

**PEMBANGUNAN APLIKASI PENILAIAN UJIAN SKRIPSI  
BERBASIS ANDROID DENGAN MENGGUNAKAN METODE  
PROTOTYPING  
(Studi Kasus: Fakultas Ilmu Komputer Universitas Brawijaya  
Malang)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh :  
Edwin Nurwansyah  
NIM: 145150401111080



PROGRAM STUDI SISTEM INFORMASI  
JURUSAN SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PEMBANGUNAN APLIKASI PENILAIAN UJIAN SKRIPSI BERBASIS ANDROID  
DENGAN MENGGUNAKAN METODE PROTOTYPING  
(STUDI KASUS : FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA MALANG)

### SKRIPSI

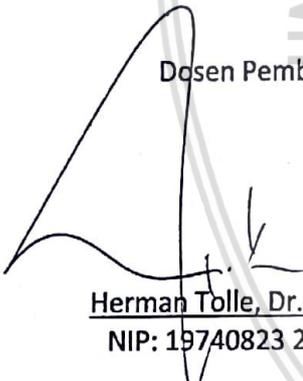
Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

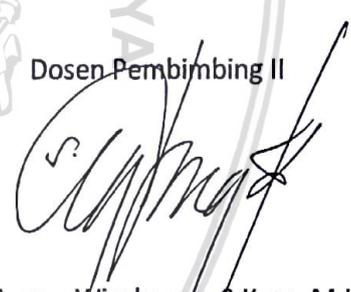
Disusun Oleh :  
Edwin Nurwansyah  
NIM: 145150401111080

Skripsi ini telah diuji dan dinyatakan lulus pada  
05 Juli 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

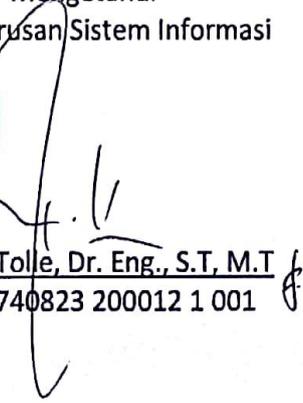
  
Herman Tolle, Dr. Eng., S.T, M.T  
NIP: 19740823 200012 1 001

  
Satrio Agung Wicaksono, S.Kom, M.Kom  
NIP: 19860521 201212 1 001

Mengetahui

Ketua Jurusan Sistem Informasi



  
Herman Tolle, Dr. Eng., S.T, M.T  
NIP: 19740823 200012 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Juli 2018



Edwin Nurwansyah

NIM: 145150401111080

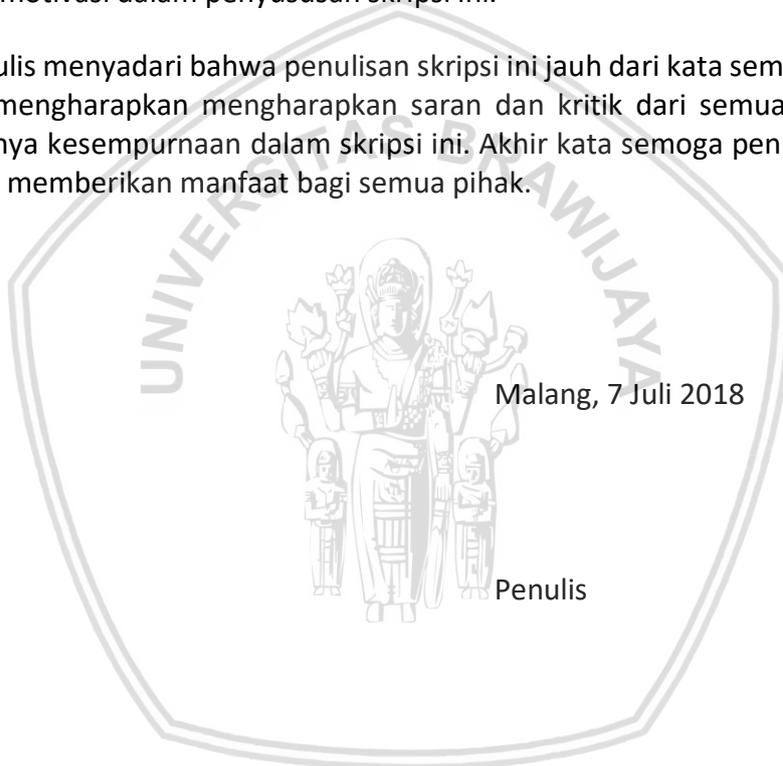
## KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Allah SWT karena atas berkat rahmat dan hidayah-Nya lah maka skripsi yang berjudul “Pembangunan Aplikasi Penilaian Ujian Skripsi Denga Menggunakan Metode Prototyping (Studi Kasus : Fakultas Ilmu Komputer Universitas Brawijaya) dapat saya selesaikan. Dengan selesainya penulisan skripsi ini, maka saya selaku penulis ingin menyampaikan ucapan terima kasih sebesar-besarnya pada pihak-pihak berikut yang telah mendukung, yaitu:

1. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku dosen pembimbing 1 yang telah membimbing dan memberikan ilmu, kritik dan sarannya dalam penyusunan skripsi.
2. Bapak Satrio Agung Wicaksono, S.Kom, M.Kom selaku dosen pembimbing 2 yang telah membimbing dan memberikan ilmu, kritik dan sarannya dalam penyusunan skripsi.
3. Bapak Aryo Pinandito, S.T, M.MT selaku dosen penasehat akademik yang telah memberikan ilmu, kritik dan sarannya selama perkuliahan.
4. Bapak Denny Sagita Rusdianto, S.Kom,M.Kom selaku ketua PSIK yang telah memberikan izin dan bantuannya untuk melakukan penelitian terkait FILKOM APPS.
5. Ibu Thusti Dwi Purnami, A.Md selaku pengembang FILKOM APPS yang telah bersedia meluangkan waktunya untuk diwawancara sehingga peneliti dapat memperoleh data untuk penelitian ini.
6. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc, Bapak Fitra Abdurrachman Bachtiar, Dr.Eng., S.T, M.Eng, Bapak Himawat Aryadita, S.T, M.Sc, Bapak Lutfi Fanani, S.Kom., M.T., M.Sc, Bapak Mahardeka Tri Ananta, S.Kom., M.T., M.Sc, Ibu Ratih Kartika Dewi, S.T., M.Kom, Ibu Retno Indah Rokhmawati, S.Pd., M.Pd, dan Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng yang telah bersedia meluangkan waktunya untuk memberikan masukan dan saran terhadap aplikasi yang dibangun dalam skripsi ini.
7. Orang tua, kakak, dan seluruh keluarga besarr yang telah memotivasi dan memberikan dukungan kepada penulis.

8. Mas Naufal, Mas Oddy, Mas Imam dan Mas Arel yang telah memberikan masukan dan motivasinya dalam menyusun skripsi.
9. Aldriyan Dicky, Annisa Nurfitri, Annisa Mursyidah, Dimas Haryanto , Dinda Agnes, Firnanda Iftitah, Higam Saiful, Nadya Mardiana, dan Windi Prahasetyo yang telah menemani perjalanan semasa kuliah hingga akhir serta memberikan motivasi kepada penulis.
10. Jawara, Danu, Danniar, Rosalia serta teman-teman Sistem Informasi 2014 yang tidak dapat saya sebutkan namanya yang telah memberikan masukan dan motivasi dalam penyusunan skripsi ini.

Penulis menyadari bahwa penulisan skripsi ini jauh dari kata sempurna. Maka penulis mengharapkan mengharapkan saran dan kritik dari semua pihak demi tercapainya kesempurnaan dalam skripsi ini. Akhir kata semoga penulisan skripsi ini dapat memberikan manfaat bagi semua pihak.



Malang, 7 Juli 2018

Penulis

## ABSTRAK

Fakultas Ilmu Komputer merupakan salah satu fakultas di Universitas Brawijaya Malang yang menjalankan proses bisnisnya dengan menggunakan sistem. Sistem yang dinamakan FILKOM APPS telah beroperasi untuk menangani proses akademik. Akan tetapi, keberadaan sistem ini masih kurang optimal dan belum menyesuaikan kebutuhan pengguna saat ini, yaitu kebutuhan adanya sistem berbasis perangkat bergerak. Hal ini dikarenakan pengguna lebih sering menggunakan perangkat bergerak dalam menggunakan sistem tersebut dan ketika itu pengguna merasa tampilan yang dari sistem ini kurang menyesuaikan tampilan di perangkat bergerak, sehingga membuat pengguna kesulitan dalam mengakses sistem melalui perangkat bergerak mereka. Maka dari itu diperlukannya sistem yang dapat berjalan di perangkat bergerak untuk menyesuaikan kebutuhan pengguna saat ini. Pembangunan sistem berbasis perangkat bergerak menggunakan siklus pengembangan *prototyping* dengan pendekatan *object-oriented*. Pembangunan sistem berbasis perangkat bergerak ini hanya dilakukan pada sistem operasi android dan menggunakan pendekatan pengembangan perangkat bergerak *native*. Tingkat ketermudahan penggunaan (*usability*) menjadi salah satu hal yang harus diperhatikan dalam mengembangkan sistem berbasis perangkat bergerak. Hasil yang didapatkan dari penelitian ini berupa rancangan dan implementasi sistem serta tingkat ketermudahan (*usability*) yang terdiri dari tiga aspek, yaitu 100% pada aspek *effectiveness*, 0.169 pada aspek *efficiency* dan 6.1 pada *satisfaction*.

Kata kunci : Aplikasi perangkat bergerak, *Usability*, *Prototyping Model*, *Native Apps*, *Object Oriented*, *Model-View-Presenter*, Sistem Informasi,

## ABSTRACT

Faculty of Computer Science is one of the faculty at Brawijaya University Malang which runs its business process using the system. The system called FILKOM APPS has operated to handle the academic process. However, the existence of this system is still less than optimal and has not adjusted the needs of current users. One of them needs a mobile device-based system. This is because users more often use mobile devices to use the system and when the user feels the display of this system is less to adjust the display on mobile devices, thus making the user difficulty in accessing the system through their mobile device. Therefore the need for systems that can run on mobile devices to match the needs of today's users. The system is built in advance on one of the features, namely the assessment of thesis exams. The development of mobile device-based systems uses a prototyping development cycle with an object-oriented approach. The development of this mobile device based system is only done on the android operating system and using the native mobile device development approach. The usability level is one of the things that must be considered in developing a mobile device-based system. The results obtained from this research are the design and implementation of the system and the usability level consisting of three aspects, namely 100% on effectiveness aspect, 0.169 on efficiency and 6.1 aspects of satisfaction.

*Keywords: Mobile Application, Usability, Prototyping Model, Native Apps, Object Oriented, , Model-View-Presenter, Information System*

## DAFTAR ISI

PERSETUJUAN .....	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR .....	iv
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xviii
DAFTAR LAMPIRAN .....	xx
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	4
1.5 Batasan masalah .....	4
1.6 Sistematika pembahasan.....	4
<b>BAB 2 LANDASAN KEPUSTAKAAN.....</b>	<b>6</b>
2.1 Kajian Pustaka .....	6
2.2 Profil Organisasi .....	7
2.3 Mekanisme Dan Prosedur Pelaksanaan Ujian Skripsi .....	7
2.3.1 Standard Penilaian Kelulusan Skripsi .....	12
2.4 Sistem Informasi .....	14
2.5 Perangkat Bergerak Dan Android .....	15
2.6 Metode <i>Prototyping</i> .....	18
2.7 <i>Unified Modeling Language</i> .....	20
2.7.2 Proses Bisnis .....	24
2.8 <i>Physical Data Model</i> .....	29
2.9 <i>Web Seivice</i> .....	30
2.10 <i>JavaScript Object Notation (JSON)</i> .....	30
2.11 Pengujian Perangkat Lunak .....	31
2.11.1 <i>Black-box Testing</i> .....	32



2.11.2 Usability Testing .....	32
2.11.3 Security Testing.....	34
2.12 Model View Presenter (MVP) .....	34
2.13 A/B Testing.....	35
2.14 Framework Lumen .....	35
2.14.1 Eloquent .....	36
2.15 Sampling Purposive .....	36
2.16 Skala Likert .....	36
<b>BAB 3 METODOLOGI .....</b>	<b>38</b>
3.1 Studi Literatur .....	38
3.2 Pengumpulan Data.....	38
3.3 Analisis dan Perancangan .....	39
3.4 Implementasi Prototyping .....	39
3.5 Evaluasi Prototyping.....	39
3.6 Implementasi dan Pengujian Sistem .....	39
3.7 Penutup .....	39
<b>BAB 4 ANALISIS .....</b>	<b>40</b>
4.1 Analisis Proses Bisnis .....	40
4.1.1 Penilaian Komponen Proses Pengerjaan Skripsi Dan Kualitas Penulisan Skripsi.....	40
4.1.2 Pelaksanaan Ujian Skripsi.....	41
4.1.3 Pengecekan Kehadiran Majelis.....	44
4.1.4 Kode dan Aktivitas Proses Bisnis .....	44
4.2 Analisis Kebutuhan .....	45
4.2.1 Penggalan Informasi Alur Kerja Sistem <i>Existing</i> .....	46
4.2.2 Identifikasi Pengguna .....	47
4.2.3 Kebutuhan Fungsional .....	48
4.2.4 Kebutuhan Non-Fungsional.....	50
4.2.5 Pemodelan Use Case <i>Diagram</i> .....	51
4.2.6 Pemodelan Use Case Scenario .....	52
<b>BAB 5 Perancangan.....</b>	<b>62</b>
5.1.1 Perancangan Arsitektur Sistem .....	62
5.1.2 Perancangan <i>Activity Diagram</i> .....	62



5.1.3 Perancangan <i>Sequence Diagram</i> .....	66
5.1.4 Perancangan <i>Class Diagram</i> .....	72
5.1.5 Perancangan Basis Data .....	75
5.1.6 Perancangan API .....	85
5.1.7 Perancangan Antarmuka Pengguna .....	87
5.1.8 Perancangan Algoritme .....	94
<b>BAB 6 EVALUASI PROTOTIPE</b> .....	<b>97</b>
6.1 Iterasi Satu .....	97
6.1.1 Perancangan Antarmuka Pengguna .....	99
6.2 Iterasi Dua .....	105
6.2.1 <i>Effectiveness</i> .....	106
6.2.2 <i>Efficiency</i> .....	107
6.2.3 <i>Satisfaction</i> .....	108
<b>BAB 7 IMPLEMENTASI</b> .....	<b>109</b>
7.1.1 Spesifikasi Sistem .....	109
7.1.2 Batasan Implementasi .....	110
7.1.3 Implementasi Aplikasi .....	110
7.1.4 Implementasi Web Service .....	118
7.1.5 Implementasi Basis Data .....	121
7.1.6 Implementasi Antarmuka .....	129
<b>BAB 8 PENGUJIAN DAN ANALISIS</b> .....	<b>133</b>
8.1 Pengujian Fungsional .....	133
8.1.1 Pengujian Fungsional Login .....	133
8.1.2 Pengujian Fungsional Melihat Notifikasi .....	135
8.1.3 Pengujian Fungsional Melihat Daftar Jadwal Ujian Skripsi .....	135
8.1.4 Pengujian Fungsional Melihat Rincian Ujian Skripsi .....	137
8.1.5 Pengujian Fungsional Melihat Dokumen Skripsi .....	137
8.1.6 Pengujian Fungsional Memverifikasi Kehadiran Mahasiswa, Dosen Pembimbing Dan Dosen Penguji .....	138
8.1.7 Pengujian Fungsional Memberikan Penilaian .....	139
8.1.8 Pengujian Fungsional Memperbarui Penilaian .....	140
8.1.9 Pengujian Fungsional Melihat Rekap Nilai .....	141
8.1.10 Pengujian Fungsional Melihat <i>Gap</i> Nilai .....	141

8.1.11 Pengujian Fungsional Melihat Rincian Penilaian Dosen Pembimbing Dan Dosen Penguji .....	142
8.1.12 Pengujian Fungsional Membuat Berita Acara .....	143
8.1.13 Pengujian Fungsional Melihat Daftar Riwayat Ujian Skripsi ...	145
8.1.14 <i>Pengujian Fungsional Melihat Rincian Riwayat Ujian Skripsi</i> .	145
8.1.15 Pengujian Fungsional Menjalankan Pengatur Waktu Ujian....	146
8.2 Pengujian Non-Fungsional .....	147
8.2.1 Pengujian Keamanan .....	147
8.2.2 Pengujian <i>Usability</i> .....	151
8.3 Analisis Hasil Pengujian .....	153
BAB 9 PENUTUP .....	155
9.1 Kesimpulan .....	155
9.2 Saran .....	155
DAFTAR PUSTAKA .....	157



## DAFTAR TABEL

Tabel 2.1 Nilai akhir skripsi.....	14
Tabel 2.2 <i>Method lifecycle android</i> .....	16
Tabel 2.3 Notasi pada <i>use case</i> .....	21
Tabel 2.4 Notasi pada activity diagram.....	22
Tabel 2.5 Notasi pada class diagram.....	23
Tabel 2.6 Notasi pada <i>sequence diagram</i> .....	23
Tabel 2.7 Notasi BPMN .....	27
Tabel 2.8 Contoh penerapan eloquent.....	36
Tabel 2.9 Bobot Nilai Jawaban .....	37
Tabel 4.1 Kode proses bisnis .....	45
Tabel 4.2 Kode aktivitas proses bisnis .....	45
Tabel 4.3 Pengguna aplikasi penilaian ujian skripsi.....	47
Tabel 4.4 Analisis kebutuhan fungsional dosen pembimbing.....	48
Tabel 4.5 Analisis kebutuhan fungsional dosen penguji.....	48
Tabel 4.6 Analisis kebutuhan fungsional ketua majelis .....	49
Tabel 4.7 Analisis kebutuhan non-fungsional .....	50
Tabel 4.8 <i>Use Case Scenario</i> Login.....	52
Tabel 4.9 <i>Use Case Scenario</i> Melihat Notifikasi .....	52
Tabel 4.10 <i>Use Case Scenario</i> Melihat Notifikasi (Lanjutan).....	53
Tabel 4.11 <i>Use Case Scenario</i> Melihat Daftar Jadwal Ujian Skripsi.....	53
Tabel 4.12 <i>Use Case Scenario</i> Melihat Rincian Ujian Skripsi.....	54
Tabel 4.13 <i>Use Case Scenario</i> Melihat Dokumen Skripsi .....	54
Tabel 4.14 <i>Use Case Scenario</i> Memverifikasi Kehadiran Mahasiswa, Dosen Pembimbing Dan Dosen Penguji .....	55
Tabel 4.15 <i>Use Case Scenario</i> Memberikan Penilaian .....	55
Tabel 4.16 <i>Use Case Scenario</i> Memberikan Penilaian (Lanjutan) .....	56
Tabel 4.17 <i>Use Case Scenario</i> Memperbarui Penilaian .....	56
Tabel 4.18 <i>Use Case Scenario</i> Melihat Rekap Nilai.....	56
Tabel 4.19 <i>Use Case Scenario</i> Melihat Rekap Nilai (Lanjutan).....	57
Tabel 4.20 <i>Use Case Scenario</i> Melihat <i>Gap</i> Nilai .....	57
Tabel 4.21 <i>Use Case Scenario</i> Melihat Rincian Penilaian Dosen Pembimbing dan Dosen Penguji.....	57

abel 4.22 <i>Use Case Scenario</i> Melihat Rincian Penilaian Dosen Pembimbing dan Dosen Penguji (Lanjutan) .....	58
Tabel 4.23 <i>Use Case Scenario</i> Membuat Berita Acara Ujian Telah Selesai.....	58
Tabel 4.24 <i>Use Case Scenario</i> Membuat Berita Acara Ujian Telah Selesai (Lanjutan) .....	59
Tabel 4.25 <i>Use Case Scenario</i> Membuat Berita Acara Penjadwalan Ulang .....	59
Tabel 4.26 <i>Use Case Scenario</i> Melihat Daftar Riwayat Ujian Skripsi .....	60
Tabel 4.27 <i>Use Case Scenario</i> Melihat Rincian Riwayat Ujian Skripsi .....	60
Tabel 4.28 <i>Use Case Scenario</i> Melihat Rincian Riwayat Ujian Skripsi (Lanjutan) ..	61
Tabel 4.29 <i>Use Case Scenario</i> Menjalankan Pengatur Waktu Ujian.....	61
Tabel 5.1 Rancangan tabel basis data dosen .....	75
Tabel 5.2 Rancangan tabel basis data mahasiswa.....	76
Tabel 5.3 Rancangan tabel basis data skripsi.....	76
Tabel 5.4 Rancangan tabel basis data bidang penelitian.....	77
Tabel 5.5 Rancangan tabel basis data tipe penelitian .....	77
Tabel 5.6 Rancangan tabel basis data keminatan .....	77
Tabel 5.7 Rancangan tabel basis data mata kuliah pendukung skripsi .....	78
Tabel 5.8 Rancangan tabel basis data mata kuliah .....	78
Tabel 5.9 Rancangan tabel basis data jurusan .....	78
Tabel 5.10 Rancangan tabel basis data program studi.....	79
Tabel 5.11 Rancangan tabel basis data bidang dosen.....	79
Tabel 5.12 Rancangan tabel basis data bimbingan dosen.....	79
Tabel 5.13 Rancangan tabel basis data nilai skripsi.....	80
Tabel 5.14 Rancangan tabel basis data nilai semhas.....	80
Tabel 5.15 Rancangan tabel basis data info ujian skripsi .....	80
Tabel 5.16 Rancangan tabel basis data jadwal dosen .....	81
Tabel 5.17 Rancangan tabel basis data berita acara reschedule .....	81
Tabel 5.18 Rancangan tabel basis data status kelulusan.....	81
Tabel 5.19 Rancangan tabel basis data komponen penilaian .....	82
Tabel 5.20 Rancangan tabel basis data indikator penilaian.....	82
Tabel 5.21 Rancangan tabel basis data pengajuan penguji pengganti.....	82
Tabel 5.22 Rancangan tabel basis data status ujian .....	83
Tabel 5.23 Rancangan tabel basis data berita acara .....	83

Tabel 5.24 Perancangan API memverifikasi kehadiran .....	85
Tabel 5.25 Perancangan API memberikan penilaian .....	86
Tabel 5.26 Perancangan API membuat berita acara .....	87
Tabel 5.27 Perancangan Algoritme memverifikasi kehadiran mahasiswa, pembimbing dan penguji .....	94
Tabel 5.29 Perancangan algoritme fungsi memeriksa <i>gap</i> nilai .....	95
Tabel 5.30 Perancangan algoritme fungsi rata-rata nilai skripsi pembimbing .....	96
Tabel 5.31 Perancangan algoritme membuat berita acara .....	96
Tabel 6.1 Kuisisioner <i>A/B Testing</i> .....	97
Tabel 6.2 Hasil uji coba <i>A/B Testing</i> .....	99
Tabel 6.3 Daftar <i>Task</i> Pengguna Berdasarkan Fungsional .....	106
Tabel 6.4 Data <i>effectiveness</i> berdasarkan fitur .....	106
Tabel 6.5 Kesalahan pengguna.....	107
Tabel 6.6 Waktu yang dibutuhkan pengguna dalam mengerjakan task berdasarkan fitur .....	107
Tabel 6.7 <i>Efficiency</i> aplikasi penilaian ujian skripsi .....	108
Tabel 6.8 Hasil penilaian kuisisioner Single Ease Question .....	108
Tabel 7.1 Spesiifikasi perangkat keras (Server) .....	109
Tabel 7.2 Spesifikasi perangkat lunak.....	109
Tabel 7.3 Implementasi kode program memverifikasi kehadiran pada <i>class</i> verifikasiactivity .....	110
Tabel 7.4 Implementasi Kode Program Memverifikasi Kehadiran Pada <i>Class</i> verifikasipresenter .....	111
Tabel 7.5 Implementasi memberikan kode program penilaian pada kelas indikatorpresenter .....	113
Tabel 7.6 Implementasi kode program memberikan penilaian pada kelas penilaianpresenter.....	114
Tabel 7.7 Implementasi kode program merekap nilai.....	115
Tabel 7.8 Implementasi kode program membuat berita acara .....	117
Tabel 7.9 Implementasi <i>Web Service</i> .....	118
Tabel 7.10 Implementasi tabel berita acara .....	121
Tabel 7.11 Implementasi asal berita acara reschedule .....	121
Tabel 7.12 implementas tabel bidang dosen .....	121
Tabel 7.13 Implementasi tabel bidang penelitian .....	122



Tabel 7.14 Implementasi tabel bimbingan dosen .....	122
Tabel 7.15 Implementasi tabel dosen.....	123
Tabel 7.16 Implementasi tabel indikator penilaian .....	123
Tabel 7.17 Implementasi tabel info ujian skripsi.....	123
Tabel 7.18 Implementasi tabel jadwal dosen .....	124
Tabel 7.19 Implementasi tabel jurusan .....	124
Tabel 7.20 Implementasi tabel keminatan .....	124
Tabel 7.21 Implementasi tabel komponen penilaian .....	125
Tabel 7.22 Implementasi tabel mahasiswa.....	125
Tabel 7.23 Implementasi tabel matkul .....	125
Tabel 7.24 Implementasi tabel matkul pendukung skripsi .....	125
Tabel 7.25 Implementasi tabel nilai semhas.....	126
Tabel 7.26 Implementasi tabel nilai skripsi .....	126
Tabel 7.27 Implementasi tabel pengajuan penguji pengganti.....	127
Tabel 7.28 Implementasi tabel prodi.....	127
Tabel 7.29 Implementasi tabel skripsi .....	127
Tabel 7.30 Implementasi tabel status ujian .....	128
Tabel 7.31 Implementasi tabel tipe penelitian .....	128
Tabel 8.1 Kasus uji login <i>basic flow</i> .....	133
Tabel 8.2 Kasus uji login <i>alternative flow 1</i> .....	134
Tabel 8.3 Kasus uji login <i>alternative flow 2</i> .....	134
Tabel 8.4 Kasus uji melihat notifikasi.....	135
Tabel 8.5 Kasus uji melihat daftar jadwal ujian skripsi <i>basic flow</i> .....	135
Tabel 8.6 Kasus uji melihat daftar jadwal ujian skripsi <i>alternative flow 1</i> .....	136
Tabel 8.7 Kasus uji melihat daftar jadwal ujian skripsi <i>alternative flow 2</i> .....	136
Tabel 8.8 kasus uji melihat rincian ujian skripsi .....	137
Tabel 8.9 Kasus uji melihat melihat dokumen skripsi.....	137
Tabel 8.10 Kasus uji memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji <i>basic flow</i> .....	138
Tabel 8.11 Kasus uji memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji <i>alternative flow</i> .....	139
Tabel 8.12 Kasus uji memberikan penilaian <i>basic flow</i> .....	139
Tabel 8.13 Kasus uji memberikan penilaian <i>basic flow</i> (Lanjutan).....	140



Tabel 8.14 Kasus uji memberikan penilaian <i>alternative flow</i> .....	140
Tabel 8.15 Kasus uji memperbaiki penilaian .....	140
Tabel 8.16 Kasus uji memperbaiki penilaian (Lanjutan) .....	141
Tabel 8.17 Kasus uji melihat rekap nilai .....	141
Tabel 8.18 Kasus uji melihat <i>gap</i> nilai .....	142
Tabel 8.19 Kasus uji melihat rincian penilaian dosen pembimbing dan dosen penguji .....	142
Tabel 8.20 Kasus uji melihat rincian penilaian dosen pembimbing dan dosen penguji (Lanjutan) .....	143
Tabel 8.21 Kasus uji membuat berita acara ujian telah selesai <i>basic flow</i> .....	143
Tabel 8.22 Kasus uji membuat berita acara ujian telah selesai <i>alternative flow</i> .....	143
Tabel 8.23 Kasus uji membuat berita acara ujian telah selesai <i>alternative flow</i> (Lanjutan) .....	144
Tabel 8.24 Kasus uji membuat berita acara ujian penjadwalan ulang .....	144
Tabel 8.25 Kasus uji membuat berita acara ujian telah selesai <i>alternative flow</i> .....	144
Tabel 8.26 Kasus uji membuat berita acara ujian telah selesai <i>alternative flow</i> (Lanjutan) .....	145
Tabel 8.27 Kasus uji melihat daftar riwayat ujian skripsi .....	145
Tabel 8.28 Kasus uji melihat rincian riwayat ujian skripsi .....	146
Tabel 8.29 Kasus uji menjalankan waktu pengatur waktu ujian .....	146
Tabel 8.30 Kasus uji menjalankan waktu pengatur waktu ujian (Lanjutan) .....	147
Tabel 8.31 Hasil pengujian keamanan verifikasi kehadiran tanpa menggunakan token .....	148
Tabel 8.32 Hasil pengujian keamanan verifikasi kehadiran menggunakan token yang valid .....	148
Tabel 8.33 Hasil pengujian keamanan verifikasi kehadiran menggunakan token yang tidak valid .....	148
Tabel 8.34 Hasil pengujian keamanan memberikan penilaian tanpa menggunakan token .....	149
Tabel 8.35 Hasil pengujian keamanan memberikan penilaian menggunakan token yang valid .....	149
Tabel 8.36 Hasil pengujian keamanan memberikan penilaian menggunakan token yang tidak valid .....	149
Tabel 8.37 Hasil pengujian keamanan memberikan penilaian menggunakan token yang tidak valid (Lanjutan) .....	150



Tabel 8.38 Hasil pengujian keamanan membuat berita acara tanpa menggunakan token ..... 150

Tabel 8.39 Hasil pengujian keamanan membuat berita acara menggunakan token yang valid..... 150

Tabel 8.40 Hasil pengujian keamanan membuat berita acara menggunakan token yang valid (Lanjutan) ..... 151

Tabel 8.41 Hasil pengujian keamanan membuat berita acara menggunakan token yang tidak valid ..... 151

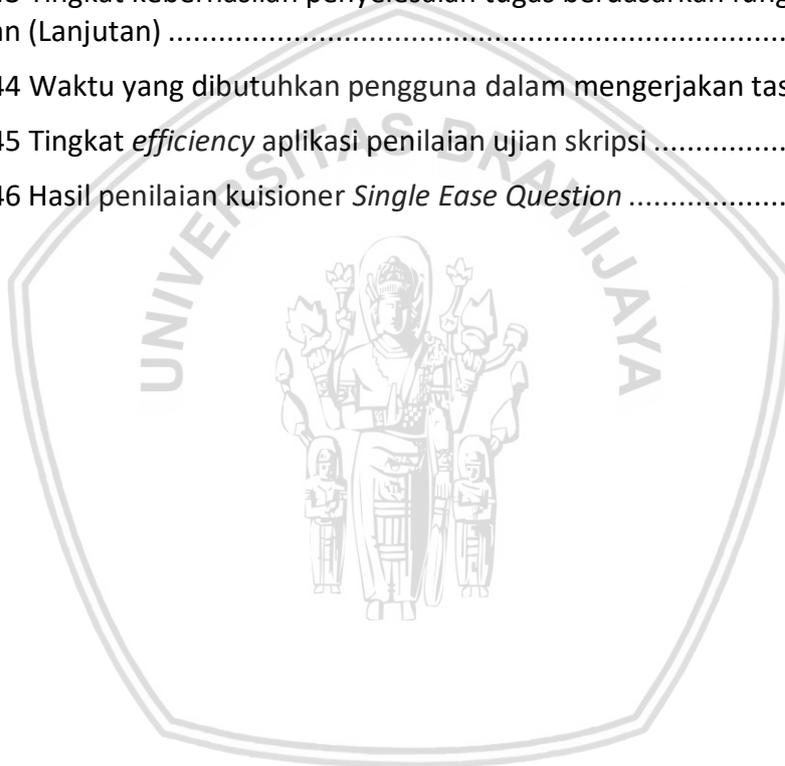
Tabel 8.42 Tingkat keberhasilan penyelesaian tugas berdasarkan fungsi yang digunakan ..... 151

Tabel 8.43 Tingkat keberhasilan penyelesaian tugas berdasarkan fungsi yang digunakan (Lanjutan) ..... 152

Tabel 8.44 Waktu yang dibutuhkan pengguna dalam mengerjakan task ..... 152

Tabel 8.45 Tingkat *efficiency* aplikasi penilaian ujian skripsi ..... 153

Tabel 8.46 Hasil penilaian kuisisioner *Single Ease Question* ..... 153



## DAFTAR GAMBAR

Gambar 1.1 Jumlah pengguna perangkat bergerak .....	3
Gambar 1.2 Jumlah pengguna aplikasi <i>mobile</i> dan <i>desktop</i> .....	3
Gambar 2.1 Struktur organisasi fakultas ilmu komputer.....	7
Gambar 2.2 Diagram alir pelaksanaan ujian skripsi bagian 1 .....	10
Gambar 2.3 Diagram alir pelaksanaan ujian skripsi bagian 2 .....	11
Gambar 2.4 Diagram alir pengecekan kehadiran majelis .....	12
Gambar 2.5 Komponen Sistem Informasi .....	15
Gambar 2.6 Android <i>activity lifecycle</i> .....	17
Gambar 2.7 Tahapan metode prototyping .....	19
Gambar 2.8 Contoh <i>private</i> bisnis proses.....	25
Gambar 2.9 Contoh <i>public</i> bisnis proses.....	25
Gambar 2.10 Contoh sub-model collaboration.....	25
Gambar 2.11 Contoh choreographies.....	26
Gambar 2.12 Contoh <i>Physical Data Model</i> .....	30
Gambar 2.13 JSON Building Block .....	31
Gambar 2.14 Contoh Kuisisioner Single Ease Question .....	34
Gambar 2.15 <i>Benchmark Framework</i> Lumen .....	35
Gambar 2.16 Contoh skala likert .....	37
Gambar 3.1 Diagram metodologi penelitian.....	38
Gambar 4.1 Proses bisnis penilaian komponen proses pengerjaan skripsi dan kualitas penulisan .....	40
Gambar 4.2 Proses bisnis pelaksanaan ujian skripsi.....	43
Gambar 4.3 Proses bisnis pengecekan kehadiran majelis .....	44
Gambar 4.4 Aturan penomoran analisis kebutuhan .....	46
Gambar 4.5 <i>Use Case Diagram</i> aplikasi penilaian ujian skripsi.....	51
Gambar 5.1 Rancangan arsitektur sistem .....	62
Gambar 5.2 <i>Activity diagram</i> memverifikasi kehadiran mahasiswa, pembimbing dan penguji .....	63
Gambar 5.3 <i>Activity diagram</i> melakukan penilaian .....	64
Gambar 5.4 <i>Activity diagram</i> melihat rekap nilai.....	65
Gambar 5.5 <i>Activity diagram</i> membuat berita acara .....	66

Gambar 5.6 Perancangan <i>Sequence Diagram</i> memverifikasi kehadiran .....	67
Gambar 5.7 Perancangan <i>Sequence Diagram</i> memberikan penilaian .....	68
Gambar 5.8 Perancangan <i>Sequence Diagram</i> rekap nilai.....	70
Gambar 5.9 Perancangan <i>Sequence Diagram</i> membuat berita acara .....	71
Gambar 5.10 Perancangan <i>Class Diagram</i> Model.....	72
Gambar 5.11 Perancangan <i>Class Diagram</i> Presenter Dan View .....	74
Gambar 5.12 Physical Data Model .....	84
Gambar 5.13 <i>Wireframe</i> halaman login .....	88
Gambar 5.14 <i>Wireframe</i> halaman beranda .....	88
Gambar 5.15 <i>Wireframe</i> halaman rincian jadwal ujian.....	89
Gambar 5.16 <i>Wireframe</i> halaman verifikasi kehadiran.....	90
Gambar 5.17 <i>Wireframe</i> halaman penilaian: (a) halaman utama penilaian (b)halaman indikator; (c) halaman penilaian naskah publikasi .....	91
Gambar 5.18 <i>Wireframe</i> halaman rekap nilai dan berita acara: (a) halaman rekap nilai dan berita acara; (b) kolom informasi <i>gap</i> nilai.....	92
Gambar 5.19 Aliran antarmuka pengguna aplikasi .....	93
Gambar 6.1 <i>Wireframe</i> halaman beranda .....	100
Gambar 6.2 <i>Wireframe</i> halaman rincian jadwal ujian.....	101
Gambar 6.3 <i>Wireframe</i> halaman verifikasi kehadiran.....	101
Gambar 6.4 <i>Wireframe</i> halaman penilaian: (a) halaman utama penilaian (b)halaman indikator; (c) halaman penilaian naskah publikasi .....	102
Gambar 6.5 <i>Wireframe</i> halaman rekap nilai.....	103
Gambar 6.6 Aliran sntarmuka penggunaan .....	105
Gambar 7.1 Implementasi antarmuka halaman beranda: (a) halaman beranda; (b)halaman beranda bagian menu <i>navigation drawer</i> .....	129
Gambar 7.2 Implementasi antarmuka halaman jadwal ujian .....	130
Gambar 7.3 Implementasi antarmuka halaman rincian ujian .....	130
Gambar 7.4 Implementasi antarmuka halaman verifikasi.....	131
Gambar 7.5 Implementasi antarmuka halaman penilaian: (a) halaman utama; (b) halaman komponen ujian; (c) halaman komponen naskah publikasi .....	132
Gambar 7.6 Implementasi antarmuka halaman rekap nilai .....	132
Gambar 8.1 Aturan penomoran pengujian validasi .....	133
Gambar 8.2 Aturan Penomoran Pengujian Keamanan.....	147

## DAFTAR LAMPIRAN

LAMPIRAN A HASIL VALIDASI MASALAH.....	160
LAMPIRAN B WAWANCARA .....	162
LAMPIRAN C RINCIAN UJI PROTOTIPE .....	163
LAMPIRAN D RINCIAN PENGUJIAN USABILITY APLIKASI.....	168



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Mobile device atau yang disebut dengan teknologi perangkat bergerak merupakan teknologi yang akhir-akhir ini mengalami perkembangan yang sangat pesat, yang dimana hal tersebut membuat jumlah pengguna dari teknologi perangkat bergerak setiap tahunnya meningkat. Hal ini ditunjukkan dari *Data Digital* pada Januari 2017 di situs We are Social sebagaimana ditunjukkan dalam Gambar 1.1 menunjukkan bahwa pengguna perangkat bergerak diseluruh dunia tidak kurang dari 4.917 miliar atau sekitar 66% populasi penduduk dunia. Sedangkan pada tahun 2016 pengguna perangkat bergerak diseluruh dunia tidak kurang dari 2.790 miliar atau sekitar 51% populasi penduduk dunia. Terjadinya peningkatan jumlah pengguna yang sangat besar dimulai sejak kehadiran perangkat bergerak cerdas (*smartphone*) pada tahun 2008 dengan ditandai munculnya iPhone dan Android yang dapat memberikan berbagai fungsionalitas dan aplikasi dari perangkat bergerak (Tolle, et al., 2017).

Salah satu yang menyebabkan adanya peningkatan jumlah pengguna perangkat bergerak tersebut adalah adanya berbagai fungsionalitas selain fungsi komunikasi yang dimiliki oleh perangkat bergerak cerdas (*smartphone*) dan ditunjang dengan aplikasi yang membantu aktivitas manusia. Kehidupan manusia saat ini dapat dikatakan telah memasuki era "mobility life" karena saat ini penggunaan perangkat bergerak tidak terlepas dari kehidupan manusia, dimulai penggunaan perangkat bergerak untuk alarm hingga sekarang digunakan untuk memesan ojek dari perangkat bergerak (Tolle, et al., 2017). Selain itu, aplikasi yang dimiliki oleh perangkat bergerak sudah menggeser penggunaan aplikasi berbasis web. Menurut Morgan Stanley Research dari situs BusinessInsider sebagaimana yang ditunjukkan oleh Gambar 1.2 menunjukkan aplikasi mobile terus mengalami peningkatan setiap tahunnya dan menggeser aplikasi desktop. Hal ini terlihat di tahun 2013 pengguna aplikasi perangkat bergerak telah menggeser penggunaan aplikasi desktop.

Aplikasi perangkat bergerak memiliki keunggulan dibandingkan dengan aplikasi desktop, diantaranya atraktif dan mudahnya kontrol pengguna pada perangkat bergerak dengan memanfaatkan layer sentuh, perangkat bergerak memiliki fitur khusus yang dapat memberikan notifikasi kepada pengguna saat ada informasi atau pembaharuan yang terjadi melalui aplikasi, dan perangkat bergerak dapat dibawa kemana-mana dan mudah digunakan dengan konten yang dapat diakses *online* maupun *offline* (Tolle, et al., 2017). Selain itu juga aplikasi perangkat bergerak dapat mengakses *functionality* yang terdapat pada sistem operasi perangkat bergerak dan hardware. Sehingga membuat aplikasi perangkat bergerak memiliki performance yang lebih maksimal dibandingkan dengan aplikasi web (Jobe, 2013).

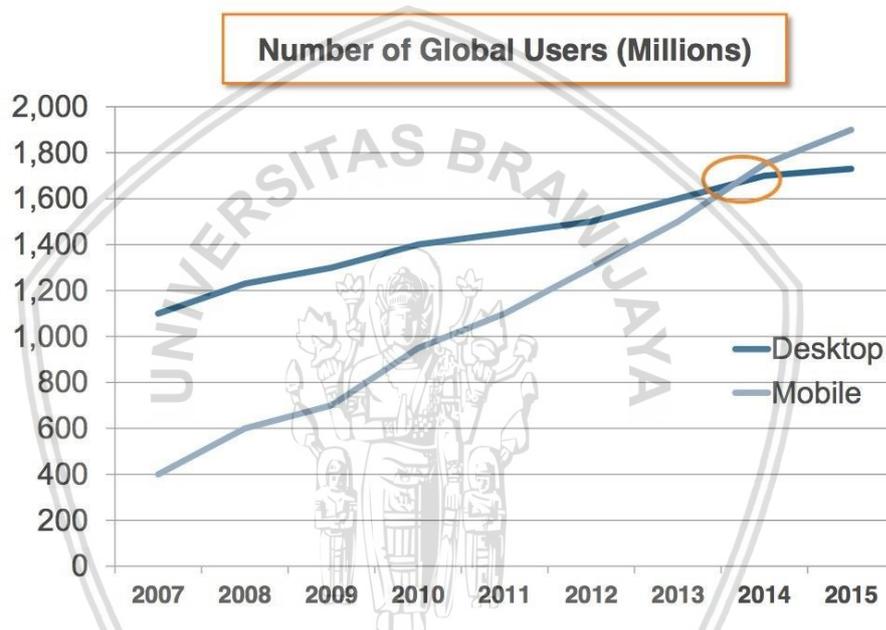
Fakultas Ilmu Komputer atau FILKOM adalah salah satu fakultas Universitas Brawijaya yang telah memiliki sistem informasi berbasis web untuk menunjang

kegiatan civitas akademik. Akan tetapi, sistem yang dikenal dengan FILKOM APPS masih belum berjalan optimal dalam memenuhi kebutuhan penggunanya, terutama ketika pengguna mengaksesnya melalui *smartphone*. Berdasarkan hasil survey yang telah dilakukan kepada 19 pengguna, 64% setuju dengan tampilan dari FILKOM APPS kurang *mobile friendly* ketika mengakses melalui perangkat bergerak mereka dan sekitar 57% setuju bahwa pengguna merasa kesulitan dalam mengakses FILKOM APPS. Selain itu juga, menurut pemaparan yang disampaikan oleh salah satu pengembang FILKOM APP (Lampiran B), sistem tersebut memang belum dikembangkan untuk *mobile* dan ia menyarankan untuk tidak mengaksesnya melalui *smarthpone*. Padahal, sekitar 71% pengguna mengakses FILKOM APPS dengan menggunakan *smarthphone*. Munculnya permasalahan yang terdapat pada FILKOM APPS berbasis *web* ketika diakses melalui *smartphone* menunjukkan ketidaksesuaian FILKOM APPS dengan kebutuhan pengguna, yang dimana kebutuhan pengguna saat ini adalah adanya dukungan FILKOM APPS untuk dapat diakses melalui *mobile*. Terbukti perolehan presentase dari hasil survey yang telah dilakukan, sekitar 86% setuju jika FILKOM APPS perlu dikembangkan di perangkat bergerak.

Berdasarkan pemaparan tersebut dapat disimpulkan bahwa kebutuhan pengguna FILKOM APPS saat ini adalah sistem yang dapat berjalan melalui perangkat bergerak. Maka dari itu, dibutuhkannya pengembangan FILKOM APPS berbasis *mobile application*. Sehingga dengan adanya FILKOM APPS berbasis *mobile* dapat memenuhi kebutuhan pengguna saat ini. Dalam mengembangkan suatu aplikasi *mobile* yang baik dan bermanfaat, ada beberapa hal yang perlu diperhatikan, seperti startup time, responsivitas aplikasi, kejelasan tujuan dan fungsi utama, komabilitas interaksi dengan sumber informasi eksternal, dan konsistensi (Tolle, et al., 2017). Selain itu, sudut pandang ketergunaan (*usability*) menjadi hal yang perlu diperhatikan dalam mengembangkan aplikasi perangkat bergerak. Oleh karena itu, diperlukan sebuah perancangan yang matang untuk menghasilkan sebuah aplikasi yang baik. Pemilihan siklus tahapan pengembangan yang tepat menjadi pendukung dalam menghasilkan sebuah rancangan dan implementasi suatu aplikasi. Ada banyak siklus pengembangan perangkat lunak yang dapat diterapkan, seperti Waterfall, Prototyping, Spiral maupun konsep lainnya. Akan tetapi konsep prototyping menjadi salah satu konsep yang dikedepankan dalam mengembangkan aplikasi perangkat bergerak. Hal ini dikarenakan konsep tersebut memungkinkan pengembang dapat memberikan gambaran atau demonstrasi dari sistem yang akan dibangun, yang nantinya akan memperjelas spesifikasi kebutuhan yang diinginkan oleh pengguna, serta dapat memperoleh umpan balik dari pengguna yang dapat dijadikan bahan evaluasi (S & Shalahuddin, 2014). Oleh karena itu, metode *prototyping* menjadi alasan peneliti memilih metode tersebut dalam proses pengembangannya. Selain itu juga, peneliti memilih salah satu jenis metode *prototyping*, yaitu *evolutionary prototyping* yang dimana dari jenis ini, peneliti melakukan pengembangan terhadap *prototyping* yang dibuat dengan tidak membuang *prototyping* pada iterasi sebelumnya sehingga prototipe yang sudah dibuat terus dilakukan pengembangan hingga menjadi sistem yang sebenarnya.



Gambar 1.1 Jumlah pengguna perangkat bergerak  
Sumber: WeAreSocial



Gambar 1.2 Jumlah pengguna aplikasi *mobile* dan *desktop*  
Sumber : BussinessInsider

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dipaparkan diatas, maka menghasilkan rumusan masalah sebagai berikut:

1. Bagaimana rancangan dan implementasi aplikasi perangkat bergerak berbasis android?
2. Bagaimana tingkat kemudahan aplikasi penilaian ujian skripsi berbasis android dalam menjalankan proses penilaian ujian skripsi?

## 1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam melakukan penelitian ini antara lain:

1. Untuk mengetahui tingkat kemudahan pengguna terhadap aplikasi penilai ujian skripsi berbasis Android.



2. Untuk mengetahui bagaimana rancangan aplikasi penilai ujian skripsi berbasis android.
3. Untuk mengetahui bagaimana implementasi aplikasi penilaian ujian skripsi berbasis android.

#### 1.4 Manfaat

Manfaat yang di dapatkan dari penelitian ini adalah memberikan kemudahan kepada pengguna dalam melakukan penilai ujian skripsi mahasiswa dan juga hasil dari penelitian ini dapat menjadi pertimbangan bagi Fakultas Ilmu Komputer Universitas Brawijaya dalam menerapkan FILKOM APPS berbasis Android.

#### 1.5 Batasan masalah

Agar diperoleh hasil pembahasan sesuai dengan yang diharapkan dan tidak melebar maka penelitian ini dibatasi oleh hal-hal sebagai berikut:

1. Aplikasi ini hanya dapat digunakan di Sistem Operasi Android.
2. Pembuatan aplikasi ini menggunakan *Object Oriented Programming* .
3. Pembuatan aplikasi menggunakan pendekatan *Native* dan dalam proses pengembangannya menggunakan metode *prototyping*.
4. Peneliti hanya mengembangkan salah satu fitur FILKOM APPS yaitu penilaian ujian skripsi.

#### 1.6 Sistematika pembahasan

##### BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

##### BAB II LANDASAN KEPUSTAKAAN

Bab ini membahas tentang dasar teori dan pendukung, temuan, dan atau bahan penelitian sebelumnya yang digunakan dalam penelitian.

##### BAB III METODOLOGI

Bab ini membahas tentang metode yang digunakan dalam penelitian, yang dimana terdiri dari langkah-langkah yang digunakan dalam menyelesaikan penelitian dan juga terdapat metode *software development life cycle* untuk menunjang metodologi.

##### BAB IV ANALISIS

Bab ini menjelaskan tentang analisis aplikasi yang akan dibangun. Bab ini terdiri dari analisis proses bisnis dan analisis kebutuhan.

**BAB V PERANCANGAN**

Bab ini menjelaskan tentang rancangan dari aplikasi yang akan dibuat. Bab ini terdiri dari perancangan arsitektur sistem, perancangan *activity diagram*, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, perancangan API, dan perancangan algoritme.

**BAB VI EVALUASI PROTOTIPE**

Bab ini menjelaskan tentang pembahasan evaluasi prototipe yang sudah dilakukan. Bab ini terdiri atas pembahasan iterasi satu dan iterasi dua.

**BAB VII IMPLEMENTASI**

Bab ini menjelaskan tentang pembahasan implementasi aplikasi yang dibuat menjadi aplikasi rilis. Bab ini terdiri dari spesifikasi sistem, batasan implementasi, implementasi aplikasi, implementasi *web service*, implementasi basis data, dan implementasi antarmuka pengguna.

**BAB VIII PENGUJIAN DAN ANALISIS**

Bab ini menjelaskan tentang pengujian terhadap aplikasi yang telah dibuat dan menjelaskan hasil analisis berdasarkan hasil pengujian yang telah dilakukan. Bab ini terdiri dari pengujian fungsional, pengujian non-fungsional dan analisis hasil pengujian.

**BAB IX PENUTUP**

Bab ini berisi kesimpulan dan saran. Kesimpulan dan saran didasarkan atas pencapaian dari penelitian yang telah dilakukan serta saran yang nantinya digunakan untuk penelitian selanjutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

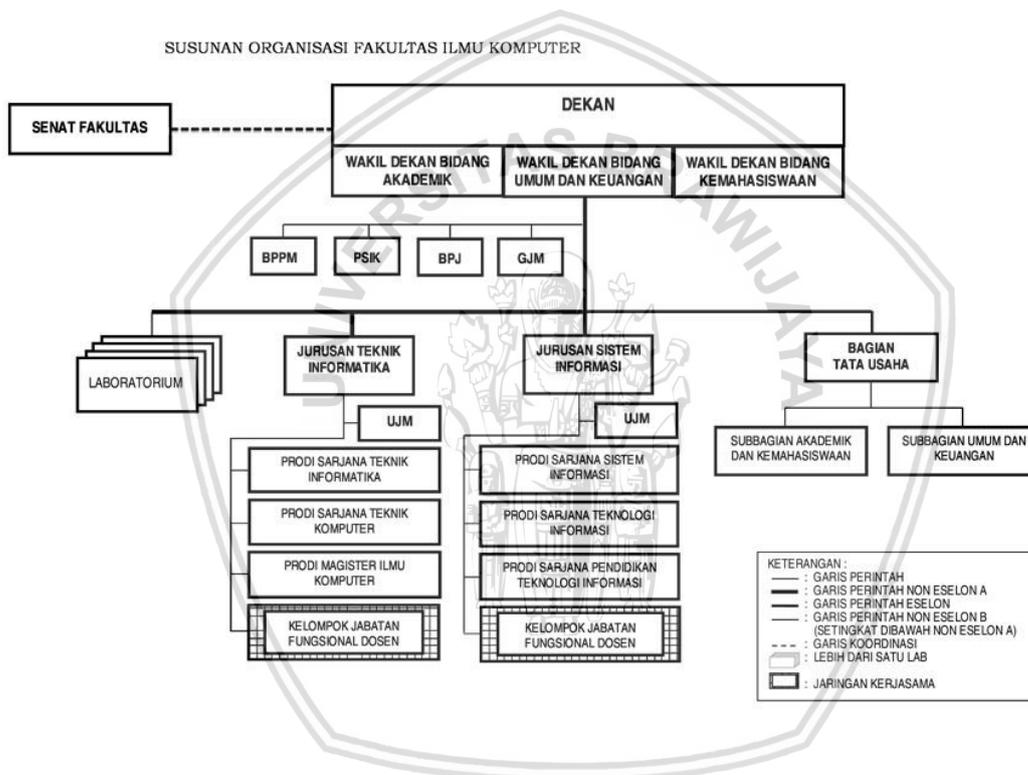
### 2.1 Kajian Pustaka

Penelitian ini membahas tentang penerapan dari metode prototyping dalam membangun aplikasi berbasis android, dengan judul dari penelitian ini adalah “Pembangunan Aplikasi Penilaian Skripsi Berbasis Android Dengan Menggunakan Metode *Prototyping*”. Sebelumnya pernah dilakukan penelitian dengan menerapkan metode prototyping dalam proses pengembangan perangkat lunak yang dilakukan oleh Faisal Prampudinantaka (2015), yang berjudul “Pengembangan Aplikasi Web Terintegrasi Pada Perpustakaan Universitas Brawijaya Dengan Model Prototyping”. Penelitian ini membahas tentang bagaimana membangun aplikasi web yang dapat berintegrasi pada perpustakaan Universitas Brawijaya dengan menerapkan metode *prototyping* dalam pengembangannya. Penelitian ini menggunakan salah satu model *prototyping* yaitu *evolutionary prototyping* dalam mengembangkan sistem yang dibangun. Tujuan dari penelitian ini adalah untuk membangun aplikasi web yang dapat menampung dan mengelola permintaan bahan pustaka baru yang terdapat pada Perpustakaan Universitas Brawijaya dan di integrasikan dengan data bahan yang tersebar. Hal ini dikarenakan sistem informasi belum mampu mengintegrasikan data-data bahan pustaka maupun karya ilmiah yang ada disetiap ruang baca dengan perpustakaan pusat. Hasil dari penelitian ini adalah penggunaan pendekatan *evolutionary prototyping* dalam membangun aplikasi web perpustakaan Universitas Brawijaya dapat menyesuaikan dengan kebutuhan yang diharapkan pengguna.

Penelitian kedua juga berkaitan dengan metode *prototyping*, yaitu “Rancang Bangun Sistem Estimasi Proyek Pada Konveksi Pakaian Dengan Metode Pert-CPM Dan *Prototyping Model*” yang dilakukan oleh Vina Angelia (2015). Penelitian ini bertujuan untuk membangun sebuah sistem estimasi proyek dengan memastikan kebutuhan pengguna benar-benar terpenuhi. Sehingga untuk mencapai hal tersebut, peneliti menggunakan *model prototyping* dalam proses pengembangannya. Selain itu, peneliti juga melakukan pengujian *usability* untuk memperoleh hasil yang akurat terkait ketertudahan pengguna terhadap sistem yang dibangun. Kemudian, penelitian ketiga berkaitan dengan pembangunan sistem informasi ber-*platform* android, yaitu “Sistem Informasi Akademik Universitas Islam Negeri Sunan Kalijaga Yogyakarta Berbasis Android” yang dilakukan oleh Suryana Wijaya (2013). Penelitian ini bertujuan untuk membangun sebuah sistem informasi akademik berbasis android. Latar belakang peneliti membangun dikarenakan tidak adanya pengembangan sistem informasi akademik ber-*platform* android dan juga peneliti melihat kebutuhan aplikasi *mobile* yang begitu tinggi. Penelitian menggunakan metode *waterfall* dalam mengembangkan sistem. Hasil diperoleh dari penelitian ini adalah peneliti berhasil membuat sebuah sistem informasi yang berjalan di perangkat bergerak berbasis android.

## 2.2 Profil Organisasi

Fakultas Ilmu Komputer atau yang disingkat FILKOM merupakan salah satu Fakultas Universitas Brawijaya Malang yang memiliki tujuan menghasilkan lulusan di bidang Teknologi Informasi. FILKOM telah berdiri sejak tahun 2011 dan telah memiliki dua jurusan, yaitu Sistem Informasi dan Teknik Informatika. Jurusan Sistem Informasi memiliki tiga macam program studi, yaitu Program Studi Sistem Informasi, Program Studi Pendidikan Teknologi Informasi dan Program Teknologi Informasi. Sementara jurusan Teknik Informatika terdiri atas Program Studi Magister Ilmu Komputer, Program Studi Teknik Informatika dan Program Studi Teknik Komputer. Adapun struktur organisasi yang dimiliki oleh Fakultas Ilmu Komputer sebagai berikut:



Sumber: FILKOM (2018)

## 2.3 Mekanisme Dan Prosedur Pelaksanaan Ujian Skripsi

Pelaksanaan ujian skripsi pada Fakultas Ilmu Komputer terdapat mekanisme dan prosedur yang harus dijalankan ketika pelaksanaannya. Prosedur ini menjadi panduan dalam melaksanakan ujian skripsi. Berikut mekanisme dan prosedur pelaksanaan ujian skripsi:

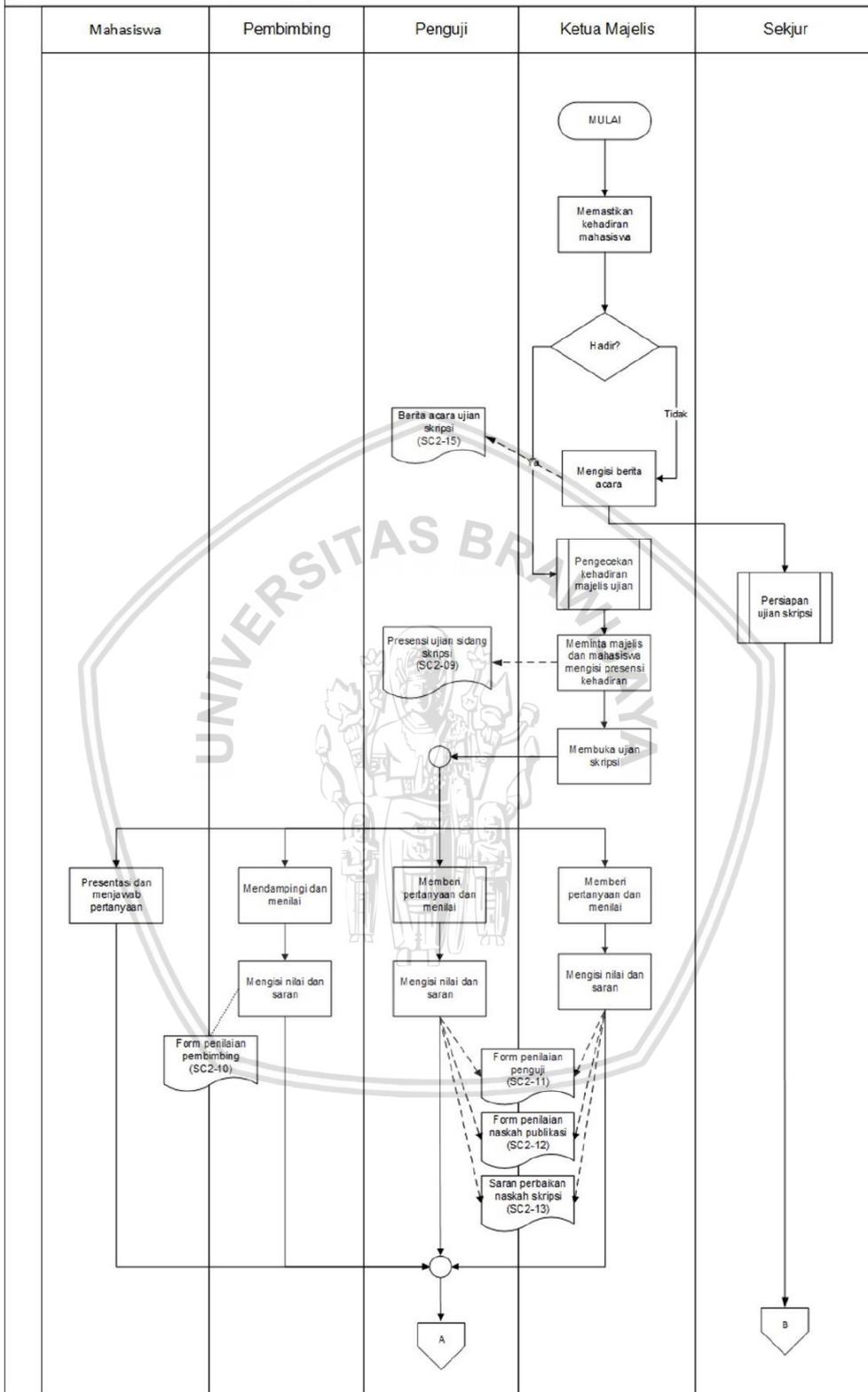
1. Ujian dapat dilaksanakan apabila dihadiri oleh mahasiswa, 2 (dua) orang Penguji dan sekurang-kurangnya satu Pembimbing dibuktikan dengan



- presensi (SC2-09). Ketua Majelis Ujian Skripsi diangkat dari salah satu Penguji berdasarkan surat tugas dari Kajur
- a. Kepastian kehadiran Pembimbing dan/atau Penguji harus sudah diselesaikan oleh Sekjur paling lambat 10 menit dari jadwal yang telah ditentukan.
  - b. Jika mahasiswa tidak hadir, maka ujian skripsi dijadwalkan ulang. Ketidakhadiran tanpa alasan yang kuat berdasarkan sepengetahuan Pembimbing, nilai maksimal ujian berikutnya adalah satu tingkat dibawah nilai maksimal normal. Alasan yang dianggap kuat adalah kondisi2 darurat pada hari ujian sehingga menghalangi mahasiswa untuk bisa hadir sesuai jadwal, yang meliputi: (i) sakit mendadak, (ii) keluarga dekat meninggal, atau (iii) kecelakaan dalam perjalanan menuju tempat ujian.
  - c. Penguji yang tidak dapat hadir pada ujian skripsi, harus memberitahukan kepada Sekjur paling lambat 1 hari sebelum pelaksanaan ujian berlangsung untuk segera dicarikan calon pengganti Penguji.
  - d. Jika Penguji yang sudah ditetapkan berhalangan hadir yang mengakibatkan hanya ada satu Penguji, maka Ketua Majelis memberitahukan kepada Sekjur agar dapat dicarikan satu Penguji pengganti. Jika Penguji tidak didapatkan maka dilakukan penjadwalan ulang ujian skripsi
  - e. Pembimbing yang tidak dapat hadir pada ujian skripsi, harus memberitahukan kepada Sekjur paling lambat 1 hari sebelum pelaksanaan ujian skripsi dan menyerahkan penilaian proses pelaksanaan skripsi.
  - f. Jika Ketua Majelis tidak hadir, maka Penguji yang hadir akan bertindak sebagai Ketua Majelis.
  - g. Mahasiswa berpakaian atasan putih dan bawahan hitam serta menggunakan jaket almamater. Untuk mahasiswa putra diwajibkan memakai dasi berwarna hitam.
  - h. Kejadian pada point b sampai dengan e harus dicatat dalam Berita Acara Ujian Skripsi (SC2-15).
2. Mahasiswa, Pembimbing dan tim penguji, melaksanakan ujian skripsi paling lama 55 menit dengan tahapan:
- a. Pembukaan oleh ketua majelis selama maksimum 1 menit.
  - b. Presentasi ujian oleh mahasiswa termasuk demo artefak selama maksimum 10 menit
  - c. Tanya Jawab oleh seluruh penguji selama maksimum 15 menit per penguji
  - d. Sidang hasil ujian skripsi oleh majelis ujian skripsi dipimpin oleh ketua majelis selama maksimum 10 menit
3. Pembimbing dan Penguji mengisi formulir penilaian :
- a. Pembimbing memberikan penilaian mengenai proses pengerjaan skripsi, kualitas penulisan dokumen skripsi, kualitas presentasi,

- kualitas argumentasi jawaban, dan kualitas artefak dengan menggunakan formulir Penilaian Pembimbing (SC2-10).
- b. Penguji memberikan penilaian mengenai kualitas penulisan dokumen skripsi, kualitas presentasi, kualitas argumentasi jawaban, dan kualitas artefak dengan menggunakan formulir Penilaian Penguji (SC2-11).
  - c. Penguji memberikan penilaian mengenai kualitas naskah publikasi (SC2-12).
  - d. Pembimbing yang tidak hadir hanya berhak memberikan penilaian terhadap proses pengerjaan skripsi dan kualitas penulisan dokumen skripsi pada formulir Penilaian Pembimbing (SC2-10). Formulir ini harus diserahkan ke Akademik Jurusan sebelum pelaksanaan ujian skripsi. Jika nilai tersebut tidak diberikan sebelum ujian dimulai, maka nilai *default* 5 (dari nilai maksimum 7) akan diberikan.
4. Pembimbing dan Penguji yang hadir melakukan rekapitulasi nilai skripsi yang meliputi nilai akhir skripsi dan nilai peringkat awal buku skripsi (SC2-14). Nilai akhir skripsi diperoleh dari seluruh komponen nilai skripsi sedangkan nilai peringkat awal skripsi diambil dari point kualitas penulisan dokumen skripsi.
  5. Ketua Majelis mengisi formulir Berita Acara Ujian Skripsi (SC2-15).
  6. Nilai akhir ujian skripsi diputuskan dalam sidang hasil ujian skripsi.
    - a. Apabila Ketua Majelis belum dapat memutuskan hasil ujian skripsi dikarenakan salah satu Pembimbing belum mengisi nilai, maka diberikan waktu maksimal 7x24 jam untuk menyelesaikannya. Jika sampai dengan batas waktu yang telah ditentukan belum dapat diputuskan, maka nilai dari Pembimbing yang tidak hadir akan disamakan dengan Pembimbing yang hadir.
    - b. Apabila Ketua Majelis tidak dapat memutuskan hasil ujian skripsi dikarenakan perbedaan selisih nilai yang tidak bisa diselesaikan oleh Majelis, maka permasalahan tersebut bisa dinaikkan ke tingkat jurusan untuk mendapatkan solusi oleh Ketua Jurusan dengan mempertimbangkan berbagai masukan dari Majelis Ujian Skripsi. Waktu maksimal 7x24 jam bagi Ketua Majelis untuk menyelesaikannya.
  7. Status hasil akhir ujian dapat dinyatakan dengan :
    - a. **Lulus**  
Lulus tanpa perbaikan skripsi
    - b. **Lulus dengan perbaikan.**  
Perbaikan skripsi harus selesai dalam waktu 30 (tiga puluh) hari setelah ujian skripsi
    - c. **Tidak Lulus**  
Mengulangi ujian skripsi dapat dilakukan maksimal dilakukan 2 kali, paling cepat 14 (empat belas) hari setelah ujian ini.
    - d. **Tidak Lulus (kembali ke preproposal)**

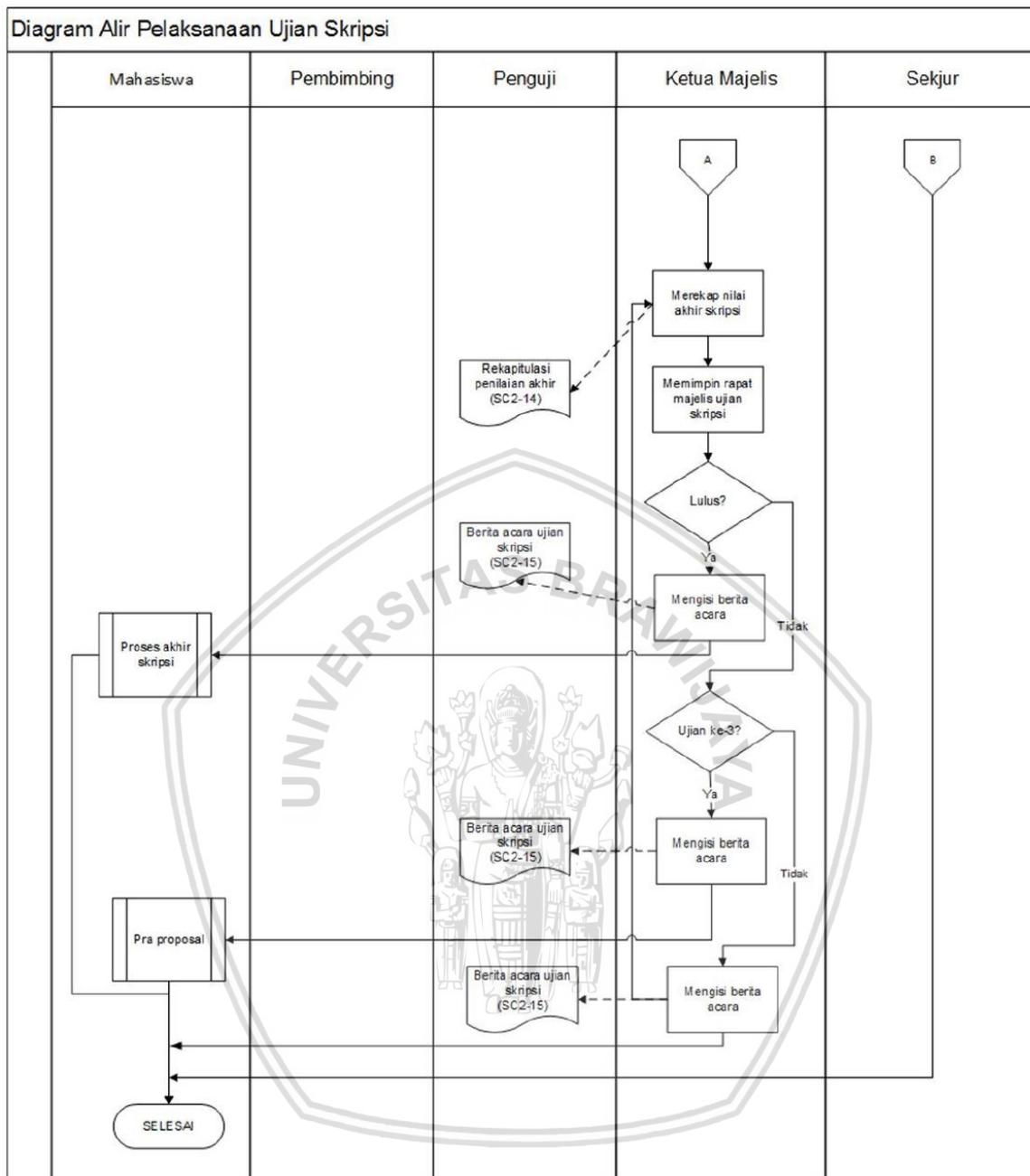
Diagram Alir Pelaksanaan Ujian Skripsi



Gambar 2.2 Diagram alir pelaksanaan ujian skripsi bagian 1

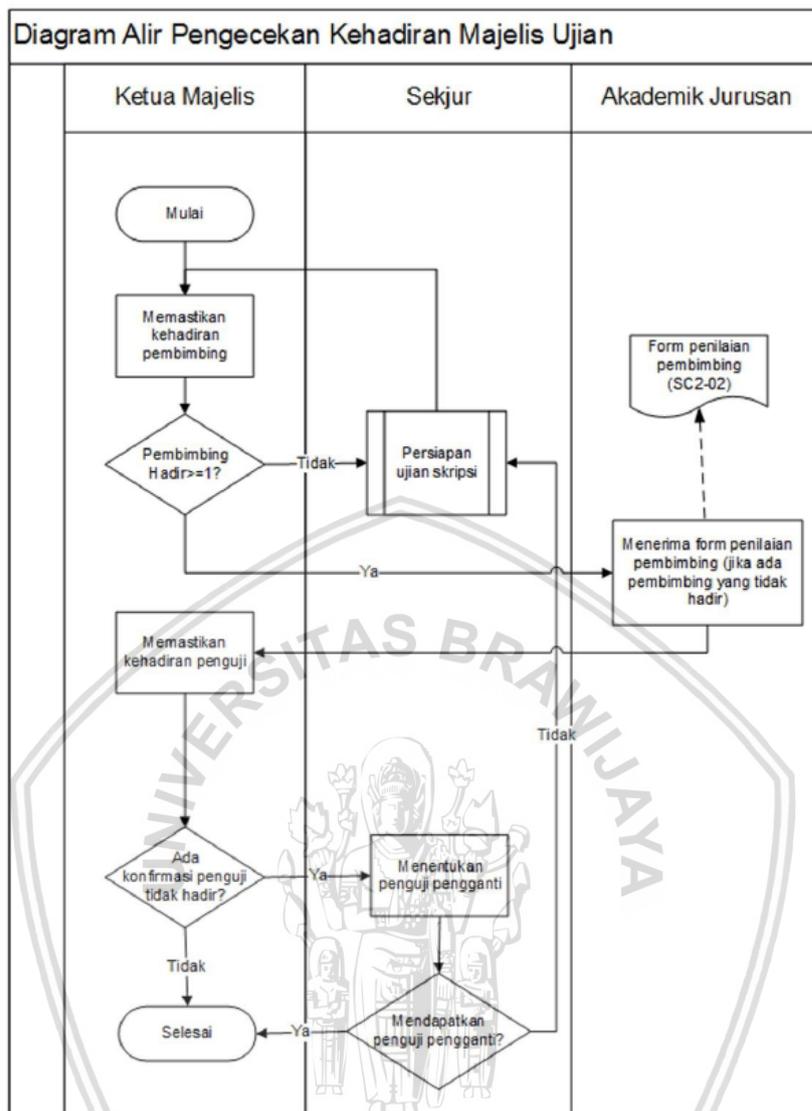
Sumber: Panduan Skripsi FILKOM UB (2017)





**Gambar 2.3 Diagram alir pelaksanaan ujian skripsi bagian 2**

Sumber: Panduan Skripsi FILKOM UB (2017)



**Gambar 2.4 Diagram alir pengecekan kehadiran majelis**

Sumber: Panduan Skripsi FILKOM UB (2017)

### 2.3.1 Standard Penilaian Kelulusan Skripsi

1. Nilai skripsi adalah nilai yang diperoleh dari pembimbing dan penguji
2. Penilaian skripsi dan ujian skripsi dilakukan oleh pembimbing dan penguji. Jumlah penguji adalah 2 orang.
3. Tahapan penilaian mencakup 3 tahap penilaian dan masing-masing memiliki bobot nilai yang berbeda. 3 tahapan penilaiannya yaitu
  - 1) Nilai seminar hasil, tahap ini memiliki bobot 5%
  - 2) Nilai pada ujian skripsi, tahap ini memiliki bobot 85%
  - 3) Nilai naskah publikasi, tahap ini memiliki bobot 10%
4. Nilai seminar hasil menggunakan rentan nilai 1 – 7

5. Nilai naskah publikasi menggunakan rentang nilai 0 – 100
6. Komponen nilai ujian skripsi mencakup 5 kriteria, yaitu:
  - a. Kriteria 1 ( $X_1$ ) : Proses pengerjaan skripsi
  - b. Kriteria 2 ( $X_2$ ) : Kualitas penulisan dokumen akhir skripsi
  - c. Kriteria 3 ( $X_3$ ) : Kualitas presentasi
  - d. Kriteria 4 ( $X_4$ ) : Kualitas argumentasi jawaban
  - e. Kriteria 5 ( $X_5$ ) : Kualitas artefak
7. Nilai ujian skripsi menggunakan rentang nilai 1 – 7 dari setiap komponen penilaian
8. Penilaian pembimbing mencakup kriteria 1, 2, 3, 4, dan 5. Bobot kriteria 3 bernilai 0.5x, bobot kriteria 5 bernilai 1.5x, dan bobot kriteria 1, 2, 4 bernilai 1x.

- a. Nilai ujian skripsi ( $Pb$ ) untuk pembimbing yang hadir menggunakan persamaan sebagai berikut:

$$Pb = \frac{x_1 + x_2 + 0.5x_3 + x_4 + 1.5x_5}{5} \quad (2.1)$$

- b. Nilai ujian skripsi ( $Pb$ ) untuk pembimbing yang tidak hadir menggunakan persamaan sebagai berikut:

$$Pb = \frac{x_1 + x_2}{2} \quad (2.2)$$

9. Penilaian penguji mencakup kriteria 2, 3, 4, dan 5. Bobot kriteria 3 bernilai 0.5x, bobot kriteria 5 bernilai 1.5x, dan bobot kriteria 2 dan 4 bernilai 1x.

$$Pb = \frac{x_2 + 0.5x_3 + x_4 + 1.5x_5}{4} \quad (2.3)$$

10. Nilai rata-rata dari pembimbing ( $Npb$ ) adalah nilai total dibagi jumlah pembimbing ( $n$ ) diperoleh dengan persamaan:

$$Npb = \frac{\sum_{i=1}^n Pbi}{n} \quad (2.4)$$

11. Nilai rata-rata dari penguji ( $Npu$ ) adalah nilai total dibagi jumlah penguji ( $n$ ) diperoleh dengan persamaan:

$$Npu = \frac{\sum_{i=1}^n Pui}{n} \quad (2.5)$$

12. Nilai ujian skripsi mahasiswa ( $Nu$ ) adalah nilai rata-rata dari nilai pembimbing dan nilai rata-rata dari penguji. Nilai rata-rata pembimbing memiliki bobot nilai sebesar 55% dan bobot nilai rata-rata penguji sebesar 30%.

$$Nu = \frac{(0.55 \times Npb + 0.3 \times Npu)}{7 \times 100} \quad (2.6)$$

13. Nilai akhir skripsi mahasiswa (  $NA$  ) adalah nilai akumulasi dari nilai seminar hasil (  $Nsh$  ) dengan bobot 5%, nilai ujian skripsi (  $Nu$  ) dengan bobot 85%, nilai naskah publikasi (  $Npb$  ) dengan bobot 10%, yang dihitung dengan persamaan:

$$NA = (0,05 \times Nsh + 0.85 \times Nu + 0.1 \times Npb) \quad (2.7)$$

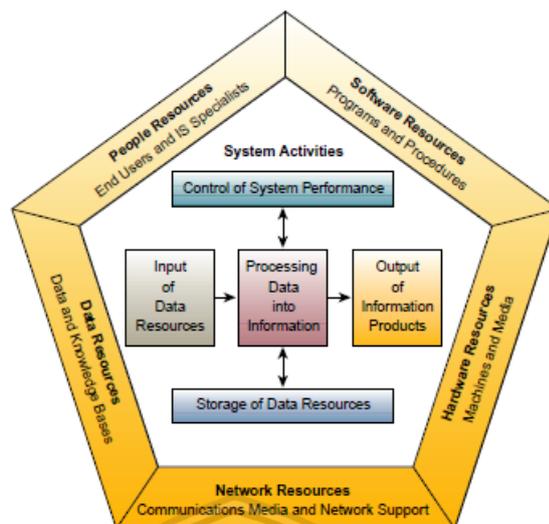
14. Mahasiswa peserta ujian skripsi dinyatakan lulus jika nilai akhir angka adalah lebih besar atau sama dengan 56. Jika nilainya kurang dari 56 maka mahasiswa tersebut dinyatakan tidak lulus dan wajib melakukan ujian ulang dalam rentang waktu 60 hari. Ujian ulang dibatasi hanya sampai 2x. Apabila setelah 2x melakukan ujian ulang nilai akhir skripsi masih kurang dari 56, status mahasiswa kembali ke preproposol.
15. Nilai akhir yang diperoleh dari ujian skripsi dinyatakan dengan nilai huruf seperti pada Tabel 2.1.

**Tabel 2.1 Nilai akhir skripsi**

Nilai Angka	Nilai Huruf	Bobot	Predikat
$100 \geq NA \geq 80$	A	4,00	Baik Sekali
$80 \geq NA \geq 75$	B+	3,50	Lebih dari baik
$75 \geq NA \geq 70$	B	3,00	Baik
$70 \geq NA \geq 60$	C+	2,50	Lebih dari cukup
$60 \geq NA \geq 56$	C	2,00	Cukup
$< 56$	D		Gagal (tidak lulus)

## 2.4 Sistem Informasi

Sistem informasi merupakan sekumpulan komponen atau elemen yang saling berkaitan dalam mengumpulkan (*input*), memanipulasi (*process*), menyimpan, dan menyebarluaskan data & informasi serta menyediakan umpan balik untuk mencapai tujuan tertentu (Stair & Reynolds, 2014). Sehingga sistem informasi terdiri beberapa komponen yang tidak dapat dipisahkan, karena di dalam suatu sistem informasi apabila salah satu unsur tidak ada, maka tidak bisa disebut sebuah sistem informasi (Darmawan & Fauzi, 2013). Komponen – Komponen sistem informasi diantaranya, perangkat keras (*hardware*), perangkat lunak (*software*), manusia (*brainware*), Prosedur (*procedure*), basis data (*database*), dan jaringan komunikasi.



**Gambar 2.5** Komponen Sistem Informasi

Sumber: O'Brien & Marakas (2011)

## 2.5 Perangkat Bergerak Dan Android

Perangkat bergerak merupakan salah satu teknologi yang sampai saat ini terus mengalami perkembangan. Perkembangan teknologi perangkat bergerak diawali dari tahun 1974 dengan kemunculan dari perangkat *pager*. Kemudian, di tahun 1983, telepon seluler pertama muncul, yaitu *Motorola™ DynaTAC 8000X*. Perkembangan ponsel tersebut berlangsung hingga tahun 1988 dan di tahun tersebut berkembang menjadi *Personal Digital Assistant* (PDA) dengan memiliki ukuran yang lebih kecil, dan ringan daripada ponsel sebelumnya. Perkembangan perangkat bergerak kemudian berlanjut pada tahun 1999 dengan munculnya ponsel-ponsel yang dirilis oleh Nokia dengan memiliki kemampuan mengakses halaman situs secara nirkabel. Lalu, ditahun 2002 – sekarang, hadirnya ponsel yang memiliki sistem operasi khusus seperti Android. Ponsel ditahun ini tidak hanya memiliki fungsi telepon biasa tetapi juga memiliki fungsi pemutar lagu, MMS, dan game. Sehingga pada tahun tersebut dikenal dengan era *smartphone* (Tolle, et al., 2017). Kemunculan *smartphone* ini kemudian dimanfaatkan oleh pengembang aplikasi dengan menghadirkan aplikasi-aplikasi perangkat bergerak yang dapat membantu aktivitas pengguna. Aplikasi perangkat bergerak ini juga memiliki keunggulan, seperti portabilitas, adanya fitur *push* notification dan memiliki kontrol yang lebih mudah dan lebih atraktif (Tolle, et al., 2017). Selain itu juga, pemanfaatan teknologi perangkat bergerak ini juga dimanfaatkan oleh perusahaan-perusahaan untuk membantu proses bisnis yang terjadi pada suatu perusahaan. Hal ini dikarenakan penggunaan aplikasi perangkat bergerak dapat meningkatkan proses bisnis terhadap empat aspek, yaitu *time*, *cost*, *quality*, dan *flexibility*.

Android merupakan perangkat lunak sistem operasi yang berbasis kernel linux yang dikembangkan untuk perangkat bergerak dan dimana mencakup sistem

operasi, *middleware* dan aplikasi (Safaat, 2012). Sistem operasi ini awalnya dikembangkan oleh Android Inc dan kemudian dikembangkan kembali oleh google. Menurut Nazarrudin, Sistem operasi Android ini memiliki beberapa keunggulan,yaitu:

1. Lengkap (*Complete Platform*)

Android merupakan sistem operasi yang menyediakan banyak tools dalam membangun software. Hal ini memberikan peluang bagi pengembang aplikasi

2. Terbuka (*Open Source Platform*)

Android memberikan Lisensi yang bersifat *open source*. Sehingga hali ini dapat dimanfaatkan oleh pengembang dalam mengembangkan aplikasi.

3. *Free Platform*

Android yang memberikan sifat lisensi yang *open source* membuat aplikasi Android tidak memerlukan biaya apa pun dalam mengembangkan dan mendistribusikan aplikasi.

**2.5.1.1 Dasar Aplikasi Android**

Aplikasi android dikembangkan dengan menggunakan bahasa pemrograman JAVA dan C++ dan baru-baru ini dalam mengembangkan aplikasi android google telah mendukung satu bahasa pemrograman lain, yaitu kotlin. Menurut Imaduddin (2017), terdapat enam jenis komponen pada aplikasi android yang perlu diketahui dalam mengembangkan aplikasi yaitu:

**1. Activity**

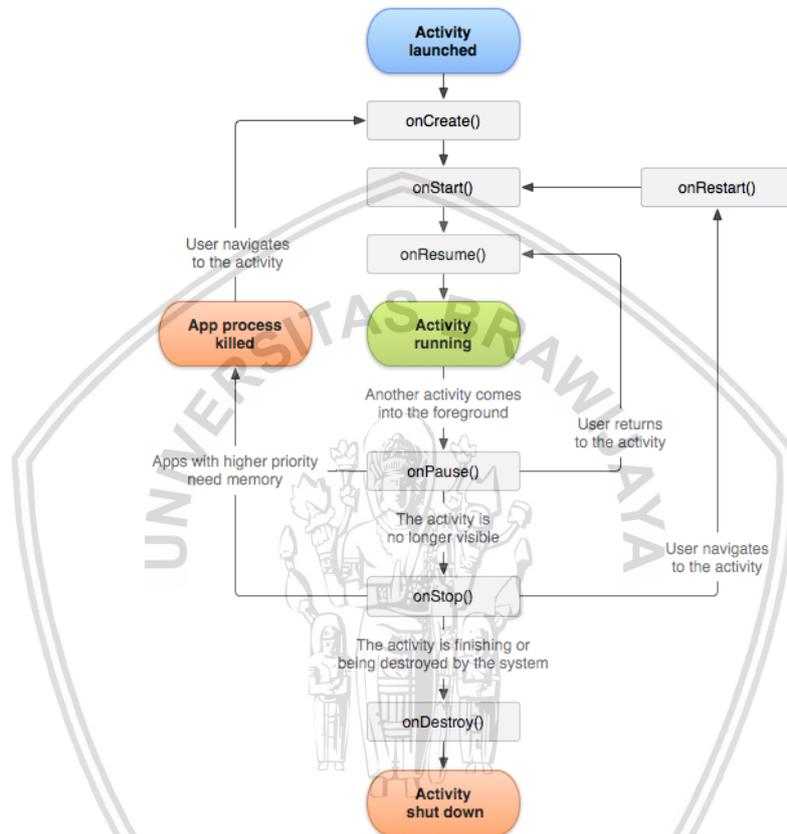
*Activities* merupakan komponen yang bertugas menangani tampilan kepada pengguna (Imaduddin & Permana, 2017). Komponen ini merupakan salah satu komponen yang berhubungan langsung dengan pengguna. Pada activity memiliki lifecycle, seperti yang ditunjukkan dalam Gambar 2.6. Pada setiap method memiliki fungsi yang berbeda-beda, berikut penjelasan dari setiap method.

**Tabel 2.2 Method lifecycle android**

No	Method	Deskripsi
1	onCreate	Dipanggil ketika activity pertama dijalankan
2	onStart	Dipanggil ketika activity ditampilkan
3	onResume	Dipanggil ketika activity berinteraksi dengan pengguna
4	onPause	Dipanggil ketika activity sedang tidak dilihat oleh pengguna
5	onStop	Dipanggil ketika activity tidak lagi dilihat oleh pengguna

Tabel 2.2 Method lifecycle android (Lanjutan)

No	Method	Deskripsi
6	onRestart	Dipanggil setelah activity anda berhenti, sebelum memulai start kembali
7	onDestroy	Dipanggil sebelum activity hilang



Gambar 2.6 Android activity lifecycle

Sumber : Android Developers

## 2. Service

Service merupakan komponen yang berjalan di *background* dan dapat digunakan untuk menjalankan suatu proses di dalam aplikasi. Misalnya, aplikasi pemutar musik, aplikasi ini memungkinkan pengguna memutar musik tanpa menjalankan aplikasi. Service dijalankan pada thread utama dari proses aplikasi.

## 3.Intent

Intent merupakan komponen yang digunakan untuk komunikasi antar komponen di dalam sebuah aplikasi. Fungsi yang dimiliki intent, antara lain, untuk menjalankan activity, mengirimkan pesan ke Broadcast Receiver, dan dapat digunakan untuk berkomunikasi dengan service yang sedang berjalan (Imaduddin & Permana, 2017).



#### 4.Fragment

Fragment merupakan komponen view selain activity, yang berfungsi untuk memberikan tampilan kepada pengguna. Fragment dapat disisipkan pada sebuah tampilan dari activity dan merupakan komponen view yang fleksibel.

#### 5.Threads

Threads merupakan komponen yang mawadahi semua proses yang ada di dalam sebuah aplikasi. Proses yang terjadi dilayar merupakan proses yang ada didalam MainThreads, akan tetapi tidak hanya MainThread saja yang digunakan dalam mawadahi jalannya proses dari sebuah aplikasi.

#### 6.Receiver

Receiver merupakan komponen yang menggunakan pola perancangan publish-subscribe. Komponen ini dapat memberitahu komponen lain ketika terjadi sebuah proses. Sehingga komponen lain dapat bereaksi terhadap event yang terjadi.

##### 2.5.1.2 ROOM Persistence Library

*Room persistence library* merupakan *library* yang dikembangkan oleh google yang bertujuan untuk memudahkan aplikasi dalam mengakses database SQL Lite android. *Library* ini memungkinkan aplikasi dapat menyalin dan meyimpan informasi yang diterima dari aplikasi, sehingga pengguna dapat melihat informasi yang diinginkan secara berulang tanpa harus aplikasi mendapatkan informasi tersebut dengan cara yang sama. Penggunaan *library* pada penelitian ini digunakan untuk menyimpan data penilaian dan pengguna. Hal ini ini dikarenakan dengan adanya penggunaan *library* tersebut aplikasi tidak perlu mengulangi permintaan ke *web service* untuk memperoleh data tersebut.

##### 2.5.1.3 Firebase Cloud Messaging

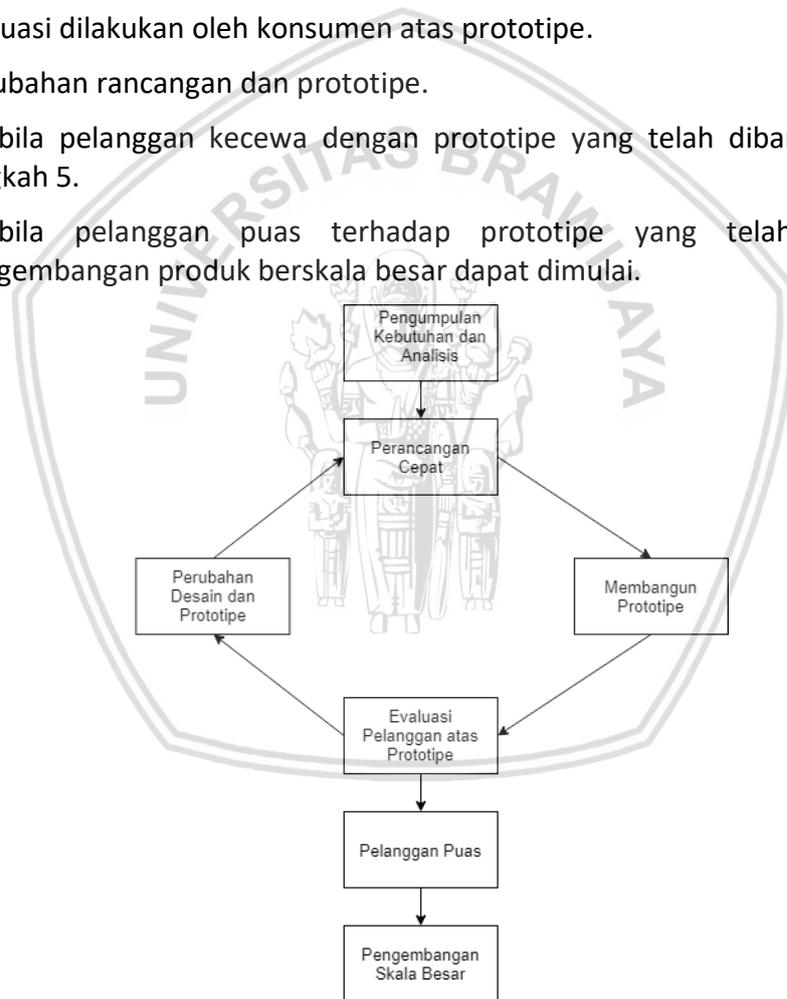
*Firebase cloud messaging* merupakan layanan dari google yang memungkinkan kita membangun aplikasi yang dapat mengirim pesan tanpa biaya (Firebase, 2018). Layanan ini dapat memberikan kemampuan aplikasi dalam menerima atau mengirim pesan kepada pengguna. Pesan yang dikirim dapat mencapai 4 kb. Pada penelitian ini, firebase cloud messaging digunakan untuk mengirim dan menerima penilaian. Sehingga dengan menggunakan layanan ini aplikasi dapat secara otomatis pengguna menerima dan mengirim penilaian.

#### 2.6 Metode Prototyping

Metode *prototyping* adalah salah satu dari *Software Development Lifecycle* (SDLC), yang dimana metode ini pengguna dan tim pengembang dapat mengklarifikas kebutuhan dan interpretasi mereka proses pengembangan sistem sehingga dapat tercipta suatu sistem yang sesuai dengan harapan dan kebutuhan pengguna (Simarmata, 2010). Sebelum tahapan konstruksi aktual dilakukan, pengembang dapat mendemonstrasikan atau menggambarkan secara langsung bagaimana sebuah perangkat lunak atau komponen-komponen perangkat lunak

akan bekerja dalam lingkungannya. Metode prototyping memiliki keuntungan yaitu, pengguna akan lebih mudah memahami sistem yang akan dikembangkan, dikarenakan pada metode ini terdapat model yang digunakan untuk menggambarkan sistem kepada pengguna, pengguna dapat terlibat aktif dalam pengembangan sehingga pengembang akan jauh lebih cepat menerima umpan balik dari pengguna dan kesalahan bisa dideteksi jauh lebih awal (ISTQB, n.d). Sebagaimana yang ditunjukkan pada Gambar 2.7, tahapan pada metode *prototyping* diantaranya:

1. Mengumpulkan dan menganalisis kebutuhan.
2. Melakukan perancangan cepat.
3. Membangun sebuah prototipe.
4. Evaluasi dilakukan oleh konsumen atas prototipe.
5. Perubahan rancangan dan prototipe.
6. Apabila pelanggan kecewa dengan prototipe yang telah dibangun, ulangi langkah 5.
7. Apabila pelanggan puas terhadap prototipe yang telah dibangun, pengembangan produk berskala besar dapat dimulai.



**Gambar 2.7 Tahapan metode prototyping**

Sumber : Simarmata (2010)

Pada metode *prototyping* memiliki 2 jenis pemodelan, yaitu *throw-away prototyping* dan *evolutionary prototyping*.

1. *Throw-away Prototyping*



*Throw-away prototyping* merupakan salah satu jenis pemodelan pada metode *prototyping* yang digunakan untuk mengembangkan sistem dan dimana pengembangan tidak memiliki pemahaman yang jelas tentang perangkat lunak yang akan dikembangkan. Maka model ini berfungsi untuk menyempurnakan dan memperjelas spesifikasi kebutuhan perangkat lunak. Prototipe yang dibangun akan terus di evaluasi sampai kebutuhan pengguna dapat terpenuhi. Setelah proses evaluasi, prototipe akan dibuang dan tidak dipakai sebagai dasar pengembangan perangkat lunak lebih lanjut (Sommerville, 2007).

## 2. *Evolutionary prototyping*

*Evolutionary prototyping* adalah model dari metode *prototyping* yang dimana sistem dikembangkan secara bertahap sehingga mudah untuk dimodifikasi dalam menanggapi *feedback* dari pengguna. *Evolutionary prototyping* tidak membuang kode *prototyping* yang telah dibuat tetapi mengembangkan menjadi kode yang dibutuhkan user. Penyempurnaan prototipe terus dilakukan untuk mendapatkan fungsionalitas yang dibutuhkan pengguna dari sistem yang akan dibuat. Prototipe ini kemudian dilanjutkan dengan produksi, yang nantinya satu prototipe evolusioner akan menjadi sistem aktual (McConnell, 1996).

## 2.7 *Unified Modeling Language*

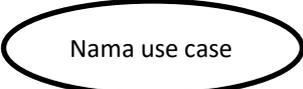
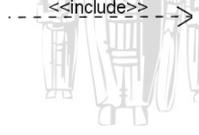
UML (*Unified Modeling Language*) merupakan sebuah bahasa grafik/gambar standar yang digunakan untuk memvisualisasi, menspesifikasi, membangun, dan pendokumentasian sebuah sistem perangkat lunak yang menggunakan pendekatan *Object Oriented Programming*. Pemodelan digunakan untuk menyederhanakan permasalahan yang kompleks sehingga lebih mudah di pahami dan dipelajari (Nugroho, 2010). Adapun tujuan penggunaan UML, antara lain:

- a. Sebagai sarana analisis, pemahaman, visualisasi, dan komunikasi antaravanggota tim pengembang.
- b. Sebagai sarana dokumentasi untuk menelaah perilaku perangkat lunak secara seksama.

### 2.7.1.1 *Use Case*

*Use case* merupakan salah satu pemodelan UML, yang digunakan untuk memodelkan kelakuan (*behavior*) dari sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (S & Shalahuddin, 2014) . Selain itu juga *use case* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang terdapat pada sistem. Pada *use case* memiliki simbol-simbol dalam memodelkan suatu sistem. Berikut beberapa simbol dan deksripsi yang biasa digunakan dalam *use case*.

Tabel 2.3 Notasi pada *use case*

Nama	Notasi	Deskripsi
<i>Use case</i>		Fungsionalitas yang disediakan sistem yang bertukar pesan antar unit atau aktor
Aktor		Aktor yang berinteraksi dengan sistem yang akan dibuat. Aktor bisa berupa orang atau sistem lain
Asosiasi		Simbol yang digunakan untuk menggambarkan komunikasi antara aktor dengan <i>use case</i>
Ekstensi/ <i>extend</i>		Simbol yang menggambarkan sebuah relasi tambahan ke suatu <i>use case</i> dan <i>use case</i> yang ditambahkan dapat berdiri sendiri
Menggunakan/ <i>include</i>		Simbol yang menggambarkan sebuah relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dan <i>use case</i> yang ditambahkan memerlukan <i>use case</i> lain

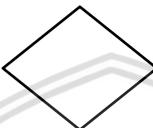
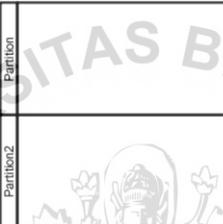
Sumber : S & Shalahuddin (2014)

### 2.7.1.2 Activity Diagram

*Activity diagram* merupakan UML diagram yang digunakan untuk menggambarkan aliran kerja dari suatu aktivitas sebuah sistem (S & Shalahuddin, 2014). Diagram ini tidak hanya menggambarkan aliran kerja dari aktivitas sebuah sistem tetapi juga dapat menggambarkan proses bisnis atau menu yang ada pada perangkat lunak. Menurut Booch (2005), *activity diagram* digunakan untuk memodelkan aspek dinamis dari suatu sistem. Adapun notasi-notasi yang perlu diperhatikan dalam membuat *activity diagram* sebagaimana yang ditunjukkan pada Tabel 2.4.



Tabel 2.4 Notasi pada activity diagram

Nama	Notasi	Deskripsi
Status awal		Status awal sistem, yang menandakan awal dari aliran kerja.
Aktivitas		Menggambarkan serangkaian kegiatan
Percabangan		Menggambarkan aliran yang memiliki dua pilihan aktivitas
Swimlane		Menggambarkan objek yang memiliki tanggung jawab terhadap aktivitas tertentu
Status Akhir		Menggambarkan akhir dari alur kerja

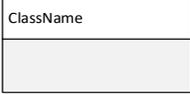
Sumber: S & Shalahuddin (2014)

### 2.7.1.3 Class Diagram

*Class diagram* merupakan salah satu diagram UML yang digunakan untuk menggambarkan jenis-jenis objek dalam sistem dan hubungan statis yang terdapat diantara mereka (Fowler, 2005). *Class diagram* umumnya ditemukan di pemodelan sistem dengan pendekatan *object oriented*. *Class diagram* terdiri dari properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. Selain itu juga pada *class diagram* memiliki notasi- notasi yang digunakan dalam memodelkan *class diagram* sebagaimana ditunjukkan pada Tabel 2.5.



**Tabel 2.5 Notasi pada class diagram**

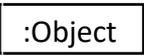
No	Nama	Notasi	Deskripsi
1	<i>Class/kelas</i>		Kelas yang terdapat pada struktur sistem.
3	<i>Asocitaion/asosiasi</i>		Relasi antara kelas dengan makna umum.
4	<i>Generalisasi</i>		Notasi yang menggambarkan relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
5	<i>Dependency/kebergantungan</i>		Notasi yang menggambarkan relasi antar kelas dengan makna kebergantungan antar kelas.
6	<i>Aggregation/agregasi</i>		Notasi yang menggambarkan relasi antara kelas dengan makna semua bagian.

Sumber : Kirill Fakhroutdinov (2009)

#### 2.7.1.4 Sequence Diagram

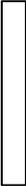
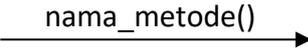
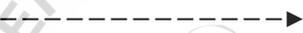
*Sequence diagram* merupakan diagram yang digunakan dalam menjelaskan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan dikirimkan dan diterima antar objek (S & Salahuddin, 2014). Oleh karena itu, sebelum *sequence diagram* ini digambarkan, terlebih dahulu mengetahui objek-objek yang terlibat dalam *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Pada Tabel 2.6 menunjukkan deskripsi dan notasi yang ada pada *sequence diagram*.

**Tabel 2.6 Notasi pada sequence diagram**

No	Nama	Notasi	Deskripsi
1	<i>Actor</i>		Aktor adalah orang, proses atau sistem lain yang berada diluar sistem dan berinteraksi dengan sistem.
2	<i>Lifeline</i>		Lifeline atau garis putus-putus menggambarkan urutan kehidupan.
3	<i>Object</i>		Menggambarkan sebuah objek.



Tabel 2.6 Notasi pada *sequence diagram* (Lanjutan)

No	Nama	Notasi	Deskripsi
4	Bar aktifasi		Bar aktifasi menggambarkan periode waktu.
5	Pesan tipe call		Pesan tipe call merupakan notasi yang menjelaskan suatu objek memanggil fungsi yang ada pada objek lain.
6	Pesan Tipe send		Notasi yang menggambarkan suatu objek mengirimkan data/masukan/informasi ke objek lainnya.
7	Pesan tipe return		Notasi yang menggambarkan suatu objek menghasilkan suatu kembalian ke objek tertentu.

Sumber : S. & Shalahuddin (2014)

## 2.7.2 Proses Bisnis

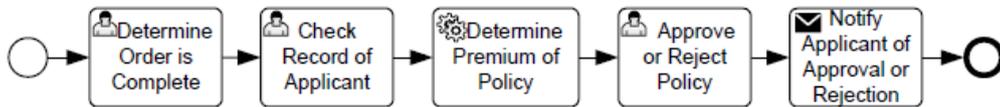
Proses bisnis merupakan serangkaian aktivitas yang saling bekerjasama dalam organisasi dan lingkungan teknis untuk mencapai tujuan bisnis (Weske, 2007). Aktivitas proses bisnis dapat dilakukan dengan bantuan sistem informasi atau dilakukan secara manual. Selain itu, menurut Michael hammer dan James Champy yang dikutip oleh Weske bahwa proses bisnis merupakan sekumpulan aktifitas yang memiliki input dan menghasilkan output yang bernilai kepada pelanggan. Sehingga berdasarkan dua pengertian tersebut dapat disimpulkan bahwa proses bisnis merupakan sekumpulan aktivitas yang saling bekerjasama untuk menghasilkan output yang bernilai kepada pelanggan dalam mencapai tujuan bisnis suatu organisasi.

### 2.7.2.1 Business Process Model Notation (BPMN)

*Business Process Model Notation* merupakan standard yang dibuat oleh Object Management Group yang bertujuan untuk menyediakan notasi yang dapat dimengerti oleh *business user*, mulai dari bisnis analis yang membuat draft awal dari proses bisnis, *technical developer* yang bertanggung jawab untuk menerapkan teknologi yang berkaitan dengan jalannya proses bisnis, hingga kepada *business people* yang akan mengelola dan memantau proses tersebut. Selain itu juga, BPMN berfungsi untuk menjembatani gap antara rancangan proses bisnis dan proses implementasi. Menurut Object Management Group BPM memiliki tiga tipe sub-model, yaitu:

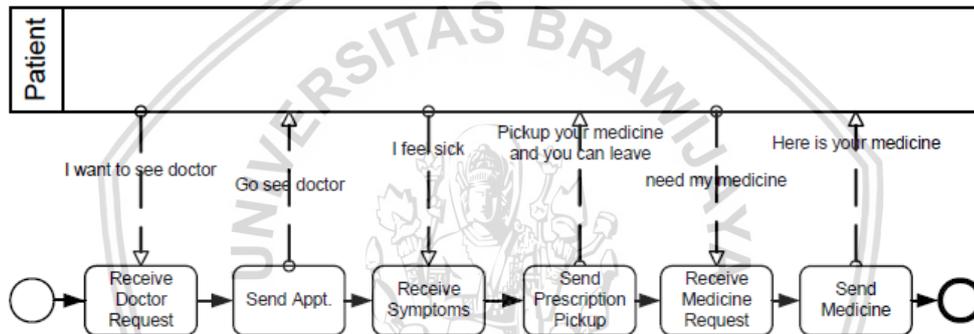
1. Orchestration

Proses merupakan tipe sub-model yang menggambarkan proses yang terjadi didalam atau diluar perusahaan. Ada dua tipe *orchestration*, yaitu bisnis proses *private* dan bisnis proses *public*. Bisnis proses *private* adalah proses yang terjadi di internal organisasi seperti departemen, divisi atau semacamnya yang dimiliki organisasi. Sedangkan, bisnis proses *public* merepresentasikan interaksi antara *private business process* dan proses lain atau participant.



Gambar 2.8 Contoh *private* bisnis proses

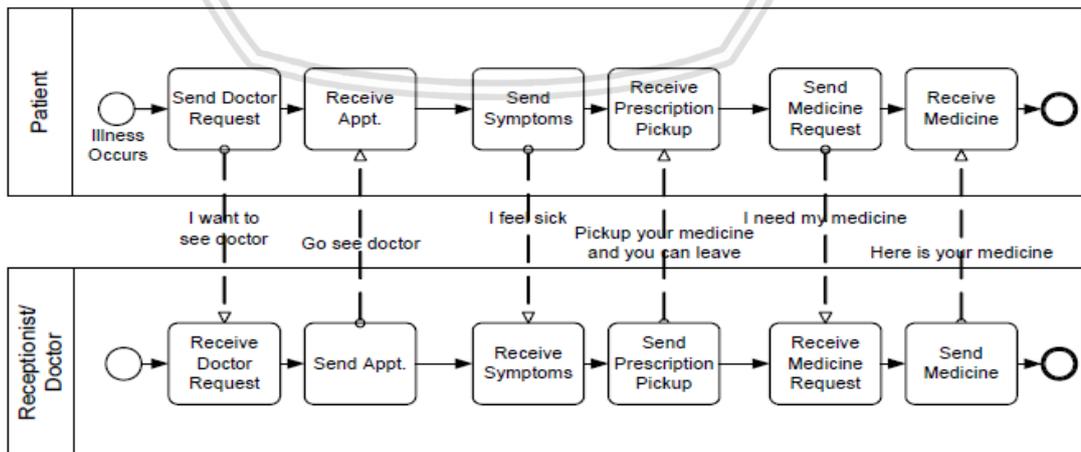
Sumber : Object Management Group (2011)



Gambar 2.9 Contoh *public* bisnis proses

Sumber : Object Management Group (2011)

2. Collaboration



Gambar 2.10 Contoh sub-model collaboration

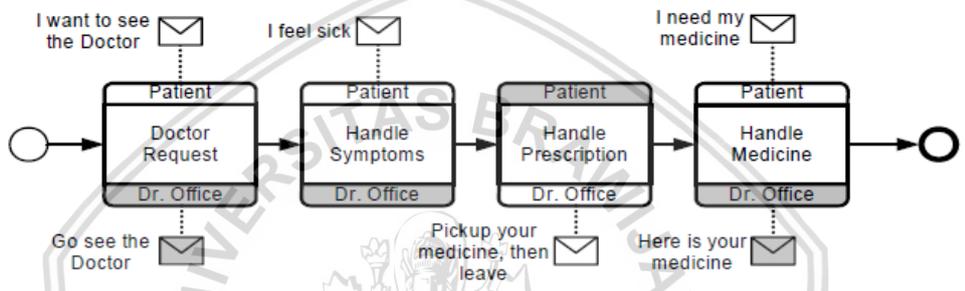
Sumber : Object Management Group (2011)



Collaboration merupakan salah satu tipe sub-model BPMN yang menggambarkan interaksi antara dua atau lebih entitas bisnis. Collaboration biasanya terdapat dua atau lebih pool yang di setiap poolnya terdapat *participant*. Collaboration juga dapat ditunjukkan sebagai dua atau lebih proses *public*. Contoh dari collaboration ditunjukkan dalam Gambar 2.10.

3. Choreographies

Choreographies merupakan kontrak prosedural, interaksi antara *participant*. Choreographies terlihat sama dengan proses bisnis *private* karena terdapat hubungan antar aktivitas, *events*, dan *gateway*. Namun, choreographies berbeda di aktivitas yang setiap interaksi mewakili satu atau lebih pertukaran pesan, yang melibatkan dua atau lebih *participant*.



Gambar 2.11 Contoh choreographies

Sumber : Object Management Group (2011)

Selain tiga sub-model yang dimiliki oleh BPMN, BPMN juga memiliki lima elemen dasar dalam memodelkan proses bisnis (Object Management Group,2011), yaitu:

1. Flow Object
 

Flow Object adalah elemen prafis utama yang mendefinisikan aturan atau peran dalam suatu proses bisnis. Pada BPMN terdapat tiga flow object:

  1. Events
  2. Activities
  3. Gateways
  
2. Data yang direpresentasikan dengan empat elemen berikut:
  1. Data Objects
  2. Data Inputs
  3. Data Outputs
  4. Data Stores
  
3. Connection Object merupakan elemen yang menghubungkan flow object satu sama lain. Ada empat connection object dalam BPMN:
  1. Sequence Flows
  2. Message Flows
  3. Associations
  4. Data Associations

4. Swimlane

Swimlane digunakan untuk mengelompokkan elemen pemodelan utama, ada dua cara dalam mengelompokkan, yaitu:

1. Pool
2. Lanes

5. Artifact

Artifact digunakan untuk memberikan informasi tambahan mengenai proses yang dimodelkan, ada dua artifacts standar yang dimiliki oleh BPMN yaitu:

1. Group
2. Text Annotation

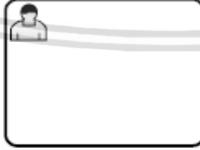
Setiap elemen dasar memiliki notasi tersendiri untuk memudahkan dalam memodelkan proses bisnis. Penjelasan notasi dapat dilihat pada Tabel 2.7.

**Tabel 2.7 Notasi BPMN**

Nama	Notasi	Keterangan
<b>Event</b>		
<i>Start Event</i>		Start Event digunakan untuk menandakan mulainya dari suatu proses.
<i>End Event</i>		Start Event digunakan untuk menandakan akhir dari suatu proses.
<i>Intermediet Event</i>		Intermediet event merupakan notasi yang menggambarkan kegiatan lain diantara awal dan akhir proses. Intermediet event akan mempengaruhi aliran proses atau choreographic. Tetapi tidak pada mulai atau akhir dari suatu proses.
<i>Message Start Event</i>		<i>Message start event</i> merupakan notasi yang menggambarkan sebuah pesan diterima dari participant dan mempengaruhi mulainya suatu proses.

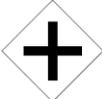


Tabel 2.7 Notasi BPMN (Lanjutan)

Nama	Notasi	Keterangan
<i>Message Intermediet Event</i>	<p><i>Throw</i></p>  <p><i>Catch</i></p> 	<i>Message Intermediet Event</i> merupakan notasi yang digunakan untuk menerima pesan (catch) dan mengirim pesan (throw) dari <i>participant</i> lain dan dapat mengganggu proses lain.
<b>Sequence flow</b>		
<i>Normal flow</i>		<i>Normal flow</i> digunakan untuk menggambarkan aliran dari aktiviti ke aktiviti lain yang berjalan pada proses.
<i>Message flow</i>		<i>Message flow</i> menggambarkan aliran pesan antara dua <i>participant</i> .
<b>Task</b>		
<i>Activity task</i>		Aktivitas merupakan notasi yang menggambarkan aktivitas dari perusahaan dalam suatu proses.
<i>Manual task</i>		Manual task merupakan notasi yang menggambarkan suatu aktivitas yang dilakukan secara manual.
<i>User task</i>		<i>User task</i> merupakan notasi yang menggambarkan aktifitas manusia dalam menjalankan sebuah tugas dengan bantuan aplikasi perangkat lunak .
<i>Receive Task</i>		<i>Receive task</i> merupakan notasi yang menggambarkan aktifitas menunggu sebuah pesan dari <i>external participant</i> .



Tabel 2.7 Notasi BPMN (Lanjutan)

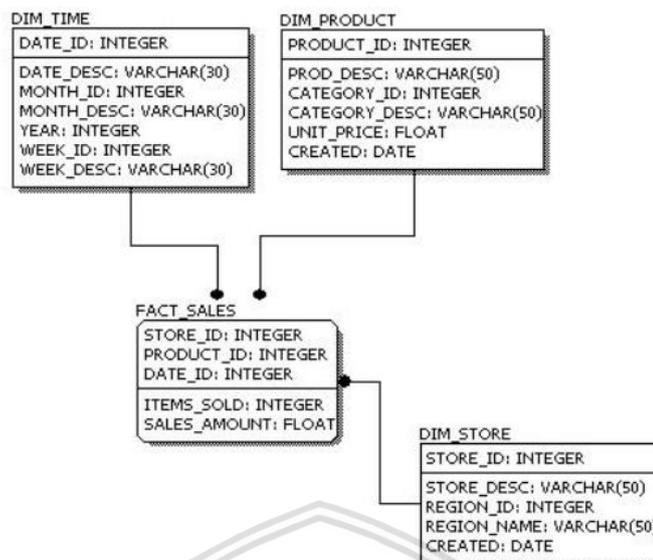
Nama	Notasi	Keterangan
<b>Gateway</b>		
<i>Exclusive Gateway</i>	 or 	<i>Exclusive gateway</i> merupakan notasi yang digunakan sebagai control flow, yang dimana berdasarkan dua kondisi “Ya” atau “Tidak”.
<i>Parallel Gateway</i>		<i>Parallel gateway</i> merupakan notasi yang digunakan untuk menggambarkan suatu kondisi yang dapat berjalan secara <i>parallel</i> .
<i>Parallel Event Based Gateway</i>		<i>Parallel event based gateway</i> merupakan notasi yang digunakan untuk menggambarkan suatu kondisi yang dapat berjalan secara <i>parallel</i> akan tetapi dapat berlangsung jika dipengaruhi oleh aktivitas lain.
<b>Swimlane</b>		
<i>Pool</i>		<i>Pool</i> merupakan notasi yang digunakan untuk merepresentasikan sebuah <i>participant</i> .
<i>Lane</i>		<i>Lane</i> merupakan notasi dari sub bagian dari pool.

Sumber: Object Management Group (2011)

## 2.8 Physical Data Model

*Physical Data Model* merupakan model yang digunakan untuk menggambarkan data serta hubungan antar data dalam sebuah basis data (S & Shalahuddin, 2014). *Physical Data Model* atau yang disingkat PDM biasanya terdiri atas sekumpulan tabel dan setiap tabel terdiri atas nama kolom dan tipe data kolom. PDM merupakan perancangan basis data yang sudah siap diimplementasikan. Sehingga nama table yang terdapat pada PDM merupakan nama asli yang digunakan pada tabel basis data. Gambar 2.12 merupakan contoh dari *physical data model*.





**Gambar 2.12 Contoh Physical Data Model**

Sumber : 1keydata (2018)

## 2.9 Web Seivice

*Web service* merupakan sebuah sistem perangkat lunak yang mendukung interaksi *machine-to-machine* diatas jaringan (Oracle, 2013). *Web service* dapat mengintegrasikan sistem yang berbeda dan menggunakan kembali sebagai fungsi bisnis. Pada penerapannya, *web service* dapat diterapkan dengan dua acara, yaitu, Representational State Transfer (RESTful) dan Simple Object Access Protocol (SOAP) . Penggunaan SOAP telah digunakan dan dikembangkan oleh banyak vendor (misal: Microsoft, Sund dan IBM). SOAP memiliki tiga komponen utama, yaitu, *service provider*, *service requester* dan *service broker*, dan didukung oleh komponen lain, seperti XML, SOAP-XML, WSDL, serta UDDI. Sedangkan di metode REST didasari oleh empat prinsip utama teknologi, yaitu, *Resource identifier through Uniform Resourse Identifier (URI)*, *Uniform Interface*, *self-descriptive messages* dan *satetful interactions through hyperlinks*. Metode REST menggunakan format standar (HTTP, JSON, XML, URI) sehingga metode ini lebih sederhana dari SOAP (Edhy & Khabib, 2012). Metode REST telah menjadi salah satu metode web service yang paling sesuai untuk aplikasi perangkat bergerak (Nolan, et al., 2014)

## 2.10 JavaScript Object Notation (JSON)

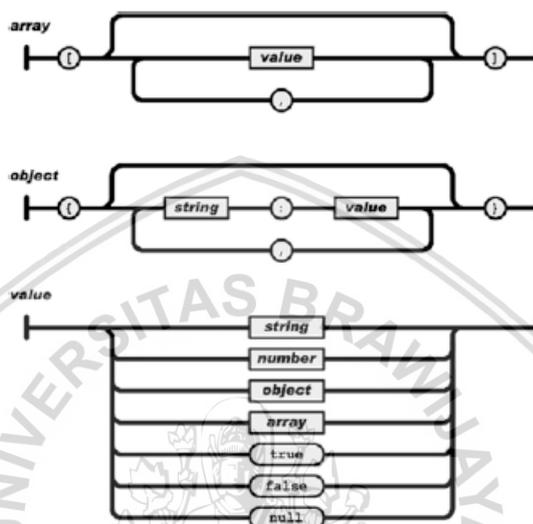
*JavaScript Object Nation (JSON)* merupakan format pertukaran data yang sering digunakan dalam pengembangan aplikasi perangkat bergerak (Wickham, 2018). Menurut Mark Wickham, ada beberapa alasan JSON menjadi salah satu pilihan format pertukaran data di aplikasi perangkat bergerak, yaitu:

1. JSON merupakan format pertukaran data yang ringan. JSON juga memberikan kemudahan pada mesin dalam membaca dan menulis format JSON.



2. Terdapat library yang dapat menangani format JSON.
3. Penggunaan JSON pada aplikasi akan menciptakan fleksibilitas dan perluasan arsitektur pada aplikasi.

JSON memiliki dua struktur data primitive untuk membangun sebuah format JSON yang dapat mewakili jenis hubungan data, dua strukturnya yaitu, JSONObject dan JSONArray. JSONObject merupakan kumpulan nilai yang tidak berurutan. Sedangkan JSONArray merupakan jenis struktur data JSON yang berurutan.



Gambar 2.13 JSON Building Block

Sumber: Wickham (2017)

## 2.11 JSON Web Token (JWT)

JSON Web Token (JWT) adalah standar terbuka (RFC 7519) yang digunakan untuk mentransmisikan informasi secara aman antar pihak dengan menggunakan objek JSON (JWT, n.d). Informasi ini dapat diverifikasi dan dipercaya karena ditandatangani secara digital. JWT dapat ditandatangani menggunakan rahasia (dengan algoritma HMAC) atau pasangan kunci publik / privat menggunakan RSA atau ECDSA.

## 2.12 Pengujian Perangkat Lunak

Pengujian atau *testing* merupakan tahap dimana suatu sistem yang telah dibuat, diuji untuk menemukan kesalahan yang dengan waktu yang relative singkat dan usaha yang minimum. Hal ini bertujuan untuk mencegah kesalahan terjadi ketika sistem sudah berjalan dan digunakan oleh pengguna. Metode perancangan kasus uji ini juga menyediakan mekanisme yang dapat membantu kelengkapan dari pengujian dan memberikan kemungkinan tertinggi untuk dapat menemukan berbagai kesalahan dalam sebuah sistem atau perangkat lunak (Presman, 2010).

### 2.12.1 *Black-box Testing*

Pengujian *black-box* salah satu metode pengujian yang digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang sudah dibuat dan *black-box* berfokus pada persyaratan fungsional perangkat, dimana memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional program (Presman, 2010). Menurut Pressman, terdapat kategori kesalahan dalam pengujian *black-box*:

1. Fungsi salah atau hilang
2. Kesalahan antarmuka
3. Kesalahan struktur data
4. Kesalahan perilaku atau kinerja
5. Kesalahan inisiasi atau penghentian

#### 2.12.1.2 *Validation Testing*

*Validation testing* merupakan pengujian yang dilakukan berdasarkan tindakan dan keluaran dari sistem yang dapat dilihat oleh pengguna (Presman, 2010). *Validation testing* dilakukan melalui serangkaian pengujian untuk memastikan kesesuaian perangkat lunak dengan persyaratan. Menurut Pressman (2010), setelah dilakukan pengujian terdapat salah satu dari dua kondisi yang memungkinkan terjadi, yaitu:

1. Karakteristik fungsi dan kinerja perangkat lunak sesuai dengan spesifikasi dan dapat diterima oleh pengguna.
2. Terungkapnya penyimpangan dari spesifikasi perangkat lunak dan kekurangan perangkat lunak akan dicatat.

#### 2.12.2 *Usability Testing*

*Usability* merupakan suatu aspek yang menilai seberapa mudah sistem dapat digunakan oleh pengguna (Nielsen, 2012). Sedangkan, *usability testing* merupakan proses pengujian untuk mengukur tingkat ketertudahan suatu sistem. Pengujian ini juga bertujuan untuk membatasi dan menghilangkan kesulitan pengguna dalam menggunakan sistem. Pengujian *usability* idealnya melibatkan pengguna langsung maupun tidak langsung (mengamati perilaku). Jumlah pengguna dari pengujian ini membutuhkan setidaknya lima orang (Lazar, et al., 2017). Menurut Mifsud (2015), *usability* dapat diukur dengan menggunakan 3 matrik yang berdasarkan ISO/IEC 9126-4, yaitu:

##### 1. *Effectiveness*

*Effectiveness* merupakan matrik yang digunakan untuk mengetahui tingkat keakuratan dan penyelesaian pengguna dalam mencapai sasaran. Matrik ini dapat diukur dengan menilai penyelesaian pengguna dalam menggunakan sistem. Penilaian ini menggunakan nilai 1 yang menggambarkan jika pengguna berhasil

menyelesaikan tugas dan 0 jika pengguna tidak berhasil menyelesaikan tugas. Berikut Persamaan 2.8 yang digunakan untuk mengukur *effectiveness*:

$$Effectiveness = \frac{\text{Jumlah tugas yang berhasil dilakukan}}{\text{Jumlah tugas yang dilakukan}} \times 100 \% \quad (2.8)$$

Hasil yang diperoleh dari matrik ini minimal mencapai rata-rata 78%, untuk mencapai hasil yang baik pada matrik *effectiveness* (Sauro, 2011). Selain menggunakan persamaan diatas matrik ini juga melihat jumlah kesalahan yang terjadi pengujian *usability*.

## 2. Efficiency

*Efficiency* merupakan matrik yang berhubungan dengan keakuratan dan penyelesaian pengguna dalam mencapai sasaran. Matrik ini diukur dengan menggunakan *task time*. *Task time* kemudian dihubungkan dengan tingkat penyelesaian. Berikut ini Persamaan 2.9 dan Persamaan 2.10 merupakan persamaan yang digunakan dalam mengukur *efficiency*:

$$Task\ Time = End\ Time - Start\ Time \quad (2.9)$$

$$Time\ Based\ Efficiency = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR} \quad (2.10)$$

**N:** Total tugas yang dikerjakan

**R:** Total pengguna

**n<sub>ij</sub>:** Hasil tingkat penyelesaian tugas pengguna, jika berhasil bernilai 1 dan jika tidak bernilai 0.

**t<sub>ij</sub>:** Waktu yang dihabiskan dalam menyelesaikan tugas. Apabila pengguna tidak dapat menyelesaikan tugas, mengukurnya dengan waktu yang dihabis pengguna sampai dengan pengguna tidak mampu menyelesaikan tugas.

## 3. Satisfaction

*Satisfaction* merupakan matrik yang digunakan untuk mengetahui kenyamanan dan penerimaan dari penggunaan sistem. Pengukuran *satisfaction* dapat dilakukan dengan menggunakan kuisisioner *satisfaction* yang sudah memiliki standar. Kuisisioner tersebut dapat diberikan setelah tugas dan/atau setelah pengujian *usability*. Menurut Mifsud (2015), ada beberapa kuisisioner yang dapat digunakan untuk mengukur *satisfaction*, yaitu, SUS, SUPR-Q, CSUQ, QUIS, SUMI, ASQ, NASA-TLX, SMEQ, UME, dan SEQ. Adapun kuisisioner yang disarankan

menurut Sauro (2010), yaitu SEQ dan SUS. Hal ini dikarenakan SEQ lebih mudah untuk ditanggapi oleh responden, mengelola dan membuat skor. Sedangkan SUS, menurut Sauro dapat memberikan hasil yang sangat akurat.

### 2.12.2.1 Single Ease Question

Single Ease Question merupakan salah satu kuisisioner untuk mengukur tingkat *satisfaction* dari suatu sistem (Sauro, 2012). Penilaian dengan menggunakan kuisisioner ini dapat dilakukan setelah pengguna menyelesaikan tugas yang telah diberikan ketika pengujian *usability*. Kuisisioner SEQ menggunakan satu pertanyaan mengenai seberapa mudah pengguna dalam menyelesaikan tugas dan penilaian SEQ menggunakan skala rating 1-7. Ada beberapa yang harus diperhatikan dalam menggunakan kuisisioner SEQ, yaitu rata-rata yang di dapatkan harus memperoleh 5 dan jika setiap tugas mendapatkan nilai dibawah 5 penguji harus menanyakan alasan pengguna memberikan nilai dibawah 5.

Overall, how difficult or easy did you find this task?

Very Difficult 1 2 3 4 5 6 7 Very Easy

Gambar 2.14 Contoh Kuisisioner Single Ease Question

Sumber : Sauro(2012)

### 2.12.3 Security Testing

*Security Testing* merupakan pengujian untuk memverifikasi bahwa perangkat lunak telah memenuhi persyaratan keamanan (Naik & Tripathy, 2008). Persyaratan keamanan terdiri dari *confidentiality*, *integrity*, dan *availability*. *Confidentiality* adalah persyaratan data yang dilindungi dari pengungkapan yang tidak sah. *Integrity* adalah persyaratan data yang dilindungi dari modifikasi yang tidak sah. *Availability* adalah persyaratan data yang dilindungi dari penyangkalan layanan kepada pengguna yang berwenang. Pengujian keamanan harus mencakup *scenario negative* seperti penyalahgunaan sistem.

## 2.13 Model View Presenter (MVP)

Model View Presenter merupakan salah satu pola yang biasa digunakan dalam pengembangan aplikasi android. Model View Presenter merupakan turunan dari MVC, yang dimana pola ini memisahkan *presentation layer* dengan *logic* (Leiva, 2014). Menurut Antonio Leiva, yang menjadi alasan adanya MVP ini, karena faktanya bahwa aktivitas android sering digabungkan dengan mekanisme antarmuka dan akses data.

Pada penerapannya MVP tidak memiliki standar akan tetapi secara konsep MVP terbagi menjadi 3, yaitu Model, View dan Presenter (Leiva, 2014).

#### 1. Model

Model bertugas sebagai penyedia data yang ditampilkan ke view.

## 2. View

View bertugas untuk memberikan tampilan kepada pengguna. View biasanya diterapkan dengan activity atau fragment tergantung kepada aplikasi yang akan dibuat.

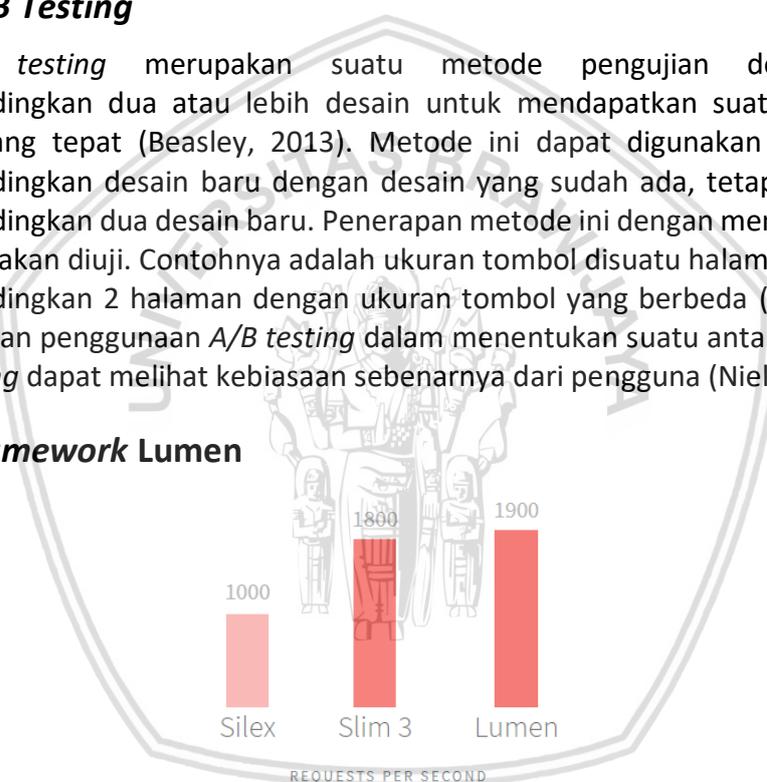
## 3. Presenter

Presenter bertanggung jawab untuk menjembatani antara view dan model. Presenter bertugas mengambil data dan mengembalikannya ke view. Terlihat seperti controller di MVC, akan tetapi presenter juga menentukan apa yang terjadi jika pengguna berinteraksi dengan view

### 2.14 A/B Testing

*A/B testing* merupakan suatu metode pengujian dengan cara membandingkan dua atau lebih desain untuk mendapatkan suatu keputusan desain yang tepat (Beasley, 2013). Metode ini dapat digunakan tidak hanya membandingkan desain baru dengan desain yang sudah ada, tetapi juga dapat membandingkan dua desain baru. Penerapan metode ini dengan memutuskan hal apa yang akan diuji. Contohnya adalah ukuran tombol disuatu halaman. Kita akan membandingkan 2 halaman dengan ukuran tombol yang berbeda (Gallo, 2017). Keuntungan penggunaan *A/B testing* dalam menentukan suatu antarmuka, yaitu *A/B testing* dapat melihat kebiasaan sebenarnya dari pengguna (Nielsen, 2005).

### 2.15 Framework Lumen



Gambar 2.15 Benchmark Framework Lumen

Sumber: Lumen (2018)

Lumen merupakan sebuah *micro framework* yang dibangun diatas *framework* Laravel. Lumen biasanya menjadi salah satu solusi dalam membangun REST API. Lumen juga memiliki fitur yang sama dengan Laravel seperti *eloquent*, *caching*, *routing*, *middleware* dan fitur laravel lainnya. *Framework* ini merupakan salah satu *micro framework* tercepat, hal ini dapat dilihat *benchmark* dari lumen yang ditunjukkan dalam Gambar 2.15.

### 2.15.1 Eloquent

Eloquent merupakan salah satu kemampuan yang dimiliki oleh *micro framework* lumen, yang dimana eloquent dapat bekerja langsung dengan database. Eloquent memungkinkan kita memiliki model yang dapat langsung berinteraksi dengan tabel basis data. Berikut Tabel 2.8 merupakan contoh kode sumber penerapan dari eloquent.

**Tabel 2.8 Contoh penerapan eloquent**

```

1  <?php
2  namespace App;
3  use Illuminate\Database\Eloquent\Model;
4
5  class Flight extends Model
6  {
7      /**
8       * The table associated with the model.
9       *
10     * @var string
11     */
12     protected $table = 'my_flights';
13 }

```

Sumber : Laravel (n.d)

### 2.16 Sampling Purposive

Sampling purposive merupakan salah satu teknik sampling dalam penelitian, yang dimana penentuan sampel berdasarkan pertimbangan tertentu (Sugiyono, 2017). Sampling purposive dapat disebut juga judgmental sampling, yaitu pengambilan sampel berdasarkan penilaian peneliti terkait siapa saja yang memenuhi syarat untuk dijadikan sampel. Selain itu juga, teknik sampling purposive tidak menggunakan angka pasti untuk menentukan ukuran sampel. Sampel diambil sebanyak mungkin sampai data purposive tersebut dianggap sudah menggambarkan apa yang menjadi tujuan dan permasalahan penelitian.

### 2.17 Skala Likert

Skala *Likert* merupakan skala yang sering digunakan dalam kuesioner untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang (Sugiyono, 2017). Skala likert umumnya menggunakan skala yang terdiri dari 5 poin secara terurut dimulai dari “Sangat Tidak Setuju” hingga “Sangat Setuju” (Bertram, 2016). Terkadang juga digunakan skala 4 poin atau angka genap lainnya yang bertujuan untuk menghasilkan ukuran ipsatif atau pilihan paksa. Setiap tingkat pada skala diberikan nilai numerik yang biasanya dimulai dari 1 dan

ditambah satu untuk tiap tingkatnya. Berikut pada Gambar 2.16 merupakan contoh dari skala *likert* dengan menggunakan skala 5 poin.



**Gambar 2.16 Contoh skala likert**  
Sumber : Bertram (2016)

Dalam implementasi skala *likert* terdapat beberapa pernyataan yang masing-masing memiliki bobot nilai tertentu. Proses perhitungan hasil dari skala *likert* diawali dengan memberikan bobot nilai pada tiap jawaban. Berikut pada Tabel 2.9 merupakan rincian bobot nilai dari jawaban.

**Tabel 2.9 Bobot Nilai Jawaban**

Pilihan Jawaban	Bobot Nilai
A = Sangat Setuju	5
B = Setuju	4
C = Netral	4
D = Kurang	2
E = Sangat Kurang	1

Sumber: Sugiyono (2017)

Setelah bobot nilai dan persentase nilai telah ditentukan, selanjutnya adalah mengkalikan jumlah jawaban dengan bobot nilainya. Untuk perhitungannya akan dijelaskan pada Persamaan 2.11 berikut:

1. Jawaban Sangat Setuju (A) =  $n \times 5$
2. Jawaban Setuju (B) =  $n \times 4$
3. Jawaban Netral (C) =  $n \times 3$
4. Jawaban Tidak Setuju (D) =  $n \times 2$
5. Jawaban Sangat Tidak Setuju (E) =  $n \times 1$

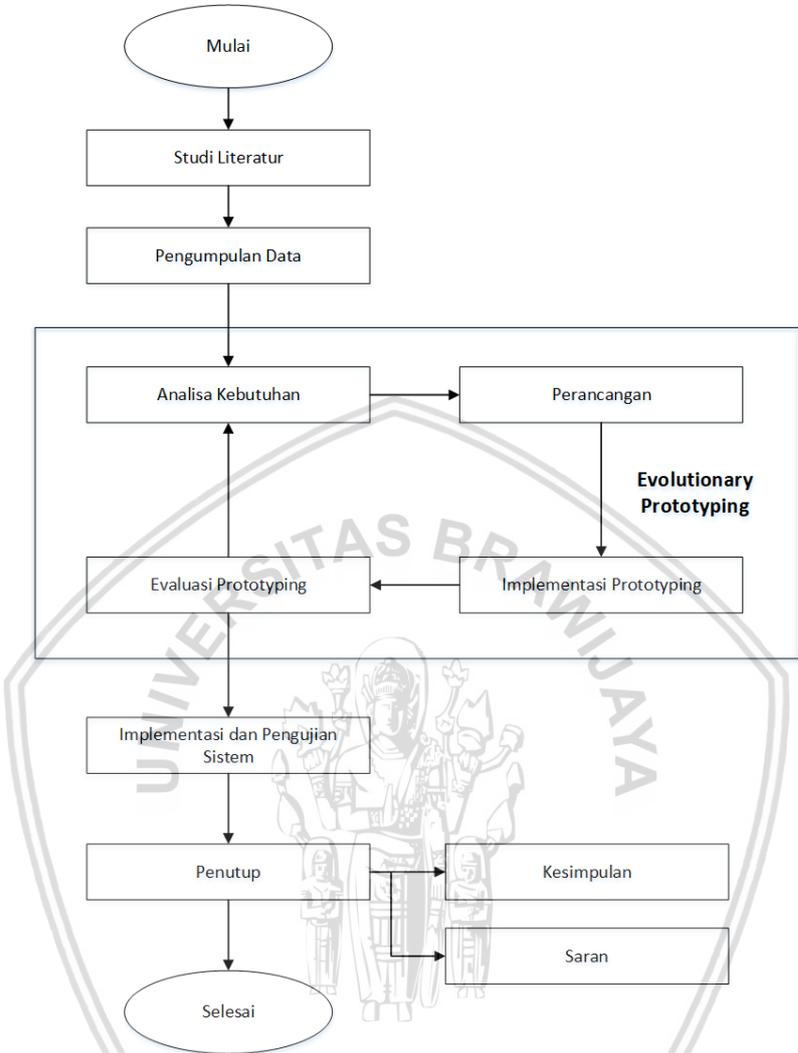
$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.11)$$

$n$  merupakan jumlah dari responden yang menjawab. Perhitungan selanjutnya adalah mencari presentase dari tingkat persetujuan terhadap suatu pernyataan yang dijelaskan pada Persamaan 2.12 dan Persamaan 2.13.

$$\text{Jumlah Skor Ideal} = 5 \times \text{Jumlah Responden} \quad (2.12)$$

$$\text{Tingkat Persetujuan (\%)} = (\text{Total Nilai} : \text{Jumlah Skor Ideal}) \times 100\% \quad (2.13)$$

### BAB 3 METODOLOGI



Gambar 3.1 Diagram metodologi penelitian

#### 3.1 Studi Literatur

Studi literatur merupakan tahap dimana mencari dan mempelajari refrensi yang memuat teori-teori atau penelitian terdahulu yang berkaitan dengan penelitian yang akan dilakukan. Sehingga hal ini semakin mempermudah dalam penyelesaian penelitian. Adapun sumber yang digunakan dalam studi literatur yaitu, *e-book*, jurnal, media masa ataupun penelitian sebelumnya.

#### 3.2 Pengumpulan Data

Pengumpulan data menjadi tahap dimana membantu penelitian dalam memvalidasi permasalahan dan juga digunakan dalam mengembangkan sistem filkom apps berbasis Android. Selain itu juga pengumpulan data ini digunakan

untuk menggali kebutuhan pengguna dan menggali informasi sistem yang sudah berjalan. Pada pengumpulan data ini menggunakan teknik observasi dan wawancara.

### 3.3 Analisis dan Perancangan

Tahap Analisis dan Perancangan merupakan tahap yang digunakan untuk menentukan kebutuhan sistem yang diperoleh dari hasil pengumpulan data pada tahap sebelumnya dan dikemudian kebutuhan yang sudah didefinisikan diterjemahkan ke dalam sebuah model perangkat lunak seperti *Unified Modeling Language* serta perancangan antarmuka pengguna yang akan dibuat sesuai dengan kebutuhan pengguna.

### 3.4 Implementasi Prototyping

Setelah melakukan tahap perancangan, kemudian dilanjutkan ke implementasi prototipe. Tahap implementasi prototipe dilakukan berdasarkan rancangan yang sudah dibuat. Pada tahap ini nantinya akan menghasilkan prototipe versi 0.0 untuk perancangan pertama kali, dan kemudian akan ada evaluasi prototipe dari pengguna. Versi dari prototipe akan disesuaikan dengan jumlah iterasi yang dilakukan.

### 3.5 Evaluasi Prototyping

Evaluasi *prototyping* merupakan tahap mengevaluasi dari hasil pembuatan prototyping, yang kemudian diuji oleh pengguna, untuk memastikan kebutuhan dari pengguna sudah tercapai atau tidak. Apabila hasil dari evaluasi ini tidak sesuai dengan kebutuhan pengguna, akan dilakukan kembali analisis kebutuhan berdasarkan hasil dari evaluasi prototyping. Evaluasi prototyping menggunakan *A/B testing* dan *usability testing*.

### 3.6 Implementasi dan Pengujian Sistem

Implementasi dan Pengujian sistem merupakan tahap dimana hasil dari evaluasi *prototyping*, sistem disempurnakan menjadi sebuah sistem yang sudah terintegrasi dengan data dan fungsional - fungsional yang terdapat pada sistem dapat digunakan dengan baik. Sistem ini juga tidak lagi sebuah prototipe tetapi sudah menjadi sebuah sistem versi rilis yang siap digunakan oleh pengguna. Selain itu juga, dilakukan pengujian sistem untuk menguji jalannya sistem, apakah sesuai dengan rancangan atau tidak. Metode pengujian yang digunakan adalah *validation testing*, *security testing* dan *usability testing*.

### 3.7 Penutup

Penutup merupakan tahap terakhir dari penelitian ini, yang dimana terdiri dari kesimpulan dan saran. kesimpulan merupakan hasil dari penelitian dan kemudian disertai saran yang nantinya dapat digunakan dan dikembangkan kembali dengan baik untuk penelitian selanjutnya.

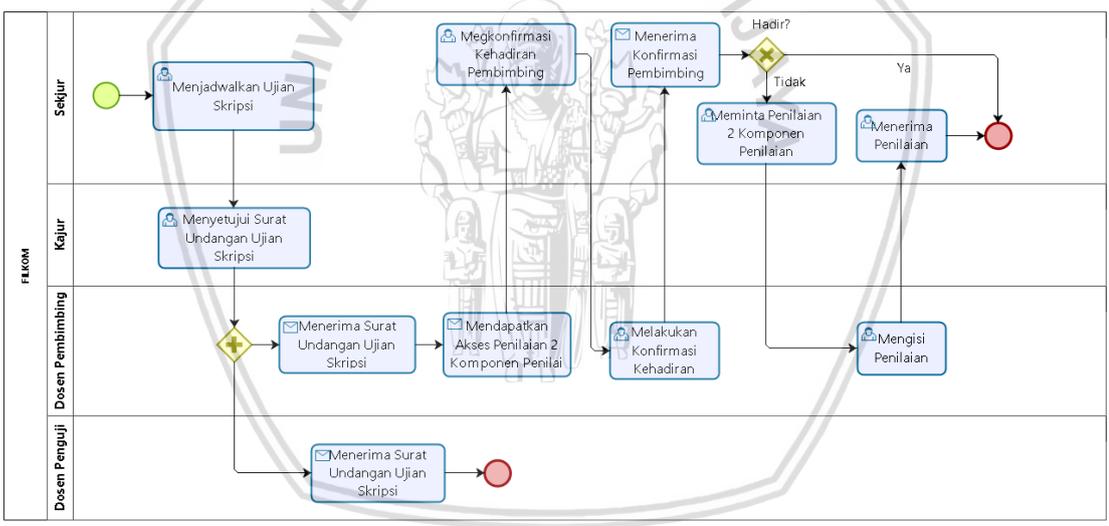
## BAB 4 ANALISIS

Pada bab ini membahas mengenai analisis dan perancangan Aplikasi Penilaian Ujian Skripsi Berbasis Android. Bab ini terdiri dari analisis proses bisnis, analisa kebutuhan dan perancangan. Tahap pertama peneliti melakukan analisis proses bisnis melalui *literature* yaitu Pedoman Skripsi Fakultas Ilmu Komputer Universitas Brawijaya Versi 2. Kemudian, peneliti melakukan analisis kebutuhan dari pengamatan pada fungsional penilaian ujian skripsi di *website* FILKOM APPS, mekanisme dan prosedur penilaian ujian skripsi dan wawancara dengan pengembang sistem penilaian ujian skripsi berbasis *web*.

### 4.1 Analisis Proses Bisnis

Bagian ini menjelaskan proses bisnis yang terjadi pada penilaian ujian skripsi dan direpresentasikan dengan *Business Process Model Notation (BPMN)*. Berikut ini merupakan hasil analisis proses bisnis penilaian ujian skripsi:

#### 4.1.1 Penilaian Komponen Proses Pengerjaan Skripsi Dan Kualitas Penulisan Skripsi.



**Gambar 4.1** Proses bisnis penilaian komponen proses pengerjaan skripsi dan kualitas penulisan

Proses ini menjabarkan bagaimana alur penilaian skripsi pada komponen proses pengerjaan skripsi dan kualitas penulisan skripsi. Komponen ini hanya dapat dilakukan pembimbing. Pada Gambar 4.1 merupakan penggambaran dari proses bisnis penilaian komponen proses pengerjaan skripsi dan kualitas penulisan dan berikut adalah penjelasan proses penilaian komponen proses pengerjaan skripsi dan kualitas penulisan pada Gambar 4.1.

1. Sekjur menjadwalkan ujian skripsi.
2. Kajur menyetujui surat undangan ujian skripsi.

3. Dosen pembimbing menerima surat undangan ujian skripsi dan mendapatkan akses penilaian 2 komponen penilaian, yaitu proses pengerjaan skripsi dan kualitas penulisan.
4. Sekjur mengkonfirmasi kehadiran pembimbing.
5. Dosen pembimbing melakukan konfirmasi kehadiran.
6. Sekjur menerima konfirmasi pembimbing
  - a. Jika konfirmasi dari pembimbing tidak dapat hadir, maka sekjur akan meminta penilaian komponen proses pengerjaan skripsi dan kualitas dokumen dan dosen pembimbing wajib memberikan penilaian pengerjaan skripsi dan kualitas dokumen.
  - b. Jika konfirmasi dari pembimbing dapat hadir, maka proses akan berakhir. Pembimbing dapat melakukan penilaian proses pengerjaan skripsi dan kualitas dokumen sebelum atau ketika ujian skripsi berlangsung.

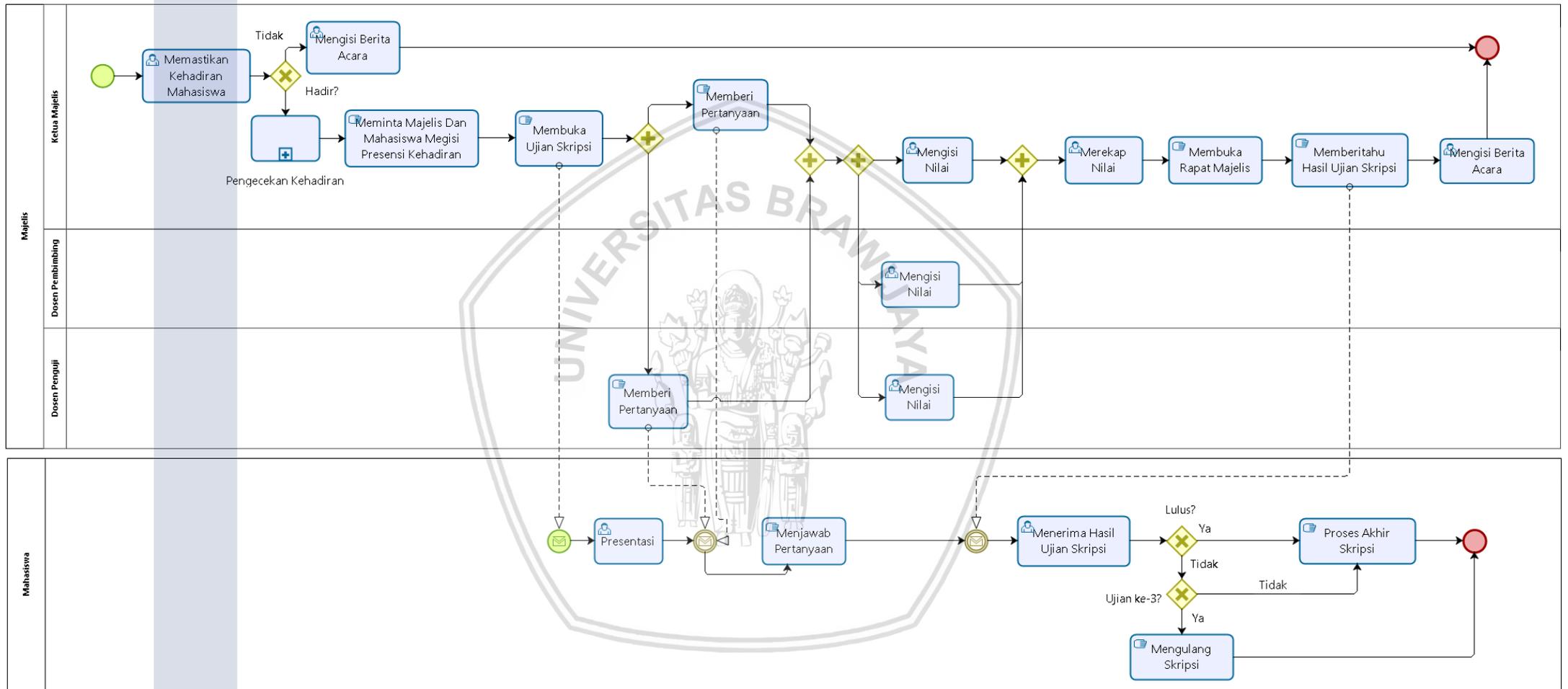
#### 4.1.2 Pelaksanaan Ujian Skripsi

Pada bagian ini menjabarkan bagaimana alur dari proses pelaksanaan ujian skripsi, sehingga nantinya hasil dari pelaksanaan ujian skripsi akan menentukan kelulusan mahasiswa. Pada Gambar 4.2 merupakan penggambaran dari proses pelaksanaan ujian skripsi. Berikut ini penjabaran dari Gambar 4.2:

1. Ketua majelis memastikan kehadiran mahasiswa
  - a. Jika pada saat pelaksanaan ujian skripsi, mahasiswa tidak dapat hadir, maka ketua majelis mengisi berita acara.
  - b. Jika mahasiswa hadir, maka ketua majelis melakukan pengecekan kehadiran anggota majelis. Pengecekan kehadiran anggota majelis merupakan *sub-process* dari proses pelaksanaan ujian skripsi yang akan dijelaskan di bagian proses pengecekan kehadiran mahasiswa.
2. Setelah melakukan pengecekan kehadiran majelis, ketua majelis meminta mahasiswa dan majelis mengisi presensi.
3. Ketua majelis membuka ujian skripsi.
4. Mahasiswa melakukan presentasi.
5. Dosen penguji dan ketua majelis memberikan pertanyaan
6. Mahasiswa menjawab pertanyaan.
7. Ketua majelis, dosen pembimbing dan penguji mengisi nilai.
8. Ketua majelis merekap nilai.
9. Ketua majelis membuka rapat majelis. Rapat majelis dilakukan untuk menentukan kelulusan mahasiswa.
10. Ketua majelis memberitahu hasil ujian skripsi kepada mahasiswa dan mahasiswa menerima hasil ujian skripsi.

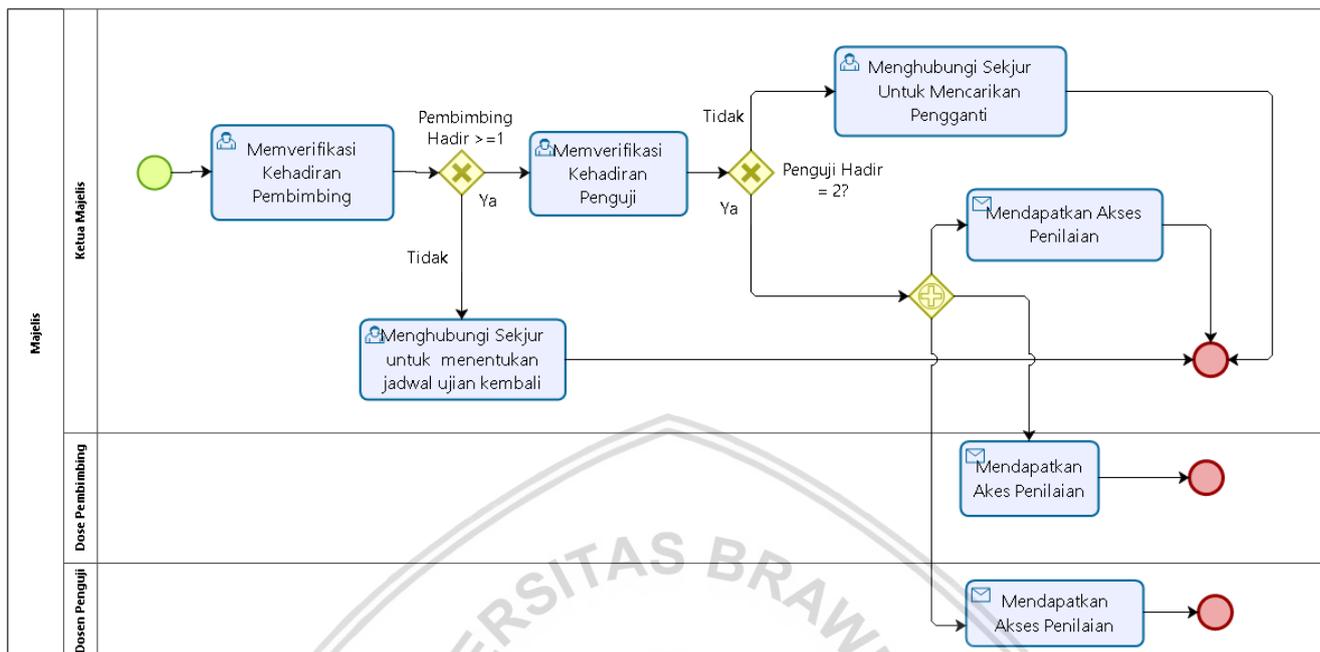
- a. Jika dinyatakan lulus, maka mahasiswa dapat melanjutkan ke proses akhir skripsi.
  - b. Jika dinyatakan tidak lulus, maka mahasiswa mengulang kembali ujian skripsi. Jika mahasiswa tidak mengulang ujian skripsi sebanyak 3 kali. Mahasiswa dapat melanjutkan ke proses akhir skripsi.
  - c. Jika dinyatakan tidak lulus, maka mahasiswa mengulang kembali ujian skripsi. Jika mahasiswa sudah mengulang ujian skripsi sebanyak 3 kali. Mahasiswa mengulang kembali pengerjaan skripsi.
11. Ketua majelis mengisi berita acara.





Gambar 4.2 Proses bisnis pelaksanaan ujian skripsi

### 4.1.3 Pengecekan Kehadiran Majelis



**Gambar 4.3** Proses bisnis pengecekan kehadiran majelis

Pada bagian ini menjabarkan bagaimana alur dari pengecekan kehadiran majelis. Pengecekan kehadiran majelis merupakan *sub-proses* dari proses bisnis pelaksanaan ujian skripsi. Pada Gambar 4.3 merupakan penggambaran dari proses bisnis pengecekan kehadiran majelis dan berikut ini penjabaran proses bisnis pengecekan kehadiran majelis.

1. Ketua majelis memastikan kehadiran pembimbing
  - a. Jika pembimbing yang hadir lebih dari sama dengan 1, maka ketua majelis memverifikasi kehadiran penguji menggunakan sistem.
  - b. Jika pembimbing yang hadir tidak lebih dari sama dengan 1, maka ketua majelis menghubungi sekjur untuk menentukan jadwal ujian kembali.
2. Ketua majelis memverifikasi kehadiran penguji.
  - a. Jika penguji yang hadir sama dengan 2, maka ketua majelis, dosen pembimbing dan penguji mendapatkan hak akses penilaian.
  - b. Jika penguji yang hadir tidak sama dengan 2, maka ketua majelis menghubungi sekjur untuk mencari pengganti.

### 4.1.4 Kode dan Aktivitas Proses Bisnis

Bagian ini membahas mengenai kode dan aktivitas proses bisnis. Kode aktivitas ini nantinya akan digunakan di analisis kebutuhan. Kode dan aktivitas

proses bisnis dibuat berdasarkan hasil analisis proses bisnis. Berikut kode dan aktivitas proses bisnis:

**Tabel 4.1 Kode proses bisnis**

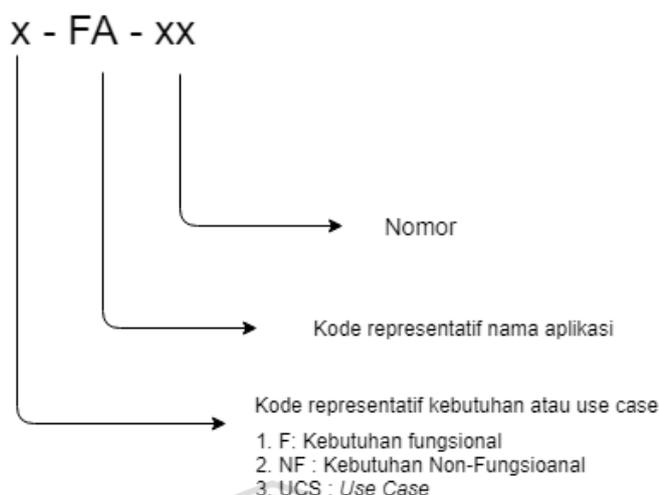
Kode Proses Bisnis	Nama Proses Bisnis
<b>PB-FA-1</b>	Penilaian Komponen Pengerjaan Skripsi Dan Kualitas Penulisan
<b>PB-FA-2</b>	Pelaksanaan Ujian Skripsi
<b>PB-FA-3</b>	Pengecekan Kehadiran Majelis

**Tabel 4.2 Kode aktivitas proses bisnis**

Kode Aktivitas	Kode Proses Bisnis	Nama Aktivitas
<b>APB-FA-1</b>	PB-FA-1	Menerima Surat Undangan Skripsi
<b>APB-FA-2</b>	PB-FA-2	Mengisi Nilai
<b>APB-FA-3</b>	PB-FA-2	Memastikan Kehadiran Mahasiswa
<b>APB-FA-4</b>	PB-FA-3	Memverifikasi Kehadiran Dosen Pembimbing
<b>APB-FA-5</b>	PB-FA-3	Memverifikasi Kehadiran Dosen Penguji
<b>APB-FA-6</b>	PB-FA-2	Merekap Nilai
<b>APB-FA-7</b>	PB-FA-2	Mengisi Berita Acara

## 4.2 Analisis Kebutuhan

Analisis Kebutuhan merupakan tahap dimana peneliti melakukan analisis sistem penilain ujian skripsi berbasis website. Tahap ini dilakukan berdasarkan observasi dan wawancara yang telah dilaksanakan. Pengamatan dilakukan untuk mengetahui proses yang terjadi pada sistem, seperti, proses input penilaian ujian skripsi, proses rekapitulasi dan proses input berita acara pengujian skripsi melalui *website* FILKOM APPS. Selain itu, dilakukan wawancara dengan pengembang sistem penilain ujian skripsi untuk mengetahui lebih lanjut proses yang terjadi pada aplikasi penialian ujian skripsi. Tidak hanya observasi dan wawancara, peneliti menggunakan buku panduan skripsi v2 sebagai literature terkait informasi pelaksanaan ujian skripsi yang terdapat pada Fakultas Ilmu Komputer. Pada analisis kebutuhan memiliki aturan penomoran sebagaimana yang ditunjukkan pada Gambar 4.4



**Gambar 4.4 Aturan penomoran analisis kebutuhan**

#### 4.2.1 Penggalan Informasi Alur Kerja Sistem *Existing*

Sebelum membangun Sistem Penilaian Ujian Skripsi Berbasis Android, peneliti terlebih dahulu mempelajari proses penilaian ujian skripsi yang telah ditetapkan oleh FILKOM dan sistem penilaian ujian skripsi berbasis *web*. Beberapa hal yang perlu diamati adalah proses penilaian, rekapitulasi nilai ujian, proses mengisi berita acara ujian skripsi serta alur mendapatkan informasi jadwal skripsi mahasiswa. Berdasarkan hasil pengamatan dan wawancara yang telah dilakukan, alur kerja sistem penilaian ujian skripsi adalah sebagai berikut.

- a. Alur kerja pada sistem penilaian ujian skripsi berbasis website
  1. *User* (dosen penguji, ketua majelis dand osen pembimbing) harus terlebih dahulu login dengan akun yang telah di daftarkan di *website* FILKOM APPS.
  2. Sistem akan memberikan akses kepada pembimbing untuk menilai 2 komponen penilaian, yaitu proses pengerjaan skripsi dan kualitas penulisan dokumen skripsi ketika surat undangan ujian skripsi telah diterbitkan.
  3. Sistem hanya memberikan izin untuk menilai komponen proses pengerjaan skripsi hanya kepada dosen pembimbing
  4. Sistem memberikan akses kepada pembimbing dalam menilai komponen kualitas presentasi, kualitas argumentasi jawaban dan penilaian kuitas artifak ketika ujian berlangsung.
  5. Sistem dapat memberikan rekomendasi penguji ketika tedapat penguji yang tidak dapat hadir pada saat ujian.
  6. Sistem tidak akan memberikan akses penilaian ujian skripsi ketika ketua majelis belum melakukan verifikasi kehadiran dosen pembimbing dan dosen penguji.

7. Sistem tidak memberikan akses dalam proses penilaian ketika syarat pelaksanaan ujian skripsi tidak terpenuhi.
8. Sistem akan memberikan peringatan kepada dosen pembimbing dan dosen penguji apabila penilaian terdapat perbedaan dua angka atau lebih antara dosen pembimbing dan dosen penguji.
9. Sistem dapat menampilkan hasil rekapitulasi nilai
10. Sistem dapat mengisi berita acara
11. Berita acara dan hasil rekapitulasi nilai hanya dapat diakses oleh ketua majelis.
12. Sistem memberikan deskripsi dari setiap nilai.
13. Sistem dapat menampilkan rincian nilai majelis di sisi ketua majelis
14. Sistem dapat memberitahu *gap* nilai di sisi ketua majelis

Alur kerja sistem penilaian ujian skripsi yang telah didapatkan akan menjadi acuan peneliti dalam membangun Aplikasi Penilaian Ujian Skripsi Berbasis Android.

#### 4.2.2 Identifikasi Pengguna

Identifikasi pengguna merupakan tahap dimana menjelaskan siapa saja yang akan menggunakan sistem penilaian ujian skripsi. Hasil identifikasi pengguna berdasarkan penggalan informasi alur sistem penilaian ujian skripsi dari buku panduan skripsi v2 serta observasi sistem, dan interview, terdapat 3 pengguna yaitu dosen pembimbing, dosen penguji dan ketua majelis. Berikut penjelasan mengenai pengguna dapat dilihat di Tabel 4.3.

**Tabel 4.3 Pengguna aplikasi penilaian ujian skripsi**

Pengguna	Deskripsi
<b>Dosen Pembimbing</b>	Dosen pembimbing merupakan pihak yang berkewajiban untuk membimbing mahasiswa dalam mengerjakan skripsi dan juga mendampingi mahasiswa ketika pelaksanaan ujian skripsi. Selain itu juga dosen pembimbing dalam proses penilaian berperan dalam menilai ke 5 komponen penilaian, yaitu proses pengerjaan skripsi, kualitas dokumen skripsi, kualitas argumentasi jawaban dan kualitas artifak.
<b>Dosen Penguji</b>	Dosen penguji merupakan pihak yang berkewajiban untuk menguji skripsi yang sudah dikerjakan oleh mahasiswa dan juga memberikan penilaian berdasarkan 4 komponen penilaian ujian, yaitu kualitas presentasi, kualitas penulisan dokumen skripsi, kualitas argumentasi jawaban dan kualitas artifak. Selain itu penguji juga menilai naskah publikasi. Penguji juga dapat merangkap sebagai ketua majelis.

**Tabel 4.3 Pengguna aplikasi penilaian ujian skripsi (Lanjutan)**

Pengguna	Deskripsi
<b>Ketua Majelis</b>	Ketua majelis memiliki peran mengatur jalannya pelaksanaan ujian skripsi, menguji dan menilai skripsi.

#### 4.2.3 Kebutuhan Fungsional

Kebutuhan fungsional merupakan tahap dimana mendefinisikan kebutuhan yang harus dipenuhi oleh sistem. Kebutuhan fungsional ini didefinisikan berdasarkan hasil analisis proses bisnis dan analisis kebutuhan.

**Tabel 4.4 Analisis kebutuhan fungsional dosen pembimbing**

Kode Fungsional	Kode Aktivitas	Deskripsi
<b>F-FA-1</b>	-	Sistem harus menyediakan fungsional login.
<b>F-FA-2</b>	APB-FA-1	Sistem dapat memberikan informasi pelaksanaan ujian skripsi yang terdiri dari informasi mahasiswa, informasi skripsi, informasi jadwal dan tempat pelaksanaan ujian skripsi dan informasi dosen penguji, ketua majelis dan dosen pembimbing.
<b>F-FA-3</b>	-	Sistem dapat menerima dan menyimpan notifikasi penilaian.
<b>F-FA-4</b>	APB-FA-2	Sistem harus menyediakan fungsional penilaian pada 5 komponen penilaian ujian skripsi, yaitu proses pengerjaan skripsi, kualitas dokumen skripsi, kualitas argumentasi jawaban dan kualitas artefak.
<b>F-FA-5</b>	-	Sistem menyediakan fungsional riwayat penilaian ujian skripsi. Riwayat penilaian ujian skripsi berisi informasi mahasiswa, informasi skripsi, informasi jadwal dan tempat pelaksanaan ujian skripsi dan informasi dosen penguji, ketua majelis dan dosen pembimbing dan hasil penilaian skripsi.

**Tabel 4.5 Analisiskebutuhan fungsional dosen penguji**

Kode Fungsional	Kode Aktivitas	Deskripsi
<b>F-FA-6</b>	-	Sistem harus menyediakan fungsional login.
<b>F-FA-7</b>	-	Sistem dapat menerima dan menyimpan notifikasi penilaian.

**Tabel 4.5 Analisis kebutuhan fungsional dosen penguji (Lanjutan)**

Kode Fungsional	Kode Aktivitas	Deskripsi
<b>F-FA-8</b>	APB-FA-1	Sistem dapat memberikan informasi pelaksanaan ujian skripsi yang terdiri dari informasi mahasiswa, informasi skripsi, informasi jadwal dan tempat pelaksanaan ujian skripsi dan informasi dosen penguji, ketua majelis dan dosen pembimbing.
<b>F-FA-9</b>	APB-FA-2	Sistem harus menyediakan fungsional penilaian pada 4 komponen penilaian ujian , yaitu kualitas presentasi, kualitas penulisan dokumen skripsi, kualitas argumentasi jawaban dan kualitas artifak serta penilaian naskah publikasi.
<b>F-FA-10</b>	-	Sistem menyediakan fungsional riwayat penilaian ujian skripsi. Riwayat penilaian ujian skripsi berisi informasi mahasiswa, informasi skripsi, informasi jadwal dan tempat pelaksanaan ujian skripsi dan informasi dosen penguji, ketua majelis dan dosen pembimbing dan hasil penilaian skripsi.

**Tabel 4.6 Analisis kebutuhan fungsional ketua majelis**

Kode Kebutuhan	Kode Aktivitas	Deskripsi
<b>F-FA-11</b>	-	Sistem harus menyediakan fungsional login.
<b>F-FA-12</b>	APB-FA-1	Sistem dapat memberikan informasi pelaksanaan ujian skripsi yang terdiri dari informasi mahasiswa, informasi skripsi, informasi jadwal dan tempat pelaksanaan ujian skripsi dan informasi dosen penguji, ketua majelis dan dosen pembimbing.
<b>F-FA-13</b>	-	Sistem dapat menerima dan menyimpan notifikasi penilaian.
<b>F-FA-14</b>	APB-FA-2	Sistem harus menyediakan fungsional penilaian pada 4 komponen penilaian ujian , yaitu kualitas presentasi, kualitas penulisan dokumen skripsi, kualitas argumentasi jawaban dan kualitas artifak serta penilaian naskah publikasi.

Tabel 4.6 Analisis kebutuhan fungsional ketua majelis (Lanjutan)

Kode Kebutuhan	Kode Aktivitas	Deskripsi
<b>F-FA-15</b>	-	Sistem menyediakan fungsional riwayat penilaian ujian skripsi. Riwayat penilaian ujian skripsi berisi informasi mahasiswa, informasi skripsi, informasi jadwal dan tempat pelaksanaan ujian skripsi dan informasi dosen penguji, ketua majelis dan dosen pembimbing dan hasil penilaian skripsi.
<b>F-FA-16</b>	APB-FA-3, APB-FA-4 & APB-FA-5	Sistem harus menyediakan fungsional untuk verifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji.
<b>F-FA-17</b>	-	Sistem harus menyediakan fungsional mengatur waktu ujian.
<b>F-FA-18</b>	APB-FA-6	Sistem dapat melakukan rekapitulasi nilai dari semua aspek penilaian yang telah dinilai oleh dosen pembimbing dan penguji.
<b>F-FA-19</b>	APB-FA-7	Sistem harus menyediakan fungsional untuk menuliskan berita acara.

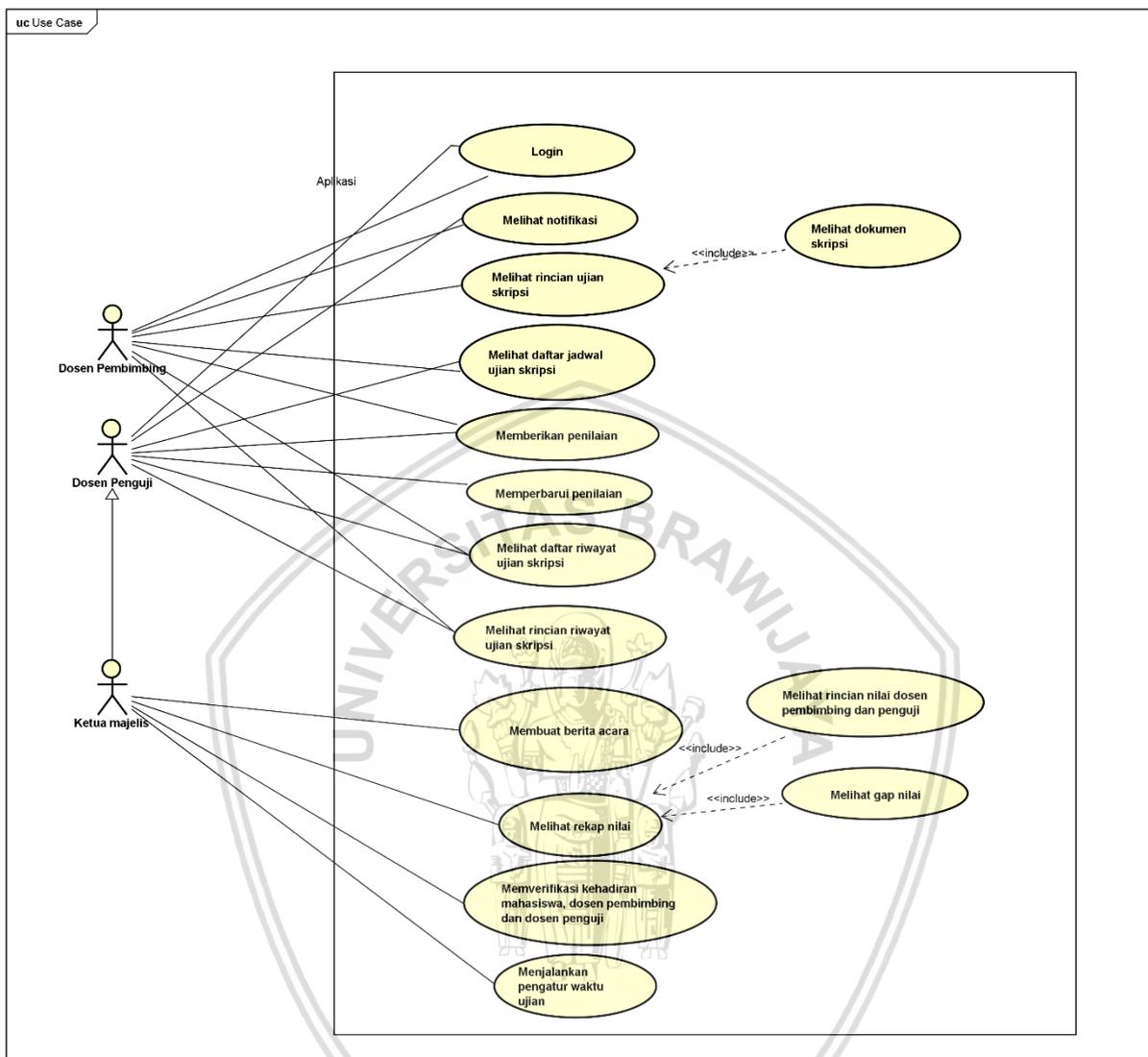
#### 4.2.4 Kebutuhan Non-Fungsional

Kebutuhan Non-Fungsional merupakan kebutuhan tambahan yang dibutuhkan oleh sistem. Kebutuhan Fungsional didapatkan berdasarkan wawancara dan pengamatan. Berikut kebutuhan fungsional dari sistem penilaian ujian skripsi berbasis android:

Tabel 4.7 Analisis kebutuhan non-fungsional

Kode Kebutuhan	Nama	Deskripsi
<b>NF-FA-1</b>	Security	Sistem memiliki keamanan yang terjamin untuk mencegah dari pihak-pihak yang ingin mengganggu jalannya sistem. Keamanan ini di fokuskan pada <i>webserice</i> , yang dimana aplikasi memerukan <i>otentikasi</i> di beberapa <i>service</i> . <i>Otentikasi</i> yang diterapkan adalah <i>JSON Web Token (JWT)</i> dan <i>web service</i> berjalan diatas protocol <i>HTTPS</i> .
<b>NF-FA-2</b>	Usability	Sistem memiliki tingkat kemudahan yang baik ketika digunakan oleh pengguna.

## 4.2.5 Pemodelan Use Case Diagram



**Gambar 4.5 Use Case Diagram aplikasi penilaian ujian skripsi**

Berdasarkan Gambar 4.5, Aplikasi Penilaian Ujian Skripsi memiliki 3 aktor yaitu dosen pembimbing, dosen penguji dan ketua majelis. Dosen penguji dan dosen pembimbing dapat login, melihat notifikasi, melihat dokumen skripsi, memberikan penilaian, melihat daftar jadwal ujian skripsi, melihat rincian ujian skripsi, melihat daftar riwayat ujian skripsi dan melihat rincian riwayat ujian skripsi. Ketua majelis memiliki fungsional yang sama dengan dosen pembimbing dan dosen penguji, akan tetapi ketua majelis dapat membuat berita acara, melihat rekap nilai, melihat rincian nilai dosen pembimbing dan dosen penguji, memverifikasi kehadiran mahasiswa, dosen pembimbing & dosen penguji, dan menjalankan pengatur waktu ujian.

#### 4.2.6 Pemodelan Use Case Scenario

Setelah perancangan *use case diagram*, maka dilakukan perancangan *use case scenario* yang bertujuan untuk menjabarkan setiap aktivitas pada *use case diagram*. Berikut perancangan *use case scenario*.

##### 4.2.6.1 Login

Tabel 4.8 Use Case Scenario Login

Item	Deskripsi
Kode	UCS-FA-01
Nama Use Case	Login
Aktor	Dosen penguji dan dosen pembimbing
Deskripsi	Use case login menjelaskan bagaimana aktor dapat login dengan menggunakan akun yang telah terdaftar.
Pra-Kondisi	Aplikasi menampilkan halaman login.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan <i>username</i> dan <i>password</i> yang sesuai.</li> <li>2. Pengguna menekan tombol <i>sign in</i>.</li> </ol>
Post-Kondisi	Aplikasi menampilkan halaman beranda.
Alternatif	<ol style="list-style-type: none"> <li>1. Akun tidak ditemukan Aplikasi menampilkan <i>pemberitahuan</i> bahwa <i>username</i> atau <i>password</i> tidak sesuai.</li> <li>2. Pengguna tidak melengkapi <i>username</i> dan/atau <i>password</i> Aplikasi menampilkan <i>pemberitahuan</i> bahwa pengguna belum melengkapi <i>username</i>, <i>password</i>, atau keduanya.</li> </ol>
Kode Kebutuhan Fungsional	F-FA-1, F-FA-6, dan F-FA-10

##### 4.2.6.2 Melihat Notifikasi

Tabel 4.9 Use Case Scenario Melihat Notifikasi

Item	Deskripsi
Kode	UCS-FA-02
Nama Use Case	Melihat notifikasi
Aktor	Dosen penguji dan dosen pembimbing
Deskripsi	Use case melihat notifikasi bagaimana aktor dapat melakukan melihat notifikasi .

Tabel 4.10 *Use Case Scenario* Melihat Notifikasi (Lanjutan)

Pra-Kondisi	Aplikasi menampilkan halaman beranda.
Tindakan	1. Pengguna memilih menu notifikasi.
Post-Kondisi	Aplikasi menampilkan halaman notifikasi.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-3, F-FA-7, dan F-FA-13

#### 4.2.6.3 Melihat Daftar Jadwal Ujian Skripsi

Tabel 4.11 *Use Case Scenario* Melihat Daftar Jadwal Ujian Skripsi

Item	Deskripsi
Kode	UCS-FA-03
Nama Use Case	Melihat daftar jadwal ujian
Aktor	Dosen pembimbing dan dosen penguji
Deskripsi	<i>Use case</i> melihat daftar jadwal ujian menjelaskan bagaimana aktor dapat melihat daftar ujian.
Pra-Kondisi	Aplikasi menampilkan halaman beranda
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu jadwal ujian pada menu <i>navigation drawer</i>.</li> <li>2. Pengguna memilih tab menu jadwal ujian terbaru</li> </ol>
Post-Kondisi	Aplikasi menampilkan daftar jadwal ujian
Alternatif	<ol style="list-style-type: none"> <li>1. Mencari daftar jadwal ujian dengan nama mahasiswa Pengguna dapat mencari daftar jadwal ujian dengan memasukkan nama mahasiswa yang ingin dicari.</li> <li>2. Memfilter daftar jadwal ujian Penguan dapat memfilter daftar jadwal ujian berdasarkan peran, sehingga pengunadapat memilih peran apa saja yang ingin ditampilkan</li> </ol>
Kode Kebutuhan Fungsional	F-FA-2 , F-FA-8, dan F-FA-12

#### 4.2.6.4 Melihat Rincian Ujian Skripsi

**Tabel 4.12 Use Case Scenario Melihat Rincian Ujian Skripsi**

Item	Deskripsi
Kode	UCS-FA-04
Nama Use Case	Melihat rincian ujian skripsi
Aktor	Dosen pembimbing dan dosen penguji
Deskripsi	<i>Use case</i> melihat rincian informasi ujian skripsi menjelaskan bagaimana aktor dapat melihat rincian ujian skripsi.
Pra-Kondisi	Aplikasi menampilkan halaman daftar jadwal ujian.
Tindakan	1. Pengguna memilih daftar jadwal ujian.
Post-Kondisi	Aplikasi menampilkan rincian informasi ujian skripsi.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-2 , F-FA-8, dan F-FA-12

#### 4.2.6.5 Melihat Dokumen Skripsi

**Tabel 4.13 Use Case Scenario Melihat Dokumen Skripsi**

Item	Deskripsi
Kode	UCS-FA-05
Nama Use Case	Melihat dokumen skripsi
Aktor	Dosen pembimbing dan dosen penguji
Deskripsi	<i>Use case</i> melihat dokumen skripsi menjelaskan bagaimana aktor dapat melihat dokumen skripsi.
Pra-Kondisi	Aplikasi menampilkan halaman rincian informasi ujian skripsi.
Tindakan	1. Pengguna menekan tombol dokumen. Aplikasi menampilkan dokumen skripsi.
Post-Kondisi	Aplikasi menampilkan dokumen skripsi.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-2 , F-FA-8, dan F-FA-12.

#### 4.2.6.6 Memverifikasi Kehadiran Mahasiswa, Dosen Pembimbing Dan Dosen Penguji

**Tabel 4.14 Use Case Scenario Memverifikasi Kehadiran Mahasiswa, Dosen Pembimbing Dan Dosen Penguji**

Item	Deskripsi
Kode	UCS-FA-06
Nama Use Case	Memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji
Aktor	Ketua majelis
Deskripsi	Use case memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji menjelaskan bagaimana aktor dapat melakukan verifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji.
Pra-Kondisi	Aplikasi menampilkan halaman verifikasi
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna melakukan verifikasi.</li> <li>2. Pengunan menekan tombol submit.</li> </ol>
Post-Kondisi	Aplikasi menampilkan pemberitahuan verifikasi berhasil.
Alternatif	<ol style="list-style-type: none"> <li>1. Belum mengisi kehadiran Aplikasi menampilkan pemberitahuan apabila terdapat kehadiran yang belum terisi .</li> </ol>
Kode Kebutuhan Fungsional	F-FA-16

#### 4.2.6.7 Memberikan Penilaian

**Tabel 4.15 Use Case Scenario Memberikan Penilaian**

Item	Deskripsi
Kode	UCS-FA-07
Nama Use Case	Memberikan penilaian skripsi
Aktor	Dosen pembimbing dan dosen penguji
Deskripsi	Use case memberikan penilaian menjelaskan bagaimana aktor dapat memberikan penilaian.
Pra-Kondisi	Aplikasi menampilkan halaman penilaian.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna melakukan penilaian.</li> <li>2. Penguan menekan tombol submit.</li> </ol>

**Tabel 4.16 Use Case Scenario Memberikan Penilaian (Lanjutan)**

Post-Kondisi	Aplikasi menampilkan pemberitahuan data penilaian sudah tersimpan.
Alternatif	1. Penilaian Belum Lengkap Apabila pengguna belum melengkapi penilaian. Aplikasi akan memberikan peringatan.
Kode Kebutuhan Fungsional	F-FA-4, F-FA-9, dan F-FA-15.

**4.2.6.8 Memperbarui Penilaian****Tabel 4.17 Use Case Scenario Memperbarui Penilaian**

Item	Deskripsi
Kode	UCS-FA-08
Nama Use Case	Memperbarui penilaian
Aktor	Dosen pembimbing dan dosen penguji
Deskripsi	<i>Use case</i> memperbarui penilaian menjelaskan bagaimana aktor dapat memperbarui penilaian.
Pra-Kondisi	Aplikasi menampilkan halaman hasil penilaian.
Tindakan	1. Pengguna menekan tombol <i>edit</i> . 2. Aplikasi menampilkan halaman penilaian. 3. Pengguna mengubah penilaian. 4. Penggunaan menekan tombol submit
Post-Kondisi	Aplikasi akan menampilkan pemberitahuan data penilaian berhasil diperbarui.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-4, F-FA-9, dan F-FA-15.

**4.2.6.9 Melihat Rekap Nilai****Tabel 4.18 Use Case Scenario Melihat Rekap Nilai**

Item	Deskripsi
Kode	UCS-FA-09
Nama Use Case	Melihat Rekap Nilai

**Tabel 4.19 Use Case Scenario Melihat Rekap Nilai (Lanjutan)**

Aktor	Ketua majelis
Deskripsi	<i>Use case</i> melihat rekap nilai menjelaskan bagaimana aktor dapat melihat rekap nilai.
Pra-Kondisi	Pengguna telah melakukan penilaian.
Tindakan	1. Pengguna menekan tab menu rekap nilai & berita acara.
Post-Kondisi	Aplikasi menampilkan hasil rekap nilai.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-18

**4.2.6.10 Melihat Gap Nilai****Tabel 4.20 Use Case Scenario Melihat Gap Nilai**

<b>Item</b>	<b>Deskripsi</b>
Kode	UCS-FA-10
Nama Use Case	Melihat <i>gap</i> nilai
Aktor	Ketua majelis.
Deskripsi	<i>Use case</i> melihat <i>gap</i> menjelaskan bagaimana aktor dapat melihat <i>gap</i> nilai pembimbing dan penguji.
Pra-Kondisi	Aplikasi menampilkan halaman rekap nilai.
Tindakan	1. Pengguna menekan tombol selengkapnya pada bagian <i>gap</i> nilai.
Post-Kondisi	Aplikasi menampilkan halaman <i>gap</i> nilai.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-18

**4.2.6.11 Melihat Rincian Penilaian Dosen Pembimbing Dan Dosen Penguji****Tabel 4.21 Use Case Scenario Melihat Rincian Penilaian Dosen Pembimbing dan Dosen Penguji**

<b>Item</b>	<b>Deskripsi</b>
Kode	UCS-FA-11

**Tabel 4.22 Use Case Scenario Melihat Rincian Penilaian Dosen Pembimbing dan Dosen Penguji (Lanjutan)**

Nama Use Case	Melihat rincian penilaian dosen pembimbing dan penguji.
Aktor	Ketua majelis.
Deskripsi	<i>Use case</i> melihat rincian nilai penilaian dosen pembimbing dan penguji menjelaskan bagaimana aktor dapat melihat rincian penilaian dosen pembimbing dan penguji.
Pra-Kondisi	Pengguna telah berada di halaman rekap nilai.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna menekan tombol selengkapnya pada nilai ujian.</li> <li>2. Aplikasi menampilkan halaman rincian dosen penguji dan pembimbing.</li> </ol>
Post-Kondisi	Aplikasi menampilkan halaman rincian penilaian dosen pembimbing dan dosen penguji.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-18

#### 4.2.6.12 Membuat Berita Acara

**Tabel 4.23 Use Case Scenario Membuat Berita Acara Ujian Telah Selesai**

Item	Deskripsi
Kode	UCS-FA-12
Nama Use Case	Membuat berita acara ujian telah selesai
Aktor	Ketua majelis.
Deskripsi	<i>Use case</i> membuat berita acara ujian telah selesai menjelaskan bagaimana aktor dapat membuat berita ketika ujian telah selesai.
Pra-Kondisi	Pengguna berada di halaman rekap nilai.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna mengisikan berita acara.</li> <li>2. Pengguna menekan tombol submit.</li> </ol>
Post-Kondisi	Aplikasi menampilkan pesan bahwa data telah tersimpan.

**Tabel 4.24 Use Case Scenario Membuat Berita Acara Ujian Telah Selesai (Lanjutan)**

Alternatif	1. Berita acara belum dimasukkan Aplikasi akan menampilkan pesan bahwa pengguna belum mengisi berita acara.
Kode Kebutuhan Fungsional	F-FA-19

**Tabel 4.25 Use Case Scenario Membuat Berita Acara Penjadwalan Ulang**

Item	Deskripsi
Kode	UCS-FA-13
Nama Use Case	Membuat berita acara penjadwalan ulang ujian
Aktor	Ketua majelis.
Deskripsi	Pengguna dapat membuat berita penjadwalan ulang ujian.
Pra-Kondisi	Status kehadiran mahasiswa tidak hadir dan aplikasi menampilkan halaman berita acara.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna mengisi berita acara.</li> <li>2. Pengguna menekan tombol submit berita acara.</li> </ol>
Post-Kondisi	Aplikasi Menampilkan Pesan Bahwa Data Telah Tersimpan.
Alternatif	<ol style="list-style-type: none"> <li>1. Berita acara belum dimasukkan Aplikasi akan menampilkan pesan bahwa pengguna belum mengisi berita acara.</li> <li>2. Tidak Ada Koneksi Aplikasi akan menampilkan pesan tidak ada koneksi apabila koneksi pengguna sedang bermasalah.</li> </ol>
Kode Kebutuhan Fungsional	F-FA-19

#### 4.2.6.13 Melihat Daftar Riwayat Ujian Skripsi

Tabel 4.26 *Use Case Scenario* Melihat Daftar Riwayat Ujian Skripsi

Item	Deskripsi
Kode	UCS-FA-14
Nama Use Case	Melihat daftar riwayat ujian skripsi
Aktor	Ketua majelis, pembimbing dan penguji.
Deskripsi	<i>Use case</i> riwayat ujian skripsi menjelaskan bagaimana aktor dapat melihat daftar riwayat ujian skripsi .
Pra-Kondisi	Aplikasi menampilkan halaman beranda.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna memilih menu jadwal ujian pada menu <i>navigation drawer</i>.</li> <li>2. Pengguna memilih tab menu riwayat.</li> </ol>
Post-Kondisi	Aplikasi menampilkan daftar informasi ujian skripsi yang telah diselesaikan pengguna.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-5 , F-FA-10, dan F-FA-15

#### 4.2.6.14 Melihat Rincian Riwayat Ujian Skripsi

Tabel 4.27 *Use Case Scenario* Melihat Rincian Riwayat Ujian Skripsi

Item	Deskripsi
Kode	UCS-FA-15
Nama Use Case	Melihat rincian riwayat ujian skripsi
Aktor	Dosen pembimbing, dosen penguji, dan ketua majelis.
Deskripsi	<i>Use case</i> melihat rincian riwayat ujian skripsi menjelaskan bagaimana aktor dapat melihat rincian informasi ujian skripsi yang telah diselesaikan pengguna.
Pra-Kondisi	Aplikasi menampilkan halaman daftar riwayat ujian.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna memilih daftar riwayat ujian.</li> </ol>
Post-Kondisi	Aplikasi menampilkan rincian informasi ujian skripsi yang telah diselesaikan pengguna.
Alternatif	-

**Tabel 4.28 Use Case Scenario Melihat Rincian Riwayat Ujian Skripsi (Lanjutan)**

Kode Kebutuhan Fungsional	F-FA-5 , F-FA-10, dan F-FA-15
---------------------------	-------------------------------

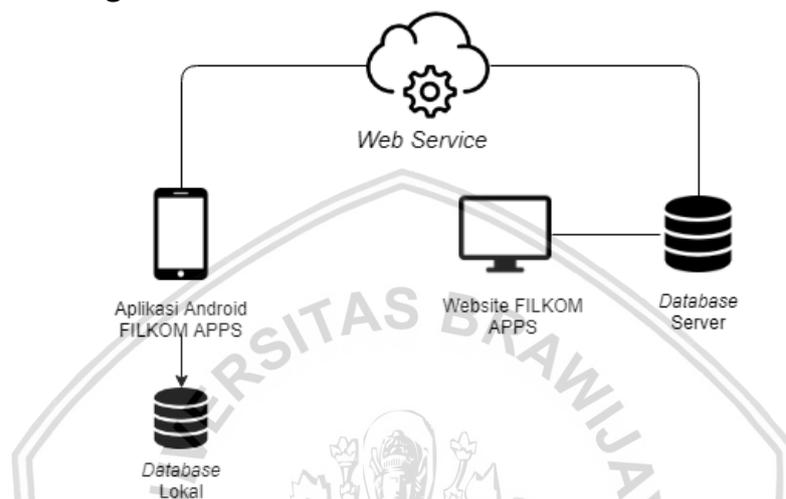
**4.2.6.15 Menjalankan Pengatur Waktu Ujian****Tabel 4.29 Use Case Scenario Menjalankan Pengatur Waktu Ujian**

Item	Deskripsi
Kode	UCS-FA-16
Nama Use Case	Menjalankan timer waktu ujian
Aktor	Ketua majelis.
Deskripsi	<i>Use case</i> pengatur waktu ujian menjelaskan bagaimana aktor dapat menjalankan pengatur waktu ujian.
Pra-Kondisi	Aplikasi menampilkan halaman penilaian.
Tindakan	<ol style="list-style-type: none"> <li>1. Pengguna menekan menu waktu.</li> <li>2. Pengguna menekan tombol mulai.</li> </ol>
Post-Kondisi	Aplikasi menjalankan pengatur waktu.
Alternatif	-
Kode Kebutuhan Fungsional	F-FA-20

## BAB 5 PERANCANGAN

Pada bagian ini akan membahas rancangan dari aplikasi penilaian ujian skripsi berbasis android. Adapun rancangan yang akan dibahas pada bab ini, yaitu perancangan arsitektur sistem, activity diagram, sequence diagram, class diagram, perancangan basis data, perancangan struktur data, dan perancangan antarmuka pengguna.

### 5.1.1 Perancangan Arsitektur Sistem



**Gambar 5.1 Rancangan arsitektur sistem**

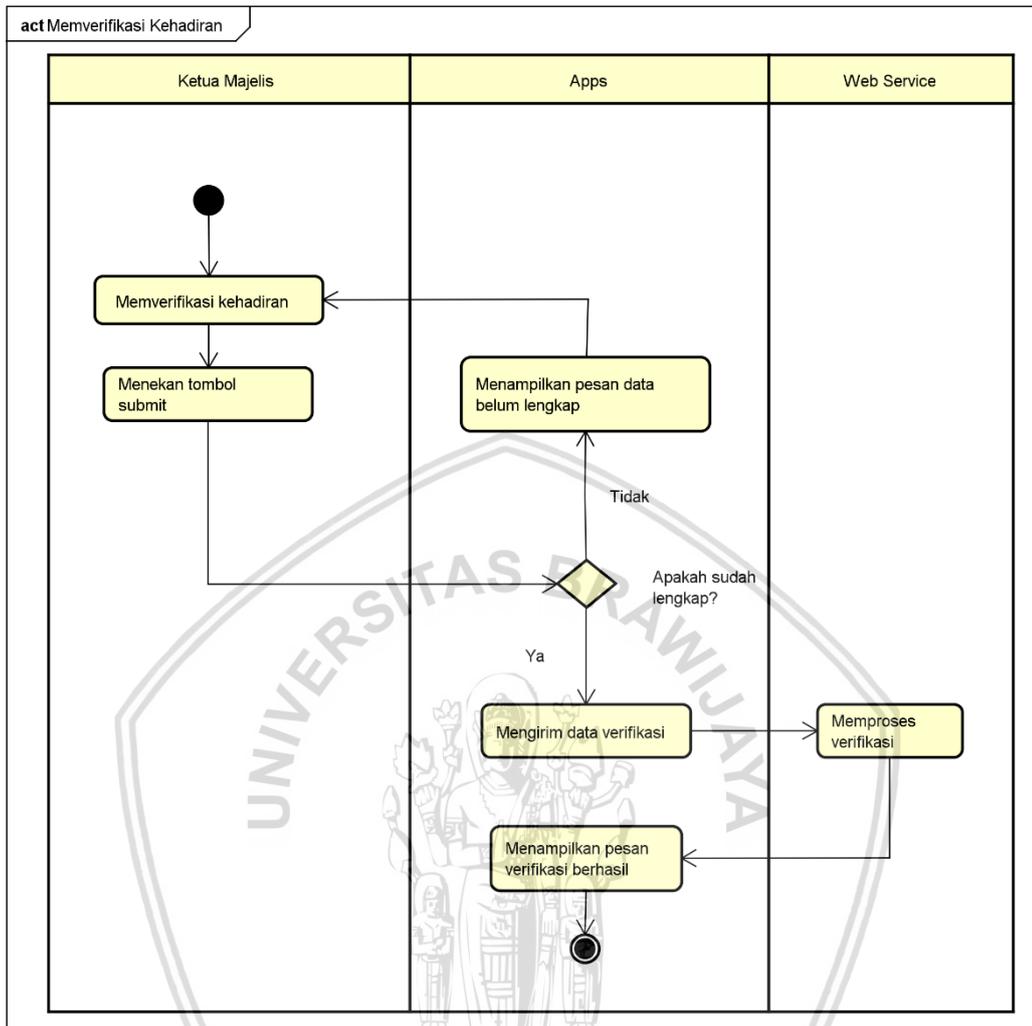
Pada Gambar 5.1 merupakan rancangan dari arsitektur sistem yang baru. Pada proses di arsitektur yang baru terdapat penambahan aplikasi FILKOM APPS berbasis android. Aplikasi ini tidak dapat langsung mengakses data yang terdapat pada *database server*. Sehingga untuk mengakses data tersebut diperlukannya *web service* untuk menjembatani antara aplikasi FILKOM APPS berbasis android dengan database server.

*Web service* ini nantinya akan melakukan komunikasi dengan aplikasi android dengan menggunakan metode REST API. Kemudian, dengan metode REST API Aplikasi android akan terlebih dahulu mengakses *url web service* dan *web service* akan menampilkan data yang diakses melalui *database server* dengan menggunakan format Javascript Object Notation (JSON). Aplikasi kemudian mengolahnya menjadi suatu tampilan yang dapat dimengerti oleh pengguna.

### 5.1.2 Perancangan Activity Diagram

Subbab ini membahas aliran kerja dari *use case* yang telah dijelaskan sebelumnya. Aliran kerja digambarkan dengan menggunakan *activity diagram*. Tidak semua *use case* yang digambarkan dalam *activity diagram*, adapun *use diagram* yang akan digambarkan, yaitu memverifikasi kehadiran mahasiswa, pembimbing dan penguji (UCS-FA-06), memberikan penilaian (UCS-FA-07), melihat rekap nilai (UCS-FA-09) dan membuat berita acara (UCS-FA-12).

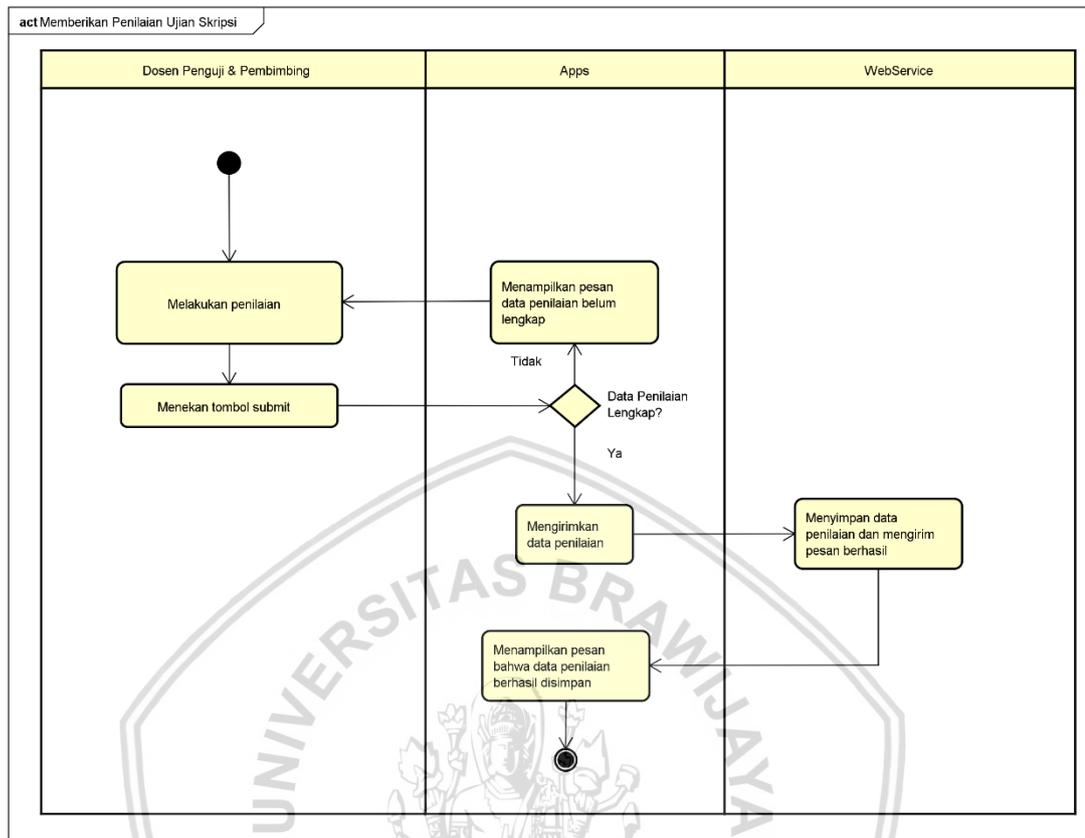
### 5.1.2.1 Memverifikasi Kehadiran Mahasiswa, Pembimbing Dan Penguji



**Gambar 5.2 Activity diagram memverifikasi kehadiran mahasiswa, pembimbing dan penguji**

Pada Gambar 5.2 merupakan alur kerja memverifikasi kehadiran mahasiswa, pembimbing dan penguji yang dimodelkan menggunakan *activity diagram*. Penggambaran alur kerja ini berdasarkan *use case* UCS-FA-06. Alur kerja dari memverifikasi kehadiran dimulai dari pengguna memverifikasi kehadiran mahasiswa, pembimbing, dan penguji. Lalu, pengguna menekan tombol submit dan terdapat pengecekan kondisi. Pertama, kondisi ketika pengguna sudah memverifikasi kehadiran dan data yang diisi sudah lengkap maka aplikasi akan mengirimkan data verifikasi dan diproses oleh *web service*. *Web service* kemudian mengirimkan pesan berhasil dan ditampilkan melalui aplikasi. Kedua, kondisi kelengkapan data verifikasi yang dimasukan pengguna tidak lengkap, aplikasi akan menampilkan pemberitahuan data belum lengkap dan akan kembali ke langkah memverifikasi kehadiran.

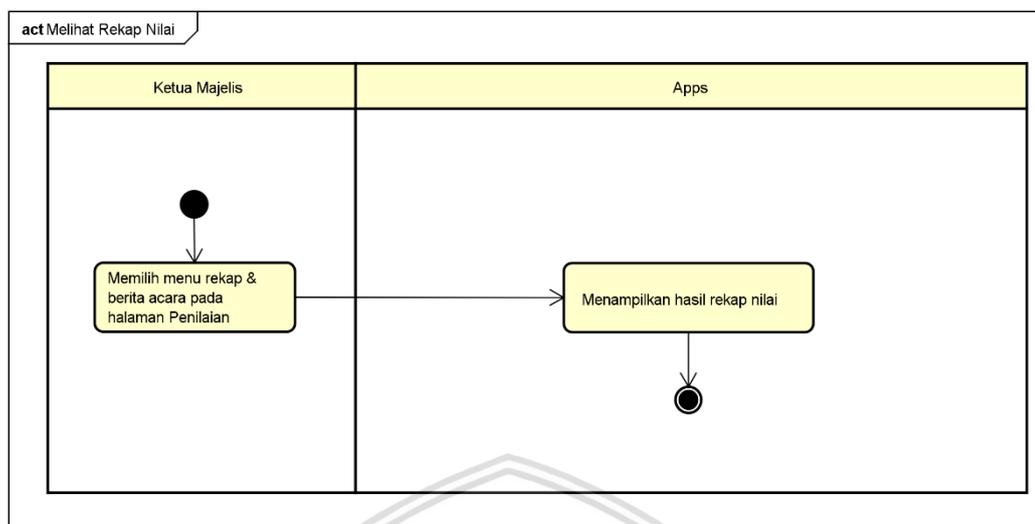
### 5.1.2.2 Memberikan Penilaian



**Gambar 5.3 Activity diagram melakukan penilaian**

Pada Gambar 5.3 merupakan alur kerja melakukan penilaian yang dimodelkan menggunakan *activity diagram*. Penggambaran alur kerja ini berdasarkan *use case* UCS-FA-07. Alur kerja dari memberikan penilaian dimulai dari pengguna melakukan penilaian. Lalu, pengguna menekan tombol submit dan terdapat pengecekan kondisi. Kondisi pertama ketika pengguna sudah melakukan penilaian dan data yang dimasukan sudah lengkap maka aplikasi akan mengirimkan data penilaian dan disimpan oleh *web service*. *Web service* kemudian mengirimkan pesan berhasil dan ditampilkan melalui aplikasi. Kedua, kondisi kelengkapan data penilaian yang dimasukan pengguna tidak lengkap, aplikasi akan menampilkan pemberitahuan data belum lengkap dan akan kembali ke langkah melakukan penilaian.

### 5.1.2.3 Melihat Rekap Nilai

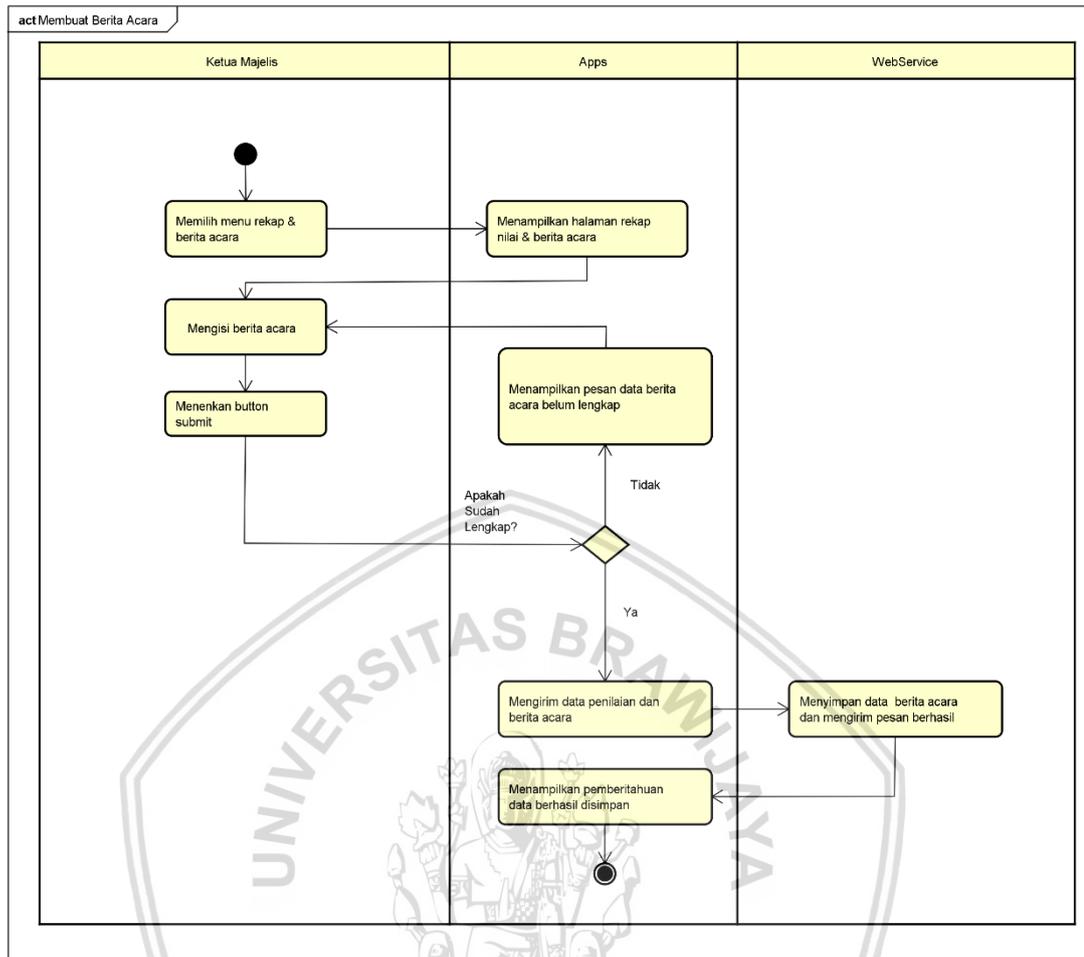


**Gambar 5.4 Activity diagram melihat rekap nilai**

Pada Gambar 5.4 merupakan alur kerja melihat rekap nilai yang dimodelkan menggunakan *activity diagram*. Penggambaran alur kerja ini berdasarkan *use case* UCS-FA-08. Alur kerja dari merekap nilai dimulai dari pengguna menekan memilih menu rekap nilai & berita acara pada halaman penilaian. Kemudian aplikasi menampilkan hasil rekap nilai.

### 5.1.2.4 Membuat Berita Acara

Pada Gambar 5.5 merupakan alur kerja membuat berita acara yang dimodelkan menggunakan *activity diagram*. Penggambaran alur kerja ini berdasarkan *use case* UCS-FA-12. Alur kerja dari membuat berita acara dimulai dari pengguna memilih menu rekap nilai & berita acara. Aplikasi menampilkan halaman rekap nilai & berita acara, kemudian, pengguna mengisi berita acara dan menekan tombol submit. Setelah itu, terdapat pemeriksaan kondisi. Kondisi pertama, kondisi dimana data berita acara yang dimasukkan pengguna sudah lengkap, aplikasi akan mengirim data berita acara ke *web service*. *Web service* akan memproses verifikasi kehadiran dan mengirim pesan berhasil. Lalu, aplikasi akan menampilkan pesan berhasil. Kondisi kedua, kondisi data berita acara tidak lengkap, aplikasi akan menampilkan pesan data berita acara belum lengkap dan alur kembali ke mengisi berita acara.



Gambar 5.5 Activity diagram membuat berita acara

### 5.1.3 Perancangan Sequence Diagram

Bagian ini untuk menggambarkan kelakuan objek dengan mendeskripsikan waktu hidup objek dan pesan dikirimkan dan diterima antar objek dan dibuat berdasarkan use case diagram dan class diagram yang telah dibuat sebelumnya. Sequence diagram yang digambarkan tidak mencakup keseluruhan *use case*, akan tetapi hanya menggambarkan fungsi utama dari aplikasi, yaitu verifikasi kehadiran, penilaian ujian skripsi, berita acara, dan rekap nilai.

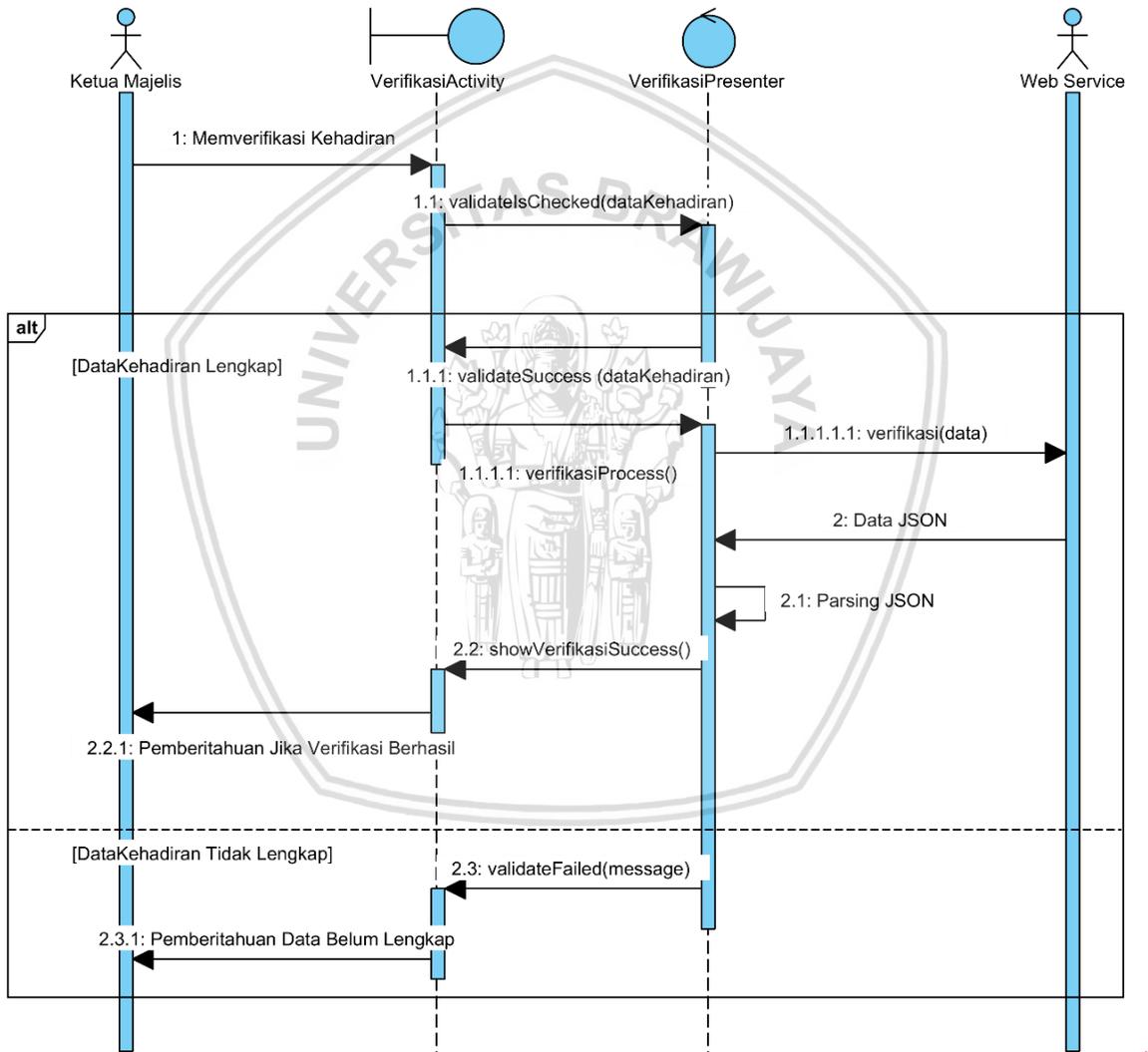
#### 5.1.3.1 Memverifikasi Kehadiran

Perancangan *sequence diagram* memverifikasi kehadiran merupakan menggambarkan interaksi antar objek yang terjadi pada saat aplikasi digunakan oleh aktor (ketua majelis) ketika memverifikasi kehadiran dari pembimbing, penguji dan mahasiswa. Sebagaimana yang ditunjukkan dalam Gambar 5.6, objek yang terlibat dalam interaksi untuk memverifikasi kehadiran diantaranya adalah ketua majelis dan web service sebagai aktor, VerifikasiActivity sebagai objek *boundary* dan VerifikasiPresenter sebagai objek *controller*.

Pertukaran pesan dimulai dari aktor (ketua majelis) memverifikasi kehadiran melalui objek VerifikasiActivity. Kemudian, pesan tersebut diteruskan kepada



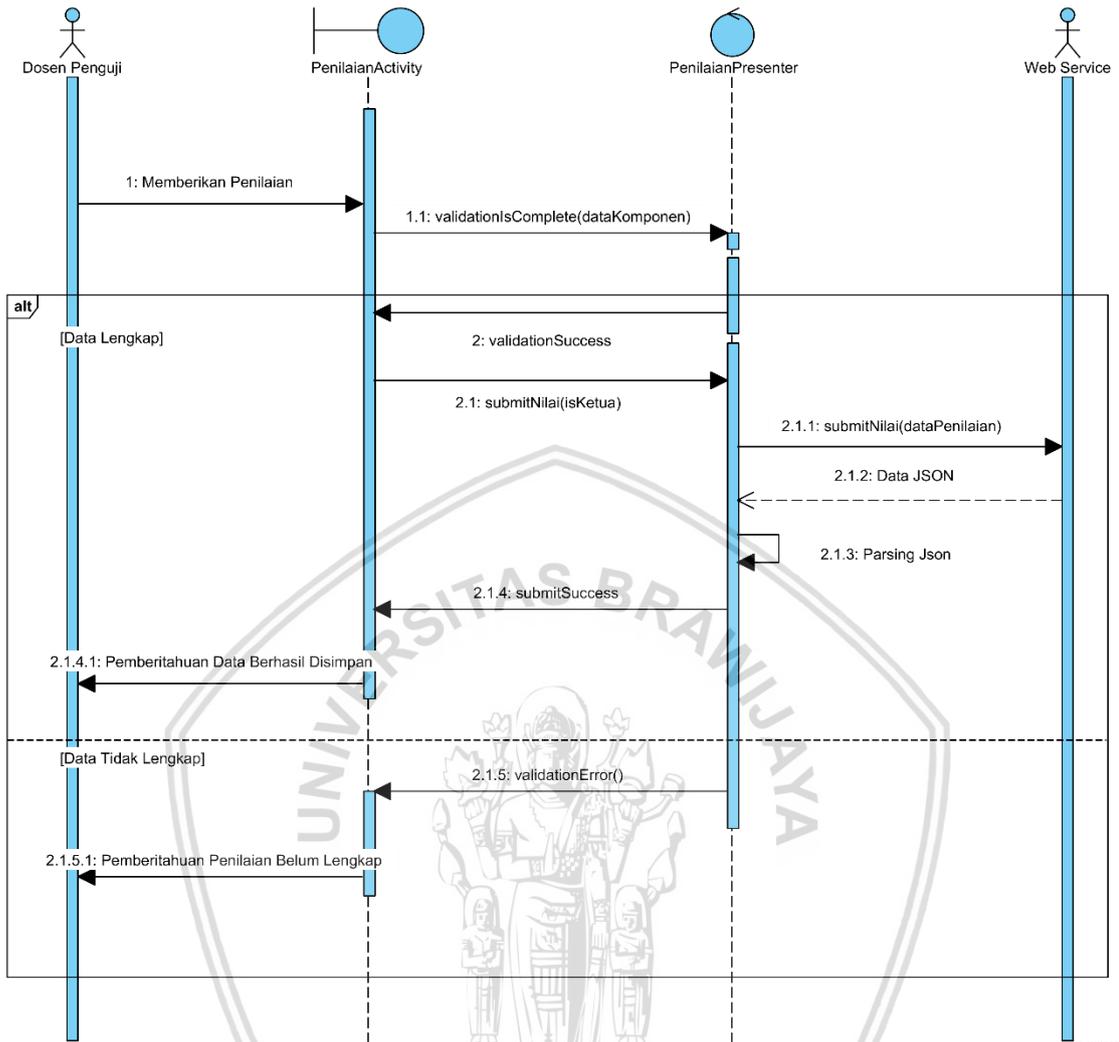
VerifikasiPresenter. Pada VerifikasiPresenter pesan tersebut dilakukan pengecekan. Apabila hasil pengecekan adalah tidak lengkapnya data kehadiran dari pesan yang dikirimkan oleh VerifikasiActivity maka VerifikasiPresenter akan mengirimkan pesan gagal kepada VerifikasiActivity dan VerifikasiActivity meneruskannya ke aktor. Akan tetapi jika sudah lengkap, maka VerifikasiPresenter mengirimkan pesan berhasil kepada VerifikasiActivity. Lalu, VerifikasiActivity mengirimkan pesan ke VerifikasiPresenter untuk diteruskan ke *Web Service*. *Web Service* akan mengirimkan pesan dalam bentuk JSON kepada VerifikasiPresenter, yang nantinya JSON tersebut di *parsing* oleh VerifikasiPresenter dan dikirim ke VerifikasiActivity. Pesan yang diterima VerifikasiActivity akan dikirim ke aktor (ketua majelis).



Gambar 5.6 Perancangan *Sequence Diagram* memverifikasi kehadiran



### 5.1.3.2 Memberikan Penilaian



**Gambar 5.7 Perancangan *Sequence Diagram* memberikan penilaian**

Perancangan *sequence diagram* memberikan penilaian merupakan menggambarkan interaksi antar objek yang terjadi pada saat aplikasi digunakan oleh aktor (dosen penguji) ketika memberikan penilaian melalui aplikasi. Sebagaimana yang ditunjukkan dalam Gambar 5.7, objek yang terlibat dalam interaksi untuk memberikan penilaian diantaranya adalah dosen penguji dan web service sebagai aktor, PenilaianActivity sebagai objek *boundary* dan PenilaianPresenter sebagai *objek controller*.

Pertukaran pesan dimulai dari aktor (dosen penguji) memberikan penilaian melalui PenilaianActivity. Kemudian, pesan tersebut diteruskan kepada PenilaianPresenter. Pada PenilaianPresenter pesan tersebut dilakukan pengecekan kelengkapan data penilaian. Apabila hasil pengecekan adalah tidak lengkapnya data penilaian dari pesan yang dikirimkan oleh PenilaianActivity maka PenilaianPresenter akan mengirimkan pesan gagal kepada PenilaianActivity dan PenilaianActivity meneruskannya ke aktor. Akan tetapi jika sudah lengkap, maka PenilaianPresenter mengirimkan pesan berhasil kepada PenilaianActivity. Lalu,

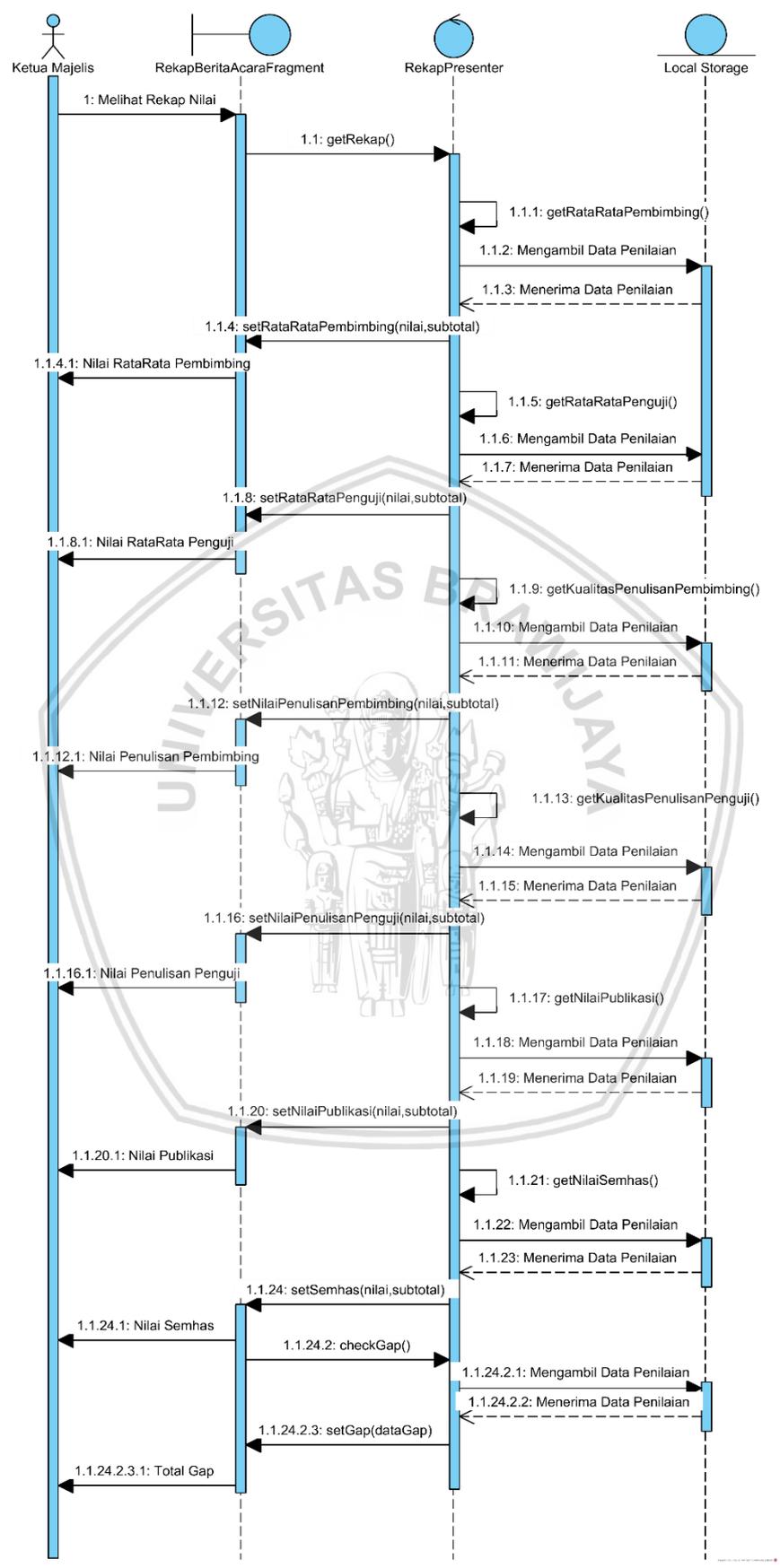


PenilaianActivity mengirimkan pesan ke PenilaianPresenter untuk diteruskan ke *Web Service*. *Web Service* akan mengirimkan pesan dalam bentuk JSON kepada PenilaianPresenter, yang nantinya JSON tersebut di *parsing* oleh PenilaianPresenter dan dikirim ke PenilaianActivity. Pesan yang diterima PenilaianActivity akan dikirim ke aktor (dosen penguji).

### 5.1.3.3 Melihat rekap Nilai

Perancangan *sequence diagram* memberikan penilain merupakan menggambarkan interaksi antar objek yang terjadi pada saat aplikasi digunakan oleh aktor (ketua majelis) ketika melihat rekap nilai melalui aplikasi. Sebagaimana yang ditunjukkan dalam Gambar 5.8, objek yang terlibat dalam interaksi untuk melihat rekap nilai diantaranya adalah ketua majelis sebagai aktor, RekapBeritaAcaraFragment sebagai objek *boundary*, RekapPresenter sebagai *objek controller* dan *Local Storage* sebagai objek *entity*.

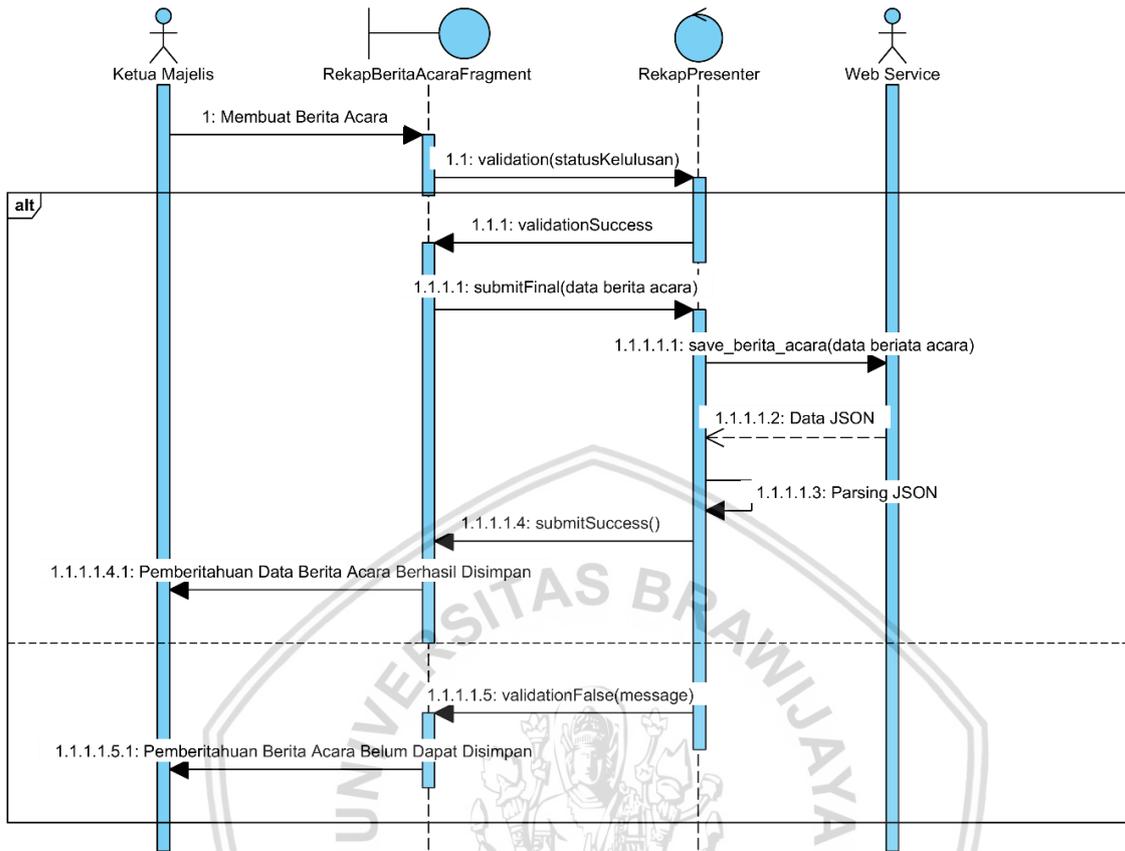
Pertukaran pesan dimulai dari ketua majelis melihat rekap nilai melalui RekapBeritaAcaraFragment. Kemudian, pesan tersebut diteruskan kepada RekapPresenter. RekapPresenter mengambil data dan melakukan perhitungan nilai. Perhitungan nilai yang selesai kemudian dikirim kepada RekapBeritaAcaraFragment dan diteruskan ke aktor. Setelah itu, RekapBeritaAcaraFragment mengirim pesan kembali ke RekapPresenter. RekapPresenter melakukan memeriksa gap nilai dengan meminta data penilaian dari local storage dan local storage mengirimkan data penilaian ke RekapPresenter. Pesan kemudian dilanjutkan ke RekapBeritaAcaraFragment dan dikirim kembali ke aktor.



Gambar 5.8 Perancangan Sequence Diagram rekap nilai



### 5.1.3.4 Membuat Berita Acara



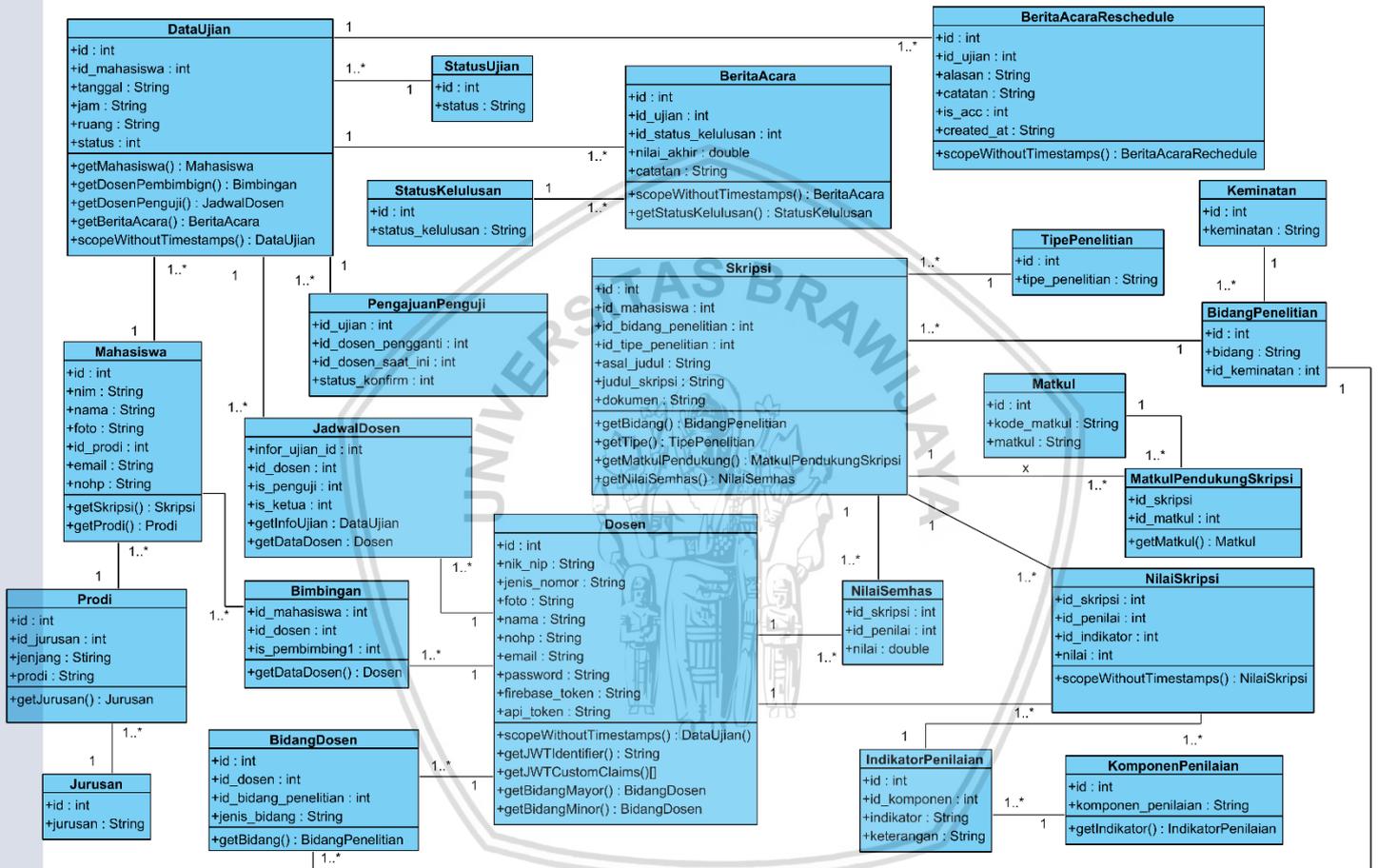
**Gambar 5.9 Perancangan *Sequence Diagram* membuat berita acara**

Perancangan *sequence diagram* memberikan penilain merupakan menggambarkan interaksi antar objek yang terjadi pada saat aplikasi digunakan oleh aktor (ketua majelis) ketika membuat berita acara melalui aplikasi. Sebagaimana yang ditunjukkan dalam Gambar 5.9, objek yang terlibat dalam interaksi untuk membuat berita acara diantaranya adalah ketua majelis dan *web service* sebagai aktor, *RekapBeritaAcaraFragment* sebagai objek *boundary*, dan *RekapPresenter* sebagai *objek controller*.

Pertukaran pesan dimulai dari aktor (dosen penguji) membuat beritaacara melalui *RekapBeritaAcaraFragment*. Kemudian, pesan tersebut diteruskan kepada *RekapPresenter*. Pada *RekapPresenter* pesan tersebut dilakukan pengecekan kelengkapan data berita acara. Apabila hasil pengecekan adalah tidak lengkapnya data berita acara dari pesan yang dikirimkan oleh *RekapBeritaAcaraFragment* maka *RekapPresenter* akan mengirimkan pesan gagal kepada *RekapBeritaAcaraFragment* dan diteruskannya ke aktor. Akan tetapi jika sudah lengkap, maka *PenilaianPresenter* mengirimkan pesan berhasil kepada *PenilaianActivity*. Lalu, *RekapBeritaAcaraFragment* mengirimkan pesan ke *RekapPresenter* untuk diteruskan ke *web service*. *Web service* akan mengirimkan pesan dalam bentuk JSON kepada *RekapPresenter*, yang nantinya JSON tersebut di *parsing* oleh *RekapPresenter* dan dikirim ke *RekapBeritaAcaraFragment*. Pesan yang diterima *RekapBeritaAcaraFragment* akan dikirim ke aktor (ketua majelis).

### 5.1.4 Perancangan Class Diagram

Subbab ini membahas perancangan dari *class diagram* aplikasi penilaian ujian skripsi berbasis android. Perancangan ini menggambarkan atribut dan operator dari setiap kelas serta hubungan setiap kelas dengan kelas lainnya. Perancangan *class diagram* mengikuti pola perancangan pengembangan aplikasi android, yaitu *Model-View-Presenter*. Sehingga perancangan ini terdiri dari *class diagram* model, presenter dan view sebagaimana ditunjukkan dalam Gambar 5.10, dan Gambar 5.11.



Gambar 5.10 Perancangan Class Diagram Model

Gambar 5.10 merupakan perancangan *class diagram* model yang terdapat pada *web service* dan memiliki hubungan dengan *database* utama dari aplikasi. Perancangan *class diagram* model menghasilkan 23 kelas model, yaitu DataUjian, BeritaAcara, BeritaAcaraReschedule, Bimbingan, JadwalDosen, Mahasiswa, Skripsi, Prodi, Jurusan, BidangPenelitian, Keminatan, TipePenelitian, NilaiSkripsi, Matkul, Dosen, PengajuanPenguji, NilaiSemhas, StatusKelulusan, IndikatorPenilaian, KomponenPenilaian, MatkulPendukungSkripsi, StatusUjian dan BidangDosen.

Gambar 5.11 merupakan perancangan *class diagram presenter* dan view yang terdapat pada aplikasi FILKOM APPS berbasis android. Perancangan *class diagram presenter* terdiri atas 10 *class*, yaitu RekapContract, RekapPresenter,

IndikatorContract, IndikatorPresenter, PenilaianContract, PenilaianPresenter, PenilaianNpContract, PenilaianNpPresenter, VerifikasiContract, dan VerifikasiPresenter. Pada *class* RekapContract, IndikatorContract, PenilaianContract, PenilaianNpContract, dan VerifikasiContract masing-masing memiliki *inner class* yaitu View dan Presenter. Kemudian, *class diagram view* terdiri atas 6 *class*, yaitu VerifikasiActivity, PenilaianActivity, PenilaianUjianActivity, StepFragment, PenilaianNaskahPublikasi, dan RekapBeritaAcaraFragment.





## 5.1.5 Perancangan Basis Data

Pada bagian membahas perancangan basis data yang digunakan dalam mengimplementasikan basis data aplikasi penilaian ujian skripsi. Perancangan basis data terbagi menjadi dua, yaitu perancangan tabel dan physical data model.

### 5.1.5.1 Perancangan Tabel

Pada bagian ini membahas table-table yang akan digunakan dalam mengimplementasikan database pada aplikasi penilaian ujian skripsi. Rancangan tabel dibuat berdasarkan diagram kelas model pada web service yang telah dibuat sebelumnya. Rancangan table terbagi menjadi dua, yaitu, rancangan tabel di web service dan rancangan tabel di android. Selain itu juga setiap nama tabel pada database menggunakan prefix “tbs”. Adapun rincian tabel sebagai berikut:

#### a) Rancangan Tabel Di Web Service

##### 1. Tabel Dosen

Nama tabel: tbs\_dosen

Nama kelas model: Dosen

Fungsi: Menyimpan data dosen

**Tabel 5.1 Rancangan tabel basis data dosen**

No	Nama Kolom	Tipe Data	Panjang
1	id	integer	11
2	nik_nip	varchar	25
3	jenis_nomor	Enum	'Nik', 'Nip'
4	Foto	varchar	150
5	Nama	varchar	150
6	Nohp	varchar	15
7	Email	varchar	150
8	Password	varchar	60
9	firebase_token	varchar	200
10	api_token	varchar	400

##### 2. Tabel Mahasiswa

Nama tabel: tbs\_mahasiswa

Nama kelas model : Mahasiswa

Fungsi : Menyimpan data mahasiswa

**Tabel 5.2 Rancangan tabel basis data mahasiswa**

No	Kolom	Tipe Data	Panjang
1	id	Varchar	30
2	Nim	Varchar	20
3	Nama	Varchar	150
4	Foto	varchar	150
5	id_prodi	Integer	11
6	Email	Varchar	150
7	Nohp	Varchar	15

**3. Tabel Skripsi**

Nama tabel: tbs\_skripsi

Nama kelas model : Skripsi

Fungsi: Menyimpan data skripsi

**Tabel 5.3 Rancangan tabel basis data skripsi**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	id_mahasiswa	Integer	11
3	judul_skripsi	Varchar	800
4	id_bidang_penelitian	Integer	11
5	id_tipe_penelitian	Integer	11
6	asal_judul	Integer	11
7	Dokumen	Varchar	100

**4. Tabel Bidang Penelitian**

Nama tabel: tbs\_bidang\_penelitian

Nama kelas model : BidangPenelitian

Fungsi: Menyimpan data bidang penelitian, seperti, pengembangan sistem informasi.

**Tabel 5.4 Rancangan tabel basis data bidang penelitian**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	Bidang	Varchar	150
3	id_keminatan	Integer	11

**5. Tabel Tipe Penelitian**

Nama tabel: tbs\_tipe\_penelitian

Nama kelas model: TipePenelitian

Fungsi: Menyimpan data tipe penelitian, seperti, Implementatif-Pengembangan.

**Tabel 5.5 Rancangan tabel basis data tipe penelitian**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	tipe_penelitian	Varchar	50

**6. Tabel Keminatan**

Nama tabel: tbs\_keminatan

Nama kelas: Keminatan

Fungsi: Menyimpan data keminatan yang terdapat di jurusan.

**Tabel 5.6 Rancangan tabel basis data keminatan**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	Keminatan	Varchar	50

**7. Tabel Mata Kuliah Pendukung Skripsi**

Nama tabel: tbs\_matkul\_pendukung\_skripsi

Nama kelas model: MatkulPendukungSkripsi

Fungsi: Menyimpan data mata kuliah yang mendukung skripsi

**Tabel 5.7 Rancangan tabel basis data mata kuliah pendukung skripsi**

No	Kolom	Tipe Data	Panjang
1	id_skripsi	Integer	11
2	id_matkul	Integer	11

**8. Tabel Mata Kuliah**

Nama tabel: tbs\_matkul

Nama kelas model: Matkul

Fungsi: Menyimpan data mata kuliah yang dimiliki oleh Fakultas Ilmu Komputer

**Tabel 5.8 Rancangan tabel basis data mata kuliah**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	kode_matkul	Varchar	10
3	Matkul	Varchar	70

**9. Tabel Jurusan**

Nama tabel: tbs\_jurusan

Nama kelas model: Jurusan

Fungsi: Menyimpan data jurusan

**Tabel 5.9 Rancangan tabel basis data jurusan**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	Jurusan	Varchar	50

**10. Tabel Program Studi**

Nama tabel: tbs\_prodi

Nama kelas model : Prodi

Fungsi: Menyimpan Data Program Studi

**Tabel 5.10 Rancangan tabel basis data program studi**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	id_jurusan	Integer	11
3	Jenjang	Enum	'S1'
4	Prodi	Varchar	50

**11. Tabel Bidang Dosen**

Nama tabel: tbs\_bidang\_dosen

Nama kelas model : BidangDosen

Fungsi: Menyimpan data bidang dosen

**Tabel 5.11 Rancangan tabel basis data bidang dosen**

No	Kolom	Tipe Data	Panjang
1	id_dosen	Integer	11
2	id_bidang_penelitian	Integer	11
3	jenis_bidang	Enum	'Mayor','Minor'

**12. Tabel Bimbingan Dosen**

Nama tabel: tbs\_bimbingan\_dosen

Nama kelas model : Bimbingan

Fungsi: Menyimpan data bimbingan skripsi dari setiap dosen

**Tabel 5.12 Rancangan tabel basis data bimbingan dosen**

No	Kolom	Tipe Data	Panjang
1	id_dosen	Integer	11
2	id_mahasiswa	Integer	11
3	is_pembimbing1	tinyint	1

**13. Tabel Nilai Skripsi**

Nama tabel: tbs\_nilai\_skripsi

Nama kelas model: NilaiUjian

Fungsi: Menyimpan data nilai skripsi berdasarkan komponen penilaian ujian

**Tabel 5.13 Rancangan tabel basis data nilai skripsi**

No	Kolom	Tipe Data	Panjang
1	id_skripsi	Integer	11
2	id_penilai	Integer	11
3	id_indikator	Integer	11
4	Nilai	integer	11

**14. Tabel Nilai Semhas**

Nama tabel: tbs\_nilai\_semhas

Nama kelas model: NilaiSemhas

Fungsi : Menyimpan data nilai semhas

**Tabel 5.14 Rancangan tabel basis data nilai semhas**

No	Kolom	Tipe Data	Panjang
1	id_skripsi	Integer	11
2	id_penilai	Integer	11
3	Nilai	Integer	double

**15. Tabel Info Ujian Skripsi**

Nama tabel : tbs\_info\_ujian\_skripsi

Nama kelas model: DataUjian

Fungsi : Menyimpan informasi ujian skripsi

**Tabel 5.15 Rancangan tabel basis data info ujian skripsi**

No	Kolom	Tipe Data	Panjang
1	id	Integer	11
2	id_mahasiswa	Integer	11
3	tanggal	Date	-
4	jam	Time	-
5	ruang	Varchar	50
6	status	Integer	1

**16. Tabel Jadwal Dosen**

Nama tabel: tbs\_jadwal\_dosen

Nama kelas model: JadwalDosen

Fungsi: Menyimpan informasi jadwal ujian skripsi setiap dosen

**Tabel 5.16 Rancangan tabel basis data jadwal dosen**

No	Kolom	Tipe Data	Panjang
1	info_ujian_id	Integer	11
2	id_dosen	Integer	11
3	is_penguji	Char	1
4	is_ketua	Tinyint	1

**17. Tabel Berita Acara Reschedule**

Nama tabel: tbs\_berita\_acara\_reschedule

Nama kelas model: BeritaAcaraReschedule

Fungsi : Menyimpan berita acara apabila jadwal ujian perlu di jadwalkan kembali

**Tabel 5.17 Rancangan tabel basis data berita acara reschedule**

No	Kolom	Tipe Data	Panjang
1	id	Integer	11
2	id_ujian	Integer	11
3	alasan	Varchar	100
4	catatan	Varchar	500
5	created_at	Timetamp	-

**18. Tabel Status Kelulusan**

Nama tabel: tbs\_status\_kelulusan

Nama kelas model: StatusKelulusan

Fungsi: Menyimpan status kelulusan

**Tabel 5.18 Rancangan tabel basis data status kelulusan**

No	Kolom	Tipe Data	Panjang
1	id	Integer	11
2	status_kelulusan	Varchar	50

**19. Tabel Komponen Penilaian**

Nama tabel: tbs\_komponen\_penilaian

Nama kelas model: KomponenPenilaian

Fungsi: Menyimpan komponen penilaian

**Tabel 5.19 Rancangan tabel basis data komponen penilaian**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	komponen_penilaian	Varchar	100

**20. Tabel Indikator Penilaian**

Nama tabel : tbs\_indikator\_penilaian

Nama kelas model: IndikatorPenilaian

Fungsi : Menyimpan Indikator setiap komponen penilaian

**Tabel 5.20 Rancangan tabel basis data indikator penilaian**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	id_komponen	Integer	11
3	indikator	Varchar	150
4	keterangan	Varchar	300

**21. Tabel Pengajuan Penguji Pengganti**

Nama tabel: tbs\_pengajuan\_penguji\_pengganti

Nama model: PengajuanPenguji

Fungsi: menyimpan data pengajuan penguji pengganti

**Tabel 5.21 Rancangan tabel basis data pengajuan penguji pengganti**

No	Kolom	Tipe Data	Panjang
1	id_ujian	Integer	11
2	id_dosen_pengganti	Integer	11
3	id_dosen_saat_ini	Integer	11
4	status_confirm	Integer	1

## 22. Tabel Status Ujian

Nama tabel: tbs\_status\_ujian

Nama model: StatusUjian

Fungsi: menyimpan data status ujian

**Tabel 5.22 Rancangan tabel basis data status ujian**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	status	Varchar	30

## 23. Tabel Berita Acara

Nama tabel: tbs\_berita\_acara

Nama kelas model: BeritaAcara

Fungsi : Menyimpan data berita acara ujian skripsi

**Tabel 5.23 Rancangan tabel basis data berita acara**

No	Kolom	Tipe Data	Panjang
1	Id	Integer	11
2	id_ujian	Integer	11
3	id_status_kelulusan	Integer	11
4	nilai_akhir	Double	-
5	catatan	Varchar	250

### 5.1.5.2 Physical Data Model

Gambar 5.12 merupakan physical data model dari database aplikasi penilaian ujian skripsi. Physical data model dibuat berdasarkan rancangan table dan kelas diagram model yang telah dibuat sebelumnya. Masing-masing kelas pada kelas diagram model di representasikan dengan tabel yang terdapat pada physical data model. Kemudian, atribut yang dimiliki oleh setiap kelas di representasikan oleh kolom pada tabel. Hubungan antara tabel berdasarkan hubungan yang terdapat di kelas diagram model.

Pada rancangan basis data terdapat 23 tabel. Tabel tbs\_jadwal\_ujian, tbs\_status\_ujian, tbs\_info\_ujian\_skripsi, tbs\_mahasiswa digunakan untuk menyimpan data jadwal ujian. Tabel tbs\_skripsi, tbs\_tipe\_penelitian dan tbs\_bidang\_penelitian, tbs\_keminatan, tbs\_matkul, tbs\_matkul\_pendukung, dan tbs\_bimbingan\_dosen digunakan untuk menyimpan data skripsi mahasiswa. Tabel tbs\_prodi, tbs\_jurusan dan tbs\_mahasiswa digunakan untuk menyimpan data



## 5.1.6 Perancangan API

API merupakan metode yang digunakan dalam melakukan pertukaran data dari aplikasi ke basis data. Bagian ini merupakan perancangan API yang menjelaskan endpoint, autentikasi, parameter dan method yang digunakan aplikasi dalam melakukan pertukaran data. Selain itu juga di perancangan ini menjelaskan *response* yang akan diterima oleh aplikasi. Perancangan ini hanya menggambarkan API yang akan digunakan pada fungsional utama. Akan tetapi tidak semua fungsional utama menggunakan API. Adapun Fungsional utama yang menggunakan API, yaitu memberikan penilaian, memverifikasi kehadiran dan membuat berita acara.

### 5.1.6.1 Memverifikasi Kehadiran

Tabel 5.24 Perancangan API memverifikasi kehadiran

Nama <i>Endpoint</i>	ujian/verifikasi
Autentikasi	JSON Web Token (JWT)
Parameter	<pre>{   "id_ujian": "1",   "update_majelis": "0",   "dosen": [     {       "id_dosen": "3",       "is_hadir": "1"     },     {       "id_dosen": "8",       "is_hadir": "1"     },     {       "id_dosen": "6",       "is_hadir": "1"     }   ] }</pre>
Tipe Parameter	RAW
<i>Method</i>	POST
<i>Response</i>	<pre>{   "success": "true",   "message": "verification successfull",   "data": [] }</pre>

Tabel 5.24 merupakan perancangan dari API yang akan digunakan di fungsional memverifikasi kehadiran. API ini diakses melalui *endpoint* ujian/verifikasi dengan menggunakan autentikasi JSON Web Token. Fungsional memverifikasi kehadiran memerlukan parameter dengan tipe RAW dan menggunakan *method* post. Parameter yang dikirim yaitu, *id\_ujian*, *update\_majelis* dan *dosen*. Parameter *dosen* merupakan *array of object* yang terdiri dari beberapa objek. Setiap *index*-nya menyimpan objek yang terdiri dari

id\_dosen, dan is\_hadir. Kemudian *response* yang diterima aplikasi berupa data JSON yang terdiri dari *success*, *message* dan data.

### 5.1.6.2 Memberikan Penilaian

Tabel 5.25 Perancangan API memberikan penilaian

Nama Endpoint	ujian/save_nilai
Authentikasi	JSON Web Token (JWT)
Parameter	<pre>{   "id_ujian": "1",   "id_skripsi": "1",   "nilai_ujian": [     { "id_penilai": "1", "id_indikator": "1", "id_komponen": "1", "nilai": 7 },     { "id_penilai": "1", "id_indikator": "2", "id_komponen": "1", "nilai": 7 },     { "id_penilai": "1", "id_indikator": "3", "id_komponen": "1", "nilai": 7 },     { "id_penilai": "1", "id_indikator": "4", "id_komponen": "1", "nilai": 7 },     { "id_penilai": "1", "id_indikator": "5", "id_komponen": "2", "nilai": 7 },     { "id_penilai": "1", "id_indikator": "6", "id_komponen": "2", "nilai": 7 }   ] }</pre>
Tipe Parameter	RAW
Method	POST
Response	<pre>{   "success": "true",   "message": "sending successfull",   "data": [] }</pre>

Tabel 5.25 merupakan perancangan dari API yang akan digunakan di fungsional memberikan penilaian. API ini diakses melalui *endpoint* ujian/save\_nilai dengan menggunakan autentikasi JSON Web Token. Fungsional memberikan penilaian memerlukan parameter dengan tipe RAW dan menggunakan *method* post. Parameter yang dikirim yaitu, id\_ujian, id\_skripsi dan nilai ujian. Parameter nilai ujian merupakan *array of object*. Setiap *index*-nya menyimpan objek yang terdiri dari id\_penilai, id\_indikator, id\_komponen dan nilai. Kemudian *response* yang diterima aplikasi berupa data JSON yang terdiri dari *success*, *message* dan data.

### 5.1.6.3 Membuat Berita Acara

Tabel 5.26 Perancangan API membuat berita acara

Nama <i>Endpoint</i>	ujian/save_berita_acara
Authentikasi	JSON Web Token (JWT)
Parameter	{ " id_ujian": "1", " id_status_kelulusan": "2", " nilai_akhir": "85", " catatan": "" }
Tipe Parameter	RAW
<i>Method</i>	POST
<i>Response</i>	{ " success": "true", " message": "berita acara berhasil disimpan", " data": [] }

Tabel 5.26 merupakan perancangan dari API yang akan digunakan di fungsional memberikan penilaian. API ini diakses melalui *endpoint* ujian/save\_berita\_acara dengan menggunakan autentikasi JSON Web Token. Fungsional memberikan penilaian memerlukan parameter dengan tipe RAW dan menggunakan *method* post. Parameter yang dikirim yaitu, id\_ujian, id\_status\_kelulusan, nilai\_akhir dan catatan. Kemudian *response* yang diterima aplikasi berupa data JSON yang terdiri dari *success*, *message* dan *data*.

### 5.1.7 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan bagian yang menjelaskan rancangan antarmuka atau *user interface* dari aplikasi yang akan di implementasikan. Perancangan ini dibuat berdasarkan *web* FILKOM APP dan mengacu pada *Google Material Design Guideline* yang dibuat oleh Google untuk menjadi panduan untuk merancang *user interface*.

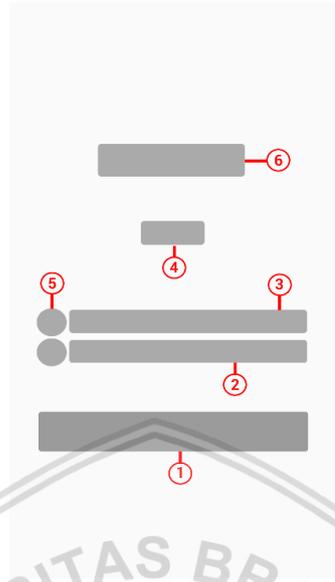
#### 5.1.7.1 Wireframe

Wireframe merupakan sketsa awal yang digunakan sebagai acuan dalam membuat *user interface* pada aplikasi penilaian ujian skripsi. Pada bagian ini akan membahas halaman-halaman yang terkait dengan proses penilaian ujian skripsi. Halaman yang akan dibahas, yaitu halaman login, halaman beranda, halaman rincian ujian, halaman verifikasi, halaman penilaian, dan halaman rekap nilai & berita acara.

#### 1. Halaman *Login*

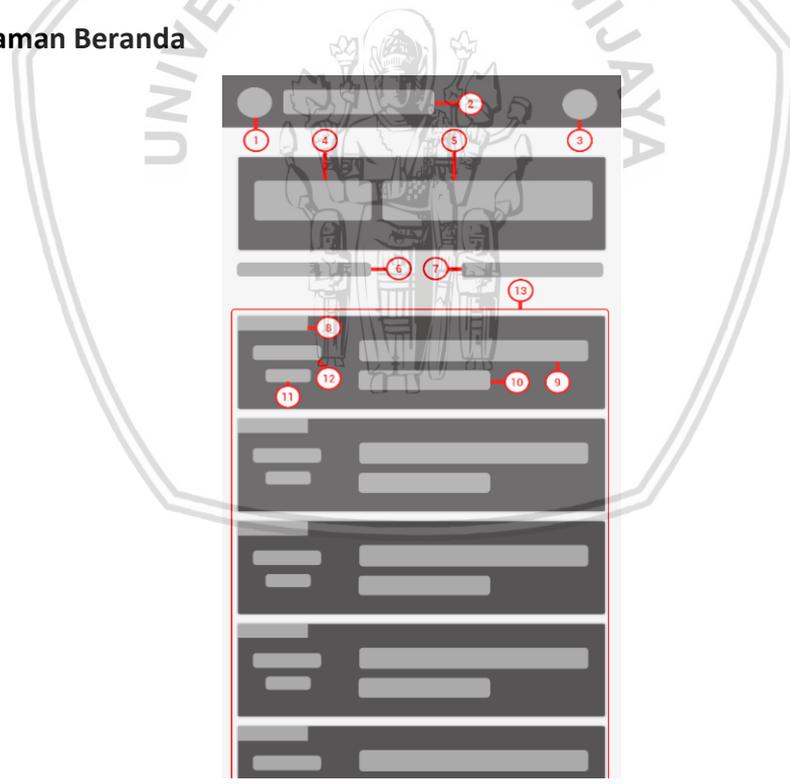
Halaman *login* merupakan halaman yang berfungsi untuk proses otentikasi identitas pengguna sebelum pengguna menggunakan mengakses informasi melalui aplikasi. Sebagaimana yang ditunjukkan dalam Gambar 5.13, halaman ini terdiri dari tombol untuk menjalankan proses otentikasi identifikasi (1), kolom untuk memasukkan password (2), kolom untuk memasukkan *username*, *icon* pada

username dan password (5), teks label yang bertuliskan *sigi in* yang menunjukkan bahwa halaman tersebut merupakan halaman *login*(4) dan logo aplikasi (6).



Gambar 5.13 Wireframe halaman login

## 2. Halaman Beranda



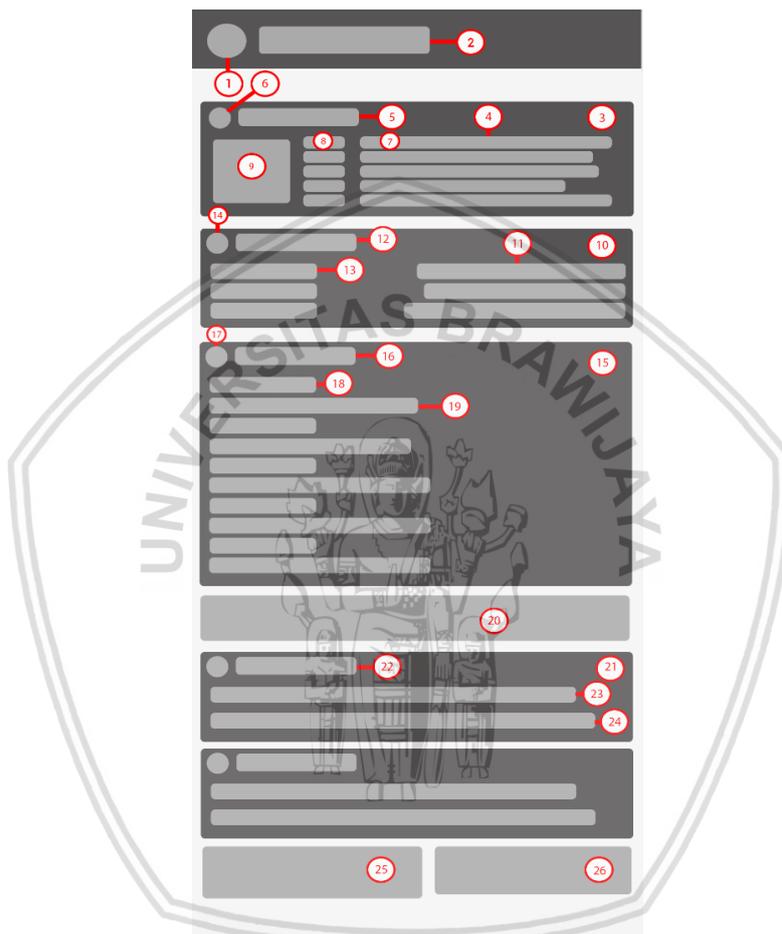
Gambar 5.14 Wireframe halaman beranda

Halaman beranda merupakan halaman utama dari aplikasi atau halaman setelah *user* melakukan login. Halaman ini menampilkan informasi daftar jadwal ujian. Sebagaimana ditunjukkan dalam Gambar 5.14, halaman ini terdiri dari *navigation drawer* (1) yang digunakan mengarahkan ke beberapa menu yang ada



pada *navigation drawer*, nama halaman (2), menu notifikasi (3), logo filkom (4), *text* fakultas ilmu computer (5), teks jadwal ujian hari ini (6), daftar jadwal ujian (12). Pada daftar ujian terdiri dari teks yang berisi informasi peran dosen (penguji, pembimbing atau ketua majelis), teks yang berisi informasi jam ujian (11), teks tanggal ujian (9), teks yang berisi informasi judul skripsi, dan teks yang berisi informasi nama mahasiswa (10).

### 3. Halaman Rincian Jadwal Ujian



**Gambar 5.15 Wireframe** halaman rincian jadwal ujian

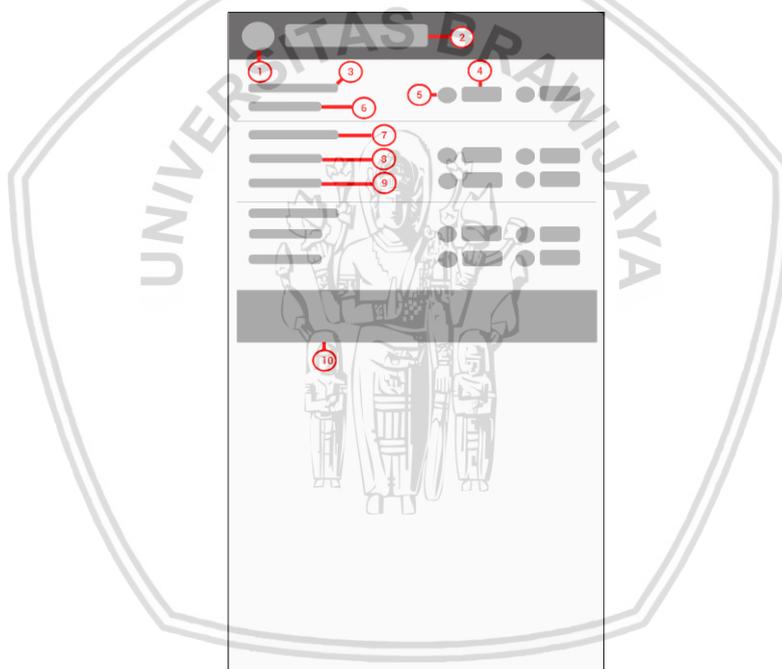
Halaman detail jadwal ujian merupakan halaman yang berfungsi untuk menampilkan informasi rincian dari jadwal ujian. Sebagaimana yang ditunjukkan dalam Gambar 5.15, halaman ini terdiri dari *icon back* untuk kembali ke halaman sebelumnya (1, teks nama halaman (2), kolom informasi mahasiswa (3), kolom informasi jadwal pelaksanaan ujian skripsi (10), kolom informasi skripsi (15), tombol untuk melihat dokumen (20), kolom informasi pembimbing (21), tombol untuk melakukan verifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji (25) dan tombol untuk melakukan penilaian (26).

Pada kolom informasi mahasiswa terdiri dari, *icon* kolom (6), teks nama kolom, teks yang berisi informasi nama mahasiswa, jenjang, email, jurusan dan nomor *handphone* mahasiswa (7), teks label dari setiap informasi mahasiswa, dan

foto mahasiswa (9). Pada kolom informasi jadwal pelaksanaan ujian skripsi terdiri dari *icon* kolom (14), teks nama kolom (12), teks label setiap informasi jadwal pelaksanaan (13), teks informasi jadwal pelaksanaan (11). Informasi yang ditampilkan pada kolom jadwal pelaksanaan ujian skripsi terdiri dari tanggal, waktu dan ruangan.

Pada kolom informasi skripsi menampilkan informasi skripsi, yaitu judul skripsi, bidang skripsi, tipe penelitian, asal judul dan mata kuliah pendukung. Kolom informasi terdiri dari *icon* kolom (17), teks nama kolom (16), teks label setiap informasi skripsi (18) dan teks setiap informasi skripsi (19). Pada kolom informasi dosen pembimbing menampilkan informasi nama dosen pembimbing. Bagian ini terdiri dari *icon* kolom dan teks nama kolom (22) dan teks nama dosen (24). Kemudian pada bagian bawah dari kolom informasi dosen pembimbing merupakan informasi penguji. Tampilan dari kolom informasi penguji sama seperti kolom informasi pembimbing.

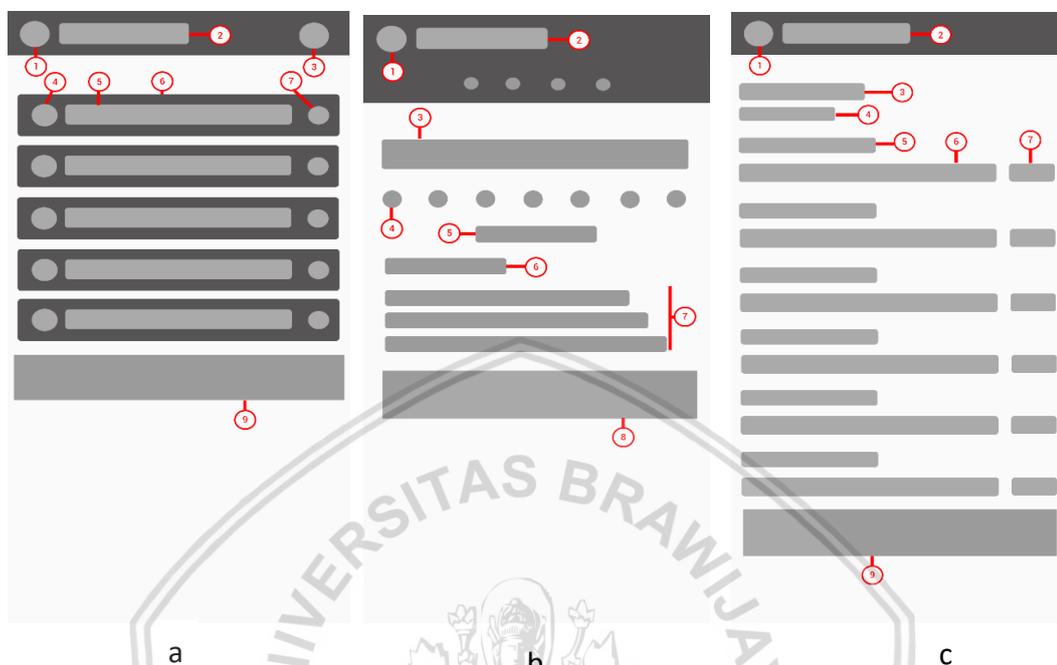
#### 4. Halaman Verifikasi Kehadiran



**Gambar 5.16 Wireframe halaman verifikasi kehadiran**

Halaman verifikasi kehadiran merupakan halaman yang berfungsi melakukan verifikasi kehadiran mahasiswa, pembimbing dan penguji. Sebagaimana yang ditunjukkan dalam Gambar 5.16, halaman ini terdiri atas *icon back* untuk kembali ke halaman sebelumnya (1), teks nama halaman (2), teks label mahasiswa, teks nama mahasiswa (7), radio button kehadiran dengan teks hadir atau tidak (4 dan 5), teks label pembimbing (7), teks nama dosen (8 dan 9). Pada bagian dosen penguji juga memiliki tampilan yang sama seperti bagian dosen pembimbing.

## 5. Halaman Penilaian



**Gambar 5.17 Wireframe halaman penilaian: (a) halaman utama penilaian (b)halaman indikator; (c) halaman penilaian naskah publikasi**

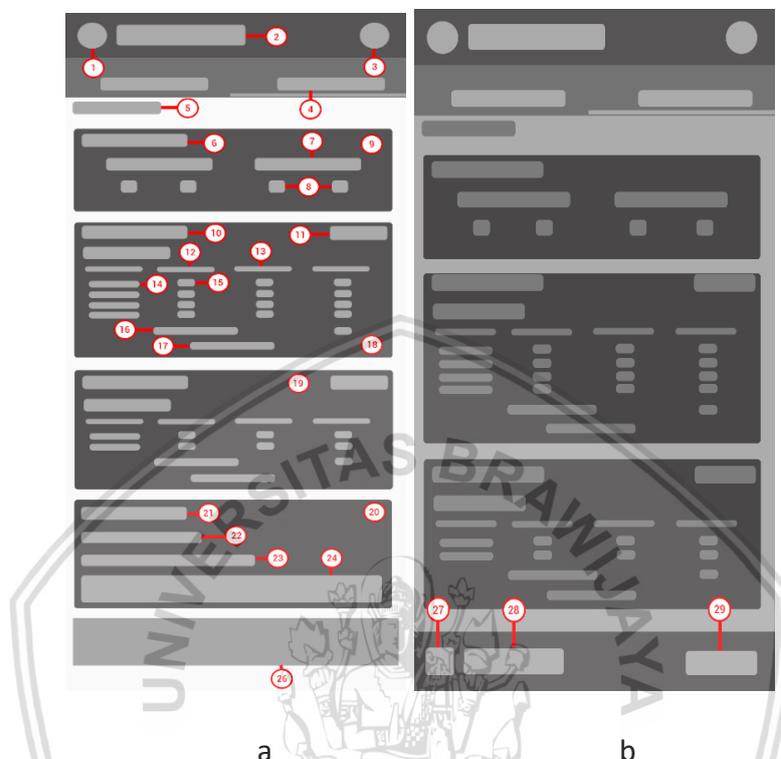
Halaman penilaian merupakan halaman dimana dosen penguji, dosen pembimbing memberikan penilaian skripsi. Pada halaman ini terdiri dari 3 tampilan. Tampilan pertama yaitu tampilan utama dari halaman penilaian, tampilan kedua dan ketiga merupakan tampilan dalam menilai setiap komponen penilaian ujian skripsi. Sebagaimana yang ditunjukkan dalam Gambar 5.17 (a) terdiri atas *icon back* untuk kembali ke halaman sebelumnya(1), teks nama halaman(2), tombol untuk memproses penilaian (9), dan daftar komponen penilaian (6). Pada daftar komponen terdiri atas *icon* yang digunakan untuk memberitahu jika komponen penilaian sudah dilengkapi (4), teks nama komponen penilaian (5) dan *icon* panah merepresentasikan bahwa dalam melakukan penilaian harus menekan salah satu daftar komponen penilaian (7).

Tampilan kedua merupakan tampilan dari setiap komponen ketika pengguna menekan salah satu daftar komponen kecuali pada komponen naskah publikasi. Sebagaimana yang ditunjukkan Gambar 5.17 (b), tampilan kedua dari halaman penilaian terdiri atas *icon back* untuk kembali ke halaman utama (1), teks nama komponen (2), teks nama indikator (3), radio button nilai 1-7 (4), teks kategori nilai (5), teks label keterangan nilai (6), teks informasi keterangan nilai (7) dan tombol untuk menyimpan penilaian (8).

Tampilan ketiga merupakan tampilan dari penilaian komponen naskah publikasi. Sebagaimana yang ditunjukkan Gambar 5.13 (c), terdiri atas atas *icon back* untuk kembali ke halaman utama (1), teks penilaian naskah publikasi (2), teks

naskah publikasi (3), *teks* nilai 0-100 (4), *teks* nama indikator (5), *teks* keterangan indikator (6), kolom untuk memasukan penialain (7) dan tombol untuk menyimpan penilaian (8).

## 6. Halaman Rekap Nilai Dan Berita Acara



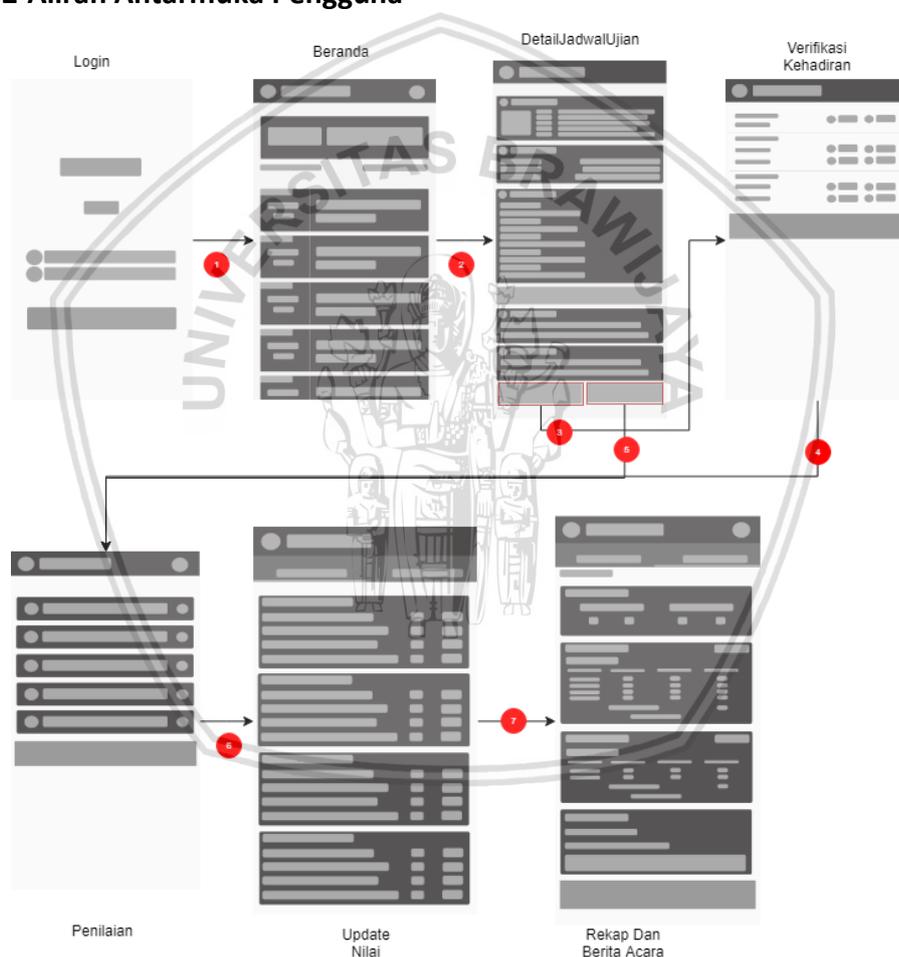
**Gambar 5.18 Wireframe** halaman rekap nilai dan berita acara: (a) halaman rekap nilai dan berita acara; (b) kolom informasi *gap* nilai.

Halaman rekap nilai merupakan halaman yang bertujuan untuk menampilkan hasil perhitungan nilai dari semua aspek penilaian skripsi serta memasukan berita acara ujian skripsi. Halaman ini tergabung menjadi satu halaman dengan halaman *update* nilai dan terdapat *tab menu* (4) sebagai pemisah antar halaman. Sebagaimana yang ditunjukkan dalam Gambar 5.18, halaman ini terdiri dari *icon back* untuk kembali ke halaman sebelumnya (1), *text* nama halaman (2), menu notifikasi (2), *text* yang menunjukkan keterangan waktu terakrit perubahan data, kolom informasi *status* proses penilaian setiap penilai yang berfungsi untuk mengetahui siapa saja yang sudah menyelesaikan penilaian atau yang sedang dalam proses penialain (9), kolom informasi nilai skripsi (18), kolom informasi nilai kualitas penulisan skripsi (19), kolom memasukan beriat acara (20) dan tombol untuk memproses berita acara (26).

Pada kolom informasi status proses penilaian (9) dan kolom informasi nilai kualitas penulisan skripsi terdiri atas *teks* nama kolom (6), *teks* peran pembimbing dan penguji (7) dan status proses penilaian yang di gambarkan menggunakan *checkbox* (8). Kemudian, pada kolom informasu nilai skripsi terdiri atas *teks* nama kolom (10), *teks* nilai akhir ujian skripsi yang ditampilkan dalam bentuk angka dan huruf (11), *teks* label penilaian yang digunakan untuk menandakan bagian-bagaian

penilaian seperti keterangan, rata-rata, bobot dan sub total (12 dan 13), teks label terkait nama penilaian seperti seminar hasil, publikasi, ujian skripsi pembimbing, ujian skripsi penguji (14), teks yang digunakan untuk menampilkan nilai (15), teks nilai akhir dalam bentuk angka (16), teks selengkapannya yang berfungsi sebagai tombol untuk melihat rincian nilai (17). Kolom beritanya kolom berita acara yang terdiri dari teks label dan teks informasi nilai akhir skripsi (21), *dropdown* status kelulusan untuk memilih status kelulusan mahasiswa (23) dan kolom untuk memasukan catatan khusus (24). Selain itu, halaman ini juga menampilkan informasi *gap* nilai sebagaimana ditunjukkan dalam Gambar 5.19 (b). Kolom tersebut terdiri atas *icon gap* (28), *text total gap* (29), dan *text* selengkapannya yang digunakan sebagai tombol untuk melihat rincian *gap* nilai.

### 5.1.7.2 Aliran Antarmuka Pengguna



**Gambar 5.19 Aliran antarmuka pengguna aplikasi**

Subab ini membahas aliran antarmuka pengguna aplikasi penilaian ujian skripsi. Aliran antarmuka ini hanya membahas pada aliran antarmuka dalam proses penilaian ujian skripsi yang dimulai verifikasi kehadiran sampai dengan pengguna memasukan data berita acara.. Aliran antarmuka dimulai dari halaman login. Ketika pengguna telah melakukan login aplikasi akan menampilkan halaman beranda. Pada halaman ini pengguna akan memilih jadwal ujian.

Kemudian aplikasi akan menampilkan rincian jadwal ujian yang pengguna telah pilih. Pengguna terlebih dahulu memverifikasi kehadiran majelis dan aplikasi akan menampilkan halaman verifikasi. Setelah verifikasi telah dilakukan, alur terbagi dua sebagaimana yang ditunjukkan Gambar 5.19 pada nomor 4 dan 5. Alur nomor 4 merupakan alur dimana pengguna dapat langsung menilai tanpa harus terlebih dahulu berada di halaman rincian ujian. Pada nomor 5 pengguna dapat melakukan penilaian pada halaman rincian ujian. Kondisi pada alur ini terjadi, apabila pengguna menginginkan terlebih dahulu melihat informasi ujian skripsi.

Alur selanjutnya yaitu halaman penilaian dan dilanjutkan ke halaman *update* nilai. Selanjutnya aplikasi akan menampilkan halaman rekap nilai dan berita acara. Pengguna masih dapat bisa kembali ke halaman *update* nilai ketika sudah berada di rekap nilai dan berita acara. Hal memungkinkan pengguna merubah nilai ketika terdapat GAP nilai. Alur nomor 7 merupakan alur antarmuka terakhir dalam proses penilaian, yang kemudian alur akan kembali ke halaman beranda.

### 5.1.8 Perancangan Algoritme

Pada bagian ini menjelaskan perancangan algoritme. Perancangan algoritme merupakan rancangan berupa prosedur sistematis yang berisi sekumpulan operasi logika sebagai konsep pembangunan fungsi-fungsi dari aplikasi yang akan dikembangkan. Perancangan algoritme ini dirancang dengan menggunakan *pseudocode*. Algoritme yang sudah dirancang akan menjadi panduan dalam tahap implementasi.

#### 5.1.8.1 Algoritme Memverifikasi Kehadiran Mahasiswa, Pembimbing Dan Penguji

Subbab ini menjelaskan rancangan algoritme dari fungsional yang disediakan oleh aplikasi untuk melakukan verifikasi kehadiran mahasiswa, pembimbing, dan penguji. Rancangan algoritme yang ditunjukkan pada Tabel 5.27 merupakan perancangan algoritme proses verifikasi kehadiran mahasiswa, pembimbing, dan penguji.

**Tabel 5.27 Perancangan Algoritme memverifikasi kehadiran mahasiswa, pembimbing dan penguji**

No	Pseudocode
1	Mulai
2	Set variabel list kehadiran pembimbing
3	Set variabel list kehadiran penguji
4	Set variabel kehadiran mahasiswa
5	IF kehadiran mahasiswa sama dengan hadir maka
6	Set variabel kehadiran mahasiswa sama dengan 1
7	ELSE
8	Set variabel kehadiran mahasiswa sama dengan 0
9	END IF
10	
11	IF kehadiran pembimbing sama dengan hadir maka
12	Tambahkan kehadiran pembimbing sama dengan 1 ke dalam
13	variabel list kehadiran pembimbing
14	ELSE



**Tabel 5.27 Perancangan algoritme memverifikasi kehadiran mahasiswa, pembimbing dan penguji (Lanjutan)**

15	Tambahkan kehadiran pembimbing sama dengan 0 ke dalam variabel list kehadiran pembimbing
16	
17	END IF
18	IF kehadiran penguji sama dengan hadir maka
19	Tambahkan kehadiran penguji sama dengan 1 ke dalam variabel list kehadiran penguji
20	
21	ELSE
22	Tambahkan kehadiran penguji sama dengan 0 ke dalam variabel list kehadiran penguji
23	
24	END IF
25	
26	IF kehadiran pembimbing telah diisi dan kehadiran penguji telah diisi dan kehadiran mahasiswa telah diisi
27	
28	Kirim data kehadiran pembimbing, penguji dan mahasiswa
29	ELSE
30	Tampilkan pesan error.
31	END IF
32	Selesai

### 5.1.8.2 Algoritme Penilaian

Subbab ini menjelaskan rancangan algoritme dari fungsional yang disediakan oleh aplikasi untuk melakukan penilaian. Rancangan algoritme yang ditunjukkan pada Tabel 5.28 merupakan perancangan algoritme proses penilaian.

**Tabel 5.28 Perancangan algoritme memberikan penilaian**

No	Pseudocode
1	Mulai
2	Set variable nilai
3	Masukan nilai
4	Tambahkan nilai ke dalam variable nilai
5	Simpan Nilai ke penyimpanan lokal
6	IF penilaian sudah terisi semua maka
7	Kirim penilaian
8	ELSE
9	Tampilkan pesan error kepada pengguna
10	END IF
11	Akhir kondisi
	Selesai

### 5.1.8.3 Algoritme Merekap Nilai

Subbab ini menjelaskan rancangan algoritme dari fungsional yang disediakan oleh aplikasi untuk melakukan proses rekapitulasi nilai sehingga pengguna dapat melihat rekap nilai. Rancangan algoritme ini terdiri atas dua algoritme, yaitu algoritme menghitung rata-rata dan memeriksa *gap* nilai. Rancangan algoritme rata-rata nilai pembimbing yang ditunjukkan pada Tabel 5.30 dan Tabel 5.29 merupakan rancangan algoritme untuk memeriksa *gap* nilai.

**Tabel 5.28 Perancangan algoritme fungsi memeriksa *gap* nilai**

No	Pseudocode
1	Mulai
2	Ambil data penilaian;
3	Set variable list gap;
4	Ulangi sebanyak data penilaian
5	SET variabel gap = data penilaian pembimbing - data
6	Penilaian penguji
7	Jika gap lebih besar sama dengan 2 maka

**Tabel 5.29 Perancangan algoritme fungsi memeriksa *gap* nilai (Lanjutan)**

9	Tambahkan data penilaian ke variabel list gapt
10	Akhir kondisi
11	Akhir perulangan
12	Tampilkan jumlah gap nilai
	Selesai

**Tabel 5.29 Perancangan algoritme fungsi rata-rata nilai skripsi pembimbing**

No	Pseudocode
1	Mulai
2	Mengambil data pembimbing
3	IF pembimbing hadir maka
4	Mengambil data rata-rata nilai dari
5	setiap komponen penilaian pembimbing
6	Ulangi sebanyak data penilaian pembimbing
7	IF komponen merupakan komponen ke 3
8	Rata-rata komponen dikalikan 0.5
9	ELSE
10	Rata-rata komponen dikalikan 1
11	END IF
12	
13	IF komponen merupakan komponen ke 5
14	Rata-rata komponen dikalikan 1.5
15	ELSE
16	Rata-rata komponen dikalikan 1
17	END IF
18	
19	Jumlahkan rata-rata komponen
20	Lakukan operasi jumlah rata-rata
21	Komponen/5
22	Akhir perulangan
23	Lakukan operasi jumlah rata-rata
24	Keseluruhan/2
25	ELSE
26	Mengambil data rata-rata nilai dari
27	Komponen 1 dan 2
28	Jumlahkan rata-rata nilai komponen 1
29	Dan 2
30	Lakukan operasi jumlah rata-rata
31	Komponen/2
32	END IF
33	Tampilkan rata-rata pembimbing
	Selesai

#### 5.1.8.4 Algortima Membuat Berita Acara

Subbab ini menjelaskan rancangan algoritme dari fungsional yang disediakan oleh aplikasi untuk membuat berita acara. Rancangan algoritme yang ditunjukkan pada Tabel 5.31 merupakan rancangan algoritme untuk membuat berita acara.

**Tabel 5.30 Perancangan algoritme membuat berita acara**

No	Pseudocode
1	Mulai
2	Masukan data berita acara
3	IF data berita acara belum lengkap
4	Tampilkan pesan error
5	ELSE
6	Ubah data berita acara menjadi JSON
7	Kirim data berita acara yang telah diubah menjadi JSON
8	END IF
9	Selesai

## BAB 6 EVALUASI PROTOTIPE

Pada bab ini membahas hasil yang di dapatkan dari evaluasi prototipe. Evaluasi prototipe dilakukan sebanyak dua kali iterasi. Pada iterasi pertama peneliti menggunakan A/B dan pada iterasi kedua peneliti menggunakan pengukuran *usability*. Selain itu juga prototipe diujikan kepada lima pengguna.

### 6.1 Iterasi Satu

Pada iterasi satu peneliti menggunakan metode *A/B testing* untuk mendapatkan desain dan *flow* yang tepat dari dua versi desain yang diusulkan (Lampiran c). Metode *A/B testing* dilakukan dengan menggunakan kuisisioner yang terdiri 20 pernyataan sebagaimana yang ditunjukkan pada Tabel 6.1. Selain itu, iterasi ini, pengguna dapat memberikan saran terhadap sistem yang akan dibangun dan saran ini akan menjadi pertimbangan dalam menentukan rancangan antarmuka aplikasi.

Tabel 6.1 Kuisisioner *A/B Testing*

No	Prototipe A	Prototipe B
1	Menu notifikasi, jadwal ujian menggunakan navigation menu	Menu notifikasi, jadwal ujian menggunakan tab menu
2	Halaman beranda menampilkan logo filkom dan 5 list jadwal ujian	Halaman beranda menampilkan 3 tab menu, dan masing - masing tab menampilkan informasi jadwal ujian pembimbing, penguji dan informasi notifikasi
3	Menggunakan dropdown untuk filter peran pembimbing dan penguji	Menggunakan tab menu untuk filter peran pembimbing dan Penguji
4	Untuk mengetahui peran sebagai ketua majelis, prototipe A menggunakan filter dropdown atau melihat keterangan peran di setiap list jadwal	Untuk mengetahui peran sebagai ketua mejelis harus melihat di halaman detail jadwal ujian.
5	Penilaian dilakukan menggunakan <i>seekbar</i> 1-7	Penilaian dilakukan menggunakan popup list pilihan 1-7
6	fungsiional membaca dokumen di representasikan dengan icon dan teks	fungsiional membaca dokumen di representasikan dengan <i>button</i>
7	Komponen penilaian ditampilkan dalam bentuk <i>step by step</i>	Komponen penilaian ditampilkan dalam bentuk list

Tabel 6.1 Kuisiner A/B Testing (Lanjutan)

8	Letak keterangan, informasi dan <i>text input</i> yang ditampilkan dibuat secara vertical (berita acara)	Letak keterangan, informasi dan <i>text input</i> yang ditampilkan dibuat secara vertical dan horizontal (berita acara)
9	Menggunakan warna filkom apps versi terbaru	Menggunakan warna filkom apps versi lama
10	Informasi jadwal pelaksanaan ujian ditampilkan terpisah dari informasi skripsi	Informasi jadwal pelaksanaan ujian ditampilkan jadi satu dengan informasi skripsi
11	Informasi nama dosen pembimbing dan penguji ditampilkan jadi satu <i>cardview</i>	Informasi nama dosen pembimbing dan penguji ditampilkan terpisah
12	Informasi data mahasiswa menggunakan label nama, nim, jurusan, no.hp, email	Informasi data mahasiswa tidak menggunakan label dan pada no.hp dan email mahasiswa menggunakan icon
14	Tidak memiliki history penilaian	Memiliki history penilaian
15	Penggunaan <i>fungsi</i> onal verifikasi kehadiran dapat dilakukan ketika pengguna menekan button kehadiran	Penggunaan <i>fungsi</i> onal verifikasi kehadiran dapat dilakukan ketika pengguna menekan button menilai
16	Menu lihat nilai penguji dan pembimbing ditampilkan dengan <i>button top menu</i>	Menu lihat nilai penguji dan pembimbing ditampilkan tab menu
17	Menu timer ditampilkan dengan <i>button top menu</i>	Menu timer ditampilkan dengan <i>floating button</i>
18	Rekap nilai dan berita acara ditampilkan di halaman yang berbeda	Rekap nilai dan berita acara ditampilkan di halaman yang sama
19	Tampilan dan penggunaan <i>fungsi</i> onal Timer lebih nyaman	Tampilan dan penggunaan <i>fungsi</i> onal Timer lebih nyaman
20	Secara keseluruhan tampilan dan penggunaan lebih nyaman	Secara keseluruhan tampilan dan penggunaan lebih nyaman

Berdasarkan hasil yang didapatkan pengguna lebih memilih prototipe a dibandingkan prototipe b. Adapun bagian-bagian yang menurut pengguna lebih tepat di implementasikan pada aplikasi penilaian ujian skripsi. Bagian-bagian tersebut terdapat pada Tabel 6.2.

Tabel 6.2 Hasil uji coba A/B Testing

No.	Hasil
1	Menu notifikasi, jadwal ujian menggunakan navigation menu
2	Halaman beranda menampilkan logo filkom dan 5 list jadwal ujian
3	Menggunakan dropdown untuk filter peran pembimbing dan penguji
4	Untuk mengetahui peran sebagai ketua majelis, prototipe A menggunakan filter dropdown atau melihat keterangan peran di setiap list jadwal
5	Penilaian dilakukan menggunakan seekbar pilihan 1-7
6	Fungsional membaca dokumen di representasikan dengan <i>button</i>
7	Komponen penilaian ditampilkan dalam bentuk list
8	Letak keterangan, informasi dan <i>text input</i> yang ditampilkan dibuat secara vertical (berita acara)
9	Menggunakan warna filkom apps versi terbaru
10	Informasi jadwal pelaksanaan ujian ditampilkan terpisah dari informasi skripsi
11	Informasi nama dosen pembimbing dan penguji ditampilkan terpisah
12	Informasi data mahasiswa menggunakan label nama, nim, jurusan, no.hp, email
13	Penggunaan <i>fungsional</i> verifikasi kehadiran dapat dilakukan ketika pengguna menekan <i>button</i> kehadiran
14	Menu lihat nilai penguji dan pembimbing ditampilkan dengan <i>button top menu</i>
15	Menu timer ditampilkan dengan <i>button top menu</i>
16	Rekap nilai dan berita acara ditampilkan di halaman yang sama
17	Tampilan dan penggunaan fungsional timer prototipe A lebih nyaman
18	Secara keseluruhan tampilan dan penggunaan prototipe A lebih nyaman

### 6.1.1 Perancangan Antarmuka Pengguna

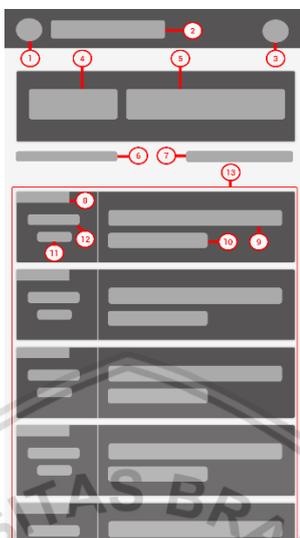
Bagian ini menggambarkan rancangan antarmuka aplikasi yang berdasarkan hasil evaluasi. Rancangan yang telah dibuat akan di implementasikan kembali ke dalam bentuk prototipe, yang kemudian dilakukan kembali evaluasi untuk mengukur *usability*. Rancangan antarmuka yang dibahas pada bagian ini hanya rancangan yang berkaitan dengan hasil evaluasi. Perancangan antarmuka pengguna terdiri dari *wireframe* dan aliran antarmuka pengguna (*screenflow*).

#### 6.1.1.1 Wireframe

Wireframe merupakan sketsa awal yang digunakan sebagai acuan dalam membuat *user interface* pada aplikasi penilaian ujian skripsi. Pada bagian ini akan membahas halaman-halaman yang terkait dengan proses penilaian ujian skripsi. Halaman yang akan dibahas, yaitu halaman login, halaman beranda, halaman

rincian ujian, halaman verifikasi, halaman penilaian, dan halaman rekap nilai & berita acara.

## 1. Halaman Beranda

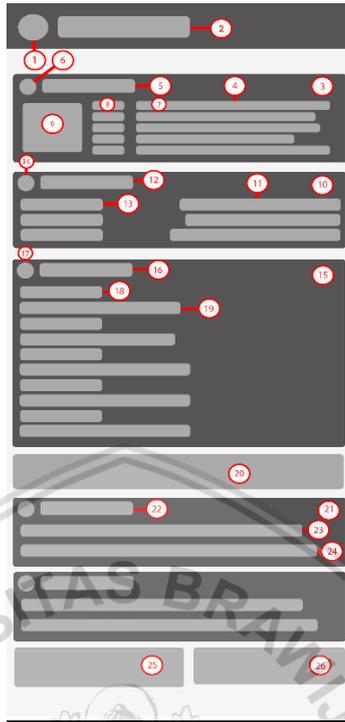


**Gambar 6.1 Wireframe halaman beranda**

Halaman beranda merupakan halaman awal yang ditampilkan setelah pengguna melakukan login. Gambar 6.1 merupakan tampilan halaman beranda yang akan diimplementasikan di prototipe iterasi kedua Rancangan antarmuka ini berdasarkan pernyataan nomor dua yang terdapat di kuisioner. Selain itu juga, pengguna lebih memilih tampilan seperti Gambar 6.1 karena lebih sederhana dan lebih mudah dioperasikan. Sebagaimana ditunjukkan dalam Gambar 6.1, halaman ini terdiri dari *navigation drawer* (1) yang digunakan mengarahkan ke beberapa menu yang ada pada *navigation drawer*, nama halaman (2), menu notifikasi (3), logo filkom (4), *text* fakultas ilmu komputer (5), teks jadwal ujian hari ini (6), daftar jadwal ujian (12). Pada daftar ujian terdiri dari teks yang berisi informasi peran dosen (penguji, pembimbing atau ketua majelis), teks yang berisi informasi jam ujian (11), teks tanggal ujian (9), teks yang berisi informasi judul skripsi, dan teks yang berisi informasi nama mahasiswa (10).

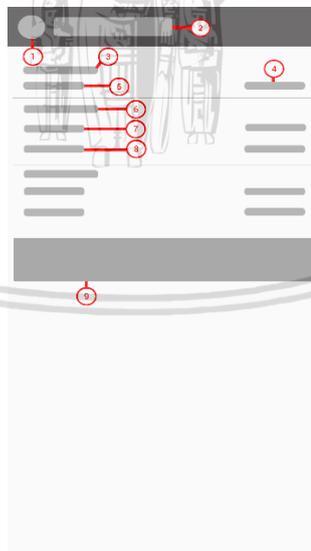
## 2. Halaman Rincian Ujian

Gambar 6.2 merupakan rancangan antarmuka halaman rincian jadwal ujian. Rancangan ini berdasarkan hasil kuisioner pada nomor 6, 7, 8, 10 dan 12. Pengguna lebih memilih tampilan tersebut karena peletakan informasi yang ditampilkan lebih tersusun dengan baik dan pengelompokan informasi terlihat lebih jelas. Sebagaimana yang ditunjukkan dalam Gambar 6.2, halaman ini terdiri dari *icon back* untuk kembali ke halaman sebelumnya (1), teks nama halaman (2), kolom informasi mahasiswa (3), kolom informasi jadwal pelaksanaan ujian skripsi (10), kolom informasi skripsi (15), tombol untuk melihat dokumen (20), kolom informasi pembimbing (21), tombol untuk melakukan verifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji (25) dan tombol untuk melakukan penilaian (26).



Gambar 6.2 Wireframe halaman rincian jadwal ujian

### 3. Halaman Verifikasi Kehadiran



Gambar 6.3 Wireframe halaman verifikasi kehadiran

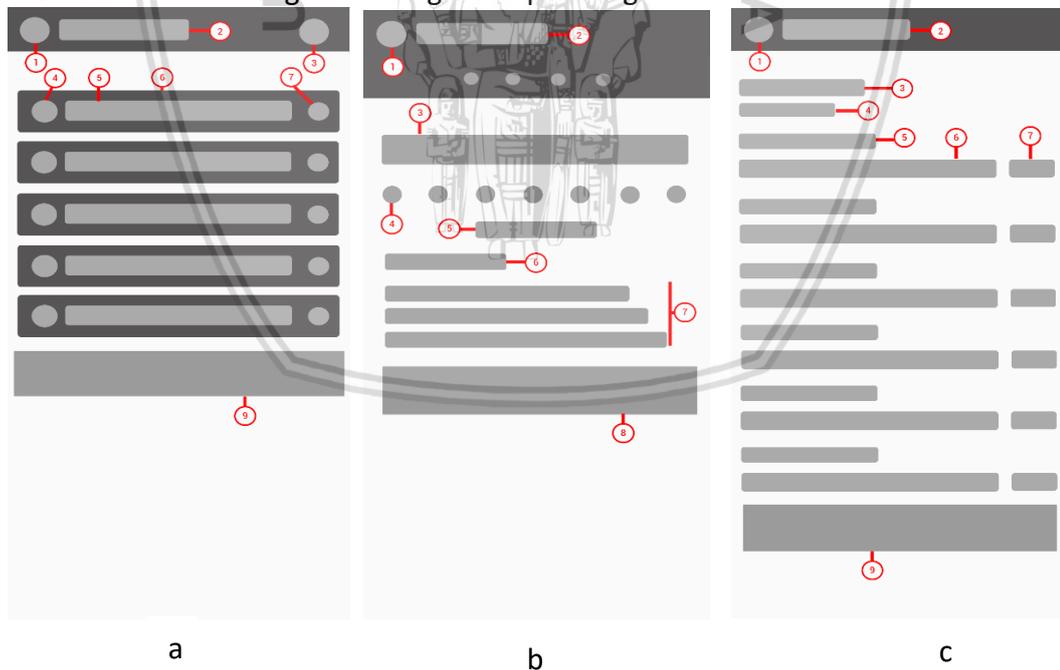
Halaman verifikasi kehadiran merupakan halaman yang berfungsi melakukan verifikasi kehadiran mahasiswa, pembimbing dan penguji. Sebagaimana yang ditunjukkan dalam Gambar 6.3, halaman ini terdiri atas *icon back* untuk kembali ke halaman sebelumnya (1), teks nama halaman (2), teks label mahasiswa (3), teks nama mahasiswa

(5), *switch button* (4), teks label pembimbing (6), teks nama dosen (7 dan 8) dan tombol verifikasi (9). Pada nomor 6, 7, dan 8 juga berlaku pada kolom kehadiran penguji.

#### 4. Halaman Penilaian

Gambar 6.4 merupakan rancangan antarmuka halaman penilaian. Rancangan antarmuka penilaian terdiri dari dua halaman. Halaman pertama menampilkan komponen penilaian yang harus di nilai oleh dosen penguji ataupun dosen pembimbing. Sedangkan, Halaman kedua menampilkan indikator dari setiap komponen yang harus dinilai. Dosen penguji ataupun dosen pembimbing harus terlebih dahulu memilih salah satu komponen penilaian. Rancangan berdasarkan hasil kuisisioner pada nomor 5 dan 7. Selain itu juga, mempertimbangkan saran dari pengguna. Rancangan antarmuka penilain di rancangan menjadi dua halaman dikarenakan adanya perubahan nilai ketika proses ujian. Ketika halaman dibuat seperti yang ditunjukkan pada Gambar 6.4 (a) memberikan kemudahan untuk bisa melakukan perubahan nilai dikomponen tertentu.

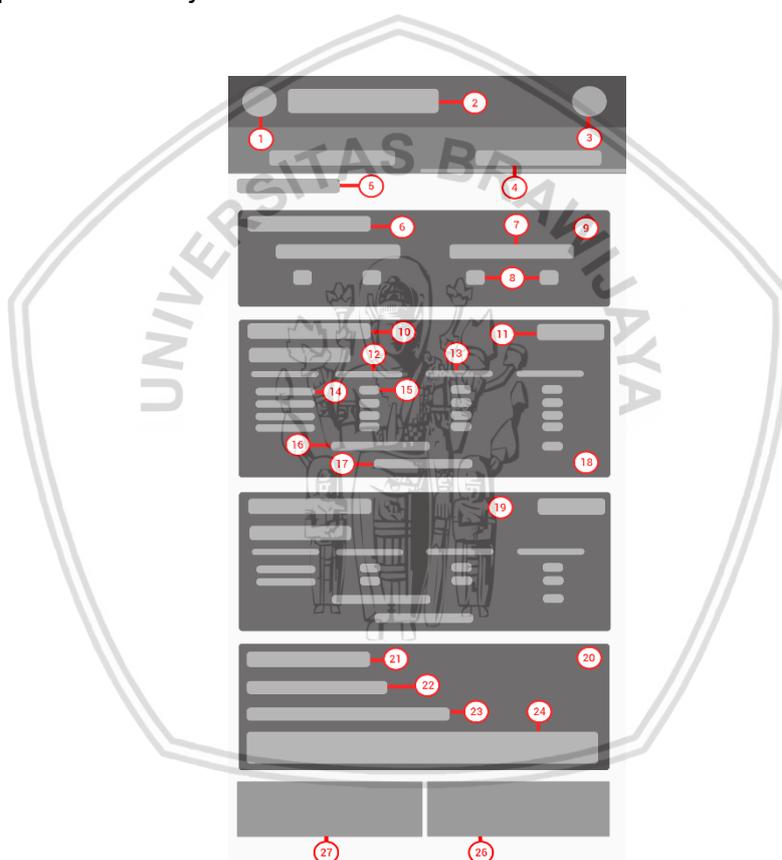
Pada Gambar 6.4 (b) merupakan rancangan anatarmuka penilaian pada indikator. Terlihat pada bagian untuk melakukan penilaian, rancangan ini menggunakan *seekbar* 0-7. Penggunaan *seekbar* ini karena tidak memberikan pandangan awal untuk menilai tinggi atau rendah. Sedangkan Gambar 6.4 (c) merupakan halaman penilaian naskah publikasi, yang dimana pengguna memasukan nilai dengan rentang 0 sampai dengan 100.



**Gambar 6.4 Wireframe** halaman penilaian: (a) halaman utama penilaian (b)halaman indikator; (c) halaman penilaian naskah publikasi

## 5. Halaman Rekap Nilai & Berita Acara

Halaman rekap nilai merupakan halaman yang bertujuan untuk menampilkan hasil perhitungan nilai dari semua aspek penilaian skripsi. Halaman ini dirancang menjadi satu halaman dengan penilaian. Hal ini dirancang untuk mempermudah ketua majelis untuk kembali melakukan perubahan nilai dan mengontrol hasil penilaian yang telah dilakukan oleh majelis. Rancangan ini didukung oleh hasil kuisisioner yang diberikan oleh pengguna pada nomor 16. Selain itu juga terdapat dua tombol penilaian pembimbing dan penguji. Dua tombol itu berdasarkan hasil kuisisioner pada nomor 14, yang dimana pengguna lebih memilih tombol dibandingkan *tab menu*. Penempatan dua tombol tersebut untuk memfokuskan antara proses penilaian dengan proses mengatur jalannya sidang pada ketua majelis.



**Gambar 6.5 Wireframe halaman rekap nilai**

Sebagaimana yang ditunjukkan dalam Gambar 6.5, halaman ini terdiri dari *icon back* untuk kembali ke halaman sebelumnya (1), *text* nama halaman (2), menu notifikasi (3), *text* yang menunjukkan keterangan waktu terakhir perubahan data, kolom informasi *status* proses penilaian setiap penilai yang berfungsi untuk mengetahui siapa saja yang sudah menyelesaikan penilaian atau yang sedang dalam proses penilaian (4-9), kolom informasi nilai skripsi (10-14), kolom informasi nilai kualitas penulisan skripsi (15-17), kolom memasukkan berita acara (18-20), tombol untuk memproses berita acara (21-22) dan tombol untuk memeriksa *gap* nilai (23-24).

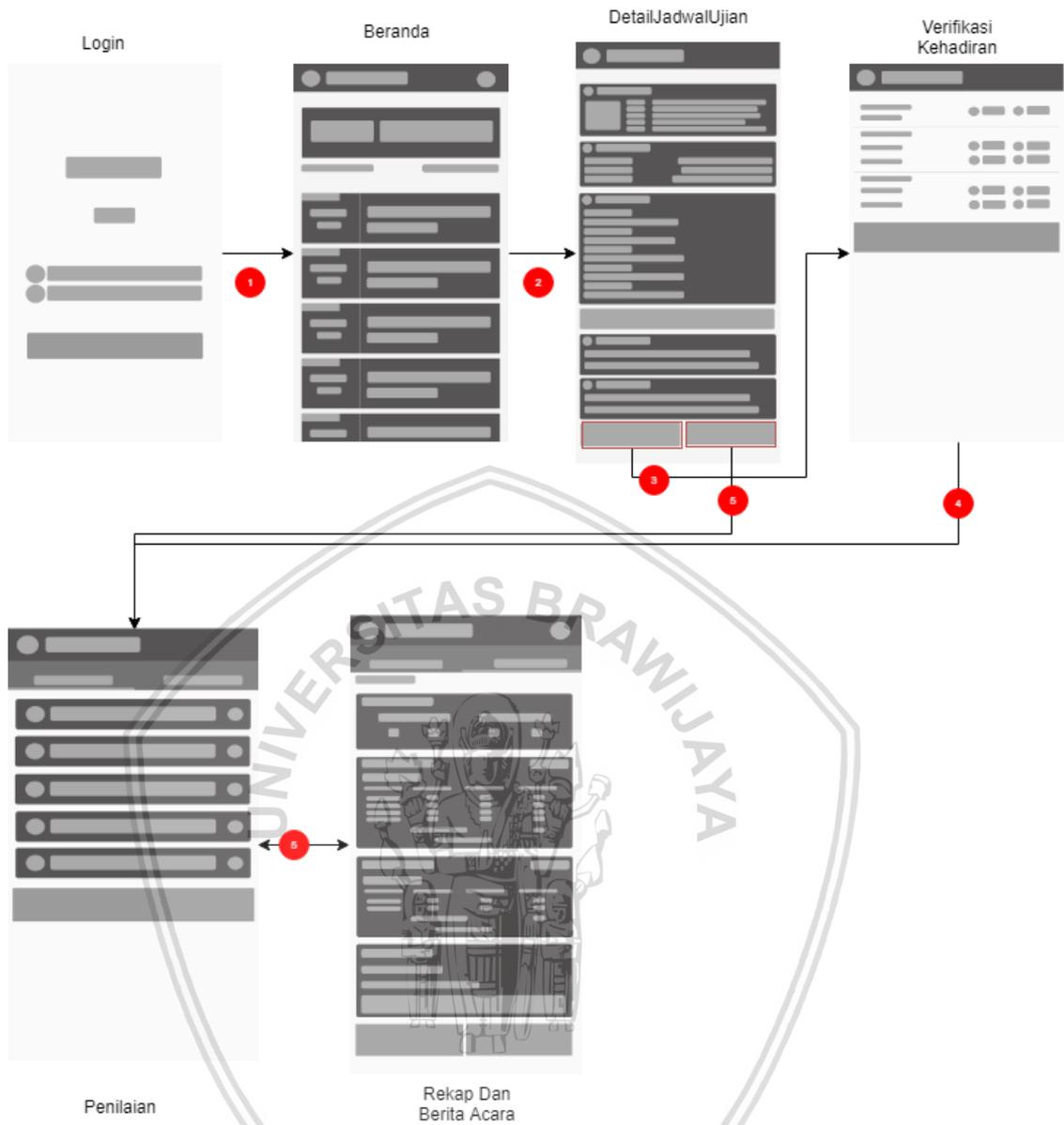
Pada kolom informasi status proses penilaian (9) dan kolom informasi nilai kualitas penulisan skripsi terdiri atas teks nama kolom (6), teks peran pembimbing dan penguji (7) dan status proses penilaian yang di gambarkan menggunakan *checkbox* (8). Kemudian, pada kolom informasi nilai skripsi terdiri atas teks nama kolom (10), teks nilai akhir ujian skripsi yang ditampilkan dalam bentuk angka dan huruf (11), teks label penilaian yang digunakan untuk menandakan bagian-bagaian penilaian seperti keterangan, rata-rata, bobot dan sub total (12 dan 13), teks label terkait nama penilaian seperti seminar hasil, publikasi, ujian skripsi pembimbing, ujian skripsi penguji (14), teks yang digunakan untuk menampilkan nilai (15), teks nilai akhir dalam bentuk angka (16), teks selengkapnya yang berfungsi sebagai tombol untuk melihat rincian nilai (17). Kolom beritanya kolom berita acara yang terdiri dari teks label dan teks informasi nilai akhir skripsi (21), *dropdown* status kelulusan untuk memilih status kelulusan mahasiswa (23) dan kolom untuk memasukan catatan khusus (24). Selain itu, halaman ini juga menampilkan informasi *gap* nilai sebagaimana ditunjukkan dalam Gambar 5.15 (b). Kolom tersebut terdiri atas *icon gap* (28), *text total gap* (29), dan *text* selengkapnya yang digunakan sebagai tombol untuk melihat rincian *gap* nilai.

#### 6.1.1.2 Aliran Antarmuka Pengguna

Subab ini membahas aliran antarmuka pengguna aplikasi penilaian ujian skripsi. Aliran antarmuka ini hanya membahas pada aliran antarmuka dalam proses penilaian ujian skripsi yang terdapat di ketua majelis. Aliran antarmuka dimulai dari halaman login. Ketika pengguna telah melakukan login aplikasi akan menampilkan halaman beranda. Pada halaman ini pengguna akan memilih jadwal ujian.

Kemudian aplikasi akan menampilkan rincian jadwal ujian yang pengguna telah pilih. Pengguna terlebih dahulu memverifikasi kehadiran majelis dan aplikasi akan menampilkan halaman verifikasi. Setelah verifikasi telah dilakukan, alur terbagi dua sebagaimana yang ditunjukkan Gambar 6.5 pada nomor 4 dan 5. Alur nomor 4 merupakan alur dimana pengguna dapat langsung menilai tanpa harus terlebih dahulu berada di halaman rincian ujian. Pada nomor 5 pengguna dapat melakukan penilaian pada halaman rincian ujian. Kondisi pada alur ini terjadi, apabila pengguna menginginkan terlebih dahulu melihat informasi ujian skripsi.

Alur selanjutnya yaitu halaman ketua majelis. Pada halaman ini terdiri dari penilaian dan rekap nilai & berita acara. Aplikasi akan terlebih dahulu menampilkan penilaian. Bagian penilaian ini merupakan bagian dimana ketua majelis memasukan penilaiannya. Pada halaman ketua majelis terdapat tab menu rekap nilai & berita acara. Tab menu tersebut untuk menampilkan rekap nilai & berita acara ujian skripsi. Bagian tersebut untuk melihat hasil rekap nilai, yang dimana rekap nilai terdiri dari rincian nilai ujian, rincian nilai kualitas naskah publikasi, dan *gap* nilai. Kemudian, terdapat kolom berita acara untuk memasukan berita acara ujian skripsi.



Gambar 6.6 Aliran sntarmuka pengguna

### 6.2 Iterasi Dua

Pada iterasi kedua peneliti tidak menggunakan metode *A/B testing*. Hal ini dikarenakan data yang di peroleh terkait desain yang diinginkan oleh pengguna dirasa sudah cukup. Sehingga di iterasi kedua ini peneliti melakukan pengukuran *usability* berdasarkan *usability metrics* ISO/IEC 9126-4. Skenario dari pengujian prototipe ini dilakukan dengan memberikan empat *task* yang setiap *task*-nya menggunakan empat fungsional utama aplikasi, yaitu verifikasi kehadiran, penilaian, rekap nilai, dan berita acara. Empat task yang dikerjakan oleh pengguna sebagai berikut:

Tabel 6.3 Daftar *Task* Pengguna Berdasarkan Fungsional

No	Pernyataan	Fungsional
1	Anda akan menilai ujian skripsi mahasiswa dengan nama en doni. Pada saat ujian, dua pembimbing dan penguji serta mahasiswa hadir pada saat ujian, sebagai ketua majelis andamengharuskan melakukan verifikasi kehadiran pembimbing, penguji dan mahasiswa.	Verifikasi Kehadiran
2	Anda akan menilai ujian skripsi mahasiswa dengan nama en doni. Pada saat ujian, dua pembimbing dan penguji serta mahasiswa hadir pada saat ujian, sebagai ketua majelis andamengharuskan melakukan verifikasi kehadiran pembimbing, penguji dan mahasiswa.	Penilaian
3	Sebagai ketua majelis dalam ujian skripsi, anda melihat hasil rekapitulasi nilai dari penilaian yang telah dilakukan dan memastikan tidak ada GAP nilai diantara pembimbing dan penguji.	Merekap Nilai
4	Setelah memastikan tidak ada GAP, ketua majelis mengisi berita acara	Membuat Berita Acara

### 6.2.1 Effectiveness

Tabel 6.4 Data *effectiveness* berdasarkan fitur

Responden	Complete			
	Verifikasi Kehadiran	Penilaian	Rekap Nilai	Berita Acara
Pengguna 1	1	1	1	1
Pengguna 2	1	1	1	1
Pengguna 3	1	1	1	1
Pengguna 4	1	1	1	1
Pengguna 5	1	1	1	1
Completion Rate	100%	100%	100%	100%

Pada aspek *effectiveness*, peneliti melakukan pengukuran tingkat keberhasilan dalam menyelesaikan *task* yang diberikan. Berdasarkan Tabel 6.4, tingkat keberhasilan memperoleh 100% dari keseluruhan *task* yang dikerjakan. Lima responden berhasil mengerjakan tugas dengan menggunakan fungsi verifikasi kehadiran, penilaian, rekap nilai dan berita acara pada prototipe aplikasi

penilaian ujian skripsi. Selain itu dari tingkat keberhasilan *task* yang mencapai 100%, terdapat kesalahan yang dilakukan oleh pengguna. Kesalahan yang sering terjadi terdapat di fungsional penilaian dan berita acara. Pada fungsional penilaian pengguna gagal memahami perbedaan dua tombol (tombol kirim nilai dan tombol selesai) yang terdapat di halaman penilaian. Kelima pengguna melakukan kesalahan yang sama yaitu menekan tombol selesai, yang seharusnya menekan tombol kirim nilai ketika penilaian telah dilakukan. Kemudian, kesalahan yang terjadi di fungsional berita acara, terdapat pada bagian menentukan predikat (lulus, lulus denganrevisi, tidak lulus, tidak lulus kembali ke preproposai) mahasiswa. Predikan mahasiswa sering sekali terlewati oleh pengguna. Tiga dari lima pengguna langsung menekan tombol submit ketika mengerjakan *task* yang berkaitan dengan fungsional berita acara. Pengguna beranggapan bahwa bagian dalam menentukan predikat mahasiswa kurang menggambarkan adanya *field* yang harus ditentukan. Berikut Tabel 6.5 rangkuman dari kesalahan-kesalahan yang dialami pengguna.

**Tabel 6.5 Kesalahan pengguna**

Fungsional	Kesalahan	Jumlah Pengguna
Verifikasi Kehadiran	-	-
Penilaian	Kesalahan dalam menekan tombol untuk menyimpan dan mengirimkan penilaian.	5
Rekap Nilai	-	-
Berita Acara	Pengguna lupa menentukan status kelulusan mahasiswa.	3

### 6.2.2 Efficiency

Pada aspek *efficiency*, peneliti mengukur dengan menggunakan persamaan *time based efficiency* berdasarkan waktu yang dibutuhkan pengguna dalam menyelesaikan tugas. Tabel 6.6 merupakan waktu yang dibutuhkan pengguna dalam menyelesaikan *task* dengan menggunakan fungsional verifikasi kehadiran, penilaian, rekap nilai, dan berita acara. Kemudian, Tabel 6.7 merupakan *efficiency* setiap *task* dan secara keseluruhan. Berdasarkan Tabel 6.7 diperoleh tingkat *efficiency* pada fungsional verifikasi kehadiran sebesar 0.240 *goals/sec*, fungsional penilaian sebesar 0.083 *goals/sec*, fungsional rekap nilai sebesar 0.228 *goals/sec*, fungsional berita acara 0.228 *goals/sec*. Lalu, secara keseluruhan aplikasi ini memiliki tingkat *efficiency* sebesar 0.192 *goals/sec*.

**Tabel 6.6 Waktu yang dibutuhkan pengguna dalam mengerjakan task berdasarkan fitur**

Pengguna	Waktu			
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
Pengguna 1	21	65	18	25

**Tabel 6.5 Waktu yang dibutuhkan pengguna dalam mengerjakan task berdasarkan fitur (Lanjutan)**

Pengguna 2	28	84	20	33
Pengguna 3	25	58	22	36
Pengguna 4	33	123	28	28
Pengguna 5	26	85	23	31

**Tabel 6.7 Efficiency aplikasi penilaian ujian skripsi**

Efficiency (goals/sec)			
Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
0.240	0.083	0,228	0.228
<b>Keseluruhan</b> 0.203993516			

### 6.2.3 Satisfaction

**Tabel 6.8 Hasil penilaian kuisioner Single Ease Question**

Pengguna	Nilai			
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
Pengguna 1	6	6	5	6
Pengguna 2	6	6	3	4
Pengguna 3	6	7	7	7
Pengguna 4	5	6	6	6
Pengguna 5	7	5	7	6
<b>RataRata</b>	6	6	5.6	5.8
<b>Rata-Rata Keseluruhan</b>	5.85			

Pada aspek *satisfaction* peneliti melakukan pengukuran dengan menggunakan Single Ease Question. Penerapan dari Single Ease Question dengan mengajukan pertanyaan setelah pengguna menyelesaikan *task* yang diberikan. Penilaian dari pertanyaan tersebut menggunakan skala likert dari 1-7. Tabel 6.8 merupakan hasil tingkat *satisfaction* yang diperoleh dari prototipe aplikasi penilaian ujian skripsi versi dua. Hasil yang diperoleh dari pengukuran *satisfaction* mencapai 5.85 untuk rata-rata keseluruhan. Nilai tersebut dapat dikatakan sudah melebihi nilai rata-rata yang harus diperoleh dalam aspek *satisfaction*.



## BAB 7 IMPLEMENTASI

Pada bab ini membahas hasil implementasi aplikasi yang berdasarkan perancangan yang telah dibahas di bab 5. Pada bab ini terbagi menjadi beberapa pembahasan, antara lain spesifikasi sistem, batasan sistem, implementasi kelas dan implementasi antarmuka.

### 7.1.1 Spesifikasi Sistem

Spesifikasi sistem merupakan rincian mengenai perangkat keras dan perangkat lunak yang digunakan ketika melakukan proses implementasi dari aplikasi penilaian ujian skripsi. Spesifikasi sistem terbagi menjadi 2, yaitu spesifikasi perangkat keras dan spesifikasi perangkat lunak.

#### 7.1.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras menjelaskan spesifikasi dari perangkat keras yang digunakan dalam mendukung proses implementasi. Perangkat keras yang digunakan dalam proses implementasi adalah server. Server ini digunakan sebagai tempat berjalannya *web service* dalam mengelola data yang diterima dari aplikasi. Spesifikasi server yang digunakan dijelaskan pada Tabel 7.1.

**Tabel 7.1 Spesiifikasi perangkat keras (Server)**

Disk	1 GB
Protokol	HTPP & HTTPS
Database	MYSQL

#### 7.1.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak menjelaskan spesifikasi dari perangkat lunak yang digunakan dalam mendukung proses implementasi. Berikut Tabel 7.2 yang menjelaskan spesifikasi dari perangkat lunak.

**Tabel 7.2 Spesifikasi perangkat lunak**

IDE	Android Studio 3.1.2
Virtual Web Server	XAMPP v3.2.2
Database	10.1.24
Library Pendukung	<ul style="list-style-type: none"> <li>• Retrofit</li> <li>• Glide</li> <li>• Butterknife</li> <li>• ROOM</li> <li>• Firebase Cloud Messaging</li> <li>• Tymon JWT AUTH</li> </ul>

**7.1.2 Batasan Implementasi**

Batasan implementasi merupakan syarat minimal yang dibutuhkan dalam mengimplementasi aplikasi penilaian ujian skripsi. Berikut ini batasan implementasi aplikasi penilaian ujian skripsi:

- a. Aplikasi berjalan minimal di android 4.03 Ice Scream Sandwich (API 15)
- b. Aplikasi membutuhkan pihak ketiga yaitu web service, dalam melakukan pertukaran data.
- c. Aplikasi membutuhkan jaringan internet untuk dapat terhubung dengan *web service*.
- d. *Web service* menggunakan RESTFULL API dengan format JSON
- e. Keamanan *web service* menggunakan autentikasi JSON Web Token (JWT)

**7.1.3 Implementasi Aplikasi**

Bagian ini membahas implementasi aplikasi, berdasarkan perancangan yang telah dibahas pada bab 5. Implementasi fungsi ini disesuaikan dengan *sequence diagram*, *class diagram* dan perancangan algoritma. Implementasi aplikasi terdiri dari fungsional utama aplikasi, yaitu memverifikasi kehadiran, memberikan penilaian, merekap nilai dan membuat berita acara.

**7.1.3.1 Implementasi Memverifikasi Kehadiran**

Bagian ini membahas Implementasi dari fungsional aplikasi yaitu memverifikasi kehadiran. Implementasi ini di berdasarkan rancangan *class diagram*, *sequence diagram* dan algoritme memverifikasi kehadiran yang sudah dibuat sebelumnya. Implementasi memverifikasi kehadiran terdapat di *class VerifikasiPresenter* dan *VerifikasiActivity* sebagaimana yang ditunjukkan pada Tabel 7.3 dan Tabel 7.4.

**Tabel 7.3 Implementasi kode program memverifikasi kehadiran pada *class verifikasiactivity***

No	VerifikasiActivity.Class
1	@OnClick({R.id.radio_hadir_mahasiswa,R.id.radio_tdkhadir_mahasiswa}
2	)
3	public void onRbClickMahasiswa (RadioButton radioButton){
4	if (radioButton.getId()==R.id.radio_hadir_mahasiswa){
5	rgPembimbing1.setVisibility(View.VISIBLE);
6	rgPembimbing2.setVisibility(View.VISIBLE);
7	rgPengujil.setVisibility(View.VISIBLE);
8	rgPenguji2.setVisibility(View.VISIBLE);
9	}else{
10	rgPembimbing1.setVisibility(View.GONE);
11	rgPembimbing2.setVisibility(View.GONE);
12	rgPengujil.setVisibility(View.GONE);
13	rgPenguji2.setVisibility(View.GONE);
14	}
15	}
16	
17	@OnClick({R.id.radio_hadir_pembimbing1,R.id.radio_tdkhadir_pembimbi
18	ng1})
19	public void onRbClickPembimbing1 (RadioButton radioButton){

**Tabel 7.3 Implementasi kode program memverifikasi kehadiran pada class verifikasiactivity (Lanjutan)**

20	if (radioButton.getId()==R.id.radio_hadir_pembimbing1){
21	verifikasiPresenter.setKehadiranDosenPembimbing1("1");
22	}else{
23	verifikasiPresenter.setKehadiranDosenPembimbing1("0");
24	}
25	}
26	@OnClick({R.id.radio_hadir_pembimbing2,R.id.radio_tdkhadir_pembimbing2})
28	public void onRbClickPembimbing2(RadioButton radioButton){
29	
30	if (radioButton.getId()==R.id.radio_hadir_pembimbing2){
31	verifikasiPresenter.setKehadiranDosenPembimbing2("1");
32	}else{
33	verifikasiPresenter.setKehadiranDosenPembimbing2("0");
34	}
35	}
36	@OnClick({R.id.radio_hadir_penguji1,R.id.radio_tdkhadir_penguji1})
36	public void onRbClickPenguji1(RadioButton radioButton){
38	if (radioButton.getId()==R.id.radio_hadir_penguji1){
39	verifikasiPresenter.setKehadiranDosenPenguji1("1");
40	}else{
41	verifikasiPresenter.setKehadiranDosenPenguji1("1");
42	}
43	}
44	@OnClick({R.id.radio_hadir_penguji2,R.id.radio_tdkhadir_penguji2})
45	public void onRbClickPenguji2(RadioButton radioButton){
46	if (radioButton.getId()==R.id.radio_hadir_penguji2){
47	verifikasiPresenter.setKehadiranDosenPenguji2("1",false);
48	lnDosenPenguji.setVisibility(View.GONE);
49	updatePengganti=false;
50	
51	}else{
52	updatePengganti=true;
53	lnDosenPenguji.setVisibility(View.VISIBLE);
54	
55	}
56	}
57	

Tabel 7.3 merupakan implementasi memverifikasi kehadiran pada *class* VerifikasiActivity. Proses yang dilakukan pada *class* tersebut untuk menyimpan data kehadiran mahasiswa, penguji dan pembimbing dan kemudian data tersebut dilakukan pemeriksaan dan pengiriman data di *class presenter* sebagaimana yang ditunjukkan pada Tabel 7.4. Pada Tabel 7.4, Baris 8 – 39 merupakan fungsi yang menjalankan proses pemeriksaan kelengkapan data kehadiran. Kemudian, baris 43 – 89 merupakan fungsi yang menjalankan proses pengiriman data ke *web service*.

**Tabel 7.4 Implementasi Kode Program Memverifikasi Kehadiran Pada Class verifikasipresenter**

No	VerifikasiPresenter.class
1	private HashMap<String, String> kehadiranPembimbing1;
2	private HashMap<String, String> kehadiranPembimbing2;
3	private HashMap<String, String> kehadiranPenguji1;
4	private HashMap<String, String> kehadiranPenguji2;

**Tabel 7.4 Implementasi Kode Program Memverifikasi Kehadiran Pada Class verifikasiPresenter (Lanjutan)**

```

5  @Override
6  public void validateIsChecked(int mahasiswaIsChecked, int
7  pembimbing1IsChecked,
8                               int pembimbing2IsChecked, int
9  pengujilIsChecked, int penguji2IsChecked) {
10     kehadiran = new ArrayList<>();
11     if (mahasiswaIsChecked == -1 || pembimbing1IsChecked == -1 ||
12     pembimbing2IsChecked == -1
13         || pengujilIsChecked == -1 || penguji2IsChecked == -1)
14     {
15
16         if (mahasiswaIsChecked == R.id.radio_tdkhadir_mahasiswa) {
17             view.validateSuccess(kehadiran);
18         } else {
19
20             view.validateFailed("Anda Tidak Dapat Melakukan
21     Verifikasi, Terdapat Kehadiran Yang Belum Anda Checklist");
22
23         }
24
25     } else {
26         if (pengujilIsChecked == R.id.radio_tdkhadir_pengujil) {
27             view.validateFailed("Anda Tidak Dapat Melakukan
28     Verifikasi, Ketika Ketua Majelis Tidak Hadir");
29
30         } else {
31             setDataKehadiran();
32             view.validateSuccess(kehadiran);
33
34         }
35     }
36
37 }
38
39 @Override
40 public void verifikasiProcess() {
41     user = UserHelper.getUser(context);
42     final HashMap<String, Object> param = new HashMap<>();
43     param.put("id_ujian", PrefUtil.getIdUjianTemp(context));
44     if (dosenPengganti != null) {
45         param.put("update_majelis", "1");
46         param.put("data_pengganti", dosenPengganti);
47     } else {
48         param.put("update_majelis", "0");
49     }
50
51     param.put("dosen", kehadiran);
52     view.showProgress();
53     ServiceGenerator.createService(EndpointAPI.class,
54     user.getApiToken()).verifikasi(param)
55     .enqueue(new Callback<GetVerifikasiResponse>() {
56         @Override
57         public void onResponse(Call<GetVerifikasiResponse>
58     call, Response<GetVerifikasiResponse> response) {
59             view.dismissProgress();
60             if (response.code() == 200) {
61                 PrefUtil.setIsDoneVerifikasi(context,
62     true);
63                 PrefUtil.setAccessPenilaian(context,
64     PrefUtil.getIdUjianTemp(context), true);
65
66

```

**Tabel 7.4 Implementasi Kode Program Memverifikasi Kehadiran Pada Class verifikasipresenter (Lanjutan)**

67	if
68	(String.valueOf(param.get("update_majelis")).equalsIgnoreCase("0"))
69	{
70	view.showVerifikasiSuccess();
71	
72	} else {
73	view.showWaitingConfirm();
74	}
75	} else {
76	
77	view.showVerifikasiFailed(response.message());
78	}
79	}
80	
81	@Override
82	public void onFailure(Call<GetVerifikasiResponse>
83	call, Throwable t) {
84	view.dismissProgress();
85	view.showVerifikasiFailed(t.getMessage());
86	}
87	});
88	
89	}
90	
91	

### 7.1.3.2 Implementasi Memberikan Penilaian

Bagian ini membahas Implementasi dari salah satu fungsional aplikasi yaitu memberikan penilaian. Implementasi ini di berdasarkan rancangan *class diagram*, *sequence diagram* dan algoritme memberikan penilaian yang sudah dibuat sebelumnya. Implementasi memberikan penilaian terdapat di *class* PenilaianPresenter dan IndikatorPresenter sebagaimana yang ditunjukkan pada Tabel 7.5 dan Tabel 7.6.

**Tabel 7.5 Implementasi memberikan kode program penilaian pada kelas indikatorpresenter**

No	IndikatorPresenter.class
1	for (int i = 0; i < dataKomponen.getIndicators().size(); i++) {
2	Nilai nilaiRincian = new Nilai();
3	
4	nilaiRincian.setIndikatorId(dataKomponen.getIndicators().get(i).getId());
5	nilaiRincian.setIdDosen(user.getId());
6	
7	
8	nilaiRincian.setNilai(Integer.parseInt(editTexts.get(dataKomponen.getIndicators().get(i).getIndikator()).getText().toString()));
9	PenilaianHelper.insertNilai(context, nilaiRincian);
10	
11	
12	}

Pada Tabel 7.5 merupakan kode program yang berfungsi untuk menyimpan penilaian setiap indikator ke dalam penyimpanan lokal pada perangkat bergerak. Kemudian pada Tabel 7.6 merupakan implementasi kode program untuk mengirimkan penilaian ke *web service*.

**Tabel 7.6 Implementasi kode program memberikan penilaian pada kelas penilaianpresenter**

No	PenilaianPresenter.class
1	@Override
2	public void submitNilai(String isKetua) {
3	view.showProgress();
4	HashMap<String, Object> param = new HashMap<>();
5	param.put("id_skripsi", PrefUtil.getIdSkripsiTemp(context));
6	param.put("id_ujian", PrefUtil.getIdUjianTemp(context));
7	List<Nilai> nilaiUjian =
8	PenilaianHelper.getPenilaian(context, user.getId());
9	param.put("nilai_ujian", nilaiUjian);
10	ServiceGenerator.createService(EndpointAPI.class,
11	user.getApiToken()).saveNilai(param).enqueue(new
12	Callback<GetSaveResponse>() {
13	@Override
14	public void onResponse(Call<GetSaveResponse> call,
15	Response<GetSaveResponse> response) {
16	view.dismissProgress();
17	if (response.isSuccessful()) {
18	if (response.code() == 200) {
19	if (response.body().getSuccess() {
20	if
21	(PrefUtil.getIsPenguji(context).equalsIgnoreCase("1")) {
22	PrefUtil.setIsSubmitNilai(context,
23	PrefUtil.getIdSkripsiTemp(context), true);
24	PenilaianHelper.insertStatusKelulusan(context,
25	response.body().getData());
26	} else {
27	if
28	(PrefUtil.getAccessPenilaian(context,
29	PrefUtil.getIdUjianTemp(context)) {
30	PrefUtil.setIsSubmitNilai(context,
31	PrefUtil.getIdSkripsiTemp(context), true);
32	}
33	}
34	view.submitSuccess();
35	}
36	}
37	}
38	}
39	}
40	
41	
42	}
43	}
44	}
45	
46	@Override
47	public void onFailure(Call<GetSaveResponse> call,
48	Throwable t) {
49	view.dismissProgress();
50	view.showSubmitError(t.getMessage());
51	}
52	});
53	}

### 7.1.3.3 Implementasi Merekap Nilai

Bagian ini membahas Implementasi dari salah satu fungsional aplikasi yaitu merekap nilai. Implementasi ini di berdasarkan rancangan *class diagram*, *sequence diagram* dan algoritme merekap nilai yang sudah dibuat sebelumnya.



Implementasi merekap nilai terdapat di *class* RekapPresenter sebagaimana yang ditunjukkan pada Tabel 7.7.

Implementasi merekap nilai terdiri dari dua fungsi yaitu menghitung rata-rata dan memeriksa *gap*. Fungsi menghitung rata-rata dapat dilihat pada baris 1 - 72. Pada baris tersebut merupakan salah satu fungsi menghitung rata-rata pada pembimbing. Sedangkan, untuk menghitung rata-rata pada penguji secara implementasi sama dengan rata-rata pembimbing, yang membedakan adalah rumus perhitungan rata-rata. Kemudian, untuk fungsi memeriksa *gap* terdapat pada baris 74-98. Fungsi bertujuan untuk melakukan proses memeriksa *gap* antara penguji dan pembimbing.

**Tabel 7.7 Implementasi kode program merekap nilai**

No	RekapPresenter.class
1	public void getRataRataPembimbing() {
2	List<RataRataKomponen> rataRataKomponens =
3	PenilaianHelper.getRataRataPembimbing(context);
4	double ratarataKeseluruhan = 0;
5	try {
6	if (rataRataKomponens.size()>0){
7	String peranTemp = rataRataKomponens.get(0).peran;
8	double rataRataPembimbing = 0;
9	int jumlahPembimbing = 1;
10	for (int i = 0; i < rataRataKomponens.size(); i++) {
11	if
12	(rataRataKomponens.get(i).peran.equalsIgnoreCase(peranTemp)) {
13	switch (rataRataKomponens.get(i).id) {
14	case "1":
15	rataRataPembimbing +=
16	rataRataKomponens.get(i).avg;
17	break;
18	case "2":
19	rataRataPembimbing +=
20	rataRataKomponens.get(i).avg;
21	break;
22	case "3":
23	rataRataPembimbing =
24	rataRataPembimbing + (0.5 * rataRataKomponens.get(i).avg);
25	break;
26	case "4":
27	rataRataPembimbing +=
28	rataRataKomponens.get(i).avg;
29	break;
30	case "5":
31	rataRataPembimbing += (1.5 *
32	rataRataKomponens.get(i).avg);
33	break;
34	}
35	if (i == rataRataKomponens.size() - 1) {
36	if
37	(rataRataKomponens.get(i).isHadir.equalsIgnoreCase("1")) {
38	rataRataPembimbing =
39	((rataRataPembimbing / 5) * 100) / 7;
40	} else {
41	rataRataPembimbing =
42	((rataRataPembimbing / 2) * 100) / 7;
43	}
44	ratarataKeseluruhan += rataRataPembimbing;
45	rataRataPembimbing = 0;
46	ratarataKeseluruhan = ratarataKeseluruhan/
47	jumlahPembimbing;

Tabel 7.7 Implementasi kode program merekap nilai (Lanjutan)

```

48         double subtotal = (ratarataKeseluruhan *
49         55) / 100;
50         DecimalFormat decimalFormat = new
51         DecimalFormat("#0.0");
52
53         view.setRataRataPembimbing(decimalFormat.format(ratarataKeseluruha
54         n),
55         decimalFormat.format(subtotal));
56     }
57     } else {
58         peranTemp = rataRataKomponens.get(i).peran;
59         rataRataPembimbing = ((rataRataPembimbing / 5)
60         * 100) / 7;
61         ratarataKeseluruhan += rataRataPembimbing;
62         rataRataPembimbing = 0;
63         jumlahPembimbing += 1;
64         i -= 1;
65     }
66 }
67 }
68 }catch (Exception e){
69 }
70 }
71 }
72 }
73 }
74 public void compare() {
75     String peran[][] = {"pembimbing1", "pembimbing2"},
76     {"penguji1", "penguji2"},
77     {"pembimbing1", "penguji1"}, {"pembimbing1",
78     "penguji2"}, {"pembimbing2", "penguji1"},
79     {"pembimbing2", "penguji2"};
80     for (int i = 0; i < peran.length; i++) {
81         List<Majelis> penguji =
82         MajelisHelper.getMajelisByPeran(context, peran[i][0],
83         peran[i][1]);
84         List<CompareNilai> nilai =
85         PenilaianHelper.getPenilaian(context, penguji.get(0).getId(),
86         penguji.get(1).getId());
87
88         for (int j = 0; j < nilai.size(); j++) {
89             int gap = nilai.get(j).penilai1 -
90             nilai.get(j).penilai2;
91             if (Math.abs(gap) >= 2) {
92                 this.gap.put(nilai.get(j).indikatorId,
93                 nilai.get(j).idKomponen);
94             }
95         }
96     }
97 }
98 }

```

#### 7.1.3.4 Implementasi Membuat Berita Acara

Bagian ini membahas Implementasi dari salah satu fungsional aplikasi yaitu membuat berita acara. Implementasi ini di berdasarkan rancangan *class diagram*, *sequence diagram* dan algoritme membuat berita acara yang sudah dibuat sebelumnya. Implementasi membuat berita terdapat di *class* RekapPresente sebagaimana yang ditunjukkan pada Tabel 7.8. Implementasi membuat berita acara terdiri dari dua fungsi, yaitu fungsi untuk memeriksa kelengkapan berita

acara dan mengirim berita acara ke *web service*. Fungsi memeriksa berita acara terdapat pada baris 1-31 dan fungsi mengirim berita acara terdapat pada baris 33-68.

**Tabel 7.8 Implementasi kode program membuat berita acara**

No	RekapPresenter.class
1	@Override
2	public void validation(int s) {
3	List<Majelis> majelis =
4	MajelisHelper.getMajelisIsHadir(context);
5	boolean isComplete = true;
6	for (int i = 0; i < majelis.size(); i++) {
7	List<Nilai> nilaiList =
8	PenilaianHelper.getPenilaian(context, majelis.get(i).getId());
9	if (nilaiList.size() > 0) {
10	isComplete = false;
11	break;
12	}
13	}
14	if (!isComplete    !isGap    s == 0) {
15	if (isComplete) {
16	view.validationFalse("Penilaian Belum Lengkap");
17	} else {
18	if (this.isGap) {
19	view.validationFalse("Masih Terdapat GAP Nilai");
20	} else {
21	if (s == 0) {
22	view.validationFalse("Anda Belum Menentukan
23	Status Kelulusan Mahasiswa");
24	} else {
25	view.validationSuccess();
26	}
27	}
28	}
29	}
30	}
31	
32	@Override
33	public void submitFinal(String catatan, String predikat, String
34	na) {
35	view.showProgress();
36	StatusKelulusan statusKelulusan =
37	PenilaianHelper.getStatusKelulusan(context, predikat);
38	HashMap<String, String> param = new HashMap<>();
39	param.put("id_ujian", PrefUtil.getIdUjianTemp(context));
40	param.put("id_status_kelulusan", statusKelulusan.getId());
41	param.put("catatan", catatan);
42	param.put("nilai_akhir", na);
43	ServiceGenerator.createService(EndpointAPI.class,
44	user.getApiToken()).submitBeritaAcara(param)
45	.enqueue(new Callback<GetBeritaAcaraResponse>() {
46	@Override
47	public void
48	onResponse(Call<GetBeritaAcaraResponse> call,
49	Response<GetBeritaAcaraResponse> response) {
50	view.dismissProgress();
51	if (response.code() == 200) {
52	view.submitSuccess();
53	deleteDataTemp();
54	} else {
55	view.showFailMessage(response.message());
56	}
57	}
58	}

**Tabel 7.8 Implementasi kode program membuat berita acara (Lanjutan)**

59	
60	@Override
61	public void onFailure(Call<GetBeritaAcaraResponse>
62	call, Throwable t) {
63	view.showFailMessage(t.getMessage());
64	
65	}
66	});
67	}

**7.1.4 Implementasi Web Service**

Bagian ini membahas mengenai implementasi dari *web service*. Bagian web service hanya menjelaskan bagian yang mendukung pertukaran data pada fungsional memverifikasi kehadiran, memberikan penilaian dan membuat berita ac. Fungsional tersebut ditangani oleh satu kelas, yaitu PenilaianController. Berikut Tabel 7.9 yang merupakan implementasi dari *web service* penilaian dan rekap nilai.

**Tabel 7.9 Implementasi Web Service**

No	PenilaianController.php
1	public function simpanNilai(Request \$request){
2	if(\$this->repos->checkStatusUjian(\$request->
3	id_ujian)==1){
4	if(\$this->repos->checkKehadiran(\$request->
5	id_ujian,\$request["nilai_ujian"][0]['id_penilai'])!=null){
6	\$insert=\$this->repos->
7	prosesInsertNilai(\$request);
8	
9	if(\$this->repos->
10	checkKetua(\$request["nilai_ujian"][0]['id_penilai'])==0){
11	if(\$insert){
12	\$this-> kirimNilai(\$request,"Anda Tela
13	Menerima Penilaian Dari ");
14	return response()->json(['success'=>
15	'true', 'message'=>'Nilai berhasil disimpan','data'=>[]],200);
16	}else{
17	return response()->json(['success'=>
18	'false', 'message'=>'Nilai gagal disimpan','data'=>[]],200);
19	}
20	}else{
21	\$list_status=\$this->repos->
22	getStatusPenilaian();
23	return response()->json(['success'=>
24	'true', 'message'=>'Nilai berhasil
25	disimpan','data'=>\$list_status],200);
26	}
27	}else{
28	return response()->json(['success'=> 'false',
29	'message'=>'Nilai Tidak Dapat Disimpan','data'=>[]],200);
30	}
31	
32	}else{
33	if(\$this->repos->
34	checkPeran(\$request["nilai_ujian"][0]['id_penilai'],\$request->
35	id_ujian)==0){
36	\$list['id_ujian']=\$request->id_ujian;
37	\$list['id_skripsi']=\$request->id_skripsi;
38	
39	\$i=0;
40	foreach(\$request["nilai_ujian"] as



Tabel 7.9 Implementasi Web Service (Lanjutan)

No	PenilaianController.php
41	<code>\$nilaiData){</code>
42	
43	<code>if(\$request["nilai_ujian"][\$i]['id_komponen']==1   </code>
44	<code>\$request["nilai_ujian"][\$i]['id_komponen']==2){</code>
45	<code>    \$list['nilai_ujian'][\$i++]=\$nilaiData;</code>
46	
47	<code>    }</code>
48	<code>    }</code>
49	<code>    \$insert=\$this-&gt;repos-</code>
50	<code>&gt;prosesInsertNilai(\$list);</code>
51	<code>    if(\$insert){</code>
53	<code>        return response()-&gt;json(['success'=&gt;</code>
54	<code>'true', 'message'=&gt;'Nilai berhasil disimpan','data'=&gt;[]],200);</code>
55	<code>    }else{</code>
56	<code>        return response()-&gt;json(['success'=&gt;</code>
57	<code>'false', 'message'=&gt;'Nilai Tidak Dapat</code>
58	<code>Disimpan','data'=&gt;[]],200);</code>
59	<code>    }</code>
60	<code>    }else{</code>
61	<code>        return response()-&gt;json(['success'=&gt; 'false',</code>
62	<code>'message'=&gt;'Nilai Tidak Dapat Disimpan','data'=&gt;[]],200);</code>
63	
64	<code>    }</code>
65	<code>    }</code>
66	
67	<code>    }</code>
68	
69	<code>    public function kirimNilai(\$request,\$message){</code>
70	<code>        \$dosen= \$this-&gt;repos-&gt;getFirebaseKetua(\$request-</code>
71	<code>&gt;id_ujian);</code>
72	<code>        if(\$dosen!=null){</code>
73	<code>            \$client = new Client();</code>
74	<code>            \$namaPenilai=\$this-&gt;repos-</code>
75	<code>&gt;getNamaPenilai(\$request["nilai_ujian"][0]['id_penilai']);</code>
76	<code>            \$data=array(</code>
77	<code>                "to"=&gt; \$dosen-&gt;firebase_token ,</code>
78	<code>                "data"=&gt;</code>
79	<code>array("jenis_notif"=&gt;2,"id_penilai"=&gt;\$request["nilai_ujian"][0]</code>
80	<code>['id_penilai'],</code>
81	<code>                "nilai_ujian"=&gt;\$request-</code>
82	<code>&gt;nilai_ujian,"detail_notif"=&gt;\$message.\$namaPenilai)</code>
83	<code>            );</code>
84	<code>            \$response = \$client-&gt;request('POST',</code>
85	<code>'https://fcm.googleapis.com/fcm/send', [</code>
86	<code>                'json' =&gt; \$data,</code>
87	<code>                'headers' =&gt; [</code>
88	<code>                    'Authorization' =&gt;</code>
89	<code>"key=AIzaSyCI5BlyFkqLVE4Az-AC24BK01CI6sxjdsY",</code>
90	<code>                    'Content-Type' =&gt; "application/json"</code>
91	<code>                ],</code>
92	<code>                'connect_timeout' =&gt; 300</code>
93	<code>            ]);</code>
94	<code>            \$response=json_decode(\$response-&gt;getBody());</code>
95	
96	<code>        }</code>
97	<code>    }</code>
98	<code>    public function simpanBeritaAcara(Request \$request){</code>
99	<code>        \$insertBeritaAcara=\$this-&gt;repos-</code>
100	<code>&gt;insertBeritaAcara(\$request);</code>
101	<code>        if(\$insertBeritaAcara){</code>
102	<code>            \$this-&gt;kirimNotifFinish(\$request);</code>
103	

Tabel 7.9 Implementasi Web Service (Lanjutan)

No	PenilaianController.php
104	return response()->json(['success'=> 'true',
105	'message'=>'berita acara berhasil disimpan','data'=>[]],200);
106	}else{
107	return response()->json(['success'=> 'false',
108	'message'=>'berita acara gagal disimpan','data'=>[]],200);
109	}
110	}
111	
112	public function kirimNotifFinish(\$request){
113	\$listdosen= \$this->repos->getKehadiran(\$request-
114	>id_ujian);
115	
116	if(\$listdosen!=null){
117	foreach(\$listdosen as \$dosen){
118	\$firebase=\$this->repos-
119	>getFirebaseDosenDirect(\$dosen->id_dosen);
120	\$client = new Client();
121	\$data=array(
122	"to"=> \$firebase->firebase_token ,
123	"data"=>
124	array("jenis_notif"=>4,"detail_notif"=>"Penilaian Ujian Telah
124	Selesai")
125	);
126	\$response = \$client->request('POST',
127	'https://fcm.googleapis.com/fcm/send', [
128	'json' => \$data,
129	'headers' => [
130	'Authorization' =>
131	"key=AiZaSyCI5BlyFkqLVE4Az-AC24BKO1CI6sxjdsY",
132	'Content-Type' =>
133	"application/json"
134	],
135	'connect_timeout' => 300
136	]);
137	\$response=json_decode(\$response->getBody());
138	}
139	}
140	}
141	}

Pada Tabel 7.9, terdapat 4 fungsi, yaitu `simpanNilai`, `kirimNilai`, `simpanBeritaAcara`, dan `kirimNotifFinish`. Fungsi `simpanNilai` bertujuan untuk menangani dalam menyimpan nilai yang diberikan oleh pengguna. Fungsi `simpanNilai` ditunjukkan pada baris 1 – 67. Fungsi `kirimNilai` merupakan fungsi yang menangani pengiriman nilai ke ketua majelis sehingga ketua majelis dapat menerima penilaian secara *real time*. Fungsi tersebut ditunjukkan pada baris 69 - 97 Sehingga ketua majelis dapat melihat hasil rekapitulasi nilai pada aplikasi. Selain itu ada dua fungsi yang digunakan dalam membuat berita acara, yaitu `simpanBeritaAcara` dan `kirimNotifFinish`. Fungsi `simpanBeritaAcara` merupakan fungsi yang digunakan untuk menyimpan berita acara ke *database* utama aplikasi. Lalu, terdapat fungsi `kirimNotifFinish` yang digunakan untuk mengirimkan notifikasi bahwa pengujian telah berakhir dan fungsi tersebut yang membuat halaman penilaian dari aplikasi tertutup atau pengguna tidak dapat melakukan proses penilaian. Fungsi tersebut ditunjukkan pada baris 111 - 140.

### 7.1.5 Implementasi Basis Data

Bagian ini berisi implementasi dari perancangan basis data yang telah dijelaskan pada bab 5. Penjelasan dari implementasi basis data dijelaskan dalam bentuk Data Definition Language (DDL). Berikut adalah penjelasan implementasi basis data:

**Tabel 7.10 Implementasi tabel berita acara**

No	Tabel Berita Acara
1	CREATE TABLE tbs_berita_acara (
2	id int(11) NOT NULL AUTO_INCREMENT,
3	id_ujian int(11) NOT NULL,
4	id_status_kelulusan int(11) NOT NULL,
5	nilai_akhir double NOT NULL,
6	catatan varchar(250) NOT NULL,
7	PRIMARY KEY (id),
8	UNIQUE KEY id_ujian_unik (id_ujian),
9	KEY id_status_kelulusan (id_status_kelulusan),
10	CONSTRAINT tbs_berita_acara_ibfk_1 FOREIGN KEY
11	(id_status_kelulusan) REFERENCES tbs_status_kelulusan (id),
12	CONSTRAINT tbs_berita_acara_ibfk_2 FOREIGN KEY (id_ujian)
13	REFERENCES tbs_info_ujian_skripsi (id)
14	)

Tabel 7.10 merupakan implementasi tabel berita acara dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data berita acara ketika proses ujian telah selesai. Tabel ini memiliki 5 atribut, yaitu id, id\_ujian, id\_status\_kelulusan, nilai\_akhir, catatan. Selain itu juga tabel ini memiliki 1 primary key (id), 1 unique key (id\_ujian) dan 2 foreign key (id\_ujian dan id\_status\_kelulusan).

**Tabel 7.11 Implementasi asal berita acara reschedule**

No	Tabel Berita Acara Reschedule
1	CREATE TABLE `tbs_berita_acara_reschedule` (
2	`id` int(11) NOT NULL AUTO_INCREMENT,
3	`id_ujian` int(11) NOT NULL,
4	`alasan` varchar(100) NOT NULL,
5	`catatan` varchar(500) NOT NULL,
6	`is_acc` tinyint(1) NOT NULL,
7	`created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
8	UPDATE CURRENT_TIMESTAMP,
9	PRIMARY KEY (`id`),
10	UNIQUE KEY `id_ujian_2` (`id_ujian`),
11	CONSTRAINT `tbs_berita_acara_reschedule_ibfk_1` FOREIGN KEY
12	(`id_ujian`) REFERENCES `tbs_info_ujian_skripsi` (`id`)
13	)

Tabel 7.11 merupakan implementasi tabel berita acara reschedule dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data berita acara ketika ujian skripsi harus dilakukan penjadwalan ulang. Tabel ini memiliki 6 atribut, yaitu id, id\_ujian, alasan, catatan, is\_acc, dan created\_at. Selain itu juga tabel ini memiliki 1 primary key (id), 1 unique key (id\_ujian) dan 1 foreign key (id\_ujian).

**Tabel 7.12 implementas tabel bidang dosen**

No	Tabel Bidang Dosen
1	CREATE TABLE `tbs_bidang_dosen` (
2	`id_dosen` int(11) NOT NULL,
3	`id_bidang_penelitian` int(11) NOT NULL,
4	`jenis_bidang` enum('Mayor', 'Minor') NOT NULL,
5	)



**Tabel 7.12 Implementas tabel bidang dosen (Lanjutan)**

No	Tabel Bidang Dosen
6	PRIMARY KEY (`id_dosen`,`id_bidang_penelitian`),
7	KEY `id_bidang_penelitian` (`id_bidang_penelitian`),
8	CONSTRAINT `tbs_bidang_dosen_ibfk_1` FOREIGN KEY (`id_dosen`)
9	REFERENCES `tbs_dosen` (`id`),
10	CONSTRAINT `tbs_bidang_dosen_ibfk_2` FOREIGN KEY
11	(`id_bidang_penelitian`) REFERENCES `tbs_bidang_penelitian` (`id`)
12	)

Tabel 7.12 merupakan implementasi tabel bidang dosen dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data bidang dosen. Tabel ini memiliki 3 atribut, yaitu id\_dosen, id\_bidang\_penelitian, dan jenis\_bidang Selain itu juga tabel ini memiliki 2 primary key (id\_dosen dan id\_bidang\_penelitian) dan 2 foreign key (id\_dosen id\_bidang\_penelitian dan id\_jenis\_bidang\_dosen).

**Tabel 7.13 Implementasi tabel bidang penelitian**

No	Tabel Bidang Penelitian
1	CREATE TABLE `tbs_bidang_penelitian` (
2	`id` int(11) NOT NULL AUTO_INCREMENT,
3	`bidang` varchar(150) NOT NULL,
4	`id_keminaran` int(11) NOT NULL,
5	PRIMARY KEY (`id`),
6	KEY `id_keminaran` (`id_keminaran`),
7	CONSTRAINT `tbs_bidang_penelitian_ibfk_1` FOREIGN KEY
8	(`id_keminaran`) REFERENCES `tbs_keminaran` (`id`)
9	)

Tabel 7.13 merupakan implementasi tabel bidang penelitian dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data bidang penelitian . Tabel ini memiliki 4 atribut, yaitu id, bidang dan id\_keminaran. Selain itu juga tabel ini memiliki 1 primary key (id) dan 1 foreign key (id\_keminaran).

**Tabel 7.14 Implementasi tabel bimbingan dosen**

No	Tabel Bimbingan Dosen
1	CREATE TABLE `tbs_bimbingan_dosen` (
2	`id_dosen` int(11) NOT NULL,
3	`id_mahasiswa` int(11) NOT NULL,
4	`is_pembimbing1` tinyint(1) NOT NULL,
5	PRIMARY KEY (`id_mahasiswa`,`id_dosen`),
6	KEY `id_dosen` (`id_dosen`),
7	KEY `id_mahasiswa` (`id_mahasiswa`),
8	CONSTRAINT `tbs_bimbingan_dosen_ibfk_3` FOREIGN KEY (`id_dosen`)
9	REFERENCES `tbs_dosen` (`id`),
10	CONSTRAINT `tbs_bimbingan_dosen_ibfk_4` FOREIGN KEY
11	(`id_mahasiswa`) REFERENCES `tbs_mahasiswa` (`id`)
12	)

Tabel 7.14 merupakan implementasi tabel bimbingan dosen dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data bimbingan yang dimiliki setiap dosen. Tabel ini memiliki 3 atribut, yaitu id\_dosen, id\_mahasiswa dan is\_pembimbing1. Selain itu juga tabel ini memiliki 2 primary key (id\_dosen dan id\_mahasiswa) dan 2 foreign key (id\_dosen dan id\_mahasiswa).

**Tabel 7.15 Implementasi tabel dosen**

No	Tabel Dosen
1	CREATE TABLE `tbs_dosen` (
2	`id` int(11) NOT NULL,
3	`nik_nip` varchar(25) NOT NULL,
4	`jenis_nomor` enum('NIK','NIP') NOT NULL,
5	`foto` varchar(150) NOT NULL,
6	`nama` varchar(150) NOT NULL,
7	`nohp` varchar(15) NOT NULL,
8	`email` varchar(150) NOT NULL,
9	`password` varchar(60) NOT NULL,
10	`firebase_token` varchar(200) NOT NULL,
11	`api_token` varchar(400) NOT NULL,
12	PRIMARY KEY (`id`),
13	UNIQUE KEY `nik_nip` (`nik_nip`)
14	)

Tabel 7.15 merupakan implementasi tabel dosen dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data dosen. Tabel ini memiliki 10 atribut, yaitu id, nik\_nip, jenis\_nomor, foto, nama, nohp, email, password, firebase\_token dan api\_token. Selain itu juga tabel ini memiliki 1 primary key (id) dan 1 unique key (nik\_nip).

**Tabel 7.16 Implementasi tabel indikator penilaian**

No	Tabel Indikator Penilaian
1	CREATE TABLE tbs_indikator_penilaian (
2	id int(11) NOT NULL AUTO_INCREMENT,
3	id_komponen int(11) NOT NULL,
4	indikator varchar(150) NOT NULL,
5	keterangan varchar(300) NOT NULL,
6	PRIMARY KEY ( id ),
7	KEY id_komponen ( id_komponen ),
8	CONSTRAINT tbs_indikator_penilaian_ibfk_1 FOREIGN KEY (
9	id_komponen ) REFERENCES tbs_komponen_penilaian ( id )
10	)

Tabel 7.16 merupakan implementasi tabel indikator penilaian dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data indikator penilaian. Tabel ini memiliki 4 atribut, yaitu id, id\_komponen, indikator dan keterangan. Selain itu juga tabel ini memiliki 1 primary key (id) dan 1 foreign key (id\_komponen).

**Tabel 7.17 Implementasi tabel info ujian skripsi**

No	Tabel Info Ujian Skripsi
1	CREATE TABLE `tbs_info_ujian_skripsi` (
2	`id` int(11) NOT NULL AUTO_INCREMENT,
3	`id_mahasiswa` int(11) NOT NULL,
4	`tanggal` date NOT NULL,
5	`jam` time NOT NULL,
6	`ruang` varchar(50) NOT NULL,
7	`status` int(1) NOT NULL,
8	PRIMARY KEY (`id`),
9	KEY `id_mahasiswa` (`id_mahasiswa`),
10	KEY `status` (`status`),
11	CONSTRAINT `tbs_info_ujian_skripsi_ibfk_1` FOREIGN KEY (`status`)
12	REFERENCES `tbs_status_ujian` (`id`),
13	CONSTRAINT `tbs_info_ujian_skripsi_ibfk_2` FOREIGN KEY
14	(`id_mahasiswa`) REFERENCES `tbs_mahasiswa` (`id`)
15	)

Tabel 7.17 merupakan implementasi tabel info ujian skripsi dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk



menyimpan data info ujian skripsi. Tabel ini memiliki 6 atribut, yaitu id, id\_mahasiswa, tanggal, jam, ruang dan status. Selain itu juga tabel ini memiliki 1 primary key (id) dan 1 foreign key (id\_mahasiswa).

**Tabel 7.18 Implementasi tabel jadwal dosen**

No	Tabel Jadwal Dosen
1	CREATE TABLE `tbs_jadwal_dosen` (
2	`info_ujian_id` int(11) NOT NULL,
3	`id_dosen` int(11) NOT NULL,
4	`is_penguji` tinyint(1) NOT NULL,
5	`is_ketua` tinyint(1) NOT NULL,
6	`is_hadir` tinyint(1) DEFAULT NULL,
7	PRIMARY KEY (`info_ujian_id`,`id_dosen`) USING BTREE,
8	KEY `id_info_ujian` (`info_ujian_id`),
9	KEY `id_dosen` (`id_dosen`),
10	CONSTRAINT `tbs_jadwal_dosen_ibfk_2` FOREIGN KEY (`info_ujian_id`)
11	REFERENCES `tbs_info_ujian_skripsi` (`id`) ON DELETE CASCADE ON
12	UPDATE CASCADE,
13	CONSTRAINT `tbs_jadwal_dosen_ibfk_4` FOREIGN KEY (`id_dosen`)
14	REFERENCES `tbs_dosen` (`id`)
15	)
16	)

Tabel 7.18 merupakan implementasi tabel jadwal dosen dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data jadwal dosen. Tabel ini memiliki 5 atribut, yaitu info\_ujian\_id, id\_dosen, is\_penguji, is\_ketua dan is\_hadir. Selain itu juga tabel ini memiliki 2 primary key (info\_ujian\_id dan id\_dosen) dan 2 foreign key (info\_ujian\_id dan id\_dosen).

**Tabel 7.19 Implementasi tabel jurusan**

No	Tabel Jurusan
1	CREATE TABLE tbs_jurusan (
2	id int(11) NOT NULL AUTO_INCREMENT,
3	jurusan varchar(50) NOT NULL,
4	PRIMARY KEY ( id )
5	)

Tabel 7.19 merupakan implementasi tabel jurusan dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data jurusan. Tabel ini memiliki 2 atribut, yaitu id dan jurusan. Selain itu juga tabel ini memiliki 1 primary key, yaitu id.

**Tabel 7.20 Implementasi tabel keminatan**

No	Tabel Keminatan
1	CREATE TABLE tbs_keminatan (
2	id int(11) NOT NULL,
3	keminatan varchar(50) NOT NULL,
4	PRIMARY KEY ( id )
5	)

Tabel 7.20 merupakan implementasi tabel keminatan dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data keminatan. Tabel ini memiliki 2 atribut, yaitu id, dan keminatan. Selain itu juga tabel ini memiliki 1 primary key, yaitu id.

**Tabel 7.21 Implementasi tabel komponen penilaian**

No	Tabel Komponen Penilaian
1	CREATE TABLE tbs_komponen_penilaian (
2	id int(11) NOT NULL AUTO_INCREMENT,
3	komponen_penilaian varchar(100) NOT NULL,
4	PRIMARY KEY ( id )
5	)

Tabel 7.21 merupakan implementasi tabel komponen penilaian dengan menggunakan Data Definition Language (DDL). Tabel ini berfungsi untuk menyimpan data komponen penilaianskripsi. Tabel ini memiliki 2 atribut, yaitu id, dan komponen\_penilaian. Selain itu juga tabel ini memiliki 1 primary key, yaitu id.

**Tabel 7.22 Implementasi tabel mahasiswa**

No	Tabel Mahasiswa
1	CREATE TABLE tbs_mahasiswa (
2	id int(11) NOT NULL,
3	nim varchar(20) NOT NULL,
4	nama varchar(150) NOT NULL,
5	foto varchar(150) NOT NULL,
6	id_prodi int(11) NOT NULL,
7	email varchar(150) NOT NULL,
8	nohp varchar(15) NOT NULL,
9	PRIMARY KEY ( id ),
10	UNIQUE KEY nim ( nim ),
11	KEY id_prodi ( id_prodi ),
12	CONSTRAINT tbs_mahasiswa_ibfk_1 FOREIGN KEY ( id_prodi )
13	REFERENCES tbs_prodi ( id )
14	)

Tabel 7.22 merupakan implementasi tabel mahasiswa dengan menggunakan Data Definition Language (DDL). Tabel mahasiswa berfungsi untuk menyimpan data mahasiswa. Tabel ini memiliki 7 atribut, yaitu id, nim, nama, foto, id\_prodi, emai l dan nohp. Selain itu juga tabel ini memiliki 1 primary key (id), unique key (nim) dan 1 foreign key (id\_prodi).

**Tabel 7.23 Implementasi tabel matkul**

No	Tabel Matkul
1	CREATE TABLE tbs_matkul (
2	id int(11) NOT NULL AUTO_INCREMENT,
3	kode_matkul varchar(10) NOT NULL,
4	matkul varchar(70) NOT NULL,
5	PRIMARY KEY ( id ),
6	UNIQUE KEY kode_matkul ( kode_matkul )
7	)

Tabel 7.23 merupakan implementasi tabel matkul dengan menggunakan Data Definition Language (DDL). Tabel matkul berfungsi untuk menyimpan data mata kuliah. Tabel ini memiliki 3 atribut, yaitu id, kode\_matkul dan matkul. Selain itu juga tabel ini memiliki 1 primary key (id) dan unique key (kode\_matkul).

**Tabel 7.24 Implementasi tabel matkul pendukung skripsi**

No	Tabel Matkul Pendukung Skripsi
1	CREATE TABLE tbs_matkul_pendukung_skripsi (
2	id_skripsi int(11) NOT NULL,
3	id_matkul int(11) NOT NULL,
4	PRIMARY KEY ( id_matkul , id_skripsi ) USING BTREE,

**Tabel 7.24 Implementasi tabel matkul pendukung skripsi (Lanjutan)**

No	Tabel Matkul Pendukung Skripsi
6	KEY id_skripsi ( id_skripsi ),
7	KEY id_matkul ( id_matkul ),
8	CONSTRAINT tbs_matkul_pendukung_skripsi_ibfk_1 FOREIGN KEY ( id_skripsi )
9	REFERENCES tbs_skripsi ( id ),
10	CONSTRAINT tbs_matkul_pendukung_skripsi_ibfk_2 FOREIGN KEY ( id_matkul )
11	REFERENCES tbs_matkul ( id )
12	)

Tabel 7.24 merupakan implementasi tabel matkul pendukung skripsi dengan menggunakan Data Definition Language (DDL). Tabel matkul pendukung skripsi berfungsi untuk menyimpan data mata kuliah pendukung skripsi. Tabel ini memiliki 2 atribut, yaitu id\_skripsi dan id\_matkul. Selain itu juga tabel ini memiliki 2 primary key dan 2 foreign key, yaitu id\_skripsi dan id\_matkul.

**Tabel 7.25 Implementasi tabel nilai semhas**

No	Tabel Nilai Semhas
.	
1	CREATE TABLE tbs_nilai_semhas (
2	id_skripsi int(11) NOT NULL,
3	id_penilai int(11) NOT NULL,
4	nilai double NOT NULL,
5	PRIMARY KEY ( id_skripsi , id_penilai ),
6	KEY id_skripsi ( id_skripsi ),
7	KEY id_penilai ( id_penilai ),
8	CONSTRAINT tbs_nilai_semhas_ibfk_1 FOREIGN KEY ( id_skripsi )
9	REFERENCES tbs_skripsi ( id ),
10	CONSTRAINT tbs_nilai_semhas_ibfk_2 FOREIGN KEY ( id_penilai )
11	REFERENCES tbs_dosen ( id )
12	)

Tabel 7.25 merupakan implementasi tabel nilai semhas dengan menggunakan Data Definition Language (DDL). Tabel nilai semhas berfungsi untuk menyimpan data nilai seminar hasil. Tabel ini memiliki 3 atribut, yaitu id\_skripsi, id\_penilai dan nilai. Selain itu juga tabel ini memiliki 2 primary key dan 2 foreign key, yaitu id\_skripsi dan id\_penilai.

**Tabel 7.26 Implementasi tabel nilai skripsi**

No	Tabel Nilai Skripsi
.	
1	CREATE TABLE tbs_nilai_skripsi (
2	id_skripsi int(11) NOT NULL,
3	id_penilai int(11) NOT NULL,
4	id_indikator int(11) NOT NULL,
5	nilai int(11) NOT NULL,
6	PRIMARY KEY ( id_skripsi , id_penilai , id_indikator ),
7	KEY id_skripsi ( id_skripsi ),
8	KEY id_penilai ( id_penilai ),
9	KEY id_indikator ( id_indikator ),
10	CONSTRAINT tbs_nilai_skripsi_ibfk_2 FOREIGN KEY (
11	id_skripsi ) REFERENCES tbs_skripsi ( id ),
12	CONSTRAINT tbs_nilai_skripsi_ibfk_3 FOREIGN KEY (
13	id_penilai ) REFERENCES tbs_dosen ( id ),
14	CONSTRAINT tbs_nilai_skripsi_ibfk_4 FOREIGN KEY (
15	id_indikator ) REFERENCES tbs_indikator_penilaian ( id )
16	)
17	



Tabel 7.26 merupakan implementasi tabel nilai skripsi dengan menggunakan Data Definition Language (DDL). Tabel nilai skripsi berfungsi untuk menyimpan data nilai skripsi. Tabel ini memiliki 4 atribut, yaitu id\_skripsi, id\_penilai, id\_indikator dan nilai. Selain itu juga tabel ini memiliki 3 primary key dan 3 foreign key, yaitu id\_skripsi, id\_penilai dan id\_indikator.

**Tabel 7.27 Implementasi tabel pengajuan penguji pengganti**

No	Tabel Pengajuan Penguji Pengganti
1	CREATE TABLE tbs_pengajuan_penguji_pengganti (
2	id_ujian int(11) NOT NULL,
3	id_dosen_pengganti int(11) NOT NULL,
4	id_dosen_saat_ini int(11) NOT NULL,
5	status_confirm int(1) NOT NULL DEFAULT '0',
6	PRIMARY KEY ( id_ujian ),
7	KEY tbs_pengajuan_penguji_pengganti_ibfk_1 ( id_dosen_pengganti
8	),
9	KEY tbs_pengajuan_penguji_pengganti_ibfk_2 ( id_ujian ),
10	KEY id_dosen_saat_ini ( id_dosen_saat_ini ),
11	CONSTRAINT tbs_pengajuan_penguji_pengganti_ibfk_1 FOREIGN KEY (
12	id_dosen_pengganti ) REFERENCES tbs_dosen ( id ),
13	CONSTRAINT tbs_pengajuan_penguji_pengganti_ibfk_2 FOREIGN KEY (
14	id_ujian ) REFERENCES tbs_info_ujian_skripsi ( id ),
15	CONSTRAINT tbs_pengajuan_penguji_pengganti_ibfk_3 FOREIGN KEY (
16	id_dosen_saat_ini ) REFERENCES tbs_dosen ( id )
17	)

Tabel 7.27 merupakan implementasi tabel pengajuan penguji pengganti dengan menggunakan Data Definition Language (DDL). Tabel pengajuan penguji pengganti berfungsi untuk menyimpan data pengajuan penguji pengganti. Tabel ini memiliki 4 atribut, yaitu id\_ujian, id\_dosen\_pengganti, id\_dosen\_saat\_ini dan status\_konfirm. Selain itu juga tabel ini memiliki 1 primary key (id\_ujian) dan 2 foreign key (id\_dosen\_pengganti dan id\_dosen\_saat\_ini)

**Tabel 7.28 Implementasi tabel prodi**

No	Tabel Prodi
1	CREATE TABLE `tbs_prodi` (
2	`id` int(11) NOT NULL AUTO_INCREMENT,
3	`id_jurusan` int(11) NOT NULL,
4	`jenjang` enum('S1') NOT NULL,
5	`prodi` varchar(50) NOT NULL,
6	PRIMARY KEY (`id`),
7	KEY `id_jurusan` (`id_jurusan`),
8	CONSTRAINT `tbs_prodi_ibfk_2` FOREIGN KEY (`id_jurusan`)
9	REFERENCES `tbs_jurusan` (`id`)
10	)

Tabel 7.28 merupakan implementasi tabel prodi dengan menggunakan Data Definition Language (DDL). Tabel prodi berfungsi untuk menyimpan data program studi. Tabel ini memiliki 4 atribut, yaitu id, id\_jurusan, jenjang dan prodi. Selain itu juga tabel ini memiliki 1 primary key (id) dan 1 foreign key (id\_jurusan).

**Tabel 7.29 Implementasi tabel skripsi**

No	Tabel Skripsi
1	CREATE TABLE `tbs_skripsi` (
2	`id` int(11) NOT NULL AUTO_INCREMENT,
3	`id_mahasiswa` int(11) NOT NULL,
4	`id_bidang_penelitian` int(11) NOT NULL,
5	`id_tipe_penelitian` int(11) NOT NULL,
6	`asal_judul` enum('Mahasiswa','Dosen') NOT NULL,



**Tabel 7.29 Implementasi tabel skripsi (Lanjutan)**

No	Tabel Skripsi
8	<code>`judul_skripsi` varchar(800) NOT NULL,</code>
9	<code>`dokumen` varchar(100) NOT NULL,</code>
10	<code>PRIMARY KEY (`id`),</code>
11	<code>KEY `tbs_skripsi_ibfk_5` (`id_bidang_penelitian`),</code>
12	<code>KEY `id_tipe_penelitian` (`id_tipe_penelitian`),</code>
13	<code>KEY `id_mahasiswa` (`id_mahasiswa`),</code>
14	<code>CONSTRAINT `tbs_skripsi_ibfk_5` FOREIGN KEY</code>
15	<code>(`id_bidang_penelitian`) REFERENCES `tbs_bidang_penelitian` (`id`),</code>
16	<code>CONSTRAINT `tbs_skripsi_ibfk_6` FOREIGN KEY (`id_tipe_penelitian`)</code>
17	<code>REFERENCES `tbs_tipe_penelitian` (`id`),</code>
18	<code>CONSTRAINT `tbs_skripsi_ibfk_8` FOREIGN KEY (`id_mahasiswa`)</code>
	<code>REFERENCES `tbs_mahasiswa` (`id`)</code>
	<code>)</code>

Tabel 7.29 merupakan implementasi tabel skripsi dengan menggunakan Data Definition Language (DDL). Tabel skripsi berfungsi untuk menyimpan data skripsi mahasiswa. Tabel ini memiliki 7 atribut, yaitu id, id\_mahasiswa, id\_bidang\_penelitian, id\_tipe\_penelitian, asal\_judul, judul skripsi dan dokumen. Selain itu juga tabel ini memiliki 1 primary key (id) dan 4 foreign key (id\_mahasiswa, id\_bidang\_penelitian dan id\_tipe\_penelitian).

**Tabel 7.30 Implementasi tabel status kelulusan**

No	Tabel Status Kelulusan
1	<code>CREATE TABLE `tbs_status_kelulusan` (</code>
2	<code>  `id` int(11) NOT NULL,</code>
3	<code>  `status_kelulusan` varchar(50) NOT</code>
4	<code>  NULL,</code>
5	<code>  PRIMARY KEY (`id`)</code>
6	<code>)</code>

Tabel 7.30 merupakan implementasi tabel status kelulusan dengan menggunakan Data Definition Language (DDL). Tabel status kelulusan berfungsi untuk menyimpan data status kelulusan. Tabel ini memiliki 2 atribut, yaitu id, dan status\_kelulusan. Selain itu juga tabel ini memiliki 1 primary key, yaitu id.

**Tabel 7.30 Implementasi tabel status ujian**

No	Tabel Status Ujian
1	<code>CREATE TABLE `tbs_status_ujian` (</code>
2	<code>  `id` int(11) NOT NULL AUTO INCREMENT,</code>
3	<code>  `status` varchar(30) NOT NULL,</code>
4	<code>  PRIMARY KEY (`id`)</code>
5	<code>)</code>

Tabel 7.31 merupakan implementasi tabel status ujian dengan menggunakan Data Definition Language (DDL). Tabel status ujian berfungsi untuk menyimpan data status ujian. Tabel ini memiliki 2 atribut, yaitu id, dan status. Selain itu juga tabel ini memiliki 1 primary key, yaitu id.

**Tabel 7.31 Implementasi tabel tipe penelitian**

No	Tabel Tipe Penelitian
1	<code>CREATE TABLE `tbs_tipe_penelitian` (</code>
2	<code>  `id` int(11) NOT NULL AUTO INCREMENT,</code>
3	<code>  `tipe_penelitian` varchar(50) NOT NULL,</code>
4	<code>  PRIMARY KEY (`id`)</code>
5	<code>)</code>

Tabel 7.32 merupakan implementasi tabel status ujian dengan menggunakan Data Definition Language (DDL). Tabel status ujian berfungsi untuk menyimpan

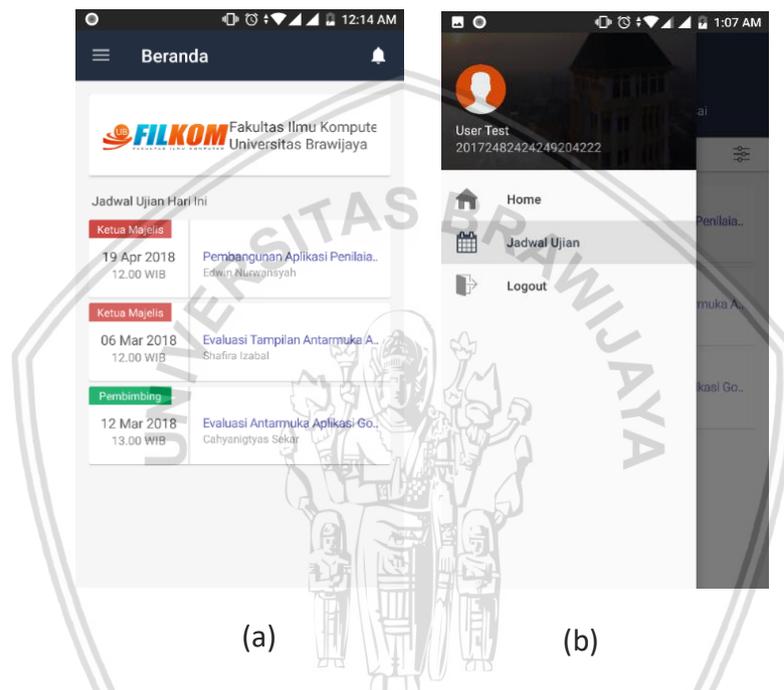


data status ujian. Tabel ini memiliki 2 atribut, yaitu id, dan tipe\_penelitian. Selain itu juga tabel ini memiliki 1 primary key, yaitu id.

### 7.1.6 Implementasi Antarmuka

Bagian ini membahas implementasi antarmuka aplikasi, berdasarkan perancangan antarmuka yang telah dibahas di bab 5. Penjelasan implementasi antarmuka terdiri dari halaman login, halaman beranda, halaman jadwal ujian, halaman rincian ujian, halaman verifikasi, halaman penilaian dan halaman rekap nilai.

#### 7.1.6.1 Halaman Beranda

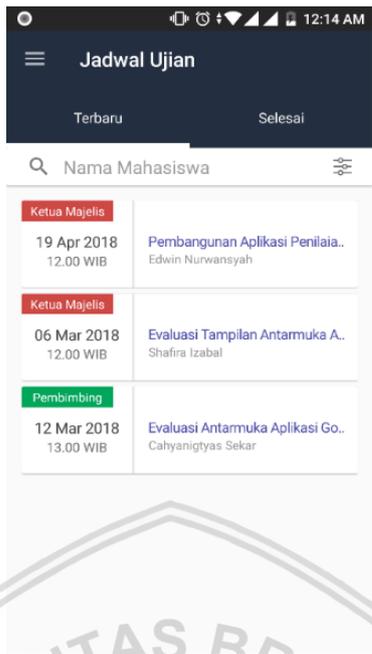


**Gambar 7.1 Implementasi antarmuka halaman beranda: (a) halaman beranda; (b)halaman beranda bagian menu *navigation drawer***

Gambar 7.1 merupakan implementasi dari perancangan antarmuka halaman beranda. Halaman ini menampilkan teks Fakultas Ilmu Komputer beserta logo dan juga jadwal ujian pada hari tersebut. Selain itu juga, halaman tersebut terdapat *navigation drawer* dan menu notifikasi. Navigation drawer terdiri dari menu home, jadwal ujian dan logout.

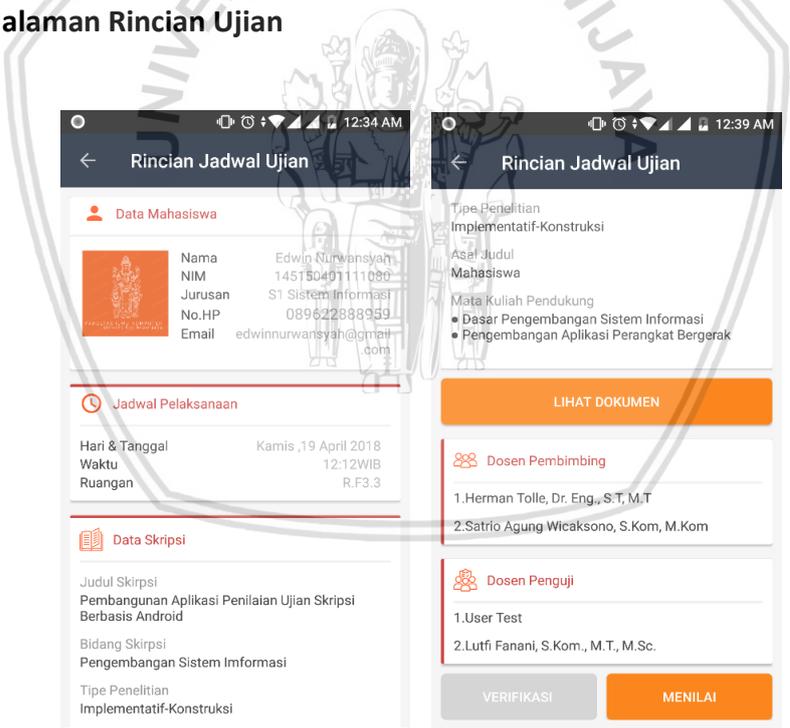
#### 7.1.6.2 Halaman Jadwal Ujian

Gambar 7.2 merupakan hasil implementasi dari perancangan antarmuka halaman jadwal ujian. Halaman ini menampilkan data jadwal ujian pengguna. Pada halaman ini terdapat fitur search dan fitur filter berdasarkan peran. Selain itu juga halaman ini tidak hanya menampilkan jadwal ujian terbaru tetapi juga jadwal ujian yang sudah dilaksanakan.



Gambar 7.2 Implementasi antarmuka halaman jadwal ujian

### 7.1.6.3 Halaman Rincian Ujian

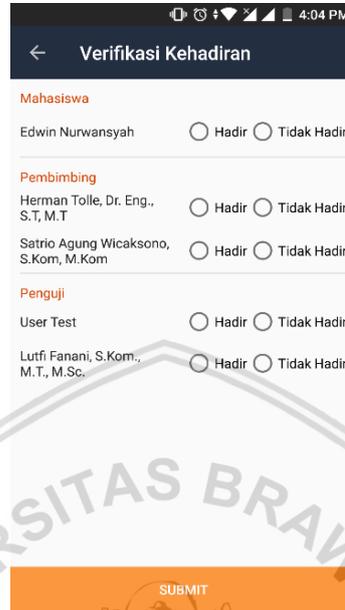


Gambar 7.3 Implementasi antarmuka halaman rincian ujian

Gambar 7.3 merupakan implementasi dari perancangan antarmuka halaman rincian. Halaman ini menampilkan data mahasiswa, jadwal pelaksanaan, data

skripsi, data pembimbing dan data penguji. Halaman ini juga menjadi halaman rincian ujian untuk jadwal terbaru ataupun jadwal yang sudah terlaksana.

#### 7.1.6.4 Halaman Verifikasi



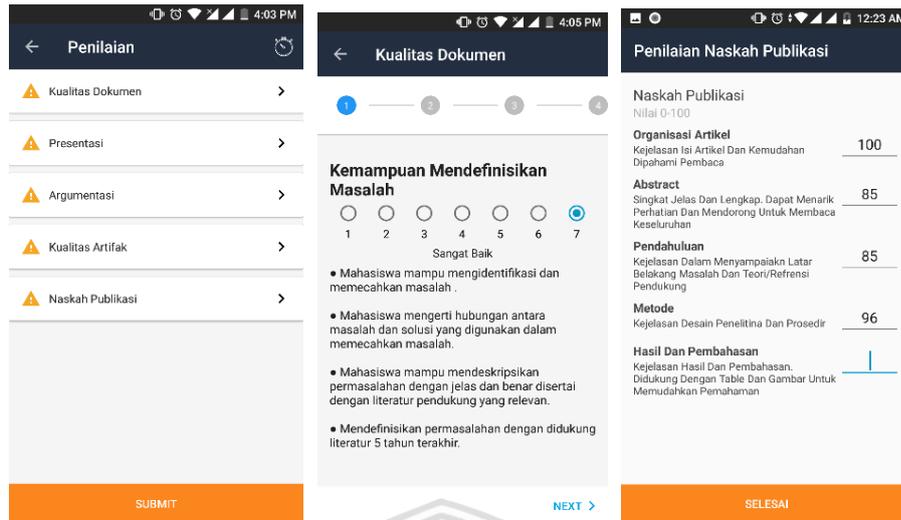
**Gambar 7.4 Implementasi antarmuka halaman verifikasi**

Gambar 7.4 merupakan implementasi dari perancangan antarmuka memverifikasi kehadiran. Halaman ini terdapat radio button untuk memverifikasi kehadiran dari setiap dosen dan juga dilengkapi nama dosen. Halaman verifikasi menjadi halaman pengecekan kehadiran majelis. Penguji ataupun pembimbing tidak akan mendapatkan akses penilaian ketika ketua majelis belum melakukan verifikasi.

#### 7.1.6.5 Halaman Penilaian

Gambar 7.5 merupakan implementasi dari perancangan antarmuka halaman penilaian. Halaman penilaian terdiri dari 3 bagian, yaitu halaman utama, halaman pada komponen ujian (proses pengerjaan skripsi, kualitas dokumen, presentasi, argumentasi, kualitas artefak) dan halaman pada komponen naskah publikasi. Selain itu juga terdapat menu pengaturan waktu yang terdapat di sisi kanan atas pada halaman utama penilaian. Menu tersebut hanya terdapat di ketua majelis.



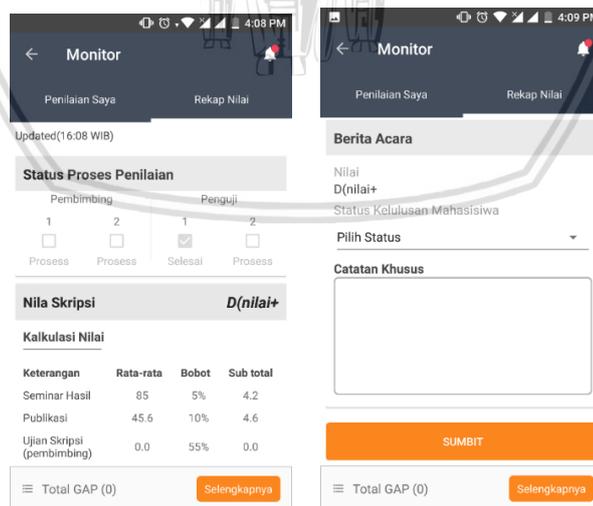


(a) (b) (c)

**Gambar 7.5 Implementasi antarmuka halaman penilaian: (a) halaman utama; (b) halaman komponen ujian; (c) halaman komponen naskah publikasi**

### 7.1.6.6 Halaman Rekap Nilai

Gambar 7.6 merupakan implementasi dari perancangan antarmuka halaman rekap nilai. Halaman ini terdapat dua fungsional, yaitu fungsional merekap nilai dan fungsional berita acara. Halaman ini menjadi halaman dimana ketua majelis dapat merundingkan nilai-nilai yang telah dilakukan oleh majelis. Selain itu juga mengisikan hasil dari ujian skripsi.



**Gambar 7.6 Implementasi antarmuka halaman rekap nilai**

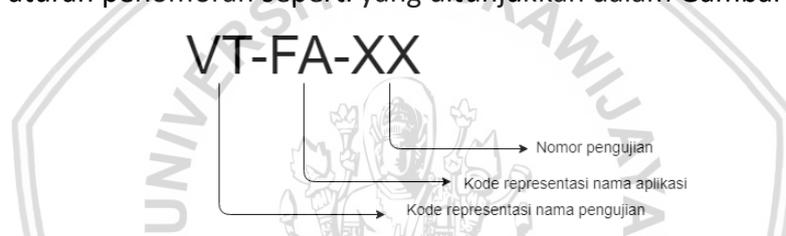


## BAB 8 PENGUJIAN DAN ANALISIS

Bagian ini membahas mengenai proses untuk memastikan perangkat lunak yang telah dibuat sesuai dengan spesifikasi dan persyaratan yang telah identifikasi sebelumnya. Bab ini terdiri dari 2 pembahasan, yaitu pengujian fungsional, dan pengujian non-fungsional.

### 8.1 Pengujian Fungsional

Pengujian fungsional merupakan pengujian yang digunakan untuk menguji fungsi-fungsi khusus dari perangkat lunak yang sudah dibuat dan berfokus pada persyaratan fungsional perangkat lunak. Subbab ini membahas hasil pengujian fungsional terhadap fungsi-fungsi yang dimiliki oleh aplikasi penilaian ujian skripsi. Pengujian ini dilakukan dengan menggunakan pengujian validasi. Pengujian terlebih dahulu membuat kasus uji yang merujuk kepada *use case scenario*. Selanjutnya, hasil pengujian diidentifikasi dengan membandingkan hasil yang telah didefinisikan *use case scenario* dengan hasil yang di dapatkan. Pengujian ini memiliki aturan penomoran seperti yang ditunjukkan dalam Gambar 8.1



Gambar 8.1 Aturan penomoran pengujian validasi

#### 8.1.1 Pengujian Fungsional Login

Pada Tabel 8.1, Tabel 8.2, dan Tabel 8.3 menjelaskan mengenai kasus uji dari pengujian validasi login dengan kode VT-FA-01, VT-FA-02, dan VT-FA-03. Kasus uji VT-FA-01, VT-FA-02, dan VT-FA-03 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-01 Login. Tujuan kasus uji VT-FA-01 untuk memastikan aplikasi dapat melakukan proses login, kasus uji VT-FA-02 untuk memastikan aplikasi menampilkan pesan akun tidak ditemukan, dan kasus uji VT-FA-03 untuk memastikan aplikasi menampilkan pesan error di *username* dan/atau *password*. Berdasarkan pada Tabel 8.1, Tabel 8.2 dan Tabel 8.3 hasil status pengujian validasi untuk login menunjukkan hasil yang *valid*.

Tabel 8.1 Kasus uji login *basic flow*

<b>Kode Pengujian</b>	VT-FA-01
<b>Kode Use Case</b>	UCS-FA-01
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat melakukan proses login
<b>Prosedur Uji</b>	1. Penguji memasukan <i>username</i> dan password 2. Penguji menekan tombol <i>sign in</i>



Tabel 8.1 Kasus uji login *basic flow* (Lanjutan)

Hasil yang Diharapkan	Aplikasi menampilkan halaman beranda.
Hasil Pengujian	Aplikasi berhasil menampilkan halaman beranda.
Status Uji	<i>Valid</i>

Tabel 8.2 Kasus uji login *alternative flow 1*

Kode Pengujian	VT-FA-02
Kode Use Case	UCS-FA-0
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem akan menolak apabila <i>username</i> dan <i>password</i> tidak valid
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji memasukan <i>username</i> dan <i>password</i> yang tidak valid</li> <li>2. Penguji menekan tombol <i>sign in</i></li> </ol>
Hasil yang Diharapkan	Aplikasi menampilkan pesan akun tidak ditemukan.
Hasil Pengujian	Aplikasi berhasil menampilkan pesan akun tidak ditemukan.
Status Uji	<i>Valid</i>

Tabel 8.3 Kasus uji login *alternative flow 2*

Kode Pengujian	VT-FA-03
Kode Use Case	UCS-FA-0
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat dapat melakukan proses login ketika dan/atau password tidak diisi.
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji tidak memasukan <i>username</i> dan/atau <i>password</i></li> <li>2. Penguji menekan tombol <i>sign in</i></li> </ol>
Hasil yang Diharapkan	Aplikasi menampilkan pesan kolom <i>username</i> dan/atau <i>password</i> wajib diisi.
Hasil Pengujian	Aplikasi berhasil menampilkan pesan kolom <i>username</i> dan/atau <i>password</i> wajib diisi.
Status Uji	<i>Valid</i>

### 8.1.2 Pengujian Fungsional Melihat Notifikasi

Pada Tabel 8.4 menjelaskan mengenai kasus uji dari pengujian validasi melihat notifikasi dengan kode VT-FA-04. Kasus uji VT-FA-04 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-02 melihat notifikasi. Tujuan kasus uji VT-FA-04 untuk memastikan aplikasi dapat menampilkan notifikasi. Berdasarkan pada Tabel 8.4 hasil status pengujian validasi untuk melihat notifikasi menunjukkan hasil yang *valid*.

**Tabel 8.4 Kasus uji melihat notifikasi**

<b>Kode Pengujian</b>	VT-FA-04
<b>Kode Use Case</b>	UCS-FA-02
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan daftar notifikasi.
<b>Prosedur Uji</b>	1. Penguji berada di halaman beranda 2. Penguji menekan menu notifikasi di halaman beranda
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan daftar notifikasi.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan daftar notifikasi.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.3 Pengujian Fungsional Melihat Daftar Jadwal Ujian Skripsi

Pada Tabel 8.5, Tabel 8.6, dan Tabel 8.7 menjelaskan mengenai kasus uji dari pengujian validasi melihat daftar jadwal ujian skripsi dengan kode VT-FA-03, VT-FA-04, dan VT-FA-05. Kasus uji kode VT-FA-03, VT-FA-04, dan VT-FA-05 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-03 melihat daftar jadwal ujian skripsi. Tujuan kasus uji VT-FA-03 untuk memastikan aplikasi dapat menampilkan daftar jadwal ujian, kasus uji VT-FA-04 untuk memastikan aplikasi dapat menampilkan daftar jadwal ujian skripsi berdasarkan nama yang dicari, dan VT-FA-05 untuk memastikan aplikasi dapat menampilkan daftar jadwal ujian skripsi sesuai dengan filter peran yang dipilih. Berdasarkan pada Tabel 8.5, Tabel 8.6, dan Tabel 8.7 hasil status pengujian validasi untuk melihat daftar jadwal ujian skripsi menunjukkan hasil yang *valid*.

**Tabel 8.5 Kasus uji melihat daftar jadwal ujian skripsi *basic flow***

<b>Kode Pengujian</b>	VT-FA-04
<b>Kode Use Case</b>	UCS-FA-03
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan daftar jadwal ujian skripsi.
<b>Prosedur Uji</b>	1. Penguji telah berada di halaman beranda 2. Penguji menekan menu jadwal ujian

Tabel 8.5 Kasus uji melihat daftar jadwal ujian skripsi *basic flow* (Lanjutan)

<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan daftar jadwal ujian
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan daftar jadwal ujian.
<b>Status Uji</b>	<i>Valid</i>

Tabel 8.6 Kasus uji melihat daftar jadwal ujian skripsi *alternative flow 1*

<b>Kode Pengujian</b>	VT-FA-05
<b>Kode Use Case</b>	UCS-FA-03
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan daftar jadwal ujian skripsi sesuai dengan nama mahasiswa yang dimasukan dikolom pencarian.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah berada di halaman beranda</li> <li>2. Penguji menekan menu jadwal ujian</li> <li>3. Penguji memasukan nama mahasiswa pada kolom pencarian</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan daftar jadwal ujian sesuai dengan nama yang dicari.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan daftar jadwal ujian sesuai dengan nama yang dicari.
<b>Status Uji</b>	<i>Valid</i>

Tabel 8.7 Kasus uji melihat daftar jadwal ujian skripsi *alternative flow 2*

<b>Kode Pengujian</b>	VT-FA-06
<b>Kode Use Case</b>	UCS-FA-03
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan daftar jadwal ujian skripsi sesuai dengan peran yang di filter.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah berada dihalaman beranda</li> <li>2. Penguji menekan menu jadwal ujian</li> <li>3. Penguji menekan tombol fiter</li> <li>4. Penguji memilih peran</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan daftar jadwal ujian sesuai dengan peran yang difilter.

**Tabel 8.7 Kasus uji melihat daftar jadwal ujian skripsi *alternative flow 2* (Lanjutan)**

<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan daftar jadwal ujian sesuai dengan peran yang difilter.
<b>Status Uji</b>	<i>Valid</i>

#### 8.1.4 Pengujian Fungsional Melihat Rincian Ujian Skripsi

Pada Tabel 8.8 menjelaskan mengenai kasus uji dari pengujian validasi melihat rincian ujian skripsi dengan kode VT-FA-07. Kasus uji VT-FA-07 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-07 melihat daftar rincian ujian skripsi. Tujuan kasus uji VT-FA-07 untuk memastikan aplikasi dapat menampilkan rincian ujian skripsi. Berdasarkan pada Tabel 8.8 hasil status pengujian validasi untuk melihat rincian ujian skripsi menunjukkan hasil yang *valid*.

**Tabel 8.8 kasus uji melihat rincian ujian skripsi**

<b>Kode Pengujian</b>	VT-FA-07
<b>Kode Use Case</b>	UCS-FA-04
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan rincian ujian skripsi.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah berada di halaman beranda</li> <li>2. Penguji menekan menu jadwal ujian</li> <li>3. Penguji memilih jadwal ujian dari daftar jadwal ujian</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan rincian ujian skripsi.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan rincian ujian skripsi.
<b>Status Uji</b>	<i>Valid</i>

#### 8.1.5 Pengujian Fungsional Melihat Dokumen Skripsi

Pada Tabel 8.9 menjelaskan mengenai kasus uji dari pengujian validasi melihat dokumen skripsi dengan kode VT-FA-08. Kasus uji VT-FA-08 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-05 melihat dokumen skripsi. Tujuan kasus uji VT-FA-08 untuk memastikan aplikasi dapat menampilkan dokumen skripsi. Berdasarkan pada Tabel 8.9 hasil status pengujian validasi untuk melihat dokumen skripsi menunjukkan hasil yang *valid*.

**Tabel 8.9 Kasus uji melihat melihat dokumen skripsi**

<b>Kode Pengujian</b>	VT-FA-08
<b>Kode Use Case</b>	UCS-FA-05

Tabel 8.9 Kasus uji melihat melihat dokumen skripsi (Lanjutan)

<b>Tujuan Pengujian</b>	Pengujian untuk memastikan bahwa sistem dapat menampilkan dokumen skripsi.
<b>Prosedur Uji</b>	1. Penguji berada di halaman rincian jadwal ujian 2. Penguji menekan tombol dokumen
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan dokumen skripsi.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan dokumen skripsi.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.6 Pengujian Fungsional Memverifikasi Kehadiran Mahasiswa, Dosen Pembimbing Dan Dosen Penguji

Pada Tabel 8.10 dan Tabel 8.11 menjelaskan mengenai kasus uji dari pengujian validasi memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji dengan kode VT-FA-09 dan VT-FA-10. Kasus uji VT-FA-09 dan VT-FA-10 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-06 memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji. Tujuan kasus uji VT-FA-09 untuk memastikan aplikasi dapat memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji dan VT-FA-10 untuk memastikan aplikasi menampilkan pesan bahwa terdapat data kehadiran yang belum dilengkapi. Berdasarkan pada Tabel 8.10 dan Tabel 8.11 hasil status pengujian validasi untuk memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji menunjukkan hasil *valid*.

Tabel 8.10 Kasus uji memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji *basic flow*

<b>Kode Pengujian</b>	VT-FA-09
<b>Kode Use Case</b>	UCS-FA-06
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji.
<b>Prosedur Uji</b>	1. Penguji telah berada di halaman verifikasi kehadiran 2. Penguji memverifikasi kehadiran 3. Penguji menekan tombol <i>submit</i>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pemberitahuan verifikasi berhasil.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pemberitahuan verifikasi berhasil.
<b>Status Uji</b>	<i>Valid</i>

**Tabel 8.11 Kasus uji memverifikasi kehadiran mahasiswa, dosen pembimbing dan dosen penguji *alternative flow***

<b>Kode Pengujian</b>	VT-FA-10
<b>Kode Use Case</b>	UCS-FA-06
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi tidak melakukan proses verifikasi kehadiran dan menampilkan pesan data kehadiran belum lengkap
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah berada di halaman verifikasi kehadiran</li> <li>2. Penguji mengosongkan data kehadiran</li> <li>3. Penguji menekan tombol <i>submit</i></li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pesan bahwa data kehadiran belum lengkap.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pesan bahwa data kehadiran belum lengkap.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.7 Pengujian Fungsional Memberikan Penilaian

Pada Tabel 8.12 dan Tabel 8.13 menjelaskan mengenai kasus uji dari pengujian validasi memberikan penilaian dengan kode VT-FA-11 dan VT-FA-12. Kasus uji VT-FA-11 dan VT-FA-12 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-07 memberikan penilaian. Tujuan kasus uji VT-FA-11 untuk memastikan aplikasi dapat memberikan penilaian dan VT-FA-12 untuk memastikan aplikasi akan menampilkan peringatan apabila penilaian belum dilengkapi. Berdasarkan pada Tabel 8.12 dan Tabel 8.13 hasil status pengujian validasi untuk memberikan penilaian menunjukkan hasil yang *valid*.

**Tabel 8.12 Kasus uji memberikan penilaian *basic flow***

<b>Kode Pengujian</b>	VT-FA-11
<b>Kode Use Case</b>	UCS-FA-07
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat memberikan penilaian.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji berada di halaman penilaian</li> <li>2. Penguji mengisikan penilaian</li> <li>3. Penguji menekan tombol <i>submit</i>.</li> </ol>

Tabel 8.13 Kasus uji memberikan penilaian *basic flow* (Lanjutan)

<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pemberitahuan data penilaian sudah tersimpan.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pemberitahuan data penilaian sudah tersimpan.
<b>Status Uji</b>	<i>Valid</i>

Tabel 8.14 Kasus uji memberikan penilaian *alternative flow*

<b>Kode Pengujian</b>	VT-FA-12
<b>Kode Use Case</b>	UCS-FA-07
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat memberikan peringatan apabila penilaian belum lengkap
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji berada di halaman penilaian</li> <li>2. Penguji mengosongkan penilaian</li> <li>3. Penguji menekan tombol <i>submit</i>.</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan peringatan data penilaian belum lengkap
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan peringatan data penilaian belum lengkap.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.8 Pengujian Fungsional Memperbarui Penilaian

Pada Tabel 8.14 menjelaskan mengenai kasus uji dari pengujian validasi memperbarui penilaian dengan kode VT-FA-13. Kasus uji VT-FA-08 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-08 memperbarui penilaian. Tujuan kasus uji VT-FA-13 untuk memastikan aplikasi dapat memperbarui penilaian. Berdasarkan pada Tabel 8.14 hasil status pengujian validasi untuk memperbarui penilaian menunjukkan hasil yang *valid*.

Tabel 8.15 Kasus uji memperbarui penilaian

<b>Kode Pengujian</b>	VT-FA-13
<b>Kode Use Case</b>	UCS-FA-08
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat memperbarui penilaian.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji berada di halaman hasil penilaian</li> <li>2. Penguji menekan tombol edit</li> <li>3. Penguji melakukan perubahan nilai</li> <li>4. Penguji menekan tombol <i>update</i></li> </ol>

**Tabel 8.16 Kasus uji memperbarui penilaian (Lanjutan)**

<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pemberitahuan data penilaian berhasil diperbarui.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pemberitahuan data penilaian berhasil diperbarui.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.9 Pengujian Fungsional Melihat Rekap Nilai

Pada Tabel 8.15 menjelaskan mengenai kasus uji dari pengujian validasi melihat rekap nilai dengan kode VT-FA-14. Kasus uji VT-FA-14 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-09. Tujuan kasus uji VT-FA-14 untuk memastikan aplikasi dapat menampilkan hasil rekap nilai. Berdasarkan pada Tabel 8.15 hasil status pengujian validasi untuk melihat rekap nilai menunjukkan hasil yang *valid*.

**Tabel 8.17 Kasus uji melihat rekap nilai**

<b>Kode Pengujian</b>	VT-FA-14
<b>Kode Use Case</b>	UCS-FA-09
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menampilkan hasil rekap nilai.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai ketua majelis dan telah melakukan penilaian</li> <li>2. Penguji menekan tab menu rekap nilai &amp; berita acara.</li> <li>3. Penguji menekan tombol selengkapnya pada bagian <i>gap</i> nilai</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan hasil rekap nilai.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan hasil rekap nilai.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.10 Pengujian Fungsional Melihat *Gap* Nilai

Pada Tabel 8.16 menjelaskan mengenai kasus uji dari pengujian validasi melihat *gap* nilai dengan kode VT-FA-15. Kasus uji VT-FA-15 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-10 melihat *gap* Nilai. Tujuan kasus uji VT-FA-15 untuk memastikan aplikasi dapat menampilkan *gap* nilai. Berdasarkan pada Tabel 8.10 hasil status pengujian validasi untuk melihat *gap* nilai menunjukkan hasil yang *valid*.

**Tabel 8.18 Kasus uji melihat *gap* nilai**

<b>Kode Pengujian</b>	VT-FA-15
<b>Kode Use Case</b>	UCS-FA-10
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menampilkan <i>gap</i> nilai.
<b>Prosedur Uji</b>	1. Penguji berada di halaman rekap nilai & berita acara 2. Penguji menekan tombol selengkapnya pada bagian <i>gap</i> nilai
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan daftar <i>gap</i> nilai.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan menampilkan daftar <i>gap</i> nilai.
<b>Status Uji</b>	<i>Valid</i>

#### 8.1.11 Pengujian Fungsional Melihat Rincian Penilaian Dosen Pembimbing Dan Dosen Penguji

Pada Tabel 8.17 menjelaskan mengenai kasus uji dari pengujian validasi melihat rincian penilaian dosen pembimbing dan dosen penguji dengan kode VT-FA-16. Kasus uji VT-FA-16 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-11 melihat rincian penilaian dosen pembimbing dan dosen penguji. Tujuan kasus uji VT-FA-16 untuk memastikan aplikasi dapat menampilkan rincian penilaian dosen pembimbing dan dosen penguji. Berdasarkan pada Tabel 8.17 hasil status pengujian validasi untuk melihat rincian Penilaian dosen pembimbing dan dosen penguji menunjukkan hasil yang *valid*.

**Tabel 8.19 Kasus uji melihat rincian penilaian dosen pembimbing dan dosen penguji**

<b>Kode Pengujian</b>	VT-FA-16
<b>Kode Use Case</b>	UCS-FA-11
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menampilkan rincian penilaian dosen pembimbing dan dosen penguji.
<b>Prosedur Uji</b>	1. Penguji berada di halaman rekap nilai & berita acara. 2. Penguji menekan tombol selengkapnya pada bagian nilai skripsi
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan rincian penilaian dosen pembimbing dan dosen penguji.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan rincian penilaian dosen pembimbing dan dosen penguji.

**Tabel 8.20 Kasus uji melihat rincian penilaian dosen pembimbing dan dosen penguji (Lanjutan)**

Status Uji	<i>Valid</i>
------------	--------------

### 8.1.12 Pengujian Fungsional Membuat Berita Acara

Pada Tabel 8.18 dan Tabel 8.19 menjelaskan mengenai kasus uji dari pengujian validasi membuat berita acara ujian telah selesai kode VT-FA-16. Kasus uji VT-FA-16 dan VT-FA-17 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-12 membuat berita acara ujian telah selesai. Tujuan kasus uji VT-FA-16 untuk memastikan aplikasi dapat membuat berita acara ujian telah selesai dan VT-FA-17 memastikan aplikasi menampilkan pesan apabila pengguna belum melengkapi data berita acara. Berdasarkan pada Tabel 8.18 dan Tabel 8.19 hasil status pengujian validasi untuk membuat berita acara telah selesai menunjukkan hasil yang *valid*.

**Tabel 8.21 Kasus uji membuat berita acara ujian telah selesai *basic flow***

Kode Pengujian	VT-FA-16
Kode <i>Use Case</i>	UCS-FA-12
Tujuan Pengujian	Pengujian untuk memastikan aplikasi dapat membuat berita acara ujian telah selesai.
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji berada di halaman rekap nilai &amp; berita acara</li> <li>2. Penguji mengisi berita acara pada bagian berita acara</li> <li>3. Penguji menekan tombol submit.</li> </ol>
Hasil yang Diharapkan	Aplikasi menampilkan data telah tersimpan.
Hasil Pengujian	Aplikasi berhasil menampilkan data telah tersimpan.
Status Uji	<i>Valid</i>

**Tabel 8.22 Kasus uji membuat berita acara ujian telah selesai *alternative flow***

Kode Pengujian	VT-FA-17
Kode <i>Use Case</i>	UCS-FA-12
Tujuan Pengujian	Pengujian untuk memastikan aplikasi dapat menampilkan pesan bahwa data berita acara belum lengkap
Prosedur Uji	<ol style="list-style-type: none"> <li>1. Penguji berada di halaman rekap nilai &amp; berita acara</li> <li>2. Penguji mengosongkan berita acara pada bagian berita acara</li> <li>3. Penguji menekan tombol submit.</li> </ol>

**Tabel 8.23 Kasus uji membuat berita acara ujian telah selesai *alternative flow* (Lanjutan)**

<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pesan bahwa data berita acara belum lengkap.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pesan bahwa data berita acara belum lengkap.
<b>Status Uji</b>	<i>Valid</i>

Pada Tabel 8.20 dan Tabel 8.21 menjelaskan mengenai kasus uji dari pengujian validasi membuat berita acara penjadwalan ulang kode VT-FA-18 dan VT-FA-19 . Kasus uji VT-FA-18 dan VT-FA-19 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-13 membuat berita acara ujian penjadwalan ulang. Tujuan kasus uji VT-FA-18 untuk memastikan aplikasi dapat membuat berita acara ujian penjadwalan ulang dan VT-FA-19 untuk memastikan aplikasi menampilkan pesan berita acara penjadwalan ulang belum lengkap. Berdasarkan pada Tabel 8.20 dan Tabel 8.21 hasil status pengujian validasi untuk membuat berita penjadwalan ulang menunjukkan hasil yang *valid*.

**Tabel 8.24 Kasus uji membuat berita acara ujian penjadwalan ulang**

<b>Kode Pengujian</b>	VT-FA-18
<b>Kode Use Case</b>	UCS-FA-13
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat membuat berita acara ujian penjadwalan ulang.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai ketua majelis dan status kehadiran mahasiswa tidak hadir</li> <li>2. Penguji mengisi berita acara</li> <li>3. Penguji menekan tombol submit.</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pesan data telah tersimpan.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pesan data telah tersimpan.
<b>Status Uji</b>	<i>Valid</i>

**Tabel 8.25 Kasus uji membuat berita acara ujian telah selesai *alternative flow***

<b>Kode Pengujian</b>	VT-FA-19
<b>Kode Use Case</b>	UCS-FA-13
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menampilkan pesan bahwa data berita acara belum lengkap

**Tabel 8.26 Kasus uji membuat berita acara ujian telah selesai *alternative flow* (Lanjutan)**

<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai ketua majelis dan status kehadiran mahasiswa tidak hadir</li> <li>2. Penguji mengosongkan berita acara</li> <li>3. Penguji menekan tombol submit.</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan pesan bahwa data berita acara belum lengkap.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan pesan bahwa data berita acara belum lengkap.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.13 Pengujian Fungsional Melihat Daftar Riwayat Ujian Skripsi

Pada Tabel 8.22 menjelaskan mengenai kasus uji dari pengujian validasi melihat daftar riwayat ujian skripsi kode VT-FA-20. Kasus uji VT-FA-20 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-14 melihat daftar riwayat ujian skripsi. Tujuan kasus uji VT-FA-20 untuk memastikan aplikasi dapat menampilkan daftar riwayat ujian skripsi. Berdasarkan pada Tabel 8.22 hasil status pengujian validasi untuk melihat daftar riwayat ujian skripsi menunjukkan hasil yang *valid*.

**Tabel 8.27 Kasus uji melihat daftar riwayat ujian skripsi**

<b>Kode Pengujian</b>	VT-FA-20
<b>Kode Use Case</b>	UCS-FA-14
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menampilkan daftar riwayat ujian skripsi.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah login</li> <li>2. Penguji menekan menu jadwal ujian</li> <li>3. Penguji menekan tab menu riwayat</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan daftar informasi ujian skripsi yang telah diselesaikan pengguna.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan daftar informasi ujian skripsi yang telah diselesaikan pengguna.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.14 Pengujian Fungsional Melihat Rincian Riwayat Ujian Skripsi

Pada Tabel 8.23 menjelaskan mengenai kasus uji dari pengujian validasi melihat rincian riwayat ujian skripsi kode VT-FA-21. Kasus uji VT-FA-21 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-15 melihat

rincian riwayat ujian skripsi. Tujuan kasus uji VT-FA-21 untuk memastikan aplikasi dapat menampilkan rincian riwayat ujian skripsi. Berdasarkan pada Tabel 8.23 hasil status pengujian validasi untuk melihat rincian riwayat ujian skripsi menunjukkan hasil yang *valid*.

**Tabel 8.28 Kasus uji melihat rincian riwayat ujian skripsi**

<b>Kode Pengujian</b>	VT-FA-21
<b>Kode Use Case</b>	UCS-FA-15
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menampilkan rincian riwayat ujian skripsi.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji telah melakukan login</li> <li>2. Penguji menekan menu jadwal ujian</li> <li>3. Penguji menekan tab menu riwayat</li> <li>4. Penguji memilih jadwal ujian dari daftar riwayat jadwal ujian</li> </ol>
<b>Hasil yang Diharapkan</b>	Aplikasi menampilkan rincian riwayat ujian skripsi.
<b>Hasil Pengujian</b>	Aplikasi berhasil menampilkan rincian riwayat ujian skripsi.
<b>Status Uji</b>	<i>Valid</i>

### 8.1.15 Pengujian Fungsional Menjalankan Pengatur Waktu Ujian

Pada Tabel 8.24 menjelaskan mengenai kasus uji dari pengujian validasi menjalankan pengatur waktu ujian kode VT-FA-22. Kasus uji VT-FA-22 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCS-FA-16 menjalankan pengatur waktu ujian. Tujuan kasus uji VT-FA-22 untuk memastikan aplikasi dapat menjalankan pengatur waktu ujian. Berdasarkan pada Tabel 8.24 hasil status pengujian validasi untuk menjalankan pengatur waktu ujian menunjukkan hasil yang *valid*.

**Tabel 8.29 Kasus uji menjalankan waktu pengatru waktu ujian**

<b>Kode Pengujian</b>	VT-FA-22
<b>Kode Use Case</b>	UCS-FA-16
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan aplikasi dapat menjalankan pengatur waktu ujian.
<b>Prosedur Uji</b>	<ol style="list-style-type: none"> <li>1. Penguji teridentifikasi sebagai ketua majelis dan telah mendapatkan aksespenilaian</li> <li>2. Penguji menekan menu pengatur waktu</li> <li>3. Penguji menjalankan waktu ujian dari salah satu daftar waktu ujian.</li> </ol>

Tabel 8.30 Kasus uji menjalankan waktu pengatur waktu ujian (Lanjutan)

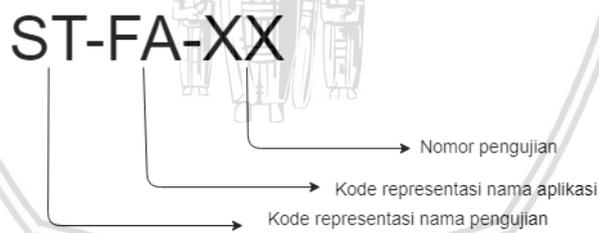
<b>Hasil yang Diharapkan</b>	Aplikasi menjalankan pengatur waktu.
<b>Hasil Pengujian</b>	Aplikasi berhasil menjalankan pengatur waktu.
<b>Status Uji</b>	<i>Valid</i>

## 8.2 Pengujian Non-Fungsional

Subab ini membahas mengenai pengujian non-fungsional sistem. Pengujian non-fungsional terdiri atas pengujian keamanan (*security testing*) dan pengujian *usability*. Pengujian keamanan digunakan untuk menguji keamanan dari *web service* dan pengujian *usability* digunakan untuk menguji tingkat kemudahan sistem.

### 8.2.1 Pengujian Keamanan

Subab ini menjelaskan hasil pengujian keamanan aplikasi penilaian ujian skripsi. Pengujian keamanan ini untuk memastikan kebutun non fungsional yaitu *security* telah terpenuhi. Keamanan pada aplikasi terfokus pada keamanan dari *web service*. Sehingga pengujian ini dilakukan untuk menguji keamanan dari *web service*. Pengujian keamanan dilakukan dengan bantuan *tool* yaitu postman. API yang diuji hanya fungsional utama dari aplikasi, yaitu verifikasi kehadiran, memberikan penilain, merubah penilaian dan mengisi berita acara. Selain itu, pengujian ini memiliki aturan penomoran seperti yang ditunjukkan dalam Gambar 8.2.



Gambar 8.2 Aturan Penomoran Pengujian Keamanan

#### 8.2.1.1 Pengujian Keamanan Memverifikasi Kehadiran

Pada Tabel 8.25 , Tabel 8.26 dan Tabel 8.27 menjelaskan mengenai kasus uji dari pengujian keamanan API verifikasi kehadiran. Pengujian keamanan API verifikasi kehadiran dilakukan dengan tiga kasus uji, yaitu kasus uji dengan menggunakan token yang valid, kasus uji tanpa menggunakan token dan kasus uji menggunakan token yang tidak valid. Hasil yang didapatkan berdasarkan tiga kasus uji, API verifikasi kehadiran tidak dapat berjalan apabila API tersebut di akses tanpa menggunakan token yang valid. Token ini dijadikan izin akses API verifikasi kehadiran.

**Tabel 8.31 Hasil pengujian keamanan verifikasi kehadiran tanpa menggunakan token**

<b>Kode Pengujian</b>	ST-FA-1
<b>Endpoint</b>	/ujian/verifikasi
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API verifikasi kehadiran.
<b>Prosedur Uji</b>	Mengakses API tanpa menggunakan token.
<b>Hasil yang Diharapkan</b>	API tidak memberikan izin untuk melakukan proses verifikasi kehadiran.
<b>Hasil Pengujian</b>	API tidak memberikan izin untuk melakukan proses verifikasi kehadiran.

**Tabel 8.32 Hasil pengujian keamanan verifikasi kehadiran menggunakan token yang valid**

<b>Kode Pengujian</b>	ST-FA-2
<b>Endpoint</b>	/ujian/verifikasi
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API verifikasi kehadiran.
<b>Prosedur Uji</b>	Mengakses API dengan menggunakan token yang <i>valid</i> .
<b>Hasil yang Diharapkan</b>	API dapat melakukan proses verifikasi kehadiran.
<b>Hasil Pengujian</b>	API dapat melakukan proses verifikasi kehadiran.

**Tabel 8.33 Hasil pengujian keamanan verifikasi kehadiran menggunakan token yang tidak valid**

<b>Kode Pengujian</b>	ST-FA-3
<b>Endpoint</b>	/ujian/verifikasi
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API verifikasi kehadiran.
<b>Prosedur Uji</b>	Mengakses API dengan menggunakan token yang tidak <i>valid</i> .
<b>Hasil yang Diharapkan</b>	API tidak memberikan izin untuk melakukan proses verifikasi kehadiran.
<b>Hasil Pengujian</b>	API tidak memberikan izin untuk melakukan proses verifikasi kehadiran.

### 8.2.1.2 Pengujian Keamanan Memberikan Penilaian

Pada Tabel 8.28 , Tabel 8.29 dan Tabel 8.30 menjelaskan mengenai kasus uji dari pengujian keamanan API merubah penilai. Pengujian keamanan API merubah penilaian dilakukan dengan tiga kasus uji, yaitu kasus uji dengan menggunakan token yang valid, kasus uji tanpa menggunakan token dan kasus uji menggunakan token yang tidak valid. Hasil yang didapatkan berdasarkan tiga kasus uji, API memberikan penilaian tidak dapat berjalan apabila API tersebut di akses tanpa menggunakan token yang valid. Token ini dijadikan izin akses API memberikan penilaian.

**Tabel 8.34 Hasil pengujian keamanan memberikan penilaian tanpa menggunakan token**

<b>Kode Pengujian</b>	ST-FA-4
<b>Endpoint</b>	/ujian/save_nilai
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API memberikan penilaian.
<b>Prosedur Uji</b>	Mengakses API tanpa menggunakan token.
<b>Hasil yang Diharapkan</b>	API tidak memberikan izin untuk melakukan proses menyimpan penilaian.
<b>Hasil Pengujian</b>	API tidak memberikan izin untuk melakukan proses menyimpan penilaian.

**Tabel 8.35 Hasil pengujian keamanan memberikan penilaian menggunakan token yang valid**

<b>Kode Pengujian</b>	ST-FA-5
<b>Endpoint</b>	/ujian/save_nilai
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API memberikan penilaian.
<b>Prosedur Uji</b>	Mengakses API dengan menggunakan token yang <i>valid</i> .
<b>Hasil yang Diharapkan</b>	API dapat melakukan proses menyimpan penilaian.
<b>Hasil Pengujian</b>	API dapat melakukan proses menyimpan penilaian.

**Tabel 8.36 Hasil pengujian keamanan memberikan penilaian menggunakan token yang tidak valid**

<b>Kode Pengujian</b>	ST-FA-6
<b>Endpoint</b>	/ujian/save_nilai

**Tabel 8.37 Hasil pengujian keamanan memberikan penilaian menggunakan token yang tidak valid (Lanjutan)**

<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API memberikan penilaian.
<b>Prosedur Uji</b>	Mengakses API dengan menggunakan token yang tidak <i>valid</i> .
<b>Hasil yang Diharapkan</b>	API tidak memberikan izin untuk melakukan proses menyimpan penilaian.
<b>Hasil Pengujian</b>	API tidak memberikan izin untuk melakukan proses menyimpan penilaian.

### 8.2.1.3 Pengujian Keamanan Membuat Berita Acara

Pada Tabel 8.31, Tabel 8.32 dan Tabel 8.33 menjelaskan mengenai kasus uji dari pengujian keamanan API membuat berita acara. Pengujian keamanan API membuat berita acara dilakukan dengan tiga kasus uji, yaitu kasus uji dengan menggunakan token yang *valid*, kasus uji tanpa menggunakan token dan kasus uji menggunakan token yang tidak *valid*. Hasil yang didapatkan berdasarkan tiga kasus uji, API membuat berita acara tidak dapat berjalan apabila API tersebut di akses tanpa menggunakan token yang *valid*. Token ini dijadikan izin akses API membuat berita acara.

**Tabel 8.38 Hasil pengujian keamanan membuat berita acara tanpa menggunakan token**

<b>Kode Pengujian</b>	ST-FA-7
<b>Endpoint</b>	/ujian/save_berita_acara
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API membuat berita acara.
<b>Prosedur Uji</b>	Mengakses API tanpa menggunakan token.
<b>Hasil yang Diharapkan</b>	API tidak memberikan izin untuk melakukan proses membuat berita acara.
<b>Hasil Pengujian</b>	API tidak memberikan izin untuk melakukan proses membuat berita acara.

**Tabel 8.39 Hasil pengujian keamanan membuat berita acara menggunakan token yang valid**

<b>Kode Pengujian</b>	ST-FA-8
<b>Endpoint</b>	/ujian/save_berita_acara

**Tabel 8.40 Hasil pengujian keamanan membuat berita acara menggunakan token yang valid (Lanjutan)**

<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API membuat berita acara.
<b>Prosedur Uji</b>	Mengakses API dengan menggunakan token yang <i>valid</i> .
<b>Hasil yang Diharapkan</b>	API dapat melakukan proses membuat berita acara.
<b>Hasil Pengujian</b>	API dapat melakukan proses membuat berita acara.

**Tabel 8.41 Hasil pengujian keamanan membuat berita acara menggunakan token yang tidak valid**

<b>Kode Pengujian</b>	ST-FA-9
<b>Endpoint</b>	/ujian/save_berita_acara
<b>Tujuan Pengujian</b>	Pengujian untuk memastikan keamanan dari API membuat berita acara.
<b>Prosedur Uji</b>	Mengakses API dengan menggunakan token yang tidak <i>valid</i> .
<b>Hasil yang Diharapkan</b>	API tidak memberikan izin untuk melakukan proses membuat berita acara.
<b>Hasil Pengujian</b>	API tidak memberikan izin untuk melakukan proses membuat berita acara.

## 8.2.2 Pengujian *Usability*

Bagian ini membahas mengenai hasil pengujian *usability* sistem. Pengujian *usability* digunakan untuk menguji tingkat kemudahan penggunaan sistem. Pengujian *usability* terdiri dari tiga aspek *effectiveness*, *efficiency*, dan *satisfaction*. Berikut ini hasil dari pengujian *usability* sistem.

### 8.2.2.1 Effectiveness

**Tabel 8.42 Tingkat keberhasilan penyelesaian tugas berdasarkan fungsi yang digunakan**

Responden	Complete			
	Verifikasi Kehadiran (Task 1)	Penilaian (Task 2)	Rekap Nilai (Task 3)	Berita Acara (Task 4)
Pengguna 1	1	1	1	1
Pengguna 2	1	1	1	1

**Tabel 8.43 Tingkat keberhasilan penyelesaian tugas berdasarkan fungsi yang digunakan (Lanjutan)**

Pengguna 3	1	1	1	1
Pengguna 4	1	1	1	1
Pengguna 5	1	1	1	1
<b>Completion Rate</b>	100%			

Pada aspek *effectiveness*, peneliti melakukan pengukuran tingkat keberhasilan dalam menyelesaikan setiap *task* yang diberikan. Apabila pengguna tidak dapat menyelesaikan *task* tersebut akan memperoleh nilai 0 dan sebaliknya jika pengguna berhasil menyelesaikan *task* maka akan memperoleh nilai 1. Berdasarkan Tabel 8.34, tingkat keberhasilan penyelesaian tugas berdasarkan fungsi menghasilkan nilai 1 dari semua *task* yang dikerjakan oleh 5 pengguna. Tingkat keberhasilan ini juga tidak terdapat kesalahan – kesalahan yang dilakukan oleh pengguna. Sehingga tingkat keberhasilan pengguna dalam menyelesaikan tugas memperoleh presentase sebesar 100% dari keseluruhan *task* yang dikerjakan.

#### 8.2.2.2 Efficiency

Pada aspek *efficiency*, peneliti mengukur dengan menggunakan persamaan *time based efficiency* berdasarkan waktu yang dibutuhkan pengguna dalam menyelesaikan tugas. Tabel 8.35 merupakan waktu yang dibutuhkan pengguna dalam menyelesaikan *task* dengan menggunakan fungsional verifikasi kehadiran, penilaian, rekap nilai, dan berita acara. Kemudian, Tabel 8.36 merupakan *efficiency* setiap *task* dan secara keseluruhan. Berdasarkan Tabel tersebut diperoleh tingkat *efficiency* pada fungsional verifikasi kehadiran sebesar 0.196 *goals/sec*, fungsional penilaian sebesar 0.072 *goals/sec*, fungsional rekap nilai sebesar 0.206 *goals/sec*, fungsional berita acara 0.202 *goals/sec*. Lalu, secara keseluruhan aplikasi ini memiliki tingkat *efficiency* sebesar 0.169 *goals/sec*.

**Tabel 8.44 Waktu yang dibutuhkan pengguna dalam mengerjakan task**

Pengguna	Waktu (Second)			
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
Pengguna 1	32	90	30	35
Pengguna 2	27	87	26	26
Pengguna 3	45	92	38	36
Pengguna 4	26	80	28	25
Pengguna 5	35	85	32	37

Tabel 8.45 Tingkat *efficiency* aplikasi penilaian ujian skripsi

Efficiency (goals/sec)			
Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
0.196	0.072	0.206	0.202
Keseluruhan			
0.169			

### 8.2.2.3 Satisfaction

Pada aspek *satisfaction* peneliti melakukan pengukuran dengan menggunakan *Single Ease Question*. Penerapan dari *Single Ease Question* dengan mengajukan pertanyaan setelah pengguna menyelesaikan *task* yang diberikan. Penilaian dari pertanyaan tersebut menggunakan skala likert dari 1-7. Tabel 8.37 merupakan hasil tingkat *satisfaction* yang diperoleh dari aplikasi penilaian ujian skripsi. Hasil yang diperoleh dari pengukuran *satisfaction* mencapai 6.1 untuk rata-rata keseluruhan. Nilai tersebut dapat dikatakan sudah melebihi nilai rata-rata yang harus diperoleh dalam aspek *satisfaction*.

Tabel 8.46 Hasil penilaian kuisioner *Single Ease Question*

Responden	Nilai			
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
Responden 1	5	7	6	7
Responden 2	7	6	7	5
Responden 3	5	6	5	5
Responden 4	6	6	6	6
Responden 5	7	7	7	6
<b>RataRata</b>	6	6.4	6.2	5.8
<b>Rata-Rata Keseluruhan</b>	6.1			

### 8.3 Analisis Hasil Pengujian

Bagian ini merupakan hasil dari pengujian fungsional, pengujian keamanan dan pengujian usability yang telah dilakukan. Hasil ini menjelaskan kesimpulan dari setiap pengujian. Berikut ini merupakan hasil pengujiannya.

1. Pada pengujian fungsional yang dilakukan dengan menggunakan pengujian validasi menghasilkan status uji yang valid dari setiap fungsionalnya. Sehingga dapat disimpulkan 15 fungsional aplikasi penilaian ujian skripsi telah berjalan dengan baik.

2. Pada pengujian keamanan dari 3 kasus uji di 3 API menghasilkan hasil yang diharapkan, yang dimana API tidak dapat diakses tanpa menggunakan *authorization*. Dari hasil tersebut dapat disimpulkan bahwa API memiliki keamanan yang baik sehingga pengguna atau seseorang yang ingin mencoba mengakses API tanpa melalui aplikasi, API tidak akan memberikan izin akses.
3. Pada pengujian *usability* hasil yang didapatkan yaitu, 100% pada aspek *effectiveness*, 0.169 *goals/sec* pada aspek *efficiency* dan 6.1 pada aspek *satisfaction*. Berdasarkan teori dari pengujian *usability* yang telah dijelaskan pada bab 2 bahwa pada aspek *effectiveness* dan *efficiency* telah mencapai hasil yang baik dengan memiliki tingkat *efficiency* yang mencapai 0.169 *goals/sec*.



## BAB 9 PENUTUP

### 9.1 Kesimpulan

Setelah semua tahap penelitian dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada hasil rancangan dan implementasi dari proses pengembangan aplikasi menghasilkan rancangan dan implementasi, sebagai berikut:
  - a. Hasil perancangan dari proses pengembangan aplikasi terdiri atas perancangan arsitektur sistem yang menggambarkan arsitektu sistem yang baru ketika adanya aplikasi penilaian ujian skripsi berbasis android, perancangan *sequence diagram* yang digunakan untuk memodelkan interaksi antar objek, perancangan *class diagram* yang digunakan untuk memodelkan objek, perancangan basis data, perancangan API, perancangan antarmuka pengguna dan perancangan algoritme. Pada perancangan *class diagram* menghasilkan 49 *class* dan 23 tabel pada perancangan basis data.
  - b. Aplikasi penilaian ujian skripsi merupakan aplikasi berbasis android sebagai hasil pengembangan yang dilakukan berdasarkan hasil perancangan. Aplikasi ini telah berjalan dengan baik melalui proses pengujian yang telah dilakukan. Selain itu juga, *web service* sangat diperlukan pada aplikasi untuk melakukan pertukaran data, dan untuk melindungi proses pertukaran data, *web service* menggunakan *authorization JSON Web Token*.
2. Aplikas penilaian ujian skripsi yang dibangun dengan menggunakan metode model *prototyping* menghasilkan suatu aplikasi yang mengikuti kebutuhan pengguna akan kemudahan dalam menggunakan aplikasi. Hal ini berdasarkan dari hasil pengujian *usability* yang telah dilakukan, yang dimana menghasilkan, 100% pada aspek *effectiveness* dan 6.1 pada rata-rata keseluruhan pada aspek *satisfaction*. Kedua aspek tersebut telah mencapai hasil yang baik dengan tingkat *efficiency* sebesar 0.169 *goals/sec*.

### 9.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya terkait pengembangan aplikasi FILKOM APPS berbasis android diantaranya adalah:

1. Perlu dilakukannya analisis lebih lanjut mengenai kesiapan Fakultas Ilmu Komputer terhadap pengembangan sistem FILKOM APPS berbasis android sehingga hasilnya dapat memaksimalkan pengembangan sistem FILKOM APPS berbasis android lebih baik dan juga dapat mencegah resiko-resiko yang memungkinkan terjadi ketika pengembangan sistem FILKOM APPS menjadi berbasis android.

2. Perlu dilakukannya penelitian lebih lanjut pada keamanan *web service*. Hal ini dikarenakan keamanan dari *web service* menjadi tantangan Fakultas Ilmu Komputer dalam menerapkan sistem berbasis android.



## DAFTAR PUSTAKA

- 1Keydata, 2018. *Physical Data Model*. [Online] Tersedia di: <https://www.1keydata.com/datawarehousing/physical-data-model.html> [Diakses 25 Februari 2018].
- Android Developer, 2014. *Understand the Activity Lifecycle*. [Online] Tersedia di: <https://developer.android.com/guide/components/activities/activity-lifecycle> [Diakses 27 September 2017].
- Angelia, V, 2015. Rancang Bangun Sistem Estimasi Proyek Pada Konveksi Pakaian Dengan Metode Pert-CPM Dan *Prototyping Model*. S1. Universitas Brawijaya.
- Beasley, M., 2013. *Practical Web Analytics for User Experience 1nd*. s.l.:Morgan Kaufman.
- Bertram, D., 2016. *Likert Scale*. s.l.:s.n.
- Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide SECOND EDITION*. s.l.:Addison Wesley Professional.
- Darmawan, D. & Fauzi, K. N., 2013. *Sistem Informasi Manajemen*. Bandung: PT Remaja Rosdakarya.
- Edhy, S. & Khabib, M., 2012. Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-GOV Di Pemkab Bantul Yogyakarta. *Jurnal Teknologi Informasi dan Komunikasi UGM*, Volume 1. Tersedia melalui: Repository Universitas Gajah Mada < <https://repository.ugm.ac.id/> > [Diakses 15 Februari 2018]
- Edwards, J., 2014. *Mobile Apps Are Killing The Free Web, Handing A Censored Duopoly to Google And Apple*. [Online] Tersedia di: <http://www.businessinsider.com/mobile-web-vs-app-usage-statistics-2014-4/?IR=T> [Diakses 20 January 2018].
- Fakhroutdinov, K., 2009. *UML Class Diagrams Reference*. [Online] Tersedia di: <https://www.uml-diagrams.org/class-reference.html> [Diakses 07 Februari 2018].
- Fakultas Ilmu Komputer Universitas Brawijaya, n.d. *Struktur Organisasi*. [Online] Tersedia di: <http://filkom.ub.ac.id/page/read/struktur-organisasi/5105a1cf9bc2e1> >[Diakses 22 Februari 2018].
- Fakultas Ilmu Komputer Universitas Brawijaya, 2017. Panduan Skripsi Fakultas Ilmu Komputer Universitas Brawijaya 2017. Malang: Fakultas Ilmu Komputer
- Firebase, 2018. *Firebase Cloud Messaging*. [Online] Tersedia di: <https://firebase.google.com/docs/cloud-messaging/> [Diakses 6 Februari 2018].
- Fowler, M., 2005. *UML Distilled 3th*. Yogyakarta: Andi.

- Gallo, A., 2017. *A Refresher on A/B Testing*. [Online] Tersedia di: <<https://hbr.org/2017/06/a-refresher-on-ab-testing>> [Diakses 10 Februari 2018].
- Hammer M, Champy J., 1993. *Reengineering the Corporation: A Manifesto for Business Revolution*. s.l.:s.n.
- Hoos, E., Groger, C., Kramer, S. & Mitschang, B., 2014. *Improving Business Processes through Mobile Apps*.
- Imaduddin, A. & Permana, S., 2017. *Menjadi Android Developer Expert*. Bandung: PT. Presentologics.
- ISTQB, n.d. *What is Prototype model- advantages, disadvantages and when to use it?*. [Online] Tersedia di: <<http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/>> [Diakses 05 Februari 2018].
- Jobe, W., 2013. *Native Apps Vs Mobile Web Apps*. *Researchgate*.
- JWT, n.d. *Introduction JSON Web Token*. [Online] Tersedia di: <<https://jwt.io/introduction/>> [Diakses 06 Februari 2018].
- Kemp, S., 2018. *Digital In 2018: World's Internet Users Pass The 4 Billion Mark*. [Online] Tersedia di: <<https://wearesocial.com/blog/2018/01/global-digital-report-2018>> [Diakses 10 January 2018].
- Lazar, J., Feng, J. H. & Hochheiser, H., 2017. *Research Methods in Human-Computer Interaction Second Edition*. Cambridge: Todd Green.
- Leiva, A., 2014. *MVP for Android: how to organize the presentation layer*. [Online] Tersedia di: <<https://antonioleiva.com/mvp-android/>> [Diakses 14 Maret 2018].
- Lumen, 2015. *Lumen*. [Online] Tersedia di: <<https://lumen.laravel.com/>> [Diakses 8 Februari 2018].
- McConnell, S., 1996. *Rapid Development: taming wild software schedules*. Washington: Microsoft Press.
- Mifsud, J., 2015. *Usability Metrics – A Guide To Quantify The Usability Of Any System*. [Online] Tersedia di: <<https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>> [Diakses 28 January 2018].
- Naik, K. & Tripathy, P., 2008. *Software Testing And Quality Assurance Theory and Practice*. s.l: John Wiley & Sons, Inc.
- Nielsen, J., 2005. *Putting A/B Testing Its Place*. [Online] Tersedia di: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>> [Diakses 5 Februari 2018].
- Nielsen, J., 2012. *Usability 101: Introduction to Usability*. [Online] Tersedia di: <<https://www.nngroup.com/articles/putting-ab-testing-in-its-place/>> [Diakses 5 Februari 2018].

- Nolan, G., O. C. & Truxall, D., 2014. *Android Best Practice*. s.l.:Apress.
- Nugroho, A., 2010. *Rekayasa perangkat lunak berorientasi objek dengan metode USDP (Unified Software Development Process)*. Yogyakarta: Andi.
- Object Management Group, 2011. *Business Process Model and Notation (BPMN)*. s.l.:Object Management Group.
- O'Brien, J. A. & Marakas, G. M., 2011. *Management Information System*. New York: McGraw-Hill.
- Oracle, 2013. *What Are Web Services?*. [Online] Tersedia di: <<https://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html>> [Diakses 5 February 2018].
- Pramudinantaka, F, 2015. Pengembangan Aplikasi Web Terintegrasi Pada Perpustakaan Universitas Brawijaya Dengan Model Prototyping. S1. Universitas Brawijaya.
- Presman, R. S., 2010. *Software Engineering - A Practitioners Approach 7nd*. New York: MacGraw - Hill Companies, Inc.
- Safaat, N., 2012. *Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*. Bandung: Informatika.
- Sauro, J., 2012. *10 Things To Know About The Single Ease Question (SEQ)*. [Online] Tersedia di: <<https://measuringu.com/seq10/>> [Diakses 29 January 2018].
- Simarmata, J., 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Sommerville, I., 2007. *Software Engineering 8nd*. s.l.:Pearson Education.
- S, R. A. & Shalahuddin, M., 2014. *Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*. Bandung: Informatika.
- Stair, R. M. & Reynolds, G. W., 2014. *Principles Of Information Systems*. United States: Course Technology Cengage Learning 11th.
- Sugiyono, 2017. *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Tolle, H., Pinandito, A., Dewi, R. K. & Putra, A. K., 2017. *Pengembangan Aplikasi Perangkat Bergerak*. Malang: UBPress.
- Weske, M., 2007. *Business Process Management*. New York: Springer-Verlag Berlin Heidelberg.
- Wickham, M., 2018. *Practical Android*. s.l.:Apress.
- Wijaya, S, 2013. Sistem Informasi Akademik Universitas Islam Negeri Sunan Kalijaga Yogyakarta Berbasis Android. S1. Universitas Islam Negeri Sunan Kalijaga Yogyakarta.

## LAMPIRAN A HASIL VALIDASI MASALAH

Lampiran ini merupakan hasil validasi masalah yang telah dilakukan untuk mendukung penelitian ini. Validasi masalah dilakukan dengan menggunakan kuisioner dan skala *likert*. Setiap pernyataan pada kuisioner dilakukan pengukuran untuk menghasilkan tingkat persetujuan terhadap suatu pandangan terkait masalah yang diangkat pada penelitian ini.

No	Pernyataan	Penilaian				
		1	2	3	4	5
1	Saya lebih sering menggunakan smarthphone untuk mengakses filkom apps	2	4	4	5	5
2	Saya rasa tampilan dari filkom apps masih kurang mobile friendly	0	6	4	8	1
3	Saya cukup kesulitan ketika mengakses filkom apps melalui smarthphone	0	9	4	6	0
4	Saya rasa filkom apps perlu dikembangkan ke arah mobile (native)	0	1	0	10	8

Pernyataan	Saya lebih sering menggunakan smarthphone untuk mengakses filkom apps
<b>Nilai</b>	<b>Total</b>
1	2
2	8
3	12
4	20
5	25
Total Nilai	67
Presentase	71%

Pernyataan	Saya rasa tampilan dari filkom apps masih kurang <i>mobile friendly</i>
<b>Nilai</b>	<b>Total</b>
1	0
2	12
3	12
4	32
5	5
Total Nilai	61
Presentase	64%

Pernyataan	Saya cukup kesulitan ketika mengakses filkom apps melalui smarthphone.
<b>Nilai</b>	<b>Total</b>
1	0
2	18
3	12
4	24
5	0
Total Nilai	54
Presentase	57%

Pernyataan	Saya rasa filkom apps perlu dikembangkan ke arah mobile (native)
<b>Nilai</b>	<b>Total</b>
1	0
2	2
3	0
4	40
5	40
Total Nilai	82
Presentase	86%

## LAMPIRAN B WAWANCARA

### Hasil Wawancara

Nama Responden : Thusti Dwi Purnami, A.Md

Jabatan : Staff PSIK FILKOM UB

Tanggal Wawancara : Jumat, 10 November 2017

1. Bagaimana flow dari fitur penilain ujian skripsi?  
Fitur penilaian ujian skripsi pada pembimbing akan terbuka ketika sudah terbitnya surat undangan ujian skripsi. Ketika itu, pembimbing dapat menilai skripsi bimbingannya hanya pada komponen proses pengerjaan skripsi dan kualitas penulisan dokumen skripsi. Terkait penilain pada komponen kualitas presentasi dan kualitas argumentasi dapat dilakukan ketika telah diverifikasi kehadiran oleh ketua majelis pada saat pengujian. Verifikasi kehadiran ini juga dapat direvisi melalui sistem. Pada fitur penilaian ujian skripsi ini dosen pembimbing dan dosen penguji tidak boleh memiliki perbedaan dua angka. Hal ini dikarenakan untuk mencegah gap nilai yang terlalu jauh antara dosen pembimbing dengan dosen penguji. Setelah dosen pembimbing dan dosen penguji memberikan nilai, ketua majelis merekap nilai dan mengisi berita acara di sistem.
2. Bagaimana tampilan fitur penilain ujian skripsi pada dosen pembimbing, dosen penguji dan ketua majelis?  
Pada dosen pembimbing dan dosen penguji hanya terdapat 1 tampilan form penilaian sedangkan pada ketua majelis terdapat tampilan penilain ujian skripsi pembimbing dan penguji, tampilan untuk memverifikasi kehadiran dosen pembimbing dan dosen penguji dan juga tampilan nilai akhir dan berita acara. Selain itu juga terdapat detail mahasiswa yang berisikan data mahasiswa, informasi skripsi mahasiswa serta nama pembimbing/penguji skripsi mahasiswa yang dapat dilihat oleh dosen penguji dan dosen pembimbing.
3. Bagaimana tampilan fitur penilain ujian skripsi ketiak di akses melalui *mobile*?  
Filkom apps untuk saat ini tidak disarankan mengaksesnya melalui *mobile*.

Malang, Senin 11 November 2017



Thusti Dwi P.

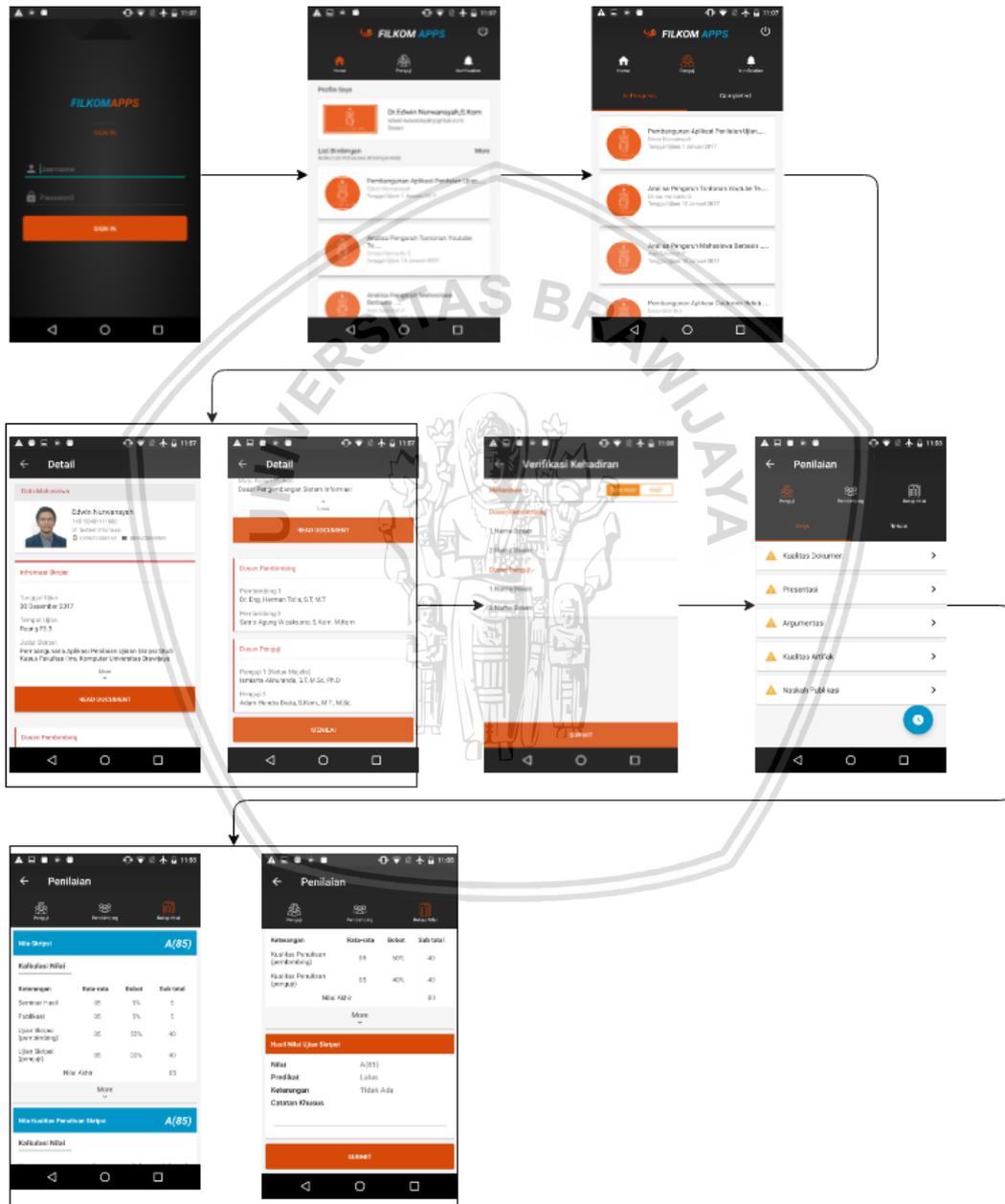


## LAMPIRAN C RINCIAN UJI PROTOTIPE

Lampiran ini merupakan lampiran yang digunakan untuk memperlihatkan implementasi dua prototipe yang didasarkan pada konsep *A/B testing* dan sebagai lembar pendukung dari bab evaluasi prototipe pada subbab iterasi satu.

### C.1 Iterasi Satu

#### C.1.1 Implementasi Prototipe A



### C.1.2 Implementasi Prototipe B



## C.1.3 Saran Pengguna

No	Pengguna	Saran
1	Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.	Alangkah baiknya memperlihatkan prototype berikutnya memperlihatkan kondisi ketika terjadi konflik.
2	Lutfi Fanani, S.Kom., M.T., M.Sc.	<ul style="list-style-type: none"> <li>• Pada bagian pengatuaran waktu alangkah baiknya dibuat <i>minimize</i>.</li> <li>• Verfikasi kehadiran dibuat kondisi awal hadir</li> <li>• Rincian Penilaian tidak tidak perlu ditampilkan per poin</li> </ul>
3	Retno Indah Rokhmawati, S.Pd., M.Pd.	<ul style="list-style-type: none"> <li>• Tambahkan dropdown nama dosen penguji pengganti</li> <li>• Memerlukan riwayat pengujian skripsi</li> </ul>
4	Himawat Aryadita, S.T, M.Sc	<ul style="list-style-type: none"> <li>• Tidak perlu membuat aplikasi yang memiliki proses yang terlalu banyak.</li> <li>• Penggabungan prototype a dan b pada penilaian, yang dimana menggunakan poin tapi pada kolom untuk memasukan nilai seperti prototype b.</li> <li>• Pada time antara prototype a dan b dipadukan.</li> </ul>
5	Adam Hendra Brata, S.Kom., M.T., M.Sc.	<ul style="list-style-type: none"> <li>• Jika menggunakan icon sebagai tombol,alangka baiknya lebih dipertegas.</li> <li>• Verfikasi kehadiran, tombolnya sebaiknya diletakkan sebelum penilaian</li> <li>• Memerlukan fitur riwayat pengujian</li> </ul>

## C.2 Iterasi Dua

### C.2.1 Effectiveness

Responden	Complete			
	Verifikasi Kehadiran	Penilaian	Rekap Nilai	Berita Acara
Retno Indah Rokhmawati, S.Pd., M.Pd.	1	1	1	1
Himawat Aryadita, S.T, M.Sc	1	1	1	1
Adam Hendra Brata, S.Kom., M.T., M.Sc.	1	1	1	1
Fitra Abdurrachman Bachtiar, Dr.Eng., S.T, M.Eng	1	1	1	1
Ratih Kartika Dewi, S.T., M.Kom	1	1	1	1

### C.2.2 Efficiency

Pengguna	Waktu			
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
Retno Indah Rokhmawati, S.Pd., M.Pd.	21	65	18	25
Himawat Aryadita, S.T, M.Sc	28	84	20	33
Adam Hendra Brata, S.Kom., M.T., M.Sc.	25	58	22	36
Fitra Abdurrachman Bachtiar, Dr.Eng., S.T, M.Eng	33	123	28	28
Ratih Kartika Dewi, S.T., M.Kom	26	85	23	31

### C.2.3 Satisfaction

Pengguna	Nilai				Komentar
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara	
Retno Indah Rokhmawati, S.Pd., M.Pd.	6	6	5	6	<ul style="list-style-type: none"> <li>• Dua tombol pada halaman penilain membingungkan</li> <li>• Penjelan <i>gap</i> nilai lebih diperjelas.</li> <li>• Kolom untuk memasukan predikat kelulusan lebih diperjelas.</li> </ul>
Himawat Aryadita, S.T, M.Sc	6	6	3	4	<ul style="list-style-type: none"> <li>• Pada verifikasi kehadiran, lebihbaik ditampilkan semua <i>checkbox</i> kehadiran</li> <li>• Pada penilaian memiliki dua tombol yang membingungkan.</li> <li>• Pada berita acara kolom masukan untuk memilih predikat kelulusan tidak informatif</li> </ul>
Adam Hendra Brata, S.Kom., M.T., M.Sc.	6	7	7	7	<ul style="list-style-type: none"> <li>• Tambahkan rincian <i>gap</i> nilai</li> </ul>
Fitra Abdurrachman Bachtian Bachtiar, Dr.Eng., S.T, M.Eng	5	6	6	6	<ul style="list-style-type: none"> <li>• Tombol untuk mengubah kehadiran majelis lebih baik ditampilkan semua.</li> <li>• Kurang menambahkan nama komponen ketika berada di halaman komponen penilaian</li> <li>• Ditambahkan rincian nilai pada <i>gap</i></li> <li>• Predikat lebih dibuat informatif</li> </ul>
Ratih Kartika Dewi, S.T., M.Kom	7	5	7	6	<ul style="list-style-type: none"> <li>• Terdapat kekeliruan antara dua tombol pada penilaian</li> </ul>

## LAMPIRAN D RINCIAN PENGUJIAN USABILITY APLIKASI

### D.1 Effectiveness

Responden	Complete			
	Verifikasi Kehadiran	Penilaian	Rekap Nilai	Berita Acara
Lutfi Fanani, S.Kom., M.T., M.Sc.	1	1	1	1
Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.	1	1	1	1
Sabriansyah Rizqika Akbar, S.T, M.Eng	1	1	1	1
Retno Indah Rokhmawati, S.Pd., M.Pd.	1	1	1	1
Fitra Abdurrachman Bachtiar, Dr.Eng., S.T, M.Eng	1	1	1	1

### D.2 Efficiency

Pengguna	Waktu			
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara
Lutfi Fanani, S.Kom., M.T., M.Sc.	32	90	30	35
Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.	27	87	26	26
Sabriansyah Rizqika Akbar, S.T, M.Eng	45	92	38	36
Retno Indah Rokhmawati, S.Pd., M.Pd.	26	80	28	25
Fitra Abdurrachman Bachtiar, Dr.Eng., S.T, M.Eng	35	85	32	37

### D.3 Satisfaction

Pengguna	Nilai				Komentar
	Verifikasi kehadiran	Penilaian	Rekap Nilai	Berita acara	
Lutfi Fanani, S.Kom., M.T., M.Sc.	5	7	6	7	-
Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.	7	6	7	5	-
Sabriansyah Rizqika Akbar, S.T, M.Eng	5	6	5	5	-
Retno Indah Rokhmawati, S.Pd., M.Pd.	6	6	6	6	-
Fitra Abdurrachman Bachtiar, Dr.Eng., S.T, M.Eng	7	7	7	6	-