

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

FOR <<APLIKASI PEMESANAN PADA
CAFE>>.

3411211046 Madya
Dharma
Nusantara3411211069
Rolan Firmansyah

Table Of Contents

Table Of Contents	1
Bab I Introduction	2
1.1 Purpose	2
1.2 Intended Audience and Reading Suggestions	3
1.3 Project Scope	4
1.4 References	5
Bab II Overall Description	0
2.1 Organisations	0
2.2 Product Perspective	1
2.3 User Classes and Characteristics	2
2.4 Operating Environment.....	3
2.5 Design and Implementation Constrains (optional).....	4
2.6 Assumptions and Dependencies (optional).....	5
Bab III Functional Requirements	7
3.1 Detailed Functional Requirements.....	7
3.2 Use Case Diagram	8
3.3 Use Case Scenario	8
Bab IV Non Functional Requirements	10
4.1 Performance Requirements (optional)	10
4.2 Safety Requirements (optional)	10
4.3 Software Quality Attributes (optional).....	10
Bab V Data Requirements	11
4.1 Input.....	11
4.2 Output.....	11
Bab VI Interface Requirements	12
4.1 User Interface	12
4.2 Hardware Interface	14
4.3 Software Interface	14
4.4 Communication Interface	15

Bab I Introduction

1.1 Purpose

<<Jelaskan latar belakang dan tujuan dari dibuatnya dokumen SRS ini, misal menjelaskan kondisi organisasi client, kebutuhan pengembangan organisasi, dan masalah masalah di organisasi saat ini yang diprediksi akan dapat terselesaikan dengan pembangunan PL.>>

Organisasi klien mengoperasikan beberapa kafe dan toko di berbagai lokasi. Saat ini, mereka menghadapi beberapa masalah terkait manajemen karyawan dan sistem penggajian. Proses penghitungan gaji, tunjangan, bonus, dan insentif masih dilakukan secara manual, yang memakan waktu dan dapat rentan terhadap kesalahan. Selain itu, perusahaan ingin meningkatkan pengelolaan karyawan, evaluasi kinerja, dan program pengembangan untuk meningkatkan efisiensi dan produktivitas.

Tujuan Pembuatan Dokumen SRS:

Dokumen SRS (Software Requirements Specification) dibuat untuk:

1. Menyediakan deskripsi yang jelas tentang kebutuhan fungsional dan non-fungsional sistem penggajian dan manajemen karyawan yang akan dikembangkan.
2. Menggambarkan persyaratan bisnis, kebutuhan pengguna, dan skenario penggunaan yang terkait dengan sistem yang diinginkan.
3. Menentukan fitur dan fungsi sistem yang dibutuhkan, seperti perhitungan gaji, tunjangan, bonus, insentif, evaluasi kinerja, dan program pelatihan karyawan.
4. Menggambarkan batasan sistem, termasuk integrasi dengan sistem yang sudah ada, keamanan data, dan kebutuhan lain yang relevan.
5. Memberikan dasar yang jelas bagi tim pengembang untuk merancang, mengembangkan, dan mengimplementasikan sistem yang sesuai dengan kebutuhan organisasi klien.
6. Menghindari ambiguitas dan salah pengertian dalam pengembangan sistem dengan menyediakan panduan yang jelas bagi semua pihak yang terlibat.

Dengan demikian, dokumen SRS ini diharapkan dapat membantu organisasi klien dalam mengatasi masalah yang mereka hadapi saat ini dan meningkatkan manajemen karyawan serta sistem penggajian di kafe dan toko mereka.

1.2 Intended Audience and Reading Suggestions

<<Jelaskan berbagai jenis pembaca yang dimaksudkan untuk dokumen SRS ini, seperti developers, project managers, staf pemasaran, user, tester, dan dokumen writer.>>

Dokumen SRS (Software Requirements Specification) ditujukan untuk berbagai jenis pembaca yang terlibat dalam pengembangan, pengelolaan, pemasaran, penggunaan, dan pengujian sistem yang akan dibangun. Berikut adalah beberapa jenis pembaca yang dimaksudkan untuk dokumen SRS ini:

1. **Pengembang (Developers):** Dokumen SRS memberikan panduan yang jelas kepada pengembang dalam merancang, mengembangkan, dan mengimplementasikan sistem. Mereka menggunakan dokumen ini untuk memahami kebutuhan fungsional dan non-fungsional sistem, arsitektur yang direncanakan, antarmuka pengguna, dan batasan teknis yang perlu dipertimbangkan selama pengembangan.
2. **Manajer Proyek (Project Managers):** Dokumen SRS membantu manajer proyek dalam mengawasi jalannya proyek pengembangan sistem. Mereka menggunakan dokumen ini untuk memastikan bahwa persyaratan proyek dipenuhi, mengelola sumber daya, dan mengoordinasikan tim pengembang.
3. **Staf Pemasaran (Marketing Staff):** Dokumen SRS memberikan pemahaman tentang fitur dan fungsionalitas sistem yang akan dikembangkan. Staf pemasaran menggunakan informasi ini untuk menyusun strategi pemasaran, mengomunikasikan nilai produk kepada calon pelanggan, dan mengidentifikasi keunggulan produk yang dapat ditawarkan kepada pasar.
4. **Pengguna (Users):** Dokumen SRS memberikan gambaran tentang bagaimana sistem akan berinteraksi dengan pengguna. Pengguna menggunakan dokumen ini untuk memahami fitur dan fungsionalitas sistem yang akan digunakan, serta untuk memberikan masukan dan umpan balik terkait kebutuhan penggunaan yang diinginkan.
5. **Penguji (Testers):** Dokumen SRS menyediakan panduan bagi penguji dalam merancang dan menjalankan skenario pengujian. Mereka menggunakan dokumen ini untuk memverifikasi bahwa sistem memenuhi persyaratan yang ditetapkan, mengidentifikasi kasus uji yang relevan, dan melaporkan temuan dalam proses pengujian.
6. **Penulis Dokumen (Document Writers):** Dokumen SRS juga dapat digunakan oleh penulis dokumen untuk menghasilkan materi dokumentasi yang lebih rinci, seperti panduan pengguna, petunjuk instalasi, atau manual teknis. Mereka menggunakan dokumen ini sebagai sumber referensi untuk menyusun dokumen-dokumen tersebut.

Dengan menyesuaikan konten dan bahasa yang digunakan, dokumen SRS dapat disesuaikan dengan kebutuhan masing-masing pembaca untuk memberikan informasi yang relevan dan berguna bagi mereka dalam peran dan tanggung jawab mereka terkait dengan pengembangan dan penggunaan sistem yang akan dibangun.

1.3 Project Scope

<<Jelaskan secara singkat tentang lingkup PL, termasuk manfaat dan tujuan PL.>>

PL (Pengembangan Perangkat Lunak) merujuk pada proses merancang, mengembangkan, menguji, dan memelihara perangkat lunak. Lingkup PL meliputi seluruh siklus hidup perangkat lunak, mulai dari analisis kebutuhan pengguna, perancangan sistem, implementasi, pengujian, hingga pemeliharaan.

Manfaat PL meliputi:

1. Meningkatkan Kualitas: PL memastikan perangkat lunak yang dikembangkan memiliki kualitas yang tinggi dengan mengikuti standar dan praktik terbaik dalam pengembangan perangkat lunak. Proses pengujian dan validasi yang terintegrasi membantu mengidentifikasi dan mengatasi masalah sejak awal.
2. Efisiensi dan Produktivitas: PL memperkenalkan metodologi dan alat yang membantu dalam mengelola proyek perangkat lunak dengan lebih efisien. Proses ini memungkinkan tim pengembang untuk bekerja secara terstruktur, berkolaborasi dengan baik, dan memanfaatkan ulang komponen atau kode yang sudah ada untuk menghemat waktu dan usaha.
3. Kepuasan Pengguna: Dengan memahami kebutuhan pengguna secara mendalam, PL membantu dalam merancang dan mengembangkan perangkat lunak yang sesuai dengan harapan dan kebutuhan pengguna. Fokus pada pengalaman pengguna yang baik membantu menciptakan perangkat lunak yang mudah digunakan dan efektif dalam memecahkan masalah pengguna.
4. Pengendalian Proyek dan Risiko: PL membantu dalam mengendalikan proyek perangkat lunak dengan mengidentifikasi, memperkirakan, dan mengelola risiko yang mungkin timbul selama siklus hidup proyek. Proses ini melibatkan perencanaan, pemantauan, dan pengendalian proyek secara terus-menerus untuk menjaga proyek tetap dalam batas waktu, anggaran, dan lingkup yang ditetapkan.

Tujuan PL meliputi:

1. Menghasilkan Perangkat Lunak yang Berkualitas: Tujuan utama PL adalah mengembangkan perangkat lunak yang memiliki kualitas tinggi, bebas dari bug, dan memenuhi kebutuhan dan harapan pengguna. Hal ini dicapai melalui penerapan metode pengembangan perangkat lunak yang terstruktur, pengujian yang komprehensif, dan validasi menyeluruh.
2. Pemenuhan Kebutuhan Pengguna: PL bertujuan untuk memahami kebutuhan dan harapan pengguna secara mendalam, serta mengembangkan solusi perangkat lunak yang memenuhi kebutuhan tersebut. Dengan menerapkan proses analisis kebutuhan yang baik, PL membantu memastikan bahwa perangkat lunak yang dikembangkan dapat memberikan nilai dan manfaat kepada pengguna.

3. Efisiensi Pengembangan: Salah satu tujuan PL adalah meningkatkan efisiensi dalam pengembangan perangkat lunak. Proses ini melibatkan penggunaan metodologi yang tepat, alat bantu yang sesuai, dan praktik pengembangan yang efektif untuk mengoptimalkan penggunaan sumber daya dan menghemat waktu dalam pengembangan perangkat lunak.
4. Keselamatan dan Keamanan: Tujuan penting dalam PL adalah memastikan bahwa perangkat lunak yang dikembangkan aman dan terlindungi dari ancaman keamanan. Hal ini mencakup perlindungan data pengguna, identifikasi dan mitigasi kerentanan keamanan, serta penerapan praktik pengembangan yang aman.

Dengan mengikuti prinsip dan praktik PL yang baik, organisasi dapat menghasilkan perangkat lunak yang andal, efisien, dan sesuai dengan kebutuhan pengguna, yang pada gilirannya dapat meningkatkan kinerja, efektivitas, dan kepuasan pengguna.

1.4 References

<<List dokumen yang menjadi referensi pada SIS ini.>>

Modul praktikum APPL

Bab II Overall Description

2.1 Organisations

<<Jelaskan tentang organisai usulan, seperti visi, misi, struktur organisasi, dll.>>

Visi:

Visi adalah gambaran masa depan yang diinginkan oleh organisasi. Ini adalah pandangan jangka panjang yang menggambarkan tujuan utama atau arah yang ingin dicapai oleh organisasi. Visi menyediakan landasan untuk menginspirasi dan memandu aktivitas dan keputusan organisasi. Biasanya, visi dirumuskan dalam kata-kata yang menggambarkan keunggulan atau keberhasilan yang ingin dicapai oleh organisasi.

Misi:

Misi adalah pernyataan tujuan dan alasan eksistensi organisasi. Ini menjelaskan tujuan inti organisasi, nilai-nilai yang dianut, dan bagaimana organisasi berkontribusi pada masyarakat atau pelanggan yang dilayani. Misi merangkum tujuan organisasi secara keseluruhan dan menjadi pedoman untuk pengambilan keputusan strategis. Pernyataan misi yang jelas membantu mengarahkan upaya organisasi dalam mencapai tujuannya.

Struktur Organisasi:

Struktur organisasi adalah kerangka atau tata cara di mana tugas, tanggung jawab, dan otoritas terorganisir dalam suatu organisasi. Ini mencakup hubungan antara bagian-bagian organisasi, hierarki pengambilan keputusan, dan aliran komunikasi. Struktur organisasi menentukan bagaimana pekerjaan diorganisir, bagaimana orang-orang saling berinteraksi, dan bagaimana otoritas dan tanggung jawab dipbagikan di seluruh organisasi.

Struktur organisasi dapat bervariasi tergantung pada jenis organisasi dan kompleksitasnya. Beberapa jenis struktur organisasi umum termasuk:

1. Struktur Fungsional: Organisasi dibagi berdasarkan fungsi, seperti pemasaran, keuangan, operasi, dan sumber daya manusia. Setiap departemen memiliki otoritas dan tanggung jawab yang jelas dalam area fungsional mereka.
2. Struktur Divisi: Organisasi dibagi menjadi divisi atau unit bisnis yang mandiri berdasarkan produk, wilayah geografis, atau pasar yang dilayani. Setiap divisi memiliki keputusan mandiri dalam area mereka, dengan tingkat otonomi yang lebih tinggi.
3. Struktur Matriks: Organisasi menggunakan kombinasi struktur fungsional dan struktur divisi. Ini memungkinkan kolaborasi dan koordinasi yang lebih baik antara fungsi dan divisi yang berbeda.
4. Struktur Tim Proyek: Organisasi memilih pendekatan tim proyek di mana tim bersifat sementara dan diberdayakan untuk menyelesaikan proyek tertentu. Ini sering digunakan dalam organisasi yang mengutamakan fleksibilitas dan inovasi.

Selain visi, misi, dan struktur organisasi, organisasi usulan juga dapat mencakup nilai-nilai inti, tujuan strategis, kebijakan, prosedur, dan budaya organisasi. Semua elemen ini membentuk kerangka kerja yang mengarahkan kegiatan organisasi dan membantu mencapai tujuan yang ditetapkan.

2.2 Product Perspective

<<Jelaskan PL dari sudut pandang organisasi dan jelaskan manfaat dirasa dengan dibangunnya PL.>>

PL (Pengembangan Perangkat Lunak) dari sudut pandang organisasi adalah pendekatan terstruktur untuk merancang, mengembangkan, menguji, dan memelihara perangkat lunak yang dibutuhkan oleh organisasi. PL melibatkan proses yang disiplin, metode pengembangan yang terorganisir, dan penggunaan alat dan praktik terbaik untuk mencapai tujuan organisasi dalam menghasilkan perangkat lunak berkualitas tinggi.

Manfaat yang dirasakan dengan dibangunnya PL dalam organisasi adalah sebagai berikut:

1. Pemenuhan Kebutuhan Organisasi: Dengan adanya PL, organisasi dapat mengidentifikasi dan memahami kebutuhan bisnis yang spesifik. Proses analisis kebutuhan yang baik membantu dalam merumuskan persyaratan perangkat lunak yang tepat. Hal ini memastikan bahwa perangkat lunak yang dikembangkan secara akurat memenuhi kebutuhan organisasi, membantu meningkatkan efisiensi operasional, memperbaiki proses bisnis, atau mengatasi tantangan bisnis yang ada.
2. Pengendalian dan Manajemen Proyek yang Lebih Baik: PL memberikan kerangka kerja yang terstruktur dan terorganisir untuk mengelola proyek pengembangan perangkat lunak. Dengan adanya metode pengembangan yang terdefinisi dengan baik, penjadwalan yang jelas, alokasi sumber daya yang tepat, dan pemantauan progres proyek, organisasi dapat mengendalikan proyek dengan lebih baik. Hal ini membantu dalam mengelola risiko, menghindari penundaan, dan memastikan proyek berjalan sesuai dengan anggaran dan jadwal yang ditentukan.
3. Peningkatan Kualitas Perangkat Lunak: PL mempromosikan pendekatan berbasis kualitas dalam pengembangan perangkat lunak. Dengan adanya proses pengujian yang terstruktur dan validasi yang cermat, organisasi dapat menghasilkan perangkat lunak yang lebih andal, bebas dari bug, dan memenuhi standar kualitas yang ditetapkan. Peningkatan kualitas perangkat lunak ini berdampak pada kepuasan pengguna yang lebih tinggi, pengurangan biaya perbaikan, dan peningkatan reputasi organisasi.
4. Pengembangan yang Lebih Efisien dan Efektif: PL membantu organisasi mengadopsi metode pengembangan perangkat lunak yang efisien dan efektif. Dengan menggunakan praktik terbaik, alat bantu, dan penggunaan kembali komponen atau kode yang sudah ada, organisasi dapat menghemat waktu, usaha, dan sumber daya dalam pengembangan perangkat lunak. Ini juga memfasilitasi komunikasi yang lebih baik antara anggota tim pengembang, kolaborasi yang efektif, dan peningkatan produktivitas.
5. Pemeliharaan dan Peningkatan yang Lebih Mudah: Dalam PL, pemeliharaan perangkat lunak menjadi fokus penting. Dengan adanya dokumentasi yang baik, penggunaan kode yang terstruktur, dan pembaruan atau perbaikan yang direncanakan dengan baik, organisasi dapat melakukan pemeliharaan dan peningkatan perangkat lunak dengan lebih mudah dan efisien. Hal ini memastikan kelangsungan fungsionalitas perangkat lunak, keamanan, dan pemenuhan kebutuhan jangka panjang organisasi.

Secara keseluruhan, PL membantu organisasi dalam mencapai tujuan bisnis dan memenuhi kebutuhan mereka dalam pengembangan perangkat lunak. Manfaat yang dirasakan termasuk pemenuhan kebutuhan organisasi yang lebih baik, pengendalian dan manajemen proyek yang lebih baik, peningkatan kualitas perangkat lunak, efisiensi pengembangan yang lebih tinggi, dan kemampuan

pemeliharaan dan peningkatan yang lebih baik. Dengan membangun PL yang solid, organisasi dapat meningkatkan efektivitas, efisiensi, dan kesuksesan proyek pengembangan perangkat lunak mereka.

2.3 User Classes and Characteristics

<<Jelaskan user yang akan terlibat dalam PL, karakteristik, hak akses, dll.>>

Dalam PL (Pengembangan Perangkat Lunak), ada beberapa jenis pengguna yang akan terlibat. Berikut adalah beberapa jenis pengguna yang umumnya terlibat dalam PL:

1. **Pengguna Akhir (End Users):** Pengguna akhir adalah individu atau kelompok yang akan menggunakan perangkat lunak yang dikembangkan. Mereka adalah pihak yang akan berinteraksi langsung dengan sistem dan mengambil manfaat dari fitur dan fungsionalitas yang disediakan. Pengguna akhir memiliki kepentingan langsung dalam kualitas, kegunaan, dan kehandalan perangkat lunak.
2. **Analisis Bisnis (Business Analysts):** Analisis bisnis bertanggung jawab untuk memahami kebutuhan bisnis dan merumuskan persyaratan fungsional dan non-fungsional untuk perangkat lunak. Mereka berinteraksi dengan pengguna akhir dan pemangku kepentingan lainnya untuk mengumpulkan informasi, menganalisis proses bisnis, dan mengkomunikasikan kebutuhan pengguna kepada tim pengembang.
3. **Pengembang (Developers):** Pengembang adalah tim yang bertanggung jawab merancang, mengembangkan, dan mengimplementasikan perangkat lunak. Mereka menggunakan persyaratan dan spesifikasi yang diberikan oleh analisis bisnis untuk membangun solusi perangkat lunak yang memenuhi kebutuhan pengguna. Pengembang juga terlibat dalam pengujian, debugging, dan pemeliharaan perangkat lunak.
4. **Penguji (Testers):** Penguji bertanggung jawab untuk melakukan pengujian perangkat lunak untuk memastikan bahwa perangkat lunak berfungsi dengan baik dan memenuhi persyaratan yang ditetapkan. Mereka merancang skenario pengujian, menjalankan tes, dan melaporkan temuan kepada tim pengembang. Penguji membantu menjamin kualitas perangkat lunak sebelum diluncurkan kepada pengguna akhir.
5. **Pemangku Kepentingan (Stakeholders):** Pemangku kepentingan adalah individu atau kelompok yang memiliki kepentingan atau pengaruh terhadap proyek PL. Mereka dapat mencakup manajemen senior, pemilik bisnis, departemen fungsional terkait, atau pihak lain yang memiliki kepentingan terhadap kesuksesan perangkat lunak. Pemangku kepentingan berkontribusi dalam menetapkan prioritas, memberikan perspektif bisnis, dan memastikan perangkat lunak memenuhi harapan mereka.

Karakteristik dan hak akses pengguna dalam PL dapat bervariasi tergantung pada peran dan tanggung jawab mereka. Beberapa karakteristik umum yang relevan dengan pengguna dalam PL meliputi:

- **Pengetahuan Domain:** Pengguna harus memahami proses bisnis yang terkait dengan perangkat lunak yang dikembangkan.
- **Kemampuan Komunikasi:** Pengguna harus dapat berkomunikasi dengan jelas dan efektif dengan tim pengembang dan analisis bisnis.
- **Pengertian Teknologi:** Pengguna harus memiliki pemahaman tentang teknologi yang terkait dengan perangkat lunak, sehingga mereka dapat memberikan wawasan yang berharga dalam pengembangan.

- Kesabaran dan Fleksibilitas: Pengguna harus memiliki kesabaran dan kesiapan untuk beradaptasi dengan perubahan dan iterasi dalam pengembangan perangkat lunak.

Setiap pengguna memiliki hak akses yang sesuai dengan perannya. Hak akses mungkin meliputi akses ke dokumentasi persyaratan, akses ke lingkungan pengembangan dan pengujian, kemampuan untuk memberikan masukan dan umpan balik, serta hak untuk menguji dan menggunakan perangkat lunak yang dikembangkan.

Dalam PL, kolaborasi dan komunikasi yang efektif antara pengguna, analis bisnis, pengembang, penguji, dan pemangku kepentingan lainnya penting untuk mencapai tujuan yang diinginkan dalam pengembangan perangkat lunak yang berhasil.

2.4 Operating Environment

<<Jelaskan environment dari PL, termasuk hardware platform, operating system and version, dan komponen software lainnya yang mendukung PL.>>

Environment dalam PL (Pengembangan Perangkat Lunak) merujuk pada lingkungan teknis yang diperlukan untuk mendukung pengembangan, pengujian, dan peluncuran perangkat lunak. Berikut adalah komponen umum dari environment PL:

1. Hardware Platform: Hardware platform merujuk pada perangkat keras yang digunakan dalam pengembangan perangkat lunak. Ini mencakup komputer, server, jaringan, dan perangkat keras lainnya yang digunakan oleh tim pengembang. Hardware platform yang digunakan dapat bervariasi tergantung pada skala proyek, kompleksitas perangkat lunak, dan persyaratan khusus.
2. Sistem Operasi (Operating System) dan Versi: Sistem operasi adalah perangkat lunak dasar yang mengelola sumber daya perangkat keras dan menyediakan layanan untuk perangkat lunak aplikasi. Pilihan sistem operasi yang digunakan dalam PL tergantung pada preferensi tim pengembang dan kebutuhan perangkat lunak yang dikembangkan. Contoh sistem operasi yang umum digunakan adalah Windows, macOS, Linux, dan berbagai varian mereka. Selain itu, versi spesifik dari sistem operasi juga penting untuk dipertimbangkan, karena perangkat lunak mungkin memerlukan fitur atau perubahan tertentu yang hanya tersedia dalam versi tertentu.
3. Lingkungan Pengembangan Terpadu (Integrated Development Environment/IDE): IDE adalah perangkat lunak yang menyediakan lingkungan terpadu untuk pengembangan perangkat lunak. Ini mencakup alat-alat untuk menulis, menguji, dan mengelola kode. Contoh IDE yang populer termasuk Visual Studio, Eclipse, Xcode, dan PyCharm. IDE biasanya mendukung bahasa pemrograman tertentu dan memfasilitasi proses pengembangan dengan fitur seperti penyorotan sintaksis, pengkodean otomatis, debugging, dan integrasi dengan sistem kontrol versi.
4. Bahasa Pemrograman: Bahasa pemrograman digunakan untuk mengembangkan perangkat lunak. Pilihan bahasa pemrograman tergantung pada kebutuhan proyek, preferensi tim pengembang, dan karakteristik perangkat lunak yang akan dikembangkan. Contoh bahasa pemrograman yang umum digunakan meliputi Java, C++, Python, JavaScript, dan banyak lagi.
5. Basis Data (Database) dan Sistem Manajemen Basis Data (Database Management System/DBMS): Basis data digunakan untuk menyimpan, mengelola, dan mengakses data dalam perangkat lunak. DBMS adalah perangkat lunak yang menyediakan antarmuka untuk mengelola dan mengakses basis data. Pilihan basis data dan DBMS tergantung pada jenis data yang akan disimpan dan

persyaratan perangkat lunak. Contoh basis data dan DBMS yang umum digunakan termasuk MySQL, Oracle, Microsoft SQL Server, dan PostgreSQL.

6. **Alat Pengujian (Testing Tools):** Alat pengujian digunakan untuk menguji perangkat lunak dan memverifikasi kecocokan dengan persyaratan yang ditetapkan. Ini mencakup alat untuk pengujian unit, pengujian fungsional, pengujian kinerja, dan pengujian keamanan. Contoh alat pengujian yang umum digunakan termasuk Selenium, JUnit, Postman, dan Apache JMeter.

Selain komponen di atas, ada juga komponen perangkat lunak lainnya yang dapat mendukung PL seperti sistem kontrol versi (version control systems), alat manajemen proyek (project management tools), alat kolaborasi tim, alat dokumentasi, dan alat pemantauan kualitas perangkat lunak.

Pemilihan komponen hardware, sistem operasi, bahasa pemrograman, dan alat pengembangan serta pengujian tergantung pada persyaratan proyek, preferensi tim pengembang, skala proyek, dan konteks organisasi yang terlibat dalam PL.

2.5 Design and Implementation Constrains (optional)

<<Jelaskan Batasan dari desain dan implementasi PL.>>

Dalam desain dan implementasi PL (Pengembangan Perangkat Lunak), terdapat beberapa batasan yang perlu dipertimbangkan. Berikut adalah beberapa batasan umum yang sering terjadi:

1. **Batasan Waktu:** Waktu adalah faktor kunci dalam desain dan implementasi PL. Terdapat batasan waktu yang harus dipatuhi untuk menyelesaikan pengembangan perangkat lunak. Hal ini dapat disebabkan oleh tenggat waktu proyek yang telah ditetapkan, kebutuhan bisnis yang mendesak, atau faktor lain yang mempengaruhi jadwal pengembangan.
2. **Batasan Sumber Daya:** Sumber daya termasuk anggaran, personel, perangkat keras, perangkat lunak, dan infrastruktur yang tersedia untuk pengembangan perangkat lunak. Batasan sumber daya dapat mempengaruhi lingkup dan kompleksitas proyek, serta kemampuan untuk mengimplementasikan fitur atau fungsionalitas tertentu.
3. **Batasan Teknis:** Terdapat batasan teknis yang perlu dipertimbangkan dalam desain dan implementasi PL. Ini meliputi batasan teknis pada platform perangkat keras, sistem operasi, bahasa pemrograman, dan teknologi yang digunakan. Selain itu, batasan teknis juga dapat berkaitan dengan kemampuan alat pengembangan, integrasi dengan sistem lain, atau batasan keamanan dan privasi.
4. **Batasan Lingkungan:** Lingkungan di mana perangkat lunak akan diimplementasikan juga dapat mempengaruhi desain dan implementasi PL. Faktor-faktor seperti konektivitas jaringan, infrastruktur IT yang ada, peraturan keamanan, atau kebutuhan integrasi dengan sistem yang ada dapat membatasi desain dan implementasi perangkat lunak.
5. **Batasan Fungsional:** Batasan fungsional mengacu pada batasan pada fitur atau fungsionalitas perangkat lunak yang akan dikembangkan. Ini dapat disebabkan oleh kebutuhan bisnis yang spesifik, kebijakan atau peraturan industri, atau kendala teknis yang mempengaruhi implementasi fitur tertentu.
6. **Batasan Keamanan dan Privasi:** Dalam beberapa kasus, ada batasan keamanan dan privasi yang harus dipertimbangkan dalam desain dan implementasi perangkat lunak. Ini termasuk kebutuhan untuk mematuhi standar keamanan data, melindungi informasi sensitif, atau menerapkan

mekanisme keamanan tertentu untuk melindungi perangkat lunak dari ancaman yang mungkin ada.

7. Batasan Skala dan Pertumbuhan: Batasan ini berkaitan dengan kapasitas perangkat lunak untuk menangani volume data atau pengguna yang besar, serta kemampuan perangkat lunak untuk berkembang seiring waktu. Dalam desain dan implementasi, harus dipertimbangkan kemampuan perangkat lunak untuk skala dan pertumbuhan yang diantisipasi.

Penting untuk mengidentifikasi dan memahami batasan-batasan ini saat merancang dan mengimplementasikan PL. Dengan mempertimbangkan batasan-batasan ini secara proaktif, tim pengembang dapat mengelola ekspektasi, merencanakan dengan baik, dan menghasilkan perangkat lunak yang sesuai dengan kebutuhan dan ketersediaan sumber daya.

2.6 Assumptions and Dependencies (optional)

<<Jelaskan semua factor yang dianggap berlawanan dengan fakta yang diketahui yang dapat mempengaruhi requirements yang dinyatakan dalam SRS ini.>>

Beberapa faktor yang dianggap berlawanan dengan fakta yang diketahui yang dapat mempengaruhi persyaratan yang dinyatakan dalam SRS (Software Requirements Specification) adalah:

1. Kendala Anggaran: Kendala anggaran yang ketat dapat membatasi kemampuan untuk mengimplementasikan persyaratan yang ada dalam SRS. Jika anggaran terbatas, beberapa persyaratan mungkin perlu disesuaikan, diprioritaskan ulang, atau bahkan dihilangkan untuk memenuhi batasan anggaran yang ada.
2. Kendala Waktu: Ketika ada batasan waktu yang ketat, mungkin tidak cukup waktu untuk mengimplementasikan semua persyaratan yang telah ditetapkan dalam SRS. Dalam situasi ini, beberapa persyaratan mungkin perlu ditunda atau dimodifikasi agar sesuai dengan jadwal yang ada.
3. Kendala Sumber Daya: Keterbatasan sumber daya seperti personel, perangkat keras, perangkat lunak, atau infrastruktur IT dapat mempengaruhi implementasi persyaratan. Jika sumber daya terbatas, mungkin perlu dilakukan penyesuaian atau perubahan dalam persyaratan yang telah ditetapkan.
4. Teknologi Tersedia: Faktor teknis seperti ketersediaan teknologi atau keterbatasan platform perangkat keras dan perangkat lunak dapat mempengaruhi persyaratan. Jika teknologi yang diperlukan tidak tersedia atau tidak kompatibel dengan lingkungan yang ada, persyaratan mungkin perlu disesuaikan atau alternatif solusi teknis harus dipertimbangkan.
5. Kebijakan atau Peraturan: Persyaratan dalam SRS juga harus mempertimbangkan kebijakan atau peraturan yang berlaku dalam industri atau organisasi. Jika ada peraturan baru atau perubahan kebijakan yang mempengaruhi persyaratan, maka persyaratan mungkin perlu disesuaikan atau diubah sesuai dengan kepatuhan yang diperlukan.
6. Kendala Lingkungan: Lingkungan tempat perangkat lunak akan diimplementasikan juga dapat mempengaruhi persyaratan. Kendala lingkungan seperti konektivitas jaringan, infrastruktur yang ada, atau persyaratan keamanan khusus dapat memerlukan penyesuaian atau perubahan dalam persyaratan yang telah ditetapkan.
7. Perubahan Kebutuhan Pengguna: Kebutuhan pengguna dapat berubah seiring waktu atau mungkin berbeda dari yang awalnya diketahui. Perubahan ini dapat mempengaruhi persyaratan

yang dinyatakan dalam SRS. Jika ada perubahan signifikan dalam kebutuhan pengguna, persyaratan mungkin perlu direvisi atau diperbarui agar tetap relevan dengan kebutuhan aktual.

Penting untuk mempertimbangkan faktor-faktor ini dan memiliki fleksibilitas dalam mengelola persyaratan yang dinyatakan dalam SRS. Jika ada perubahan atau kendala yang mempengaruhi persyaratan, perlu dilakukan manajemen perubahan yang tepat untuk memastikan bahwa persyaratan tetap sesuai dengan kebutuhan dan ketersediaan yang ada.

Bab III Functional Requirements

Kebutuhan Fungsional adalah kebutuhan yang berisi proses-proses apa saja / layanan apa saja yang nantinya harus disediakan oleh PL, mencakup bagaimana sistem harus bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu.

3.1 Detailed Functional Requirements

1. Manajemen Pengguna:

PL harus memiliki kemampuan untuk mendaftarkan pengguna baru, mengelola hak akses, dan mengatur pengguna ke dalam peran atau grup yang sesuai.

Pengguna harus dapat melakukan autentikasi ke PL menggunakan mekanisme seperti username dan password, autentikasi sosial, atau otentikasi berbasis token.

PL harus menyediakan fitur pemulihan kata sandi jika pengguna lupa kata sandi mereka.

2. Pemesanan dan Transaksi:

PL harus memungkinkan pengguna untuk melakukan pemesanan produk/jasa yang tersedia.

Pengguna harus dapat memasukkan rincian pemesanan seperti jumlah, tanggal/jam,

3. Penjual

Menerima pesanan

Membuat surat pesanan

Memproses pesanan

Mengirim pesanan

4. Pembeli

Membuat pesanan

Menerima surat pesanan

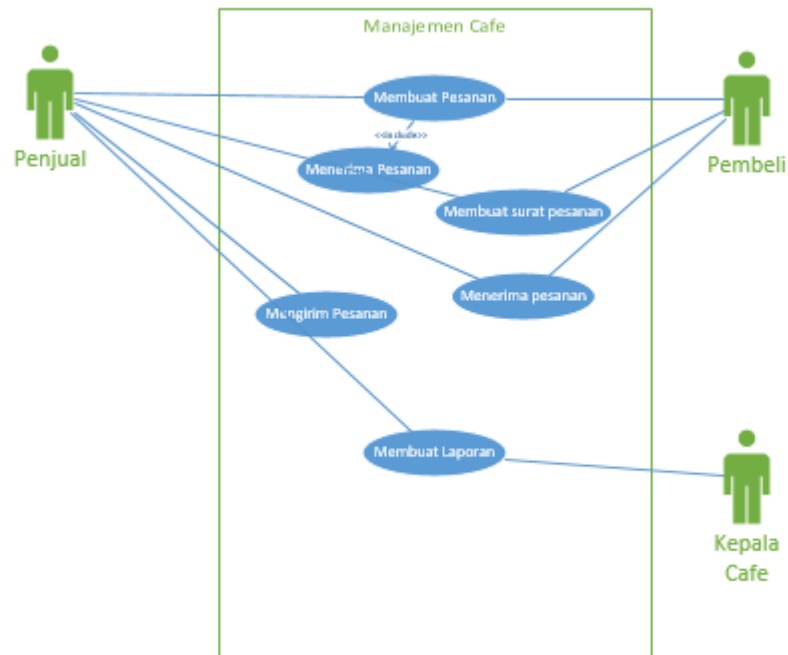
Menerima pesanan

5. Kepala cafe

Menerima laporan

3.2 Use Case Diagram

<<Berikan gambaran kebutuhan fungsionalitas PL dengan menggunakan use case diagram>>



3.3 Use Case Scenario

Penjual	Pembeli	Sistem
	1. Memilih Barang	2. Menampilkan pemilihan barang
	3. Memilih pesanan	
		4. Sistem menyimpan pesanan
5. Penjual menerima pesanan		
6. Membuat surat pesanan		
		7. Sistem mencatatkan selesai pesanan
	8. Menerima surat pesanan	
9. Menerima surat pesanan		
10. . Mengirim pesanan		

		11. Menyelesaikan pesanan
	12. Menerima pesanan	
		13. Pesanan selesai

Bab IV Non Functional Requirements

Kebutuhan Non Fungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. kebutuhan fungsional juga sering disebut sebagai batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dan lain lain. Contoh :

- Menggunakan SO Windows NT
- Spesifikasi komputer Minimal pentium II
- Kebutuhan RAM 256
- Kebutuhan Hardisk 10 GB
- Printer
- Sistem aplikasi dan database dilengkapi dengan password.
- Dilengkapi dengan CCTV diruang baca dan ruang penyimpanan tas.
- Dibutuhkan pemberitahuan bila terjadi kesalahan dalam pengisian form
- Memiliki antarmuka yang mudah dipahami
- waktu peminjaman buku dibatasi 3 menit
- Waktu pengembalian buku dibatasi 2 menit

4.1 Performance Requirements (optional)

1. Responsif dan Cepat:

- Organisasi menginginkan PL yang responsif dan cepat dalam menjalankan tugas-tugasnya.
- Pengguna mengharapkan waktu respons yang singkat saat berinteraksi dengan PL, seperti saat melakukan pencarian, memuat halaman, atau mengirimkan data.
- Performa yang baik akan meningkatkan pengalaman pengguna dan efisiensi operasional.

2. Kepatuhan dan Keamanan:

- Organisasi membutuhkan performa yang mendukung kepatuhan dan keamanan.
- PL harus mematuhi persyaratan kepatuhan yang relevan (misalnya, GDPR, HIPAA) dan memiliki mekanisme keamanan yang kuat untuk melindungi data pengguna dan sistem dari ancaman keamanan.

4.2 Safety Requirements (optional)

1. Keamanan Autentikasi Pengguna:

- Aplikasi harus memiliki mekanisme autentikasi yang kuat, seperti username dan password, autentikasi dua faktor, atau integrasi dengan layanan autentikasi pihak ketiga.
- Ini memastikan bahwa hanya pengguna yang sah yang dapat mengakses aplikasi dan data sensitif.

2. Keamanan Data Pelanggan:

Aplikasi harus melindungi data pelanggan yang sensitif, seperti informasi pribadi, detail pembayaran, atau riwayat pesanan.

4.3 Software Quality Attributes (optional)

Bab V Data Requirements

Deskripsikan dengan rinci setiap data yang dibutuhkan oleh pengguna.

4.1 Input

Atribut data :

Nama pemesan :

No pesanan :

Nama barang :

Alamat :

Nama penjual :

CRUD

Penjual

Buat: Penjual dapat membuat data

Baca: Penjual dapat membaca data

Perbarui: Penjual dapat memperbarui data

Hapus : Penjual dapat menghapus data

Pembeli

Pilih : Pembeli dapat membuat atau memilih barang

Baca : Pembeli dapat membaca pesanan

Perbarui : Pembeli dapat memperbarui pesanan

4.2 Output

Aplikasi pemesanan pada cafe dapat menghasilkan keluaran, terutama dalam bentuk laporan yang berguna untuk melacak dan menganalisis informasi terkait jual beli pada cafe. Berikut ini adalah detail keluaran yang mungkin dihasilkan oleh

aplikasi tersebut:

Laporan cafe :

Laporan jual beli pada cafe

Bab VI Interface Requirements

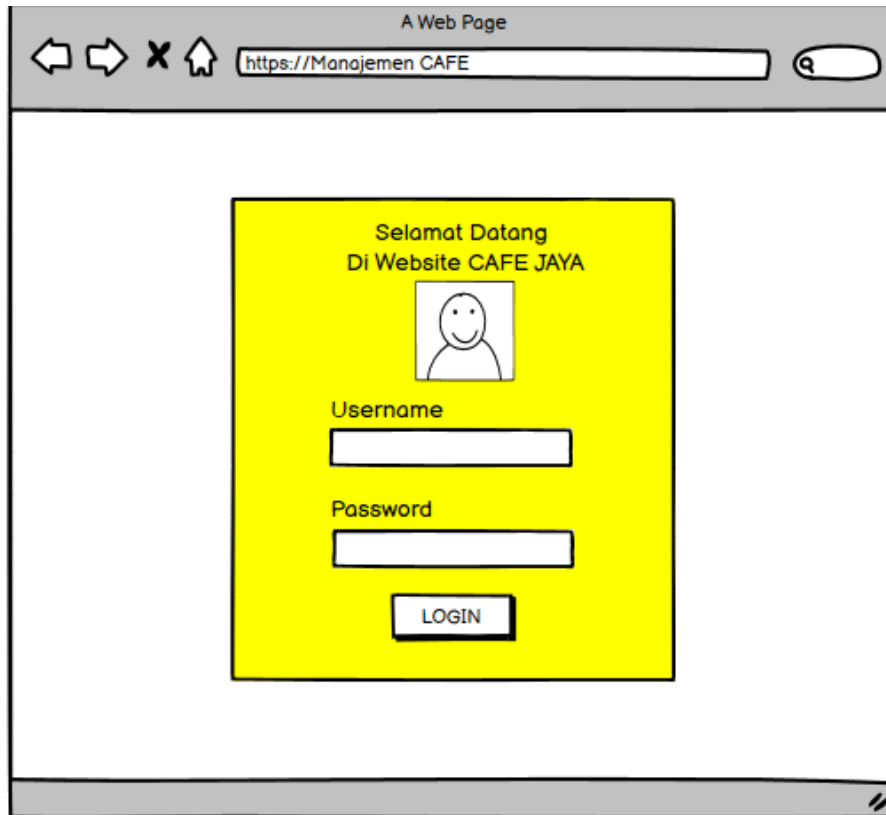
Deskripsikan dengan rinci setiap kebutuhan antarmuka yang dibutuhkan oleh pengguna.

4.1 User Interface

<<Jelaskan karakteristik dari setiap interface antara PL dengan pengguna, jika bisa lengkapi dengan gambar interface.>>

1. Antarmuka Login:

Fungsi: Memungkinkan pengguna untuk masuk ke dalam sistem dengan mengotentikasi identitas mereka.



2. Antarmuka Menu:

Fungsi: Memungkinkan pengguna untuk mengelola menu kafe.

Karakteristik: Biasanya berisi daftar menu dengan informasi seperti nama item,

Menu

MENU

☐
Coffe

☐
Kue

☐
Makanan cepat saji

3. Antarmuka Pemesanan/pembayaran

Fungsi: Memungkinkan pengguna untuk menerima dan mengelola pesanan dari pelanggan.

A Web Page

⬅ ➡ ✕ 🏠

🔍

PEMBAYARAN

Pembayaran

Pembayaran

Rp.

Pesanan	Tanggal Bayar	harga
Giacomo Guilizzoni Founder & CEO	22/09/2003	1.000.000

4.2 Hardware Interface

1. Antarmuka Login:

Fungsi: Memungkinkan pengguna untuk masuk ke dalam sistem dengan mengotentikasi identitas mereka.

Karakteristik: Biasanya terdiri dari kolom input untuk username dan password, serta tombol "Masuk" untuk mengakses akun pengguna

2. Antarmuka Menu:

Fungsi: Memungkinkan pengguna untuk mengelola menu kafe.

Karakteristik: Biasanya berisi daftar menu dengan informasi seperti nama item, deskripsi, harga, dan gambar makanan/minuman

3. Antarmuka Pemesanan:

Fungsi: Memungkinkan pengguna untuk menerima dan mengelola pesanan dari pelanggan.

Karakteristik: Biasanya terdiri dari daftar pesanan yang masuk dengan rincian seperti nomor pesanan, menu yang dipesan, jumlah, dan status pesanan

4.3 Software Interface

<<Jelaskan karakteristik dari setiap interface antara PL dengan komponen software (lengkapi dengan nama dan versi).>>

1. Antarmuka Pemrograman Aplikasi (Application Programming Interface/API):

Fungsi: Mengizinkan komunikasi antara dua perangkat lunak.

Karakteristik: API menyediakan sekumpulan protokol, aturan, dan alat yang memungkinkan aplikasi berinteraksi dengan komponen perangkat lunak lainnya. Beberapa contoh API yang populer adalah:

RESTful API (Representational State Transfer): Misalnya, Django REST Framework (versi 3.x), Express.js (versi 4.x).

SOAP API (Simple Object Access Protocol): Misalnya, Apache Axis (versi 1.x), Microsoft WCF (Windows Communication Foundation).

2. Antarmuka Penghubung Basis Data (Database Interface):

Fungsi: Menghubungkan perangkat lunak dengan sistem basis data untuk mengakses, mengelola, dan memanipulasi data.

Karakteristik: Antarmuka ini biasanya bergantung pada jenis basis data yang digunakan. Contoh antarmuka basis data yang populer adalah:

JDBC (Java Database Connectivity): Misalnya, JDBC 4.3 untuk Java 8.

ODBC (Open Database Connectivity): Misalnya, ODBC 3.8 untuk Microsoft Windows.

3. Antarmuka Pengguna Grafis (Graphical User Interface/GUI):

Fungsi: Menyediakan tampilan visual dan interaksi dengan pengguna.

Karakteristik: GUI berbeda tergantung pada teknologi dan platform yang digunakan. Beberapa contoh antarmuka GUI yang umum adalah:

JavaFX: Misalnya, JavaFX 17.

4.4 Communication Interface

1. Komunikasi antara pengguna dan PL:

Pengguna perlu dapat berinteraksi dengan PL melalui antarmuka yang mudah dipahami dan intuitif.

PL harus menerima input dari pengguna, seperti perintah, data, preferensi, dan instruksi lainnya.

PL harus memberikan umpan balik kepada pengguna, termasuk pesan kesalahan, notifikasi, atau hasil operasi yang dijalankan.

2. Komunikasi antara PL dan komponen perangkat keras:

PL mungkin perlu berkomunikasi dengan perangkat keras eksternal, seperti sensor, printer, atau perangkat lainnya.

PL harus dapat membaca dan menulis data ke perangkat keras serta menerima umpan balik dari mereka.

Komunikasi ini dapat melibatkan protokol khusus, antarmuka perangkat keras, atau API yang disediakan oleh pihak ketiga.

3. Komunikasi antara komponen internal PL:

PL yang kompleks sering terdiri dari beberapa komponen internal yang saling berinteraksi.

Komunikasi ini dapat terjadi melalui antarmuka, API internal, panggilan fungsi, pertukaran pesan, atau pemrosesan acara (event processing).

Tujuannya adalah untuk mentransfer data, koordinasi tugas, atau menghubungkan komponen untuk menjalankan fungsionalitas yang diperlukan.