*SOFTWARE DESIGN DOKUMEN (sdd)*

FOR SYSTEM PERPUSTAKAAN UNJANI

# DAFTAR ISI

[DAFTAR ISI i](#_Toc42292130)

[BAB I Introduction 1](#_Toc42292131)

[1.1 Purpose 1](#_Toc42292132)

[1.2 Scope 1](#_Toc42292133)

[1.3 Overview 1](#_Toc42292134)

[1.4 References 2](#_Toc42292135)

[BAB II System Overview 3](#_Toc42292136)

[BAB III Application Design 4](#_Toc42292137)

[3.1 Use Case Diagram 4](#_Toc42292138)

[3.2 UseCaseScenario 4](#_Toc42292139)

[3.3 Class Diagram 12](#_Toc42292140)

[3.4 Sequence Diagram 13](#_Toc42292141)

[3.5 Activity Diagram 16](#_Toc42292142)

[3.6 State Diagram 21](#_Toc42292143)

[3.7 Deployment Diagram 25](#_Toc42292144)

[BAB IV Non-Functional Requirements 26](#_Toc42292145)

[4.1 Logical Design 26](#_Toc42292146)

[4.2 Physical Design 27](#_Toc42292147)

[BAB V User Interface Design 30](#_Toc42292148)

[BAB VI Interface Requirements 34](#_Toc42292149)

[6.1 User Interface 34](#_Toc42292150)

[6.2 Hardware Interface 34](#_Toc42292151)

[6.3 *Software* Interface 34](#_Toc42292152)

[6.4 Communication Interface 35](#_Toc42292153)

# BAB I Introduction

## Purpose

Perpustakaan UNJANI merupakan perpustakaan pusat milik Universitas Achmad Yani yang menyediakan berbagai jenis koleksi buku. Perpustakaan ini telah memiliki *website* resmi dan melakukan semi *online* untuk proses transaksinya. Dari hasil analisis kekurangan yang dimiliki *website* perpustakan ini yaitu tidak dapat melakukan pendaftaran anggota secara *online*, mahasiswa harus datang ke perpustakaan dan melapor ke petugas untuk melakukan pendaftaran anggota, hanya dapat melihat koleksi buku tapi tidak dengan *request* peminjamn.

Perangakat Lunak yang dirancang ini dapat mengatasi masalah tersebut. Perangkat Lunak ini dapat melakukan pendaftaran anggota secara *online* dengan ngisi *form* secara benar kemudian petugas memvalidasi permintaan pendaftaran tersebut, mahasiswa akan mendapat username dan *password* serta kartu anggota virtual yang dapat dicetak langsung diperpustakaan. Perangkat Lunak ini dapat melakukan peminjamn buku secara *online* berdasarkan hasil pencarian yang tersedia, sehingga tidak ada lagi kejadian mahsiswa berputar-putar di perpustakaan mencari buku dan buku yang dicari tidak ada/habis.

## Scope

Ruang Lingkup dari pembuatan perangkat lunak ini yaitu:

1. Perangkat Lunak ini bersifat *online*, berbasis *website* dan hanya biasa diakses melalui *web browser*.
2. Hanya melayani proses peminjaman, pencarian buku dan registrasi anggota.
3. Mahasiswa harus tetap datang ke perpustakaan untuk mengambil dan mengembalikan buku yang dipinjam.
4. Fitur Pengembalian pada perangkat lunak ini hanya dapat dilakukan oleh petugas.
5. Hanya mahasiswa di universitas terkait (dalam hal ini hanya mahasiswa UNJANI saja) yang dapat mendaftar sebagai anggota.
6. Setiap mahasiswa hanya boleh mendaftar sekali saja.

## Overview

Dokumen ini secara garis besar terdiri dari 6 bagian dengan perincian sebagai berikut:

* Bagian 1, Pendahuluan, merupakan pengantar *Software* *Design Document* yang berisi tujuan SDD, ruang lingkup SDD, daftar definisi dan singkatan yang digunakan, referensi yang menjadi acuan, serta overview mengenai isi dari *Software* *Design Document* itu sendiri. 2.
* Bagian 2, Rancangan lingkungan implementasi, mendefinisikan lingkungan perangkat keras (*hardware*), perangkat lunak (*software*), dan basis data yang akan digunakan untuk implementasi.
* Bagian 3, Perancangan data, mendefinisikan tabel-tabel yang akan digunakan oleh perangkat keras, CDM (*Conceptual DataModel*) atau E-R Diagram, dekomposisi fungsional modul, dan menjelaskan struktur tabel yang digunakan pada aplikasi gaji pegawai.
* Bagian 4, perancangan arsitektural, mendefinisikan kajian data dan aliran data serta menjelaskan struktur program yang diperoleh dengan cara menggambarkan bagan struktur program.
* Bagian 5. Bagan antarmuka rancangan antarmuka, mendefinisikan spesifikasi antarmuka, aturan perancangan antarmuka, dan perancangan antarmuka eksternal.
* Bagian 6, perancangan prosedural, mendefinisikan fungsi prosedural dari suatu modul melalui naratif pemrosesan, deskripsi antarmuka deskripsi perancangan bahasa, modul-modul yang digunakan, struktur data internal, dan keterangan/larangan/batasan dalam perancangan.

## References

1. Booch G, Maksimchuk RA, Engle MW, Young BJ, Connallen J, Houston KA. *Object-Oriented Analysis and Design with Applications, Third Edition*. Vol 33.; 2008. doi:10.1145/1402521.1413138

2. Satzinger JW, Jackson RB, Burd SD, Jackson RB, Burd SD. *Systems Analysis and Design in A Changing World (6th Edition)*. Vol 5th ed..; 2012. doi:10.1109/TMAG.2015.2421281

3. Rumpe B. *Agile Modeling with UML: Code Generation, Testing, Refactoring*.; 2017. doi:10.1007/978-3-319-58862-9

4. Approach AO. *SYSTEMS ANALYSIS & DESIGN An Object-Oriented Approach with UML*.

# BAB II System Overview

Pada perangkat lunak yang dibuat ini yaitu untuk membuat sistem aplikasi pada pengelolaan perpustakaan di UNJANI. Dimana nantinya petugas dapat mengelola data baik data; pendaftaran anggota, peminjaman, dan pengembalian. Tidak hanya petugas yang akan diuntungkan pada pernagkat lunak yang dibuat ini akan meneguntukna anggota perpus untuk meminjam buku di perpustakaan karena akan dapat menampilkan buku masih tersedia atau tidak. Jadi anggota tidak perlu repot untuk mencari buku yang ingin dipinjam.

Adapun fungsi-fungsi yang terdapat pada perangkat lunak yang dibangun, diantaranya yaitu:

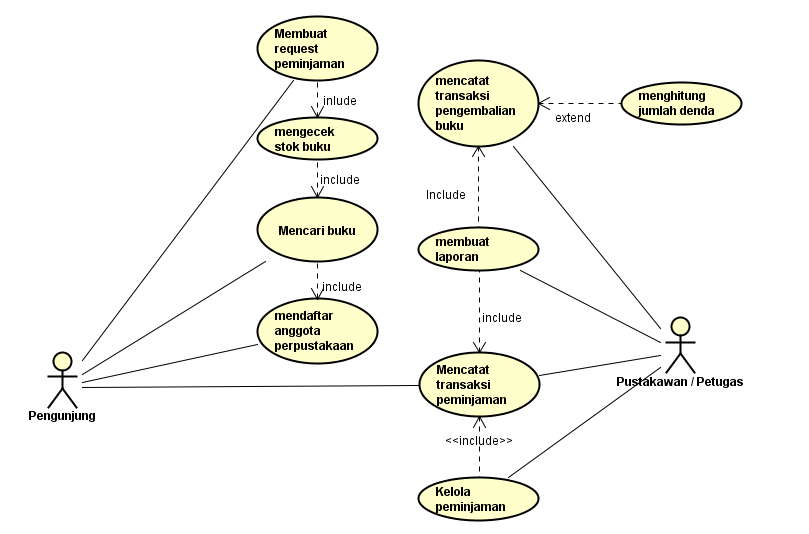
1. Memudahkan anggota mencari buku yang akan dipinjam
2. Memberikan informasi data buku yang dipinjam
3. Memberikan informasi data buku yang sudah dikembalikan
4. Dapat Melakukan pendataan buku baru, kategori dan koleksi buku di perpustakaan
5. Sistem dapat menghitung keuangan yang berasal dari denda

Fitur-fitur yang terdapat pada perangkat lunakn ini yaitu;

1. *Log*-in atau Registrasi
2. Menu Utama
3. Pencarian katalog buku
4. *Form* Peminjaman
5. *Form* Pengembalian

# BAB III Application Design

## Use Case Diagram



Gambar Use Case Diagram Perpustakaan

## UseCaseScenario

1. *Use case scenario* 1

*Use case scenario* untuk proses pengunjung perpustakaan mendaftar sebagai anggota perpustakaan dapat dilihat pada *Use case scenario* 1 Mendaftar Sebagai Anggota Perpustakaan di bawah

|  |  |
| --- | --- |
| **Name** | Mendaftar anggota perpustakaan |
| **ID** | **01** |
| **Description** | *Use Case* ini menggambarkan proses pengguna mendaftar sebagai anggota perpustakaan |
| **Actors** | Mahasiswa/pengunjung perpustakaan |
| **Secondar Actors** | Pegawai/petugas perpustakaan |
| **Pre-Condition** | Mahasiswa telah mengakses *web* apilkasi pinjam buku perpustakaan Unjani Cimahi |
| **Post-Condition** | Mahasiswa dapat *login* ke dalam *web* Aplikasi Pinjam Buku Perpustakaan Unjani Cimahi dengan *username* dan *password* yang telah diterima melalui *email* yang dikirimkan. |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. mahasiswa memilih menu *login* pada Aplikasi Pinjam Buku Perpustakaan Unjani Cimahi. |  |
|  | 1. Aplikasi menampilkan *form* *login*. |
| 1. Mahaisiwa memilih menu daftar sebagai anggota pada halaman *login* |  |
|  | 1. Aplikasi menampilkan *form* registrasi/pendaftaran sebagai anggota perpustakaan. |
| 1. Mahasiswa mengisi *form*ulir yang telah disediakan, termasuk data diri mahasiswa seperti NIM, jurusan, dan fakultas mahasiswa yang bersangkutan. |  |
| 1. Mahasiswa memilih tombol simpan. |  |
| 1. Pegawai/petugas perpustakaan menyetujui *form* *request* mahasiswa untuk mendaftar sebagai anggota perpustakaan *online*. |  |
|  | 1. Aplikasi menyimpan data anggota perpustakaan baru. |
|  | 1. Aplikasi mengirimkan *username* dan *password* ke *email* mahasiswa, yaitu *email* universitas. |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Pengguna bukan merupakan mahasisw Unjani. |  |
|  | 1. Apilkasi tidak akan menyimpan data diri pengguna dan muncul alert bahwa pendaftar harus merupakan mahasiswa di kampus Unjani Cimahi |

1. *Use case scenario* 2

*Use case scenario* untuk proses mencari buku di library perpustakaan *web* Aplikasi Pinjam Buku Perpustakaan UNJANI dapat dilihat pada *Use case scenario* 2 Mencari Buku di Perpustakaan di bawah.

|  |  |
| --- | --- |
| **Name** | Mencari buku perpustakaan. |
| **ID** | **02** |
| **Description** | *Use Case* ini menggunakan proses pengguna melakukan pencarian buku di *web* Aplikasi Pinjam Buku Perpustakaan Unjani Cimahi. |
| **Actors** | Mahasiswa/pengunjung perpustakaan. |
| **Trigers** | - |
| **Pre-Condition** | Mahasiswa telah *login* ke *web* Aplikasi Pinjam Buku Perpustakaan Unjani CImahi |
| **Post-Condition** | Aplikasi menampilkan hasil pencarian buku yang paling sesuai dengan kata kunci yang dimasukan dan mahasiswa dapat memilih buku yang paling sesuai/diinginkan. |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mahasiswa memilih menu Daftar Buku. |  |
|  | 1. Aplikasi menampilkan halaman daftar buku. |
| 1. Mahasiswa memasukan kata kunci pencarian pada mesin pencari yang ada di halaman daftar buku. |  |
|  | 1. Aplikasi menampilkan hasil pencarian yang paling sesuai dengan kata kunci yang dimasukkan oleh pengguna. |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Mahasiswa mencari buku yang tidak terdaftar/tidak ada di Aplikasi Pinjam Buku Perpustakaan Unjani Cimahi |  |
|  | 1. Aplikasi menampilkan pesan bahwa pencarian tidak ditemukan. |

1. *Use case scenario* 3

*Use case scenario* untuk proses membuat *request* peminjaman buku dapat dilihat pada *Use case scenario* 3 Membuat *Request* Peminjaman Buku di bawah.

|  |  |
| --- | --- |
| **Name** | Membuat *request* pinjam buku. |
| **ID** | **03** |
| **Description** | *Use Case* ini menggambarkan proses pengguna membuat permohonan peminjam buku. |
| **Actors** | Mahasiswa/pengunjung perpustakaan |
| **Secondary actors** | - |
| **Pre-Condition** | Mahasiswa telah melakukan pencarian buku dan menemukan buku yang akan dipinjam |
| **Post-Condition** | Aplikasi menampilkan kode *request* peminjaman buku jika buku tersedia dan dapat dipinjam. Kode *request* digunakan untuk validasi peminjaman di perpustakaan. |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Pengguna memilih menu pinjam buku pada halaman buku hasil pencarian. |  |
|  | 1. Aplikasi mengecek ketersediaan buku. |
|  | 1. Jika buku yang ingin dipinjam tersedia, aplikasi menampilkan *form*ulir peminjaman buku. |
| 1. Pengguna mengisi *form*ulir peminjaman buku. |  |
|  | 1. Aplikasi menampilkan kode *request* peminjaman untuk validasi peminjaman buku di perpustakaan jika buku dapat dipinjam. |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
|  | 1. Buku tidak tersedia. |
|  | 1. Aplikasi menampilkan alert bahwa buku tidak tersedia sehingga tidak dapat dipinjam. |

1. *Use case scenario* 4

|  |  |
| --- | --- |
| **Name** | Mencatat transaksi peminjaman |
| **ID** | **04** |
| **Description** | *Use Case* ini menggambarkan proses mencatat transaksi peminjaman |
| **Actors** | Pustakawan |
| **Trigers** |  |
| **Pre-Condition** | Pengunjung menyelesaikan proses transaksi peminjaman |
| **Post-Condition** | Pustakawan membuat laporan dari hasil transaksi-transaksi peminjaman |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Pustakawan menerima detail transaksi peminjaman |  |
|  | 1. Sistem mencatat/menyimpan data transaksi peminjaman. |
|  | 1. Sistem merekap data transaksi |
| 1. Pustakawan membuat laporan dari hasil transaksi-transaksi peminjaman |  |

1. *Use case scenario* 5

|  |  |
| --- | --- |
| **Name** | Kelola Peminjaman |
| **ID** | **05** |
| **Description** | *Use Case* ini menggambarkan proses menampilkan daftar buku yang dipinjam |
| **Actors** | Pustakawan |
| **Trigers** |  |
| **Pre-Condition** | Buku sudah atau sedang dipinjam |
| **Post-Condition** | Menampilkan daftar buku yang dipinjam |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Pustakawan *login* sebagai admin |  |
| 1. Pustakawan meng klik daftar peminjam buku |  |
|  | 1. Sistem menampilkan daftar buku yang dipinjam beserta keterang peminjaman dan identitas peminjam |

1. *Use case scenario* 6

|  |  |
| --- | --- |
| **Name** | Membuat kode peminjaman |
| **ID** | **06** |
| **Description** | *Use Case* ini menggambarkan proses membuat kode peminjaman |
| **Actors** | Pengunjung |
| **Trigers** |  |
| **Pre-Condition** | Pengunjung telah memverifikasi kode booking |
| **Post-Condition** | Situs mengirimkan pemberitahuan kepada pengunjung bahwa buku telah dapat dipinjam |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Pengunjung menyelesaikan verfikasi kode booking. |  |
|  | 1. Membuat dan mengirim salinan cetakkan kode peminjaman ke pengunjung. |
| 1. Pengunjung menerima kode peminjaman. |  |
|  | 1. Situs mengirimkan pemberitahuan kepada pengunjung bahwa buku telah dapat dipinjam |

1. *Use case scenario* 7

|  |  |
| --- | --- |
| **Name** | Mencatat pengembalian buku |
| **ID** | **07** |
| **Description** | *Use Case* ini menggambarkan proses pengunjung mengembalikan buku. |
| **Actors** | Pustakawan |
| **Secondary actors** | Mahasiswa/pengunjung perpustakaan |
| **Pre-Condition** | Mahasiswa meminjaman buku dan tercatat di sistem |
| **Post-Condition** | Aplikasi akan mencatat pengembalian buku. |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Pengunjung mengembalikan buku yang telah dipinjam kepada petugas |  |
| 1. Petugas mengimput data buku |  |
|  | 1. Sistem mencatat data pengembalian |
|  | 1. Sistem mengecek denda |
| 1. Petugas mengecek kondisi buku |  |
| 1. Buku dalam keadaan baik, petugas menaruh kembali buku |  |
| **Skenario Alternatif** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Buku yang telat dikembalikan |  |
|  | 1. Sistem menampilkan data denda yang harus dibayar |
| 1. Peminjam membayar denda |  |

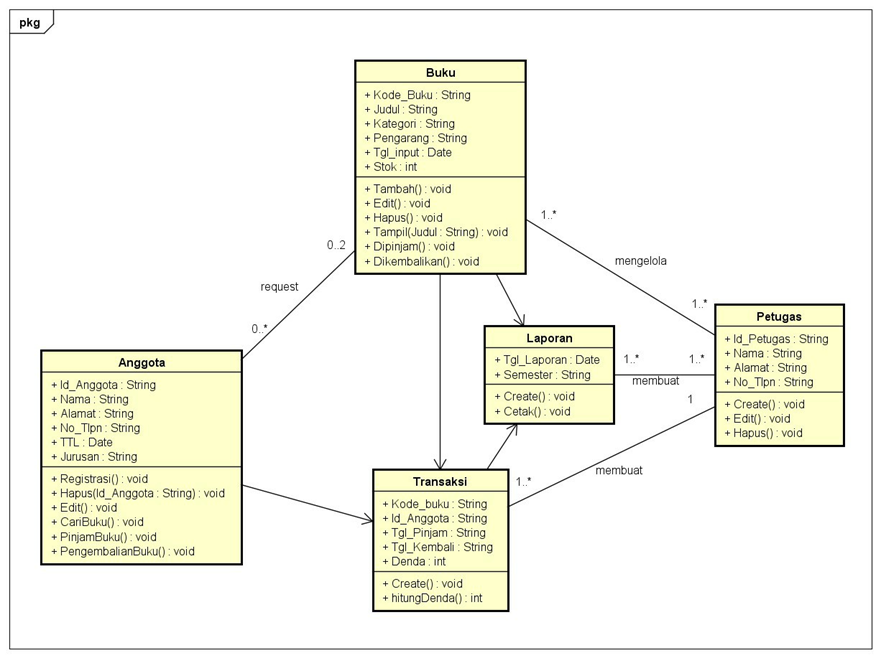
1. *Use case scenario* 8

|  |  |
| --- | --- |
| **Name** | Menghitung jumlah denda |
| **ID** | **08** |
| **Description** | *Use Case* ini menampilkan proses penghitungan denda dan menempilkan hasilnya. |
| **Actors** | Pustakawan |
| **Triger** | Peminjam telat mengembalikan buku |
| **Pre-Condition** | Denda masih nol rupiah |
| **Post-Condition** | Menampilkan denda yang harus dibayar sesuai keterlambatan |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Petugas mengimput data buku yang dipinjam |  |
|  | 1. Sistem mencatat data penjaman |
|  | 1. Sistem menghitung denda yang harus dibayar dengan ketentuan > 7 hari kerja setiap harinya akan ditambahakan denda sebesar Rp 500. |
|  | 1. Sistem menampilkan denda yang harus dibayar |

1. *Use case scenario* 8

|  |  |
| --- | --- |
| **Name** | Membuat laporan |
| **ID** | **09** |
| **Description** | *Use Case* ini menggambarkan proses pembuatan laporan |
| **Actors** | Pustakawan / petugas perpustakaan |
| **Triger** | Transaksi tercatat di database |
| **Pre-Condition** | Transaksi belum dibuat laporan |
| **Post-Condition** | Menampilkan laporan suatu transaksi |
| **Main Course** | |
| **Aksi Aktor** | **Reaksi Sistem** |
| 1. Petugas melalukan *log in* |  |
| 1. Petugas memilih laporan apa yang ingin ditampilkan |  |
|  | 1. Sistem menampilakan laporan sesuai dengan laporan yang diinginkan petugas |

## Class Diagram



Gambar Class Diagram Perpustakaan UNJANI

Diagram diatas merupakan diagram class untuk transaksi peminjaman dan pengembalian anggota. Didalam diagram tersebut kita memiliki beberapa objek diantaranya Transaksi, Anggota, Buku, Petugas dan Laporan.

Objek Anggota mempunyai relasi Assocition dengan *class* buku. Maksudnya ialah mahasiswa mencari buku mana yang akan dipinajam (*request*).

*Class* buku ke *class* Petugas perpustakaan menggunakan relasi Association karena untuk mengelola proses transaksinya.

*Class* buku ke transaksi ada relasi Composition. Karena harus adanya transaksi peminjaman tentang buku apa yang akan dipinjam.

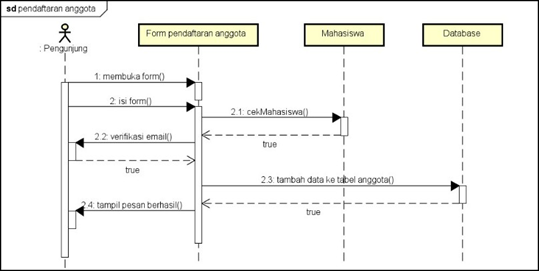
Kemudian dari Petugas perpustakaan ke *class* Laporan ada relasi *Association*, artinya petugas melakuakn pembuat laporan. Dan *class* Transaksi ke Laporan ada relasi *Composition* karena harus ada transaksi untuk membuat laporan.

## Sequence Diagram

Sequence Diagram menjelaskan detail urutan proses yang dilakukan oleh bagian-bagian yang terlibat didalam system untuk mencapai tujuan dari *use* *case* interaksi antara *class*, operasi apa yang terlibat, urutan antara operasi dan informasi yang diperlukan oleh:

1. Pendaftaran Anggota

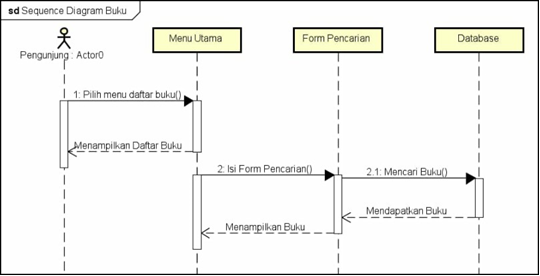
Sequence Diagram Pendaftaran Anggota menjelaskan proses tahap-tahap melakukan pendaftraan. Proses dimulai dari tampilan *form* pendaftaran, proses input data hingga menghasilkan tampilan berhasil.



Gambar Sequence Diagram Pendaftaran Anggota

1. Mencari Buku

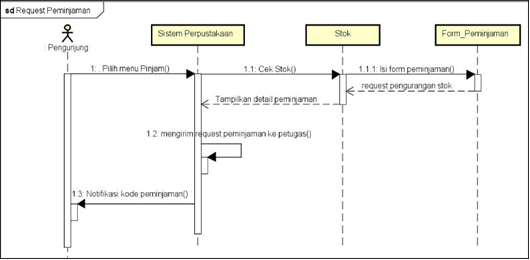
Sequence Diagram Mencari Buku menjelaskan proses tahap-tahap melakukan pencarian buku. Proses dimulai dari tampilan menu daftar buku, proses input data hingga menghasilkan tampilan buku yang dicari.



Gambar Sequence Diagram Pencarian Buku

1. *Request* Peminjaman

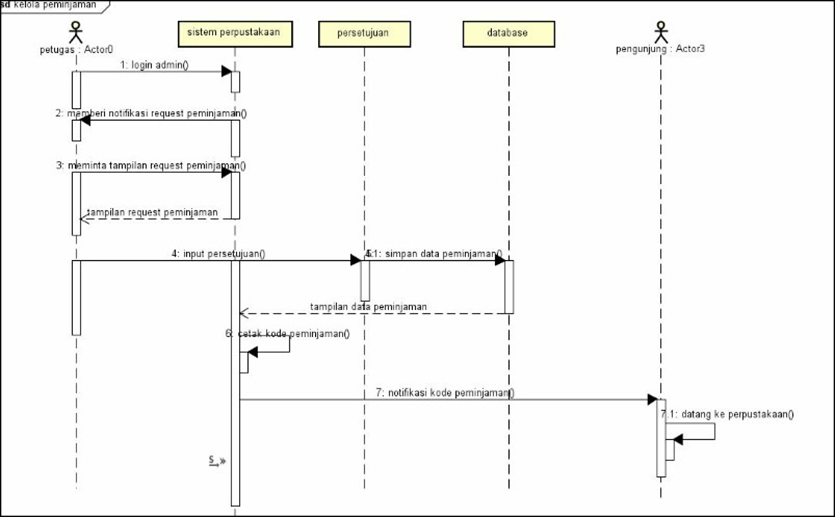
Sequence Diagram *Request* Peminjaman menjelaskan proses tahap-tahap melakukan pengajuan pinjam buku. Proses dimulai dari tampilan *form* pinjam, proses input data hingga mendapatkan notifikasi kode peminjaman.



Gambar Sequence Diagram Request Peminjaman

1. Kelola Peminjaman

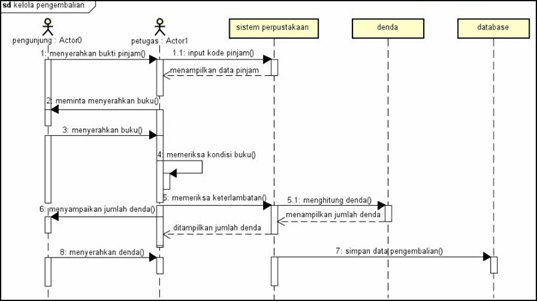
Sequence Diagram Kelola Peminjaman menjelaskan proses tahap-tahap melakukan kelola peminjaman dan saling berhubungan dengan Proses *Request* Peminjaman. Proses dimulai dari melakukan *login* sebagi admin, tampilan *request* peminjamn, melakukan persetujuan hingga mengirimkan notifikasi kode peminjaman ke pengunjung.



Gambar Sequence Diagram Kelola Peminjaman

1. Kelola Pengembalian

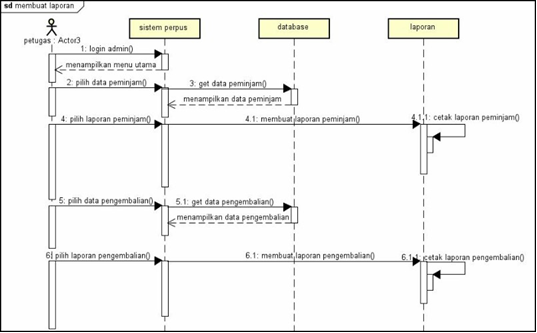
Sequence Diagram Kelola pengembalian menjelaskan proses tahap-tahap kelola pengembalian buku. Proses dimulai dari menyerahkan bukti pinjam ke petugas, proses input data hingga simpan data pengembalian.



Gambar Sequence Diagram Kelola Pengembalian

1. Membuat Laporan

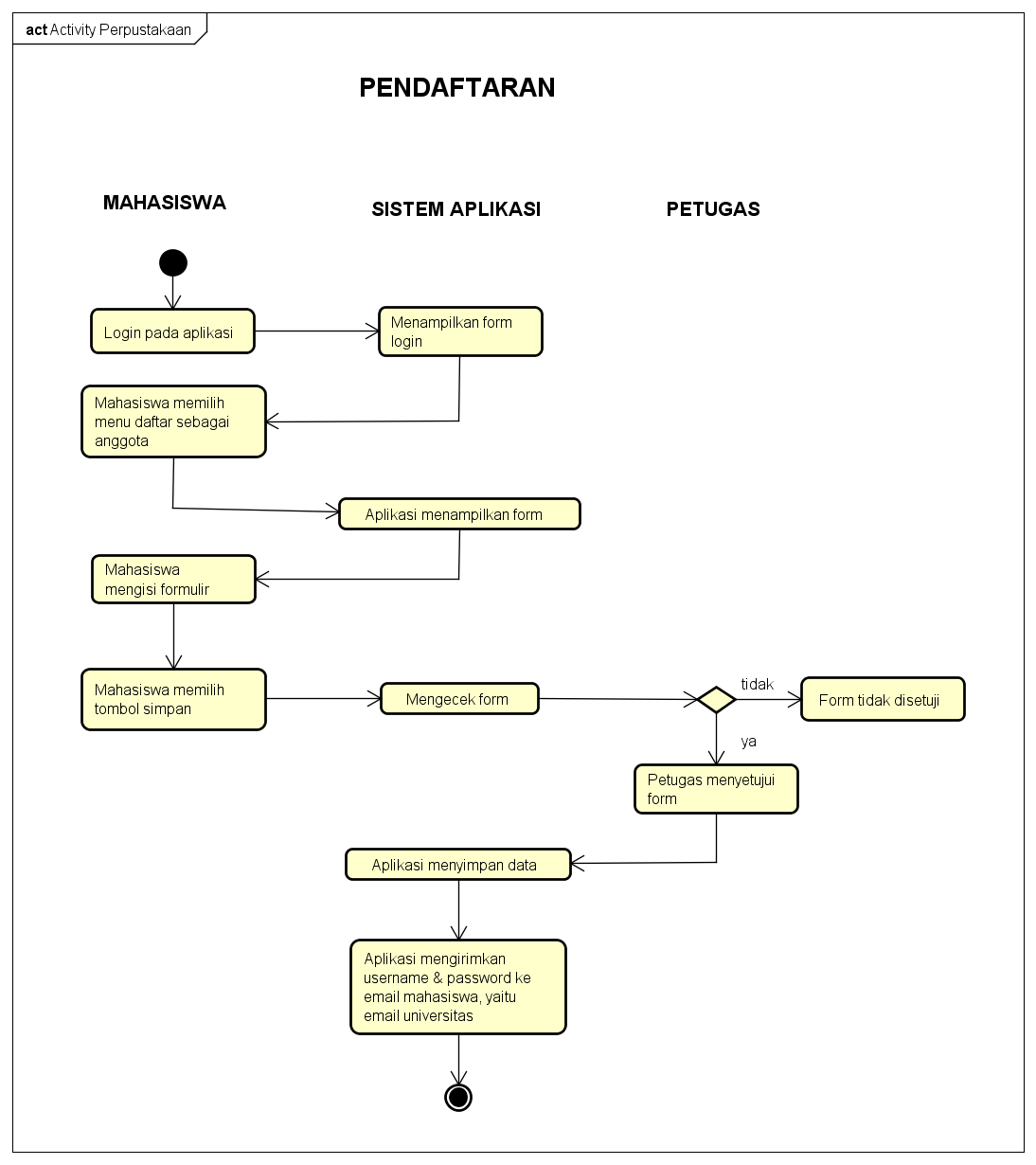
Sequence Diagram Membuat Laporan menjelaskan proses tahap-tahap melakukan pembuatan laporan. Proses dimulai dari *login* sebagi admin, hingga mencetak laporan.



Gambar Sequence Diagram Pembuatan Laporan

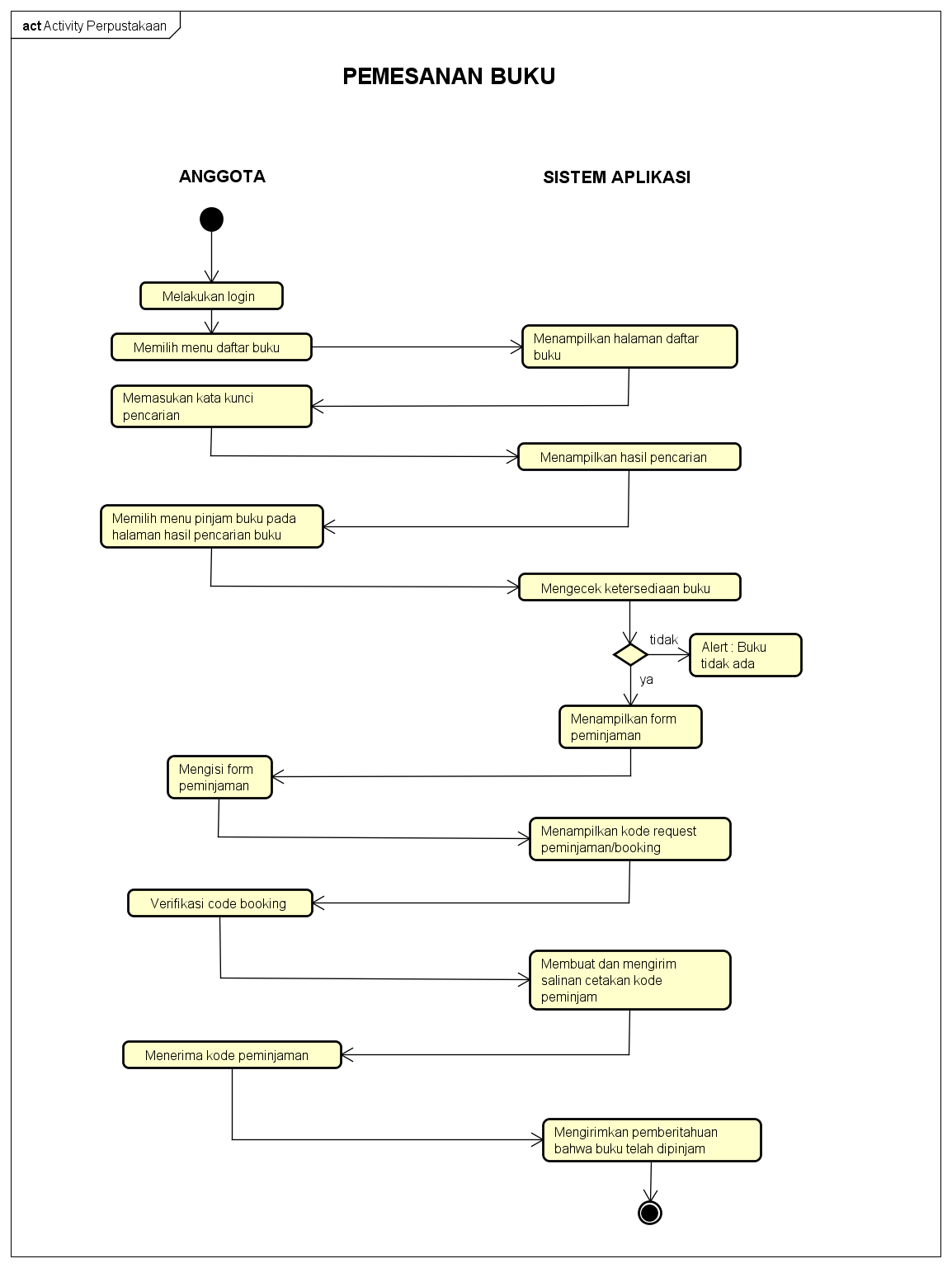
## Activity Diagram

1. Pendaftaran



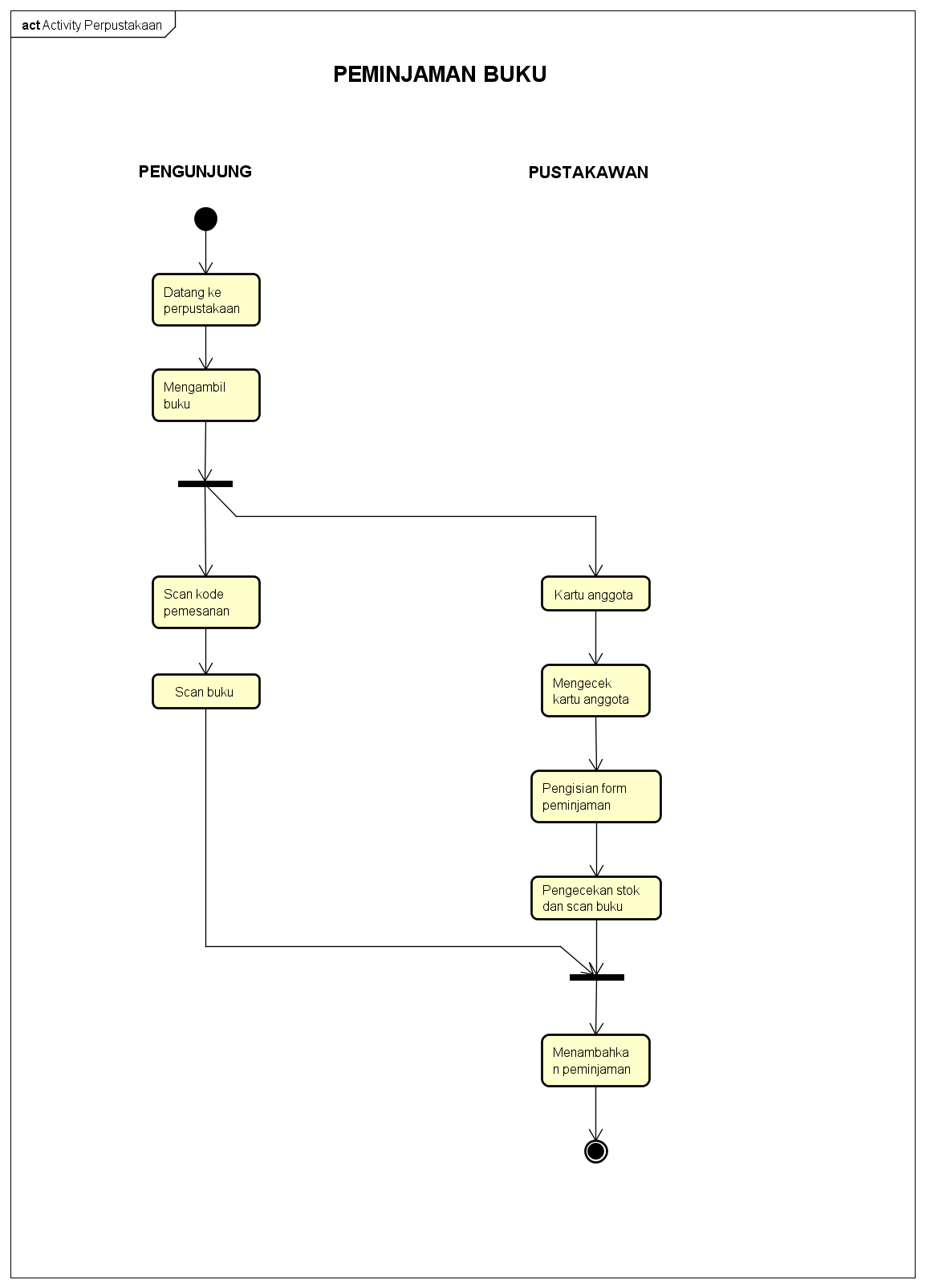
Gambar Activity Diagram Pendaftaran

1. Pemesanan Buku

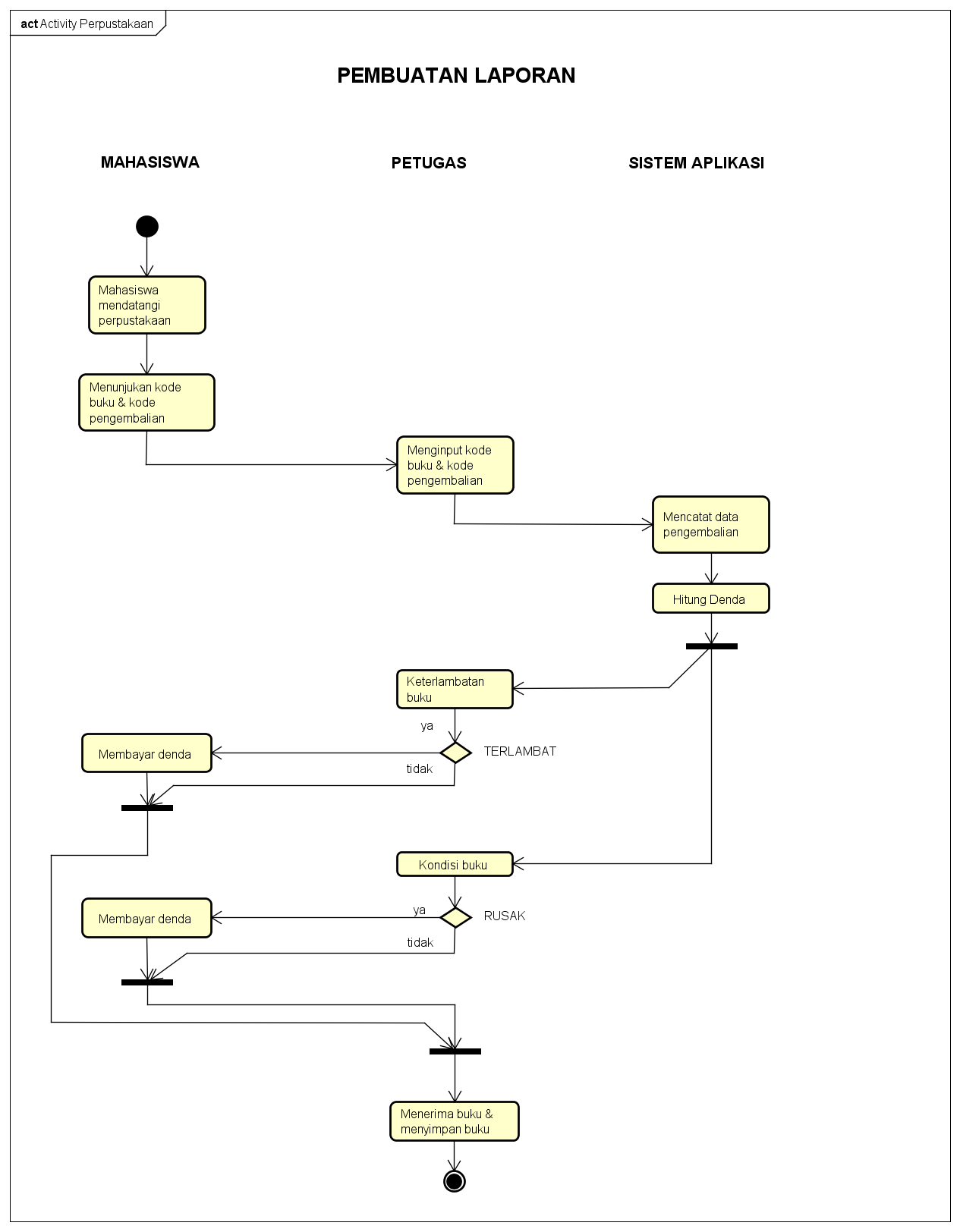


Gambar Activity Diagram Pemesanan Buku

1. Peminjaman Buku

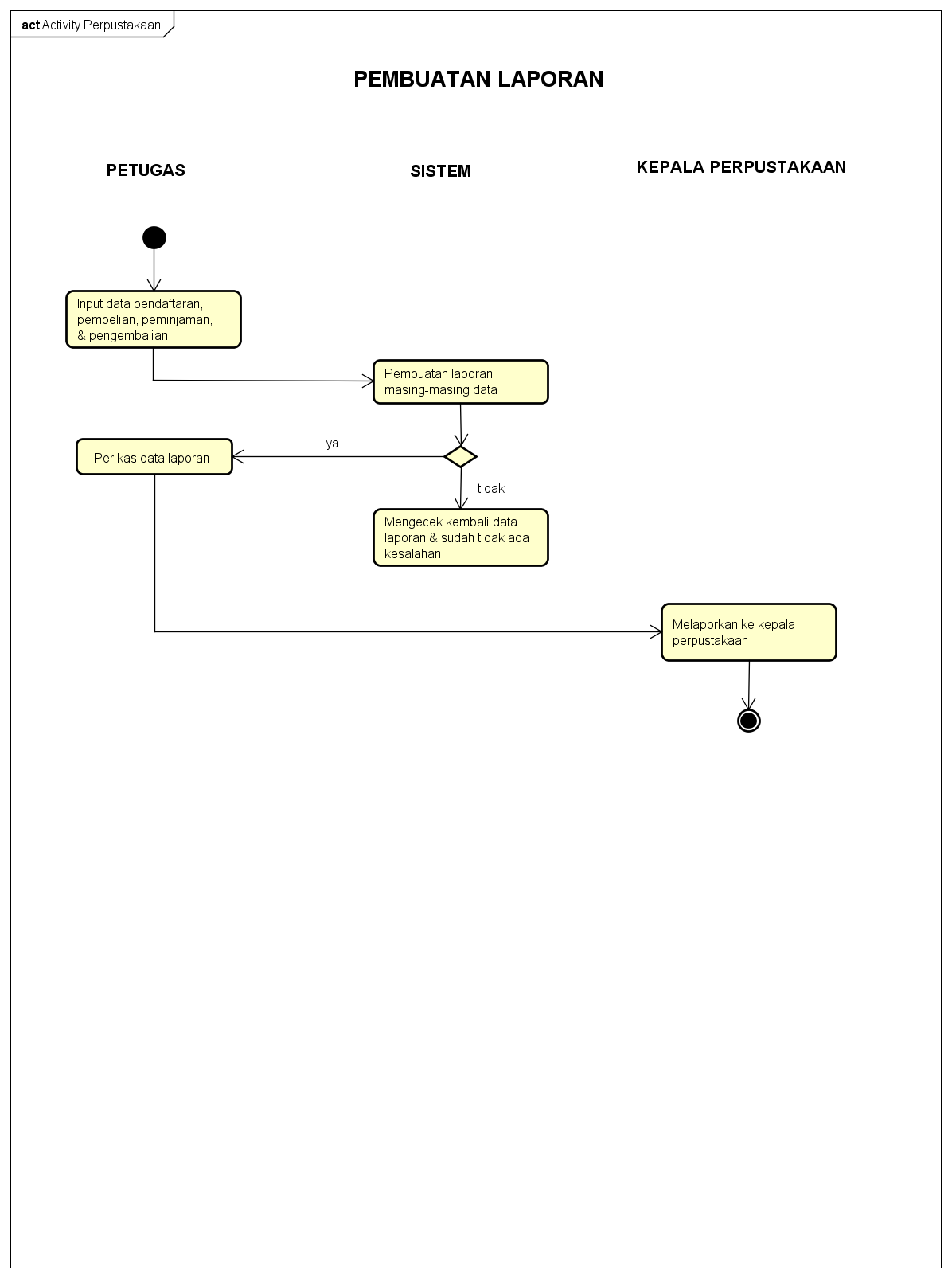


Gambar Activity Diagram Peminjaman Buku

1. Pengembalian Buku

Gambar Activity Diagram Pengembalian Buku

1. Laporan

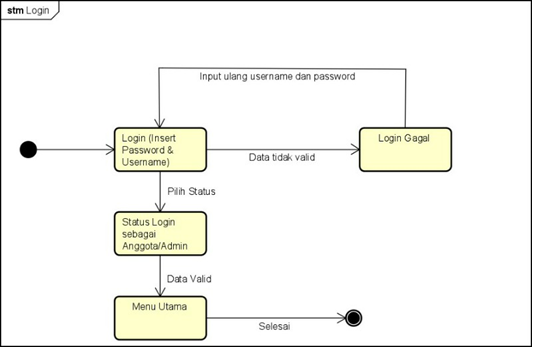


Gambar Activity Diagram Pembuatan Laporan

## State Diagram

1. *Login*

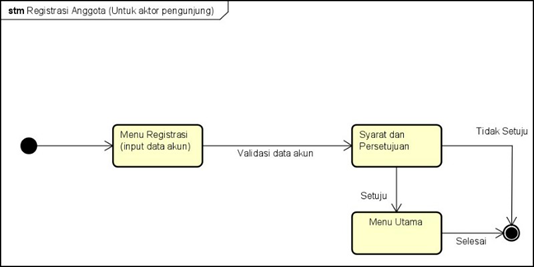
Sistem diawali pada pengisian *form* *login* kemudian memeriksa apakah username dan *password* valid. Jika valid, maka ke status *login* kemudian halaman menu utama. Jika tidak valid maka kembali ke *form* *login* diminta input ulang username dan *password*.



Gambar State Diagram Login

1. Registrasi Anggota

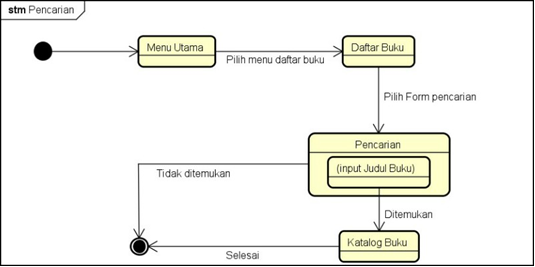
Sistem diawali pada input menu registrasi kemudian meminta syarat dan persetujuan. Jika setuju, maka ke halaman menu utama. Jika tidak setuju maka proses selesai.



Gambar State Diagram Registrasi Anggota

1. Pencarian

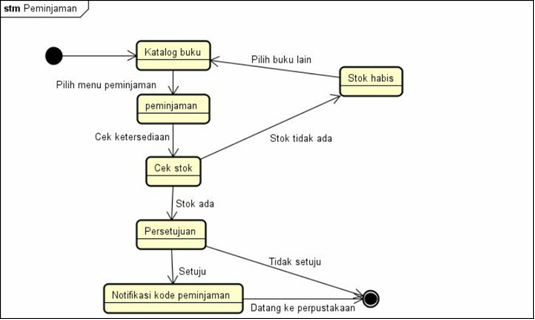
Sistem diawali tampilan menu utama kemudian daftar buku dan *form* pencarian. Isi buku yang dicari di *form* pencarian. Jika ditemukan, maka akan menampilkan katalog buku. Jika tidak valid proses selesai.



Gambar State Diagram Pencarian buku

1. Peminjaman

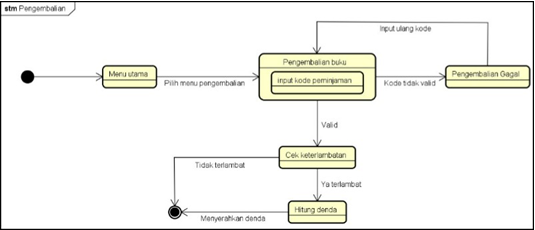
Sistem diawali tampilan katalog buku kemudian *form* peminjaman dan cek stok. Jika stok mencukupi, maka akan ke persetujuan. Jika disetujuan akan mengirim notifikasi kode peminjaman. Jika tidak maka proses selesai.



Gambar State Diagram Peminjaman Buku

1. Pengembalian

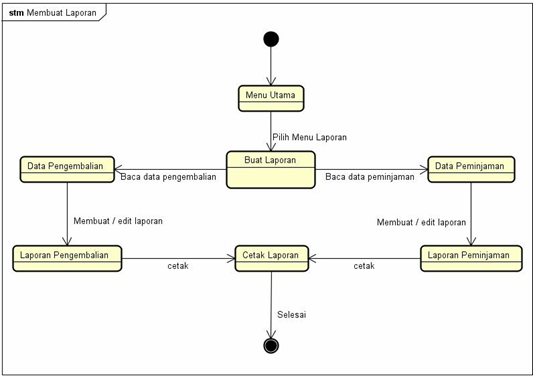
Sistem diawali tampilan menu utama kemudian *form* pengembalian. Isi *form* pengembalian dengan kode peminjaman, jika data sesuai/valid maka ke proses selanjutnya cek keterlambatan. Jika tidak maka akan kembali ke *form* pengembalian, diminta input ulang kode. Cek keterlambatan, jika terlambat maka akan menghitung denda. Jika tidak terlambat maka proses selesai.



Gambar State Diagram Pengembalian Buku

1. Laporan

Sistem diawali menu utama kemudian buat laporan. Kemudian memilih data apa yang akan dilaporkan setelah itu cetak dan proses selesai.

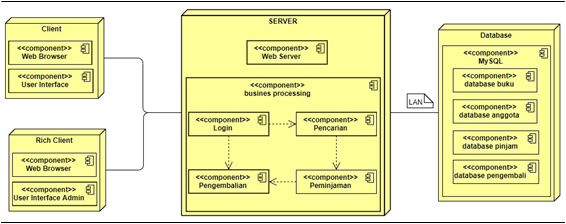


Gambar State Diagram Pembuatan Laporan

## Deployment Diagram

Diagram Deployment menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut:

1. Sistem tambahan (embedded system) yang menggambarkan rancangan device, node dan hardware.
2. Sistem client/server.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.



Gambar Deployment Diagram Sistem Perpustakaan UNJANI

# BAB IV Non-Functional Requirements

## Logical Design

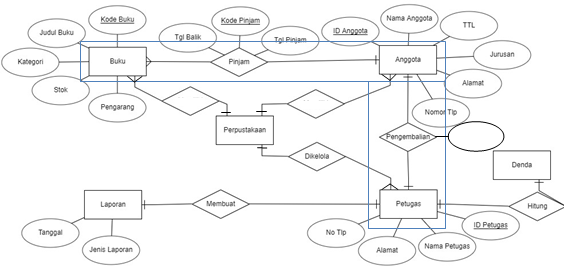
*Entity Relationship Diagram* (ERD) merupakan salah satu diagram atau notasi yang digunakan untuk mendesain data dengan pendekatan *Top* – *Down*. Bentuk data yang dirancang adalah Tabel yang memiliki relasi atau lebih dikenal sebagai *Relation*al Database.

Entitas Buku memiliki atribut Judul Buku, Kategori, Stok, Pengarang dan atribut Kode Buku sebagai *Key*. Entitas ini memiliki Relasi Pinjam dengan Anggota dan Relasi Koleksi dengan Perpustakaan. Untuk Relasi Pinjam memiliki Atribut khusus yaitu Tgl Balik (Pengembalian), Tgl Pinjam dan atribut Kode Pinjam sebagai *Key*.

Entitas Anggota memiliki atribut Nama Anggota, TTL, Jurusan, Alamat, Nomor Tlp dan atribut ID Anggota sebagai *Key*. Entitas ini memiliki Relasi Pengembalian dengan Petugas dan Relasi Daftar dengan Perpustakaan.

Entitas Petugas memiliki atribut Nama Petugas, Alamat, Nomor Tlp dan atribut ID Petugas sebagai *Key*. Entitas ini memiliki Relasi Pinjam dengan Anggota, Relasi Membuat dengan Laporan dan Relasi Hitung dengan Denda.

Terdapat Agregasi pada Relasi entitas Petugas dan Proses Peminjaman, artinya untuk melakukan pengembalian terlebih dahulu melakukan peminjaman. Dan terdapat Agregasi pada Relasi antara entitas Buku dan Anggota, artinya sebelum melakukan pinjam Mahasiswa ini harus terdaftar dulu sebagai Anggota dan Buku yang akan dipinjam harus sudah terdapat/dikoleksi oleh Perpustakaan.



Gambar ERD SIstem Perpustakaan UNJANI

## Physical Design

1. Peminjaman

* Unnormalized



* 1NF



* 2NF



* 3NF



1. Pengembalian

* Unnormalized



* 1NF



* 2NF

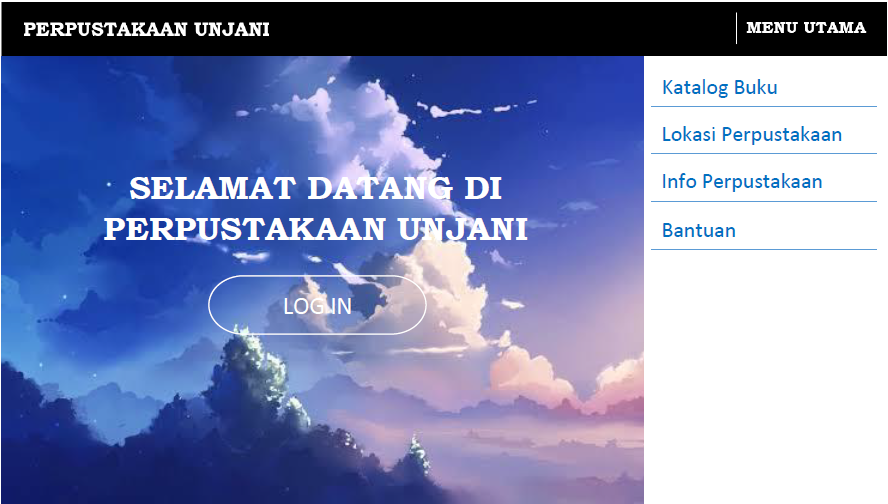


* 3NF



# BAB V User Interface Design

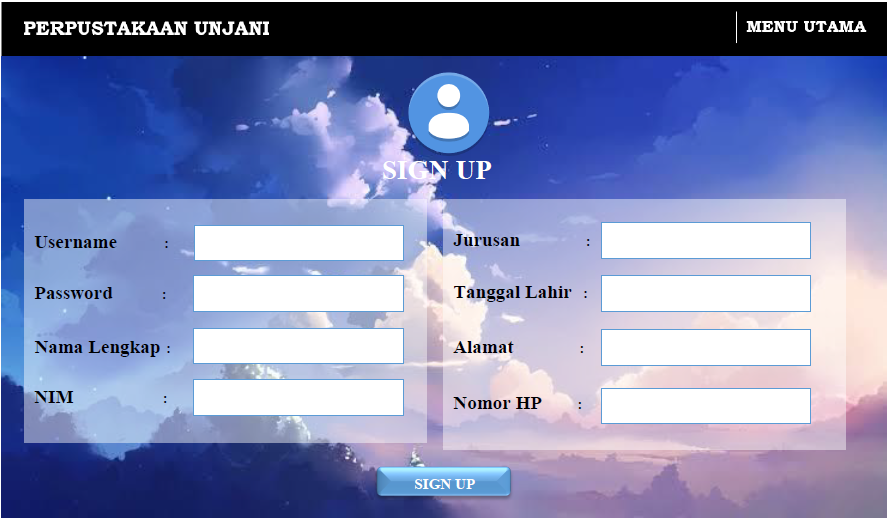
* **Tampilan Menu Utama**



Gambar Tampilan Menu Utama

Merupakan tampilan awal ketika dijalankan. Terdapat tombol/ button *LOG IN* untuk menuju ke *form* *login*. Dan terdapat pilihan menu tambahan lainnya seperti Katalog Buku, Lokasi Perpustakaan, Info Perpustakaan dan Bantuan

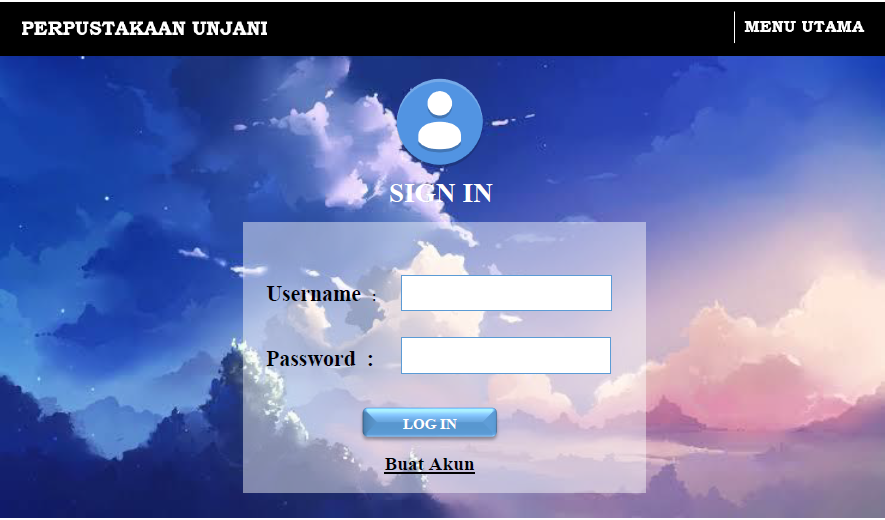
* **Tampilan *Form* Pendaftaran**



Gambar Tampilan Form Pendaftaran

Pada *form* ini pengunjung akan melakukan pendaftaran sebagai anggota dengan mengisi 6 *Textfield* berikut Username, *Password*, Nama Lengkap, NIM, Jurusan, Tanggal Lahir, Alamat dan Nomor HP. Setiap data yang diinput harus valid.

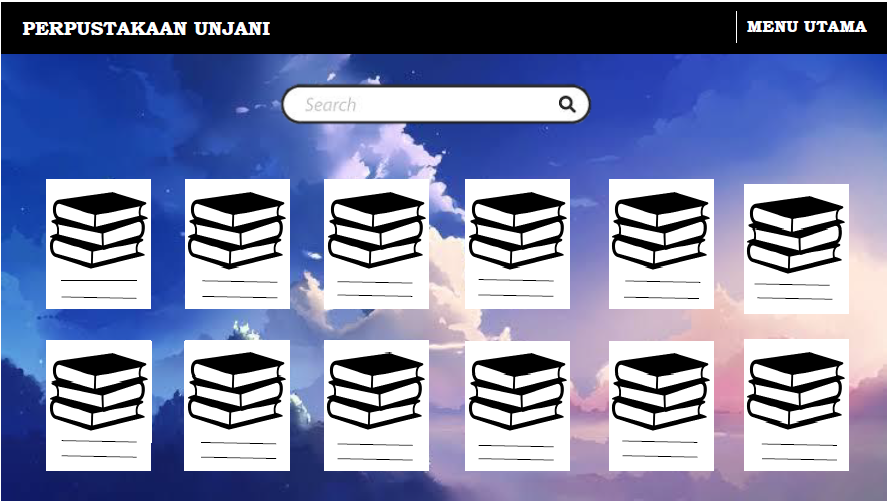
* **Tampilan *Form* *Log in***



Gambar Tampilan Form Log in

Merupakan tampilan SIGN IN atau *LOG IN* setiap pengunjung yang ingin menggunkan fitur-fitur yang ada di perangkat lunak ini harus melakukan *LOG IN* terlebih dahulu. Jika Pengunjung sudah memiliki akun atau sudah terdaftar sebagai anggota maka tinggal mengisi username dan *password* kemudian clik *LOG IN* jika tidak maka klik Buat Akun untuk mendaftar

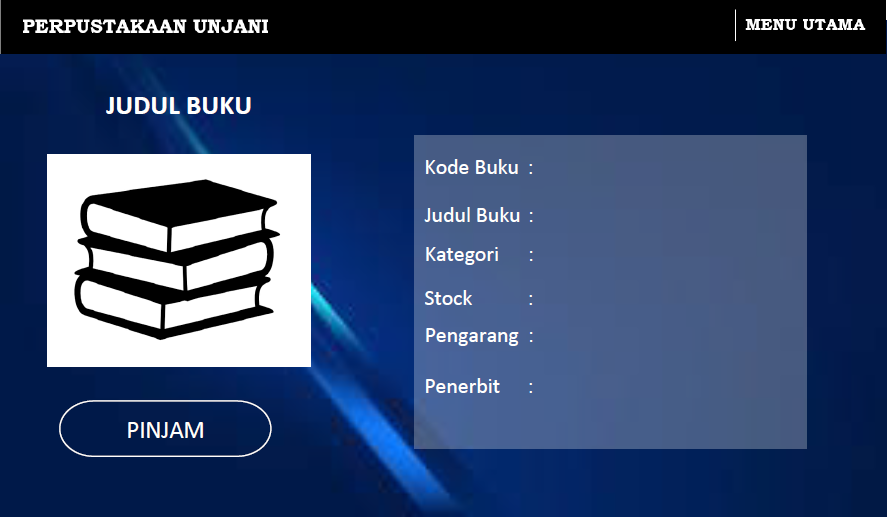
* **Tampilan Katalog dan *Form* Pencarian Buku**



Gambar Tampilan Katalog dan Pencarian

Merupakan tampilan katalog buku atau daftar buku dan terdapat pula *form* pencarian untuk menacari buku secara cepat. Ketika meng-click icon buku maka akan menuju tampilan Detail Buku

* **Tampilan Detail Buku**



Gambar Tampilan Detail Buku

Menampilkan detail buku seperti Kode Buku, Judul Buku, Kategori, Stock, Pengarang dan Penerbit. Dan juga terdapat tombol PINJAM untuk melakukan transaksi peminjaman

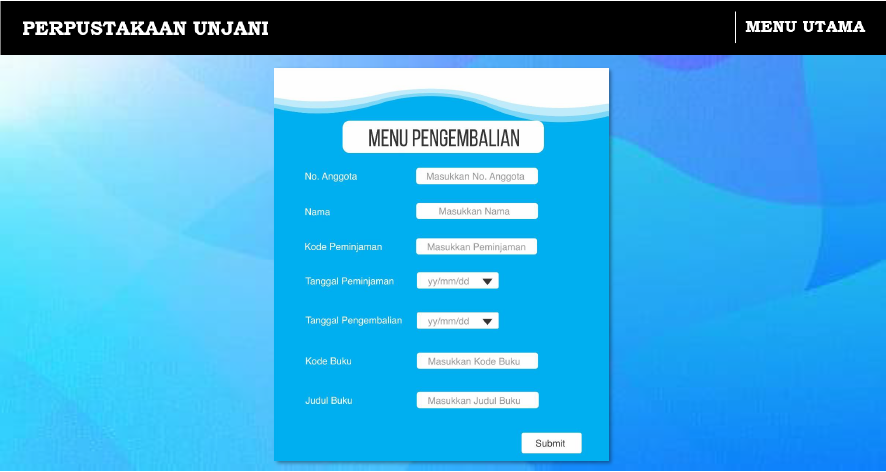
* **Tampilan *Form* Peminjaman**

****

Gambar Tampilan Form Peminjaman

Merupakan tampilan yang ada pada petugas perpustakaan. *Form* ini sebenarnya hanya akan diisi apabila anggota yang akan meminjam buku tidak melakukab booking terlebih dahulu.

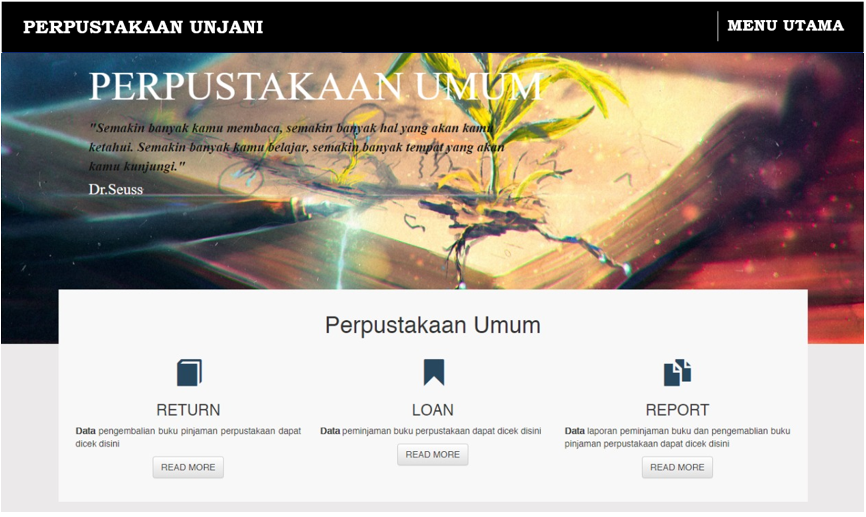
* **Tampilan *Form* Pengembalian**



Gambar Tampilan Form Pengembalian

Merupakan tampilan *form* pengembalian, proses ini dilakukan oleh petugas. Pada *form* ini perlu mengisi No Anggota, Nama, Kode Peminjaman, Tanggal Peminjaman, Tanggal Pengembalian, Kode Buku dan Judul Buku. Setiap data yang diinput harus valid

* **Tampilan *Form* Laporan**



Gambar Tampilan Form Laporan awal

# BAB VI Interface Requirements

## User Interface

Pada aplikasi ini terdapat 2 jenis user yaitu petugas perpustakaan dan anggota perpustakaan. Untuk petugas fitur yang dapat diakses yaitu *form* pengembalian, laporan transaksi, edit koleksi buku dan notifikasi persetujuan anggota. Untuk anggota fitur yang dapat diakses yaitu *form* registrasi anggota, *form* pencarian, katalog buku, tampilan detai buku dan *form* peminjaman. Setiap user baik petugas maupun anggota perlu melakukan *login* untuk mengakses fitur-fitur tersebut sebagai bentuk keamanan, tampilan aplikasi akan berbeda sesuai identitas user.

## Hardware Interface

Aplikasi dapat berjalan dengan minimum spec:

1. Processor : Intel® Celeron® Processor N Series
2. Memory : 2 GB DDR3 L
3. Hardisk : 500 GB
4. VGA : Intel® HD Graphics Processor Z3700 Series
5. Monitor : Monitor LED AOC E1670Sw-USB Powered

## *Software* Interface

*Software* *Interface* adalah titik interaksi antara berbagai komponen, interaksi ini melibatkan bagaimana pengguna dapat berinteraksi dengan aplikasi melibatkan perangkat keras dan perangkat lunak itu sendiri. Perangkat Lunak terkait untuk dapat menjalankan aplikasi perpustakaan berbasis *website* ini adalah:

1. *MySQL* sebagai perangkat lunak system manajemen basis data yang digunakan untuk mengolah database koleksi buku, data anggota dan data transaksi yang terjadi.
2. PHP (*Hypertext* *Preprocessor*), *Javascript*, CSS (*Cascading Style Sheets*), HTML (*Hyper Text Markup Language*) merupakan bahasa-bahasa pemrograman yang digunkan untuk merancang aplikasi perpustakaan berbasis website ini.
3. *Browser* *Crome*, Opera Mini, UC dll sebagai *software* untuk mengakses aplikasi perpustakaan berbasis *website* ini.

## Communication Interface

Semua *transfer* data antara perangkat lunak(*server*) dan pengguna dilakukan melalui internet dengan berbasis *web* *server*. Jadi perangkat yang akan mengakses apliksi harus memiliki koneksi internet