Software design document (SDD)

FOR <<TUGAS AKHIR INFORMATIKA UNIVERSITAS XYZ>>

# Daftar Isi

[Daftar Isi 1](#_Toc155920174)

[Bab I Introduction 2](#_Toc155920175)

[1.1 Purpose 2](#_Toc155920176)

[1.2 Scope 2](#_Toc155920177)

[1.3 Overview 3](#_Toc155920178)

[1.4 Reference 4](#_Toc155920179)

[1.5 Definitions and Acronyms 5](#_Toc155920180)

[Bab II System Overview 16](#_Toc155920181)

[Bab III Application Design 18](#_Toc155920182)

[3.2 Use Case Diagram 18](#_Toc155920183)

[3.3 Use Case Scenario 18](#_Toc155920184)

[3.3 Class Diagram 20](#_Toc155920185)

[3.3 Sequence Diagran 21](#_Toc155920186)

[3.3 Activity Diagram 23](#_Toc155920187)

[3.3 State Diagram 26](#_Toc155920188)

[3.3 Deployment Diagram 30](#_Toc155920189)

[Bab IV Data Design 31](#_Toc155920190)

[4.1 Logical Design 31](#_Toc155920191)

[4.2 Physical Design 31](#_Toc155920192)

[Bab V User Interface Design 33](#_Toc155920193)

[Bab VI Interface Requirements 34](#_Toc155920194)

[4.1 User Interface 34](#_Toc155920195)

[4.2 Hardware Interface 37](#_Toc155920196)

[4.3 Software Interface 38](#_Toc155920197)

[4.4 Communication Interface 40](#_Toc155920198)

# Bab I Introduction

## 1.1 Purpose

Tujuan dari Dokumen Software Design Document (SDD) dalam Tugas Akhir Informatika XYZ adalah memberikan panduan rinci tentang rancangan dan implementasi perangkat lunak yang akan dibangun, memastikan kepatuhan terhadap persyaratan, dan membantu kolaborasi tim pengembang dalam mengimplementasikan proyek dengan efisien dan efektif.. Dokumen SDD ini dibuat untuk membantu mahasiswa memahami persyaratan dan prosedur yang harus mereka ikuti dalam menyelesaikan Tugas Akhir mereka.

Beberapa tujuan utama SDD dalam Tugas Akhir Informatika Universitas XYZ adalah:

1. Menggambarkan Rancangan Perangkat Lunak: SDD menjelaskan rincian arsitektur, desain, dan komponen perangkat lunak yang akan dikembangkan dan bagaimana komponen-komponennya berinteraksi.
2. Memastikan Kepatuhan terhadap Persyaratan: SDD mencakup persyaratan fungsional dan non-fungsional yang telah ditetapkan sebelumnya. Dokumen ini membantu memastikan bahwa desain perangkat lunak memenuhi persyaratan yang telah diperhitungkan dan diimplementasikan dengan benar.
3. Memfasilitasi Pengujian: SDD menyediakan informasi yang diperlukan untuk merencanakan dan melaksanakan pengujian perangkat lunak dan memahami rancangan perangkat lunak dan mengidentifikasi skenario pengujian yang sesuai.
4. Meningkatkan Efisiensi Pengembangan: SDD membantu menghindari perubahan desain yang tidak perlu selama tahap implementasi. Dengan memiliki rancangan yang jelas dan terstruktur sejak awal.
5. Dokumentasi dan Referensi: SDD berfungsi sebagai dokumen acuan yang komprehensif untuk proyek Tugas Akhir. Ini membantu dalam memahami dan mereplikasi rancangan perangkat lunak.

## 1.2 Scope

Scope (lingkup) Dokumen Software Design Document (SDD) dalam Tugas Akhir Informatika Universitas XYZ mencakup aspek-aspek berikut:

1. Rancangan Perangkat Lunak: SDD mencakup rancangan perangkat lunak yang akan dikembangkan dalam Tugas Akhir. Ini meliputi arsitektur perangkat lunak, komponen utama, antarmuka pengguna, dan rincian teknis terkait.
2. Komponen Perangkat Lunak: SDD mencakup penjelasan detail tentang setiap komponen perangkat lunak yang akan dibangun. Ini termasuk kelas-kelas, fungsi-fungsi, dan interaksi antara komponen-komponen tersebut.
3. Persyaratan Fungsional dan Non-fungsional: SDD mencakup persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh perangkat lunak. Persyaratan ini harus secara jelas dijelaskan dan dihubungkan dengan desain yang diusulkan.
4. Desain Antarmuka Pengguna: Jika perangkat lunak yang dibangun memiliki antarmuka pengguna, SDD mencakup rancangan antarmuka pengguna. Ini termasuk tata letak, aliran navigasi, elemen-elemen antarmuka, dan interaksi dengan pengguna.
5. Desain Database: Jika perangkat lunak melibatkan penggunaan database, SDD dapat mencakup rancangan database termasuk skema database, tabel, hubungan, dan pertanyaan database yang relevan.
6. Batasan Teknis: SDD harus mencakup batasan teknis yang relevan, seperti bahasa pemrograman yang akan digunakan, platform atau framework yang diperlukan, ketergantungan pihak ketiga, dan kapasitas perangkat keras atau jaringan yang diharapkan.
7. Pengujian: SDD harus memberikan panduan tentang bagaimana pengujian akan dilakukan. Ini mencakup skenario pengujian, metode pengujian, dan lingkungan pengujian yang relevan.
8. Keterbatasan dan Asumsi: SDD harus mencakup keterbatasan dan asumsi yang relevan terkait dengan perancangan dan implementasi perangkat lunak.
9. Rencana Implementasi: SDD dapat mencakup rencana implementasi yang mencakup estimasi waktu, alokasi sumber daya, dan tugas-tugas yang harus diselesaikan untuk menyelesaikan pembangunan perangkat lunak.

Penting untuk menjaga agar SDD tetap relevan dengan Tugas Akhir dan tidak mencakup hal-hal yang di luar lingkup proyek. Lingkup SDD harus ditentukan dan disepakati dengan pembimbing atau tim pengawas Tugas Akhir untuk memastikan keterkaitan yang jelas dengan proyek yang akan dilakukan.

## 1.3 Overview

Overview dari Dokumen Software Design Document (SDD) dalam Tugas Akhir Informatika Universitas XYZ adalah sebagai berikut:

1. Pendahuluan: Bagian ini memberikan gambaran umum tentang proyek Tugas Akhir, termasuk latar belakang, tujuan, dan ruang lingkup proyek. Ini juga menjelaskan struktur dan isi dari SDD.
2. Persyaratan: Bagian ini mencakup persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh perangkat lunak yang akan dibangun. Persyaratan ini diperoleh dari analisis kebutuhan dan menggambarkan apa yang diharapkan dari perangkat lunak.
3. Desain Arsitektur: Bagian ini menjelaskan arsitektur perangkat lunak yang akan dibangun. Ini termasuk pemilihan pola desain, komponen utama, dan hubungan antara komponen-komponen tersebut. Bagian ini memberikan gambaran tentang struktur sistem secara keseluruhan.
4. Desain Detail: Bagian ini memberikan rincian teknis tentang setiap komponen perangkat lunak. Ini mencakup kelas-kelas, fungsi-fungsi, algoritma-algoritma, dan desain antarmuka pengguna. Bagian ini menjelaskan bagaimana komponen-komponen tersebut akan diimplementasikan.
5. Pengujian: Bagian ini menjelaskan rencana pengujian yang akan dilakukan untuk memverifikasi dan memvalidasi perangkat lunak. Ini mencakup skenario pengujian, metode pengujian, dan lingkungan pengujian yang akan digunakan. Bagian ini juga mencakup rencana untuk mengatasi masalah yang terdeteksi selama pengujian.
6. Rencana Implementasi: Bagian ini memberikan jadwal dan rencana implementasi untuk proyek Tugas Akhir. Ini mencakup estimasi waktu, alokasi sumber daya, dan tugas-tugas yang harus diselesaikan untuk menyelesaikan pembangunan perangkat lunak. Rencana ini membantu dalam mengatur dan melacak kemajuan proyek.
7. Referensi: Bagian ini mencakup daftar referensi yang digunakan dalam penulisan SDD. Ini termasuk buku, artikel, dokumen teknis, atau sumber daya lain yang relevan dengan proyek.

SDD memberikan panduan rinci tentang perancangan dan implementasi perangkat lunak dalam Tugas Akhir. Dokumen ini membantu memastikan bahwa tim pengembang memiliki pemahaman yang jelas tentang perangkat lunak yang akan dibangun, memenuhi persyaratan yang ditetapkan, dan mengikuti desain yang telah ditetapkan. SDD juga berfungsi sebagai dokumen acuan yang komprehensif untuk proyek, membantu dalam kolaborasi tim, dan memfasilitasi pengujian dan implementasi yang efisien.

## 1.4 Reference

1. Modul Pratikum APPL dimana di dalamnya berisi pedoman dan panduan lengkap mengenai proses, persyaratan, dan prosedur yang terkait dengan Tugas Akhir di Program Studi Informatika Universitas XYZ.
2. Sumber Internet menyediakan akses ke berbagai sumber daya online, seperti situs web akademik, repositori institusi, database penelitian, blog, dan forum diskusi. Mahasiswa dapat menggunakan sumber-sumber ini untuk mencari informasi, memperoleh data, atau mempelajari pandangan dan penelitian terbaru dalam bidang yang mereka teliti.
3. Jurnal ilmiah, jurnal ilmiah yang terkait dengan topik penelitian mereka. Jurnal ilmiah menyediakan penelitian-penelitian sebelumnya yang telah dilakukan oleh ahli dalam bidang yang sama, memberikan dasar teoritis dan konteks yang diperlukan untuk penelitian mereka sendiri.

## 1.5 Definitions and Acronyms

1. SDD (Software Design Document)

Dokumen Desain Perangkat Lunak adalah dokumen yang menjelaskan rancangan dan rencana implementasi dari sebuah perangkat lunak. SDD mencakup detail arsitektur, komponen, antarmuka pengguna, dan desain teknis lainnya yang akan diimplementasikan dalam perangkat lunak.

1. PL (Perangkat Lunak)

Perangkat Lunak adalah program atau aplikasi komputer yang dirancang untuk menjalankan tugas atau memberikan fungsi-fungsi tertentu kepada pengguna. Ini mencakup program-program komputer, aplikasi web, aplikasi seluler, dan sistem informasi.

1. TA (Tugas Akhir)

Tugas Akhir adalah proyek penelitian atau pengembangan yang dilakukan oleh mahasiswa sebagai syarat untuk menyelesaikan pendidikan tingkat sarjana atau gelar akademik lainnya. Tugas Akhir sering kali melibatkan pembangunan perangkat lunak, penelitian ilmiah, atau proyek-proyek inovatif lainnya.

1. SKS (Status Kredit Semester)

Status Kredit Semester adalah jumlah kredit yang diperoleh oleh seorang mahasiswa untuk menyelesaikan program studi atau gelar akademik tertentu. Setiap mata kuliah yang selesai dengan sukses akan memberikan sejumlah kredit, dan SKS menggambarkan akumulasi total kredit yang diperoleh oleh seorang mahasiswa.

1. NIM (Nomor Induk Mahasiswa)

Nomor Induk Mahasiswa adalah nomor identifikasi unik yang diberikan kepada setiap mahasiswa di sebuah institusi pendidikan. NIM digunakan untuk mengidentifikasi dan melacak rekam jejak akademik dan administratif mahasiswa.

1. NIP (Nomor Induk Pegawai)

Nomor Induk Pegawai adalah nomor identifikasi unik yang diberikan kepada seorang pegawai di sebuah lembaga atau organisasi. NIP digunakan untuk mengidentifikasi dan melacak data pribadi, jabatan, dan rekam jejak kerja pegawai.

1. KBK (Ketua Bidang Keahlian)

Ketua Bidang Keahlian adalah posisi kepemimpinan akademik di sebuah perguruan tinggi atau institusi pendidikan yang mengawasi bidang keahlian atau jurusan tertentu. KBK bertanggung jawab atas pengembangan kurikulum, pengajaran, penelitian, dan pengabdian masyarakat dalam bidang keahliannya.

1. UI (User Interface)

Antarmuka Pengguna adalah elemen-elemen visual dan interaktif dari sebuah perangkat lunak atau sistem yang memungkinkan pengguna untuk berinteraksi dengan perangkat lunak tersebut. Ini mencakup elemen seperti tombol, formulir, ikon, menu, dan tata letak yang memungkinkan pengguna berinteraksi dengan perangkat lunak.

1. UX (User Experience)

Pengalaman Pengguna adalah keseluruhan pengalaman dan persepsi pengguna dalam menggunakan sebuah perangkat lunak atau sistem. UX mencakup aspek-aspek seperti kegunaan, kepuasan pengguna, efisiensi, dan kesan keseluruhan yang dirasakan oleh pengguna saat berinteraksi dengan perangkat lunak.

1. UML (Unified Modelling Language)

Unified Modeling Language adalah bahasa pemodelan visual yang digunakan untuk merancang, memvisualisasikan, dan mendokumentasikan perangkat lunak. UML menggunakan berbagai diagram, seperti diagram use case, diagram kelas, diagram aktivitas, dan diagram sekuensial, untuk menggambarkan berbagai aspek perangkat lunak.

# Bab II System Overview

System Overview (Gambaran Sistem) dalam Software Design Document (SDD) Tugas Akhir Informatika Universitas XYZ memberikan pemahaman umum tentang sistem yang akan dibangun. Ini mencakup deskripsi singkat tentang tujuan sistem, fungsionalitas utama, dan interaksi dengan pengguna atau sistem lainnya. Berikut adalah beberapa komponen yang mungkin ada dalam System Overview:

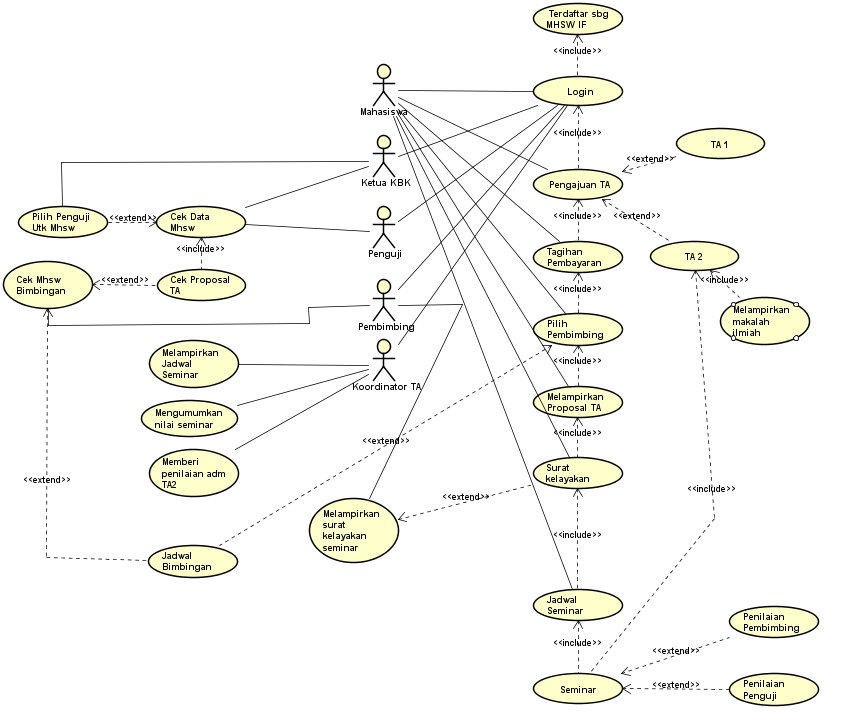
1. Deskripsi Sistem: Menjelaskan secara umum tentang sistem yang akan dibangun. Ini mencakup tujuan sistem, lingkup fungsionalitas, dan manfaat yang diharapkan.
2. Pengguna Sistem: Menjelaskan siapa yang akan menggunakan sistem dan peran atau hak akses yang mereka miliki. Ini dapat mencakup pengguna akhir, pengguna administratif, atau pengguna dengan peran khusus.
3. Fungsionalitas Utama: Menjelaskan fungsi-fungsi utama yang akan ditawarkan oleh sistem. Ini mencakup kegiatan atau fitur yang dapat dilakukan oleh pengguna, seperti membuat, membaca, memperbarui, atau menghapus data, atau melakukan tugas khusus lainnya.
4. Interaksi Sistem: Menjelaskan bagaimana sistem akan berinteraksi dengan pengguna atau sistem lainnya. Misalnya, apakah sistem akan memiliki antarmuka pengguna grafis, antarmuka pemrograman aplikasi (API), atau integrasi dengan sistem lain melalui protokol komunikasi tertentu.
5. Arsitektur Sistem: Memberikan gambaran umum tentang arsitektur sistem yang akan digunakan. Ini dapat mencakup pola desain yang akan diterapkan, seperti arsitektur berbasis lapisan (layered architecture), arsitektur mikrojasa (microservices architecture), atau arsitektur berorientasi layanan (service-oriented architecture).
6. Teknologi dan Platform: Menyebutkan teknologi dan platform yang akan digunakan dalam pengembangan sistem. Ini dapat mencakup bahasa pemrograman, framework, basis data, infrastruktur jaringan, atau alat pengembangan yang relevan.
7. Batasan Sistem: Menjelaskan batasan-batasan yang ada dalam sistem. Ini mencakup keterbatasan dalam hal fungsionalitas, skala, kinerja, keamanan, atau kompatibilitas dengan sistem yang ada.

8. Integrasi dengan Sistem Lain: Jika sistem perlu berintegrasi dengan sistem lain, System Overview dapat menyajikan gambaran umum tentang integrasi tersebut. Ini mencakup sistem yang terlibat, protokol komunikasi yang digunakan, dan jenis data yang akan ditukar.

System Overview dalam SDD membantu para pembaca untuk mendapatkan pemahaman awal tentang sistem yang akan dibangun. Ini memberikan gambaran tentang tujuan, fungsionalitas utama, interaksi, dan batasan sistem. Hal ini juga membantu tim pengembang dan pihak terkait dalam mengarahkan dan memahami proyek secara keseluruhan.

# Bab III Application Design

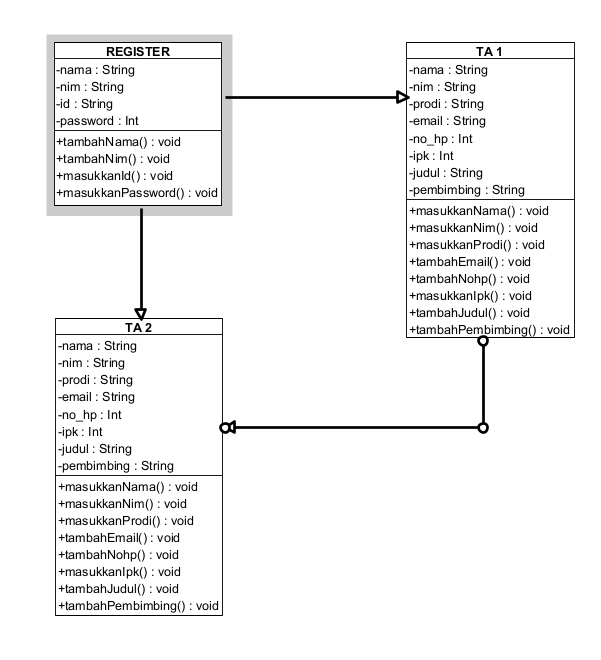
## 3.2 Use Case Diagram



## 3.3 Use Case Scenario

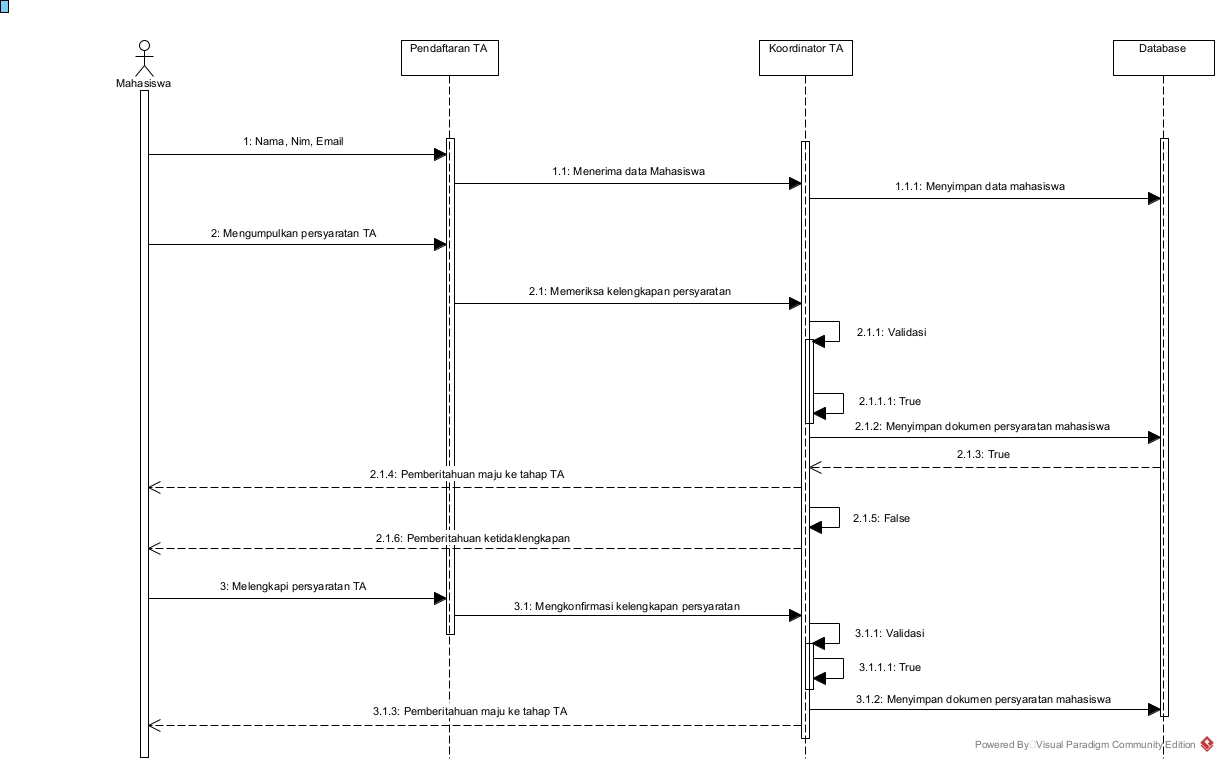
|  |  |
| --- | --- |
| AKSI MAHASISWA | REAKSI SYSTEM |
| 1. Melakukan login |  |
|  | 1. Menampilkan halaman login |
| 1. Memilih menu pengajuan |  |
|  | 1. Menampilkan halaman pengajuan |
| 1. Memilih TA |  |
|  | 1. Menampilkan pilihan TA1 atau TA2 |
| 1. Membayar tagihan pembayaran TA |  |
|  | 1. Menampilkan tagihan TA |
| 1. Memilih dosen pembimbing |  |
|  | 1. Menampilkan daftar dosen pembimbing |
| 1. Melakukan bimbingan |  |
|  | 1. Menampilkan jadwal bimbingan |
| 1. Melampirkan proposal TA |  |
|  | 1. Menampilkan tampilan untuk mengupload file proposal |
| 1. Menunggu surat kelayakan untuk seminar |  |
|  | 1. Pembimbing akan melampirkan surat kelayakan seminar |
| 1. Seminar |  |
|  | 1. Koordinator TA melampirkan jadwal seminar |
| 1. Menunggu hasil penilaian |  |
|  | 1. Pembimbing dan Penguji akan mengupload nilai |

## Class Diagram

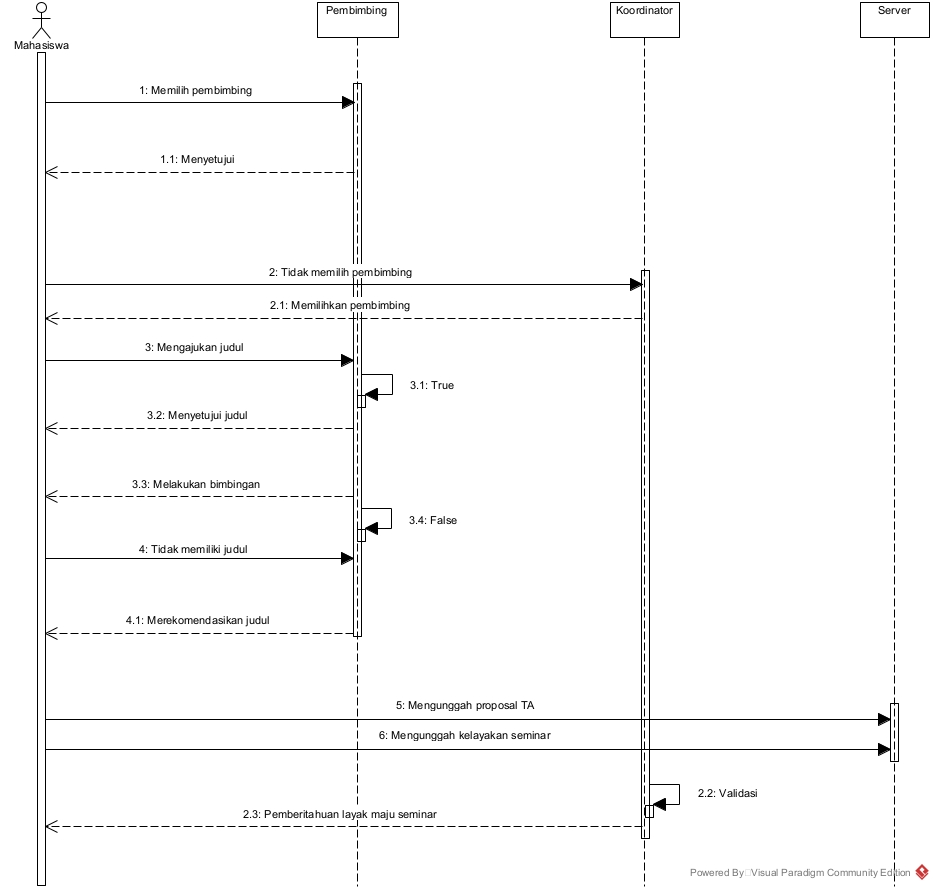


## 3.3 Sequence Diagran

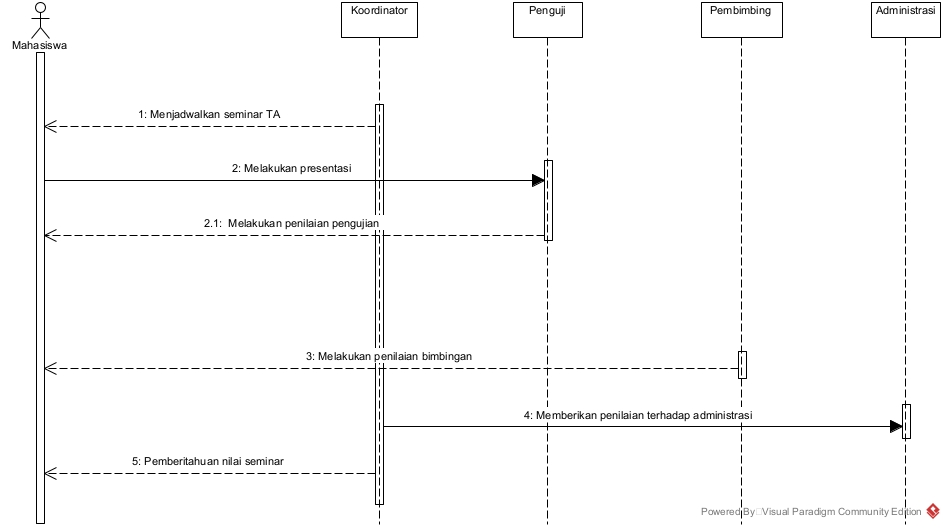
1. Pendaftaran Tugas Akhir



1. Pelaksanaan Tugas Akhir

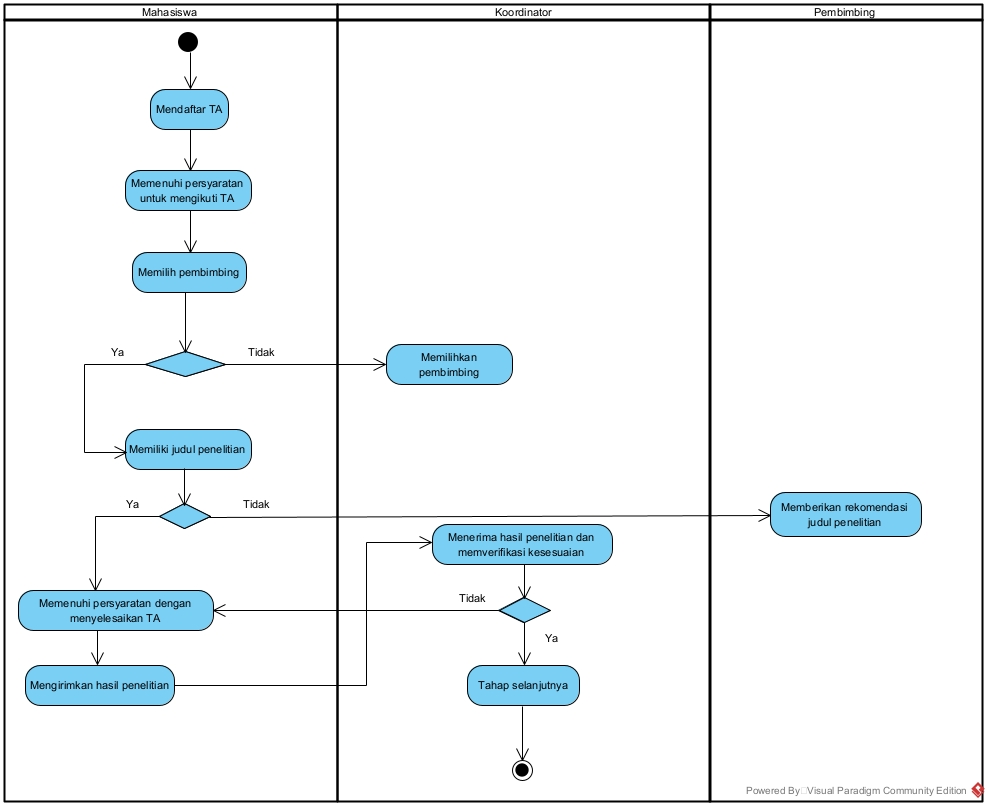


1. Seminar Tugas Akhir

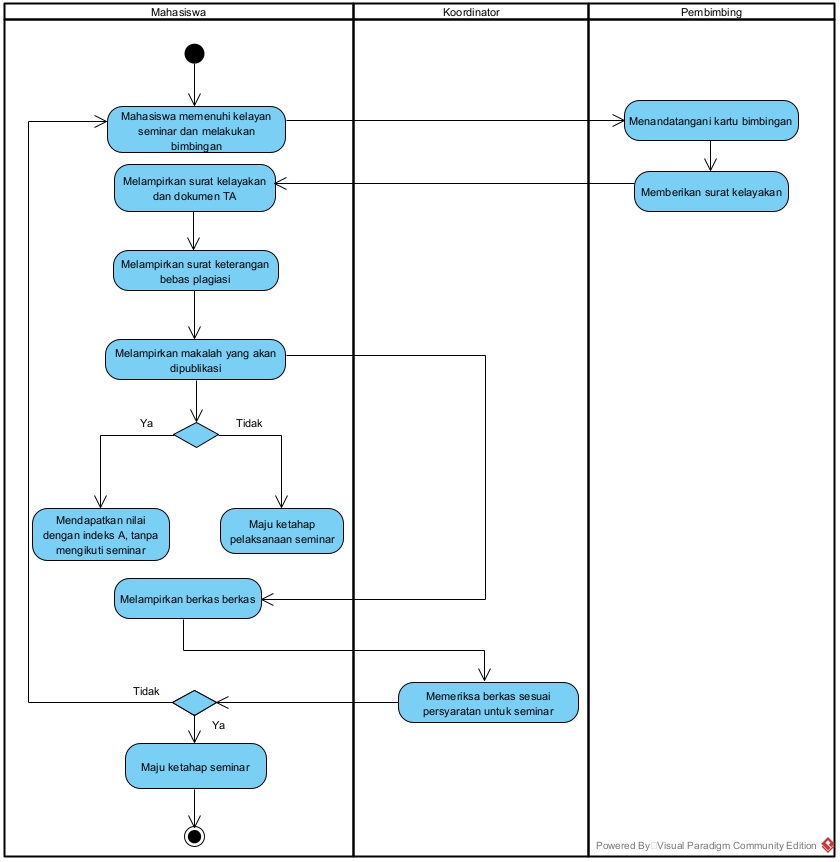


## 3.3 Activity Diagram

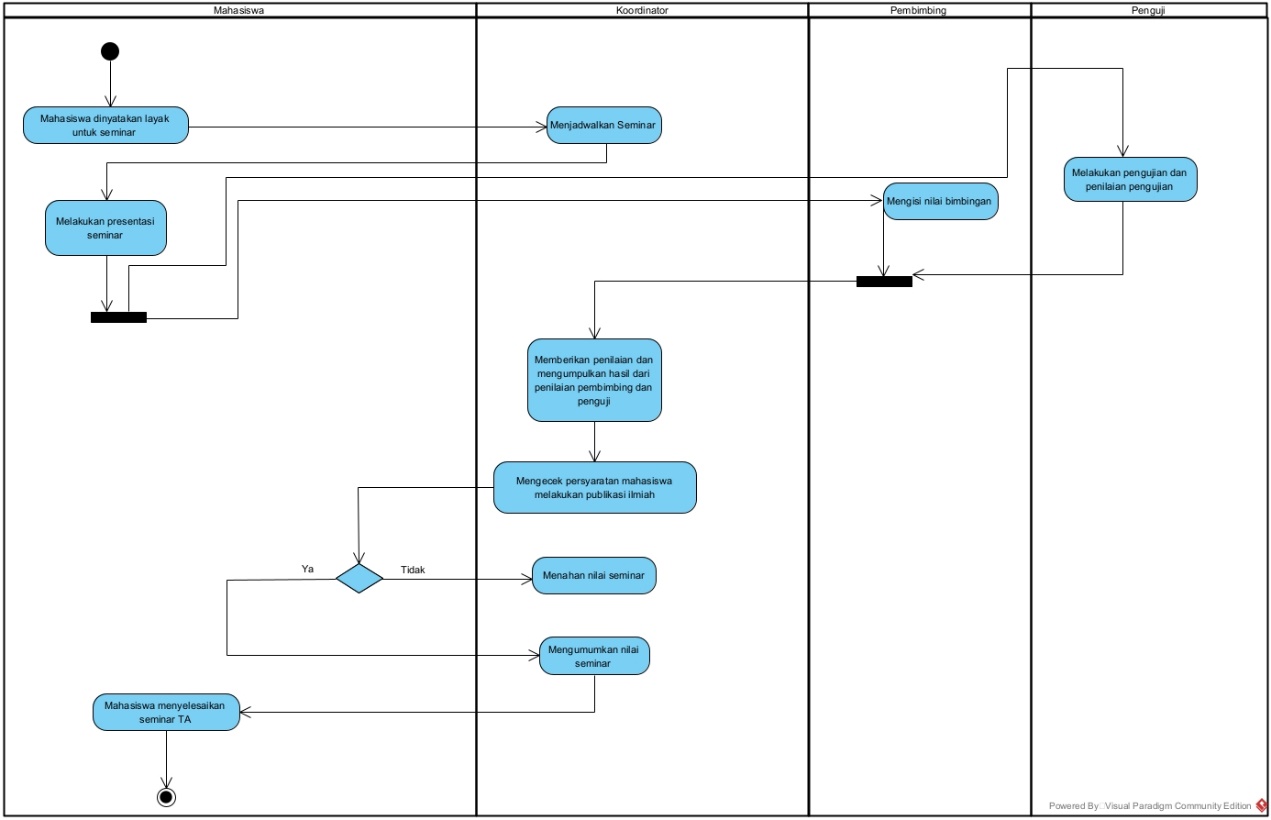
1. Daftar TA



1. Kelayakan Seminar

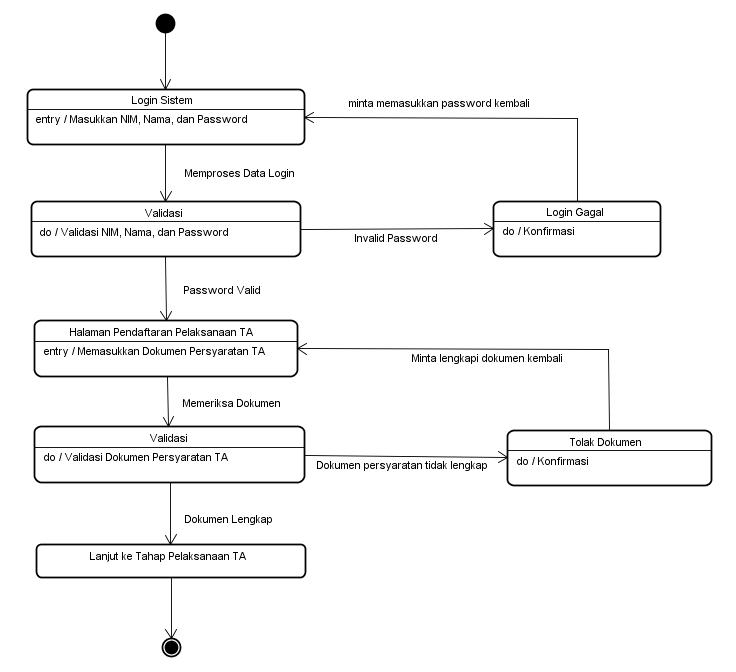


1. Seminar

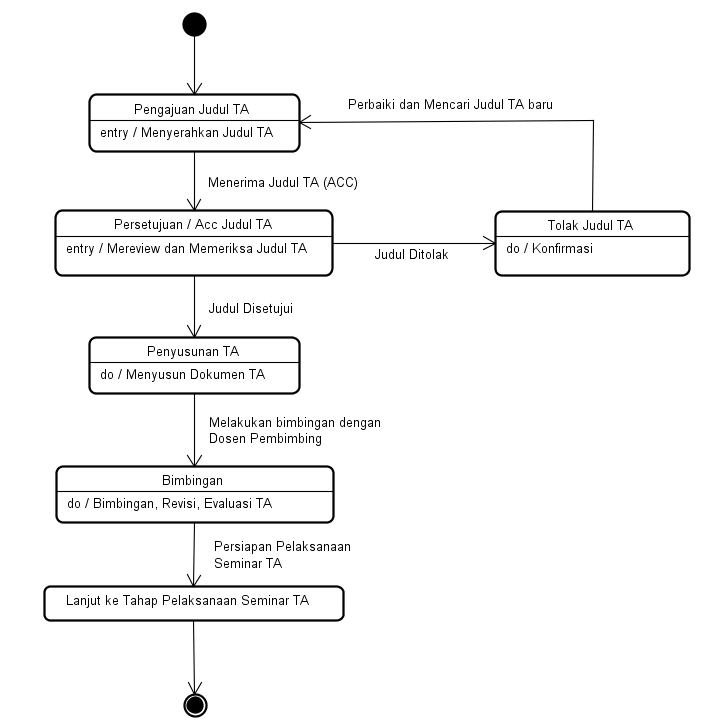


## 3.3 State Diagram

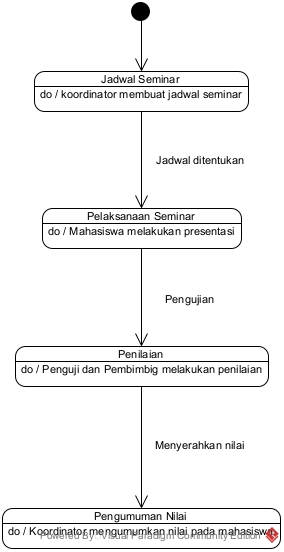
1. Pendaftaran TA



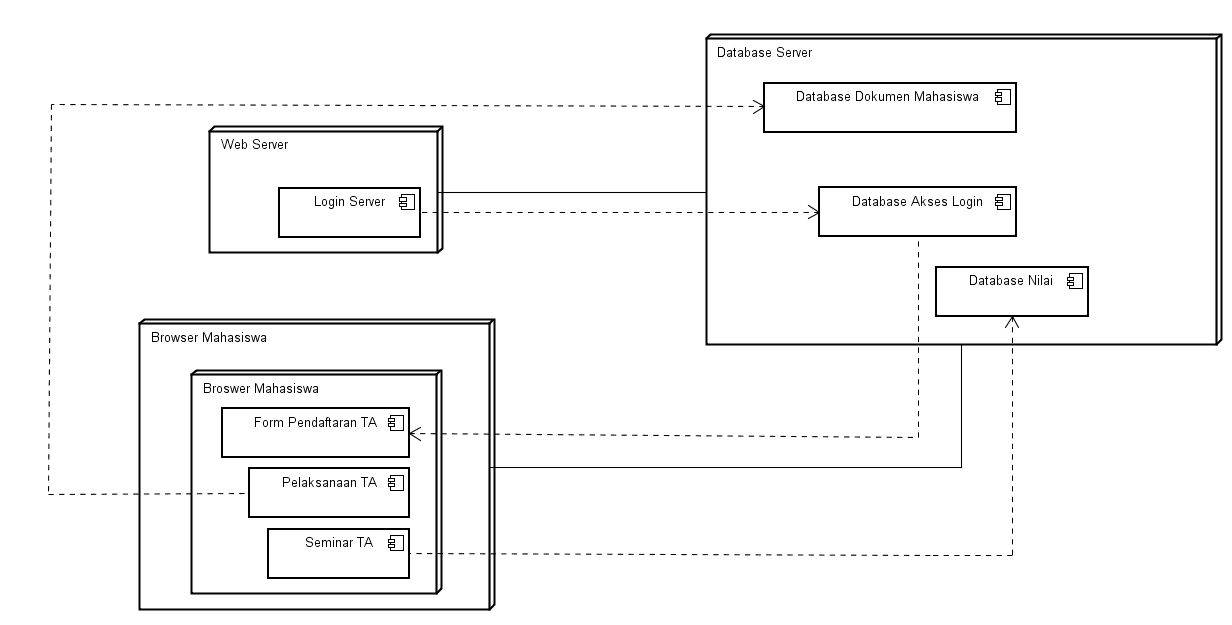
1. Pelaksanaan TA



1. Pelaksanaan Seminar TA



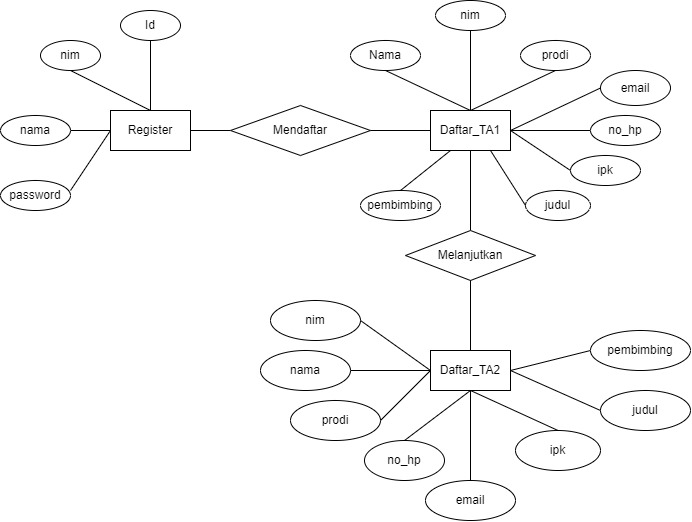
## 3.3 Deployment Diagram



# Bab IV Data Design

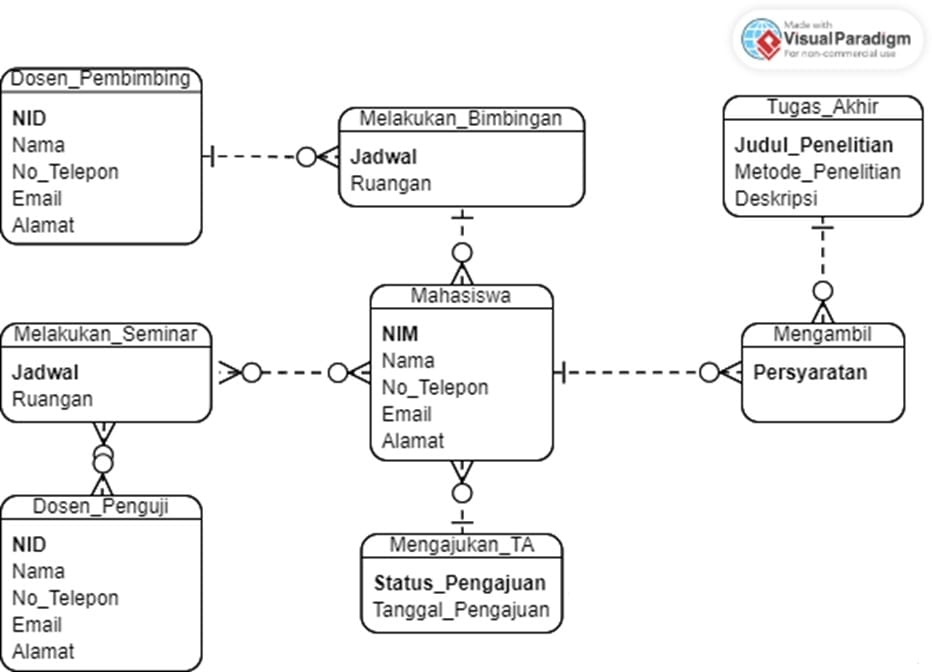
## 4.1 Logical Design

Logical Design adalah langkah untuk merencanakan dan merancang struktur logis sistem atau perangkat lunak yang akan dibangun. Ini mencakup identifikasi entitas, atribut, dan hubungan yang diperlukan untuk menggambarkan logika sistem secara menyeluruh. Logical Design membantu dalam memahami dan merencanakan komponen dan fungsionalitas yang akan dibangun dalam Tugas Akhir.



## 4.2 Physical Design

Physical Design melibatkan pemilihan dan spesifikasi perangkat keras dan perangkat lunak yang akan digunakan untuk membangun sistem. Ini mencakup pemilihan bahasa pemrograman, framework, database, server, dan perangkat keras yang sesuai dengan kebutuhan dan persyaratan proyek. Physical Design juga mencakup perencanaan infrastruktur dan lingkungan implementasi yang akan digunakan.



1.Entitas

1. Entitas Mahasiswa

Atribut: NIM (Primary Key), Nama, No\_Telepon, Email, Alamat

1. Entitas Tugas\_Akhir

Atribut: Judul\_Penelitian (Primary Key), Metode\_Penelitian, Deskrips

1. Entitas Dosen\_Pembimbing

Atribut: NID (Primary Key), Nama, No\_Telepon, Email, Alamat

1. Entitas Dosen\_Penguji

Atribut: NID (Primary Key), Nama, No\_Telepon, Email, Alamat

2. Relasi

1. Mahasiswa mengambil Tugas\_Akhir
2. Tugas\_Akhir dibimbing Dosen\_Pembimbing
3. Dosen\_Penguji menguji Tugas\_Akhir(Seminar)

# Bab V User Interface Design

User Interface (UI) Design dalam Tugas Akhir mahasiswa melibatkan perancangan antarmuka pengguna yang intuitif, menarik, dan mudah digunakan. Berikut adalah beberapa aspek yang perlu dipertimbangkan dalam User Interface Design Tugas Akhir:

1. Analisis Pengguna: Lakukan analisis pengguna untuk memahami profil dan kebutuhan pengguna potensial. Identifikasi karakteristik pengguna, tingkat pengalaman, preferensi, dan tugas-tugas yang ingin mereka lakukan melalui antarmuka.
2. Tujuan Antarmuka: Tetapkan tujuan utama dari antarmuka pengguna, apakah itu untuk mempermudah pencarian informasi, melakukan tugas spesifik, atau menyediakan interaksi yang intuitif.
3. Tata Letak dan Organisasi: Rancang tata letak yang logis dan intuitif untuk menempatkan elemen-elemen antarmuka dengan baik. Pemilihan struktur navigasi, menu, dan pengelompokan konten harus memudahkan pengguna untuk menavigasi dan menemukan informasi dengan cepat.
4. Komponen Antarmuka: Pilih dan desain komponen antarmuka seperti tombol, formulir, tabel, ikon, dan elemen lainnya yang diperlukan untuk interaksi pengguna. Pastikan desain tersebut konsisten dengan kebutuhan proyek dan mengikuti prinsip-prinsip desain yang baik.
5. Konsistensi Visual: Buat tampilan visual yang konsisten dengan menggunakan elemen desain yang seragam, termasuk warna, jenis huruf, ukuran, dan gaya. Ini membantu pengguna mengenali pola dan memahami bagaimana berbagai elemen berinteraksi.
6. Responsif dan Mobile-Friendly: Jika perlu, pastikan antarmuka pengguna dirancang untuk responsif, sehingga dapat diakses dan digunakan secara optimal di berbagai perangkat, termasuk desktop, tablet, dan ponsel.
7. Visualisasi Data: Jika sistem Anda melibatkan pemrosesan dan visualisasi data, pertimbangkan cara terbaik untuk menggambarkan data secara efektif melalui grafik, diagram, atau representasi visual lainnya.
8. Uji Pengguna: Melakukan uji coba pengguna untuk mendapatkan umpan balik tentang kemudahan penggunaan dan keefektifan antarmuka pengguna. Gunakan hasil tes pengguna untuk memperbaiki desain dan mengoptimalkan pengalaman pengguna.
9. Dokumentasi: Dokumentasikan desain antarmuka pengguna secara rinci dalam SDD untuk memberikan panduan kepada pengembang selanjutnya dan memastikan konsistensi implementasi.

# Bab VI Interface Requirements

## 4.1 User Interface

Antarmuka Pengguna atau User Interface adalah elemen yang memungkinkan interaksi antara pengguna dengan perangkat lunak Tugas Akhir. Antarmuka pengguna haruslah mudah digunakan, intuitif, dan menyediakan semua fitur yang diperlukan oleh pengguna. Berikut adalah beberapa karakteristik dari setiap interface antara PL dengan pengguna.

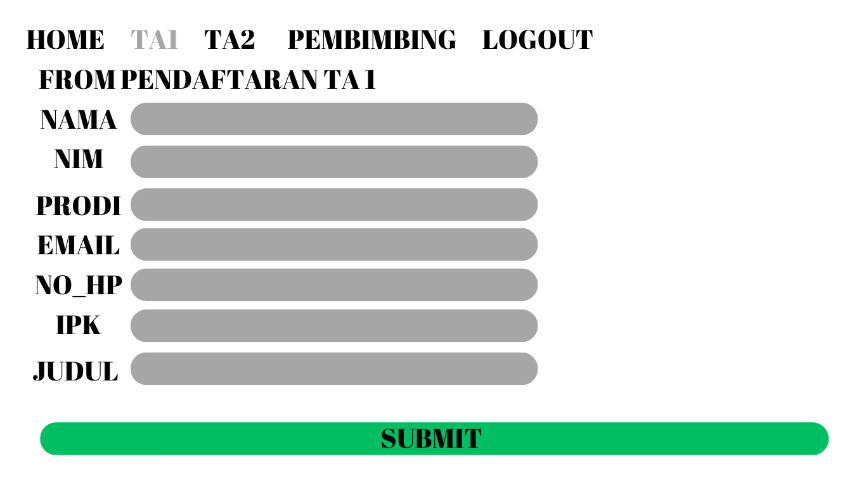
1. Halaman Login



1. Register

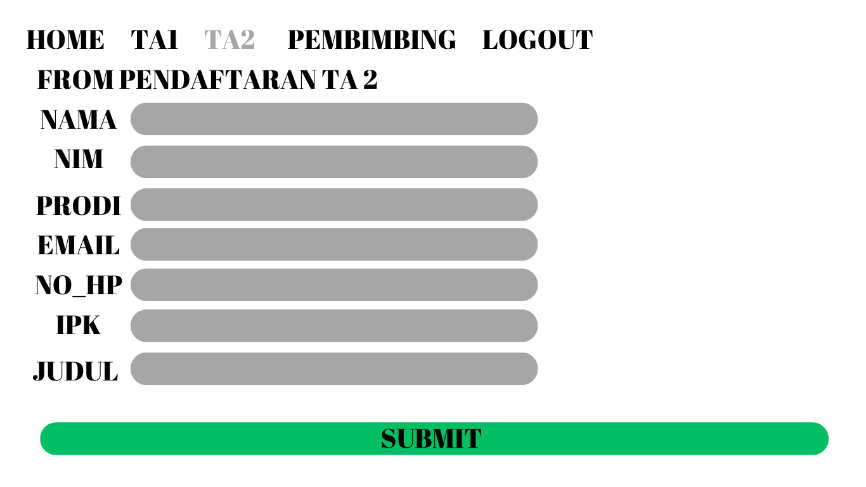


1. Formulir TA 1

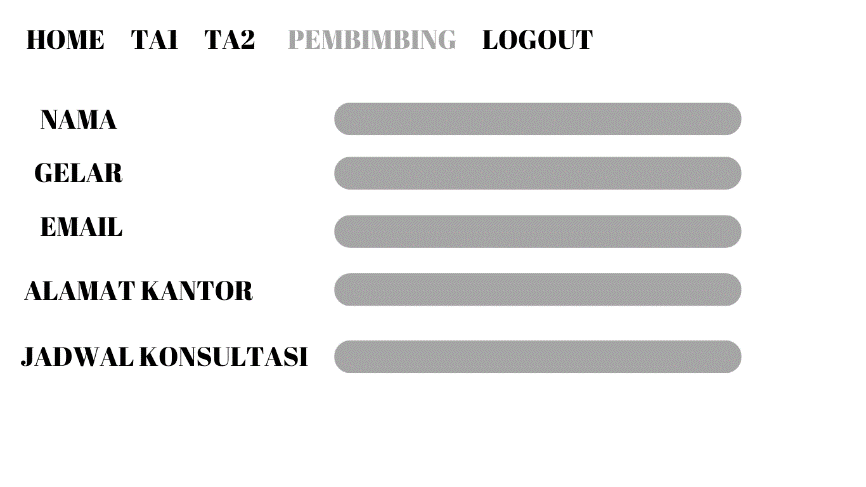


.

1. TA 2

.

1. Pembimbing



## 4.2 Hardware Interface

Hardware Interface (Antarmuka Perangkat Keras) dalam Tugas Akhir mahasiswa merujuk pada koneksi dan interaksi antara perangkat lunak yang dikembangkan dengan perangkat keras yang terhubung. Berikut adalah beberapa contoh Hardware Interface yang mungkin diperlukan dalam Tugas Akhir:

1. Komputer/Laptop: Antarmuka ini memungkinkan sistem Tugas Akhir untuk berinteraksi dengan komputer atau laptop yang digunakan oleh pengguna (mahasiswa) untuk mengakses sistem. Sistem harus kompatibel dengan berbagai tipe dan model komputer/laptop yang umum digunakan.
2. Sensor atau Perangkat Input: Jika perangkat lunak yang dikembangkan membutuhkan input dari sensor fisik, seperti sensor suhu, sensor gerakan, atau perangkat penunjuk (misalnya, mouse), maka perlu ditentukan antarmuka dan metode untuk menerima dan memproses input dari perangkat tersebut.
3. Perangkat Output: Jika perangkat lunak yang dikembangkan harus berinteraksi dengan perangkat output fisik, seperti layar, printer, atau speaker, maka perlu ditentukan antarmuka dan metode untuk mengirimkan data dan instruksi ke perangkat output tersebut.
4. Jaringan dan Komunikasi: Jika perangkat lunak yang dikembangkan perlu berkomunikasi dengan perangkat lain melalui jaringan, seperti protokol jaringan atau komunikasi nirkabel, maka perlu ditentukan antarmuka dan protokol komunikasi yang sesuai.
5. Perangkat Keras Tertentu: Jika perangkat lunak yang dikembangkan harus berinteraksi dengan perangkat keras spesifik, seperti mikrokontroler, perangkat FPGA (Field-Programmable Gate Array), atau perangkat I/O khusus, maka perlu ditentukan antarmuka dan protokol komunikasi yang diperlukan untuk berinteraksi dengan perangkat tersebut.
6. Perangkat Penyimpanan: Jika perangkat lunak yang dikembangkan harus berinteraksi dengan perangkat penyimpanan fisik, seperti hard drive, flash drive, atau perangkat cloud storage, maka perlu ditentukan antarmuka dan metode akses yang sesuai untuk membaca dan menulis data ke perangkat penyimpanan tersebut.
7. Aksesibilitas Perangkat Keras: Jika perangkat lunak yang dikembangkan harus berfungsi dengan perangkat keras tertentu, seperti tipe perangkat atau spesifikasi yang spesifik, maka perlu ditentukan persyaratan kompatibilitas yang jelas untuk memastikan bahwa perangkat lunak dapat berjalan dengan baik di perangkat keras yang ditentukan.
8. Dokumentasi: Dokumentasikan antarmuka perangkat keras dengan jelas dalam SDD dan sertakan spesifikasi teknis, koneksi fisik, dan protokol komunikasi yang relevan untuk memudahkan pengembang selanjutnya dalam mengimplementasikan perangkat lunak dengan perangkat keras yang sesuai.

## 4.3 Software Interface

Software Interface (Antarmuka Perangkat Lunak) dalam Tugas Akhir mahasiswa merujuk pada antarmuka yang digunakan oleh perangkat lunak yang dikembangkan untuk berinteraksi dengan komponen lain dalam sistem atau dengan perangkat lunak lainnya. Berikut adalah beberapa contoh Software Interface yang mungkin diperlukan dalam Tugas Akhir:

1. Antarmuka Pengguna (User Interface): Ini mencakup elemen-elemen antarmuka yang digunakan oleh pengguna untuk berinteraksi dengan perangkat lunak, seperti tampilan grafis, tombol, formulir, dan menu. Desain dan implementasi antarmuka pengguna harus mempertimbangkan kemudahan penggunaan, responsivitas, dan tampilan yang menarik.
2. Antarmuka Pemrograman Aplikasi (API): Jika perangkat lunak yang dikembangkan akan digunakan oleh aplikasi atau sistem lain sebagai komponen terintegrasi, maka API perlu ditentukan. API menyediakan metode, fungsi, atau protokol komunikasi yang digunakan untuk berinteraksi dengan perangkat lunak melalui panggilan pemrograman.
3. Antarmuka Basis Data: Jika perangkat lunak yang dikembangkan berkomunikasi dengan basis data, antarmuka basis data harus ditentukan. Ini mencakup koneksi ke basis data, metode akses data, permintaan query, dan manipulasi data.
4. Antarmuka Jaringan: Jika perangkat lunak yang dikembangkan berinteraksi dengan sistem lain melalui jaringan, antarmuka jaringan harus ditentukan. Ini mencakup protokol komunikasi, metode pengiriman data, format pertukaran data, dan metode autentikasi.
5. Antarmuka Sistem Operasi: Jika perangkat lunak yang dikembangkan berjalan di atas sistem operasi tertentu, antarmuka sistem operasi harus dipertimbangkan. Ini mencakup metode akses sistem operasi, manajemen sumber daya, dan interaksi dengan komponen sistem operasi, seperti sistem file, layanan jaringan, atau perangkat keras.
6. Antarmuka Layanan Web: Jika perangkat lunak yang dikembangkan menyediakan layanan web yang dapat diakses melalui protokol web (seperti HTTP), antarmuka layanan web harus ditentukan. Ini mencakup endpoint, metode permintaan (GET, POST, PUT, DELETE), format respons, dan protokol keamanan (seperti HTTPS).
7. Antarmuka dengan Perangkat Keras Eksternal: Jika perangkat lunak yang dikembangkan berinteraksi dengan perangkat keras eksternal, seperti perangkat periferal atau sensor, antarmuka perangkat keras eksternal harus ditentukan. Ini mencakup protokol komunikasi, format data, dan metode pengontrolan perangkat keras.
8. Antarmuka Integrasi dengan Sistem Lain: Jika perangkat lunak yang dikembangkan harus berintegrasi dengan sistem lain, antarmuka integrasi harus ditentukan. Ini mencakup format pertukaran data, protokol komunikasi, metode sinkronisasi, dan interaksi dengan sistem lain melalui API atau protokol kustom.

## 4.4 Communication Interface

Communication Interface (Antarmuka Komunikasi) dalam Tugas Akhir mahasiswa merujuk pada metode atau protokol yang digunakan untuk komunikasi antara komponen sistem, perangkat keras, atau perangkat lunak yang terlibat dalam proyek. Berikut adalah beberapa contoh Communication Interface yang mungkin diperlukan dalam Tugas Akhir:

1. Protokol Komunikasi: Menentukan protokol yang akan digunakan untuk komunikasi antara komponen atau sistem yang berbeda. Ini bisa termasuk protokol jaringan seperti TCP/IP, HTTP, atau MQTT, atau protokol kustom yang dikembangkan sesuai kebutuhan proyek.
2. Format Pertukaran Data: Menentukan format data yang akan digunakan dalam pertukaran informasi antara komponen atau sistem. Ini dapat mencakup format seperti JSON (JavaScript Object Notation), XML (eXtensible Markup Language), atau protokol biner kustom.
3. Metode Pengiriman Pesan: Menentukan metode atau teknologi yang digunakan untuk mengirim dan menerima pesan antara komponen atau sistem. Ini bisa termasuk pengiriman pesan melalui protokol HTTP, pesan publik-subscribe melalui MQTT, atau penggunaan antrian pesan.
4. Antarmuka Pemrograman Aplikasi (API): Jika proyek melibatkan integrasi dengan sistem atau layanan eksternal, persyaratan antarmuka pemrograman aplikasi (API) harus ditentukan. Ini mencakup endpoint, metode permintaan (GET, POST, PUT, DELETE), parameter, dan format respons yang digunakan dalam komunikasi dengan sistem atau layanan eksternal.
5. Antarmuka Sensor atau Perangkat Keras Eksternal: Jika proyek melibatkan interaksi dengan sensor atau perangkat keras eksternal, antarmuka komunikasi dengan perangkat tersebut harus ditentukan. Ini dapat meliputi protokol komunikasi seperti I2C, SPI, UART, atau komunikasi nirkabel seperti Bluetooth atau Zigbee.
6. Pengamanan Komunikasi: Jika komunikasi perlu aman dan terenkripsi, persyaratan pengamanan komunikasi harus ditentukan. Ini melibatkan penggunaan protokol seperti SSL/TLS untuk enkripsi, atau metode autentikasi seperti OAuth atau JWT (JSON Web Tokens) untuk mengamankan komunikasi.
7. Kecepatan dan Latensi: Jika proyek membutuhkan komunikasi real-time atau berkinerja tinggi, persyaratan terkait kecepatan dan latensi harus ditentukan. Ini mencakup persyaratan terkait kecepatan transfer data, waktu respons, atau batasan latensi dalam komunikasi.
8. Skema Komunikasi dan Struktur Pesan: Menentukan skema atau struktur komunikasi yang akan digunakan untuk mengatur pertukaran pesan antara komponen atau sistem. Ini mencakup format dan tata letak pesan yang akan dikirim dan diterima.