

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

**SKRIPSI**

*Untuk Memenuhi Sebahagian Persyaratan  
Mencapai Gelar Sarjana Komputer*

**Program Studi : Sistem Informasi  
Jenjang Pendidikan : Strata 1**



*Diajukan Oleh :*

**EGO DAFMA DASA  
14101152610565**

**FAKULTAS ILMU KOMPUTER (FILKOM)  
UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG  
2018**

## **LEMBAR PERNYATAAN**

Saya yang bertandatangan di bawah ini :

Nama : Ego Dafma Dasa

Nomor Bp : 14101152610565

Fakultas : Ilmu Komputer

Jurusan : Sistem Informasi

Menyatakan bahwa :

1. Sesungguhnya skripsi yang saya susun ini merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam tugas akhir yang saya peroleh dari hasil karya tulis orang lain, telah saya tuliskan sumbernya dengan jelas, sesuai dengan kaedah penulisan ilmiah.
2. Jika dalam pembuatan skripsi secara keseluruhan ternyata terbukti dibuat oleh orang lain, maka saya bersedia menerima sanksi yang diberikan akademik berupa pembatalan skripsi dan mengulang penelitian serta mengajukan judul baru.

Demikian surat pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari pihak manapun.

Padang, Januari 2018

Penulis

**EGO DAFMA DASA**  
**14101152610565**

## LEMBAR PENGESAHAN PENGUJI SIDANG SKRIPSI

### PERANCANGAN COMPUTER ASSISTED TEST (CAT) MENGGUNAKAN BAHASA PEMROGRAMAN NODE.JS BERBASIS WEB/MOBILE PADA UNIVERSITAS PUTRA INDONESIA "YPTK" PADANG

Oleh :

EGO DAFMA DASA  
14101152610565

PROGRAM STUDI SISTEM INFORMASI

Skripsi ini telah dinyatakan LULUS oleh  
Penguji Materi Pada Sidang Skripsi Program Studi Strata 1 Ilmu Komputer  
Program Studi Sistem Informasi  
Universitas Putra Indonesia "YPTK" Padang  
Pada Hari/Tgl : 01/02/2018

TIM PENGUJI :

1. Hari Marfalino, S. Kom, M. Kom .....  
.....
2. Eva Rianti, S. Kom, M. Kom .....  
.....

Padang, Februari 2018

Mengetahui  
Dekan Fakultas Ilmu Komputer  
Universitas Putra Indonesia (UPI) "YPTK" Padang

(Dr. Julius Santony, S.Kom, M.Kom)  
NIDN: 1029077301

## LEMBARAN PENGESAHAN LULUS SIDANG SKRIPSI

### PERANCANGAN COMPUTER ASSISTED TEST (CAT) MENGGUNAKAN BAHASA PEMROGRAMAN NODE.JS BERBASIS WEB/MOBILE PADA UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG

Yang Dipersiapkan dan Disusun Oleh :

**EGO DAFMA DASA**  
**14101152610565**

Telah Dipertahankan Didepan Dewan Penguji  
Pada Tanggal 1 Februari 2018  
Dinyatakan Telah Lulus Memenuhi Syarat

**Pembimbing I**

**(Silvia Andini, S.Kom, M.Kom)**  
**NIDN: 1003056801**

**Pembimbing II**

**(Novi Trisna S.Kom, M.Kom)**  
**NIDN: 1007119001**

Padang, Februari 2018

**Dekan Fakultas Ilmu Komputer  
Universitas Putra Indonesia (UPI) ”YPTK” Padang**

**(Dr. Julius Santony, S.Kom, M.Kom)**  
**NIDN: 1029077301**

## **KATA PERSEMBAHAN**

***Dialah Allah, tidak ada tuhan selain dia. Mengetahui yang gaib dan yang nyata, Dialah yang maha pengasih, maha penyayang.***

***(QS. Al-Hasyr:22)***

***Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari seesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain) dan hanya kepada tuhanmu lah engkau berharap***

***(QS. Al-Insyirah:6-8)***

**Yang Utama Dari Segalanya...**

Sembah sujud serta syukur kepada Allah SWT. Taburan cinta dan kasih sayang-Mu telah memberikanku kekuatan, membekaliku dengan ilmu serta memperkenalkanku dengan cinta. Dari semua yang telah engkau tetapkan baik itu rencana indah yang engkau siapkan untuk masa depanku sebagai harapan kesuksesan. Atas karunia serta kemudahan yang Engkau berikan akhirnya skripsi yang sederhana ini dapat terselesaikan. Sholawat dan salam selalu terlimpahkan keharibaan

Rasullah Muhammad SAW.

**Kupersembahkan karya sederhana ini kepada orang yang sangat kukasihi dan kusayangi**

Ibunda, Ayahanda, Adikku, dan Keluargaku Tercinta  
Sebagai tanda bakti, hormat, dan rasa terima kasih yang tiada terhingga kupersembahkan karya kecil ini kepada keluargaku yang telah memberikan kasih sayang, segala dukungan, dan cinta kasih yang tiada terhingga yang tiada mungkin dapat kubalas hanya dengan selembar kertas yang bertuliskan kata cinta dan persembahan. Untuk Ibu, Ayah dan Adikku yang selalu membuatku termotivasi dan selalu menyirami kasih sayang, selalu mendoakanku, selalu menasehatiku menjadi lebih baik,

Terima Kasih Ibu.... Ibu.... Ibu.....Terima Kasih Ayah, Terima Kasih Keluargaku semuanya, Terima Kasih Ya Allah yang telah mengirimkan insan terbaik dalam hidupku.

**Dosen Pembimbing Tugas Akhirku...**

Ibu Sifia Andini, S.Kom, M.Kom dan Ibu Novi Trisna S.Kom, M.Kom selaku dosen pembimbing tugas akhir, terima kasih banyak...buk..., sudah begitu banyak membantu selama ini, sudah dinasehati, sudah diajari, bantuan dan kesabaran Ibu akan selalu terukir dihati.

Manisnya keberhasilan akan menghapus pahitnya kesabaran. Nikmatnya memperoleh kemenangan akan menghilangkan letihnya perjuangan menuntaskan pekerjaan. Hidup adalah perjuangan yang harus dimenangkan. Pengalaman akan membawa kita pada kegagalan dan keberhasilan, yang keduanya bersama-sama akan menempah kita untuk terus berkembang dan akhirnya menggapai kesuksesan.

Allah selalu memberikan apa yang kita butuhkan, bukan apa yang kita inginkan, karena apa yang terbaik bagi kita belum tentu baik bagi Allah SWT, namun apa yang baik bagi Allah SWT itulah yang terbaik buat kita. tetapi sering kali kita tidak bisa melihat apa yang kita butuhkan, melainkan selalu melihat apa yang kita inginkan.

## ABSTRAK

<b>JUDUL</b>	<b>: PERANCANGAN COMPUTER ASSISTED TEST (CAT) MENGGUNAKAN BAHASA PEMROGRAMAN NODE.JS DAN DATABASE POSTGRESQL BERBASIS WEB/ MOBILE PADA UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG</b>
<b>NAMA</b>	<b>: EGO DAFMA DASA</b>
<b>NOBP</b>	<b>: 14101152610565</b>
<b>PROGRAM STUDI</b>	<b>: SISTEM INFORMASI</b>
<b>PEMBIMBING</b>	<b>: 1. SILFIA ANDINI, S.KOM., M.KOM. 2. NOVI TRISNA, S. KOM., M.KOM.</b>

Tujuan dari penelitian ini adalah untuk merancang aplikasi computer assisted test pada Universitas Putra Indonesia “YPTK” Padang. Metode yang digunakan pada penelitian ini adalah Penelitian Lapangan (Field research), Penelitian Keperpustakaan (Library Research) dan Penelitian Laboratorium (Laboratory Research). Alat bantu yang digunakan untuk perancangan adalah menggunakan UML (Unified Modelling Language). Hasil dari penelitian ini adalah untuk mempermudah pihak Universitas Putra Indonesia “YPTK” Padang dalam melaksanakan ujian UTS (Ujian Tengah Semester), UAS (Ujian Akhir Semester) dan ujian lainnya.

*Kata Kunci : tes,ujian,berbasis,computer,nodejs,postgresql*

## **ABSTRACT**

<b>TITLE</b>	<b>:</b> COMPUTER ASSISTED TEST DESIGN USING NODE.JS PROGRAMMING LANGUAGES AND POSTGRESQL DATABASE BASED WEB / MOBILE AT UNIVERSITY OF PUTRA INDONESIA "YPTK" PADANG
<b>NAME</b>	<b>:</b> EGO DAFMA DASA
<b>STUDENT NUMBER</b>	<b>:</b> 14101152610565
<b>PROGRAM OF</b>	<b>:</b> SISTEM INFORMASI
<b>STUDY</b>	
<b>ADVISORS</b>	<b>:</b> 1. SILFIA ANDINI, S.KOM., M.KOM. 2. NOVI TRISNA, S. KOM., M.KOM.

The purpose of this research is to design computer assisted test application at Universitas Putra Indonesia "YPTK" Padang. The methods used in this research are Field Research, Library Research and Laboratory Research. The tools used for the design are UML (Unified Modeling Language). The results of this research is to facilitate the University of Putra Indonesia "YPTK" Padang in carrying out Mid Semester Exam, Final Exam Semester and other exams.

Keywords: *test,exam,based,computer,nodejs,postgresql*

## KATA PENGANTAR



Puji dan syukur penulis panjatkan kehadirat Allah Swt, Karena hanya dengan rahmat dan karunia-Nya akhirnya penulis dapat menyelesaikam skripsi ini, yang berjudul **“PERANCANGAN COMPUTER ASSISTED TEST (CAT) MENGGUNAKAN BAHASA PEMOGRAMAN NODE.JS DAN DATABASE POSTGRESQL BERBASIS WEB/MOBILE PADA UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG”** diajukan untuk memenuhi persyaratan akademis Program Studi Strata-1 pada Universitas Putra Indonesia “YPTK”, Jurusan Sistem Informasi pada Fakultas Ilmu Komputer (FILKOM).

Penulisan tugas akhir ini merupakan salah satu implementasi ilmu pengetahuan yang didapatkan sewaktu penulis mengikuti perkuliahan. Didalam penulisan skripsi ini, penulis banyak mendapatkan bimbingan dan arahan serta bantuan dari berbagai pihak baik secara langsung maupun tidak langsung, pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Bapak H. Herman Nawas, selaku Ketua Yayasan Perguruan Tinggi Komputer (YPTK) Padang.
2. Bapak Prof. Dr. Sarjon Defit, S.Kom, M.sc. selaku Rektor Universitas Putra Indonesia “YPTK” Padang.
3. Bapak Dr. Julius Santony,S.Kom, M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK” .

4. Ibu Silfia Andini, S.Kom, M.Kom selaku dosen pembimbing I yang juga telah membimbing dan mengarahkan penulis dalam menyelesaikan skripsi ini.
5. Ibu Novi Trisna, S.Kom, M.Kom selaku dosen pembimbing II yang juga telah membimbing dan mengarahkan penulis dalam menyelesaikan skripsi ini.
6. Bapak/Ibu Dosen, Karyawan dan karyawati Sekretariat Universitas Putra Indonesia “YPTK” yang telah memberikan bantuan segala proses administrasi.

Penulis menyadari masih banyak terdapat kekurangan dalam penulisan skripsi ini baik pembahasan maupun aturan dalam penulisan, untuk itu penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan skripsi ini. Akhir kata semoga skripsi ini dapat bermanfaat baik untuk penulis khususnya maupun pembaca pada umumnya.

Padang, Januari 2018

Penulis

## DAFTAR ISI

<b>LEMBAR PERNYATAAN.....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN.....</b>	<b>ii</b>
<b>LEMBAR PERSETUJUAN.....</b>	<b>iii</b>
<b>KATA PERSEMBAHAN.....</b>	<b>iv</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>ABSTRACT.....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>DAFTAR TABEL.....</b>	<b>xxiii</b>

## BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Hipotesis.....	2
1.4 Ruang Lingkup Penelitian.....	3
1.5 Tujuan dan Manfaat Penelitian.....	4
1.6 Metodologi Penelitian.....	4
1.7 Tinjauan Umum.....	6
1.7.1 Sejarah Berdirinya Universitas Putra Indonesia “YPTK” Padang.....	6
1.7.2 Struktur Organisasi.....	6

1.7.3	Pembagian Tugas Dan Fungsi.....	7
<b>BAB II LANDASAN TEORI</b>		
2.1	Konsep Dasar Sistem.....	16
2.1.1	Pengertian Sistem.....	16
2.1.2	Karakteristik Sistem.....	16
2.1.3	Klasifikasi Sistem.....	18
2.1.4	Daur Hidup Sistem.....	19
2.2	Konsep Informasi.....	21
2.2.1	Pengertian Data dan Informasi.....	21
2.2.2	Atribut Informasi.....	22
2.2.3	Siklus Data dan Informasi.....	24
2.3	Konsep Dasar Sistem Informasi.....	26
2.3.1	Pengertian Sistem Informasi.....	26
2.3.2	Komponen Sistem Informasi.....	26
2.3.3	Perencanaan Sistem Informasi.....	30
2.4	Computer Assisted Test (CAT).....	30
2.5	Konsep Basis Data.....	31
2.5.1	Sistem Basis Data.....	32
2.5.2	Hierarki Data.....	32
2.5.3	Tujuan Pengembangan Basis Data.....	34
2.5.4	Keuntungan Pengembangan Basis Data.....	34
2.5.5	PostgreSQL.....	35
2.6	Bahasa Pemrograman <i>Javascript</i> .....	36

2.6.1	NodeJS.....	36
2.7	<i>Unified Modelling Language (UML)</i> .....	37
2.7.1	Use Case Diagram.....	38
2.7.2	Activity Diagram.....	40
2.7.3	Sequence Diagram.....	41
2.7.4	Class Diagram.....	44

### **BAB III ANALISA DAN PERANCANGAN**

3.1	Analisa Sistem Yang Sedang Berjalan.....	46
3.1.1	Analisa <i>Input</i> , Proses dan <i>Output</i> .....	47
3.1.2	Evaluasi Sistem Yang Sedang Berjalan.....	49
3.1.3	Usulan Sistem Baru.....	50
3.2	Desain Sistem Baru.....	51
3.2.1	Desain Secara Global.....	51
3.2.1.1	Usecase Diagram.....	51
3.2.1.2	Skenario <i>Usecase</i> .....	53
3.2.1.2.1	Skenario <i>Usecase</i> Beranda.....	54
3.2.1.2.2	Skenario <i>Usecase</i> Kelola <i>User</i> .....	55
3.2.1.2.3	Skenario <i>Usecase</i> Kelola Mahasiswa.....	57
3.2.1.2.4	Skenario <i>Usecase</i> Kelola Dosen.....	59
3.2.1.2.5	Skenario <i>Usecase</i> Kelola Matkul.....	60
3.2.1.2.6	Skenario <i>Usecase</i> Kelola Jenis Soal.....	62
3.2.1.2.7	Skenario <i>Usecase</i> Kelola Jenis Ujian.....	63
3.2.1.2.8	Skenario <i>Usecase</i> Kelola Kelas.....	65

3.2.1.2.9 Skenario <i>Usecase</i> Kelola Ujian.....	66
3.2.1.2.10 Skenario <i>Usecase</i> Kelola Soal Ujian.....	68
3.2.1.2.11 Skenario <i>Usecase</i> Kelola Peserta Kuliah.....	69
3.2.1.2.12 Skenario <i>Usecase</i> Hasil Ujian.....	71
3.2.1.2.13 Skenario <i>Usecase</i> Periksa Jawaban Peserta.....	73
3.2.1.2.14 Skenario <i>Usecase</i> <i>Login</i> .....	75
3.2.1.2.15 Skenario <i>Usecase</i> Petunjuk Ujian.....	77
3.2.1.2.16 Skenario <i>Usecase</i> Halaman Ujian.....	78
3.2.1.2.17 Skenario <i>Usecase</i> Kumpulkan Ujian.....	80
3.2.1.2.18 Skenario <i>Usecase</i> Pilih Ujian.....	82
3.2.1.2.19 Skenario <i>Usecase</i> Logout.....	83
3.2.1.2.20 Skenario <i>Usecase</i> Kelola Kuliah.....	85
3.2.1.2.21 Skenario <i>Usecase</i> Cetak Nilai.....	87
3.2.1.3 <i>Class Diagram</i> .....	88
3.2.1.4 <i>Activity Diagram</i> .....	89
3.2.1.4.1 <i>Activity Diagram</i> Admin.....	90
3.2.1.4.2 <i>Activity Diagram</i> Dosen.....	90
3.2.1.4.3 <i>Activity Diagram</i> Mahasiswa.....	91
3.2.1.5 <i>Sequence Diagram</i> .....	92
3.2.1.5.1 <i>Sequence Diagram</i> Beranda.....	92
3.2.1.5.2 <i>Sequence Diagram</i> Kelola User.....	92
3.2.1.5.3 <i>Sequence Diagram</i> Kelola Mahasiswa.....	93
3.2.1.5.4 <i>Sequence Diagram</i> Kelola Dosen.....	94

3.2.1.5.5 <i>Sequence Diagram</i> Kelola Mata Kuliah.....	95
3.2.1.5.6 <i>Sequence Diagram</i> Kelola Jenis Ujian.....	96
3.2.1.5.7 <i>Sequence Diagram</i> Kelola Tipe Soal.....	97
3.2.1.5.8 <i>Sequence Diagram</i> Kelola Kelas.....	98
3.2.1.5.9 <i>Sequence Diagram</i> Kelola Kuliah.....	99
3.2.1.5.10 <i>Sequence Diagram</i> Kelola Peserta Kuliah.....	100
3.2.1.5.11 <i>Sequence Diagram</i> Kelola Ujian.....	101
3.2.1.5.12 <i>Sequence Diagram</i> Kelola Soal Ujian.....	102
3.2.1.5.13 <i>Sequence Diagram</i> Hasil Ujian.....	103
3.2.1.5.14 <i>Sequence Diagram</i> Login.....	104
3.2.1.5.15 <i>Sequence Diagram</i> Pilih Ujian.....	104
3.2.1.5.16 <i>Sequence Diagram</i> Petunjuk Ujian.....	105
3.2.1.5.17 <i>Sequence Diagram</i> Halaman Ujian.....	105
3.2.1.5.18 <i>Sequence Diagram</i> Kumpulkan Ujian.....	106
3.2.1.5.19 <i>Sequence Diagram</i> Logout.....	106
3.2.1.5.20 <i>Sequence Diagram</i> Cetak Nilai.....	107
3.2.1.5.21 <i>Sequence Diagram</i> Periksa Jawaban.....	108
3.2.2 Desain Terinci.....	108
3.2.2.1 Desain Output.....	108
3.2.2.2 Desain <i>Input</i> .....	116
3.2.2.3 Desain <i>File</i> .....	125

## **BAB IV IMPLEMENTASI DAN PENGUJIAN**

4.1 Implementasi Sistem.....	136
------------------------------	-----

4.1.1	Kebutuhan dan Instalasi Sistem.....	136
4.1.2	Kebutuhan Perangkat Keras.....	136
4.1.3	Kebutuhan Perangkat Lunak.....	137
4.1.4	Instalasi Program.....	137
4.1.4.1	Instalasi <i>Postgresql</i> .....	137
4.1.4.2	Impor <i>Database</i> ke <i>Postgresql</i> .....	138
4.1.4.3	Instalasi <i>NodeJS</i> .....	145
4.1.4.4	Menjalankan <i>NodeJS</i> .....	145
4.2	Pengujian Sistem.....	146
4.2.1	Halaman <i>Login</i> .....	146
4.2.2	Halaman Beranda.....	146
4.2.3	Halaman Kelola <i>User</i> .....	147
4.2.4	Halaman Kelola Tipe Soal.....	148
4.2.5	Halaman Kelola Jenis Ujian.....	149
4.2.6	Halaman Kelola Mahasiswa.....	150
4.2.7	Halaman Kelola Dosen.....	151
4.2.8	Halaman Kelola Matkul.....	152
4.2.9	Halaman Kelola Kelas.....	153
4.2.10	Halaman Kelola Kuliah.....	154
4.2.11	Halaman Kelola Peserta Kuliah.....	155
4.2.12	Halaman Kelola Ujian.....	156
4.2.13	Halaman Kelola Soal Ujian.....	157
4.2.14	Halaman Pilih Ujian.....	159

4.2.15 Halaman Petunjuk Ujian.....	160
4.2.16 Halaman Ujian.....	160
4.2.17 Halaman Periksa Jawaban.....	162
4.2.18 Halaman Hasil Ujian.....	162

## **BAB V PENUTUP**

5.1 Kesimpulan.....	165
5.2 Saran-Saran.....	165

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

Listing Program

## DAFTAR GAMBAR

1.1	Struktur Organisasi Universitas Putra Indonesia YPTK Padang.....	7
2.1	Daur Hidup Sistem.....	21
2.2	Hubungan Antara Data dan Informasi.....	22
2.3	Atribut-atribut Informasi.....	23
2.4	Siklus Data dan Informasi.....	25
2.5	Komponen Sistem Informasi.....	28
2.6	Hierarki Data.....	33
2.7	Contoh kelas diagram.....	45
3.1	Lembar Soal Ujian.....	48
3.2	Lembar Jawaban.....	49
3.3	Usecase Diagram.....	53
3.4	Class Diagram.....	89
3.5	Activity Diagram Admin.....	90
3.6	Activity Diagram Dosen.....	91
3.7	Activity Diagram Mahasiswa.....	91
3.8	Sequence Diagram Beranda.....	92
3.9	Sequence Diagram Kelola User.....	93
3.10	Sequence Diagram kelola mahasiswa.....	94
3.11	Sequence Diagram kelola dosen.....	95
3.12	Sequence Diagram kelola mata kuliah.....	96
3.13	Sequence Diagram kelola jenis ujian.....	97

3.14 Sequence Diagram kelola tipe soal.....	98
3.15 Sequence Diagram kelola kelas.....	99
3.16 Sequence Diagram kelola kuliah.....	100
3.17 Sequence Diagram kelola peserta kuliah.....	101
3.18 Sequence Diagram kelola ujian.....	102
3.19 Sequence Diagram kelola soal ujian.....	103
3.20 Sequence Diagram hasil ujian.....	103
3.21 Sequence Diagram login.....	104
3.22 Sequence Diagram pilih ujian.....	104
3.23 Sequence Diagram petunjuk ujian.....	105
3.24 Sequence Diagram halaman ujian.....	106
3.25 Sequence Diagram kumpulkan ujian.....	106
3.26 Sequence Diagram logout.....	107
3.27 Sequence Diagram cetak nilai.....	107
3.28 Sequence Diagram periksa jawaban.....	108
3.29 Halaman Daftar Mahasiswa.....	109
3.30 Halaman Daftar Dosen.....	109
3.31 Halaman Daftar Mata Kuliah.....	110
3.32 Halaman Daftar Kelas.....	110
3.33 Halaman Daftar Tipe Soal.....	111
3.34 Halaman Daftar Jenis Ujian.....	111
3.35 Halaman Daftar User.....	112
3.36 Halaman Daftar Kuliah.....	112

3.37 Halaman Daftar Peserta Kuliah.....	113
3.38 Halaman Daftar Ujian.....	114
3.39 Halaman Daftar Soal Ujian.....	114
3.40 Halaman Daftar Hasil Ujian.....	115
3.41 Laporan Hasil Ujian.....	116
3.42 Form Mahasiswa.....	117
3.43 Form Mahasiswa.....	117
3.44 Form Matkul.....	118
3.45 Form Kelas.....	118
3.46 Form Tipe Soal.....	119
3.47 Form Jenis Ujian.....	119
3.48 Form User.....	120
3.49 Form Kuliah.....	120
3.50 Form Peserta Kuliah.....	121
3.51 Form Ujian.....	122
3.52 Form Soal Ujian Essai.....	122
3.53 Form Soal Ujian Pilihan Ganda.....	123
3.54 Form Halaman Ujian Essai.....	124
3.55 Form Halaman Ujian Pilihan Ganda.....	125
4.1 Proses Instalasi Postgresql.....	138
4.2 Membuat database pada postgresql.....	138
4.3 Import database ke postgresql.....	139
4.4 Masuk ke database.....	139

4.5 Struktur tabel tbmahasiswa.....	140
4.6 Struktur tabel tbdosen.....	140
4.7 Struktur tabel tbmatkul.....	140
4.8 Struktur tabel tbjenis_soal.....	141
4.9 Struktur tabel tbjenis_ujian.....	141
4.10 Struktur tabel tbuser.....	141
4.11 Struktur tabel tbkelas.....	142
4.12 Struktur tabel tbkuliah.....	142
4.13 Struktur tabel tbpeserta_kuliah.....	142
4.14 Struktur tabel tbujian.....	143
4.15 Struktur tabel tbsoal.....	143
4.16 Struktur tabel tbsoal_ujian.....	143
4.17 Struktur tabel tbhasil_ujian.....	144
4.18 Struktur tabel kelas_kuliah.....	144
4.19 Struktur tabel tbjenis_user.....	144
4.20 Proses Instalasi NodeJS.....	145
4.21 Menjalankan Nodejs.....	145
4.22 Tampilan Halaman Login.....	146
4.23 Tampilan Halaman Beranda.....	147
4.24 Tampilan Halaman Kelola User.....	147
4.25 Tampilan Form Tambah User.....	148
4.26 Tampilan Halaman Kelola Tipe Soal.....	148
4.27 Tampilan Form Tambah Tipe Soal.....	149

4.28 Tampilan Halaman Kelola Jenis Ujian.....	150
4.29 Tampilan Form Tambah Jenis Ujian.....	150
4.30 Tampilan Halaman Kelola Mahasiswa.....	151
4.31 Tampilan Form Tambah Mahasiswa.....	151
4.32 Tampilan Halaman Kelola Dosen.....	152
4.33 Tampilan Form Tambah Dosen.....	152
4.34 Tampilan Halaman Kelola Matkul.....	153
4.35 Tampilan Form Tambah Matkul.....	153
4.36 Tampilan Form Tambah Matkul.....	154
4.37 Tampilan Form Tambah Kelas.....	154
4.38 Tampilan Halaman Kelola Kuliah.....	155
4.39 Form Tambah Kuliah.....	156
4.40 Tampilan Halaman Kelola Peserta Kuliah.....	156
4.41 Form Tambah Peserta Kuliah.....	157
4.42 Tampilan Halaman Kelola Ujian.....	157
4.43 Form Tambah Ujian.....	158
4.44 Tampilan Halaman Ujian Tipe Soal Essai.....	159
4.45 Form Tambah Soal Pilihan Ganda.....	160
4.46 Form Tambah Soal Essai.....	160
4.47 Tampilan Halaman Piih Ujian.....	161
4.48 Tampilan Halaman Piih Ujian.....	162
4.49 Tampilan Halaman Ujian Tipe Soal Pilihan Ganda.....	162
4.50 Tampilan Halaman Ujian Tipe Soal Essai.....	163

4.51 Tampilan Halaman Periksa Jawaban.....	164
4.52 Tampilan Halaman Hasil Ujian.....	165
4.53 Laporan Hasil Ujian Format File CSV.....	165

## DAFTAR TABEL

2.1	Simbol-simbol pada diagram usecase.....	38
2.2	Simbol-simbol pada activity diagram.....	40
2.3	Simbol-simbol pada diagram sekuen.....	42
3.1	Defenisi Aktor.....	52
3.2	Skenario Usecase Beranda.....	54
3.3	Skenario Usecase Kelola User.....	56
3.4	Skenario Usecase Kelola Mahasiswa.....	57
3.5	Skenario Usecase Kelola Dosen.....	59
3.6	Skenario Usecase Kelola Matkul.....	60
3.7	Skenario Usecase Kelola Jenis Soal.....	62
3.8	Skenario Usecase Kelola Jenis Ujian.....	63
3.9	Skenario Usecase Kelola Kelas.....	65
3.10	Skenario Usecase Kelola Ujian.....	67
3.11	Skenario Usecase Kelola Soal Ujian.....	68
3.12	Skenario Usecase Kelola Peserta Kuliah.....	70
3.13	Skenario Usecase Hasil Ujian.....	71
3.14	Skenario Usecase Periksa Jawaban Peserta.....	73
3.15	Skenario Usecase Login.....	75
3.16	Skenario Usecase Petunjuk Ujian.....	77
3.17	Skenario Usecase Halaman Ujian.....	79
3.18	Skenario Usecase Periksa Kumpulkan Ujian.....	80

3.19 Skenario Usecase Pilih Ujian.....	82
3.20 Skenario Usecase Periksa Jawaban Peserta.....	84
3.21 Skenario Usecase Kelola Kuliah.....	85
3.22 Skenario Usecase Kelola Kuliah.....	87
3.23 Desain File Kelas Kuliah.....	125
3.24 Desain File Kuliah.....	126
3.25 Desain File Peserta Kuliah.....	127
3.26 Desain File Hasil Ujian.....	127
3.27 Desain File Jawaban.....	128
3.28 Desain File Soal.....	129
3.29 Desain File Kelas.....	130
3.30 Desain File Jenis Soal.....	130
3.31 Desain File Jenis Ujian.....	131
3.32 Desain File Jenis User.....	131
3.33 Desain File User.....	132
3.34 Desain File Ujian.....	132
3.35 Desain File Mahasiswa.....	133
3.36 Desain File Dosen.....	134
3.37 Desain File Matkul.....	134
3.38 Desain File Soal Ujian.....	135

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Penerapan sistem komputerisasi didalam kehidupan sehari-hari di saat 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 di mana komputer atau perangkat smartphone digunakan sebagai media pengakses informasi pembelajaran.

Tak hanya itu saja, sekarang sudah berkembang aplikasi komputer yang bernama “*Computer Assisted Test*” di mana kegiatan tes atau ujian atau semacamnya dilakukan berbasis komputer. Hal tersebut memungkinkan peserta melakukan pengisian jawaban dan pembacaan soal serta pengeraannya 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 di atas, maka penulis tertarik untuk menerapkan aplikasi *Computer Assisted Test* dalam bentuk skripsi dengan judul **“PERANCANGAN COMPUTER ASSISTED TEST (CAT) MENGGUNAKAN BAHASA PEMROGRAMAN NODE.JS BERBASIS WEB/MOBILE PADA UNIVERSITAS PUTRA INDONESIA “YPTK” PADANG”**

### **1.2 Perumusan Masalah**

Berdasarkan latar belakang di atas, maka rumusan masalah pada penelitian ini adalah sebagai berikut :

1. Bagaimana merancang aplikasi *computer assisted test* yang dapat memudahkan mahasiswa serta pihak kampus dalam melangsungkan ujian?
2. Bagaimana aplikasi *computer assisted test* dapat memudahkan dosen dalam melakukan pemeriksaan serta mengelola hasil ujian mahasiswa?
3. Bagaimana aplikasi *computer assisted test* dapat meningkatkan efisiensi pelaksanaan ujian?

### **1.3 Hipotesis**

Berdasarkan perumusan masalah di atas, maka dapat diperoleh suatu hipotesis yaitu :

1. Dengan diterapkannya *Computer Assisted Test* dalam pelaksanaan ujian, maka kesalahan mahasiswa dalam melaksanakan ujian atau kesalahan teknis lainnya bisa diminimalkan.
2. Dengan adanya *Computer Assisted Test*, diharapkan hasil ujian mahasiswa dapat dihasilkan lebih mudah.
3. Dengan menggunakan kemampuan bahasa pemrograman *javascript* dan dengan bantuan *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 Ruang Lingkup Penelitian**

Agar pembahasan lebih terarah serta sesuai harapan, maka sebagai ruang lingkup dibuat pembatasan masalah yang meliputi :

1. Penelitian terhadap proses pelaksanaan ujian baik ujian tengah semester maupun ujian akhir semester
2. Menggunakan *NodeJS* dan basis data *Postgresql*
3. Penelitian hanya dilakukan pada Universitas Putra Indonesia “YPTK”

Padang

## 1.5 Tujuan dan Manfaat Penelitian

Berdasarkan perumusan masalah yang telah dijelaskan, maka tujuan dari penelitian ini adalah sebagai berikut :

1. Menciptakan aplikasi *Computer Assisted Test* yang dapat mempermudah mahasiswa serta dosen dalam pelaksanaan ujian.
2. Menciptakan aplikasi *Computer Assisted Test* yang dapat diterapkan pada Universitas Putra Indonesia “YPTK” Padang.
3. Menciptakan aplikasi *Computer Assisted Test* yang dapat mempermudah dosen dalam pemeriksaan hasil ujian mahasiswa.

Berdasarkan rumusan masalah dan tujuan yang telah dijelaskan sebelumnya, diharapkan memperoleh suatu manfaat yaitu sebagai berikut :

1. Bagi pihak Universitas Putra Indonesia “YPTK” Padang  
Diharapkan dengan dilakukannya penelitian ini, pihak Universitas Putra Indonesia “YPTK” Padang dapat dengan lebih mudah dalam melengkungkan 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.

## 1.6 Metodologi Penelitian

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

### 1. Penelitian Lapangan

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

### 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 sekali-gus sebagai pemecahan masalah yang ada.

### 3. Penelitian Laboratorium

Pada penelitian ini digunakan sebuah laptop yang mana *hardware* dan *softwere* yang terpasang pada laptop tersebut adalah dengan spesifikasi sebagai berikut :

#### a. Hardware

- 1) Prosessor AMD A8-7410
- 2) Memori RAM 4 Gb
- 3) Media penyimpanan hardisk 1 Tb
- 4) Printer canon MP287

#### b. Software

- 1) Sistem Operasi Ubuntu 16.04 32bit
- 2) *NodeJS* versi 8.9.1
- 3) *Node Package Manager* (NPM) versi 5.4.2
- 4) *Text editor* Geany versi 1.29

- 5) *Postgresql* versi 9.5
- 6) Software pendukung lainnya

## **1.7 Tinjauan Umum**

### **1.7.1 Sejarah Berdirinya Universitas Putra Indonesia “YPTK” Padang**

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

### **1.7.2 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 dari Universitas Putra Indonesia “YPTK” Padang dapat dilihat pada Gambar 1.1 :



*Sumber : (Website Resmi Universitas UPI YPTK)*

**Gambar 1.1 Struktur Organisasi Universitas Putra Indonesia YPTK Padang**

### 1.7.3 Pembagian Tugas Dan Fungsi

Berdasarkan Gambar 1.1 di atas, berikut adalah penjelasan dari pembagian tugas dan fungsi pada Universitas Putra Indonesia “YPTK” Padang :

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 kebi-

jaksanaan 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 kerumah-tanggaan 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 dibawah 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
  - 7. 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.
  - 8. 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 Kemasasiswaan. 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.

9. 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

## 10. 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 pembinaan-

nya 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 Indonesia “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)

##### **2.1.1 Pengertian Sistem**

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

##### **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)

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

## 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 mempengaruhi 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.

### 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)

#### 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.

##### 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. Daur hidup sistem dapat dijelaskan sebagai berikut (Tata Sutabri, 2016) :

###### 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 ke-

butuhan 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. Instalasi Sistem

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

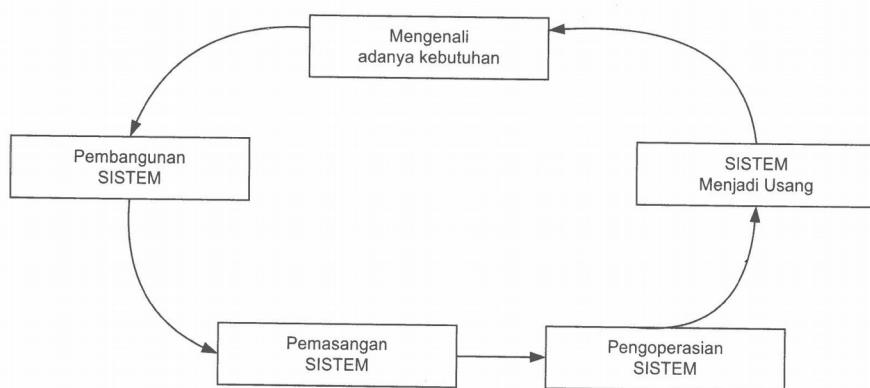
## 4. Pengoperasian Sistem

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. Untuk lebih jelasnya mengenai daur hidup sistem dapat dilihat pada Gambar 2.1 :



Sumber : (Tata Sutabri, 2016)

**Gambar 2.1 Daur Hidup Sistem**

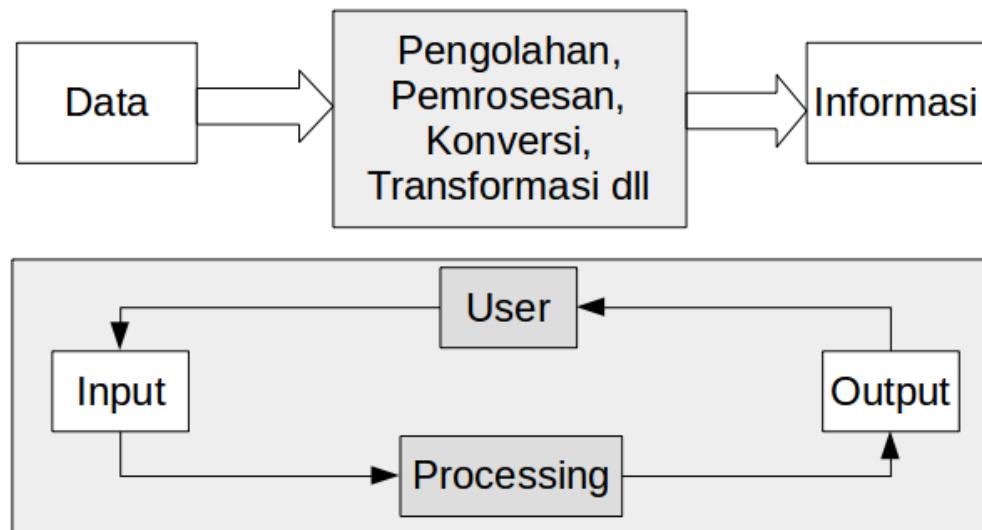
## 2.2 Konsep Informasi

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)

### 2.2.1 Pengertian Data dan Informasi

Menurut Eddy Prahasta (2014) 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 peneri-

manya. Untuk lebih jelasnya mengenai hubungan antara data dan informasi dapat dilihat pada Gambar 2.2 :

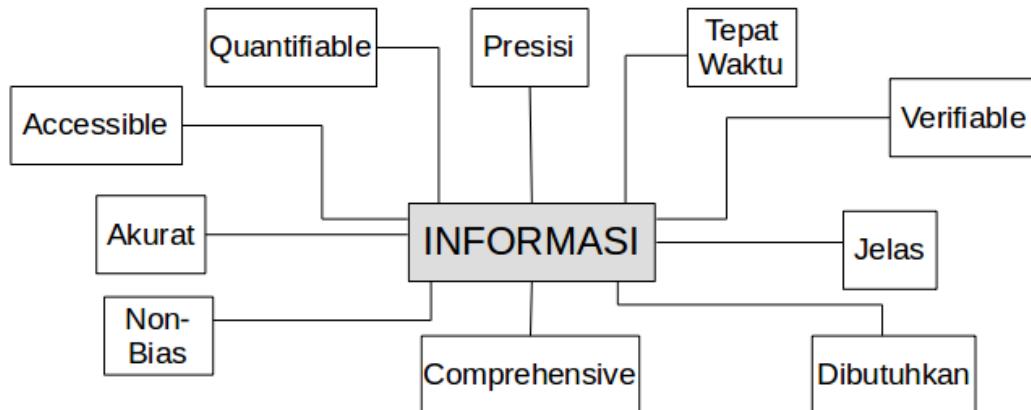


*Sumber : (Eddy Prahasta, 2014)*

**Gambar 2.2 Hubungan Antara Data dan Informasi**

### **2.2.2 Atribut Informasi**

Banyak atribut atau kualitas yang berkaitan dengan konsep informasi dapat membantu perancang dalam mengidentifikasi kebutuhan informasi. Untuk lebih jelasnya mengenai atribut-atribut informasi dapat dilihat pada Gambar 2.3 :



Sumber : (Eddy Prahasta, 2014)

**Gambar 2.3 Atribut-atribut Informasi**

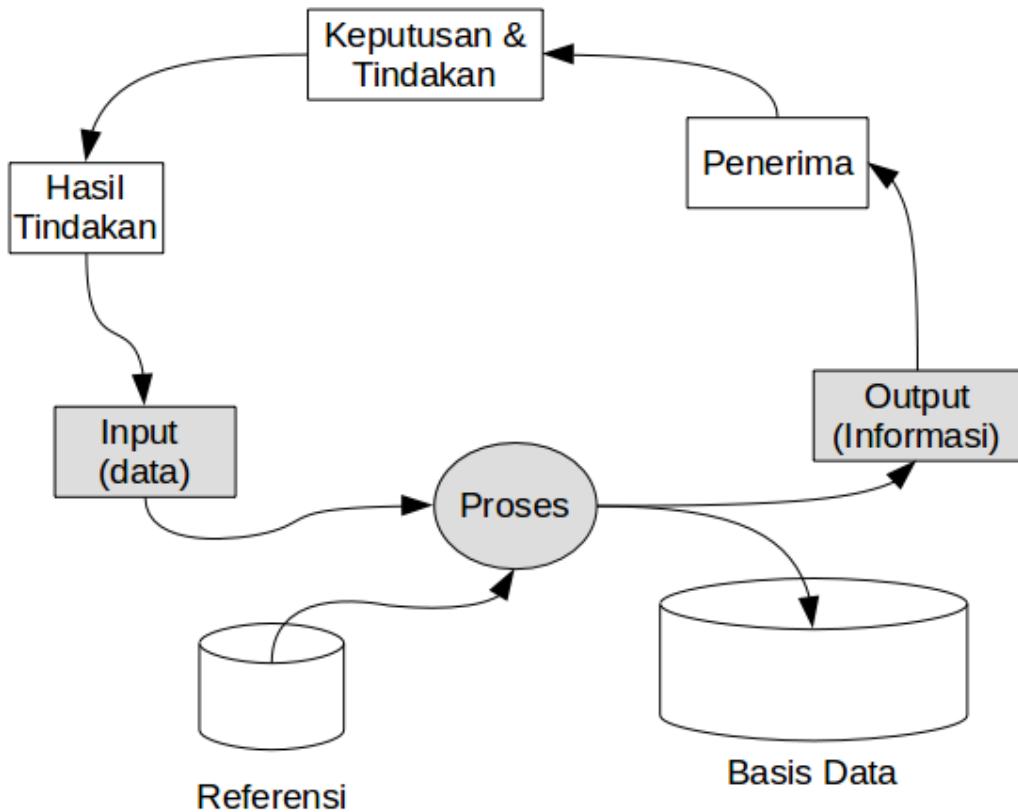
Dari Gambar 2.3 dapat dijelaskan sebagai berikut :

1. Akurat: tingkat kebebasan informasi dari kesalahan.
2. *Quantifiable*: tingkat kemampuan menyatakan informasi dalam bentuk numerik.
3. Presisi: ukuran detil untuk menyediakan infomasi.
4. Tepat waktu: penerimaan informasi masih dalam toleransi waktu yang dibutuhkan oleh penerima; tidak kadaluarsa.
5. Jelas: tingkat kebebasan informasi dari keraguan.
6. Dibutuhkan: tingkat relevansi informasi dengan kebutuhan pengguna.
7. *Verifiable*: tingkat kesamaan nilai sebagai hasil pengujian informasi yang sama oleh berbagai pengguna.
8. *Accessible*: tingkat kemudahan dan kecepatan memperoleh informasi.
9. Non-bias: tingkat perubahan untuk memodifikasi informasi dengan tujuan mempengaruhi para penerimanya.
10. *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 atributnya pada perancangan sistem informasi. Jadi, yang diperlukan adalah penyediaan informasi yang benar pada waktu yang tepat (Eddy Prahasta, 2014)

### **2.2.3 Siklus Data dan Informasi**

Menurut Eddy Prahasta (2014) 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*. Untuk lebih jelasnya mengenai siklus data dan informasi dapat dilihat pada Gambar 2.4 :



Sumber : (Eddy Prahasta, 2014)

**Gambar 2.4 Siklus Data dan Informasi**

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)

## 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)

### 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)

### 2.3.2 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan yang saling bekerjasama. Adapun komponen sistem informasi, (Tata Sutabri, 2016) yaitu :

#### 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

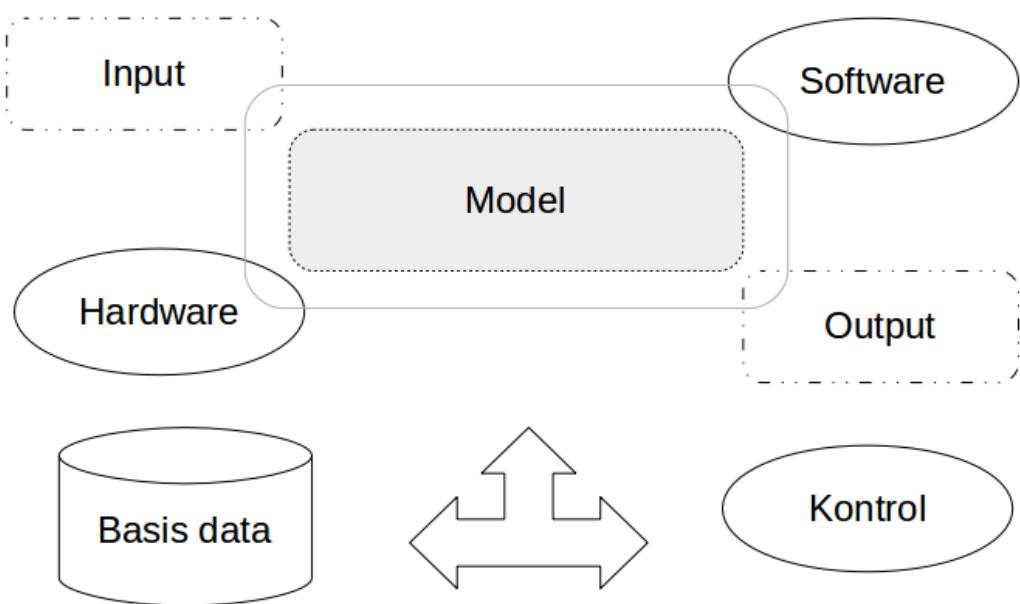
Teknologi merupakan *toolbox* dalam sistem informasi. Teknologi didunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian yaitu teknisi, perangkat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

## 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 di atasi. Untuk lebih jelas mengenai hubungan blok-blok pada sistem informasi dapat dilihat pada Gambar 2.5 :



Sumber : (Tata Sutabri, 2016)

**Gambar 2.5 Komponen Sistem Informasi**

Berdasarkan komponen fisik penyusunnya, komponen sistem informasi terdiri atas (Tata Sutabri, 2016) :

## 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 .

### 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) :

1. Tingkat I : Ide, mengetahui perlu adanya perubahan.
2. Tingkat II : Desain, merancang cara pemecahannya.
3. Tingkat III : Pelaksanaan, menerapkan desain kedalam sistem.
4. Tingkat IV : Kontrol, memeriksa tingkat pelaksanaan dijalankan sesuai dengan desain.
5. Tingkat V : Evaluasi, memeriksa apakah perubahan yang terjadi sesuai dengan tujuan semula.
6. Tingkat VI : Tindak lanjut, melaksanakan perubahan sesuai dengan hasil evaluasi yang ada.

### 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)

Menurut Khusnul Khotimah (2016), tujuan dari penggunaan Computer Assisted Test adalah sebagai berikut :

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

Menurut BKN dalam jurnal I Ketut Buana dan Made Gede Wirakusuma (2015) 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, menampilkan dan menjawab soal.
3. Soal ditampilkan secara acak pada setiap peserta.
4. Pemeriksaan hasil tes langsung dilakukan oleh aplikasi CAT secara otomatis.

## 2.5 Konsep Basis Data

Menurut Eddy Sutanta (2011), 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.

Sedangkan menurut Sri Rahmawati (2014) Basis data atau *database* adalah sebuah tempat untuk menyimpan data-data penting dan disamping itu data-data yang ada didalam database dapat dipanggil kembali kapan saja saat kita membutuhkannya. Dalam kehidupan sehari hari basis data ini dapat dicontohkan seperti lemari penyimpanan arsip-arsip penting yang telah diberi tanda atau kode tertentu agar saat dilakukan pencarian menjadi lebih mudah dan cepat.

### **2.5.1 Sistem Basis Data**

Menurut Eddy Sutanta (2011) 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.

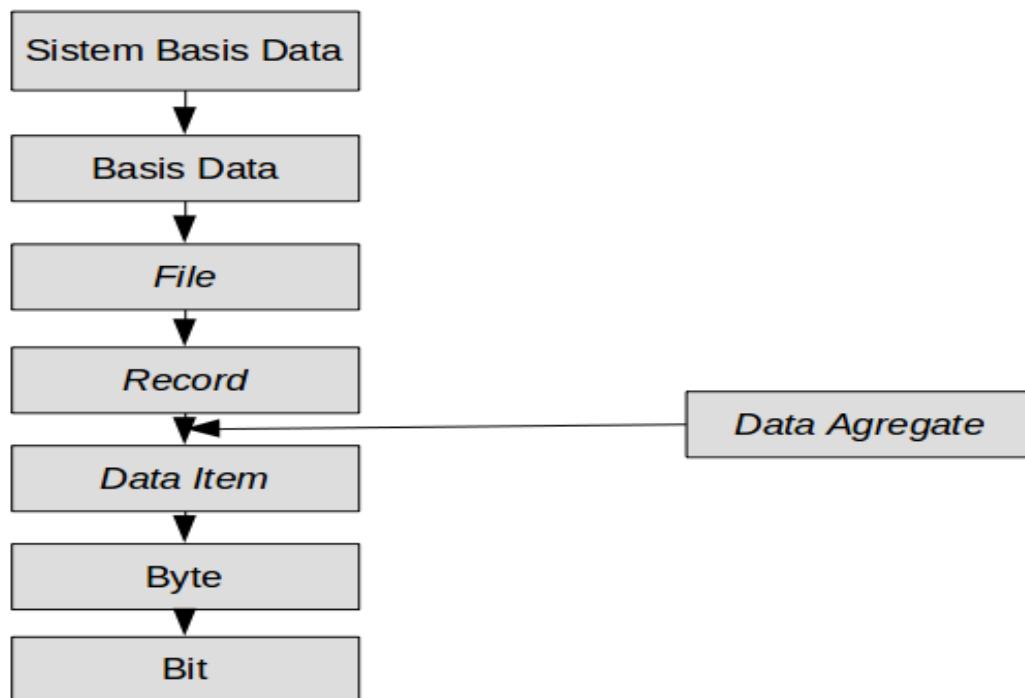
### **2.5.2 Hierarki Data**

Hierarki data yang dimulai dari paling kompleks hingga yang paling sederhana dapat dijelaskan sebagai berikut (Eddy Sutanto, 2011) :

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.
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 Agregate*, 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.

Untuk lebih jelasnya mengenai Hierarki data dapat dilihat pada Gambar 2.6 :



Sumber : (Eddy Sutanta, 2011)

**Gambar 2.6 Hierarki Data**

### **2.5.3 Tujuan Pengembangan Basis Data**

Setiap tindakan pasti memiliki tujuan yang akan dicapai. Tujuan utama dari pengembangan basis data adalah sebagai berikut (Eddy Sutanta, 2011) :

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.

### **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 (Eddy Sutanto, 2011) :

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.

### 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* (Sukhdeep Kaur, 2016)

*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* (Sukhdeep Kaur, 2016)

*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)

Menurut Mochamad Alfan Rosid (2016) *Postgresql* sebagai *ORDBMS (Object Relational Database Management System)* 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.

## 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 (Betah Sidik, 2011)

### 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., Muchammad 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. Sebagaimana 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

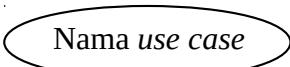
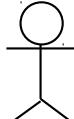
Menurut Rosa A.S dan M. Shalahuddin (2015), *use case* merupakan pemodelan yang digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakannya.

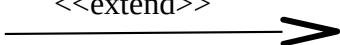
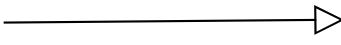
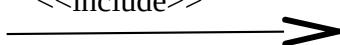
Ada dua hal utama pada *use case* yaitu pendefinisian apa yang akan disebut aktor dan *use case*. Berikut adalah penjelasan dari aktor dan *use case* :

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.

Dalam menggambarkan *use case*, digunakan simbol-simbol yang memiliki makna-makna tertentu. Simbol-simbol yang digunakan pada *use case* dapat dilihat pada Tabel 2.1 :

**Tabel 2.1 Simbol-simbol pada diagram *usecase***

No	Simbol	Deskripsi
1	2	3
1	<p><i>Use case</i></p> 	<p>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>.</p>
2	<p>Aktor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>
3	<p>Asosiasi</p> 	<p>komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

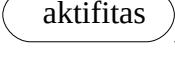
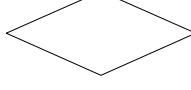
1	2	3
4	Ekstensi 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.
5	Generalisasi 	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimanafungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Menggunakan <i>include</i> 	Relasi <i>usecase</i> tambahan ke sebuah <i>usecse</i> dimana <i>usecase</i> yang ditambahkan memerlukan <i>usecase</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>usecase</i> ini.

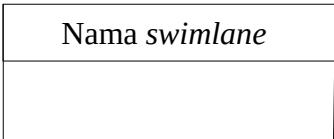
Sumber : Rosa A.S dan M. Shalahudin, 2015

### 2.7.2 *Activity Diagram*

Diagram aktivitas menggambarkan aliran kerja dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Simbol-simbol pada diagram aktivitas dapat dilihat pada Tabel 2.2 :

**Tabel 2.2 Simbol-simbol pada *activity diagram***

No	Simbol	Deskripsi
1	2	3
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	aktivitas 	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja
3	Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

1	2	3
6	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap yang terjadi.

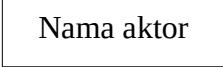
Sumber : Rosa A.S dan M. Shalahudin, 2015

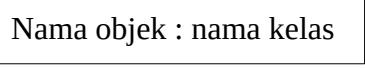
### 2.7.3 Sequence Diagram

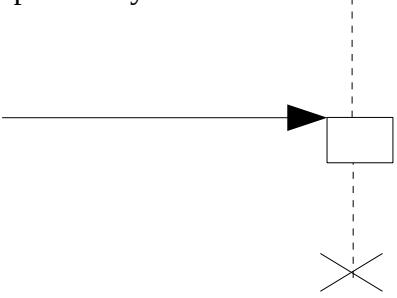
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang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

Simbol-simbol pada diagram sekuen dapat dilihat pada Tabel 2.3 :

**Tabel 2.3 Simbol-simbol pada diagram sekuen**

No	Simbol	Deskripsi
1	2	3
1	Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.

1	2	3
2	Garis hidup 	Menyatakan kehidupan suatu objek.
3	Objek  Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan.
4	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5	Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	Pesan tipe call 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7	Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data ke objek lainnya, arah panah mengarah pada objek yang dikirim.

1	2	3
8	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.
9	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 : 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 – M. Shalahudin, 2015)

Menurut Rosa A.S dan M. Shalahudin (2015) Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas sebagai 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.

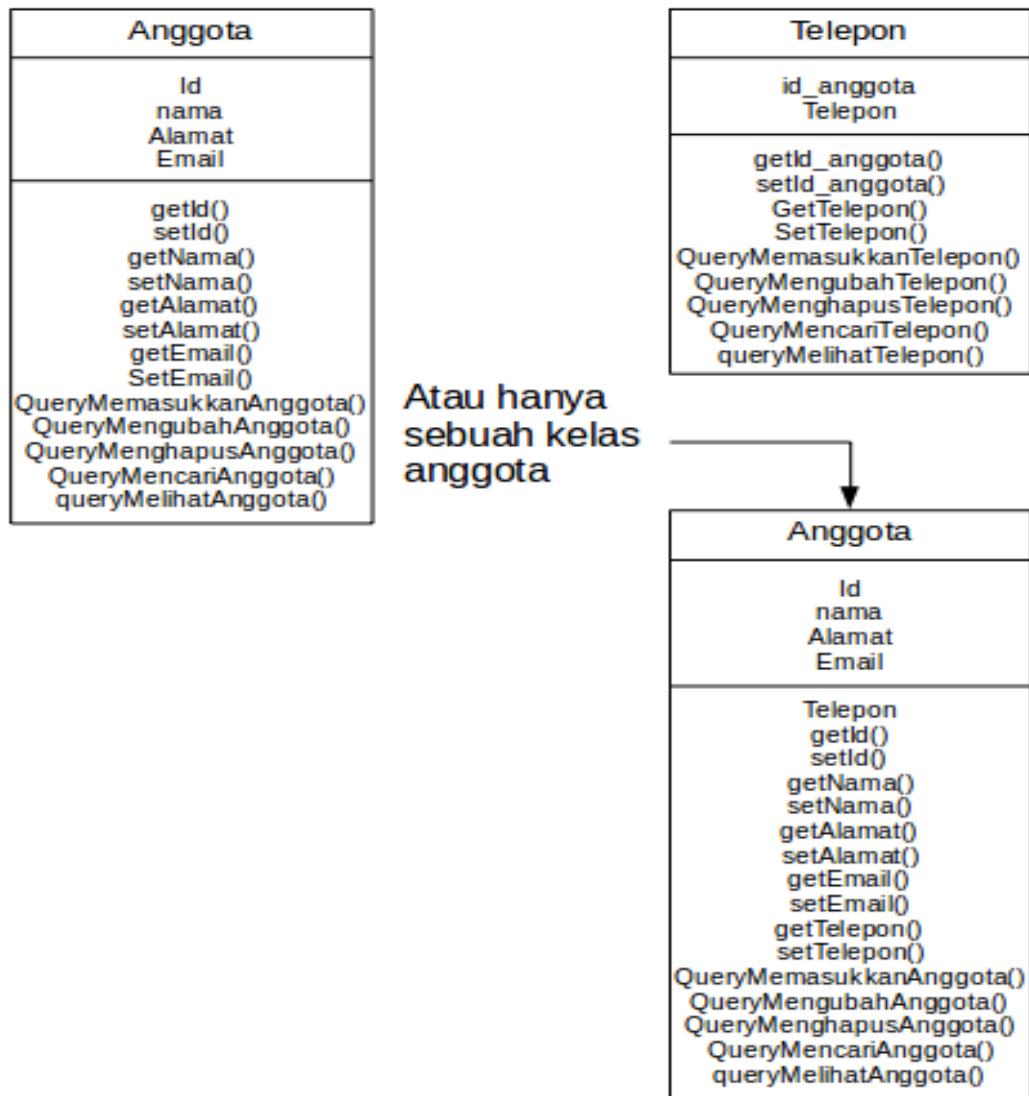
3. Kelas yang diambil dari pendefinisian *use case*

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 diperlengkungjawabkan atau tetap ada di dalam perangcangan kelas.

Untuk lebih jelasnya contoh kelas diagram dapat dilihat pada Gambar 2.7 :



Sumber : (Rosa A.S – M. Shalahudin, 2015)

**Gambar 2.7 Contoh kelas diagram**

## **BAB III**

### **ANALISA DAN PERANCANGAN**

#### **3.1 Analisa Sistem Yang Sedang Berjalan**

Analisa sistem dapat diartikan sebagai kegiatan memahami sebuah sistem. Dengan memahami sebuah sistem, maka dapat diketahui tentang cara kerja sistem, kelebihan sistem dan kekurangan.

Dari hasil penelitian yang telah dilakukan di Universitas Putra Indonesia “YPTK” Padang baik secara langsung melalui *survey* maupun tidak langsung, maka dapat diketahui bahwa sistem pelaksanaan ujian di Universitas Putra Indonesia “YPTK” Padang masih manual. Dimana kertas sangat dibutuhkan sebagai media soal dan media jawaban. Proses pemeriksaan jawaban masih dilakukan secara manual oleh dosen sehingga cukup menyulitkan dosen mengelola jumlah kertas jawaban yang jumlahnya cukup banyak. Cara kerja sistem pelaksanaan ujian yang sedang berjalan pada Universitas Putra Indonesia “YPTK” Padang dapat dijelaskan sebagai berikut :

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.

### **3.1.1 Analisa *Input*, Proses dan *Output***

Pada sebuah sistem, terdapat 3 (tiga) komponen utama yaitu *input* yang merupakan masukkan kesistem, proses yang merupakan pengolahan *input* dan *output* yang merupakan hasil dari proses. *Input*, proses dan *output* yang ditemukan pada sistem pelaksanaan ujian Universitas Putra Indonesia “YPTK” Padang dapat dijelaskan sebagai berikut :

#### **1. Analisa *input***

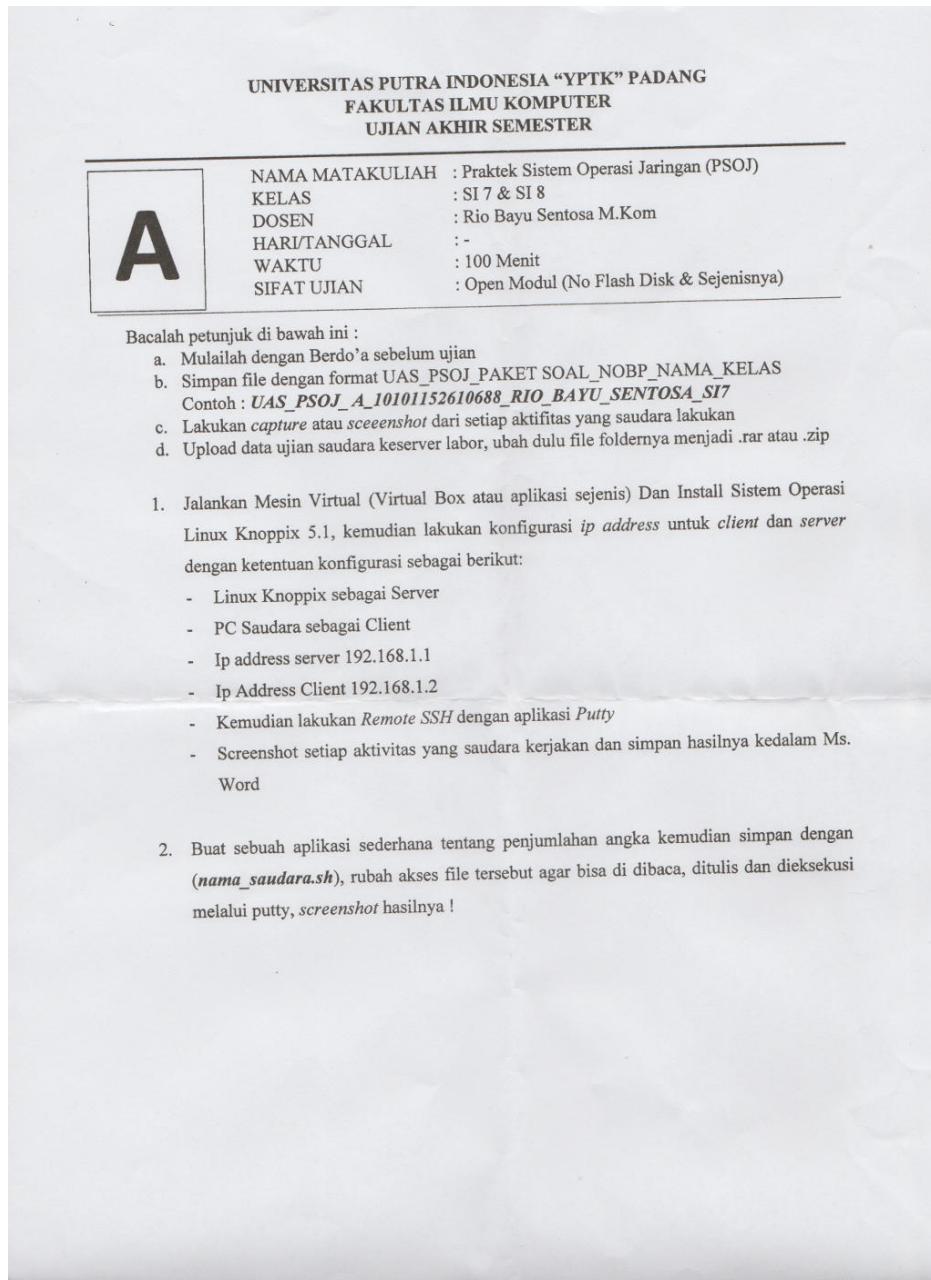
Input yang ditemukan adalah data mahasiswa, data dosen, data mata kuliah, data kuliah, data soal dan data jawaban.

#### **2. Analisa proses**

Proses yang ditemukan adalah proses pemeriksaan jawaban peserta ujian oleh dosen. Proses pemeriksaan jawaban dilakukan secara manual oleh dosen dimana dosen memiliki lembar jawaban peserta dan mengecek lembar jawaban tersebut satu per satu.

### 3. Analisa *output*

*Ouput* yang ditemukan adalah lembar soal ujian dan jawaban ujian. Adapun lembar soal ujian dan jawaban ujian dapat dilihat pada Gambar 3.1 dan 3.2 :



**Gambar 3.1 Lembar Soal Ujian**

59		FAKULTAS : FAKULTAS KOMPUTER NOMBOR BP : 1401119610866 NAMA : EGO DAFMA PATA KELAS : ST-4 MATA UJIAN : KEAMANAN KOMPUTER HARI/TGL : Sabtu/28 Oktober 2017 T. TANGAN : <i>[Signature]</i>
----	--	--

1) Keamanan komputer adalah suatu bidang ilmu yang mempelajari upaya untuk melindungi informasi & komputer dari pihak yang tidak bertanggung jawab. Komponen yang harus dijaga pada sistem komputer:  
 - Main memory  
 - CPU  
 - Input/Output

2) CIA (Confidential, Integrity, Availability)  
 - Confidential merupakan suatu keadaan yang menggambarkan kerahasiaan sebuah informasi yang tersimpan di dalam komputer. Informasi tersebut tidak bisa diakses oleh orang lain yang tidak punya akses.  
 - Integrity merupakan suatu keadaan yang menggambarkan keaslian dan keutuhan informasi yang tersimpan di dalam komputer. Informasi tersebut tidak berubah dan tidak diubah oleh orang yang tidak bertanggung jawab sehingga informasi yang dihasilkan benar.

- Availability merupakan suatu keadaan yang menggambarkan ketersediaan informasi terhadap pengguna. Dimana informasi yang hanya tersedia untuk pengguna yang sudah melalui tahap autentikasi dan akses.

Keamanan komputer menggambarkan upaya untuk melindungi komputer dari pengguna yang tidak bertanggung jawab dan CIA tersebut menggambarkan tentang bagaimana keamanan komputer bisa tercipta.

3) a) chmod -u ~~775~~ /home/budi/belajar -pft chmod -u 125 /home/budi/belajar. *[X]*  
 b) chmod -u ~~771~~ /home/budi/belajar -pft chmod -u 741 /home/budi/belajar. *[X]*  
 c) chmod -u ~~774~~ /home/budi/belajar -pft chmod -u 644 /home/budi/belajar. *[X]*

4) salah satu bagian dari SO adalah kernel yang merupakan jantung dari sebuah SO.  
 Cara mengamankan SO:  
 - menginstall patch terbaru.  
 - memasang keamanan tambahan seperti Firewall dan antivirus.  
 - mengkonfigurasi permission dari user, group dan other.  
 - menguji keamanan SO  
 - memblokir aplikasi atau protokol yang tidak diinginkan.

**Gambar 3.2 Lembar Jawaban**

### 3.1.2 Evaluasi Sistem Yang Sedang Berjalan

Berdasarkan analisa di atas, diketahui bahwa soal yang telah dibuat dosen dicetak pada kertas dan mahasiswa menjawab soal tersebut menggunakan kertas *doublefolio* sebagai media lembar jawaban. Kendala yang ditemukan pada sistem lama dapat dijelaskan sebagai berikut :

1. Pada sistem lama, kertas sangat dibutuhkan. Karena lembar soal dan jawaban menggunakan kertas. Kejadian kurangnya lembar jawaban atau lembaran soal bisa menjadi masalah jika jumlah kertas yang akan digunakan tidak diprediksi dengan baik.
2. Pada beberapa kasus, dosen tidak bisa membaca tulisan mahasiswa karena setiap mahasiswa memiliki tulisan yang berbeda-beda sehingga hal tersebut membingungkan dosen dalam memberikan nilai akhir.
3. Dosen sedikit kewalahan dalam mengelola lembar jawaban mahasiswa karena rata-rata dalam sebuah kelas, terdapat lebih dari 30 mahasiswa dan seorang dosen tidak hanya mengajar sebuah kelas dengan mata kuliah yang sama, sehingga dosen akan cukup kewalahan dalam mengelola lembar jawaban mahasiswa yang jumlahnya cukup banyak.

### **3.1.3 Usulan Sistem Baru**

Berdasarkan evaluasi sistem yang telah dijelaskan sebelumnya, akan dilakukan usulan sistem baru yang akan dikembangkan :

1. Merancang sebuah aplikasi *computer assisted test* yang dapat mempermudah pelaksanaan ujian.
2. Aplikasi tersebut dikembangkan menggunakan bahasa pemrograman *javascript* dengan bantuan *nodejs*. Karena pada pelaksanaan ujian, akan terjadi keadaan dimana banyak mahasiswa akan mengakses aplikasi *computer assisted test* dalam waktu bersamaan. Dan *nodejs* mampu menangani hal tersebut lebih baik karena cara kerjanya yang bersifat *non blocking* sehingga penggunaan *resource* dapat lebih efisien.

## 3.2 Desain Sistem Baru

Salah satu tujuan dari pengembangan sistem baru yaitu membenahi keku-rangan yang ada pada sistem lama. Dengan demikian, desain sistem baru haruslah lebih baik daripada sistem lama baik dari segi biaya maupun dari segi efisiensi.

Desain sistem baru terdiri atas 2 (dua) bagian yaitu desain sistem secara global atau secara umum dan desain sistem secara terinci.

### 3.2.1 Desain Secara Global

Desain secara global diartikan sebagai gambaran secara umum kepada peng-guna tentang sistem yang akan dikembangkan. Desain sistem secara global meru-pakan persiapan untuk membangun desain yang lebih rinci. Desain yang tidak da-pat dijelaskan secara global nantinya akan dijelaskan pada desain terinci. Dan pada penelitian ini, penulis akan menggunakan *Unified Modelling Language* atau UML dalam mendesain sistem secara global.

#### 3.2.1.1 Usecase Diagram

Use-case *diagram* merupakan gambaran dari aktifitas-aktifitas yang akan di-lakukan oleh pengguna sistem. Use-case *diagram* juga memberikan gambaran ten-tang bagaimana sistem berinteraksi dengan orang yang berada diluar sistem atau biasa disebut sebagai aktor. Daftar aktor pada desain sistem baru dapat dijelaskan pada Tabel 3.1 :

**Tabel 3.1 Defenisi 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.

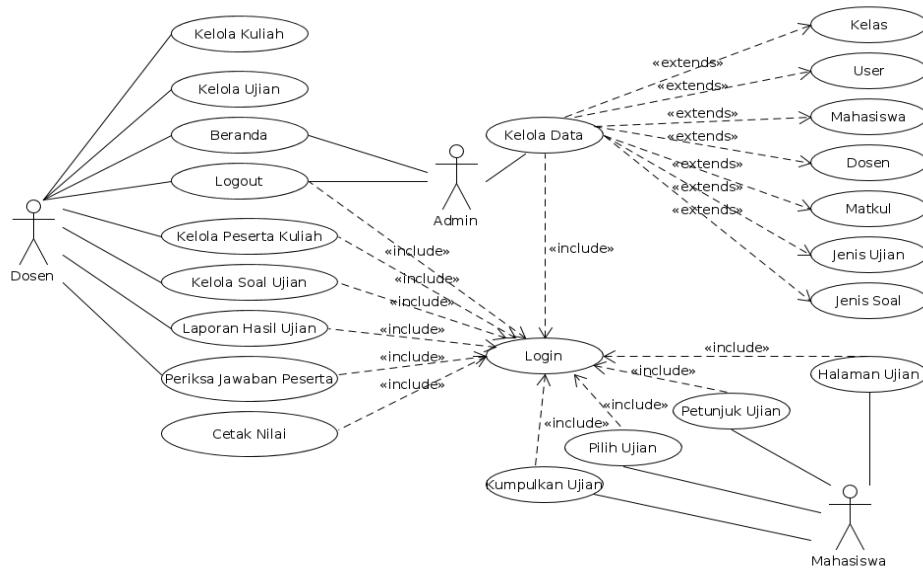
Adapun daftar *usecase* beserta aktornya pada desain sistem baru adalah sebagai berikut :

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 : Dosen
10. Kelola Soal Ujian : Dosen
11. Kelola Peserta Kuliah : Dosen
12. 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 : Dosen
21. Cetak Nilai : Dosen

Berdasarkan daftar *usecase* di atas, dapat digambarkan *usecase diagram*

seperti pada Gambar 3.3 :



**Gambar 3.3 Use Case Diagram**

### 3.2.1.2 Skenario Use Case

Skenario *usecase* merupakan gambaran detail sebuah proses yang terjadi pada suatu *usecase*, dimana aktifitas aktor dan sistem dalam menanggapi aktifitas aktor akan dijelaskan pada skenario *usecase*.

Pada skenario *usecase*, dijelaskan tentang kondisi awal aktor terhadap *usecase*, kemudian penjelasan detail dari skenario sukses sebuah *usecase* dan dilanjutkan dengan penjelasan skenario gagal sebuah *usecase*.

### 3.2.1.2.1 Skenario *Usecase* Beranda

Usecase beranda merupakan *usecase* yang diakses saat proses *login* berhasil. Beranda hanya akan dijumpai oleh aktor admin dan dosen. Agar lebih jelas mengenai skenario *usecase* beranda dapat dilihat pada Tabel 3.2 :

**Tabel 3.2 Skenario *Usecase* Beranda**

Identifikasi	
1	2
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	

1	2
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	

### 3.2.1.2.2 Skenario *UseCase Kelola User*

*UseCase* kelola user bertujuan untuk mengelola data pengguna sistem dimana hanya pengguna terdaftar yang dapat masuk ke sistem. Jika muncul suatu keadaan dimana dibutuhkan lebih banyak admin, maka proses penambahan admin baru dapat dilakukan di *usecase* kelola user. Agar lebih jelas mengenai skenario *usecase* kelola user dapat dilihat pada Tabel 3.3 :

**Tabel 3.3 Skenario Usecase Kelola User**

Identifikasi	
1	2
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	
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	

1	2
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	

### 3.2.1.2.3 Skenario *Usecase Kelola Mahasiswa*

*Usecase* kelola mahasiswa bertujuan untuk mengelola data mahasiswa. Data mahasiswa dikelola oleh admin dan jika terdapat mahasiswa baru, maka data mahasiswa baru tersebut dapat ditambahkan pada *usecase* kelola mahasiswa. Agar lebih jelas mengenai skenario *usecase* kelola mahasiswa dapat dilihat pada Tabel 3.4 :

**Tabel 3.4 Skenario *Usecase Kelola Mahasiswa***

Identifikasi	
1	2
No	3
Nama	Kelola mahasiswa
Tujuan	Aktor mengatur data mahasiswa

1	2
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, cari)	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	

### 3.2.1.2.4 Skenario Usecase Kelola Dosen

Use case kelola dosen bertujuan untuk mengelola data dosen. Admin bertugas mengelola data dosen tersebut. Jika terdapat dosen baru, maka admin harus menambahkan data dosen baru tersebut. Agar lebih jelas mengenai skenario usecase kelola dosen dapat dilihat pada Tabel 3.5 :

**Tabel 3.5 Skenario Usecase Kelola Dosen**

Identifikasi	
1	2
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

1	2
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	

### 3.2.1.2.5 Skenario *Usecase Kelola Matkul*

*Usecase* kelola mata kulliah bertujuan untuk mengelola data mata kulliah. Admin bertugas mengelola data mata kulliah tersebut. Jika terdapat mata kulliah baru, maka admin harus menambahkan data mata kulliah baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola mata kulliah dapat dilihat pada Tabel 3.6 :

**Tabel 3.6 Skenario *Usecase Kelola Matkul***

Identifikasi	
1	2
No	5

1	2
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

1	2
Submit/Ok kembali	Informasi kelola data matkul
Kondisi Akhir : Aktor berhasil mengelola data matkul	

### 3.2.1.2.6 Skenario *Usecase Kelola Jenis Soal*

*Usecase* kelola jenis soal bertujuan untuk mengelola data jenis soal. Admin bertugas mengelola data jenis soal tersebut. Jika terdapat jenis soal baru, maka admin harus menambahkan data jenis soal baru tersebut jika diperlukan. Agar lebih jelas mengenai skenario *usecase* kelola jenis soal dapat dilihat pada Tabel 3.7 :

**Tabel 3.7 Skenario *Usecase Kelola Jenis Soal***

Identifikasi	
1	2
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	

1	2
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	

### 3.2.1.2.7 Skenario *Usecase Kelola Jenis Ujian*

*Usecase* kelola jenis ujian bertujuan untuk mengelola data jenis ujian. Admin bertugas mengelola data jenis ujian tersebut. Jika terdapat jenis ujian baru, maka admin harus menambahkan data jenis ujian baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola jenis ujian dapat dilihat pada Tabel 3.8 :

**Tabel 3.8 Skenario Usecase Kelola Jenis Ujian**

Identifikasi	
1	2
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	
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	

1	2
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	

### 3.2.1.2.8 Skenario *Usecase Kelola Kelas*

*Usecase* kelola kelas bertujuan untuk mengelola data kelas. Admin bertugas mengelola data kelas tersebut. Jika terdapat kelas baru, maka admin harus menambahkan data kelas baru tersebut. Agar lebih jelas mengenai skenario *usecase* kelola kelas dapat dilihat pada Tabel 3.9 :

**Tabel 3.9 Skenario *Usecase Kelola Kelas***

Identifikasi	
1	2
No	8
Nama	Kelola Kelas
Tujuan	Aktor mengatur data Kelas
Deskripsi	Aktor membaca, mengedit, membuat dan menghapus data Kelas

1	2
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	

### 3.2.1.2.9 Skenario Usecase Kelola Ujian

*Usecase* kelola ujian dilakukan oleh aktor dosen. Dosen dapat membuat ujian baru dan nanti ujian tersebut akan diikuti oleh mahasiswa sesuai dengan kuliahan yang diambil. Agar lebih jelas mengenai skenario *usecase* kelola ujian dapat dilihat pada Tabel 3.10 :

**Tabel 3.10 Skenario Usecase Kelola Ujian**

Identifikasi	
1	2
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

1	2
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	

### 3.2.1.2.10 Skenario *Usecase Kelola Soal Ujian*

*Usecase* kelola soal ujian dilakukan oleh dosen. Dosen dapat menambahkan soal ujian baru dengan tipe soal essai dan pilihan ganda. Agar lebih jelas mengenai skenario *usecase* kelola soal ujian dapat dilihat pada Tabel 3.11 :

**Tabel 3.11 Skenario *Usecase Kelola Soal Ujian***

Identifikasi	
1	2
No	10
Nama	Kelola soal ujian

1	2
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

1	2
Kondisi Akhir : Aktor berhasil mengelola data soal ujian	

### 3.2.1.2.11 Skenario Usecase Kelola Peserta Kuliah

*Usecase* kelola peserta kuliah bertujuan untuk mengelola peserta mahasiswa yang mengikuti suatu kuliah. Peserta kuliah nantinya akan otomatis menjadi peserta ujian. Agar lebih jelas mengenai skenario *usecase* kelola peserta kuliah dapat dilihat pada Tabel 3.12 :

**Tabel 3.12 Skenario Usecase Kelola Peserta Kuliah**

Identifikasi	
1	2
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	

1	2
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	

### 3.2.1.2.12 Skenario *UseCase* Hasil Ujian

*UseCase* hasil ujian merupakan kegiatan dosen melihat hasil ujian peserta kuliah. Dosen masih perlu melakukan pemeriksaan jawaban jika tipe soal ujian adalah essai. Sedangkan tipe soal pilihan ganda akan menghasilkan nilai pada halaman hasil ujian secara otomatis. Agar lebih jelas mengenai skenario *usecase* hasil ujian dapat dilihat pada Tabel 3.13 :

**Tabel 3.13 Skenario Usecase Hasil Ujian**

Identifikasi	
1	2
No	12
Nama	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	

1	2
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	

### 3.2.1.2.13 Skenario UseCase Periksa Jawaban Peserta

Skenario *usecase* berikut menjelaskan proses dosen dalam melakukan pemeriksaan jawaban peserta. Pemeriksaan jawaban hanya berlaku jika tipe soal ujian adalah essai. Agar lebih jelas mengenai skenario *usecase* periksa jawaban peserta dapat dilihat pada Tabel 3.14 :

**Tabel 3.14 Skenario UseCase Periksa Jawaban Peserta**

Identifikasi	
1	2
No	13
Nama	Periksa Jawaban
Tujuan	Aktor memeriksa jawaban ujian peserta
Deskripsi	Aktor membaca dan memberikan nilai bobot pada setiap jawaban peserta

1	2
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

1	2
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	

### 3.2.1.2.14 Skenario *Usecase Login*

*Usecase login* diakses oleh semua aktor. Karena hampir semua halaman pada aplikasi yang akan dikembangkan hanya bisa diakses oleh aktor mahasiswa, dosen dan admin. Agar lebih jelas mengenai skenario *usecase login* dapat dilihat pada Tabel 3.15 :

**Tabel 3.15 Skenario *Usecase Login***

Identifikasi	
1	2
No	14
Nama	Login

1	2
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	Reaksi Sistem
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

1	2
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	

### 3.2.1.2.15 Skenario UseCase Petunjuk Ujian

UseCase ini akan diakses oleh mahasiswa yang akan mengikuti ujian. Halaman ini bertujuan untuk memberi sedikit gambaran tentang penggunaan aplikasi *computer assisted test*. Agar lebih jelas mengenai skenario usecase petunjuk ujian dapat dilihat pada Tabel 3.16 :

**Tabel 3.16 Skenario UseCase Petunjuk Ujian**

Identifikasi	
1	2
No	15
Nama	Petunjuk Ujian
Tujuan	Aktor mengetahui petunjuk ujian

1	2
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 ke halaman 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 ke halaman petunjuk ujian

1	2
	Sistem gagal menampilkan petunjuk ujian
Login kembali	Sistem berhasil menampilkan petunjuk ujian
Kondisi Akhir : Aktor berhasil mengetahui petunjuk ujian	

### 3.2.1.2.16 Skenario *UseCase* Halaman Ujian

*UseCase* halaman ujian diakses oleh mahasiswa. Mahasiswa akan menjawab soal ujian pada halaman ini. Hanya mahasiswa yang sudah terdaftar yang dapat mengikuti ujian. Agar lebih jelas mengenai skenario *usecase* halaman ujian dapat dilihat pada Tabel 3.17 :

**Tabel 3.17 Skenario *UseCase* Halaman Ujian**

Identifikasi	
1	2
No	16
Nama	Halaman Ujian
Tujuan	Aktor mengikuti ujian
Deskripsi	Aktor menjawab pertanyaan yang diujikan

1	2
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

1	2
Aktor klik kumpulkan ujian	Sistem tidak berhasil mengumpulkan ujian
Klik kembali kumpulkan ujian	Sistem berhasil mengumpulkan ujian
Kondisi Akhir : Ujian peserta terlaksana	

### 3.2.1.2.17 Skenario *Usecase Kumpulkan Ujian*

*Usecase* ini dilakukan mahasiswa setelah semua soal terjawab atau waktu ujian sudah habis. Semua jawaban yang sudah dikumpulkan akan diproses untuk menghasilkan nilai ujian mahasiswa. Agar lebih jelas mengenai skenario *usecase* kumpulkan ujian dapat dilihat pada Tabel 3.18 :

**Tabel 3.18 Skenario *Usecase Periksa Kumpulkan Ujian***

Identifikasi	
1	2
No	17
Nama	Kumpulkan Ujian
Tujuan	Aktor mengumpulkan ujian
Deskripsi	Aktor mengumpulkan ujian setelah selesai ujian

1	2
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman ujian	
Aksi aktor	Reaksi Sistem
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	

### 3.2.1.2.18 Skenario *UseCase Pilih Ujian*

UseCase pilih ujian muncul setelah mahasiswa login. Pada *usecase* ini, mahasiswa dihadapkan pada daftar ujian yang bisa di ikuti. Mahasiswa hanya dapat melaksanakan 1 (satu) ujian dalam waktu bersamaan. Agar lebih jelas mengenai skenario *usecase* pilih ujian dapat dilihat pada Tabel 3.19 :

**Tabel 3.19 Skenario Usecase Pilih Ujian**

Identifikasi	
1	2
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	
Entri username dan password	
Sistem cek username dan password	
Sistem menampilkan halaman pilih ujian	
Aktor memilih ujian	
Skenario Gagal	

1	2
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 mencocokkan username dan password
	Sistem menampilkan halaman pilih ujian
Aktor memilih ujian	
Kondisi Akhir : Aktor berhasil memilih ujian	

### 3.2.1.2.19 Skenario Usecase Logout

Use case ini dilakukan oleh admin dan dosen. Aktor dianjurkan untuk melakukan *logout* saat mereka tidak akan menggunakan aplikasi *computer assisted test* dalam waktu tertentu agar lebih aman terhadap orang yang tidak bertanggung jawab. Agar lebih jelas mengenai skenario usecase *logout* dapat dilihat pada Tabel 3.20 :

**Tabel 3.20 Skenario Usecase Periksa Jawaban Peserta**

Identifikasi	
1	2
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	
Klik tombol logout	
Sistem membawa aktor keluar sistem	
Sistem menampilkan halaman login	
Skenario Gagal	
Aksi Aktor	

1	2
Klik tombol logout	Sistem gagal membawa aktor keluar sistem
Klik tombol logout kemmbali	Sistem berhasil membawa aktor keluar
Kondisi Akhir : Aktor berhasil logout	

### 3.2.1.2.20 Skenario *Usecase Kelola Kuliah*

*Usecase* ini dilakukan oleh dosen untuk mengelola kuliah yang di ampu. Kuliah yang telah ditambahkan nantinya akan muncul pada saat dosen ingin menambahkan ujian. Agar lebih jelas mengenai skenario *usecase* kelola kuliah dapat dilihat pada Tabel 3.21 :

**Tabel 3.21 Skenario *Usecase Kelola Kuliah***

Identifikasi	
1	2
No	20
Nama	Kelola kuliah
Tujuan	Aktor mengatur data kuliah
Deskripsi	Aktor membaca, mengedit, membuat dan menghapus data kuliah

1	2
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman beranda	
Aksi aktor	Reaksi Sistem
Aktor mengelola data kuliah (simpan, tambah, hapus, cari)	Validasi data kuliah dan sistem menyimpan data kelola kuliah
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	

### 3.2.1.2.21 Skenario *Usecase Cetak Nilai*

*Usecase* ini bertujuan untuk mencetak hasil ujian mahasiswa dalam bentuk *file* seperti *file excel*. Nantinya *file* tersebut bisa digunakan oleh dosen untuk dio-

lah lebih lanjut. Agar lebih jelas mengenai skenario *usecase* cetak nilai dapat dilihat pada Tabel 3.22 :

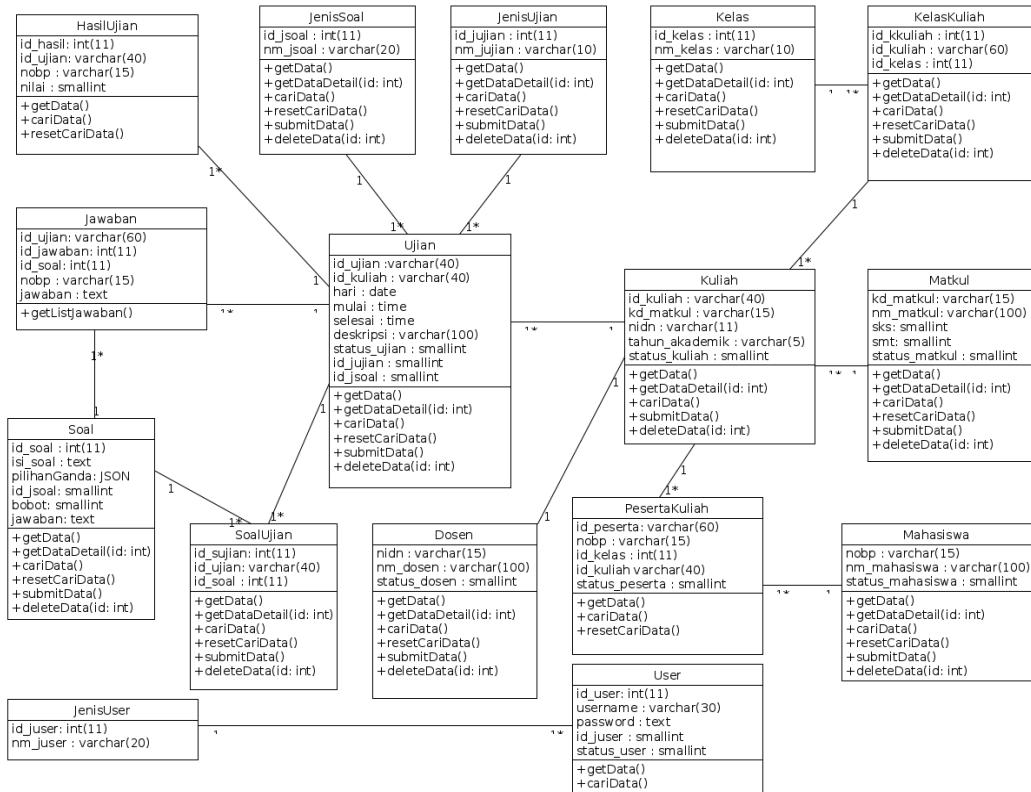
**Tabel 3.22 Skenario Usecase Kelola Kuliah**

Identifikasi	
1	2
No	21
Nama	Cetak Nilai
Tujuan	Aktor Mencetak Nilai kedalam bentuk file
Deskripsi	Aktor mencetak nilai ujian mahasiswa
Tipe :-	
Aktor : Dosen	
Skenario Utama	
Kondisi awal : Halaman Hasil Ujian	
Aksi aktor	
Aktor mengklik tombol cetak dan memilih pilihan file untuk dicetak	Sistem memberikan file nilai yang diminta aktor
Skenario Gagal	

1	2
Aksi Aktor	Reaksi Sistem
Aktor mengklik tombol cetak dan memilih pilihan file untuk dicetak	Sistem gagal memberikan file nilai yang diminta aktor
Aktor mengklik kembali tombol cetak dan memilih pilihan file untuk dicetak	Sistem memberikan file nilai yang diminta aktor
Kondisi Akhir : Aktor berhasil mendapatkan file nilai mahasiswa	

### 3.2.1.3 *Class Diagram*

*Class diagram* menggambarkan struktur *class* serta deskripsi *class* dan *object* serta hubungan antar data-data yang terdapat pada basis data. Agar lebih jelas mengenai *class diagram* pada aplikasi yang akan dikembangkan dapat dilihat pada Gambar 3.4 :



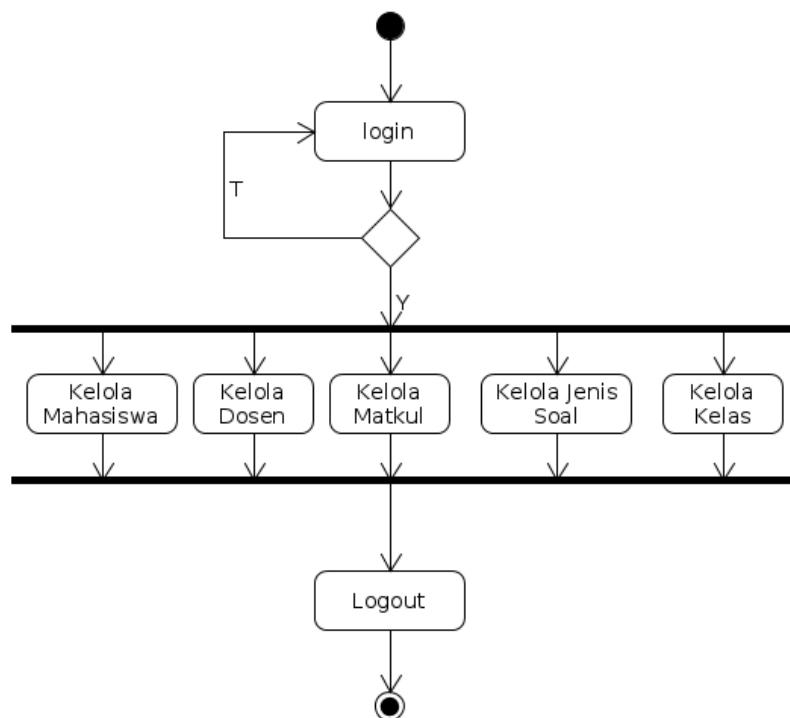
**Gambar 3.4 Class Diagram**

### 3.2.1.4 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. Agar lebih jelas mengenai *activity diagram* pada penelitian ini dapat dijelaskan sebagai berikut :

### 3.2.1.4.1 *Activity Diagram Admin*

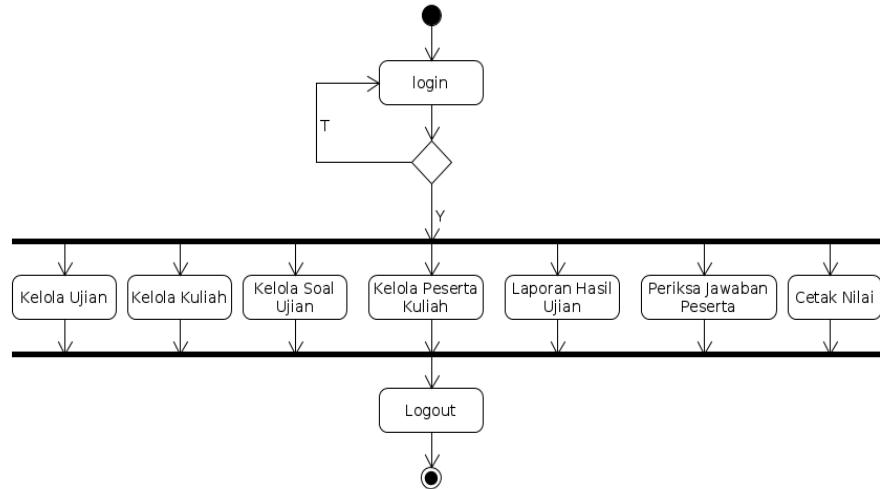
Seorang admin biasanya mengelola data utama dari sebuah sistem. Data utama tersebut nantinya akan digunakan oleh aktor lainnya. Adapun *activity diagram* seorang admin dapat digambarkan seperti Gambar 3.5 :



**Gambar 3.5 Activity Diagram Admin**

### 3.2.1.4.2 *Activity Diagram Dosen*

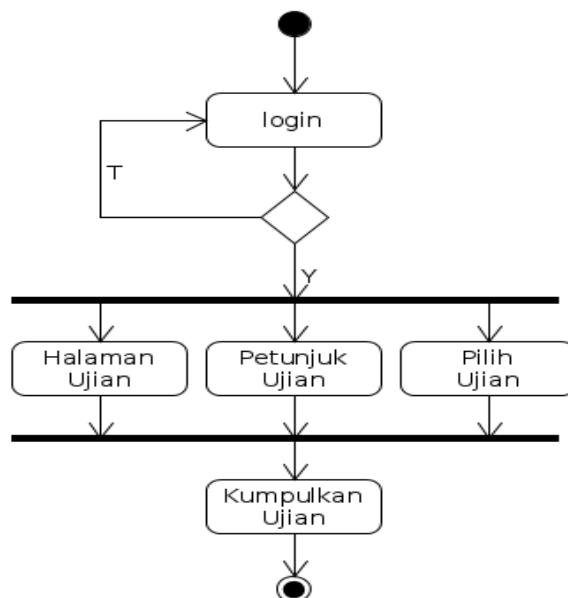
Pada sistem baru, dosen nantinya bertugas mengelola kuliah dan ujian yang telah Ia daftarkan. Dan juga dosen akan melakukan pemeriksaan terhadap jawaban peserta ujian. Adapun *activity diagram* seorang dosen dapat digambarkan seperti Gambar 3.6 :



**Gambar 3.6 Activity Diagram Dosen**

#### 3.2.1.4.3 Activity Diagram Mahasiswa

Mahasiswa merupakan aktor yang akan mengikuti ujian yang telah dibuat oleh seorang dosen. Adapun *activity diagram* seorang mahasiswa dapat digambar seperti Gambar 3.7 :



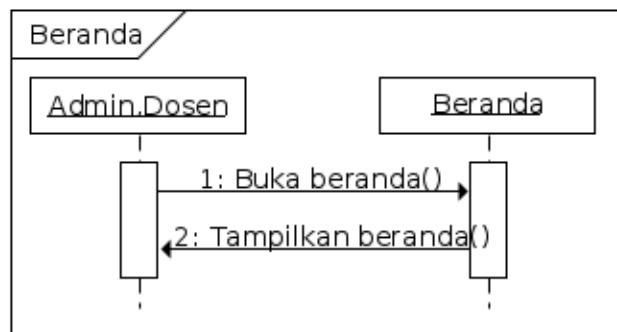
**Gambar 3.7 Activity Diagram Mahasiswa**

### 3.2.1.5 Sequence Diagram

*Sequence diagram* merupakan gambaran detail dari aktifitas seorang aktor pada sebuah *usecase*. Aktifitas-aktifitas itu sendiri dimulai dari aktor dan kemudian sistem membalas aktifitas tersebut. Bisa berupa menampilkan tampilan lain atau menampilkan informasi tertentu. Adapun adalah rancangan dari *sequence diagram* pada penelitian ini adalah sebagai berikut :

#### 3.2.1.5.1 Sequence Diagram Beranda

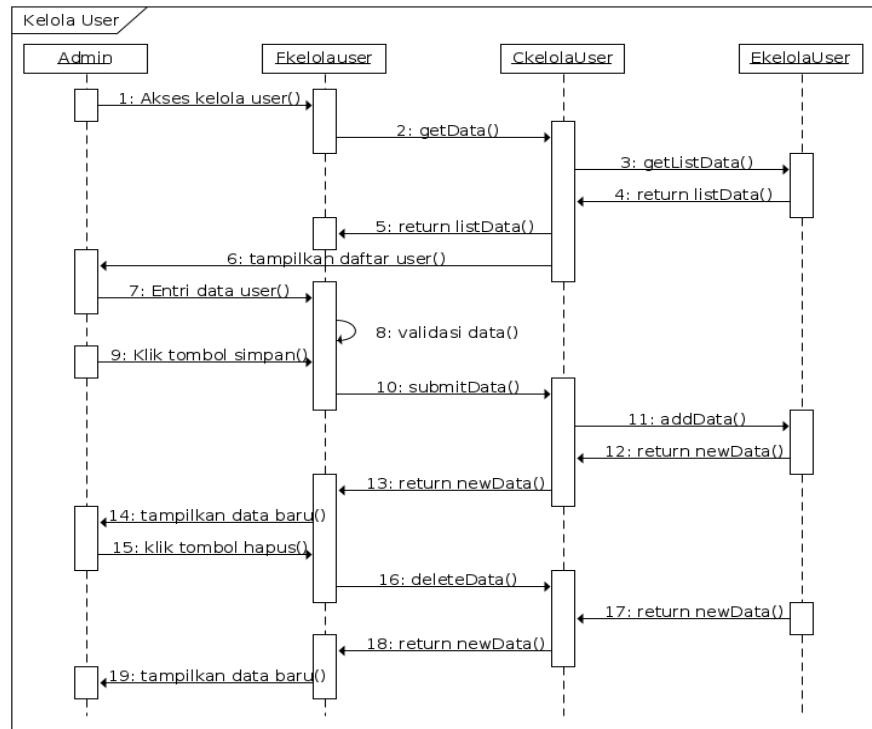
Diagram ini menggambarkan bagaimana aktor mengakses halaman beranda. Halaman ini nantinya akan muncul pertama kali saat dosen dan admin berhasil login. Adapun *sequence diagram* beranda dapat digambarkan seperti pada Gambar 3.8 :



**Gambar 3.8 Sequence Diagram Beranda**

#### 3.2.1.5.2 Sequence Diagram Kelola User

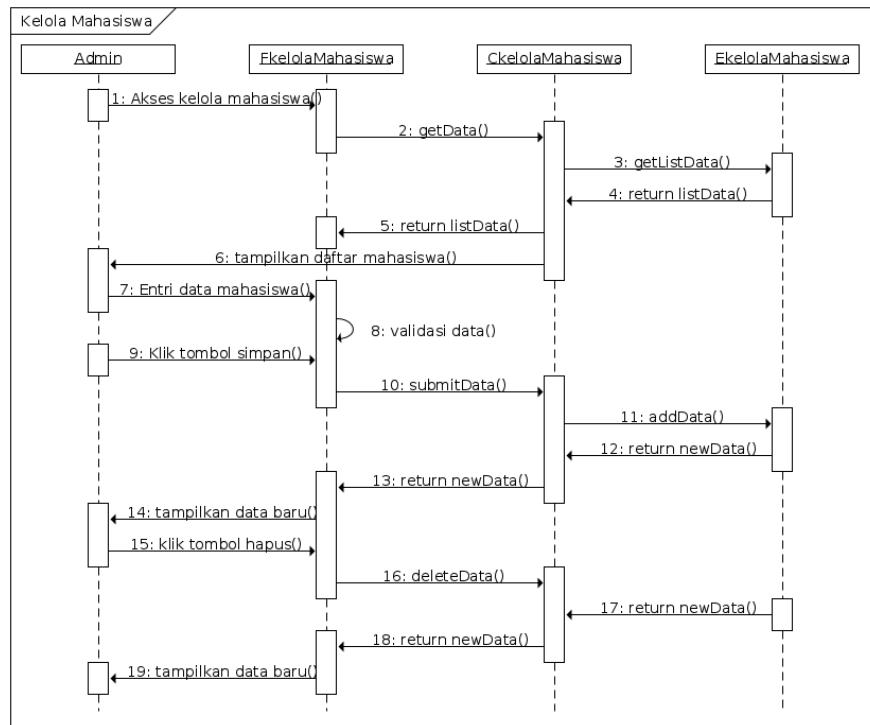
Sequence diagram ini menggambarkan tentang aktifitas admin dalam mengelola *user*. Admin dapat mengatur siapa saja yang dapat masuk kesistem melalui halaman ini. Adapun *sequence diagram* kelola user yang dapat digambarkan seperti pada Gambar 3.9 :



**Gambar 3.9 Sequence Diagram Kelola User**

### 3.2.1.5.3 Sequence Diagram Kelola Mahasiswa

*Sequence diagram* ini menjelaskan tentang bagaimana aktifitas admin dalam mengelola mahasiswa. Nantinya mahasiswa ini akan mengikuti ujian yang dibuat oleh dosen. Adapun *sequence diagram* kelola mahasiswa dapat digambarkan seperti Gambar 3.10 :

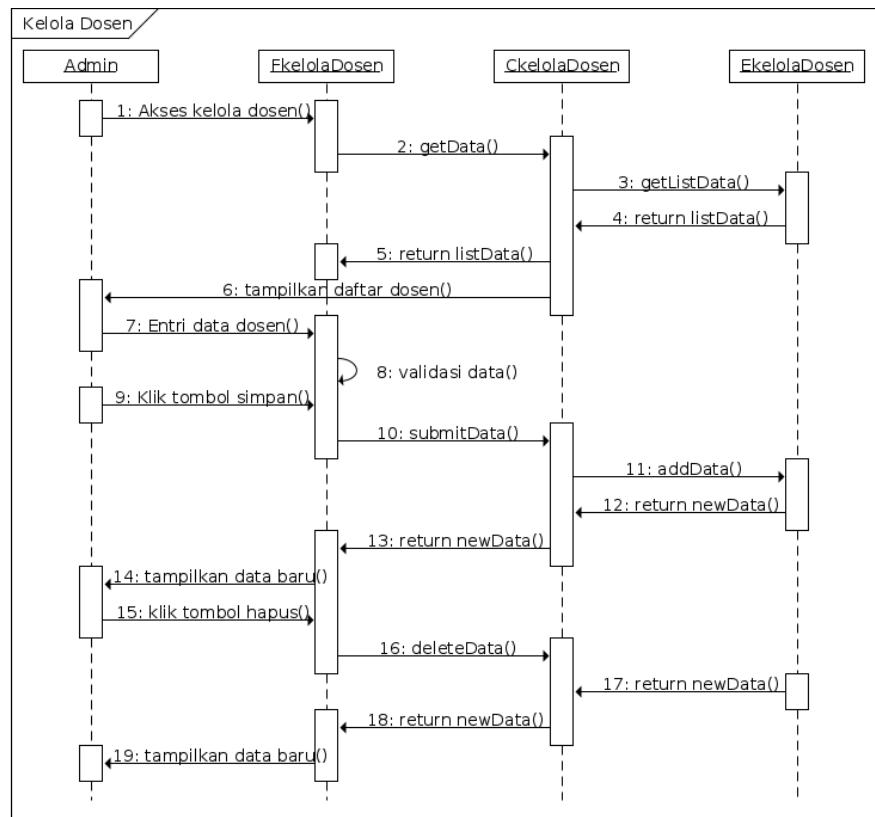


**Gambar 3.10 Sequence Diagram kelola mahasiswa**

#### 3.2.1.5.4 Sequence Diagram Kelola Dosen

Diagram ini menggambarkan aktifitas admin dalam mengelola data dosen.

Dosen yang sudah terdaftar dapat menggunakan aplikasi *CAT* ini. Setiap dosen baru ditambahkan, data dosen tersebut akan bertambah otomatis pada data user sehingga dosen dapat langsung masuk ke sistem. Adapun *sequence diagram* kelola dosen pada Gambar 3.11 :



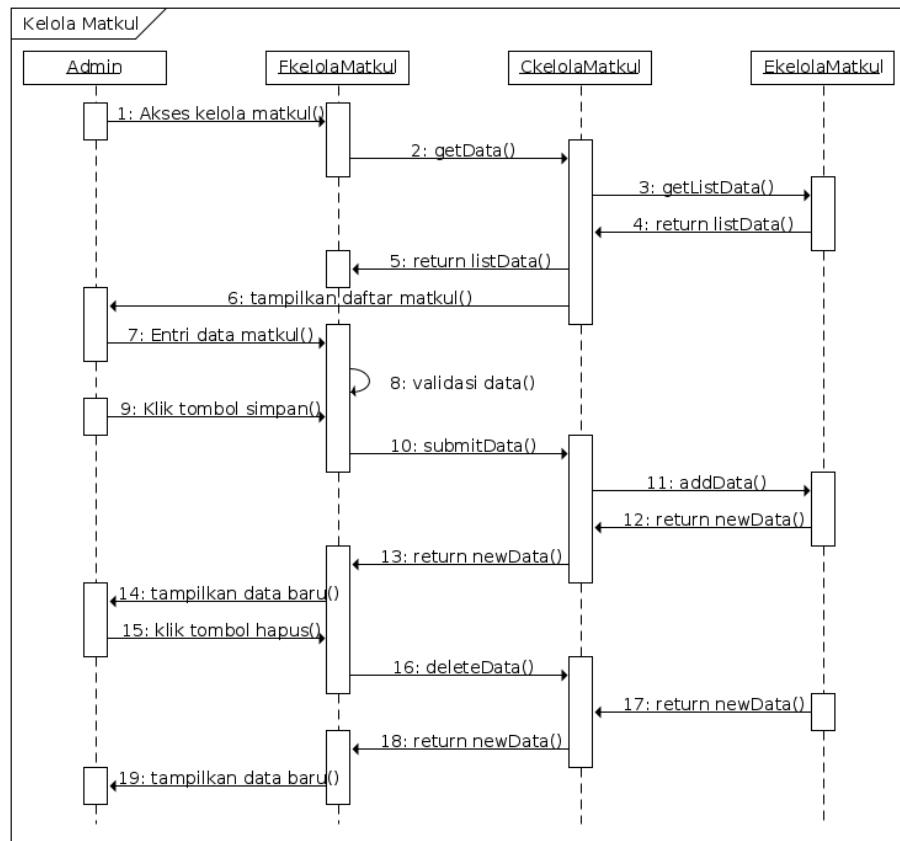
**Gambar 3.11 Sequence Diagram kelola dosen**

### 3.2.1.5.5 Sequence Diagram Kelola Mata Kuliah

Diagram ini menggambarkan detail proses dari mengelola mata kuliah.

Dimulai dari proses pengambilan daftar mata kuliah hingga proses penyimpanan mata kuliah. Adapun *sequence diagram* kelola mata kuliah dapat dilihat pada

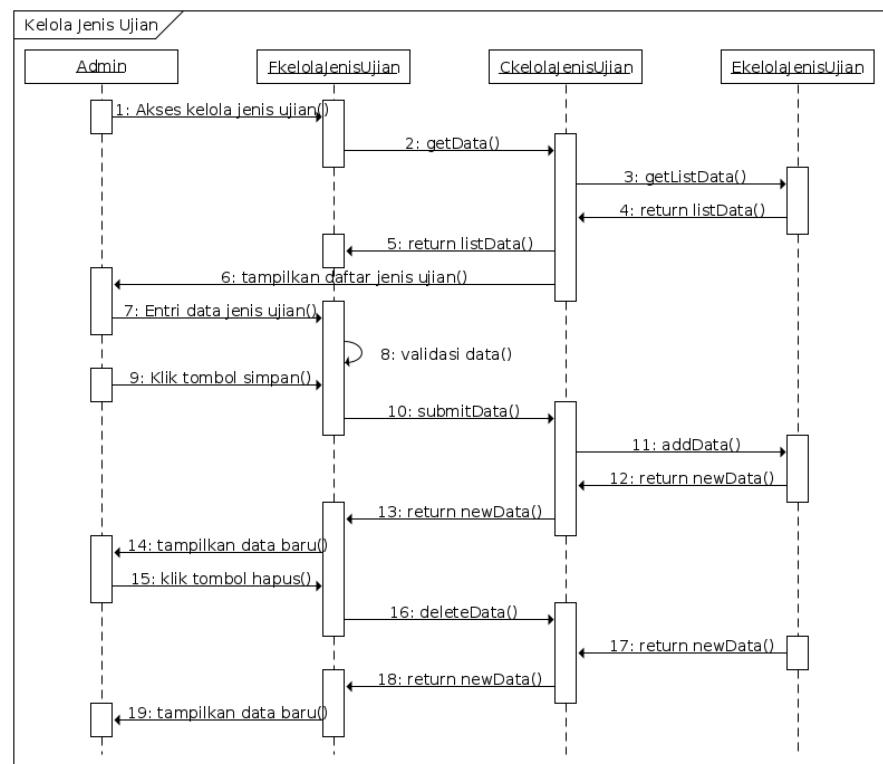
Gambar 3.12 :



**Gambar 3.12 Sequence Diagram kelola mata kuliah**

### 3.2.1.5.6 Sequence Diagram Kelola Jenis Ujian

Diagram ini menggambarkan detail proses dari kegiatan mengelola jenis ujian. Dimulai dari proses pengambilan daftar jenis ujian yang sudah ditambahkan hingga proses penyimpanan jenis ujian baru ke basis data. Adapun *sequence diagram* kelola jenis ujian dapat dilihat pada Gambar 3.13 :

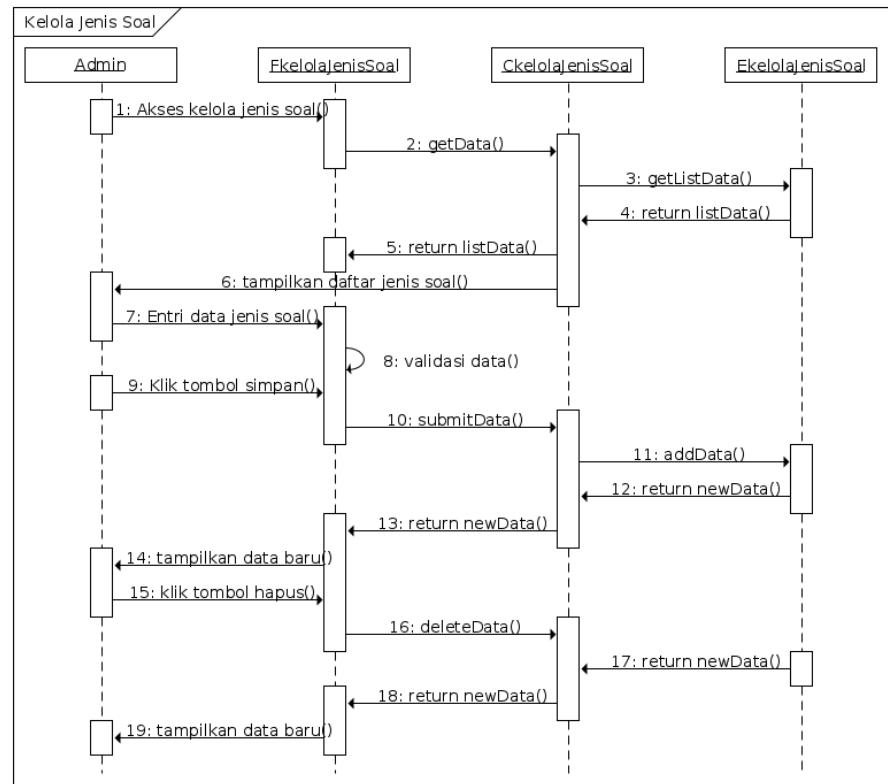


**Gambar 3.13 Sequence Diagram kelola jenis ujian**

### 3.2.1.5.7 Sequence Diagram Kelola Tipe Soal

Diagram ini menggambarkan detail proses dari kegiatan mengelola tipe soal.

Dimulai dari proses pengambilan daftar tipe soal yang sudah ditambahkan hingga proses penyimpanan tipe soal baru ke basis data. Adapun *sequence diagram* kelola tipe soal dapat dilihat pada Gambar 3.14 :

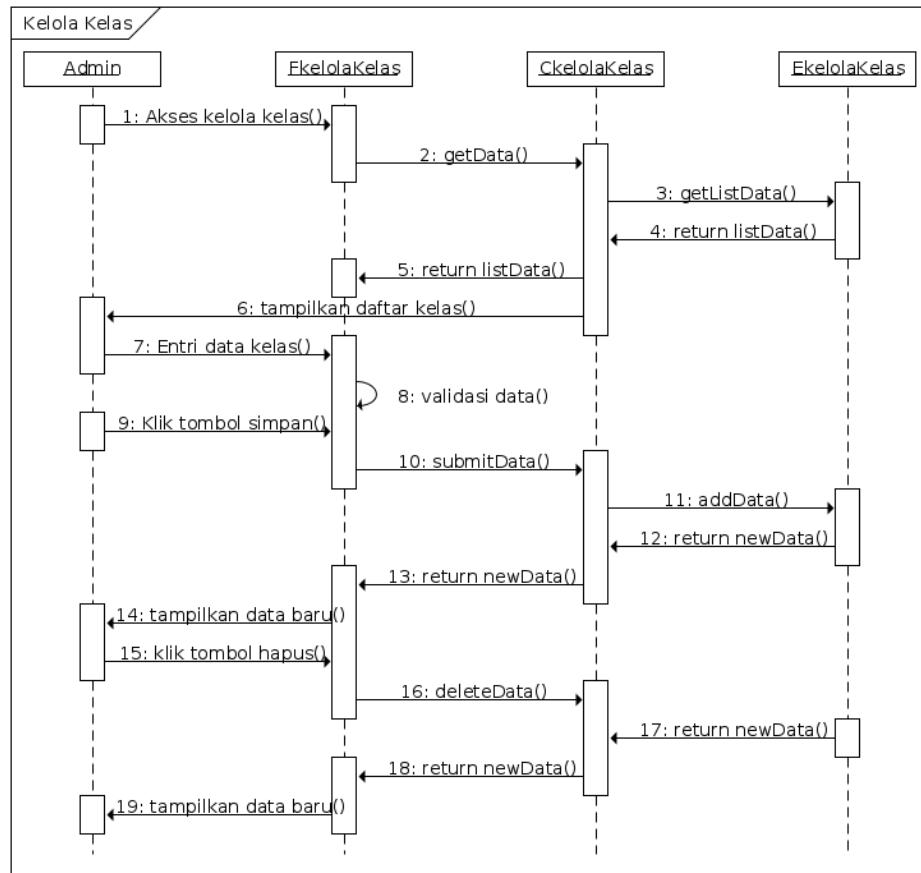


**Gambar 3.14 Sequence Diagram kelola tipe soal**

### 3.2.1.5.8 Sequence Diagram Kelola Kelas

Diagram ini menggambarkan detail proses dari kegiatan mengelola kelas.

Dimulai dari proses pengambilan daftar kelas yang sudah ditambahkan hingga proses penyimpanan kelas baru ke basis data. Adapun *sequence diagram* kelola kelas dapat dilihat pada Gambar 3.15 :

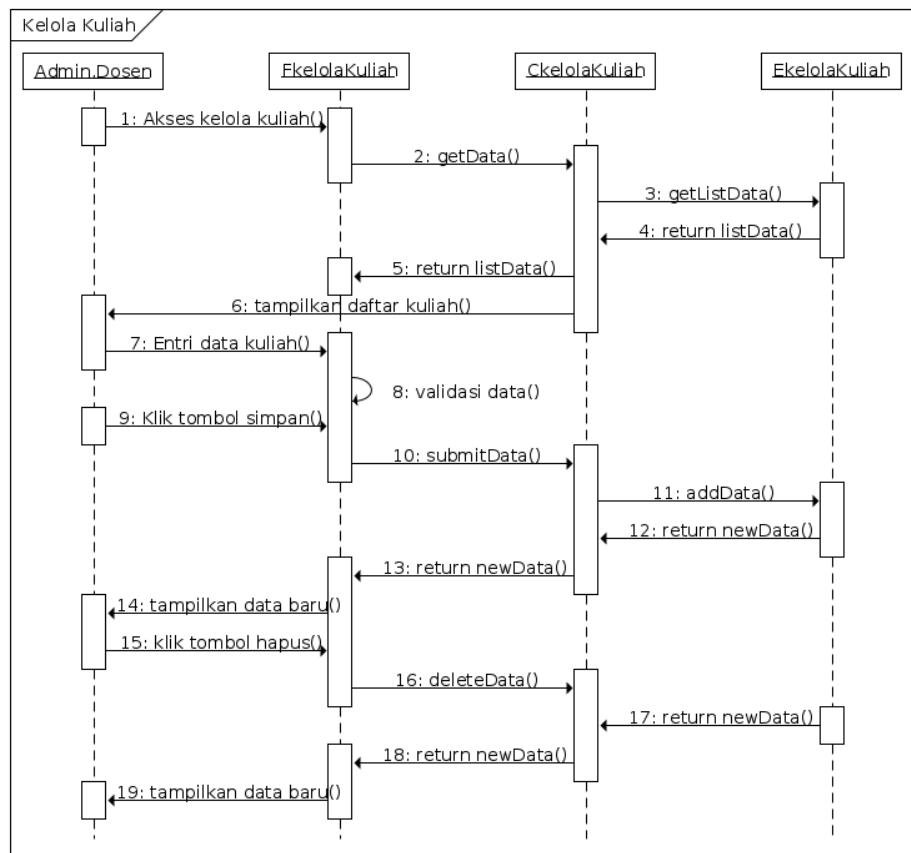


**Gambar 3.15 Sequence Diagram kelola kelas**

#### 3.2.1.5.9 Sequence Diagram Kelola Kuliah

Diagram ini menggambarkan detail proses dari kegiatan mengelola kuliah.

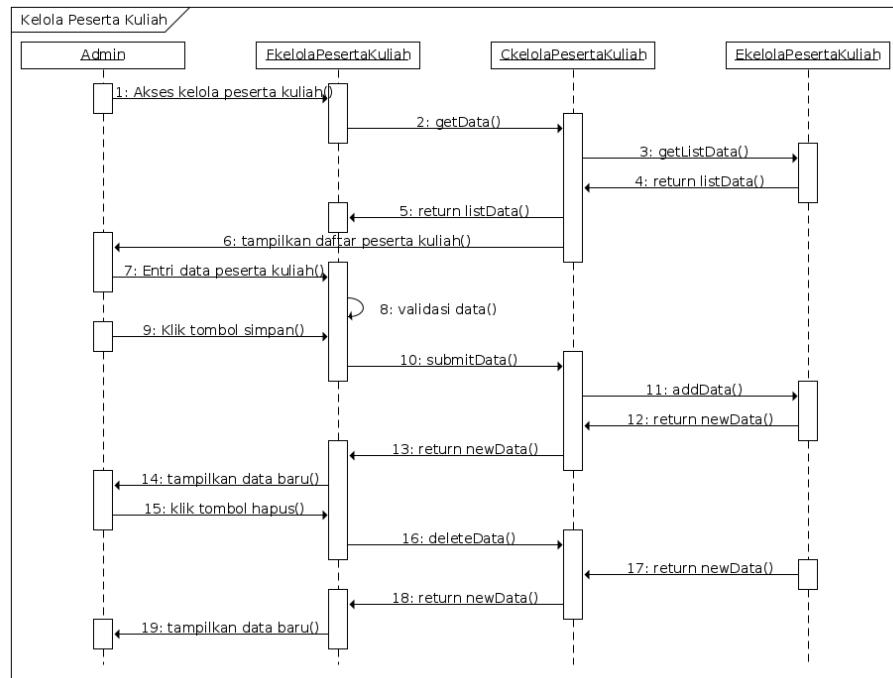
Dimulai dari proses pengambilan daftar kuliah yang sudah ditambahkan hingga proses penyimpanan kuliah baru ke basis data. Adapun *sequence diagram* kelola kuliah dapat dilihat pada Gambar 3.16 :



**Gambar 3.16 Sequence Diagram kelola kuliah**

#### 3.2.1.5.10 Sequence Diagram Kelola Peserta Kuliah

Diagram ini menggambarkan detail proses dari kegiatan mengelola jenis ujian. Dimulai dari proses pengambilan daftar jenis ujian yang sudah ditambahkan hingga proses penyimpanan jenis ujian baru ke basis data. Adapun *sequence diagram* kelola jenis ujian dapat dilihat pada Gambar 3.17 :

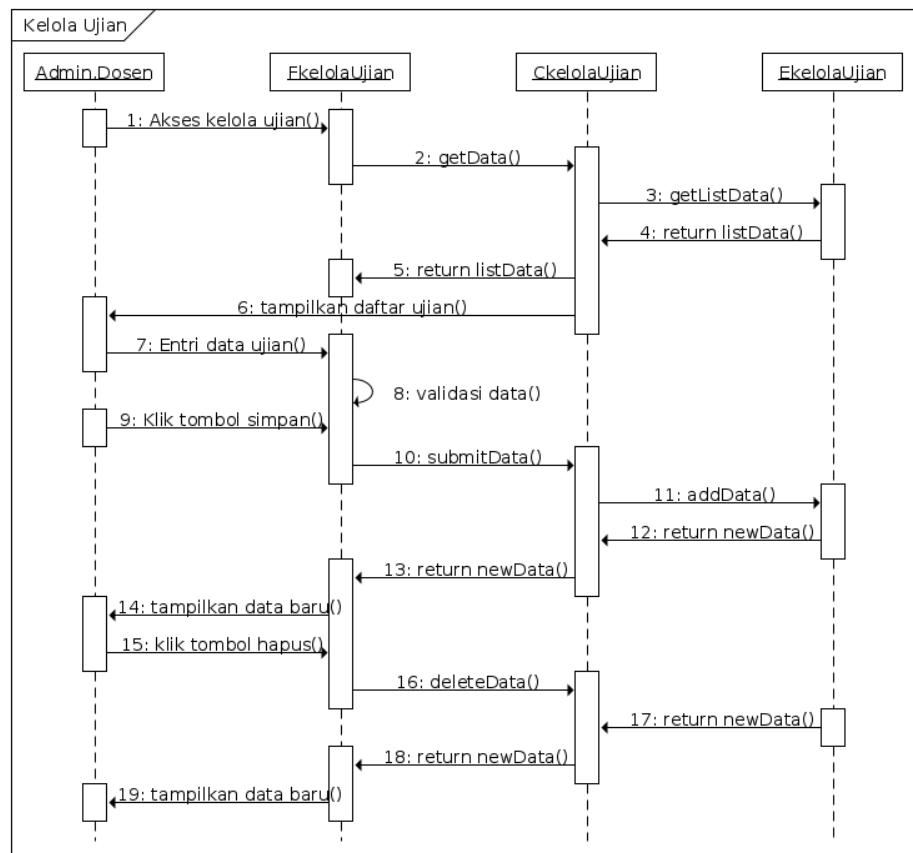


**Gambar 3.17 Sequence Diagram kelola peserta kuliah**

### 3.2.1.5.11 Sequence Diagram Kelola Ujian

Diagram ini menggambarkan detail proses dari kegiatan mengelola ujian.

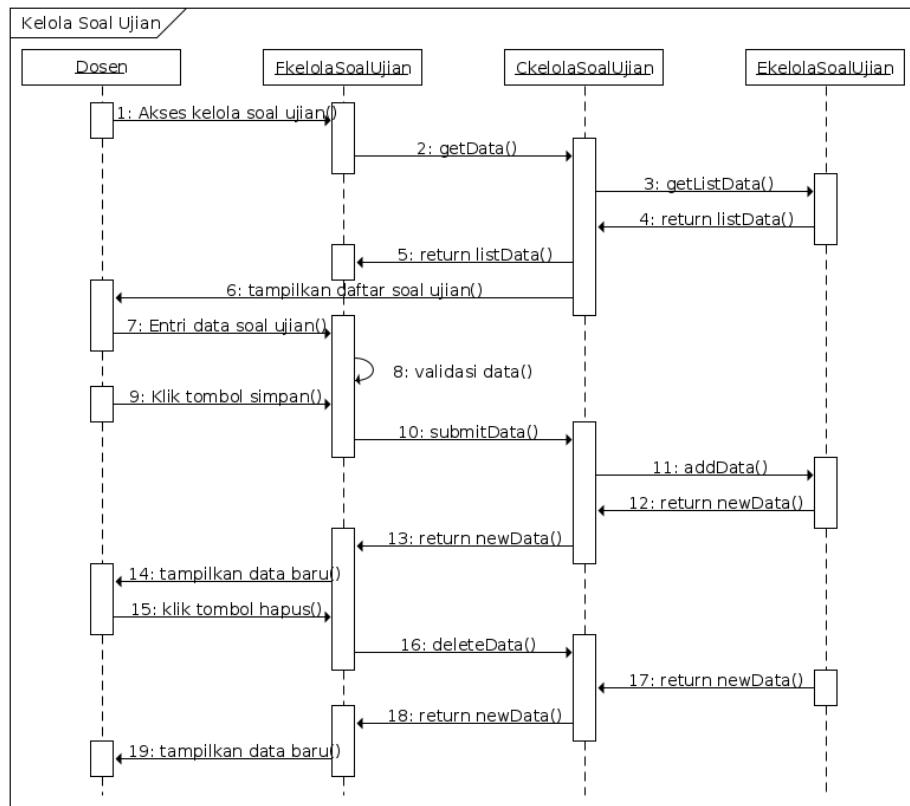
Dimulai dari proses pengambilan daftar ujian yang sudah ditambahkan hingga proses penyimpanan ujian baru ke basis data. Adapun *sequence diagram* kelola ujian dapat dilihat pada Gambar 3.18 :



**Gambar 3.18 Sequence Diagram kelola ujian**

#### 3.2.1.5.12 Sequence Diagram Kelola Soal Ujian

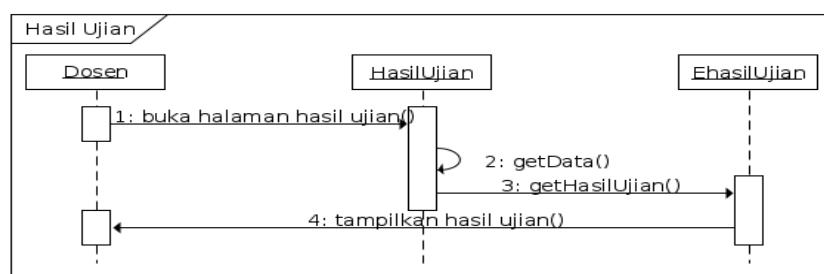
Diagram ini menggambarkan detail proses dari kegiatan mengelola soal ujian. Dimulai dari proses pengambilan daftar soal ujian yang sudah ditambahkan hingga proses penyimpanan soal ujian baru ke basis data. Adapun *sequence diagram* kelola soal ujian dapat dilihat pada Gambar 3.19 :



**Gambar 3.19 Sequence Diagram kelola soal ujian**

#### 3.2.1.5.13 Sequence Diagram Hasil Ujian

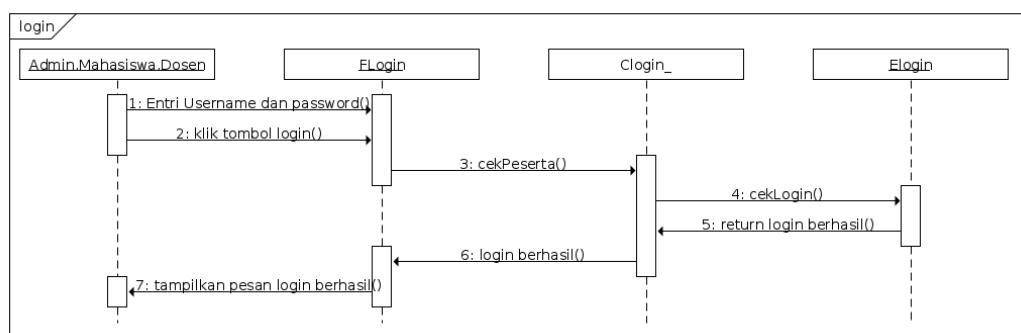
Diagram ini menggambarkan detail proses dari kegiatan mengelola hasil ujian. Dimulai dari proses pengambilan daftar hasil ujian yang sudah ditambahkan hingga proses penyimpanan hasil ujian baru ke basis data. Adapun *sequence diagram* kelola hasil ujian dapat dilihat pada Gambar 3.20 :



**Gambar 3.20 Sequence Diagram hasil ujian**

### 3.2.1.5.14 Sequence Diagram Login

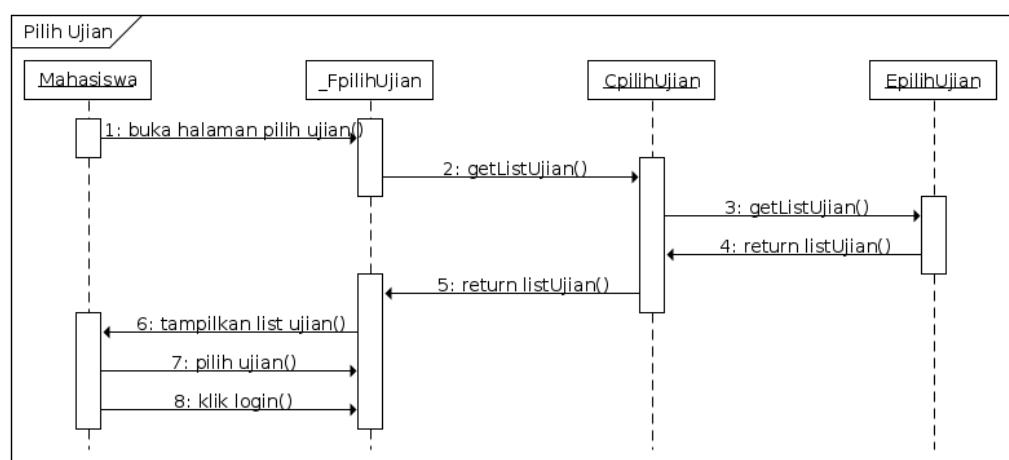
Diagram ini menggambarkan proses seorang pengguna melakukan *login* ke sistem. Dimulai dari mengetikkan *username* dan *password* hingga proses pengecekan hak akses pengguna. Adapun *sequence diagram login* dapat digambarkan seperti Gambar 3.21 :



**Gambar 3.21 Sequence Diagram login**

### 3.2.1.5.15 Sequence Diagram Pilih Ujian

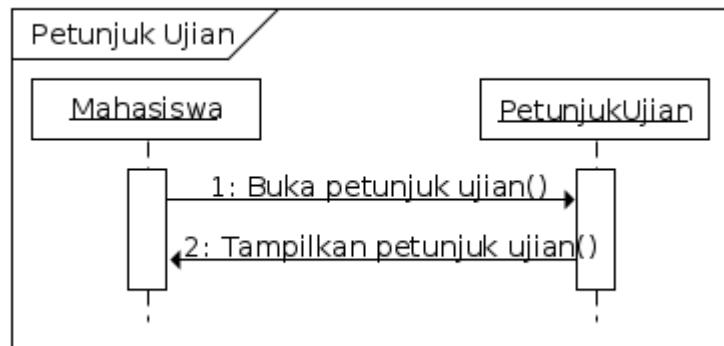
Diagram ini menggambarkan proses mahasiswa dalam memilih ujian yang ingin di ikuti. Adapun *sequence diagram pilih ujian* dapat digambarkan seperti Gambar 3.22 :



**Gambar 3.22 Sequence Diagram pilih ujian**

### 3.2.1.5.16 Sequence Diagram Petunjuk Ujian

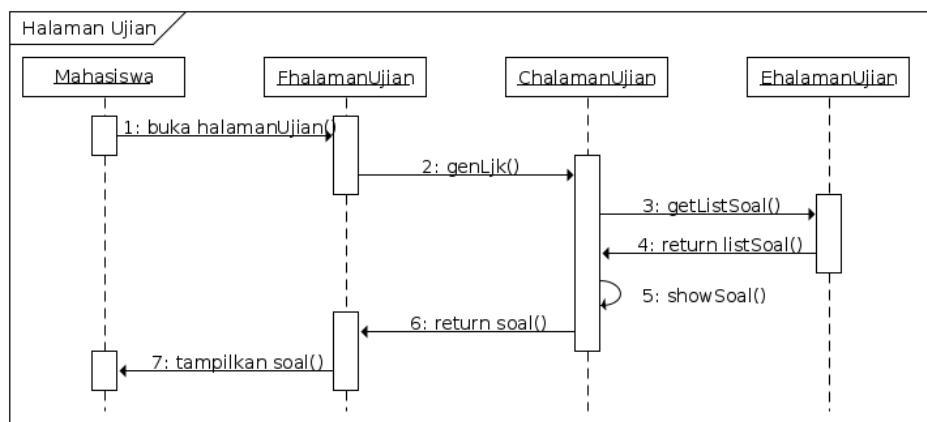
Diagram ini menggambarkan proses seorang mahasiswa dalam mengakses halaman petunjuk ujian. Adapun *sequence diagram* petunjuk ujian dapat digambarkan seperti Gambar 3.23 :



**Gambar 3.23 Sequence Diagram petunjuk ujian**

### 3.2.1.5.17 Sequence Diagram Halaman Ujian

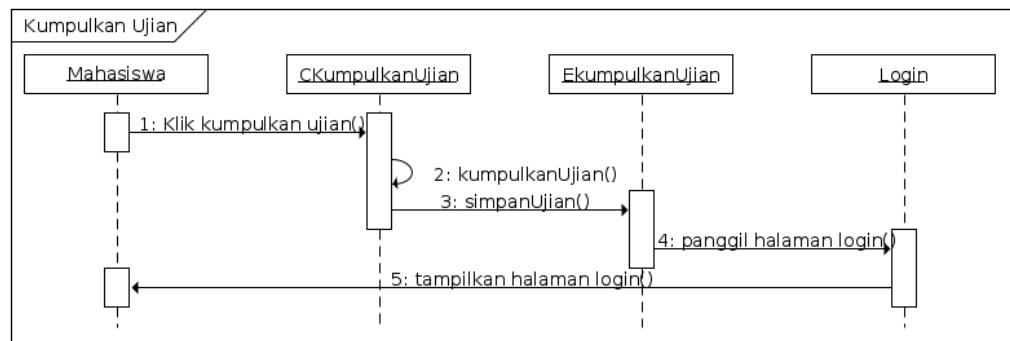
Diagram ini menggambarkan proses seorang mahasiswa dalam mengerjakan ujian pada aplikasi yang akan dikembangkan. Adapun *sequence diagram* halaman ujian dapat digambarkan seperti Gambar 3.24 :



**Gambar 3.24 Sequence Diagram halaman ujian**

### 3.2.1.5.18 Sequence Diagram Kumpulkan Ujian

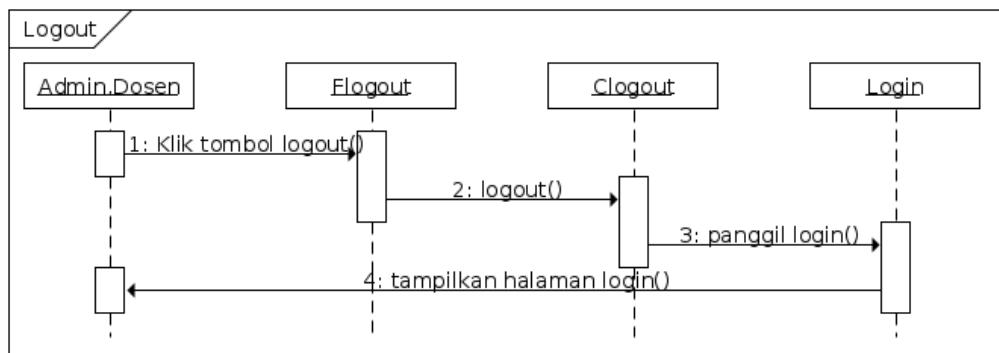
Diagram ini menggambarkan proses seorang mahasiswa pada saat mengumpulkan ujian. Adapun *sequence diagram* kumpulkan ujian dapat digambarkan seperti Gambar 3.25 :



**Gambar 3.25 Sequence Diagram kumpulkan ujian**

### 3.2.1.5.19 Sequence Diagram Logout

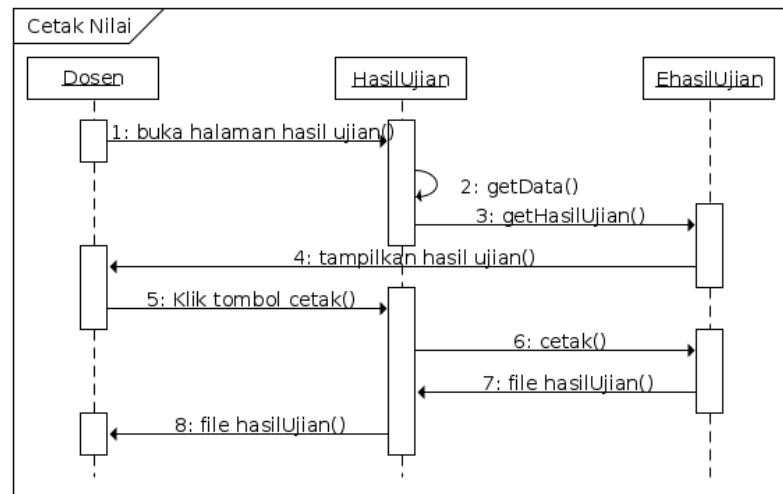
Diagram ini menggambarkan proses pengguna dalam melakukan *logout* atau keluar dari sistem. Adapun *sequence diagram* logout dapat digambarkan seperti Gambar 3.26 :



**Gambar 3.26 Sequence Diagram logout**

### 3.2.1.5.20 Sequence Diagram Cetak Nilai

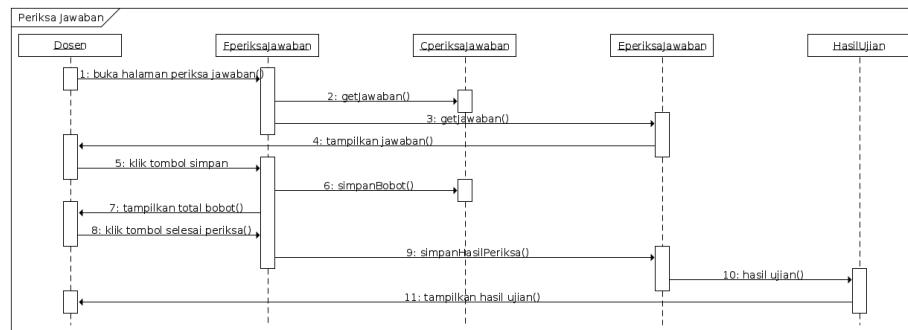
Diagram ini menggambarkan proses seorang dosen dalam mengakses nilai ujian mahasiswa serta mencetaknya ke dalam bentuk *file*. Adapun *sequence diagram* cetak nilai dapat digambarkan seperti Gambar 3.27 :



**Gambar 3.27 Sequence Diagram cetak nilai**

### 3.2.1.5.21 Sequence Diagram Periksa Jawaban

Diagram ini menunjukkan proses seorang dosen dalam melakukan pemeriksaan jawaban peserta ujian dengan tipe soal essai. Adapun *sequence diagram* periksa ujian dapat digambarkan seperti Gambar 3.28 :



**Gambar 3.28 Sequence Diagram periksa jawaban**

### 3.2.2 Desain Terinci

Desain terinci merupakan pengembangan lebih lanjut dari desain global. Desain terinci terbagi atas 3 (tiga) macam yaitu desain *output*, *input* dan *file*.

#### 3.2.2.1 Desain Output

Desain *output* merupakan desain atau tampilan dari hasil proses sistem. *Output* dapat ditampilkan pada monitor atau dicetak kedalam bentuk kertas. Desain *output* dari aplikasi yang akan dikembangkan dapat dijelaskan sebagai berikut :

##### 1. Daftar Mahasiswa

Halaman ini digunakan untuk melihat daftar mahasiswa secara keseluruhan, dengan bentuk rancangan seperti pada Gambar 3.29 :

Header				
Username	Daftar Mahasiswa			
	<input type="button" value="Tambah Data"/>	<input type="button" value="Refresh"/>	<input type="button" value="Pencarian ..."/>	<input type="button" value="Reset"/>
Menu	No	NOBP	Nama Mahasiswa	Aksi
	99 99	Varchar(15) Varchar(15)	Varchar(100) Varchar(100)	
<input type="button" value="Sebelumnya"/>		<input type="button" value="99"/>	<input type="button" value="Selanjutnya"/>	Banyak Data Perhalaman <input type="button" value="99"/>

**Gambar 3.3.29 Halaman Daftar Mahasiswa**

##### 2. Daftar Dosen

Halaman ini berisi daftar dosen secara keseluruhan, dengan bentuk rancangan seperti pada Gambar 3.30 :

Header		Daftar Dosen		
Username				
			Pencarian ...	Reset
Menu	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>			
	No	NIDN	Nama Dosen	Aksi
99	Varchar(15)	Varchar(100)		
99	Varchar(15)	Varchar(100)		
<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman		99

### **Gambar 3.30 Halaman Daftar Dosen**

### 3. Daftar Mata Kuliah

Halaman ini berisi daftar mata kuliah secara keseluruhan, dengan bentuk rancangan seperti pada Gambar 3.31 :

Header		Daftar Matkul						
Username	Tambah Data			Refresh		Pencarian ...		Reset
	No	Kode Matkul	Nama Matkul	Semester	SKS	Aksi		
99	Varchar(15)	Varchar(100)	Int(2)	Int(2)				
99	Varchar(15)	Varchar(100)	Int(2)	Int(2)				
<a href="#">Sebelumnya</a> <a href="#">99</a> <a href="#">Selanjutnya</a>			Banyak Data Perhalaman				<a href="#">99</a>	

**Gambar 3.31 Halaman Daftar Mata Kuliah**

#### 4. Daftar Kelas

Halaman ini berisi daftar kelas secara keseluruhan, dengan bentuk rangcangan seperti pada Gambar 3.32 :

Header			
Username	Daftar Kelas		
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>		
Menu	No	Nama Kelas	Aksi
	99	Varchar(10)	
	99	Varchar(10)	
<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>	

**Gambar 3.32 Halaman Daftar Kelas**

#### 5. Daftar Tipe Soal

Halaman ini berisi daftar tipe soal secara keseluruhan, dengan bentuk rancangan seperti pada Gambar 3.33 :

Header			
Username	Daftar Tipe Soal		
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>		
Menu	No	Nama Tipe Soal	Aksi
	99	Varchar(20)	
	99	Varchar(20)	
<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="button" value="99"/>	

**Gambar 3.33 Halaman Daftar Tipe Soal**

#### 6. Daftar Jenis Ujian

Halaman ini berisi daftar jenis ujian secara keseluruhan, dengan bentuk rancangan seperti pada Gambar 3.34 :

Header		Daftar Jenis Ujian		
Username	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/> <input type="text" value="Pencarian ..."/> <input type="button" value="Reset"/>			
	No	Nama Jenis Ujian	Aksi	
99	Varchar(10)	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>		
99	Varchar(10)	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>		

Banyak Data Perhalaman

### **Gambar 3.34 Halaman Daftar Jenis Ujian**

## 7. Daftar *User*

Halaman ini berisi daftar user secara keseluruhan, dengan bentuk rangcangan seperti pada Gambar 3.35 :

Header		Daftar User		
Username	Menu	<a href="#">Tambah Data</a> <a href="#">Refresh</a>		<a href="#">Pencarian ...</a> <a href="#">Reset</a>
		No	Username	Jenis User
99	Varchar(30)	Varchar(10)		
99	Varchar(30)	Varchar(10)		
<a href="#">Sebelumnya</a> <a href="#">99</a> <a href="#">Selanjutnya</a>		Banyak Data Perhalaman		<a href="#">99</a>

### Gambar 3.35 Halaman Daftar User

## 8. Daftar Kuliah

Halaman ini berisi daftar kuliah yang telah ditambahkan oleh dosen. Kuliah ini nantinya akan didaftarkan pada ujian, dengan bentuk rancangan seperti pada Gambar 3.36 :

Header				
Username	Daftar Kuliah			
	Tambah Data Refresh		Tahun Akademik	Varchar(5)
Menu	No	Mata Kuliah	Kelas	Status
	99 99	Varchar(100) Varchar(100)	Varchar(10) Varchar(10)	Varchar(10) Varchar(10)
Sebelumnya 99 Selanjutnya		Banyak Data Perhalaman 99		

**Gambar 3.36 Halaman Daftar Kuliah**

#### 9. Daftar Peserta Kuliah

Halaman ini berisi daftar peserta kuliah yang telah ditambahkan oleh dosen. Peserta kuliah ini nantinya akan mengikuti ujian yang telah dibuat dosen, dengan bentuk rancangan seperti pada Gambar 3.37 :

Header				
Username	Informasi Kuliah			
	Nama Dosen Mata Kuliah Kelas Tahun Akademik	Varchar(100) Varchar(100) Varchar(10) Varchar(5)		
Menu	Daftar Peserta Kuliah			
	Tambah Data Refresh		Pencarian ...	Reset
No	NOBP	Nama Mahasiswa	Aksi	
99 99	Varchar(15) Varchar(15)	Varchar(100) Varchar(100)	Aksi	
Sebelumnya 99 Selanjutnya		Banyak Data Perhalaman		99

**Gambar 3.37 Halaman Daftar Peserta Kuliah**

## 10. Daftar Ujian

Halaman ini berisi daftar ujian yang telah ditambahkan oleh dosen. Peserta kuliah nantinya akan mengikuti ujian yang telah dibuat dosen, dengan bentuk rancangan seperti pada Gambar 3.38 :

Header		Daftar Ujian					
Username	Menu	Tambah Data		Refresh	Tahun Akademik		VARCHAR(5)
Menu		No	Mata Kuliah	Jenis Ujian	Waktu	Status	Aksi
99	Varchar(100)	Varchar(10)	Date	Varchar(10)	Varchar(10)	Varchar(10)	
							99
Sebelumnya		99	Selanjutnya		Banyak Data Perhalaman		

### **Gambar 3.38 Halaman Daftar Ujian**

## 11. Daftar Soal Ujian

Halaman ini berisi daftar soal ujian yang telah ditambahkan oleh dosen. Soal ini nantinya akan keluar pada saat ujian secara acak, dengan bentuk rancangan seperti pada Gambar 3.39 :

Header		Informasi Ujian			
Username		Mata Kuliah	Varchar(100)	Dosen	Varchar(100)
Menu		Kelas	Varchar(10)	Tahun Akademik	Varchar(5)
		Hari	Varchar(15)	Waktu	Date
		Jenis Ujian	Varchar(5)	Jenis Soal	Varchar(5)
		Soal Ujian		Hasil Ujian	
	<input type="button" value="Tambah Data"/> <input type="button" value="Refresh"/>		<input type="button" value="Pencarian ..."/> <input type="button" value="Reset"/>		
No	Isi Soal	Bobot	Aksi		
99	<input type="text" value="Text"/>  	<input type="text" value="Int(5)"/>  			
99	<input type="text" value="Text"/>	<input type="text" value="Int(5)"/>			
<input type="button" value="Sebelumnya"/> <input type="button" value="99"/> <input type="button" value="Selanjutnya"/>		Banyak Data Perhalaman <input type="text" value="99"/>			

### **Gambar 3.39 Halaman Daftar Soal Ujian**

## 12. Daftar Hasil Ujian

Halaman ini berisi daftar hasil ujian peserta. Status ujian beserta nilai ujian mahasiswa akan ditampilkan dihalaman ini, dengan bentuk rancangan seperti pada Gambar 3.40 :

Header						
Username Menu	Informasi Ujian					
	Mata Kuliah	Varchar(100)	Dosen	Varchar(100)		
	Kelas	Varchar(10)	Tahun Akademik	Varchar(5)		
	Hari	Varchar(15)	Waktu	Date		
	Jenis Ujian	Varchar(5)	Jenis Soal	Varchar(5)		
	Soal Ujian	Hasil Ujian				
	<a href="#">Tambah Data</a>	<a href="#">Refresh</a>		<a href="#">Pencarian ...</a>	<a href="#">Reset</a>	
	No	NOBP	Nama	Kelas	Status	Nilai
	99	Varchar(15)	Varchar(100)	Varchar(5)	Varchar(15)	Int(3)
	99	Varchar(15)	Varchar(100)	Varchar(5)	Varchar(15)	Int(3)
		<a href="#">Sebelumnya</a>	99	<a href="#">Selanjutnya</a>	Banyak Data Perhalaman	99

**Gambar 3.3.40 Halaman Daftar Hasil Ujian**

## 13. Laporan Hasil Ujian

Laporan isi berisi daftar mahasiswa serta nilai ujian yang didapatkannya. Pada laporan terdapat ketentuan, jika mahasiswa belum melaksanakan ujian maka nilainya akan menjadi 0 dan statusnya menjadi “Belum Ujian”. Sedangkan jika mahasiswa sudah ujian dan jawaban mahasiswa sudah diperiksa (tipe soal essai) maka status akan berubah menjadi “Sudah Ujian” dan pada kolom nilai akan keluar nilai sesuai dengan hasil ujian mahasiswa tersebut, dengan bentuk rancangan seperti pada Gambar 3.41 :

<b>LAPORAN HASIL UJIAN</b>				
Mata Kuliah	Varchar(100)			
Dosen	Varchar(100)			
Kelas	Varchar(30)			
Tahun Akademik	Varchar(5)			
Hari	Date			
Jam	Time			
Jenis Ujian	Varchar(10)			
Jenis Soal	Varchar(10)			
No	NOBP	Nama	Kelas	Nilai
99	Varchar(15)	Varchar(100)	Varchar(10)	Int(3)
99	Varchar(15)	Varchar(100)	Varchar(10)	Int(3)

**Gambar 3.41 Laporan Hasil Ujian**

### 3.2.2.2 Desain *Input*

Dalam setiap sistem, pasti ada masukkan atau *input* yang nantinya akan diproses sistem. Dari proses *input* tadi, didapat hasil *output* seperti yang telah dilakukan sebelumnya. Berikut adalah desain *input* yang akan dikembangkan :

#### 1. Form Mahasiswa

Form ini digunakan untuk menambahkan mahasiswa baru kedalam sistem. Pengguna harus memasukkan NOBP dan nama kedalam form yang telah disediakan, dengan bentuk rancangan seperti pada Gambar 3.42 :

<b>Header</b>	
<b>Username</b>	<b>Tambah Data</b>
<b>Menu</b>	<b>NOBP</b> <input type="text"/> Varchar(15)
	<b>Nama</b> <input type="text"/> Varchar(100)
	<input type="button"/> Simpan <input type="button"/> Reset <input type="button"/> Batal

**Gambar 3.42 Form Mahasiswa**

## 2. Form Dosen

Form ini digunakan untuk menambahkan dosen baru kedalam sistem.

Pengguna harus memasukkan NIDN dan nama kedalam form yang telah disediakan, dengan bentuk rancangan seperti pada Gambar 3.43 :

<b>Header</b>	
<b>Username</b>	<b>Tambah Data</b>
<b>Menu</b>	<b>NIDN</b> <input type="text"/> Varchar(15)
	<b>Nama</b> <input type="text"/> Varchar(100)
	<input type="button"/> Simpan <input type="button"/> Reset <input type="button"/> Batal

**Gambar 3.43 Form Mahasiswa**

## 3. Form Matkul

Form ini digunakan untuk menambahkan mata kuliah baru kedalam sistem. Pengguna harus memasukkan kode mata kuliah, nama mata kuliah,

jumlah SKS dan semester kedalam form yang telah disediakan, dengan bentuk rancangan seperti pada Gambar 3.44 :

Header	
Username	<b>Tambah Data</b>
Menu	<p><b>Kode Matkul</b>  <input type="text" value="Varchar(15)"/></p> <p><b>Nama Matkul</b>  <input type="text" value="Varchar(100)"/></p> <p><b>SKS</b>  <input type="text" value="Int(3)"/></p> <p><b>Semester</b>  <input type="text" value="Int(3)"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/></p>

**Gambar 3.44 Form Matkul**

#### 4. Form Kelas

Form ini digunakan untuk menambahkan nama kelas baru, dengan bentuk rancangan seperti pada Gambar 3.45 :

Header	
Username	<b>Tambah Data</b>
Menu	<p><b>Nama Kelas</b>  <input type="text" value="Varchar(10)"/></p> <p><input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/></p>

**Gambar 3.45 Form Kelas**

#### 5. Form Tipe Soal

Form ini digunakan untuk menambahkan tipe soal baru jika diperlukan, dengan bentuk rancangan seperti pada Gambar 3.46 :

Header	
Username	<b>Tambah Data</b>
Menu	Nama Tipe Soal <input type="text" value="Varchar(20)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.46 Form Tipe Soal**

#### 6. Form Jenis Ujian

Form ini digunakan untuk menambahkan jenis ujian baru jika diperlukan, dengan bentuk rancangan seperti pada Gambar 3.47 :

Header	
Username	<b>Tambah Data</b>
Menu	Nama Jenis Ujian <input type="text" value="Varchar(10)"/> <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.47 Form Jenis Ujian**

#### 7. Form User

Form ini digunakan untuk menambahkan *user* baru. *User* baru dapat berupa admin baru, dosen baru atau mahasiswa baru, dengan bentuk rancangan seperti pada Gambar 3.48 :

Header	
Username	<b>Tambah Data</b>
Menu	Username <input type="text" value="Varchar(30)"/>
	Password <input type="text" value="Varchar(12)"/>
	Jenis User <input type="text" value="Varchar(10)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.48 Form User**

#### 8. Form Kuliah

Form ini digunakan oleh dosen untuk menambahkan kuliah yang diampu. Kuliah yang sudah ditambahkan nantinya akan muncul di form tambah ujian. Pada form ini, dosen diminta untuk memilih mata kuliah, memilih kelas-kelas yang diampu serta mengentri tahun akademik, dengan bentuk rancangan seperti pada Gambar 3.49 :

Header	
Username	<b>Tambah Data</b>
Menu	Pilih Mata Kuliah <input type="text" value="Varchar(100)"/>
	Tambahkan Kelas <input type="text" value="Varchar(12)"/>
	Tahun Akademik <input type="text" value="Varchar(5)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.49 Form Kuliah**

#### 9. Form Peserta Kuliah

Form ini digunakan oleh dosen untuk menambah peserta kuliah. Dosen diminta untuk memilih salah satu kelas, kemudian memilih mahasiswa

yang ingin ditambahkan, dengan bentuk rancangan seperti pada Gambar 3.50 :

Header	
Username	<b>Tambah Data</b>
Menu	<b>Pilih Kelas</b> <input type="text" value="Varchar(10)"/> <input type="button" value="▼"/>
	<b>Tambahkan Mahasiswa</b> <input type="text" value="Varchar(100)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.50 Form Peserta Kuliah**

#### 10. Form Ujian

Form ini digunakan oleh dosen untuk menambahkan ujian baru. Dosen diminta untuk memilih kuliah yang ingin diujikan, jenis soal ujian, jenis ujian, jadwal ujian serta deskripsi ujian jika perlu, dengan bentuk rancangan seperti pada Gambar 3.51 :

Header	
Username	<b>Tambah Data</b>
<b>Menu</b>	<b>Pilih Kuliah</b> <input type="text" value="Varchar(100)"/> 
	<b>Pilih Jenis Ujian</b> <input type="text" value="Varchar(10)"/> 
	<b>Pilih Jenis Soal</b> <input type="text" value="Varchar(15)"/> 
	<b>Hari Ujian</b> <input type="text" value="Date"/>
	<b>Jam Mulai</b> <input type="text" value="Time"/>
	<b>Jam Selesai</b> <input type="text" value="Time"/>
	<b>Deskripsi</b> <input type="text" value="Varchar(100)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.51 Form Ujian**

#### 11. Form Soal Ujian

Form ini digunakan oleh dosen untuk menambahkan soal ujian baru.

Form soal ujian memiliki 2 tampilan tergantung jenis soal ujian yang dipilih, dengan bentuk rancangan seperti pada Gambar 3.52 :

Header	
Username	<b>Tambah Data</b>
<b>Menu</b>	<b>Isi Soal</b> <input type="text" value="Text"/>
	<b>Bobot Nilai</b> <input type="text" value="Integer(3)"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>

**Gambar 3.52 Form Soal Ujian Essai**

Pada jenis soal ujian essai, dosen hanya diminta untuk memasukkan isi soal serta bobot soal. Namun pada soal ujian pilihan ganda, dosen diminta untuk memasukkan pilihan ganda yang disediakan dan bobot nilai untuk tipe soal pilihan ganda otomatis diatur menjadi 1 (satu), dengan bentuk rancangan seperti pada Gambar 3.53 :

Header							
Username	<b>Tambah Data</b> <b>Isi Soal</b> <input type="text" value="Text"/> <table border="1"> <tr> <td>A</td> <td>Varchar(200)</td> </tr> <tr> <td>B</td> <td>Varchar(200)</td> </tr> <tr> <td>C</td> <td>Varchar(200)</td> </tr> </table> <b>Jawaban</b> <input type="radio"/> A <input type="radio"/> B <input type="radio"/> C <input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>	A	Varchar(200)	B	Varchar(200)	C	Varchar(200)
A	Varchar(200)						
B	Varchar(200)						
C	Varchar(200)						
Menu							

**Gambar 3.53 Form Soal Ujian Pilihan Ganda**

#### 12. Form Halaman Ujian

Form ini digunakan oleh mahasiswa untuk menjawab soal ujian. Form halaman ujian memiliki 2 desain untuk tipe soal essai dan untuk tipe soal pilihan ganda, dengan bentuk rancangan seperti pada Gambar 3.54 :

Header	
<b>Pilih Soal</b> <input type="button" value="99"/> <input type="button" value="99"/> <input type="button" value="99"/>	Total Soal : 99      Sisa Waktu : Date <input type="button" value="Selesai : 99"/> <input type="button" value="Belum Selesai : 99"/> <i>Jawablah soal dibawah dengan benar!</i> <b>Isi Soal</b> Text <input type="radio"/> Text <input type="radio"/> Text <input type="radio"/> Text <div style="display: flex; justify-content: space-around;"> <input type="button" value="&lt;-"/> <input type="button" value="-&gt;"/> </div>
<b>Informasi Ujian</b> Varchar(100) Varchar(100) Date Time <input type="button" value="Kumpulkan Ujian"/>	

**Gambar 3.54 Form Halaman Ujian Pilihan Ganda**

Pada jenis soal ujian essai, mahasiswa diminta untuk mengetikkan jawabannya secara lengkap pada form yang telah disediakan. Dan jika sudah selesai, peserta wajib mengklik tombol “Simpan Jawaban” agar jawaban tersimpan. Pada form halaman ujian dengan tipe soal pilihan ganda, mahasiswa diminta untuk memilih jawaban yang dianggap benar. Dan jawaban peserta akan tersimpan otomatis, dengan bentuk rancangan seperti pada Gambar 3.55 :

Header	
<b>Pilih Soal</b> <input type="button" value="99"/> <input type="button" value="99"/> <input type="button" value="99"/>	<input type="button" value="Total Soal : 99"/> <input type="button" value="Sisa Waktu : Date"/> <input type="button" value="Selesai : 99"/> <input type="button" value="Belum Selesai : 99"/> <i>Jawablah soal dibawah dengan benar!</i>
<b>Informasi Ujian</b> Varchar(100) Varchar(100) Date Time <input type="button" value="Kumpulkan Ujian"/>	<b>Isi Soal</b> <b>Text</b> <b>Jawaban Anda</b> <input type="text" value="Text"/> <input type="button" value="Simpan Jawaban"/> <input type="button" value="&lt;-"/> <input type="button" value="-&gt;"/>

**Gambar 3.55 Form Halaman Ujian Essai**

### 3.2.2.3 Desain *File*

*File* merupakan tabel yang digunakan pada sebuah basis data. Semua data dan informasi yang terdapat pada aplikasi *computer assisted test* akan disimpan pada tabel-tabel tersebut. Agar lebih jelas mengenai *file-file* pada aplikasi *computer assisted test* dapat dilihat pada tabel-tabel dibawah ini :

#### 1. *File* Kelas Kuliah

*File* ini digunakan untuk menampung daftar kelas yang berada pada sebuah kuliah, dengan bentuk rancangan struktur seperti pada Tabel 3.23

**Tabel 3.23 Desain *File* Kelas Kuliah**

Nama *Database* : dbcat

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	60	Id kuliah
id_kelas	integer	11	Id kelas

## 2. *File* Kuliah

*File* ini digunakan untuk menampung semua daftar kuliah, dengan bentuk rancangan struktur seperti pada Tabel 3.24 :

**Tabel 3.24 Desain *File* Kuliah**

Nama *Database* : dbcat

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
status_kuliah	smallint	3	Status kuliah

### 3. *File* Peserta Kuliah

*File* ini digunakan untuk menampung daftar mahasiswa yang mengikuti sebuah kuliah, dengan bentuk rancangan struktur seperti pada Tabel 3.25

**Tabel 3.25 Desain *File* Peserta Kuliah**

Nama *Database* : dbcat

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

### 4. *File* Hasil Ujian

*File* ini digunakan untuk menampung daftar mahasiswa serta nilai ujian mereka, dengan bentuk rancangan struktur seperti pada Tabel 3.26 :

**Tabel 3.26 Desain *File* Hasil Ujian**

Nama *Database* : dbcat

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

##### 5. *File* Jawaban

*File* ini digunakan untuk menampung jawaban ujian peserta yang bertipe soal essai, dengan bentuk rancangan struktur seperti pada Tabel 3.27 :

**Tabel 3.27 Desain *File* Jawaban**

Nama *Database* : dbcat

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

## 6. *File* Soal

*File* ini digunakan untuk menampung soal ujian, dengan bentuk rancangan struktur seperti pada Tabel 3.28 :

**Tabel 3.28 Desain *File* Soal**

Nama *Database* : dbcat

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

## 7. *File* Kelas

*File* ini digunakan untuk menampung daftar nama kelas yang bisa digunakan pada kuliah, dengan bentuk rancangan struktur seperti pada Tabel 3.29 :

**Tabel 3.29 Desain *File* Kelas**

Nama *Database* : dbcat

Nama Tabel : tbkelas

*Primary Key* : id\_kelas

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Panjang data</b>	<b>Deskripsi</b>
id_kelas	integer	11	Id kelas
nm_kelas	varchar	10	Nama kelas

#### 8. *File* Jenis Soal

*File* ini digunakan untuk menampung nama jenis soal, dengan bentuk rancangan struktur seperti pada Tabel 3.30 :

**Tabel 3.30 Desain *File* Jenis Soal**

Nama *Database* : dbcat

Nama Tabel : tbjenis\_soal

*Primary Key* : id\_jsoal

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

#### 9. *File* Jenis Ujian

*File* ini digunakan untuk menampung nama jenis ujian, dengan bentuk rancangan struktur seperti pada Tabel 3.31 :

**Tabel 3.31 Desain *File* Jenis Ujian**

Nama *Database* : dbcat

Nama Tabel : tbjenis\_ujian

*Primary Key* : id\_jujian

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

#### 10. *File* Jenis User

*File* ini digunakan untuk menampung nama jenis *user*, dengan bentuk rancangan struktur seperti pada Tabel 3.32 :

**Tabel 3.32 Desain *File* Jenis User**

Nama *Database* : dbcat

Nama Tabel : tbjenis\_user

*Primary Key* : id\_juser

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

#### 11. *File* User

*File* ini digunakan untuk menampung daftar *user*, dengan bentuk rancangan struktur seperti pada Tabel 3.33 :

**Tabel 3.33 Desain *File User***

Nama *Database* : dbcat

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

## 12. *File Ujian*

*File* ini digunakan untuk menampung daftar ujian, dengan bentuk ran-cangan struktur seperti pada Tabel 3.34 :

**Tabel 3.34 Desain *File Ujian***

Nama *Database* : dbcat

Nama Tabel : tbujian

*Primary Key* : id\_ujian

Nama Kolom	Tipe Data	Panjang data	Deskripsi
1	2	3	4
id_ujian	varchar	40	Id ujian

1	2	3	4
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

### 13. *File* Mahasiswa

*File* ini digunakan untuk menampung seluruh data mahasiswa, dengan bentuk rancangan struktur seperti pada Tabel 3.35 :

**Tabel 3.35 Desain *File* Mahasiswa**

Nama *Database* : dbcat

Nama Tabel : tbmahasiswa

*Primary Key* : nobp

Nama Kolom	Tipe Data	Panjang data	Deskripsi
1	2	3	4
nobp	varchar	15	NOBP

1	2	3	4
nm_mahasiswa	varchar	100	Nama mahasiswa yang terdaftar
status_mahasiswa	smallint	3	Status mahasiswa

#### 14. *File* Dosen

*File* ini digunakan untuk menampung seluruh data dosen, dengan bentuk rancangan struktur seperti pada Tabel 3.36 :

**Tabel 3.36 Desain *File* Dosen**

Nama *Database* : dbcat

Nama Tabel : tb\_dosen

*Primary Key* : nidn

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Panjang data</b>	<b>Deskripsi</b>
nidn	varchar	15	NIDN
nm_dosen	varchar	100	Nama dosen
status_dosen	smallint	3	Status dosen

#### 15. *File* Matkul

*File* ini digunakan untuk menampung seluruh data matkul, dengan bentuk rancangan struktur seperti pada Tabel 3.37 :

**Tabel 3.37 Desain *File* Matkul**

Nama *Database* : dbcat

Nama Tabel : tb\_matkul

*Primary Key* : kd\_matkul

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Panjang data</b>	<b>Deskripsi</b>
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

#### 16. *File Soal Ujian*

*File* ini digunakan untuk menampung data soal dan ujian serta kaitannya, dengan bentuk rancangan struktur seperti pada Tabel 3.38 :

**Tabel 3.38 Desain *File Soal Ujian***

*Nama Database* : dbcat

*Nama Tabel* : tbsoal\_ujian

*Primary Key* : id\_sujian

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

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Implementasi Sistem**

Implementasi sistem merupakan kegiatan menerapkan sistem baru pada sistem lama yang sudah diteliti. Sebelum melakukan implementasi, perlu dilakukan pengkajian ulang terhadap sistem baru yang telah dibuat. Dan pengguna sistem perlu diberitahu tentang cara menggunakan sistem baru tersebut.

##### **4.1.1 Kebutuhan dan Instalasi Sistem**

Sebelum melakukan implementasi dan menjalankan aplikasi *computer assisted test* pada Universitas Putra Indonesia “YPTK” Padang, maka dibutuhkan spesifikasi perangkat keras (hardware) dan perangkat lunak (software), serta langkah-langkah yang harus dilakukan untuk dapat menjalankan aplikasi ini agar dapat berfungsi sebagaimana mestinya.

##### **4.1.2 Kebutuhan Perangkat Keras**

Untuk dapat menjalankan sistem yang dibuat ini diperlukan perangkat keras dan perangkat lunak dengan spesifikasi tertentu. Untuk menjalankan aplikasi ini sebagai *server* membutuhkan komputer dengan spesifikasi minimum sebagai berikut:

1. Prosessor dengan kecepatan minimal 0,6 Ghz.
2. *RAM* 512 MB.
3. *Harddisk* 10 Gb .

4. Monitor dengan resolusi minimal 1024 x 768.
5. VGA Card 8 MB, Mouse, dan keyboard.

#### **4.1.3 Kebutuhan Perangkat Lunak**

Perangkat lunak adalah komponen non fisik yang digunakan untuk membuat sistem komputer dapat berjalan dan melakukan tugasnya. Adapun perangkat lunak yang dibutuhkan dan telah diujicobakan pada komputer *server* yaitu :

1. Sistem Operasi *Ubuntu* 16.04 32bit.
2. *NodeJS* versi 6 keatas.
3. *Postgresql* versi 9.5

#### **4.1.4 Instalasi Program**

Untuk menjalankan aplikasi *computer assisted test* pada Universitas Putra Indonesia “YPTK” Padang, perlu dilakukan instalasi program serta melakukan konfigurasi pada program tersebut. Aplikasi *computer assisted test* bekerja pada sistem operasi *Ubuntu* 16.04.

##### **4.1.4.1 Instalasi *Postgresql***

*Postgresql* merupakan perangkat lunak yang bertugas mengatur basis data. Instalasi *postgresql* pada sistem operasi *ubuntu* dapat dilakukan dengan cara membuka terminal kemudian ketik “*sudo apt-get install postgresql*” dan tekan tombol *Enter*. Proses instalasi pun akan berlangsung, seperti pada Gambar 4.1 :

```
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 ... 222294 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 4.1 Proses Instalasi Postgresql**

#### 4.1.4.2 Impor Database ke Postgresql

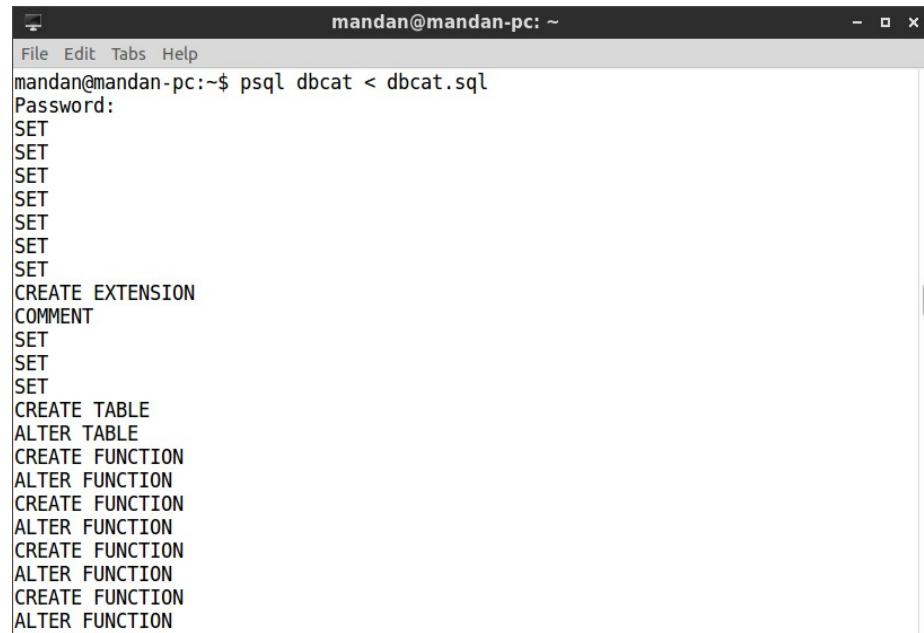
Proses impor *database* ke *postgresql* perlu dilakukan agar aplikasi *computer assissted test* dapat berjalan dengan benar. Adapun langkah-langkah impor *database* ke *postgresql* dapat dijelaskan sebagai berikut :

1. Buka terminal, kemudian ketik “`createdb dbcat`”. Perintah tersebut berfungsi untuk membuat *database* baru pada *postgresql* dengan nama “*dbcat*”. Setelah mengetik perintah tersebut, masukkan kata sandi dari *server*, seperti pada Gambar 4.2 :

```
mandan@mandan-pc:~$ createdb dbcat
Password:
mandan@mandan-pc:~$
```

**Gambar 4.2 Membuat database pada postgresql**

2. Setelah berhasil membuat *database*, ketik “`psql dbcat < dbcat.sql`” untuk melakukan impor *database*, seperti pada Gambar 4.3 :



```
mandan@mandan-pc: ~
File Edit Tabs Help
mandan@mandan-pc:~$ psql dbcat < dbcat.sql
Password:
SET
SET
SET
SET
SET
SET
SET
SET
CREATE EXTENSION
COMMENT
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE FUNCTION
ALTER FUNCTION
```

**Gambar 4.3 Import database ke postgresql**

3. Setelah berhasil melakukan impor, masuk ke *database* yang telah dibuat dengan mengetikkan perintah “`psql dbcat`” dan masukkan kata sandi *server*, seperti pada Gambar 4.4 :



```
mandan@mandan-pc: ~
File Edit Tabs Help
mandan@mandan-pc:~$ psql dbcat
Password:
psql (9.5.10)
Type "help" for help.

dbcat=#
```

**Gambar 4.4 Masuk ke database**

4. Tampilan ini menjelaskan struktur tabel `tbmahasiswa`. Untuk melihat struktur tabel `tbmahasiswa`, ketik “`\d tbmahasiswa`”, seperti pada Gambar 4.5 :

```
dbcat=# \d tbmahasiswa
          Table "public.tbmahasiswa"
  Column |          Type          |      Modifiers
-----+-----+-----+
 nm_mahasiswa | character varying(100) | not null
 nobp | character varying(15) | not null
 status_mahasiswa | smallint | not null default 1
Indexes:
 "tbmahasiswa_pkey" PRIMARY KEY, btree (nobp)
 "tbmahasiswa_nobp" UNIQUE CONSTRAINT, btree (nobp)
```

**Gambar 4.5 Struktur tabel tbmahasiswa**

5. Tampilan ini menjelaskan struktur tabel tbdosen. Untuk melihat struktur tabel tbdosen, ketik “\d tbdosen”, seperti pada Gambar 4.6 :

```
dbcat=# \d tbdosen;
          Table "public.tbdosen"
  Column |          Type          |      Modifiers
-----+-----+-----+
 nm_dosen | character varying(100) | not null
 nidn | character varying(15) | not null
 status_dosen | smallint | not null default 1
Indexes:
 "tbdosen_pkey" PRIMARY KEY, btree (nidn)
 "tbdosen_nidn" UNIQUE CONSTRAINT, btree (nidn)
```

**Gambar 4.6 Struktur tabel tbdosen**

6. Tampilan ini menjelaskan struktur tabel tbmatkul. Untuk melihat struktur tabel tbmatkul, ketik “\d tbmatkul”, seperti pada Gambar 4.7 :

```
dbcat=# \d tbmatkul;
          Table "public.tbmatkul"
  Column |          Type          |      Modifiers
-----+-----+-----+
 kd_matkul | character varying(15) | not null
 nm_matkul | character varying(100) | not null
 sks | smallint | not null
 smt | smallint | not null
 status_matkul | smallint | not null default 1
Indexes:
 "tbmatkul_pkey" PRIMARY KEY, btree (kd_matkul)
```

**Gambar 4.7 Struktur tabel tbmatkul**

7. Tampilan ini menjelaskan struktur tabel tbjenis\_soal. Untuk melihat struktur tabel tbjenis\_soal ketik “\d tbjenis\_soal”, seperti pada Gambar

4.8 :

```
dbcat=# \d tbjenis_soal;
          Table "public.tbjenis_soal"
  Column |      Type      |          Modifiers
-----+-----+-----+
 id_jsoal | integer | not null default nextval('tbjenis_soal_id_jsoal_seq'::regclass)
 nm_jsoal | character varying(20) | not null
Indexes:
 "tbjenis_soal_id_jsoal" PRIMARY KEY, btree (id_jsoal)
```

**Gambar 4.8 Struktur tabel tbjenis\_soal**

8. Tampilan ini menjelaskan struktur tabel tbjenis\_ujian. Untuk melihat struktur tabel tbjenis\_ujian, ketik “\d tbjenis\_ujian”, seperti pada Gambar 4.9 :

```
dbcat=# \d tbjenis_ujian;
          Table "public.tbjenis_ujian"
  Column |      Type      |          Modifiers
-----+-----+-----+
 id_ujian | integer | not null default nextval('tbjenis_ujian_id_ujian_seq'::regclass)
 nm_ujian | character varying(10) | not null
Indexes:
 "tbjenis_ujian_id_ujian" PRIMARY KEY, btree (id_ujian)
```

**Gambar 4.9 Struktur tabel tbjenis\_ujian**

9. Tampilan ini menjelaskan struktur tabel tbuser. Untuk melihat struktur tabel tbuser, ketik “\d tbuser”, seperti pada Gambar 4.10 :

```
dbcat=# \d tbuser;
          Table "public.tbuser"
  Column |      Type      |          Modifiers
-----+-----+-----+
 id_user | integer | not null default nextval('tbuser_id_user_seq'::regclass)
 username | character varying(30) | not null
 password | text | not null
 id_juser | smallint | not null
 status_user | smallint | not null default 1
Indexes:
 "tbuser_id_user" PRIMARY KEY, btree (id_user)
 "tbuser_username" UNIQUE CONSTRAINT, btree (username)
```

**Gambar 4.10 Struktur tabel tbuser**

10. Tampilan ini menjelaskan struktur tabel tbkelas. Untuk melihat struktur tabel tbkelas, ketik “\d tbkelas”, seperti pada Gambar 4.11 :

```
dbcat=# \d tbkelas;
          Table "public.tbkelas"
  Column |      Type      |      Modifiers
-----+-----+-----+
 id_kelas | integer | not null default nextval('tnama_kelas_id_kelas_seq'::regclass)
 nm_kelas | character varying(10) | not null
Indexes:
"tbkelas_id_kelas" PRIMARY KEY, btree (id_kelas)
```

**Gambar 4.11 Struktur tabel tbkelas**

11. Tampilan ini menjelaskan struktur tabel tbkuliah Untuk melihat struktur tabel tbkuliah, ketik “\d tbkuliah”, seperti pada Gambar 4.12 :

```
dbcat=# \d tbkuliah;
          Table "public.tbkuliah"
  Column |      Type      |      Modifiers
-----+-----+-----+
 id_kuliah | character varying(40) | not null
 nidn | character varying(11) | not null
 kd_matkul | character varying(15) | not null
 tahun_akademik | character varying(5) | not null
 status_kuliah | smallint | not null default '0'::smallint
Indexes:
"tbkuliah_id_kuliah" PRIMARY KEY, btree (id_kuliah)
"tbkuliah_kd_matkul" btree (kd_matkul)
"tbkuliah_nidn" btree (nidn)
```

**Gambar 4.12 Struktur tabel tbkuliah**

12. Tampilan ini menjelaskan struktur tabel tbpeserta\_kuliah. Untuk melihat struktur tabel tbpeserta\_kuliah, ketik “\d tbpeserta\_kuliah”, seperti pada Gambar 4.13 :

```
dbcat=# \d tbpeserta_kuliah;
          Table "public.tbpeserta_kuliah"
  Column |      Type      |      Modifiers
-----+-----+-----+
 id_peserta | character varying(60) | not null
 nobp | character varying(15) | not null
 id_kuliah | character varying(40) | not null
 status_peserta | smallint | not null default 1
 status_kelas_peserta | smallint | not null default '3'::smallint
 id_kelas | integer | not null default 0
Indexes:
"tbpeserta_kuliah_id_peserta" PRIMARY KEY, btree (id_peserta)
"tbpeserta_kuliah_id_kuliah" btree (id_kuliah)
"tbpeserta_kuliah_nobp" btree (nobp)
```

**Gambar 4.13 Struktur tabel tbpeserta\_kuliah**

13. Tampilan ini menjelaskan struktur tabel tbujian. Untuk melihat struktur tabel tbujian, ketik “\d tbujian”, seperti pada Gambar 4.14 :

```
dbcat=# \d tbujian;
          Table "public.tbujian"
  Column |      Type       |      Modifiers
-----+-----+-----+
  id_ujian | character varying(40) | not null
  hari | date | not null
  mulai | time without time zone | not null
  selesai | time without time zone | not null
  deskripsi | character varying(100) | not null
  status_ujian | smallint | not null default 0
  id_jujian | smallint | not null
  id_jsoal | smallint | not null
  id_kuliah | character varying(40) | not null
Indexes:
  "tbujian_id_ujian" PRIMARY KEY, btree (id_ujian)
  "tbujian_id_kuliah" btree (id_kuliah)
```

**Gambar 4.14 Struktur tabel tbujian**

14. Tampilan ini menjelaskan struktur tabel tbsoal. Untuk melihat struktur tabel tbsoal, ketik “\d tbsoal”, seperti pada Gambar 4.15 :

```
dbcat=# \d tbsoal;
          Table "public.tbsoal"
  Column |      Type       |      Modifiers
-----+-----+-----+
  id_soal | integer | not null default nextval('tbsoal_id_soal_seq'::regclass)
  isi_soal | text | not null
  pilihanGanda | json | 
  id_jsoal | smallint | not null
  bobot | smallint | not null default 1
  jawaban | character varying(5) | 
Indexes:
  "tbsoal_id_soal" PRIMARY KEY, btree (id_soal)
```

**Gambar 4.15 Struktur tabel tbsoal**

15. Tampilan ini menjelaskan struktur tabel tbsoal\_ujian. Untuk melihat struktur tabel tbsoal\_ujian, ketik “\d tbsoal\_ujian”, seperti pada Gambar 4.16 :

```
dbcat=# \d tbsoal_ujian
          Table "public.tbsoal_ujian"
  Column |      Type       |      Modifiers
-----+-----+-----+
  id_sujian | integer | not null default nextval('tbsoal_ujian_id_sujian_seq1'::regclass)
  id_ujian | character varying(40) | not null
  id_soal | integer | not null
Indexes:
  "tbsoal_ujian_id_sujian" PRIMARY KEY, btree (id_sujian)
  "tbsoal_ujian_id_soal" btree (id_soal)
  "tbsoal_ujian_id_ujian" btree (id_ujian)
```

**Gambar 4.16 Struktur tabel tbsoal\_ujian**

16. Tampilan ini menjelaskan struktur tabel tbhasil\_ujian. Untuk melihat struktur tabel tbhasil\_ujian, ketik “\d tbhasil\_ujian”, seperti pada Gambar 4.17 :

```
dbcat=# \d tbhasil_ujian;
          Table "public.tbhasil_ujian"
  Column |      Type      |                         Modifiers
-----+-----+-----+
  id_hasil | integer | not null default nextval('tbhasil_ujian_id_hasil_seq'::regclass)
  id_ujian | character varying(40) | not null
  nobp | character varying(15) | not null
  nilai | smallint | not null
Indexes:
  "tbhasil_ujian_id_hasil" PRIMARY KEY, btree (id_hasil)
  "tbhasil_ujian_id_ujian" btree (id_ujian)
  "tbhasil_ujian_nobp" btree (nobp)
```

**Gambar 4.17 Struktur tabel tbhasil\_ujian**

17. Tampilan ini menjelaskan struktur tabel tbkelas\_kuliah. Untuk melihat struktur tabel tbkelas\_kuliah, ketik “\d tbkelas\_kuliah”, seperti pada Gambar 4.18 :

```
dbcat=# \d tbkelas_kuliah;
          Table "public.tbkelas_kuliah"
  Column |      Type      |                         Modifiers
-----+-----+-----+
  id_kkuliah | integer | not null default nextval('tbkelas_ujian_id_kujian_seq'::regclass)
  id_kuliah | character varying(60) | not null
  id_kelas | integer | not null
Indexes:
  "tbkelas_kuliah_id_kkuliah" PRIMARY KEY, btree (id_kkuliah)
```

**Gambar 4.18 Struktur tabel kelas\_kuliah**

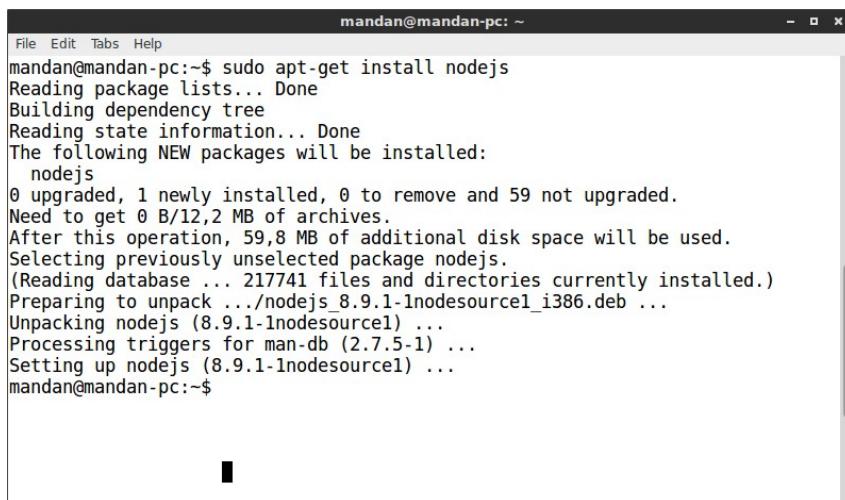
18. Tampilan ini menjelaskan struktur tabel tbjenis\_user. Untuk melihat struktur tabel tbjenis\_user, ketik “\d tbjenis\_user”, seperti pada Gambar 4.19 :

```
dbcat=# \d tbjenis_user;
          Table "public.tbjenis_user"
  Column |      Type      |                         Modifiers
-----+-----+-----+
  id_juser | integer | not null default nextval('tbjenis_user_id_juser_seq'::regclass)
  nm_juser | character varying(20) | not null
Indexes:
  "tbjenis_user_id_juser" PRIMARY KEY, btree (id_juser)
```

**Gambar 4.19 Struktur tabel tbjenis\_user**

#### 4.1.4.3 Instalasi NodeJS

*NodeJS* merupakan perangkat lunak yang menjadikan bahasa pemrograman *javascript* bisa berjalan pada sisi *server* layaknya *PHP*. Proses instalasi *nodejs* pada sistem operasi *ubuntu* tidaklah susah. Pada monitor, buka *terminal* dengan cara menekan tombol “*ctrl + alt + t*”. Jika *terminal* terbuka, ketik “*sudo apt-get install nodejs*” dan tekan tombol *Enter*. Dan proses instalasi pun akan berlangsung, seperti pada Gambar 4.20 :



```
mandan@mandan-pc:~$ sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 59 not upgraded.
Need to get 0 B/12,2 MB of archives.
After this operation, 59,8 MB of additional disk space will be used.
Selecting previously unselected package nodejs.
(Reading database ... 217741 files and directories currently installed.)
Preparing to unpack .../nodejs_8.9.1-1nodesource1_i386.deb ...
Unpacking nodejs (8.9.1-1nodesource1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up nodejs (8.9.1-1nodesource1) ...
mandan@mandan-pc:~$
```

**Gambar 4.20 Proses Instalasi NodeJS**

#### 4.1.4.4 Menjalankan NodeJS

Setelah berhasil melakukan instalasi *nodejs*, buka terminal dan masuk ke *folder* aplikasi *computer assisted test*. Kemudian ketik “*npm start*” dan *server nodejs* akan berjalan, seperti pada Gambar 4.21 :

```
mandan@mandan-pc:~/project/aplikasiujianberbasisweb$ npm start
> aplikasiujianberbasisweb@2.7.8 start /home/mandan/project/aplikasiujianberbasisweb
> node ./bin/www
```

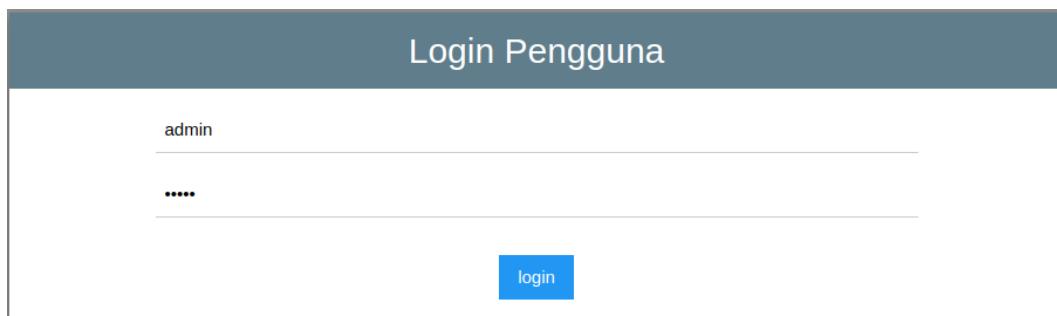
**Gambar 4.21 Menjalankan Nodejs**

## 4.2 Pengujian Sistem

Pada bagian pengujian program ini akan dijelaskan mengenai penggunaan dari aplikasi yang dibuat. Penjelasan aplikasi yang dibuat meliputi tampilan aplikasi, fungsi kontrol dalam aplikasi, serta cara penggunaanya. Pada sub bab akan dijelaskan tentang penggunaan aplikasi per sistem menu, mulai dari tampilan menu utama, fungsi dan cara penggunaannya sampai selesai.

### 4.2.1 Halaman Login

Halaman *login* digunakan oleh semua pemakai aplikasi dan setelah *login*, pemakai akan dibawa kehalaman yang sesuai dengan fungsi si pemakai. Pengguna diminta untuk mengentrikan *username* dan *password*, seperti pada Gambar 4.22 :

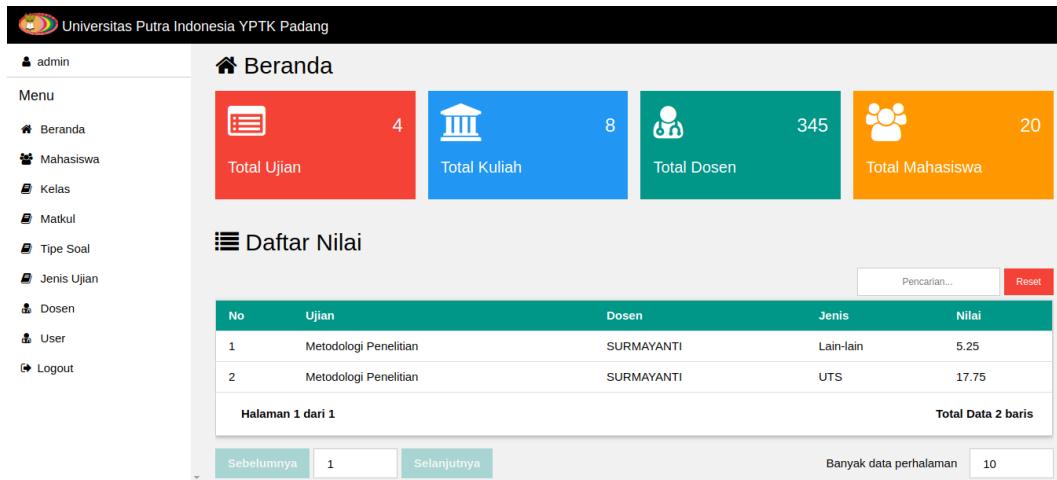


The image shows a screenshot of a web-based user login form. The title bar is dark blue with the text "Login Pengguna" in white. Below the title, there are two text input fields. The first field is labeled "username" and contains the text "admin". The second field is labeled "password" and contains the text ".....". At the bottom of the form is a blue rectangular button with the word "login" in white.

**Gambar 4.22 Tampilan Halaman Login**

### 4.2.2 Halaman Beranda

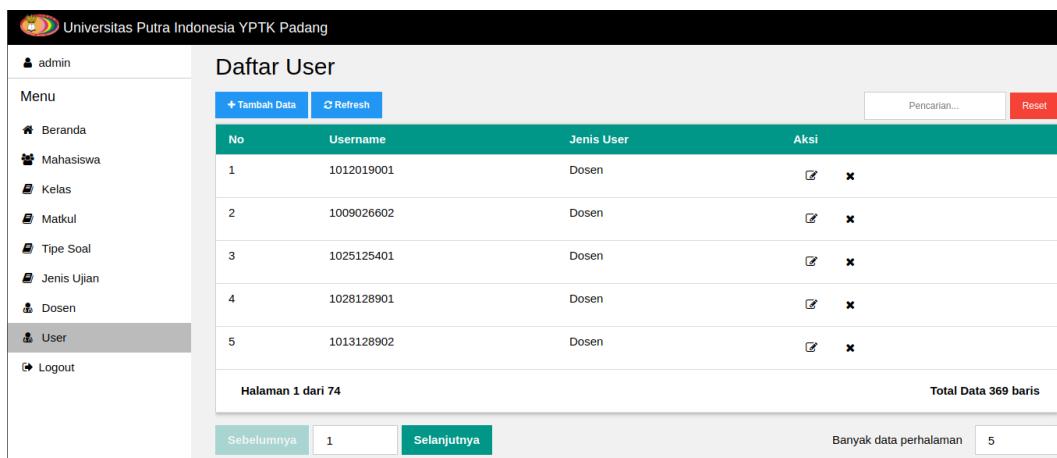
Halaman ini akan terlihat jika pengguna berhasil *login* ke sistem. Dihalaman ini ditampilkan daftar nilai rata-rata seluruh mata kuliah yang telah terdaftar pada aplikasi *CAT* ini, seperti pada Gambar 4.23 :



**Gambar 4.23 Tampilan Halaman Beranda**

#### 4.2.3 Halaman Kelola User

Halaman ini digunakan untuk mengelola *user* yang bisa masuk ke sistem. Admin dapat melihat, menambah, menghapus dan mengedit data *user*. Halaman ini dapat diakses dengan cara mengklik menu *user* yang terletak pada bagian sebelah kiri dan halaman kelola *user* akan muncul, seperti pada Gambar 4.24 :



**Gambar 4.24 Tampilan Halaman Kelola User**

Admin dapat menambah admin baru pada halaman ini. Sedangkan untuk *user* dengan jenis dosen dan mahasiswa, data *user* mereka dapat ditambahkan juga

dihalaman ini. Namun data *user* dari mahasiswa dan dosen sudah dibuat otomatis melalui halaman kelola mahasiswa dan dosen.

Admin dapat menambah *user* baru dengan mengklik tombol “Tambah Data” dan form tambah user akan muncul, seperti pada Gambar 4.25 :

Universitas Putra Indonesia YPTK Padang

Daftar User

Tambah Data

Username: admin2

Password: \*\*\*\*\*

Jenis User: Admin

Simpan Reset Batal

**Gambar 4.25 Tampilan Form Tambah User**

#### 4.2.4 Halaman Kelola Tipe Soal

Halaman ini digunakan untuk mengelola nama tipe soal. Halaman ini dapat diakses dengan mengklik menu tipe soal pada sebelah kiri dan halaman kelola tipe soal akan muncul, seperti pada Gambar 4.26 :

Universitas Putra Indonesia YPTK Padang

Daftar Tipe Soal

+ Tambah Data Refresh

No	Nama Tipe Soal	Aksi
1	Pilihan Ganda	<input checked="" type="checkbox"/> <input type="button" value="X"/>
2	Essai	<input checked="" type="checkbox"/> <input type="button" value="X"/>

Halaman 1 dari 1 Total Data 2 baris

Sebelumnya 1 Selanjutnya

Banyak data per halaman 5

**Gambar 4.26 Tampilan Halaman Kelola Tipe Soal**

Admin dapat menambahkan tipe soal baru jika diperlukan. Admin dapat mengklik tombol “Tambah Data” dan form tambah tipe soal akan muncul, seperti pada Gambar 4.27 :

**Gambar 4.27 Tampilan Form Tambah Tipe Soal**

#### 4.2.5 Halaman Kelola Jenis Ujian

Halaman ini digunakan untuk mengelola nama jenis ujian. Halaman ini dapat diakses dengan mengklik menu jenis ujian pada sebelah kiri dan halaman kelola jenis ujian akan muncul, seperti pada Gambar 4.28 :

No	Nama Jenis Ujian	Aksi
1	UTS	<input checked="" type="checkbox"/>
2	UAS	<input checked="" type="checkbox"/>
3	Lain-lain	<input checked="" type="checkbox"/>

**Gambar 4.28 Tampilan Halaman Kelola Jenis Ujian**

Admin dapat menambahkan jenis ujian baru jika diperlukan. Admin dapat mengklik tombol “Tambah Data” dan form tambah jenis ujian akan muncul, seperti pada Gambar 4.29 :

**Gambar 4.29 Tampilan Form Tambah Jenis Ujian**

#### 4.2.6 Halaman Kelola Mahasiswa

Halaman ini digunakan untuk mengelola data mahasiswa. Halaman ini hanya dapat diakses oleh admin. Admin dapat melihat, mengedit, menambah dan menghapus mahasiswa. Halaman ini dapat diakses dengan mengklik menu mahasiswa pada sebelah kiri dan halaman kelola mahasiswa akan muncul, seperti pada Gambar 4.30 :

No	Nama	NOBP	Aksi
1	Ego Dafma Dasa	14101152610565	<input checked="" type="checkbox"/> <input type="button" value="x"/>
2	Aidil	14101152610544	<input checked="" type="checkbox"/> <input type="button" value="x"/>
3	Irvan Rultio	14101152610550	<input checked="" type="checkbox"/> <input type="button" value="x"/>
4	Davidsyah Permana Putra	14101152610545	<input checked="" type="checkbox"/> <input type="button" value="x"/>
5	Yogi Anggara	14101152610590	<input checked="" type="checkbox"/> <input type="button" value="x"/>

**Gambar 4.30 Tampilan Halaman Kelola Mahasiswa**

Semua mahasiswa yang telah ditambahkan, otomatis ia juga akan ditambahkan ke tabel user agar mahasiswa dapat melakukan login ke sistem. Info user untuk mahasiswa akan terbentuk otomatis, dimana nobp akan menjadi username dan password user merupakan kombinasi dari kata “upi” + nobp + “yptk”.

Untuk menambahkan mahasiswa baru, admin dapat mengklik tombol “Tambah Data” dan form tambah mahasiswa pun akan muncul, seperti pada Gambar 4.31 :

Universitas Putra Indonesia YPTK Padang

Daftar Mahasiswa

Tambah Data

Nama Mahasiswa  
Ego Dafma Dasa

NOBP  
14101152610565

Simpan Reset Batal

**Gambar 4.31 Tampilan Form Tambah Mahasiswa**

#### 4.2.7 Halaman Kelola Dosen

Halaman ini digunakan untuk mengelola data dosen. Admin dapat melihat, menambah, menghapus dan mengedit data dosen, seperti pada Gambar 4.32 :

Universitas Putra Indonesia YPTK Padang

Daftar Dosen

+ Tambah Data Refresh

No	NIDN	Nama Dosen	Aksi
1	1012019001	ANITA ADE RAHMA	<input checked="" type="checkbox"/>
2	1009026602	ARFIS	<input checked="" type="checkbox"/>
3	1023116001	AZIZUL HAKIM	<input checked="" type="checkbox"/>
4	1025125401	BUCHYARDI GOND	<input checked="" type="checkbox"/>
5	1028128901	DESFRIANA SARI	<input checked="" type="checkbox"/>

**Gambar 4.32 Tampilan Halaman Kelola Dosen**

Sama seperti dengan data mahasiswa, data dosen juga akan otomatis ditambahkan kedalam tabel user. NIDN akan menjadi username dari dosen dan kombinasi kata “upi” + nidn + “yptk” menjadi passwordnya.

Untuk menambah dosen baru, Admin dapat mengklik tombol “Tambah Data” dan form tambah dosen akan muncul, seperti pada Gambar 4.33 :

Tambah Data

NIDN  
1029108702

Nama Dosen  
Ashabul Fadhl

Simpan Reset Batal

**Gambar 4.33 Tampilan Form Tambah Dosen**

#### 4.2.8 Halaman Kelola Matkul

Halaman ini digunakan untuk mengelola data mata kuliah. Admin dapat melihat, menambah, menghapus dan mengedit mata kuliah, seperti pada Gambar 4.34 :

Universitas Putra Indonesia YPTK Padang

Daftar Mata Kuliah

+ Tambah Data Refresh

Pencarian... Reset

No Kode Matkul Nama Matkul Semester SKS Aksi

1 KKKI12106 Pengantar Teknologi Info. 1 2   x

2 KKKI12116 Paket Pemrograman I () 1 2   x

3 KKKI13105 Pengantar Manajemen 1 3   x

4 KKKI13111 Kalkulus 1 3   x

5 KPKI12101 Pendidikan Pancasila 1 2   x

Halaman 1 dari 11 Total Data 55 baris

Sebelumnya 1 Selanjutnya Banyak data per halaman 5

**Gambar 4.34 Tampilan Halaman Kelola Matkul**

Admin dapat mengklik tombol “Tambah Data” untuk menampilkan form tambah matkul, seperti pada Gambar 4.35 :

Universitas Putra Indonesia YPTK Padang

**Tambah Data**

Kode Matkul  
KPKI12102

Nama Matkul  
Pendidikan Agama

SKS  
2

Semester  
1

Pencarian...

Aksi

<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>
<input checked="" type="checkbox"/>	<input type="button" value="x"/>

**Gambar 4.35 Tampilan Form Tambah Matkul**

#### 4.2.9 Halaman Kelola Kelas

Halaman ini digunakan untuk mengelola daftar nama kelas. Jika terdapat nama kelas baru, Admin dapat menambahkannya dihalaman ini, seperti pada Gambar 4.36 :

Universitas Putra Indonesia YPTK Padang

**Daftar Nama Kelas**

Pencarian...

No	Nama Kelas	Aksi
1	SI-1	<input checked="" type="checkbox"/> <input type="button" value="x"/>
2	SI-2	<input checked="" type="checkbox"/> <input type="button" value="x"/>
3	SI-3	<input checked="" type="checkbox"/> <input type="button" value="x"/>
4	SI-4	<input checked="" type="checkbox"/> <input type="button" value="x"/>
5	SI-5	<input checked="" type="checkbox"/> <input type="button" value="x"/>
Halaman 1 dari 3		
Total Data 11 baris		
<input type="button" value="Sebelumnya"/>	1	<input type="button" value="Selanjutnya"/>
Banyak data per halaman		5

**Gambar 4.36 Tampilan Form Tambah Matkul**

Admin dapat menambahkan nama kelas baru dengan mengklik tombol “Tambah Data” dan form tambah kelas akan muncul, seperti pada Gambar 4.37 :

**Gambar 4.37 Tampilan Form Tambah Kelas**

#### 4.2.10 Halaman Kelola Kuliah

Halaman ini berisi daftar kuliah yang telah ditambahkan dosen sebelumnya. Dosen dapat melihat, menambah dan menghapus kuliah serta akses menu ke halaman kelola peserta kuliah, seperti pada Gambar 4.38 :

**Gambar 4.38 Tampilan Halaman Kelola Kuliah**

Seperi dapat dilihat pada Gambar 4.38, status kuliah tidak akan aktif jika belum ada mahasiswa yang ditambahkan pada kuliah tersebut. Hal tersebut bertujuan agar kuliah tersebut tidak bisa diikutsertakan pada ujian nantinya. Sehingga kasus ujian tanpa mahasiswa tidak muncul.

Dosen dapat menambahkan kuliah baru dengan mengklik tombol “Tambah Data” dan form tambah kuliah akan muncul. Pada saat form tambah kuliah

muncul, dosen diminta untuk memilih mata kuliah, memilih kelas yang diampu serta tahun akademik, seperti pada Gambar 4.39 :

**Gambar 4.39 Form Tambah Kuliah**

#### 4.2.11 Halaman Kelola Peserta Kuliah

Setelah dosen berhasil menambahkan kuliah, selanjutnya dosen harus menambahkan peserta kuliah. Hal tersebut dapat dilakukan pada tombol “Peserta Kuliah” yang terletak didekat tombol hapus kuliah. Menu tersebut akan membawa dosen ke halaman kelola peserta kuliah, seperti pada Gambar 4.40 :

**Gambar 4.40 Tampilan Halaman Kelola Peserta Kuliah**

Pada halaman ini, dosen dapat menambah dan menghapus peserta kuliah. Dosen dapat menambah peserta kuliah dengan mengklik tombol “Tambah Data” dan form tambah peserta kuliah akan muncul seperti pada Gambar 4.41 :

**Gambar 4.41 Form Tambah Peserta Kuliah**

Pada form ini, dosen diminta untuk memilih kelas dan memilih mahasiswa yang ingin ditambahkan ke kelas tersebut. Semua peserta kuliah yang telah ditambahkan otomatis akan bisa mengikuti ujian sesuai dengan kuliah yang diujiankan.

#### 4.2.12 Halaman Kelola Ujian

Halaman ini berisi daftar ujian yang telah ditambahkan oleh dosen. Dosen dapat melihat, menambah, menghapus dan mengedit data ujian, seperti pada Gambar 4.42 :

**Gambar 4.42 Tampilan Halaman Kelola Ujian**

Seperti dapat dilihat pada Gambar 4.42, status ujian akan berubah menjadi aktif jika ujian tersebut sudah terdapat 1 (satu) soal. Hal tersebut bertujuan agar tidak munculnya ujian tanpa soal.

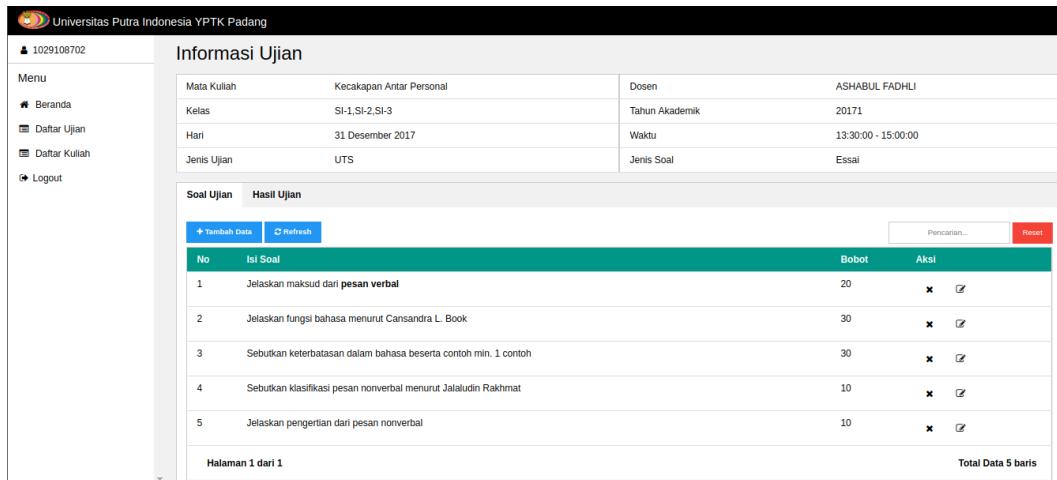
Dosen dapat menambahkan ujian baru dengan mengklik tombol “Tambah Data” dan form tambah ujian akan muncul, seperti pada Gambar 4.43 :

**Gambar 4.43 Form Tambah Ujian**

Pada form tambah ujian, dosen diminta untuk memilih kuliah yang ingin diujangkan. Jika kuliah telah dipilih, maka akan muncul pilihan jenis ujian yang tersedia. Jenis ujian yang muncul pada form ini adalah jenis ujian yang sebelumnya belum pernah dipilih sehingga dosen tidak akan salah dalam memilih jenis ujian. Kemudian dosen diminta untuk mengisi waktu kapan ujian akan diadakan serta deskripsi ujian.

#### 4.2.13 Halaman Kelola Soal Ujian

Halaman ini digunakan dosen untuk mengelola soal ujian. Dosen dapat melihat, menambah, menghapus dan mengubah soal ujian, seperti pada Gambar 4.44 :

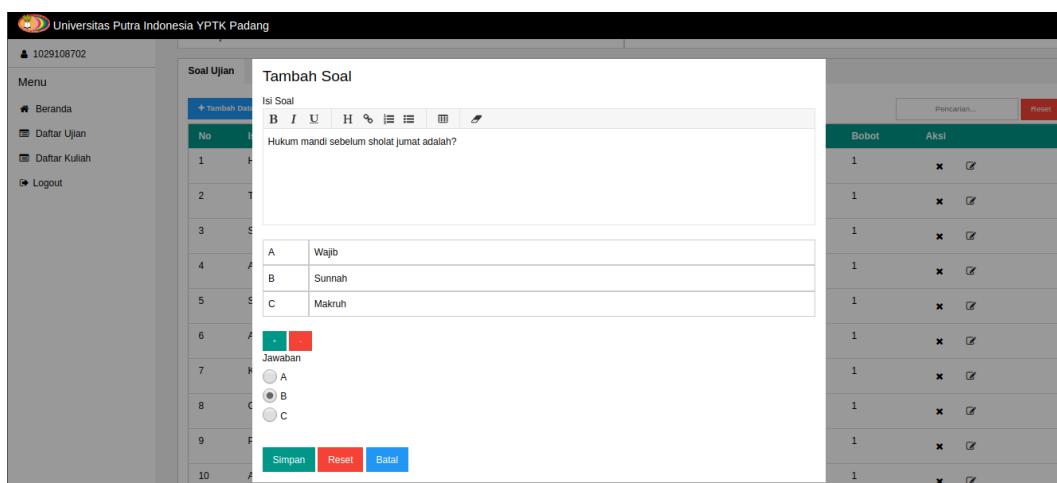


No	Isi Soal	Bobot	Aksi
1	Jelaskan maksud dari <b>pesan verbal</b>	20	<input checked="" type="checkbox"/> <input type="checkbox"/>
2	Jelaskan fungsi bahasa menurut Cansandra L. Book	30	<input checked="" type="checkbox"/> <input type="checkbox"/>
3	Sebutkan keterbatasan dalam bahasa beserta contoh min. 1 contoh	30	<input checked="" type="checkbox"/> <input type="checkbox"/>
4	Sebutkan klasifikasi pesan nonverbal menurut Jalaludin Rakmat	10	<input checked="" type="checkbox"/> <input type="checkbox"/>
5	Jelaskan pengertian dari pesan nonverbal	10	<input checked="" type="checkbox"/> <input type="checkbox"/>

Halaman 1 dari 1 Total Data 5 baris

**Gambar 4.44 Tampilan Halaman Ujian Tipe Soal Essai**

Untuk menambah soal ujian baru, dosen dapat mengklik tombol “Tambah Data” yang terdapat pada bagian atas tabel. Jika di klik maka akan muncul form untuk menambahkan soal ujian baru. Jenis soal ujian sendiri terbagi atas 2 macam yaitu pilihan ganda dan essai. Form tambah soal pilihan ganda dapat dilihat seperti pada Gambar 4.45 :



No	Isi Soal	Bobot	Aksi								
1	Hukum mandi sebelum sholat jumat adalah?	1	<input checked="" type="checkbox"/> <input type="checkbox"/>								
2	Tujuan sholat jumat adalah?	1	<input checked="" type="checkbox"/> <input type="checkbox"/>								
3	Sebutkan 3 hukum mandi?	1	<input checked="" type="checkbox"/> <input type="checkbox"/>								
4	<table border="1"> <tr> <td>A</td> <td>Wajib</td> </tr> <tr> <td>B</td> <td>Sunnah</td> </tr> <tr> <td>C</td> <td>Makruh</td> </tr> <tr> <td>D</td> <td></td> </tr> </table>	A	Wajib	B	Sunnah	C	Makruh	D		1	<input checked="" type="checkbox"/> <input type="checkbox"/>
A	Wajib										
B	Sunnah										
C	Makruh										
D											
5	<table border="1"> <tr> <td>A</td> <td>Wajib</td> </tr> <tr> <td>B</td> <td>Sunnah</td> </tr> <tr> <td>C</td> <td>Makruh</td> </tr> <tr> <td>D</td> <td></td> </tr> </table>	A	Wajib	B	Sunnah	C	Makruh	D		1	<input checked="" type="checkbox"/> <input type="checkbox"/>
A	Wajib										
B	Sunnah										
C	Makruh										
D											
6	<table border="1"> <tr> <td>A</td> <td>B</td> <td>C</td> <td>D</td> </tr> </table>	A	B	C	D	1	<input checked="" type="checkbox"/> <input type="checkbox"/>				
A	B	C	D								
7	<table border="1"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> </table>	A	B	C	1	<input checked="" type="checkbox"/> <input type="checkbox"/>					
A	B	C									
8	<table border="1"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> </table>	A	B	C	1	<input checked="" type="checkbox"/> <input type="checkbox"/>					
A	B	C									
9	<table border="1"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> </table>	A	B	C	1	<input checked="" type="checkbox"/> <input type="checkbox"/>					
A	B	C									
10	<table border="1"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> </table>	A	B	C	1	<input checked="" type="checkbox"/> <input type="checkbox"/>					
A	B	C									

Jawaban

A B C D

Simpan Reset Batal

**Gambar 4.45 Form Tambah Soal Pilihan Ganda**

Seperti dapat dilihat pada Gambar 4.45, dosen diminta untuk memasukkan isi soal beserta pilihan ganda dan jawaban benar. Pada tipe soal pilihan ganda,

bobot otomatis bernilai satu di setiap soal. Dan untuk mendapatkan hasil akhir dari ujian dengan tipe soal pilihan ganda, digunakan rumus “jawaban benar dikali dengan hasil pembagian 100 dengan total soal” sehingga nilai maksimal masih tetap 100. Form tambah soal dengan tipe soal essai dapat dilihat pada Gambar 4.46 :

**Gambar 4.46 Form Tambah Soal Essai**

Sama seperti pilihan ganda, dosen diminta untuk memasukkan isi soal. Namun pada tipe soal essai, dosen diminta untuk memasukkan bobot nilai soal secara manual sehingga nilai maksimal yang didapat pada ujian tersebut adalah total dari nilai bobot secara keseluruhan di setiap soal.

#### 4.2.14 Halaman Pilih Ujian

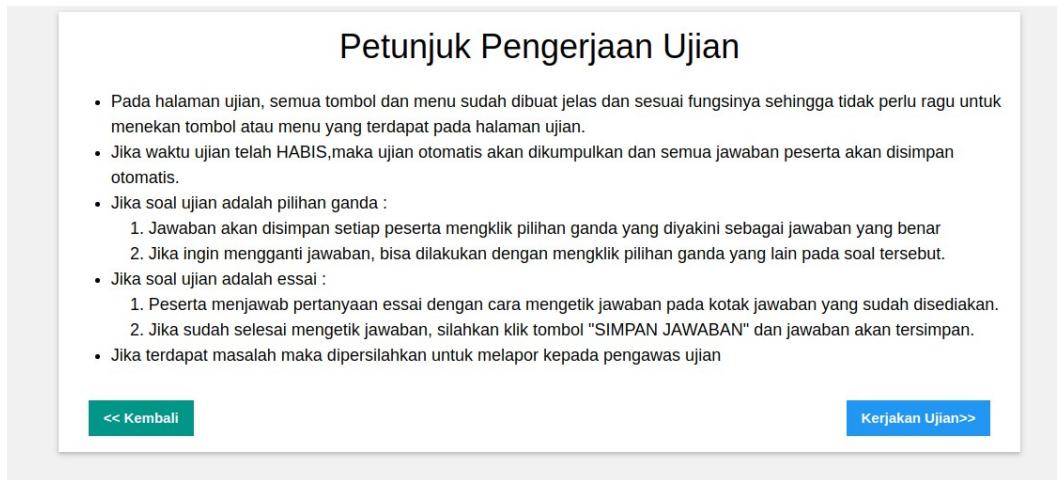
Halaman ini diakses oleh mahasiswa untuk memilih ujian yang akan diikuti. Mahasiswa harus *login* terlebih dahulu sebelum bisa masuk kehalaman ini. Pada halaman ini akan muncul daftar ujian yang bisa dilaksanakan saat itu juga, seperti pada Gambar 4.47 :



**Gambar 4.47 Tampilan Halaman Pilih Ujian**

#### 4.2.15 Halaman Petunjuk Ujian

Halaman ini akan muncul setelah mahasiswa memilih ujian. Halaman ini berisi penjelasan tentang cara penggerjaan ujian, seperti pada Gambar 4.48 :



**Gambar 4.48 Tampilan Halaman Pilih Ujian**

#### 4.2.16 Halaman Ujian

Halaman ini digunakan oleh mahasiswa untuk mengerjakan ujian. Pada halaman ujian, mahasiswa bisa mengerjakan ujian sesuai dengan tipe soal ujian. Jika tipe soal ujian adalah pilihan ganda, maka mahasiswa tinggal mengklik pilihan ganda yang dianggap benar sebagai jawaban, seperti pada Gambar 4.49:

Universitas Putra Indonesia YPTK Padang

Pilih Soal

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Total Soal : 16

Sisa Waktu : 1 Jam 19 Menit 5 Detik

Selesai : 5

Belum Selesai : 11

**Jawablah soal dibawah ini dengan benar!**

12. As-sidiq berarti

Membenarkan

Menguatkan

Memperbaiki

◀ ▶

Khalilq Alifikrizal  
Pendidikan Agama  
31 Desember 2017  
0 1:50:55

Kumpulkan Ujian

#### **Gambar 4.49 Tampilan Halaman Ujian Tipe Soal Pilihan Ganda**

Jika tipe soal adalah essai, maka akan ditampilkan kotak isian untuk mengetik jawaban, seperti pada Gambar 4.50 :

 Universitas Putra Indonesia YPTK Padang						
<b>Pilih Soal</b>						
<b>1</b>	<b>2</b>	<b>3</b>				
<b>4</b>						
<b>5</b>						
<b>Informasi Ujian</b>						
 Ego Dafma Dasa						
 Kecakapan Antar Personal						
31 Desember 2017						
0 1:40:40						
<b>Kumpulkan Ujian</b>						
<table border="1"><tr><td>Total Soal : 5</td><td>Sisa Waktu : 1 Jam 19 Menit 20 Detik</td></tr><tr><td>Selesai : 1</td><td>Belum Selesai : 4</td></tr></table>			Total Soal : 5	Sisa Waktu : 1 Jam 19 Menit 20 Detik	Selesai : 1	Belum Selesai : 4
Total Soal : 5	Sisa Waktu : 1 Jam 19 Menit 20 Detik					
Selesai : 1	Belum Selesai : 4					
<b>Jawablah soal dibawah ini dengan benar!</b>						
1. Jelaskan maksud dari <b>pesan verbal</b>						
Jawaban Anda						
<p>B I U S ॥ ॥/ H<sub>1</sub> H<sub>2</sub> ॥ ॥ x<sub>1</sub> x<sup>2</sup> ॥ ॥ * ¶ Normal ⇡ Normal ⇡ A ¶ Sans Serif ⇡ ॥ ॥ ॥ ॥</p>						
<p> Pesan verbal adalah semua jenis simbol yang menggunakan satu kata atau lebih. Pesan verbal terdiri atas kata-kata terucap atau tertulis.</p>						
<p>Pesan verbal terbagi atas 2 jenis yaitu :</p>						
<p>1. Fungsional yaitu alat yang dimiliki bersama untuk mengungkap gagasan</p>						
<p>2. Formal yaitu rangkaian kata-kata yang disusun sesuai dengan tata bahasa.</p>						
<p><b>Simpan Jawaban</b></p>						
<p>← →</p>						

**Gambar 4.50 Tampilan Halaman Ujian Tipe Soal Essai**

Disebelah kiri halaman, terdapat menu “Pilih Soal” dimana mahasiswa dapat melihat jumlah soal yang ada pada ujian tersebut. Mahasiswa juga dapat menampilkan soal dengan nomor soal tertentu dengan cara mengklik kotak-kotak berisi angka pada “Pilih Soal” tersebut. Warna merah pada kotak menunjukkan bahwa soal belum dikerjakan dan warna hijau menunjukkan soal sudah dikerjakan.

Jika mahasiswa sudah selesai mengerjakan semua soal, mahasiswa dapat mengklik tombol “Kumpulkan Ujian” dan mahasiswa akan keluar dari halaman ujian.

#### 4.2.17 Halaman Periksa Jawaban

Halaman periksa jawaban merupakan halaman yang digunakan oleh dosen untuk melakukan pemeriksaan jawaban pada jawaban ujian mahasiswa. Dosen akan melakukan pemeriksaan jawaban pada ujian yang bertipe soal essai. Dosen akan memberikan nilai disetiap soal hingga semua soal sudah memiliki nilai. Dan total nilai pada setiap soal akan ditotalkan dan dimasukkan ke tabel nilai, seperti pada Gambar 4.51 :

Universitas Putra Indonesia YPTK Padang

Pemeriksaan Jawaban

1029100702

Menu

- Beranda
- Dafat Ujian
- Dafat Kuliah
- Logout

Pilih Soal

1 2 3 4 5

Soal

Jelaskan maksud dari pesan verbal

Jawaban

Pesan verbal adalah semua jenis simbol yang menggunakan satu kata atau lebih. Pesan verbal terdiri atas kata-kata terucap atau tertulis.

Pesan verbal bagi atas 2 jenis yaitu :

1. Fungsional yaitu alat yang dimiliki bersama untuk mengungkap gagasan
2. Formal yaitu rangkulan kata-kata yang disusun sesuai dengan tata bahasa.

Nilai yang diberikan

Maks. Nilai Soal ini : 20

17

Simpan

Nilai Akhir

Total Nilai Peserta : 17

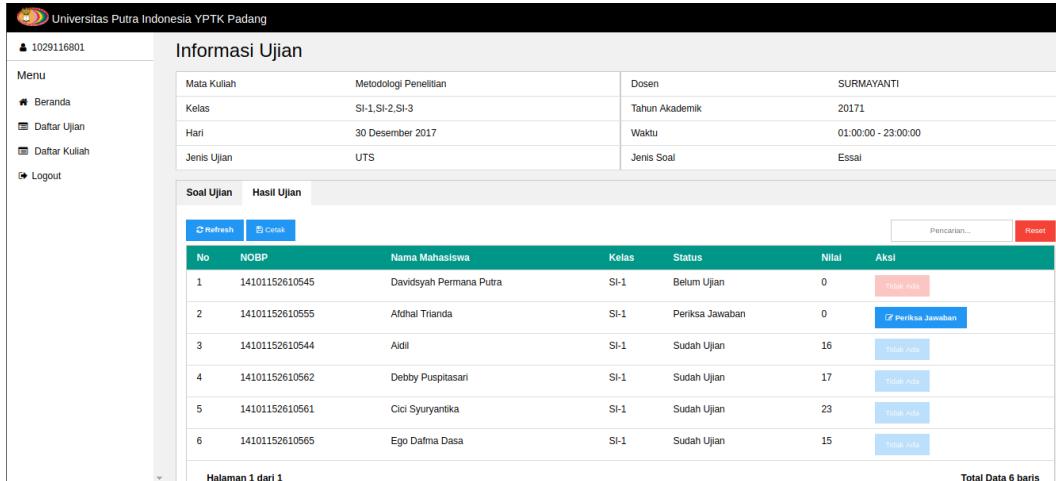
Simpan Hasil Pemeriksaan

**Gambar 4.51 Tampilan Halaman Periksa Jawaban**

#### 4.2.18 Halaman Hasil Ujian

Halaman ini digunakan oleh dosen untuk melihat hasil ujian dari mahasiswa. Dosen dapat melihat status mahasiswa dan aksi yang dapat dilakukan terhadap mahasiswa tersebut. Jika soal ujian bertipe essai, maka dosen dapat meng-

lik tombol “Periksa Jawaban” pada kolom aksi. Adapun desain dari halaman hasil ujian dapat dilihat pada Gambar 4.52 :



Informasi Ujian						
Mata Kuliah		Metodologi Penelitian		Dosen		SURMAYANTI
Kelas		SI-1,SI-2,SI-3		Tahun Akademik		20171
Hari		30 Desember 2017		Waktu		01:00:00 - 23:00:00
Jenis Ujian		UTS		Jenis Soal		Essai
Soal Ujian		Hasil Ujian				
<input type="button" value="REFRESH"/> <input type="button" value="Cetak"/>				<input type="text" value="Pencarian..."/> <input type="button" value="Reset"/>		
No	NOBP	Nama Mahasiswa	Kelas	Status	Nilai	Aksi
1	14101152610545	Davidsyah Permana Putra	SI-1	Belum Ujian	0	<input type="button" value="Periksa Jawaban"/>
2	14101152610555	Afdhal Trianda	SI-1	Periksa Jawaban	0	<input type="button" value="Periksa Jawaban"/>
3	14101152610544	Aidil	SI-1	Sudah Ujian	16	<input type="button" value="Total Akhir"/>
4	14101152610562	Debby Puspitasari	SI-1	Sudah Ujian	17	<input type="button" value="Total Akhir"/>
5	14101152610561	Cici Syuryantika	SI-1	Sudah Ujian	23	<input type="button" value="Total Akhir"/>
6	14101152610565	Ego Dafma Dasa	SI-1	Sudah Ujian	15	<input type="button" value="Total Akhir"/>

**Gambar 4.52 Tampilan Halaman Hasil Ujian**

Dosen juga dapat mencetak hasil ujian kedalam bentuk *file* seperti *file excel* dan *csv* dengan cara mengklik tombol “Cetak” dan kemudian akan muncul 2 (dua) pilihan format *file* yang bisa digunakan. Jika sudah dipilih, maka proses cetak akan dimulai dan jika sudah selesai, *browser* secara otomatis akan mengunduh *file* hasil cetak tersebut, seperti pada Gambar 4.53 :

	A	B	C	D	E
1	LAPORAN HASIL UJIAN				
2	Mata Kuliah	Kecakapan Antar Personal			
3	Dosen	ASHABUL FADHLI			
4	Kelas	SI-1,SI-2,SI-3			
5	Tahun Akademik	20171			
6	Hari	31 Desember 2017			
7	Jam	13:30:00 - 15:00:00			
8	Jenis Ujian	UTS			
9	Jenis Soal	Essai			
10					
11	No	NOBP	Nama	Kelas	Nilai
12	1	13101152610342	Gilang Kovalen	SI-1	50
13	2	14101152610544	Aidil	SI-1	80
14	3	14101152610545	Davidsyah Permana Putra	SI-1	90
15	4	14101152610555	Afdhal Trianda	SI-1	80
16	5	14101152610561	Cici Syuryantika	SI-1	100
17	6	14101152610562	Debby Puspitasari	SI-1	95
18	7	14101152610565	Ego Dafma Dasa	SI-1	60

**Gambar 4.53 Laporan Hasil Ujian Format File CSV**

Salah satu kelebihan dari *file* dengan format *csv* adalah bisa dibuka dengan berbagai macam aplikasi sehingga *file* dengan format *csv* cocok jika dosen tidak memiliki aplikasi seperti *microsoft office*. *File* laporan hasil ujian dengan format *excel* dapat dilihat pada Gambar 4.54 :

	A	B	C	D	E
1	<b>LAPORAN HASIL UJIAN</b>				
2	Mata Kuliah	Kecakapan Antar Personal			
3	Dosen	ASHABUL FADHLI			
4	Kelas	SI-1,SI-2,SI-3			
5	Tahun Akademik	20171			
6	Hari	31 Desember 2017			
7	Jam	13:30:00 - 15:00:00			
8	Jenis Ujian	UTS			
9	Jenis Soal	Essai			
10					
11	No	NOBP	Nama	Kelas	Nilai
12	1	13101152610342	Gilang Kovalen	SI-1	50
13	2	14101152610544	Aidil	SI-1	80
14	3	14101152610545	Davidsyah Permana Putra	SI-1	90
15	4	14101152610555	Afdhal Trianda	SI-2	80
16	5	14101152610561	Cici Syuryantika	SI-2	100
17	6	14101152610562	Debby Puspitasari	SI-3	95
18	7	14101152610565	Ego Dafma Dasa	SI-3	60

**Gambar 4.54 Laporan Hasil Ujian Format *File* Excel**

*File* dengan format *excel* dapat dibuka dengan aplikasi seperti *microsoft office*, *wps office* dan *libreoffice*. Selain itu, *file* dengan format *excel* dapat memiliki *style* pada hasil cetaknya seperti tulisan tebal, tulisan dengan warna tertentu dll.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan pembahasan pada bab-bab sebelumnya, maka pada bab ini penulis dapat mengambil kesimpulan sebagai berikut :

1. Pada sistem lama, pelaksanaan ujian membutuhkan banyak kertas sebagai media soal dan jawaban. Dan kadang terjadi kasus kekurangan lembar soal atau lembar jawaban atau peserta membutuhkan lebih banyak lembar jawaban untuk menjawab soal.
2. Dengan menggunakan program berbasis web, maka proses pelaksanaan ujian dapat berlangsung tanpa kertas pada mata kuliah ujian tertentu. Dan hasil ujian untuk soal pilihan ganda dapat dilihat langsung oleh dosen.
3. Selain itu, jika soal ujian adalah essai maka dosen dapat memeriksa jawaban peserta secara langsung melalui aplikasi sehingga lebih hemat dari segi penggunaan kertas. Dan dosen tidak akan kesulitan dalam mengelola lembar jawaban peserta ujian karena semua jawaban sudah tersimpan didalam aplikasi.

#### **5.2 Saran-Saran**

Berdasarkan kesimpulan di atas maka dapat diberikan saran-saran sebagai berikut :

1. Dalam penerapan sistem baru sebaiknya didukung oleh perangkat yang memadai baik dari segi manusia (*brainware*) maupun dari segi peralatan (*hardware* dan *software*) dan perlu dilakukan pemeliharaan atau perawatan terhadap sistem tersebut.
2. Mengadakan pelatihan terhadap pihak terkait yang akan mengoperasikan sistem yang baru. Terutama bagi pihak dosen dan mahasiswa. Setelah dirasakan sistem yang baru dapat dioperasikan dengan baik dan pihak terkait telah mampu mengoperasikan sistem tersebut, maka sistem lama dapat digantikan dengan sistem yang baru secara keseluruhan.

## **Daftar Pustaka**

- Prahasta, Eddy. 2014. *Sistem Informasi geografis: Konsep-Konsep Dasar (Perspektif Geodesi dan Geomatika)*. Bandung : Informatika
- Rosa AS dan M.Shalahuddin. 2015. Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek. Bandung : INFORMATIKA
- Sidik, Betha. 2011. *Java Script*. Bandung : Informatika
- Sutabri, Tata. 2016. *Sistem Informasi Manajemen (Edisi Revisi)*. Yogyakarta : ANDI
- Sutanta, Edhy. 2011. *Basis Data Dalam Tinjauan Konseptual*. Yogyakarta : ANDI
- Rahmawati, Sri. 2014. *PENGOLAHAN DATA UNTUK KEAMANAN DATABASE AKADEMIK DENGAN METODE KRIPTOGRAFI MENGGUNAKAN BAHASA PEMROGRAMAN PHP DAN DATABASE MYSQL*. Padang: Majalah Ilmiah UPI YPTK. Volume 21, No.21
- Kaur, Sukhdeep. 2016. *The Implementation of Column-Oriented Database in Postgresql for Improving Performance of Queries*. International Journal of Research. Volume 03, No.4
- Alfan Rosid, Mochamad. 2017. *Implementasi JSON untuk Minimasi Penggunaan Jumlah Kolom Suatu Tabel Pada Database PostgreSQL*. JOINCS (Journal of Informatics, Network, and Computer Science). Volume 1, No.1
- Khotimah, Khusnul. 2016. *PENGEMBANGAN PROTOTIPE COMPUTER ASISTED TEST (CAT) MENGGUNAKAN ARSITEKTUR MODEL VIEW CONTROLLER PADA BADAN KEPEGAWAIAN NEGARA*. Padang: Universitas Muhammadiyah Jakarta. Volume 8, No.2

Muhammad Iqbal C. R., Muchammad Husni dan Hudan Studiawan. 2012. *Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan Node.js*. Surabaya : Institut Teknologi Sepuluh Nopember. Volume 1

I Ketut Buana dan Made Gede Wirakusuma. 2015. *PENGARUH PENGUNAAN SISTEM COMPUTER ASSISTED TEST PADA EFISIENSI BIAYA DAN AKUNTABILITAS PUBLIKASIAN HASIL (Studi pada Tes Seleksi Penerimaan CPNS di Kabupaten Jembrana dan Karangasem Tahun 2014)*. Bali : Fakultas Ekonomi dan Bisnis Universitas Udayana

## LISTING PROGRAM

```
// The following line loads the standalone build of Vue instead of the runtime-only
build,
// so you don't have to do: import Vue from 'vue/dist/vue'
// This is done with the browser options. For the config, see package.json
import Vue from 'vue'
import VueSession from 'vue-session'
import VueRouter from 'vue-router'
import App from './App.vue'
import routes from './routes.js'
import VeeValidate, { Validator } from 'vee-validate';
import bahasa from './node_modules/vee-validate/dist/locale/id.js'
import vSelect from 'vue-select'
import wysiwyg from "vue-wysiwyg";
import Notifications from 'vue-notification'
import store from 'store'
import axios from 'axios'
import lodash from 'lodash'
import { Bus } from './bus.js'
import vuecookies from 'vue-cookies'
import VueQuillEditor from 'vue-quill-editor'
import fp2 from 'fingerprintjs2'
```

```
import jwt_decode from 'jwt-decode'

Vue.use(VueQuillEditor)

Vue.use(wysiwyg, {
  hideModules: { "bold":false,
  "headings":true,
  "hyperlink":true,
  "image":true,
  "italic":false,
  "list_ordered":false,
  "list_unordered":false,
  "removeFormat":false,
  "table":false
}
})

Vue.use(Notifications)

Vue.component('v-select', vSelect)

VeeValidate.Validator.addLocale(bahasa)

Vue.use(VueRouter);

Vue.use(VueSession);

Vue.use(VeeValidate, {locale: 'id', delay: "1000"});

const cks = {

  setCookies (x,y,z = null){

```

```
z == null ? vuecookies.set(x,y) : vuecookies.set(x,y,z)

},

getCookies (x){

    return jwt_decode(vuecookies.get(x)).data

},

clearCookies (x){

    vuecookies.remove(x)

},

isCookies (x){

    return vuecookies.isKey(x)

}

}

const localStorage = {

    setLcs (x,y){

        store.set(x,y)

    },

    getLcs (x){

        return store.get(x)

    },

    clearLcs (){

        store.clearAll()

    },

    removeLcs (x){


```

```
        store.remove(x)

    }

}

Vue.prototype.$cekLogin = null

Vue.prototype.$lcs = localStorage

Vue.prototype.$cks = cks

Vue.prototype.$fp2 = new fp2({

    excludeUserAgent : true,
    excludeLanguage : true,
    excludeColorDepth : true,
    excludeScreenResolution : true,
    excludeAvailableScreenResolution : true,
    excludeSessionStorage : true,
    excludePlatform : true
})

axios.interceptors.request.use((config)=>{

    if(vuecookies.isKey('infoLogin')){

        config.headers.common['Authorization'] = 'Bearer ' +
        vuecookies.get('infoLogin')

    }

    return config;
}, (error)=>{

    // Do something with request error
})
```

```
        return Promise.reject(error);
    });

window.bus = Bus

window._ = lodash

Vue.prototype.$ajx = axios

// We create te router instance here.

const router = new VueRouter({

    routes: routes

});

new Vue({ // eslint-disable-line no-new

    el: '#app',

    router,

    render: (h) => h(App)

})

var express = require('express');

var path = require('path');

var favicon = require('serve-favicon');

var logger = require('morgan');

var cookieParser = require('cookie-parser');

var bodyParser = require('body-parser');

var validator = require('express-validator');

var fileUpload = require('express-fileupload');

var Promise = require('promise');
```

```
var url = require('url');

var knexLogger = require('knex-logger');

const expressGraphQL = require('express-graphql');

var jwt = require('jsonwebtoken')

const sMahasiswa = require('./schema/mahasiswa.js');

const sJujian = require('./schema/jenis_ujian.js');

const sKuliah = require('./schema/kuliah.js');

const sStatistik = require('./schema/statistik.js');

const sDosen = require('./schema/dosen.js');

const sUser = require('./schema/user.js');

const sUjian = require('./schema/ujian.js');

db = require('knex')({  
  client: 'pg',  
  connection: process.env.DATABASE_URL  
});  
process.env.TZ= 'Asia/Jakarta'  
var app = express();  
{}  
// view engine setup  
app.set('views', path.join(__dirname, 'views'));  
app.set('view engine', 'jade');  
app.disable('X-Powered-By');
```

```
// uncomment after placing your favicon in /public
//app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));

app.use(knexLogger(db));

app.use(logger('dev'));

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: false }));

app.use(cookieParser());

app.use(express.static(path.join(__dirname, 'public')));

app.use(fileUpload());

app.use('/^/api/:params*',function(req,res,next){

  let token = req.header('Authorization')

  res.set({'Access-Control-Allow-Origin' : '*'});

  if(token){

    jwt.verify(token.split(" ")[1], 'Panther A 7.5cm kwk 42 L/70', function(err,
decoded) {

      if(err) res.status(401).json({status:false,error:"Anda harus login terlebih
dahulu!"})

      else next()

    })

  }else res.status(401).json({status:false,error:"Anda harus login terlebih
dahulu!"})

});
```

```
app.use.validator({  
    customValidators : require('./validator/custom_validator')  
});  
  
//ROUTES  
  
app.use('/login', require('./routes/login'));  
  
app.use('/api/soal', require('./routes/soal'));  
  
app.use('/api/ujian', require('./routes/ujian'));  
  
app.use('/api/mahasiswa', require('./routes/mahasiswa'));  
  
app.use('/api/matkul', require('./routes/matkul'));  
  
app.use('/api/kelas', require('./routes/kelas'));  
  
app.use('/api/jenis_ujian', require('./routes/jenis_ujian'));  
  
app.use('/api/jenis_soal', require('./routes/jenis_soal'));  
  
app.use('/api/dosen', require('./routes/dosen'));  
  
app.use('/api/kuliah', require('./routes/kuliah'));  
  
app.use('/api/cek', require('./routes/cekPesertaUjian'));  
  
app.use('/api/user', require('./routes/user'));  
  
app.use('/api/jenis_user', require('./routes/jenis_user'));  
  
app.use('/api/v2/mahasiswa', expressGraphQL({  
    schema: sMahasiswa,  
    graphiql: true,  
});  
  
app.use('/api/v2/jenis_ujian', expressGraphQL({
```

```
schema: sJujian,
graphiql: true,
}));  
  
app.use('/api/v2/kuliah', expressGraphQL({  
  schema: sKuliah,  
  graphiql: true,  
}));  
  
app.use('/api/v2/statistik', expressGraphQL({  
  schema: sStatistik,  
  graphiql: true,  
}));  
  
app.use('/api/v2/dosen', expressGraphQL({  
  schema: sDosen,  
  graphiql: true,  
}));  
  
app.use('/api/v2/user', expressGraphQL({  
  schema: sUser,  
  graphiql: true,  
}));  
  
app.use('/api/v2/ujian', expressGraphQL({  
  schema: sUjian,  
  graphiql: true,  
}));
```

```
// ^.+\\(api)\\.+$  
//EOF ROUTES  
  
// catch 404 and forward to error handler  
  
app.use(function(req, res, next) {  
  
  var err = new Error('Not Found');  
  
  err.status = 404;  
  
  next(err);  
  
});  
  
// error handler  
  
app.use(function(err, req, res, next) {  
  
  // set locals, only providing error in development  
  
  res.locals.message = err.message;  
  
  res.locals.error = req.app.get('env') === 'development' ? err : {};  
  
  // render the error page  
  
  res.status(err.status || 500);  
  
  res.render('error');  
  
});  
  
module.exports = app;
```

```
<template>

<div id="genForm">

<div class="w3-modal" :style="showForm ? 'display:block;' : 'display:none;'">

<div class="w3-modal-content">

<div class="w3-card-8 w3-container w3-section">

<template v-if="contentType == 'form'">

<form v-if="showForm" id="addData" @submit.prevent="submitData()"
name="addData" method="POST">

<h3>{{edit ? 'Edit Data' : 'Tambah Data'}}</h3>

<span v-for="x in input">

<template v-if="x.jenis == 'textField'">

<template v-if="x.tipe == 'number'">

<label>{{x.caption}}</label>

<input :disabled="x.disabled || false" class="w3-input w3-
border" :placeholder="x.caption"
type="number" :name="x.name" :min="x.min" :max="x.max" v-
model.number="output[x.name]" />

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<template v-else-if="x.tipe == 'email'">

<label>{{x.caption}}</label>
```

```

<input :disabled="x.disabled || false" class="w3-input w3-
border" :placeholder="x.caption" type="email" :name="x.name" v-
model="output[x.name]" />

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<template v-else>

<label>{{ x.caption }}</label>

<input :disabled="x.disabled || false" class="w3-input w3-
border" :placeholder="x.caption" :type="x.tipe" :name="x.name" v-
model="output[x.name]" />

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<br/>

</template>

<template v-else-if="x.jenis == 'textArea'">

<label>{{ x.caption }}</label>

<textarea :disabled="x.disabled || false" class="w3-input w3-
border" :placeholder="x.caption" :name="x.name" v-model="output[x.name]">
{{ x.value }}</textarea>

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<template v-else-if="x.jenis == 'radioButton'">

<label for="x.name" v-for="(y,index,key) in x.option">

```

```

<span v-if="index < 1">{ {x.caption}}<br/></span>

<input class="w3-radio" :disabled="x.disabled ||

false" :value="y[x.valueSelect]" type="radio" :name="x.name" v-
model="output[x.name]" /> { {y[x.captionSelect]}}<br/>

</label>

<span class="w3-text-red">{ { error[x.name] }}</span>

</template>

<template v-else-if="x.jenis == 'selectOption'">

<label>{ {x.caption}}</label>

<select class="w3-select w3-white w3-border w3-text-
gray" :disabled="x.disabled || false" :name="x.name" v-model="output[x.name]">

<template v-if="x.option.length == 0">

<option class="w3-white w3-text-gray" value="null"-

disabled>{ {x.caption}}</option>

</template>

<template v-else>

<template v-for="(y,index,key) in x.option">

<template v-if="index == 0">

<option class="w3-white w3-text-gray" value="null"-

disabled>{ {x.caption}}</option>

<option

:value="y[x.valueSelect]">{ {y[x.captionSelect]}}</option>

</template>

```

```
<template v-else>

    <option
        :value="y[x.valueSelect]">{{ y[x.captionSelect] }}</option>

    </template>

</template>

</select>

<span class="w3-text-red">{{ error[x.name] }}</span><br/>

</template>

<template v-else-if="x.jenis == 'select2'">

    <v-select :valueSelect="x.valueSelect" v-
        model="output[x.name]" :label="x.captionSelect" :placeholder="x.placeholder" :o-
        ptions="x.option"></v-select>

    <span class="w3-text-red">{{ error[x.name] }}</span>

    </template>

<template v-else-if="x.jenis == 'richEditor'">

    <wysiwyg v-model="output[x.name]" />

    <span class="w3-text-red">{{ error[x.name] }}</span>

    </template>

<template v-else-if="x.jenis == 'datePick'">

    <label>{{ x.caption }}</label>
```

```

<datepicker language="id" placeholder="Pilih
Tanggal ..." :value="output[x.value]" input-class="w3-input w3-border" v-
model="output[x.name]"></datepicker>

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<template v-else-if="x.jenis == 'timePick'">

<label>{{ x.caption }}</label>

<time-picker v-model="output[x.name]"></time-picker>

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<template v-else-if="x.jenis == 'selectize'">

<label>{{ x.caption }}</label>

<selectize v-model="output[x.name]" :settings="x"></selectize>

<span class="w3-text-red">{{ error[x.name] }}</span>

</template>

<br/>

</span>

<button type="submit" :disabled="Bsimpan.disabled" class="w3-button
w3-teal w3-section"><span v-html="Bsimpan.caption"></span></button>

<button type="button" @click="resetForm()" class="w3-button w3-
reset w3-red w3-section" :disabled="Breset.disabled"><span v-
html="Breset.caption"></span></button>

```

```
<button type="button" class="w3-button w3-blue w3-section"
@click="cancelForm()" :disabled="Bbatal.disabled"><span v-
html="Bbatal.caption"></span></button>

</form>

</template>

<template v-else>

<slot></slot>

</template>

</div>

</div>

</div>

</div>

</div>

</template>

<script>

import Datepicker from 'vuejs-datepicker';

import TimePicker from 'vue-timepicker'

import selectize from 'vue2-selectize'

export default {

  name : 'genForm',

  components : {

    selectize, Datepicker, TimePicker
```

```
  },  
  props : {  
    url : String,  
    input : Array,  
    pk : {  
      type : String,  
      required : false,  
      default : 'id'  
    },  
    contentType : {  
      type : String,  
      required : false,  
      default : "form"  
    }  
  },  
  data () {  
    return {  
      showForm : false,  
      idData : null,  
      output : {},  
      error : {},  
      Bsimpan : {  
        disabled : false,  
        value : null  
      }  
    }  
  }  
}
```

```
caption : '<i class="fa fa-save w3-small"></i> Simpan',  
},  
  
Bbatal : {  
  
    disabled : false,  
  
    caption : '<i class="fa fa-remove w3-small"></i> Batal'  
},  
  
Breset : {  
  
    disabled : false,  
  
    caption : '<i class="fa fa-repeat w3-small"></i> Reset'  
},  
  
edit : false  
  
}  
  
},  
  
created () {  
  
    bus.$on('getDataDetail', x =>{  
  
        this.getDataDetail(x)  
    })  
  
    bus.$on('toggleFormData', () =>{  
  
        this.toggleFormData()  
    })  
  
    this.showForm = false  
  
    _.forEach(this.input,(v,k)=>{  
  
        this.$set(this.error,v.name, null)  
    })  
}
```

```
  })

//SET DAFTAR V-MODEL DI OBJECT OUTPUT{}

_.forEach(this.input,(v,k)=>{

  this.output[v.name] = v.value || null //looping ke object input dan ambil
  properti name yang merupakan vmodel

  this.error[v.name] = null //looping ke object input dan ambil properti name
  yang merupakan error

})

//EOF SET DAFTAR V-MODEL DI OBJECT OUTPUT{}

},  
methods : {  
  resetForm (){  
    var prop = Object.keys(this.error)  
    _.forEach(prop, (v,k)=>{  
      this.error[v] = null  
      this.output[v] = null  
    })  
  },  
  resetError () {  
    var prop = Object.keys(this.error)  
    _.forEach(prop, (v,k)=>{  
      this.error[v] = null  
    })  
  }  
}
```

```
    },  
  
    cancelForm () {  
  
        this.resetForm()  
  
        this.edit = false  
  
        this.showForm = false  
  
        this.buttonSubmit(0)  
  
        this.idData = null  
  
    },  
  
    buttonSubmit (x) {  
  
        if(x == 1){  
  
            this.Bsimpan.disabled = true  
  
            this.Bsimpan.caption = '<i class="fa fa-spinner w3-spin w3-small"></i>  
Menyimpan Data'  
  
            this.Breset.disabled = true  
  
            this.Bbatal.disabled = true  
  
        }else{  
  
            this.Bsimpan.disabled = false  
  
            this.Bsimpan.caption = '<i class="fa fa-save w3-small"></i> Simpan'  
  
            this.Breset.disabled = false  
  
            this.Bbatal.disabled = false  
  
        }  
  
    },  
  
    toggleFormData (){
```

```
        this.showForm = !this.showForm
        this.resetForm()
        this.buttonSubmit(0)
    },
    submitData () {
        this.buttonSubmit(1)
        this.resetError()
        if(this.edit == false){
            var method = 'POST'
            var url = ""
        }else {
            var method = 'PUT'
            var url = '/' +this.idData
        }
        this.$ajx({
            method : method,
            data : this.output,
            url : '/api/' +this.url +url,
        })
        .then(res=>{
            bus.$emit('newData')
            this.toggleFormData()
            this.buttonSubmit(0)
        })
    }
}
```

```
        this.edit = false
    })
    .catch(err=>{
    var kode = err.response.status
    if(kode == 422){
        var listError = Object.keys(err.response.data.error)
        _.forEach(listError,(v,k)=>{
            this.error[v] = err.response.data.error[v].msg
        })
    }else if(kode >= 500){
        bus.$emit('showAlert','Kesalahan!','Terjadi kesalahan pada
server.','bottom right','danger')
    }
    this.buttonSubmit(0)
})
},
getDataDetail (x){
    this.edit = true
    this.$ajax.get('/api/'+this.url+'/'+x)
    .then(res=>{
        this.output = res.data.data[0]
        this.showForm = !this.showForm
        this.idData = this.output[this.pk]
    })
}
```

```
  })
  .catch(err=>{
    var kode = err.response.status
    if(kode > 300){
      bus.$emit('showAlert','Peringatan!','Tidak dapat mengambil data.
      Silahkan ulangi kembali.','warning')
    }else{
      bus.$emit('showAlert','Peringatan!','Terjadi kesalahan pada
      server!','warning')
    }
  })
  },
  destroyed () {
    //bus.$off('getDataDetail')
  }
}

</script>
```

```
<style scoped>
```

```
</style>
```