikut detail gambar pemasangannya :

A screenshot of a terminal window titled "mandan@mandan-pc: ~". The window shows the command "sudo apt-get install postgresql" being run and its output. The output includes: "Reading package lists... Done", "Building dependency tree", "Reading state information... Done", "Suggested packages: postgresql-doc", "The following NEW packages will be installed: postgresql", "0 upgraded, 1 newly installed, 0 to remove and 59 not upgraded.", "Need to get 0 B/5.450 B of archives.", "After this operation, 59,4 kB of additional disk space will be used.", "Selecting previously unselected package postgresql.", "(Reading database ... 217738 files and directories currently installed.)", "Preparing to unpack .../postgresql_9.5+173ubuntu0.1_all.deb ...", "Unpacking postgresql (9.5+173ubuntu0.1) ...", "Setting up postgresql (9.5+173ubuntu0.1) ...". The terminal prompt "mandan@mandan-pc:~\$" is visible at the bottom.

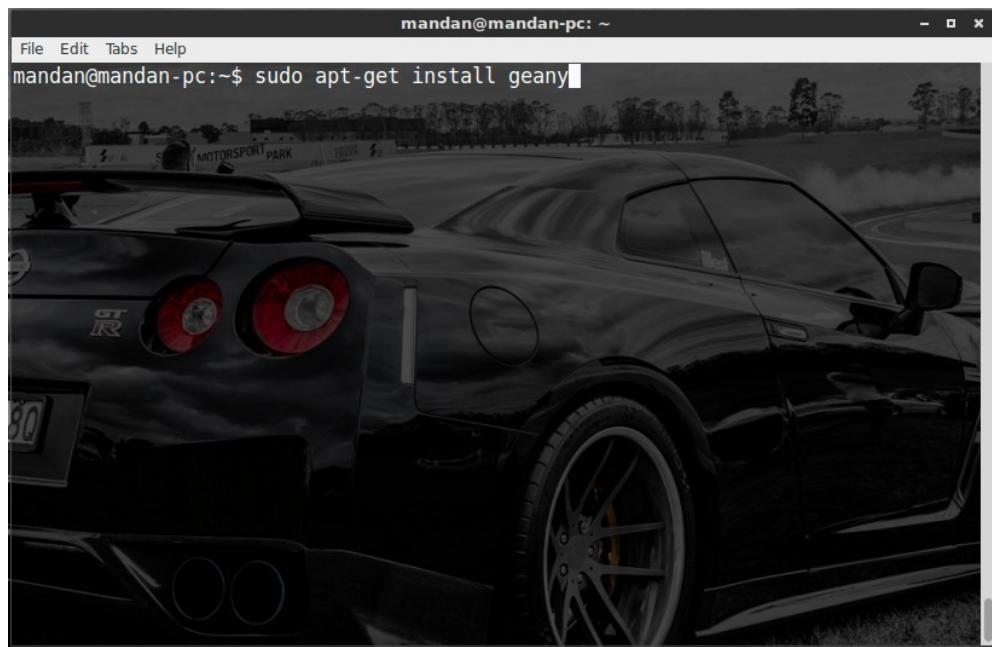
mandan@mandan-pc:~\$ sudo apt-get install postgresql
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
 postgresql-doc
The following NEW packages will be installed:
 postgresql
0 upgraded, 1 newly installed, 0 to remove and 59 not upgraded.
Need to get 0 B/5.450 B of archives.
After this operation, 59,4 kB of additional disk space will be used.
Selecting previously unselected package postgresql.
(Reading database ... 217738 files and directories currently installed.)
Preparing to unpack .../postgresql_9.5+173ubuntu0.1_all.deb ...
Unpacking postgresql (9.5+173ubuntu0.1) ...
Setting up postgresql (9.5+173ubuntu0.1) ...
mandan@mandan-pc:~\$

Gambar 4.4: Proses pemasangan *postgresql*

4.1.3 Pemasangan *geany*

Geany merupakan *text editor* yang digunakan dalam pembuatan aplikasi *computer assisted test* pada penelitian ini. Berikut adalah proses pemasangannya :

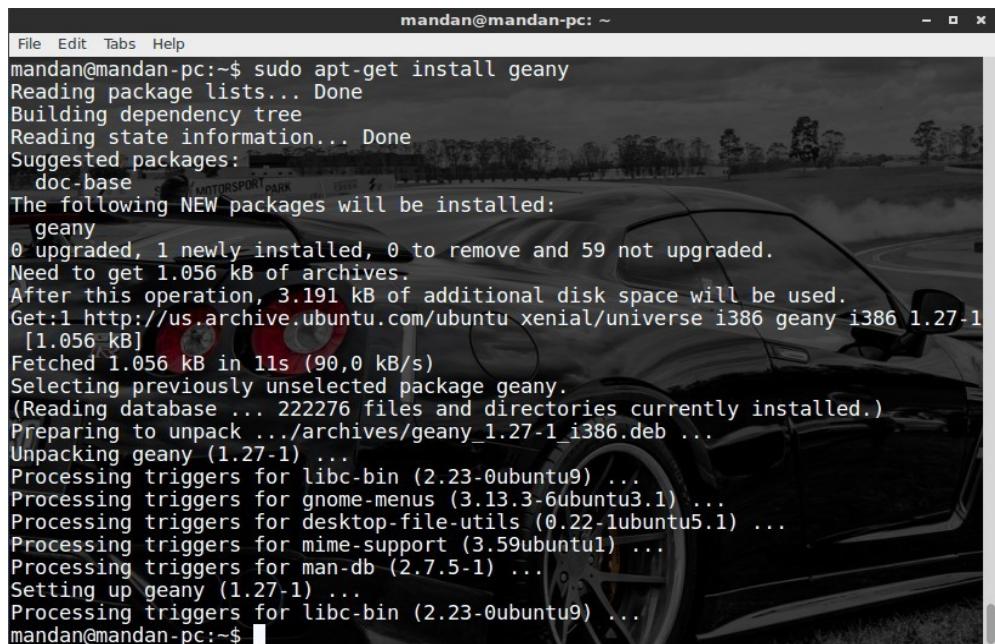
1. Buka *terminal* dan ketik “*sudo apt-get install geany*”. Berikut adalah detail gambar langkah pertama :



Gambar 4.5: Langkah pertama proses pemasangan *geany*

2. Kemudian tekan “*enter*” dan proses pemasangan akan dilakukan.

Berikut gambar detail proses pemasangan :



```
mandan@mandan-pc:~$ sudo apt-get install geany
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  doc-base
The following NEW packages will be installed:
  geany
0 upgraded, 1 newly installed, 0 to remove and 59 not upgraded.
Need to get 1.056 kB of archives.
After this operation, 3.191 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe i386 geany i386 1.27-1
[1.056 kB]
Fetched 1.056 kB in 11s (90,0 kB/s)
Selecting previously unselected package geany.
(Reading database ... 222276 files and directories currently installed.)
Preparing to unpack .../archives/geany_1.27-1_i386.deb ...
Unpacking geany (1.27-1) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
Processing triggers for gnome-menus (3.13.3-6ubuntu3.1) ...
Processing triggers for desktop-file-utils (0.22-1ubuntu5.1) ...
Processing triggers for mime-support (3.59ubuntu1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up geany (1.27-1) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
mandan@mandan-pc:~$
```

Gambar 4.6: Proses pemasangan *geany*