

**SOFTWARE REQUIREMENTS SPESIFICATION
APLIKASI KOPERASI**

DISUSUN OLEH :

ANWAR SYARIFUDDIN – 2250081025

HERLY SETIAWAN NURHIDAYAT - 2250081062

ASTRIANI KOMERI – 2250081067



**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN INFORMATIKA
UNIVERSITAS JENDERAL ACHMAD YANI
TAHUN 2024**

DAFTAR ISI

DAFTAR ISI.....	1
DAFTAR GAMBAR	2
BAB I INTRODUCING	3
I.1. Purpose	3
I.2. Intended Audience and Reading Suggestions.....	3
I.3. Project Scope	4
BAB II OVERALL DESCRIPTION	5
II.1. Organizations.....	5
II.2. Product Perspective	5
II.3. User Classes and Characteristics	6
II.4. Operating Environment	7
II.5. Design and Implementation Constraint	7
II.6. Assumptions and Dependencies	8
BAB III FUNCTIONAL REQUIREMENTS	9
III.1. Detailed Functional Requirements	9
III.2. Use Case Diagram.....	10
III.3. Use Case Scenario.....	10
BAB IV FUNCTIONAL REQUIREMENTS	12
IV.1. Performance Requirements	12
IV.2. Safety Requirements	12
IV.3. Software Quality Attributes	13
BAB V DATA REQUIREMENTS.....	14
V.1. Input	14
V.2. Output.....	14
BAB VI INTERFACE REQUIREMENTS	16
VI.1. User Interface.....	16
VI.2. Hardware Interface.....	16
VI.3. Software Interface	17
VI.4. Communication Interface.....	18

DAFTAR GAMBAR

Gambar 3. 1 Use Case Diagram.....	10
-----------------------------------	----

BAB I

INTRODUCING

I.1. Purpose

SRS (Software Requirements Specification) atau Spesifikasi Kebutuhan Perangkat Lunak untuk aplikasi manajemen administrasi tugas akhir di Universitas InXYZ adalah dokumen yang digunakan untuk menjelaskan secara rinci kebutuhan fungsional dan nonfungsional suatu perangkat lunak yang akan dikembangkan. Tujuan utama dari pembuatan SRS adalah sebagai berikut:

1. Memahami kebutuhan pengguna: SRS membantu mengidentifikasi dan mendokumentasikan kebutuhan pengguna atau pemangku kepentingan terkait dengan perangkat lunak yang akan dikembangkan.
2. Mengkomunikasikan persyaratan: SRS berfungsi sebagai dokumen komunikasi antara tim pengembang, pemangku kepentingan, dan pengguna akhir. Dokumen ini menyediakan rincian yang jelas dan terstruktur tentang apa yang diharapkan dari perangkat lunak, sehingga semua pihak yang terlibat memiliki pemahaman yang seragam tentang kebutuhan yang harus dipenuhi.
3. Merencanakan pengembangan perangkat lunak: SRS membantu dalam perencanaan pengembangan perangkat lunak dengan mengidentifikasi kebutuhan yang harus dipenuhi, batasan dan kendala yang ada, serta lingkup pengembangan yang diharapkan. Hal ini membantu tim pengembang dalam merencanakan sumber daya, jadwal, dan pendekatan pengembangan yang sesuai.

I.2. Intended Audience and Reading Suggestions

1. Tim Pengembang Perangkat Lunak: Anggota tim pengembang perangkat lunak adalah pembaca utama SRS. Dokumen ini memberikan mereka panduan yang jelas tentang kebutuhan fungsional dan nonfungsional yang harus dipenuhi dalam pengembangan perangkat lunak.
2. Pemangku Kepentingan (Stakeholders): Pemangku kepentingan, seperti manajer proyek, pengguna akhir, atau pemilik produk, juga merupakan pembaca yang penting. SRS memberikan mereka gambaran lengkap tentang apa yang diharapkan dari perangkat lunak yang akan dikembangkan.
3. Tim QA (Quality Assurance): Tim QA bertanggung jawab untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi persyaratan yang ditetapkan dalam SRS. Mereka

menggunakan SRS sebagai acuan untuk merencanakan dan melaksanakan pengujian perangkat lunak.

4. Pemelihara Perangkat Lunak: Pada masa depan, ketika perubahan, perbaikan, atau perluasan perangkat lunak diperlukan, SRS akan menjadi dokumen referensi yang berharga. Tim pemelihara perangkat lunak dapat merujuk kembali ke SRS untuk memahami kebutuhan awal dan melaksanakan perubahan dengan benar.

I.3. Project Scope

Project Scope (Lingkup Proyek) dalam pembuatan SRS mengacu pada batasan dan cakupan pekerjaan yang akan dilakukan dalam pengembangan perangkat lunak. Dokumen SRS harus mencakup informasi yang jelas tentang apa yang termasuk dan tidak termasuk dalam proyek tersebut. Berikut adalah beberapa poin yang harus dipertimbangkan dalam menentukan scope dalam SRS:

1. Fungsi dan Fitur: SRS harus menjelaskan secara rinci fungsi dan fitur perangkat lunak yang akan dikembangkan. Ini mencakup deskripsi tentang apa yang dapat dilakukan oleh perangkat lunak dan apa yang tidak dapat dilakukan.
2. Batasan: SRS harus mencantumkan batasan-batasan yang ada dalam proyek. Ini mungkin mencakup batasan waktu, anggaran, sumber daya manusia, atau teknologi yang tersedia. 3 Batasan-batasan ini membantu memastikan bahwa ekspektasi realistis ditegakkan dan mencakup dalam pengembangan perangkat lunak.
3. Ketergantungan Eksternal: Jika proyek perangkat lunak bergantung pada sistem atau layanan eksternal, SRS harus mencakup ketergantungan tersebut. Hal ini membantu dalam merencanakan integrasi dengan sistem atau layanan tersebut, serta menentukan batasan yang relevan.

BAB II OVERALL DESCRIPTION

II.1. Organizations

1. Visi
Menyediakan solusi perangkat lunak inovatif, transparan, dan integrasi untuk mendukung pemberdayaan anggota koperasi melalui layanan yang mudah diakses, efisien, dan terpercaya.
2. Misi
Mengembangkan sistem yang mampu mengotomatisasi proses administrasi koperasi, seperti pencatatan simpanan, pinjaman, dan pembagian SHU (Sisa Hasil Usaha).
3. Struktur Organisasi
Organisasi kami terdiri dari departemen pengembangan perangkat lunak, manajemen proyek, departemen pemasaran, dan departemen dukungan pelanggan. Tim pengembang akan bekerja di bawah manajer proyek yang melapor ke direktur teknis. Tim dukungan pelanggan akan berada di bawah manajer dukungan pelanggan yang melapor ke direktur operasional.
4. Keahlian dan Pengalaman Organisasi
Organisasi kami memiliki pengalaman yang luas dalam pengembangan perangkat lunak berbasis web. Tim pengembang kami memiliki keahlian dalam penggunaan teknologi terkini dan memahami persyaratan unik yang terkait dengan pengelolaan tugas akhir. Selama beberapa tahun terakhir, kami telah berhasil mengembangkan aplikasi manajemen tugas akhir untuk beberapa institusi pendidikan terkemuka, dengan umpan balik positif dari pengguna.

II.2. Product Perspective

Dalam konteks pembuatan SRS untuk aplikasi sistem koperasi, PL akan merujuk pada aplikasi perangkat lunak yang dikembangkan untuk memenuhi kebutuhan tersebut. PL ini akan memiliki fitur-fitur dan fungsionalitas yang ditentukan dalam SRS.

1. Hubungan dengan Sistem yang ada : Mengintegrasikan proses yang saat ini dilakukan secara manual atau menggunakan aplikasi terpisah.
2. Antarmuka Pengguna : Sistem menyediakan antarmuka yang intuitif untuk berbagai jenis pengguna, seperti pengurus koperasi, anggota koperasi, dan auditor. Aksesibilitas diutamakan untuk memastikan pengguna dapat mengoperasikan sistem tanpa memerlukan pelatihan teknis yang mendalam.

3. Konteks Sistem Baru : Sistem ini dirancang untuk berjalan pada platform berbasis web untuk memastikan aksesibilitas yang luas. Menggunakan arsitektur berbasis cloud untuk akses jarak jauh.
4. Asumsi dan Ketergantungan : Diasumsikan bahwa pengguna memiliki akses ke koneksi internet yang stabil untuk mengoperasikan sistem. Ketergantungan pada teknologi pihak ketiga.

II.3. User Classes and Characteristics

Dalam aplikasi sistem koperasi, beberapa pengguna yang berbeda akan terlibat dalam menggunakan perangkat lunak (PL) tersebut. Berikut adalah beberapa pengguna yang mungkin terlibat dan beberapa karakteristik, hak akses, dan kewajiban yang terkait:

1. Pengurus Koperasi
 - Karakteristik : Memiliki akses penuh ke fitur manajemen data, seperti pencatatan simpanan, pinjaman, dan SHU.
 - Kebutuhan : Antarmuka yang intuitif, sistem pelaporan yang fleksibel dan detail, dan keamanan tinggi untuk melindungi data keuangan koperasi.
2. Anggota Koperasi
 - Karakteristik : Kemampuan teknologi bervariasi, dari pengguna awam hingga yang terbiasa dengan aplikasi digital.
 - Kebutuhan : Antarmuka yang sederhana dan ramah pengguna, aksesibilitas melalui aplikasi fitur notifikasi untuk pengingat pembayaran atau informasi penting.
3. Auditor
 - Karakteristik : Akses hanya ke data dan laporan yang relevan tanpa kemampuan untuk mengubah data, memiliki pemahaman teknis yang cukup untuk membaca laporan dan menganalisis data.
 - Kebutuhan : Fitur laporan yang detail dan mudah diakses, sistem pencatatan audit trail untuk memantau perubahan data, dan hak akses terbatas sesuai dengan perannya.

II.4. Operating Environment

Dalam SRS (Software Requirements Specification) untuk aplikasi manajemen tugas akhir, environment (lingkungan) merujuk pada persyaratan perangkat keras (hardware) dan perangkat lunak (software) yang diperlukan untuk menjalankan aplikasi dengan baik. Berikut adalah beberapa komponen yang perlu dipertimbangkan dalam environment untuk aplikasi sistem operasi :

1. Hardware Platform
Persyaratan Minimum: Komputer atau laptop dengan prosesor yang cukup kuat, memori RAM yang memadai, dan penyimpanan yang mencukupi untuk menginstal dan menjalankan aplikasi.
2. Sistem Operasi
Aplikasi ini dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, atau Linux. 9 Persyaratan Minimum: Misalnya, Windows 10, macOS 10.14 (Mojave), atau Ubuntu 18.04 LTS. Dalam hal ini, versi minimum yang diperlukan harus ditentukan agar aplikasi berfungsi dengan baik.
3. Web Browser
Aplikasi ini mungkin membutuhkan akses melalui web browser. Persyaratan Browser: Misalnya, Google Chrome versi terbaru, Mozilla Firefox versi terbaru, atau Safari versi terbaru. Jelaskan browser dan versi minimum yang mendukung aplikasi secara optimal.
4. Database
Aplikasi mungkin menggunakan database untuk menyimpan dan mengelola data terkait tugas akhir. Persyaratan Database: Misalnya, MySQL versi terbaru, PostgreSQL versi terbaru, atau MongoDB versi terbaru. Tentukan database dan versi minimum yang mendukung aplikasi.

II.5. Design and Implementation Constraint

Batasan desain dan implementasi dalam konteks pembuatan SRS (Software Requirements Specification) untuk aplikasi sistem operasi dapat mencakup beberapa hal sebagai berikut:

1. Lingkup Fungsionalitas
Tentukan dengan jelas fungsionalitas yang akan diimplementasikan dalam aplikasi. Jelaskan batasan mengenai fitur-fitur yang akan disertakan dan fitur-fitur yang tidak akan dimasukkan dalam versi awal aplikasi
2. Keterbatasan Teknis
Identifikasi keterbatasan teknis yang perlu diperhatikan dalam desain dan implementasi aplikasi.

3. Ketersediaan Sumber Daya
Pertimbangkan ketersediaan sumber daya yang tersedia, seperti tenaga kerja, anggaran, dan waktu yang diperlukan untuk desain dan implementasi aplikasi.
4. Keamanan dan Privasi
Identifikasi batasan keamanan dan privasi yang harus diperhatikan dalam desain dan implementasi aplikasi.

II.6. Assumptions and Dependencies

Ada beberapa faktor yang bisa bertentangan dengan fakta yang diketahui dan dapat mempengaruhi requirements yang dinyatakan dalam SRS (Software Requirements Specification). Berikut adalah beberapa faktor yang perlu dipertimbangkan:

5. Ketersediaan Sumber Daya: Faktor seperti keterbatasan anggaran, waktu, atau tenaga kerja dapat mempengaruhi requirements yang dapat diimplementasikan.
6. Batasan Teknis: Faktor-faktor teknis seperti keterbatasan perangkat keras, sistem operasi, atau lingkungan IT yang ada dapat mempengaruhi requirements yang dapat diimplementasikan.
7. Kebijakan dan Regulasi: Kebijakan dan regulasi yang berlaku, baik di tingkat institusi pendidikan maupun di tingkat pemerintah, dapat mempengaruhi requirements yang dijelaskan dalam SRS..
8. Kebutuhan Pengguna yang Bertentangan: Berbagai pengguna aplikasi mungkin memiliki kebutuhan atau persyaratan yang bertentangan.
9. Perubahan Kebutuhan: Fakta bahwa kebutuhan pengguna dan lingkungan bisnis dapat berubah seiring waktu dapat mempengaruhi requirements yang dinyatakan dalam SRS.
10. Keterbatasan Teknologi: Kemajuan teknologi yang cepat dapat menyebabkan adanya keterbatasan dalam hal kemampuan teknologi yang tersedia saat ini.

Dalam menghadapi faktor-faktor ini, tim pengembang dan pihak terkait harus melakukan evaluasi dan pengelolaan risiko yang tepat untuk menentukan requirements yang realistis dan dapat diimplementasikan dalam proyek pembangunan aplikasi.

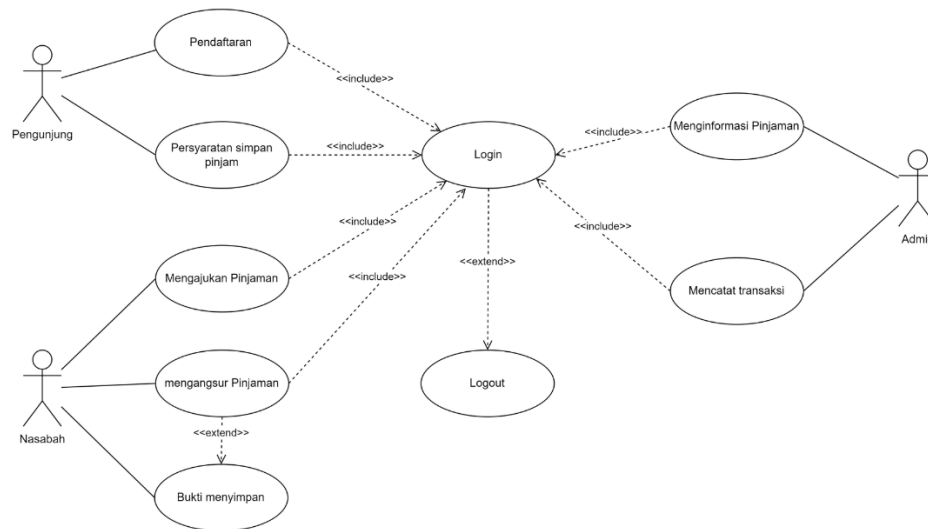
BAB III

FUNCTIONAL REQUIREMENTS

III.1. Detailed Functional Requirements

1. Manajemen Anggota
 - Pendaftaran anggota baru, termasuk pengumpulan data pribadi, kontak, dan informasi keanggotaan.
 - Pembaruan dan pengelolaan data anggota.
 - Pencarian dan penampilan profil anggota.
 - Pengelolaan status keanggotaan, termasuk suspense atau penghapusan keanggotaan.
2. Manajemen Simpanan
 - Pengelolaan simpanan wajib, sukarela, dan berjangka.
 - Catatan transaksi simpanan dan penarikan.
 - Perhitungan bunga dan distribusi kepada anggota.
3. Manajemen Pinjaman
 - Pengajuan dan persetujuan pinjaman.
 - Perhitungan bunga dan jadwal pembayaran pinjaman.
 - Pencatatan pembayaran cicilan dan pelacakan keterlambatan pembayaran.
4. Manajemen Keuangan
 - Pembukuan dan pencatatan transaksi keuangan.
 - Laporan keuangan, termasuk neraca, laporan laba dan rugi, dan laporan arus kas.
 - Manajemen kas dan bank.
5. Manajemen Inventory
 - Pencatatan dan pembaruan inventori barang atau jasa.
 - Manajemen pembelian dan penjualan inventori.
 - Laporan stok dan pemantauan barang yang kurang.
6. Laporan dan Analisis
 - Laporan keanggotaan, keuangan, pinjaman, dan simpanan.
 - Analisis tren dan performa koperasi.
 - Dashboard untuk visualisasi data dan metrik kunci.

III.2. Use Case Diagram



Gambar 3. 1 Use Case Diagram

III.3. Use Case Scenario

1. Pendaftaran
 - Pengunjung membuka halaman pendaftaran.
 - Pengunjung mengisi informasi pribadi, nama, alamat, dan kontak.
 - Sistem memvalidasi data yang diinput.
 - Sistem menyimpan data pendaftaran dan memberikan notifikasi keberhasilan pendaftaran.
2. Persyaratan Simpan Pinjam
 - Pengunjung memilih opsi di menu utama.
 - Sistem menampilkan dokumen atau daftar persyaratan yang perlu dipenuhi.
 - Pengunjung membaca dan memahami informasi tersebut.
3. Login
 - Pengguna membuka halaman login.
 - Pengguna memasukkan username dan password.
 - Sistem memverifikasi kredensial pengguna.
 - Sistem memberikan akses ke fitur sesuai peran pengguna.
4. Mengajukan Pinjaman
 - Nasabah login ke sistem.
 - Nasabah memilih menu.
 - Nasabah mengisi formulir pengajuan pinjaman.
 - Sistem memvalidasi data pengajuan dan menyimpannya.
 - Nasabah mendapatkan notifikasi bahwa pengajuan sedang diproses.

5. Mengangsur Pinjaman
 - Nasabah login ke sistem.
 - Nasabah memilih menu.
 - Nasabah memasukkan jumlah angsuran yang akan dibayarkan.
 - Sistem memvalidasi data pembayaran dan mencatat transaksi.
 - Sistem memberikan konfirmasi bahwa pembayaran berhasil.
6. Bukti Menyimpan
 - Nasabah login ke sistem.
 - Nasabah memilih menu.
 - Sistem menampilkan riwayat setoran simpanan, termasuk bukti pembayaran.
7. Menginformasikan Pinjaman
 - Admin login ke sistem.
 - Admin memilih menu “Informasi Pinjaman”.
 - Admin memasukkan data nasabah yang ingin diperiksa.
 - Sistem menampilkan detail pinjaman nasabah.
 - Admin menyampaikan informasi kepada nasabah.
8. Mencatat Transaksi
 - Admin login ke sistem.
 - Admin memilih menu “Catat Transaksi”.
 - Admin mengisi data transaksi.
 - Sistem memvalidasi data dan menyimpannya.

BAB IV

FUNCTIONAL REQUIREMENTS

IV.1. Performance Requirements

Kebutuhan performa umum yang sering diinginkan oleh organisasi :

- Waktu Respons
Organisasi mungkin menginginkan sistem yang memberikan waktu respons yang cepat terhadap interaksi pengguna.
- Kecepatan Pemrosesan
Organisasi dapat menginginkan sistem yang mampu memproses data dengan kecepatan tinggi.
- Ketersediaan dan Downtime Minimal
Organisasi mungkin membutuhkan sistem yang memiliki tingkat ketersediaan yang tinggi dengan waktu downtime yang minimal.
- Skalabilitas
Organisasi mungkin membutuhkan sistem yang dapat dengan mudah diukur untuk menangani pertumbuhan dan meningkatkan kapasitas sesuai kebutuhan.
- Kinerja Saat Beban Tinggi
Jika organisasi berharap sistem akan digunakan oleh banyak pengguna atau dalam situasi dengan lalu lintas yang tinggi, mereka mungkin membutuhkan sistem yang mampu menjaga kinerja yang baik bahkan saat ada lonjakan aktivitas atau beban pengguna yang tinggi.

IV.2. Safety Requirements

- Organisasi mungkin menginginkan sistem yang memiliki mekanisme otentikasi yang kuat, seperti penggunaan username dan password.
- Organisasi dapat mengharapkan sistem yang menerapkan enkripsi untuk melindungi data saat penyimpanan, pengiriman, atau saat beristirahat.
- Organisasi mungkin menginginkan sistem yang memiliki prosedur dan mekanisme yang jelas untuk menangani kejadian keamanan, termasuk deteksi, respons, dan pemulihan dari serangan atau pelanggaran keamanan.

IV.3. Software Quality Attributes

- Memberikan pengalaman pengguna yang baik dan memuaskan.
- Dapat diandalkan dan stabil dalam operasinya. Sistem harus mampu berjalan tanpa gangguan atau kerusakan berkepanjangan, dan mengatasi situasi kesalahan dengan pemulihan yang cepat.
- Memiliki performa tinggi dan responsif terhadap tindakan pengguna.
- Efisien dalam penggunaan sumber daya seperti CPU, memori, dan penyimpanan.
- Mudah dipelihara dan diperbaiki.

BAB V

DATA REQUIREMENTS

V.1. Input

1. Pendaftaran
 - Jumlah pinjaman (Numeric, wajib, validasi)
 - Tujuan pinjaman (String, opsional)
 - Tenor pinjaman (Integer, wajib, validasi)
2. Persyaratan Simpan Pinjam
 - Tidak ada (hanya pilihan menu oleh pengguna)
3. Login
 - Username (String, wajib)
 - Password (String, wajib)
4. Mengajukan Pinjaman
 - Jumlah pinjaman (Numeric, wajib)
 - Tujuan pinjaman (String, opsional)
5. Mengangsur Pinjaman
 - Nomor Pinjaman (String, wajib, validasi harus valid di database)
 - Jumlah angsuran (Numeric, wajib, validasi)
6. Bukti Menyimpan
 - Tidak ada (hanya pilihan menu oleh pengguna)
7. Menginformasikan Pinjaman
 - Nomor nasabah (String, wajib)
8. Mencatat Transaksi
 - Nomor transaksi (String, wajib, unik)
 - Jenis transaksi (String, wajib : “Penyimpanan” atau “Angsuran”)
 - Jumlah transaksi (Numeric, wajib, validasi)
 - Tanggal transaksi (Date, wajib)

V.2. Output

1. Pendaftaran
 - Notifikasi “Pengajuan pinjaman berhasil dikirim dan sedang diproses.”
 - Data pengajuan pinjaman tersimpan di database.
2. Persyaratan Simpan Pinjam
 - KTP
 - KK
 - Slip Gaji
3. Login
 - Jika valid “Login berhasil”
 - Jika tidak valid “Username atau password salah”
4. Mengajukan Pinjaman

- Notifikasi “Pengajuan pinjaman berhasil dikirim dan sedang diproses.”
- Data pengajuan pinjaman tersimpan di database.
- 5. Mengangsur Pinjaman
 - Notifikasi “Pembayaran Berhasil”
 - Data transaksi tercatat di database.
- 6. Bukti Menyimpan
 - Tanggal transaksi
 - Jumlah transaksi
 - Bukti pembayaran
- 7. Menginformasikan Pinjaman
 - Nomor pinjaman
 - Total pinjaman
 - Sisa pinjaman
 - Riwayat angsuran
- 8. Mencatat Transaksi
 - Notifikasi “Transaksi berhasil dicatat”
 - Data transaksi tersimpan di database.

BAB VI

INTERFACE REQUIREMENTS

VI.1. User Interface

Berikut adalah beberapa karakteristik yang umumnya ada dalam interface antara PL (Perangkat Lunak) dan pengguna:

1. User-Friendly (Mudah digunakan): Antarmuka pengguna dirancang dengan prinsip kegunaan dan keterbacaan yang tinggi. Tombol dan elemen navigasi yang jelas dan mudah dipahami. Penggunaan ikon dan simbol yang intuitif untuk mempermudah pemahaman pengguna. Organisasi informasi yang logis dan hierarkis.
2. Responsif: Antarmuka merespons input pengguna dengan cepat dan menyajikan perubahan dengan cepat. Aksi dan transisi yang halus dan lancar. Tampilan yang disesuaikan dengan perangkat yang digunakan, seperti responsif terhadap ukuran layar pada perangkat mobile.
3. Konsisten: Penggunaan elemen visual dan kontrol yang konsisten di seluruh antarmuka. Tata letak, ikon, dan warna yang konsisten untuk menyampaikan arti yang sama. Konsistensi dalam istilah dan frasa yang digunakan untuk menghindari kebingungan pengguna.
4. Intuitif: Antarmuka dirancang sedemikian rupa sehingga pengguna dapat dengan mudah memahami bagaimana menggunakan dan berinteraksi dengan PL. Tindakan dan alur kerja yang terprediksi dengan logika yang jelas. 24 Penggunaan visualisasi dan petunjuk yang membantu pengguna dalam navigasi dan penggunaan fitur-fitur PL.
5. Estetika yang Menarik: Antarmuka memiliki tampilan visual yang menarik dan estetik. Penggunaan warna, font, dan grafik yang sesuai dengan identitas visual PL. Tata letak yang rapi dan penataan elemen yang menyenangkan mata.

VI.2. Hardware Interface

1. Kompatibilitas Hardware:
 - PL harus kompatibel dengan komponen hardware yang digunakan, seperti sistem operasi, arsitektur prosesor, kapasitas penyimpanan, dan perangkat input/output.
 - Pengembangan PL harus mempertimbangkan kebutuhan hardware minimal yang diperlukan agar aplikasi dapat berjalan dengan baik.

2. Interaksi dengan Perangkat Input/Output:
 - PL harus dapat berinteraksi dengan perangkat input/output, seperti keyboard, mouse, monitor, printer, dan perangkat lainnya.
 - Menyediakan antarmuka yang memungkinkan pengguna untuk berinteraksi dengan perangkat tersebut dan mentransfer data dengan lancar.
3. Penanganan Koneksi Jaringan:
 - Jika PL memerlukan akses jaringan, interface harus memungkinkan pengaturan dan manajemen koneksi jaringan.
 - Mendukung protokol jaringan yang relevan dan memastikan keamanan dan privasi data saat berkomunikasi dengan komponen hardware melalui jaringan.
4. Skalabilitas dan Performa:
 - PL harus dirancang untuk mendukung skala operasi yang berbeda, dari perangkat keras dengan spesifikasi rendah hingga infrastruktur jaringan yang kompleks.
 - Mampu mengoptimalkan penggunaan sumber daya hardware yang tersedia dan memberikan performa yang baik dalam berbagai lingkungan.
5. Keamanan dan Perlindungan Hardware:
 - PL harus mempertimbangkan keamanan perangkat keras dan melindungi hardware dari kerusakan atau penyalahgunaan yang disebabkan oleh PL itu sendiri.
 - Interface harus memastikan akses terbatas atau terlindungi terhadap komponen hardware tertentu yang sensitif.

VI.3. Software Interface

1. Sistem Operasi (OS):
 - PL harus kompatibel dengan sistem operasi yang digunakan oleh pengguna, misalnya Windows 10, macOS Catalina, atau Linux Ubuntu 20.04.
 - Interface dengan sistem operasi harus mempertimbangkan fitur dan fungsi yang tersedia, serta kebijakan keamanan dan hak akses yang diberlakukan oleh sistem operasi tersebut.
2. Bahasa Pemrograman dan Framework:
 - PL yang dikembangkan mungkin menggunakan bahasa pemrograman tertentu seperti Python, Java, C#, atau JavaScript.
 - Interface dengan bahasa pemrograman dan framework harus mempertimbangkan dukungan dan kompatibilitas dengan versi spesifik yang digunakan, misalnya Python 3.9 atau Java JDK 11.
3. Database Management System (DBMS):
 - Jika PL membutuhkan penyimpanan data, interface harus memungkinkan koneksi dan interaksi dengan DBMS yang digunakan, seperti MySQL 8.0 atau PostgreSQL 13.0.

- Kompatibilitas dengan SQL (Structured Query Language) dan fitur-fitur khusus DBMS yang digunakan harus dipertimbangkan.
4. Middleware atau Library:
- PL dapat menggunakan middleware atau library tambahan untuk mengakses fitur atau layanan tertentu, seperti pengolahan gambar, komunikasi jaringan, atau analisis data.
 - Interface dengan middleware atau library harus mempertimbangkan integrasi dan ketergantungan dengan versi dan spesifikasi yang tepat, misalnya TensorFlow 2.5 atau Django 3.2.

VI.4. Communication Interface

1. Komunikasi antara Pengguna dan PL:
 - Pengguna harus dapat berinteraksi dengan PL melalui antarmuka pengguna yang disediakan.
 - Komunikasi ini melibatkan input pengguna seperti pengisian formulir, pengaturan preferensi, atau pemilihan opsi melalui elemen antarmuka seperti tombol, menu, dan formulir.
 - Respon dari PL kepada pengguna, seperti pesan kesalahan, notifikasi, atau hasil operasi, juga merupakan bagian dari komunikasi ini.
2. Komunikasi antara PL dan Basis Data:
 - Jika PL menggunakan basis data untuk penyimpanan dan pengambilan data, komunikasi antara PL dan basis data sangat penting.
 - PL harus dapat terhubung dan berkomunikasi dengan basis data untuk melakukan operasi seperti penambahan data, pembaruan, penghapusan, atau pengambilan data.
 - Protokol komunikasi yang digunakan, seperti SQL (Structured Query Language), harus dipahami dan diimplementasikan dengan benar.
3. Komunikasi antara PL dan Layanan Eksternal:
 - PL dapat memerlukan komunikasi dengan layanan eksternal seperti API (Application Programming Interface) pihak ketiga atau layanan web lainnya.
 - Contoh penggunaan termasuk integrasi dengan sistem pembayaran, pengiriman pesan, analisis data, atau sumber daya eksternal lainnya.

- Protokol komunikasi yang digunakan, seperti RESTful API atau SOAP, serta kebijakan keamanan yang diterapkan harus dipertimbangkan.
4. Komunikasi antara Komponen PL:
- PL yang kompleks sering terdiri dari berbagai komponen yang berinteraksi satu sama lain.
 - Komunikasi antara komponen ini penting untuk mentransfer data, pesan, atau instruksi antar komponen.