# **1. Introduction**

## **1.1 Purpose**

Dokumen ini berisi Spesifikasi Kebutuhan Perangkat Lunak (SKPL) atau System Requirement Spesification (SRS) untuk aplikasi Interlibary Loan (ILL). Tujuan penulisan dokumen ini adalah memberikan penjelasan mengenai hasil analisis perangkat lunak. Berupa gambaran umum maupun penjelasan detail dan menyeluruh.

Dokumen ini dapat digunakan sebagai referensi pengembangan oleh pengembang perangkat lunak ILL. Ataupun berbagai pihak yang berkaitan dengan pengembangan aplikasi ini. Dengan adanya dokumen ini, diharapkan pengembangan perangkat lunak dapat lebih terarah dan terfokus serta tidak menimbulkan multitafsir.

## **1.2 Document Conventions**

1. SKPL : Spesifikasi Kebutuhan Perangkat Lunak
2. SRS : System Requirement Spesification
3. ILL : Interlibrary Loan
4. DBMS : Database Management System
5. PHP : PHP Hypertext Preprocessor
6. Laravel : Kerangka kerja dari PHP
7. MySQL : Sebuah perangkat lunak sistem manajemen basis data SQL
8. Apache : Sebuah nama web server yang bertanggung jawab pada request-response HTTP dan logging informasi

## **1.3 Intended Audience and Reading Suggestions**

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## **1.4 Product Scope**

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## **1.5 References**

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# **2. Overall Description**

## **2.1 Product Perspective**

Interlibrary Loan (ILL) adalah aplikasi yang bertujuan untuk menciptakan integrasi pada ruang baca yang ada di seluruh departemen di Institut Teknologi Sepuluh Nopember. Aplikasi ini memungkinkan peminjaman buku antar departemen menjadi lebih mudah.

Aplikasi ILL nantinya akan berjalan dengan basis website. Dengan adanya aplikasi ini, diharapkan tidak ada lagi gap antar ruang baca di ITS. Dan juga setiap ruang baca dapat berbagi data dan terintegrasi. Hal ini diharapkan menjadi solusi untuk memberikan kemudahan bagi civitas akademika ITS untuk memperluas wawasannya.

## **2.2 Product Functions**

1. Melihat dan mencari berbagai data koleksi buku yang terdapat pada ruang baca di setiap departemen
2. Membaca e-book
3. Mengunduh e-book
4. Mendaftar sebagai anggota
5. Mengajukan peminjaman buku
6. Mengorganisir data buku
7. Mengorganisir data anggota
8. Mengorganisir data historis peminjaman
9. Mengorganisir data perpustakaan anggota ILL

## **2.3 User Classes and Characteristics**

User yang akan menggunakan sistem ini diantaranya:

1. Peminjam buku : Mahasiswa, dosen, karyawan, dan seluruh civitas akademika di ITS

* Melihat dan mencari berbagai data koleksi buku yang terdapat pada ruang baca di setiap departemen
* Membaca e-book
* Mengunduh e-book
* Mendaftar sebagai anggota
* Mengajukan peminjaman buku

1. Pustakawan : Seluruh karyawan yang memegang wewenang di ruang baca seluruh ITS

* Mengorganisir data buku
* Mengorganisir data anggota
* Mengorganisir data historis peminjaman
* Mengorganisir data perpustakaan anggota ILL

## **2.4 Operating Environment**

Sistem akan berjalan dengan baik pada server dengan spesifikasi minimal sebagai berikut:

1. Processor 3GHz
2. 1 GB RAM
3. 25GB HDD
4. 4MB/s bandwidth

Sementara itu di sisi perangkat lunak, aplikasi ini dibangun dengan bahasa pemrograman PHP. Yang dimana untuk memudahkan pengembangan, aplikasi ini berjalan pada kerangka kerja Laravel 5.6. Untuk penyimpanan data, aplikasi ini menggunakan DBMS MySQL. Sedangkan web servernya, akan menggunakan Apache.

## **2.5 Design and Implementation Constraints**

Terdapat beberapa hal yang dapat menghambat realisasi ILL, diantaranya:

1. Masih ada ruang baca yang tidak menghendaki untuk open data
2. Setiap ruang baca memiliki SOP peminjaman yang berbeda-beda
3. Adanya interfensi dari Perpustakaan Pusat ITS
4. Koneksi internet di beberapa departemen masih tidak stabil

## **2.6 User Documentation**

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## **2.7 Assumptions and Dependencies**

1. Semua ruang baca memiliki SOP yang sama
2. Semua ruang baca menghendaki open data
3. Semua ruang baca menyetujui format pencatatan historis peminjaman
4. Setidaknya terdapat sebuah komputer yang memiliki koneksi internet stabil di setiap ruang baca

# **3. External Interface Requirements**

## **3.1 User Interfaces**

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## **3.2 Hardware Interfaces**

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## **3.3 Software Interfaces**

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## **3.4 Communications Interfaces**

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# **4. System Features**

Fitur dalam Inter Library Loan ini akan banyak melibatkan Perpustakaan/ Ruang Baca yang ada sekaligus orang yang berada didalamnya. Sistem ini sendiri memilki banyak fitur yaitu:

1. Melihat koleksi buku
2. Mendaftar sebagai anggota
3. Melihat status dan deskripsi buku yang ingin dipinjam
4. Melakukan peminjaman/pemesanan buku
5. Melihat status peminjaman buku
6. Mencari buku yang ingin dipinjam
7. Melihat history peminjaman buku
8. Mengelola data buku
9. Melihat daftar peminjam buku
10. Mengelola permintaan peminjaman buku
11. Melayani pengembalian buku
12. Melihat denda atas keterlambatan buku
13. Mengelola data pustakawan

Untuk lebih jelasnya dapat dilihat pada gambar usecase diagram dibawah ini.

******

## **Melihat Koleksi Buku**

### Deskripsi dan Prioritas

Fitur melihat koleksi buku dapat dilakukan oleh Peminjam. Sistem akan menampilkan seluruh koleksi buku yang tersedia pada ruang baca yang bersangkutan.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat Koleksi Buku | 0.89 |

4.1.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melihat Koleksi Buku** |
| **Kode Use Case** | UC-01 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat melihat koleksi buku yang tersedia. |
| **Trigger** | Aktor memilih menu koleksi buku |
| **Kondisi Awal** | Aktor berada pada halaman dashboard dan sistem belum menampilkan halaman koleksi buku. |
| **Alur Normal**   1. Aktor memilih menu Koleksi Buku 2. Sistem menampilkan daftar dari koleksi buku | **Informasi yang Terkait**  1. -  2. Data Buku |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Sistem menampilkan daftar koleksi buku |
| **Eksepsi**  - | |



* + 1. Kebutuhan Fungsional

**Peminjam :**

REQ-: Melihat Koleksi Buku

## **Fitur Mendaftar Sebagai Anggota**

4.2.1 Deskripsi dan Prioritas

Fitur mendaftar sebagai anggota dilakukan oleh peminjam yang belum memiliki status sebagai anggota. Untuk dapat melakukan transaksi pada ruang baca, peminjam harus memiliki status keangotaan dengan cara mendaftar pada ruang baca jurusan masing-masing.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Mendaftar Sebagai Anggota | 0.49 |

4.2.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Mendaftar Sebagai Anggota** |
| **Kode Use Case** | UC-02 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat mendaftar sebagai anggota |
| **Trigger** | Aktor memilih pilihan daftar anggota |
| **Kondisi Awal** | Aktor berada belum memiliki data sebagai anggota |
| **Alur Normal**   1. Aktor memilih pilihan daftar anggota 2. Sistem menampilkan form daftar anggota 3. Aktor mengisi data diri 4. Aktor menekan tombol submit 5. Sistem menyimpan data diri aktor ke database 6. Sistem menampilkan halaman selesai mendaftar | **Informasi yang Terkait**   1. - 2. - 3. Data diri anggota 4. - 5. - 6. - |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor sudah mendaftar sebagai anggota dan data disimpan pada database |
| **Eksepsi**  - | |



4.2.3 Kebutuhan Fungsional

**Peminjam :**

REQ-2: Mendaftar Sebagai Anggota

## **Fitur Melihat Status dan Deskripsi Buku**

4.4.1 Deskripsi dan Prioritas

Fitur melihat status dan deskripsi buku dapat dilakukan oleh peminjam saat akan melakukan peminjaman buku. Sistem akan menampilkan data dari buku yang akan dipinjam.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat status dan deskripsi buku | 0.49 |

4.4.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melihat status dan Deskripsi Buku** |
| **Kode Use Case** | UC-03 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat melihat status dan deskripsi buku yang akan dipinjam |
| **Trigger** | Aktor memilih buku yang akan dipinjam |
| **Kondisi Awal** | Aktor belum melihat data buku yang dipilih |
| **Alur Normal**   1. Aktor memilih buku yang akan dipinjam 2. Sistem menampilkan status dan deskripsi buku yang dipilih | **Informasi yang Terkait**   1. – 2. Data Buku |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor sudah melihat status dan deskripsi buku yang dipilih |
| **Eksepsi**  - | |



* + 1. Kebutuhan Fungsional

**Peminjam**

REQ-4: Melihat Status dan Deskripsi Buku

## **Fitur Meminjam Buku**

4.3.1 Deskripsi dan Prioritas

Fitur Meminjam buku dilakukan oleh peminjam dengan masuk ke halaman peminjaman buku. Aktor dapat melakukan peminjaman buku yang terdapat pada ruang baca yang bersangkutan dengan memilih judul buku yang ingin dipinjam.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Meminjam buku | 0.59 |

4.3.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Meminjam Buku** |
| **Kode Use Case** | UC-04 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat memilih dan meminjam buku yang dipilih |
| **Trigger** | Aktor memilih menu peminjaman buku |
| **Kondisi Awal** | Aktor telah melakukan login ke sistem dan menemukan buku yang akan dipinjam, namun belum dapat meminjam buku yang dipilih |
| **Alur Normal**   1. Include UC-03 2. Aktor menekan tombol pinjam 3. Sistem menampilkan form peminjaman 4. Aktor mengisi data peminjaman 5. Aktor menekan tombol simpan 6. Sistem menyimpan data peminjaman kedalam database | **Informasi yang Terkait**   1. – 2. – 3. – 4. Data peminjaman 5. – 6. - |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor sudah melakukan peminjaman namun belum diverifikasi |
| **Eksepsi**  - | |



4.3.3 Kebutuhan Fungsional

**Peminjam:**

REQ-4: Meminjam buku

## **Melihat Status Peminjaman Buku**

4.5.1 Deskripsi dan Prioritas

Fitur melihat status peminjaman buku memungkinkan peminjam untuk melihat status dari waktu peminjaman buku yang sedang dipinjam.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat Status Peminjaman Buku | 0.23 |

4.5.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melihat Status Peminjaman Buku** |
| **Kode Use Case** | UC-05 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat melihat status dari waktu peminjaman buku yang sedang dipinjam |
| **Trigger** | Aktor memilih menu status peminjaman |
| **Kondisi Awal** | Aktor belum melihat status dari peminjaman buku |
| **Alur Normal**   * + - 1. Aktor memilih menu status peminjaman       2. Sistem menampilkan status dari peminjaman buku yang sedang dipinjam | **Informasi yang Terkait**  1. –  2. data peminjaman |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor sudah melihat status dari poeminjaman buku yang sedang dipinjam |
| **Eksepsi**  - | |



4.5.3 Kebutuhan Fungsional

**Peminjam:**

REQ-5: Melihat Status Peminjaman Buku

## **Fitur Mencari Buku**

4.6.1 Deskripsi dan Prioritas

Fitur Mencari Buku dapat dilakukan oleh peminjam ketika ingin mencari buku yang diinginkan dengan memasukkan kata kunci dari buku yang dicari..

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Mencari Buku | 0.89 |

4.6.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Mencari Buku** |
| **Kode Use Case** | UC-06 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat mencari buku dengan memasukkan kata kunci |
| **Trigger** | Menekan tombol search |
| **Kondisi Awal** | Aktor belum menemukan buku yang dicari berdasarkan kata kunci |
| **Alur Normal**  1. Aktor memilih tombol search  2. Aktor memasukkan kata kunci buku yang dicari  3. Sistem menampilkan daftar buku yang sesuai dengan kata kunci | **Informasi yang Terkait**  1. –  2. kata kunci  4. daftar buku |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor sudah menemukan buku yang dicari sesuai daengan kata kunci |
| **Eksepsi**  - | |



* + 1. Kebutuhan Fungsional

**Peminjam**

REQ-6: Mencari Buku

## **Melihat History Peminjaman Buku**

4.7.1 Deskripsi dan Prioritas

Fitur Melihat History Peminjaman Buku memungkinkan peminjam untuk melihat historis dari peminjaman buku yang pernah dilakukan.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat History Peminjaman Buku | 0.29 |

4.7.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melihat History Peminjaman Buku** |
| **Kode Use Case** | UC-07 |
| **Aktor** | Peminjam |
| **Deskripsi** | Aktor dapat melihat History dari peminjaman buku yang pernah dilakukan |
| **Trigger** | Aktor memilih menu history peminjaman |
| **Kondisi Awal** | Aktor belum melihat hitory dari peminjaman buku |
| **Alur Normal**  1. Aktor memilih menu historis peminjaman  2. Sistem menampilkan historis dari peminjaman buku yang pernah dilakukan | **Informasi yang Terkait**  1. –  2. data historis peminjaman |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor sudah melihat hitory dari peminjaman buku |
| **Eksepsi**  - | |



4.7.3 Kebutuhan Fungsional

**Peminjam:**

REQ-7: Melihat History Peminjaman Buku

## Fitur Mengelola Data Buku

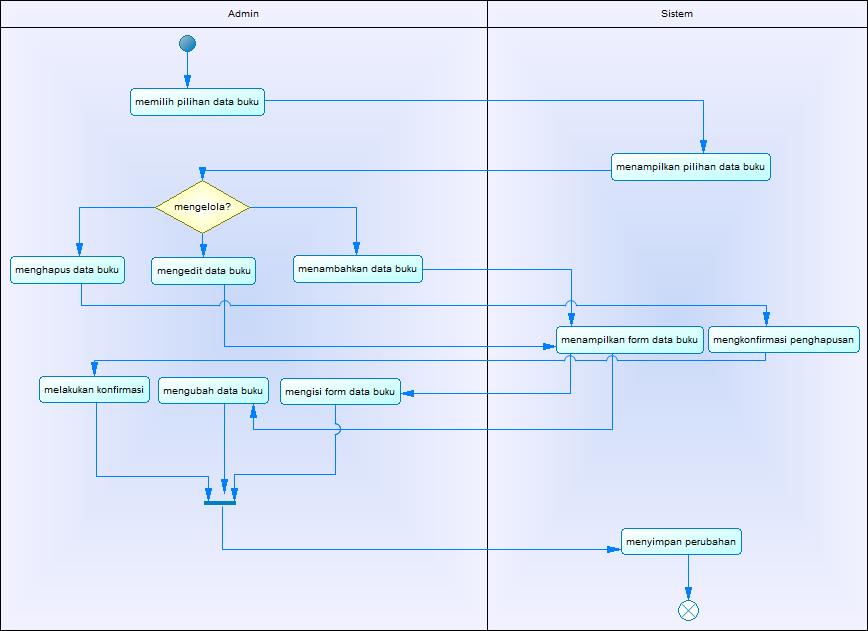
4.8.1 Deskripsi dan Prioritas

Fitur mengelola data buku dapat dilakukan oleh pustakawan. Mengelola yang dimaksud disini adalah melihat, menambah, mengubah, dan menghapus data. Data buku yang dimaksud seperti Judul, Pengarang, Tahun Terbit, Sinopsis, dan Jumlah Buku Tersedia

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat data buku | 0.79 |
| Menambah data buku | 0.79 |
| Mengubah data buku | 0.79 |
| Menghapus data buku | 0.79 |

4.8.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Mengelola Data Buku** |
| **Kode Use Case** | UC-08 |
| **Aktor** | Pustakawan |
| **Deskripsi** | Aktor dapat mengelola; melihat, menambah, mengubah, dan menghapus data buku |
| **Trigger** | Aktor memilih pilihan data buku |
| **Kondisi Awal** | Data buku belum atau sudah ada dalam database sistem |
| **Alur Normal**   1. Aktor memilih pilihan data buku 2. Sistem menampilkan halaman data buku 3. Aktor mengelola data buku    1. Aktor memilih menambahkan Buku       1. Sistem menampilkan form data buku       2. Aktor mengisi form data buku       3. Aktor memilih pilihan simpan data       4. Sistem menyimpan data buku    2. Aktor memilih pilihan edit data buku       1. Sistem menampilkan form data buku       2. Aktor mengubah data buku       3. Aktor memilih pilihan simpan data       4. Sistem menyimpan perubahan data buku    3. Aktor memilih pilihan hapus data buku       1. Sistem menampilkan konfirmasi penghapusan       2. Aktor melakukan konfirmasi       3. Sistem menghapus data buku | **Informasi yang Terkait**   1. – 2. Data buku 3. –    1. -       1. Form data buku       2. -       3. –       4. Data buku    2. –       1. Form data buku       2. –       3. Data buku    3. –       1. Notifikasi hapus data buku       2. –       3. Data buku |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Data buku sudah ada atau diperbarui |
| **Eksepsi**  - | |



4.8.3 Kebutuhan Fungsional

REQ-1: Melihat Data Buku

REQ-2: Mengubah Data Buku

REQ-3: Menambah Data Buku

REQ-4: Menghapus Data Buku

## **Fitur Melihat Daftar Peminjaman Buku**

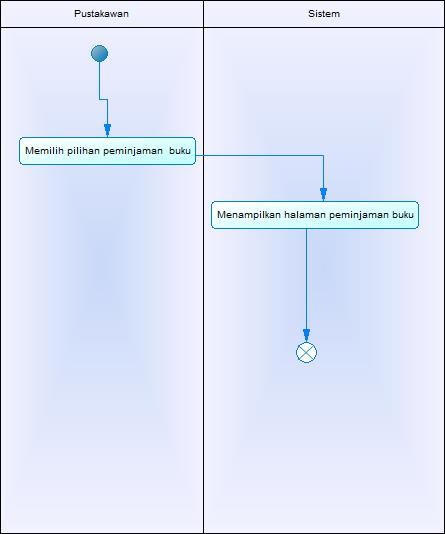
4.9.2 Deskripsi dan Prioritas

Fitur melihat daftar peminjaman buku hanya bisa dilakukan oleh pustakawan. Pustakawan dapat melihat nama-nama anggota yang sedang melakukan peminjaman buku di dalam sistem.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat daftar peminjam buku | 0.49 |

4.9.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melihat Daftar Peminjaman Buku** |
| **Kode Use Case** | UC-09 |
| **Aktor** | Pustakawan |
| **Deskripsi** | Pustakawan dapat mengetahui nama-nama orang yang sedang meminjam buku |
| **Trigger** | Aktor memilih menu peminjaman |
| **Kondisi Awal** | Daftar peminjam buku ada di database |
| **Alur Normal**   1. Aktor memilih pilihan Peminjam buku 2. Sistem menampilkan halaman peminjam buku | **Informasi yang Terkait**   1. – 2. Data peminjaman buku |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Aktor melihat daftar peminjam buku |
| **Eksepsi**  - | |



4.9.3 Functional Requirements

REQ-1: Melihat daftar peminjam buku

## **Fitur Mengelola Permintaan Peminjaman Buku**

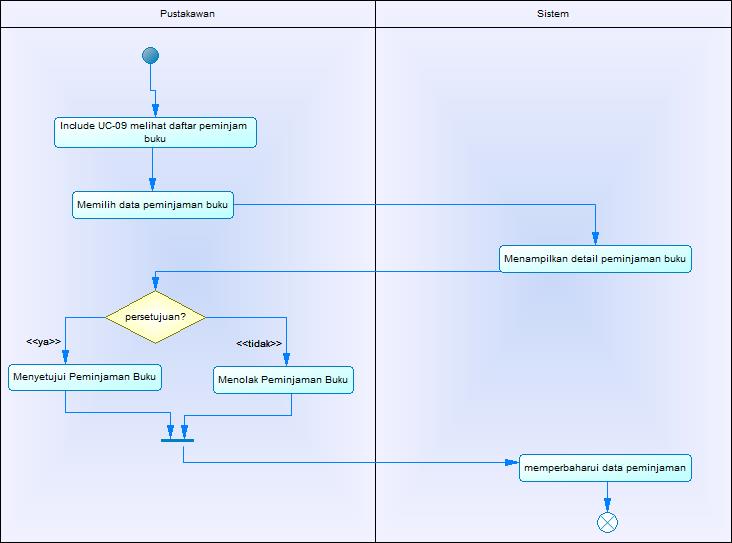
4.10.1 Deskripsi dan Prioritas

Fitur mengelola Permintaan peminjaman buku ini dilakukan oleh pustakawan. Dalam fitur ini pustakawan dapat menyetujui maupun menolak permintaan peminjaman buku yang dilakukan oleh anggota

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Menyetujui permintaan peminjaman buku | 0.89 |
| Menolak permintaan peminjam buku | 0.89 |

4.10.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Mengelola Permintaan Peminjaman Buku** |
| **Kode Use Case** | UC-10 |
| **Aktor** | Pustakawan |
| **Deskripsi** | Pustakawan dapat mengelola permintaan peminjaman buku |
| **Trigger** | Adanya permintaan peminjaman buku yang dilakukan oleh peminjam |
| **Kondisi Awal** | Permintaan peminjaman buku belum terverifikasi (disetujui) |
| **Alur Normal**   1. Include UC-09 2. Aktor memilih data peminjaman buku 3. Sistem menampilkan detail peminjaman buku 4. a. Aktor menyetujui peminjaman buku 5. Data peminjaman buku diperbaharui | **Informasi yang Terkait**   1. – 2. Data peminjaman buku 3. Deskripsi peminjaman buku 4. – 5. Data peminjaman buku |
| **Alur Alternatif**  4b. Aktor menolak peminjaman buku | |
| **Kondisi Akhir** | Permintaan peminjaman buku disetujui.  Permintaan peminjaman buku ditolak. |
| **Eksepsi**  - | |



4.10.3 Functional Requirements

REQ-1: Menyetujui permintaan peminjam buku

REQ-2: Menolak permintaan peminjam buku

## **Fitur Melayani Pengembalian Buku**

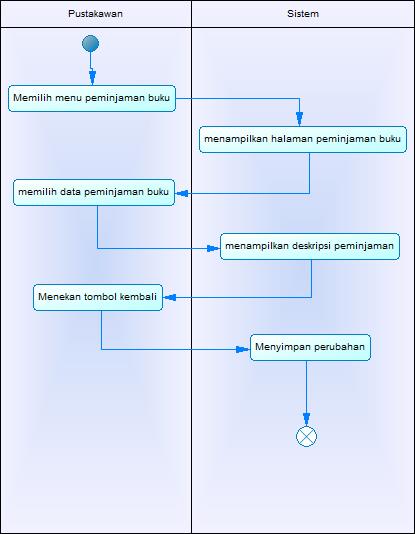
4.11.2 Deskripsi dan Prioritas

Fitur Melayani pengembalian Buku ini dilakukan oleh pustakwan. Dalam fitur ini pustakawan memeverivifikasi apabila peminjam melakukan pengembalian buku.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melayani pengembalian buku | 0.79 |

4.11.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melayani Pengembalian Buku** |
| **Kode Use Case** | UC-11 |
| **Aktor** | Pustakawan |
| **Deskripsi** | Pustakawan dapat memverifikasi pengembalian Buku |
| **Trigger** | Aktor menerima pengembalian buku dari peminjam |
| **Kondisi Awal** | Peminjaman buku sedang onprogress |
| **Alur Normal**   1. Aktor memilih menu peminjaman buku 2. Sistem menampilkan halaman peminjaman buku 3. Aktor memilih data peminjaman buku yang sesuai dengan invoice 4. Sistem menampilkan deskripsi peminjaman 5. Aktor menekan tombol kembali 6. Sistem mennyimpan perubahan | **Informasi yang Terkait**   1. – 2. list peminjaman buku 3. data peminjaman buku 4. data peminjaman buku 5. – 6. Data peminjaman buku |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Peminjaman buku telah selesai dikembalikan |
| **Eksepsi**  - | |



4.11.3 Functional Requirements

REQ-1: Melayani pengembalian Buku

## **Fitur Melihat Denda Peminjaman**

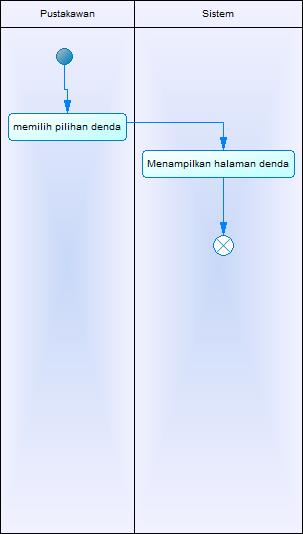
4.12.2 Deskripsi dan Prioritas

Fitur melihat denda peminjaman ini hanya bisa dilakukan oleh pustakawan. Pada fitur ini pustakawan dapat langsung mengetahui siapa saja peminjam yang telah melewati batas peminjaman dan berapa denda yang harus mereka bayar.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat denda Peminjaman | 0.29 |

4.12.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Melihat Denda Peminjaman** |
| **Kode Use Case** | UC-12 |
| **Aktor** | Pustakawan |
| **Deskripsi** | Pustakawan dapat melihat denda dari peminjaman yang telah melewati batas |
| **Trigger** | Pengingat untuk pengembalian buku |
| **Kondisi Awal** | Pustakawan belum mengetahui data denda peminjaman |
| **Alur Normal**   1. Admin memilih pilihan Denda 2. Sistem menampilkan Halaman Denda | **Informasi yang Terkait**   1. – 2. Data peminjaman |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Pustakawan mengetahui data denda |
| **Eksepsi**  - | |



4.12.3 Functional Requirements

REQ-1: Melihat denda peminjaman

## **Fitur Mengelola data Pustakawan**

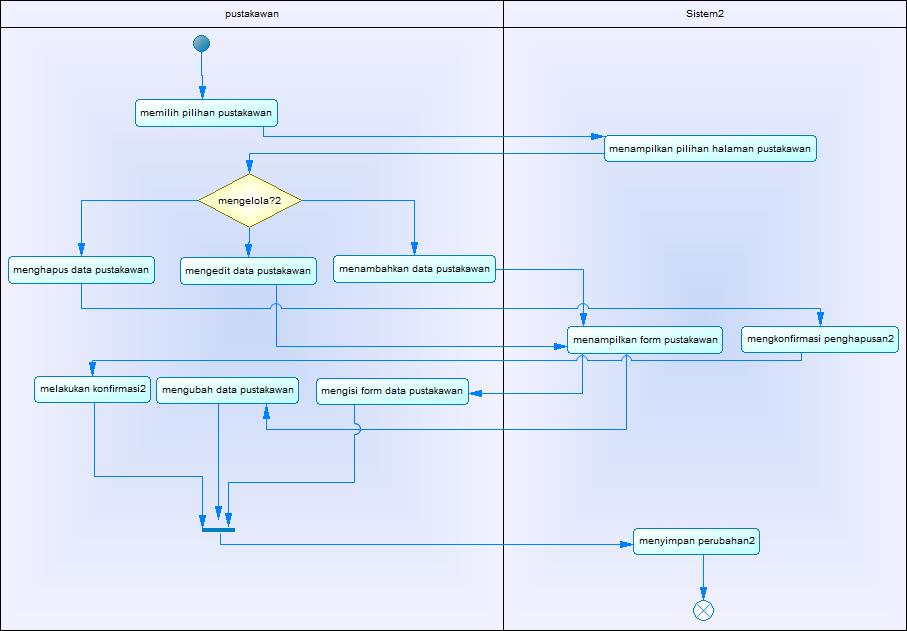
4.13.2 Deskripsi dan Prioritas

Fitur mengelola data pustakawan hanya dapat dilakukan oleh admin. Mengelola yang dimaksud disini adalah melihat, menambah, mengubah, dan menghapus data. Akun pustakawan yang dimaksud seperti nama, tanggal lahir, nomor induk, dan tanggal dibuat akun.

|  |  |
| --- | --- |
| **Kebutuhan Fungsional** | **Prioritas** |
| Melihat data pustakawan | 0.89 |
| Menambah data pustakawan | 0.89 |
| Mengubah data pustakawan | 0.89 |
| Menghapus data pustakawan | 0.89 |

4.13.2 Urutan Aksi dan Respons

|  |  |
| --- | --- |
| **Nama Use Case** | **Mengelola data Pustakawan** |
| **Kode Use Case** | UC-13 |
| **Aktor** | Admin |
| **Deskripsi** | Admin dapat mengelola; melihat, menambah, mengubah, dan menghapus data akun pustakawan |
| **Trigger** | Admin memilih pilihan pustakawan |
| **Kondisi Awal** | Data akun pustakawan atau sudah ada dalam database sistem |
| **Alur Normal**   1. Admin memilih pilihan Pustakawan 2. Sistem menampilkan halaman Pustakawan 3. Admin mengelola data pengguna    1. Admin memilih pilihan tambahkan Pustakawan       1. Sistem menampilkan form data pustakawan       2. Admin mengisi form data pustakawan       3. Admin memilih pilihan simpan data       4. Sistem menyimpan data pustakawan    2. Admin memilih pilihan edit data pustakawan       1. Sistem menampilkan form data pustakawan       2. Admin mengubah data pustakawan       3. Admin memilih pilihan simpan data       4. Sistem menyimpan perubahan data pustakawan    3. Admin memilih pilihan hapus data pustakawan       1. Sistem menampilkan konfirmasi penghapusan       2. Admin melakukan konfirmasi       3. Sistem menghapus data pustakawan | **Informasi yang Terkait**   1. – 2. Data pustakawan 3. –    1. -       1. Form data pustakawan       2. -       3. –       4. Data pustakawan    2. –       1. Form data pusatakawan       2. –       3. Data pustakawan    3. –       1. Notifikasi hapus data pengguna       2. –       3. Data pustakawan |
| **Alur Alternatif**  - | |
| **Kondisi Akhir** | Data pustakawan sudah ada atau diperbarui |
| **Eksepsi**  - | |



4.13.3 Functional Requirements

REQ-1: Melihat Data Pustakawan

REQ-2: Mengubah Data Pustakawan

REQ-3: Menambah Data Pustakawan

REQ-4: Menghapus Data Pustakawan

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

Sistem harus dapat digunakan secara responsif dengan waktu delay maksimal 10 milisekon sehingga setiap respon yang diterima sistem dapat diproses tanpa adanya waktu delay yang cukup lama. Sistem juga harus mampu memproses 100 transaksi peminjaman per detik saat peak load.

## **5.2 Safety Requirements**

Data yang dikirim ke server harus aman tanpa adanya perubahan dalam informasi yang dikirim. Sistem dapat dilakukan backup untuk mencegah kehilangan data serta dapat dilakukan recovery apabila terjadi crash.

## **5.3 Security Requirements**

Sistem harus memiliki mekanisme login user yang dapat mencegah hacking dengan cara meningkatkan keamanan apabila terjadi spam login lebih dari 3 kali oleh user tertentu. Selain itu setiap data yang dikirim baik dari server ataupun ke server harus sudah dienkripsi menggunakan metode enkripsi MD2.

## **5.4 Software Quality Attributes**

### **5.4.1 Usability**

*Untuk memudahkan user dalam melakukan peminjaman buku, sistem Inter Library Loan harus dapat memproses pendaftaran user selama waktu maksimal 1 menit.*

### **5.4.2 Availability**

*Agar dapat digunakan oleh user setiap saat, sistem Inter Library Loan harus memiliki waktu uptime hingga 99.9% per tahun.*

### **5.4.3 Maintainability**

*Apabila sistem mengalami down, sistem Inter Library Loan harus dapat kembali up dalam waktu maksimal 10 menit.*

### **5.4.4 Portability**

*Sistem Inter Library Loan harus dikembangkan pada sistem operasi Windows dan Linux, sehingga memudahkan migrasi antar kedua sistem operasi tersebut.*

### **5.4.5 Reliability**

*Setidaknya hanya 1 dari 100.000 transaksi yang gagal diproses oleh sistem Inter Library Loan per tahun.*

### **5.4.6 Robustness**

*Ketika terjadi disk crash, estimasi kehilangan data harus lebih kecil dari 0.01%.*

## **5.5 Business Rules**

Untuk melakukan peminjaman, peminjam harus memiliki akun yang terdaftar pada sistem Inter Library Loan. Segala keterlambatan pengembalian buku harus didenda.

# **6. Other Requirements**

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>