



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI E-LEARNING PADA FITUR MANAJEMEN PERKULIAHAN BERBASIS PERANGKAT BERGERAK ANDROID MENGUNAKAN PARADIGMA REACTIVE PROGRAMMING

WILLIAM SUHUD
NRP. 0511144000002

Dosen Pembimbing 1
Rizky Januar Akbar, S.Kom., M.Eng.

Dosen Pembimbing 2
Abdul Munif, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - KI141502

**RANCANG BANGUN APLIKASI E-LEARNING
PADA FITUR MANAJEMEN PERKULIAHAN
BERBASIS PERANGKAT BERGERAK ANDROID
MENGUNAKAN PARADIGMA REACTIVE
PROGRAMMING**

**WILLIAM SUHUD
NRP. 0511144000002**

**Dosen Pembimbing 1
Rizky Januar Akbar, S.Kom., M.Eng.**

**Dosen Pembimbing 2
Abdul Munif, S.Kom., M.Sc.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - KI141502

**DEVELOPMENT APPLICATION OF E-LEARNING
ON CASE: LECTURE MANAGEMENT BASED
ANDROID MOBILE APPLICATION USING
REACTIVE PROGRAMMING PARADIGM**

**WILLIAM SUHUD
NRP. 0511144000002**

**Supervisor 1
Rizky Januar Akbar, S.Kom., M.Eng.**

**Supervisor 2
Abdul Munif, S.Kom., M.Sc.**

**Informatics Department
Faculty of Information And Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI E-LEARNING PADA FITUR MANAJEMEN PERKULIAHAN BERBASIS PERANGKAT BERGERAK ANDROID MENGGUNAKAN PARADIGMA REACTIVE PROGRAMMING

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Rekayasa Perangkat Lunak
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

WILLIAM SUHUD

NRP. 5114 101 003

Disetujui oleh Pembimbing Tugas Akhir

1. Rizky Januar Akbar, S.Kom., M.Eng.
(NIP. 198701032014041001) (pembimbing 1)
2. Abdul Munif, S.Kom., M.Sc.
(NIP. 198608232015041004) (pembimbing 2)



**SURABAYA
JUNI, 2018**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI E-LEARNING PADA FITUR MANAJEMEN PERKULIAHAN BERBASIS PERANGKAT BERGERAK ANDROID MENGUNAKAN PARADIGMA REACTIVE PROGRAMMING

Nama Mahasiswa : William Suhud
NRP : 5114 100 002
Jurusan : Departemen Informatika FTIK-ITS
Dosen Pembimbing 1 : Rizky Januar Akbar, S.Kom., M.Eng.
Dosen Pembimbing 2 : Abdul Munif, S.Kom., M.Sc.

ABSTRAK

Di lingkungan ITS memiliki *e-Learning* berbasis *web* yang dapat diakses secara *realtime* dengan koneksi internet. *Platform* ini dikenal dengan Share ITS (Share.its.ac.id). namun produk ini masih memiliki beberapa kelemahan seperti tampilan kurang *responsive* ketika dijalankan pada perangkat bergerak Android. Hal ini menyebabkan pengguna akan kesulitan jika menggunakan fitur manajemen perkuliahan *e-Learning* melalui perangkat bergerak.

Kami mencoba membawa solusi baru dengan dibangunnya *platform e-Learning*. Aplikasi ini akan dilengkapi *Application Programming Interface* (API) yang akan mengambil data dari SIM AKADEMIK. Aplikasi ini juga akan ditunjang dengan layanan *web* berupa REST API yang berguna untuk memuat konten dari *e-Learning web*. Selain API aplikasi ini akan dikembangkan pada perangkat bergerak menggunakan paradigma *Reactive Programming* dengan *library RxAndroid*.

Dalam tugas akhir yang penulis ajukan ini akan dibuat *platform e-Learning* yang dapat membantu proses pembelajaran di lingkungan ITS yang mampu menjembatani kegiatan perkuliahan antara dosen, asisten dosen, serta mahasiswa.

Kata kunci: android, mobile, e-learning, pembelajaran, reactive, RxJava, reactive programming, aplikasi

DEVELOPMENT APPLICATION OF E-LEARNING ON CASE: LECTURE MANAGEMENT BASED ANDROID MOBILE APPLICATION USING REACTIVE PROGRAMMING PARADIGM

Student Name : William Suhud
NRP : 5114 100 002
Major : Informatics Department FTIK – ITS
Supervisor I : Rizky Januar Akbar, S.Kom., M.Eng.
Supervisor II : Abdul Munif, S.Kom., M.Sc.

ABSTRACT

In the ITS environment has a web-based e-Learning that can be accessed in realtime with internet connection. This platform is known as Share ITS (Share.its.ac.id). but this product still has some disadvantages such as less responsive display when run on Android mobile devices. This causes the user will be difficult if using e-Learning lecture management features through mobile devices.

We try to bring new solutions with the development of e-Learning platform. This application will be equipped with Application Programming Interface (API) which will retrieve data from ACADEMIC SIM. This application will also be supported with a web service in the form of REST API which is useful for loading content from e-Learning web. In addition to this application API will be developed on mobile devices using the Reactive Programming paradigm with RxAndroid library.

We will be create e-Learning platform that can help the learning process in the ITS environment that is able to bridge lectures between lecturers, assistant lecturers, and students.

Keywords: android, mobile, e-learning, learning, reactive, RxJava, reactive programming, applications

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah yang Maha Esa, karena atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“RANCANG BANGUN APLIKASI E-LEARNING PADA FITUR MANAJEMEN PERKULIAHAN BERBASIS PERANGKAT BERGERAK ANDROID MENGGUNAKAN PARADIGMA REACTIVE PROGRAMMING”**.

Pengerjaan tugas akhir ini penulis lakukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi S-1 Departemen Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan terima kasih atas bantuan dan semangat dari berbagai pihak, dari awal pengerjaan tugas akhir hingga selesai, antara lain:

1. Allah yang Maha Kuasa atas segala karunia dan rahmat-Nya yang telah diberikan selama ini.
2. Keluarga penulis, Ibu, dan saudara kandung serta keluarga yang tidak dapat penulis tuliskan satu per satu yang telah memberi dukungan moral, material, dan juga doa untuk penulis.
3. Bapak Rizky Januar Akbar, S.Kom., M.Eng. selaku dosen pembimbing I yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.
4. Bapak Abdul Munif, S.Kom., M.Sc. selaku dosen pembimbing II yang telah memberikan bimbingan dan arahan dalam pengerjaan tugas akhir ini.
5. Bapak Dr. Eng. Darlis Herumurti, S.Kom.,M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Dr. Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah banyak memberikan ilmu kepada penulis.
6. Teman-teman Administrator Laboratorium Rekayasa Perangkat Lunak, Penghuni RMP yang telah menjadi sahabat

dan keluarga selama penulis menimba ilmu di Teknik Informatika ITS.

7. Teman-teman angkatan 2014 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis.
8. Saudari Fitria Rizky Aprilina dan Staf DPTSI yang telah memberi dukungan dan ilmu kepada penulis.
9. Saudara Arya Putra Kurniawan, Steven Kurniawan, Prasetyo Nugrohadi yang telah memberi semangat, dukungan, dan ilmu kepada penulis.

Penulis menyadari bahwa masih terdapat banyak kekurangan dalam tugas akhir ini. Oleh karena itu, penulis menerima dengan rendah hati kritik dan saran untuk pembelajaran dan perbaikan ke depannya. Semoga tugas akhir ini dapat memberikan manfaat yang sebaik-baiknya.

Surabaya, Juni 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xxiii
DAFTAR TABEL	xxix
DAFTAR KODE SEMU.....	xxxv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Metodologi Pembuatan Tugas Akhir	4
a. Penyusunan proposal tugas akhir	4
b. Studi literatur	5
c. Analisis dan desain perangkat lunak.....	5
d. Implementasi perangkat lunak	5
f. Penyusunan buku tugas akhir.....	6
1.7 Sistematika Penulisan Laporan Tugas Akhir	7
BAB II TINJAUAN PUSTAKA.....	9
2.1 E-Learning	9
2.2 Share ITS	9
2.3 SIAKAD ITS	10
2.4 JSON.....	10
2.5 JAVA	11
2.6 ANDROID STUDIO	11
2.7 REST API	11
2.8 ANDROID.....	12
2.9 REACTIVE PROGRAMMING	13
2.10 RXJAVA/RXANDROID.....	13
2.11 MYSQL	13

2.12	MOODLE.....	14
2.13	MVP.....	14
2.14	FAST ANDROID NETWORKING (FAN).....	15
2.15	JAX-RS (Java API for RESTful Web Services)	15
2.16	WSO2 API Manager	15
2.17	Dagger 2.....	16
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		17
3.1	Analisis.....	17
3.1.1	Analisis Permasalahan.....	17
3.1.2	Deskripsi Umum Sistem.....	19
3.1.3	Alur Kerja Sistem.....	21
3.1.4	Spesifikasi Kebutuhan Perangkat Lunak.....	24
3.1.4.1	Kebutuhan Fungsional.....	25
3.1.4.2	Kebutuhan Non-Fungsional.....	25
3.1.4.3	Aktor.....	26
3.1.4.4	Kasus Penggunaan.....	26
3.1.4.4.1	Melihat Data Kelas (UC-001).....	28
3.1.4.4.2	Mengelola Data Peserta Kelas UC-002	29
3.1.4.4.3	Mengelola Tugas (UC-003).....	34
3.1.4.4.4	Mengelola Materi Perkuliahan (UC-004).....	39
3.1.4.4.5	Melihat Tugas (UC-005).....	44
3.1.4.4.6	Melihat Materi Perkuliahan (UC-006).....	46
3.1.4.4.7	Mengelola Detail Materi Perkuliahan (UC-007)	48
3.1.4.4.8	Mengelola Nilai	59
3.2	Perancangan	62
3.2.1	Lingkungan Perancangan Perangkat Lunak	62
3.2.2	Perancangan Struktur Data.....	63
3.2.2.1	Tabel <i>assign</i>	64
3.2.2.2	Tabel <i>assign_grades</i>	64
3.2.2.3	Tabel <i>assign_plugin_config</i>	64
3.2.2.4	Tabel <i>assign_submission</i>	64

3.2.2.5	Tabel <i>assign_user_flags</i>	64
3.2.2.6	Tabel <i>assignfeedback_comments</i>	64
3.2.2.7	Tabel <i>assignsubmission_onlinetext</i>	64
3.2.2.8	Tabel <i>assignsubmission_file</i>	64
3.2.2.9	Tabel <i>context</i>	65
3.2.2.10	Tabel <i>course</i>	65
3.2.2.11	Tabel <i>course_categories</i>	65
3.2.2.12	Tabel <i>course_modules</i>	65
3.2.2.13	Tabel <i>course_sections</i>	65
3.2.2.14	Tabel <i>files</i>	65
3.2.2.15	Tabel <i>glossary</i>	65
3.2.2.16	Tabel <i>glossary_entries</i>	65
3.2.2.17	Tabel <i>role_assignments</i>	66
3.2.2.18	Tabel <i>user</i>	66
3.2.2.19	Tabel <i>user_enrolments</i>	66
3.2.3	Perancangan Arsitektur Model View Presenter ...	66
3.2.3.1	Perancangan Model	67
3.2.3.2	Perancangan View	68
3.2.3.3	Perancangan Presenter.....	69
3.2.3.4	Perancangan Pemrograman Reaktif	69
3.2.3.5	Perancangan Dependency Injection	70
3.2.4	Perancangan Antarmuka Pengguna.....	71
3.2.4.1	Rancangan Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Dosen.....	72
3.2.4.2	Rancangan Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Mahasiswa.....	73
3.2.4.3	Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen	74
3.2.4.4	Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa.....	76
3.2.4.5	Rancangan Halaman Antarmuka Melihat Data Kelas	78

3.2.4.6	Rancangan Halaman Antarmuka Menambah Data Kelas	80
3.2.4.7	Rancangan Halaman Antarmuka Menghapus Data Kelas	82
3.2.4.8	Rancangan Halaman Antarmuka Menambah Materi Perkuliahan	83
3.2.4.9	Rancangan Halaman Antarmuka <i>Edit</i> Materi Perkuliahan.....	85
3.2.4.10	Rancangan Halaman Antarmuka Menghapus Materi Perkuliahan	86
3.2.4.11	Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen	88
3.2.4.12	Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa	90
3.2.4.13	Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	93
3.2.4.14	Rancangan Halaman Antarmuka <i>Edit</i> Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	95
3.2.4.15	Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	96
3.2.4.16	Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	98
3.2.4.17	Rancangan Halaman Antarmuka <i>Edit</i> Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	100
3.2.4.18	Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	101
3.2.4.19	Rancangan Halaman Antarmuka Menambah Tugas	103
3.2.4.20	Rancangan Halaman Antarmuka <i>Edit</i> Tugas.	104

3.2.4.21	Rancangan Halaman Antarmuka Menghapus Tugas.....	106
3.2.4.22	Rancangan Halaman Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Dosen	107
3.2.4.23	Rancangan Halaman Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa.....	108
3.2.4.24	Rancangan Halaman Antarmuka Melihat Detail Tugas.....	111
3.2.4.25	Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen.....	113
3.2.4.26	Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa	114
3.2.4.27	Rancangan Halaman Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Dosen	116
3.2.4.28	Rancangan Halaman Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Mahasiswa	119
BAB IV IMPLEMENTASI.....		121
4.1	Lingkungan Implementasi Perangkat Lunak	121
4.2	Implementasi Antarmuka Pengguna	122
4.2.1	Implementasi Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Dosen	122
4.2.2	Implementasi Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Mahasiswa	123
4.2.3	Implementasi Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen	124
4.2.4	Implementasi Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa.....	125
4.2.5	Implementasi Halaman Antarmuka Melihat Data Kelas.....	126

4.2.6	Implementasi Halaman Antarmuka Menambah Data Kelas.....	127
4.2.7	Implementasi Halaman Antarmuka Menghapus Data Kelas	128
4.2.8	Implementasi Halaman Antarmuka Menambah Materi Perkuliahan	129
4.2.9	Implementasi Halaman Antarmuka <i>Edit</i> Materi Perkuliahan.....	130
4.2.10	Implementasi Halaman Antarmuka Menghapus Materi Perkuliahan	131
4.2.11	Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen	132
4.2.12	Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa ..	133
4.2.13	Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	134
4.2.14	Implementasi Halaman Antarmuka <i>Edit</i> Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	135
4.2.15	Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	136
4.2.16	Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	137
4.2.17	Implementasi Halaman Antarmuka <i>Edit</i> Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	138
4.2.18	Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	139
4.2.19	Implementasi Halaman Antarmuka Menambah Tugas	140
4.2.20	Implementasi Halaman Antarmuka <i>Edit</i> Tugas .	141

4.2.21	Implementasi Halaman Antarmuka Menghapus Tugas	142
4.2.22	Implementasi Halaman Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Dosen 143	
4.2.23	Implementasi Halaman Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa.....	144
4.2.24	Implementasi Halaman Antarmuka Melihat Detail Tugas	145
4.2.25	Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen	145
4.2.26	Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa	147
4.2.27	Implementasi Halaman Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Dosen	148
4.2.28	Implementasi Halaman Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Mahasiswa... 149	
4.3	Implementasi Kasus Penggunaan	150
4.3.1	Implementasi Kasus Penggunaan Pengelolaan Data Kelas.....	150
4.3.2	Implementasi Kasus Penggunaan Pengelolaan Data Peserta Kelas	153
4.3.3	Implementasi Kasus Penggunaan Pengelolaan Data Tugas	155
4.3.4	Implementasi Kasus Penggunaan Pengelolaan Materi Perkuliahan	158
4.3.5	Implementasi Kasus Penggunaan Melihat Tugas 160	
4.3.6	Implementasi Kasus Penggunaan Melihat Materi Perkuliahan.....	161
4.3.7	Implementasi Kasus Penggunaan Pengelolaan Detail Materi Perkuliahan	162
4.4	Implementasi REST API.....	164
4.4.1	Koneksi ke Database	166

4.4.2	Dependency Injection.....	167
4.4.2.1	data-beans.xml.....	167
4.4.2.2	service-beans.xml.....	167
4.4.2.3	endpoint-beans.xml.....	168
4.4.3	Error Handler.....	168
4.4.4	Model.....	168
4.4.5	DAO.....	169
4.4.5.1	id.ac.its.api.ekelas.dao.....	169
4.4.5.2	id.ac.its.api.ekelas.dao.jdbc.....	169
4.4.5.3	id.ac.its.api.ekelas.dao.mapper.....	169
4.4.6	Service.....	169
4.4.7	Endpoint.....	169
BAB V PENGUJIAN DAN EVALUASI.....		173
5.1	Lingkungan Pengujian.....	173
5.2	Pengujian Fungsionalitas.....	174
5.2.1.	Pengujian Fungsionalitas Melihat <i>List</i> Kelas.....	174
5.2.2.	Pengujian Fungsionalitas Melihat Halaman Utama Kelas.....	175
5.2.3.	Pengujian Fungsionalitas Melihat Data Kelas.....	175
5.2.4.	Pengujian Fungsionalitas Menambah Data Kelas	176
5.2.5.	Pengujian Fungsionalitas Menghapus Data Kelas	176
5.2.6.	Pengujian Fungsionalitas Menambah Materi Perkuliahan.....	177
5.2.7.	Pengujian Fungsionalitas Melihat Detail Materi Perkuliahan.....	178
5.2.8.	Pengujian Fungsionalitas <i>Edit</i> Materi Perkuliahan	179
5.2.9.	Pengujian Fungsionalitas Menghapus Materi Perkuliahan.....	180
5.2.10.	Pengujian Fungsionalitas Menambah Tugas.....	180
5.2.11.	Pengujian Fungsionalitas Melihat Detail Tugas.	181
5.2.12.	Pengujian Fungsionalitas <i>Edit</i> Tugas.....	182
5.2.13.	Pengujian Fungsionalitas Menghapus Tugas.....	183

5.2.14.	Pengujian Fungsionalitas Melihat <i>List</i> Pengumpulan Tugas	184
5.2.15.	Pengujian Fungsionalitas Menilai Tugas	184
5.2.16.	Pengujian Fungsionalitas Melihat <i>List</i> Nilai Tugas	185
5.2.17.	Pengujian Fungsionalitas Menambah Detail Materi Perkuliahan.....	186
5.2.18.	Pengujian Fungsionalitas <i>Edit</i> Detail Materi Perkuliahan.....	187
5.2.19.	Pengujian Fungsionalitas Menghapus Detail Materi Perkuliahan.....	188
5.3	Pengujian REST API	189
5.3.1	Pengujian Response Body	189
5.3.1.1	Skenario Pengujian.....	190
5.3.1.2	Pengujian Kasus Penggunaan Melihat Data Kelas	191
5.3.1.3	Pengujian Kasus Penggunaan Mengelola Data Peserta Kelas	192
5.3.1.4	Pengujian Kasus Penggunaan Mengelola Data Tugas.....	194
5.3.1.5	Pengujian Kasus Penggunaan Mengelola Materi Perkuliahan.....	196
5.3.1.6	Pengujian Kasus Penggunaan Melihat Tugas	198
5.3.1.7	Pengujian Kasus Penggunaan Melihat Materi Perkuliahan.....	199
5.3.1.8	Pengujian Kasus Penggunaan Mengelola Detail Materi Perkuliahan	200
5.3.2	Pengujian Response Time	202
5.3.2.1	Pengujian 100 Pengguna	203
5.3.2.2	Pengujian 500 Pengguna	203
5.3.2.3	Pengujian 1000 Pengguna	204
5.3.2.4	Pengujian 2000 Pengguna	204
5.4	Pengujian Kepuasan Pengguna	205
5.5	Evaluasi Pengujian.....	211
5.5.1	Evaluasi Pengujian Fungsionalitas.....	211

5.6	Evaluasi Pengujian REST API.....	213
5.7	Evaluasi Pengujian Kepuasan Pengguna	215
BAB VI KESIMPULAN DAN SARAN.....		217
6.1	Kesimpulan	217
6.2	Saran	217
DAFTAR PUSTAKA.....		219
LAMPIRAN		223
BIODATA PENULIS.....		231

DAFTAR GAMBAR

Gambar 2.1. JSON Key-Value	10
Gambar 3.1 Deskripsi Umum Sistem.....	19
Gambar 3.2. Alur Kerja Sistem Dosen.....	22
Gambar 3.3. Alur Kerja Sistem Asisten Dosen.....	23
Gambar 3.4. Alur Kerja Sistem Mahasiswa	24
Gambar 3.5 Diagram Kasus Penggunaan	27
Gambar 3.6. Diagram Kasus Penggunaan UC-001	28
Gambar 3.7. Diagram Aktivitas UC-001.....	29
Gambar 3.8. Diagram Kasus Penggunaan (UC-002)	29
Gambar 3.9. Diagram Aktivitas UC-002 Tambah User	32
Gambar 3.10. Diagram Aktivitas UC-002 Hapus User.....	33
Gambar 3.11. Diagram Kasus Penggunaan UC-003	34
Gambar 3.12. Diagram Aktivitas UC-003 Menambah Tugas	36
Gambar 3.13. Diagram Aktivitas UC-003 Mengedit Tugas.....	37
Gambar 3.14. Diagram Aktivitas UC-003 Menghapus Tugas	38
Gambar 3.15. Diagram Kasus Penggunaan UC-004	39
Gambar 3.16. Diagram Aktivitas UC-004 Menambah Materi	41
Gambar 3.17. Diagram Aktivitas UC-004 Mengedit Materi.....	42
Gambar 3.18. Diagram Aktivitas UC-004 Menghapus Materi	43
Gambar 3.19. Diagram Kasus Penggunaan UC-005	44
Gambar 3.20. Diagram Aktivitas UC-005.....	46
Gambar 3.21. Diagram Kasus Penggunaan UC-006.....	46
Gambar 3.22. Diagram Aktivitas UC-006.....	47
Gambar 3.23. Diagram Kasus Penggunaan UC-007	48
Gambar 3.24. Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Melihat Detail Materi	52
Gambar 3.25 Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Menambah Detail Materi.....	53
Gambar 3.26. Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Mengedit Detail Materi	54
Gambar 3.27. Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Menghapus Detail Materi.....	55
Gambar 3.28. Diagram Aktivitas UC-007 Mahasiswa.....	58

Gambar 3.29. Diagram Kasus Penggunaan UC-008	59
Gambar 3.30. Diagram Aktivitas UC-008 Menilai dan Melihat List Nilai	61
Gambar 3.31. Diagram Aktivitas UC-008 List Pengumpulan Tugas	62
Gambar 3.32. Diagram Struktur Model.....	67
Gambar 3.33. Diagram Struktur View.....	68
Gambar 3.34. Diagram Struktur Presenter	69
Gambar 3.35. Diagram Perancangan Pemrograman Reaktif.....	69
Gambar 3.36. Diagram Struktur Dependency Injection	70
Gambar 3.37. Rancangan Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Dosen	72
Gambar 3.38. Rancangan Halaman Antarmuka Melihat List Kelas pada Aplikasi MyITS Mahasiswa.....	73
Gambar 3.39. Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen.....	75
Gambar 3.40. Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa.....	77
Gambar 3.41. Rancangan Halaman Antarmuka Melihat Data Kelas	79
Gambar 3.42. Rancangan Halaman Antarmuka Menambah Data Kelas.....	81
Gambar 3.43. Rancangan Halaman Antarmuka Menghapus Data Kelas	82
Gambar 3.44. Rancangan Halaman Antarmuka Menambah Materi Perkuliahan	84
Gambar 3.45. Rancangan Halaman Antarmuka Edit Materi Perkuliahan	86
Gambar 3.46. Rancangan Halaman Antarmuka Menghapus Materi Perkuliahan	87
Gambar 3.47. Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen	90
Gambar 3.48. Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	92

Gambar 3.49. Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	94
Gambar 3.50. Rancangan Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	96
Gambar 3.51. Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	97
Gambar 3.52. Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	99
Gambar 3.53. Rancangan Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa	101
Gambar 3.54. Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	102
Gambar 3.55. Rancangan Halaman Antarmuka Menambah Tugas	104
Gambar 3.56. Rancangan Halaman Antarmuka Edit Tugas.....	105
Gambar 3.57. Rancangan Halaman Antarmuka Menghapus Tugas	106
Gambar 3.58. Rancangan Halaman Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Dosen	108
Gambar 3.59. Rancangan Halaman Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa.....	110
Gambar 3.60. Rancangan Halaman Antarmuka Melihat Detail Tugas	112
Gambar 3.61. Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen.....	114
Gambar 3.62. Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa.....	116
Gambar 3.63. Rancangan Halaman Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Dosen	118
Gambar 3.64. Rancangan Halaman Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Mahasiswa.....	120
Gambar 4.1. Implementasi Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Dosen	122
Gambar 4.2. Implementasi Halaman Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Mahasiswa	123

Gambar 4.3. Implementasi Antarmuka Halaman Utama Kelas pada Aplikasi MyITS Dosen.....	124
Gambar 4.4. Implementasi Antarmuka Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa	125
Gambar 4.5. Implementasi Halaman Antarmuka Melihat Data Kelas.....	126
Gambar 4.6. Implementasi Halaman Antarmuka Menambah Data Kelas	127
Gambar 4.7. Implementasi Halaman Antarmuka Menghapus Data Kelas.....	128
Gambar 4.8. Implementasi Halaman Antarmuka Menambah Materi Perkuliahan	129
Gambar 4.9. Implementasi Halaman <i>Edit</i> Materi Perkuliahan..	130
Gambar 4.10. Implementasi Halaman Antarmuka Menghapus Materi Perkuliahan	131
Gambar 4.11. Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen	132
Gambar 4.12. Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	133
Gambar 4.13. Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	134
Gambar 4.14. Implementasi Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	135
Gambar 4.15. Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.....	136
Gambar 4.16. Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa ...	137
Gambar 4.17. Implementasi Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	138
Gambar 4.18. Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa ...	139
Gambar 4.19. Implementasi Halaman Antarmuka Menambah Tugas	140
Gambar 4.20. Implementasi Halaman Antarmuka Edit Tugas ..	141

Gambar 4.21. Implementasi Halaman Antarmuka Menghapus Tugas	142
Gambar 4.22. Implementasi Halaman Antarmuka Melihat List Pengumpulan Tugas pada Aplikasi MyITS Dosen	143
Gambar 4.23. Implementasi Halaman Antarmuka Melihat List Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa.....	144
Gambar 4.24. Implementasi Halaman Antarmuka Melihat Detail Tugas	145
Gambar 4.25. Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen	146
Gambar 4.26. Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa	147
Gambar 4.27. Implementasi Halaman Antarmuka Melihat List Nilai Tugas pada Aplikasi MyITS Dosen	148
Gambar 4.28. Implementasi Halaman Antarmuka Melihat List Nilai Tugas pada Aplikasi MyITS Mahasiswa.....	149
Gambar 4.29. Struktur Direktori Project REST API.....	166
Gambar 8.1. Conceptual Data Model User	223
Gambar 8.2. Conceptual Data Model Assign.....	224
Gambar 8.3. Conceptual Data Model Glossary.....	225
Gambar 8.4. Conceptual Data Model Course.....	226
Gambar 8.5. Physical Data Model User	227
Gambar 8.6. Physical Data Model Assign.....	228
Gambar 8.7. Physical Data Model Glossary	229
Gambar 8.8. Physical Data Model Course	230

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Metode HTTP dan Penggunaannya dalam REST	12
Tabel 2.2. Annotasi pada Dagger2	16
Tabel 3.1. Perbandingan Sistem	18
Tabel 3.2. Kebutuhan Fungsional.....	25
Tabel 3.3. Kasus Penggunaan.....	26
Tabel 3.4. Spesifikasi Kasus Penggunaan UC-001	28
Tabel 3.5. Spesifikasi Kasus Penggunaan UC-002	30
Tabel 3.6. Spesifikasi Kasus Penggunaan UC-003	34
Tabel 3.7. Spesifikasi Kasus Penggunaan UC-004	39
Tabel 3.8. Spesifikasi Kasus Penggunaan UC-005	44
Tabel 3.9. Spesifikasi Kasus Penggunaan UC-006	47
Tabel 3.10. Spesifikasi Kasus Penggunaan UC-007 Dosen dan Asisten Dosen.....	48
Tabel 3.11. Spesifikasi Kasus Penggunaan UC-007 Mahasiswa	55
Tabel 3.12. Spesifikasi Kasus Penggunaan UC-008	59
Tabel 3.13 Lingkungan Perancangan Perangkat Lunak	62
Tabel 3.14. Atribut Antarmuka Melihat <i>List</i> Kelas pada Aplikasi MyITS Dosen	72
Tabel 3.15. Atribut Antarmuka Melihat List Kelas pada Aplikasi MyITS Mahasiswa	73
Tabel 3.16. Atribut Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen	74
Tabel 3.17. Atribut Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa	76
Tabel 3.18. Atribut Antarmuka Melihat Data Kelas	78
Tabel 3.19. Atribut Antarmuka Menambah Data Kelas	80
Tabel 3.20. Atribut Antarmuka Menghapus Data Kelas	82
Tabel 3.21. Atribut Antarmuka Menambah Materi Perkuliahan.	83
Tabel 3.22. Atribut Antarmuka Edit Materi Perkuliahan	85
Tabel 3.23. Atribut Antarmuka Menghapus Materi Perkuliahan	87
Tabel 3.24. Atribut Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen.....	88

Tabel 3.25. Atribut Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa	91
Tabel 3.26. Atribut Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	93
Tabel 3.27. Atribut Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	95
Tabel 3.28. Atribut Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen	97
Tabel 3.29. Atribut Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	98
Tabel 3.30. Atribut Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	100
Tabel 3.31. Atribut Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa.....	102
Tabel 3.32. Atribut Antarmuka Menambah Tugas	103
Tabel 3.33. Atribut Antarmuka Edit Tugas	105
Tabel 3.34. Atribut Antarmuka Menghapus Tugas	106
Tabel 3.35. Atribut Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Dosen	107
Tabel 3.36. Atribut Antarmuka Melihat <i>List</i> Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa.....	109
Tabel 3.37. Atribut Antarmuka Melihat Detail Tugas.....	111
Tabel 3.38. Atribut Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen	113
Tabel 3.39. Atribut Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa.....	115
Tabel 3.40. Atribut Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Dosen.....	117
Tabel 3.41. Atribut Antarmuka Melihat <i>List</i> Nilai Tugas pada Aplikasi MyITS Mahasiswa	119
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak	121
Tabel 4.2. <i>List</i> Injeksi Objek ke kelas Implementasi.....	168
Tabel 4.3. <i>List</i> Injeksi Objek ke Endpoint.....	168
Tabel 4.4. <i>List</i> Endpoint	170

Tabel 5.1. Lingkungan Pengujian Fungsionalitas Perangkat Lunak	173
Tabel 5.2. Lingkungan Pengujian Fungsionalitas Perangkat Lunak	173
Tabel 5.3. Lingkungan Pengujian <i>Server</i> Basis Data	173
Tabel 5.4 Skenario Pengujian Fungsionalitas Melihat <i>List</i> Kelas	174
Tabel 5.5 Skenario Pengujian Fungsionalitas Melihat Halaman Utama Kelas	175
Tabel 5.6 Skenario Pengujian Fungsionalitas Melihat Data Kelas	175
Tabel 5.7 Skenario Pengujian Fungsionalitas Menambah Data Kelas.....	176
Tabel 5.8. Skenario Pengujian Fungsionalitas Menghapus Data Kelas.....	177
Tabel 5.9. Skenario Pengujian Fungsionalitas Menambah Materi Perkuliahan.....	177
Tabel 5.10. Skenario Pengujian Fungsionalitas Melihat Detail Materi Perkuliahan	178
Tabel 5.11. Skenario Pengujian Fungsionalitas Edit Materi Perkuliahan.....	179
Tabel 5.12. Skenario Pengujian Fungsionalitas Menghapus Materi Perkuliahan.....	180
Tabel 5.13. Skenario Pengujian Fungsionalitas Menambah Tugas	181
Tabel 5.14. Skenario Pengujian Fungsionalitas Melihat Detail Tugas	181
Tabel 5.15. Skenario Pengujian Fungsionalitas Edit Tugas	182
Tabel 5.16. Skenario Pengujian Fungsionalitas Menghapus Tugas	183
Tabel 5.17. Skenario Pengujian Fungsionalitas Melihat List Pengumpulan Tugas	184
Tabel 5.18. Skenario Pengujian Fungsionalitas Menilai Tugas	184
Tabel 5.19. Skenario Pengujian Fungsionalitas Melihat List Nilai Tugas	185

Tabel 5.20. Skenario Pengujian Fungsionalitas Menambah Detail Materi Perkuliahan	186
Tabel 5.21. Skenario Pengujian Fungsionalitas Edit Detail Materi Perkuliahan	187
Tabel 5.22. Skenario Pengujian Fungsionalitas Menghapus Detail Materi Perkuliahan	188
Tabel 5.23. Skenario Pengujian Response Body	190
Tabel 5.24. Pengujian GET <i>List Class Per User</i>	191
Tabel 5.25. Pengujian GET <i>Detail Class</i>	191
Tabel 5.26. Pengujian GET <i>List User Per Class</i>	192
Tabel 5.27. Pengujian GET Jumlah Mahasiswa Di Kelas	192
Tabel 5.28. Pengujian GET <i>Detail User From SIAKAD</i>	193
Tabel 5.29. Pengujian POST <i>User to Moodle and Class</i>	193
Tabel 5.30. Pengujian DELETE <i>User from Class</i>	193
Tabel 5.31. Pengujian GET <i>List Pengumpulan Tugas</i>	194
Tabel 5.32. Pengujian GET <i>List Nilai Tugas</i>	194
Tabel 5.33. Pengujian POST Nilai Tugas.....	195
Tabel 5.34. Pengujian POST Tugas.....	195
Tabel 5.35. Pengujian EDIT Tugas	196
Tabel 5.36. Pengujian DELETE Tugas	196
Tabel 5.37. Pengujian GET Materi Perkuliahan.....	197
Tabel 5.38. Pengujian POST Materi Perkuliahan.....	197
Tabel 5.39. Pengujian EDIT Materi Perkuliahan	197
Tabel 5.40. Pengujian DELETE Materi Perkuliahan	198
Tabel 5.41. Pengujian GET Detail Tugas.....	198
Tabel 5.42. Pengujian GET <i>List Tugas yang Sudah Dikumpulkan</i>	199
Tabel 5.43. Pengujian GET Materi Perkuliahan.....	199
Tabel 5.44. Pengujian GET Detail Materi Perkuliahan.....	200
Tabel 5.45. Pengujian GET <i>Files Per Detail Materi Perkuliahan</i>	200
Tabel 5.46. Pengujian POST Detail Materi Perkuliahan.....	201
Tabel 5.47. Pengujian EDIT Detail Materi Perkuliahan.....	201
Tabel 5.48. Pengujian DELETE Detail Materi Perkuliahan.....	202
Tabel 5.49. Hasil Pengujian <i>Response Time</i> 100 Pengguna.....	203

Tabel 5.50. Hasil Pengujian Response Time 500 Pengguna	204
Tabel 5.51. Hasil Pengujian Response Time 1000 Pengguna ...	204
Tabel 5.52. Hasil Pengujian Response Time 2000 Pengguna ...	204
Tabel 5.53. Tingkat Penilaian.....	205
Tabel 5.54. Skenario Uji Coba Kepuasan Pengguna.....	205
Tabel 5.55. Rekapitulasi Hasil Penilaian.....	207
Tabel 5.56. Rekapitulasi Hasil Penilaian Dosen	209
Tabel 5.57. Hasil Pengujian Fungsionalitas	211
Tabel 5.58. Hasil Pengujian <i>Base Response</i>	213
Tabel 5.59. Perbandingan Hasil Pengujian <i>Response Time</i>	215
Tabel 5.60. Rata-Rata Nilai Responden	216
Tabel 5.61. Rata-Rata Total Nilai	216

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SEMU

Kode Semu 4.3.1. Kasus Penggunaan Menampilkan <i>List</i> Kelas	152
Kode Semu 4.3.2. Kasus Penggunaan Menampilkan Halaman Utama Kelas	153
Kode Semu 4.3.3. Kasus Penggunaan Menampilkan <i>List</i> Peserta Kelas.....	154
Kode Semu 4.3.4. Kasus Penggunaan Menampilkan Jumlah Mahasiswa yang Mengikuti Kelas	154
Kode Semu 4.3.5. Kasus Penggunaan Menambah Data kelas ..	155
Kode Semu 4.3.6. Kasus Penggunaan Menghapus Data Kelas.	155
Kode Semu 4.3.7. Kasus Penggunaan Menampilkan <i>List</i> Pengumpulan Tiap Tugas.....	156
Kode Semu 4.3.8. Kasus Penggunaan Menampilkan Detail Tugas	156
Kode Semu 4.3.9. Kasus Penggunaan Menampilkan <i>List</i> Nilai Tugas	157
Kode Semu 4.3.10. Kasus Penggunaan Menilai Tugas.....	157
Kode Semu 4.3.11. Kasus Penggunaan Menambah Tugas	158
Kode Semu 4.3.12. Kasus Penggunaan Mengubah Tugas	158
Kode Semu 4.3.13. Kasus Penggunaan Menghapus Tugas.....	158
Kode Semu 4.3.14. Kasus Penggunaan Menampilkan Materi Perkuliahan.....	159
Kode Semu 4.3.15. Kasus Penggunaan Menambah Materi Perkuliahan.....	159
Kode Semu 4.3.16. Kasus Penggunaan Mengubah Materi Perkuliahan.....	160
Kode Semu 4.3.17. Kasus Penggunaan Menghapus Materi Perkuliahan.....	160
Kode Semu 4.3.18. Kasus Penggunaan Menampilkan List Tugas yang sudah Dikumpulkan	161
Kode Semu 4.3.19. Kasus Penggunaan Menampilkan Detail Tugas	161

Kode Semu 4.3.20. Kasus Penggunaan Menampilkan Materi Perkuliahan	162
Kode Semu 4.3.21. Kasus Penggunaan Menambah Detail Materi Perkuliahan	163
Kode Semu 4.3.22. Kasus Penggunaan Mengedit detail materi perkuliahan	163
Kode Semu 4.3.23. Kasus Penggunaan Menghapus detail materi perkuliahan	164
Kode Semu 4.3.24. Kasus Penggunaan Menampilkan detail materi perkuliahan	164

BAB I

PENDAHULUAN

Bab ini menjelaskan garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan, batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

e-Learning sebagai salah satu metode pembelajaran di perkuliahan yang sangat potensial. Tak jarang banyak instansi pendidikan yang menggunakan *platform e-Learning* sebagai media belajar para mahasiswanya. Apabila dipergunakan dengan sepenuhnya, *e-Learning* bisa menjadi alat bantu proses belajar yang efektif. Karena peserta didik bisa belajar dengan mandiri dan dari tempat yang tidak terbatas, selama masih ada koneksi internet.

Di Institut Teknologi Sepuluh Nopember (ITS) Surabaya, sistem belajar mengajar dilakukan dengan tatap muka di kelas dan sistem *e-Learning* yang masih berbasis *website* yaitu *share.its.ac.id*. Kegiatan belajar mengajar dengan sistem *e-Learning* yang masih hanya berbasis *website* sering membuat mahasiswa malas untuk menggunakannya karena harus membuka komputer/laptop terlebih dahulu untuk mengaksesnya. Tampilan *website* sistem *e-Learning* yang ada juga dirasa kurang *responsive* apabila digunakan di perangkat bergerak Android.

Kami mencoba membawa solusi baru dengan dibangunnya *platform e-Learning* berbasis perangkat bergerak yang berbasis Android. Alasan utama dipilihnya *platform* Android ini yaitu *smartphone* berbasis Android memegang posisi teratas untuk kategori *smartphone/tablet*. Dibuktikan perkembangan produk Android cukup cepat dan sangat signifikan. Menurut StatCounter, sejak Mei 2017 hingga Mei 2018 di Indonesia, *market share* produk Android mencapai angka 91,15% [1]. Mudahnya pengaplikasian dan akses secara gratis menjadi nilai tambah bagi sistem operasi Android ini. Pembangunan *platform mobile e-Learning* ini didukung dengan Android Studio sebagai *tools*

pembuatan aplikasi yang *user friendly* yang menyediakan berbagai fitur penunjang aplikasi dan dokumentasi yang lengkap [2].

Aplikasi ini akan dilengkapi *Application Programming Interface* (API) yang akan mengambil data dari SIM AKADEMIK untuk sinkronisasi tiap kelas yang ada di basis data pada seluruh jurusan di ITS. Caranya yaitu setiap *user* yang akan mengakses aplikasi wajib *login* untuk melakukan otorisasi ke Integra dengan NRP atau NIP masing-masing sehingga aplikasi dapat tersinkronisasi dengan SIAKAD AKADEMIK, SIM AKADEMIK, serta Integra ITS. *Login* ini menggunakan protokol OAuth2 yaitu sebuah protokol yang mengaktifkan fitur untuk pembatasan hak akses ke suatu HTTP *service* tertentu. Untuk dapat mengakses HTTP yang diinginkan diharuskan memasukkan kode akses untuk mendapat persetujuan terhadap keotentikan dan kesesuaian kode akses dengan basis data sistem. Aplikasi ini juga akan ditunjang dengan layanan *web* berupa REST API yang berguna untuk memuat konten dari *e-Learning web* dan akan disajikan ke *user* dalam bentuk antarmuka aplikasi dengan beban komunikasi yang ringan.

Selain API, aplikasi ini akan dikembangkan menggunakan paradigma *Reactive Programming*, yaitu pola pemrograman yang berbeda dengan pola pemrograman secara prosedural. Bila melakukan suatu pekerjaan, pada pemrograman prosedural, *user* harus melakukan *request* terus menerus sehingga menyebabkan *blocking thread*. Sedangkan pada *Reactive Programming*, *user* tidak perlu meminta *request* secara aktif atau terus menerus, karena *thread* akan mendelegasikan *task* yang diminta oleh *user* ke *thread* lainnya. *User* dapat melakukan *task* lain sambil menunggu notifikasi dari *task* yang sedang diproses oleh *thread* (*asynchronous*).

Dalam tugas akhir yang penulis ajukan ini akan dibuat *platform e-Learning* yang dapat membantu proses pembelajaran di lingkungan ITS yang mampu menjembatani kegiatan perkuliahan antara dosen, asisten dosen, serta mahasiswa. Dengan konsep REST API dan paradigma *Reactive Programming*, aplikasi *e-*

Learning ini dapat mensinkronisasikan *web e-Learning* dengan SIAKAD (yang terdiri dari Integra API dan Akademik API). Data dari REST API ini akan diimplementasikan pada perangkat bergerak Android dengan menggunakan paradigma *Reactive Programming*. Hal ini bertujuan untuk mempermudah akses *e-Learning* dari berbagai tempat selama masih ada koneksi internet.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana menerapkan konsep *REST API* untuk fitur manajemen perkuliahan pada *platform e-Learning* yang sudah ada?
2. Bagaimana menerapkan API untuk mengambil data mahasiswa dari SIAKAD ITS?
3. Bagaimana mengimplementasikan fitur manajemen perkuliahan aplikasi *e-Learning* pada perangkat bergerak Android dengan menggunakan paradigma *Reactive Programming*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Aplikasi *e-Learning* ini digunakan untuk menunjang kegiatan perkuliahan di Institut Teknologi Sepuluh Nopember Surabaya.
2. Teknologi yang digunakan dalam pembuatan aplikasi ini adalah bahasa pemrograman Java, serta *web service* REST API, serta *library* RxAndroid.
3. Aplikasi ini hanya dapat digunakan untuk mengakses konten pembelajaran saja, sedangkan untuk fitur FRS tetap dilakukan di SIAKAD.

1.4 Tujuan

Tugas akhir ini mempunyai beberapa tujuan, yaitu sebagai berikut:

1. Menerapkan API untuk mengambil data mahasiswa dari SIAKAD ITS.
2. Mengimplementasikan REST API untuk mengambil data fitur manajemen perkuliahan dari Moodle *web* yang sudah ada.
3. Membangun aplikasi *e-Learning* pada perangkat bergerak Android untuk membuat aplikasi *mobile* yang *user friendly* namun bermanfaat di lingkungan ITS.
4. Mengimplementasikan paradigma RxAndroid pada *platform e-Learning mobile*.

1.5 Manfaat

Tugas akhir ini diharapkan dapat menjadi *platform e-Learning* yang mendukung proses belajar mengajar di lingkungan ITS, terutama pada fitur manajemen perkuliahan. Tugas akhir ini akan menerapkan konsep REST API dan paradigma *Reactive Programming* pada perangkat bergerak Android sehingga lebih mudah menyinkronisasikan *platform e-Learning* pada *web* dengan versi *mobile* ini, sehingga *user* dapat mudah mengakses *platform e-Learning* di berbagai tempat, tentunya dengan adanya akses internet.

1.6 Metodologi Pembuatan Tugas Akhir

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

a. Penyusunan proposal tugas akhir

Tahap pertama dalam proses pengerjaan tugas akhir ini adalah menyusun proposal tugas akhir. Pada proposal

tugas akhir ini diajukan Aplikasi *e-Learning* berbasis perangkat bergerak *Android* dengan penerapan *REST API* untuk menunjang kegiatan perkuliahan.

b. Studi literatur

Pada tahap ini, akan dicari studi literatur yang relevan untuk dijadikan referensi dalam pengerjaan tugas akhir. Studi literatur ini didapatkan dari *ebook* yang berjudul “*Web API Design*” yang ditulis oleh Brian Mulloy [3]. Sumber lain berasal dari dokumentasi internet pada *web* “*Moodle*” [4]. Tersedia juga dokumentasi resmi GitHub dari *ReactiveX* untuk pengimplementasian *RxAndroid* [5]. Dan materi pendukung berasal dari materi-materi kuliah yang berhubungan dengan metode yang akan digunakan.

c. Analisis dan desain perangkat lunak

Analisis kebutuhan dan perancangan sistem dilakukan untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Tahap desain meliputi arsitektur perangkat lunak yang digunakan, desain kelas-kelas yang terlibat dalam aplikasi, desain antarmuka, serta diagram-diagram yang mendukung pendeskripsian sistem aplikasi.

d. Implementasi perangkat lunak

Pembangunan aplikasi *e-Learning* akan dilakukan dengan menggunakan basis data *MySQL*, bahasa pemrograman *Java*, serta *REST API*.

e. Pengujian dan evaluasi

Pengujian akan dilakukan kepada beberapa mahasiswa dan dosen ITS untuk mengetahui tingkat keberhasilan pada fitur manajemen perkuliahan di aplikasi e-Learning yang dibangun sudah berjalan dengan baik dan tidak ada *error* yang terjadi. Model pengujian dibagi menjadi 2 bagian, yaitu pengujian untuk REST API dan pengujian untuk aplikasi Android.

Pengujian pada REST API dilakukan untuk menguji kebenaran proses API dengan aplikasi Postman dan Jmeter. Pengujian pada aplikasi Android dilakukan dengan *blackbox testing* yang diuji pada beberapa *smartphone* atau tablet dengan sistem operasi Android.

f. Penyusunan buku tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini. Pada tahap ini juga disertakan hasil dari implementasi metode dan algoritma yang telah dibuat. Sistematika penulisan buku tugas akhir ini secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran

6. Daftar Pustaka

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu metodologi yang digunakan dan sistematika penulisan laporan akhir juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini berisi tentang analisis permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, lingkungan perancangan, perancangan arsitektur sistem, diagram kelas, dan struktur data.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode sumber yang digunakan untuk proses implementasi.

Bab V Pengujian dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan saran untuk pengembangan perangkat lunak selanjutnya.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan teori-teori yang berkaitan dengan pembangunan aplikasi mobile ekelas yang diajukan untuk tugas akhir ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap perangkat lunak yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 E-Learning

E-learning adalah suatu sistem atau konsep pendidikan yang memanfaatkan teknologi informasi dalam proses belajar mengajar. karakteristik *e-learning* bersifat jaringan, yang membuatnya mampu memperbaiki secara cepat, menyimpan atau memunculkan kembali, mendistribusikan, dan *sharing* pembelajaran dan informasi [6]. Pada tugas akhir ini, *e-Learning* berkaitan dengan *Learning Management System* (LMS) yang merupakan sistem perangkat lunak untuk mem-virtualisasikan proses belajar mengajar konvensional untuk konten manajemen perkuliahan [7].

2.2 Share ITS

Share ITS adalah sistem *e-Learning* berbasis *web* yang dimiliki oleh Institut Teknologi Sepuluh Nopember Surabaya. Sistem ini mengadaptasi *platform e-Learning* yang sudah banyak digunakan yaitu Moodle. Share ITS dapat diakses oleh beberapa aktor seperti dosen, mahasiswa, *admin*. Sistem ini memiliki beberapa fungsi operasional yang dapat diakses seperti manajemen pembelajaran, manajemen pengguna, manajemen sumber belajar, manajemen aktivitas, manajemen soal/kuis. Memiliki layanan proses pembelajaran yang dapat digunakan pada bidang kurikulum. Layanan berisi perencanaan proses pembelajaran yang terdiri dari rencana pembelajaran, rancangan pembelajaran *online*, partisipasi mata kuliah, sumber belajar mata kuliah, aktivitas pembelajaran berbasis Share ITS, pelaksanaan proses pembelajaran, penilaian hasil belajar, pengawasan proses pembelajaran [8].

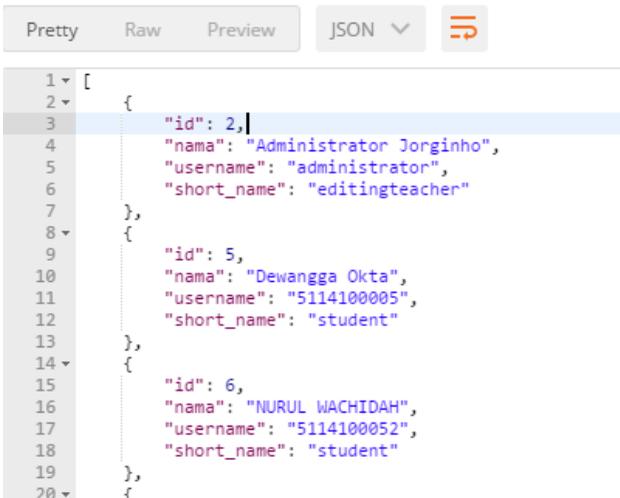
2.3 SIAKAD ITS

SIAKAD ITS adalah sistem akademik Institut Teknologi Sepuluh Nopember Surabaya yang merupakan bagian dari *integra.its.ac.id*. Sistem ini digunakan untuk proses instansiasi data *user* untuk mengisi tabel *user* pada sistem *e-Learning* perangkat bergerak Android. Data *user* juga akan digunakan untuk menambahkan data asisten dosen dan mahasiswa di kelas pada kasus penggunaan “Mengelola Data Peserta Kelas UC-002”.

2.4 JSON

JSON atau *JavaScript Object Notation* adalah *format* pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. *Format* JSON dibuat berdasarkan bagian dari bahasa pemrograman JavaScript [9].

JSON tersusun atas pasangan *key-value* dengan *format* yang umum seperti pada Gambar 2.1. Tipe data dari pasangan ini adalah *string* apapun yang memenuhi kondisi, yaitu harus di dalam petik (“*key*”). Selain *key*, *value* juga memiliki kondisi tipe data, ada 6 tipe data untuk mengisi data *value*, *strings*, *numbers*, *objects*, *booleans*, dan *null*.



```

1 [
2   {
3     "id": 2,
4     "nama": "Administrator Jorginho",
5     "username": "administrator",
6     "short_name": "editingteacher"
7   },
8   {
9     "id": 5,
10    "nama": "Dewangga Okta",
11    "username": "511410005",
12    "short_name": "student"
13  },
14  {
15    "id": 6,
16    "nama": "NURUL WACHIDAH",
17    "username": "5114100052",
18    "short_name": "student"
19  },
20 ]
  
```

Gambar 2.1. JSON Key-Value

2.5 JAVA

Java adalah bahasa pemrograman berbasis objek atau disebut *object oriented programming* (OOP) atau pemrograman yang berorientasi pada objek [10]. Bahasa ini merupakan adopsi dari bahasa pemrograman C dan C++ dimana sintaks pada Java dan kedua bahasa pemrograman ini hampir sama namun Java memiliki keunggulan yaitu dapat dijalankan pada beberapa *platform* sistem operasi yang berbeda [11]. Keunggulan ini yang menjadi salah satu pertimbangan digunakannya Java dalam penyusunan REST API maupun aplikasi Android pada tugas akhir ini, yaitu dapat dijalankan di beberapa sistem operasi yang berbeda.

2.6 ANDROID STUDIO

Android Studio adalah *platform* kerja Android berupa *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android yang berbasis Java [2]. *Platform* ini memiliki fitur yaitu sistem versi berbasis *Gradle* yang fleksibel, *emulator* yang cepat dan teratur, dapat langsung *instant run* sehingga tidak perlu membuat apk baru, serta ada dukungan C++, NDK, serta *Google Cloud Platform* untuk mempermudah pengintegrasian *Firebase Cloud Messaging* (FCM) dan *App Engine*. Pada tugas akhir ini Android Studio dipilih untuk membuat aplikasi *e-Learning* karena fitur dan *library* yang lengkap seperti RxJava, *Fast Android Networking* (FAN).

2.7 REST API

REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang sering diterapkan dalam pengembangan layanan berbasis *web*. Arsitektur REST, yang umumnya dijalankan melalui HTTP (*Hypertext Transfer Protocol*), melibatkan proses pembacaan laman *web* tertentu yang memuat sebuah *file* XML atau JSON. *File* inilah yang menguraikan dan memuat konten yang hendak disajikan. Setelah melalui sebuah proses definisi tertentu, konsumen akan bisa mengakses antarmuka aplikasi yang dimaksudkan.

Kekhasan REST terletak pada interaksi antara klien dan *server* yang difasilitasi oleh sejumlah tipe operasional (verba) dan *Universal Resource Identifiers* (URIs) yang unik bagi tiap-tiap sumberdaya. Masing-masing verba seperti *GET*, *POST*, *PUT* dan *DELETE* memiliki makna operasional khusus untuk menghindari ambiguitas [12]. Kegunaan dari masing-masing verba berbeda-beda, kegunaan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Metode HTTP dan Penggunaannya dalam REST

Metode	Deskripsi
GET	Mendapatkan (<i>read</i>) sebuah sumber daya (<i>resource</i>) yang diidentifikasi dengan URI (<i>Uniform Resource Identifier</i>).
POST	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk membuat (<i>create</i>) sumber daya baru.
PUT	Mengirimkan sumber daya (<i>resource</i>) ke server. Digunakan untuk memasukkan (<i>insert</i>) atau memperbarui (<i>update</i>) sumber daya yang tersimpan.
DELETE	Menghapus (<i>delete</i>) sumber daya (<i>resource</i>) yang diidentifikasi dengan URI.

Komponen dari *HTTP Response* adalah:

- *Status/Response Code*, mengindikasikan *status server* terhadap *resource* yang diminta. Misal: 404, artinya *resource* tidak ditemukan dan 200 *response* OK.
- *HTTP Version*, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
- *Response Header*, berisi metadata untuk *HTTP Response*. Contoh, *type server*, panjang *content*, tipe *content*, waktu *response*, dll.
- *Response Body*, konten dari data yang diberikan.

2.8 ANDROID

Android adalah sistem operasi yang dirancang untuk perangkat bergerak dengan layar sentuh seperti *smartphone* atau

tablet dengan didukung kernel yang berbasis Linux dan ditulis dengan bahasa pemrograman Java [13]. Android menjadi populer sejak dirilis pada tahun 2008 karena merupakan *platform* tak terbatas di mana aplikasi berbasis Android sangat banyak dan dapat diakses secara gratis [14].

2.9 REACTIVE PROGRAMMING

Reactive Programming adalah upaya untuk mempelajari pengetahuan yang mungkin untuk di terapkan di generasi baru dari perangkat lunak. Pemograman reaktif melakukan pendekatan koheren untuk arsitektur sistem yang dibutuhkan dengan membuat suatu sistem yang *Responsive, Resilient, Elastic and Message Driven*.

Sistem reaktif dibangun sebagai *Reactive Systems* sehingga lebih fleksible dan *scalable*. Hal ini memudahkan dalam pengembangan dan yang terpenting lebih tolerir jika ada perubahan di masa depan. Suatu sistem yang reaktif lebih toleran terhadap kegagalan sistem serta memungkinkan untuk user mendapat *feedback* secara *interactive* [15].

2.10 RXJAVA/RXANDROID

RxAndroid adalah *library* yang digunakan untuk implementasi dari *Reactive Programming*. ReactiveX ini meng-*extend Observer Pattern*. Dalam ReactiveX sumber data diletakkan dalam *class Observable* dan di-*publish* ke dalam *class Subscriber*. Pada suatu kondisi bisa saja data yang diterima oleh *Subscriber* berbeda dengan data dari *Observable*. *Subscriber* tidak peduli dengan pengolahan data yang dilakukan di *Observable*, Subscriber hanya menerima kembalian data sesuai yang telah ditetapkan. Datanya berupa *integer, string* dan lainnya [5].

2.11 MYSQL

Sebagai suatu pengelola basis data terbesar dan paling banyak digunakan tentunya MySQL ini memiliki fitur atau kapabilitas tertentu. Salah satu yang paling dicari oleh para pengguna MySQL

adalah kemampuannya yang *multi platform* dan berlisensi GPL, sehingga dapat digunakan oleh komputer hampir di semua OS. Kinerjanya juga dianggap cukup tinggi dalam hal memproses kueri yang ada meskipun masih terbatas pada basis data dalam jumlah tertentu.

Beberapa fitur lain yang ada pada MySQL saat ini tersedianya tipe data yang sangat beragam seperti *Float*, *Double*, *Char*, *Date* dan lain-lain. MySQL juga mendukung penggunaan *field* sebagai index serta memiliki tingkat keamanan yang cukup bagus dengan adanya *subnetmask*, nama *host* serta sandi yang terenkripsi [16]. MySQL juga mendukung pengelolaan *database* Moodle.

2.12 MOODLE

Moodle merupakan salah satu aplikasi dari konsep dan mekanisme belajar mengajar yang memanfaatkan teknologi informasi, yang dikenal dengan konsep pembelajaran elektronik atau *e-learning* [4].

Dengan tampilan seperti halaman *web* pada umumnya, Moodle memiliki fitur untuk menyajikan kursus (*course*), dimana pengajar bisa mengunggah materi ajar, soal dan tugas. Murid bisa mengakses *web* Moodle kemudian memilih kursus yang disediakan atau di-*enroll* untuknya. Semua aktivitas murid di Moodle dapat terpantau progres dan nilainya [17]. Di ITS sendiri juga sudah menggunakan Moodle seperti pada *share.its.ac.id* yang dapat dimodifikasi untuk diimplementasikan pada perangkat bergerak.

2.13 MVP

MVP memiliki kepanjangan yaitu *Model View Presenter*. MVP merupakan jenis arsitektur yang membagi-bagi kode sumber dari yang kompleks, menjadi bagian kecil dan sederhana. Arsitektur ini memiliki pola pembagian tugas berdasarkan tingkatan *layer* sehingga mudah untuk di-*test*. Konsep ini membuat *logic code* atau *business code* dari *view* ke *presenter* menjadi terpisah. Sehingga *presenter* memiliki peranan sebagai penyedia

data dari *model* selain dapat menyimpan *logic* yang biasanya seharusnya disimpan di *view* serta menerima *action* dari *view*. Sedangkan model tersusun atas sekumpulan data struktur menjadi satu *class*. Tugasnya yaitu menyimpan data dari API *response* atau menyimpan data untuk *request* API [18].

2.14 FAST ANDROID NETWORKING (FAN)

Fast Android Networking merupakan *library* Android yang dibangun dengan protokol OkHttp yang *support* untuk metode *request* HTTP/2 seperti *POST*, *GET*, *DELETE*, *PUT*. *Fast Android Networking* (FAN) dipilih pada implementasi fitur *networking* karena FAN *support* dengan RxJava dan arsitektur *Model View Presenter* (MVP) [19].

2.15 JAX-RS (Java API for RESTful Web Services)

JAX-RS merupakan arsitektur *web service* yang berbasis protokol HTTP, yang berguna untuk proses transfer data secara independen sehingga masing-masing *request* tidak terkait dengan *request* sebelum atau sesudahnya. Implementasi pada JAX-RS ini menggunakan Apache CXF dengan beberapa metode yang dapat diakses seperti *GET*, *POST*, *PUT*, *DELETE*, *Produces*, *Consumes*, *PathParam*, *QueryParam*, *FormParam*, dan lain-lain [20].

2.16 WSO2 API Manager

WSO2 merupakan sebuah aplikasi yang menyediakan fasilitas komunikasi antara penyedia API dan pengguna API. Aplikasi ini memiliki berbagai fitur untuk menunjang proses komunikasi diantara pengguna dan penyedia API. Dari sisi pengguna, aplikasi ini dapat memilih API yang telah tersedia dari aplikasi *store*, mengakses *url* tertentu yang telah disediakan oleh *API Manager*. Sedangkan dari sisi penyedia, berbagai fitur dapat dilakukan pada aplikasi WSO2 ini seperti, mem-*publish* API, memantau API yang telah diunggah ke *server*, mendaftarkan *endpoint* yang dapat diakses oleh pengguna serta mengaktifkan fitur *token* guna kebutuhan keamanan (*non-functionality security*) [21].

2.17 Dagger 2

Dagger2 adalah *library* yang menerapkan mekanisme *dependency injection* tanpa menulis kode berulang-ulang [22]. Ada beberapa anotasi yang menyusun Dagger2 seperti pada Tabel 2.2.

Tabel 2.2. Anotasi pada Dagger2

Anotasi	Pengertian
@Module	kelas yang menyediakan <i>dependencies</i>
@Provides	<i>method</i> yang ada di kelas @Module
@Scope	anotasi Java untuk menginisialisasikan <i>object</i> yang akan diinjeksikan
@Inject	sebuah <i>request dependency</i> , bisa berupa konstruktor, <i>method</i> , maupun <i>field</i>
@Component	sebuah jembatan penghubung antara <i>module</i> dan <i>injection</i> .

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas analisis kebutuhan dan rancangan yang akan digunakan untuk membangun perangkat lunak yang diajukan sebagai tugas akhir

3.1 Analisis

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

3.1.1 Analisis Permasalahan

Aplikasi *e-Learning* merupakan alternatif proses pembelajaran di kelas. Hal ini menjadi sarana untuk dosen dan mahasiswa berinteraksi lintas waktu. Aplikasi ini dapat menjadi solusi bila dosen berhalangan mengajar karena tugas dinas atau kegiatan lain.

Di Institut Teknologi Sepuluh Nopember (ITS) Surabaya, sistem belajar mengajar dilakukan dengan tatap muka di kelas dan sistem *e-Learning* yang masih berbasis *website* yaitu *share.its.ac.id*. Kegiatan belajar mengajar dengan sistem *e-Learning* yang masih hanya berbasis *website* sering membuat mahasiswa malas untuk menggunakannya karena harus membuka komputer/laptop terlebih dahulu untuk mengaksesnya. Tampilan *website* sistem *e-Learning* yang ada juga dirasa kurang *responsive* apabila digunakan di perangkat bergerak Android.

Mudahnya pengaplikasian dan akses secara gratis menjadi nilai tambah bagi sistem operasi Android ini. Pembangunan *platform mobile e-Learning* ini didukung dengan Android Studio sebagai *tools* pembuatan aplikasi yang *user friendly* yang menyediakan berbagai fitur penunjang aplikasi dan dokumentasi yang lengkap.

Akses dan pengambilan informasi juga semakin mudah dan cepat diakses melalui penggunaan perangkat bergerak maupun *web*. Pengambilan informasi didukung dengan *Application Programming Interface* (API) dengan menerapkan arsitektur *Representational State Transfer of Application Programming*

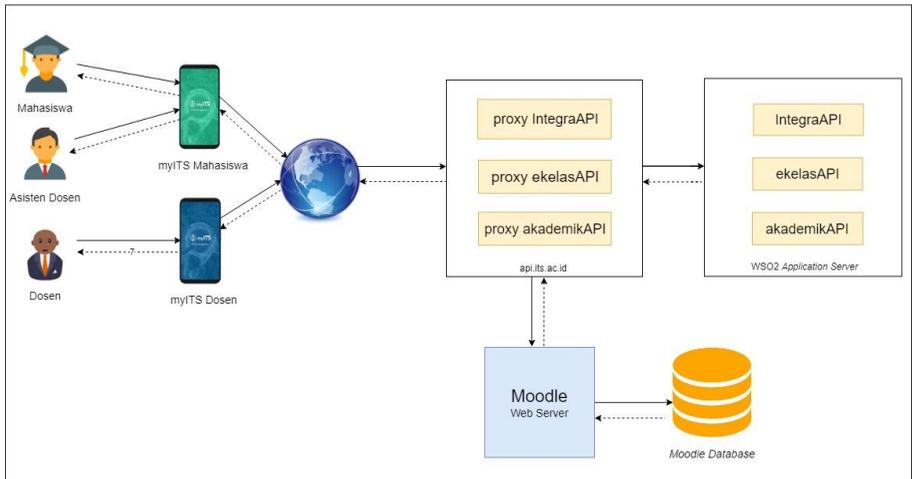
Interface (REST-API). Kegunaan dari pengaplikasian arsitektur ini adalah proses pengambilan informasi dan data dari *database* SIM AKADEMIK dan Integra ITS. Selain itu untuk diterapkan juga pola pembuatan *web services* untuk mengakses ke *web e-Learning* menggunakan *Java Programming Language for RESTful Web Services*(JAX-RS). Pola ini diterapkan supaya dapat menyajikan konten dari *web e-Learning* dengan beban komunikasi yang ringan.

Perbandingan antara sistem *e-Learning* berbasis *website* yang ada saat ini dengan sistem *e-Learning* berbasis perangkat bergerak Android yang akan diimplementasikan dapat dilihat pada Tabel 3.1.

Tabel 3.1. Perbandingan Sistem

Sistem <i>e-Learning</i> berbasis <i>website</i>	Sistem <i>e-Learning</i> berbasis perangkat bergerak Android
Tampilan kurang dapat digunakan pada ukuran layar kecil seperti perangkat bergerak Android.	Tampilan dapat menyesuaikan ukuran layar terutama digunakan pada perangkat bergerak Android.
Dosen tidak dapat mendaftarkan asisten dosen dan mahasiswa yang baru bergabung ke kelasnya secara langsung, melainkan melalui <i>admin</i> .	Terdapat fitur untuk dosen dapat mendaftarkan asisten dosen dan mahasiswa yang baru bergabung ke kelasnya.
Data peserta kelas secara manual diisikan oleh <i>admin</i> .	Data <i>user</i> terintegrasi secara otomatis dengan SIAKAD ITS.

3.1.2 Deskripsi Umum Sistem



Gambar 3.1 Deskripsi Umum Sistem

Gambar 3.1 merupakan arsitektur sistem perangkat lunak yang akan dibangun. Pada gambar ditunjukkan fokus pengerjaan pada pada perangkat lunak yang akan dibangun. Sistem yang dibangun dari beberapa *layer*. Pada gambar ini ditunjukkan *layer* *api.its.ac.id* semacam *layer security* yang akan membatasi akses langsung ke *database*. Data yang di-request akan diproses pada *layer* *WSO2 application server* dengan mengakses ke *eKelasAPI*.

Implementasi *Java Programming Language for Restful Web Services (JAX-RS)* ini dilakukan di *layer* *WSO2 application server* yang memuat *ekelasAPI*. *Application Programming Interface (API)* ini disusun pada aplikasi Eclipse dengan menerapkan arsitektur *Representational State Transfer of Application Programming Interface (REST API)*.

Web e-Learning ini disusun dan mengambil *source* dari sebuah sistem pembelajaran *online* Moodle. Sistem ini merupakan *Learning Management System (LMS)*, yaitu sebuah Sistem yang berbasis pada konsep *social constructionist pedagogy* yang menekankan model pengajaran berorientasi objek (mahasiswa). Moodle disajikan dalam bentuk *open source* dan dapat diterapkan pada sistem operasi apapun dengan bahasa pemrograman PHP dan

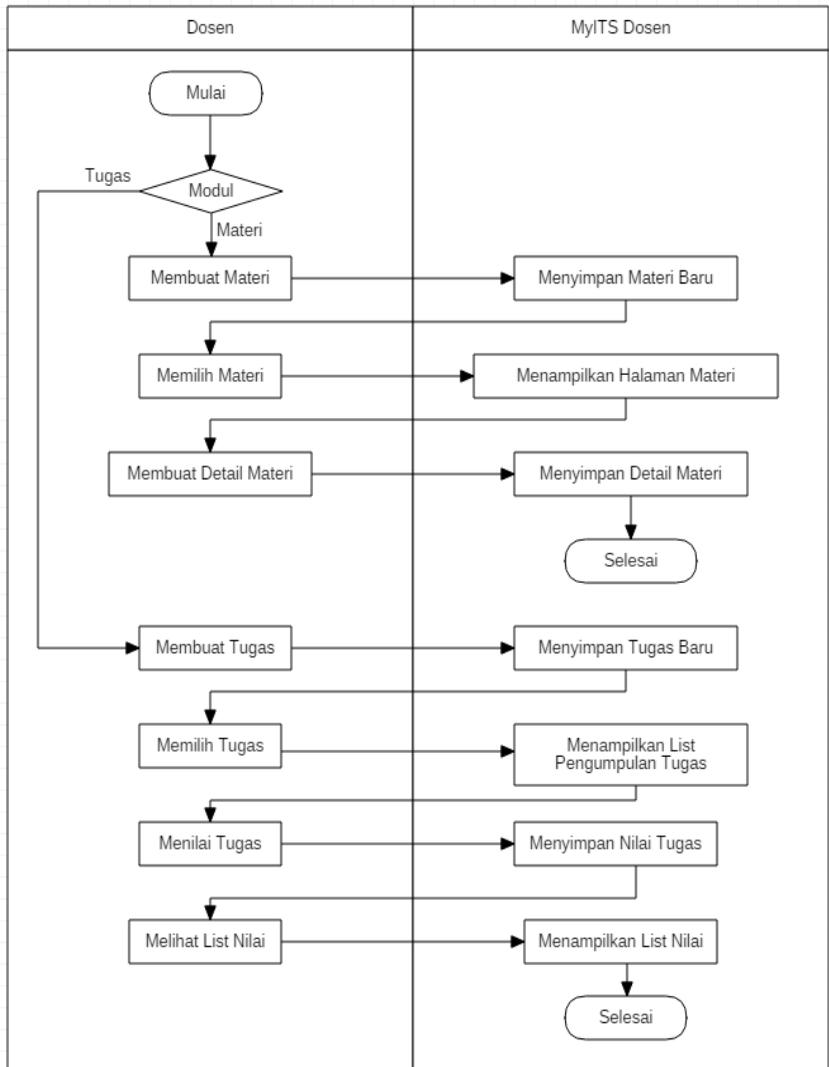
memiliki *database* SQL. Alasan digunakan Moodle sebagai sistem dalam *web e-Learning*, karena di ITS sudah menggunakan sistem *database* Moodle sehingga dapat menyinkronkan *database* antara SIM AKADEMIK dengan *database e-Learning*. Moodle juga dilengkapi dengan versi perangkat bergerak. Namun aplikasi ini memiliki beberapa kelemahan seperti perlu memasukkan *internet protocol address* secara manual setiap kali melakukan *login*, tampilan terlalu kaku, selain itu hanya terkoneksi ke *server* Moodle saja. Sehingga tidak dapat diterapkan pada tugas akhir ini, karena pada tugas akhir ini perlu mengakses ke *database* lain seperti SIM AKADEMIK, Integra ITS.

Aplikasi ini dikembangkan dengan menggunakan paradigma *Reactive Programming*, yaitu pola pemrograman yang berbeda dengan pola pemrograman secara prosedural. Bila melakukan suatu pekerjaan, pada pemrograman prosedural, *user* harus melakukan *request* terus menerus sehingga menyebabkan *blocking thread*. Sedangkan pada *Reactive Programming*, *user* tidak perlu meminta *request* secara aktif atau terus menerus, karena *thread* akan mendelegasikan *task* yang diminta oleh *user* ke *thread* lainnya. *User* dapat melakukan *task* lain sambil menunggu notifikasi dari *task* yang sedang diproses oleh *thread* (*asynchronous*). Paradigma *Reactive Programming* diimplementasikan pada *framework Model View Presenter* (MVP).

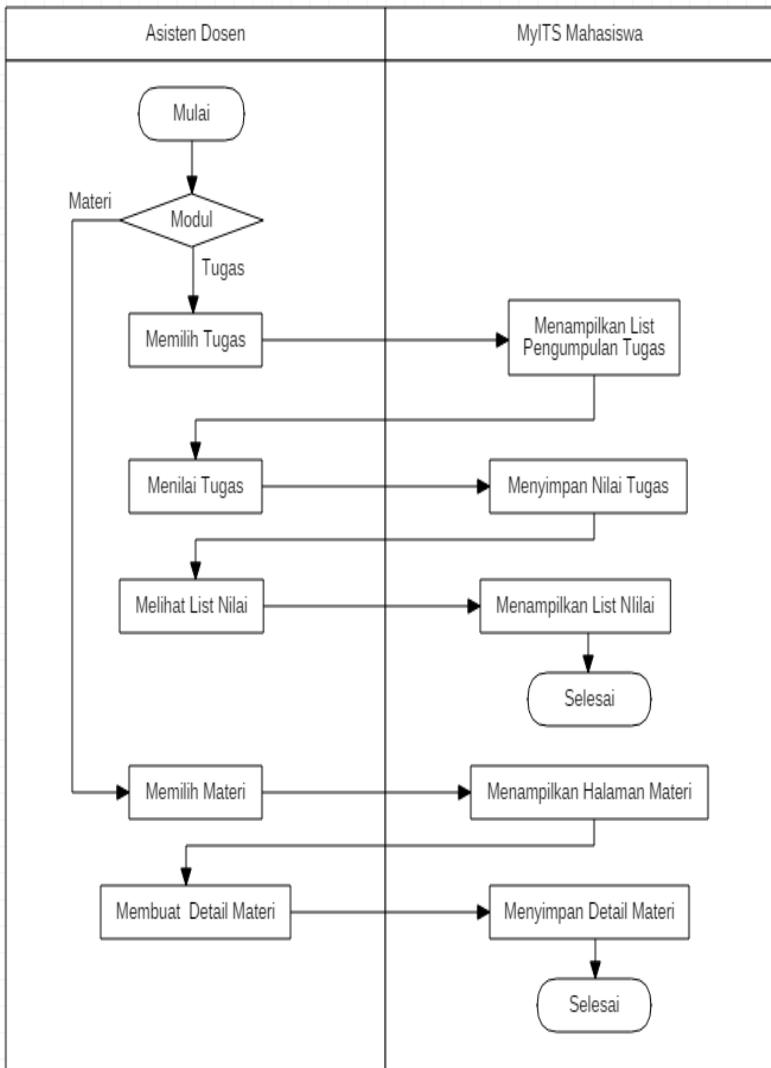
Tahapan aplikasi ini dibagi menjadi dua jenis aplikasi yaitu MyITS-Dosen untuk fitur dosen dan MyITS-Mahasiswa untuk fitur asisten dosen dan mahasiswa. Perbedaan aplikasi dilakukan karena adanya perbedaan fitur dan pembatasan hak akses. Pada fitur dosen memiliki beberapa menu yang dapat diakses, seperti melihat data peserta di kelas, mengelola tugas, mengelola materi perkuliahan serta mengolah nilai. Pada fitur asisten dosen ada beberapa menu yang dapat diakses seperti mengelola tugas, mengelola materi perkuliahan serta mengolah nilai. Sedangkan pada fitur mahasiswa hanya dapat melihat tugas masing-masing, tugas yang harus dikerjakan, materi yang telah di-*posting* oleh dosen, mengunduh materi yang telah diberikan oleh dosen.

3.1.3 Alur Kerja Sistem

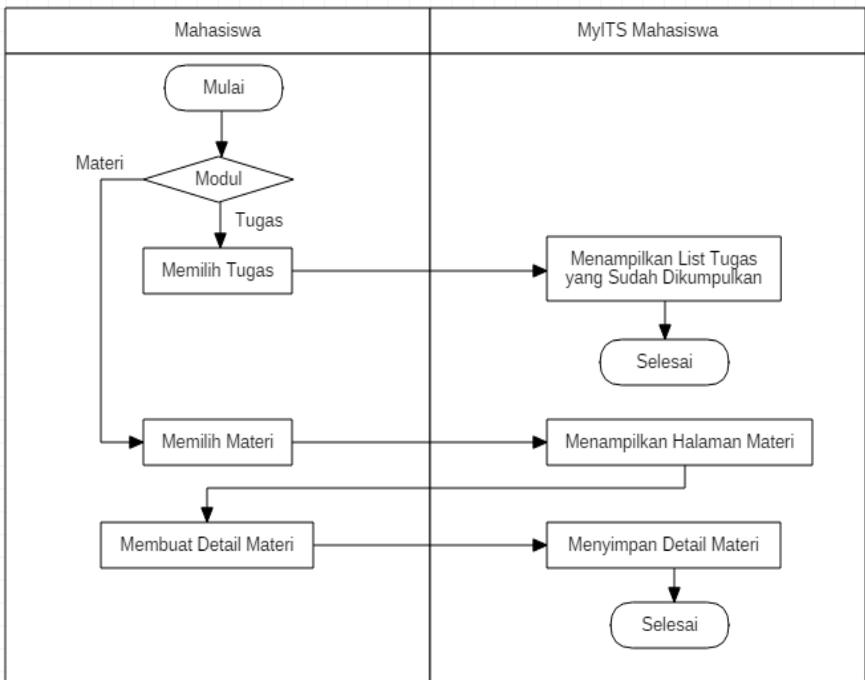
Alur kerja sistem akan menjelaskan gambaran kerja sistem secara besar. Alur kerja sistem ini dibagi menjadi tiga jenis, yaitu fitur manajemen perkuliahan pada aplikasi MyITS-Dosen untuk dosen, dan fitur manajemen perkuliahan pada aplikasi MyITS-Mahasiswa untuk asisten dosen dan mahasiswa. Perbedaan aplikasi dilakukan karena adanya perbedaan fitur dan pembatasan hak akses. Pada aplikasi MyITS-Dosen, dosen memiliki beberapa menu yang dapat diakses, yaitu pengelolaan tugas, pengelolaan nilai dan pengelolaan materi perkuliahan. Sedangkan pada aplikasi MyITS-Mahasiswa, mahasiswa hanya dapat menggunakan menu melihat materi perkuliahan, membuat detail materi perkuliahan dan melihat tugas yang telah dikumpulkan. Asisten dosen dapat melakukan pengelolaan tugas, pengelolaan nilai, dan pengelolaan detail materi perkuliahan. Alur kerja sistem dapat dilihat pada Gambar 3.2, Gambar 3.3 dan Gambar 3.4.



Gambar 3.2. Alur Kerja Sistem Dosen



Gambar 3.3. Alur Kerja Sistem Asisten Dosen



Gambar 3.4. Alur Kerja Sistem Mahasiswa

3.1.4 Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan cakupan perangkat lunak yang telah dijelaskan pada bagian deskripsi umum sistem, dibutuhkan adanya spesifikasi perangkat lunak supaya dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Diharapkan dengan adanya spesifikasi ini dapat menyesuaikan kebutuhan-kebutuhan pengguna. Spesifikasi kebutuhan perangkat lunak adalah penjelasan mengenai kebutuhan sistem yang diinginkan pengguna atau *client* dalam bentuk tulisan. Spesifikasi kebutuhan perangkat lunak pada tugas akhir ini terdiri dari kebutuhan fungsional, kebutuhan non-fungsional, aktor, dan kasus penggunaan.

3.1.4.1 Kebutuhan Fungsional

Kebutuhan fungsional dihasilkan dari analisis yang telah dijelaskan sebelumnya. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.2.

Tabel 3.2. Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Melihat Data Kelas	Menampilkan <i>list</i> kelas, serta menampilkan halaman utama kelas.
2	Mengelola Data Peserta Kelas	Menampilkan <i>list</i> peserta kelas, menampilkan jumlah mahasiswa yang mengikuti kelas, mendaftarkan peserta ke kelas, serta menghapus peserta dari kelas.
3	Mengelola Tugas	Mem- <i>posting</i> tugas, menghapus tugas, meng- <i>edit</i> tugas, menampilkan <i>detail</i> tugas.
4	Mengelola Materi Perkuliahan	Mem- <i>posting</i> materi perkuliahan, menghapus materi perkuliahan, meng- <i>edit</i> materi perkuliahan, menampilkan materi perkuliahan.
5	Melihat Tugas	Menampilkan <i>list</i> tugas yang sudah dikumpulkan, menampilkan <i>detail</i> tugas.
6	Melihat Materi Perkuliahan	Menampilkan materi perkuliahan.
7	Mengelola Detail Materi Perkuliahan	Menampilkan detail materi perkuliahan, mem- <i>posting</i> materi perkuliahan, meng- <i>edit</i> materi perkuliahan, menghapus materi perkuliahan, mengunduh materi perkuliahan.
8	Mengelola Nilai	Menampilkan <i>list</i> pengumpulan tiap tugas, menampilkan <i>list</i> nilai tiap tugas, menilai tugas yang sudah dikumpulkan oleh mahasiswa.

3.1.4.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem adalah sebagai berikut.

1. Kebutuhan Operasional

Perangkat lunak akan berjalan pada berbagai jenis perangkat bergerak dengan sistem operasi yang berbeda, ukuran layar yang berbeda serta resolusi yang berbeda.

3.1.4.3 Aktor

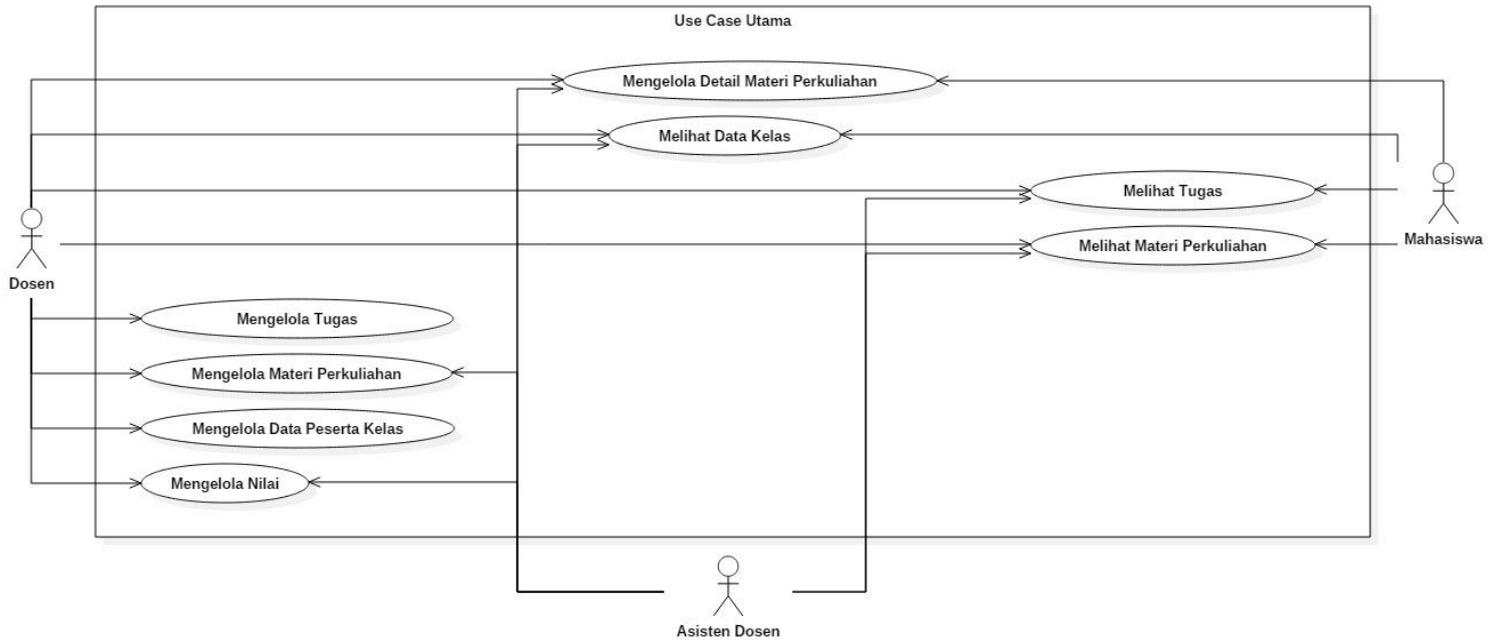
Pengertian aktor atau biasa disebut entitas luar adalah siapa saja atau pihak-pihak apapun yang harus berinteraksi dengan sistem. Aktor dapat berupa manusia, sistem lain, atau perangkat lain. Pada perangkat lunak ini terdapat tiga pengguna yaitu dosen, asisten dosen dan mahasiswa. Dosen adalah sebutan bagi orang yang mendidik dan mengajar aktor mahasiswa pada perguruan tinggi, sekolah tinggi, universitas atau institusi, dalam instansi swasta ataupun negeri. Asisten Dosen adalah mahasiswa yang membantu proses penyampaian materi oleh dosen di kelas, supaya mahasiswa dapat lebih mudah memahami materi yang diajarkan dan menyelesaikan tugas atau ujian yang diberikan. Mahasiswa adalah sebutan bagi orang yang sedang menempuh pendidikan tinggi di sebuah perguruan tinggi yang terdiri atas sekolah tinggi, akademi, universitas atau institusi.

3.1.4.4 Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan digambarkan dalam tabel penjelasan kasus penggunaan dan diagram kasus penggunaan. Tabel penjelasan kasus penggunaan dapat dilihat pada Tabel 3.3 dan diagram kasus penggunaan dapat dilihat pada Gambar 3.5.

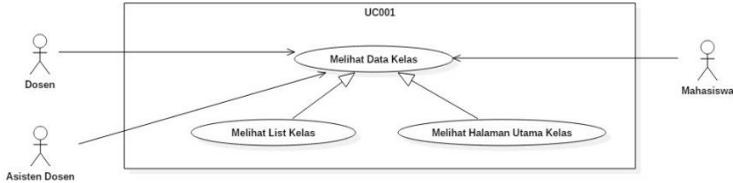
Tabel 3.3. Kasus Penggunaan

Kode Kasus Penggunaan	Nama	Aktor
UC-001	Melihat Data Kelas	Dosen, Asisten, Mahasiswa
UC-002	Mengelola Data Peserta Kelas	Dosen
UC-003	Mengelola Tugas	Dosen
UC-004	Mengelola Materi Perkuliahan	Dosen
UC-005	Melihat Tugas	Mahasiswa
UC-006	Melihat Materi Perkuliahan	Mahasiswa, Asisten
UC-007	Mengelola Detail Materi Perkuliahan	Dosen, Asisten, Mahasiswa
UC-008	Mengelola Nilai	Dosen, Asisten



Gambar 3.5 Diagram Kasus Penggunaan

3.1.4.4.1 Melihat Data Kelas (UC-001)

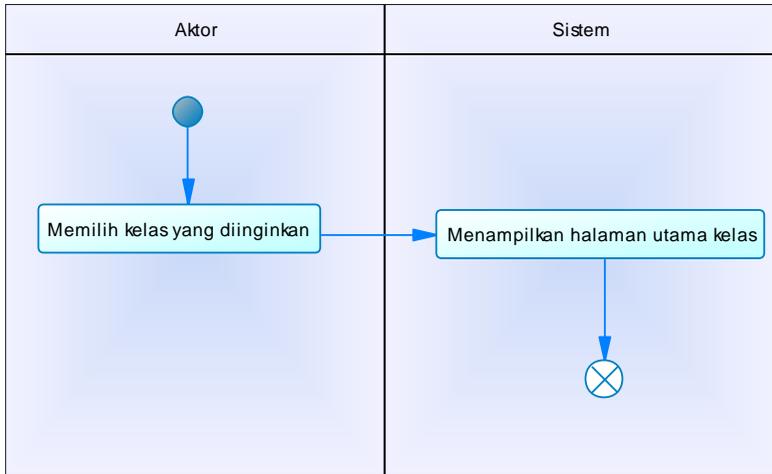


Gambar 3.6. Diagram Kasus Penggunaan UC-001

Kasus penggunaan nomor UC-001 ini diakses ketika dosen, asisten dosen, serta mahasiswa masuk ke dalam aplikasi dan memilih kelas yang diinginkan. Pada tahap selanjutnya setelah memilih kelas, aktor akan masuk ke halaman utama masing-masing kelas. Spesifikasi dan diagram aktivitas dari kasus penggunaan ini dapat dilihat pada Tabel 3.4 dan Gambar 3.7.

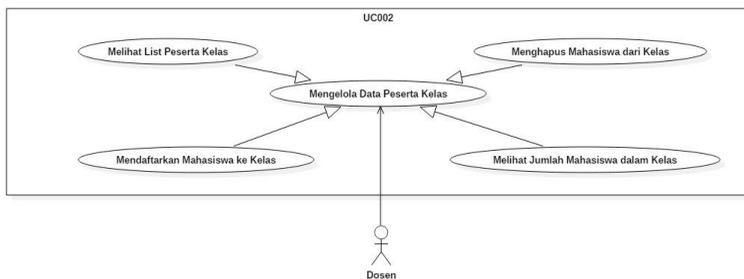
Tabel 3.4. Spesifikasi Kasus Penggunaan UC-001

Kode Use Case	UC-001	
Nama Use Case	Melihat Data Kelas	
Aktor	Dosen, Asisten Dosen, Mahasiswa	
Deskripsi	Aktor dapat memilih pilihan kelas serta mengakses halaman utama masing-masing kelas, sistem menampilkan <i>list</i> kelas dan halaman utama kelas.	
Relasi	-	
Kondisi Awal	Sistem menampilkan <i>list</i> kelas yang berhak diakses oleh aktor.	
Kondisi Akhir	Aktor berada pada halaman utama kelas di menu E-Kelas.	
Alur Kejadian Normal	Aktor	Sistem
	1. Memilih kelas yang diinginkan	
		2. Menampilkan halaman utama kelas



Gambar 3.7. Diagram Aktivitas UC-001

3.1.4.4.2 Mengelola Data Peserta Kelas UC-002



Gambar 3.8. Diagram Kasus Penggunaan (UC-002)

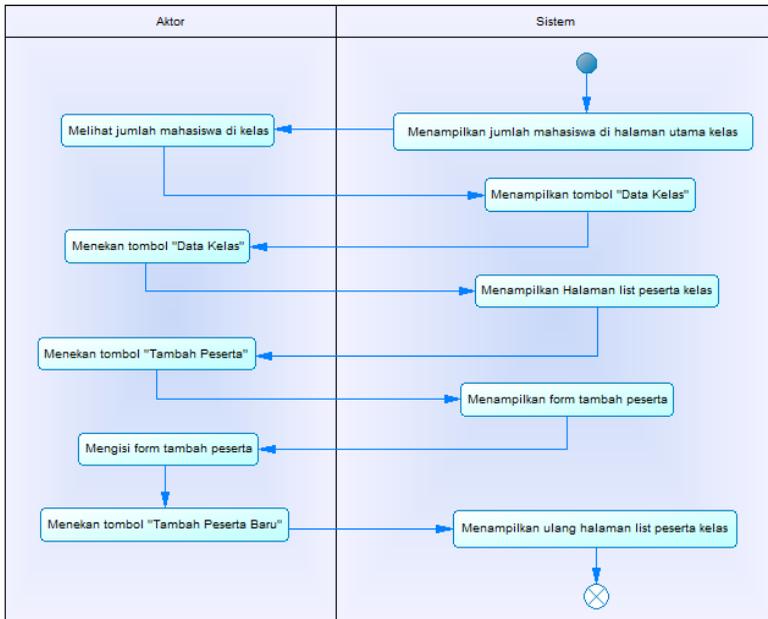
Kasus penggunaan nomor UC-002 ini diakses ketika dosen hendak melakukan pengolahan data peserta di masing-masing kelas. Spesifikasi dan diagram aktivitas dapat dilihat pada

Tabel 3.5, Gambar 3.9 dan Gambar 3.10.

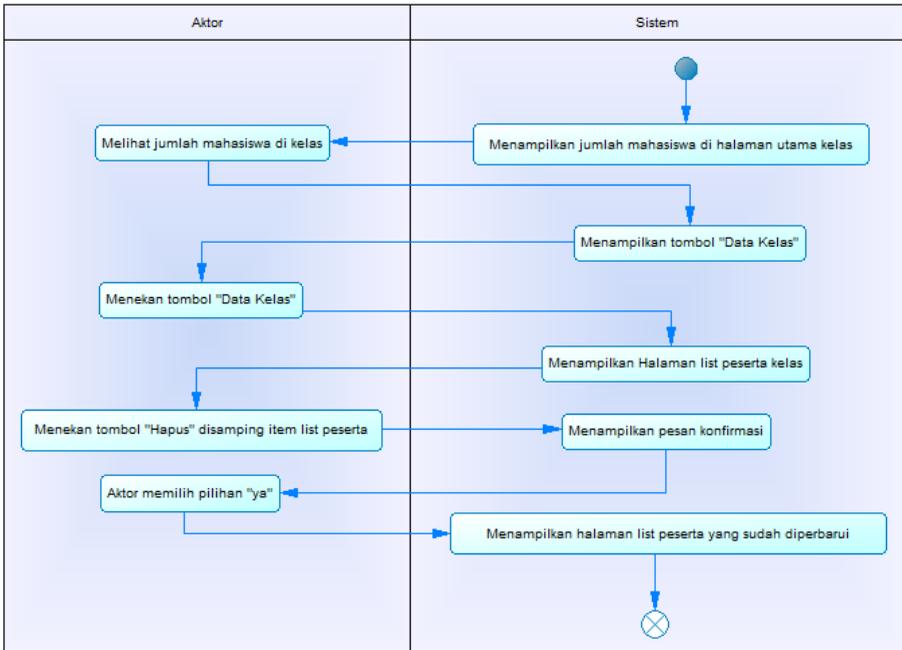
Tabel 3.5. Spesifikasi Kasus Penggunaan UC-002

Kode Use Case	UC-002	
Nama Use Case	Mengelola Data Peserta Kelas	
Aktor	Dosen	
Deskripsi	Dosen dapat melihat <i>list</i> peserta kelas, melihat jumlah mahasiswa di kelas, mendaftarkan mahasiswa, serta menghapus mahasiswa dari kelas.	
Relasi	-	
Kondisi Awal	Dosen berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan <i>list</i> peserta kelas	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan jumlah mahasiswa di kelas
	2. Melihat jumlah mahasiswa di kelas	
		3. Menampilkan tombol “Data Kelas”
	4. Menekan tombol “Data Kelas”	
		5. Menampilkan halaman <i>list</i> peserta kelas
Alur Kejadian Alternatif	Aktor	Sistem
	5a. Aktor Menambah Peserta ke kelas	
		5a.1. Menampilkan halaman <i>list</i> peserta kelas
	5a.2. Menekan tombol “Tambah Peserta”	
		5a.3. Menampilkan <i>form</i> tambah peserta
5a.4. Mengisi <i>form</i> tambah peserta		

	5a.5. Menekan tombol “Tambah Peserta Baru”	
		5a.6. Menampilkan ulang halaman <i>list</i> peserta kelas
	5b. Aktor Menghapus peserta dari kelas	
		5b.1. Menampilkan halaman <i>list</i> peserta kelas
	5b.2. Menekan tombol “Hapus” di samping <i>item list</i> peserta	
		5b.3. Menampilkan pesan konfirmasi
	5b.4. Aktor memilih pilihan “ya”	
		5b.5. Menampilkan ulang halaman <i>list</i> peserta kelas

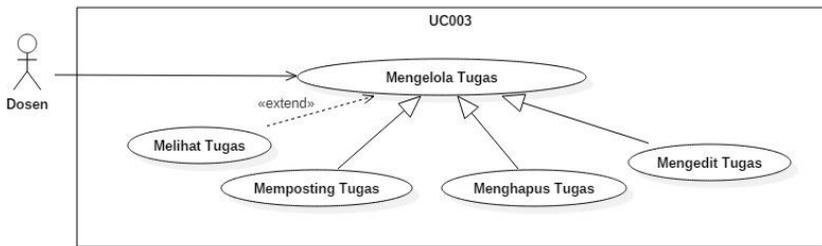


Gambar 3.9. Diagram Aktivitas UC-002 Tambah User



Gambar 3.10. Diagram Aktivitas UC-002 Hapus User

3.1.4.4.3 Mengelola Tugas (UC-003)



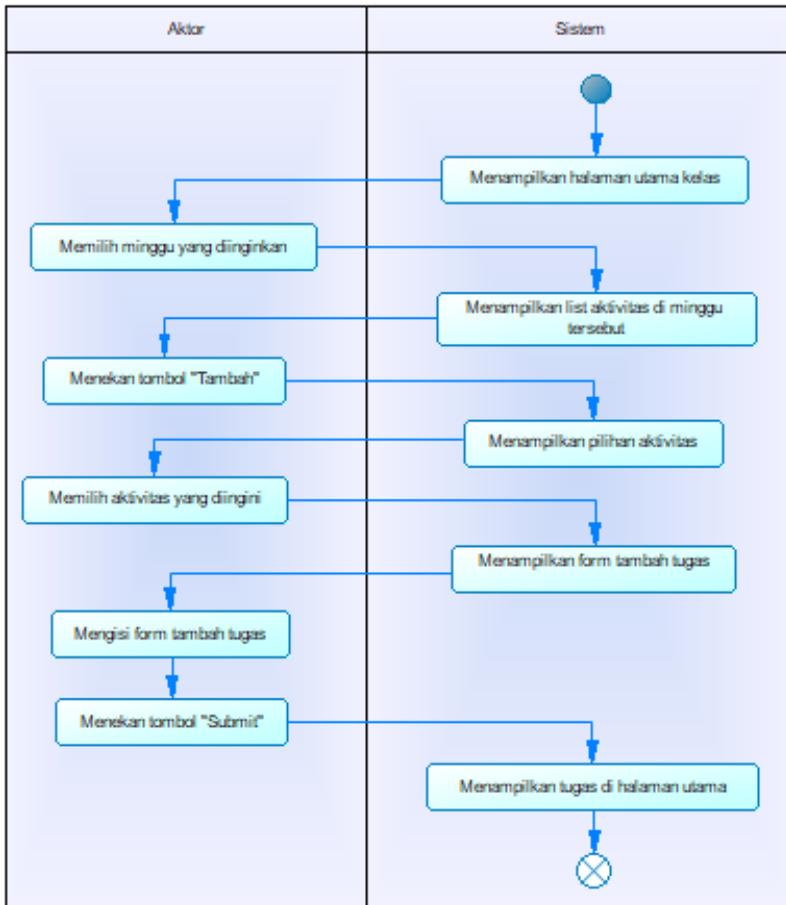
Gambar 3.11. Diagram Kasus Penggunaan UC-003

Kasus penggunaan nomor UC-003 ini diakses ketika dosen hendak melakukan pengolahan data tugas di masing-masing minggu dan masing-masing tugas. Untuk menjelaskan peran aktor maka ada spesifikasi dan diagram aktivitas dapat dilihat pada Tabel 3.6, Gambar 3.12, Gambar 3.13 dan Gambar 3.14.

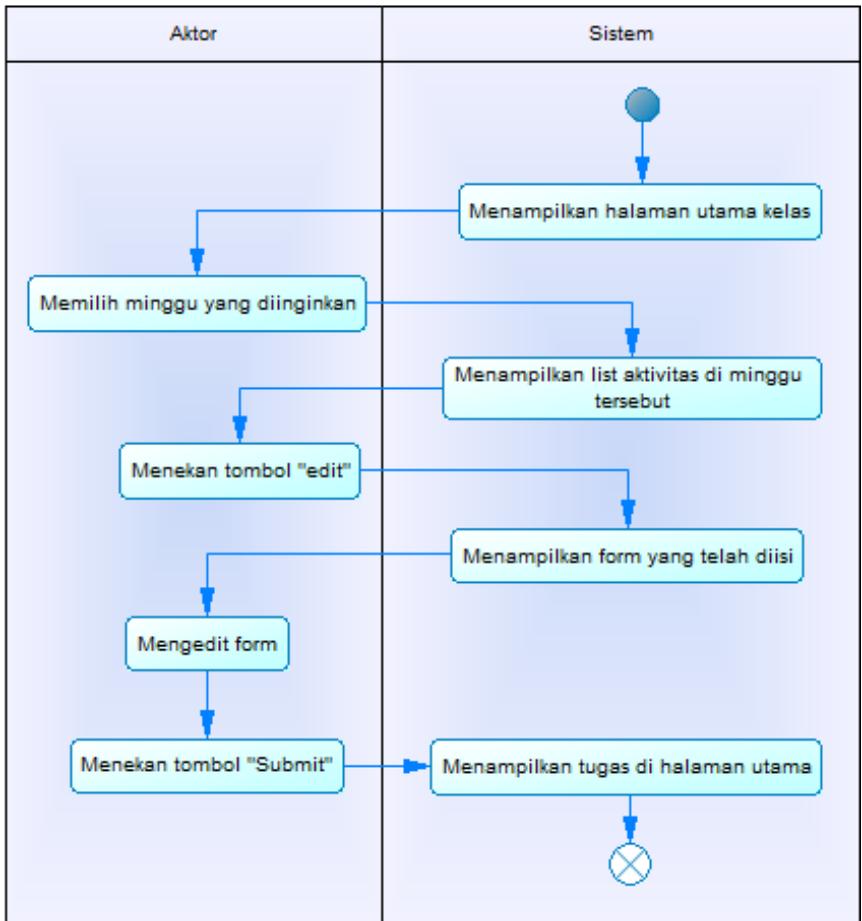
Tabel 3.6. Spesifikasi Kasus Penggunaan UC-003

Kode Use Case	UC-003	
Nama Use Case	Mengelola Tugas	
Aktor	Dosen	
Deskripsi	Dosen dapat memposting tugas, mengedit tugas, serta menghapus tugas.	
Relasi	-	
Kondisi Awal	Dosen berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan halaman tugas	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan halaman utama kelas
	2. Memilih minggu yang diinginkan	
		3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	4. Menekan tombol “Tambah”	
		5. Menampilkan pilihan aktivitas

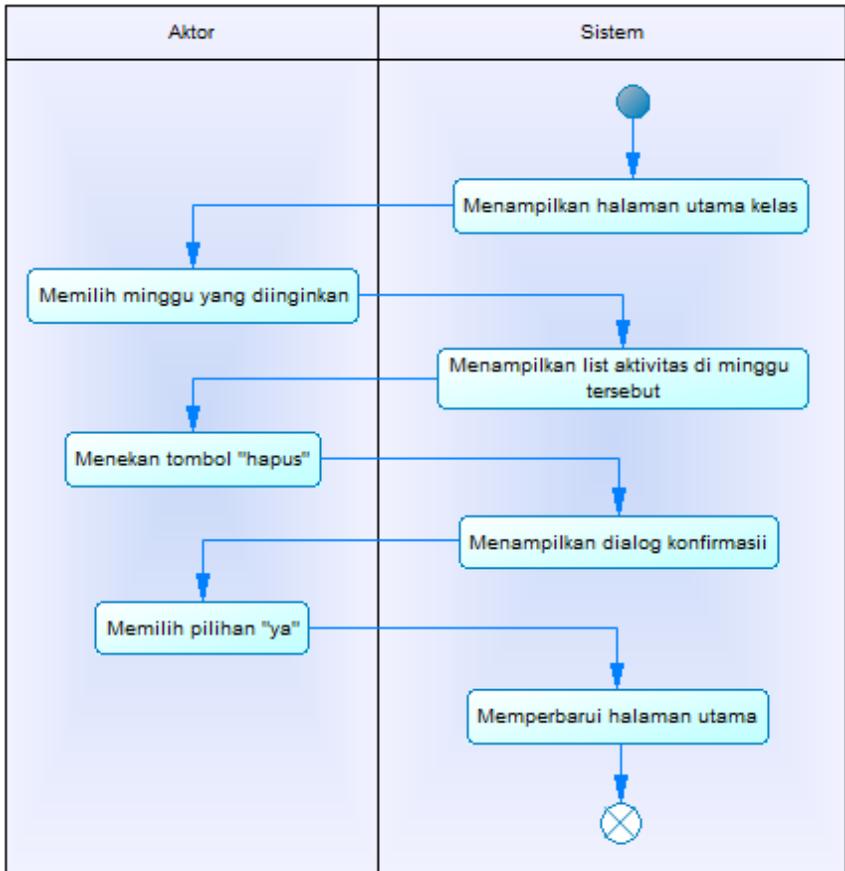
	6. Memilih aktivitas yang diinginkan	
		7. Menampilkan <i>form</i> tambah tugas
	8. Mengisi <i>form</i> tambah tugas	
	9. Menekan tombol <i>submit</i>	
		10. Menampilkan tugas di halaman utama
Alur Kejadian Alternatif	10a. Mengedit tugas.	
		10a.1. Menampilkan tugas di halaman utama
	10a.2. Menekan tombol “ <i>Edit</i> ” di samping <i>item</i> aktivitas	
		10a.3. Menampilkan <i>form edit</i> tugas
	10a.4. Mengisi <i>form edit</i> tugas	
	10a.5. Menekan tombol “ <i>Submit</i> ”	
		10a.6. Menampilkan ulang tugas di halaman utama
	10b. Aktor Menghapus tugas	
		10b.1. Menampilkan tugas di halaman utama
	10b.2. Menekan tombol “ <i>Hapus</i> ” di samping <i>item</i> aktivitas	
		10b.3. Menampilkan pesan konfirmasi
	10b.4. Aktor memilih pilihan “ <i>ya</i> ”	
		10b.5. Menampilkan halaman utama



Gambar 3.12. Diagram Aktivitas UC-003 Menambah Tugas

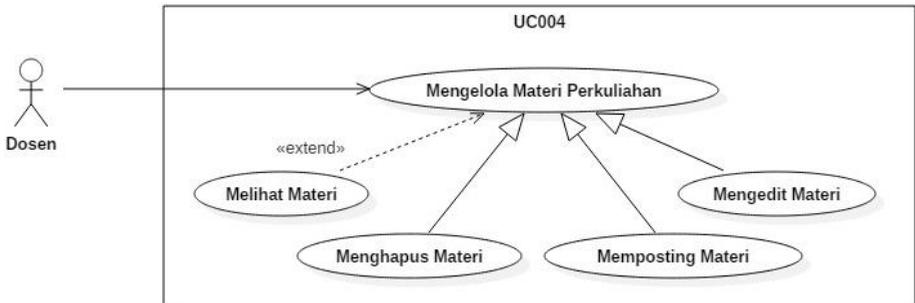


Gambar 3.13. Diagram Aktivitas UC-003 Mengedit Tugas



Gambar 3.14. Diagram Aktivitas UC-003 Menghapus Tugas

3.1.4.4.4 Mengelola Materi Perkuliahan (UC-004)



Gambar 3.15. Diagram Kasus Penggunaan UC-004

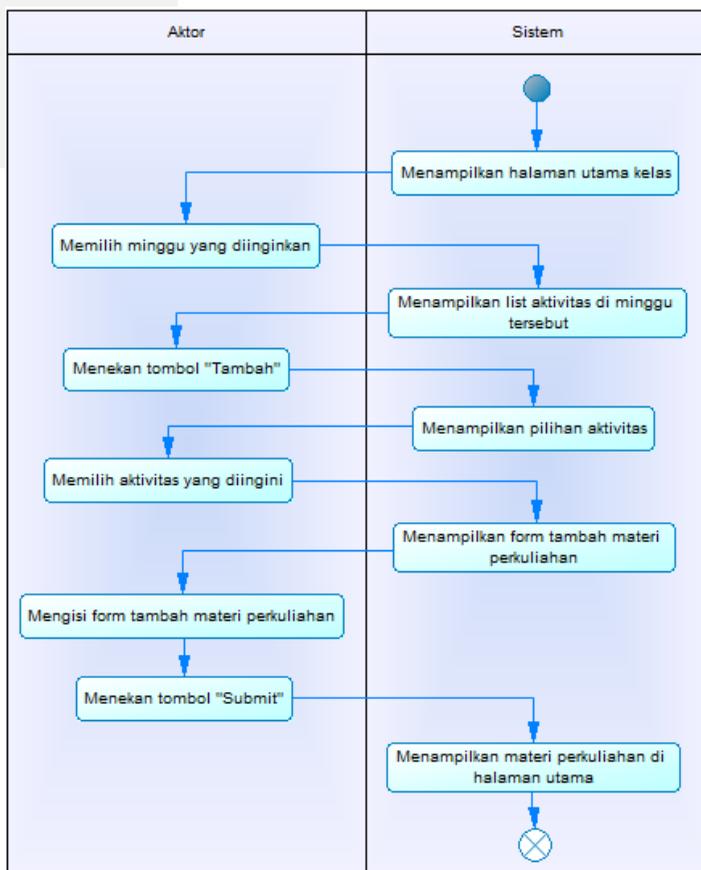
Kasus penggunaan nomor UC-004 ini diakses ketika dosen hendak melakukan pengolahan data materi perkuliahan di masing-masing kelas. Spesifikasi dan diagram aktivitas dapat dilihat pada Tabel 3.7, Gambar 3.16, Gambar 3.17 dan Gambar 3.18.

Tabel 3.7. Spesifikasi Kasus Penggunaan UC-004

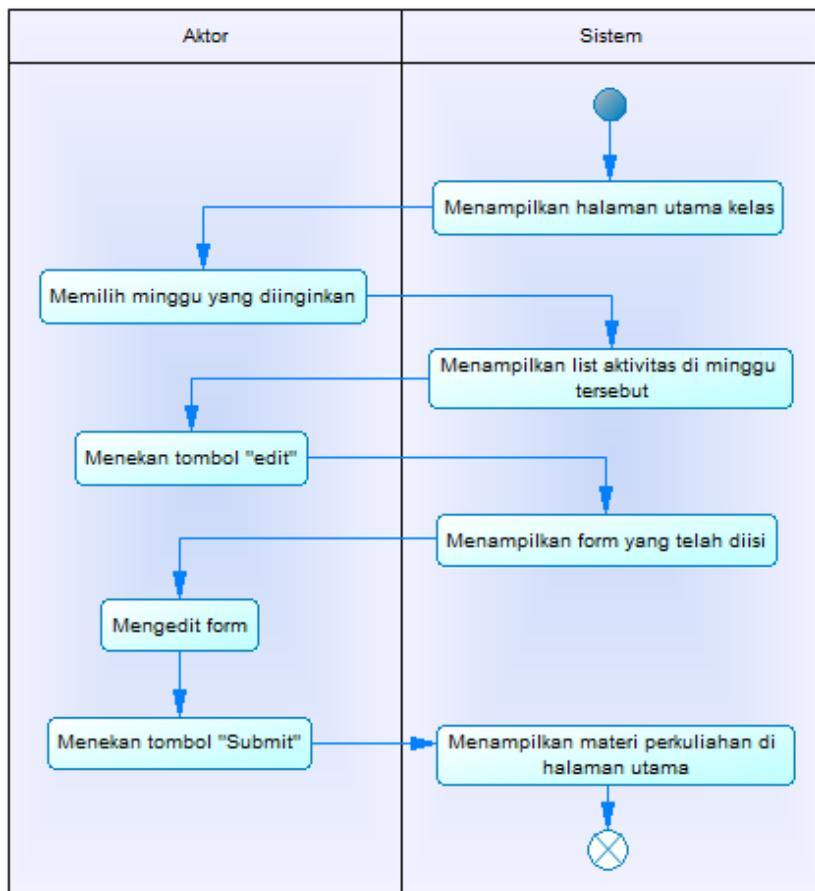
Kode Use Case	UC-004	
Nama Use Case	Mengelola Materi Perkuliahan	
Aktor	Dosen	
Deskripsi	Dosen dapat melihat materi perkuliahan, <i>memposting</i> materi perkuliahan, <i>meng-edit</i> materi perkuliahan serta menghapus materi perkuliahan.	
Relasi	-	
Kondisi Awal	Dosen berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan materi perkuliahan di halaman utama kelas	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan halaman utama kelas
	2. Memilih minggu yang diinginkan	
		3. Menampilkan <i>list</i> aktivitas di minggu tersebut

	4. Melihat <i>item</i> materi perkuliahan	
Alur Kejadian Alternatif	4a. Aktor Menambah Materi Perkuliahan	
		4a.1. Menampilkan halaman utama kelas
	4a.2. Menekan <i>item</i> tambah di minggu yang diinginkan	
		4a.3. Menampilkan <i>form</i> tambah materi perkuliahan
	4a.4. Mengisi <i>form</i> tambah materi perkuliahan	
	4a.5. Menekan tombol “Publish”	
		4a.6. Menampilkan materi pada halaman utama
	4b. Aktor Meng- <i>edit</i> Materi Perkuliahan	
		4b.1. Menampilkan halaman utama kelas
	4b.2. Menekan tombol <i>edit</i> di samping <i>item</i> materi perkuliahan	
		4b.3. Menampilkan <i>form edit</i> materi perkuliahan
	4b.4. Mengisi <i>form edit</i> materi perkuliahan	
	4b.5. Menekan tombol “Publish”	
		4b.6. Menampilkan ulang materi di halaman utama
	4c. Aktor Menghapus Materi Perkuliahan	
		4c.1. Menampilkan halaman utama kelas

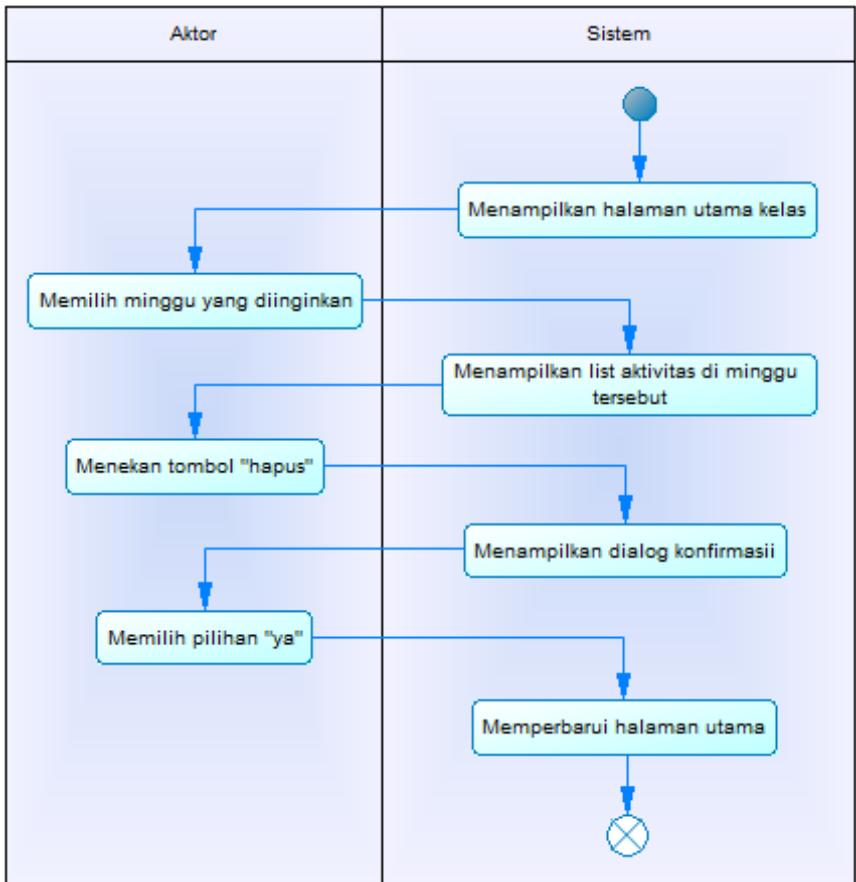
	4c.2 Menekan tombol “Hapus” di samping <i>item</i> materi	
		4c.3. Menampilkan pesan konfirmasi
	4c.6. Memilih pilihan “ya”	
		4c.7. Menampilkan ulang halaman utama



Gambar 3.16. Diagram Aktivitas UC-004 Menambah Materi

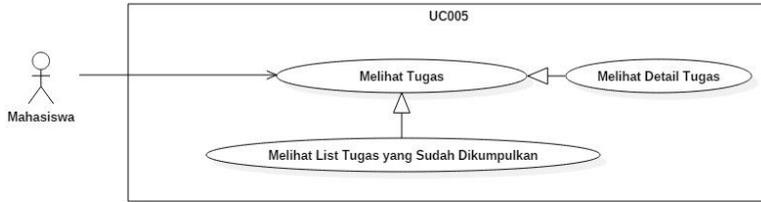


Gambar 3.17. Diagram Aktivitas UC-004 Mengedit Materi



Gambar 3.18. Diagram Aktivitas UC-004 Menghapus Materi

3.1.4.4.5 Melihat Tugas (UC-005)



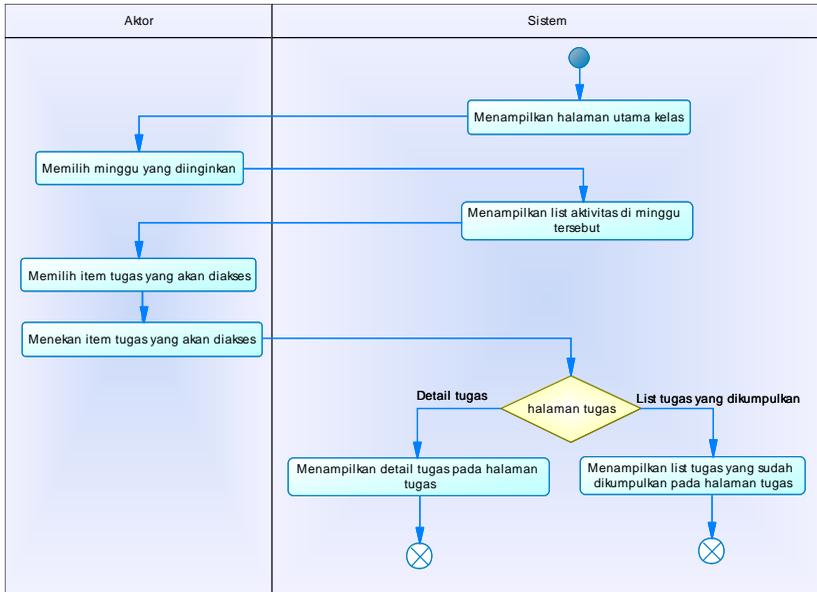
Gambar 3.19. Diagram Kasus Penggunaan UC-005

Kasus penggunaan nomor UC-005 ini diakses ketika mahasiswa hendak melihat detail tugas dan melihat *list* tugas yang sudah ia kumpulkan. Spesifikasi dan diagram aktivitas dapat dilihat pada Tabel 3.8 dan Gambar 3.20.

Tabel 3.8. Spesifikasi Kasus Penggunaan UC-005

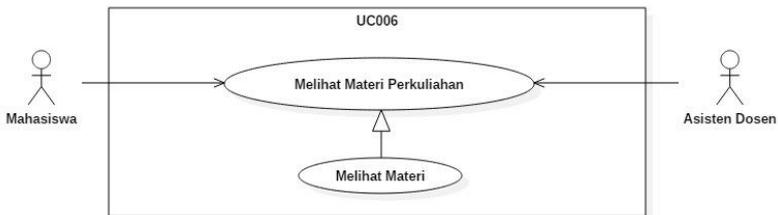
Kode Use Case	UC-005	
Nama Use Case	Melihat Tugas	
Aktor	Mahasiswa	
Deskripsi	Mahasiswa dapat melihat detail tugas pada setiap minggu dan dapat melihat <i>list</i> tugas yang sudah dikumpulkan di kelas tersebut.	
Relasi	-	
Kondisi Awal	Mahasiswa berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan halaman tugas.	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan halaman utama kelas
	2. Memilih minggu yang diinginkan	
		3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	4. Memilih <i>item</i> tugas yang akan diakses	
	5. Menekan <i>item</i> tugas yang akan diakses	

		6. Menampilkan detail tugas pada halaman tugas
Alur Kejadian Alternatif	6a. Aktor Melihat <i>list</i> tugas yang sudah dikumpulkan	
		6a.1. Menampilkan halaman utama kelas
	6a.2. Memilih minggu yang diinginkan	
		6a.3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	6a.4. Memilih <i>item</i> tugas yang akan diakses	
	6a.5. Menekan <i>item</i> tugas yang akan diakses	
		6a.6. Menampilkan <i>list</i> tugas yang sudah dikumpulkan pada halaman tugas



Gambar 3.20. Diagram Aktivitas UC-005

3.1.4.4.6 Melihat Materi Perkuliahan (UC-006)

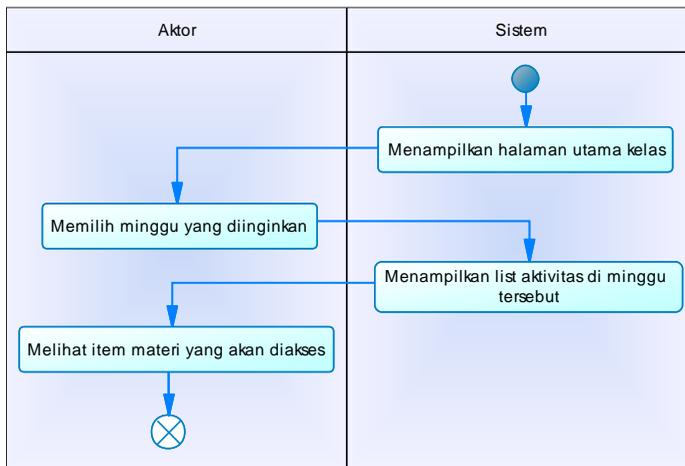


Gambar 3.21. Diagram Kasus Penggunaan UC-006

Kasus penggunaan nomor UC-006 ini diakses ketika mahasiswa dan asisten dosen hendak melihat materi perkuliahan. Spesifikasi dan diagram aktivitas dapat dilihat pada Tabel 3.9 dan Gambar 3.22. Diagram Aktivitas UC-006

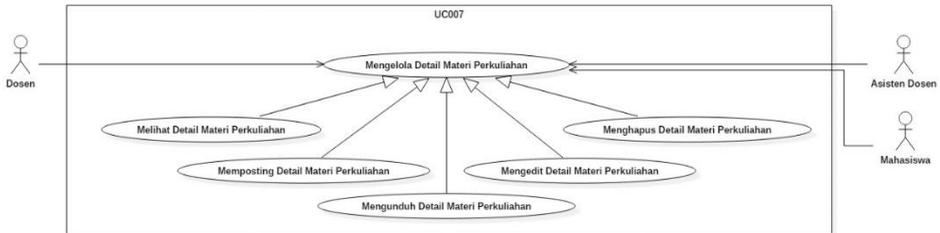
Tabel 3.9. Spesifikasi Kasus Penggunaan UC-006

Kode Use Case	UC-006	
Nama Use Case	Melihat Materi Perkuliahan	
Aktor	Mahasiswa, Asisten Dosen	
Deskripsi	Mahasiswa dan Asisten Dosen dapat melihat materi perkuliahan di masing-masing minggu dan masing-masing kelas.	
Relasi	-	
Kondisi Awal	Mahasiswa berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan materi perkuliahan pada masing-masing minggu.	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan halaman utama kelas
	2. Memilih minggu yang diinginkan	
		3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	4. Melihat <i>item</i> materi yang akan diakses	



Gambar 3.22. Diagram Aktivitas UC-006

3.1.4.4.7 Mengelola Detail Materi Perkuliahan (UC-007)



Gambar 3.23. Diagram Kasus Penggunaan UC-007

Kasus penggunaan nomor UC-007 ini diakses ketika dosen, asisten dosen serta mahasiswa hendak melakukan pengolahan detail materi perkuliahan di masing-masing *item* materi perkuliahan. Spesifikasi, diagram aktivitas dosen dan asisten dosen dapat dilihat pada Tabel 3.10, Gambar 3.24, Gambar 3.25, Gambar 3.26 dan Gambar 3.27. Sedangkan spesifikasi dan diagram aktivitas mahasiswa dapat dilihat pada Tabel 3.11 dan Gambar 3.28.

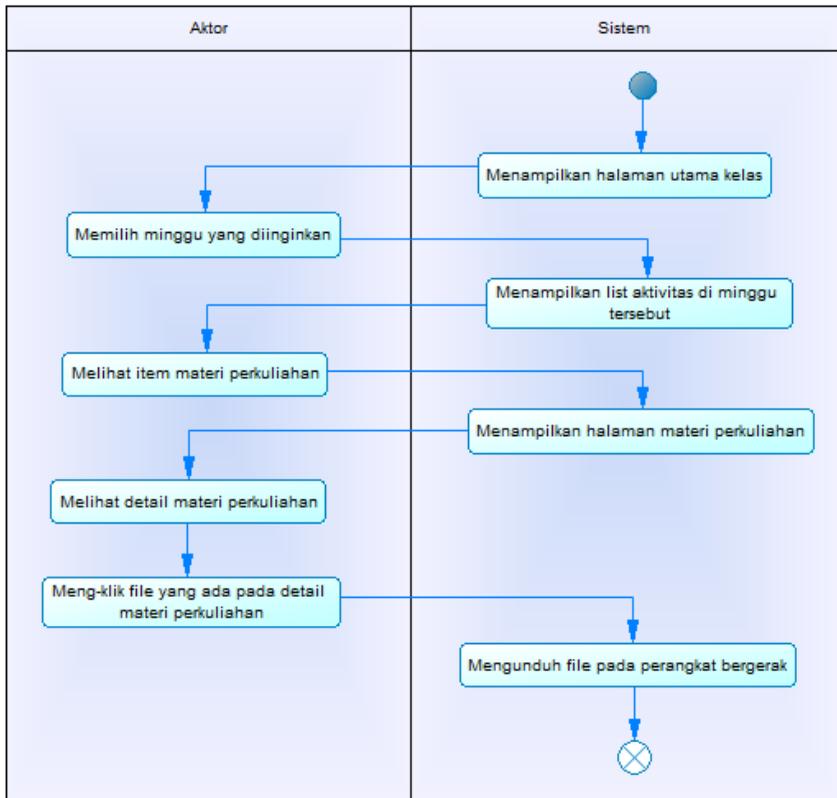
Tabel 3.10. Spesifikasi Kasus Penggunaan UC-007 Dosen dan Asisten Dosen

Kode Use Case	UC-007	
Nama Use Case	Mengelola Detail Materi Perkuliahan	
Aktor	Dosen, Asisten Dosen	
Deskripsi	Dosen dan Asisten Dosen dapat melihat dan mengunduh detail materi perkuliahan, <i>memposting</i> detail materi perkuliahan, <i>meng-edit</i> detail materi perkuliahan serta menghapus detail materi perkuliahan.	
Relasi	-	
Kondisi Awal	Dosen/ Asisten Dosen berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan detail materi perkuliahan di halaman materi perkuliahan.	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan halaman utama kelas

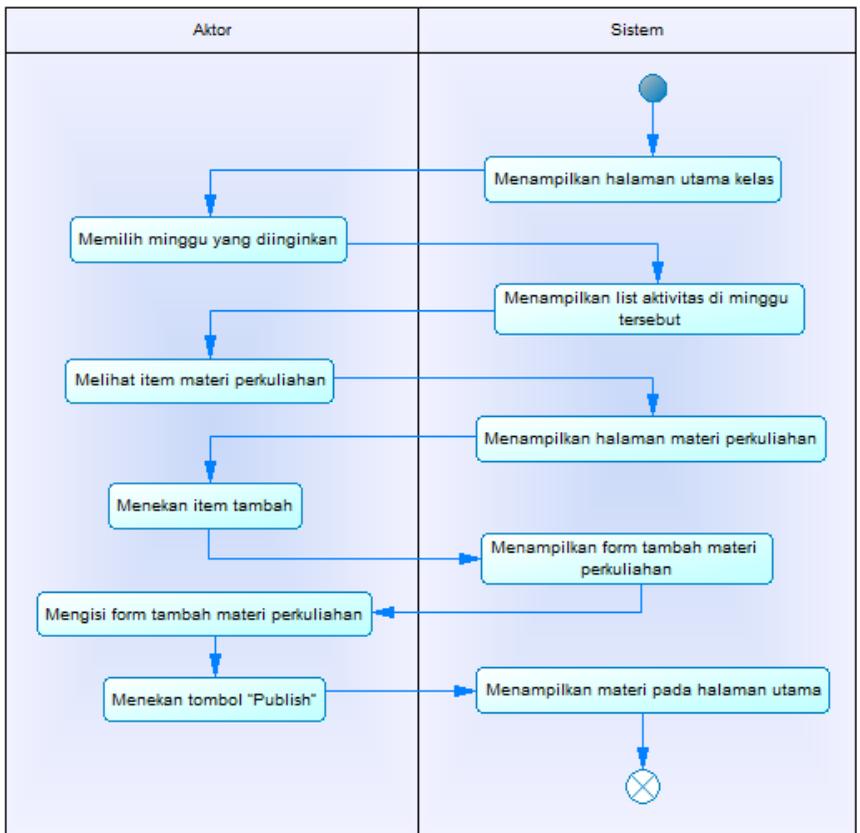
	2. Memilih minggu yang diinginkan	
		3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	4. Melihat <i>item</i> materi perkuliahan	
		5. Menampilkan halaman materi perkuliahan
	6. Melihat detail materi perkuliahan	
Alur Kejadian Alternatif	6a. Aktor Menambah Detail Materi Perkuliahan	
		6a.1. Menampilkan halaman utama kelas
	6a.2. Memilih minggu yang diinginkan	
		6a.3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	6a.4. Melihat <i>item</i> materi perkuliahan	
		6a.5. Menampilkan halaman materi perkuliahan.
	6a.2. Menekan tombol “Tambah Materi”	
		6a.3. Menampilkan <i>form</i> tambah detail materi perkuliahan
	6a.4. Mengisi <i>form</i> tambah detail materi perkuliahan	
	6a.5. Menekan tombol “Publish”	
		6a.6. Menampilkan detail materi pada halaman utama

6b. Aktor Meng- <i>edit</i> Detail Materi Perkuliahan		
		6b.1. Menampilkan halaman utama kelas
	6b.2. Memilih minggu yang diinginkan	
		6b.3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	6b.4. Melihat <i>item</i> materi perkuliahan	
		6b.5. Menampilkan halaman materi perkuliahan.
	6b.6. Menekan tombol <i>edit</i> di samping <i>item</i> detail materi perkuliahan	
		6b.7. Menampilkan <i>form edit</i> detail materi perkuliahan
	6b.8. Mengisi <i>form edit</i> detail materi perkuliahan	
	6b.9. Menekan tombol “Publish”	
	6b.10. Menampilkan ulang detail materi di halaman utama	
6c. Aktor Menghapus Detail Materi Perkuliahan		
		6c.1. Menampilkan halaman utama kelas
	6c.2. Memilih minggu yang diinginkan	
		6c.3. Menampilkan <i>list</i> aktivitas di minggu tersebut

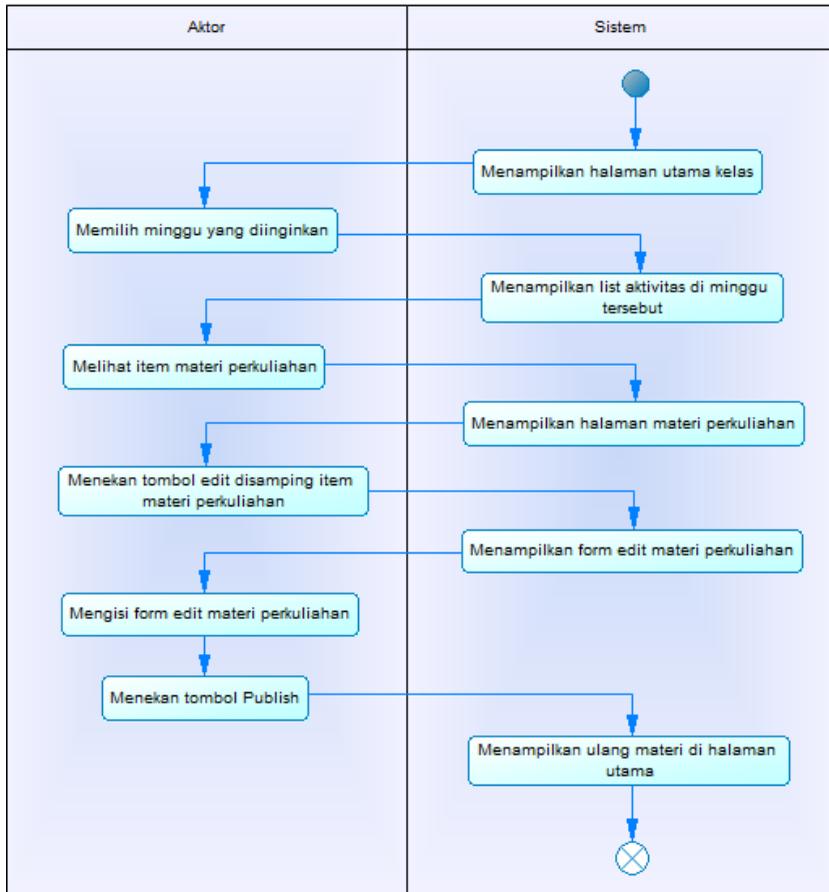
	6c.4. Melihat <i>item</i> materi perkuliahan	
		6c.5. Menampilkan halaman materi perkuliahan.
	6c.6 Menekan tombol “Hapus” di samping <i>item</i> detail materi	
		6c.7. Menampilkan pesan konfirmasi
	6c.8. Memilih pilihan “ya”	
		6c.9. Menampilkan ulang halaman materi perkuliahan
	6d. Mengunduh detail materi perkuliahan	
		6d.1. Menampilkan halaman utama kelas
	6d.2. Memilih minggu yang diinginkan	
		6d.3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	6d.4. Melihat <i>item</i> materi perkuliahan	
		6d.5. Menampilkan halaman materi perkuliahan.
	6d.6. Meng-klik <i>file</i> yang ada pada detail materi perkuliahan	
		6d.7. Mengunduh <i>file</i> pada perangkat bergerak



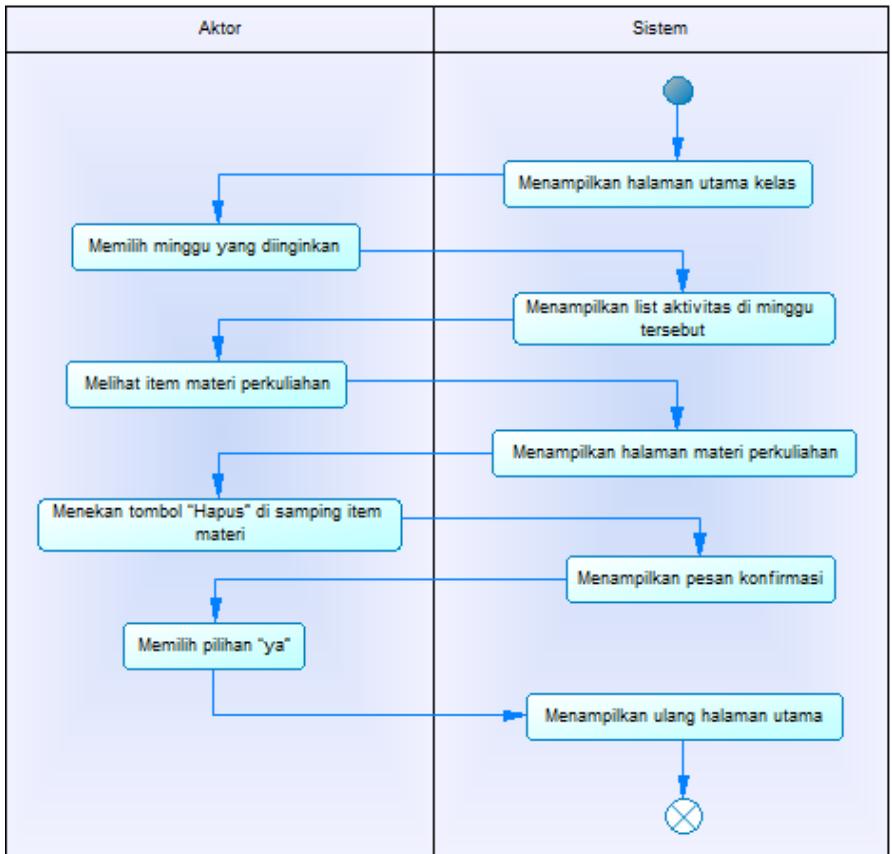
Gambar 3.24. Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Melihat Detail Materi



Gambar 3.25 Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Menambah Detail Materi



Gambar 3.26. Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Mengedit Detail Materi



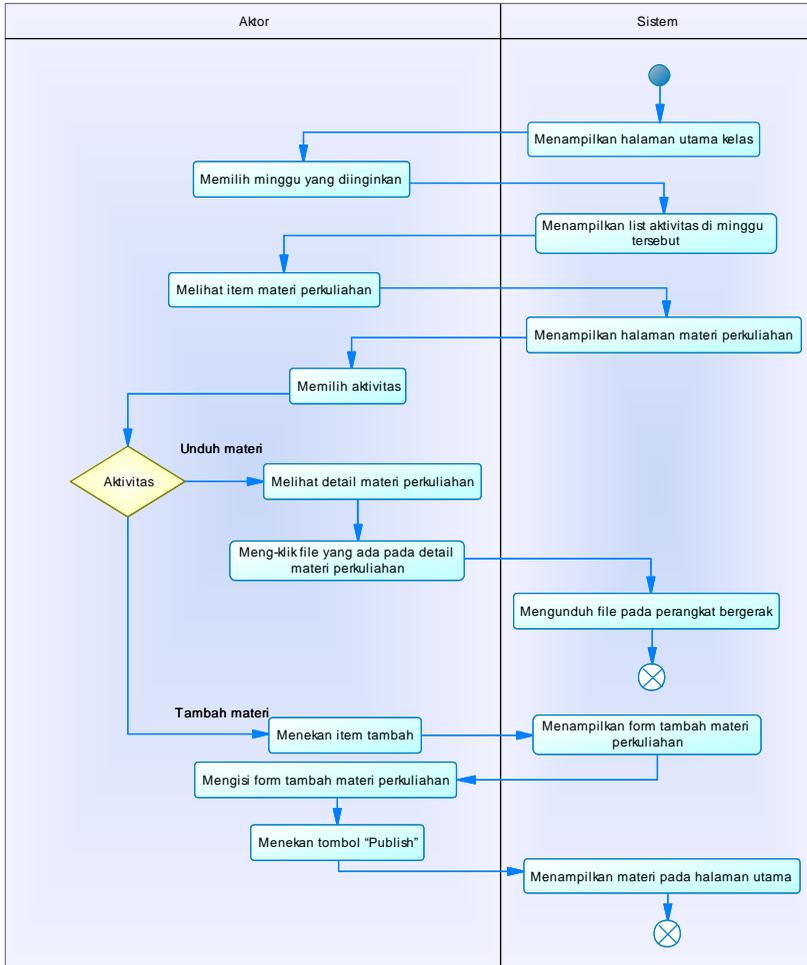
Gambar 3.27. Diagram Aktivitas UC-007 Dosen dan Asisten Dosen Menghapus Detail Materi

Tabel 3.11. Spesifikasi Kasus Penggunaan UC-007 Mahasiswa

Kode Use Case	UC-007
Nama Use Case	Mengelola Detail Materi Perkuliahan
Aktor	Mahasiswa
Deskripsi	Mahasiswa dapat melihat dan mengunduh detail materi perkuliahan, mem- <i>posting</i> detail materi perkuliahan.
Relasi	-

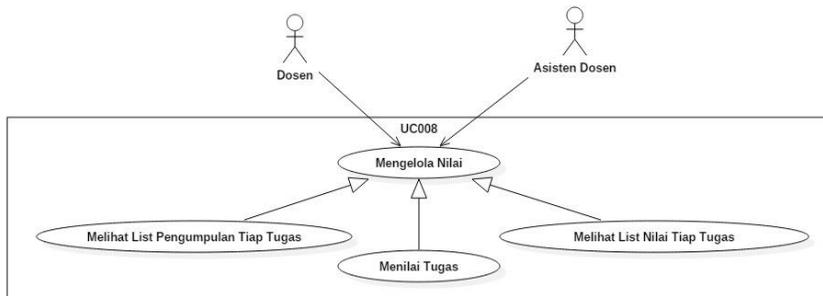
Kondisi Awal	Mahasiswa berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan detail materi perkuliahan di halaman materi perkuliahan.	
Alur Kejadian Normal	Aktor	Sistem
		1. Menampilkan halaman utama kelas
	2. Memilih minggu yang diinginkan	
		3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	4. Melihat <i>item</i> materi perkuliahan	
		5. Menampilkan halaman materi perkuliahan.
Alur Kejadian Alternatif	6a. Aktor Menambah Detail Materi Perkuliahan	
		6a.1. Menampilkan halaman utama kelas
	6a.2. Memilih minggu yang diinginkan	
		6a.3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	6a.4. Melihat <i>item</i> materi perkuliahan	
		6a.5. Menampilkan halaman materi perkuliahan.
	6a.2. Menekan tombol “Tambah Materi”	
		6a.3. Menampilkan <i>form</i> tambah detail materi perkuliahan

	6a.4. Mengisi <i>form</i> tambah detail materi perkuliahan	
	6a.5. Menekan tombol “Publish”	
		6a.6. Menampilkan detail materi pada halaman utama
	6b. Mengunduh detail materi perkuliahan	
		6b.1. Menampilkan halaman utama kelas
	6b.2. Memilih minggu yang diinginkan	
		6b.3. Menampilkan <i>list</i> aktivitas di minggu tersebut
	6b.4. Melihat <i>item</i> materi perkuliahan	
		6b.5. Menampilkan halaman materi perkuliahan.
	6b.6. Meng- <i>klik file</i> yang ada pada detail materi perkuliahan	
		6b.7. Mengunduh <i>file</i> pada perangkat bergerak



Gambar 3.28. Diagram Aktivitas UC-007 Mahasiswa

3.1.4.4.8 Mengelola Nilai



Gambar 3.29. Diagram Kasus Penggunaan UC-008

Kasus penggunaan nomor UC-008 ini diakses ketika dosen dan asisten dosen hendak melakukan pengolahan nilai di masing-masing minggu dan masing-masing tugas. Untuk menjelaskan peran aktor, maka ada spesifikasi dan diagram aktivitas untuk masing-masing aktor.

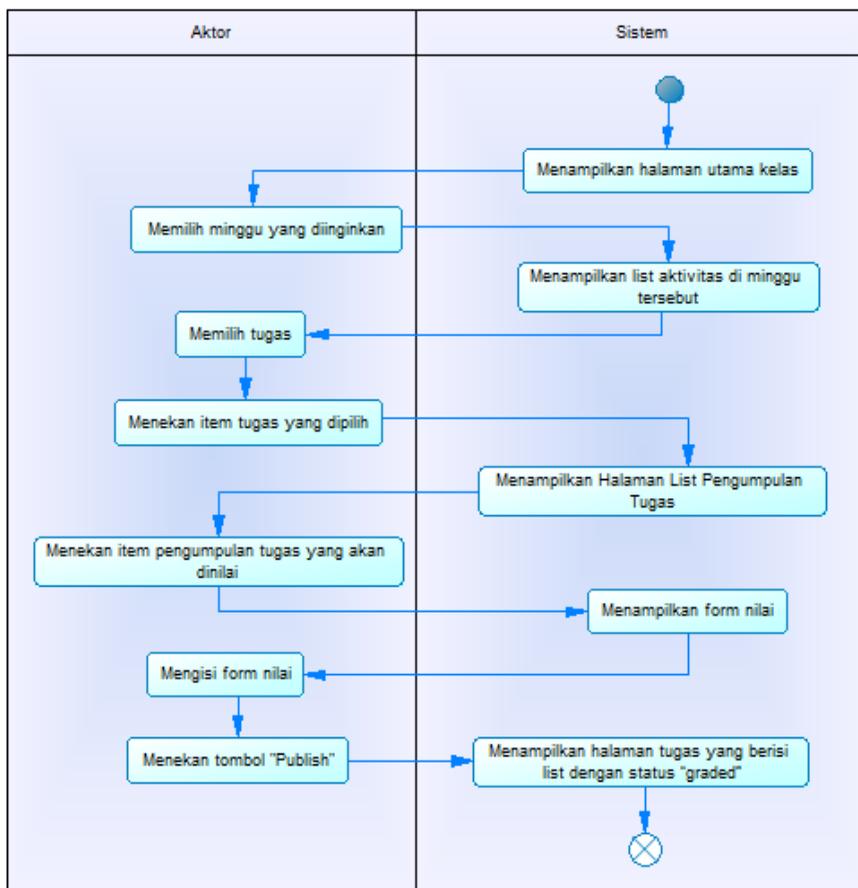
Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.12, sedangkan diagram aktivitas menilai dan melihat list nilai dapat dilihat pada Gambar 3.30 dan untuk diagram aktivitas melihat list pengumpulan tugas dapat dilihat pada Gambar 3.31.

Tabel 3.12. Spesifikasi Kasus Penggunaan UC-008

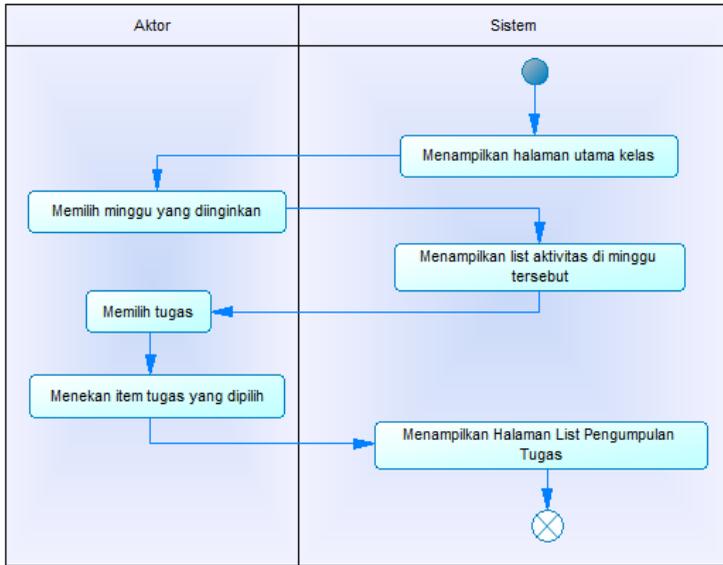
Kode Use Case	UC-008	
Nama Use Case	Mengelola Nilai	
Aktor	Dosen, Asisten Dosen	
Deskripsi	Aktor dapat melihat <i>list</i> pengumpulan tiap tugas, melihat <i>list</i> nilai tiap tugas, menilai tugas yang telah dikumpulkan oleh mahasiswa	
Relasi	-	
Kondisi Awal	Aktor berada pada halaman utama kelas di menu E-Kelas.	
Kondisi Akhir	Sistem menampilkan halaman tugas	
Alur Kejadian Alternatif	Aktor	Sistem
		1. Menampilkan tugas di halaman utama

	2. Memilih tugas yang diinginkan	
	3. Menekan <i>item</i> tugas yang dipilih	
		4. Menampilkan halaman tugas yang berisi <i>list</i> pengumpulan tugas
	4b. Aktor menilai tugas	
		4b.1. Menampilkan tugas di halaman utama
	4b.2. Memilih tugas yang diinginkan	
	4b.3. Menekan <i>item</i> tugas yang dipilih	
		4b.4. Menampilkan halaman tugas yang berisi <i>list</i> pengumpulan tugas
	4b.5. Menekan <i>item</i> pengumpulan tugas yang akan dinilai	
		4b.6. Menampilkan <i>form</i> nilai
	4b.7. Mengisi <i>form</i> nilai	
	4b.8. Menekan tombol " <i>publish</i> "	
		4b.9. Menampilkan ulang halaman tugas dengan status " <i>submitted</i> " berubah menjadi " <i>graded</i> "
	4c. Melihat <i>list</i> nilai tiap tugas	
		4c.1. Menampilkan tugas di halaman utama
	4c.2. Memilih tugas yang diinginkan	
	4c.3. Menekan <i>item</i> tugas yang dipilih	

	4c.4. Menampilkan halaman tugas yang berisi <i>list</i> dengan status “graded”
--	--



Gambar 3.30. Diagram Aktivitas UC-008 Menilai dan Melihat List Nilai



Gambar 3.31. Diagram Aktivitas UC-008 List Pengumpulan Tugas

3.2 Perancangan

Perancangan dalam subbab ini membahas perancangan dari aplikasi tugas akhir. Subbab ini terdiri dari perancangan struktur data, perancangan antarmuka pengguna. Lingkungan perancangan perangkat lunak, perancangan basis data, dan perancangan antarmuka pengguna.

3.2.1 Lingkungan Perancangan Perangkat Lunak

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap perancangan perangkat lunak tugas akhir ini seperti dijelaskan pada Tabel 3.13.

Tabel 3.13 Lingkungan Perancangan Perangkat Lunak

Perangkat Keras	Komputer	Asus ROG GL553VD-FY380
	Prosesor	Intel® Core™ i7-7700HQ CPU @ 2.80GHz (2.81GHz)

	Memori Primer	16 GB
	Memori Sekunder	128GB SSD, 1TB HDD
Perangkat Lunak	Sistem Operasi	<i>Windows</i> 10 Pro 64-bit
	Perangkat Lunak	Android Studio v3.1.1, Justinmind Prototyper 7.1.1, StarUML v2.8.1, Sybase PowerDesigner 16.5, Microsoft Word 2016, Eclipse Oxygen.1 2017 v4.7.1, WSO2 Application Manager, MySQL.

3.2.2 Perancangan Struktur Data

Dalam membuat suatu aplikasi perangkat bergerak, diperlukan analisis kebutuhan berupa perancangan basis data. Basis data yang digunakan adalah MySQL. MySQL ini dipilih menjadi basis data aplikasi ini karena sifat RDBMS yang *open source*, mudah digunakan, dan memiliki fitur yang lebih lengkap serta *multi platform*.

Perancangan basis data pada aplikasi *e-Learning* ini menggunakan skema yang sudah ada pada *platform* Moodle. Rancangan basis data ditampilkan dalam bentuk *Conceptual Data Model* (selanjutnya disebut CDM) dan *Physical Data Model* (selanjutnya disebut PDM). Penjelasan lebih lengkap mengenai PDM dapat dilihat pada halaman 223 bagian LAMPIRAN. Untuk CDM *user* dapat diakses pada Gambar 8.1, CDM *assign* dapat diakses pada Gambar 8.2, CDM *glossary* dapat diakses pada Gambar 8.3, CDM *course* dapat diakses pada Gambar 8.4. Sedangkan untuk PDM *user* dapat diakses pada Gambar 8.5, PDM *assign* dapat diakses pada Gambar 8.6, PDM *glossary* dapat diakses pada Gambar 8.7, dan PDM *course* dapat diakses pada Gambar 8.8.

3.2.2.1 Tabel *assign*

Tabel *assign* adalah tabel yang digunakan untuk menyimpan data tugas.

3.2.2.2 Tabel *assign_grades*

Tabel *assign_grades* adalah tabel yang digunakan untuk menyimpan data nilai tugas.

3.2.2.3 Tabel *assign_plugin_config*

Tabel *assign_plugin_config* adalah tabel yang digunakan untuk menyimpan data yang dapat mengaktifkan fitur pengumpulan tugas.

3.2.2.4 Tabel *assign_submission*

Tabel *assign_submission* adalah tabel yang digunakan untuk menyimpan data pengumpulan tugas.

3.2.2.5 Tabel *assign_user_flags*

Tabel *assign_user_flags* adalah tabel yang digunakan untuk menyimpan data tugas *user* yang sudah dinilai.

3.2.2.6 Tabel *assignfeedback_comments*

Tabel *assignfeedback_comments* adalah tabel yang digunakan untuk menyimpan data *comment* penilaian pada fitur menilai tugas.

3.2.2.7 Tabel *assignsubmission_onlinetext*

Tabel *assignsubmission_onlinetext* adalah tabel yang digunakan untuk menyimpan data jawaban tugas dalam bentuk narasi yang dikerjakan oleh *user*.

3.2.2.8 Tabel *assignsubmission_file*

Tabel *assignsubmission_file* adalah tabel yang digunakan untuk menyimpan data jawaban tugas dalam bentuk *file* yang diunggah oleh *user*.

3.2.2.9 Tabel *context*

Tabel *context* adalah tabel yang digunakan untuk menyimpan data tipe modul pada seluruh aplikasi Moodle.

3.2.2.10 Tabel *course*

Tabel *course* adalah tabel yang digunakan untuk menyimpan data-data kelas yang ada.

3.2.2.11 Tabel *course_categories*

Tabel *course_categories* adalah tabel yang digunakan untuk menyimpan data-data *filter* kelas yang digunakan untuk membagi *course* ke tiap-tiap kategori seperti semester.

3.2.2.12 Tabel *course_modules*

Tabel *course_module* adalah tabel yang digunakan untuk menyimpan data-data modul/aktivitas yang ada pada suatu kelas.

3.2.2.13 Tabel *course_sections*

Tabel *course_sections* adalah tabel yang digunakan untuk menyimpan data-data lokasi modul/aktivitas berada pada suatu minggu pada kelas.

3.2.2.14 Tabel *files*

Tabel *files* adalah tabel yang digunakan untuk menyimpan data-data *file* pada sistem Moodle yang digunakan untuk materi kuliah dan tugas.

3.2.2.15 Tabel *glossary*

Tabel *glossary* adalah tabel yang digunakan untuk menyimpan data-data materi perkuliahan.

3.2.2.16 Tabel *glossary_entries*

Tabel *glossary_entries* adalah tabel yang digunakan untuk menyimpan data-data detail materi perkuliahan.

3.2.2.17 Tabel *role_assignments*

Tabel *role_assignments* adalah tabel yang digunakan untuk menyimpan data-data peran *user* pada masing-masing kelas.

3.2.2.18 Tabel *user*

Tabel *user* adalah tabel yang digunakan untuk menyimpan data-data *user* pada aplikasi *e-learning*.

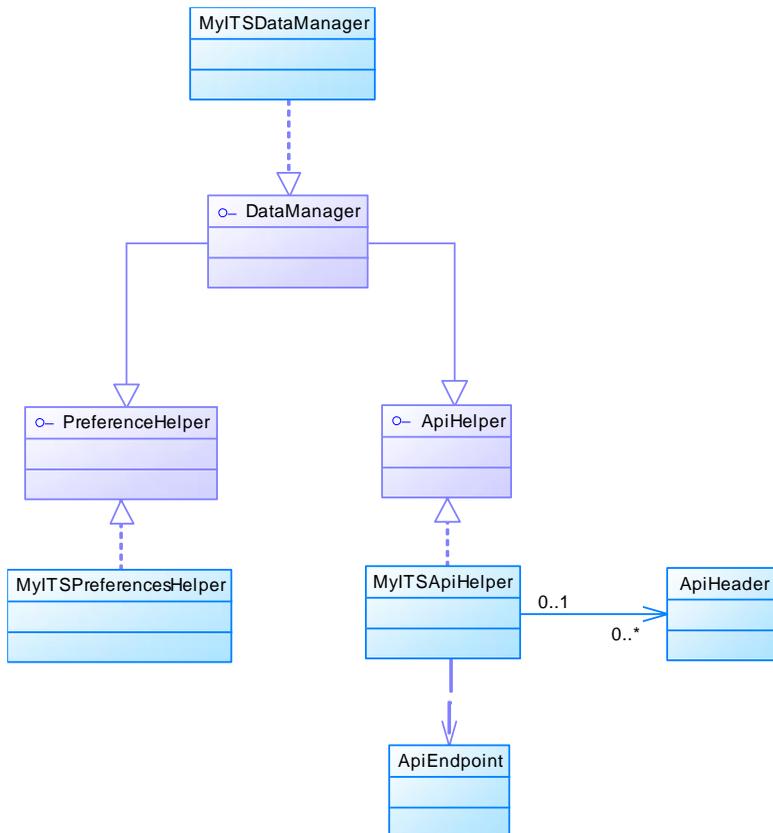
3.2.2.19 Tabel *user_enrolments*

Tabel *user_enrolments* adalah tabel yang digunakan untuk menyimpan data-data hak akses *user* ke masing-masing modul/aktivitas.

3.2.3 Perancangan Arsitektur Model View Presenter

Pada subbab ini akan dibahas secara detail perihal perancangan *Model View Presenter* untuk pembangunan fitur data kelas, data peserta kelas, nilai, tugas dan materi perkuliahan pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa. Setiap kelas memiliki sebuah kelas *interface*. Tujuan Penggunaan *interface* adalah untuk memisahkan abstraksi dan implementasi abstraksi tersebut. Hal ini sejalan dengan prinsip *dependency inversion* yang memiliki tujuan untuk mengabstraksi setiap kelas supaya kelas memiliki ketergantungan yang rendah dan mudah dibongkar pasang. Kelas *interface* mengandung nama *contract*. Jika ada sebuah kelas yang ingin mengakses atau menginstansiasi kelas lain, kelas tersebut akan menggunakan *interface* dari kelas yang ingin diakses.

3.2.3.1 Perancangan Model

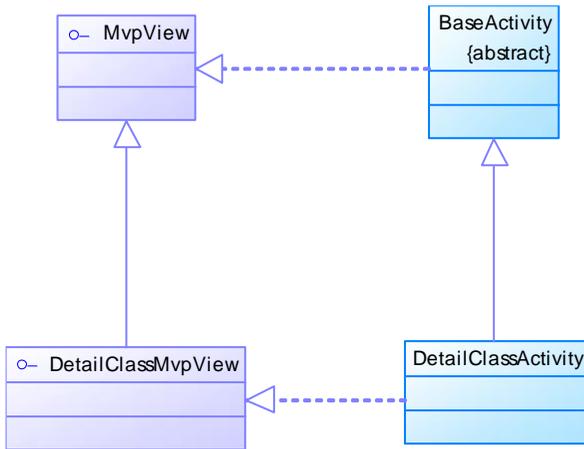


Gambar 3.32. Diagram Struktur Model

Diagram struktur *model* terdiri dari lima kelas dan tiga *interface* seperti yang dapat dilihat di Gambar 3.32. Kelas `MyITSDataManager` digunakan sebagai kelas yang diakses presenter untuk mengakses data dan yang mendelegasikan dan menggambarkan seluruh fungsi yang terdapat di kelas `MyITSPreferencesHelper` dan `ApiHelper`. Tujuan kelas `MyITSDataManager` digunakan adalah untuk membuat akses ke lapisan model dan hanya bisa dilakukan melalui kelas tersebut. Kelas `MyITSPreferencesHelper` berisi fungsi-fungsi yang

digunakan untuk melakukan operasi data dengan menggunakan *shared preference*. `MyITSApiHelper` berisi fungsi-fungsi yang digunakan untuk melakukan operasi data dengan API. Seluruh konstanta *endpoint* API diletakkan di kelas `ApiEndpoint`. Sedangkan kelas `ApiHeader` yang akan digunakan setiap kali kelas `MyITSApiHelper` melakukan *request*.

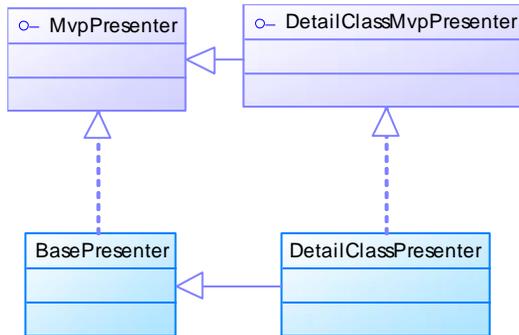
3.2.3.2 Perancangan View



Gambar 3.33. Diagram Struktur View

Diagram struktur *view* pada Gambar 3.33 memiliki kelas *abstract* dengan nama kelas yang mengandung “*Base*”. Setiap kelas dan *interface activity* atau *fragment* akan meng-*extend* kelas `BaseActivity` atau `BaseFragment`. Tujuan kelas *Base* adalah untuk meletakkan fungsi-fungsi yang akan digunakan di setiap kelas *activity* atau *fragment*. Contoh fungsi tersebut adalah *action bar*, *dependency injection* dan *binder* Butterknife. Alasan *abstract* digunakan karena tidak semua fungsi harus digunakan oleh kelas *activity* atau *fragment*. Selain itu fungsi yang didefinisikan di kelas *abstract* dapat di-*override* dengan fungsi yang lain. Kelas *activity* hanya digunakan untuk mengatur *view*, instansiasi *dependency injection*, validasi dan memanggil *interface presenter*.

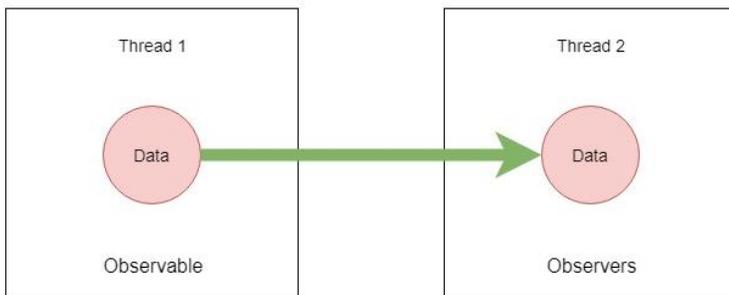
3.2.3.3 Perancangan Presenter



Gambar 3.34. Diagram Struktur Presenter

Diagram struktur *presenter* memiliki sebuah kelas *BasePresenter* yang akan di-*extend* oleh setiap kelas *presenter* berdasarkan Gambar 3.34. Kelas *BasePresenter* digunakan dengan tujuan untuk melakukan instansiasi *interface DataManager* dan *MvpView*. Kedua kelas tersebut perlu diinstansiasi karena lapisan *presenter* memiliki akses ke lapisan *model* dan *view*. Selain itu, kelas *BasePresenter* juga memiliki fungsi yang digunakan untuk menangani *error handling request API*. Setiap lapisan *view* yang membutuhkan akses dengan data pasti memiliki sebuah kelas *presenter*.

3.2.3.4 Perancangan Pemrograman Reaktif

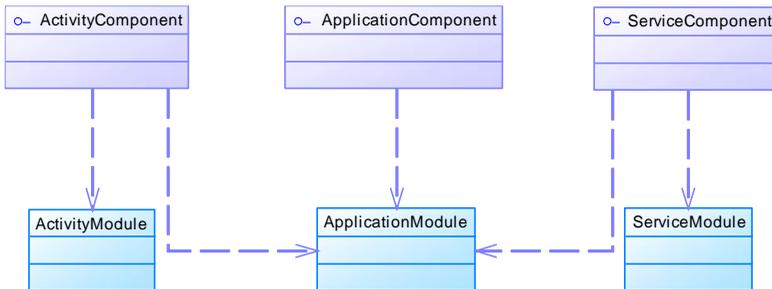


Gambar 3.35. Diagram Perancangan Pemrograman Reaktif

Pemrograman Reaktif akan diimplementasikan untuk setiap API *request* untuk fitur data kelas, data peserta kelas, nilai, tugas, dan materi perkuliahan yang dilakukan pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa. Karena operasi API *request* berpotensi memakan waktu yang tidak sebentar, maka operasi *asynchronous* dengan menggunakan pemrograman reaktif harus dilakukan. Pengaturan penggunaan *thread* akan dilakukan sehingga *request* API tidak mengganggu *main thread* aplikasi.

Berdasarkan Gambar 3.35, terdapat istilah *observable* dan *observers*. *Observable* adalah suatu operasi yang menghasilkan data. Sedangkan *observers* adalah operasi yang mengambil data dari *observable*. Dapat dilihat bahwa *observable* dan *observers* dijalankan pada *thread* yang berbeda. Bagian yang melakukan manajemen *thread* tersebut dinamakan *schedulers*. Pada aplikasi ini, seluruh *network request* ke API akan menggunakan *observable*. Sedangkan hasil data yang diterima dari API akan diproses dengan *observers*.

3.2.3.5 Perancangan Dependency Injection



Gambar 3.36. Diagram Struktur Dependency Injection

Dependency injection pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa menggunakan tiga kelas dan *interface* seperti di Gambar 3.36. Kelas yang mengandung nama “Module” digunakan sebagai kelas yang mendefinisikan parameter dan fungsi-fungsi yang dibutuhkan sebuah kelas untuk menginstansiasi kelas lain. *Interface* digunakan sebagai jembatan yang

menghubungkan kelas *module* dan kelas yang mendefinisikan *dependency injection*. Kelas `ActivityModule` dan `ServiceModule` terdiri dari fungsi-fungsi dan parameter yang dibutuhkan kelas *activity* dan *service*. Sedangkan kelas `ApplicationModule` terdiri dari konstanta *database* dan *shared preference* dan kelas-kelas yang di-*inject* menggunakan anotasi *singleton*. *Singleton* menandakan kelas tersebut hanya akan diinstansiasi hanya sekali dalam *runtime*. Seluruh kelas yang ada di lapisan *model* akan menggunakan *singleton* karena setiap kelas *presenter* pasti akan mengakses kelas `MyITSDataManager`. Jika setiap kelas *presenter* memiliki sebuah *instance* untuk `MyITSDataManager`, maka penggunaan *dependency injection* menjadi tidak efisien.

3.2.4 Perancangan Antarmuka Pengguna

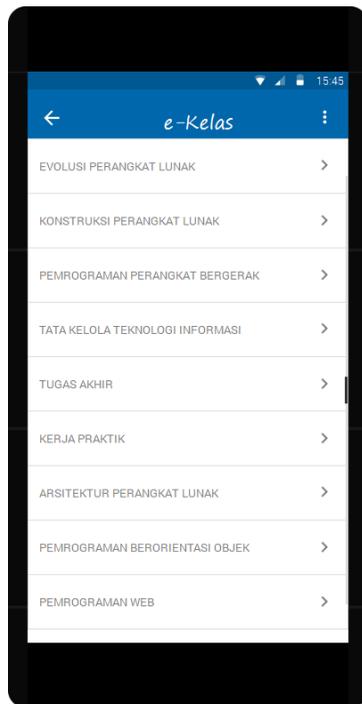
Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan dan tampilan yang menarik bagi penggunanya. Aplikasi ini memiliki tiga aktor, yaitu dosen, asisten dosen dan mahasiswa. Pada aktor dosen, aplikasi ini memiliki beberapa antarmuka, yaitu halaman *list* kelas, halaman utama kelas, halaman data kelas, halaman *form* data kelas, halaman *form* tambah materi, halaman materi kuliah, halaman *form* tambah detail materi kuliah, halaman *form* tambah tugas, halaman pengumpulan tugas, halaman *form* menilai tugas, halaman *list* nilai. Pada aktor asisten dosen, aplikasi ini memiliki beberapa antarmuka, yaitu, halaman *list* kelas, halaman utama kelas, halaman materi kuliah, halaman *form* tambah detail materi kuliah, halaman pengumpulan tugas, halaman *form* menilai tugas, serta halaman *list* nilai. Sedangkan pada aktor mahasiswa, aplikasi ini memiliki beberapa antarmuka, yaitu, halaman *list* kelas, halaman utama kelas, halaman materi kuliah, halaman *form* tambah detail materi kuliah serta halaman detail tugas mahasiswa.

3.2.4.1 Rancangan Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan pada aplikasi ketika aktor melihat *list* kelas pada tampilan antarmuka halaman *list* kelas. Atribut antarmuka dapat dilihat pada Tabel 3.14. Rancangan dapat dilihat pada Gambar 3.37.

Tabel 3.14. Atribut Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>rv_class_view</i>	<i>Recycleview</i>	Menampilkan <i>list</i> nama kelas	<i>String</i>



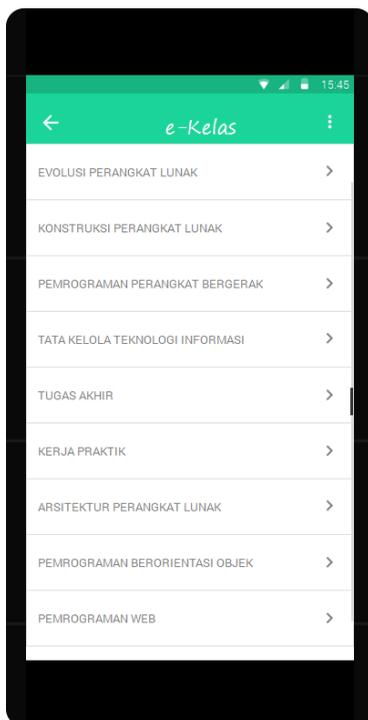
Gambar 3.37. Rancangan Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Dosen

3.2.4.2 Rancangan Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan pada aplikasi ketika aktor melihat *list* kelas pada tampilan antarmuka halaman *list* kelas. Atribut antarmuka dapat dilihat pada Tabel 3.15. Rancangan dapat dilihat pada Gambar 3.38.

Tabel 3.15. Atribut Antarmuka Melihat List Kelas pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/Keluaran
1	<i>rv_class_view</i>	<i>Recycleview</i>	Menampilkan <i>list</i> nama kelas	<i>String</i>



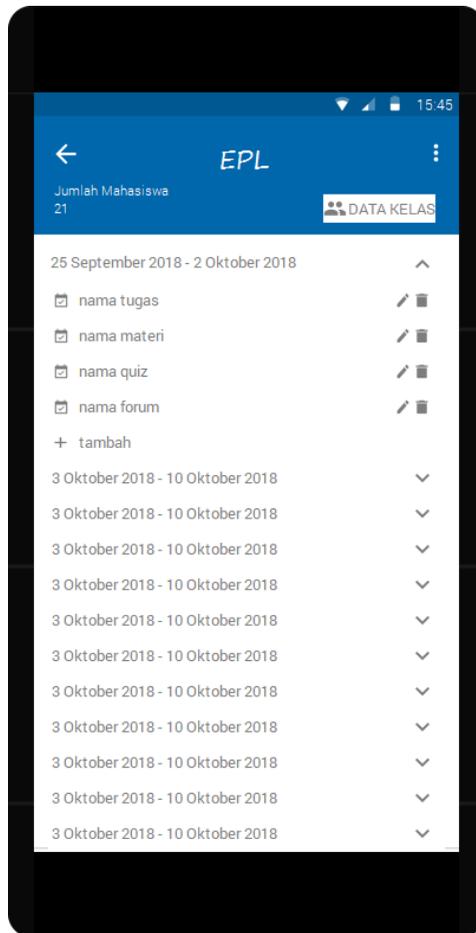
Gambar 3.38. Rancangan Halaman Antarmuka Melihat List Kelas pada Aplikasi MyITS Mahasiswa

3.2.4.3 Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen

Halaman ini ditampilkan pada aplikasi ketika aktor melihat halaman utama kelas pada tampilan antarmuka halaman utama kelas. Atribut antarmuka dapat dilihat pada Tabel 3.16. Rancangan halaman dapat dilihat pada Gambar 3.39.

Tabel 3.16. Atribut Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	tvCount UserMhs	<i>TextView</i>	Menampilkan jumlah mahasiswa di kelas tersebut	<i>String</i>
2	tvData DetailClass	<i>TextView</i>	Menampilkan data peserta kelas	<i>String</i>
3	Exp_Detail Class	<i>Expendables ListView</i>	Menampilkan minggu perkuliahan	<i>String</i>
4	tvNama Content	<i>TextView</i>	Menampilkan nama modul/ aktivitas	<i>String</i>
5	ivEkelas EditContent	<i>ImageView</i>	Tombol aksi untuk menuju ke halaman <i>form</i> tambah modul	<i>ImageClick</i>
6	ivEkelas DeleteContent	<i>ImageView</i>	Tombol aksi untuk menghapus modul dari halaman utama kelas	<i>ImageClick</i>



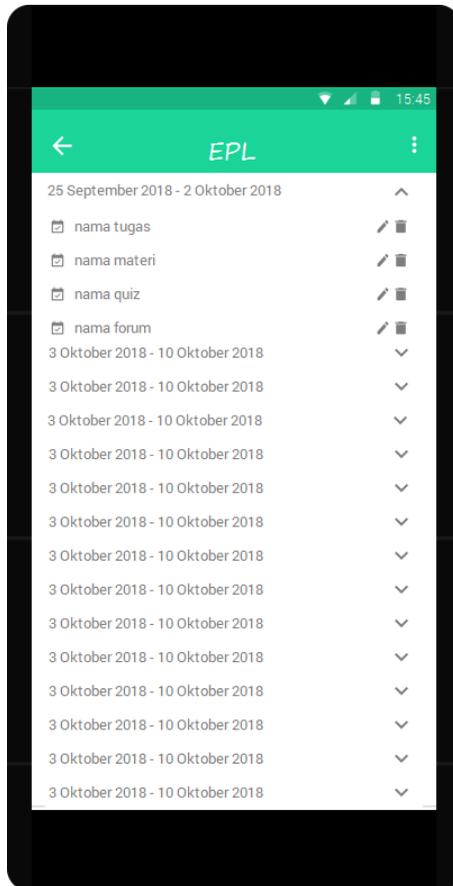
Gambar 3.39. Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen

3.2.4.4 Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa

Halaman ini ditampilkan pada aplikasi ketika aktor melihat halaman utama kelas pada tampilan antarmuka halaman utama kelas. Atribut antarmuka dapat dilihat pada Tabel 3.17. Rancangan halaman dapat dilihat pada Gambar 3.40.

Tabel 3.17. Atribut Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	Exp_Detail Class	<i>Expendables ListView</i>	Menampilkan minggu perkuliahan	<i>String</i>
2	tvNama Content	<i>TextView</i>	Menampilkan nama modul/ aktivitas	<i>String</i>
3	ivEkelas EditContent	<i>ImageView</i>	Tombol aksi untuk menuju ke halaman <i>form</i> tambah modul	<i>ImageClick</i>
4	ivEkelas DeleteContent	<i>ImageView</i>	Tombol aksi untuk menghapus modul dari halaman utama kelas	<i>ImageClick</i>



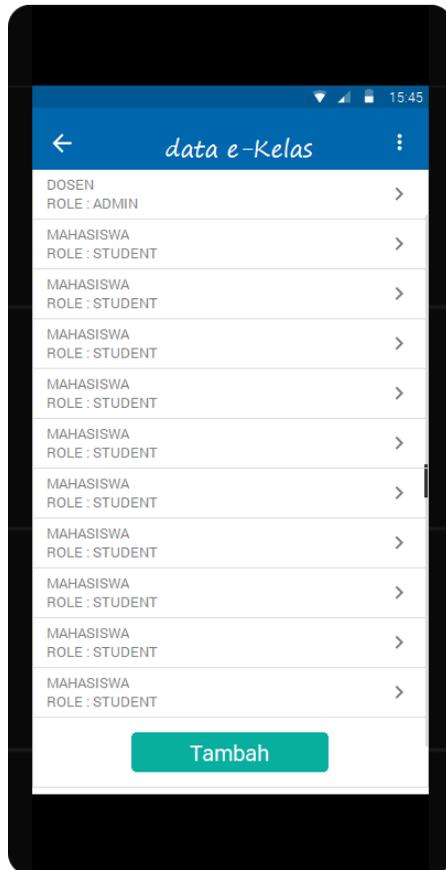
Gambar 3.40. Rancangan Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa

3.2.4.5 Rancangan Halaman Antarmuka Melihat Data Kelas

Halaman ini akan ditampilkan sistem ketika aktor melihat data kelas pada halaman antarmuka melihat data kelas. Atribut antarmuka dapat dilihat pada Tabel 3.18. Rancangan dapat dilihat pada Gambar 3.41.

Tabel 3.18. Atribut Antarmuka Melihat Data Kelas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	groups_recycler_view	<i>RecyclerView</i>	Menampilkan <i>list</i> peserta dalam kelas tersebut	<i>String</i>
2	btnTambah Groups	<i>Button</i>	Tombol untuk menuju ke halaman <i>form</i> tambah data kelas	<i>ButtonClick</i>
3	tvGroups Name	<i>TextView</i>	Menampilkan nama kelas	<i>String</i>
4	tvGroups Description	<i>TextView</i>	Menampilkan informasi data kelas	<i>String</i>



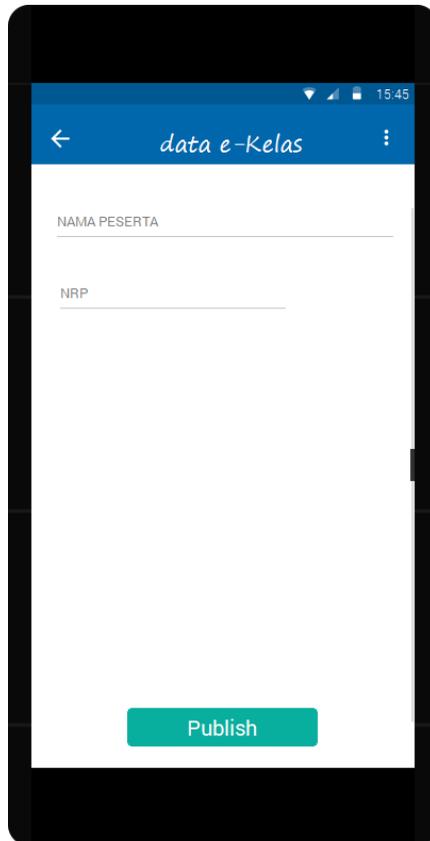
Gambar 3.41. Rancangan Halaman Antarmuka Melihat Data Kelas

3.2.4.6 Rancangan Halaman Antarmuka Menambah Data Kelas

Halaman ini akan ditampilkan ketika aktor menekan pilihan tambah data kelas pada halaman antarmuka menambah data kelas. Halaman ini akan memuat nama dan NRP mahasiswa yang sedang di-*input*-kan oleh dosen. Atribut antarmuka dapat dilihat pada Tabel 3.19. Rancangan dapat dilihat pada Gambar 3.42.

Tabel 3.19. Atribut Antarmuka Menambah Data Kelas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etInputNama GroupsMembers	<i>EditText</i>	<i>Input</i> untuk memasukkan nama mahasiswa	<i>String</i>
2	etInputNRPGroupsMembers	<i>EditText</i>	<i>Input</i> untuk memasukkan NRP mahasiswa	<i>String</i>
3	btnTambah PesertaBaru	<i>Button</i>	Tombol untuk menambah peserta baru	<i>String</i>
4	ivEkelasDeleteContent	<i>ImageView</i>	Tombol aksi untuk menghapus peserta dari kelas	<i>ImageClick</i>



The image shows a mobile application interface on a smartphone. The top status bar displays the time as 15:45. Below the status bar is a blue header with a back arrow on the left, the text "data e-Kelas" in the center, and a three-dot menu icon on the right. The main content area is white and contains two text input fields. The first field is labeled "NAMA PESERTA" and the second is labeled "NRP". At the bottom of the screen, there is a teal button with the text "Publish".

Gambar 3.42. Rancangan Halaman Antarmuka Menambah Data Kelas

3.2.4.7 Rancangan Halaman Antarmuka Menghapus Data Kelas

Halaman ini akan ditampilkan ketika aktor menekan pilihan hapus data kelas di samping *item* data kelas pada halaman antarmuka data kelas. Halaman ini akan memuat pilihan untuk menghapus data. Atribut antarmuka dapat dilihat pada Tabel 3.20. Rancangan dapat dilihat pada Gambar 3.43.

Tabel 3.20. Atribut Antarmuka Menghapus Data Kelas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>dialog_delete</i>	<i>Dialog</i>	Informasi konfirmasi penghapusan diskusi	<i>String</i>
2	<i>btn_yes</i>	<i>Button</i>	Tombol aksi untuk menyetujui penghapusan diskusi yang dipilih	<i>Button click</i>
3	<i>btn_no</i>	<i>Button</i>	Tombol aksi untuk menolak penghapusan diskusi yang dipilih	<i>Button click</i>

Konfirmasi Hapus

Apakah anda ingin menghapus ini ?

TIDAK

YA

Gambar 3.43. Rancangan Halaman Antarmuka Menghapus Data Kelas

3.2.4.8 Rancangan Halaman Antarmuka Menambah Materi Perkuliahan

Halaman ini akan ditampilkan ketika dosen menekan pilihan tambah materi perkuliahan pada halaman antarmuka menambah materi perkuliahan. Halaman ini akan memuat judul dan deskripsi materi perkuliahan yang sedang di-*input*-kan oleh dosen. Atribut antarmuka dapat dilihat pada Tabel 3.21. Rancangan dapat dilihat pada Gambar 3.44.

Tabel 3.21. Atribut Antarmuka Menambah Materi Perkuliahan

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	<i>Input</i> untuk memasukkan judul materi	<i>String</i>
2	etMessage	<i>EditText</i>	<i>Input</i> untuk memasukkan deskripsi materi	<i>String</i>
3	btnTambah MateriBaru	<i>Button</i>	Tombol untuk menambah materi baru	<i>ButtonClick</i>
4	tvDate	<i>TextView</i>	Menampilkan tanggal pada rentang minggu perkuliahan	<i>String</i>

← EPL ⋮

Tanggal
25 Sept 18 - 2 Okt 18

Judul Materi

Deskripsi Materi

Publish

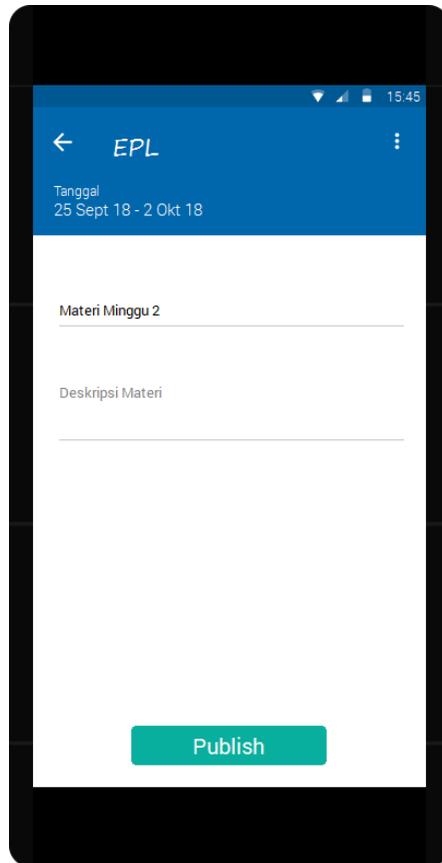
Gambar 3.44. Rancangan Halaman Antarmuka Menambah Materi Perkuliahan

3.2.4.9 Rancangan Halaman Antarmuka *Edit Materi Perkuliahan*

Halaman ini akan ditampilkan ketika aktor menekan pilihan *edit* materi perkuliahan di samping *item* modul/aktivitas pada halaman antarmuka halaman utama kelas. Halaman ini akan memuat pilihan untuk meng-*edit* materi perkuliahan. Atribut antarmuka dapat dilihat pada Tabel 3.21. Rancangan dapat dilihat pada Gambar 3.44.

Tabel 3.22. Atribut Antarmuka Edit Materi Perkuliahan

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	<i>Input</i> untuk memasukkan judul materi	<i>String</i>
2	etMessage	<i>EditText</i>	<i>Input</i> untuk memasukkan deskripsi materi	<i>String</i>
3	btnTambah MateriBaru	<i>Button</i>	Tombol untuk menambah materi baru	<i>Button Click</i>
4	tvDate	<i>TextView</i>	Menampilkan tanggal pada rentang minggu perkuliahan	<i>String</i>



Gambar 3.45. Rancangan Halaman Antarmuka Edit Materi Perkuliahan

3.2.4.10 Rancangan Halaman Antarmuka Menghapus Materi Perkuliahan

Halaman ini akan ditampilkan ketika aktor menekan pilihan hapus materi perkuliahan di samping *item* modul/aktivitas pada halaman antarmuka halaman utama kelas. Halaman ini akan memuat pilihan untuk menghapus materi perkuliahan. Atribut antarmuka dapat dilihat pada Tabel 3.23. Rancangan dapat dilihat pada Gambar 3.43.

Tabel 3.23. Atribut Antarmuka Menghapus Materi Perkuliahan

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>dialog_delete</i>	<i>Dialog</i>	Informasi konfirmasi penghapusan diskusi	<i>String</i>
2	<i>btn_yes</i>	<i>Button</i>	Tombol aksi untuk menyetujui penghapusan diskusi yang dipilih	<i>Button click</i>
3	<i>btn_no</i>	<i>Button</i>	Tombol aksi untuk menolak penghapusan diskusi yang dipilih	<i>Button click</i>

Konfirmasi Hapus

Apakah anda ingin menghapus ini ?

TIDAK

YA

Gambar 3.46. Rancangan Halaman Antarmuka Menghapus Materi Perkuliahan

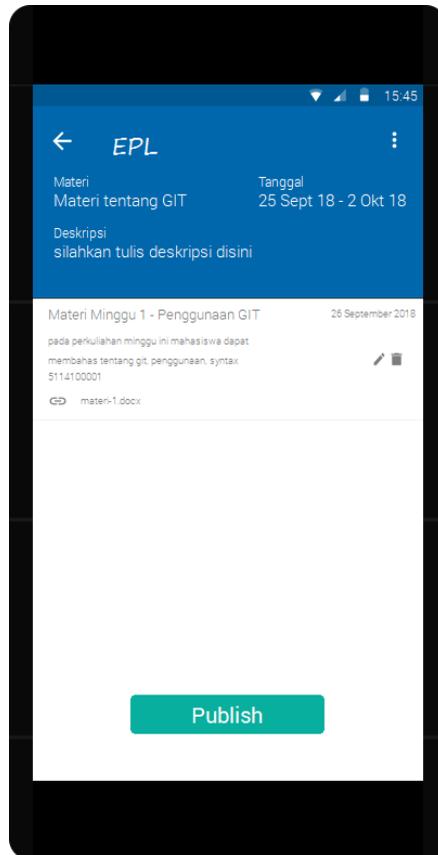
3.2.4.11 Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika aktor melihat materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat hasil *input* yang telah di-*input*-kan oleh dosen. Atribut antarmuka dapat dilihat pada Tabel 3.24. Rancangan dapat dilihat pada Gambar 3.47.

Tabel 3.24. Atribut Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	tvGlossary Description	<i>TextView</i>	Menampilkan deskripsi materi perkuliahan	<i>String</i>
2	tvGlossary Title	<i>TextView</i>	Menampilkan judul materi perkuliahan	<i>String</i>
3	tvGlossary Date	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	labelTitle	<i>TextView</i>	Menampilkan judul materi	<i>String</i>
5	labelDate	<i>TextView</i>	Menampilkan tanggal <i>post</i> detail materi perkuliahan	<i>String</i>
6	labelMessage	<i>TextView</i>	Menampilkan detail materi perkuliahan	<i>String</i>
7	labelUser	<i>TextView</i>	Menampilkan <i>username/</i>	<i>String</i>

			NRP aktor yang mem- <i>posting</i> detail materi perkuliahan	
8	labelNama File	<i>TextView</i>	Menampilkan nama <i>file</i> materi perkuliahan	<i>String</i>
9	ivAttachment File	<i>ImageView</i>	Tombol untuk melakukan unduh <i>file</i> materi perkuliahan	<i>Image Click</i>
10	ivEntriesEdit Content	<i>ImageView</i>	Tombol untuk meng- <i>edit</i> detail materi perkuliahan	<i>Image Click</i>
11	ivEntriesDeleteContent	<i>ImageView</i>	Tombol untuk menghapus detail materi perkuliahan	<i>Image Click</i>
12	btnTambah Materi	<i>Button</i>	Tombol untuk menuju ke <i>form</i> tambah detail materi perkuliahan	<i>Button Click</i>



Gambar 3.47. Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen

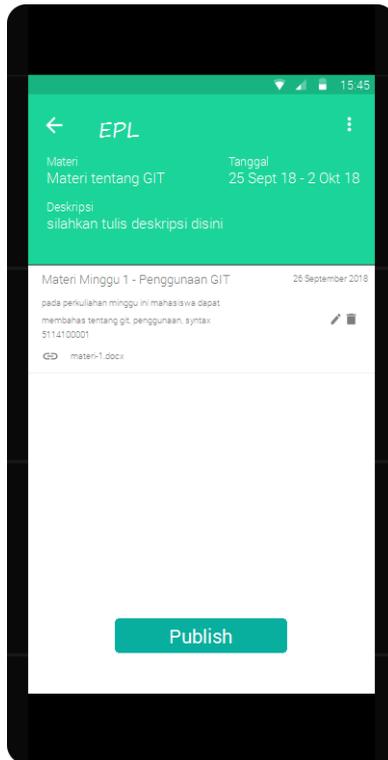
3.2.4.12 Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika aktor melihat materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat hasil *input* yang telah di-*input*-kan oleh dosen. Atribut antarmuka dapat dilihat pada Tabel 3.25. Rancangan dapat dilihat pada Gambar 3.48.

Tabel 3.25. Atribut Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	tvGlossary Description	<i>TextView</i>	Menampilkan deskripsi materi perkuliahan	<i>String</i>
2	tvGlossary Title	<i>TextView</i>	Menampilkan judul materi perkuliahan	<i>String</i>
3	tvGlossary Date	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	labelTitle	<i>TextView</i>	Menampilkan judul materi	<i>String</i>
5	labelDate	<i>TextView</i>	Menampilkan tanggal <i>post</i> detail materi perkuliahan	<i>String</i>
6	labelMessage	<i>TextView</i>	Menampilkan detail materi perkuliahan	<i>String</i>
7	labelUser	<i>TextView</i>	Menampilkan <i>username/NRP</i> aktor yang <i>posting</i> detail materi perkuliahan	<i>String</i>
8	labelNama File	<i>TextView</i>	Menampilkan nama <i>file</i> materi perkuliahan	<i>String</i>
9	ivAttachmentFile	<i>ImageView</i>	Tombol untuk melakukan unduh <i>file</i> materi perkuliahan	<i>ImageClick</i>
10	btnTambah Materi	<i>Button</i>	Tombol untuk menuju ke <i>form</i>	<i>ButtonClick</i>

			tambah detail materi perkuliahan	
--	--	--	----------------------------------	--



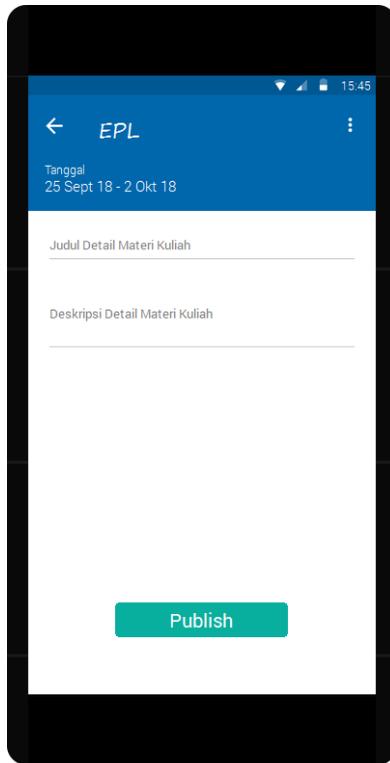
Gambar 3.48. Rancangan Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

3.2.4.13 Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika aktor menekan pilihan tambah detail materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat judul dan deskripsi detail materi perkuliahan yang sedang di-*input*-kan oleh aktor. Atribut antarmuka dapat dilihat pada Tabel 3.26. Rancangan dapat dilihat pada Gambar 3.49.

Tabel 3.26. Atribut Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	Input untuk memasukkan judul detail materi perkuliahan	<i>String</i>
2	etMessage	<i>EditText</i>	Input untuk memasukkan deskripsi detail materi perkuliahan	<i>String</i>
3	tvdate	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	btnTambah Materi	<i>Button</i>	Tombol untuk menambah detail materi perkuliahan baru	<i>ButtonClick</i>



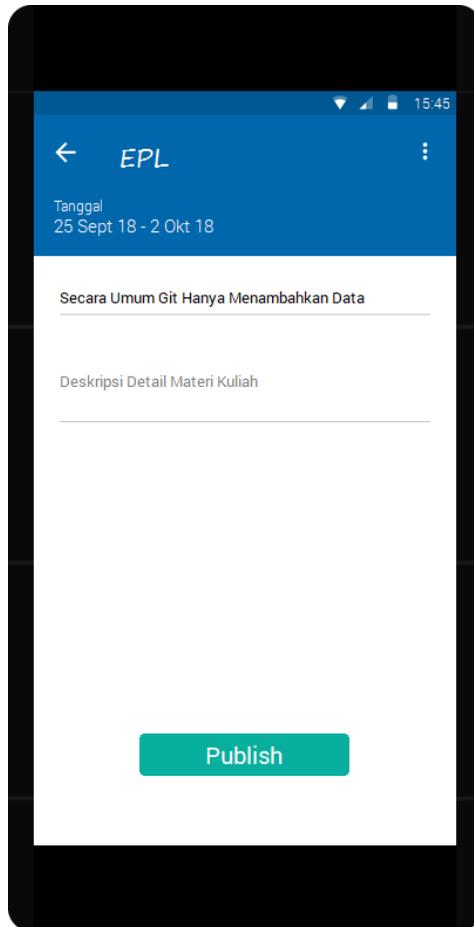
Gambar 3.49. Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

3.2.4.14 Rancangan Halaman Antarmuka *Edit* Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika aktor menekan pilihan *edit* detail materi perkuliahan di samping *item* materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat judul dan deskripsi detail materi perkuliahan yang sedang di-*input*-kan oleh aktor. Atribut antarmuka dapat dilihat pada Tabel 3.27. Rancangan dapat dilihat pada Gambar 3.50.

Tabel 3.27. Atribut Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	Input untuk memasukkan judul detail materi perkuliahan	<i>String</i>
2	etMessage	<i>EditText</i>	Input untuk memasukkan deskripsi detail materi perkuliahan	<i>String</i>
3	tvdate	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	btnTambah Materi	<i>Button</i>	Tombol untuk menambah detail materi perkuliahan baru	<i>ButtonClick</i>



Gambar 3.50. Rancangan Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

3.2.4.15 Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika aktor menekan pilihan hapus materi perkuliahan di samping *item* detail materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat pilihan untuk menghapus detail materi

perkuliahan. Atribut antarmuka dapat dilihat pada Tabel 3.28. Rancangan dapat dilihat pada Gambar 3.51.

Tabel 3.28. Atribut Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>dialog_delete</i>	<i>Dialog</i>	Informasi konfirmasi penghapusan diskusi	<i>String</i>
2	<i>btn_yes</i>	<i>Button</i>	Tombol aksi untuk menyetujui penghapusan diskusi yang dipilih	<i>Button click</i>
3	<i>btn_no</i>	<i>Button</i>	Tombol aksi untuk menolak penghapusan diskusi yang dipilih	<i>Button click</i>

Konfirmasi Hapus

Apakah anda ingin menghapus ini ?

TIDAK

YA

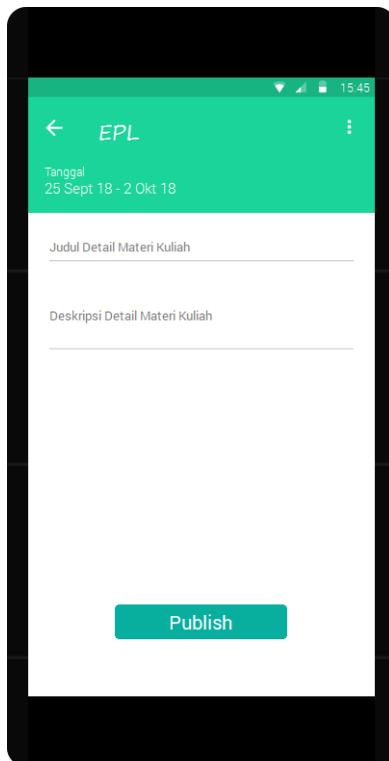
Gambar 3.51. Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

3.2.4.16 Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika aktor menekan pilihan tambah detail materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat judul dan deskripsi detail materi perkuliahan yang sedang di-*input*-kan oleh aktor. Atribut antarmuka dapat dilihat pada Tabel 3.29. Rancangan dapat dilihat pada Gambar 3.52.

Tabel 3.29. Atribut Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	Input untuk memasukkan judul detail materi perkuliahan	<i>String</i>
2	etMessage	<i>EditText</i>	Input untuk memasukkan deskripsi detail materi perkuliahan	<i>String</i>
3	tvdate	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	btnTambah Materi	<i>Button</i>	Tombol untuk menambah detail materi perkuliahan baru	<i>ButtonClick</i>



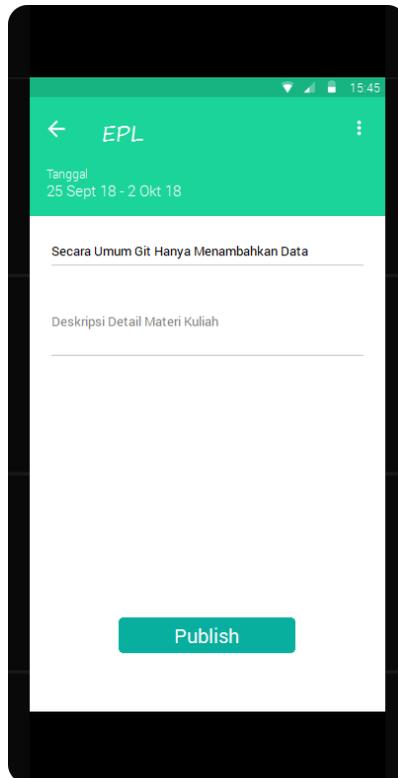
Gambar 3.52. Rancangan Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

3.2.4.17 Rancangan Halaman Antarmuka *Edit* Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika aktor menekan pilihan *edit* detail materi perkuliahan di samping *item* materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat judul dan deskripsi detail materi perkuliahan yang sedang di-*input*-kan oleh aktor. Atribut antarmuka dapat dilihat pada Tabel 3.30. Rancangan dapat dilihat pada Gambar 3.53.

Tabel 3.30. Atribut Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	Input untuk memasukkan judul detail materi perkuliahan	<i>String</i>
2	etMessage	<i>EditText</i>	Input untuk memasukkan deskripsi detail materi perkuliahan	<i>String</i>
3	tvdate	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	btnTambah Materi	<i>Button</i>	Tombol untuk menambah detail materi perkuliahan baru	<i>ButtonClick</i>



Gambar 3.53. Rancangan Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

3.2.4.18 Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika aktor menekan pilihan hapus materi perkuliahan di samping *item* detail materi perkuliahan pada halaman antarmuka materi perkuliahan. Halaman ini akan memuat pilihan untuk menghapus detail materi perkuliahan. Atribut antarmuka dapat dilihat pada Tabel 3.31. Rancangan dapat dilihat pada Gambar 3.54.

Tabel 3.31. Atribut Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>dialog_delete</i>	<i>Dialog</i>	Informasi konfirmasi penghapusan diskusi	<i>String</i>
2	<i>btn_yes</i>	<i>Button</i>	Tombol aksi untuk menyetujui penghapusan diskusi yang dipilih	<i>Button click</i>
3	<i>btn_no</i>	<i>Button</i>	Tombol aksi untuk menolak penghapusan diskusi yang dipilih	<i>Button click</i>

Konfirmasi Hapus

Apakah anda ingin menghapus ini ?

TIDAK

YA

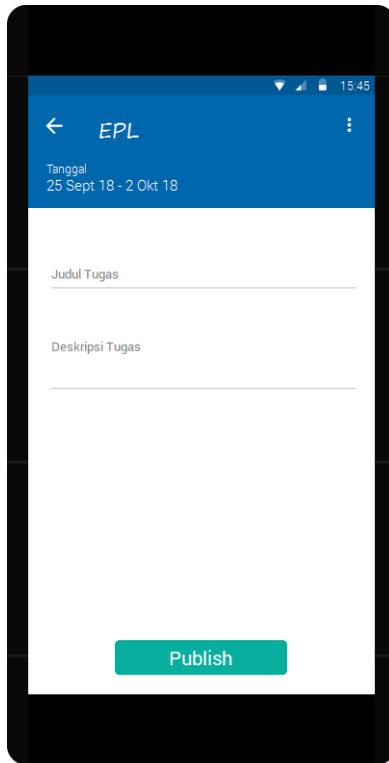
Gambar 3.54. Rancangan Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

3.2.4.19 Rancangan Halaman Antarmuka Menambah Tugas

Halaman ini akan ditampilkan ketika dosen menekan pilihan tambah tugas pada halaman antarmuka menambah tugas. Halaman ini akan memuat judul dan deskripsi tugas yang sedang di-*input*-kan oleh dosen. Atribut antarmuka dapat dilihat pada Tabel 3.32. Rancangan dapat dilihat pada Gambar 3.55.

Tabel 3.32. Atribut Antarmuka Menambah Tugas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	Input untuk memasukkan judul tugas	<i>String</i>
2	etMessage	<i>EditText</i>	Input untuk memasukkan deskripsi tugas	<i>String</i>
3	tvdate	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	btnTambahTugasBaru	<i>Button</i>	Tombol untuk menambah tugas baru	<i>ButtonClick</i>



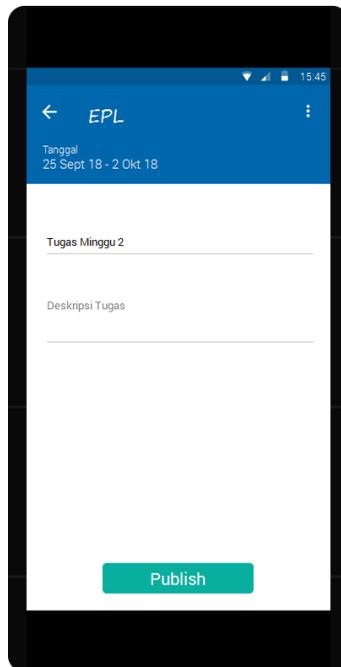
Gambar 3.55. Rancangan Halaman Antarmuka Menambah Tugas

3.2.4.20 Rancangan Halaman Antarmuka *Edit* Tugas

Halaman ini akan ditampilkan ketika dosen menekan pilihan tambah tugas pada halaman antarmuka menambah tugas. Halaman ini akan memuat judul dan deskripsi tugas yang sedang di-*input*-kan oleh dosen. Atribut antarmuka dapat dilihat pada Tabel 3.33. Rancangan dapat dilihat pada Gambar 3.56.

Tabel 3.33. Atribut Antarmuka Edit Tugas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etTitle	<i>EditText</i>	Input untuk memasukkan judul tugas	<i>String</i>
2	etMessage	<i>EditText</i>	Input untuk memasukkan deskripsi tugas	<i>String</i>
3	tvdate	<i>TextView</i>	Menampilkan rentang tanggal di minggu perkuliahan	<i>String</i>
4	btnTambah TugasBaru	<i>Button</i>	Tombol untuk menambah tugas baru	<i>ButtonClick</i>



Gambar 3.56. Rancangan Halaman Antarmuka Edit Tugas

3.2.4.21 Rancangan Halaman Antarmuka Menghapus Tugas

Halaman ini akan ditampilkan ketika aktor menekan pilihan hapus tugas di samping *item* modul/aktivitas pada halaman antarmuka halaman utama kelas. Halaman ini akan memuat pilihan untuk menghapus tugas. Atribut antarmuka dapat dilihat pada Tabel 3.34. Rancangan dapat dilihat pada Gambar 3.57.

Tabel 3.34. Atribut Antarmuka Menghapus Tugas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	<i>dialog_delete</i>	<i>Dialog</i>	Informasi konfirmasi penghapusan diskusi	<i>String</i>
2	<i>btn_yes</i>	<i>Button</i>	Tombol aksi untuk menyetujui penghapusan diskusi yang dipilih	<i>Button click</i>
3	<i>btn_no</i>	<i>Button</i>	Tombol aksi untuk menolak penghapusan diskusi yang dipilih	<i>Button click</i>

Konfirmasi Hapus

Apakah anda ingin menghapus ini ?

TIDAK

YA

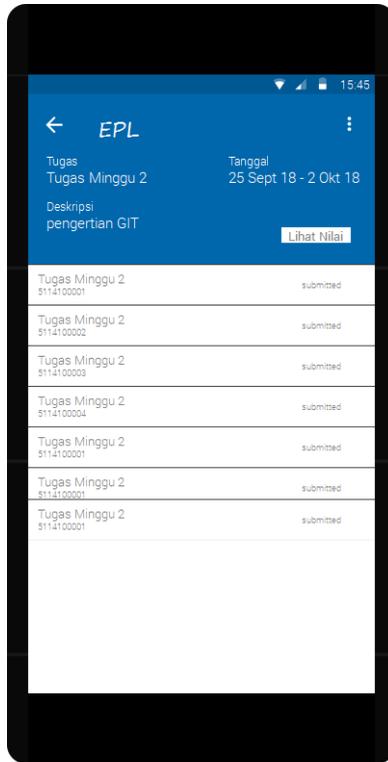
Gambar 3.57. Rancangan Halaman Antarmuka Menghapus Tugas

3.2.4.22 Rancangan Halaman Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika dosen melihat *list* pengumpulan tugas pada halaman antarmuka melihat *list* pengumpulan tugas. Halaman ini akan memuat hasil pencatatan tugas yang telah dikumpulkan oleh mahasiswa. Atribut antarmuka dapat dilihat pada Tabel 3.35. Rancangan dapat dilihat pada Gambar 3.58.

Tabel 3.35. Atribut Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	tvAssignTitle	<i>TextView</i>	Menampilkan judul tugas	<i>String</i>
2	tvAssignDate	<i>TextView</i>	Menampilkan tanggal <i>deadline</i>	<i>String</i>
3	tvAssign Description	<i>TextView</i>	Menampilkan judul tugas	<i>String</i>
4	ivLihatNilai PerTugas	<i>ImageView</i>	Tombol menuju ke halaman melihat nilai tugas	<i>ImageClick</i>
5	labelName	<i>TextView</i>	Menampilkan nama tugas yang dikumpulkan	<i>String</i>
6	labelStatus	<i>TextView</i>	Menampilkan status pengumpulan	<i>String</i>
7	labelUser	<i>TextView</i>	Menampilkan <i>username</i> /NR P aktor yang mengumpulka n	<i>String</i>



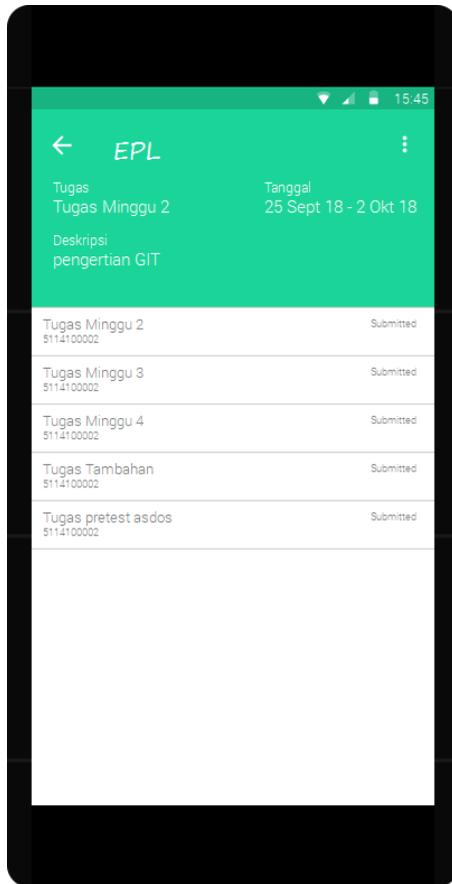
Gambar 3.58. Rancangan Halaman Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Dosen

3.2.4.23 Rancangan Halaman Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika asisten dosen melihat *list* pengumpulan tugas pada halaman antarmuka melihat *list* pengumpulan tugas. Halaman ini akan memuat hasil pencatatan tugas yang telah dikumpulkan oleh mahasiswa. Atribut antarmuka dapat dilihat pada Tabel 3.36. Rancangan dapat dilihat pada Gambar 3.59.

Tabel 3.36. Atribut Antarmuka Melihat List Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	tvAssignTitle	<i>TextView</i>	Menampilkan judul tugas	<i>String</i>
2	tvAssignDate	<i>TextView</i>	Menampilkan tanggal <i>deadline</i>	<i>String</i>
3	tvAssign Description	<i>TextView</i>	Menampilkan judul tugas	<i>String</i>
4	ivLihatNilai PerTugas	<i>ImageView</i>	Tombol menuju ke halaman melihat nilai tugas	<i>Image Click</i>
5	labelName	<i>TextView</i>	Menampilkan nama tugas yang dikumpulkan	<i>String</i>
6	labelStatus	<i>TextView</i>	Menampilkan status pengumpulan	<i>String</i>
7	labelUser	<i>TextView</i>	Menampilkan <i>username/NRP</i> aktor yang mengumpulkan	<i>String</i>



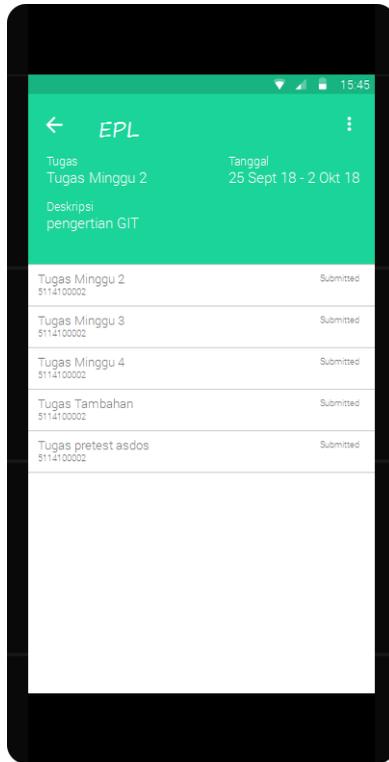
Gambar 3.59. Rancangan Halaman Antarmuka Melihat List Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa

3.2.4.24 Rancangan Halaman Antarmuka Melihat Detail Tugas

Halaman ini akan ditampilkan ketika mahasiswa menekan pilihan tugas pada halaman antarmuka halaman utama kelas. Halaman ini akan memuat detail tugas yang harus dikerjakan oleh mahasiswa serta *list* tugas yang telah dikumpulkan oleh mahasiswa pada kelas tersebut. Atribut antarmuka dapat dilihat pada Tabel 3.37. Rancangan dapat dilihat pada Gambar 3.60.

Tabel 3.37. Atribut Antarmuka Melihat Detail Tugas

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	tvAssignTitle	<i>TextView</i>	Menampilkan judul tugas	<i>String</i>
2	tvAssignDate	<i>TextView</i>	Menampilkan tanggal <i>deadline</i>	<i>String</i>
3	tvAssign Description	<i>TextView</i>	Menampilkan judul tugas	<i>String</i>
4	labelName	<i>TextView</i>	Menampilkan nama tugas yang dikumpulkan	<i>String</i>
5	labelStatus	<i>TextView</i>	Menampilkan status pengumpulan	<i>String</i>
6	labelUser	<i>TextView</i>	Menampilkan <i>username</i> /NRP aktor	<i>String</i>



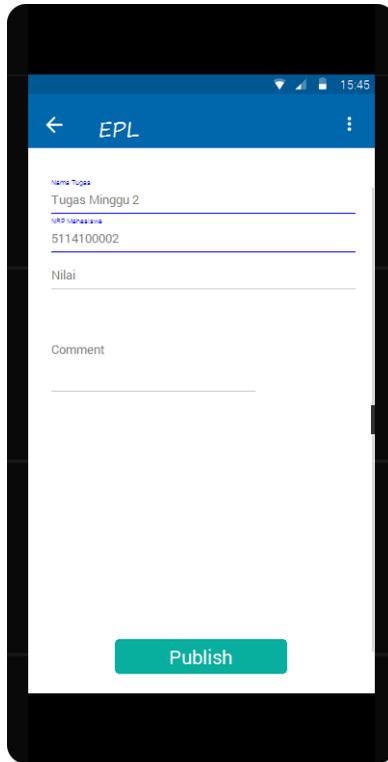
Gambar 3.60. Rancangan Halaman Antarmuka Melihat Detail Tugas

3.2.4.25 Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika dosen menekan pilihan tugas dari *list* tugas pada halaman antarmuka *list* pengumpulan tugas. Halaman ini akan memuat hasil penilaian yang digunakan dosen dan asisten dosen untuk menilai masing-masing tugas yang ditekan. Atribut antarmuka dapat dilihat pada Tabel 3.38. Rancangan dapat dilihat pada Gambar 3.61.

Tabel 3.38. Atribut Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etMark	<i>EditText</i>	<i>Input</i> untuk memasukkan nilai tugas	<i>String</i>
2	etTitle	<i>EditText</i>	Menampilkan nama tugas yang akan dinilai yang didapat dari masing-masing tugas yang diklik	<i>String</i>
3	etName	<i>EditText</i>	Menampilkan NRP mahasiswa yang akan dinilai yang didapat dari masing-masing tugas yang diklik	<i>String</i>
4	etComment	<i>EditText</i>	<i>Input</i> untuk memasukkan komentar nilai tugas	<i>String</i>
5	btnTambah NilaiBaru	<i>Button</i>	Tombol untuk menambahkan nilai tugas baru	<i>Button Click</i>



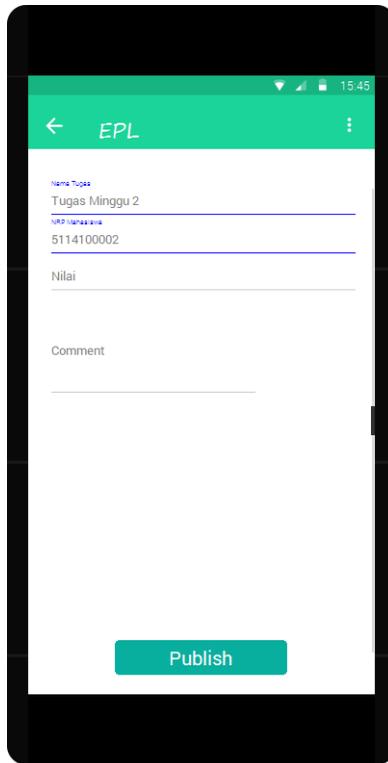
Gambar 3.61. Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen

3.2.4.26 Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika asisten dosen menekan pilihan tugas dari *list* tugas pada halaman antarmuka *list* pengumpulan tugas. Halaman ini akan memuat hasil penilaian yang digunakan dosen dan asisten dosen untuk menilai masing-masing tugas yang ditekan. Atribut antarmuka dapat dilihat pada Tabel 3.39. Rancangan dapat dilihat pada Gambar 3.62.

Tabel 3.39. Atribut Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan/ Keluaran
1	etMark	<i>EditText</i>	<i>Input</i> untuk memasukkan nilai tugas	<i>String</i>
2	etTitle	<i>EditText</i>	Menampilkan nama tugas yang akan dinilai yang didapat dari masing-masing tugas yang diklik	<i>String</i>
3	etName	<i>EditText</i>	Menampilkan NRP mahasiswa yang akan dinilai yang didapat dari masing-masing tugas yang diklik	<i>String</i>
4	etComment	<i>EditText</i>	<i>Input</i> untuk memasukkan komentar nilai tugas	<i>String</i>
5	btnTambah NilaiBaru	<i>Button</i>	Tombol untuk menambahkan nilai tugas baru	<i>ButtonClick</i>



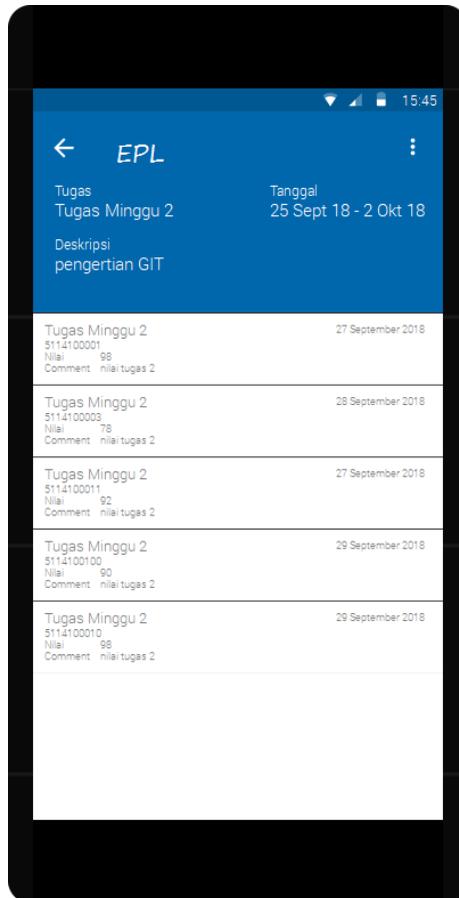
Gambar 3.62. Rancangan Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa

3.2.4.27 Rancangan Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Dosen

Halaman ini akan ditampilkan ketika dosen menekan pilihan lihat nilai tugas pada halaman antarmuka melihat *list* nilai tugas. Halaman ini akan memuat hasil pencatatan nilai tugas yang telah di-*input*-kan oleh dosen dan asisten dosen. Atribut antarmuka dapat dilihat pada Tabel 3.40. Rancangan dapat dilihat pada Gambar 3.63.

Tabel 3.40. Atribut Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Dosen

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan / Keluaran
1	tvAssignTitle	<i>TextView</i>	Menampilkan Nama Tugas	<i>String</i>
2	tvAssign Description	<i>TextView</i>	Menampilkan Deskripsi Tugas	<i>String</i>
3	tvAssignDate	<i>TextView</i>	Menampilkan rentang tanggal tugas di minggu perkuliahan	<i>String</i>
4	labelUser	<i>TextView</i>	Menampilkan <i>username/NRP</i> yang dinilai	<i>String</i>
5	labelTugas	<i>TextView</i>	Menampilkan Nama Tugas yang dinilai	<i>String</i>
6	labelDate	<i>TextView</i>	Menampilkan tanggal penilaian tugas	<i>String</i>
7	labelComment	<i>TextView</i>	Menampilkan komentar nilai tugas	<i>String</i>
8	labelNilai	<i>TextView</i>	Menampilkan nilai tugas	<i>String</i>



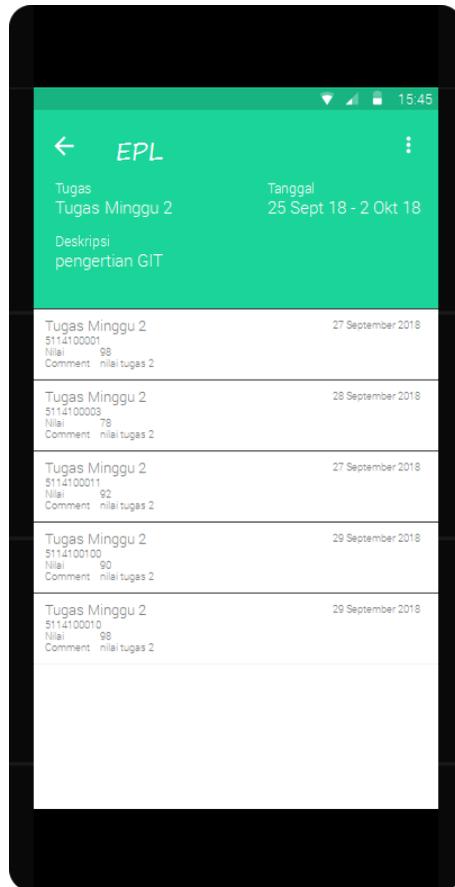
Gambar 3.63. Rancangan Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Dosen

3.2.4.28 Rancangan Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Mahasiswa

Halaman ini akan ditampilkan ketika asisten dosen menekan pilihan lihat nilai tugas pada halaman antarmuka melihat *list* nilai tugas. Halaman ini akan memuat hasil pencatatan nilai tugas yang telah di-*input*-kan oleh dosen dan asisten dosen. Atribut antarmuka dapat dilihat pada Tabel 3.41. Rancangan dapat dilihat pada Gambar 3.64.

Tabel 3.41. Atribut Antarmuka Melihat List Nilai Tugas pada Aplikasi MyITS Mahasiswa

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan	Masukan / Keluaran
1	tvAssignTitle	<i>TextView</i>	Menampilkan Nama Tugas	<i>String</i>
2	tvAssign Description	<i>TextView</i>	Menampilkan Deskripsi Tugas	<i>String</i>
3	tvAssignDate	<i>TextView</i>	Menampilkan rentang tanggal tugas di minggu perkuliahan	<i>String</i>
4	labelUser	<i>TextView</i>	Menampilkan <i>username</i> /NRP yang dinilai	<i>String</i>
5	labelTugas	<i>TextView</i>	Menampilkan Nama Tugas yang dinilai	<i>String</i>
6	labelDate	<i>TextView</i>	Menampilkan tanggal penilaian tugas	<i>String</i>
7	labelComment	<i>TextView</i>	Menampilkan komentar nilai tugas	<i>String</i>
8	labelNilai	<i>TextView</i>	Menampilkan nilai tugas	<i>String</i>



Gambar 3.64. Rancangan Halaman Antarmuka Melihat List Nilai Tugas pada Aplikasi MyITS Mahasiswa

BAB IV IMPLEMENTASI

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam aplikasi dan disertai dengan kode sumber masing-masing fungsi utama.

4.1 Lingkungan Implementasi Perangkat Lunak

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak ini dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat Keras	Komputer	Asus ROG GL553VD-FY380
	Prosesor	Intel® Core™ i7-7700HQ CPU @ 2.80GHz (2.81GHz)
	Memori Primer	16 GB
	Memori Sekunder	128GB SSD, 1TB HDD
Perangkat Lunak	Sistem Operasi	<i>Windows 10 Pro 64-bit</i>
	Perangkat Lunak	Android Studio v3.1.1, Justinmind Prototyper 7.1.1, StarUML v2.8.1, Sybase PowerDesigner 16.5, Microsoft Word 2016, Eclipse Oxygen.1 2017 v4.7.1, WSO2 Application Manager, MySQL.

4.2 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna pada fitur kuis dan diskusi pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa ini menggunakan jenis file XML yang dibangun dan diimplementasikan dalam lingkungan kerja Android Studio.

Penjelasan implementasi antarmuka aplikasi ditunjukkan melalui tampilan halaman XML yang sudah di-*render* oleh Android Studio berikut sesuai dengan rancangan antarmuka pada bab III tugas akhir ini.

4.2.1 Implementasi Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Dosen

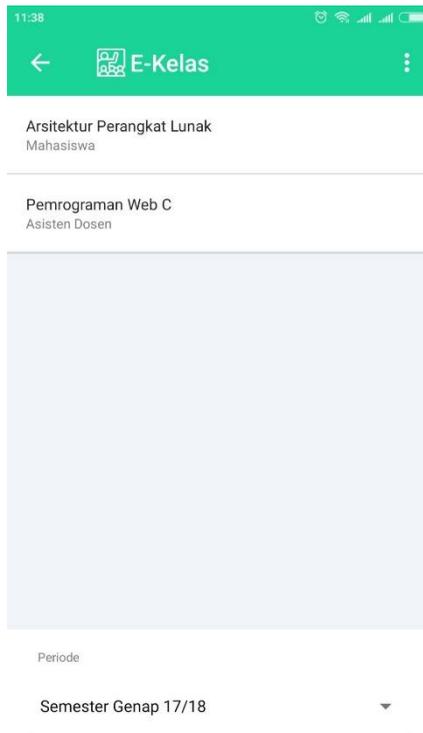
Halaman antarmuka melihat *list* kelas pada aplikasi MyITS dosen adalah halaman yang hanya dapat diakses oleh dosen untuk melihat *list* kelas yang telah dan sedang diampu. Halaman ini berisi *list* nama kelas dan *filter* berdasarkan semester. Hasil implementasi dapat dilihat pada Gambar 4.1.



Gambar 4.1. Implementasi Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Dosen

4.2.2 Implementasi Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Mahasiswa

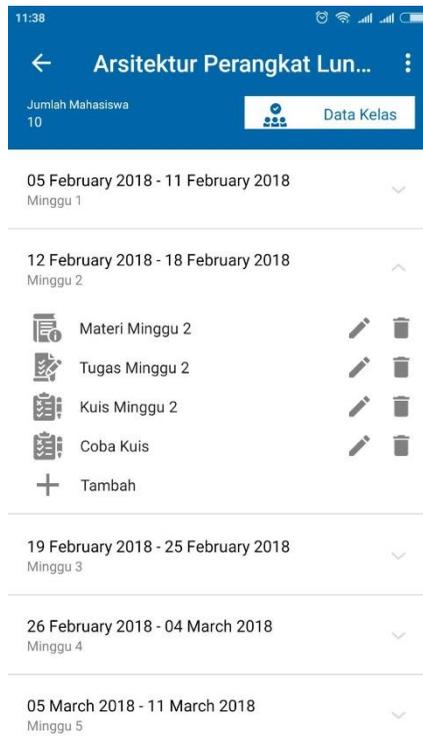
Halaman antarmuka melihat *list* kelas pada aplikasi MyITS mahasiswa adalah halaman yang hanya dapat diakses oleh asisten dosen dan mahasiswa untuk melihat *list* kelas yang telah dan sedang diikuti. Halaman ini berisi *list* nama kelas dan *filter* berdasarkan semester. Hasil implementasi dapat dilihat pada Gambar 4.2.



Gambar 4.2. Implementasi Halaman Antarmuka Melihat *List* Kelas pada Aplikasi MyITS Mahasiswa

4.2.3 Implementasi Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Dosen

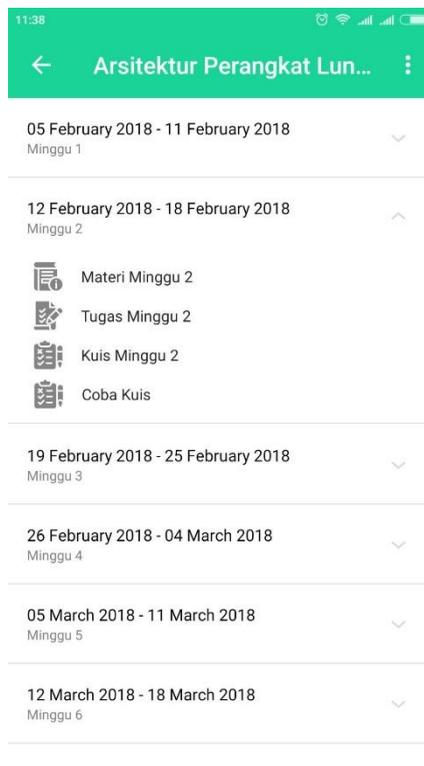
Halaman antarmuka melihat halaman utama kelas pada aplikasi MyITS dosen adalah halaman yang hanya diakses oleh dosen untuk melakukan segala aktivitas di kelas. Halaman ini berisi minggu perkuliahan dengan isi modul/ aktivitas untuk berpindah ke masing-masing aktivitas. Hasil implementasi dapat dilihat pada Gambar 4.3.



Gambar 4.3. Implementasi Antarmuka Halaman Utama Kelas pada Aplikasi MyITS Dosen

4.2.4 Implementasi Halaman Antarmuka Melihat Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa

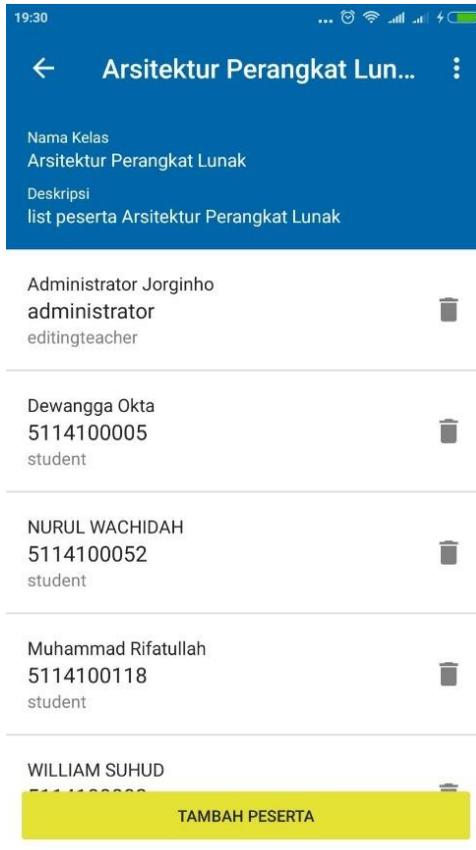
Halaman antarmuka melihat halaman utama kelas pada aplikasi MyITS mahasiswa adalah halaman yang hanya diakses oleh asisten dosen dan mahasiswa untuk melakukan segala aktivitas di kelas. Halaman ini berisi minggu perkuliahan dengan isi modul/ aktivitas untuk berpindah ke masing-masing aktivitas. Hasil implementasi dapat dilihat pada Gambar 4.4.



Gambar 4.4. Implementasi Antarmuka Halaman Utama Kelas pada Aplikasi MyITS Mahasiswa

4.2.5 Implementasi Halaman Antarmuka Melihat Data Kelas

Halaman antarmuka melihat data kelas adalah halaman yang hanya dapat diakses oleh dosen untuk melihat *list* nama peserta kelas pada kelas yang diampu. Hasil implementasi dapat dilihat pada Gambar 4.5.



Gambar 4.5. Implementasi Halaman Antarmuka Melihat Data Kelas

4.2.6 Implementasi Halaman Antarmuka Menambah Data Kelas

Halaman antarmuka menambah data kelas adalah halaman yang hanya dapat diakses oleh dosen untuk menambah peserta ke kelas. Halaman ini berisi *form* untuk menambah peserta baru. Hasil implementasi dapat dilihat pada Gambar 4.6.



20:21

← Nama Kelas

Nama Peserta

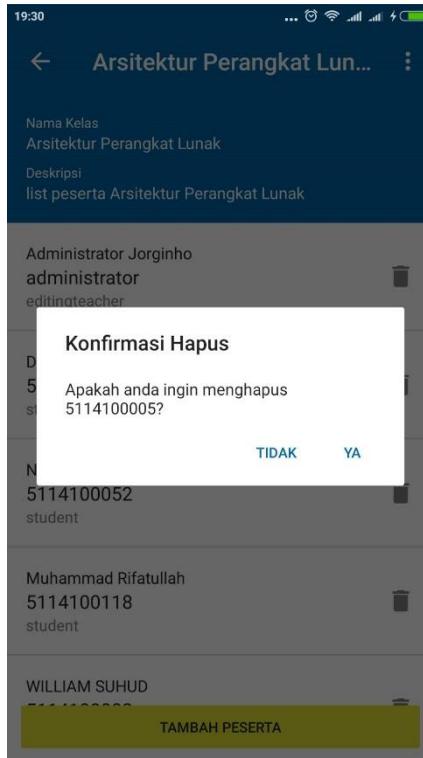
NRP Peserta

PUBLISH

Gambar 4.6. Implementasi Halaman Antarmuka Menambah Data Kelas

4.2.7 Implementasi Halaman Antarmuka Menghapus Data Kelas

Halaman antarmuka menghapus data kelas adalah halaman yang hanya dapat diakses oleh dosen untuk menghapus data peserta dari kelas. Halaman ini berisi dialog untuk konfirmasi hapus data peserta. Hasil implementasi dapat dilihat pada Gambar 4.7.



Gambar 4.7. Implementasi Halaman Antarmuka Menghapus Data Kelas

4.2.8 Implementasi Halaman Antarmuka Menambah Materi Perkuliahan

Halaman antarmuka menambah materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen untuk menambah materi perkuliahan. Halaman ini berisi *form* untuk menambah materi perkuliahan baru. Hasil implementasi dapat dilihat pada Gambar 4.8.



12:00

← Arsitektur Perangkat Lun... ⋮

Tanggal
12 February 2018 - 18 February 2018

Judul Materi Kuliah

Deskripsi Materi Kuliah

PUBLISH

Gambar 4.8. Implementasi Halaman Antarmuka Menambah Materi Perkuliahan

4.2.9 Implementasi Halaman *Edit Materi* Perkuliahan

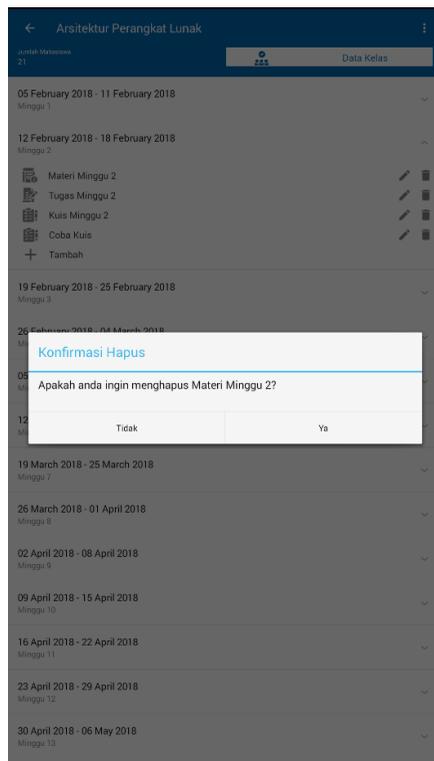
Halaman antarmuka *edit* materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen untuk meng-*edit* materi perkuliahan. Halaman ini berisi *form* untuk mengganti judul dan deskripsi materi perkuliahan, serta tombol “*publish*” untuk menampilkan materi perkuliahan yang baru. Hasil implementasi dapat dilihat pada Gambar 4.9.



Gambar 4.9. Implementasi Halaman *Edit Materi* Perkuliahan

4.2.10 Implementasi Halaman Antarmuka Menghapus Materi Perkuliahan

Halaman antarmuka hapus materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen untuk menghapus materi perkuliahan dari *list* aktivitas pada minggu perkuliahan. Halaman ini berisi dialog untuk konfirmasi hapus materi perkuliahan. Hasil implementasi dapat dilihat pada Gambar 4.10.



Gambar 4.10. Implementasi Halaman Antarmuka Menghapus Materi Perkuliahan

4.2.11 Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman antarmuka melihat materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen untuk mengelola detail materi perkuliahan. Halaman ini berisi detail materi perkuliahan, materi perkuliahan, tombol tambah materi perkuliahan, tombol *edit* dan tombol hapus. Hasil implementasi dapat dilihat pada Gambar 4.11.



Gambar 4.11. Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Dosen

4.2.12 Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman antarmuka melihat materi perkuliahan adalah halaman yang hanya dapat diakses oleh asisten dosen dan mahasiswa untuk mengelola detail materi perkuliahan. Halaman ini berisi detail materi perkuliahan, materi perkuliahan, tombol tambah materi perkuliahan, tombol *edit* dan tombol hapus. Hasil implementasi dapat dilihat pada Gambar 4.12.



Gambar 4.12. Implementasi Halaman Antarmuka Melihat Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

4.2.13 Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman antarmuka menambah detail materi perkuliahan adalah halaman yang dapat diakses oleh seluruh aktor untuk menambah detail materi perkuliahan. Halaman ini berisi *form* untuk menambah detail materi perkuliahan baru. Hasil implementasi dapat dilihat pada Gambar 4.13.



3:45

← Arsitektur Perangkat Lun... ⋮

Tanggal

Judul Detail Materi Kuliah

Deskripsi Detail Materi Kuliah

PUBLISH

Gambar 4.13. Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

4.2.14 Implementasi Halaman Antarmuka *Edit* Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

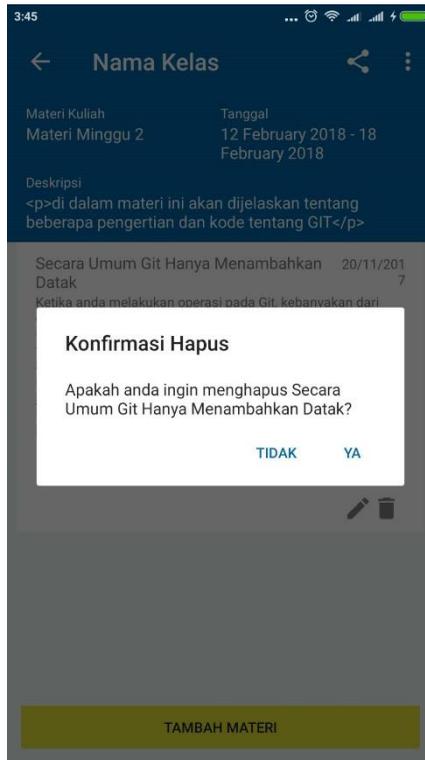
Halaman antarmuka *edit* detail materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk meng-*edit* detail materi perkuliahan. Halaman ini berisi *form* untuk mengubah detail materi perkuliahan. Hasil implementasi dapat dilihat pada Gambar 4.14.



Gambar 4.14. Implementasi Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Dosen.

4.2.15 Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

Halaman antarmuka hapus detail materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk menghapus detail materi perkuliahan. Halaman ini berisi dialog untuk konfirmasi hapus detail materi perkuliahan. Hasil implementasi dapat dilihat pada Gambar 4.15.



Gambar 4.15. Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Dosen

4.2.16 Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman antarmuka menambah detail materi perkuliahan adalah halaman yang dapat diakses oleh seluruh aktor untuk menambah detail materi perkuliahan. Halaman ini berisi *form* untuk menambah detail materi perkuliahan baru. Hasil implementasi dapat dilihat pada Gambar 4.16.



The screenshot shows a mobile application interface with a white background. At the top, there is a blue header bar with the time '3:43' on the left and icons for signal strength, Wi-Fi, and battery on the right. Below the header, the word 'Tanggal' is displayed in a light blue font. The main content area contains two text input fields. The first field is labeled 'Judul Detail Materi Kuliah' and has a light blue border. The second field is labeled 'Deskripsi Detail Materi Kuliah' and has a light gray border. At the bottom of the screen, there is a prominent red rectangular button with the word 'PUBLISH' centered in white capital letters.

Gambar 4.16. Implementasi Halaman Antarmuka Menambah Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

4.2.17 Implementasi Halaman Antarmuka *Edit* Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

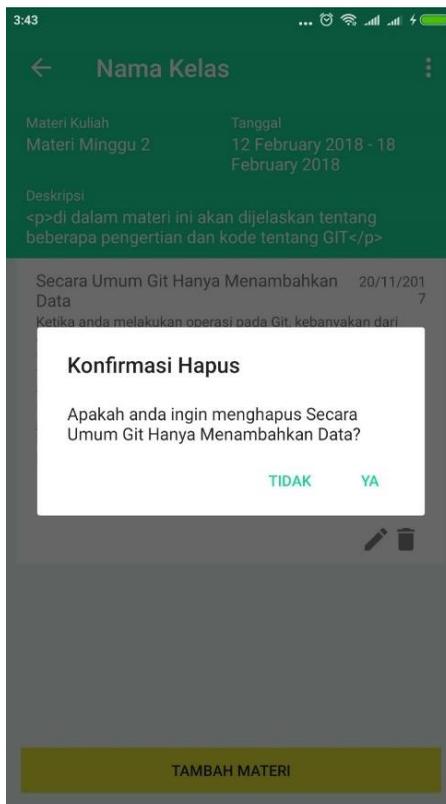
Halaman antarmuka *edit* detail materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk meng-*edit* detail materi perkuliahan. Halaman ini berisi *form* untuk mengubah detail materi perkuliahan. Hasil implementasi dapat dilihat pada Gambar 4.17.



Gambar 4.17. Implementasi Halaman Antarmuka Edit Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

4.2.18 Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

Halaman antarmuka hapus detail materi perkuliahan adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk menghapus detail materi perkuliahan. Halaman ini berisi dialog untuk konfirmasi hapus detail materi perkuliahan. Hasil implementasi dapat dilihat pada Gambar 4.18.



Gambar 4.18. Implementasi Halaman Antarmuka Menghapus Detail Materi Perkuliahan pada Aplikasi MyITS Mahasiswa

4.2.19 Implementasi Halaman Antarmuka Menambah Tugas

Halaman antarmuka menambah tugas adalah halaman yang hanya dapat diakses oleh dosen untuk menambah tugas. Halaman ini berisi *form* untuk menambahkan tugas baru. Hasil implementasi dapat dilihat pada Gambar 4.19.



12:01

← Nama Kelas

Tanggal
12 February 2018 - 18 February 2018

Nama Tugas

Deskripsi Tugas

SUBMIT

Gambar 4.19. Implementasi Halaman Antarmuka Menambah Tugas

4.2.20 Implementasi Halaman Antarmuka *Edit Tugas*

Halaman antarmuka *edit* tugas adalah halaman yang hanya dapat diakses oleh dosen untuk meng-*edit* tugas. Halaman ini berisi *form* untuk mengubah tugas. Hasil implementasi dapat dilihat pada Gambar 4.20.

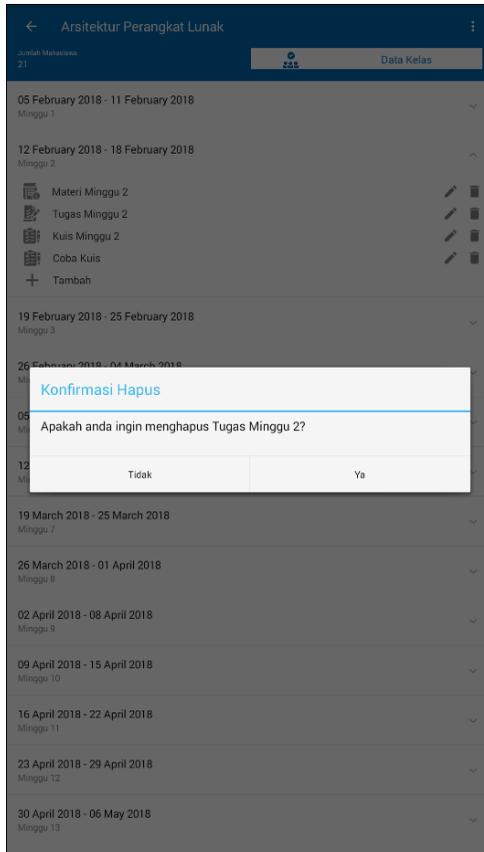


The screenshot displays a mobile application interface for editing a task. At the top, the status bar shows the time 12:01 and various system icons. Below this, a blue header bar contains a back arrow, the text "Nama Kelas", and a three-dot menu icon. The main content area has a white background and includes the following elements: a "Tanggal" (Date) field with the value "12 February 2018 - 18 February 2018"; a "Nama Tugas" (Task Name) field with the value "Tugas Minggu 2"; a "Deskripsi Tugas" (Task Description) field with the value "pengertian GIT"; and a prominent yellow button at the bottom labeled "EDIT TUGAS".

Gambar 4.20. Implementasi Halaman Antarmuka Edit Tugas

4.2.21 Implementasi Halaman Antarmuka Menghapus Tugas

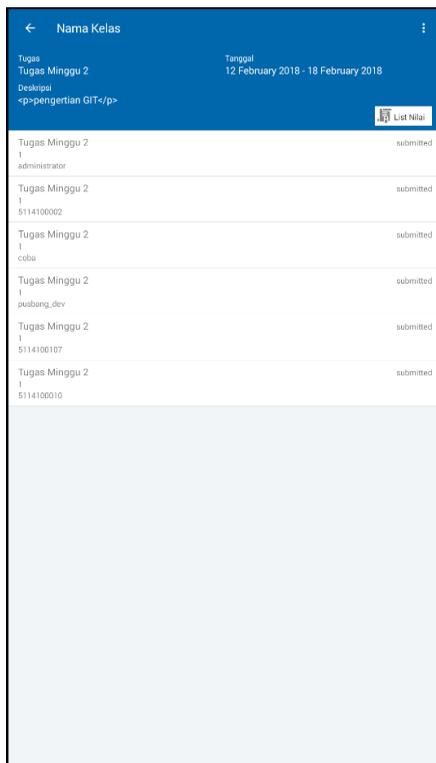
Halaman antarmuka menghapus tugas adalah halaman yang hanya dapat diakses oleh dosen untuk menghapus tugas. Halaman ini berisi dialog untuk konfirmasi hapus tugas. Hasil implementasi dapat dilihat pada Gambar 4.21.



Gambar 4.21. Implementasi Halaman Antarmuka Menghapus Tugas

4.2.22 Implementasi Halaman Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Dosen

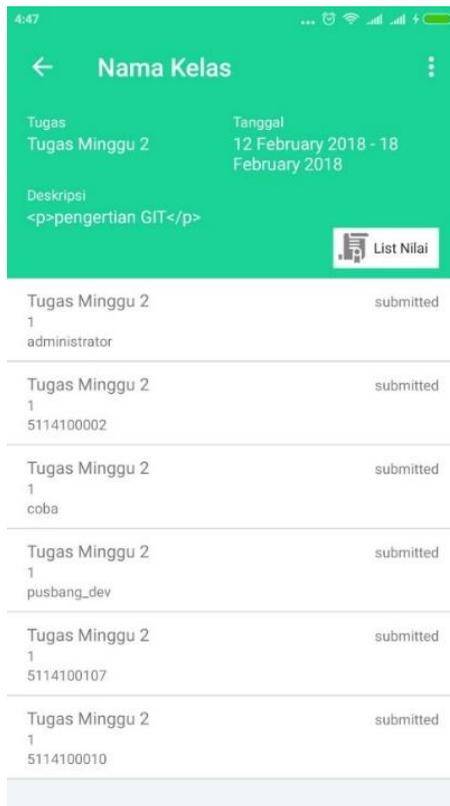
Halaman antarmuka melihat *list* pengumpulan tugas adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk melihat mahasiswa yang sudah mengumpulkan tugas. Halaman ini berisi *list* tugas yang telah dikumpulkan oleh mahasiswa. Hasil implementasi dapat dilihat pada Gambar 4.22.



Gambar 4.22. Implementasi Halaman Antarmuka Melihat List Pengumpulan Tugas pada Aplikasi MyITS Dosen

4.2.23 Implementasi Halaman Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa

Halaman antarmuka melihat *list* pengumpulan tugas adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk melihat mahasiswa yang sudah mengumpulkan tugas. Halaman ini berisi *list* tugas yang telah dikumpulkan oleh mahasiswa. Hasil implementasi dapat dilihat pada Gambar 4.23.



Gambar 4.23. Implementasi Halaman Antarmuka Melihat *List* Pengumpulan Tugas pada Aplikasi MyITS Mahasiswa

4.2.24 Implementasi Halaman Antarmuka Melihat Detail Tugas

Halaman antarmuka melihat detail tugas adalah halaman yang digunakan untuk melihat detail tugas yang harus dikerjakan oleh mahasiswa. Halaman ini berisi judul dan deskripsi tugas serta rentang waktu pengerjaan. Hasil implementasi dapat dilihat pada Gambar 4.24.



Gambar 4.24. Implementasi Halaman Antarmuka Melihat Detail Tugas

4.2.25 Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen

Halaman antarmuka menilai tugas adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk menilai tugas mahasiswa yang sudah dikumpulkan. Halaman ini berisi *form* untuk mengisi nilai dan komentar penilaian terhadap *item* tugas yang dipilih. Hasil implementasi dapat dilihat pada Gambar 4.25.



The screenshot shows a mobile application interface with a blue header bar. The header contains a back arrow, the text "Nama Kelas", and a three-dot menu icon. Below the header, the text "Nama Tugas" is followed by "Tugas Minggu 2". Underneath, "NRP Mahasiswa" is followed by "5114100107". The label "Nilai" is followed by a horizontal line. Below this is a "Comment" section with another horizontal line. At the bottom, there is a yellow button labeled "PUBLISH".

4:28

← Nama Kelas

Nama Tugas

Tugas Minggu 2

NRP Mahasiswa

5114100107

Nilai

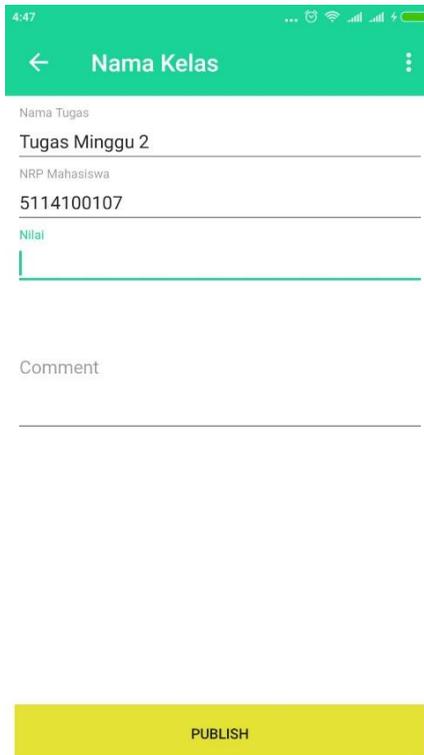
Comment

PUBLISH

Gambar 4.25. Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Dosen

4.2.26 Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa

Halaman antarmuka menilai tugas adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk menilai tugas mahasiswa yang sudah dikumpulkan. Halaman ini berisi *form* untuk mengisi nilai dan komentar penilaian terhadap *item* tugas yang dipilih. Hasil implementasi dapat dilihat pada Gambar 4.26.



The screenshot shows a mobile application interface for task evaluation. At the top, there is a green header bar with a back arrow, the text "Nama Kelas", and a menu icon. Below the header, the form contains the following fields:

- Nama Tugas:** Tugas Minggu 2
- NRP Mahasiswa:** 5114100107
- Nilai:** A vertical input field with a green border.
- Comment:** A text input field with a horizontal line below it.

At the bottom of the screen, there is a yellow button labeled "PUBLISH".

Gambar 4.26. Implementasi Halaman Antarmuka Menilai Tugas pada Aplikasi MyITS Mahasiswa

4.2.27 Implementasi Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Dosen

Halaman antarmuka melihat *list* nilai tugas adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk melihat nilai mahasiswa yang sudah dinilai. Halaman ini berisi *list* berupa nama tugas, *username*/NRP, nilai dan komentar penilaian. Hasil implementasi dapat dilihat pada Gambar 4.27.



Tugas	Tanggal
Tugas Minggu 2	15 Mei 2018
Deskripsi <p>pengertian GIT</p>	
Tugas Minggu 2 5114100002	22 Mei 2018
Nilai 98.00000	
Comment <p>nilai tugas 1</p>	
Tugas Minggu 2 198701032014041001	14 Mei 2018
Nilai 78.00000	
Comment menilai	
Tugas Minggu 2 5114100107	16 Mei 2018
Nilai 11.00000	
Comment <p>test 11</p>	
Tugas Minggu 2 pusbang_dev	22 Mei 2018
Nilai 80.00000	
Comment <p>telat kumpul minus 20</p>	

Gambar 4.27. Implementasi Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Dosen

4.2.28 Implementasi Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Mahasiswa

Halaman antarmuka melihat *list* nilai tugas adalah halaman yang hanya dapat diakses oleh dosen dan asisten dosen untuk melihat nilai mahasiswa yang sudah dinilai. Halaman ini berisi *list* berupa nama tugas, *username*/NRP, nilai dan komentar penilaian. Hasil implementasi dapat dilihat pada Gambar 4.28.



The screenshot shows a mobile application interface with a green header. The header contains a back arrow, the text "Nama Kelas", and a three-dot menu icon. Below the header, there is a section for the current assignment: "Tugas" (Tugas Minggu 2) and "Tanggal" (15 Mei 2018). A "Deskripsi" section follows, containing the text "<p>pengertian GIT</p>". Below this is a list of four assignment entries, each with a white background and a light blue border. Each entry displays the task name, a unique ID, the student's NRP, the score, and a comment.

Tugas	Tanggal
Tugas Minggu 2 5114100002 Nilai 98.00000 Comment <p>nilai tugas 1</p>	22 Mei 2018
Tugas Minggu 2 198701032014041001 Nilai 78.00000 Comment menilai	14 Mei 2018
Tugas Minggu 2 5114100107 Nilai 11.00000 Comment <p>test 11</p>	16 Mei 2018
Tugas Minggu 2 pusbang_dev Nilai 80.00000 Comment <p>telat kumpul minus 20</p>	22 Mei 2018

Gambar 4.28. Implementasi Halaman Antarmuka Melihat *List* Nilai Tugas pada Aplikasi MyITS Mahasiswa

4.3 Implementasi Kasus Penggunaan

Implementasi kasus penggunaan aplikasi berbasis perangkat bergerak ini menggunakan bahasa pemrograman Java yang dibangun dengan perangkat lunak Android Studio. Pada subbab ini akan menjelaskan dan menampilkan kode semu (*pseudocode*) pada sisi aplikasi perangkat bergerak.

Pada dasarnya, implementasi pada setiap kasus penggunaan menggunakan paradigma *reactive programming* untuk mengambil data dari API. Pemanggilan data dari API dimulai ketika memanggil *endpoint* untuk mendapatkan data. Selanjutnya, *method subscribeOn()* dipanggil dengan parameter *Schedulers.io()*. *Method* ini berfungsi untuk menentukan *thread Schedulers.io()* sebagai tempat dimana proses pengambilan data dilakukan. Kemudian *method observeOn()* dipanggil dengan parameter *AndroidSchedulers.mainThread()* yang berarti setelah pengambilan data selesai, *thread* akan berpindah ke *thread* utama atau UI *thread*.

Respon akan diterima setelah memanggil *endpoint*. Respon yang diterima berupa JSON yang telah dimodelkan menjadi objek. Respon bisa berupa sebuah objek atau *list* yang berisi objek-objek. Apabila respon objek tidak kosong, maka objek tersebut akan dikirimkan ke *view* untuk ditampilkan dan menyimpan data tersebut ke *shared preferences*. Sedangkan apabila respon objek kosong, maka dilakukan pengecekan pada *shared preferences*, apakah ada data yang tersimpan sebelumnya, jika ada maka data tersebut ditampilkan ke *view*.

4.3.1 Implementasi Kasus Penggunaan Pengelolaan Data Kelas

Subbab ini membahas mengenai implementasi kasus penggunaan pengelolaan data kelas. Kasus penggunaan ini meliputi menampilkan *list* kelas dan menampilkan halaman utama kelas. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa. Implementasi kode semu pada kasus penggunaan menampilkan *list* kelas dan halaman utama kelas dapat dilihat pada Kode Semu 4.3.1 dan Kode Semu 4.3.2.

1	Set <i>refreshing</i> pada <i>swipeRefreshLayout</i> menjadi <i>true</i>
2	Memanggil <i>endpoint</i> untuk mendapatkan informasi <i>list</i> kelas
3	SET <i>Schedulers.io()</i> pada <i>method subscribeOn()</i> untuk proses <i>request</i> data
4	SET <i>AndroidSchedulers.mainThread()</i> pada <i>method observeOn()</i>
5	Menerima respon <i>list</i> dari objek <i>ListKelas</i> yang berisi informasi <i>list</i> kelas tiap <i>user</i>
6	<p>Apabila respon <i>list</i> objek <i>ListKelas</i> tidak kosong</p> <p>IF isi <i>list</i> lebih dari 0</p> <p>Mengosongkan <i>list ListKelas</i></p> <p>FOR setiap objek <i>ListKelas</i> pada respon <i>list</i></p> <p>Tambahkan objek <i>ListKelas</i> ke <i>list ListKelas</i></p> <p>Simpan data <i>ListKelas</i> ke <i>shared preferences</i></p> <p>Set <i>visibility emptyView</i> pada <i>view</i> menjadi GONE</p> <p>Mengirim <i>list</i> objek <i>ListKelas</i> ke <i>view</i></p> <p>ELSE</p> <p>Set <i>visibility emptyView</i> pada <i>view</i> menjadi VISIBLE</p> <p>Set <i>refreshing</i> pada <i>swipeRefreshLayout</i> menjadi <i>false</i></p>
7	<p>Apabila respon <i>list</i> objek <i>ListKelas</i> kosong</p> <p>IF <i>list</i> objek <i>ListKelas</i> pada <i>shared preferences</i> tidak kosong</p> <p>Mengirim <i>list</i> objek <i>ListKelas</i> ke <i>view</i></p> <p>Set <i>visibility emptyView</i> pada <i>view</i> menjadi GONE</p> <p>ELSE</p>

	Set <i>visibility emptyView</i> pada <i>view</i> menjadi VISIBLE
8	Set <i>refreshing</i> pada <i>swipeRefreshLayout</i> menjadi <i>false</i>
9	IF internet terhubung Tampilkan pesan “Terjadi kesalahan! Mohon untuk mengulang kembali” ELSE Tampilkan pesan “Tidak ada koneksi internet”

Kode Semu 4.3.1. Kasus Penggunaan Menampilkan *List* Kelas

1	Set <i>refreshing</i> pada <i>swipeRefreshLayout</i> menjadi <i>true</i>
2	Memanggil <i>endpoint</i> untuk mendapatkan informasi halaman utama kelas
3	SET <i>Schedulers.io()</i> pada <i>method subscribeOn()</i> untuk proses <i>request</i> data
4	SET <i>AndroidSchedulers.mainThread()</i> pada <i>method observeOn()</i>
5	Menerima respon <i>list</i> dari objek Modul yang berisi informasi <i>list</i> <i>courseModel</i>
6	Apabila respon <i>list</i> objek <i>courseModel</i> tidak kosong IF isi <i>list</i> lebih dari 0 Mengosongkan <i>list</i> <i>courseModel</i> FOR setiap objek <i>courseModel</i> pada respon <i>list</i> Tambahkan objek <i>courseModel</i> ke <i>list</i> <i>courseModel</i> Simpan data <i>courseModel</i> ke <i>shared preferences</i> Set <i>visibility emptyView</i> pada <i>view</i> menjadi GONE Mengirim <i>list</i> objek <i>courseModel</i> ke <i>view</i>

	<p>ELSE</p> <p>Set <i>visibility emptyView</i> pada <i>view</i> menjadi VISIBLE</p> <p>Set <i>refreshing</i> pada <i>swipeRefreshLayout</i> menjadi <i>false</i></p>
7	<p>Apabila respon <i>list</i> objek <i>courseModel</i> kosong</p> <p>IF <i>list</i> objek <i>courseModel</i> pada <i>shared preferences</i> tidak kosong</p> <p>Mengirim <i>list</i> objek <i>courseModel</i> ke <i>view</i></p> <p>Set <i>visibility emptyView</i> pada <i>view</i> menjadi GONE</p> <p>ELSE</p> <p>Set <i>visibility emptyView</i> pada <i>view</i> menjadi VISIBLE</p>
8	<p>Set <i>refreshing</i> pada <i>swipeRefreshLayout</i> menjadi <i>false</i></p>
9	<p>IF internet terhubung</p> <p>Tampilkan pesan “Terjadi kesalahan! Mohon untuk mengulang kembali”</p> <p>ELSE</p> <p>Tampilkan pesan “Tidak ada koneksi internet”</p>

Kode Semu 4.3.2. Kasus Penggunaan Menampilkan Halaman Utama Kelas

4.3.2 Implementasi Kasus Penggunaan Pengelolaan Data Peserta Kelas

Subbab ini membahas mengenai implementasi kasus penggunaan pengelolaan data peserta kelas. Kasus penggunaan ini meliputi menampilkan *list* peserta kelas, menampilkan jumlah mahasiswa yang mengikuti kelas, mendaftarkan peserta ke kelas, serta menghapus peserta dari kelas. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Dosen. Implementasi kode semu pada kasus penggunaan menampilkan *list* peserta kelas,

menampilkan jumlah mahasiswa yang mengikuti kelas, mendaftarkan peserta ke kelas, serta menghapus peserta dari kelas dapat dilihat pada Kode Semu 4.3.3, Kode Semu 4.3.4, Kode Semu 4.3.5 dan Kode Semu 4.3.6.

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>course</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan data kelas pada halaman data kelas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman data kelas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan data kelas kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.3. Kasus Penggunaan Menampilkan List Peserta Kelas

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>course</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan jumlah mahasiswa pada halaman utama kelas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan jumlah mahasiswa kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.4. Kasus Penggunaan Menampilkan Jumlah Mahasiswa yang Mengikuti Kelas

1	Mengambil data nama dan NRP yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada nama dan NRP yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan POST dengan mengirim data nama dan NRP, id kelas, serta id <i>section</i>

5	<p>Apabila <i>method</i> POST yang dipanggil berhasil</p> <ul style="list-style-type: none"> • Menampilkan pesan data kelas berhasil ditambahkan • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman data kelas
6	<p>Apabila <i>method</i> POST yang dipanggil gagal</p> <ul style="list-style-type: none"> • Menampilkan pesan data kelas gagal ditambahkan • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.5. Kasus Penggunaan Menambah Data kelas

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan DELETE dengan mengirim id <i>user</i> , id <i>course</i>
3	<p>Apabila <i>method</i> DELETE yang dipanggil berhasil</p> <ul style="list-style-type: none"> • Menampilkan pesan NRP berhasil dihapus • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman data kelas
4	<p>Apabila <i>method</i> DELETE yang dipanggil gagal</p> <ul style="list-style-type: none"> • Menampilkan pesan NRP gagal dihapus • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.6. Kasus Penggunaan Menghapus Data Kelas

4.3.3 Implementasi Kasus Penggunaan Pengelolaan Data Tugas

Subbab ini membahas mengenai implementasi kasus penggunaan pengelolaan data tugas. Kasus penggunaan ini meliputi menampilkan list pengumpulan tiap tugas, mem-*posting* tugas, menghapus tugas, meng-*edit* tugas, menampilkan detail tugas, menampilkan list nilai tiap tugas, menilai tugas yang sudah dikumpulkan oleh mahasiswa. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa. Implementasi kode semu pada kasus penggunaan menampilkan list pengumpulan tiap tugas, mem-*posting* tugas, menghapus tugas, meng-*edit* tugas, menampilkan detail tugas, menampilkan list nilai tiap tugas serta menilai tugas yang sudah

dikumpulkan oleh mahasiswa dapat dilihat pada Kode Semu 4.3.7, Kode Semu 4.3.8, Kode Semu 4.3.9, Kode Semu 4.3.10, Kode Semu 4.3.11, Kode Semu 4.3.12 dan Kode Semu 4.3.13.

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>assign</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan <i>list</i> pengumpulan tugas tiap tugas pada halaman tugas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman tugas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan <i>list</i> pengumpulan tugas kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.7. Kasus Penggunaan Menampilkan List Pengumpulan Tiap Tugas

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>assign</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan detail tugas pada halaman tugas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman tugas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan detail tugas kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.8. Kasus Penggunaan Menampilkan Detail Tugas

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>assign</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan <i>list</i> nilai pada halaman nilai

	<ul style="list-style-type: none"> • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman nilai
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan <i>list</i> nilai kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.9. Kasus Penggunaan Menampilkan *List* Nilai Tugas

1	Mengambil data nilai dan komentar penilaian yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada nilai dan komentar penilaian yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan POST dengan mengirim data nilai dan komentar penilaian dan id <i>assign</i>
5	Apabila <i>method</i> POST yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan tugas berhasil dinilai • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman nilai tugas
6	Apabila <i>method</i> POST yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan gagal menilai tugas • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.10. Kasus Penggunaan Menilai Tugas

1	Mengambil data judul dan deskripsi yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada judul dan deskripsi yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan POST dengan mengirim data judul dan deskripsi, id kelas, serta id <i>section</i>
5	Apabila <i>method</i> POST yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan tugas berhasil ditambahkan • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
6	Apabila <i>method</i> POST yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan tugas gagal ditambahkan • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.11. Kasus Penggunaan Menambah Tugas

1	Mengambil data judul dan deskripsi yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada judul dan deskripsi yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan PUT dengan mengirim data judul dan deskripsi dan id <i>assign</i>
5	Apabila <i>method</i> PUT yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan tugas berhasil diedit • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
6	Apabila <i>method</i> PUT yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan tugas gagal diedit • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.12. Kasus Penggunaan Mengubah Tugas

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan DELETE dengan mengirim id <i>assign</i>
3	Apabila <i>method</i> DELETE yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan tugas berhasil dihapus • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
4	Apabila <i>method</i> DELETE yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan tugas gagal dihapus • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.13. Kasus Penggunaan Menghapus Tugas

4.3.4 Implementasi Kasus Penggunaan Pengelolaan Materi Perkuliahan

Subbab ini membahas mengenai implementasi kasus penggunaan pengelolaan materi perkuliahan. Kasus penggunaan ini meliputi mem-*posting* materi perkuliahan, menghapus materi perkuliahan, meng-*edit* materi perkuliahan, menampilkan materi perkuliahan. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Dosen. Implementasi kode semu pada kasus penggunaan mem-*posting* materi perkuliahan, menghapus materi perkuliahan, meng-*edit* materi perkuliahan serta menampilkan

materi perkuliahan dapat dilihat pada Kode Semu 4.3.14, Kode Semu 4.3.15, Kode Semu 4.3.16 dan Kode Semu 4.3.17.

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>glossary</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan materi perkuliahan pada halaman detail kelas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.14. Kasus Penggunaan Menampilkan Materi Perkuliahan

1	Mengambil data judul dan deskripsi yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada judul dan deskripsi yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan POST dengan mengirim data judul dan deskripsi, id kelas, serta id <i>section</i>
5	Apabila <i>method</i> POST yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan berhasil ditambahkan • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
6	Apabila <i>method</i> POST yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan gagal ditambahkan • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.15. Kasus Penggunaan Menambah Materi Perkuliahan

1	Mengambil data judul dan deskripsi yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada judul dan deskripsi yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>

4	Memanggil <i>endpoint</i> untuk melakukan PUT dengan mengirim data judul dan deskripsi dan id <i>glossary</i>
5	Apabila <i>method</i> PUT yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan berhasil diedit • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
6	Apabila <i>method</i> PUT yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan gagal diedit • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.16. Kasus Penggunaan Mengubah Materi Perkuliahan

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan DELETE dengan mengirim id <i>glossary</i>
3	Apabila <i>method</i> DELETE yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan berhasil dihapus • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
4	Apabila <i>method</i> DELETE yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan gagal dihapus • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.17. Kasus Penggunaan Menghapus Materi Perkuliahan

4.3.5 Implementasi Kasus Penggunaan Melihat Tugas

Subbab ini membahas mengenai implementasi kasus penggunaan melihat tugas. Kasus penggunaan ini meliputi menampilkan list tugas yang sudah dikumpulkan, menampilkan detail tugas. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Mahasiswa. Implementasi kode semu pada kasus penggunaan menampilkan list tugas yang sudah dikumpulkan dan menampilkan detail tugas dapat dilihat pada Kode Semu 4.3.18 dan Kode Semu 4.3.19.

1	Menampilkan <i>loading</i> pada <i>view</i>
---	---

2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>assign</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan <i>list</i> tugas yang sudah dikumpulkan pada halaman tugas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman tugas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan <i>list</i> tugas kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.18. Kasus Penggunaan Menampilkan List Tugas yang sudah Dikumpulkan

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>assign</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan detail tugas pada halaman tugas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman tugas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan detail tugas kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.19. Kasus Penggunaan Menampilkan Detail Tugas

4.3.6 Implementasi Kasus Penggunaan Melihat Materi Perkuliahan

Subbab ini membahas mengenai implementasi kasus penggunaan melihat materi perkuliahan. Kasus penggunaan ini meliputi menampilkan materi perkuliahan. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Mahasiswa. Implementasi kode semu pada kasus penggunaan menampilkan materi perkuliahan dapat dilihat pada Kode Semu 4.3.20.

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>glossary</i>

3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan materi perkuliahan pada halaman detail kelas • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman detail kelas
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan materi perkuliahan kosong • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.20. Kasus Penggunaan Menampilkan Materi Perkuliahan

4.3.7 Implementasi Kasus Penggunaan Pengelolaan Detail Materi Perkuliahan

Subbab ini membahas mengenai implementasi kasus penggunaan pengelolaan detail materi perkuliahan. Kasus penggunaan ini meliputi mem-*posting* detail materi perkuliahan, menghapus detail materi perkuliahan, meng-*edit* detail materi perkuliahan, menampilkan detail materi perkuliahan serta mengunduh detail materi perkuliahan. Kasus penggunaan ini diimplementasikan pada aplikasi MyITS-Dosen dan MyITS-Mahasiswa. Implementasi kode semu pada kasus penggunaan mem-*posting* detail materi perkuliahan, menghapus detail materi perkuliahan, meng-*edit* detail materi perkuliahan, menampilkan detail materi perkuliahan serta mengunduh detail materi perkuliahan dapat dilihat pada Kode Semu 4.3.21, Kode Semu 4.3.22, Kode Semu 4.3.23, dan Kode Semu 4.3.24.

1	Mengambil data judul dan deskripsi yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada judul dan deskripsi yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan POST dengan mengirim data judul dan deskripsi, id <i>glossary</i>
5	Apabila <i>method</i> POST yang dipanggil berhasil

	<ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan berhasil ditambahkan • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman materi perkuliahan
6	Apabila <i>method</i> POST yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan gagal ditambahkan • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.21. Kasus Penggunaan Menambah Detail Materi Perkuliahan

1	Mengambil data judul dan deskripsi yang di- <i>input</i> -kan aktor
2	Melakukan pengecekan pada judul dan deskripsi yang di- <i>input</i> -kan supaya tidak kosong
3	Menampilkan <i>loading</i> pada <i>view</i>
4	Memanggil <i>endpoint</i> untuk melakukan PUT dengan mengirim data judul dan deskripsi dan id <i>glossary</i> , id <i>entries</i>
5	Apabila <i>method</i> PUT yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan berhasil diedit • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman materi perkuliahan
6	Apabila <i>method</i> PUT yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan gagal diedit • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.22. Kasus Penggunaan Mengedit detail materi perkuliahan

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan DELETE dengan mengirim id <i>glossary</i> , id <i>entries</i>
3	Apabila <i>method</i> DELETE yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan berhasil dihapus

	<ul style="list-style-type: none"> • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman materi perkuliahan
4	Apabila <i>method</i> DELETE yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan gagal dihapus • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.23. Kasus Penggunaan Menghapus detail materi perkuliahan

1	Menampilkan <i>loading</i> pada <i>view</i>
2	Memanggil <i>endpoint</i> untuk melakukan GET dengan mengirim id <i>glossary</i> , id <i>entries</i>
3	Apabila <i>method</i> GET yang dipanggil berhasil <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan berhasil ditampilkan • Menghapus <i>loading</i> pada <i>view</i> • Memuat ulang halaman materi perkuliahan
4	Apabila <i>method</i> GET yang dipanggil gagal <ul style="list-style-type: none"> • Menampilkan pesan detail materi perkuliahan gagal ditampilkan • Menghapus <i>loading</i> pada <i>view</i>

Kode Semu 4.3.24. Kasus Penggunaan Menampilkan detail materi perkuliahan

4.4 Implementasi REST API

Pada subbab ini akan dijelaskan implementasi API menggunakan konsep REST API. API dibangun dengan menggunakan bahasa Java dan perangkat lunak Eclipse. Kerangka kerja yang digunakan adalah Spring.

Gambar 4.29. Struktur Direktori Project REST API merupakan struktur direktori pada *project* API yang dibangun. Pada *project* ini terdapat beberapa *package* yang berisi kelas-kelas dalam bahasa pemrograman Java dan terdapat berkas XML yang digunakan untuk *dependency injection*. *Package* disimpan pada direktori *src/main/java* sedangkan berkas XML disimpan pada direktori *src/main/java*.

```

|-- kelas-api
|
|-- Java Resources
|   |-- src/main/java
|       |-- id.ac.its.api.ekelas.config
|           |-- ConfigDao.java
|       |-- id.ac.its.api.ekelas.dao
|           |-- AbstractDao.java
|           |-- AssignDao.java
|           |-- AssignFeedbackCommentsDao.java
|           |-- AssignGradesDao.java
|           |-- AssignSubmissionFileDao.java
|           |-- AssignSubmissionOnlineTextDao.java
|           |-- CourseDao.java
|           |-- GlossaryDao.java
|           |-- GlossaryEntriesDao.java
|       |-- id.ac.its.api.ekelas.dao.jdbc
|           |-- JdbcAssignDao.java
|           |-- JdbcAssignFeedbackCommentsDao.java
|           |-- JdbcAssignGradesDao.java
|           |-- JdbcAssignSubmissionFileDao.java
|           |-- JdbcAssignSubmissionOnlineTextDao.java
|           |-- JdbcCourseDao.java
|           |-- JdbcFileDao.java
|           |-- JdbcGlossaryDao.java
|           |-- JdbcGlossaryEntriesDao.java
|       |-- id.ac.its.api.ekelas.dao.mapper
|           |-- AssignFeedbackCommentsUserRowMapper.java
|           |-- AssignGradesRowMapper.java
|           |-- AssignGradesUserRowMapper.java
|           |-- AssignPluginConfigRowMapper.java
|           |-- AssignRowMapper.java
|           |-- AssignSubmissionFileRowMapper.java
|           |-- AssignSubmissionOnlineTextRowMapper.java
|           |-- AssignSubmissionPerAssignmentRowMapper.java
|           |-- AssignSubmissionPerUserRowMapper.java
|           |-- AssignSubmissionRowMapper.java
|           |-- AssignUserFlagsRowMapper.java
|           |-- BlockRecentActivityRowMapper.java
|           |-- ContextRowMapper.java
|           |-- CourseModulesRowMapper.java
|           |-- CourseRowMapper.java
|           |-- CourseSectionsRowMapper.java
|           |-- CourseUserRowMapper.java
|           |-- DropDownUserIdRowMapper.java
|           |-- FileRowMapper.java
|           |-- GlossaryEntriesRowMapper.java
|           |-- GlossaryEntriesUserRowMapper.java
|           |-- GlossaryRowMapper.java
|           |-- InsertUserAkademikRowMapper.java
|           |-- ListCourseUserRowMapper.java
|           |-- ListGroupUserRowMapper.java
|           |-- ModulesRowMapper.java
|           |-- RoleAssignmentsRowMapper.java
|           |-- UrLFileRowMapper.java
|           |-- UserRowMapper.java
|       |-- id.ac.its.api.ekelas.endpoint
|           |-- AssignEndpoint.java
|           |-- BaseEndpoint.java
|           |-- CourseEndpoint.java
|           |-- FileEndpoint.java
|           |-- GlossaryEndpoint.java
|           |-- UserEndpoint.java
|       |-- id.ac.its.api.ekelas.error

```

```

|-- ErrorMessage.java
|-- ErrorMessages.java
|-- id.ac.its.api.ekelas.model
|-- Assign.java
|-- AssignFeedbackCommentsUser.java
|-- AssignGradesUser.java
|-- AssignPluginConfig.java
|-- AssignSubmission.java
|-- AssignSubmissionFile.java
|-- AssignSubmissionOnLineText.java
|-- AssignSubmissionPerAssignment.java
|-- AssignSubmissionPerUser.java
|-- AssignUserFlags.java
|-- BlockRecentActivity.java
|-- ContextActivity.java
|-- CourseActivity.java
|-- CourseModulesActivity.java
|-- CourseSectionsActivity.java
|-- CourseUserActivity.java
|-- DropDownUserIdActivity.java
|-- FileActivity.java
|-- GlossaryActivity.java
|-- GlossaryEntriesActivity.java
|-- GlossaryEntriesUserActivity.java
|-- InsertUserAkademikActivity.java
|-- ListCourseUserActivity.java
|-- ListGroupUserActivity.java
|-- ModulesActivity.java
|-- RoleAssignmentsActivity.java
|-- UrFileActivity.java
|-- UserActivity.java
|-- UserEnrolmentsActivity.java
|-- id.ac.its.api.ekelas.service
|-- AssignService.java
|-- AssignServiceImpl.java
|-- CourseService.java
|-- CourseServiceImpl.java
|-- FileService.java
|-- FileServiceImpl.java
|-- GlossaryService.java
|-- GlossaryServiceImpl.java
|-- src/main/resources
|-- data-beans.xml
|-- db-config.xml
|-- endpoint-beans.xml
|-- service-beans.xml

```

Gambar 4.29. Struktur Direktori Project REST API

4.4.1 Koneksi ke Database

Pada pembuatan API, diperlukan koneksi ke *database*. Proses koneksi ini memerlukan konfigurasi yang dilakukan pada berkas *db-config.xml*. Berkas ini berisi konfigurasi basis data, seperti nama basis data, *server* basis data, *username*, dan *password*. Konfigurasi basis data menggunakan *library* HikariCP. Selanjutnya, konfigurasi tersebut diinjeksikan pada *bean* dengan

ID *dataSource* yang merujuk ke kelas `com.zaxxer.hikari.HikariDataSource`.

4.4.2 Dependency Injection

Pada pembuatan API, *dependency injection* dilakukan pada berkas XML yang merupakan salah satu cara *dependency injection* pada kerangka kerja Spring. Pada kerangka kerja Spring, dikenal istilah *bean* yang berisi konfigurasi *metadata*. Sebuah *bean* merujuk ke suatu kelas yang akan menerima injeksi. Pada *bean*, terdapat beberapa properti yang membentuk konfigurasi *bean*. Contohnya adalah *properties* yang digunakan untuk melakukan injeksi dari *bean* melalui *method setter*. Terdapat beberapa berkas XML, yaitu :

4.4.2.1 data-beans.xml

Berkas ini berfungsi untuk menginjeksikan *dataSource* yang telah didefinisikan sebelumnya ke kelas :

1. `JdbcCourseDao`
2. `JdbcAssignDao`
3. `JdbcAssignFeedbackCommentsDao`
4. `JdbcAssignGradesDao`
5. `JdbcAssignSubmissionFileDao`
6. `JdbcAssignSubmissionOnlineTextDao`
7. `JdbcFileDao`
8. `JdbcGlossaryDao`
9. `JdbcGlossaryEntriesDao`

Artinya, data dari setiap *query* yang dilakukan pada kelas tersebut bersumber dari basis data yang telah dikonfigurasi pada *dataSource*.

4.4.2.2 service-beans.xml

Berkas ini berfungsi untuk menginjeksi objek yang terdaftar dalam `data-beans.xml` ke kelas implementasi. *List* injeksi objek ke kelas implementasi dapat dilihat pada Tabel 4.2.

Tabel 4.2. List Injeksi Objek ke kelas Implementasi

Kelas Objek	Kelas Implementasi
JdbcCourseDao	CourseServiceImpl
JdbcAssignDao	AssignServiceImpl
JdbcAssignFeedbackCommentsDao	
JdbcAssignGradesDao	
JdbcAssignSubmissionFileDao	
JdbcAssignSubmissionOnlineTextDao	
JdbcFileDao	FileServiceImpl
JdbcGlossaryDao	GlossaryServiceImpl
JdbcGlossaryEntriesDao	

4.4.2.3 endpoint-beans.xml

Berkas ini berfungsi untuk menginjeksi objek ke kelas Endpoint. *List* injeksi objek ke kelas Endpoint dapat dilihat pada Tabel 4.3.

Tabel 4.3. List Injeksi Objek ke Endpoint

Kelas Objek	Kelas Endpoint
AssignServiceImpl	AssignEndpoint
GlossaryServiceImpl	GlossaryEndpoint
FileServiceImpl	FileEndpoint
CourseServiceImpl	CourseEndpoint

4.4.3 Error Handler

Kelas ini berisi tentang penanganan *error* (*error handling*) pada setiap *error* yang terjadi. *Response* yang diberikan meliputi, kode *error*, pesan, deskripsi pesan, status.

4.4.4 Model

Model pada kerangka kerja Spring merupakan objek yang menyimpan hasil pengambilan data dari *database*. Model memiliki *getter* dan *setter* untuk mengambil atribut pada data Model ini yang akan diubah ke dalam bentuk JSON. Kemudian akan dikirim ke *endpoint*.

4.4.5 DAO

Pada *project* ini, terdapat beberapa *package* Java. Masing-masing *package* berisi kumpulan kelas dengan fungsinya masing-masing. Berikut adalah *package* yang terdapat pada *project* pembuatan API.

4.4.5.1 id.ac.its.api.ekelas.dao

Package ini berisi kelas-kelas *Data Access Object* (DAO). *Data Access Object* (DAO) merupakan *interface* berisi *method* untuk mengakses basis data.

4.4.5.2 id.ac.its.api.ekelas.dao.jdbc

Package ini berisi kelas yang mengimplementasikan *method* pada *interface* DAO. Kelas pada *package* ini berisi *query* untuk mengakses sumber data.

4.4.5.3 id.ac.its.api.ekelas.dao.mapper

Package ini berisi kelas yang memetakan hasil dari pengambilan data. Setiap baris data dimodelkan menjadi sebuah objek pada *package model*.

4.4.6 Service

Service merupakan kelas yang menggabungkan *data access object* (DAO) ke *endpoint*. Pada kelas ini terdapat logika yang diperlukan untuk memproses data. Selain itu, di kelas ini menerima parameter (*QueryParam*, *PathParam*, *FormParam*) dari *endpoint* yang kemudian akan diteruskan ke DAO setelah diproses terlebih dahulu di *service*.

4.4.7 Endpoint

Endpoint adalah URL yang digunakan sebagai komunikasi antara *client* dengan *database*. *Endpoint* pada kerangka kerja Spring yang dipakai ini dapat menerima *input* data dari *client* berupa parameter di *path*, *body*, dan dapat juga menerima *input* JSON. *Endpoint* ini juga dapat memberikan *return* nilai pada *body*

dan juga memberikan *response code*. Endpoint yang digunakan pada aplikasi ini dijelaskan pada Tabel 4.4.

Tabel 4.4. List Endpoint

Endpoint (api.its.ac.id/ekelas/1.0)	Method HTTP	Hasil
/assign/{id}	GET	Detail Tugas
/assign/{id}	PUT	Mengubah Detail Tugas
/assign/{id}	DELETE	Hapus Tugas
/assign/{id}/feedbackcomments	GET	Komentar penilaian tiap tugas
/assign/{id}/feedbackcomments	POST	Memberi komentar penilaian
/assign/{id}/feedbackcomments/{id}	PUT	Mengubah komentar penilaian
/assign/{id}/feedbackcomments/{id}	DELETE	Menghapus komentar penilaian
/assign/{id}/grades	GET	Nilai Tugas tiap Tugas
/assign/{id}/grades	POST	Menilai Tugas
/assign/{id}/grades/{id}	PUT	<i>Edit</i> Nilai
/assign/{id}/grades/{id}	DELETE	Hapus Nilai
/assign/{id}/submission/{id}	GET	Tugas mahasiswa dalam bentuk <i>essay</i>
/assign/user	GET	<i>List</i> tugas yang dikumpulkan oleh <i>user</i>
/assign/{id}/submission	GET	<i>List</i> tugas setiap tugas
/course/{id}	GET	Modul Aktivitas tiap Kelas
/course/{id}/count	GET	Jumlah Mahasiswa tiap Kelas
/course/{id}/user	GET	<i>List</i> Peserta Kelas
/course/{id}/assign	POST	<i>Mem-posting</i> Tugas

/course/{id}/glossary	POST	Mem- <i>posting</i> Materi Perkuliahan
/course/{id}/user	POST	Mendaftarkan <i>user</i> ke kelas
/course/{id}/user/{id}	DELETE	Menghapus <i>user</i> dari kelas
/files/{id}	GET	Mendapatkan Url <i>file</i>
/glossary/{id}	GET	Materi Perkuliahan
/glossary/{id}	PUT	Mengubah Materi Perkuliahan
/glossary/{id}	DELETE	Menghapus Materi Perkuliahan
/glossary/{id}/entries	GET	Detail Materi Perkuliahan
/glossary/{id}/entries	POST	Mem- <i>posting</i> Detail Materi Perkuliahan
/glossary/{id}/entries/{id}	PUT	Mengubah Detail Materi Perkuliahan
/glossary/{id}/entries/{id}	DELETE	Menghapus Detail Materi Perkuliahan
/user/listcourse	GET	<i>List</i> Kelas yang diikuti/ diampu oleh <i>user</i>
/user/	GET	Data <i>User</i>
/user/	POST	Mendaftarkan <i>User</i> ke Moodle

[Halaman ini sengaja dikosongkan]

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas tentang pengujian dan evaluasi pada perangkat lunak yang dibangun untuk tugas akhir ini. Pengujian dilakukan pada kasus penggunaan dari sistem perangkat lunak.

5.1 Lingkungan Pengujian

Pada proses pengujian fungsionalitas perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam uji fungsionalitas ini dipakai untuk semua kasus penggunaan. Spesifikasi lingkungan pengujian akan dijelaskan pada Tabel 5.1 dan Tabel 5.2.

Tabel 5.1. Lingkungan Pengujian Fungsionalitas Perangkat Lunak

Spesifikasi	Deskripsi
Jenis Perangkat	<i>Tablet</i>
Merek Perangkat	Samsung Galaxy Tab A
Sistem Operasi	Android Nougat 7.1
Memori Internal	16 GB
RAM	2 GB
Ukuran Layar	8.0 <i>inch</i>

Tabel 5.2. Lingkungan Pengujian Fungsionalitas Perangkat Lunak

Spesifikasi	Deskripsi
Jenis Perangkat	<i>Smartphone</i>
Merek Perangkat	Xiaomi Redmi Note 3 Pro
Sistem Operasi	Android Marshmallow 6.0
Memori Internal	32 GB
RAM	3 GB

Tabel 5.3. Lingkungan Pengujian Server Basis Data

Spesifikasi	Deskripsi
Prosesor	2.1 GHz Intel® Xeon® CPU E5-2450
<i>Memory</i> (RAM)	8GB
Tipe Sistem	64-bit
Sistem Operasi	Ubuntu 16.04

<i>Server</i>	Apache
Basis Data	MySQL

5.2 Pengujian Fungsionalitas

Pengujian fungsionalitas ini adalah pengujian fungsi-fungsi yang berjalan pada aplikasi berdasarkan kasus penggunaan. Pada subbab ini akan dijelaskan skenario pengujian yang digunakan. Pengujian fungsionalitas ini dilakukan dengan metode kotak hitam (*black box*), yaitu sebuah metode pengujian yang menekankan pada hasil keluaran program/aplikasi. Pengujian fungsionalitas ini adalah pengujian fungsi-fungsi yang berjalan pada aplikasi berdasarkan kasus penggunaan.

5.2.1. Pengujian Fungsionalitas Melihat *List Kelas*

Pengujian fungsionalitas melihat *list kelas* merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat *list kelas*. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.4.

Tabel 5.4 Skenario Pengujian Fungsionalitas Melihat *List Kelas*

No. Pengujian	SCF-001
Skenario Pengujian	Melihat <i>List Kelas</i>
Kondisi Awal	Data pengguna telah tersimpan
Langkah-langkah	1. Penguji masuk ke halaman <i>list kelas</i> dengan memilih menu “ <i>Ekelas</i> ” pada beranda aplikasi.
Hasil yang diharapkan	Aplikasi dapat menampilkan <i>list kelas</i> tiap pengguna
Hasil yang diperoleh	<i>list kelas</i> tiap pengguna ditampilkan
Hasil Pengujian	Berhasil

5.2.2. Pengujian Fungsionalitas Melihat Halaman Utama Kelas

Pengujian fungsionalitas melihat halaman utama kelas merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat halaman utama kelas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.5.

Tabel 5.5 Skenario Pengujian Fungsionalitas Melihat Halaman Utama Kelas

No. Pengujian	SCF-002
Skenario Pengujian	Melihat Halaman Utama Kelas
Kondisi Awal	Pengguna berada pada halaman list kelas
Langkah-langkah	1. Penguji masuk ke halaman utama kelas dengan memilih kelas pada halaman <i>list</i> kelas.
Hasil yang diharapkan	Aplikasi dapat menampilkan halaman utama kelas
Hasil yang diperoleh	Halaman utama kelas ditampilkan
Hasil Pengujian	Berhasil

5.2.3. Pengujian Fungsionalitas Melihat Data Kelas

Pengujian fungsionalitas melihat data kelas merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat data kelas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.6.

Tabel 5.6 Skenario Pengujian Fungsionalitas Melihat Data Kelas

No. Pengujian	SCF-003
Skenario Pengujian	Melihat Halaman Data Kelas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	1. Penguji masuk ke halaman data kelas dengan memilih tombol

	“data kelas” pada halaman utama kelas.
Hasil yang diharapkan	Aplikasi dapat menampilkan halaman data kelas
Hasil yang diperoleh	Halaman data kelas ditampilkan
Hasil Pengujian	Berhasil

5.2.4. Pengujian Fungsionalitas Menambah Data Kelas

Pengujian fungsionalitas menambah data kelas merupakan pengujian terhadap kemampuan pengguna untuk dapat menambah data kelas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.7.

Tabel 5.7 Skenario Pengujian Fungsionalitas Menambah Data Kelas

No. Pengujian	SCF-004
Skenario Pengujian	Menambah Data Kelas
Kondisi Awal	Pengguna berada pada halaman data kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih tombol “tambah data kelas” 2. Penguji mengisi nama dan NRP mahasiswa 3. Penguji mengklik tombol “tambah” 4. Penguji kembali ke halaman data kelas
Hasil yang diharapkan	Aplikasi dapat menampilkan halaman data kelas
Hasil yang diperoleh	Halaman data kelas ditampilkan
Hasil Pengujian	Berhasil

5.2.5. Pengujian Fungsionalitas Menghapus Data Kelas

Pengujian fungsionalitas menghapus data kelas merupakan pengujian terhadap kemampuan pengguna untuk dapat menghapus data kelas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.8.

Tabel 5.8. Skenario Pengujian Fungsionalitas Menghapus Data Kelas

No. Pengujian	SCF-005
Skenario Pengujian	Menghapus Data Kelas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji mengklik ikon data kelas 2. Penguji mengklik ikon <i>delete</i> di samping <i>item</i> data kelas 3. Penguji memilih “ya” pada pesan konfirmasi hapus 4. Penguji kembali ke halaman data kelas
Hasil yang diharapkan	Aplikasi dapat menghapus data kelas dan menghilangkannya dari <i>list</i> data kelas
Hasil yang diperoleh	data kelas berhasil dihapus dan dihilangkan dari <i>list</i> data kelas
Hasil Pengujian	Berhasil

5.2.6. Pengujian Fungsionalitas Menambah Materi Perkuliahan

Pengujian fungsionalitas menambah materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat menambah materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.9.

Tabel 5.9. Skenario Pengujian Fungsionalitas Menambah Materi Perkuliahan

No. Pengujian	SCF-006
Skenario Pengujian	Menambah Materi Perkuliahan
Kondisi Awal	Pengguna berada pada halaman utama kelas

Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji mengklik tombol “tambah” 3. Penguji mengisi <i>form</i> judul dan deskripsi materi perkuliahan 4. Penguji mengklik tombol <i>submit</i> 5. Penguji kembali ke halaman utama
Hasil yang diharapkan	Aplikasi dapat menampilkan data materi perkuliahan pada minggu yang diinginkan
Hasil yang diperoleh	data materi perkuliahan pada minggu yang diinginkan dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.7. Pengujian Fungsionalitas Melihat Detail Materi Perkuliahan

Pengujian fungsionalitas melihat detail materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat detail materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.10.

Tabel 5.10. Skenario Pengujian Fungsionalitas Melihat Detail Materi Perkuliahan

No. Pengujian	SCF-007
Skenario Pengujian	Melihat Detail Materi Perkuliahan
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> materi perkuliahan 3. Penguji mengklik <i>item</i> materi perkuliahan 4. Penguji melihat materi perkuliahan

Hasil yang diharapkan	Aplikasi dapat menampilkan data detail materi perkuliahan pada halaman materi perkuliahan
Hasil yang diperoleh	data detail materi perkuliahan pada halaman materi perkuliahan dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.8. Pengujian Fungsionalitas *Edit Materi Perkuliahan*

Pengujian fungsionalitas *edit* materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat *edit* materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.11.

Tabel 5.11. Skenario Pengujian Fungsionalitas Edit Materi Perkuliahan

No. Pengujian	SCF-008
Skenario Pengujian	<i>Edit Materi Perkuliahan</i>
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji mengklik ikon <i>edit</i> di samping <i>item</i> materi perkuliahan 3. Penguji mengubah <i>form</i> judul dan deskripsi materi perkuliahan 4. Penguji mengklik tombol <i>submit</i> 5. Penguji kembali ke halaman utama
Hasil yang diharapkan	Aplikasi dapat menampilkan data materi perkuliahan pada minggu yang diinginkan yang telah diubah
Hasil yang diperoleh	data materi perkuliahan pada minggu yang diinginkan yang telah diubah dapat ditampilkan

Hasil Pengujian	Berhasil
-----------------	----------

5.2.9. Pengujian Fungsionalitas Menghapus Materi Perkuliahan

Pengujian fungsionalitas menghapus materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat menghapus materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.12.

Tabel 5.12. Skenario Pengujian Fungsionalitas Menghapus Materi Perkuliahan

No. Pengujian	SCF-009
Skenario Pengujian	Menghapus Materi Perkuliahan
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji mengklik ikon <i>delete</i> di samping <i>item</i> materi perkuliahan 3. Penguji memilih “ya” pada pesan konfirmasi hapus 4. Penguji kembali ke halaman utama
Hasil yang diharapkan	Aplikasi dapat menghapus data materi perkuliahan dan menghilangkannya dari <i>list</i> aktivitas kelas
Hasil yang diperoleh	data materi perkuliahan berhasil dihapus dan dihilangkan dari <i>list</i> aktivitas kelas
Hasil Pengujian	Berhasil

5.2.10. Pengujian Fungsionalitas Menambah Tugas

Pengujian fungsionalitas menambah tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat menambah

tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.13.

Tabel 5.13. Skenario Pengujian Fungsionalitas Menambah Tugas

No. Pengujian	SCF-0010
Skenario Pengujian	Menambah Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji mengklik tombol “tambah” 3. Penguji mengisi <i>form</i> judul dan deskripsi tugas 4. Penguji mengklik tombol <i>submit</i> 5. Penguji kembali ke halaman utama
Hasil yang diharapkan	Aplikasi dapat menampilkan data tugas pada minggu yang diinginkan
Hasil yang diperoleh	data tugas pada minggu yang diinginkan dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.11. Pengujian Fungsionalitas Melihat Detail Tugas

Pengujian fungsionalitas melihat detail tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat detail tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.14.

Tabel 5.14. Skenario Pengujian Fungsionalitas Melihat Detail Tugas

No. Pengujian	SCF-011
Skenario Pengujian	Melihat Detail Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas

Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> tugas 3. Penguji mengklik <i>item</i> tugas 4. Penguji melihat detail tugas
Hasil yang diharapkan	Aplikasi dapat menampilkan data detail tugas pada halaman tugas
Hasil yang diperoleh	data detail tugas pada halaman tugas dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.12. Pengujian Fungsionalitas *Edit* Tugas

Pengujian fungsionalitas *edit* tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat *edit* tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.15.

Tabel 5.15. Skenario Pengujian Fungsionalitas *Edit* Tugas

No. Pengujian	SCF-012
Skenario Pengujian	<i>Edit</i> Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji mengklik ikon <i>edit</i> di samping <i>item</i> tugas 3. Penguji mengubah <i>form</i> judul dan deskripsi tugas 4. Penguji mengklik tombol <i>submit</i> 5. Penguji kembali ke halaman utama
Hasil yang diharapkan	Aplikasi dapat menampilkan data tugas pada minggu yang diinginkan yang telah diubah

Hasil yang diperoleh	data tugas pada minggu yang diinginkan yang telah diubah dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.13. Pengujian Fungsionalitas Menghapus Tugas

Pengujian fungsionalitas menghapus tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat menghapus tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.16.

Tabel 5.16. Skenario Pengujian Fungsionalitas Menghapus Tugas

No. Pengujian	SCF-013
Skenario Pengujian	Menghapus Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji mengklik ikon <i>delete</i> di samping <i>item</i> tugas 3. Penguji memilih “ya” pada pesan konfirmasi hapus 4. Penguji kembali ke halaman utama
Hasil yang diharapkan	Aplikasi dapat menghapus data tugas dan menghilangkannya dari <i>list</i> aktivitas kelas
Hasil yang diperoleh	data tugas berhasil dihapus dan dihilangkan dari <i>list</i> aktivitas kelas
Hasil Pengujian	Berhasil

5.2.14. Pengujian Fungsionalitas Melihat *List* Pengumpulan Tugas

Pengujian fungsionalitas melihat *list* pengumpulan tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat *list* pengumpulan tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.17.

Tabel 5.17. Skenario Pengujian Fungsionalitas Melihat *List* Pengumpulan Tugas

No. Pengujian	SCF-014
Skenario Pengujian	Melihat <i>List</i> Pengumpulan Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> tugas 3. Penguji mengklik <i>item</i> tugas 4. Penguji melihat <i>list</i> pengumpulan tugas
Hasil yang diharapkan	Aplikasi dapat menampilkan data <i>list</i> pengumpulan tugas pada halaman tugas
Hasil yang diperoleh	data <i>list</i> pengumpulan tugas pada halaman tugas dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.15. Pengujian Fungsionalitas Menilai Tugas

Pengujian fungsionalitas menilai tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat menilai tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.18.

Tabel 5.18. Skenario Pengujian Fungsionalitas Menilai Tugas

No. Pengujian	SCF-015
---------------	---------

Skenario Pengujian	Menilai Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> tugas 3. Penguji mengklik <i>item</i> tugas 4. Penguji melihat <i>list</i> pengumpulan tugas 5. Penguji memilih <i>item</i> tugas yang ingin dinilai 6. Penguji mengklik <i>item</i> tugas 7. Penguji mengisi <i>form</i> nilai dengan memasukkan nilai dan komentar penilaian 8. Penguji menekan tombol <i>submit</i> 9. Penguji memilih halaman lihat nilai untuk melihat hasil penilaian
Hasil yang diharapkan	Aplikasi dapat menyimpan nilai yang sudah diisi oleh penguji pada halaman nilai tugas masing-masing
Hasil yang diperoleh	Nilai yang sudah diisi oleh penguji dapat disimpan pada halaman nilai tugas masing-masing
Hasil Pengujian	Berhasil

5.2.16. Pengujian Fungsionalitas Melihat *List* Nilai Tugas

Pengujian fungsionalitas melihat *list* nilai tugas merupakan pengujian terhadap kemampuan pengguna untuk dapat melihat *list* nilai tugas. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.19.

Tabel 5.19. Skenario Pengujian Fungsionalitas Melihat List Nilai Tugas

No. Pengujian	SCF-016
---------------	---------

Skenario Pengujian	Melihat <i>List</i> Nilai Tugas
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> tugas 3. Penguji mengklik <i>item</i> tugas 4. Penguji melihat <i>list</i> pengumpulan tugas 5. Penguji memilih <i>item</i> tugas yang ingin dinilai 6. Penguji memilih tombol lihat nilai 7. Penguji melihat <i>list</i> nilai tugas
Hasil yang diharapkan	Aplikasi dapat menampilkan <i>list</i> nilai tugas pada halaman <i>list</i> nilai tugas
Hasil yang diperoleh	<i>List</i> nilai tugas pada halaman <i>list</i> nilai tugas dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.17. Pengujian Fungsionalitas Menambah Detail Materi Perkuliahan

Pengujian fungsionalitas menambah detail materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat menambah detail materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.20.

Tabel 5.20. Skenario Pengujian Fungsionalitas Menambah Detail Materi Perkuliahan

No. Pengujian	SCF-017
Skenario Pengujian	Menambah Detail Materi Perkuliahan
Kondisi Awal	Pengguna berada pada halaman utama kelas

Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> materi kuliah 3. Penguji mengklik <i>item</i> materi kuliah 4. Penguji mengklik tombol “tambah materi” 5. Penguji mengisi <i>form</i> detail materi perkuliahan 6. Penguji mengklik tombol submit 7. penguji kembali ke halaman materi perkuliahan 8. Penguji melihat detail materi perkuliahan
Hasil yang diharapkan	Aplikasi dapat menampilkan detail materi perkuliahan baru pada halaman materi perkuliahan
Hasil yang diperoleh	Detail materi perkuliahan baru pada halaman materi perkuliahan dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.18. Pengujian Fungsionalitas *Edit* Detail Materi Perkuliahan

Pengujian fungsionalitas *edit* detail materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat *edit* detail materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.21.

Tabel 5.21. Skenario Pengujian Fungsionalitas Edit Detail Materi Perkuliahan

No. Pengujian	SCF-018
Skenario Pengujian	<i>Edit</i> Detail Materi Perkuliahan
Kondisi Awal	Pengguna berada pada halaman utama kelas

Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> materi perkuliahan 3. Penguji mengklik <i>item</i> materi perkuliahan 4. Penguji memilih detail materi perkuliahan 5. Penguji mengklik <i>item edit</i> di samping detail materi perkuliahan 6. Penguji mengisi ulang <i>form</i> detail materi perkuliahan 7. Penguji mengklik tombol submit 8. penguji kembali ke halaman materi perkuliahan 9. Penguji melihat detail materi perkuliahan yang sudah diubah
Hasil yang diharapkan	Aplikasi dapat menampilkan detail materi perkuliahan yang sudah diubah pada halaman materi perkuliahan
Hasil yang diperoleh	Detail materi perkuliahan yang sudah diubah pada halaman materi perkuliahan dapat ditampilkan
Hasil Pengujian	Berhasil

5.2.19. Pengujian Fungsionalitas Menghapus Detail Materi Perkuliahan

Pengujian fungsionalitas menghapus detail materi perkuliahan merupakan pengujian terhadap kemampuan pengguna untuk dapat menghapus detail materi perkuliahan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.22.

Tabel 5.22. Skenario Pengujian Fungsionalitas Menghapus Detail Materi Perkuliahan

No. Pengujian	SCF-019
---------------	---------

Skenario Pengujian	Menghapus Detail Materi Perkuliahan
Kondisi Awal	Pengguna berada pada halaman utama kelas
Langkah-langkah	<ol style="list-style-type: none"> 1. Penguji memilih minggu yang diinginkan 2. Penguji memilih <i>item</i> materi perkuliahan 3. Penguji mengklik <i>item</i> materi perkuliahan 4. Penguji memilih detail materi perkuliahan 5. Penguji mengklik <i>item delete</i> di samping detail materi perkuliahan 6. Penguji memilih "ya"
Hasil yang diharapkan	Aplikasi dapat menghapus data detail materi perkuliahan yang dipilih dan menghilangkannya dari daftar aktivitas kelas
Hasil yang diperoleh	Data detail materi yang dipilih berhasil dihapus dan dihilangkan dari <i>list</i> detail materi perkuliahan
Hasil Pengujian	Berhasil

5.3 Pengujian REST API

Pengujian REST API dibagi menjadi dua bagian yaitu pengujian *response body* dan *respond time*. Kedua pengujian ini dilakukan untuk mengetahui kesesuaian *response body* dari *endpoint* REST API dan performa REST API ketika diakses oleh banyak pengguna.

5.3.1 Pengujian Response Body

Pengujian *response body* dilakukan pada *endpoint* di modul-modul REST API. Setiap *endpoint* diuji kesesuaian *response body* berdasarkan skenario yang telah ditentukan. Pengujian ini dilakukan dengan aplikasi REST API *client* JMeter.

5.3.1.1 Skenario Pengujian

Terdapat empat skenario pengujian yang diterapkan pada setiap endpoint. Skenario A dan B didesain untuk menguji parameter yang dikirim ketika mengakses *endpoint*. Sedangkan skenario C dan D didesain untuk menguji keamanan JWT pada setiap *endpoint*. Deskripsi dari setiap skenario dapat dilihat pada Tabel 5.23.

Tabel 5.23. Skenario Pengujian Response Body

Kode Skenario	Nama Skenario	Deskripsi
A	Parameter Lengkap	Pengujian dilakukan dengan mengirim seluruh parameter yang dibutuhkan oleh <i>endpoint</i> . Pengujian berhasil jika <i>status code</i> dari <i>response body</i> bernilai 200 atau 201.
B	Parameter Tidak Lengkap	Pengujian dilakukan dengan mengirim sebagian parameter yang dibutuhkan oleh <i>endpoint</i> . Pengujian berhasil jika <i>status code</i> dari <i>response body</i> bernilai 500 untuk metode HTTP POST atau 404 untuk metode HTTP GET
C	Token JWT Tidak Digunakan	Pengujian dilakukan dengan mengakses <i>endpoint</i> tanpa menggunakan <i>token</i> JWT. Pengujian Berhasil jika <i>status code</i> dari <i>response body</i> bernilai 401
D	Token JWT <i>Expired</i>	Pengujian dilakukan dengan mengakses <i>endpoint</i> menggunakan <i>token</i> JWT yang sudah <i>expired</i> . Pengujian berhasil jika <i>status</i>

		<i>code</i> dari <i>response body</i> bernilai 401.
--	--	---

5.3.1.2 Pengujian Kasus Penggunaan Melihat Data Kelas

Kasus penggunaan melihat data kelas memiliki dua *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.24 dan Tabel 5.25.

Tabel 5.24. Pengujian GET *List Class Per User*

Kode Pengujian		RA-001	
Nama <i>Endpoint</i>		GET <i>List Class Per User</i>	
<i>Endpoint</i>		/user/listcourse	
Deskripsi <i>Endpoint</i>		Mendapatkan <i>list</i> kelas yang diikuti oleh <i>user</i> .	
Metode HTTP		GET	
Kasus Penggunaan		Melihat Data Kelas (UC-001)	
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.25. Pengujian GET *Detail Class*

Kode Pengujian		RA-002	
Nama <i>Endpoint</i>		GET <i>Detail Class</i>	
<i>Endpoint</i>		/course/{id}	
Deskripsi <i>Endpoint</i>		Mendapatkan halaman utama kelas.	
Metode HTTP		GET	
Kasus Penggunaan		Melihat Data Kelas (UC-001)	
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.1.3 Pengujian Kasus Penggunaan Mengelola Data Peserta Kelas

Kasus penggunaan mengelola data peserta kelas memiliki lima *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.26, Tabel 5.27, Tabel 5.28, Tabel 5.29, dan Tabel 5.30.

Tabel 5.26. Pengujian GET *List User Per Class*

Kode Pengujian	RA-003		
Nama <i>Endpoint</i>	GET <i>List User Per Class</i>		
<i>Endpoint</i>	/course/{id}/user		
Deskripsi <i>Endpoint</i>	Mendapatkan <i>list user</i> pada masing-masing kelas.		
Metode HTTP	GET		
Kasus Penggunaan	Mengelola Data Peserta Kelas (UC-002)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.27. Pengujian GET Jumlah Mahasiswa Di Kelas

Kode Pengujian	RA-004		
Nama <i>Endpoint</i>	GET Jumlah Mahasiswa Di Kelas		
<i>Endpoint</i>	/course/{id}/count		
Deskripsi <i>Endpoint</i>	Mendapatkan jumlah mahasiswa pada masing-masing kelas.		
Metode HTTP	GET		
Kasus Penggunaan	Mengelola Data Peserta Kelas (UC-002)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.28. Pengujian GET Detail User From SIAKAD

Kode Pengujian	RA-005		
Nama <i>Endpoint</i>	GET <i>Detail User From SIAKAD</i>		
<i>Endpoint</i>	/mahasiswa/{nrp}		
Deskripsi <i>Endpoint</i>	Mendapatkan data mahasiswa yang akan didaftarkan ke Moodle dan ke kelas yang dipilih		
Metode HTTP	GET		
Kasus Penggunaan	Mengelola Data Peserta Kelas (UC-002)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.29. Pengujian POST User to Moodle and Class

Kode Pengujian	RA-006		
Nama <i>Endpoint</i>	POST <i>User to Moodle and Class</i>		
<i>Endpoint</i>	/course/{courseid}/user		
Deskripsi <i>Endpoint</i>	Mendaftarkan mahasiswa ke Moodle dan ke kelas yang dipilih		
Metode HTTP	POST		
Kasus Penggunaan	Mengelola Data Peserta Kelas (UC-002)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.30. Pengujian DELETE User from Class

Kode Pengujian	RA-007		
Nama <i>Endpoint</i>	DELETE <i>User from Class</i>		
<i>Endpoint</i>	/course/{courseid}/user/{username}		
Deskripsi <i>Endpoint</i>	Menghapus mahasiswa dari kelas		

Metode HTTP	DELETE		
Kasus Penggunaan	Mengelola Data Peserta Kelas (UC-002)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.1.4 Pengujian Kasus Penggunaan Mengelola Data Tugas

Kasus penggunaan mengelola data tugas memiliki enam *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.31, Tabel 5.32, Tabel 5.33, Tabel 5.34, Tabel 5.35, dan Tabel 5.36.

Tabel 5.31. Pengujian GET *List* Pengumpulan Tugas

Kode Pengujian	RA-008		
Nama <i>Endpoint</i>	GET <i>List</i> Pengumpulan Tugas		
<i>Endpoint</i>	/assign/{id}/submission		
Deskripsi <i>Endpoint</i>	Mendapatkan data <i>list</i> tugas yang telah dikumpulkan oleh mahasiswa.		
Metode HTTP	GET		
Kasus Penggunaan	Mengelola Data Tugas (UC-003)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.32. Pengujian GET *List* Nilai Tugas

Kode Pengujian	RA-009		
Nama <i>Endpoint</i>	GET <i>List</i> Nilai Tugas		
<i>Endpoint</i>	/assign/{id}/grades		
Deskripsi <i>Endpoint</i>	Mendapatkan data <i>list</i> nilai tugas.		

Metode HTTP	GET		
Kasus Penggunaan	Mengelola Data Tugas (UC-003)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.33. Pengujian POST Nilai Tugas

Kode Pengujian	RA-010		
Nama <i>Endpoint</i>	POST Nilai Tugas		
<i>Endpoint</i>	/assign/{id}/grades		
Deskripsi <i>Endpoint</i>	Menilai tugas.		
Metode HTTP	POST		
Kasus Penggunaan	Mengelola Data Tugas (UC-003)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.34. Pengujian POST Tugas

Kode Pengujian	RA-011		
Nama <i>Endpoint</i>	POST Tugas		
<i>Endpoint</i>	/assign/{id}		
Deskripsi <i>Endpoint</i>	Mem-posting tugas.		
Metode HTTP	POST		
Kasus Penggunaan	Mengelola Data Tugas (UC-003)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.35. Pengujian EDIT Tugas

Kode Pengujian	RA-012		
Nama <i>Endpoint</i>	EDIT Tugas		
<i>Endpoint</i>	/assign/{id}		
Deskripsi <i>Endpoint</i>	Mengubah tugas.		
Metode HTTP	PUT		
Kasus Penggunaan	Mengelola Data Tugas (UC-003)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.36. Pengujian DELETE Tugas

Kode Pengujian	RA-013		
Nama <i>Endpoint</i>	DELETE Tugas		
<i>Endpoint</i>	/assign/{id}		
Deskripsi <i>Endpoint</i>	Menghapus tugas.		
Metode HTTP	DELETE		
Kasus Penggunaan	Mengelola Data Tugas (UC-003)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.1.5 Pengujian Kasus Penggunaan Mengelola Materi Perkuliahan

Kasus penggunaan mengelola materi perkuliahan memiliki empat *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.37, Tabel 5.38, Tabel 5.39, dan Tabel 5.40.

Tabel 5.37. Pengujian GET Materi Perkuliahan

Kode Pengujian	RA-014		
Nama <i>Endpoint</i>	GET Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{id}/		
Deskripsi <i>Endpoint</i>	Mendapatkan data materi perkuliahan.		
Metode HTTP	GET		
Kasus Penggunaan	Mengelola Materi Perkuliahan (UC-004)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.38. Pengujian POST Materi Perkuliahan

Kode Pengujian	RA-015		
Nama <i>Endpoint</i>	POST Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{id}/		
Deskripsi <i>Endpoint</i>	Mem-posting materi perkuliahan.		
Metode HTTP	POST		
Kasus Penggunaan	Mengelola Materi Perkuliahan (UC-004)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.39. Pengujian EDIT Materi Perkuliahan

Kode Pengujian	RA-016		
Nama <i>Endpoint</i>	EDIT Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{id}/		
Deskripsi <i>Endpoint</i>	Mengubah materi perkuliahan.		
Metode HTTP	PUT		

Kasus Penggunaan	Mengelola Materi Perkuliahan (UC-004)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.40. Pengujian DELETE Materi Perkuliahan

Kode Pengujian	RA-017		
Nama <i>Endpoint</i>	DELETE Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{id}/		
Deskripsi <i>Endpoint</i>	Menghapus materi perkuliahan.		
Metode HTTP	DELETE		
Kasus Penggunaan	Mengelola Materi Perkuliahan (UC-004)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.1.6 Pengujian Kasus Penggunaan Melihat Tugas

Kasus penggunaan melihat tugas memiliki dua *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.41 dan Tabel 5.42.

Tabel 5.41. Pengujian GET Detail Tugas

Kode Pengujian	RA-018		
Nama <i>Endpoint</i>	GET Detail Tugas		
<i>Endpoint</i>	/assign/{id}/		
Deskripsi <i>Endpoint</i>	Mendapatkan data detail tugas.		
Metode HTTP	GET		

Kasus Penggunaan	Melihat Tugas (UC-005)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.42. Pengujian GET *List* Tugas yang Sudah Dikumpulkan

Kode Pengujian	RA-019		
Nama <i>Endpoint</i>	GET <i>List</i> Tugas yang Sudah Dikumpulkan		
<i>Endpoint</i>	/assign/user		
Deskripsi <i>Endpoint</i>	Mendapatkan data tugas yang sudah di submit oleh user.		
Metode HTTP	GET		
Kasus Penggunaan	Melihat Tugas (UC-005)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.1.7 Pengujian Kasus Penggunaan Melihat Materi Perkuliahan

Kasus penggunaan melihat materi perkuliahan memiliki satu *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.43.

Tabel 5.43. Pengujian GET Materi Perkuliahan

Kode Pengujian	RA-020		
Nama <i>Endpoint</i>	GET Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{id}		
Deskripsi <i>Endpoint</i>	Mendapatkan data materi perkuliahan.		
Metode HTTP	GET		

Kasus Penggunaan	Melihat Materi Perkuliahan (UC-006)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.1.8 Pengujian Kasus Penggunaan Mengelola Detail Materi Perkuliahan

Kasus penggunaan mengelola detail materi perkuliahan memiliki lima *endpoint* yang diujikan. Hasil dari pengujian dapat dilihat pada Tabel 5.44, Tabel 5.45, Tabel 5.46, Tabel 5.47, dan Tabel 5.48.

Tabel 5.44. Pengujian GET Detail Materi Perkuliahan

Kode Pengujian	RA-021		
Nama <i>Endpoint</i>	GET Detail Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{glossaryid}/entries		
Deskripsi <i>Endpoint</i>	Mendapatkan detail materi perkuliahan.		
Metode HTTP	GET		
Kasus Penggunaan	Mengelola Detail Materi Perkuliahan (UC-007)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.45. Pengujian GET *Files* Per Detail Materi Perkuliahan

Kode Pengujian	RA-022		
Nama <i>Endpoint</i>	GET <i>Files</i> Per Detail Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{glossaryid}/entries		
Deskripsi <i>Endpoint</i>	Mendapatkan <i>file</i> pada setiap detail materi perkuliahan.		

Metode HTTP	GET		
Kasus Penggunaan	Mengelola Detail Materi Perkuliahan (UC-007)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.46. Pengujian POST Detail Materi Perkuliahan

Kode Pengujian	RA-023		
Nama <i>Endpoint</i>	POST Detail Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{glossaryid}/entries		
Deskripsi <i>Endpoint</i>	Mem-posting detail materi perkuliahan.		
Metode HTTP	POST		
Kasus Penggunaan	Mengelola Detail Materi Perkuliahan (UC-007)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.47. Pengujian EDIT Detail Materi Perkuliahan

Kode Pengujian	RA-024		
Nama <i>Endpoint</i>	EDIT Detail Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{glossaryid}/entries/{idEntries}		
Deskripsi <i>Endpoint</i>	Mengubah detail materi perkuliahan.		
Metode HTTP	PUT		
Kasus Penggunaan	Mengelola Detail Materi Perkuliahan (UC-007)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

Tabel 5.48. Pengujian DELETE Detail Materi Perkuliahan

Kode Pengujian	RA-025		
Nama <i>Endpoint</i>	DELETE Detail Materi Perkuliahan		
<i>Endpoint</i>	/glossary/{glossaryid}/entries/{idEntries}		
Deskripsi <i>Endpoint</i>	Menghapus detail materi perkuliahan.		
Metode HTTP	DELETE		
Kasus Penggunaan	Mengelola Detail Materi Perkuliahan (UC-007)		
Hasil Pengujian	Skenario	A	Berhasil
		B	Berhasil
		C	Berhasil
		D	Berhasil

5.3.2 Pengujian Response Time

Pengujian *response time* dilakukan untuk mengetahui performa *server* dan REST API ketika diakses oleh banyak pengguna dalam waktu yang hampir bersamaan. Pengujian ini diujikan dengan jumlah pengguna yang berbeda yaitu 100, 500, 1000 dan 2000. Aplikasi Apache JMeter digunakan untuk melakukan pengujian ini.

Thread Group digunakan untuk menyimulasikan jumlah pengguna di aplikasi JMeter. Satu *thread* diasumsikan sebagai satu pengguna. Sehingga pengujian dengan 100 pengguna akan menggunakan 100 *thread*. Parameter *Ramp-Up-Period* menentukan waktu yang dibutuhkan untuk mempersiapkan *thread*. Karena pengujian ini mengukur performa dan *response time server* ketika diakses hampir bersamaan, maka nilai *Ramp-Up-Period* di jadikan satu detik. Parameter *Loop Count* akan diisi dengan nilai satu karena pengujian hanya dilakukan satu kali.

Fitur HTTP *Request* digunakan untuk melakukan *request* ke *server*. Pengujian ini akan menggunakan *endpoint Get Detail Class* untuk melakukan *request* ke *server*. Penjelasan dari *endpoint*

tersebut dapat dilihat pada Tabel 5.25. Dikarenakan *endpoint* tersebut dilindungi oleh JWT, maka *HTTP Header Manager* ditambahkan di *HTTP Request* untuk mengelola *request header*. *Key* dari *header* adalah *Authorization*. Sedangkan *value* adalah *Bearer* ditambah dengan token JWT yang didapatkan setelah *login*. Untuk melihat laporan dari pengujian JMeter, dibutuhkan beberapa *Listener*. Pengujian ini menggunakan tiga *Listener* yaitu *Summary Report*, *View Results Tree* dan *View Results in Table*. *Summary Report* digunakan untuk melihat ringkasan dari pengujian. Dari laporan tersebut, rata-rata, minimum dan maksimum *response time* beserta persentase *error* dapat dilihat. Informasi lebih detail mengenai hasil pengujian dapat dilihat di *View Results in Table*. Sedangkan *View Results Tree* menyajikan hasil informasi *response server* dari setiap *request* yang dilakukan. Hasil pengujian yang ditampilkan pada bab ini mengacu pada *Summary Report*.

5.3.2.1 Pengujian 100 Pengguna

Berdasarkan Tabel 5.49, pengujian 100 pengguna pada *server* menghasilkan tingkat keberhasilan 100 persen dan rata-rata *response time* 1,760 detik. *Response Time* tersebut dapat dikatakan cepat dan dapat diandalkan.

Tabel 5.49. Hasil Pengujian *Response Time* 100 Pengguna

Kode Pengujian	RB-001	
Jumlah Pengguna	100	
Persentase Keberhasilan	100%	
<i>Response Time</i> (Detik)	Minimum	1,084
	Maksimum	2,343
	Rata-rata	1,760

5.3.2.2 Pengujian 500 Pengguna

Berdasarkan Tabel 5.50, pengujian 500 pengguna pada *server* menghasilkan tingkat keberhasilan 100 persen dan rata-rata *response time* 9,748. *Response Time* tersebut dapat dikatakan cepat dan dapat diandalkan.

Tabel 5.50. Hasil Pengujian Response Time 500 Pengguna

Kode Pengujian	RB-002	
Jumlah Pengguna	500	
Persentase Keberhasilan	100%	
Response Time (Detik)	Minimum	12,445
	Maksimum	4,556
	Rata-rata	9,748

5.3.2.3 Pengujian 1000 Pengguna

Berdasarkan Tabel 5.51, pengujian 1000 pengguna pada *server* menghasilkan tingkat keberhasilan 100 persen dan rata-rata *response time* 18,585 detik. *Response Time* tersebut dapat dikatakan lambat dan tidak dapat diandalkan.

Tabel 5.51. Hasil Pengujian Response Time 1000 Pengguna

Kode Pengujian	RB-003	
Jumlah Pengguna	1000	
Persentase Keberhasilan	100%	
Response Time (Detik)	Minimum	0,512
	Maksimum	26,497
	Rata-rata	18,585

5.3.2.4 Pengujian 2000 Pengguna

Berdasarkan Tabel 5.52, pengujian 2000 pengguna pada *server* menghasilkan tingkat keberhasilan 99,4 persen dan rata-rata *response time* 42,869 detik. *Response Time* tersebut dapat dikatakan sangat lambat. Tingkat keberhasilan 99,4 persen menandakan bahwa ada beberapa *request* yang gagal menerima *response* dari *server*.

Tabel 5.52. Hasil Pengujian Response Time 2000 Pengguna

Kode Pengujian	RB-004	
Jumlah Pengguna	2000	
Persentase Keberhasilan	99,4%	
Response Time (Detik)	Minimum	9,745

	Maksimum	77,167
	Rata-rata	42,869

5.4 Pengujian Kepuasan Pengguna

Pengujian dilakukan oleh responden yang merupakan *end user* dari sistem. Tujuan dari pengujian ini adalah untuk memastikan sistem sesuai dengan kebutuhan fungsional.

Bentuk pengujian yaitu survei kepada responden yang telah melakukan uji coba pada aplikasi MyITS-Dosen dan aplikasi MyITS-Mahasiswa. Penguji berjumlah 12 orang, yang terdiri dari dosen berjumlah 6 orang, asisten dosen berjumlah 3 orang dan mahasiswa berjumlah 3 orang. Proses uji coba dan survei dilakukan pada tanggal 7 Juli 2018 – 18 Juli 2018.

Penilaian pada survei ini menggunakan parameter angka dengan *range* 1 hingga 5 dengan keterangan seperti pada Tabel 5.53. Kuisisioner dibuat berdasarkan skenario uji coba yang terdapat pada Tabel 5.54. Aspek penilaian dalam survei ini, yaitu kemudahan penggunaan fitur. Rekapitulasi hasil penilaian dari kuisisioner oleh mahasiswa dan asisten dosen disajikan pada Tabel 5.55. Rekapitulasi hasil penilaian dari kuisisioner oleh dosen disajikan pada Tabel 5.56.

Tabel 5.53. Tingkat Penilaian

Angka	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Tabel 5.54. Skenario Uji Coba Kepuasan Pengguna

No.	Pernyataan
Melihat Data Kelas	
1.	Menurut saya, aplikasi ini memudahkan saya untuk melihat list kelas

2.	Menurut saya, aplikasi ini memudahkan saya untuk melihat halaman utama kelas
Mengelola Data Peserta Kelas	
3.	Menurut saya, aplikasi ini memudahkan saya untuk melihat list peserta kelas
4.	Menurut saya, aplikasi ini memudahkan saya untuk melihat jumlah mahasiswa di kelas
5.	Menurut saya, aplikasi ini memudahkan saya untuk mendaftarkan mahasiswa
6.	Menurut saya, aplikasi ini memudahkan saya untuk menghapus mahasiswa dari kelas
Mengelola Data Tugas	
7.	Menurut saya, aplikasi ini memudahkan saya untuk melihat list pengumpulan tiap tugas
8.	Menurut saya, aplikasi ini memudahkan saya untuk melihat list nilai tiap tugas
9.	Menurut saya, aplikasi ini memudahkan saya untuk menilai tugas
10.	Menurut saya, aplikasi ini memudahkan saya untuk memposting tugas
11.	Menurut saya, aplikasi ini memudahkan saya untuk mengedit tugas
12.	Menurut saya, aplikasi ini memudahkan saya untuk menghapus tugas
Mengelola Materi Perkuliahan	
13.	Menurut saya, aplikasi ini memudahkan saya untuk melihat materi perkuliahan
14.	Menurut saya, aplikasi ini memudahkan saya untuk memposting materi perkuliahan
15.	Menurut saya, aplikasi ini memudahkan saya untuk mengedit materi perkuliahan
16.	Menurut saya, aplikasi ini memudahkan saya untuk menghapus materi perkuliahan
Mengelola Detail Materi Perkuliahan	
17.	Menurut saya, aplikasi ini memudahkan saya untuk melihat dan mengunduh detail materi perkuliahan

18.	Menurut saya, aplikasi ini memudahkan saya untuk memposting detail materi perkuliahan
19.	Menurut saya, aplikasi ini memudahkan saya untuk mengedit detail materi perkuliahan
20.	Menurut saya, aplikasi ini memudahkan saya untuk menghapus detail materi perkuliahan
Melihat Tugas	
21	Menurut saya, aplikasi ini memudahkan saya untuk melihat detail tugas tiap minggu
22	Menurut saya, aplikasi ini memudahkan saya untuk melihat list tugas yang sudah dikumpulkan di kelas tersebut
Melihat Materi Perkuliahan	
23	Menurut saya, aplikasi ini memudahkan saya untuk melihat materi perkuliahan

Tabel 5.55. Rekapitulasi Hasil Penilaian

No	Pernyataan	Nilai Responden						
		1	2	3	4	5	6	Rata-Rata
1	Menurut saya, aplikasi ini memudahkan saya untuk melihat list kelas	4	4	4	5	5	5	4,6
2	Menurut saya, aplikasi ini memudahkan saya untuk melihat halaman utama kelas	4	4	4	5	5	5	4,5
3	Menurut saya, aplikasi ini memudahkan saya untuk melihat list peserta kelas	5	5	4	4	5	4	4,3
4	Menurut saya, aplikasi ini memudahkan saya untuk melihat jumlah mahasiswa di kelas	4	3	3	5	5	5	4,5
5	Menurut saya, aplikasi ini memudahkan saya untuk mendaftarkan mahasiswa	3	4	5	4	5	4	4,2

6	Menurut saya, aplikasi ini memudahkan saya untuk menghapus mahasiswa dari kelas	4	4	5	5	5	4	4,7
7	Menurut saya, aplikasi ini memudahkan saya untuk melihat list pengumpulan tiap tugas	4	5	4	5	5	4	4,5
8	Menurut saya, aplikasi ini memudahkan saya untuk melihat list nilai tiap tugas	4	4	3	4	3	4	3,9
9	Menurut saya, aplikasi ini memudahkan saya untuk menilai tugas	4	3	3	3	4	4	3,4
10	Menurut saya, aplikasi ini memudahkan saya untuk memposting tugas	4	5	4	4	4	4	3,8
11	Menurut saya, aplikasi ini memudahkan saya untuk mengedit tugas	4	5	5	4	5	5	4,3
12	Menurut saya, aplikasi ini memudahkan saya untuk menghapus tugas	4	5	5	5	5	5	4,9
13	Menurut saya, aplikasi ini memudahkan saya untuk melihat materi perkuliahan	5	5	5	5	5	5	4,4
14	Menurut saya, aplikasi ini memudahkan saya untuk memposting materi perkuliahan	4	5	4	5	5	4	3,8
15	Menurut saya, aplikasi ini memudahkan saya untuk mengedit materi perkuliahan	4	5	5	5	4	4	4,2
16	Menurut saya, aplikasi ini memudahkan saya untuk menghapus materi perkuliahan	4	5	5	5	5	5	4,8
17	Menurut saya, aplikasi ini memudahkan saya untuk melihat dan mengunduh detail materi perkuliahan	4	4	5	3	4	3	3,8

18	Menurut saya, aplikasi ini memudahkan saya untuk memposting detail materi perkuliahan	4	4	5	5	4	4	3,8
19	Menurut saya, aplikasi ini memudahkan saya untuk mengedit detail materi perkuliahan	4	5	5	4	4	4	3,9
20	Menurut saya, aplikasi ini memudahkan saya untuk menghapus detail materi perkuliahan	4	5	5	4	5	5	4,8
21	Menurut saya, aplikasi ini memudahkan saya untuk melihat detail tugas tiap minggu	5	5	4	5	4	5	4,7
22	Menurut saya, aplikasi ini memudahkan saya untuk melihat list tugas yang sudah dikumpulkan di kelas tersebut	4	5	5	5	3	4	4,3
23	Menurut saya, aplikasi ini memudahkan saya untuk melihat materi perkuliahan	4	5	5	5	5	5	4,8

Tabel 5.56. Rekapitulasi Hasil Penilaian Dosen

No	Pernyataan Dosen	Nilai Responden						Rata-Rata
		1	2	3	4	5	6	
1	Menurut saya, aplikasi ini memudahkan saya untuk melihat list kelas	4	5	4	5	5	5	4,6
2	Menurut saya, aplikasi ini memudahkan saya untuk melihat halaman utama kelas	4	5	4	5	4	5	4,5
3	Menurut saya, aplikasi ini memudahkan saya untuk melihat list peserta kelas	4	5	3	5	5	3	4,3
4	Menurut saya, aplikasi ini memudahkan saya untuk	5	5	5	5	5	4	4,5

	melihat jumlah mahasiswa di kelas							
5	Menurut saya, aplikasi ini memudahkan saya untuk mendaftarkan mahasiswa	5	5	2	5	5	3	4,2
6	Menurut saya, aplikasi ini memudahkan saya untuk menghapus mahasiswa dari kelas	5	5	5	5	4	5	4,7
7	Menurut saya, aplikasi ini memudahkan saya untuk melihat list pengumpulan tiap tugas	5	5	3	5	5	4	4,5
8	Menurut saya, aplikasi ini memudahkan saya untuk melihat list nilai tiap tugas	4	5	3	5	5	3	3,9
9	Menurut saya, aplikasi ini memudahkan saya untuk menilai tugas	4	2	3	4	4	3	3,4
10	Menurut saya, aplikasi ini memudahkan saya untuk memposting tugas	4	2	2	5	4	3	3,8
11	Menurut saya, aplikasi ini memudahkan saya untuk mengedit tugas	5	4	2	5	5	3	4,3
12	Menurut saya, aplikasi ini memudahkan saya untuk menghapus tugas	5	5	5	5	5	5	4,9
13	Menurut saya, aplikasi ini memudahkan saya untuk melihat materi perkuliahan	4	2	4	5	5	3	4,4
14	Menurut saya, aplikasi ini memudahkan saya untuk memposting materi perkuliahan	4	2	2	5	4	2	3,8
15	Menurut saya, aplikasi ini memudahkan saya untuk mengedit materi perkuliahan	4	4	2	5	5	3	4,2
16	Menurut saya, aplikasi ini memudahkan saya untuk	4	5	5	5	5	5	4,8

	menghapus materi perkuliahan							
17	Menurut saya, aplikasi ini memudahkan saya untuk melihat dan mengunduh detail materi perkuliahan	5	2	2	5	5	3	3,8
18	Menurut saya, aplikasi ini memudahkan saya untuk memposting detail materi perkuliahan	4	2	3	4	4	3	3,8
19	Menurut saya, aplikasi ini memudahkan saya untuk mengedit detail materi perkuliahan	4	2	3	5	5	2	3,9
20	Menurut saya, aplikasi ini memudahkan saya untuk menghapus detail materi perkuliahan	4	5	5	5	5	5	4,8

5.5 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional, pengujian kualitas program dan pengujian REST API.

5.5.1 Evaluasi Pengujian Fungsionalitas

Hasil pengujian fungsionalitas secara keseluruhan dapat dilihat di Tabel 5.57. Data pada tabel tersebut menunjukkan bahwa semua kebutuhan fungsional telah berhasil diimplementasikan dan berjalan sesuai dengan skenario yang telah dirancang. Sehingga dapat ditarik kesimpulan bahwa aplikasi bekerja sesuai dengan yang diharapkan.

Tabel 5.57. Hasil Pengujian Fungsionalitas

Kode Pengujian Fungsionalitas	Kasus Penggunaan	Hasil
SCF-001	Melihat <i>List</i> Kelas	Berhasil
SCF-002	Melihat Halaman Utama Kelas	Berhasil

SCF-003	Melihat Halaman Data Kelas	Berhasil
SCF-004	Menambah Data Kelas	Berhasil
SCF-005	Menghapus Data Kelas	Berhasil
SCF-006	Menambah Materi Perkuliahan	Berhasil
SCF-007	Melihat Detail Materi Perkuliahan	Berhasil
SCF-008	<i>Edit</i> Materi Perkuliahan	Berhasil
SCF-009	Menghapus Materi Perkuliahan	Berhasil
SCF-010	Menambah Tugas	Berhasil
SCF-011	Melihat Detail Tugas	Berhasil
SCF-012	<i>Edit</i> Tugas	Berhasil
SCF-013	Menghapus Tugas	Berhasil
SCF-014	Melihat <i>List</i> Pengumpulan Tugas	Berhasil
SCF-015	Menilai Tugas	Berhasil
SCF-016	Melihat <i>List</i> Nilai Tugas	Berhasil
SCF-017	Menambah Detail Materi Perkuliahan	Berhasil
SCF-018	<i>Edit</i> Detail Materi Perkuliahan	Berhasil
SCF-019	Menghapus Detail Materi Perkuliahan	Berhasil

5.6 Evaluasi Pengujian REST API

Hasil Pengujian *base response* REST API dapat dilihat pada Tabel 5.58. Berdasarkan tabel tersebut, hasil pengujian menunjukkan bahwa semua skenario pengujian untuk setiap *endpoint* berhasil dilakukan. Sehingga dapat ditarik kesimpulan bahwa setiap *endpoint* REST API memiliki *base response* yang sesuai.

Tabel 5.58. Hasil Pengujian Base Response

Kode Pengujian Response Body	Nama Endpoint	Hasil Pengujian Skenario			
		A	B	C	D
		RA-001	GET <i>List Class Per User</i>	Berhasil	Berhasil
RA-002	GET <i>Detail Class</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-003	GET <i>List User Per Class</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-004	GET Jumlah Mahasiswa Di Kelas	Berhasil	Berhasil	Berhasil	Berhasil
RA-005	GET <i>Detail User From SIAKAD</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-006	POST <i>User to Moodle and Class</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-007	DELETE <i>User from Class</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-008	GET <i>List Pengumpulan Tugas</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-009	GET <i>List Nilai Tugas</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-010	POST <i>Nilai Tugas</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-011	POST <i>Tugas</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-012	EDIT <i>Tugas</i>	Berhasil	Berhasil	Berhasil	Berhasil
RA-013	DELETE <i>Tugas</i>	Berhasil	Berhasil	Berhasil	Berhasil

RA-014	GET Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-015	POST Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-016	EDIT Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-017	DELETE Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-018	GET Detail Tugas	Berhasil	Berhasil	Berhasil	Berhasil
RA-019	GET <i>List</i> Tugas yang Sudah Dikumpulkan	Berhasil	Berhasil	Berhasil	Berhasil
RA-020	GET Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-021	GET Detail Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-022	GET <i>Files Per</i> Detail Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-023	POST Detail Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-024	EDIT Detail Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil
RA-025	DELETE Detail Materi Perkuliahan	Berhasil	Berhasil	Berhasil	Berhasil

Perbandingan hasil pengujian *response time* REST API dapat dilihat pada Tabel 5.59. Berdasarkan data di tabel tersebut, tingkat keberhasilan pengujian semakin menurun ketika jumlah pengguna bertambah. Pada pengujian dengan jumlah pengguna 100, 500 dan 1000 memiliki tingkat keberhasilan sempurna. Sedangkan jumlah pengguna 2000 masih memiliki tingkat keberhasilan yang bagus yaitu 99,4 persen karena *error* lebih kecil

dari 1 persen. Dalam segi *response time*, jumlah pengguna 100 memiliki *response time* yang sangat cepat yaitu 1,760 detik untuk setiap *request*. Sedangkan jumlah pengguna 500 dan 1000 memiliki *response time* di rentang 9 hingga 19 detik. *Response time* untuk jumlah pengguna 2000 memiliki waktu 42 detik, tiga kali lipat dari jumlah pengguna 1000. Dapat dikatakan bahwa *response time* dari jumlah pengguna 500, 1000 dan 2000 sangat tinggi dan tidak dapat diandalkan. *Response time* yang tinggi akan menghambat penggunaan aplikasi Android dan membuat penggunaan aplikasi tidak efisien. Berdasarkan analisis di atas, dapat disimpulkan bahwa peningkatan jumlah pengguna dalam mengakses REST API dalam waktu yang hampir bersamaan membuat tingkat keberhasilan lebih rendah dan *response time* yang lebih lama. Hal ini terjadi karena spesifikasi *server* yang sangat rendah dan hanya menggunakan RAM sebesar 8 GB dan *shared virtual CPU*.

Tabel 5.59. Perbandingan Hasil Pengujian *Response Time*

Kode Pengujian <i>Response Time</i>	Jumlah Pengguna	Tingkat Keberhasilan	Rata-Rata <i>Response Time</i> (Detik)
RB-001	100	100%	1,760
RB-002	500	100%	9,748
RB-003	1000	100%	18,585
RB-004	2000	99,4%	42,869

5.7 Evaluasi Pengujian Kepuasan Pengguna

Berdasarkan respon dari survei pengujian kepuasan pengguna dengan hasil rata-rata nilai pada Tabel 5.60, dapat disimpulkan bahwa kesesuaian kebutuhan fungsionalitas dengan aplikasi yang diujicobakan sudah sesuai. Hal ini dapat dilihat pada Tabel 5.61 dengan penjelasan bahwa nilai dari rata-rata penilaian adalah 4,3 dari 5 yang mewakili keterangan setuju dengan tingkat rata-rata penilaian 86%.

Tabel 5.60. Rata-Rata Nilai Responden

Pernyataan	Rata-Rata Nilai	Rata-Rata (%)
1	4,6	92
2	4,5	90
3	4,3	87
4	4,5	90
5	4,2	83
6	4,7	93
7	4,5	90
8	3,9	78
9	3,4	68
10	3,8	75
11	4,3	87
12	4,9	98
13	4,4	88
14	3,8	77
15	4,2	83
16	4,8	97
17	3,8	75
18	3,8	77
19	3,9	78
20	4,8	95
21	4,7	93
22	4,3	87
23	4,8	97

Tabel 5.61. Rata-Rata Total Nilai

Rata-Rata Total Nilai	4,3
Rata-Rata Persentase Total (%)	86

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil pengujian yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditunjukkan untuk pengembangan perangkat lunak di masa mendatang.

6.1 Kesimpulan

Kesimpulan dari tugas akhir ini adalah:

1. Konsep *REST API* untuk fitur manajemen perkuliahan pada *platform e-Learning* yang sudah ada berhasil diimplementasikan melalui pembuatan API yang mengambil data Moodle untuk disajikan ke aplikasi MyITS.
2. Penerapan API untuk mengambil data mahasiswa dari SIAKAD ITS berhasil diterapkan pada *endpoint* `GET /user/akademik` yang akan mengirim *response* ke aplikasi MyITS Dosen.
3. Pengimplementasian fitur manajemen perkuliahan aplikasi *e-Learning* pada perangkat bergerak Android menggunakan paradigma *Reactive Programming* berhasil diterapkan dengan menggunakan *library* RxAndroid dan arsitektur *Model View Presenter* (MVP).

6.2 Saran

1. Perlu pengembangan pada fitur manajemen perkuliahan. Pengembangan dapat terpusat pada penambahan tingkatan aktor, seperti *course creator* dan *manager*. Peran dari *course creator* adalah bertugas untuk mengatur perkuliahan, kurikulum, dan aturan-aturan perkuliahan. Sedangkan peran *manager* seperti kepala departemen yaitu berperan sebagai pengambil keputusan tertinggi kepada setiap keputusan akademik, tanpa perlu *join* ke masing-masing kelas.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] StatCounter, "Mobile Operating System Market Share in Indonesia - May 2018," StatCounter , 1999-2017. [Online]. Available: <http://gs.statcounter.com/os-market-share/mobile/indonesia>. [Accessed 5 Juni 2018].
- [2] Android Studio, "Mengenal Android Studio," Android Studio, [Online]. Available: <https://developer.android.com/studio/intro/index.html?hl=id>. [Accessed 8 June 2017].
- [3] B. Mulloy, "Web API Design," Apigee, March 2012. [Online]. Available: <https://pages.apigee.com/rs/apigee/images/api-design-ebook-2012-03.pdf>. [Accessed 8 June 2017].
- [4] MoodleDocs, "Moodle Docs 3.3," Moodle, 21 April 2017. [Online]. Available: https://docs.moodle.org/33/en/Main_page. [Accessed 16 Mei 2018].
- [5] RxAndroid, "ReactiveX RxAndroid," GitHub, 2015. [Online]. Available: <https://github.com/ReactiveX/RxAndroid>. [Accessed 15 June 2017].
- [6] M. Riadi, "Pengertian, Karakteristik dan Manfaat E-Learning," KajianPustaka.com, 17 Juni 2014. [Online]. Available: <https://www.kajianpustaka.com/2014/06/pengertian-karakteristik-dan-manfaat-elearning.html>. [Accessed 10 Juli 2018].
- [7] A. Sanjaya, "Pengertian E-learning Definisi Manfaat, Komponen, Kelebihan dan kekurangan serta Filosofinya," Landasan Teori, 28 Mei 2016. [Online].

Available:

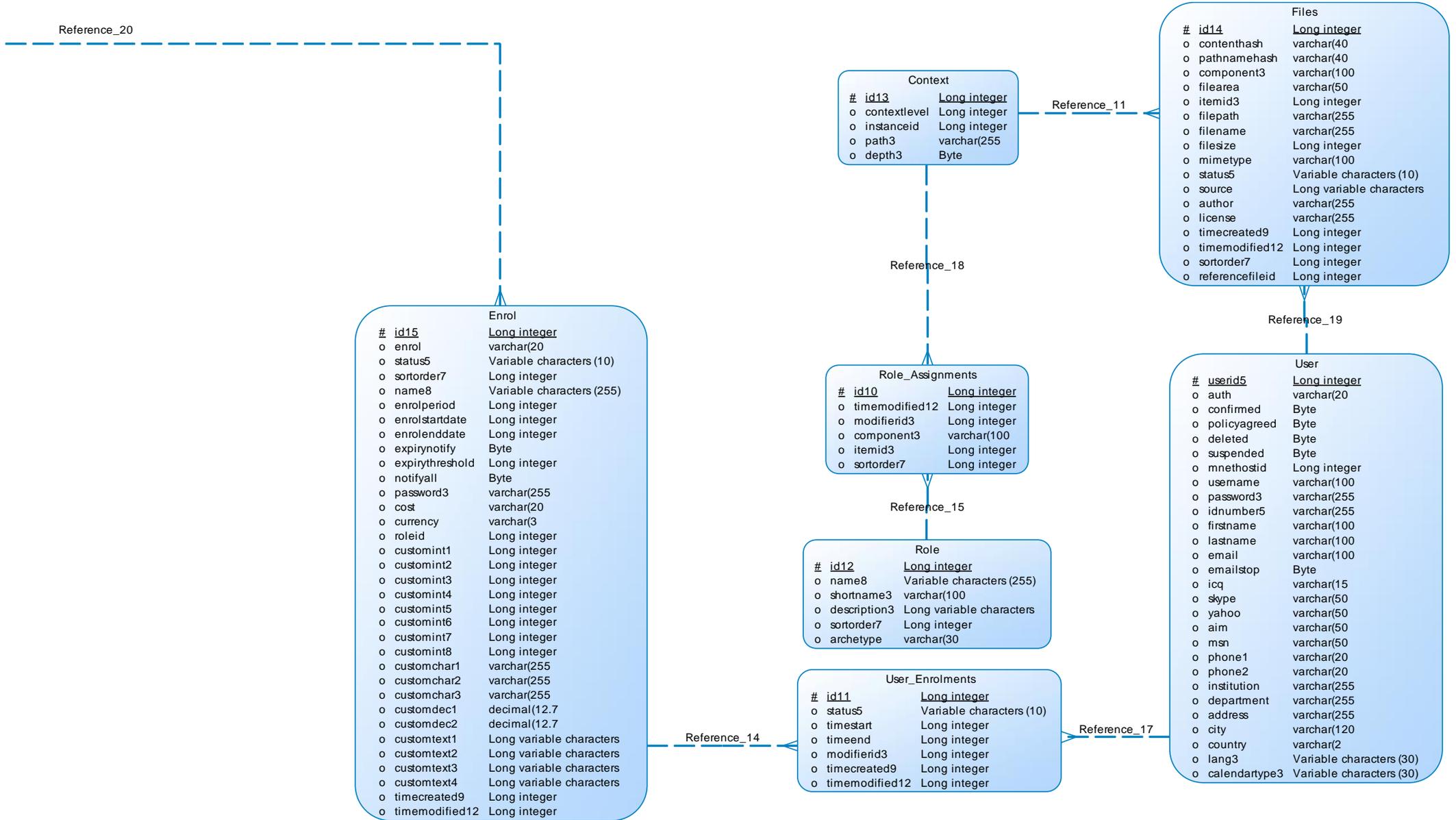
<http://www.landasanteori.com/2015/09/pengertian-e-learning-definisi-manfaat.html>. [Accessed 10 Juli 2018].

- [8] p3ai@its.ac.id, *Buku Pedoman Pengembangan & Penyelenggaraan Pembelajaran Berbasis Share ITS*, Surabaya: Pusat Pengembangan Pendidikan dan Aktivitas Instruksional, 2013.
- [9] Ecma International 2017, "The JSON Data Interchange Standard.," *ECMAScript® 2017 Language Specification*, 2017.
- [10] Sun Microsystems, Inc., "The Java Language Environment," Sun Microsystems, Inc., 1997. [Online]. Available: <http://www.oracle.com/technetwork/java/intro-141325.html>. [Accessed 8 June 2017].
- [11] N. Langley, "Write once, run anywhere?," *Computer Weekly*, [Online]. Available: <http://www.computerweekly.com/feature/Write-once-run-anywhere>. [Accessed 8 June 2017].
- [12] Feridi, "Mengenal RESTful Web Services," *CODEPOLITAN*, 25 Februari 2016. [Online]. Available: <https://www.codepolitan.com/mengenal-restful-web-services>. [Accessed 24 November 2017].
- [13] Black Duck Software, Inc. , "Android," *Open Hub*, 25 Oktober 2017. [Online]. Available: <https://www.openhub.net/p/android>. [Accessed 6 Juni 2018].
- [14] PCWorld, "Original Android Prototype Revealed During Google, Oracle Trial," *PCWorld*, 26 April 2012. [Online]. Available:

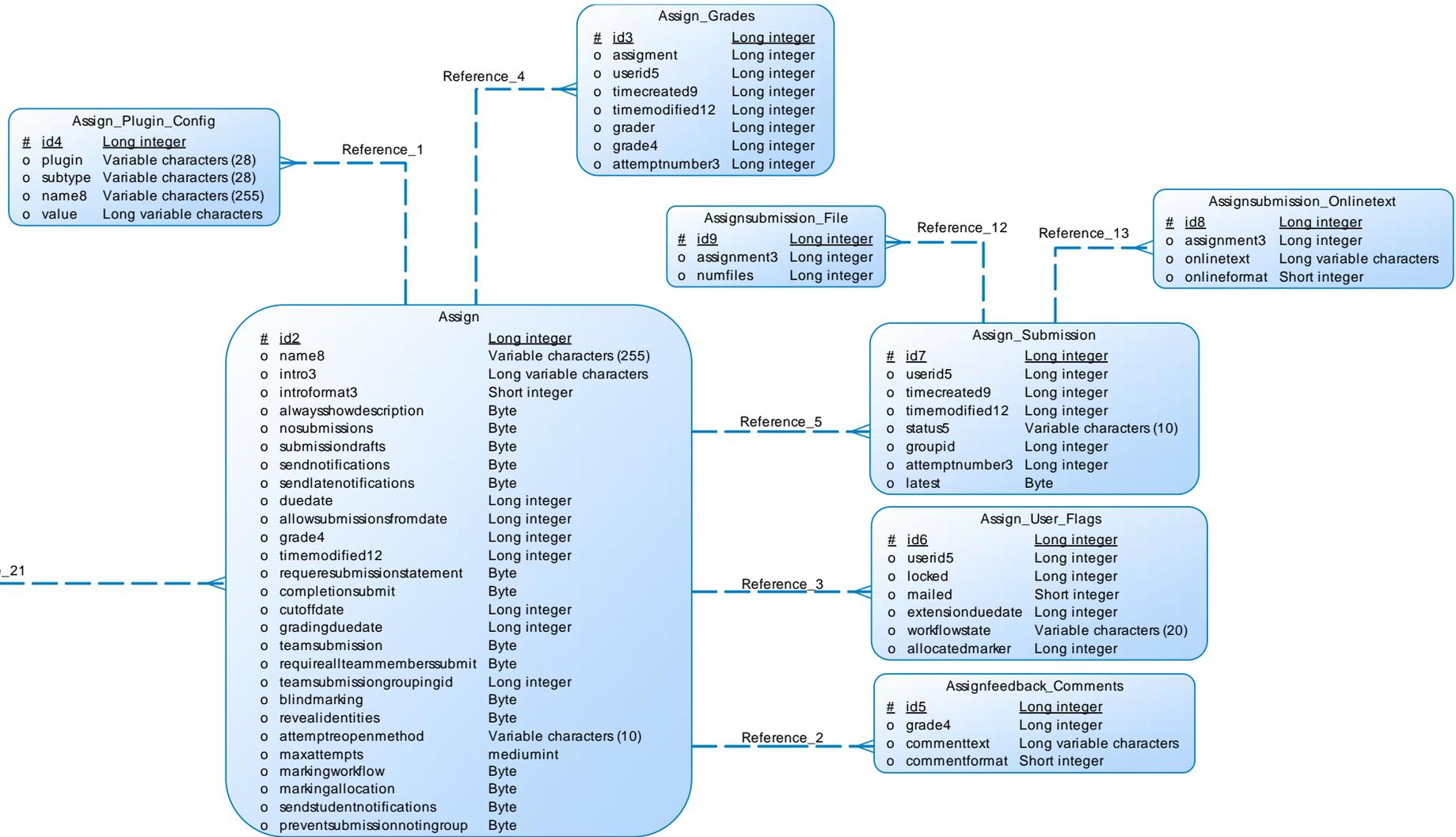
- http://www.pcworld.com/article/254539/original_android_prototype_revealed_during_google_oracle_trial.html. [Accessed 8 June 2017].
- [15] Open Source Projects, "ReactiveX," [Online]. Available: <http://reactivex.io/intro.html>. [Accessed 15 June 2017].
- [16] Firmansyah, "Pengertian MySQL Beserta Fungsi dan Sejarah Terbentuknya MySQL Secara Lengkap," Nesaba Media Copyright © 2018., 2018. [Online]. Available: <https://www.nesabamedia.com/pengertian-mysql/>. [Accessed 6 Juni 2018].
- [17] MoodleDocs, "Non-editing teacher role - MoodleDocs," [Online]. Available: https://docs.moodle.org/32/en/Non-editing_teacher_role. [Accessed 8 June 2017].
- [18] MINDORKS NEXTGEN PRIVATE LIMITED, "Android MVP Architecture: Sample App," Github, 3 April 2017. [Online]. Available: <https://github.com/MindorksOpenSource/android-mvp-architecture>. [Accessed 6 Juni 2018].
- [19] A. Shekhar, "The Best Android Networking Library for Fast and Easy Networking," Mindorks, 3 Juli 2016. [Online]. Available: <https://medium.com/mindorks/simple-and-fast-android-networking-19ed860d1455>. [Accessed 21 Mei 2018].
- [20] Lars Vogel, (c) 2009, 2017 vogella GmbH, "REST with Java (JAX-RS) using Jersey - Tutorial," Vogella, 27 November 2017. [Online]. Available: <http://www.vogella.com/tutorials/REST/article.html>. [Accessed 20 Mei 2018].

- [21] andra49, "WSO2 API Manager Tutorial," GitHub, 17 Desember 2014. [Online]. Available: <https://github.com/if-itb/IF4050-OpenBravo-Tugas2-WSO2-Identity-Server-API-Manager/wiki/WSO2-API-Manager-Tutorial>. [Accessed 20 Mei 2018].
- [22] A. Shekhar, "The New Dagger 2 Android Injector," Mindorks, 29 Desember 2017. [Online]. Available: <https://blog.mindorks.com/the-new-dagger-2-android-injector-cbe7d55afa6a>. [Accessed 31 Mei 2018].

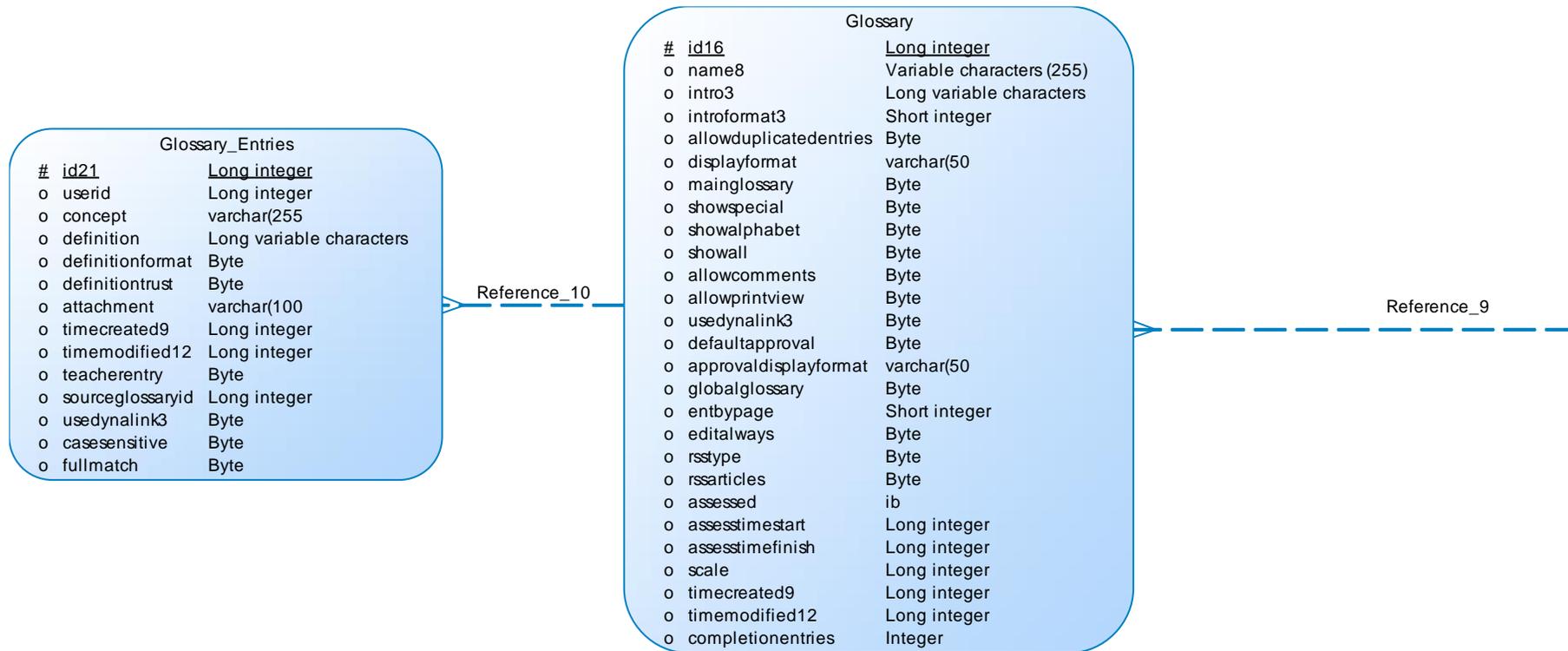
LAMPIRAN



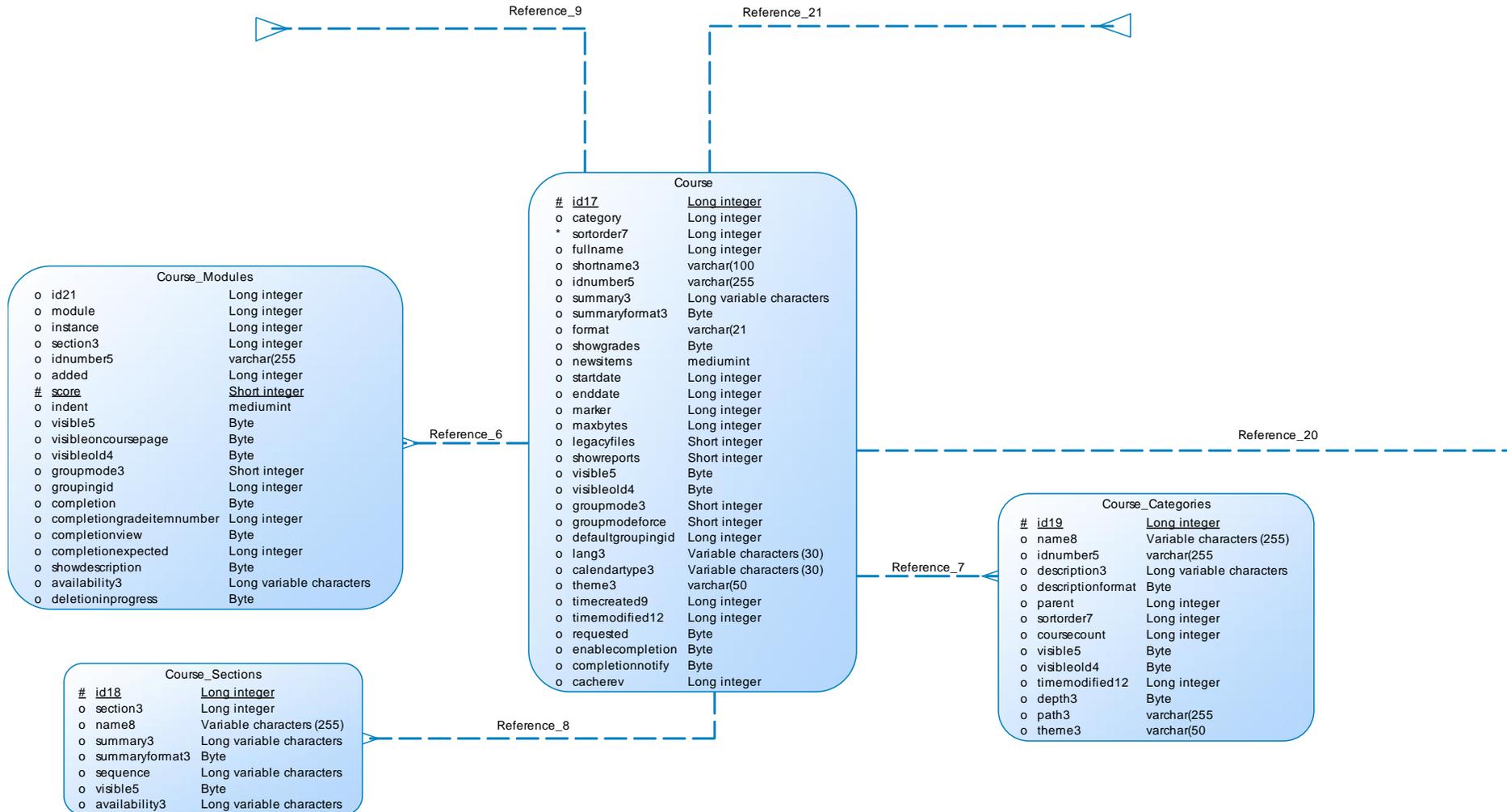
Gambar 8.1. Conceptual Data Model User



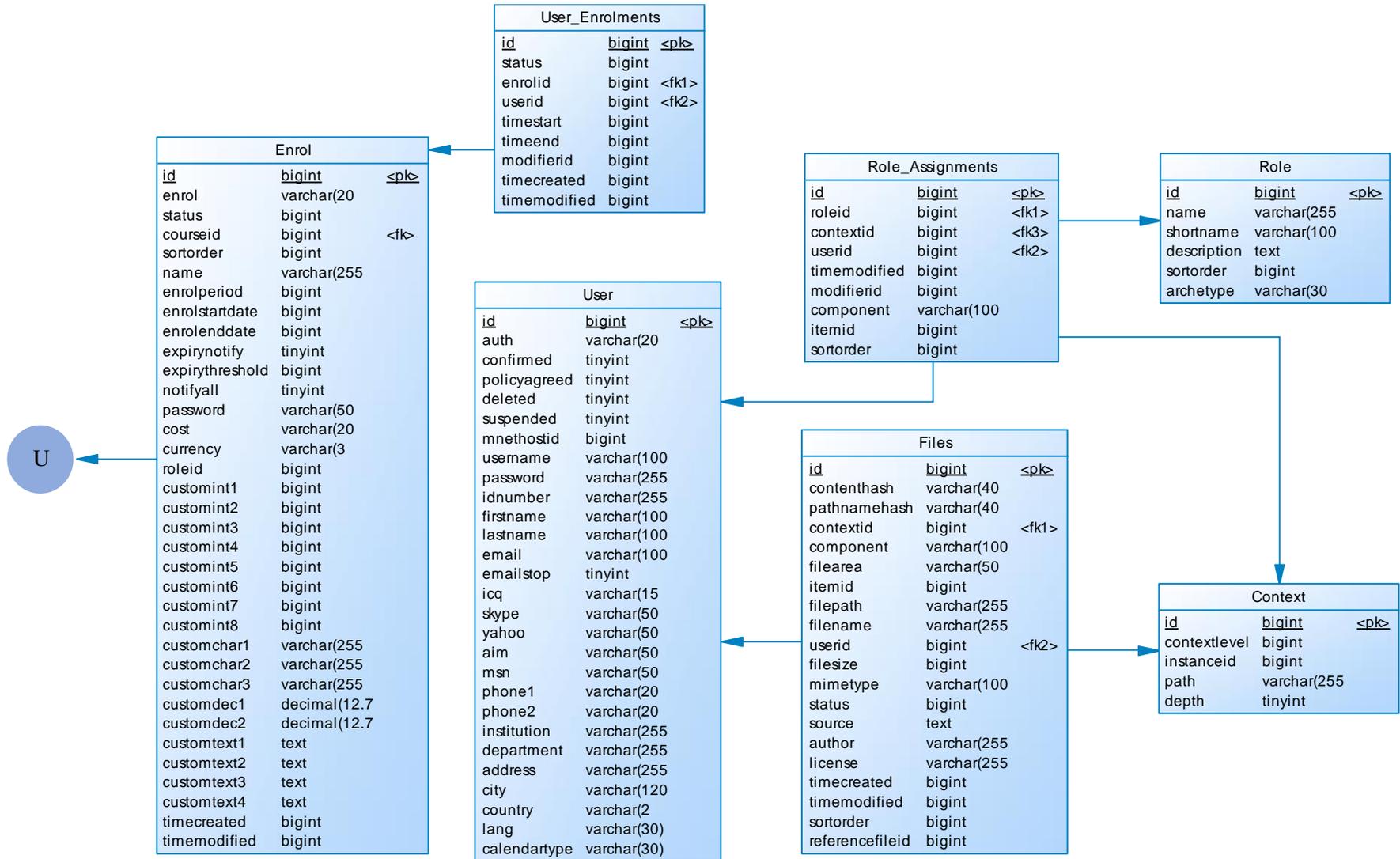
Gambar 8.2. Conceptual Data Model Assign



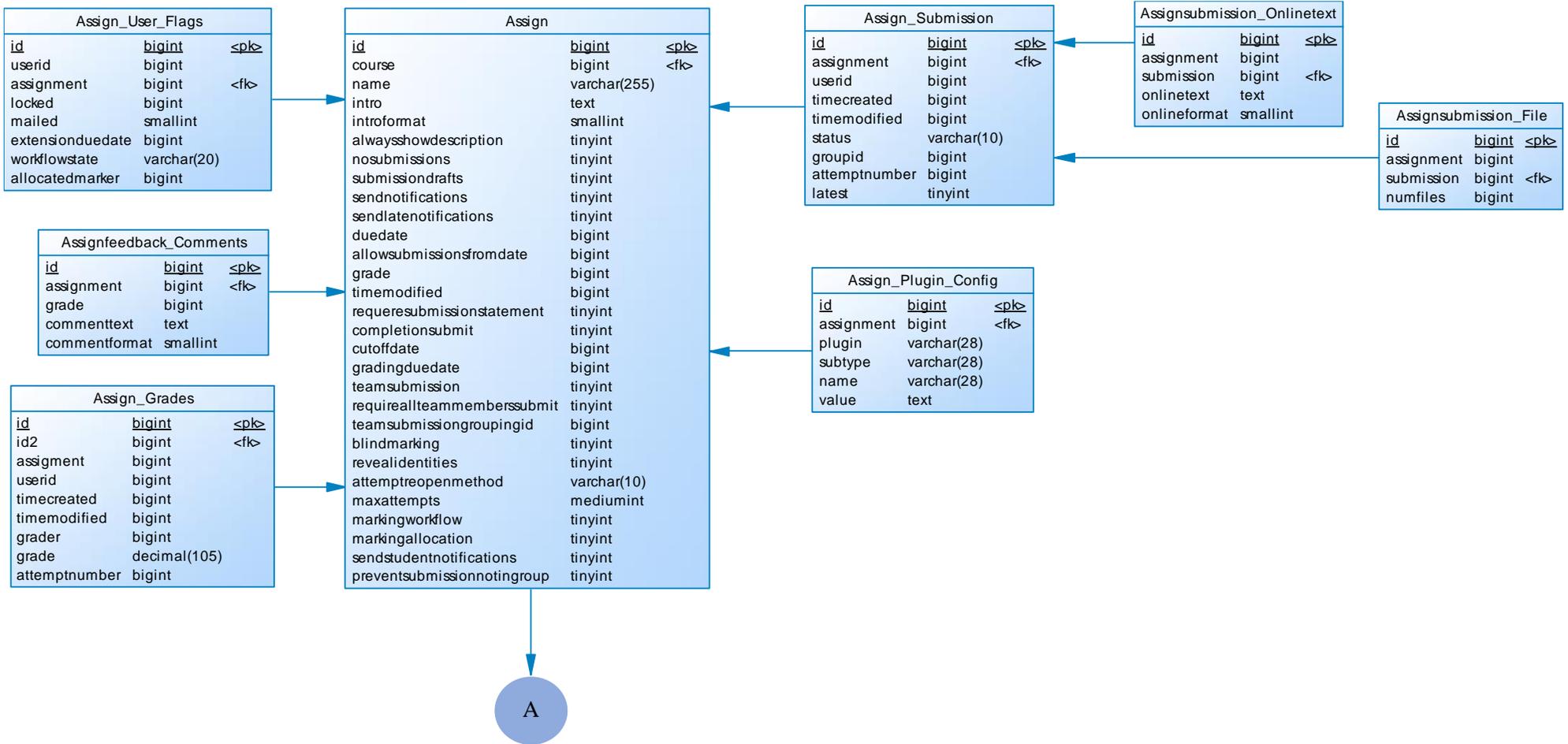
Gambar 8.3. Conceptual Data Model Glossary



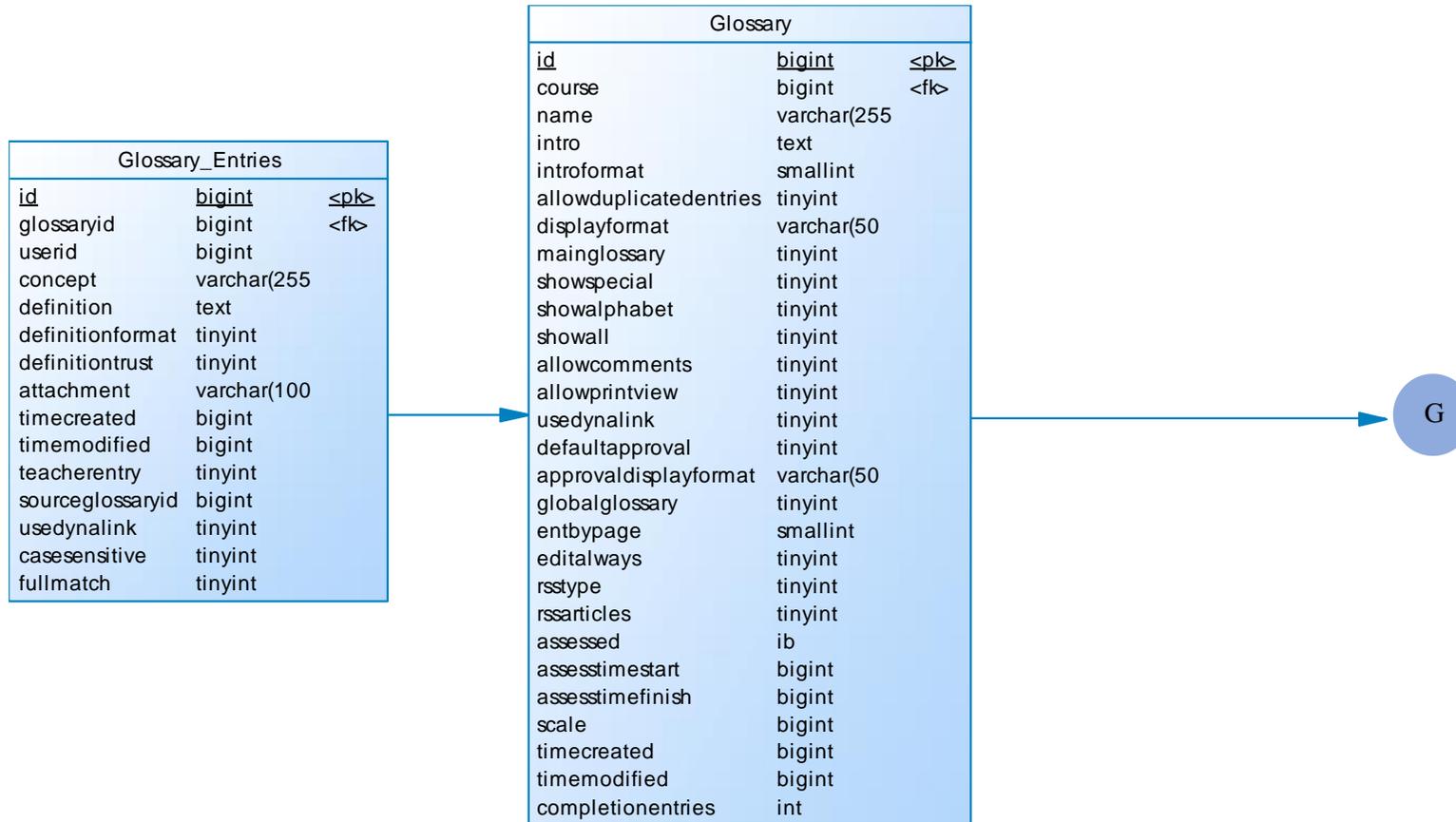
Gambar 8.4. Conceptual Data Model Course



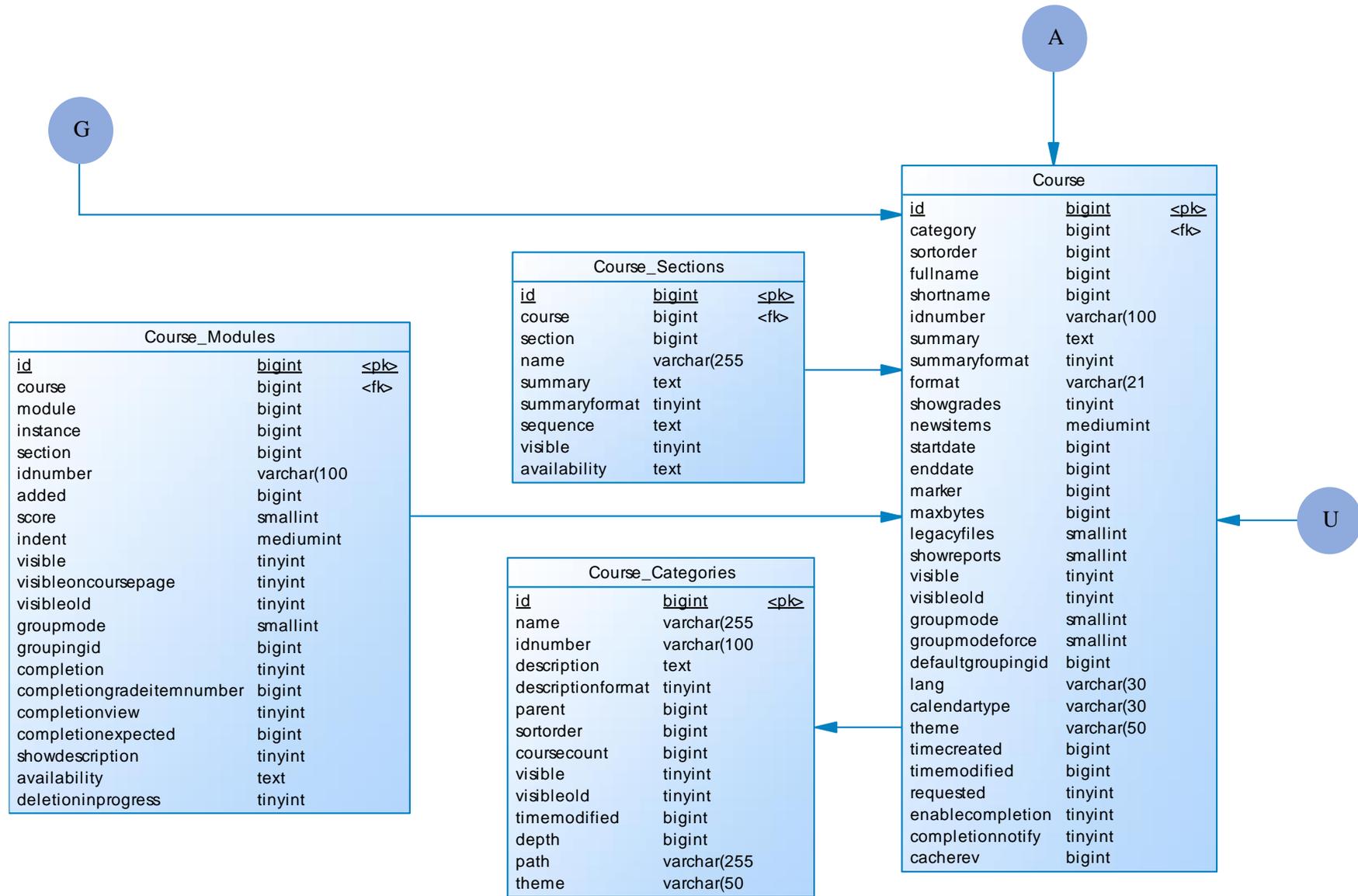
Gambar 8.5. Physical Data Model User



Gambar 8.6. Physical Data Model Assign



Gambar 8.7. Physical Data Model Glossary



Gambar 8.8. Physical Data Model Course

BIODATA PENULIS



William Suhud atau yang biasa dipanggil dengan Suhud di lingkungan kampus, lahir pada tanggal 18 Januari 1996 di Surabaya. Penulis menempuh Pendidikan SMA di SMA Kr. Petra 5 Surabaya dan S1 Teknik Informatika Fakultas Teknologi Informasi dan Komunikasi (FTIK) di Institut Teknologi Sepuluh Nopember Surabaya. Selama menempuh Pendidikan di kampus, penulis aktif dalam organisasi kemahasiswaan, antara lain Staf Departemen Hubungan Luar Himpunan Mahasiswa Teknik Computer-Informatika (HMTTC), Staf Keamanan dan Perizinan Schematics, Staf Dana dan Usaha Schematics, Staf Departemen External Affairs Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf), Staf bidang Publikasi Perlengkapan Dokumentasi Dekorasi Transportasi (Publengdekdoktrans) Big Event Persekutuan Mahasiswa Kristen (PMK) ITS, Sekretaris Departemen External Affairs Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf). Penulis dapat dihubungi melalui alamat surel **williamsuhud@gmail.com**