




Topik Capstone	<b>Smart System</b>	
Siklus / Tahun	<b>2 / 2023</b>	
Judul Dokumen	<b>Pengembangan Sistem Informasi Pengelolaan Capstone dan Tugas Akhir di Departemen Teknik Komputer Universitas Diponegoro</b>	
Jenis Dokumen	<b>DESAIN PRODUK</b> Catatan: Penggunaan dan penyebaran dokumen ini dikendalikan oleh Departemen Teknik Komputer Universitas Diponegoro	
Nomor Dokumen	<b>C300.2TA2023.2.022322</b>	
Nomor Revisi	<b>2</b>	
Nama File	<b>C300_022322.doc</b>	
Tanggal Penerbitan	<b>5 Desember 2023</b>	
Unit Penerbit	<b>Departemen Teknik Komputer Universitas Diponegoro</b>	
Jumlah Halaman	<b>44</b>	Tidak termasuk sampul

Data Pengusul			
Pengusul	Nama NIM	<b>Nanda Iqbal Hanafi</b> <b>21120120130109</b>	Jabatan: API & Backend Mobile
	Tanggal	<b>5 Desember 2023</b>	
	Nama NIM	<b>Muhammad Rofi'ul Anam</b> <b>21120120140135</b>	Jabatan: Web Developer
	Tanggal	<b>5 Desember 2023</b>	
	Nama NIM	<b>Maulana Yusuf Suradin</b> <b>21120120140051</b>	Jabatan UI UX & Frontend Android
	Tanggal	<b>5 Desember 2023</b>	
Pembimbing 1 (Utama)	Nama	<b>Adnan Fauzi, S.T., M.Kom.</b> <b>NIP. 198101272018071001</b>	Tanda Tangan
Pembimbing 2	Nama	<b>Patricia Evericho Mountaines,</b> <b>S.T., M.Cs.</b> <b>NPPU. H.7.199203222022042001</b>	Tanda Tangan
	Tanggal		

## Daftar Isi

<b>1. Pendahuluan.....</b>	<b>4</b>
<b>1.1. Ringkasan isi dokumen.....</b>	<b>4</b>
<b>1.2. Aplikasi Dokumen.....</b>	<b>4</b>
<b>1.3. Referensi .....</b>	<b>5</b>
<b>1.4. Daftar Singkatan.....</b>	<b>5</b>
<b>2. Pemilihan Desain Produk.....</b>	<b>6</b>
<b>2.1. Alternatif Solusi .....</b>	<b>6</b>
2.1.1. Alternatif Solusi untuk Sistem Berbasis Web.....	6
2.1.2. Alternatif Solusi untuk Sistem Berbasis Android .....	8
<b>2.2. Proses Pemilihan Solusi.....</b>	<b>9</b>
<b>3. Desain Produk yang Diusulkan .....</b>	<b>11</b>
<b>3.1. Arsitektur Sistem .....</b>	<b>11</b>
<b>3.2. Desain Detail Sistem .....</b>	<b>14</b>
3.2.1. Data Description .....	14
3.2.2. Diagram Alur .....	23
3.2.3. Standar - standar yang Dipergunakan .....	29
3.2.4. Method/API.....	29
<b>3.3. Traceable .....</b>	<b>30</b>
3.3.1. Data.....	30
<b>4. Verifikasi Desain Produk .....</b>	<b>35</b>
<b>4.1. Hasil Simulasi Awal Produk .....</b>	<b>35</b>
<b>5. Rencana Implementasi dan Pengujian.....</b>	<b>42</b>
<b>5.1. Gaant Chart.....</b>	<b>42</b>
<b>5.2. Metode Pengujian .....</b>	<b>42</b>

### Daftar Revisi

[versi] Tanggal	Perbaikan
[1] 5 Desember 2023	<ul style="list-style-type: none"> <li>- Daftar pustaka: harus dari jurnal, penulisan dikasih keterangan Vol. dan No. dan Halaman.</li> <li>- Pengantar dan penjelasan setiap gambar diperbaiki lagi</li> <li>- Gambar diperjelas</li> <li>- ERD → app_login dan app_login_attempt dihapus saja</li> <li>- Normalisasi database</li> <li>→ app_role_user → relasi langsung tabel role dan user saja</li> <li>→ app_role_menu → relasi langsung tabel role dan menu saja</li> <li>→ user_id dan kolom lain tolong diperhatikan estimasi panjangnya</li> <li>- Perbaiki jika posisi gambar memang harus miring (yang atas yang bagian kiri)</li> </ul>
[2] 15 Desember 2023	<ul style="list-style-type: none"> <li>- Daftar isi <i>font</i>-nya di samakan (ukurannya)</li> <li>- Menambahkan daftar singkatan untuk metode transmisi data (halaman 12 dan 13)</li> <li>- Pengantar gambar pada halaman 23 direvisi Tidak boleh memakai “pada Gambar 3.4 sampai dengan Gambar 3.8” → dibuat pengantar pergambar</li> <li>- Diagram alur superadmin disesuaikan dengan use case diagram (superadmin bukan merupakan gabungan)</li> </ul>

## **1. Pendahuluan**

### **1.1. Ringkasan isi dokumen**

Pada dokumen ini berisikan uraian spesifikasi proposal Capstone yang akan melatarbelakangi proyek “Pengembangan Sistem Informasi Pengelolaan Capstone dan Tugas Akhir di Departemen Teknik Komputer Universitas Diponegoro”. Sistem Informasi ini direncanakan dikembangkan lebih lanjut untuk menambahkan fitur dan perbaikan *bug* serta melakukan pengembangan versi Android sehingga dapat meningkatkan efektivitas dalam mendukung kegiatan akademik dan administratif mahasiswa.

Dokumen ini memuat penjelasan mengenai desain produk dari Sistem Informasi Pengelolaan Capstone dan Tugas Akhir di Departemen Teknik Komputer Universitas Diponegoro. Dokumen spesifikasi sistem ini terdiri dari 4 Bab yaitu, Bab 1 berisi ringkasan isi dokumen, aplikasi dokumen, referensi, dan daftar singkatan. Bab 2 berisi pemaparan alternatif solusi. Bab 3 memuat desain produk yang diusulkan antara lain arsitektur sistem, desain detail produk dan *traceable*. Bab 4 berisi simulasi produk. Bab 5 berisi *gant chart* dan pengujian sistem yang digunakan.

Dokumen ini digunakan sebagai acuan dalam proses pengembangan serta sebagai bahan evaluasi pada saat proses pengembangan perangkat lunak. Selain itu, dokumen ini juga dapat digunakan oleh pengembangan perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

### **1.2. Aplikasi Dokumen**

Dokumen ini berfungsi untuk menjelaskan:

- 1) Proses pemilihan desain alat dari beberapa alternatif yang ada.
- 2) Detail desain alat dari level tertinggi sampai terendah.
- 3) Menjelaskan standar-standar yang dipergunakan.
- 4) Refensi komponen/*library* yang digunakan.
- 5) Verifikasi bahwa hasil rancangan dapat diaplikasikan.
- 6) Rencana implementasi dan pengujian.

### 1.3. Referensi

- [1] A Jartarghar, H. Rao Salanke, G, A.R, A. K, G.S, S, & Dalali, S. (2022). React Apps with Server-Side Rendering: Next.js. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, Vol 14(4), halaman 25–29.
- [2] Nasution, Lizda Iswari. (2021). Penerapan React JS Pada Pengembangan FrontEnd Aplikasi Startup Ubaform. *Jurnal Start Up*, Vol 2(2), halaman 2-3.
- [3] Devi Purnama Sari, & Rony Wijanarko (2019). Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang). *Jurnal Informatika dan Rekayasa Perangkat Lunak*, Vol 2(1), halaman 33.
- [4] Panji Rachmat Setiawan, Rizdqi Akbar Ramadhan, Ause Labellapansa (2022). Pelatihan Pemrograman Flutter. *Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan*, Vol 3(1), halaman 23-25.
- [5] Feri Setyo Efendi, Lutfi Fanani, & Ahmad Afif Supianto. (2020). Rancang Bangun Aplikasi Pendukung untuk Observasi Kelas berbasis Mobile. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol 4(6), halaman 18-30.
- [6] I. A. Shaleh, J. P. Yogi, P. Pirdaus, R. Syawal, dan A. Saifudin. (2021). Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, Vol 4(1), halaman 38. doi: 10.32493/jtsi.v4i1.8960.
- [7] Munadia and A. G. Taufiq. (2019). Room Charting berdasarkan Pemetaan Gantt Chart pada Penjadwalan Kuliah Pengganti/Tambahan di Fakultas Teknik Universitas Syiah Kuala. *Jurnal Online Teknik Elektro*, Vol 4(4), halaman 25–32.

### 1.4. Daftar Singkatan

Tabel 1. 1 Daftar singkatan

Akronim	Terminologi
API	<i>Application Programming Interface</i>
AVD	<i>Android Virtual Device</i>
CSS	<i>Cascading Style Sheet</i>
CTA	<i>Call To Action</i>

HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JS	<i>JavaScript</i>
MVC	<i>Model View Controller</i>
PHP	<i>Personal Home Page: Hypertext Preprocessor</i>
PDO	<i>PHP Data Objects</i>
SI	Sistem Informasi
SQL	<i>Structured Query Language</i>
TA	Tugas Akhir
UI	<i>User Interface</i>
UX	<i>User Experience</i>
XML	<i>Extensible Markup Language</i>

## 2. Pemilihan Desain Produk

### 2.1. Alternatif Solusi

Dalam pengembangan Sistem Informasi Pengelolaan Capstone dan Tugas Akhir di Departemen Teknik Komputer Universitas Diponegoro terdapat 2 jenis alternatif solusi yaitu sebagai berikut:

#### 2.1.1. Alternatif Solusi untuk Sistem Berbasis Web

##### a. NextJs

Next.js adalah *framework* React yang kuat untuk membangun aplikasi web. Dibangun di atas React, Next.js menawarkan fitur-fitur seperti *server-side rendering* (SSR) dan *static site generation* (SSG) yang dapat meningkatkan kecepatan pemuatan halaman dan kinerja keseluruhan. Kelebihannya meliputi pendekatan modular untuk pengelolaan komponen, serta kemudahan dalam melakukan *prerendering* halaman untuk mengoptimalkan pengalaman pengguna.

Next.js dibuat karena masalah *pre-render* secara statis di beberapa halaman yang membuat SEO (*Search Engine Optimization*) terganggu, dikarenakan *file* JavaScript harus di-load dan menentukan komponen apa yang harus ditampilkan. Next.js mempunyai beberapa keunggulan seperti *automatic code splitting* yang digunakan sebagai pemecah kode agar lebih cepat saat me-load halaman. Namun, Next.js lebih fokus pada pengembangan *frontend*, sehingga memerlukan integrasi dengan *backend* terpisah. [1]

b. ReactJs

ReactJS adalah *library* JavaScript yang kuat untuk membangun antarmuka pengguna yang responsif dan dinamis. Kelebihannya termasuk pendekatan komponen yang modular untuk pengelolaan kode, serta kemampuan untuk membangun antarmuka pengguna yang menarik dan interaktif dengan dukungan untuk *single-page application* (SPA). Komunitas yang besar dan aktif juga menyediakan banyak *library* dan alat tambahan untuk membantu pengembangan. Namun, ReactJS hanya merupakan *library* UI, sehingga memerlukan kombinasi dengan *library* atau *framework* lain untuk mengelola *state*, *routing*, dan *backend*.

React memungkinkan untuk membuat *user interface* yang kompleks dengan set kode kecil yang terisolasi yang disebut "komponen". React JS ini digunakan untuk menangani lapisan tampilan dalam aplikasi satu halaman dan pengembangan aplikasi. React berusaha untuk memberikan kecepatan, kesederhanaan, dan skalabilitas. Beberapa fitur yang paling mencolok adalah JSX, Komponen Stateful, Model Objek Dokumen Virtual. [2]

c. Laravel

Laravel adalah *framework* PHP yang kuat dan populer untuk pengembangan web dengan fitur-fitur lengkap. Kelebihannya meliputi dukungan ORM (*Object-Relational Mapping*) untuk interaksi dengan *database* dan pengolahan data, serta arsitektur yang jelas dengan konvensi pengkodean yang baik untuk memudahkan kerja dalam tim. Banyak fitur bawaan seperti autentikasi, *caching*, validasi, dan lainnya, juga menghemat waktu pengembangan. Laravel sangat cocok untuk membangun aplikasi dengan kompleksitas menengah hingga tinggi. Namun, dalam beberapa kasus, performa Laravel bisa menjadi lebih lambat dibandingkan dengan *framework* PHP yang lebih ringan, dan proyek yang menggunakan Laravel memerlukan pemahaman tentang bahasa PHP dan konsep-konsep *web development*. 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*. [3]

### 2.1.2. Alternatif Solusi untuk Sistem Berbasis Android

#### a. Flutter

Flutter adalah sebuah *framework open-source* yang dikembangkan oleh Google untuk membangun antarmuka (*user interface*) aplikasi Android dan iOS [4]. Kelebihannya termasuk kemampuan untuk mengembangkan aplikasi Android dan iOS dengan satu kode, sehingga menghemat waktu dan upaya dalam pengembangan. Flutter juga menawarkan antarmuka pengguna yang menarik dan responsif, dengan dukungan untuk *widget* kustom dan animasi. Namun, karena Flutter menggunakan *rendering* sendiri, dapat menyebabkan beberapa masalah dengan integrasi perangkat Android dan keterbatasan dalam akses ke fitur perangkat yang spesifik.

#### b. Android Native Java

Dalam konteks pengembangan aplikasi Android, Java digunakan sebagai bahasa resmi untuk membuat aplikasi yang dapat dijalankan pada *platform* Android. Kelebihan penggunaan Java untuk pengembangan Android termasuk dukungan penuh dari Google, pemilik Android, dan komunitas Android yang besar, yang membuat tersedia banyak referensi dan dokumentasi untuk pengembang. Dengan Java, pengembang memiliki kontrol penuh terhadap fitur perangkat Android dan dapat mengakses berbagai API Android untuk membangun aplikasi yang kuat.

Namun, seiring berjalannya waktu, bahasa pemrograman Kotlin telah muncul sebagai alternatif yang lebih modern dan efisien dalam pengembangan aplikasi Android. Kotlin menawarkan sintaksis yang lebih bersih, pengkodean yang lebih singkat, dan lebih banyak fitur yang mendukung pengembangan yang produktif. Oleh karena itu, Java, dalam konteks pengembangan Android, dapat dianggap sebagai "usang" atau kurang relevan dalam beberapa kasus. Pengembangan dengan Java mungkin memakan lebih banyak waktu dan usaha dibandingkan dengan Kotlin atau bahasa pemrograman modern lainnya.

#### c. Android Native Kotlin

Android Native Kotlin x XML Views adalah pendekatan konvensional untuk mengembangkan aplikasi Android menggunakan bahasa Kotlin dan XML untuk membangun antarmuka pengguna. Kelebihannya termasuk adopsi dari pendekatan yang sudah mapan dengan dukungan penuh dari Android SDK. Pengembangan dengan Kotlin x XML Views dapat menggabungkan kekuatan bahasa Kotlin dengan tata letak yang didefinisikan secara deklaratif melalui XML. Namun, seperti pada pendekatan Android Native Java, pengembangan dengan



Kotlin x XML Views juga dapat menjadi lebih lambat dan memerlukan lebih banyak kode *boilerplate*. [5]

## 2.2. Proses Pemilihan Solusi

### a. Proses Pemilihan Solusi berbasis Web

Pada tabel 2. 1 akan dibandingkan kelebihan dan kekurangan dari masing-masing alternatif solusi untuk sistem berbasis web.

Tabel 2. 1 Pemilihan solusi berbasis web

Solusi	Kelebihan	Kekurangan
<b>NextJS</b>	<ul style="list-style-type: none"><li>a. <i>Server-Side Rendering</i> (SSR) dan <i>Static Site Generation</i> (SSG) untuk SEO yang baik.</li><li>b. Modularitas dan pemecahan otomatis kode (<i>automatic code splitting</i>)</li><li>c. Kemudahan <i>routing</i> dengan direktori 'pages'.</li><li>d. Dukungan untuk pengembangan aplikasi <i>real-time</i> dengan integrasi WebSocket.</li></ul>	<ul style="list-style-type: none"><li>a. Fokus pada <i>frontend</i>, memerlukan integrasi dengan <i>backend</i> terpisah</li><li>b. Perlu pemahaman mendalam tentang React.js.</li><li>c. SSG memerlukan perhatian khusus untuk data yang sering berubah.</li></ul>
<b>ReactJS</b>	<ul style="list-style-type: none"><li>a. Modularitas dan kemampuan membangun antarmuka pengguna yang responsif.</li><li>b. Komunitas yang besar dan aktif.</li><li>c. Dapat digunakan untuk mengembangkan aplikasi <i>mobile</i> dengan React Native.</li><li>d. Kemampuan untuk mengelola <i>state</i> aplikasi dengan Redux atau <i>Context API</i>.</li></ul>	<ul style="list-style-type: none"><li>a. Hanya <i>library</i> UI, memerlukan kombinasi dengan <i>library</i> atau <i>framework</i> lain.</li><li>b. Perlu pengetahuan ekstensif tentang ekosistem React.</li><li>c. <i>State management</i> yang kompleks untuk aplikasi yang lebih besar.</li></ul>

<b>Laravel</b>	a. <i>Framework</i> PHP yang kuat dengan dukungan ORM, autentikasi, dan banyak fitur bawaan. b. Struktur MVC yang jelas dan konvensi pengkodean yang baik. c. Banyak alat bantu untuk pengembangan. d. Mudah untuk mengelola basis data dengan <i>Eloquent</i> ORM. e. Dukungan <i>caching</i> yang baik untuk meningkatkan performa aplikasi.	a. Performa mungkin lebih lambat dibandingkan dengan <i>framework</i> PHP yang lebih ringan. b. Memerlukan pengetahuan tentang PHP. c. Kode menjadi kompleks jika aplikasi sangat besar dan rumit. d. SSG tidak langsung didukung, meskipun bisa diimplementasikan dengan kerja tambahan.
----------------	--	--

Setelah mempertimbangkan kelebihan dan kekurangan masing-masing solusi, dipertimbangkan penggunaan Laravel sebagai solusi teknis untuk proyek ini. Laravel memberikan dukungan lengkap untuk *backend*, termasuk interaksi dengan *database*, *routing*, autentikasi, dan lainnya sehingga dimungkinkan pengembangan sistem informasi yang kuat dan kompleks. Selain itu, kemampuan Laravel dalam mengelola data dan fungsionalitas yang luas akan sangat bermanfaat dalam membangun aplikasi yang memerlukan kompleksitas yang tinggi, seperti sistem informasi *Capstone* dan TA di Departemen Teknik Komputer.

#### b. Proses Pemilihan Solusi berbasis Android

Pada tabel 2. 2 akan dibandingkan kelebihan dan kekurangan dari masing-masing alternatif solusi untuk sistem berbasis Android.

Tabel 2. 2 Pemilihan solusi berbasis Android

<b>Solusi</b>	<b>Kelebihan</b>	<b>Kekurangan</b>
<b>Flutter</b>	a. Kemampuan untuk mengembangkan aplikasi Android dan iOS dengan satu kode. b. Antarmuka pengguna yang menarik dan responsif. c. Dukungan untuk <i>widget</i> kustom dan animasi.	a. Penggunaan <i>rendering</i> sendiri dapat menyebabkan masalah integrasi dengan perangkat Android. b. Keterbatasan akses ke fitur perangkat yang spesifik.

<b>Android Native Java</b>	<ul style="list-style-type: none"> <li>a. Dukungan penuh dari Google dan komunitas Android dengan banyak referensi dan dokumentasi.</li> <li>b. Kontrol penuh terhadap fitur perangkat Android.</li> <li>c. Akses ke berbagai API Android.</li> </ul>	<ul style="list-style-type: none"> <li>a. Pengembangan dengan Java mungkin memerlukan lebih banyak waktu dan usaha.</li> <li>b. Kurang efisien dibandingkan dengan Kotlin dalam beberapa kasus.</li> </ul>
<b>Android Native Kotlin</b>	<ul style="list-style-type: none"> <li>a. Adopsi dari pendekatan yang sudah mapan dengan dukungan penuh dari Android SDK.</li> <li>b. Kombinasi kekuatan bahasa Kotlin dengan tata letak yang didefinisikan secara deklaratif melalui XML.</li> </ul>	<ul style="list-style-type: none"> <li>a. Pengembangan dengan Kotlin x XML Views dapat menghasilkan kode <i>boilerplate</i>.</li> <li>b. Kesulitan dalam menangani tampilan yang sangat kompleks.</li> </ul>

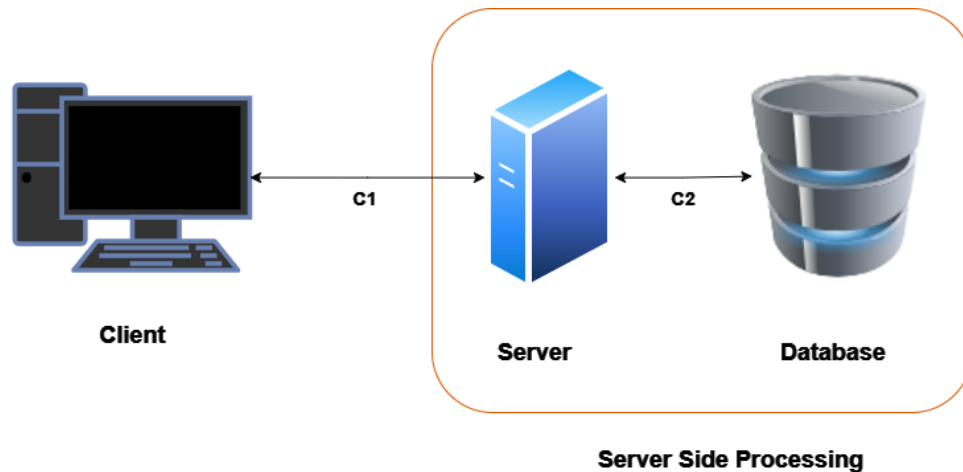
Setelah mempertimbangkan kelebihan dan kekurangan masing-masing solusi, dipilih penggunaan Android Native Kotlin x XML Views sebagai solusi teknis untuk proyek ini. Dengan menggunakan bahasa Kotlin yang ringkas dan ekspresif, didukung penuh oleh Google sebagai bahasa pemrograman resmi untuk pengembangan Android, serta pendekatan konvensional dalam pembangunan antarmuka pengguna dengan XML Views, pengembangan aplikasi akan menjadi lebih stabil dan teruji. Teknologi ini telah mapan dan mendapatkan dukungan penuh dari Android SDK, sehingga memastikan keselarasan dengan *platform* Android.

### 3. Desain Produk yang Diusulkan

#### 3.1. Arsitektur Sistem

##### a. Arsitektur Sistem berbasis Web

Arsitektur Sistem Informasi Pengelolaan *Capstone* dan TA berbasis web dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Arsitektur sistem berbasis web

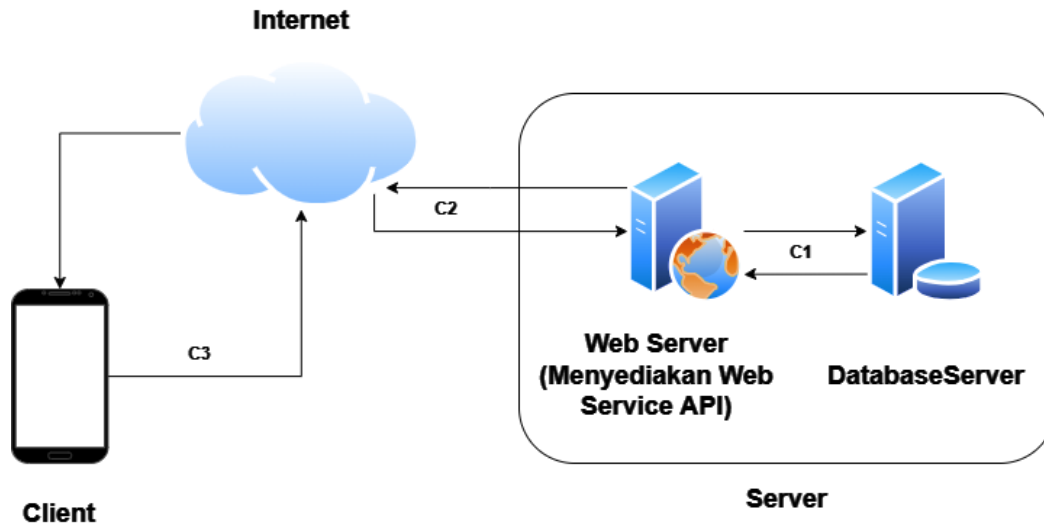
Arsitektur sistem berbasis web terdiri dari tiga komponen utama, yaitu *client*, *server*, dan *database*. *Client* adalah perangkat yang digunakan oleh pengguna. Perangkat *client* akan berinteraksi dengan *server* melalui koneksi internet dengan protokol HTTP. *Client* dapat mengirimkan *request*, kemudian dari *request* tersebut akan diteruskan ke *database* oleh *server*. *Response* kemudian akan dikembalikan ke *client* juga melalui *server*. Masing-masing komponen memiliki hubungan yang dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Hubungan komponen arsitektur sistem berbasis web

Kode	Komponen Pengirim	Komponen Penerima	Media Transmisi	Metode Transmisi Data
C1	<i>Client</i>	<i>Server</i>	Internet	HTTP
C2	<i>Server</i>	<i>Database</i>	PDO	PDO

#### b. Arsitektur Sistem berbasis Android

Arsitektur Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 3. 2.



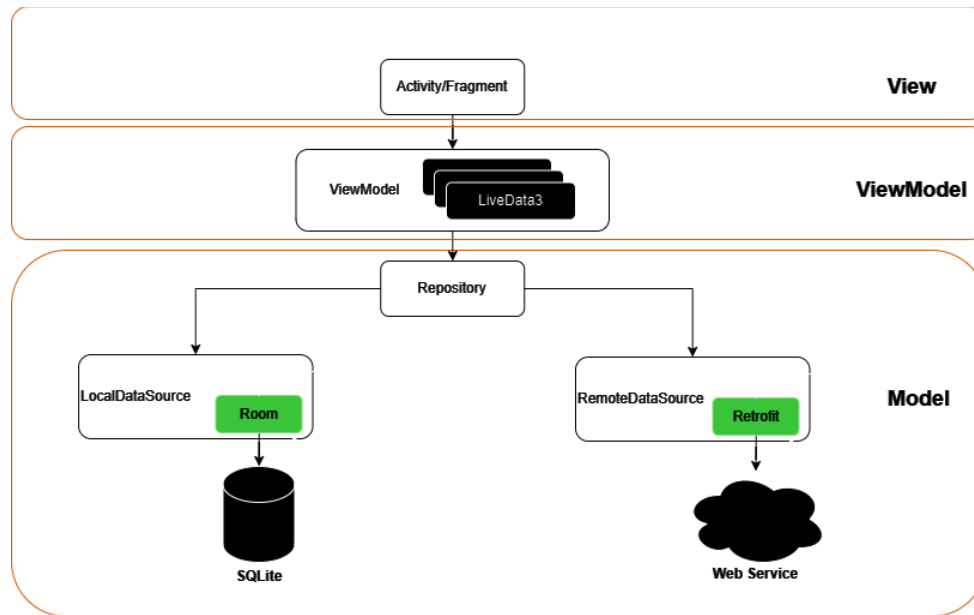
Gambar 3. 2 Arsitektur sistem berbasis Android

Arsitektur komponen sistem berbasis Android memiliki tiga komponen utama, yaitu *client*, *internet*, dan *server* (*web server* dan *database server*). API akan menjadi jembatan transportasi antara perangkat *client* dengan *database* aplikasi. Masing-masing komponen memiliki hubungan yang dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Hubungan komponen arsitektur sistem berbasis Android

Kode	Komponen Pengirim	Komponen Penerima	Media Transmisi	Metode Transmisi Data
C1	Web Server	Database Server	PDO	PDO
C2	Internet	Web Server	Internet	HTTP Request
C3	Client	Internet	Internet	HTTP Request dengan pustaka Retrofit

Aplikasi Android pada sistem informasi *Capstone* dan TA di Departemen Teknik Komputer memiliki arsitektur pengembangan yang dapat dilihat pada Gambar 3. 3.



Gambar 3. 3 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*.

## 3.2. Desain Detail Sistem

### 3.2.1. Data Description

Pada bagian desain detail system ini ada beberapa tabel yang menjelaskan mengenai 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 3.3.

Tabel 3. 3 Struktur tabel *app\_menu*

Nama Kolom	Tipe Data	Default	Constraint
menu_id	VARCHAR (3)	None	Primary Key
role_id	VARCHAR (3)	NULL	Foreign Key
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 3.4.

Tabel 3. 4 Struktur tabel *app\_role*

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
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 3.5.

Tabel 3. 5 Struktur tabel *app\_user*

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
user_id	VARCHAR (15)	<i>None</i>	<i>Primary Key</i>
user_name	VARCHAR (50)	<i>None</i>	-
user_email	VARCHAR (50)	NULL	-
role_id	VARCHAR (3)	NULL	<i>Foreign Key</i>

user_password	VARCHAR (255)	<i>None</i>	-
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 (11)	NULL	-
ipk	VARCHAR (20)	NULL	-
sks	INT (11)	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 3.6.

Tabel 3. 6 Struktur tabel *broadcast*

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (4)	<i>None</i>	<i>Primary Key</i>
id_siklus	INT (4)	NULL	<i>Foreign Key</i>
nama_event	VARCHAR (200)	NULL	-
link_pendukung	TEXT	NULL	-
keterangan	TEXT	NULL	-
tgl_mulai	DATE	NULL	-
tgl_selesai	DATE	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-



modified_date	DATETIME	NULL	-
broadcast_image_name	TEXT	<i>None</i>	-
broadcast_image_path	TEXT	<i>None</i>	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel dosen\_kelompok seperti yang ditunjukkan pada Tabel 3.7.

Tabel 3. 7 Struktur tabel dosen\_kelompok

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>
id_kelompok	INT (15)	NULL	<i>Foreign Key</i>
id_dosen	VARCHAR (15)	NULL	<i>Foreign Key</i>
status_dosen	ENUM	NULL	-
status_persetujuan	ENUM	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel jadwal\_expo seperti yang ditunjukkan pada Tabel 3.8.

Tabel 3. 8 Struktur tabel jadwal\_expo

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
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 3.9.

Tabel 3. 9 Struktur tabel jadwal\_sidang\_proposal

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>
id_kelompok	INT (15)	NULL	<i>Foreign Key</i>
siklus_id	INT (15)	NULL	<i>Foreign Key</i>
tanggal_mulai	DATE	NULL	-
waktu	TIME	NULL	-
ruangan_id	VARCHAR (255)	<i>None</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\_t\_a\_s seperti yang ditunjukkan pada Tabel 3.10.

Tabel 3. 10 Struktur tabel jadwal\_sidang\_t\_a\_s

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	BIGINT (20)	<i>None</i>	<i>Primary Key</i>
id_kelompok	INT (11)	<i>None</i>	<i>Foreign Key</i>
siklus_id	INT (11)	<i>None</i>	<i>Foreign Key</i>
tanggal_mulai	DATE	<i>None</i>	-
waktu	TIME	<i>None</i>	-
ruangan_id	BIGINT (20)	<i>None</i>	-
created_at	TIMESTAMP	NULL	-
modified_at	TIMESTAMP	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel kelompok seperti yang ditunjukkan pada Tabel 3.11.

Tabel 3. 11 Struktur tabel kelompok

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>

id_siklus	INT (15)	NULL	<i>Foreign Key</i>
nomor_kelompok	INT (25)	NULL	-
id_topik	INT (25)	NULL	<i>Foreign Key</i>
status_kelompok	ENUM	NULL	-
id_dosen_pembimbing_1	VARCHAR (15)	NULL	-
id_dosen_pembimbing_1	VARCHAR (15)	NULL	-
id_dosen_penguji_1	VARCHAR (15)	NULL	-
id_dosen_penguji_2	VARCHAR (15)	NULL	-
status_dosen_pembimbing_1	ENUM	NULL	-
status_dosen_pembimbing_1	ENUM	NULL	-
status_dosen_penguji_1	ENUM	NULL	-
status_dosen_penguji_2	ENUM	NULL	-
file_name_c100	VARCHAR (200)	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	-
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	-
progress_kelompok	ENUM	NULL	-
judul_capstone	VARCHAR (255)	<i>None</i>	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel kelompok\_mhs seperti yang ditunjukkan pada Tabel 3.12.

Tabel 3. 12 Struktur tabel kelompok\_mhs

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>
id_siklus	INT (15)	NULL	<i>Foreign Key</i>
id_kelompok	INT (15)	NULL	<i>Foreign Key</i>
id_mahasiswa	VARCHAR (15)	NULL	<i>Foreign Key</i>
id_topik_mhs	INT (15)	NULL	<i>Foreign Key</i>
status_individu	ENUM	NULL	-
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	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-
link_upload	TEXT	NULL	-
judul_ta_mhs	VARCHAR (100)	NULL	-
progress_kelompok	ENUM	NULL	-

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

Tabel 3. 13 Struktur tabel peminatan

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
-------------------	------------------	----------------	-------------------

id	INT (4)	<i>None</i>	<i>Primary Key</i>
id_mahasiswa	VARCHAR (15)	NULL	<i>Foreign Key</i>
id_kel_mhs	INT (15)	NULL	<i>Foreign Key</i>
peminatan	ENUM	NULL	-
prioritas	VARCHAR (15)	NULL	-

Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel pendaftaran\_capstone seperti yang ditunjukkan pada Tabel 3.14.

Tabel 3. 14 Struktur tabel pendaftaran\_capstone

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>
siklus_id	INT (15)	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 3.15.

Tabel 3. 15 Struktur tabel pendaftaran\_expo

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>
id_kelompok	INT (15)	NULL	<i>Foreign Key</i>
id_expo	INT (15)	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 ruang\_sidangs seperti yang ditunjukkan pada Tabel 3.16.

Tabel 3. 16 Struktur tabel ruang\_sidangs

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	BIGINT (20)	<i>None</i>	<i>Primary Key</i>
kode_ruang	VARCHAR (255)	NULL	-
nama_ruang	VARCHAR (255)	NULL	-
created_at	TIMESTAMP	NULL	-
modified_at	TIMESTAMP	NULL	-

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

Tabel 3. 17 Struktur tabel siklus

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
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 3.18.

Tabel 3. 18 Struktur tabel topik

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
-------------------	------------------	----------------	-------------------

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	-

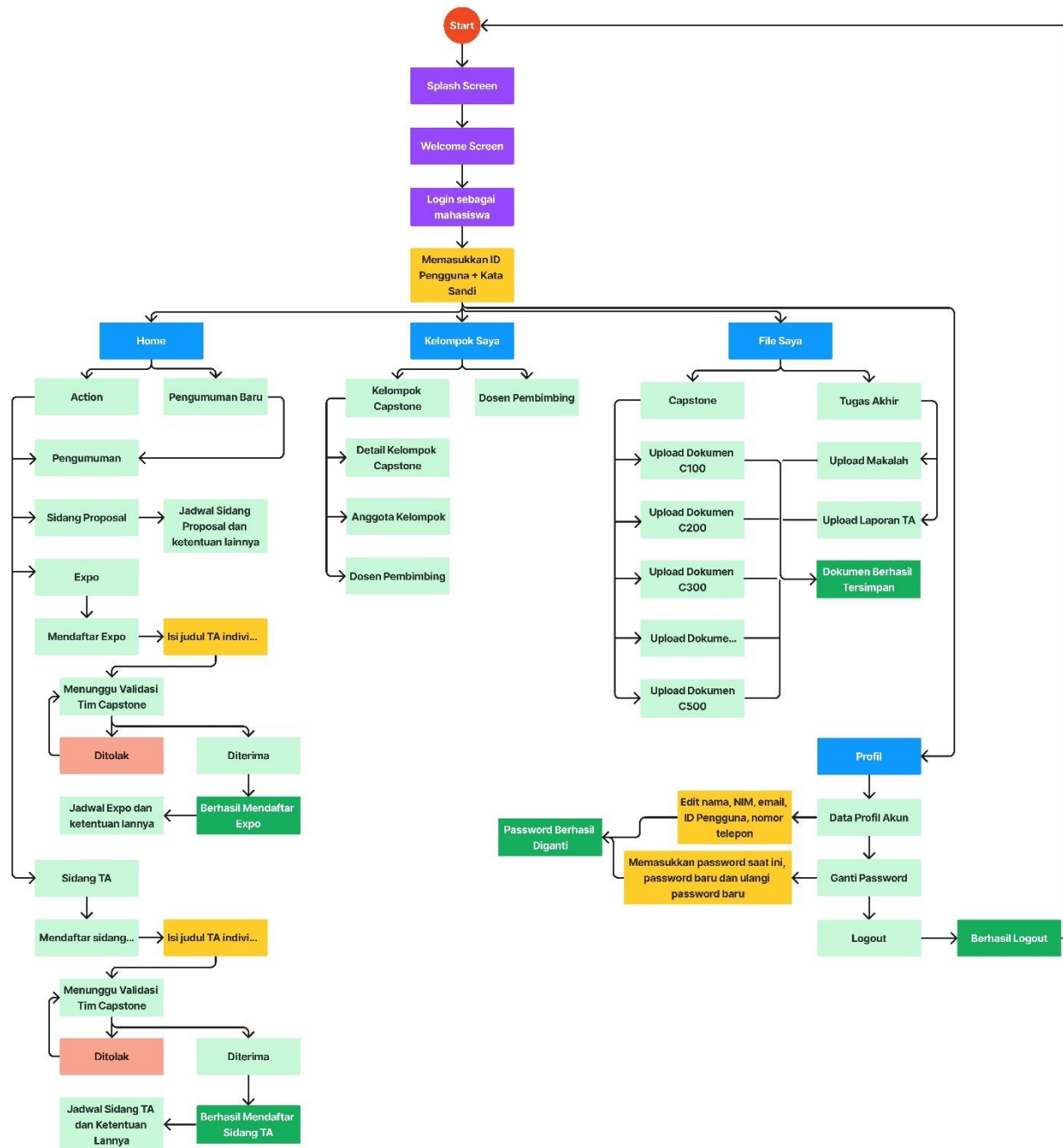
Berikut ini merupakan bagian data *database* yang menjelaskan mengenai struktur tabel topik\_mhs seperti yang ditunjukkan pada Tabel 3.19.

Tabel 3. 19 Struktur tabel topik\_mhs

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Default</b>	<b>Constraint</b>
id	INT (15)	<i>None</i>	<i>Primary Key</i>
id_mahasiswa	VARCHAR (15)	NULL	<i>Foreign Key</i>
id_kel_mhs	INT (15)	NULL	<i>Foreign Key</i>
id_topik	INT (15)	NULL	<i>Foreign Key</i>
prioritas	VARCHAR (15)	NULL	-
created_by	VARCHAR (15)	NULL	-
created_date	DATETIME	NULL	-
modified_by	VARCHAR (15)	NULL	-
modified_date	DATETIME	NULL	-

### 3.2.2. Diagram Alur

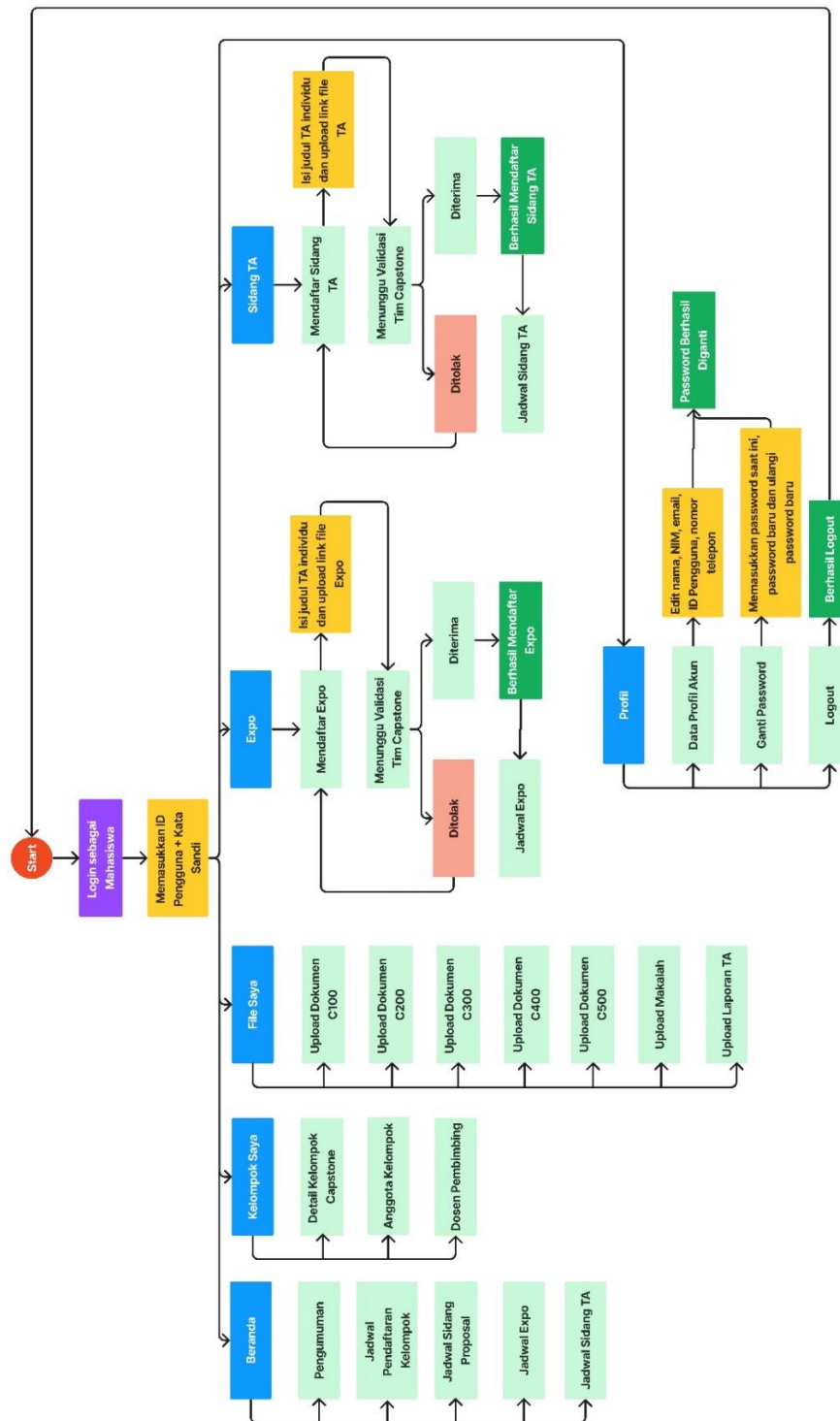
Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android memiliki diagram alur. Diagram alur adalah diagram yang berisi langkah atau proses untuk mengoperasikan sebuah program. Diagram alur dari Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dan web terdapat beberapa diagram alur sistem. Berikut ini merupakan diagram alur Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android seperti yang ditunjukkan pada Gambar 3.4.



Gambar 3. 4 Diagram alur dari Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

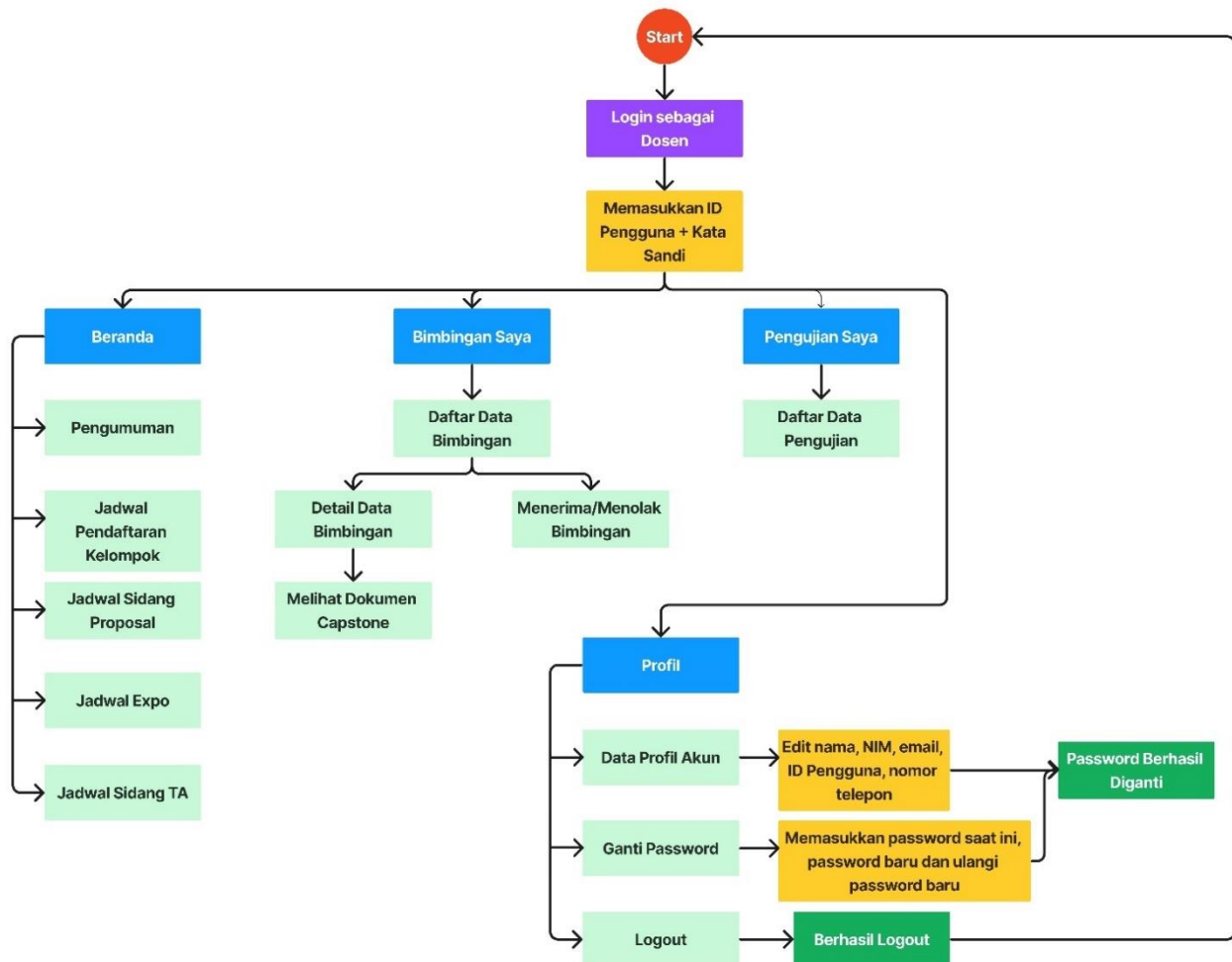


Berikut ini merupakan diagram alur mahasiswa pada Sistem Informasi Pengelolaan Capstone dan TA berbasis website seperti yang ditunjukkan pada Gambar 3.5.



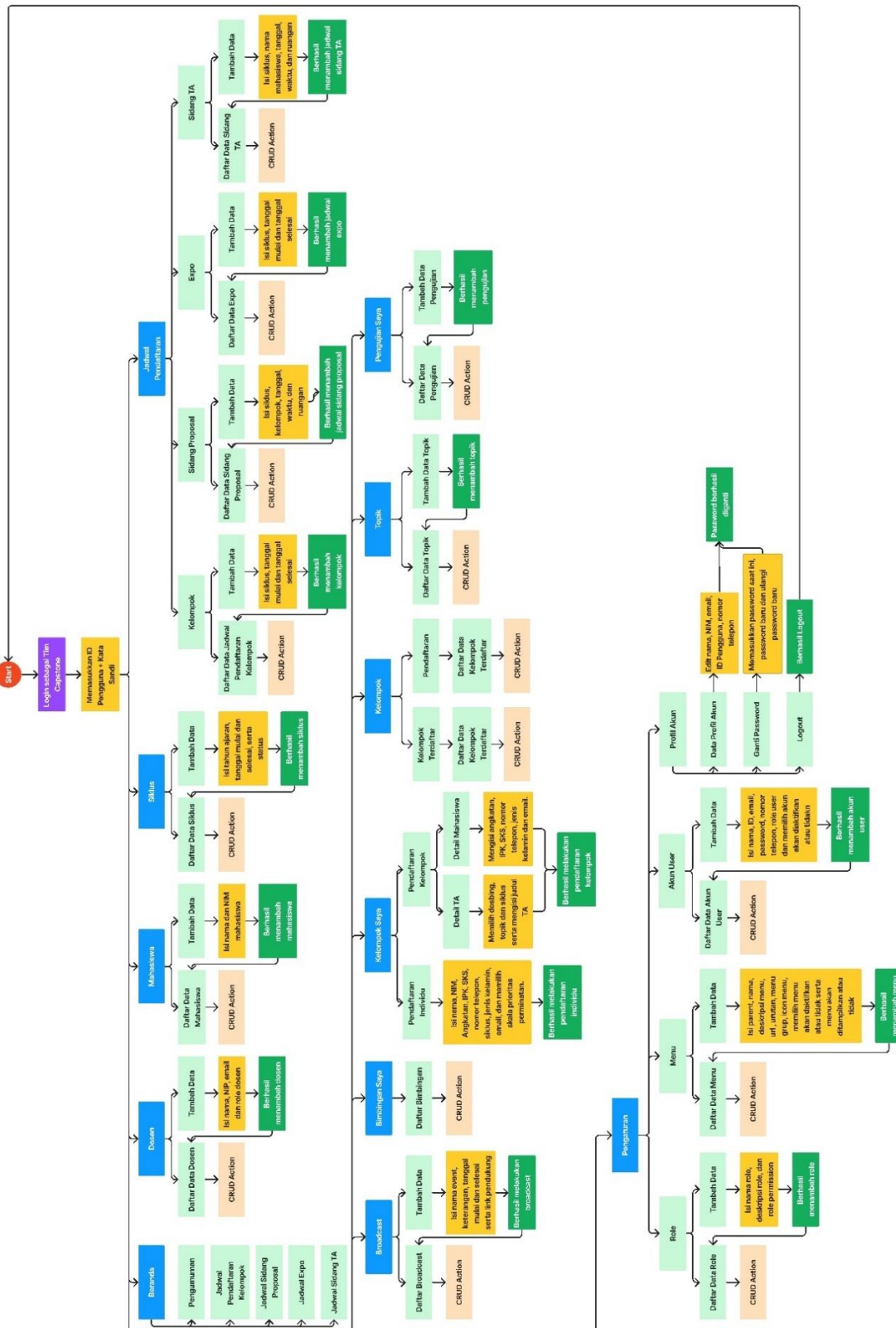
Gambar 3. 5 Diagram alur Mahasiswa dari Sistem Informasi Pengelolaan *Capstone* dan berbasis web

Berikut ini merupakan diagram alur dosen pada Sistem Informasi Pengelolaan Capstone dan TA berbasis website seperti yang ditunjukkan pada Gambar 3.6.



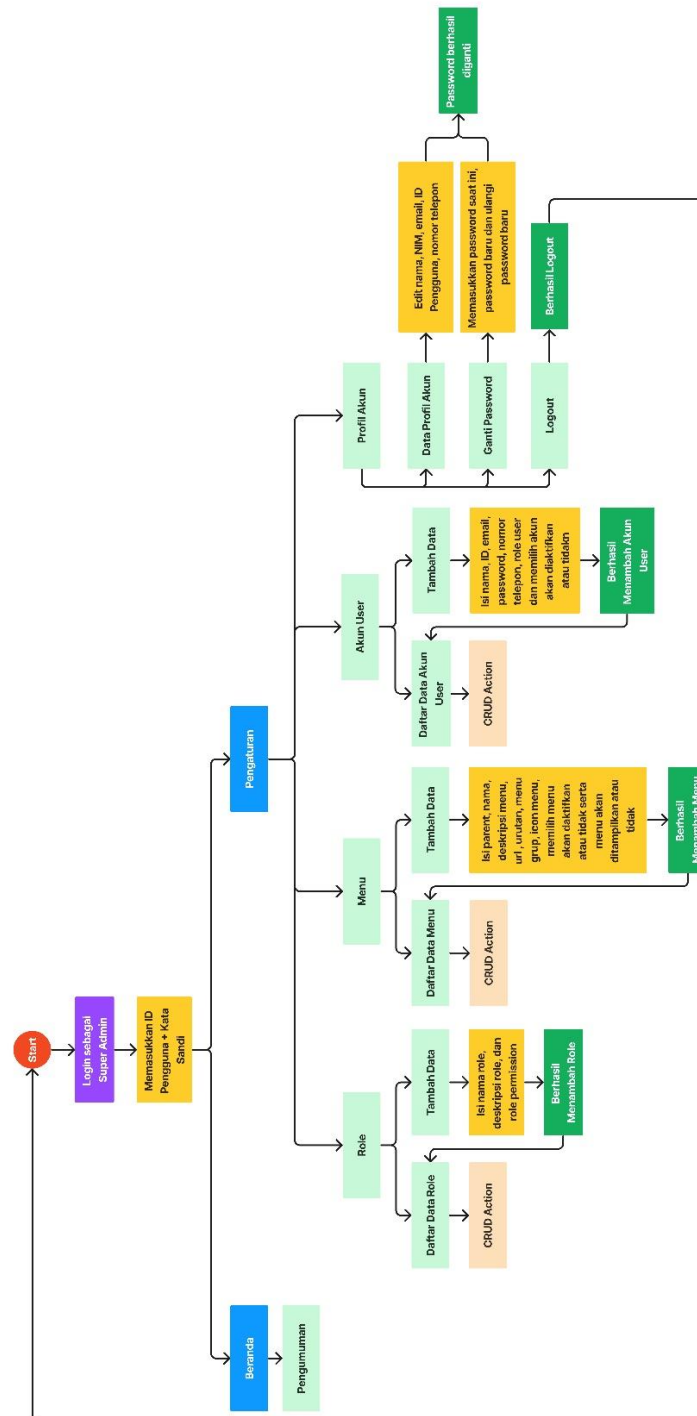
Gambar 3. 6 Diagram alur Dosen dari Sistem Informasi Pengelolaan *Capstone* dan TA berbasis web

Berikut ini merupakan diagram alur tim Capstone pada Sistem Informasi Pengelolaan Capstone dan TA berbasis website seperti yang ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Diagram alur Tim Capstone dari Sistem Informasi Pengelolaan Capstone dan TA berbasis web

Berikut ini merupakan diagram alur tim Capstone pada Sistem Informasi Pengelolaan Capstone dan TA berbasis website seperti yang ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Diagram alur Superadmin dari Sistem Informasi Pengelolaan *Capstone* dan TA berbasis web

### 3.2.3. Standar - standar yang Dipergunakan

Dalam pengembangan Sistem Informasi Pengelolaan *Capstone* dan TA digunakan beberapa standar untuk memastikan kualitasnya. Untuk komunikasi antara aplikasi dan *database*, diterapkan standar protokol HTTP. Proses transfer data dalam aplikasi dapat dilakukan melalui REST API atau melalui operasi CRUD, yang mana data diambil dari *database manager* di web *hosting*. Aplikasi ini juga menyertakan mekanisme *session login* dan *logout* untuk mencegah pengguna melakukan *bypass* pada halaman tertentu sebelum melakukan autentikasi oleh sistem. Untuk menjaga keamanan data privasi pengguna, terutama *password*, diterapkan fungsi *hash* untuk mengenkripsi *password*.

### 3.2.4. Method/API

Pada pembuatan Sistem Informasi Pengelolaan *Capstone* dan TA akan terdapat sekumpulan API *endpoint* (*web service*) dari aplikasi *backend* yang akan digunakan untuk menjalankan logika bisnis sistem. Salah satu fitur yang dapat dijadikan sebagai acuan bentuk API yang akan dikembangkan secara umum dan mengaplikasikan CRUD adalah fitur pengelolaan data mahasiswa. Berikut pada Tabel 3.20 merupakan daftar API *endpoint* untuk fitur pengelolaan mahasiswa.

Tabel 3. 20 Daftar API mahasiswa

<i>Method</i>	<i>Route Path</i>	Keterangan
<i>GET</i>	/admin/mahasiswa	<i>Endpoint</i> ini digunakan untuk mendapatkan daftar mahasiswa yang ada pada <i>database</i> .
<i>POST</i>	/admin/mahasiswa/add-process	<i>Endpoint</i> ini digunakan untuk menambahkan mahasiswa ke dalam tabel mahasiswa pada <i>database</i> .
<i>POST</i>	/admin/mahasiswa/edit-process	<i>Endpoint</i> ini digunakan untuk melakukan perubahan data dalam tabel mahasiswa pada <i>database</i> .
<i>POST</i>	/admin/mahasiswa/delete-process/{user_id}	<i>Endpoint</i> ini digunakan untuk menghapus data dalam tabel mahasiswa pada <i>database</i> .

### 3.3. Traceable

#### 3.3.1. Data

Hubungan tabel-tabel dalam basis-data dengan desain ER yang telah dibuat dapat dilihat pada Tabel 3. 21.

Tabel 3. 21 Status tabel basis data

<b>Nama Tabel</b>	<b>Primary key</b>	<b>Entity Class</b>	<b>ER</b>	<b>Deskripsi isi</b>
app_menu	menu_id	parent_menu_id, menu_name, role_id, menu_description, menu_url, menu_sort, menu_group, menu_icon, menu_active, menu_display, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi daftar menu yang bisa diakses oleh pengguna.
app_role	role_id	role_name, role_description, role_permission, created_by, created_date, modified_by, modified_date	-	Berisi data hak akses tiap pengguna.
app_user	user_id	user_name, user_email, role_id, user_password, user_active, user_img_path, user_img_name, nomor_induk, no_telp, angkatan, ipk, sks, alamat, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi data pengguna aplikasi.
broadcast	id	id_siklus, nama_event, link_pendukung, keterangan, tgl_mulai, tgl_selesai, created_by, created_date, modified_by, modified_date, broadcast_image_name, broadcast_image_path	<i>Need</i>	Berisi data pengumuman dari departemen.

dosen_kelompok	id	id_kelompok, id_dosen, status_dosen, status_persetujuan	<i>Need</i>	Berisi data status dosen sebagai pembimbing dan penguji.
jadwal_expo	id	id_siklus, tanggal_mulai, tanggal_selesai, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi data jadwal <i>expo</i> masing-masing siklus.
jadwal_sidang_proposal	id	id_kelompok, siklus_id, tanggal_mulai, waktu, ruangan_id, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi data jadwal sidang proposal kelompok tiap siklus.
jadwal_sidang_tas	id	id_kelompok, siklus_id, tanggal_mulai, waktu, ruangan_id, created_at, modified_at	<i>Need</i>	Berisi data jadwal sidang TA tiap siklus.
kelompok	id	id_siklus, nomor_kelompok, id_topik, status_kelompok, id_dosen_pembimbing_1, id_dosen_pembimbing_1, id_dosen_penguji_1, id_dosen_penguji_2, status_dosen_pembimbing_1, status_dosen_pembimbing_1, status_dosen_penguji_1, status_dosen_penguji_2, file_name_c100, file_path_c100, file_name_c200, file_path_c200,	<i>Need</i>	Berisi data dokumen yang diunggah oleh setiap kelompok mahasiswa.

		file_name_c300, file_path_c300, file_name_c400, file_path_c400, file_name_c500, file_path_c500, created_by, created_date, modified_by, modified_date, progress_kelompok, judul_capstone		
kelompok_mhs	id	id_siklus, id_kelompok, id_mahasiswa, id_topik_mhs, status_individu, file_name_laporan_ta, file_path_laporan_ta, file_name_makalah, file_path_makalah, created_by, created_date, modified_by, modified_date, link_upload, judul_ta_mhs, progress_kelompok	<i>Need</i>	Berisi data laporan setiap anggota kelompok.
peminatan	id	id_mahasiswa, id_kel_mhs, peminatan, prioritas	<i>Need</i>	Berisi data skala prioritas peminatan yang dipilih oleh setiap mahasiswa.
pendaftaran_capstone	id	siklus_id, tanggal_mulai, tanggal_selesai, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi periode pendaftaran <i>Capstone</i> berlangsung setiap siklus.
pendaftaran_expo	id	id_kelompok, id_expo, status, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi id yang digunakan untuk menyatukan



				data pada saat pendaftaran expo.
ruang_sidangs	id	kode_ruang, nama_ruang, created_at, modified_at	-	Berisi data ruangan sidang di Departemen Teknik Komputer.
siklus	id	tahun_ajaran, tanggal_mulai, tanggal_selesai, status, created_by, created_date, modified_by, modified_date	-	Berisi data durasi periode siklus.
topik	id	nama, created_by, created_date, modified_by, modified_date	-	Berisi data topik.
topik_mhs	id	id_mahasiswa, id_kel_mhs, id_topik, prioritas, created_by, created_date, modified_by, modified_date	<i>Need</i>	Berisi data skala prioritas topik yang dipilih oleh mahasiswa.

### 3.3.2. Requirements

Spesifikasi kebutuhan perangkat lunak pada Sistem Informasi Pengelolaan *Capstone* dan TA dapat dilihat pada Tabel 3.22.

Tabel 3. 22 Method HTTP

SRS-ID	Nama <i>Method</i>	Keterangan
SW-SIPCTA-SRS01	<i>POST</i>	Pengguna mengirimkan data <i>login</i> ketika menekan tombol <i>login</i> .

SW-SIPCTA-SRS02	<i>POST</i>	Mahasiswa mengirimkan data yang dimasukkan pada formulir pendaftaran setelah menekan tombol <i>submit</i> .
SW-SIPCTA-SRS03	<i>GET</i>	Admin memperoleh data kelompok dari data dosen dan data pendaftaran.
SW-SIPCTA-SRS04	<i>POST</i>	Admin mengirimkan data kelompok yang telah dikelola setelah menekan tombol simpan.
SW-SIPCTA-SRS05	<i>GET</i>	Mahasiswa dan Dosen memperoleh informasi pembagian kelompok pada halaman kelompok.
SW-SIPCTA-SRS06	<i>POST</i>	Admin mengirimkan data jadwal sidang proposal yang telah dikelola setelah menekan tombol simpan.
SW-SIPCTA-SRS07	<i>GET</i>	Mahasiswa dan Dosen memperoleh informasi jadwal sidang proposal.
SW-SIPCTA-SRS08	<i>POST</i>	Mahasiswa mengirimkan data pada formulir pendaftaran expo dengan menekan tombol <i>submit</i> .
SW-SIPCTA-SRS09	<i>GET</i>	Admin memperoleh data pendaftaran expo.
SW-SIPCTA-SRS10	<i>POST</i>	Admin mengelola data pendaftaran expo yang didapat, kemudian mengirimkan data yang telah dikelola dengan menekan tombol tambah.
SW-SIPCTA-SRS11	<i>GET</i>	Mahasiswa dan Dosen memperoleh informasi jadwal expo.
SW-SIPCTA-SRS12	<i>POST</i>	Mahasiswa mengunggah setiap dokumen yang telah disetujui oleh dosen.
SW-SIPCTA-SRS13	<i>GET</i>	Admin dan Dosen melihat dan melakukan pengecekan dokumen yang diunggah oleh mahasiswa.
SW-SIPCTA-SRS14	<i>POST</i>	Admin mengirim data jadwal sidang akhir dengan menekan tombol tambah.
SW-SIPCTA-SRS15	<i>GET</i>	Mahasiswa dan Dosen memperoleh informasi jadwal sidang tugas akhir.

Keterangan:

SW = *Software*

SIPCTA = Sistem Informasi Pengelolaan Capstone dan TA

SRS = *Software Requirements Specification*

## 4. Verifikasi Desain Produk

### 4.1. Hasil Simulasi Awal Produk

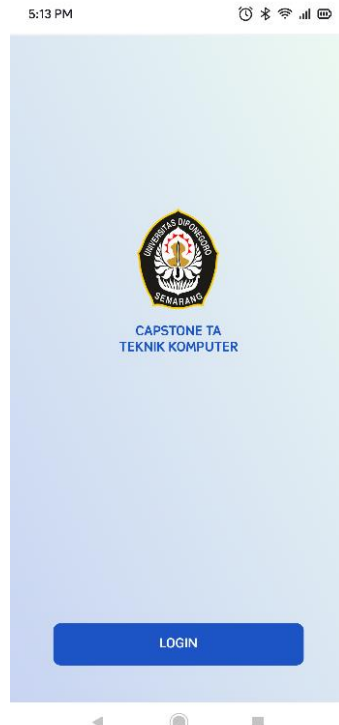
Simulasi tampilan desain halaman *splash screen* Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 4. 1. Tampilan ini akan muncul sesaat pada saat pengguna pertama kali mengakses aplikasi ini.



Gambar 4. 1 Halaman *splash screen* Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

*Splash screen* ini digunakan untuk tampilan pertama pada aplikasi ini untuk memberitahu kepada pengguna bahwa aplikasi yang sedang dibuka saat ini yaitu aplikasi Sistem Informasi Pengelolaan Capstone dan TA. Pada *splash screen* ini terdapat logo Universitas Diponegoro beserta tulisan “CAPSTONE TA TEKNIK KOMPUTER” yang menjadi ciri khas dari aplikasi berbasis Android ini.

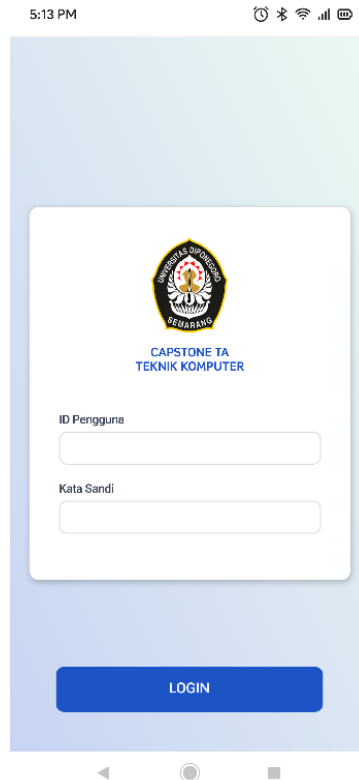
Setelah tampilan *splashscreen* selesai, akan muncul halaman *landing page*. Simulasi tampilan desain halaman *landing page* Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 4. 2.



Gambar 4. 2 Halaman *landing page* Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

Tidak jauh berbeda dengan tampilan desain *splash screen*, halaman *landing page* ini terdapat logo Universitas Diponegoro beserta tulisan “CAPSTONE TA TEKNIK KOMPUTER”. Letak perbedaan antara halaman *splash screen* dan halaman *landing page* ini terdapat pada penambahan tombol CTA (*Call to Action*) bertuliskan “LOGIN” yang apabila ditekan akan menuju ke halaman *login* untuk masuk ke dalam aplikasi.

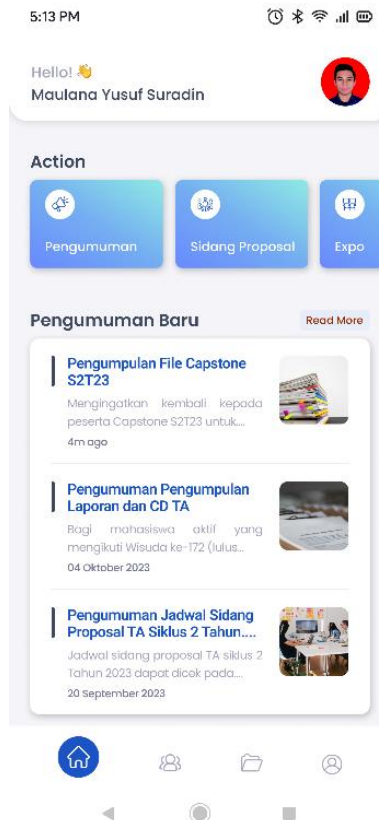
Setelah masuk ke dalam aplikasi, aplikasi akan menampilkan halaman *login* untuk memvalidasi bahwa pengguna yaitu mahasiswa aktif Program Studi Teknik Komputer Universitas Diponegoro merupakan mahasiswa yang telah terdaftar pada *database* Sistem Informasi Pengelolaan *Capstone* TA. Simulasi tampilan desain halaman *login* dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Halaman *login* pengguna Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

Pada tampilan halaman *login* terdapat sebuah *card* yang berisi logo Universitas Diponegoro beserta tulisan “CAPSTONE TA TEKNIK KOMPUTER” serta dua *form* yang terdiri dari *form* ID pengguna dan *form* kata sandi. ID pengguna merupakan nomor unik yang dimiliki oleh setiap mahasiswa yang telah terdaftar serta kata sandi merupakan kata kunci terenkripsi yang telah dibuat oleh tiap mahasiswa yang tentunya kedua hal tersebut merupakan kata kunci unik yang masing-masing mahasiswa berbeda sehingga mahasiswa satu dan lainnya tidak dapat mengakses satu sama lain. Selain dua *form* itu, terdapat tombol CTA (*Call to Action*) bertuliskan “LOGIN” yang apabila ditekan akan menuju ke halaman aplikasi.

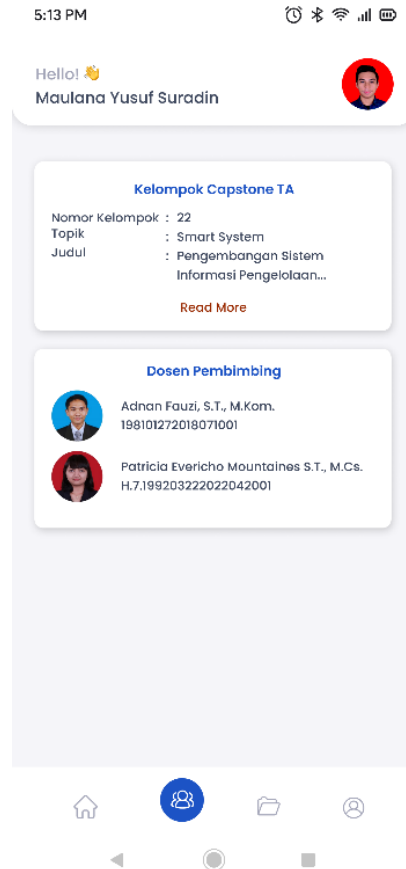
Halaman utama aplikasi hanya dapat diakses saat pengguna berhasil melakukan *login*. Simulasi tampilan desain halaman utama Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 4. 4.



Gambar 4. 4 Halaman utama Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

Halaman utama ini terdapat berbagai hal yaitu terdapat nama pengguna, foto profil pengguna yang ketika ditekan akan mengarahkan pengguna ke menu profil, beberapa *card* serta empat buah menu yang berbeda. *Card Action* pada halaman utama ini merupakan *card* untuk menampilkan berbagai macam pengumuman yang terdiri dari *card* pengumuman *broadcast* yang dapat diterima oleh pengguna dari admin, *card* sidang proposal, *card* expo, dan *card* sidang tugas akhir yang ketika ditekan masing-masing akan menampilkan pengumuman dan detail lainnya mengenai *card* yang sedang dibuka tersebut. Pada halaman utama ini terdapat satu *card* berukuran besar yang digunakan untuk menampilkan tiga pengumuman terbaru yang telah diterima oleh pengguna pada aplikasi ini. Aplikasi ini memiliki empat buah menu yaitu menu utama, menu kelompok, menu *file* dan menu profil.

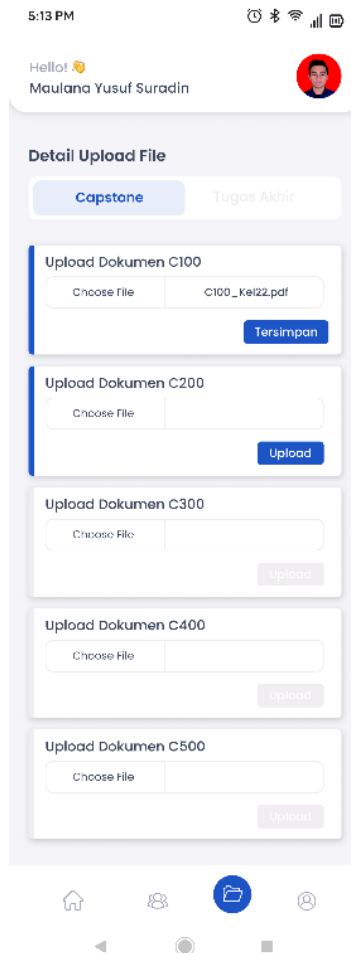
Selain halaman utama, aplikasi juga memiliki halaman kelompok. Simulasi tampilan desain halaman kelompok Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Halaman kelompok Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

Pada halaman ini terdapat nama pengguna, foto profil pengguna yang ketika ditekan akan mengarahkan pengguna ke menu profil, serta terdapat dua buah *card*. *Card* pertama merupakan *card* informasi singkat tentang kelompok Capstone TA pengguna dan *card* kedua merupakan *card* yang menampilkan dosen pembimbing Capstone TA pengguna. Pada *card* pertama terdapat tulisan CTA (*Call to Action*) yang jika ditekan akan mengarah ke halaman informasi lebih detail tentang *Capstone* TA pengguna.

Kemudian, terdapat halaman *file* untuk mengumpulkan seluruh *file* Capstone dan TA. Simulasi tampilan desain halaman file Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 4.6.

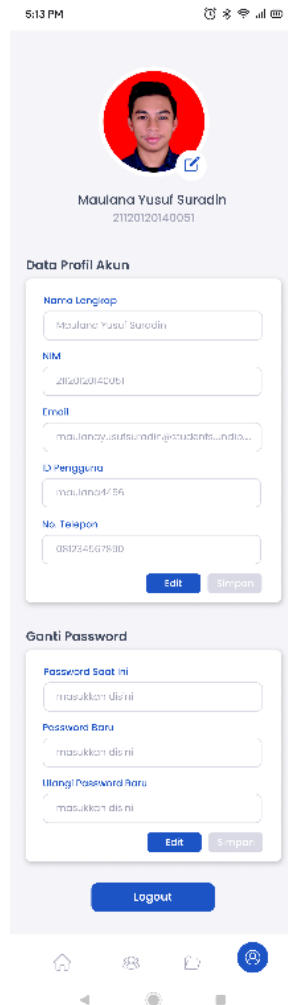


Gambar 4. 6 Halaman *file* Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

Pada halaman ini terdapat dua bagian detail *upload file* pada halaman ini. Bagian pertama yaitu bagian untuk mengunggah dokumen *Capstone* dari dokumen C100 sampai dengan C500 yang dipisahkan dengan *card* dan masing-masing *card* terdapat tombol CTA (*Call to Action*) yang dapat ditekan ketika pengguna telah memilih *file* yang akan di-*upload* dan ketika tombol tersebut ditekan, sistem akan langsung menyimpan *file* tersebut ke *database*. Bagian kedua yaitu bagian untuk mengunggah dokumen TA yang terdiri dari dua *card* yaitu *card upload* makalah TA dan laporan TA yang masing-masing *card* tersebut terdapat tombol CTA (*Call to Action*) *upload*.

Kemudian, terdapat halaman profil yang berisi informasi mahasiswa. Simulasi tampilan desain halaman profil Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android dapat dilihat pada Gambar 4.7.





Gambar 4. 7 Halaman profil Sistem Informasi Pengelolaan *Capstone* dan TA berbasis Android

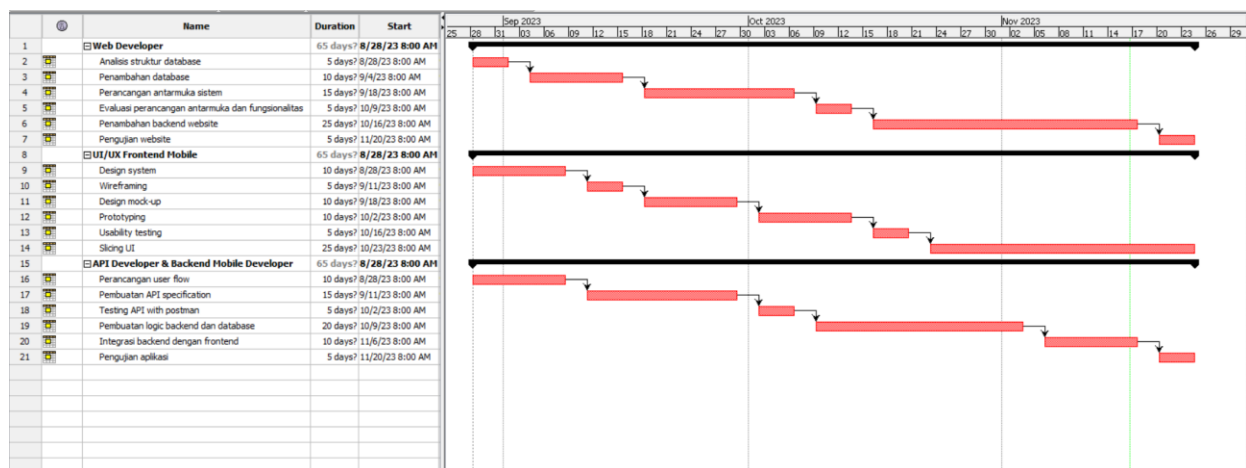
Pada halaman ini, terdapat foto pengguna, nama dan NIM pengguna serta terdapat 2 *card*. *Card* pertama merupakan *card* data profil akun yang terdiri dari nama lengkap, NIM, email, ID pengguna dan nomor telepon serta terdapat dua tombol CTA (*Call to Action*) *edit* yang ketika ditekan akan mengaktifkan *form* sehingga dapat dilakukan perubahan dari masing-masing data serta tombol *simpan* untuk melakukan simpan data dari data yang telah dilakukan perubahan ke *database*. *Card* kedua merupakan *card* ganti *password* yang berfungsi untuk melakukan perubahan *password*. Tahapan untuk mengubah *password* yaitu memasukkan *password* saat ini terlebih dahulu, kemudian memasukkan *password* baru dan setelah itu melakukan verifikasi ulang *password* baru tersebut. Setelah semua tahapan dilakukan, pengguna menekan tombol CTA (*Call to Action*) *simpan* untuk melakukan simpan data dari data yang telah diubah. Selain itu, pada halaman ini terdapat CTA *logout* yang berfungsi untuk mengeluarkan akun yang sedang digunakan

oleh pengguna sehingga nantinya halaman akan kembali ke halaman *login* dan pengguna harus kembali memasukkan *username* dan *password* saat akan *login* kembali pada aplikasi ini.

## 5. Rencana Implementasi dan Pengujian

### 5.1. Gaant Chart

Gaant Chart adalah sejenis grafik batang (*Bar Chart*) yang digunakan untuk menunjukkan tugas-tugas pada proyek serta jadwal dan waktu pelaksanaannya, seperti waktu dimulainya tugas tersebut dan juga batas waktu yang digunakan untuk menyelesaikan tugas yang bersangkutan. Gantt chart dari Sistem Informasi Pengelolaan *Capstone* dan TA bisa dilihat pada Gambar 5. 1.



Gambar 5. 1 Gaant Chart Sistem Informasi Pengelolaan Capstone dan TA

### 5.2. Metode Pengujian

Metode pengujian yang akan digunakan untuk UI (*User Interface*) Sistem Informasi Pengelolaan *Capstone* dan TA Departemen Teknik Komputer Universitas Diponegoro adalah *usability testing*. *Usability testing* adalah salah satu kategori metode dalam evaluasi *usability* yang digunakan untuk mengevaluasi sebuah produk dengan mengujinya langsung pada pengguna [6].

Selain itu, metode pengujian yang diterapkan pada Sistem Informasi Pengelolaan *Capstone* dan TA yaitu *blackbox*. *Blackbox* sendiri merupakan sebuah teknik pengujian yang berfokus pada spesifikasi fungsionalitas dari perangkat lunak program. Proses *blackbox testing* dilakukan dengan cara mencoba sebuah *software* yang telah selesai dibangun lalu pada setiap *form*-nya akan dicoba memasukkan sampel data. Pengujian ini sangat dibutuhkan untuk mengetahui apakah *software* yang dibangun telah berjalan sesuai dengan apa yang diinginkan oleh perusahaan [7].