



UNIVERSITAS DIPONEGORO

**IMPLEMENTASI POLA DESAIN MVVM PADA
PEMBUATAN BACKEND ANDROID DAN RESTFUL API UNTUK
SISTEM PENGOLAAN 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 : Implementasi Pola Desain MVVM Pada Pembuatan Backend
Android Dan RESTFUL API untuk Sistem Pengolaan
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 : Semarang, 21 Februari 2024

**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:

Implementasi Pola Desain MVVM Pada Pembuatan Backend Android Dan RESTFUL API untuk Sistem Pengolaan 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 **“Implementasi Pola Desain MVVM Pada Pembuatan Backend Android Dan RESTFUL API untuk Sistem Pengolaan 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 Staf 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	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan	6
BAB II KAJIAN PUSTAKA	8
2.1 Kajian Penelitian Terdahulu.....	8
2.2 Landasan Teori.....	14
2.2.1 Capstone TA	14
2.2.2 Sistem Informasi.....	16
2.2.3 Android	16
2.2.4 Kotlin	16
2.2.5 REST API.....	16
2.2.6 <i>Model View ViewModel</i>	17
2.2.7 PHP17	
2.2.8 Laravel.....	18

2.2.9	Postman	18
2.2.10	Android Studio	19
2.2.11	Pengujian <i>Black Box</i>	19
2.2.12	Pengujian <i>Usability</i>	19
2.2.13	Metode RAD	19
BAB III PERANCANGAN SISTEM		21
3.1.	Gambaran Umum Sistem	21
3.1.1.	Fungsi Utama Produk	21
3.1.2.	Karakteristik Pengguna	21
3.2.	Proses Bisnis Saat Ini	22
3.2.1.	Proses Rekayasa/Proses Bisnis	22
3.2.2.	Prosedur	23
3.3.	Target Sistem	24
3.3.1.	Ruang Lingkup Sistem	24
3.3.2.	Proses Rekayasa	24
3.3.3.	Prosedur	32
3.3	Perancangan Sistem	33
3.3.1.	Arsitektur Sistem	33
3.3.2.	Antarmuka Eksternal	35
3.3.3.	Perancangan Diagram Alur	35
3.3.4.	Perancangan Diagram <i>Use Case</i>	36
3.3.5.	Perancangan Skenario <i>Use Case</i>	37
3.4.6.	Perancangan Basis Data	44
3.4.7.	Perancangan End-Point	53
3.4.8.	Kebutuhan Fungsional	61
3.4.9.	Kebutuhan Non-Fungsional	61
3.4	Metode Pengujian	62
BAB IV HASIL DAN PEMBAHASAN		63
BAB V PENUTUP		64

5.1. Kesimpulan	64
5.2. Saran.....	64
DAFTAR PUSTAKA	65
LAMPIRAN 1 BIODATA MAHASISWA	69

DAFTAR GAMBAR

Gambar 3.1 Diagram BPMN <i>Capstone</i> TA	22
Gambar 3.2 Diagram proses pendaftaran dan pengelompokan <i>Capstone</i> TA	25
Gambar 3.3 Diagram proses bimbingan dan <i>tracking</i> progres kelompok	26
Gambar 3.4 Diagram proses penjadwalan sidang C100	27
Gambar 3.5 Diagram proses unggah dokumen C100, C200, dan C300	28
Gambar 3.6 Diagram proses unggah dokumen lengkap	29
Gambar 3.7 Diagram proses <i>Expo Project</i>	30
Gambar 3.8 Diagram proses pendaftaran Tugas Akhir.....	31
Gambar 3.9 Diagram proses penjadwalan sidang Tugas Akhir.....	32
Gambar 3.10 Arsitektur sistem berbasis Android	33
Gambar 3.11 Arsitektur pengembangan aplikasi Android.....	34
Gambar 3.12 Diagram alur dari Sistem Informasi Pengelolaan <i>Capstone</i> TA berbasis Android	36
Gambar 3.13 Diagram <i>use case</i>	37
Gambar 3.14 ERD Sistem	45

DAFTAR TABEL

Tabel 2.1 Perbandingan penelitian Dharmawan dkk. dengan penelitian penulis	8
Tabel 2.2 Perbandingan penelitian Sari dkk. dengan penelitian penulis	9
Tabel 2.3 Perbandingan penelitian Lander dkk. dengan penelitian penulis.....	9
Tabel 2.4 Perbandingan penelitian Nurwicaksono dkk. dengan penelitian penulis....	10
Tabel 2.5 Perbandingan penelitian Mardiani dkk dengan penelitian penulis	10
Tabel 2.6 Perbandingan penelitian Ridhoi dkk dengan penelitian penulis	10
Tabel 2.7 Perbandingan penelitian Riyadhhi dkk dengan penelitian penulis	11
Tabel 2.8 Perbandingan penelitian Purnamasari dkk dengan penelitian penulis	11
Tabel 2.9 Tabel kajian penelitian terdahulu	12
Tabel 3.1 Prosedur <i>Capstone</i> TA	23
Tabel 3.2 Tabel prosedur proses pengelolaan <i>Capstone</i> TA.....	32
Tabel 3.3 Hubungan komponen arsitektur sistem berbasis Android	34
Tabel 3.4 Skenario <i>use case</i> masuk aplikasi versi Android	37
Tabel 3.5 Skenario <i>use case</i> melihat pengumuman	38
Tabel 3.6 Skenario <i>use case</i> mahasiswa mendaftar <i>capstone</i> dan mengakses informasi kelompok.....	39
Tabel 3.7 Skenario <i>use case</i> mahasiswa mengakses dan mengunggah dokumen.....	40
Tabel 3.8 Skenario <i>use case</i> profil mahasiswa.....	41
Tabel 3.9 Skenario <i>use case</i> mahasiswa melihat jadwal sidang proposal.....	41
Tabel 3.10 Skenario <i>use case</i> mahasiswa melihat jadwal expo dan mendaftar expo. 42	
Tabel 3.11 Skenario <i>use case</i> mahasiswa yang telah memenuhi syarat sidang Tugas Akhir mendaftar Sidang Tugas Akhir	43
Tabel 3.12 Struktur tabel app_menu	46
Tabel 3.13 Struktur tabel app_role.....	46
Tabel 3.14 Struktur tabel app_user	47
Tabel 3.15 Struktur tabel broadcast	47
Tabel 3.16 Struktur tabel jadwal_expo	48
Tabel 3.17 Struktur tabel jadwal_sidang_proposal	48

Tabel 3.18 Struktur tabel jadwal_sidang_ta.....	49
Tabel 3.19 Struktur tabel kelompok.....	49
Tabel 3.20 Struktur tabel kelompok_mhs	50
Tabel 3.21 Struktur tabel peminatan	51
Tabel 3.22 Struktur tabel pendaftaran_capstone	51
Tabel 3.23 Struktur tabel pendaftaran_expo	52
Tabel 3.24 Struktur tabel jadwal_periode_sidang_ta.....	52
Tabel 3.25 Struktur tabel ruang_sidangs.....	52
Tabel 3.26 Struktur tabel siklus	53
Tabel 3.27 Struktur tabel topik	53
Tabel 3.28 <i>Endpoint</i> REST API.....	54
Tabel 3.29 Tabel kebutuhan fungsional sistem.....	61
Tabel 3.30 Tabel kebutuhan non-fungsional sistem	61

ABSTRAK

Kata Kunci:

ABSTRACT

Keywords:

BAB I

PENDAHULUAN

1.1 Latar Belakang

Departemen Teknik Komputer, Fakultas Teknik, Universitas Diponegoro mengintegrasikan konsentrasi Informatika dan Komputer (INFOKOM) dalam upaya membekali para lulusan menjadi ahli di bidang INFOKOM. Mahasiswa akan mempelajari ilmu di bidang jaringan komputer, pemrograman, robotika dan multimedia didukung oleh dosen pengajar yang berkompeten di bidangnya.

Salah satu persyaratan kelulusan adalah menyelesaikan *Capstone* Tugas Akhir (*Capstone* TA). Ini merupakan model pelaksanaan Tugas Akhir yang mengharuskan mahasiswa menerapkan pengetahuannya langsung dalam waktu dua semester atau satu tahun. Selama *Capstone* TA, mahasiswa dapat mengaplikasikan semua 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 dalam desain teknik. [1]. Sistem *capstone* diterapkan di Departemen Teknik Komputer Universitas Diponegoro sejak Oktober 2022, dengan pengerjaan berkelompok maksimal 3 mahasiswa dan mengikuti *timeline capstone*.

Meskipun Departemen Teknik Komputer telah memiliki sistem informasi pengelolaan *Capstone* TA, yaitu pada <https://capstone-ta.ce.undip.ac.id/>, sistem ini masih terbatas pada pengumuman departemen, *timeline capstone*, dan format dokumen. Belum ada mekanisme pendaftaran dan penjadwalan di dalam sistem tersebut. Oleh karena itu, proses pendaftaran, bidding kelompok, *plotting* dosen pembimbing, penjadwalan sidang, dan hal terkait *capstone* masih menggunakan sistem manual dengan Google Form dan Google Spreadsheet.

Pada siklus 1 tahun 2022, ada kelompok *capstone* yang telah 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 browser. Oleh karena itu, diperlukan pengembangan fitur, perbaikan bug, dan pengembangan versi Android dari sistem informasi ini.

Dalam penerapannya, sistem informasi versi Android harus terintegrasi dalam satu basis data dengan sistem informasi versi web agar data-data sistem terhubung secara *real-time*. Hal tersebut dapat dipenuhi dengan REST 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 REST API untuk memenuhi kebutuhan pembuatan sistem informasi versi Android.

Kemudian, pembuatan sistem informasi versi Android memerlukan sisi *frontend* yaitu tampilan sistem yang akan berinteraksi dengan pengguna dan *backend* yang menangani sisi pengelolaan data dan integrasi data dengan *frontend*. Sistem informasi versi Android dikembangkan dengan bahasa pemrograman Kotlin dan menerapkan *pattern Model-View-View-Model* (MVVM *pattern*). Salah satu kesalahan yang sering dilakukan oleh *developer* pada saat proses pengembangan aplikasi di antaranya yaitu penulisan kode logika proses bisnis dan kode logika tampilan yang ditulis dalam satu *class* yang sama. Hal ini dapat menyebabkan kode menjadi susah untuk dibaca, diperbaharui, dites, dan dirawat.

Kurangnya penerapan *design pattern* dan arsitektur pada pengembangan sebuah aplikasi juga menyebabkan kesulitan bagi *developer* untuk memahami *flow* dari kode proyek tersebut. Apalagi jika proyek tersebut sering mengalami pergantian *developer*, maka *developer* baru akan mengalami kesulitan untuk membaca dan

memahami kode pada proyek tersebut. Untuk menghindari hal tersebut dapat dilakukan melalui penggunaan *design pattern* Model View ViewModel (MVVM). Penerapan *design pattern* MVVM memungkinkan mempertahankan data dari perubahan konfigurasi yang terjadi pada perangkat pengguna, serta mempermudah akses data [3].

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, rumusan suatu masalah yang akan dibahas pada Tugas Akhir kali ini yaitu sebagai berikut.

1. Bagaimana mengintegrasikan sistem versi Android dengan basis data yang digunakan dalam sistem informasi *Capstone* TA?
2. Bagaimana pembuatan REST API (*Representational State Transfer Application Programming Interface*) yang diperlukan untuk komunikasi antara basis data dan sistem versi Android pada sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro?
3. Bagaimana merancang aplikasi Android dengan menggunakan menggunakan *design pattern Model-View-ViewModel* (MVVM)?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian yang dilakukan ini adalah sebagai berikut.

1. Merancang dan membuat REST API (*Representational State Transfer Application Programming Interface*) sebagai media komunikasi perangkat Android dengan basis data MySQL.
2. Merancang dan membuat sistem informasi pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro versi Android.
3. Menerapkan kerangka-kerja Android *Native* dalam merancang dan membuat 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 kerangka-kerja Android *Native* dan berkomunikasi dengan basis-data MySQL menggunakan REST API (*Representational State Transfer Application Programming Interface*).
3. Aplikasi yang dikembangkan berjalan hanya saat perangkat Android terhubung dengan jaringan internet.

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 ke dalam 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 sistem versi Android yang dibuat menggunakan bahasa pemrograman Kotlin dan menerapkan *pattern Model-View-View Model* (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, untuk memperoleh hasil yang akurat dan relevan digunakanlah metode penelitian. Berikut metode 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 implementasi desain sistem terlebih dahulu kemudian implementasi antarmuka aplikasi dan dilanjutkan dengan implementasi REST API.

Penulis menggunakan Kotlin sebagai bahasa pemrograman dengan kerangka-kerja Android *Native* dan *pattern Model-View-View Model* (MVVM).

5. Pengujian Sistem

Tahap akhir dari pengembangan sistem informasi *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro versi Android adalah pengujian sistem. Penulis menggunakan pengujian *black box* untuk menguji implementasi agar sesuai dengan tujuan pembuatan dan pengujian *usability* untuk mengidentifikasi masalah uji ketergunaan [4].

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 1 PENDAHULUAN

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

BAB II TINJAUAN PUSTAKA

Dalam bab kedua terdapat penjelasan mengenai konsep dasar dalam penelitian dan pengembangan sistem. Tinjauan Pustaka ini 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 REST API (*Representational State Transfer Application Programming Interface*) dan pembuatan sistem informasi

pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro versi Android.

BAB IV HASIL DAN PEMBAHASAN

Berisi tulisan dan gambar yang menjelaskan hasil dari tahap implementasi sistem informasi pengelolaan *Capstone* TA Departemen Teknik Komputer Universitas Diponegoro versi Android.

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*”[5] 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”[6], 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 Java
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”[7], 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” [8], 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” [9], 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” [10], 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>blackbox testing</i> dan <i>usability 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”[11], 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</i> TA
Metode pengembangan yang digunakan adalah metode SCRUM	Metode pengembangan yang digunakan adalah metode <i>Rapid Application Development</i>

Pada penelitian yang dilakukan Purnamasari dkk pada tahun 2020 dengan penelitian yang berjudul, “Implementasi *Usability Testing* dalam Evaluasi Website Sekolah”[12], dijelaskan bahwa Sample yang diambil dari populasi adalah pengguna yang nantinya akan dijadikan responden dalam usability testing. Pengguna yang akan dijadikan sample harus mewakili dari seluruh populasi (pengguna). Terdapat beberapa perbedaan antara penelitian Purnamasari dkk. dan penelitian penulis yang akan dijelaskan secara lebih rinci dengan perbandingan pada Tabel 2.8.

Tabel 2.8 Perbandingan penelitian Purnamasari dkk dengan penelitian penulis

Penelitian Purnamasari dkk	Penelitian Penulis
<i>Usability testing</i> digunakan untuk menguji sistem berbasis web	<i>Usability testing</i> digunakan untuk menguji sistem berbasis Android

Penelitian terdahulu memiliki peran penting sebagai sumber referensi bagi penulis dalam menyelesaikan karya ilmiahnya. 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.	Penelitan 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.	Penelitan 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.	Penelitan menggunakan metode SCRUM, dengan menerapkan Nginx sebagai perangkat <i>web server</i> .	Laravel <i>framework</i> dapat menjadi solusi dalam pengembangan REST API. Imlementasi 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 Oleh: Riyadhhi dkk (2022)	Membangun aplikasi pengaduan masyarakat berbasis Android dengan menerapkan arsitektur MVVM [8].	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 pemeliharaan kode yang telah ada

Tabel 2.9 Tabel kajian penelitian terdahulu (lanjutan)

No	Judul dan Peneliti	Tujuan	Metode Penelitian	Hasil
8	Implementasi Usability Testing dalam Evaluasi Website Sekolah Oleh: Purnamasari dkk (2020)	Melihat tingkat <i>usability</i> dari website SMK Negeri Sumsel	Metode pengujian yang digunakan adalah metode <i>usability testing</i> .	Disimpulkan bahwa pengukuran <i>usability</i> pada variabel <i>Learnability</i> dinyatakan sudah cukup baik, karena bagi pengguna cukup mengerti dan memahami pada saat mengunjungi situs website SMK Negeri Sumsel Palembang

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. 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 *blackbox testing* dan *usability* dapat digunakan untuk memastikan kualitas aplikasi Android.

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. Sementara itu, menurut H. F. Hoffman dalam bukunya yang berjudul “*The Engineering*

Capstone Course”, para insinyur lulusan teknik harus memiliki banyak atribut yang dibutuhkan di tempat kerja mereka yang berhubungan dengan teknik. Atribut tersebut tidak hanya mencakup dasar-dasar teknik dan pemahaman proses manufaktur saja, tetapi juga pemahaman terkait kebutuhan ekonomi, lingkungan, dan sosial. 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. Tidak seluruh bagian harus dirancang dari *scratch* (dapat berupa *redesign* atau *reprocessing*) namun harus mengandung perancangan *hardware*, *software*, simulasi, atau kombinasi dari ketiga hal tersebut. Aspek kebaruan dalam penelitian bukan menjadi hal utama yang harus dipenuhi dalam *TA* ini. *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

2.2.3 Android

2.2.4 Kotlin

Kotlin ialah bahasa pemrograman yang diakui oleh Google sebagai bahasa resmi pemrograman Android yang dapat bekerja sama dengan pustaka yang dibuat dari bahasa Java[10]. Kotlin memiliki beberapa keunggulan dibandingkan dengan Java. Pertama, Kotlin dirancang untuk bekerja secara *interoperable* 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.

2.2.5 REST API

Representatif State Transition Application Programming Interface (REST 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 [13].

2.2.6 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 [14]. MVVM terbagi atas beberapa *layer*, yaitu *Model*, *View*, dan *ViewModel*.

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 [15].

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* [15].

3. *ViewModel*

ViewModel merupakan *layer* yang digunakan untuk berinteraksi kepada model untuk meneruskan data pada *View* [15].

2.2.7 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[16]. *Web base programming*, pada umumnya menggunakan bahasa pemrograman php, dengan pasangan aplikasi pengelola database mySQL.

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 *database* seperti MySQL dan PostgreSQL,

serta integrasi yang mudah dengan HTML, memungkinkan pengembang mencampur logika pemrograman dengan konten web untuk pengembangan aplikasi dinamis.

2.2.8 Laravel

Laravel adalah sebuah framework web berbasis PHP yang open-source dan tidak berbayar, diciptakan oleh Taylor Otwell dan diperuntukkan untuk pengembangan aplikasi web yang menggunakan pola MVC. Struktur pola MVC pada laravel sedikit berbeda pada struktur pola MVC pada umumnya. Di laravel terdapat routing yang menjembatani antara request dari user dan controller. Jadi controller tidak langsung menerima request tersebut[17]. 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.

Laravel juga menawarkan ORM yang kuat, yaitu *Eloquent*, yang memudahkan pengembang dalam berinteraksi dengan database menggunakan objek dan mengurangi kerumitan dalam manipulasi data. Dukungan bawaan untuk fitur-fitur seperti otentikasi, manajemen sesi, dan *caching* membuat pengembangan lebih cepat dan efisien.

2.2.9 Postman

Postman adalah sebuah aplikasi (berupa plugin) untuk browser chrome, yang berfungsi sebagai REST Client, yang digunakan untuk melakukan uji coba REST API[18]. 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.

2.2.10 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform* android[19]. Android studio ini berbasis pada IntelliJ IDEA. Untuk membuat tampilan atau layout, 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.

2.2.11 Pengujian *Black Box*

Pengujian *Black Box* merupakan salah satu jenis metode pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para penguji memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tapi cukup dikenai proses pengujian dibagian luar [20].

2.2.12 Pengujian *Usability*

Pengujian *usability* adalah salah satu metode dalam evaluasi *usability* yang mengobservasi penggunaan sebuah aplikasi kemudian diambil data dan menganalisisnya. Pengujian ini bertujuan untuk mengidentifikasi masalah kegunaan, mengumpulkan data kuantitatif dan kualitatif serta menentukan kepuasan pengguna terhadap sistem [21].

2.2.13 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 [16]. Berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi.

1. *Requirements Planning* (Perencanaan Syarat-Syarat)

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.

2. *RAD Design Workshop* (Workshop Desain RAD)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna.

3. *Implementation* (Implementasi)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi

BAB III

PERANCANGAN SISTEM

3.1. Gambaran Umum Sistem

3.1.1. Fungsi Utama Produk

Sistem yang penulis kembangkan memiliki fungsi utama sebagai sistem informasi bagi mahasiswa Departemen Teknik Komputer untuk mengakses informasi mengenai *Capstone* TA pada Departemen Teknik Komputer. Sistem ini diharapkan mampu membantu mahasiswa dalam mengakses informasi termasuk pendaftaran, jadwal, dan unggah dokumen.

3.1.2. 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 dengan tujuan utama yaitu mengakses informasi, mengatur profil pengguna, melakukan pendaftaran, dan melakukan unggah dokumen. Maka dari itu, hak akses yang dimiliki mahasiswa yaitu menambahkan, menyunting dan menghapus data.

Description of User : Melakukan *read* data, *create* data, *update* data, dan *delete* data.

Role : Mahasiswa

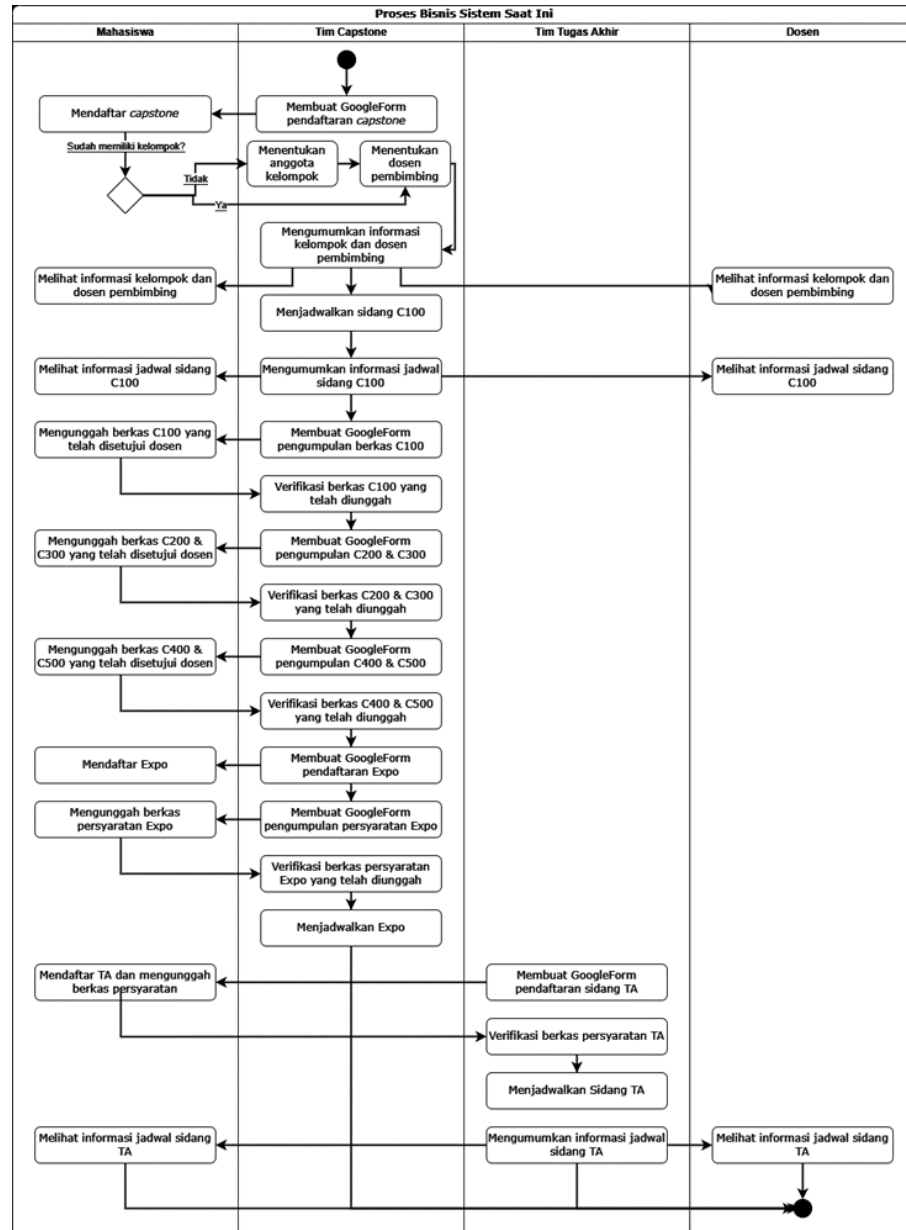
Prerequisite : Mahasiswa mengakses aplikasi versi android dengan perangkat Android yang sudah terkoneksi dengan koneksi internet. Mahasiswa melakukan *login* terlebih dahulu untuk dapat menggunakan sistem.

Task description : Melakukan pendaftaran, mengunggah dan mengunduh dokumen, menerima informasi, mengelola data profil akun.

3.2. Proses Bisnis Saat Ini

3.2.1. Proses Rekayasa/Proses Bisnis

Saat ini, proses manajemen *Capstone* TA di Departemen Teknik Komputer Universitas Diponegoro dikelola oleh tim *capstone* menggunakan GoogleForm dan GoogleSpreadsheet. Proses pengelolaan *Capstone* TA dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram BPMN *Capstone* TA

Proses manajemen meliputi proses pendaftaran *capstone*, *plotting* kelompok, *plotting* dosen pembimbing, penjadwalan sidang C100, pendaftaran *expo*, pendaftaran TA dan penjadwalan sidang TA.

Departemen Teknik Komputer sudah mengembangkan sistem informasi pengelolaan *Capstone* TA berbasis web. Sistem Informasi *capstone* TA di Departemen Teknik Komputer Universitas Diponegoro saat ini masih memiliki *bug* dan memerlukan tambahan fitur serta memerlukan pengembangan sistem berbasis Android yang berfokus pada *role* mahasiswa.

3.2.2. Prosedur

Terdapat beberapa prosedur pada *role* mahasiswa dalam proses *Capstone* TA di Departemen Teknik Komputer Universitas Diponegoro. Prosedur rinci dalam proses *Capstone* TA dapat dilihat pada Tabel 3.1.

Tabel 3.1 Prosedur *Capstone* TA

Aktor	Aksi	Prosedur
Mahasiswa	Mendaftar <i>capstone</i> .	Mahasiswa mengisi <i>form</i> pada GoogleForm yang dibuat oleh tim <i>capstone</i> .
Mahasiswa	Melakukan bimbingan dokumen dengan dosen pembimbing.	Mahasiswa menghubungi dosen, kemudian mengirim dokumen melalui surel.
Mahasiswa	Mendaftar <i>expo</i> .	Mahasiswa mengunggah dokumen dan mengisi <i>form</i> pada GoogleForm yang dibuat oleh tim <i>capstone</i> .
Mahasiswa	Mendaftar TA.	Mahasiswa mengunggah dokumen dan mengisi <i>form</i> pada GoogleForm yang dibuat oleh tim <i>capstone</i> .
Mahasiswa	Melihat kelompok, dosen pembimbing, dan jadwal sidang.	Mahasiswa mengakses GoogleSpreadsheet yang dibuat oleh tim <i>capstone</i> .

3.3. Target Sistem

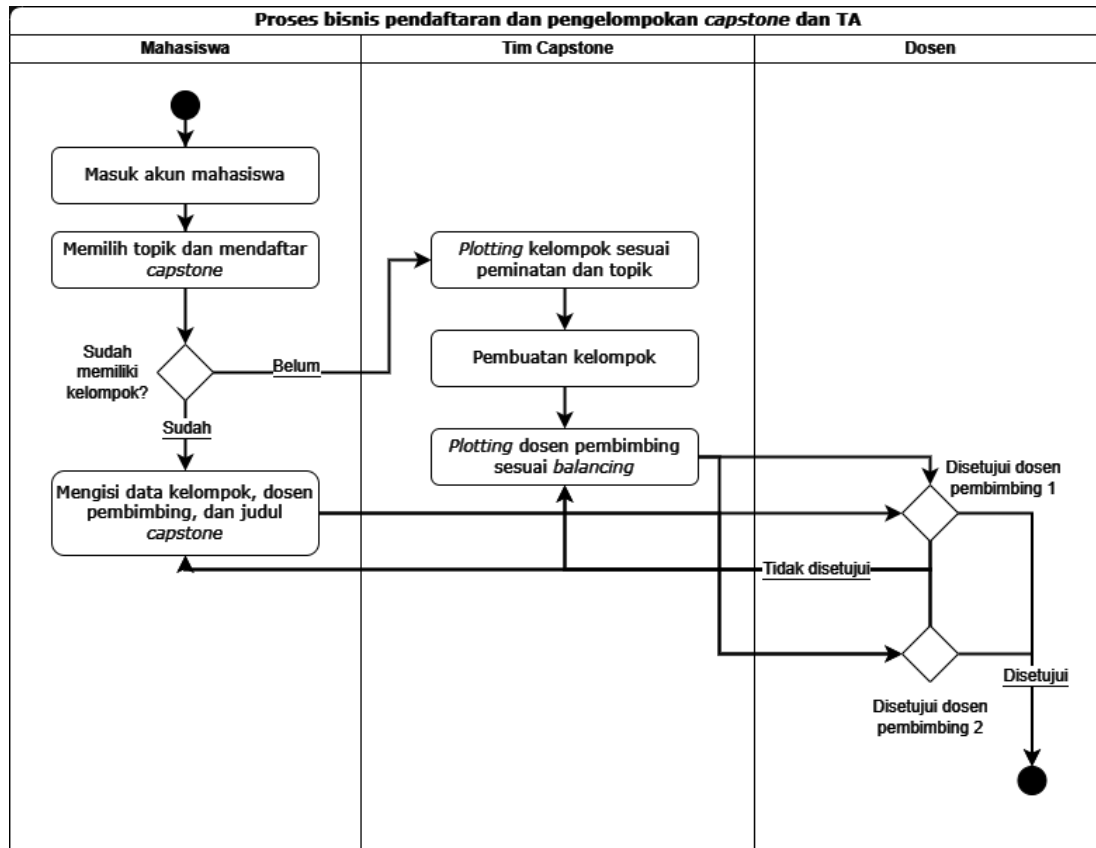
3.3.1. Ruang Lingkup Sistem

Sistem yang akan dikembangkan merupakan pengembangan, perbaikan *bug*, dan pembuatan versi Android dari sistem yang pernah dikembangkan sebelumnya. Sistem yang akan dikembangkan terdiri dari sistem pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro. Sistem memiliki dua versi, yaitu versi web dan versi Android. Sistem web memiliki 4 *role* pengguna yaitu superadmin, dosen, tim *capstone*, dan mahasiswa. Sedangkan sistem berbasis 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. Mahasiswa dapat melakukan proses pendaftaran, melihat informasi, mengelola dokumen, dan mengelola profil.

3.3.2. Proses Rekayasa

Proses rekayasa dari sistem informasi pengeolaan yang akan dikembangkan berupa web dan aplikasi Android. Berikut merupakan gambaran proses rekayasa baru dari sistem informasi pengelolaan *Capstone* TA pada Departemen Teknik Komputer Universitas Diponegoro. Proses bisnis pengelolaan *Capstone* TA di Departemen Teknik Komputer dimulai dengan pendaftaran dan pengelompokan mahasiswa. Proses bisnis dan pengelompokan *Capstone* TA dapat dilihat pada Gambar 3.2.

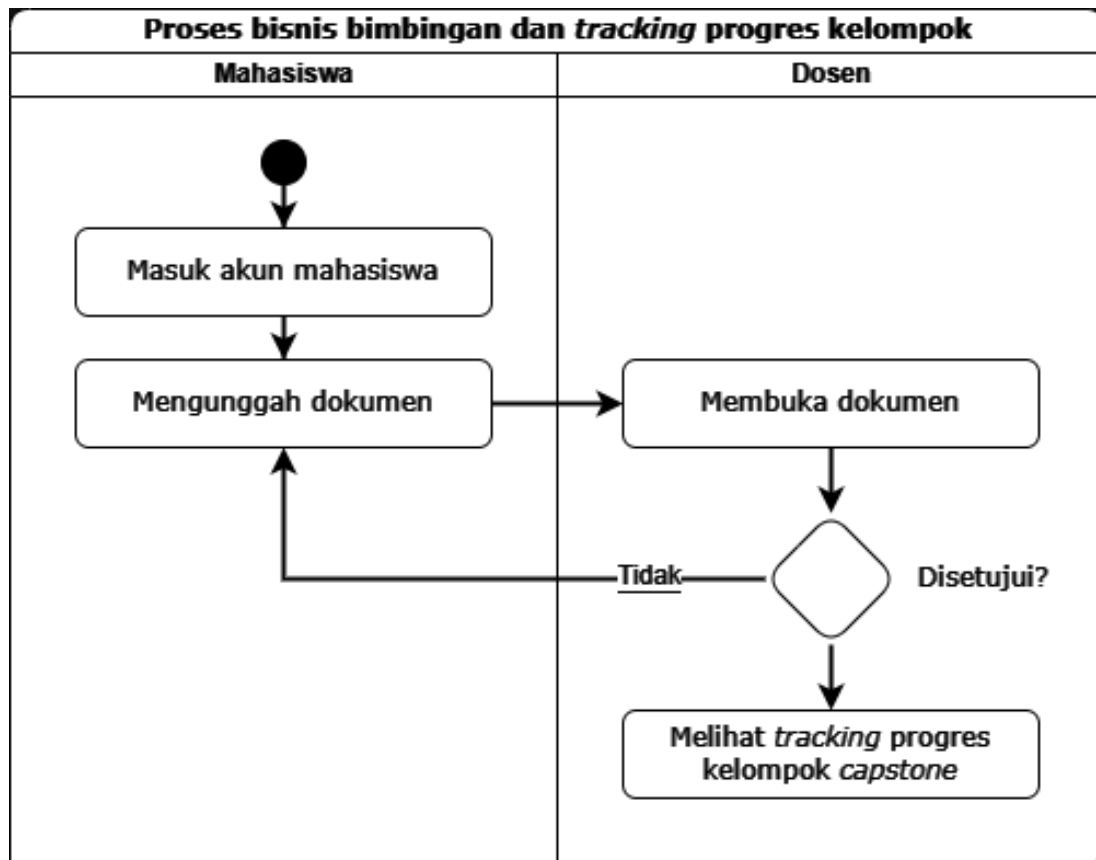


Gambar 3.2 Diagram proses pendaftaran dan pengelompokan *Capstone* TA

Mahasiswa dapat mendaftar *capstone* dengan status sudah berkelompok dan mendaftar *capstone* dengan status mendaftar secara individu. Jika mendaftar dengan status sudah berkelompok, maka mahasiswa diminta untuk mengisi data kelompok dan judul yang diajukan. Jika mahasiswa mendaftar secara individu, maka *plotting* kelompok dan dosen pembimbing akan dilakukan oleh tim *capstone* dengan mempertimbangkan peminatan, topik, dan *balancing* dosen pembimbing. Setelah kelompok *capstone* selesai dibuat, maka akan diajukan ke dosen pembimbing, apakah kelompok tersebut disetujui oleh dosen pembimbing atau ada revisi sebelum disetujui.

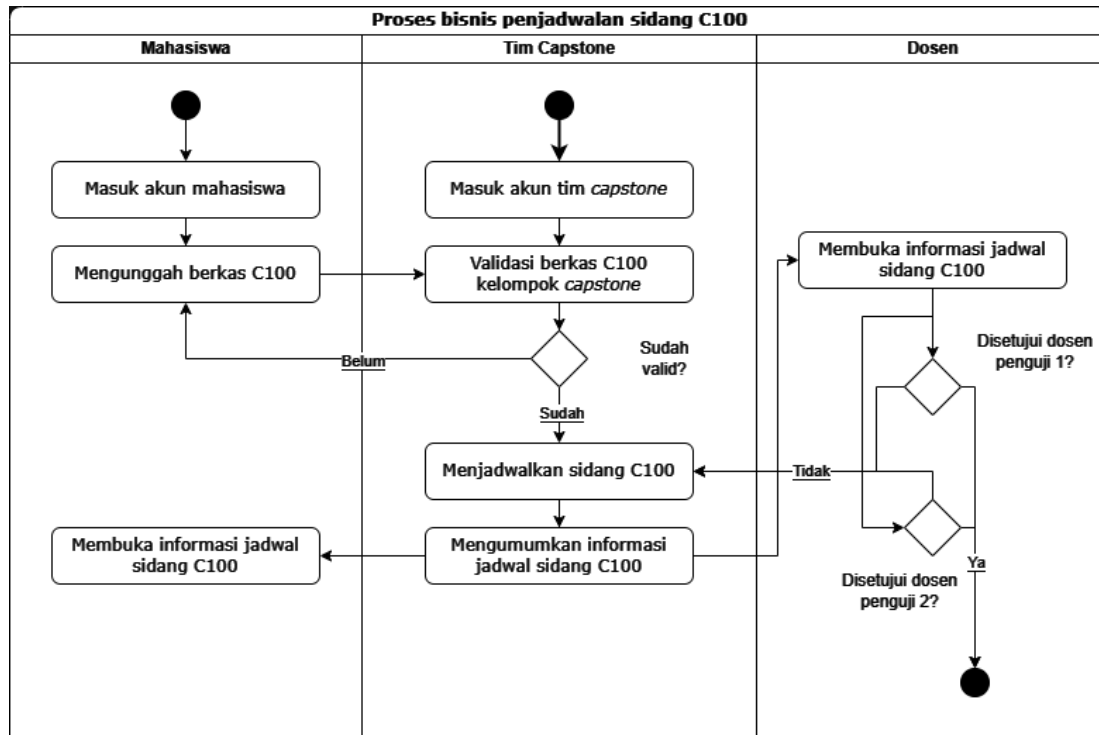
Setelah mahasiswa mendapatkan kelompok dan dosen pembimbing, maka proses bisnis selanjutnya adalah kelompok mahasiswa mulai mengerjakan dokumen *capstone* yang dimulai dengan dokumen C100. Setelah mahasiswa selesai mengerjakan dokumen C100, mahasiswa melakukan bimbingan dokumen dengan dosen

pembimbingnya. Proses bisnis bimbingan dan *tracking* progres kelompok *capstone* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram proses bimbingan dan *tracking* progres kelompok

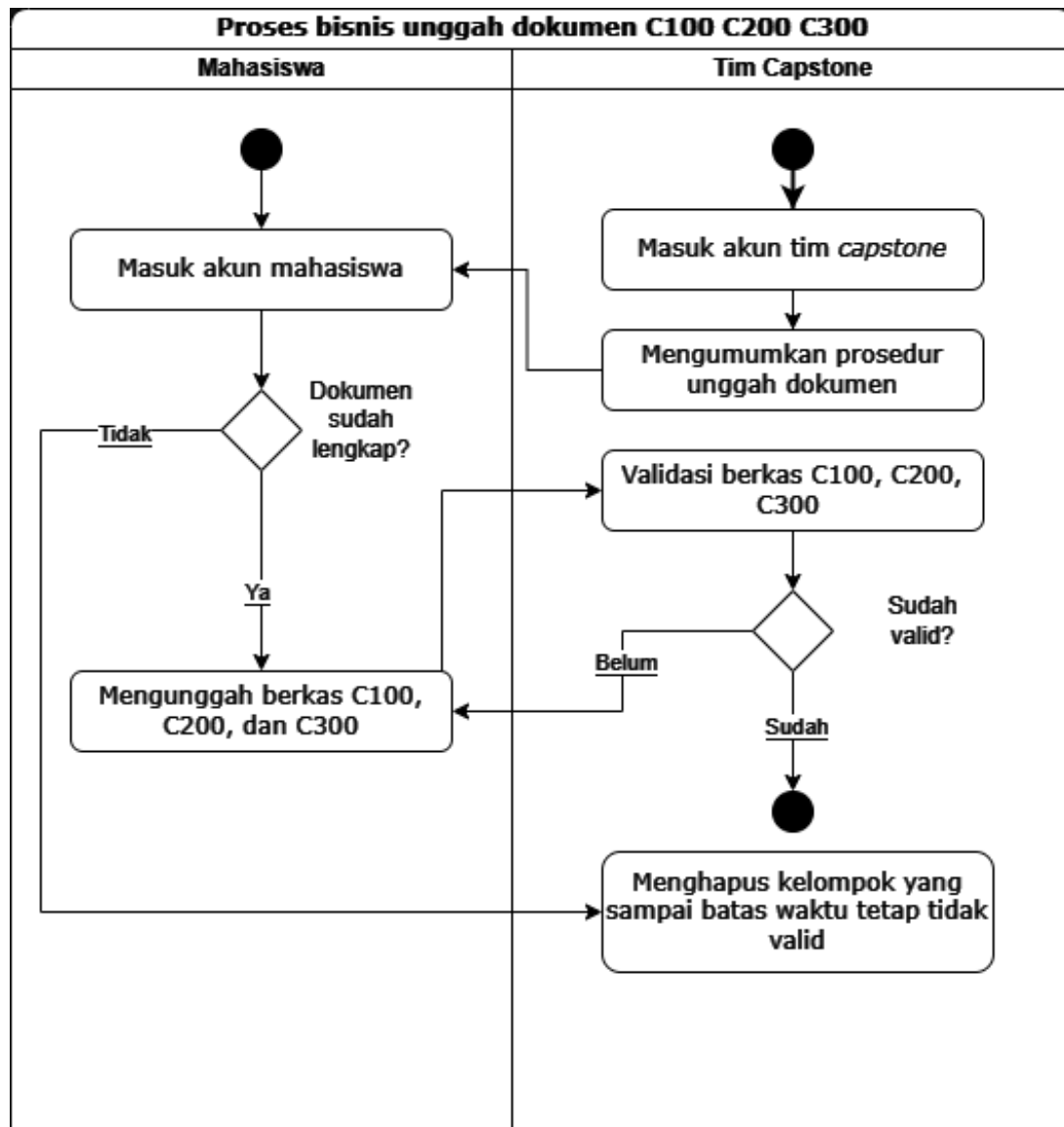
Mahasiswa dapat mengunggah dokumen yang sudah disetujui melalui sistem. Dokumen tersebut kemudian dapat diakses oleh tim *capstone* dan dosen pembimbing terkait. Kemudian, setelah dokumen C100 disetujui oleh dosen pembimbing 1 dan dosen pembimbing 2, maka kelompok mahasiswa dijadwalkan untuk sidang proposal. Proses bisnis penjadwalan sidang C100 dapat dilihat pada Gambar 3.4.



Gambar 3.4 Diagram proses penjadwalan sidang C100

Proses penjadwalan sidang C100 atau sidang proposal dilakukan oleh tim *capstone*, sebelum melakukan penjadwalan, tim *capstone* akan melakukan validasi terlebih dahulu untuk memastikan bahwa dokumen C100 dari setiap kelompok sudah disetujui oleh kedua dosen pembimbing kelompok terkait. Hal itu untuk meminimalisir terjadinya kelompok yang sudah dijadwalkan untuk sidang namun ternyata mangkir dari siklus *capstone* terkait. Selanjutnya, jika tim *capstone* sudah selesai melakukan penjadwalan sidang C100, jadwal tersebut akan diajukan ke masing-masing dosen penguji hingga kedua dosen penguji menyetujui jadwal sidang terkait. Kemudian, jika sudah memastikan semua dosen penguji menyetujui, tim *capstone* akan mengirimkan *broadcast* atau pengumuman jadwal sidang C100.

Setelah sidang C100 atau sidang proposal selesai, kelompok mahasiswa diminta untuk mengunggah dokumen C100 setelah revisi, dan mulai mengerjakan dokumen C200 dan C300. Setelah dokumen C200 dan C300 mendapatkan persetujuan, maka kelompok mahasiswa diminta untuk mengunggah dokumen tersebut. Proses bisnis unggah dokumen C100 C200 dan C300 dapat dilihat pada Gambar 3.5.

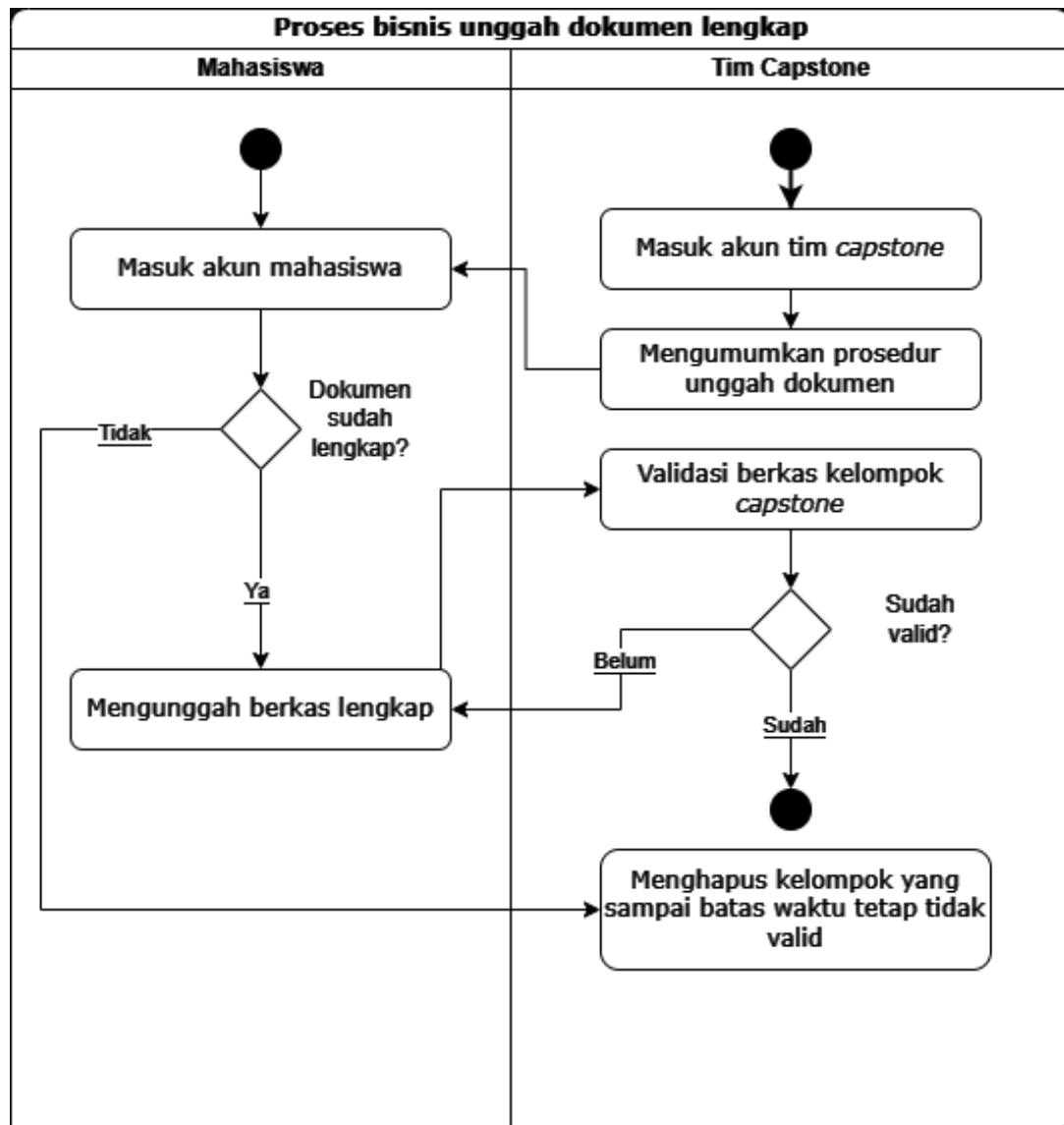


Gambar 3.5 Diagram proses unggah dokumen C100, C200, dan C300

Mahasiswa akan diminta untuk mengunggah dokumen C100, C200, dan C300 untuk mendapatkan nilai mata kuliah Proyek Desain Capstone. Setiap berkas yang diunggah akan divalidasi oleh tim *capstone*.

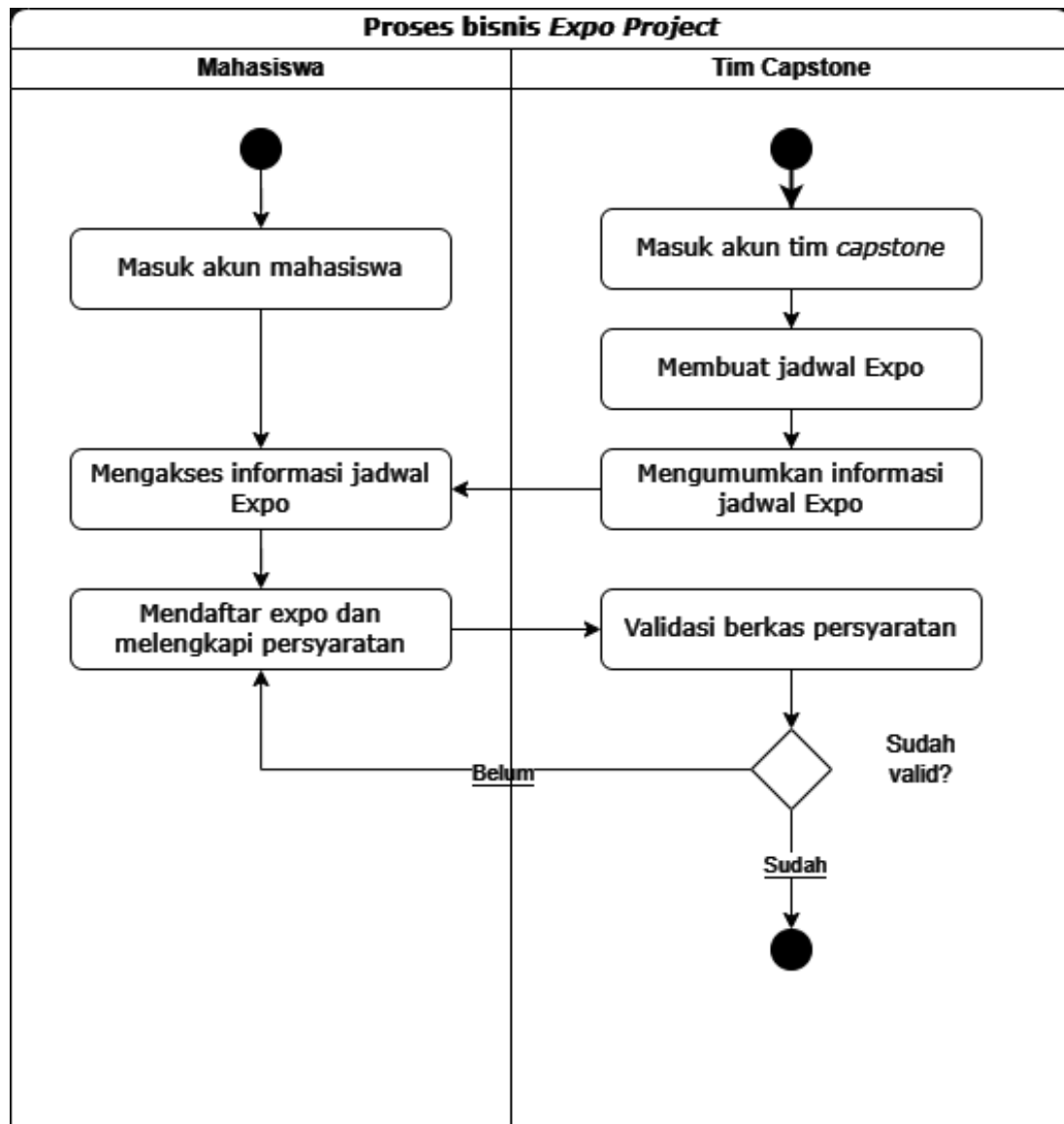
Lalu, proses bisnis selanjutnya adalah mahasiswa mengerjakan *project* yang diusulkan hingga selesai dan mendapatkan persetujuan dari *stakeholder* terkait dan dosen pembimbing. Setelah itu, akan masuk periode pendaftaran *expo project*, yang mana pada periode ini, mahasiswa diminta untuk mengunggah dokumen lengkap

sebagai syarat mengikuti *expo*. Proses bisnis unggah dokumen lengkap dapat dilihat pada Gambar 3.6.



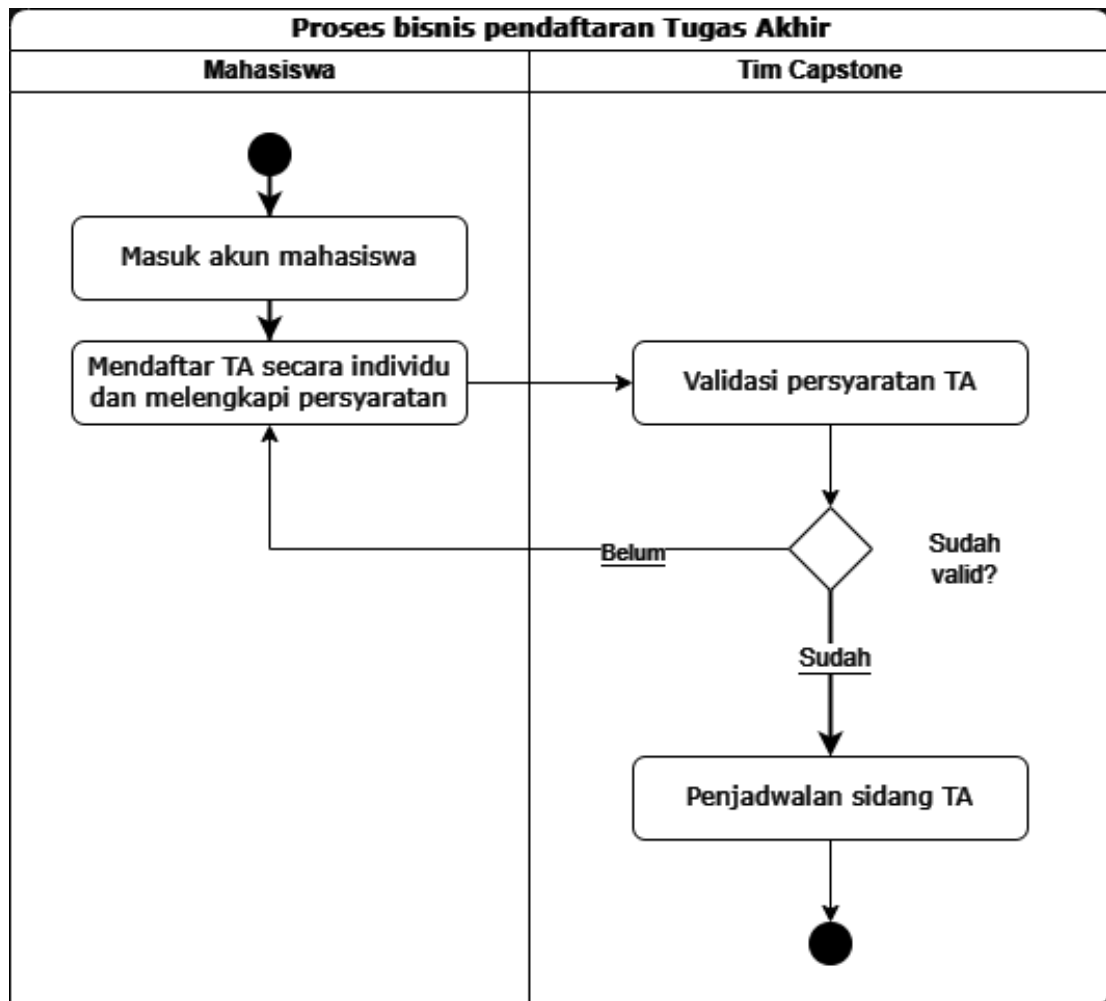
Gambar 3.6 Diagram proses unggah dokumen lengkap

Mahasiswa harus mengunggah dokumen lengkap sesuai batas waktu yang telah ditentukan oleh tim *capstone*, kemudian tim *capstone* akan melakukan validasi dokumen terkait. Proses bisnis *expo project* dapat dilihat pada Gambar 3.7.



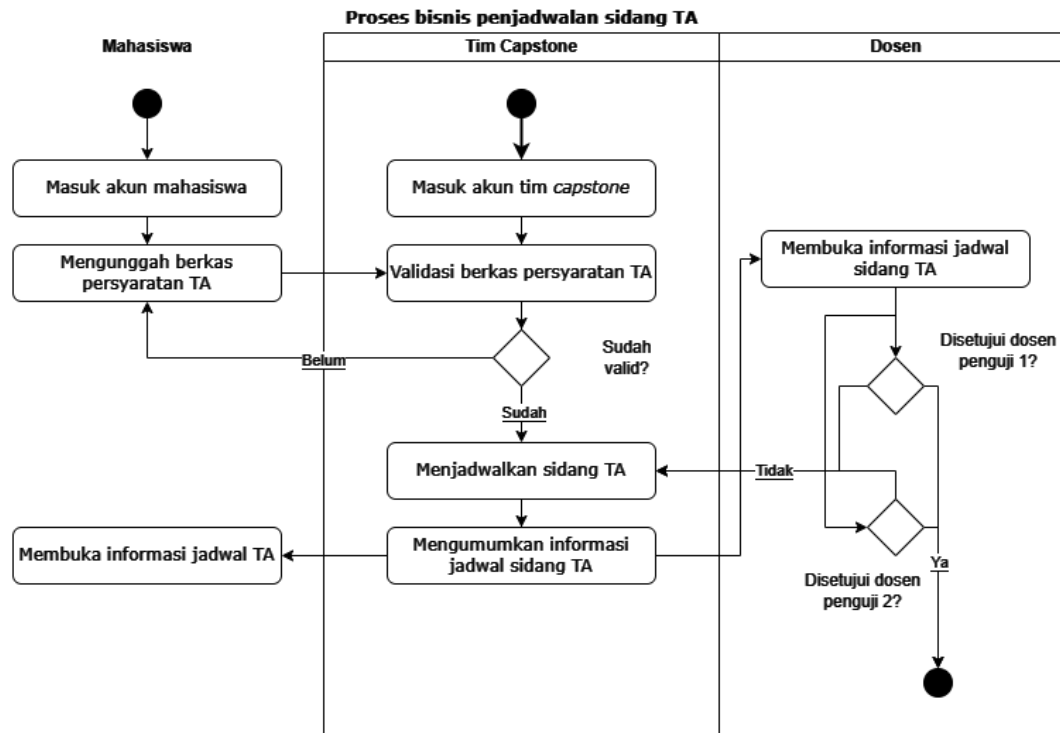
Gambar 3.7 Diagram proses *Expo Project*

Mahasiswa yang dapat mengikuti *expo project* adalah hanya kelompok mahasiswa yang berkasnya sudah valid dan lolos validasi oleh tim *capstone*. Kelompok kelompok yang sudah valid selanjutnya akan mengikuti *expo project* yaitu pameran dari hasil proyek yang telah dikembangkan. Setelah *expo project*, mahasiswa dapat mendaftar untuk mengikuti sidang Tugas Akhir secara individu. Proses bisnis pendaftaran sidang Tugas Akhir dapat dilihat pada Gambar 3.8.



Gambar 3.8 Diagram proses pendaftaran Tugas Akhir

Mahasiswa yang dapat mengikuti sidang Tugas Akhir adalah hanya mahasiswa yang berkasnya sudah valid dan lolos validasi oleh tim Tugas Akhir. Kelompok kelompok yang sudah valid selanjutnya akan mengikuti sidang Tugas Akhir sesuai jadwal. Proses bisnis penjadwalan sidang Tugas Akhir dapat dilihat pada Gambar 3.9



Gambar 3.9 Diagram proses penjadwalan sidang Tugas Akhir

3.3.3. Prosedur

Terdapat aturan, prosedur, atau *business rules* yang baru akibat dengan pengembangan sistem yang ditawarkan. Prosedur proses pengelolaan *Capstone* TA pada Departemen Teknik Komputer versi Android setelah sistem yang diusulkan dibuat dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tabel prosedur proses pengelolaan *Capstone* TA

Aktor	Aksi	Prosedur
Mahasiswa	Mendaftar <i>capstone</i> .	Mahasiswa mengisi <i>form</i> yang tersedia melalui sistem informasi berbasis web atau berbasis Android.
Mahasiswa	Melakukan bimbingan dokumen dengan dosen pembimbing.	Mahasiswa menghubungi dosen, kemudian mengunggah dokumen-dokumen keperluan pada sistem informasi berbasis Android. Hal tersebut mengurangi kemungkinan mahasiswa harus mencari tautan terkait <i>capstone</i> dan tugas akhir karena formulir bisa langsung diakses melalui halaman pada sistem.
Mahasiswa	Mendaftar <i>expo</i> .	Mahasiswa mengunggah dokumen dan mengisi <i>form</i> yang tersedia pada sistem informasi berbasis web dan Android. Hal tersebut mengurangi kemungkinan mahasiswa harus mencari tautan terkait <i>capstone</i> dan tugas akhir.

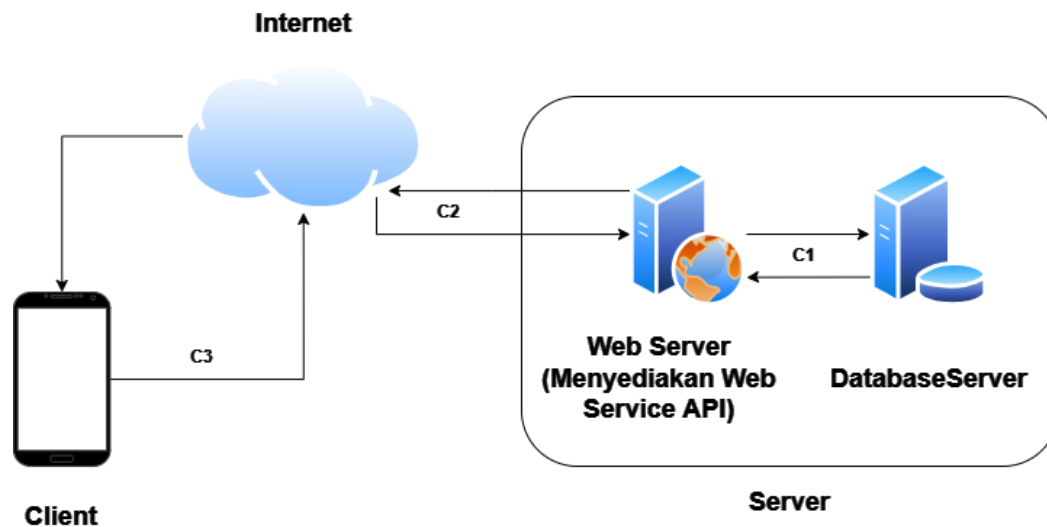
Tabel 3.2 Tabel prosedur proses pengelolaan *Capstone* TA (lanjutan)

Aktor	Aksi	Prosedur
Mahasiswa	Mendaftar tugas akhir.	Mahasiswa mengunggah dokumen dan mengisi <i>form</i> tersedia pada sistem informasi berbasis Android. Hal tersebut mengurangi kemungkinan mahasiswa harus mencari tautan terkait <i>capstone</i> dan tugas akhir.
Mahasiswa	Melihat kelompok, dosen pembimbing, dan jadwal sidang.	Mahasiswa mengakses informasi melalui sistem informasi berbasis Android. Hal tersebut mengurangi kemungkinan mahasiswa harus mencari tautan terkait <i>capstone</i> dan tugas akhir.
Mahasiswa	Mengelola dokumen	Mahasiswa mengelola dokumen melalui sistem informasi berbasis Android. Hal tersebut mengurangi kemungkinan mahasiswa harus mencari tautan terkait pengelolaan dokumen <i>capstone</i> dan Tugas Akhir.
Mahasiswa	Mengelola profil	Mahasiswa mengelola profil sistem informasi berbasis Android. Hal tersebut mengurangi kemungkinan mahasiswa harus mencari tautan terkait pengisian profil mahasiswa.

3.3 Perancangan Sistem

3.3.1. Arsitektur Sistem

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



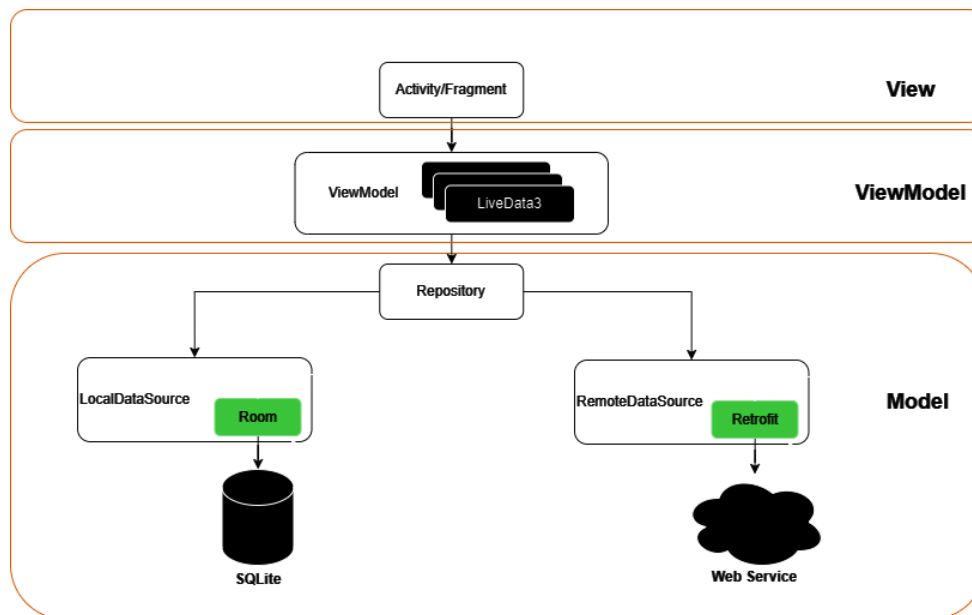
Gambar 3.10 Arsitektur sistem berbasis Android

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 *database* 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 3.11 Arsitektur pengembangan aplikasi Android

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

3.3.2. Antarmuka Eksternal

1. Hardware Interface

Perangkat keras yang digunakan sebagai antarmuka untuk interaksi manusia ke sistem adalah berupa antarmuka *touch screen* digunakan untuk melihat tampilan dan mengakses sistem informasi pada *platform* Android.

2. Software Interface

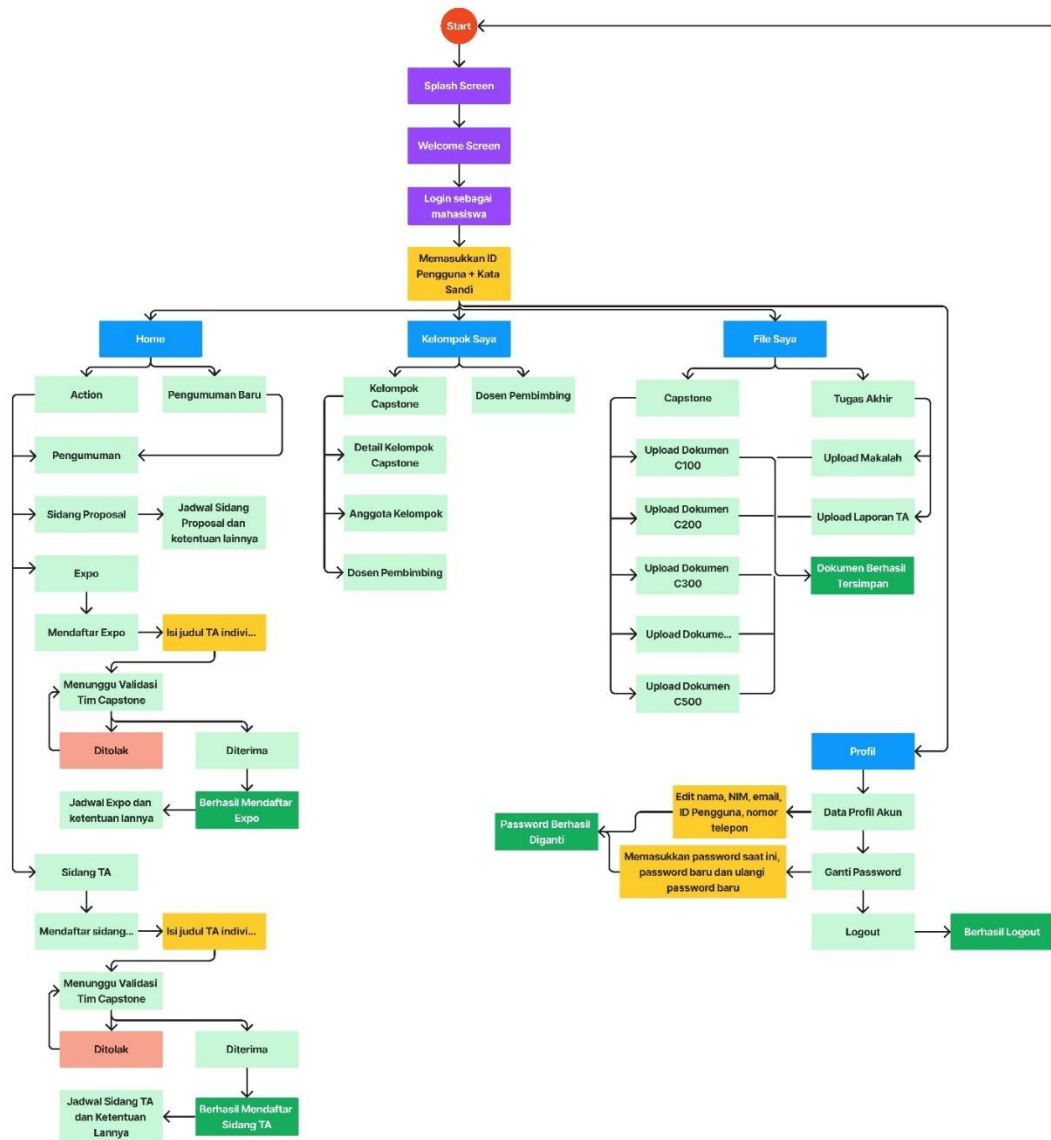
Sistem ini menggunakan antarmuka perangkat lunak berupa sistem operasi Android versi 5.0 (Lollipop) ke atas agar dapat atas digunakan untuk memasang dan menggunakan aplikasi pada Android.

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.

3.3.3. Perancangan Diagram Alur

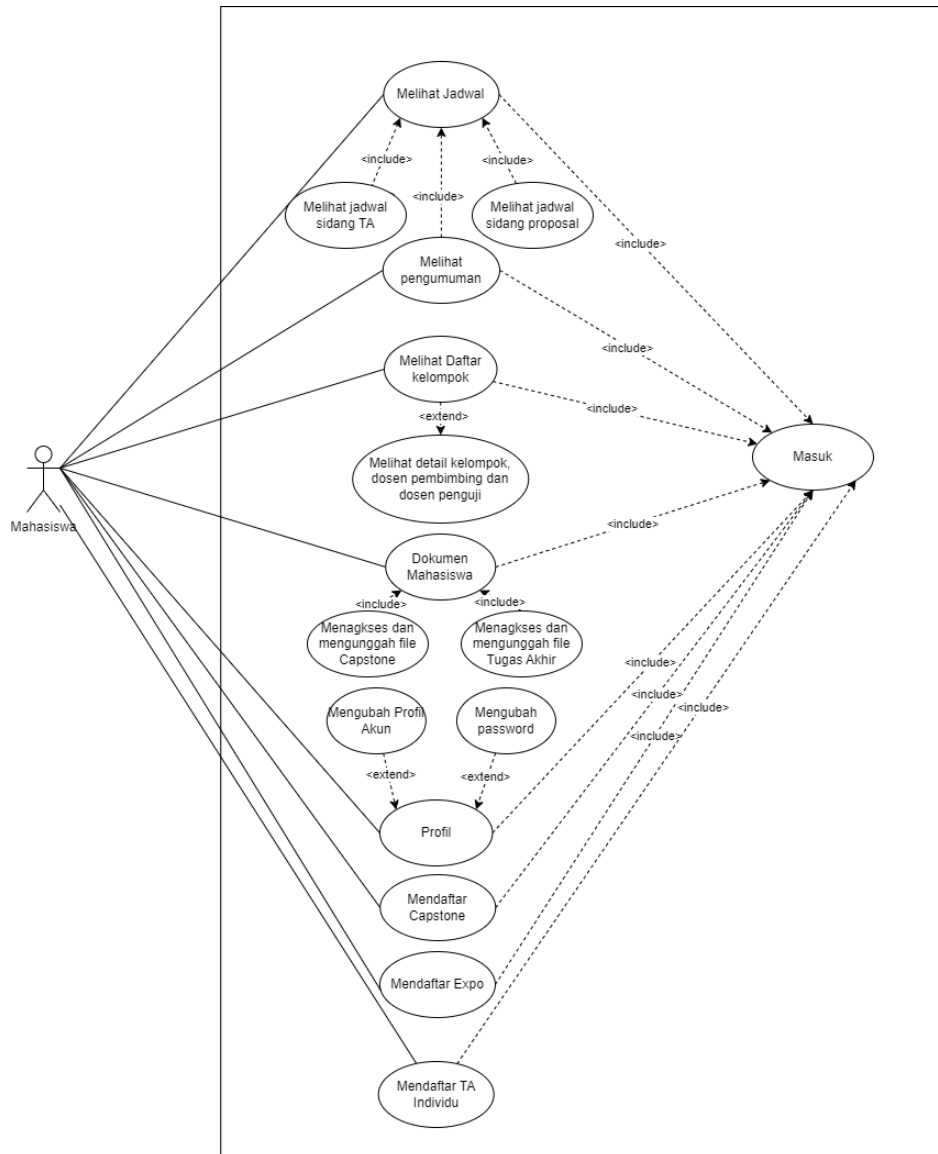
Sistem informasi pengelolaan *Capstone* TA berbasis Android memiliki diagram alur. Diagram alur adalah diagram yang berisi langkah atau proses untuk mengoperasikan sebuah program. Berikut ini merupakan diagram alur sistem informasi pengelolaan *Capstone* TA berbasis Android ditunjukkan pada Gambar 3.12.



Gambar 3.12 Diagram alur dari Sistem Informasi Pengelolaan *Capstone* TA berbasis Android

3.3.4. Perancangan Diagram *Use Case*

Sistem informasi pengelolaan *Capstone* TA versi Android pada Departemen Teknik Komputer Universitas Diponegoro memiliki kebutuhan fungsional yang akan dikembangkan. Kebutuhan tersebut digambarkan dalam sebuah diagram *use case* dengan 1 aktor yaitu mahasiswa. *Use case diagram* sistem dapat dilihat pada Gambar 3.13.

Gambar 3.13 Diagram *use case*

3.3.5. Perancangan Skenario *Use Case*

Diagram *use case* pada gambar 3.13 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

<i>Use Case ID Number</i>	1
<i>Use Case Name</i>	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 pengumuman.
	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 <i>capstone</i> .

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.	
Error Flow of Events		4. Sistem menampilkan halaman profil pengguna yang telah diubah.
	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 pengumuman.
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 memasukkan 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

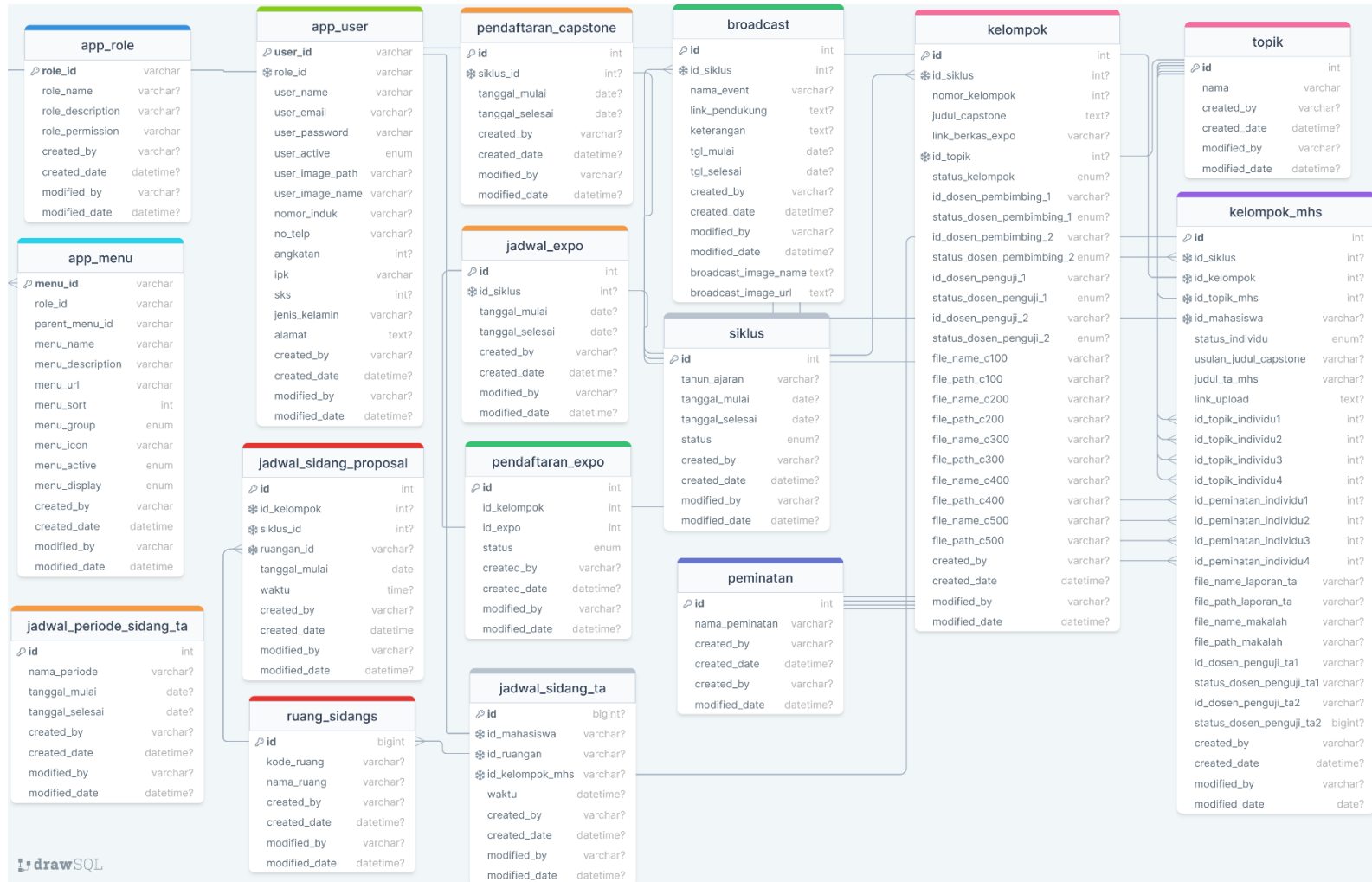
Use Case ID Number	8	
Use Case Name	Melihat dan Mendaftar Sidang Tugas Akhir	
Use Case Description	<i>Use Case</i> ini menggambarkan proses mahasiswa mengakses jadwal sidang Tugas Akhir dan Mendaftar sidang Tugas Akhir.	
Primary Actor	Mahasiswa	
Secondary Actor	Tim Capstone	
Pre-Condition	Mahasiswa telah 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.	

3.4.6. 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 3.14 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	-

3.4.7. 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 Authorization 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 Authorization 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 Authorization 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 <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>{ "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 <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- 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 <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>{ "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 <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>{ "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 Authorization 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>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>{ "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>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>{ "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>

3.4.8. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan akan fasilitas yang dibutuhkan serta aktivitas apa saja yang dilakukan oleh sistem secara umum [16]. Kebutuhan fungsional sistem dapat dilihat pada Tabel 3.29.

Tabel 3.29 Tabel kebutuhan fungsional sistem

No	Kategori Pengguna	Kode	Deskripsi Kebutuhan	Prioritas
1	Mahasiswa	SW-SIPCTA-F01	Masuk ke dalam aplikasi.	Tinggi
2	Mahasiswa	SW-SIPCTA-F02	Mengakses pengumuman.	Tinggi
3	Mahasiswa	SW-SIPCTA-F03	Melakukan pendaftaran <i>capstone</i>	Tinggi
4	Mahasiswa	SW-SIPCTA-F04	Mengakses dan mengunggah dokumen	Tinggi
5	Mahasiswa	SW-SIPCTA-F05	Melihat informasi pembagian kelompok dan dosen pembimbing	Tinggi
6	Mahasiswa	SW-SIPCTA-F06	Melihat informasi jadwal sidang proposal	Tinggi
7	Mahasiswa	SW-SIPCTA-F07	Melakukan pendaftaran expo	Tinggi
8	Mahasiswa	SW-SIPCTA-F08	Mengakses dan mendaftar sidang Tugas Akhir.	Tinggi
9	Mahasiswa	SW-SIPCTA-F09	Mengakses dan mengubah profil mahasiswa.	Tinggi
10	Mahasiswa	SW-SIPCTA-F10	Keluar dari akun mahasiswa.	Tinggi

3.4.9. Kebutuhan Non-Fungsional

Selain kebutuhan fungsional, sistem juga memiliki kebutuhan non-fungsional sistem yang dapat dilihat pada Tabel 3.30.

Tabel 3.30 Tabel kebutuhan non-fungsional sistem

SRS-Id	Parameter	Requirement
SW-SIPCTA-NF01	<i>Availability</i>	Aplikasi ini dapat beroperasi 7 hari dalam seminggu dan 24 jam dalam satu hari.
SW-SIPCTA-NF02	<i>Reliability</i>	Sistem akan menjamin minimalisasi tingkat kegagalan dalam pengoperasian.
SW-SIPCTA-NF03	<i>Portability</i>	Sistem dapat dioperasikan pada perangkat Android dengan bermacam versi OS.
SW-SIPCTA-NF04	<i>Response time</i>	Memberikan waktu respon maksimal kira-kira 5 detik.
SW-SIPCTA-NF05	<i>Safety</i>	Semua data pada sistem dijamin aman.
SW-SIPCTA-NF06	<i>Security</i>	Aplikasi menyimpan informasi yang sudah divalidasi.

Penjelasan kode:

SW = *Software*

SIPCTA = Sistem Informasi Pengelolaan *Capstone* TA

F = *Functional*

NF = *Non-Functional*

3.4 Metode Pengujian

Pengujian perangkat lunak yang dikembangkan dalam proyek ini akan dievaluasi dengan menggunakan metode *black box* dan pengujian *usability*. Pengujian ini dijadwalkan untuk dilakukan pada tahap akhir pembuatan perangkat lunak dengan tujuan untuk menilai sejauh mana perangkat lunak tersebut dapat berfungsi. Metode *black box* digunakan untuk menguji fungsionalitas perangkat lunak tanpa memerhatikan implementasi internalnya. Selain itu, dalam pengujian *usability* ini, sampel peran pengguna yang diambil adalah mahasiswa. Dengan mengambil sampel *role* pengguna mahasiswa, diharapkan dapat mendapatkan wawasan yang lebih baik tentang pengalaman pengguna yang sebenarnya dan menilai sejauh mana perangkat lunak ini dapat memenuhi kebutuhan.

BAB IV
HASIL DAN PEMBAHASAN

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil

5.2. Saran

Dengan adanya.

DAFTAR PUSTAKA

- [1] Panduan Pelaksanaan Capstone Tugas Akhir 2023. Departemen Teknik Komputer, Universitas Diponegoro. 2023.
- [2] P. P. Arhandi, "Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan dengan Menggunakan Metode Back End dan Front End," *Jurnal Teknologi Informasi*, vol. 7, no. 1, pp. 39-48, 2016. doi: <https://doi.org/10.36382/jti-tki.v7i1.192>
- [3] V. Makarenko, O. Olshevska, dan Y. Kornienko, "an Architectural approach for Quality Improving of Android Applications Development Which Implemented To Communication Application foe Mechatronics Robot Laboratory Onaft," *Jurnal Teknologi*, vol. 9, no. 3, pp. 8–13, 2017, doi: <https://doi.org/10.15673/atbp.v9i3.714>
- [4] P. Ni Luh Putri Ari Wedayanti, "Evaluasi Aspek Usability pada Aplikasi Simalu Menggunakan Metode Usability Testing," *Jurnal Merpati*, vol. 7, pp. 113-124, 2019. doi: <https://doi.org/10.24843/JIM.2019.v07.i02.p03>
- [5] M.A. Dharmawan, R. Indriati, Sucipto, "Implementasi Sistem Informasi Tugas Akhir Menggunakan Metode Classic Life Cycle" Vol. 3 No. 1 (2019): Prosiding Semnas Inotek Ke-III Tahun 2019
- [6] S.M Sari, L.M Tua, E. Krisnanik, "Sistem Informasi Monitoring Pembimbingan Skripsi/Tugas Akhir (SIMP-S/TA) Berbasis Android" Vol. 2 No. 1 (2019): Seminar Nasional Informatika, Sistem Informasi dan Keamanan Siber (SEINASI-KESI) 2019
- [7] A. Lander, D.Q. Utama, Putrada, "Desain dan Implementasi REST-API Pada Website Marketplace Katering “Ketringan” Menggunakan Framework Laravel," *Jurnal e-Proceeding of Engineering* , vol.8, no.5, p. 10693, 2021.
- [8] A.B. Nurwicaksono, Ikmah "Perancangan Sistem Manajemen Kasir Berbasis Android pada Blessing of Shoes," *Journal of Information System Management*

- (*JOISM*), Vol. 5, No. 2 pp. 108-114. 2024 doi: <https://doi.org/10.24076/joism.2024v5i2.1111>
- [9] D. Murdiani and H. Hermawan, "Perbandingan Metode Waterfall dan RAD (Rapid Application Development) pada Pengembangan Sistem Informasi," *Jurnal Teknologi Informasi*, vol. 6, no. 1, pp. 14–23. 2022. doi: <https://doi.org/10.51401/jinteks.v4i4.2008>
- [10] M.F. Ridhoi, "Rancang Bangun Mobile Learning Berbasis Android di SMKN Poncol" *Jurnal Seminar Nasional Teknologi Informasi dan Komunikasi*, vol, 5, no. 1, pp. 85–94, 2022.
- [11] I.M. Riyadhi, "Penerapan Pola Arsitektur MVVM Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android," *Jurnal Infotech*, vol. 9 no. 1, pp. 146-158, 2023. doi: <https://doi.org/10.31949/infotech.v9i1.5246>
- [12] S.D. Purnamasari, F. Syakti " Implementasi Usability Testing dalam Evaluasi Website Sekolah," *Jurnal SISFOKOM (Sistem Informasi dan Komputer)*, Vol 09, No 03, pp 420 – 426. 2020
- [13] H. A. B. H. Hasanuddin Hasanuddin, "Rancang Bangun REST API Aplikasi Weshare sebagai Upaya Mempermudah Pelayanan Donasi Kemanusiaan," *Jurnal Informatika Teknologi dan Sains*, vol. 4, p. 8, 2022. doi: <https://doi.org/10.51401/jinteks.v4i1.1474>
- [14] F. Maulana, "Aplikasi Manajemen Laboratorium Menggunakan Metode MVVM Berbasis Android," *Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 3, p. 88, 2022. doi: <https://doi.org/10.30630/jitsi.3.3.94>
- [15] Arribe, Edo, "Aplikasi E-marketplace Menggunakan Arsitektur MVVM (Model-view-viewmodel) Berbasis Android." *Jurnal Fasikom*, vol. 11, no. 2, 2021, pp. 75-78, doi: <https://doi.org/10.37859/jf.v11i2.2762>
- [16] Subianto, "Penerapan Metode Rapid Application Development dalam Perancangan Sistem Informasi Pendataan," *Jurnal Infokam*, Vol. XVI, No. 1, pp.47-49, 2020, doi: <https://doi.org/10.53845/infokam.v16i1.218>

- [17] D.P. Sari, R. Wijanarko "Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang)" *Jurnal Informatika Dan RPL*, Vol. 2, No. 1, pp. 32-36. 2019.
- [18] W.G. Wardhana, I. Arwani, B. Rahayudi "Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)" *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 4, No. 2, pp. 680-689. 2020.
- [19] S. Sibuea, M.I. Saputro, A. Annan, Y.B. Widodo "Aplikasi Mobile Collection Berbasis Android pada PT. Suzuki Finance Indonesia" *Jurnal Informatika Dan Teknologi Komputer*, Vol 2 No. 1. pp 31-42. 2022.
- [20] F. N. K. Umi Salamah, "Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing," *Information Management For Educators And Professionals*, vol. 2, pp. 35-46, 2017.
- [21] H. H. B. P. Aisyah Sriwulandari, "Analisis dan Evaluasi Aspek Usability Pada Web HRMIS Telkom University Menggunakan Usability Testing," *eProceedings of Engineering*, vol. 1, pp. 537-542, 2014.
- [22] K. 'Afifah, Z. F. Azzahra, dan A. D. Anggoro, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review", *Jurnal Intech*, vol. 3, no. 1, pp. 9-11, 2022, doi: <http://dx.doi.org/10.54895/intech.v3i2.1682>
- [23] L. Setiyani, dan E. Tjandra, "Analisis Kebutuhan Fungsional Aplikasi Penanganan Keluhan Mahasiswa Studi Kasus: STMIK Rosma Karawang," *Jurnal Inovasi Pendidikan dan Teknologi Informasi*, vol. 2, no. 1, p. 11, 2021, DOI: <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