Software Requirements specification (SRS)

FOR <<Project Name>>.

…

# Table Of Contents

[Table Of Contents 1](#_Toc152871024)

[Bab I Introduction 4](#_Toc152871025)

[1.1 Purpose 4](#_Toc152871026)

[1.2 Overview 4](#_Toc152871027)

[1.3 Project Scope 5](#_Toc152871028)

[1.4 References 5](#_Toc152871029)

[1.5 Definitions and Acronyms 5](#_Toc152871030)

[Bab II System Overview 0](#_Toc152871031)

[Bab III Overall Description 2](#_Toc152871032)

[3.1 Organitations 2](#_Toc152871033)

[3.2 Product Perspective 2](#_Toc152871034)

[3.3 User Classes and Characteristics 2](#_Toc152871035)

[3.4 Operating Environment 3](#_Toc152871036)

[3.5 Design and Implementation Constrains (optional) 3](#_Toc152871037)

[3.6 Assumptions and Dependencies (optional) 4](#_Toc152871038)

[Bab IV Functional Requirements 5](#_Toc152871039)

[4.1 Detailed Functional Requirements 5](#_Toc152871040)

[4.2 Use Case Diagram 7](#_Toc152871041)

[4.3 Use Case Scenario 7](#_Toc152871042)

[4.4 Class Diagram 8](#_Toc152871043)

[4.5 Sequence Diagram 8](#_Toc152871045)

[4.6 Activity Diagram 11](#_Toc152871046)

[4.7 State Diagram 14](#_Toc152871047)

[4.8 Deployment Diagram 18](#_Toc152871048)

[Bab V Non Functional Requirements 19](#_Toc152871049)

[5.1 Performance Requirements (optional) 20](#_Toc152871050)

[5.2 Safety Requirements (optional) 20](#_Toc152871051)

[5.3 Software Quality Attributes (optional) 21](#_Toc152871052)

[Bab VI Data Design 22](#_Toc152871053)

[6.1 Logical Design 22](#_Toc152871054)

[6.2 Physical Design 23](#_Toc152871055)

[Bab VII Data Requirements 24](#_Toc152871056)

[7.1 Input 24](#_Toc152871057)

[7.2 Output 24](#_Toc152871058)

[Bab VIII User Interface Design 26](#_Toc152871059)

[Bab IX Interface Requirements 30](#_Toc152871060)

[9.1 User Interface 30](#_Toc152871061)

[9.2 Hardware Interface 30](#_Toc152871062)

[9.3 Software Interface 31](#_Toc152871063)

[9.4 Communication Interface 31](#_Toc152871064)

# Bab I Introduction

## 1.1 Purpose

Tujuan dari dokumen Spesifikasi Persyaratan Perangkat Lunak (SRS) ini adalah untuk memberikan deskripsi detail tentang persyaratan untuk Divisi Keuangan di dalam organisasi. Dokumen ini menjelaskan persyaratan fungsional dan non-fungsional, batasan sistem, dan antarmuka yang perlu dipertimbangkan selama pengembangan dan implementasi sistem perangkat lunak keuangan.

## 1.2 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.

## 1.3 Project Scope

Lingkup proyek untuk sistem perangkat lunak keuangan dalam Divisi Keuangan organisasi meliputi hal-hal berikut:

1. Pencatatan otomatis dan pengelolaan transaksi keuangan, termasuk utang usaha, piutang usaha, dan entri buku besar.
2. Pembuatan laporan keuangan, seperti neraca, laporan laba rugi, dan laporan arus kas.
3. Integrasi dengan departemen dan sistem lainnya, seperti Divisi Penjualan, Manajemen Inventaris, dan Sumber Daya Manusia.
4. Penyimpanan yang aman dan kontrol akses untuk data keuangan.
5. Kepatuhan dengan peraturan keuangan dan standar yang relevan.

## 1.4 References

Pengembangan dokumen SRS ini mungkin merujuk pada sumber daya berikut:

1. Prosedur dan dokumentasi Divisi Keuangan yang sudah ada.
2. Standar industri dan praktik terbaik dalam manajemen keuangan.
3. Persyaratan peraturan terkait pelaporan keuangan dan keamanan data.
4. Konsultasi dengan personel Divisi Keuangan, pengembang perangkat lunak, dan pemangku kepentingan terkait.

## 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 Overall Description

## 3.1 Organitations

Organisasi usulan adalah entitas yang sedang direncanakan atau diusulkan untuk dibentuk. Dalam konteks ini, beberapa elemen yang perlu dipertimbangkan adalah visi, misi, dan struktur organisasi.

1. Visi: Visi organisasi adalah gambaran jangka panjang yang diinginkan oleh organisasi. Ini mencerminkan tujuan utama atau keadaan yang ingin dicapai dalam jangka waktu yang lebih luas. Visi memberikan arah dan fokus kepada organisasi.
2. Misi: Misi organisasi adalah pernyataan yang menjelaskan tujuan inti dan alasan eksistensi organisasi. Ini menggambarkan peran organisasi dalam masyarakat atau industri tertentu dan mengidentifikasi nilai-nilai yang ingin ditekankan dalam menjalankan kegiatan.
3. Struktur Organisasi: Struktur organisasi merujuk pada susunan hierarki dan hubungan antara unit atau departemen dalam organisasi. Ini mencakup pembagian tugas, wewenang, dan tanggung jawab, serta aliran komunikasi dan pengambilan keputusan di dalam organisasi.

## 3.2 Product Perspective

Perspektif produk melibatkan penjelasan tentang produk yang akan dikembangkan, termasuk deskripsi dan manfaatnya dari sudut pandang organisasi.

1. Deskripsi Produk: Deskripsi produk mencakup penjelasan rinci tentang karakteristik, fungsi, dan fitur produk yang akan dibangun. Ini mencakup informasi tentang tujuan produk dan bagaimana produk tersebut akan memenuhi kebutuhan pengguna atau pasar.
2. Manfaat Produk: Manfaat produk menjelaskan dampak positif yang akan dirasakan oleh organisasi dengan dibangunnya produk tersebut. Ini mungkin meliputi peningkatan efisiensi, peningkatan kualitas produk atau layanan, pengurangan biaya, atau peningkatan kepuasan pelanggan.

## 3.3 User Classes and Characteristics

Pada tahap ini, perlu dijelaskan pengguna yang akan terlibat dalam penggunaan produk, termasuk karakteristik mereka, hak akses, dan sejenisnya.

1. Kelas Pengguna: Kelas pengguna mengidentifikasi kelompok pengguna yang akan menggunakan produk. Kelompok ini dapat mencakup pengguna internal (misalnya karyawan organisasi) dan pengguna eksternal (misalnya pelanggan atau mitra bisnis).
2. Karakteristik Pengguna: Karakteristik pengguna mencakup informasi tentang latar belakang, pengetahuan, dan keahlian pengguna dalam menggunakan produk. Ini membantu dalam merancang antarmuka atau fitur yang sesuai dengan kebutuhan dan kemampuan pengguna.
3. Hak Akses: Hak akses mengacu pada tingkat akses atau izin yang dimiliki oleh setiap kelas pengguna terhadap fitur dan fungsi produk. Ini dapat mencakup tingkat akses berbeda antara pengguna internal dan eksternal, serta perbedaan akses antara peran atau posisi dalam organisasi.

## 3.4 Operating Environment

Lingkungan operasional merujuk pada lingkungan fisik dan teknologi di mana produk akan dioperasikan. Ini mencakup platform perangkat keras, sistem operasi, dan komponen perangkat lunak lain yang mendukung penggunaan produk.

1. Platform Perangkat Keras: Platform perangkat keras merujuk pada jenis dan spesifikasi perangkat keras yang diperlukan untuk menjalankan produk. Misalnya, apakah produk ini akan dijalankan pada komputer desktop, server, atau perangkat mobile.
2. Sistem Operasi dan Versi: Sistem operasi dan versi yang mendukung produk perlu dijelaskan. Ini penting karena aplikasi perangkat lunak dapat memiliki persyaratan khusus terkait sistem operasi yang kompatibel.
3. Komponen Perangkat Lunak Lainnya: Jika ada komponen perangkat lunak tambahan yang diperlukan untuk mendukung penggunaan produk, seperti basis data atau perangkat lunak pihak ketiga, perlu dijelaskan juga.

## 3.5 Design and Implementation Constrains (optional)

Batasan desain dan implementasi mengacu pada kendala atau pembatasan yang harus diperhatikan selama proses pengembangan produk. Contohnya dapat mencakup:

1. Keterbatasan Teknis: Batasan terkait dengan keterbatasan teknologi yang ada, seperti keterbatasan perangkat keras atau kebutuhan integrasi dengan sistem yang sudah ada.
2. Keterbatasan Waktu dan Sumber Daya: Batasan terkait dengan batasan waktu yang tersedia untuk pengembangan produk dan ketersediaan sumber daya, seperti tenaga kerja, anggaran, dan infrastruktur.
3. Kebijakan atau Hukum: Batasan yang mungkin ada akibat kebijakan atau peraturan tertentu yang harus dipatuhi dalam pengembangan dan penggunaan produk.

## 3.6 Assumptions and Dependencies (optional)

Asumsi dan ketergantungan mengacu pada faktor-faktor yang diasumsikan atau bergantung pada dalam mengembangkan persyaratan dalam dokumen ini. Ini mencakup faktor-faktor yang mungkin berlawanan dengan fakta yang diketahui dan dapat mempengaruhi persyaratan produk. Misalnya, asumsi dapat mencakup adanya akses internet yang stabil, kepatuhan pengguna dalam mengikuti prosedur tertentu, atau ketergantungan pada sistem pihak ketiga untuk mengintegrasikan data. Hal-hal tersebut mencerminkan asumsi yang mendasari persyaratan produk dan mempengaruhi cara produk dirancang dan diimplementasikan.

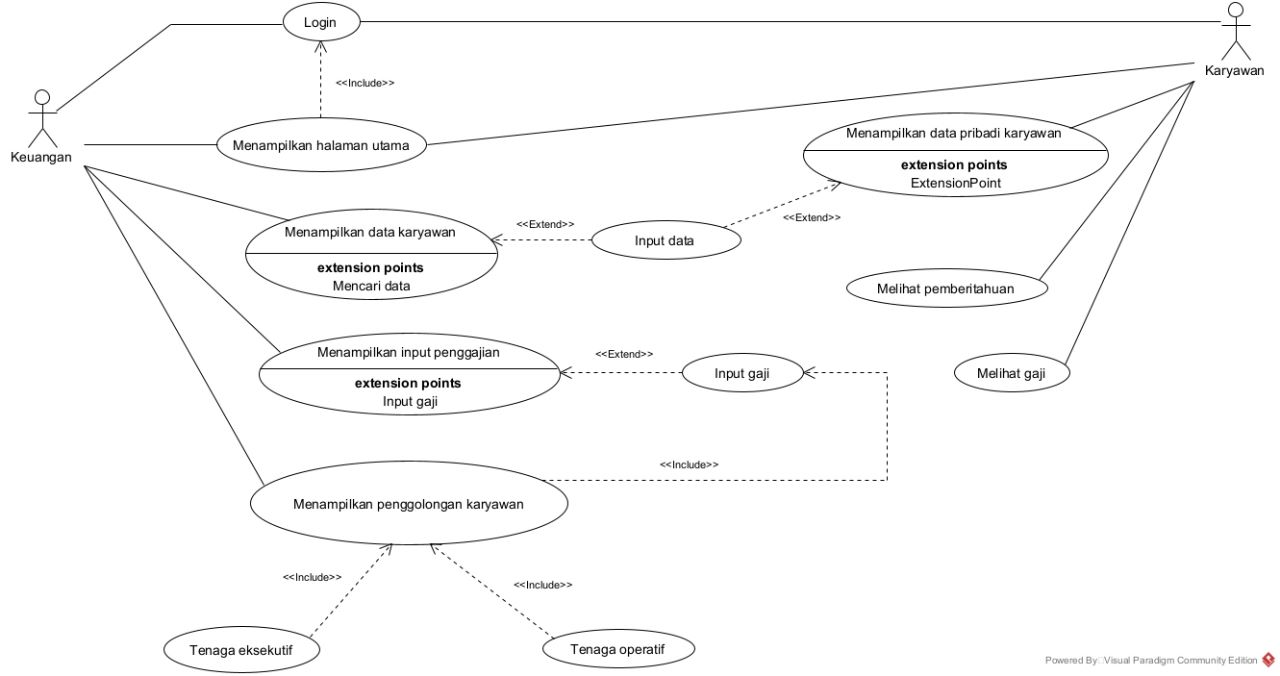
# Bab IV Functional Requirements

Kebutuhan fungsional adalah persyaratan atau spesifikasi yang menjelaskan fungsi-fungsi atau fitur-fitur yang harus ada dalam suatu sistem atau produk. Kebutuhan fungsional menentukan apa yang sistem harus lakukan, bagaimana sistem harus berperilaku, dan bagaimana interaksi dengan pengguna atau komponen lainnya. Contoh kebutuhan fungsional termasuk pemrosesan data, operasi matematika, validasi input, dan tampilan laporan.

## 4.1 Detailed Functional Requirements

1. Manajemen Data Karyawan:
2. Sistem harus dapat mencatat dan menyimpan data karyawan, termasuk informasi personal seperti nama, alamat, tanggal lahir, dan informasi pekerjaan seperti jabatan, departemen, tanggal mulai kerja, dan kontrak kerja.
3. Sistem harus mendukung operasi CRUD (Create, Read, Update, Delete) untuk mengelola data karyawan.
4. Sistem harus memiliki mekanisme untuk mengatur dan memperbarui informasi karyawan, termasuk perubahan jabatan, promosi, atau penambahan tunjangan.
5. Penghitungan Gaji dan Upah:
6. Sistem harus dapat menghitung gaji berdasarkan kontrak kerja yang ditandatangani, termasuk gaji pokok dan tunjangan yang relevan.
7. Sistem harus dapat menghitung upah berdasarkan output yang dihasilkan oleh setiap individu, seperti jumlah barang yang diproduksi atau layanan yang diberikan.
8. Sistem harus memperhitungkan faktor-faktor seperti absensi, ketepatan/kedisiplinan waktu, dan durasi kerja karyawan dalam penghitungan gaji dan upah.
9. Penggolongan Karyawan:
10. Sistem harus mampu mengelompokkan karyawan berdasarkan penggolongan, seperti tenaga eksekutif (manajerial) dan tenaga operatif (keterampilan teknis).
11. Sistem harus mempertimbangkan perbedaan dalam penghitungan gaji dan tunjangan antara karyawan dalam golongan yang berbeda.
12. Manajemen Tunjangan:
13. Sistem harus mendukung pencatatan dan pengelolaan tunjangan yang diberikan kepada karyawan, seperti tunjangan transportasi, tunjangan makan, atau tunjangan kesehatan.
14. Sistem harus memperhitungkan tunjangan yang relevan dalam penghitungan total gaji yang diterima oleh karyawan.
15. Perhitungan Potongan Pajak:
16. Sistem harus dapat menghitung potongan pajak penghasilan yang harus dipotong dari gaji karyawan sesuai dengan peraturan perpajakan yang berlaku.
17. Sistem harus mempertimbangkan faktor-faktor seperti tarif pajak, status pernikahan, dan jumlah tanggungan dalam perhitungan potongan pajak.
18. Pembuatan Slip Gaji:
19. Sistem harus dapat menghasilkan slip gaji yang berisi rincian gaji karyawan, termasuk gaji pokok, tunjangan, potongan pajak, dan gaji bersih yang diterima.
20. Slip gaji harus mencakup informasi yang relevan, seperti nama karyawan, ID karyawan, jabatan, departemen, periode gaji, tanggal pembayaran, dan rincian tunjangan dan potongan.
21. Pelaporan Penggajian:
22. Sistem harus mampu menghasilkan laporan penggajian yang merangkum total pengeluaran gaji dalam periode tertentu.
23. Laporan penggajian harus mencakup informasi seperti total gaji yang dibayarkan, rincian tunjangan dan potongan, dan jumlah gaji bersih yang diterima oleh karyawan.
24. Integrasi dengan Sistem Keuangan:
25. Sistem penggajian harus dapat terintegrasi dengan sistem keuangan perusahaan untuk memfasilitasi pembayaran gaji kepada karyawan.
26. Integrasi tersebut harus memungkinkan transfer dana langsung ke rekening bank karyawan dan memperbarui catatan keuangan secara otomatis.

## 4.2 Use Case Diagram



## 4.3 Use Case Scenario

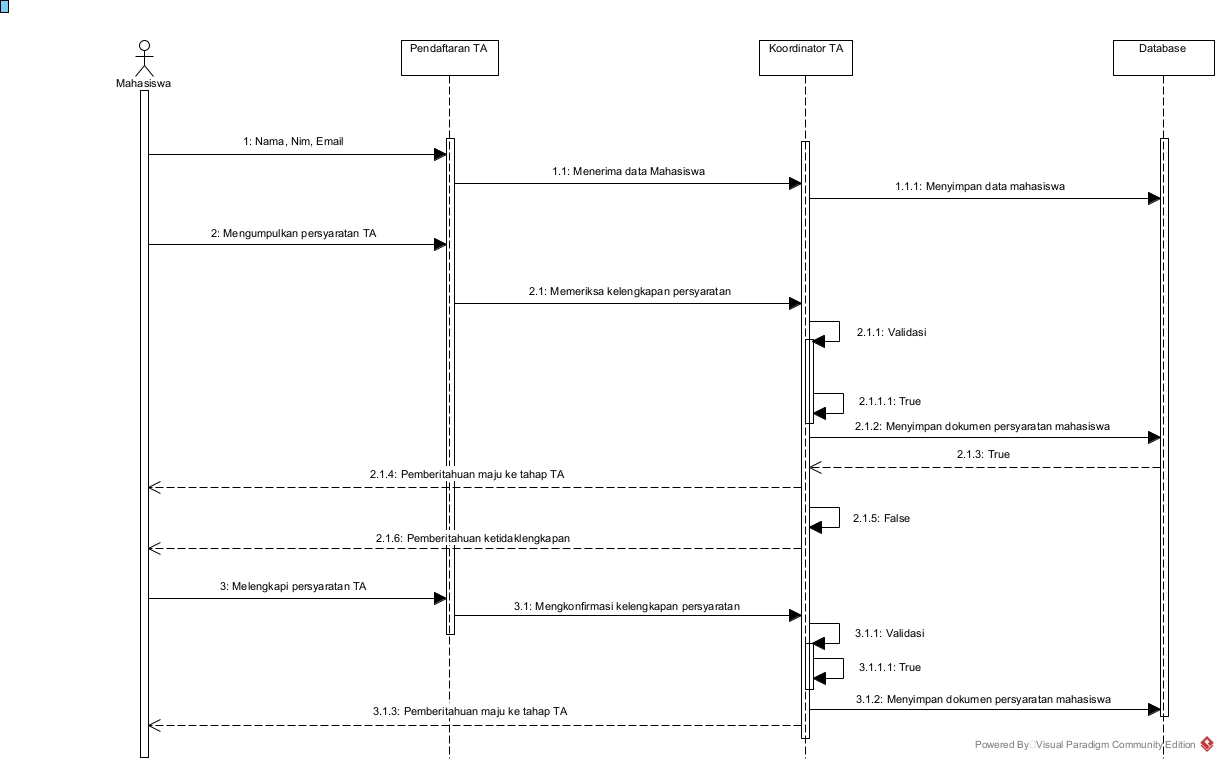
|  |  |  |
| --- | --- | --- |
| **Keuangan** | **Karyawan** | **Sistem Keuangan** |
| 1. Membuka situs Penggajian |  |  |
|  |  | 1. Menampilakn tampilan login |
| 1. Melakukan login |  |  |
|  |  | 1. Memvalidasi data user |
| 1. Menginput data karyawan |  |  |
|  |  | 1. Menyimpan data karyawan |
| 1. Menginput gaji karyawan |  |  |
|  |  | 1. Menyimpan inputan gaji karyawan |
|  | 1. Membuka situs penggajian |  |
|  |  | 1. Menampilakn tampilan login |
|  | 1. Melakukan login |  |
|  |  | 1. Memvalidasi data user |
|  |  | 1. Menampilkan data pribadi karyawan |
|  | 1. Memilih menu tampilkan gaji |  |
|  |  | 1. Menampilkan nominal gaji karyawan |

## 4.4 Class Diagram

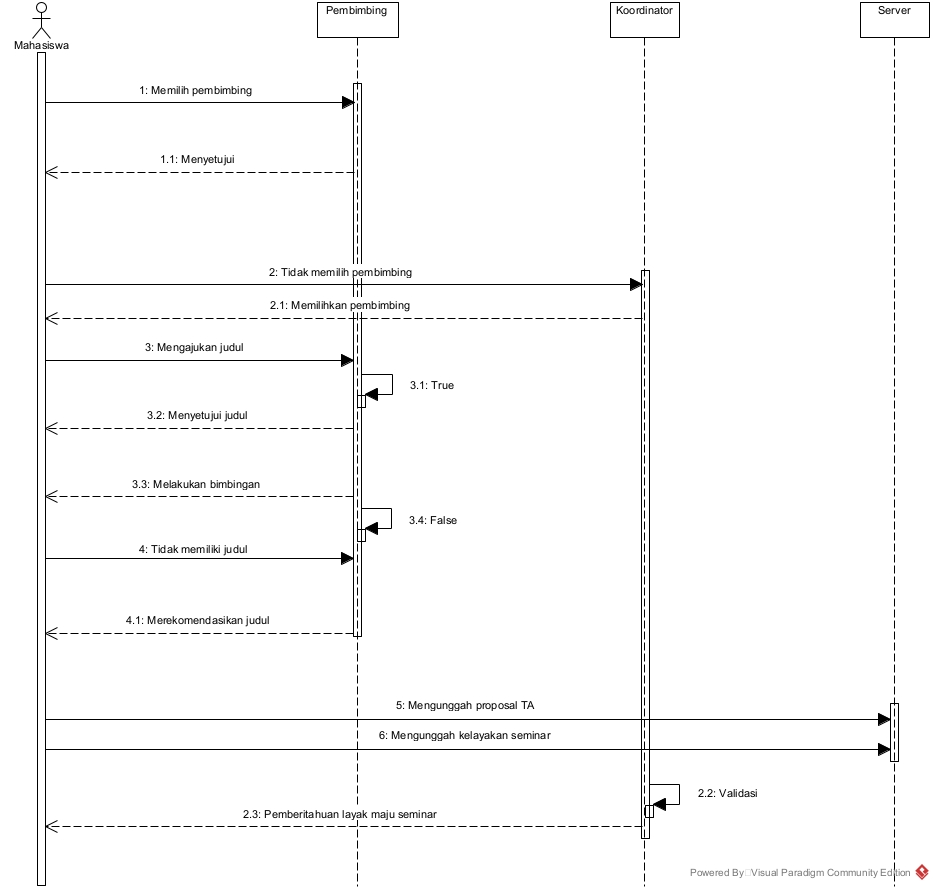
## 

## 4.5 Sequence Diagram

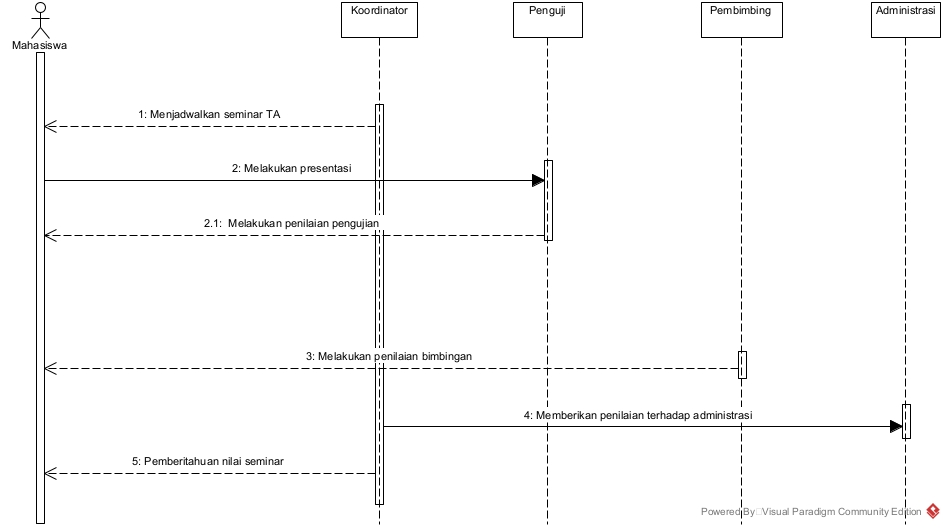
1. Pendaftaran Tugas Akhir



1. Pelaksanan Tugas Akhir

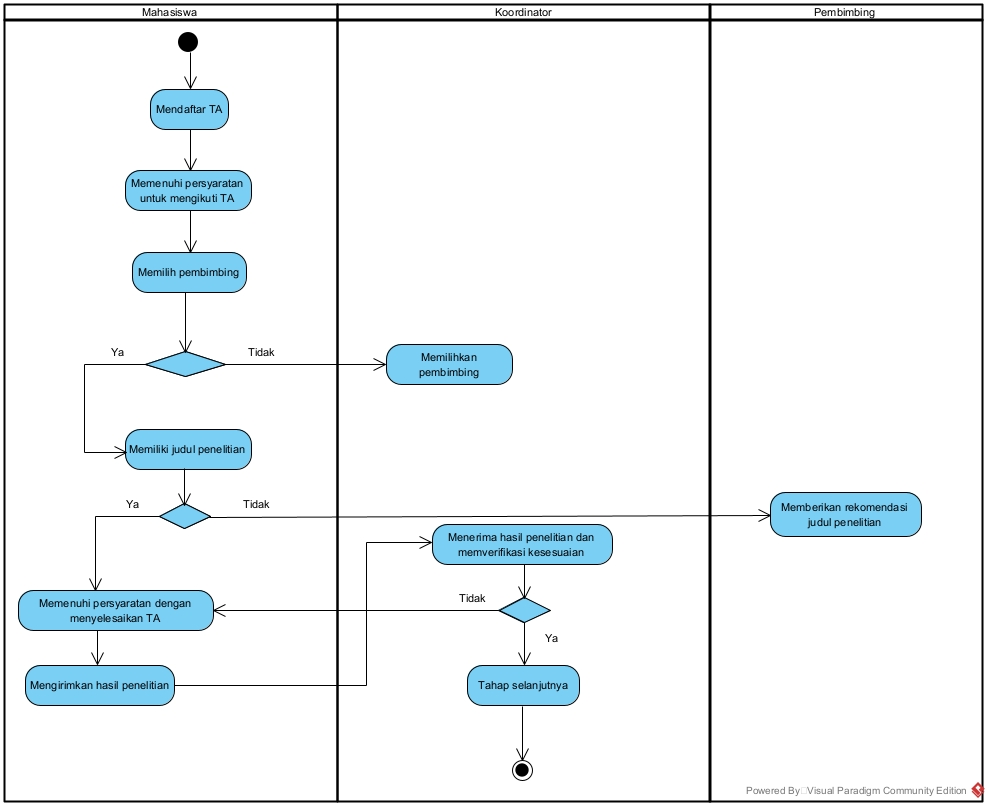


1. Seminar Tugas Akhir

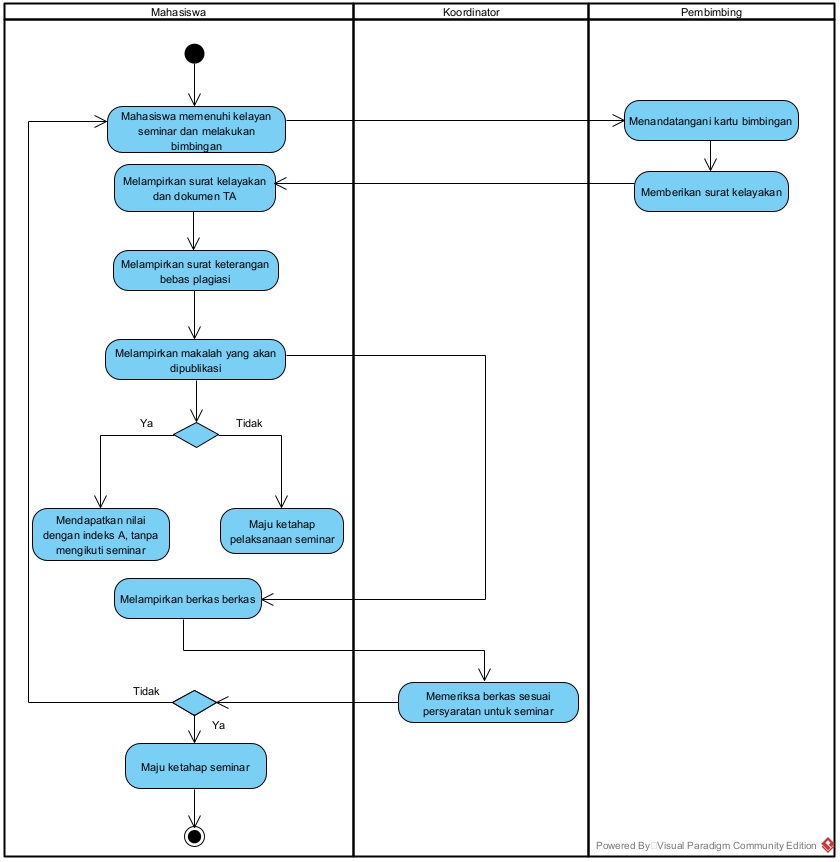


## 4.6 Activity Diagram

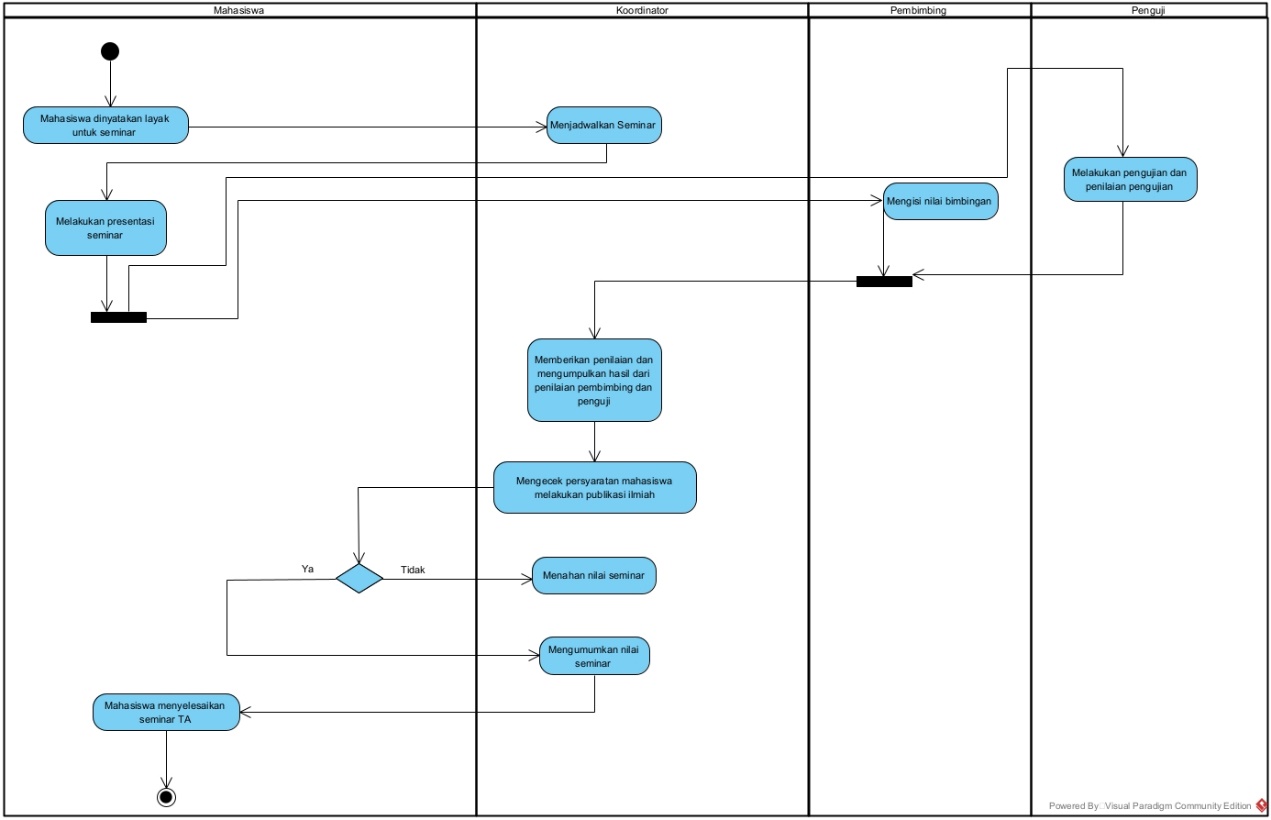
1. Daftar TA



1. Kelayakan Seminar

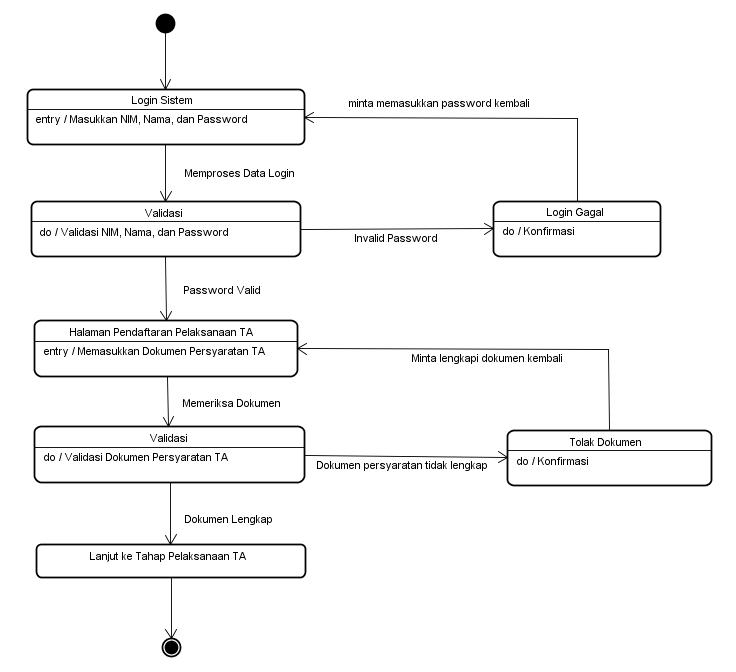


1. Seminar

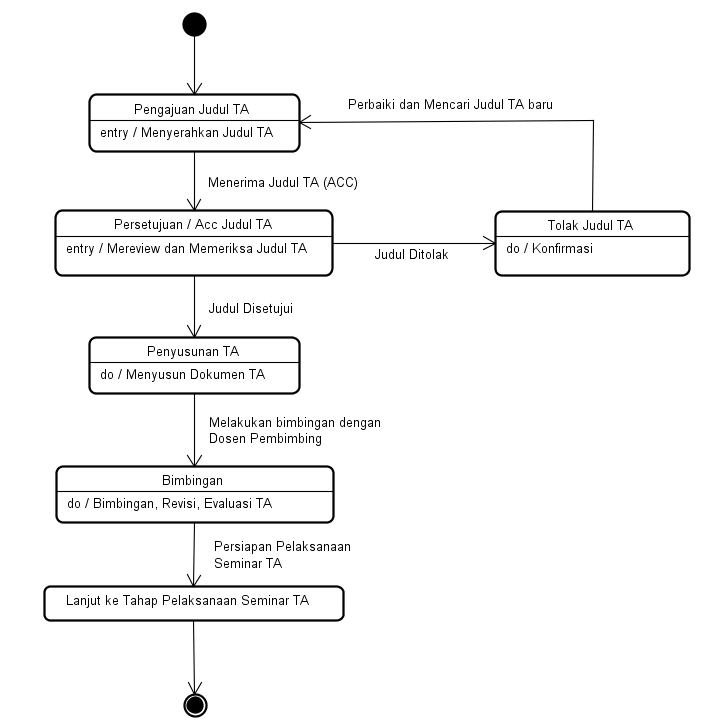


## 4.7 State Diagram

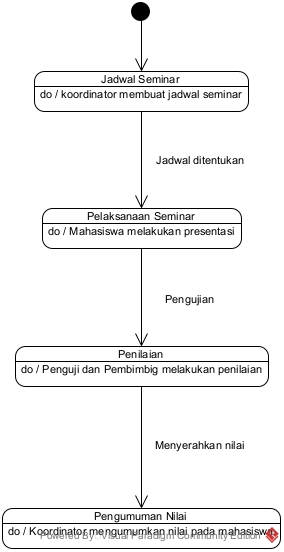
1. Pendaftaran TA



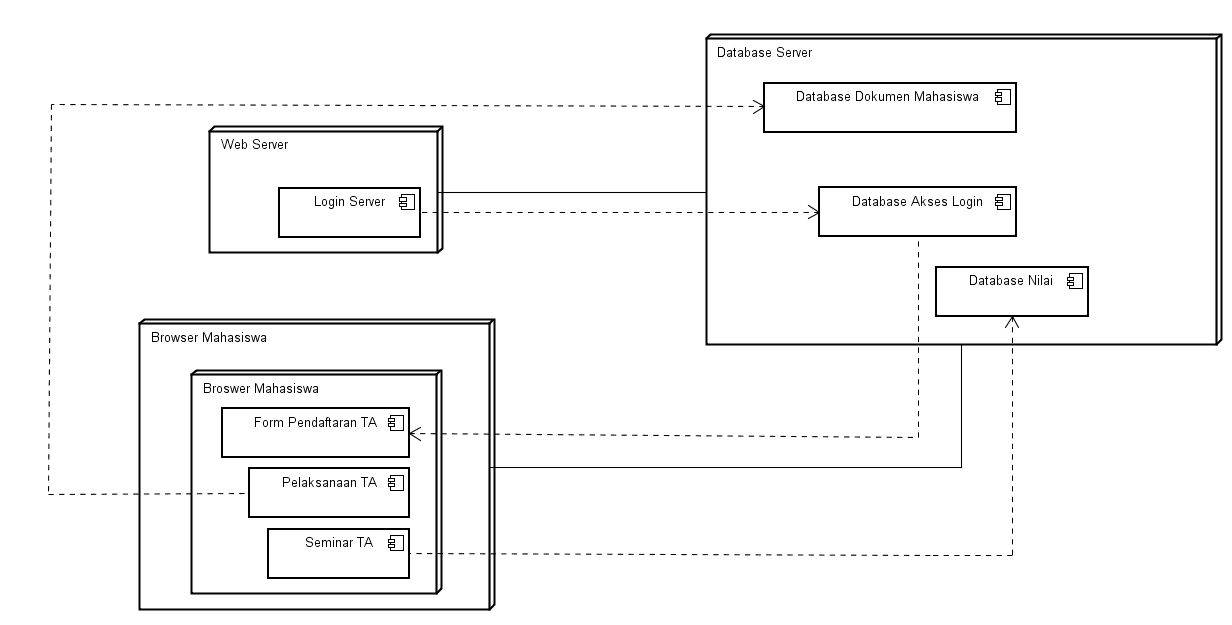
1. Pelaksanaan TA



1. Pelaksanaan Seminar TA



## 4.8 Deployment Diagram



# Bab V Non Functional Requirements

Kebutuhan non-fungsional adalah persyaratan atau spesifikasi yang menentukan atribut-atribut yang harus dimiliki oleh suatu sistem atau produk. Kebutuhan non-fungsional berkaitan dengan kualitas, performa, keamanan, ketersediaan, dan aspek lain yang tidak berkaitan langsung dengan fungsi-fungsi spesifik sistem. Contoh kebutuhan non-fungsional termasuk kecepatan respons sistem, kehandalan, keamanan data, user interface yang ramah pengguna, skalabilitas, dan ketersediaan sistem.

1. Sistem penggajian harus menjaga keamanan data karyawan, termasuk informasi pribadi dan gaji yang bersifat rahasia. Hal ini melibatkan penggunaan mekanisme keamanan seperti enkripsi data, otorisasi akses, dan pemantauan aktivitas yang mencurigakan.
2. Sistem penggajian harus mampu menangani pertumbuhan jumlah karyawan dalam organisasi. Ini mencakup kemampuan untuk mengelola dan menghasilkan slip gaji dengan cepat dan efisien tanpa mengalami penurunan kinerja seiring dengan peningkatan jumlah karyawan.
3. Sistem penggajian harus memiliki performa yang baik, dengan kemampuan untuk menghitung gaji dan tunjangan dengan cepat dan akurat. Ini penting agar proses penggajian dapat diselesaikan dalam waktu yang singkat dan menghindari keterlambatan pembayaran gaji kepada karyawan.
4. Sistem penggajian harus tersedia dan beroperasi secara stabil setiap saat. Dalam situasi di mana karyawan mengharapkan pembayaran gaji tepat waktu, sistem harus dapat diandalkan untuk memproses penggajian secara konsisten
5. Sistem penggajian harus mematuhi peraturan dan ketentuan hukum terkait penggajian, termasuk peraturan perpajakan, peraturan ketenagakerjaan, dan peraturan keuangan yang berlaku. Hal ini mencakup penghitungan dan pemotongan pajak penghasilan sesuai dengan aturan yang berlaku.
6. Sistem penggajian harus mudah digunakan oleh pengguna, seperti bagian kepegawaian atau HR, untuk mengelola data karyawan, menghitung gaji, dan menghasilkan slip gaji. Antarmuka yang intuitif dan panduan pengguna yang jelas akan membantu pengguna dalam menjalankan tugas mereka dengan efektif.
7. Sistem penggajian harus dapat terintegrasi dengan sistem lain yang relevan dalam organisasi, seperti sistem kehadiran karyawan, sistem manajemen kinerja, dan sistem keuangan. Integrasi ini memungkinkan aliran data yang lancar dan pembaruan otomatis antara sistem yang terkait.
8. Sistem penggajian harus memiliki mekanisme backup dan pemulihan data yang teratur dan dapat diandalkan. Hal ini penting untuk melindungi data penggajian dari kehilangan atau kerusakan, serta memastikan kelangsungan proses penggajian dalam situasi darurat atau kegagalan sistem.
9. Sistem penggajian harus dapat mengakomodasi kebutuhan penggajian yang kompleks, termasuk pembayaran tunjangan, bonus, potongan, dan perhitungan khusus lainnya. Kemampuan untuk menyesuaikan aturan penggajian dan memproses perubahan dengan cepat dan akurat sangat penting.
10. Sistem penggajian harus memiliki kemampuan untuk melacak dan mengaudit aktivitas yang terkait dengan penggajian, termasuk perubahan data karyawan, penghitungan gaji, dan pembayaran. Hal ini penting untuk memastikan keakuratan dan keabsahan proses penggajian serta mendukung kepatuhan audit internal dan eksternal.

## 5.1 Performance Requirements (optional)

Dalam konteks penggajian dan tunjangan pegawai, beberapa kebutuhan kinerja yang diinginkan organisasi dapat mencakup:

1. Waktu Respons Sistem: Sistem penggajian harus memberikan respons yang cepat saat menghitung dan memproses gaji karyawan. Hal ini penting agar proses penggajian dapat diselesaikan dengan efisien dan tepat waktu.
2. Skalabilitas: Sistem harus mampu menangani pertumbuhan jumlah karyawan dan volume data yang berkaitan dengan penggajian. Kemampuan sistem untuk tetap berkinerja baik dan menghasilkan gaji secara akurat ketika jumlah karyawan meningkat adalah hal yang penting.
3. Kecepatan Pengolahan Data: Sistem harus dapat mengolah data penggajian dengan cepat, termasuk perhitungan gaji, tunjangan, potongan, dan perhitungan khusus lainnya. Proses pengolahan data yang efisien akan membantu mempercepat proses penggajian secara keseluruhan.
4. Ketersediaan Sistem: Sistem penggajian harus tersedia dan dapat diakses oleh pengguna saat dibutuhkan. Dalam situasi di mana karyawan mengharapkan pembayaran gaji tepat waktu, sistem harus dapat beroperasi secara stabil dan tersedia setiap saat.

## 5.2 Safety Requirements (optional)

Dalam konteks penggajian dan tunjangan pegawai, beberapa kebutuhan keamanan yang diinginkan organisasi dapat mencakup:

1. Keamanan Data: Sistem penggajian harus melindungi data karyawan, termasuk informasi pribadi dan keuangan, dari akses yang tidak sah. Mekanisme keamanan seperti enkripsi data dan kontrol akses harus diterapkan untuk melindungi kerahasiaan dan integritas data.
2. Privasi Data: Sistem harus memastikan bahwa data karyawan dan informasi keuangan yang sensitif tidak diungkapkan kepada pihak yang tidak berwenang. Perlindungan privasi data adalah hal yang penting untuk menjaga kepercayaan karyawan dan mematuhi peraturan privasi yang berlaku.

## 5.3 Software Quality Attributes (optional)

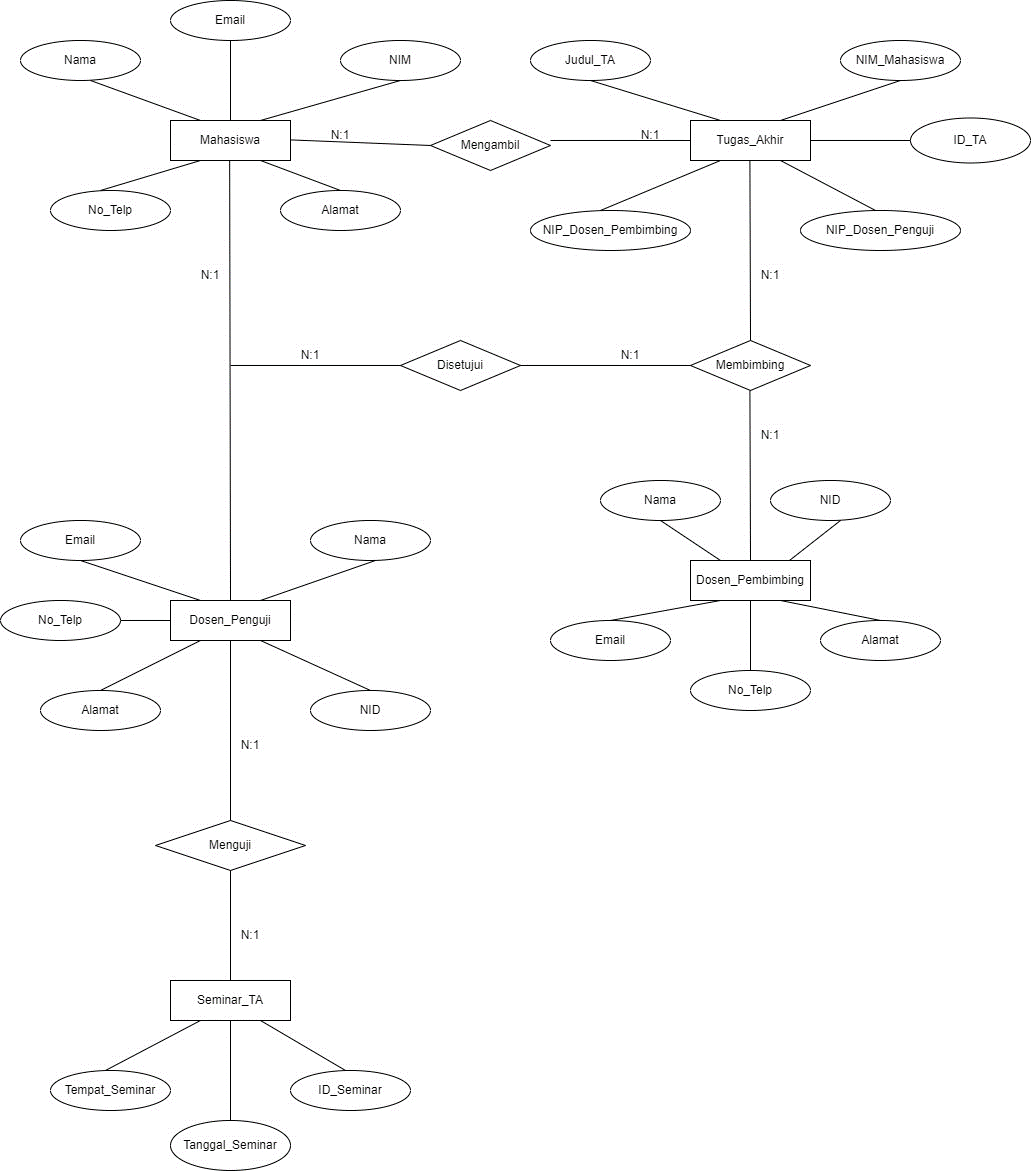
Dalam konteks penggajian dan tunjangan pegawai, beberapa atribut kualitas perangkat lunak yang diinginkan organisasi dapat mencakup:

1. Akurasi: Sistem harus menghasilkan perhitungan gaji dan tunjangan yang akurat berdasarkan aturan dan kebijakan yang berlaku. Kesalahan perhitungan dapat menyebabkan ketidakpuasan karyawan dan masalah hukum.
2. Keandalan: Sistem penggajian harus dapat beroperasi secara konsisten dan dapat diandalkan. Ketersediaan dan kehandalan sistem akan memastikan kelancaran proses penggajian dan meminimalkan potensi kesalahan atau kegagalan.
3. User-Friendly: Sistem penggajian harus mudah digunakan dan memiliki antarmuka yang ramah pengguna. Pengguna, seperti bagian kepegawaian, harus dapat dengan mudah mengelola data karyawan, menghitung gaji, dan menghasilkan slip gaji tanpa kesulitan.
4. Maintability: Sistem penggajian harus mudah dipelihara dan diperbarui. Kemampuan untuk memperbaiki bug, menyesuaikan aturan penggajian, atau menambahkan fitur baru dengan mudah akan membantu dalam menjaga kinerja dan kegunaan sistem.
5. Scalability: Sistem harus dapat dengan mudah ditingkatkan untuk menyesuaikan pertumbuhan organisasi dan kebutuhan penggajian yang lebih kompleks di masa depan. Kemampuan sistem untuk berkembang seiring dengan kebutuhan bisnis adalah faktor penting dalam jangka panjang.

# Bab VI Data Design

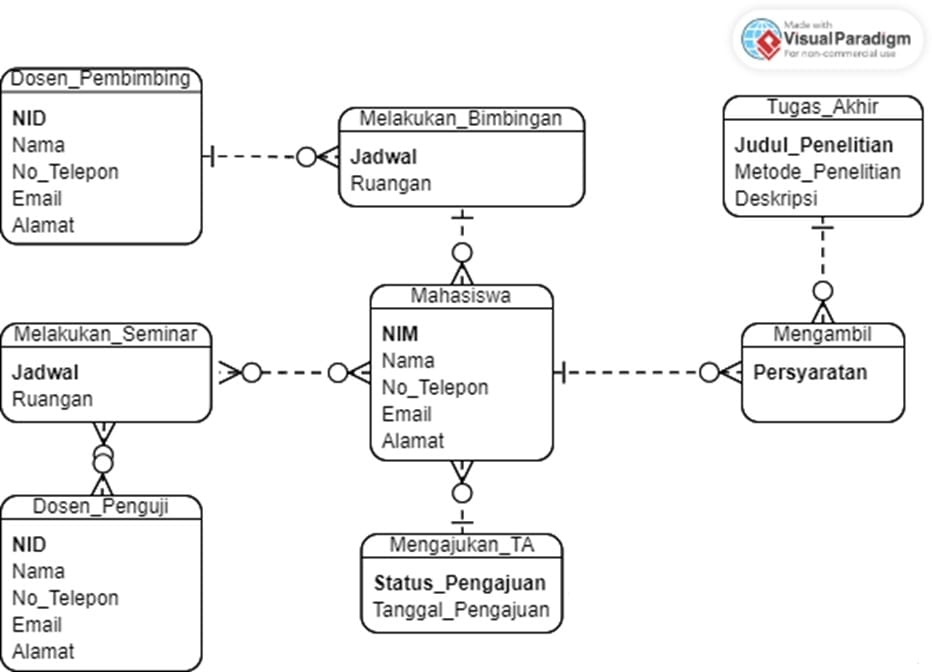
## 6.1 Logical Design

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.



## 6.2 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 VII Data Requirements

## 7.1 Input

Dalam pengelolaan data pada sistem penggajian dan tunjangan pegawai, terdapat beberapa kebutuhan data yang perlu diperhatikan. Berikut adalah detail kebutuhan data yang umum dalam sistem tersebut, beserta atribut dalam setiap datanya:

Data Karyawan:

1. Nama: Nama lengkap karyawan.
2. ID Karyawan: Identifikasi unik untuk setiap karyawan.
3. Jabatan: Jabatan atau posisi karyawan dalam organisasi.
4. Departemen: Departemen atau unit kerja di mana karyawan bekerja.
5. Kontak: Informasi kontak seperti alamat, nomor telepon, atau alamat email.
6. Tanggal Mulai Kerja: Tanggal karyawan mulai bekerja di perusahaan.
7. Informasi Bank: Informasi rekening bank karyawan untuk proses pembayaran gaji.

Data Gaji:

1. Gaji Pokok: Jumlah gaji pokok yang diterima karyawan.
2. Tunjangan: Jumlah tunjangan atau manfaat tambahan yang diberikan kepada karyawan.
3. Potongan: Potongan gaji seperti potongan pajak, asuransi, atau potongan lainnya.
4. Gaji Bersih: Jumlah gaji yang diterima karyawan setelah dikurangi potongan.
5. Periode Gaji: Periode waktu di mana gaji dibayarkan, misalnya bulanan atau mingguan.
6. Tanggal Pembayaran: Tanggal ketika gaji dibayarkan kepada karyawan.

Matriks Akses Pengguna terhadap Data (Matriks CRUD dan Pengguna):

Matriks akses pengguna terhadap data adalah alat yang digunakan untuk menggambarkan hak akses pengguna terhadap data dalam sistem. Matriks ini menggunakan konsep CRUD (Create, Read, Update, Delete) untuk menggambarkan aktivitas apa yang dapat dilakukan oleh pengguna terhadap setiap jenis data.

## 7.2 Output

PL dalam sistem penggajian dan tunjangan pegawai menghasilkan berbagai keluaran yang penting untuk manajemen dan karyawan. Berikut adalah detail beberapa keluaran yang umum dihasilkan oleh PL:

1. Slip Gaji: Laporan yang berisi rincian gaji karyawan pada periode gaji tertentu. Atribut dalam slip gaji dapat mencakup nama karyawan, ID karyawan, jabatan, departemen, gaji pokok, tunjangan, potongan, gaji bersih, periode gaji, dan tanggal pembayaran.
2. Laporan Pengeluaran Gaji: Laporan yang merangkum pengeluaran gaji keseluruhan dalam periode tertentu. Laporan ini mencakup total gaji yang dibayarkan kepada seluruh karyawan serta rincian tunjangan dan potongan yang terkait.
3. Laporan Kehadiran: Laporan yang menyajikan informasi tentang kehadiran karyawan dalam periode waktu tertentu. Atribut dalam laporan kehadiran meliputi nama karyawan, ID karyawan, tanggal kehadiran, durasi kerja, dan keterangan tambahan seperti izin atau cuti.
4. Laporan Pajak: Laporan yang merinci potongan pajak penghasilan yang dilakukan pada gaji karyawan. Laporan ini mencakup informasi tentang pajak yang dipotong dari gaji karyawan, termasuk perhitungan dan rincian pajak.

# Bab VIII 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.

Halaman Login



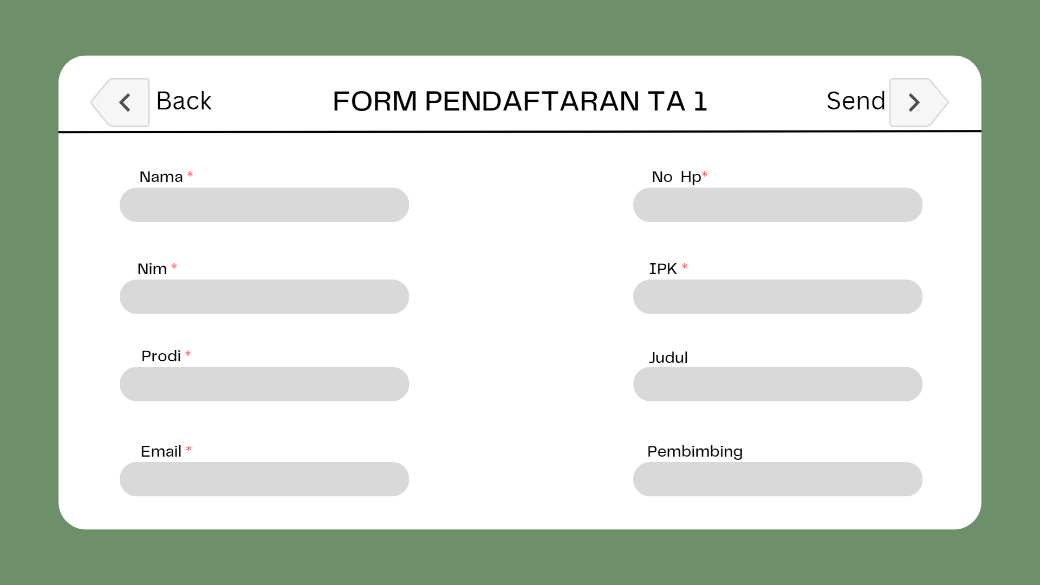
Lupa Password



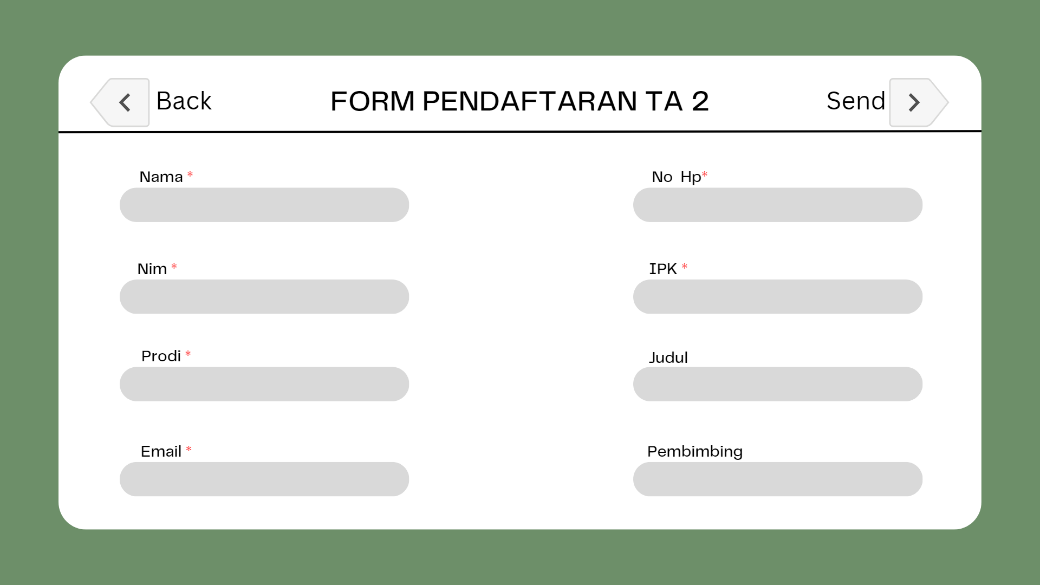
Beranda



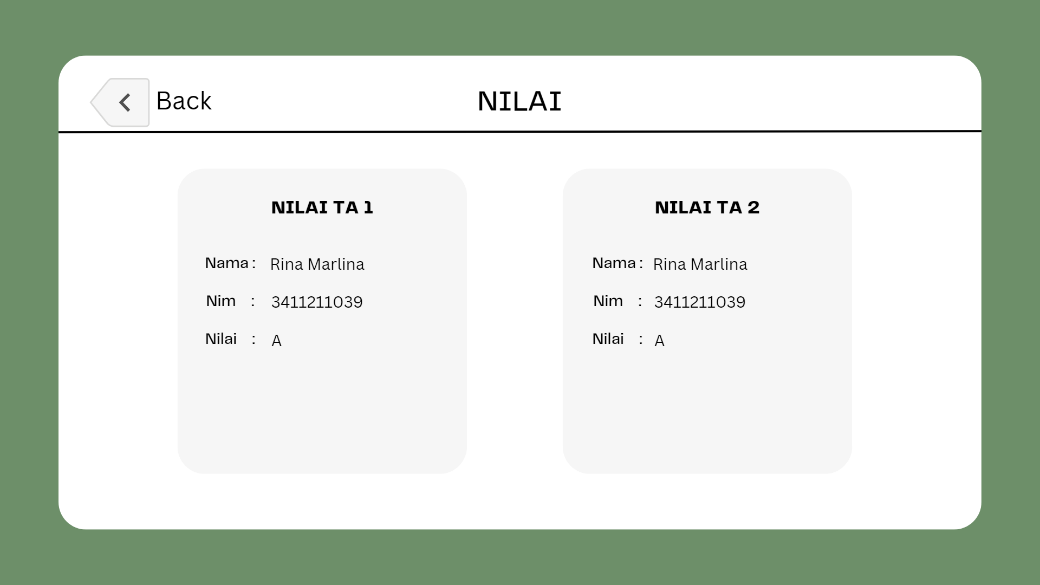
Formulir TA 1



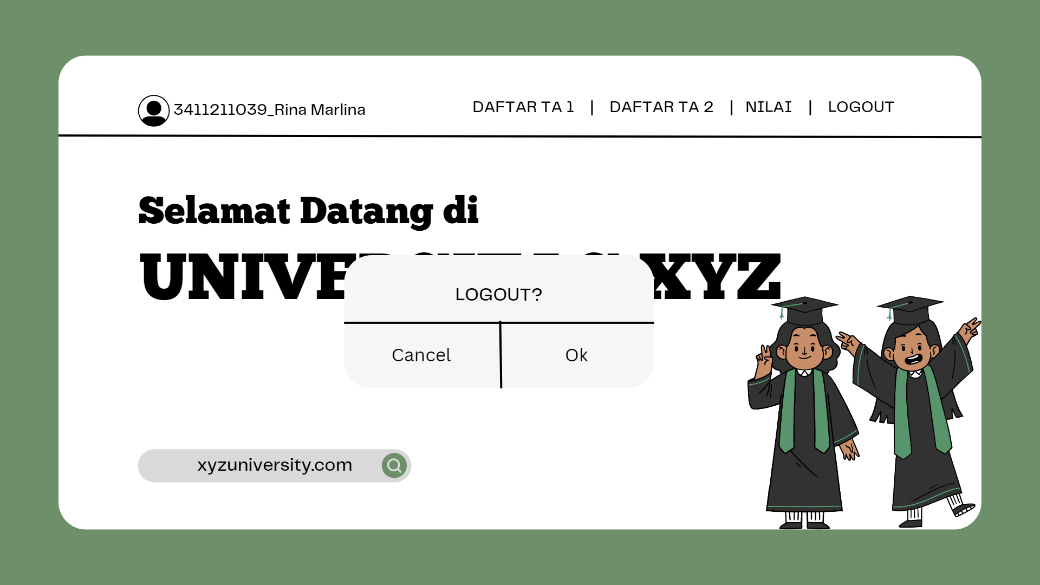
Formulir TA 2



Nilai



Logout



# Bab IX Interface Requirements

## 9.1 User Interface

Karakteristik dari setiap interface antara PL (sistem penggajian dan tunjangan pegawai) dengan pengguna dapat bervariasi tergantung pada desain dan implementasi sistem yang spesifik. Namun, dalam konteks ini, berikut adalah beberapa karakteristik umum dari antarmuka pengguna:

1. User-Friendly (Mudah digunakan): Antarmuka pengguna harus dirancang dengan prinsip kesederhanaan dan kemudahan penggunaan. Hal ini mencakup tata letak yang intuitif, ikon yang jelas, dan navigasi yang mudah dipahami agar pengguna dapat dengan mudah berinteraksi dengan sistem.
2. Visual yang Menarik: Antarmuka pengguna harus memiliki tampilan yang menarik dan estetis, menggunakan elemen desain seperti warna yang sesuai, tipografi yang mudah dibaca, dan grafik yang relevan. Hal ini dapat meningkatkan pengalaman pengguna dan membuatnya lebih menyenangkan dalam menggunakan sistem.
3. Responsif: Antarmuka pengguna harus responsif terhadap tindakan pengguna. Ini berarti interaksi dengan sistem harus memiliki waktu respons yang cepat, menjaga pengguna terlibat dan memberikan umpan balik yang sesuai setelah setiap tindakan.
4. Konsistensi: Antarmuka pengguna harus konsisten dalam seluruh sistem, sehingga pengguna dapat dengan mudah memahami dan menggunakan fitur yang berbeda tanpa perlu mempelajari ulang. Konsistensi dapat meliputi penggunaan ikon yang seragam, penempatan elemen yang konsisten, dan tata letak yang sama di berbagai bagian sistem.
5. Customizable (Dapat Dikustomisasi): Antarmuka pengguna yang memungkinkan pengguna untuk mengkustomisasi preferensi mereka, seperti mengubah tampilan tema, mengatur preferensi pengguna, atau mengatur notifikasi sesuai kebutuhan, dapat meningkatkan kepuasan dan kenyamanan pengguna.

## 9.2 Hardware Interface

Karakteristik dari setiap interface antara PL dengan komponen hardware dapat meliputi:

1. Perangkat Input: Sistem penggajian dapat berinteraksi dengan perangkat input seperti keyboard atau mouse untuk memasukkan data atau melakukan aksi di antarmuka pengguna.
2. Perangkat Output: Sistem penggajian dapat berinteraksi dengan perangkat output seperti monitor atau printer untuk menampilkan atau mencetak laporan, slip gaji, atau dokumen terkait penggajian.
3. Perangkat Penyimpanan: Sistem penggajian membutuhkan akses ke perangkat penyimpanan seperti hard disk atau server untuk menyimpan dan mengelola data karyawan, data gaji, dan informasi terkait lainnya.

## 9.3 Software Interface

Karakteristik dari setiap interface antara PL dengan komponen software dapat meliputi:

1. Basis Data: Sistem penggajian dapat berinteraksi dengan sistem basis data untuk menyimpan, mengelola, dan mengambil data karyawan, data gaji, dan informasi terkait lainnya.
2. Sistem Operasi: Sistem penggajian membutuhkan kompatibilitas dengan sistem operasi tertentu, seperti Windows, macOS, atau Linux, untuk menjalankan aplikasi dan menyediakan fungsionalitas yang diperlukan.
3. Aplikasi Tambahan: Sistem penggajian dapat berintegrasi dengan aplikasi tambahan, seperti sistem keuangan, sistem manajemen SDM, atau sistem perpajakan, untuk pertukaran data dan fungsi yang lebih terintegrasi.

## 9.4 Communication Interface

Kebutuhan komunikasi untuk PL (sistem penggajian dan tunjangan pegawai) dapat mencakup:

1. Komunikasi Internal: PL harus mendukung komunikasi internal antara departemen atau unit yang terlibat dalam penggajian, seperti antara bagian kepegawaian, keuangan, dan manajemen SDM. Komunikasi internal dapat dilakukan melalui email, pesan instan, atau sistem komunikasi internal perusahaan.
2. Komunikasi Eksternal: PL dapat memerlukan komunikasi eksternal dengan pihak-pihak eksternal, seperti bank atau institusi keuangan untuk pemrosesan pembayaran gaji atau perpajakan. Komunikasi eksternal dapat dilakukan melalui protokol dan layanan yang relevan, seperti protokol perbankan elektronik atau API.