



UNIVERSITAS DIPONEGORO

**PEMBUATAN RESTFUL API DAN BACKEND ANDROID
UNTUK SISTEM PENGELOLAAN CAPSTONE TUGAS AKHIR
(Studi Kasus di Departemen Teknik Komputer Universitas Diponegoro)**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

NANDA IQBAL HANAFI

21120120130109

DEPARTEMEN TEKNIK KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS DIPONEGORO

SEMARANG

2024

HALAMAN PENGESAHAN

Tugas akhir ini diajukan oleh:

Nama : Nanda Iqbal Hanafi
NIM : 21120120130109
Departemen : Teknik Komputer
Judul Tugas Akhir : Pembuatan RESTful API dan Backend Android untuk Sistem
Pengelolaan Capstone Tugas Akhir (Studi Kasus di
Departemen Teknik Komputer Universitas Diponegoro)

Telah berhasil dipertahankan di hadapan Tim Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Departemen Teknik Komputer, Fakultas Teknik, Universitas Diponegoro.

TIM PENGUJI

Pembimbing I : Adnan Fauzi, S.T., M.Kom. ()
Pembimbing II : Patricia Evericho Mountaines, S.T., M.Cs. ()
Ketua Penguji : ()
Anggota Penguji : ()

Semarang,

Ketua Departemen Teknik Komputer

Prof. Dr. Adian Fatchur Rochim S.T., M.T.

NIP. 197302261998021001

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.

Nama : Nanda Iqbal Hanafi

NIM : 21120120130109

Tanda Tangan :

Tanggal :

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Diponegoro, saya yang bertanda tangan di bawah ini:

Nama : Nanda Iqbal Hanafi
NIM : 21120120130109
Departemen : TEKNIK KOMPUTER
Fakultas : TEKNIK
Jenis Karya : TUGAS AKHIR

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya berjudul:

Pembuatan RESTful API dan Backend Android untuk Sistem Pengelolaan Capstone Tugas Akhir (Studi Kasus di Departemen Teknik Komputer Universitas Diponegoro) beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan memublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang

Pada tanggal : 21 Februari 2024

Yang menyatakan,

(Nanda Iqbal Hanafi)

KATA PENGANTAR

Puji syukur penulis panjatkan atas ke hadirat Allah SWT, karena atas berkat, rahmat, dan karunia-Nya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“Pembuatan RESTful API dan Backend Android untuk Sistem Pengelolaan Capstone Tugas Akhir (Studi Kasus di Departemen Teknik Komputer Universitas Diponegoro)”**.

Laporan Tugas Akhir ini dibuat sebagai syarat kelulusan serta untuk memenuhi kewajiban di Departemen Teknik Komputer, Fakultas Teknik, Universitas Diponegoro. Dalam penyusunan laporan Tugas Akhir ini, penulisan tentunya ingin mengucapkan terima kasih yang sebesar-besarnya kepada beberapa pihak yang telah memberikan dukungan, bimbingan, bantuan, doa, dan arahan. Oleh karena itu, melalui kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Allah SWT, yang telah memberikan penulis kesehatan, kekuatan dan anugrah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir penulis tepat waktu.
2. Kedua orang tua, khususnya ibu penulis yang telah memberikan dukungan secara moral dan materi, kasih sayang, doa, nasehat, serta atas kesabaran yang luar biasa dalam setiap langkah hidup penulis.
3. Kepada kedua saudara penulis, Sigit Hanafi dan Dimas Hanafi yang telah membagikan banyak pengalaman perkuliahan dan nilai-nilai kehidupan kepada penulis.
4. Bapak Adnan Fauzi, S.T., M.Kom selaku dosen pembimbing I yang telah memberikan bimbingan, saran, arahan, dan waktu kepada penulis selama pembuatan Tugas Akhir ini.
5. Ibu Patricia Evericho Mountaines, S.T., M.Cs. selaku dosen pembimbing II yang telah memberikan bimbingan, saran, arahan, dan waktu kepada penulis selama pembuatan Tugas Akhir ini.
6. Bapak Dr. Adian Fatchur Rochim S.T., M.T. selaku Ketua Departemen Teknik Komputer Universitas Diponegoro.

7. Bapak Ilmam Fauzi Hashbil Alim S.T., M.Kom. selaku Koordinator Tugas Akhir.
8. Seluruh Bapak dan Ibu dosen Jurusan Teknik Komputer yang telah memberikan ilmunya kepada penulis.
9. Seluruh Staff Tata Usaha dan/atau Tenaga Kependidikan Departemen Teknik Komputer yang telah bekerja dengan baik.
10. Teman-teman terdekat, khususnya teman-teman *Rogue* yang telah mendukung, membantu, dan menemani penulis dalam menjalani masa perkuliahan.
11. Teman-teman Jurusan Teknik Komputer angkatan 2020 dan tim *Capstone* yang turut serta memberikan dukungan dan menemani selama proses kuliah hingga Tugas Akhir ini.
12. Muhammad Rofi'ul Anam dan Maulana Yusuf Suradin yang telah membantu dan menemani penulis dalam menyelesaikan Tugas Akhir ini.
13. Khanuun Maulida Puspita Hasyim yang dengan hati baiknya selalu memberikan dukungan dan motivasi secara tulus untuk menyelesaikan Tugas Akhir ini.
14. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu hingga selesainya Tugas Akhir ini.

Penulis menyadari bahwa dalam pembuatan Tugas Akhir ini masih banyak kekurangan. Tugas Akhir ini masih sangat membutuhkan kritik, saran, serta masukan yang dapat membantu agar Tugas Akhir ini berkembang lebih baik lagi. Penyampaian terima kasih ini akan penulis akhiri dengan harapan semoga laporan Tugas Akhir ini dapat bermanfaat dan menjadi motivasi serta inspirasi bagi pembaca.

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
ABSTRAK	xiii
ABSTRACT.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	6
BAB II KAJIAN PUSTAKA	7
2.1 Kajian Penelitian Terdahulu.....	7
2.2 Landasan Teori.....	13
2.2.1 <i>Capstone</i> TA	13

2.2.2	Sistem Informasi	15
2.2.3	RESTful API	15
2.2.4	MySQL.....	15
2.2.5	PHP	16
2.2.6	Laravel.....	16
2.2.7	Postman	17
2.2.8	Android	17
2.2.9	Android Studio.....	18
2.2.10	Kotlin	18
2.2.11	<i>Model View ViewModel</i>	19
2.2.12	Metode RAD	19
2.2.13	<i>Unit Testing</i>	20
2.2.14	Black Box Testing.....	20
2.2.15	<i>Load Testing</i>	21
BAB III PERANCANGAN SISTEM		22
3.1	<i>Analysis and Quick Design</i>	23
3.1.1	Kebutuhan Fungsional Sistem	23
3.1.2	Kebutuhan Non Fungsional Sistem.....	24
3.1.3	Karakteristik Pengguna	24
3.1.4	Arsitektur Sistem.....	25
3.1.5	Antarmuka Eksternal.....	26
3.2	Menentukan Target Sistem	27
3.2.1	Ruang Lingkup Sistem.....	27
3.2.2	Proses Bisnis Sistem	27
3.3	Perancangan Sistem	28
3.3.1	Perancangan Diagram Alur	28
3.3.2	Perancangan Diagram <i>Use Case</i>	30

3.3.3 Perancangan Skenario <i>Use Case</i>	31
3.3.6 Perancangan Basis Data	37
3.3.7 Perancangan <i>End-Point</i>	46
3.4 Metode Pengujian.....	54
BAB IV HASIL DAN PEMBAHASAN	55
BAB V PENUTUP.....	56
5.1. Kesimpulan	56
5.2. Saran.....	56
DAFTAR PUSTAKA	57
LAMPIRAN 1 BIODATA MAHASISWA	60

DAFTAR GAMBAR

Gambar 3.1 Metode <i>Rapid Application Development</i> dalam pengembangan API dan <i>backend</i> Android	22
Gambar 3.2 Arsitektur sistem berbasis Android	25
Gambar 3.3 Arsitektur pengembangan aplikasi Android.....	26
Gambar 3.4 Diagram alur proses mahasiswa masuk ke aplikasi	29
Gambar 3.5 Diagram alur proses mahasiswa mengakses pengumuman.....	29
Gambar 3.6 Diagram <i>use case</i>	30
Gambar 3. 7 ERD sistem.....	38

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian Dharmawan dkk. dengan penelitian penulis	7
Tabel 2.2 Perbandingan penelitian Sari dkk. dengan penelitian penulis	8
Tabel 2.3 Perbandingan penelitian Lander dkk. dengan penelitian penulis.....	8
Tabel 2.4 Perbandingan penelitian Nurwicaksono dkk. dengan penelitian penulis.....	9
Tabel 2.5 Perbandingan penelitian Mardiani dkk dengan penelitian penulis	9
Tabel 2.6 Perbandingan penelitian Ridhoi dkk dengan penelitian penulis	9
Tabel 2.7 Perbandingan penelitian Riyadhhi dkk dengan penelitian penulis	10
Tabel 2. 8 Perbandingan penelitian Padang dkk dengan penelitian penulis	10
Tabel 2.9 Tabel kajian penelitian terdahulu	11
Tabel 3.1 Tabel kebutuhan fungsional sistem.....	23
Tabel 3.2 Tabel kebutuhan non-fungsional sistem	24
Tabel 3. 3 Hubungan komponen arsitektur sistem berbasis Android	25
Tabel 3.4 Skenario <i>use case</i> masuk aplikasi versi Android	31
Tabel 3.5 Skenario <i>use case</i> melihat pengumuman	31
Tabel 3.6 Skenario <i>use case</i> mahasiswa mendaftar <i>capstone</i> dan mengakses informasi kelompok.....	32
Tabel 3.7 Skenario <i>use case</i> mahasiswa mengakses dan mengunggah dokumen.....	33
Tabel 3.8 Skenario <i>use case</i> profil mahasiswa.....	34
Tabel 3.9 Skenario <i>use case</i> mahasiswa melihat jadwal sidang proposal.....	35
Tabel 3.10 Skenario <i>use case</i> mahasiswa melihat jadwal expo dan mendaftar expo.	35
Tabel 3.11 Skenario <i>use case</i> mahasiswa yang telah memenuhi syarat sidang Tugas Akhir mendaftar Sidang Tugas Akhir	36
Tabel 3.12 Struktur tabel app_menu	39
Tabel 3.13 Struktur tabel app_role.....	39
Tabel 3.14 Struktur tabel app_user	40
Tabel 3.15 Struktur tabel broadcast	40

Tabel 3.16 Struktur tabel jadwal_expo	41
Tabel 3.17 Struktur tabel jadwal_sidang_proposal	41
Tabel 3.18 Struktur tabel jadwal_sidang_ta.....	42
Tabel 3.19 Struktur tabel kelompok.....	42
Tabel 3.20 Struktur tabel kelompok_mhs	43
Tabel 3.21 Struktur tabel peminatan	44
Tabel 3.22 Struktur tabel pendaftaran_capstone	44
Tabel 3.23 Struktur tabel pendaftaran_expo	45
Tabel 3.24 Struktur tabel jadwal_periode_sidang_ta.....	45
Tabel 3.25 Struktur tabel ruang_sidangs.....	45
Tabel 3.26 Struktur tabel siklus	46
Tabel 3.27 Struktur tabel topik	46
Tabel 3.28 <i>Endpoint</i> REST API.....	47

ABSTRAK

Kata Kunci:

ABSTRACT

Keywords:

BAB I

PENDAHULUAN

1.1 Latar Belakang

Departemen Teknik Komputer merupakan salah satu departemen yang berada di bawah naungan Fakultas Teknik, Universitas Diponegoro. Salah satu persyaratan kelulusan di Departemen Teknik Komputer adalah menyelesaikan *Capstone* Tugas Akhir (*Capstone* TA). *Capstone* TA merupakan model pelaksanaan Tugas Akhir (TA) yang mengharuskan mahasiswa menerapkan pengetahuannya langsung dalam waktu dua semester atau satu tahun. Selama *Capstone* TA, mahasiswa dapat mengaplikasikan ilmu yang telah dipelajari untuk menyelesaikan permasalahan dunia nyata. Menurut ABET (*Accreditation Board for Engineering and Technology*), *Capstone Design* merupakan puncak pengalaman mahasiswa sarjana dalam menciptakan inovasi desain teknik[1]. Sistem *capstone* telah diterapkan oleh Departemen Teknik Komputer Universitas Diponegoro sejak Oktober 2022, dengan pengerjaan berkelompok minimal tiga mahasiswa dan mengikuti linimasa *capstone*.

Saat ini, Departemen Teknik Komputer telah memiliki sistem informasi pengelolaan *Capstone* TA, yaitu pada <https://capstone-ta.ce.undip.ac.id/>. Namun, sistem yang sudah dimiliki Departemen Teknik Komputer masih terbatas pada pengumuman departemen, linimasa *capstone*, dan format dokumen. Sistem informasi tersebut belum mencakup mekanisme pendaftaran, pengelompokan, penetapan dosen pembimbing, dan penjadwalan. Oleh karena itu, mekanisme-mekanisme dan hal terkait *capstone* dilakukan secara manual oleh Tim *Capstone* menggunakan Google Form dan Google Spreadsheet. Pada siklus 1 tahun 2022, terdapat kelompok *capstone* yang mengembangkan sistem informasi pengelolaan *capstone*. Meskipun telah ada pengembangan, sistem ini masih memerlukan penambahan fitur dan perbaikan *bug*.

Saat ini, sistem hanya mencakup mekanisme *capstone* dan belum mencakup mekanisme Tugas Akhir. Aksesnya juga terbatas hanya **melalui web**. Oleh karena itu, diperlukan pengembangan fitur, perbaikan *bug*, dan pengembangan versi Android dari sistem informasi ini. Keberadaan versi Android dapat memperluas jangkauan pengguna, mengingat penggunaan perangkat berbasis Android yang semakin meningkat. Dengan memiliki versi Android, pengguna dapat mengakses sistem informasi ini melalui perangkat mobile mereka di mana saja dan kapan saja.

Dalam penerapannya, sistem informasi versi Android harus terintegrasi dalam satu basis data dengan sistem informasi versi web agar data-data sistem saling terkoneksi. Hal tersebut dapat dipenuhi dengan RESTful API (*Representational State Transfer Application Programming Interface*). Pemrosesan data melalui API kemudian akan digunakan di perangkat Android. Pemrosesan data yang dimaksud mencakup validasi data, pemrosesan logika bisnis, penyimpanan data ke basis data, melakukan proses autentikasi dan otorisasi terhadap permintaan pengguna, dan pemrosesan lainnya yang diperlukan untuk menjalankan fungsi sistem informasi (misalnya seperti perhitungan matematis dan analisis data)[2]. Oleh karena itu, diperlukan pembuatan RESTful API untuk memenuhi kebutuhan pembuatan sistem informasi versi Android. Kemudian, *backend* **Android** sistem informasi versi Android dikembangkan dengan bahasa Kotlin dan menerapkan pola *Model-View-ViewModel* (MVVM). Penerapan pola desain MVVM bertujuan untuk memisahkan logika bisnis dan tampilan dalam kelas yang berbeda[3].

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, rumusan masalah yang akan dibahas pada Tugas Akhir kali ini adalah sebagai berikut:

1. Bagaimana mengintegrasikan sistem versi Android dengan basis data yang digunakan dalam sistem informasi *Capstone* TA?

2. Bagaimana pembuatan RESTful API yang diperlukan untuk komunikasi antara basis data dan sistem versi Android pada sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro?
3. Bagaimana membuat *backend* Android untuk sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian yang dilakukan adalah sebagai berikut:

1. Merancang dan membuat RESTful API sebagai media komunikasi perangkat Android dengan basis data MySQL.
2. Merancang dan membuat *backend* Android untuk sistem informasi pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro.
3. Menerapkan kerangka kerja Android *Native* dengan bahasa Kotlin dalam merancang dan membuat *backend* Android sistem informasi pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro versi Android.

1.4 Batasan Masalah

Dalam penulisan Tugas Akhir ini, pembahasan dibatasi dalam hal berikut, agar pembahasan berfokus pada permasalahan utama dan tidak melebar ke topik-topik lainnya, yaitu sebagai berikut:

1. Tidak membahas secara detail mengenai sistem informasi pengelolaan *Capstone* TA versi web.
2. Pengembangan sistem versi android menggunakan bahasa pemrograman Kotlin dan berkomunikasi dengan basis data MySQL menggunakan RESTful API.
3. Sistem informasi pengelolaan *Capstone* TA versi android hanya memiliki satu *role* yaitu sebagai mahasiswa.

4. Tidak membahas secara detail perbandingan pembuatan *backend* Android menggunakan *pattern* MVVM dan tidak menggunakan *pattern* MVVM.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini diantaranya adalah sebagai berikut.

1. Manfaat untuk Penulis
 - a. Menerapkan ilmu pengetahuan yang sudah didapatkan dari perkuliahan di dalam proses pembuatan sistem.
 - b. Menambah wawasan ilmu pengetahuan yang dimiliki melalui referensi literatur yang digunakan dalam pembuatan sistem.
 - c. Menambah pengalaman dalam merencanakan, membuat, dan menganalisa *backend* sistem versi Android yang dibuat menggunakan bahasa pemrograman Kotlin dan menerapkan pola MVVM.
2. Manfaat untuk Pengguna
 - a. Bagi pengguna mahasiswa Departemen Teknik Komputer Universitas Diponegoro dapat mengakses sistem informasi *Capstone* TA melalui perangkat Android.

1.6 Metodologi Penelitian

Dalam pengerjaan Tugas Akhir ini, penulis melakukan studi pustaka, analisis kebutuhan, perancangan sistem, implementasi, dan pengujian sistem. Berikut penjelasan masing-masing langkah dalam metodologi penelitian yang digunakan dalam pengerjaan Tugas Akhir:

1. Studi Pustaka

Pada studi pustaka akan dipelajari tentang dasar teori yang digunakan untuk membuat sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro versi Android.

2. Analisis Kebutuhan

Analisis kebutuhan merupakan proses yang digunakan untuk merancang sistem secara lengkap. Identifikasi kebutuhan sistem dilakukan dengan diskusi bersama *stakeholder* yaitu tim *capstone* Departemen Teknik Komputer Universitas Diponegoro. Setelah analisis kebutuhan selesai, maka dapat dihasilkan kebutuhan pengguna, deskripsi sistem, tujuan sistem, fungsionalitas sistem, karakteristik pengguna, serta batasan sistem.

3. Perancangan Sistem

Perancangan sistem ini berisi tentang gambaran perancangan sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro versi Android. Dalam perancangan ini penulis membahas tentang gambaran umum sistem, kebutuhan alat untuk pengembangan, dan alur kerja sistem. Perancangan sistem dibutuhkan untuk menjadi acuan dalam tahap selanjutnya.

4. Implementasi

Implementasi sistem merupakan hasil dari perancangan sistem. Implementasi ini menghasilkan produk berupa sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro versi Android. Proses implementasi dilakukan dengan pembuatan RESTful API, kemudian implementasi *backend* Android. Penulis menggunakan bahasa pemrograman PHP dengan kerangka kerja Laravel untuk pembuatan RESTful API dan bahasa Kotlin sebagai bahasa pemrograman untuk mengembangkan *backend* Android dengan pola MVVM.

5. Pengujian Sistem

Tahap akhir dari pengembangan sistem informasi *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro versi Android adalah pengujian sistem. Penulis menggunakan pengujian *unit testing* yaitu PHPUnit untuk menguji RESTful API dan *black box testing* untuk menguji *backend* Android. Selain itu, penulis juga melakukan pengujian *load testing* dan *energy usage testing* untuk menguji performa aplikasi.

1.7 Sistematika Penulisan

Untuk menggambarkan mengenai isi dari laporan Tugas Akhir ini, berikut diuraikan secara singkat dalam sistematika penulisan. Sistematika penulisan laporan Tugas Akhir ini terdiri dari lima bab dengan susunan sebagai berikut:

BAB I PENDAHULUAN

Bab pertama berisi tentang penjelasan mengenai latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab kedua menjelaskan mengenai konsep dasar dalam penelitian dan pengembangan sistem. Tinjauan Pustaka menguraikan kajian hasil penelitian terdahulu serta landasan teori yang digunakan.

BAB III PERANCANGAN SISTEM

Bab ketiga ini menjelaskan tentang rencana perancangan dan pengembangan sistem, diantaranya dijabarkan mengenai pembuatan RESTful API dan pembuatan *backend* Android sistem informasi pengelolaan *Capstone* TA.

BAB IV HASIL DAN PEMBAHASAN

Berisi tulisan dan gambar yang menjelaskan hasil dari tahap implementasi dan pengujian sistem informasi pengelolaan *Capstone* TA.

BAB V PENUTUP

Di bab terakhir ini berisi tentang kesimpulan yang didapatkan dari hasil pengembangan sistem yang telah dibuat serta apakah sistem tersebut telah sesuai dengan rancangan awal yang telah dibuat. Selain itu, terdapat pula saran yang digunakan untuk perbaikan dan pengembangan yang dapat dilakukan selanjutnya.

BAB II

KAJIAN PUSTAKA

2.1 Kajian Penelitian Terdahulu

Dalam pelaksanaan penelitian ini diperlukan beberapa penelitian yang telah dilaksanakan sebelumnya sebagai acuan dan pembanding. Penelitian-penelitian sebelumnya yang digunakan adalah penelitian dengan tema, metode, atau kerangka perancangan yang mirip dengan penelitian yang dilakukan oleh penulis.

Pada penelitian yang dilakukan oleh Dharmawan dkk. pada tahun 2019, dengan judul penelitian “Implementasi Sistem Informasi Tugas Akhir Menggunakan Metode *Classic Life Cycle*”[4] menjelaskan bahwa Tugas Akhir bertujuan agar mahasiswa mampu menyusun karya ilmiah yang sesuai dengan bidang ilmunya. Selain itu, penerapan teknologi pada proses bisnis sistem informasi Tugas Akhir dijelaskan dapat mempercepat proses pengolahan data dan mempercepat proses validasi. Terdapat beberapa perbedaan antara penelitian Dharmawan dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.1.

Tabel 2.1 Perbandingan penelitian Dharmawan dkk. dengan penelitian penulis

Penelitian Dharmawan dkk	Penelitian Penulis
Sistem yang dikembangkan adalah sistem berbasis web	Sistem yang dikembangkan adalah sistem berbasis Android
Menggunakan metode pengembangan <i>Waterfall</i>	Menggunakan metode pengembangan <i>Rapid Application Development</i>
Metode pengujian menggunakan <i>blackbox testing</i>	Metode pengujian menggunakan <i>blackbox testing</i> dan <i>usability testing</i>

Pada penelitian yang dilakukan oleh Sari dkk. pada tahun 2019 dengan judul penelitian, “Sistem Informasi Monitoring Pembimbingan Skripsi/Tugas Akhir (SIMP-S/TA) Berbasis Android”[5], dijelaskan bahwa dalam proses pengerjaan Tugas Akhir, pertemuan tatap muka secara langsung seringkali tidak dapat terlaksana. Hal ini disebabkan karena perbedaan jadwal dan kesibukkan masing-masing individu yaitu mahasiswa dan dosen pembimbing untuk mengadakan pertemuan dalam rangka bimbingan. Terdapat beberapa perbedaan antara penelitian Sari dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.2

Tabel 2.2 Perbandingan penelitian Sari dkk. dengan penelitian penulis

Penelitian Sari dkk	Penelitian Penulis
Aplikasi dikembangkan menggunakan bahasa pemrograman Java	Aplikasi dikembangkan menggunakan bahasa pemrograman Kotlin
Menggunakan metode pengembangan <i>Waterfall</i>	Menggunakan metode pengembangan <i>Rapid Application Development</i>

Pada penelitian yang dilakukan oleh Lander dkk. pada tahun 2021 dengan judul penelitian “Desain dan Implementasi REST-API Pada Website *Marketplace* Katering “Ketringan” Menggunakan *Framework* Laravel”[6], menjelaskan bahwa REST-API dibangun untuk memenuhi kebutuhan sistem yang dikembangkan, yaitu agar API bisa diakses dan digunakan oleh perangkat *mobile*. Terdapat beberapa perbedaan antara penelitian Alvin Lander dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.3.

Tabel 2.3 Perbandingan penelitian Lander dkk. dengan penelitian penulis

Penelitian Lander dkk	Penelitian Penulis
API digunakan untuk memenuhi kebutuhan pada sistem pemesanan katering	API digunakan untuk memenuhi kebutuhan pada sistem informasi pengelolaan <i>Capstone</i> TA
Menggunakan metode pengembangan SCRUM	Menggunakan metode pengembangan <i>Rapid Application Development</i>
Menggunakan Nginx sebagai perangkat <i>web server</i>	Menggunakan XAMPP sebagai perangkat <i>web server</i>

Pada penelitian yang dilakukan Nurwicaksono dkk pada tahun 2024 dengan penelitian yang berjudul, “Perancangan Sistem Manajemen Kasir Berbasis Android pada Blessing of Shoes” [7], dijelaskan bahwa penerapan sistem berbasis Android akan dapat meminimalisir kesalahan pencatatan keuangan. Selain itu, dijelaskan juga metode *Rapid Application Development* menggunakan pendekatan iteratif (berulang) dalam pengembangan sistem, di mana sebuah model kerja sistem dibangun pada tahap awal pengembangan dengan memperhatikan kebutuhan pengguna. Dijelaskan juga bahwa metode *Rapid Application Development* mampu menangani sistem dengan rentang waktu yang relatif singkat. Terdapat beberapa perbedaan antara penelitian Nurwicaksono dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.4.

Tabel 2.4 Perbandingan penelitian Nurwicaksono dkk. dengan penelitian penulis

Penelitian Nurwicaksono dkk	Penelitian Penulis
Pengembangan aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem manajemen kasir.	Pengembangan aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem informasi pengelolaan <i>Capstone</i> TA
Metode pengujian menggunakan <i>blackbox testing</i>	Metode pengujian menggunakan <i>blackbox testing</i> dan <i>usability testing</i>
Basis data yang digunakan adalah basis data lokal yaitu Room SQLite <i>database</i>	Basis data yang digunakan adalah basis data <i>remote web server</i> yaitu MySQL

Pada penelitian yang dilakukan Murdiani dkk pada tahun 2022 dengan penelitian yang berjudul, “Perbandingan Metode *Waterfall* dan RAD (*Rapid Application Development*) pada Pengembangan Sistem Informasi”[8], dijelaskan bahwa Selain waterfall ada juga model RAD yang merupakan pengembangan lebih lanjut dari waterfall. pengembangan model RAD merupakan model inkremental yang menekankan pada sedikitnya siklus pengembangan. Terdapat beberapa perbedaan antara penelitian Mardiani dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.5.

Tabel 2.5 Perbandingan penelitian Mardiani dkk dengan penelitian penulis

Penelitian Mardiani dkk	Penelitian Penulis
Hanya membandingkan metode pengembangan dalam <i>Software Development Life Cycle</i> .	Membuat aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem informasi pengelolaan <i>Capstone</i> TA

Pada penelitian yang dilakukan Ridhoi dkk pada tahun 2022 dengan penelitian yang berjudul, “Rancang Bangun Mobile Learning Berbasis Android di SMKN Poncol”[9], dijelaskan bahwa penelitian ini menargetkan smartphone Android untuk dapat mengoperasikan mobile learning yang akan dibuat. Mobile learning ini dibuat dengan menggunakan bahasa pemrograman Kotlin dan menerapkan metode RAD (*Rapid Application Development*). Terdapat beberapa perbedaan antara penelitian Ridhoi dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.6.

Tabel 2.6 Perbandingan penelitian Ridhoi dkk dengan penelitian penulis

Penelitian Ridhoi dkk	Penelitian Penulis
Pengembangan aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem <i>mobile learning</i>	Pengembangan aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem informasi pengelolaan <i>Capstone</i> TA

Metode pengujian menggunakan <i>blackbox testing</i>	Metode pengujian menggunakan <i>unit testing</i> dan <i>performance testing</i> .
--	---

Pada penelitian yang dilakukan Riyadhhi dkk pada tahun 2022 dengan penelitian yang berjudul, “Penerapan Pola Arsitektur MVVM Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android”[10], dijelaskan bahwa penerapan pola arsitektur MVVM ini memungkinkan aplikasi untuk mempertahankan keadaan dari perubahan konfigurasi yang terjadi pada *smartphone* pengguna, serta mempermudah akses data. Pola arsitektur MVVM juga memecah kode menjadi beberapa komponen, yaitu *Model*, *View*, dan *ViewModel*. Terdapat beberapa perbedaan antara penelitian Riyadhhi dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.7.

Tabel 2.7 Perbandingan penelitian Riyadhhi dkk dengan penelitian penulis

Penelitian Riyadhhi dkk	Penelitian Penulis
Pengembangan aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem pengaduan masyarakat	Pengembangan aplikasi berbasis Android digunakan untuk memenuhi kebutuhan sistem informasi pengelolaan <i>Capstone TA</i>
Metode pengembangan yang digunakan adalah metode SCRUM	Metode pengembangan yang digunakan adalah metode <i>Rapid Application Development</i>

Pada penelitian yang dilakukan Thohari dkk pada tahun 2018 dengan penelitian yang berjudul, “Implementasi Test Driven Development dalam Pengembangan Aplikasi Berbasis Web”[11], menjelaskan tentang pengujian dalam dunia pemrograman. Dalam dunia pemrograman, pengujian adalah metode verifikasi perangkat lunak di mana seorang programmer menguji bahwa suatu unit atau bagian terkecil atau fitur dari program layak untuk dipakai. Terdapat beberapa perbedaan antara penelitian Padang dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.8.

Tabel 2. 8 Perbandingan penelitian Padang dkk dengan penelitian penulis

Penelitian Afandi dkk	Penelitian Penulis
<i>Unit testing</i> digunakan untuk menguji website IMDB.	<i>Unit testing</i> digunakan untuk menguji RESTful API yang akan digunakan oleh sistem berbasis Android.

Penelitian terdahulu memiliki peran penting sebagai sumber referensi bagi penulis dalam menyelesaikan penelitian penulis. Melalui penelitian-penelitian sebelumnya, penulis dapat memperoleh pemahaman mendalam tentang topik yang

diteliti, mengidentifikasi kesenjangan pengetahuan, dan merumuskan pertanyaan penelitian yang relevan. Selain itu, referensi dari penelitian terdahulu membantu penulis dalam merancang metodologi penelitian yang tepat dan memilih pendekatan yang sesuai. Berikut ringkasan yang berisi judul, tujuan, metode penelitian dan hasil dari penelitian-penelitian terdahulu seperti yang dijelaskan pada Tabel 2.9.

Tabel 2.9 Tabel kajian penelitian terdahulu

No	Judul dan Peneliti	Tujuan	Metode Penelitian	Hasil
1	Implementasi Sistem Informasi Tugas Akhir Menggunakan Metode <i>Classic Life Cycle</i> Oleh: Dharmawan dkk. (2019)	Penelitian ini bertujuan untuk membangun sistem informasi pengajuan judul skripsi, guna membantu mahasiswa Universitas Nusantara PGRI Kediri pada Progam Studi Sistem Informasi yang telah memenuhi syarat akademik.	Penelitian menggunakan metode <i>Waterfall</i> . Tahap pengujian menggunakan pengujian <i>blackbox</i> .	Menghasilkan sistem informasi Tugas Akhir untuk proses pengelolaan pengajuan dan validasi judul Tugas Akhir mahasiswa.
2	Sistem Informasi Monitoring Pembimbingan Skripsi/Tugas Akhir (SIMP-S/TA) Berbasis Android Oleh: Sari dkk. (2019)	Membuat sistem informasi monitoring tugas akhir berbasis android.	Penelitian menggunakan metode <i>Waterfall</i> . Bahasa pemrograman yang digunakan dalam mengembangkan aplikasi adalah bahasa Java.	Aplikasi android dapat memonitoring capaian hasil pembimbingan yang telah dilakukan oleh mahasiswa sehingga dosen pembimbing dapat memantau capaian pekerjaan mahasiswa berdasarkan laporan per bab
3	Desain dan Implementasi REST-API Pada Website <i>Marketplace</i> Katering “Ketringan” Menggunakan <i>Framework</i> Laravel Oleh: Alvin Lander dkk. (2021)	Bertujuan untuk membuat perangkat lunak yang memiliki protokol transfer HTTP yaitu REST API dengan tujuan agar dapat digunakan untuk berkomunikasi antar aplikasi.	Penelitian menggunakan metode SCRUM, dengan menerapkan Nginx sebagai perangkat <i>web server</i> .	Laravel <i>framework</i> dapat menjadi solusi dalam pengembangan REST API. Implementasi REST-API ini berjalan dengan baik dan tidak memiliki kendala ketika client melakukan <i>request</i> . <i>Response</i> yang diberikan juga sudah cukup baik berdasarkan beberapa <i>user</i> yang telah mengujinya.

Tabel 2.9 Tabel kajian penelitian terdahulu (lanjutan)

No	Judul dan Peneliti	Tujuan	Metode Penelitian	Hasil
4	Perancangan Sistem Manajemen Kasir berbasis Android pada Blessing of Shoes Oleh: Nurwicaksono dkk (2024)	Mewujudkan sistem kasir pada Toko Blessing Of Shoes untuk mempermudah karyawan dalam melakukan transaksi dengan kostumer	Penelitian menggunakan metode <i>Rapid Application Development</i> , dengan basis data lokal Room SQLite dan pengujian yang digunakan adalah <i>blackbox testing</i> .	Aplikasi Sistem Manajemen Kasir berbasis Android yang dikembangkan dengan bahasa pemrograman Kotlin.
5	Perbandingan Metode <i>Waterfall</i> dan RAD (<i>Rapid Application Development</i>) pada Pengembangan Sistem Informasi Oleh: Mardiani dkk (2022)	Membandingkan 2 metode pengembangan sistem metode mana yang paling tepat digunakan dalam mengembangkan sebuah sistem perangkat lunak.	Membandingkan metode pengembangan <i>Rapid Application Development</i> dan <i>Waterfall</i> .	RAD lebih efektif dibandingkan dengan metode <i>waterfall</i> . hal ini disebabkan karena pada RAD menggunakan konsep <i>iterative</i> (berulang) sedangkan pada metode <i>waterfall</i> tidak dapat berulang dan waktu pengerjaan dilakukan secara bertahap.
6	Rancang Bangun Mobile Learning Berbasis Android di SMKN Poncol Oleh: Rhidoi dkk (2022)	Membangun aplikasi <i>mobile learning</i> untuk SMKN Poncol dapat menciptakan proses pembelajaran di berbagai konteks melalui interaksi sosial dan konten menggunakan perangkat elektronik pribadi dengan bahasa pemrograman Kotlin.	Metode yang digunakan adalah metode <i>Rapid Application Development</i> , dengan bahasa yang digunakan untuk mengembangkan aplikasi Android adalah bahasa Kotlin.	<i>Mobile learning</i> ini mampu membuat kegiatan belajar mengajar di SMKN Poncol menjadi lebih menarik dan terdigitalisasi.
7	Penerapan Pola Arsitektur MVVM Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android	Membangun aplikasi pengaduan masyarakat berbasis Android dengan menerapkan arsitektur MVVM.	Metode yang digunakan adalah <i>Rapid Application Development</i> dengan bahasa yang digunakan untuk mengembangkan aplikasi Android adalah bahasa Kotlin dengan pola MVVM.	Penelitian ini menghasilkan aplikasi pengaduan masyarakat berbasis Android dengan menggunakan pola arsitektur MVVM. Dengan menerapkan pola arsitektur ini dapat mempermudah untuk melakukan

	Oleh: Riyadhhi dkk (2022)			pemeliharaan kode yang telah ada
--	---------------------------	--	--	----------------------------------

Tabel 2.9 Tabel kajian penelitian terdahulu (lanjutan)

No	Judul dan Peneliti	Tujuan	Metode Penelitian	Hasil
8	Implementasi Test Driven Development dalam Pengembangan Aplikasi Berbasis Web Oleh: Thohari dkk (2018)	Memastikan perangkat sudah siap digunakan dengan melakukan pengujian.	<i>Test driven development</i> (TDD) yaitu pengembangan perangkat lunak dilakukan dengan membuat test case terlebih dahulu baru kemudian melakukan <i>producing code</i> .	Penerapan metode TDD membantu pengembang dalam menerapkan proses bisnis dan kebutuhan dari pengguna aplikasi IMDB.

Berdasarkan penelitian-penelitian sebelumnya, terbukti bahwa penggunaan bahasa pemrograman Kotlin dan pola desain MVVM dapat membuat struktur kode dan proyek lebih mudah dirawat dan dikembangkan oleh pengembang[10]. Pola desain MVVM juga direkomendasikan oleh Google dalam proses pengembangan aplikasi Android native. Oleh karena itu, dapat disimpulkan bahwa Kotlin dan MVVM memiliki manfaat untuk membuat kode lebih terstruktur, mudah dirawat, dan mudah dikembangkan. Selain itu, penerapan unit testing dengan PHPUnit untuk menguji API dan *black box* testing untuk menguji backend Android, dapat digunakan untuk memastikan aplikasi Android berjalan sebagaimana yang diharapkan[11].

2.2 Landasan Teori

2.2.1 Capstone TA

Capstone Tugas Akhir, atau yang biasa disebut *Capstone* TA, merupakan suatu model pelaksanaan Tugas Akhir yang menuntut mahasiswa mengaplikasikan kemampuannya secara langsung dalam jangka waktu pelaksanaan dua semester atau satu tahun[1]. Selama melaksanakan *Capstone* TA, mahasiswa dapat menggunakan seluruh ilmu yang telah mereka pelajari untuk menyelesaikan suatu permasalahan riil di dunia nyata. Sesuai dengan definisi berdasarkan ABET (*Accreditation Board for Engineering and Technology*), *Capstone Design* merupakan puncak dari pengalaman

mahasiswa sarjana dalam menciptakan cetak biru untuk inovasi dalam desain teknik. Topik *Capstone* TA, adalah sebagai berikut.

1. *Early Warning System*
2. *Building/area monitoring or controlling system*
3. *Smart business/organization platform/support system*
4. *Smart city and transportation*

Judul-judul yang diajukan di dalam *Capstone* TA harus berada di bawah topik-topik di atas sebagai bagian dari Rencana Induk Penelitian Program Studi S-1 Teknik Komputer. Luaran dapat berbentuk purwarupa maupun produk yang merupakan implementasi dari perancangan dalam proses problem solving. *Output* Laporan *Capstone* TA:

1. Dokumen Proposal (C100): dokumen berisi latar belakang dan rumusan masalah yang akan diselesaikan.
2. Dokumen Spesifikasi Produk (C200): dokumen berisi spesifikasi produk sebagai solusi terhadap permasalahan dalam proposal.
3. Dokumen Perancangan (C300): dokumen berisi penjabaran desain untuk merealisasikan produk (baik dari sisi *hardware*, *software*, simulasi, maupun kombinasi ketiganya).
4. Dokumen Implementasi (C400): dokumen berisi hasil desain, seperti prototipe produk (2D/3D), *source code*, layout hasil produk, tabel/grafik hasil simulasi, dan lain-lain.
5. Dokumen Pengujian (C500): dokumen berisi hasil pengujian dari produk yang dibuat
6. Presentasi, Makalah, Artikel Publikasi, Poster, dan Banner: pemaparan hasil dari *capstone project*, baik secara lisan (presentasi langsung saat ujian Proposal TA dan *Expo* TA), tertulis (melalui makalah dan artikel publikasi), maupun visual (melalui media poster dan *banner*).

2.2.2 Sistem Informasi

Sistem informasi adalah sebuah sistem formal, sosioteknikal, dan organisasional yang dirancang untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi. Dalam dunia teknologi internet sangatlah diperlukan dalam menyampaikan informasi. Internet sudah banyak diaplikasikan dalam berbagai bidang, salah satunya dalam bidang pendidikan seperti Universitas. Dukungan sistem informasi dalam peningkatan mutu pelayanan akademik yang terdiri dari banyaknya proses diantaranya perencanaan perkuliahan hingga proses akhir seperti halnya bimbingan skripsi[4].

2.2.3 RESTful API

Representatif State Transition Application Programming Interface (RESTful API) adalah standar arsitektur komunikasi yang diterapkan dalam pengembangan web dan layanan berbasis aplikasi, sedangkan API atau *Application Programming Interface* merupakan antarmuka yang dibangun oleh pengembang sistem berupa tautan yang dapat mengakses beberapa fungsi atau semua fungsi secara terprogram sehingga aplikasi dapat berinteraksi dan berbagi data[12].

Selain itu, RESTful API merupakan standar arsitektur komunikasi yang banyak digunakan dalam pengembangan web dan layanan berbasis aplikasi. Melalui RESTful API, pengguna dapat mengakses dan memanipulasi data dari sistem secara terprogram melalui antarmuka yang telah ditentukan. Dengan membangun REST API yang baik dan representatif, sistem dapat memfasilitasi interaksi antara aplikasi dengan sistem secara efisien dan aman, memungkinkan integrasi yang mulus dengan berbagai platform dan layanan eksternal.

2.2.4 MySQL

MySQL merupakan *software* basis data *open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa SQL[13]. MySQL merupakan

tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan. Keuntungan menyimpan data di basis data adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk tabel. MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan para pemrogram aplikasi web. Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di *bundling* dengan *web server* sehingga proses instalasinya jadi lebih mudah.

2.2.5 PHP

PHP singkatan dari *Hypertext Preprocessor* adalah sebuah bahasa pemrograman yang berbentuk *scripting*, yang digunakan untuk membuat halaman web yang dinamis walaupun tidak tertutup kemungkinan untuk digunakan pemakai lain. *Web base programming*, pada umumnya menggunakan bahasa pemrograman PHP, dengan pasangan aplikasi pengelola basis data MySQL[14].

Dengan kemampuannya untuk disematkan langsung ke dalam kode HTML, PHP dieksekusi di sisi *server*, menghasilkan *output* yang dikirim ke peramban web pengguna. Beberapa karakteristik utama PHP meliputi kemampuan *server-side scripting*, dukungan terhadap berbagai jenis basis data seperti MySQL dan PostgreSQL, serta integrasi yang mudah dengan HTML, memungkinkan pengembang mencampur logika pemrograman dengan konten web untuk pengembangan aplikasi dinamis.

2.2.6 Laravel

Laravel adalah sebuah kerangka kerja web berbasis PHP yang *open-source* dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC[15]. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di laravel terdapat routing

yang menjembatani antara *request* dari pengguna dan *controller*. Jadi *controller* tidak langsung menerima *request* tersebut. Salah satu fitur kunci Laravel adalah kemampuannya dalam menyederhanakan tugas-tugas umum pengembangan web melalui konsep-konsep seperti *routing* yang mudah, kontroler yang terorganisir, dan sistem *templating* menggunakan blade.

2.2.7 Postman

Postman adalah sebuah aplikasi yang berfungsi sebagai REST *Client*, yang digunakan untuk melakukan uji coba RESTful API[16]. Dengan antarmuka grafis yang intuitif, pengguna dapat membuat, mengorganisir, dan menguji permintaan API, serta menyusunnya dalam koleksi dan lingkungan yang dapat diatur. Fitur-fitur seperti variabel, skrip uji, dan kemampuan dokumentasi API membuat Postman menjadi alat yang sangat berguna dalam pengembangan aplikasi yang berinteraksi dengan API. Selain itu, kemampuan untuk berkolaborasi, memonitor, dan melacak permintaan API memberikan nilai tambah dalam siklus pengembangan perangkat lunak. Postman telah menjadi pilihan populer di kalangan pengembang untuk mempermudah pengujian dan pengelolaan API.

2.2.8 Android

Android merupakan sistem operasi dengan kode sumber terbuka, dan Google merilis kode tersebut di bawah Lisensi Apache[7]. Dengan sifatnya yang terbuka untuk umum (sumber terbuka) dan menggunakan lisensi Apache, Android memungkinkan pengguna untuk secara bebas memodifikasi perangkat lunak dan mendistribusikannya oleh pembuat perangkat, operator nirkabel, maupun pengembang aplikasi. Selain itu, Android juga memiliki komunitas pengembang aplikasi yang sangat besar, yang menciptakan aplikasi-aplikasi kustom untuk memperluas fungsionalitas perangkat. Android dirancang khusus untuk perangkat bergerak layar sentuh seperti telepon pintar dan tablet. Pada awalnya, Android dikembangkan oleh perusahaan bernama Android

dengan dukungan finansial dari Google, namun kemudian Google membeli Android pada tahun 2005.

2.2.9 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform* android[17]. Android studio ini berbasis pada IntelliJ IDEA. Untuk membuat tampilan, digunakan bahasa XML. Android studio juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk *deploy* ke perangkat android. Android Studio juga merupakan pengembangan dari eclipse, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya Android Studio IDE, Android SDK *tools*.

2.2.10 Kotlin

Kotlin adalah bahasa pemrograman yang diakui oleh Google sebagai bahasa resmi pemrograman Android yang dapat bekerja sama dengan pustaka yang dibuat dari bahasa Java[9]. Kotlin memiliki beberapa keunggulan dibandingkan dengan Java. Pertama, Kotlin dirancang untuk bekerja secara interoperabilitas dengan Java, sehingga memudahkan pengembang yang sudah familiar dengan Java untuk beralih ke Kotlin tanpa kesulitan. Selain itu, Kotlin menawarkan sintaksis yang lebih ringkas dan ekspresif, mengurangi *boilerplate code* yang sering ditemui dalam Java. Kotlin juga mendukung fitur-fitur modern seperti *null safety*, yang membantu mengurangi kesalahan terkait *null pointer*, dan ekstensi fungsi yang memungkinkan penambahan fungsionalitas pada kelas tanpa perlu mengubah kode sumber aslinya. Keseluruhan, Kotlin memberikan pengalaman pengembangan yang lebih nyaman, efisien, dan aman dibandingkan dengan Java.

2.2.11 Model View ViewModel

Model-View-ViewModel (MVVM) merupakan pola desain atau paradigma yang baru untuk memisahkan antarmuka pengguna dari logika bisnis. Google selalu merekomendasikan hal ini kepada pengembang dalam setiap proyek pengembangan aplikasi android sejak MVVM diperkenalkan Google pada tahun 2018. MVVM terbagi atas beberapa *layer*, yaitu *Model*, *View*, dan *ViewModel*[18].

1. *Model*

Model atau entitas merupakan *layer* yang merepresentasikan data yang akan digunakan pada logika bisnis. Kelas-kelas yang ada pada model umumnya merupakan kelas *Plain Old Java Object* (POJO) pada bahasa pemrograman Java dan *Data Classes* pada bahasa pemrograman Kotlin[18].

2. *View*

View merupakan *layer* yang berisi *User Interface* (UI) aplikasi untuk mengatur bagaimana data atau informasi akan ditampilkan kepada pengguna. *Layer* ini berisi kelas *Activity* atau *Fragment*.

3. *ViewModel*

ViewModel merupakan *layer* yang digunakan untuk berinteraksi kepada model untuk meneruskan data pada *View*.

2.2.12 Metode RAD

Rapid Application Development (RAD) adalah proses model perangkat lunak inkremental yang menekankan siklus pengembangan yang singkat. Model RAD adalah sebuah adaptasi “kecepatan tinggi” dari model *waterfall*, di mana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika tiap-tiap kebutuhan dan batasan ruang lingkup proyek telah diketahui dengan baik, proses RAD memungkinkan tim pengembang untuk menciptakan sebuah sistem yang berfungsi penuh dalam jangka waktu yang sangat singkat[14].

2.2.13 Unit Testing

Pengujian dapat menjadi alat ukur kualitas perangkat lunak yang sangat diperlukan sebelum sebuah perangkat lunak tersebut dirilis atau diluncurkan. Hal tersebut perlu dilakukan untuk meyakinkan pelanggan atau pengguna bahwa perangkat lunak tersebut sudah layak untuk digunakan dan memenuhi kinerja sesuai kebutuhan dari pelanggan atau pengguna.

Unit testing adalah teknik untuk menguji apakah kode program perangkat lunak sudah efektif dan bebas dari kesalahan[19]. Pada beberapa kasus pengembangan perangkat lunak, sering terjadi bahwa kode program yang dikembangkan ternyata kurang efektif atau bahkan tidak pernah digunakan. Pengujian jenis ini dapat membantu dalam menemukan kode program yang kurang efektif tersebut serta dapat mengukur seberapa efektif kode program pada pengembangan perangkat lunak tersebut. Selain itu, pengujian ini sering digunakan oleh para pengembang untuk menemukan kesalahan dan cacat yang tidak terlihat saat program dieksekusi.

2.2.14 Black Box Testing

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian *black box* bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi[20]. Pengujian *black box* bertumpu pada memastikan tiap proses sudah berfungsi sesuai dengan kebutuhan yang diharapkan. Penguji dapat menartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari sistem. Sehingga pengujian merupakan suatu cara pelaksanaan program yang bertujuan menemukan kesalahan kemudian memperbaikinya sehingga sistem dapat dikatakan layak untuk digunakan.

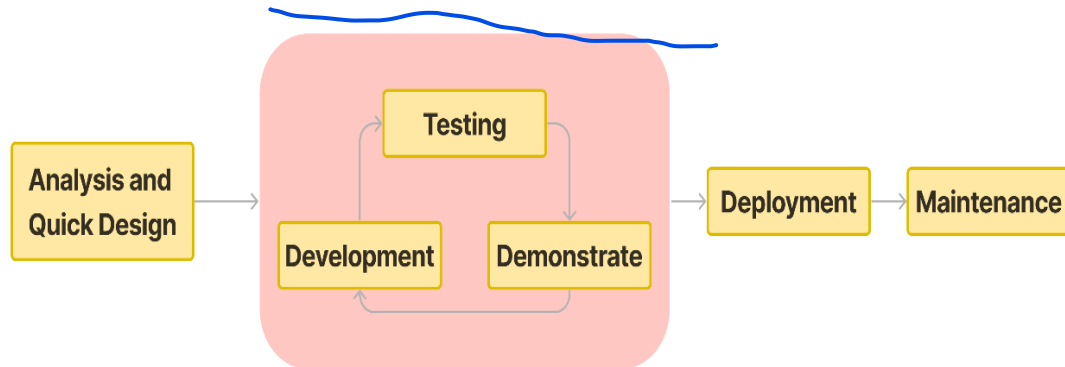
2.2.15 Load Testing

Load Testing merupakan teknik pengujian performa dimana respon sistem diukur dalam berbagai kondisi dan beban. Pengujian ini membantu menentukan bagaimana *software* berperilaku ketika pengguna mengakses *software*. *Load testing* digunakan untuk memeriksa bagaimana sebuah sistem yang sedang dikembangkan bisa menangani masalah atau beban yang diujikan yang disesuaikan dengan keadaan sebenarnya[21].

Load testing berfokus pada tiga indikator yaitu, *response time* merupakan ukuran waktu tunggu saat pengguna melakukan *request* ke *server* hingga *server* memberikan *response* kembali. *Throughput* merupakan jumlah *request* pengguna yang diproses *server* dalam satuan detik / *request per second* (rps). *Error rate* merupakan perhitungan persentase proses yang gagal saat proses request dan response berlangsung

BAB III

PERANCANGAN SISTEM



Gambar 3.1 Metode *Rapid Application Development* dalam pengembangan API dan *backend* Android

Pengembangan sistem ini menggunakan metode *Rapid Application Development*. Proses di mulai dengan *analysis and quick design*, yaitu pada proses ini Penulis melakukan wawancara dengan *stakeholder*. Wawancara dilakukan untuk menemukan permasalahan yang dialami oleh *stakeholder* dan berdiskusi tentang solusi yang bisa dilakukan untuk menyelesaikan permasalahan tersebut. Kemudian, proses berlanjut pada tahap iterasi yaitu *development*, *testing*, dan *demonstrate*. Tahap *development* merupakan tahap implementasi dari rancangan yang telah dibuat. Tahap *development* menghasilkan API dan hasil integrasi API dengan antarmuka pengguna melalui *backend* Android yaitu berupa aplikasi Android. API dan aplikasi android kemudian diuji untuk memastikan aplikasi sudah layak didemonstrasikan kepada *stakeholder*. Dari demonstrasi, *stakeholder* memberikan masukan dan permintaan untuk menyesuaikan kebutuhan, lalu proses berulang sampai aplikasi dapat diterima oleh *stakeholder*. Selanjutnya, masuk ke tahapan *deployment* untuk rilis aplikasi agar dapat digunakan oleh mahasiswa Departemen Teknik Komputer, bersamaan dengan *maintenance* atau pemeliharaan sistem.

3.1 *Analysis and Quick Design*

Tahap *analysis and quick design* dilakukan dengan wawancara dengan *stakeholder* yaitu Tim Capstone Departemen Teknik Komputer. Dari wawancara tersebut, didapatkan kebutuhan fungsional dan kebutuhan non fungsional sistem. Lalu, pada tahap *analysis and quick design*, Penulis juga menentukan arsitektur sistem berdasarkan kebutuhan sistem.

3.1.1 Kebutuhan Fungsional Sistem

Sistem yang Penulis kembangkan memiliki fungsi utama agar mahasiswa Departemen Teknik Komputer dapat mengakses dan mengelola informasi mengenai *Capstone TA* pada. Sistem ini akan menjadi aplikasi mahasiswa dalam mengakses informasi termasuk pendaftaran, jadwal, dan unggah dokumen. Kebutuhan fungsional merupakan kebutuhan akan fasilitas yang dibutuhkan serta aktivitas apa saja yang dilakukan oleh sistem secara umum[22]. Kebutuhan fungsional sistem dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel kebutuhan fungsional sistem

No	Kategori Pengguna	Deskripsi Kebutuhan	Prioritas
1	Mahasiswa	Masuk ke dalam aplikasi.	Tinggi
2	Mahasiswa	Mengakses pengumuman terkait <i>capstone TA</i> .	Tinggi
3	Mahasiswa	Melakukan pendaftaran <i>capstone</i> , mengakses informasi, dan mengelola data kelompok <i>capstone</i> .	Tinggi
4	Mahasiswa	Mengakses dan mengunggah dokumen <i>capstone TA</i> .	Tinggi
5	Mahasiswa	Melihat informasi pembagian kelompok dan dosen pembimbing.	Tinggi
6	Mahasiswa	Melihat informasi jadwal sidang proposal dan dosen penguji sidang proposal.	Tinggi
7	Mahasiswa	Mengakses data <i>expo</i> tersedia, melakukan pendaftaran <i>expo</i> dan mengakses jadwal <i>expo</i> .	Tinggi
8	Mahasiswa	Mengakses periode sidang TA, mendaftar sidang TA, dan mengakses informasi penguji sidang TA.	Tinggi
9	Mahasiswa	Mengakses dan mengubah profil mahasiswa.	Tinggi
10	Mahasiswa	Keluar dari akun mahasiswa.	Tinggi

3.1.2 Kebutuhan Non Fungsional Sistem

Selain kebutuhan fungsional, sistem juga memiliki kebutuhan non-fungsional sistem yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel kebutuhan non-fungsional sistem

Parameter	Requirement
Portability	Sistem dapat dioperasikan pada perangkat Android dengan bermacam versi OS.
Response time	Memberikan waktu respon maksimal 10 detik.

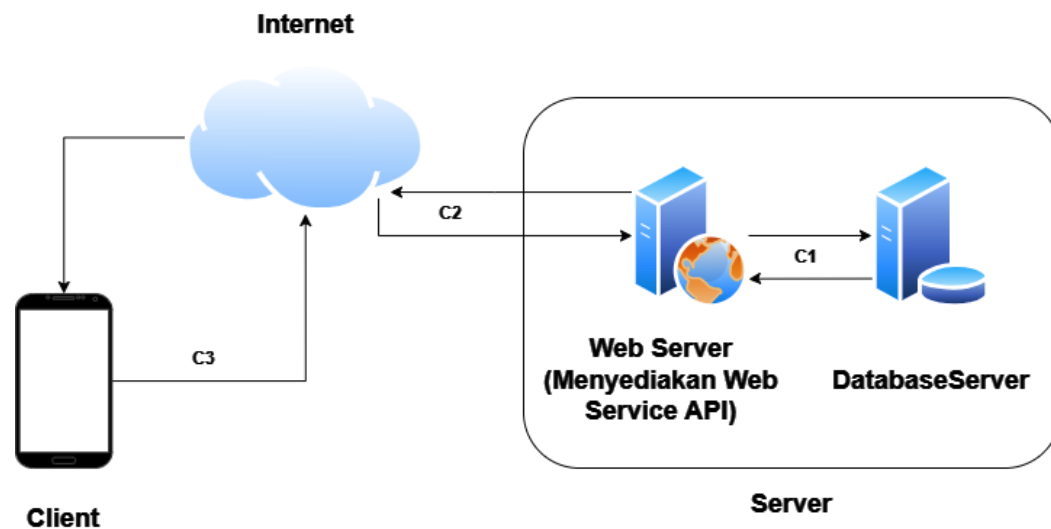
3.1.3 Karakteristik Pengguna

Sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro memiliki karakteristik pengguna yang dilihat dari aspek pekerjaan dan hak akses pengguna. Pengguna sistem versi Android adalah mahasiswa Departemen Teknik Komputer. Maka dari itu, hak akses yang dimiliki mahasiswa yaitu menambahkan, menyunting dan menghapus data.

Role	: Mahasiswa
Description of User	: Melakukan <i>read</i> data, <i>create</i> data, <i>update</i> data, dan <i>delete</i> data.
Prerequisite	: Mahasiswa mengakses aplikasi Android dengan perangkat Android yang sudah terkoneksi dengan koneksi internet.
Task description	: Mengakses pengumuman terkait <i>capstone</i> , melakukan pendaftaran <i>capstone</i> , mengakses informasi kelompok, dosen pembimbing, dan dosen penguji serta mengakses jadwal sidang, mengunggah dan mengunduh dokumen, mengelola data profil akun.

3.1.4 Arsitektur Sistem

Arsitektur sistem informasi pengelolaan *Capstone* TA versi Android dapat dilihat pada Gambar 3.10.



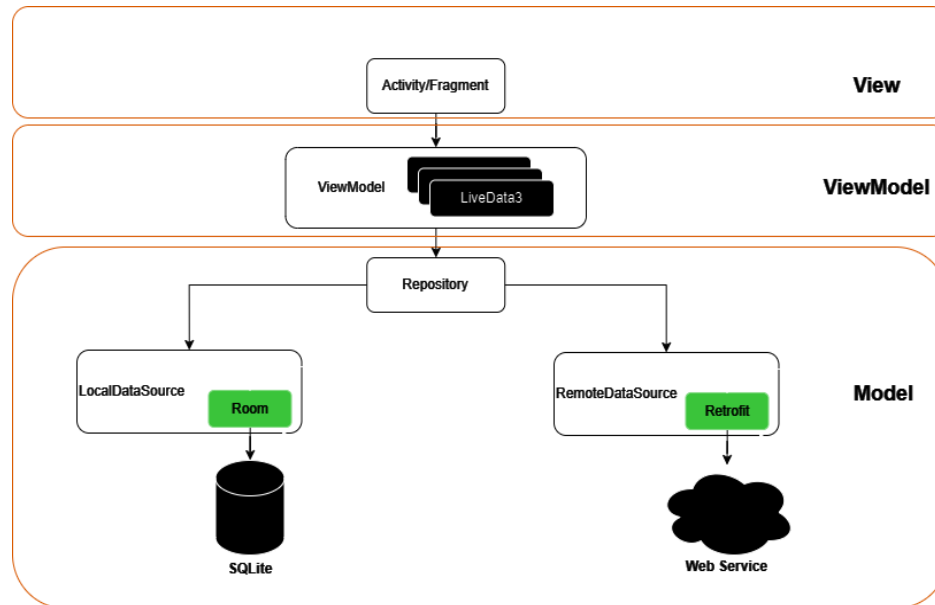
Gambar 3.2 Arsitektur sistem berbasis Android

Arsitektur komponen sistem berbasis Android memiliki tiga komponen utama, yaitu *client*, *internet*, dan *server* (*web server* dan *database server*). API akan menjadi jembatan transportasi antara perangkat *client* dengan basis data aplikasi. Masing-masing komponen memiliki hubungan yang dapat dilihat pada Tabel 3.3.

Tabel 3. 3 Hubungan komponen arsitektur sistem berbasis Android

Kode	Komponen Pengirim	Komponen Penerima	Media Transmisi	Metode Transmisi Data
C1	Web Server	Database Server	PHP Data Objects	PHP Data Objects
C2	Internet	Web Server	Internet	Hypertext Transfer Protocol Request
C3	Client	Internet	Internet	Hypertext Transfer Protocol Request dengan pustaka Retrofit

Aplikasi Android pada sistem informasi *Capstone* TA pada Departemen Teknik Komputer memiliki arsitektur pengembangan dapat dilihat pada Gambar 3.11.



Gambar 3.3 Arsitektur pengembangan aplikasi Android

Arsitektur pengembangan aplikasi Android menggunakan MVVM (*Model*, *View*, *ViewModel*) yaitu arsitektur pembuatan aplikasi yang berfokus pada pemisahan antara kode untuk logika bisnis dan tampilan aplikasi. Dalam penerapannya, MVVM terbagi atas beberapa *layer*, yaitu *Model*, *View*, dan *ViewModel*.

3.1.5 Antarmuka Eksternal

1. *Hardware Interface*

Perangkat keras yang digunakan sebagai antarmuka untuk interaksi manusia ke sistem adalah *touch screen*.

2. *Software Interface*

Sistem ini menggunakan antarmuka perangkat lunak berupa sistem operasi Android versi 5.0 (Lollipop) ke atas agar dapat digunakan untuk memasang dan menggunakan aplikasi pada Android.

3. *Communication Interface*

Sistem yang dikembangkan membutuhkan jaringan internet agar dapat bekerja dengan baik. Jaringan tersebut digunakan untuk berkomunikasi dengan basis data melalui REST API.

3.2 Menentukan Target Sistem

3.2.1 Ruang Lingkup Sistem

Sistem yang akan dikembangkan merupakan pengembangan, perbaikan *bug*, dan pembuatan versi Android dari sistem yang pernah dikembangkan sebelumnya, yaitu sistem pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro. Sistem memiliki dua versi, yaitu versi web dan versi Android. Sistem versi web memiliki 4 *role* pengguna yaitu superadmin, dosen, tim *capstone*, dan mahasiswa. Sedangkan sistem versi Android memiliki pengguna dengan *role* mahasiswa.

Penulis mengembangkan sistem berbasis Android dengan tujuan yaitu sistem informasi pengelolaan *Capstone* TA dapat diakses oleh mahasiswa melalui perangkat Android. Sistem ini dijalankan pada lingkungan Departemen Teknik Komputer Undip. Secara keseluruhan sistem, pihak-pihak yang terlibat dalam sistem ini adalah Superadmin, Tim *Capstone*, Dosen, dan Mahasiswa. Untuk sistem versi Android, sistem hanya akan melibatkan mahasiswa.

3.2.2 Proses Bisnis Sistem

Perlu diketahui bahwa Penulis hanya mengembangkan sistem versi Android dengan aktor mahasiswa, tetapi Penulis tetap mencantumkan aktor lain yaitu dosen dan tim *capstone* dikarenakan satu aktor dengan aktor yang lain saling terkait dan tidak dapat dipisahkan. Berikut merupakan daftar proses bisnis dari versi Android sistem informasi pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro:

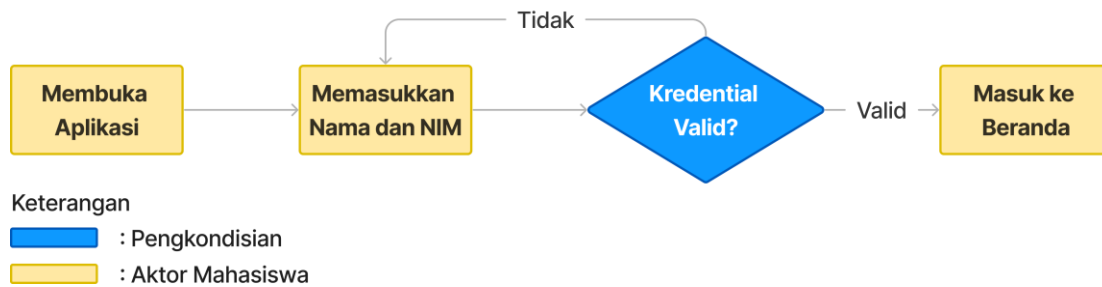
1. Proses bisnis mahasiswa masuk ke dalam akun mahasiswa melalui aplikasi Android.
2. Proses bisnis mahasiswa mengakses pengumuman terkait *capstone* TA melalui aplikasi Android.

3. Proses bisnis mahasiswa mendaftar *capstone* TA secara individu melalui aplikasi Android.
4. Proses bisnis mahasiswa mendaftar *capstone* TA secara berkelompok melalui aplikasi Android.
5. Proses bisnis mahasiswa mengakses dan mengelola informasi kelompok *capstone* melalui aplikasi Android.
6. Proses bisnis mahasiswa mengakses dan mengunggah dokumen *capstone* TA melalui aplikasi Android.
7. Proses bisnis mahasiswa mengakses waktu sidang proposal, penguji sidang proposal, dan tempat sidang proposal melalui aplikasi Android.
8. Proses bisnis mahasiswa mengakses informasi *expo* dan mendaftar *expo* melalui aplikasi Android.
9. Proses bisnis mahasiswa mengakses informasi sidang TA dan mendaftar sidang TA melalui aplikasi Android.
10. Proses bisnis mahasiswa mengubah profil dan kata sandi mahasiswa melalui aplikasi Android.

3.3 Perancangan Sistem

3.3.1 Perancangan Diagram Alur

Dari daftar proses bisnis sistem, Penulis akan menjabarkan proses bisnis tersebut dalam bentuk gambar diagram alur. Diagram alur adalah diagram yang berisi langkah atau proses untuk mengoperasikan sebuah program. Berikut ini gambar diagram alur mahasiswa masuk ke dalam akun mahasiswa melalui aplikasi Android dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram alur proses mahasiswa masuk ke aplikasi

Untuk masuk ke dalam halaman utama aplikasi, mahasiswa harus melakukan autentikasi pengguna terlebih dahulu, yaitu dengan memasukkan nama dan NIM mahasiswa. Jika autentikasi gagal, maka aplikasi akan menampilkan pesan gagal. Jika berhasil, maka mahasiswa dapat mengakses halaman beranda aplikasi.

Selanjutnya, berikut ini gambar diagram alur mahasiswa mengakses pengumuman terkait *capstone* TA melalui aplikasi Android dapat dilihat pada Gambar 3.5.

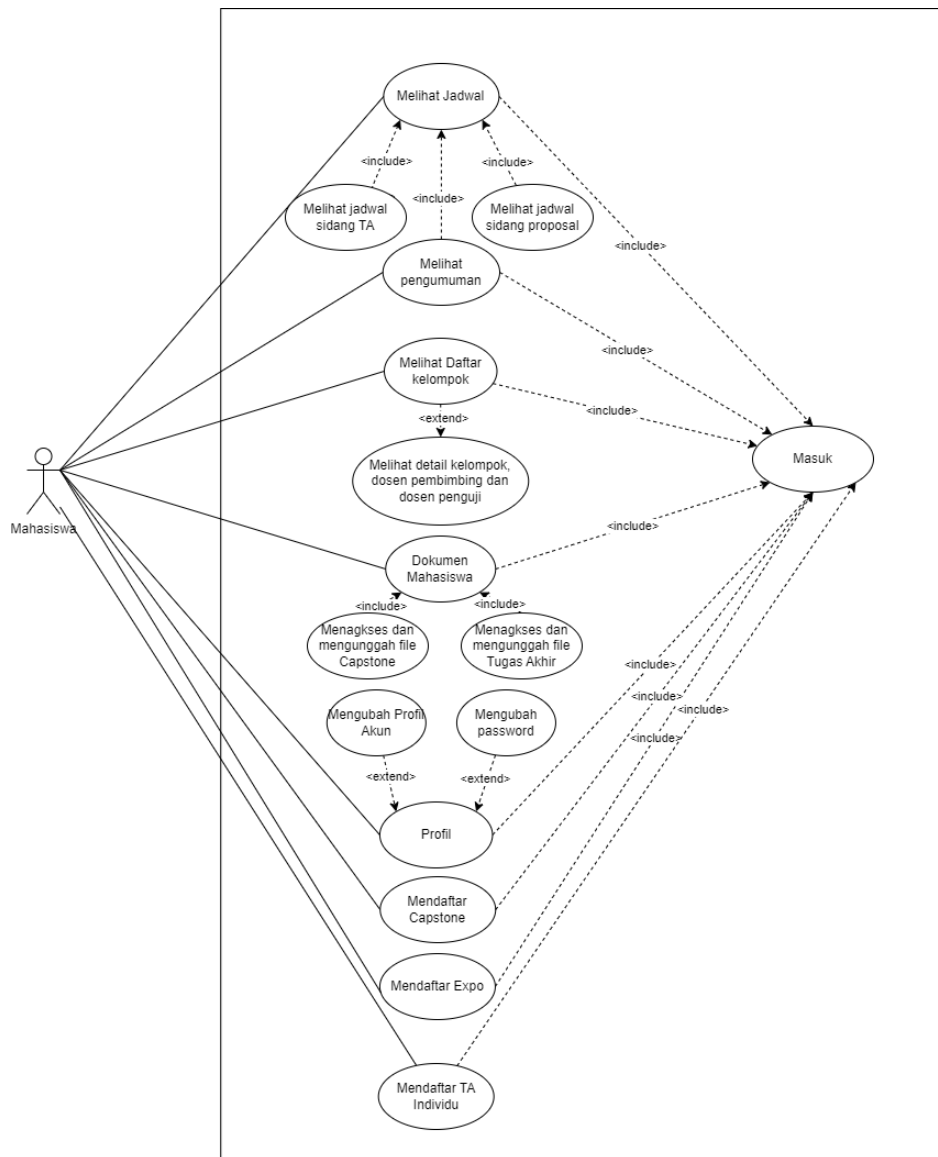


Gambar 3.5 Diagram alur proses mahasiswa mengakses pengumuman

Mahasiswa dapat mengakses pengumuman terbaru melalui halaman beranda. Pengumuman terbaru adalah pengumuman yang paling baru diunggah oleh Tim Capstone. Lalu, jika mahasiswa ingin mengakses semua pengumuman, mahasiswa dapat masuk ke halaman pengumuman. Dari pengumuman terbaru dan semua pengumuman, pengguna dapat masuk ke halaman detail pengumuman jika ingin melihat informasi detail mengenai pengumuman terkait.

3.3.2 Perancangan Diagram *Use Case*

Diagram *use case* adalah representasi grafis dari interaksi antara sistem dan aktor yang berperan dalam sistem tersebut. Diagram *Use case* digunakan dalam rekayasa perangkat lunak untuk menggambarkan fungsionalitas sistem dan bagaimana aktor berinteraksi dengan sistem tersebut dalam berbagai skenario atau situasi. Diagram *use case* sistem dapat dilihat pada Gambar 3.13.



Gambar 3.6 Diagram *use case*

3.3.3 Perancangan Skenario Use Case

Diagram *use case* sistem dijelaskan lebih lanjut dalam bentuk tabel pada skenario *use case*. Skenario *use case* mahasiswa masuk ke dalam aplikasi melalui perangkat Androi mahasiswa dijelaskan pada Tabel 3.4.

Tabel 3.4 Skenario *use case* masuk aplikasi versi Android

Use Case ID Number	1	
Use Case Name	Masuk Aplikasi	
Use Case Description	<i>Use Case</i> ini menggambarkan proses pengguna masuk ke dalam aplikasi menggunakan <i>credential</i> pengguna.	
Primary Actor	Mahasiswa	
Secondary Actor	-	
Pre-Condition	Perangkat mahasiswa memiliki koneksi internet.	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa membuka aplikasi.	
		2. Sistem menampilkan halaman “masuk”, dan meminta mahasiswa memasukkan <i>credential</i> berupa ID pengguna dan kata sandi pengguna.
	3. Mahasiswa memasukkan <i>credential</i> mahasiswa yaitu berupa ID pengguna dan kata sandi pengguna.	
		4. Menampilkan hasil percobaan masuk oleh pengguna
Error Flow of Events	3a. Mahasiswa salah memasukkan ID pengguna/kata sandi, atau perangkat pengguna belum terhubung ke internet.	
		4a. Menampilkan informasi yang menyebabkan mahasiswa gagal masuk ke aplikasi.
Post-Condition	Mahasiswa berhasil masuk aplikasi dan dapat menggunakan fitur yang ada di dalam aplikasi.	

Skenario *use case* mahasiswa melihat pengumuman dan mengakses informasi terkait *Capstone* TA dijelaskan pada Tabel 3.5.

Tabel 3.5 Skenario *use case* melihat pengumuman

Use Case ID Number	2
Use Case Name	Melihat Pengumuman
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa mengakses pengumuman terkait <i>Capstone</i> TA
Primary Actor	Mahasiswa

Secondary Actor	-	
Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet.	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa membuka halaman beranda aplikasi dengan perangkat yang telah terhubung ke koneksi internet.	
		2. Sistem menampilkan daftar pengumuman terbaru pada halaman beranda, dan menampilkan <i>action</i> untuk melihat semua daftar pengumuman.
	3. Mahasiswa memilih pengumuman yang hendak dibaca secara detail, kemudian melakukan klik pada pengumuman tersebut.	
		4. Sistem menampilkan detail pengumuman.
Error Flow of Events	1a. Mahasiswa membuka aplikasi dengan perangkat yang tidak terhubung ke koneksi internet.	
		2a. Sistem akan menampilkan pesan bahwa sistem gagal memuat data pengumuman.
Post-Condition	Mahasiswa dapat melihat dan membaca detail pengumuman, dan mengakses tautan pendukung (jika ada)	

Skenario *use case* mahasiswa mendaftar *capstone* dan mengakses informasi kelompok dijelaskan pada Tabel 3.6.

Tabel 3.6 Skenario *use case* mahasiswa mendaftar *capstone* dan mengakses informasi kelompok.

Use Case ID Number	3	
Use Case Name	Mendaftar <i>capstone</i> dan melihat informasi kelompok	
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa mendaftar <i>capstone</i> dan mengakses informasi mengenai kelompok <i>capstone</i> .	
Primary Actor	Mahasiswa	
Secondary Actor	Tim <i>capstone</i>	
Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet dalam periode pendaftaran <i>capstone</i> .	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa masuk ke halaman “Kelompok” menggunakan <i>bottom navigation</i> .	

		2. Jika mahasiswa sudah mendaftar <i>capstone</i> , maka sistem akan menampilkan informasi kelompok dan informasi dosen pembimbing kelompok. Jika mahasiswa belum mendaftar <i>capstone</i> , sistem akan menampilkan formulir pendaftaran <i>capstone</i> secara kelompok dan secara individu.
	3. Mahasiswa mengisi data pada formulir.	
	4. Mahasiswa memilih tombol <i>submit</i> untuk mengunggah data pada formulir.	
		5. Sistem menyimpan formulir mahasiswa ke <i>database</i> .
Error Flow of Events	1a. Mahasiswa membuka aplikasi dengan perangkat yang tidak terhubung ke koneksi internet.	
		2a. Sistem akan menampilkan pesan bahwa sistem gagal memuat data informasi kelompok.
	4a. Mahasiswa mengisi formulir dengan format yang salah.	
		5b. Sistem menampilkan pesan kesalahan format.
Post-Condition	Tim <i>capstone</i> dapat melihat dan mengelola data mahasiswa yang mendaftar <i>capstone</i>	

Skenario *use case* mahasiswa mengakses dan mengunggah dokumen dijelaskan pada Tabel 3.7.

Tabel 3.7 Skenario *use case* mahasiswa mengakses dan mengunggah dokumen

Use Case ID Number	4	
Use Case Name	Mengakses dan Mengunggah Dokumen	
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa mengakses dokumen dan mengunggah dokumen.	
Primary Actor	Mahasiswa	
Secondary Actor	Tim <i>capstone</i>	
Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet dan mahasiswa sudah memiliki kelompok <i>capstone</i> .	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu 'Dokumen' pada <i>bottom navigation</i> .	

		2. Sistem menampilkan halaman yang terdapat formulir untuk mengunggah dokumen.
	3. Mahasiswa mengunggah dokumen pada formulir.	
	4. Mahasiswa memilih tombol <i>upload</i> untuk mengunggah dokumen pada formulir.	
		5. Sistem menyimpan dokumen mahasiswa di <i>database</i> .
		6. Sistem menampilkan dokumen laporan pada halaman dosen dan tim capstone.
Error Flow of Events	3a. Mahasiswa mengunggah dokumen dengan ukuran dokumen yang melebihi ketentuan.	
		3b. Sistem menampilkan pesan kesalahan ukuran.
Post-Condition	Tim <i>capstone</i> dan dosen dapat melihat dokumen <i>capstone</i> dan tugas akhir mahasiswa.	

Skenario *use case* mahasiswa mengakses dan mengubah profile mahasiswa dijelaskan pada Tabel 3.8.

Tabel 3.8 Skenario *use case* profil mahasiswa

Use Case ID Number	5	
Use Case Name	Profil	
Use Case Description	<i>Use case</i> ini menggambarkan proses ketika seorang pengguna ingin melihat dan/atau mengubah data profil pengguna.	
Primary Actor	Mahasiswa	
Secondary Actor	Tim <i>capstone</i>	
Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet.	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu profil melalui <i>bottom navigation</i> .	
		2. Sistem menampilkan halaman profil pengguna.
	3. Mahasiswa memilih menu ubah data profil/ubah <i>password</i> di halaman profil.	
		4. Sistem menampilkan halaman ubah data profil/ubah <i>password</i> .
	5. Mahasiswa mengisi formulir dan menekan tombol simpan untuk menyimpan perubahan pada profil.	

		4. Sistem menampilkan halaman profil pengguna yang telah diubah.
Error Flow of Events	5a. Mahasiswa mengisi formulir tidak sesuai dengan ketentuan.	
		5b. Sistem menampilkan pesan kesalahan format.
Post-Condition	Mahasiswa dapat melihat data profil mahasiswa.	

Skenario *use case* mahasiswa yang sudah memiliki kelompok *capstone* mengakses jadwal sidang proposal C100 dijelaskan pada Tabel 3.9.

Tabel 3.9 Skenario *use case* mahasiswa melihat jadwal sidang proposal

Use Case ID Number	6	
Use Case Name	Melihat Jadwal Sidang Proposal	
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa yang sudah memiliki kelompok mengakses jadwal sidang proposal C100.	
Primary Actor	Mahasiswa	
Secondary Actor	-	
Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet. Mahasiswa sudah memiliki kelompok <i>capstone</i> .	
Primary Flow of Events	User Action	System Response
	1. Mahasiswa memilih menu “Sidang Proposal” pada halaman beranda.	
		2. Sistem jadwal sidang proposal dan menampilkan opsi unduh jadwal sidang proposal.
Error Flow of Events	1a. Mahasiswa membuka aplikasi dengan perangkat yang tidak terhubung ke koneksi internet.	
		2a. Sistem akan menampilkan pesan bahwa sistem gagal memuat data informasi sidang proposal.
Post-Condition	Mahasiswa yang sudah memiliki kelompok <i>capstone</i> dapat mengakses informasi mengenai jadwal sidang proposal kelompoknya.	

Skenario *use case* mahasiswa yang sudah memiliki kelompok *capstone* mengakses jadwal Expo dijelaskan pada Tabel 3.10.

Tabel 3.10 Skenario *use case* mahasiswa melihat jadwal expo dan mendaftar expo.

Use Case ID Number	7
Use Case Name	Melihat Jadwal Expo dan Mendaftar Expo
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa mengakses jadwal Expo Project dan Mendaftar Expo Project.
Primary Actor	Mahasiswa
Secondary Actor	Tim Capstone

Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet.	
Primary Flow of Events	User Action	Primary Flow of Events
	1. Mahasiswa memilih menu “Expo” pada halaman beranda.	
		2. Sistem menampilkan halaman informasi mengenai jadwal expo dan formulir pendaftaran expo.
	3. Perwakilan kelompok mahasiswa yang sudah memenuhi syarat untuk mengikuti expo, mengisi formulir pendaftaran expo.	
		4. Sistem menampilkan status pendaftaran expo.
Error Flow of Events	1a. Mahasiswa membuka aplikasi dengan perangkat yang tidak terhubung ke koneksi internet.	
		2a. Sistem akan menampilkan pesan bahwa sistem gagal memuat data Expo.
	3a. Mahasiswa memasukan formulir pendaftaran dengan tidak valid.	
		4a. Sistem menampilkan pesan kesalahan dan kegagalan pendaftaran.
Post-Condition	Mahasiswa mendaftar expo dan dapat melihat status pendaftarannya.	

Skenario *use case* mahasiswa yang sudah memenuhi syarat sidang Tugas

Akhir melihat dan mendaftar sidang Tugas Akhir dijelaskan pada Tabel 3.11.

Tabel 3.11 Skenario *use case* mahasiswa yang telah memenuhi syarat sidang Tugas Akhir mendaftar Sidang Tugas Akhir

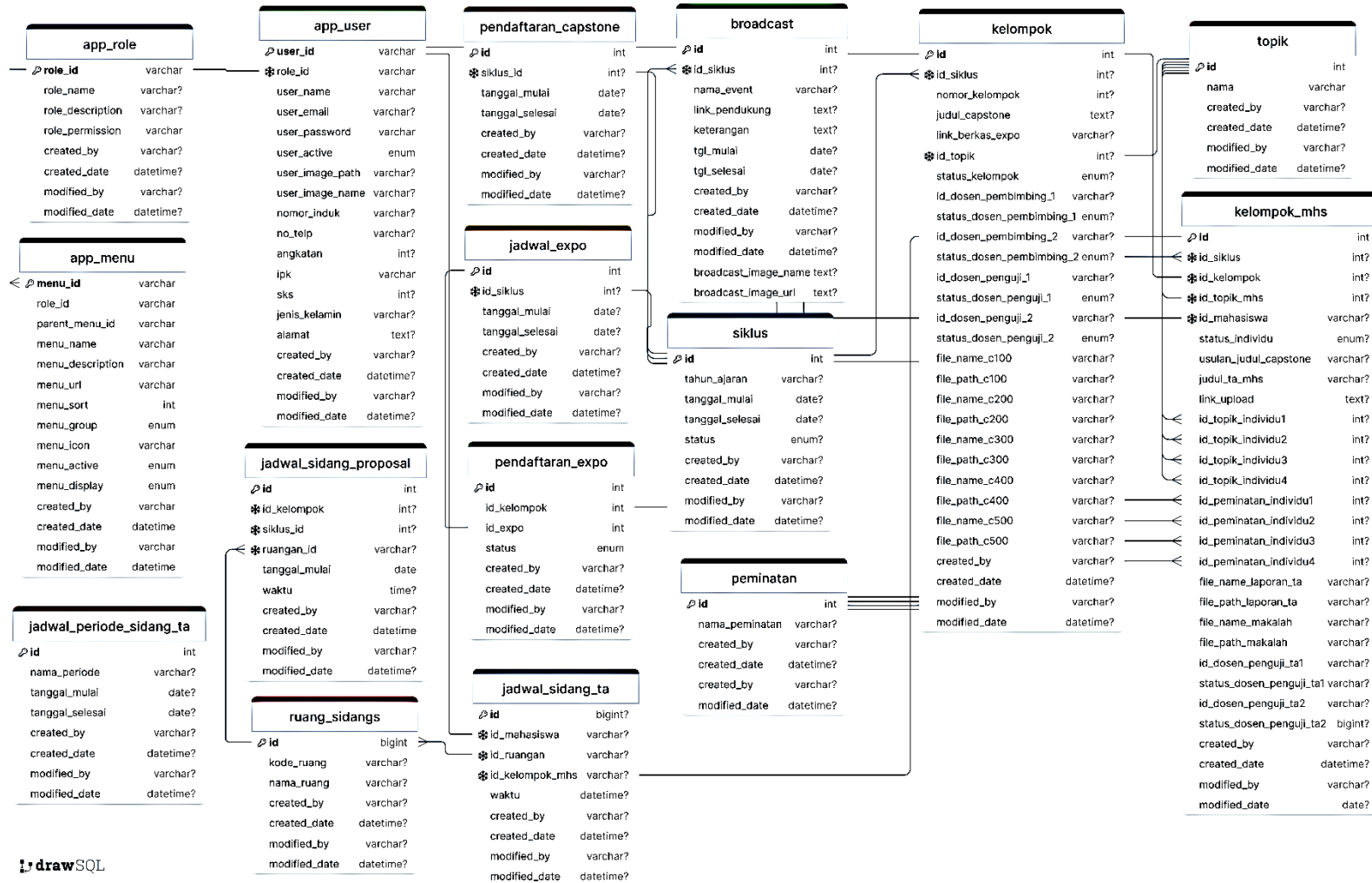
Use Case ID Number	8	
Use Case Name	Melihat dan Mendaftar Sidang Tugas Akhir	
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa mengakses jadwal sidang Tugas Akhir dan Mendaftar sidang Tugas Akhir.	
Primary Actor	Mahasiswa	
Secondary Actor	Tim Capstone	
Pre-Condition	Mahasiswa telah masuk ke dalam aplikasi, dan perangkat mahasiswa terhubung dengan koneksi internet.	
Primary Flow of Events	User Action	Primary Flow of Events
	1. Mahasiswa memilih menu “Sidang Tugas Akhir” pada halaman beranda.	

		2. Sistem menampilkan halaman informasi mengenai jadwal sidang Tugas Akhir dan formulir pendaftaran sidang Tugas Akhir.
	3. Mahasiswa yang sudah memenuhi syarat untuk mengikuti sidang Tugas Akhir, mengisi formulir pendaftaran sidang Tugas Akhir.	
		4. Sistem menampilkan status pendaftaran sidang Tugas Akhir
Error Flow of Events	1a. Mahasiswa membuka aplikasi dengan perangkat yang tidak terhubung ke koneksi internet.	
		2a. Sistem akan menampilkan pesan bahwa sistem gagal memuat data sidang Tugas Akhir.
	3a. Mahasiswa memasukan formulir pendaftaran dengan tidak valid.	
		4a. Sistem menampilkan pesan kesalahan dan kegagalan pendaftaran.
Post-Condition	Mahasiswa mendaftar expo dan dapat melihat status pendaftarannya.	

3.3.6 Perancangan Basis Data

1. ERD

Entity Relationship Diagram (ERD) adalah representasi visual yang menggunakan notasi grafis untuk menggambarkan hubungan dan interaksi antara entitas atau objek dalam suatu sistem *database*[15]. ERD digunakan sebagai alat bantu yang penting dalam fase perancangan *database*, membantu pengembang untuk memahami struktur data, mengidentifikasi ketergantungan, dan merancang hubungan antar tabel. Dengan menampilkan elemen-elemen seperti entitas, atribut, dan hubungan, ERD memberikan gambaran yang jelas tentang bagaimana data saling terkait dan bagaimana mereka akan disimpan dan dikelola dalam *database* yang sedang dibuat, menunjukkan bahwa ERD berfungsi sebagai panduan visual yang diperlukan dalam mengembangkan sistem *database*, dan diagram ERD dari sistem yang sedang dikembangkan dapat ditemukan pada Gambar 3.14.



Gambar 3. 7 ERD sistem

2. Deskripsi Tabel Basis Data

Terdapat beberapa tabel yang menjelaskan mengenai deskripsi dari data-data pada *database*. Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel *app_menu* seperti yang ditunjukkan pada Tabel 3.12.

Tabel 3.12 Struktur tabel *app_menu*

Nama Kolom	Tipe Data	Default	Constraint
menu_id	VARCHAR (3)	<i>None</i>	<i>Primary Key</i>
role_id	VARCHAR (3)	NULL	<i>Foreign Key</i>
parent_menu_id	VARCHAR (3)	NULL	-
menu_name	VARCHAR (50)	NULL	-
menu_description	VARCHAR (100)	NULL	-
menu_url	VARCHAR (100)	NULL	-
menu_sort	INT (10)	NULL	-
menu_group	ENUM	utama	-
menu_icon	VARCHAR (50)	NULL	-
menu_active	ENUM	1	-
menu_display	ENUM	1	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel *app_role* seperti yang ditunjukkan pada Tabel 3.13.

Tabel 3.13 Struktur tabel *app_role*

Nama Kolom	Tipe Data	Default	Constraint
role_id	VARCHAR (3)	<i>None</i>	<i>Primary Key</i>
role_name	VARCHAR (100)	NULL	-
role_description	VARCHAR (100)	NULL	-
role_permission	VARCHAR (4)	1000	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel *app_user* seperti yang ditunjukkan pada Tabel 3.4.

Tabel 3.14 Struktur tabel *app_user*

Nama Kolom	Tipe Data	Default	Constraint
user_id	VARCHAR (15)	None	Primary Key
user_name	VARCHAR (50)	None	-
user_email	VARCHAR (50)	NULL	-
role_id	VARCHAR (3)	NULL	Foreign Key
user_password	VARCHAR (255)	None	-
user_active	ENUM	1	-
user_img_path	VARCHAR (100)	NULL	-
user_img_name	VARCHAR (200)	NULL	-
nomor_induk	VARCHAR (25)	NULL	-
no_telp	VARCHAR (15)	NULL	-
angkatan	INT (4)	NULL	-
ipk	VARCHAR (4)	NULL	-
sks	INT (3)	NULL	-
jenis_kelamin	VARCHAR (20)	NULL	-
alamat	TEXT	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel *broadcast* seperti yang ditunjukkan pada Tabel 3.15.

Tabel 3.15 Struktur tabel *broadcast*

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	None	Primary Key
id_siklus	INT (4)	NULL	Foreign Key
nama_event	VARCHAR (200)	NULL	-
link_pendukung	TEXT	NULL	-
keterangan	TEXT	NULL	-
tgl_mulai	DATE	NULL	-
tgl_selesai	DATE	NULL	-
broadcast_image_name	VARCHAR (255)	NULL	-
broadcast_image_path	VARCHAR (255)	NULL	-
created_by	VARCHAR (15)	NULL	-

Tabel 3.15 Struktur tabel broadcast (lanjutan)

Nama Kolom	Tipe Data	Default	Constraint
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel jadwal_expo seperti yang ditunjukkan pada Tabel 3.16.

Tabel 3.16 Struktur tabel jadwal_expo

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	<i>None</i>	<i>Primary Key</i>
id_siklus	INT (4)	NULL	<i>Foreign Key</i>
tanggal_mulai	DATE	NULL	-
tanggal_selesai	DATE	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel jadwal_sidang_proposal seperti yang ditunjukkan pada Tabel 3.17.

Tabel 3.17 Struktur tabel jadwal_sidang_proposal

Nama Kolom	Tipe Data	Default	Constraint
id	INT (5)	<i>None</i>	<i>Primary Key</i>
id_kelompok	INT (5)	NULL	<i>Foreign Key</i>
siklus_id	INT (4)	NULL	<i>Foreign Key</i>
waktu	DATETIME	NULL	-
ruangan_id	INT (5)	NULL	<i>Foreign Key</i>
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel *jadwal_sidang_ta* seperti yang ditunjukkan pada Tabel 3.18.

Tabel 3.18 Struktur tabel *jadwal_sidang_ta*

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	<i>None</i>	<i>Primary Key</i>
id_mahasiswa	VARCHAR (15)	<i>None</i>	<i>Foreign Key</i>
id_ruangan	VARCHAR (4)	<i>None</i>	<i>Foreign Key</i>
id_kelompok_mhs	VARCHAR (15)	<i>None</i>	<i>Foreign Key</i>
waktu	DATETIME	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel kelompok seperti yang ditunjukkan pada Tabel 3.19.

Tabel 3.19 Struktur tabel kelompok

Nama Kolom	Tipe Data	Default	Constraint
id	INT (5)	<i>None</i>	<i>Primary Key</i>
id_siklus	INT (4)	NULL	<i>Foreign Key</i>
nomor_kelompok	INT (25)	NULL	-
id_topik	INT (2)	NULL	<i>Foreign Key</i>
status_kelompok	ENUM	NULL	-
judul_capstone	VARCHAR (255)	NULL	-
link_berkas_expo	VARCHAR (255)	NULL	-
id_dosen_pembimbing_1	VARCHAR (15)	NULL	-
status_dosen_pembimbing_1	ENUM	NULL	-
id_dosen_pembimbing_2	VARCHAR (15)	NULL	-
status_dosen_pembimbing_2	ENUM	NULL	-
id_dosen_penguji_1	VARCHAR (15)	NULL	-
status_dosen_penguji_1	ENUM	NULL	-
id_dosen_penguji_2	VARCHAR (15)	NULL	-
status_dosen_penguji_2	ENUM	NULL	-
file_path_c100	VARCHAR (200)	NULL	-
file_name_c200	VARCHAR (200)	NULL	-
file_path_c200	VARCHAR (200)	NULL	-
file_name_c300	VARCHAR (200)	NULL	-

Tabel 3.19 Struktur tabel kelompok (lanjutan)

Nama Kolom	Tipe Data	Default	Constraint
file_path_c300	VARCHAR (200)	NULL	-
file_name_c400	VARCHAR (200)	NULL	-
file_path_c400	VARCHAR (200)	NULL	-
file_name_c500	VARCHAR (200)	NULL	-
file_path_c500	VARCHAR (200)	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel kelompok_mhs seperti yang ditunjukkan pada Tabel 3.20.

Tabel 3.20 Struktur tabel kelompok_mhs

Nama Kolom	Tipe Data	Default	Constraint
id	INT (5)	<i>None</i>	<i>Primary Key</i>
id_siklus	INT (4)	NULL	<i>Foreign Key</i>
id_kelompok	INT (5)	NULL	<i>Foreign Key</i>
id_mahasiswa	VARCHAR (15)	NULL	<i>Foreign Key</i>
id_topik_mhs	INT (4)	NULL	<i>Foreign Key</i>
status_individu	ENUM	NULL	-
link_upload	TEXT	NULL	-
usulan_judulan_capstone	VARCHAR (300)	NULL	-
judul_ta_mhs	VARCHAR (100)	NULL	-
id_topik_individu1	INT (4)	NULL	<i>Foreign Key</i>
id_topik_individu2	INT (4)	NULL	<i>Foreign Key</i>
id_topik_individu3	INT (4)	NULL	<i>Foreign Key</i>
id_topik_individu4	INT (4)	NULL	<i>Foreign Key</i>
id_peminatan_individu1	INT (4)	NULL	<i>Foreign Key</i>
id_peminatan_individu2	INT (4)	NULL	<i>Foreign Key</i>
id_peminatan_individu3	INT (4)	NULL	<i>Foreign Key</i>
id_peminatan_individu4	INT (4)	NULL	<i>Foreign Key</i>
file_name_laporan_ta	VARCHAR (300)	NULL	-
file_path_laporan_ta	VARCHAR (200)	NULL	-
file_name_makalah	VARCHAR (300)	NULL	-
file_path_makalah	VARCHAR (200)	NULL	-

Tabel 3.20 Struktur tabel kelompok_mhs (lanjutan)

Nama Kolom	Tipe Data	Default	Constraint
id_dosen_penguji_ta1	VARCHAR (15)	NULL	-
status_dosen_penguji_ta1	VARCHAR (255)	NULL	-
id_dosen_penguji_ta2	VARCHAR (15)	NULL	-
status_dosen_penguji_ta2	VARCHAR (255)	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel peminatan seperti yang ditunjukkan pada Tabel 3.21.

Tabel 3.21 Struktur tabel peminatan

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	<i>None</i>	<i>Primary Key</i>
nama_peminatan	VARCHAR (255)	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel pendaftaran_capstone seperti yang ditunjukkan pada Tabel 3.22.

Tabel 3.22 Struktur tabel pendaftaran_capstone

Nama Kolom	Tipe Data	Default	Constraint
id	INT (5)	<i>None</i>	<i>Primary Key</i>
siklus_id	INT (4)	NULL	<i>Foreign Key</i>
tanggal_mulai	DATE	NULL	-
tanggal_selesai	DATE	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel pendaftaran_expo seperti yang ditunjukkan pada Tabel 3.23.

Tabel 3.23 Struktur tabel pendaftaran_expo

Nama Kolom	Tipe Data	Default	Constraint
id	INT (15)	<i>None</i>	<i>Primary Key</i>
id_kelompok	INT (5)	NULL	<i>Foreign Key</i>
id_expo	INT (4)	NULL	<i>Foreign Key</i>
status	ENUM	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel jadwal_periode_sidang_ta seperti yang ditunjukkan pada Tabel 3.24.

Tabel 3.24 Struktur tabel jadwal_periode_sidang_ta

Nama Kolom	Tipe Data	Default	Constraint
id	INT (5)	<i>None</i>	<i>Primary Key</i>
siklus_id	INT (4)	NULL	<i>Foreign Key</i>
tanggal_mulai	DATE	NULL	-
tanggal_selesai	DATE	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel ruang_sidangs seperti yang ditunjukkan pada Tabel 3.25.

Tabel 3.25 Struktur tabel ruang_sidangs

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	<i>None</i>	<i>Primary Key</i>
kode_ruang	VARCHAR (255)	NULL	-
nama_ruang	VARCHAR (255)	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel siklus *capstone* seperti yang ditunjukkan pada Tabel 3.26.

Tabel 3.26 Struktur tabel siklus

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	<i>None</i>	<i>Primary Key</i>
tahun_ajaran	VARCHAR (100)	NULL	-
tanggal_mulai	DATE	NULL	-
tanggal_selesai	DATE	NULL	-
status	ENUM	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel topik *capstone* seperti yang ditunjukkan pada Tabel 3.27.

Tabel 3.27 Struktur tabel topik

Nama Kolom	Tipe Data	Default	Constraint
id	INT (4)	<i>None</i>	<i>Primary Key</i>
nama	VARCHAR (200)	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

3.3.7 Perancangan *End-Point*

Pada pembuatan sistem informasi pengelolaan *Capstone* TA versi Android pada Departemen Teknik Komputer Universitas Diponegoro, akan terdapat sekumpulan REST API *endpoint* dari aplikasi *backend* yang akan digunakan untuk berkomunikasi antara sistem versi Android dan basis data sistem. Daftar REST API *endpoint* untuk yang diperlukan untuk sistem informasi versi Android seperti yang tertampil pada Tabel 3.28.

Tabel 3.28 *Endpoint REST API*

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/auth/login/	<p><i>Endpoint</i> ini digunakan untuk melakukan <i>login</i> mahasiswa dan verifikasi mahasiswa valid Departemen Teknik Komputer Universitas Diponegoro. Data yang dikirimkan dalam <i>request body</i> adalah nomor induk dan <i>password</i> mahasiswa.</p> <pre>{ "nomor_induk": string, "password": string , }</pre>
GET	/api/v1/auth/logout	<p><i>Endpoint</i> ini digunakan untuk melakukan <i>logout</i> mahasiswa dan verifikasi mahasiswa valid Departemen Teknik Komputer Universitas Diponegoro. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>
GET	/api/v1/mahasiswa/profile	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mendapatkan data mahasiswa. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>
POST	/api/v1/mahasiswa/profile/editProcess	<p><i>Endpoint</i> ini digunakan mahasiswa untuk melakukan perubahan data mahasiswa. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> sebagai berikut.</p> <pre>{ "user_name": string, "no_telp": string, "user_email": string?, }</pre>

Tabel 3.28 *Endpoint REST API* (lanjutan)

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/mahasiswa/ /profile/editPassword	<p><i>Endpoint</i> ini digunakan mahasiswa untuk melakukan perubahan <i>password</i> mahasiswa. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> sebagai berikut.</p> <pre>{ "current_password": string, "new_password": string, "repeat_new_password": string, }</pre>
GET	api/v1/mahasiswa/ broadcast-home	<i>Endpoint</i> ini digunakan mahasiswa untuk mendapatkan data pengumuman terbaru mengenai <i>Capstone TA</i> .
GET	api/v1/mahasiswa/ broadcast	<i>Endpoint</i> ini digunakan mahasiswa untuk mendapatkan data pengumuman mengenai <i>Capstone TA</i> .
POST	/api/v1/mahasiswa/ /broadcast/detail- broadcast	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mendapatkan data detail pengumuman mengenai <i>capstone</i> dan tugas akhir.</p> <p>Data yang dikirimkan dalam <i>request body</i> sebagai berikut.</p> <pre>{ "id_broadcast": string, }</pre>
GET	/api/v1/mahasiswa/ /kelompok	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mendapatkan data kelompok <i>Capstone TA</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>

Tabel 3.28 *Endpoint REST API* (lanjutan)

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/mahasiswa /kelompok/add- kelompok-process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk melakukan pendaftaran <i>capstone</i> secara individu. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> sebagai berikut.</p> <pre>{ "id_siklus": string, "email": string, "angkatan": string, "jenis_kelamin": string, "ipk": string, "sks": string, "no_telp": string, "s": string, (peminatan) "e": string, (peminatan) "c": string, (peminatan) "m": string, (peminatan) "ews": string, (topik) "bac": string, (topik) "smb": string, (topik) "smc": string, (topik) }</pre>
GET	/api/v1/mahasiswa /upload-file	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mendapatkan data dokumen <i>Capstone TA</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>

Tabel 3.28 *Endpoint REST API* (lanjutan)

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/mahasiswa/ kelompok/add- punya-kelompok- process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk melakukan pendaftaran <i>capstone</i> secara berkelompok oleh salah satu anggota kelompok. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> sebagai berikut.</p> <pre>{ "judul_capstone": string, "id_topik": string, "dosbing_1": string, "dosbing_2": string, "angkatan1": string, "email1": string, "jenis_kelamin1": string, "ipk1": string, "sks1": string, "no_telp1": string, "user_id2": string, "angkatan2": string, "email2": string, "jenis_kelamin2": string, "ipk2": string, "sks2": string, "no_telp2": string, "user_id3": string, "angkatan3": string, "email3": string, "jenis_kelamin3": string, "ipk3": string, "sks3": string, "no_telp3": string, }</pre>
POST	/api/v1/mahasiswa/ upload- file/upload- makalah-process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengunggah dokumen makalah tugas akhir. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "makalah": Part File , }</pre>

Tabel 3.28 *Endpoint REST API* (lanjutan)

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/mahasiswa /upload- file/upload- laporan-process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengunggah dokumen laporan tugas akhir. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "laporan_ta": Part File , }</pre>
POST	/api/v1/mahasiswa /upload- file/upload-c100- process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengunggah dokumen c100 <i>capstone</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "c100": Part File , "id_kelompok": string , }</pre>
POST	/api/v1/mahasiswa /upload- file/upload-c200- process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengunggah dokumen c200 <i>capstone</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu <i>Authorization</i> dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "c200": Part File , "id_kelompok": string , }</pre>

Tabel 3.28 *Endpoint REST API* (lanjutan)

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/mahasiswa/ upload-file/upload-c300- process	<p><i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "c300": Part File , "id_kelompok": string , }</pre>
POST	/api/v1/mahasiswa/ upload-file/upload-c400- process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengunggah dokumen c400 <i>captone</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "c400": Part File , "id_kelompok": string , }</pre>
POST	/api/v1/mahasiswa/ upload-file/upload-c500- process	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengunggah dokumen c500 <i>captone</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "c500": Part File , "id_kelompok": string , }</pre>
GET	/api/v1/mahasiswa/ /sidang-proposal- kelompok	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengakses jadwal sidang proposal kelompok. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>
GET	/api/v1/mahasiswa/ /expo	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengakses jadwal <i>expo</i> yang tersedia, sekaligus melihat status <i>expo</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>

Tabel 3.28 *Endpoint REST API (lanjutan)*

<i>Method</i>	<i>Route Path</i>	<i>Keterangan</i>
POST	/api/v1/mahasiswa /expo-daftar	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mendaftar <i>expo project</i>. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "id_expo": string , "link_berkas_expo": string , "judul_ta_mhs": string , }</pre>
GET	/api/v1/mahasiswa /sidang-tugas-akhir-mahasiswa	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mengakses jadwal sidang Tugas Akhir mahasiswa. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre>
POST	/api/v1/mahasiswa /sidang-tugas-akhir-daftar	<p><i>Endpoint</i> ini digunakan mahasiswa untuk mendaftar sidang Tugas Akhir secara individu. <i>Endpoint</i> ini memerlukan <i>Header</i> yaitu Authorization dengan token.</p> <pre>{ "Authorization": "Bearer api_token" }</pre> <p>Data yang dikirimkan dalam <i>request body</i> adalah sebagai berikut.</p> <pre>{ "link_upload": string , "judul_ta_mhs": string , }</pre>

3.4 Metode Pengujian

Perangkat lunak yang sedang dikembangkan dalam proyek ini akan dievaluasi menggunakan metode *black box*, *unit testing*, dan *load testing*. Metode *black box* digunakan untuk memastikan fungsi keseluruhan aplikasi bekerja dengan baik tanpa memperhatikan struktur internalnya. *Unit testing* akan fokus pada pengujian setiap komponen kode secara terpisah, sementara *load testing* bertujuan untuk mengukur performa aplikasi dalam kondisi beban tinggi.

- *Black box*: Memastikan aplikasi Android dapat mengirimkan permintaan (*request*) dan menerima tanggapan (*response*) yang benar dari API sistem. Pengujian akan dilakukan menggunakan perangkat Android Studio, kemudian mengakses halaman terkait dan memastikan adanya permintaan dan tanggapan yang sesuai.
- *Unit testing*: Memastikan kode yang dikembangkan oleh penulis dalam mengembangkan API bebas dari kesalahan. Unit testing menguji masing-masing fungsi yang ada di dalam kontroler dan model kode, kemudian membuat beberapa kasus, dan memastikan hasil aktual sama dengan hasil yang diharapkan (*expected result*).
- *Load testing*: Digunakan untuk menguji performa API yang dikembangkan. Load testing akan dilakukan menggunakan Postman dan mengukur berapa lama API memberikan tanggapan terhadap permintaan pengguna.

BAB IV
HASIL DAN PEMBAHASAN

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil

5.2. Saran

Dengan adanya.

DAFTAR PUSTAKA

- [1] D. T. Komputer, *Panduan Pelaksanaan Capstone Tugas Akhir 2023*. 2023.
- [2] P. P. Arhandi, “Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan dengan Menggunakan Metode Back End dan Front End,” *J. Teknol. Inf.*, vol. 7, no. 1, pp. 39–48, 2016.
- [3] Y. K. V. Makarenko, O. Olshevska, “No Titlean Architectural approach for Quality Improving of Android Applications Development Which Implemented To Communication Application foe Mechatronics Robot Laboratory Onaft,” *J. Teknol.*, vol. 9, no. 3, pp. 8–13, 2017, [Online]. Available: <https://doi.org/10.15673/atbp.v9i3.714>
- [4] R. I. M.A. Dharmawan, “Implementasi Sistem Informasi Tugas Akhir Menggunakan Metode Classic Life Cycle,” *Pros. Semnas Inotek Ke-III Tahun 2019*, vol. 3, no. 1, 2019.
- [5] E. K. S.M Sari, L.M Tua, “Sistem Informasi Monitoring Pembimbingan Skripsi/Tugas Akhir (SIMP-S/TA) Berbasis Android,” *Semin. Nas. Inform. Sist. Inf. dan Keamanan Siber 2019*, vol. 2, no. 1, 2019.
- [6] D. Q. U. A. Lander, “Desain dan Implementasi REST-API Pada Website Marketplace Katering ‘Ketringan’ Menggunakan Framework Laravel,” *J. e-Proceeding Eng.*, vol. 8, no. 5, p. 106, 2021.
- [7] A. B. Nurwicaksono, “Perancangan Sistem Manajemen Kasir Berbasis Android pada Blessing of Shoes,” *J. Inf. Syst. Manag.*, vol. 5, no. 2, pp. 108–114, 2024, [Online]. Available: <https://doi.org/10.24076/joism.2024v5i2.1111>
- [8] H. H. D. Murdiani, “Perbandingan Metode Waterfall dan RAD (Rapid Application Development) pada Pengembangan Sistem Informasi,” *J. Teknol. Inf.*, vol. 6, no. 1, pp. 14–23, 2022, [Online]. Available: <https://doi.org/10.51401/jinteks.v4i4.2008>
- [9] M. F. Ridhoi, “Rancang Bangun Mobile Learning Berbasis Android di SMKN Poncol,” in *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2022.

- [10] I. M. Riyadhi, “Penerapan Pola Arsitektur MVVM Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android,” *J. Infotech*, vol. 9, no. 1, pp. 146–158, 2023, [Online]. Available: <https://doi.org/10.31949/infotech.v9i1.5246>
- [11] A. Thohari, “Implementasi Test Driven Development Dalam Pengembangan Aplikasi Berbasis Web,” *SITECH J. Sist. Inf. Dan Teknol.*, vol. 1, no. 1, pp. 1–10, 2018, [Online]. Available: <https://doi.org/10.24176/sitech.v1i1.2255>
- [12] B. H. Hasanuddin, “Rancang Bangun REST API Aplikasi Weshare sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan,” *J. Inform. Teknol. dan Sains*, vol. 4, no. 1, p. 8, 2022, [Online]. Available: <https://doi.org/10.51401/jinteks.v4i1.1474>
- [13] D. D. J. T. Sitinjak, “Analisa dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris pada Intensive English Course di Ciledug Tangerang,” *J. IPSIKOM*, vol. 8, no. 1, 2020.
- [14] M. Subianto, “Penerapan Metode Rapid Application Development dalam Perancangan Sistem Informasi Pendataan,” *J. Infokam*, vol. 16, no. 1, pp. 47–49, 2020, [Online]. Available: <https://doi.org/10.53845/infokam.v16i1.218>
- [15] R. W. D.P. Sari, “Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang),” *J. Inform. Dan RPL*, vol. 2, no. 1, pp. 32–36, 2019.
- [16] B. R. W.G. Wardhana, I. Arwani, “Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 2, pp. 680–689, 2020.
- [17] Y. B. W. S. Sibuea, M.I. Saputro, A. Annan, “Aplikasi Mobile Collection Berbasis Android pada PT. Suzuki Finance Indonesia,” *J. Inform. Dan Teknol. Komput.*, vol. 2, no. 1, pp. 31–42, 2022.
- [18] Arribe Edo, “Aplikasi E-marketplace Menggunakan Arsitektur MVVM (Model-

- view-viewmodel) Berbasis Android,” *J. Fasilkom*, vol. 11, no. 2, pp. 75–78, 2021, [Online]. Available: <https://doi.org/10.37859/jf.v11i2.2762>
- [19] T. D. Annisa Nauli Hasibuan, “Pengujian dengan Unit Testing dan Test case pada Proyek Pengembangan Modul Manajemen Pengguna,” *Sect. Res. Penelit.*, vol. 2, no. 1, 2021.
- [20] Y. D. Wijaya, “Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan PT INKA (Persero) Berbasis Equivalence Partitions,” *J. Digit. Teknol. Inf.*, vol. 4, no. 1, pp. 22–26, 2021.
- [21] G. H. Setiawan, “Pengujian Performa API (Application Programming Interface) dengan Metode Load Testing,” in *SEMINAR NASIONAL CORISINDO*, 2022, pp. 539–542.
- [22] E. T. L. Setiyani, “Analisis Kebutuhan Fungsional Aplikasi Penanganan Keluhan Mahasiswa Studi Kasus: STMIK Rosma Karawang,” *J. Inov. Pendidik. dan Teknol. Inf.*, vol. 2, no. 1, p. 11, 2021, [Online]. Available: <https://doi.org/10.52060/pti.v2i01.465>

LAMPIRAN 1
BIODATA MAHASISWA



Nama Mahasiswa : Nanda Iqbal Hanafi
NIM : 21120120130109
Konsentrasi : Perangkat Lunak
Tempat/Tgl. Lahir : Kulon Progo/31 Desember 2001
Alamat Sekarang : Jalan Sirojuddin, Gg. Margoyoso No.
25, Tembalang, Tembalang, Kota
Semarang
No. Telepon/HP : 083101487741
Alamat e-mail : nandaiqbalhanafii@gmail.com
Nama orang tua : Ponijan
Alamat orang tua : RT13/RW06, Sanggrahan Lor,
Bendungan, Wates, Kulon Progo

Semarang, 12 November 2023

Nanda Iqbal Hanafi