

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SISTEM INFORMASI PERPUSTAKAAN

untuk:

Telkom University


Dipersiapkan oleh :

1. Kemas M. Alfarabi (1301180236)
2. Arsenio Jusuf A (1301194043)
3. Muhammad Farhan (1301193333)
4. Rizal Adi P (1301194229)
5. Salma Salsabila F (1301194143)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen		Halaman
		DPPL-08		<35>
		Revisi	<nomor revisi>	

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

1. Pendahuluan	5
Tujuan Penulisan Dokumen	6
Lingkup Masalah	6
Definisi dan Istilah	6
Referensi	6
Sistematika Pembahasan	6
Deskripsi Perancangan Global	6
Rancangan Lingkungan Implementasi	7
Deskripsi Arsitektural	7
Deskripsi Komponen	7
Perancangan Rinci	8
Realisasi Use Case	8
Use Case <nama use case 1>	8
Identifikasi Kelas	8
Sequence Diagram	8
Diagram Kelas	8
Perancangan Detil Kelas	8
Kelas <nama kelas>	8
Kelas <nama kelas>	9
Diagram Kelas Keseluruhan	9
Algoritma/Query	9
Diagram Statechart	9
Perancangan Antarmuka	9
Perancangan Representasi Persistensi Kelas	10
Matriks Keruntutan	10
Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar	

1. Pendahuluan

1.1 Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak ini digunakan sebagai media pembelajaran di kampus Telkom University serta untuk memenuhi tugas besar mata kuliah Analisis dan Perancangan Perangkat Lunak. Tujuan penulisan dokumen ini sebagai acuan bagi dua pihak yang terkait yaitu antar pengembang dan pengguna. Untuk pengembang dokumen ini digunakan sebagai acuan dalam pembuatan aplikasi/perangkat lunak, sedangkan bagi pengguna dokumen ini digunakan sebagai referensi untuk mengetahui bagaimana sistem bekerja lengkap dengan spesifikasi dan kebutuhan perangkat.

1.2 Lingkup Masalah

Aplikasi yang kami buat adalah Sistem Informasi Perpustakaan yang merupakan sebuah aplikasi berbasis web dimana nantinya dapat menunjang proses sistem pendataan pada sebuah perpustakaan. Lingkup masalah pada aplikasi yang kami buat adalah untuk mengetahui kebutuhan informasi, proses pendataan perpustakaan, layanan dan koleksi buku. sehingga dapat memudahkan pustakawan mendata buku, mendata Mahasiswa (pelanggan) dan memudahkan pustakawan merekap laporan bulanan atau tahunan untuk diberikan kepada admin. Perangkat lunak ini dipasang pada masing-masing komputer admin dan pustakawan. admin hanya akan memantau segala proses kegiatan yang ada di perpustakaan, membuat akun pustakawan dan pustakawan akan melakukan pendataan, transaksi peminjaman dan pengembalian buku yang dilakukan oleh Mahasiswa.

1.3 Definisi dan Istilah

No.	Definisi	Keterangan
1.	SKPL	Spesifikasi Kebutuhan Perangkat Lunak dan merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.
2.	DPPL	Deskripsi Perancangan Perangkat Lunak dan merupakan dokumen sebagai penjelasan dan gambaran tentang rancangan yang dibuat.
3.	Admin	Merupakan seseorang yang bisa melakukan semua kegiatan yang ada di aplikasi
4.	Pustakawan	Merupakan seseorang yang mengelola data buku.
5.	Mahasiswa	Merupakan seseorang yang melakukan transaksi peminjaman dan pengembalian di perpustakaan.
6.	DBMS	Singkatan dari “ <i>Database Management System</i> ” yaitu sistem pengorganisasian dan sistem Database pada Komputer.

7.	MySQL	Perangkat lunak sistem manajemen basis data SQL atau DBMS
8.	PHP, HTML, JavaScript	merupakan bahasa skrip yang digunakan untuk membuat Sistem Informasi.
9.	XAMPP	berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan

1.4 Referensi

Dokumen acuan yang digunakan dalam penulisan dokumen ini adalah :

- SKPL Sistem Informasi Perpustakaan Umum
- Template dokumen Deskripsi Perancangan Perangkat Lunak (DPPL)

1.5 Sistematika Pembahasan

Dokumen perancangan perangkat lunak Sistem Informasi Perpustakaan ini berisikan rincian dari deskripsi pengembangan perangkat lunak. Pengorganisasian dokumen ini dikelompokkan dalam tiga bagian utama, yaitu :

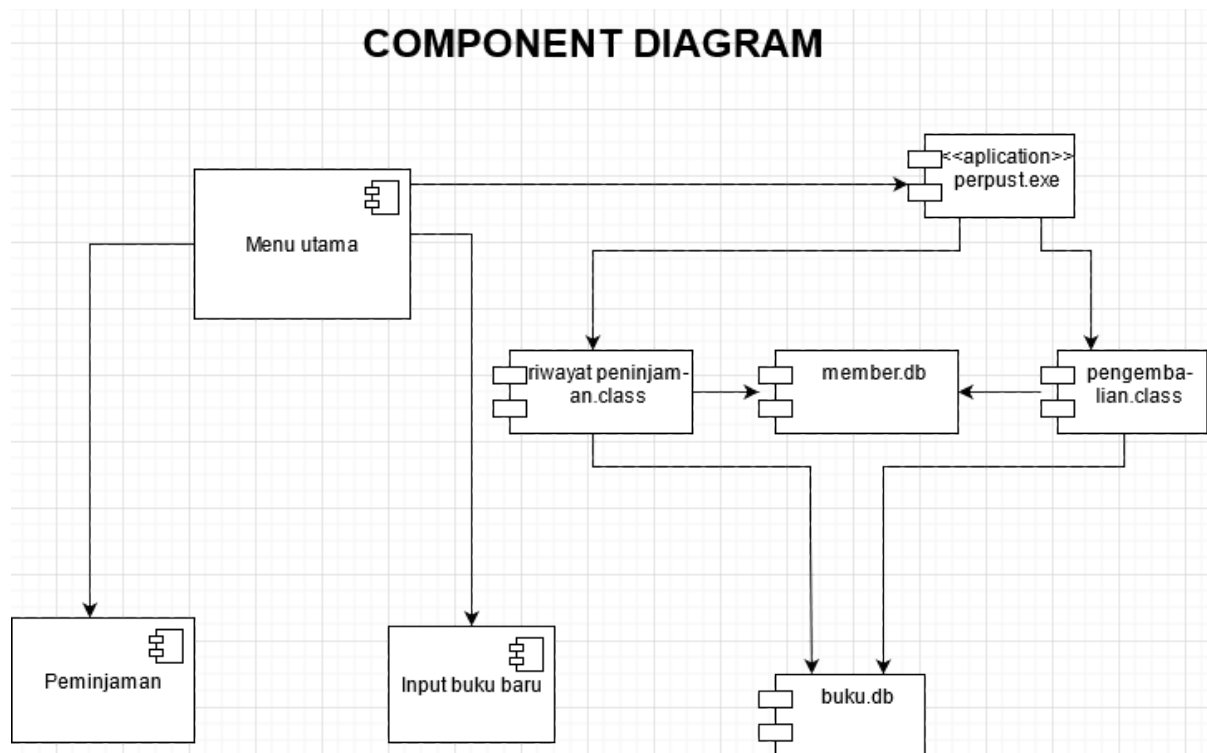
1. Bab 1 Pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah yang ditangani dalam perangkat lunak yang akan dibangun serta deskripsi umum dokumen.
2. Bab 2 Deskripsi Perancangan Global, berisi penjelasan perangkat lunak yang akan diimplementasikan di lingkungan pengguna secara global dan bagaimana aplikasi itu dibuat.
3. Bab 3 Perancangan Rinci, berisikan detail komponen, fitur yang akan diimplementasikan dan hasil analisis terhadap kebutuhan perangkat lunak yang akan digunakan secara terperinci.
4. Bab 4 Matriks Keruntutan.

2 Deskripsi Perancangan Global

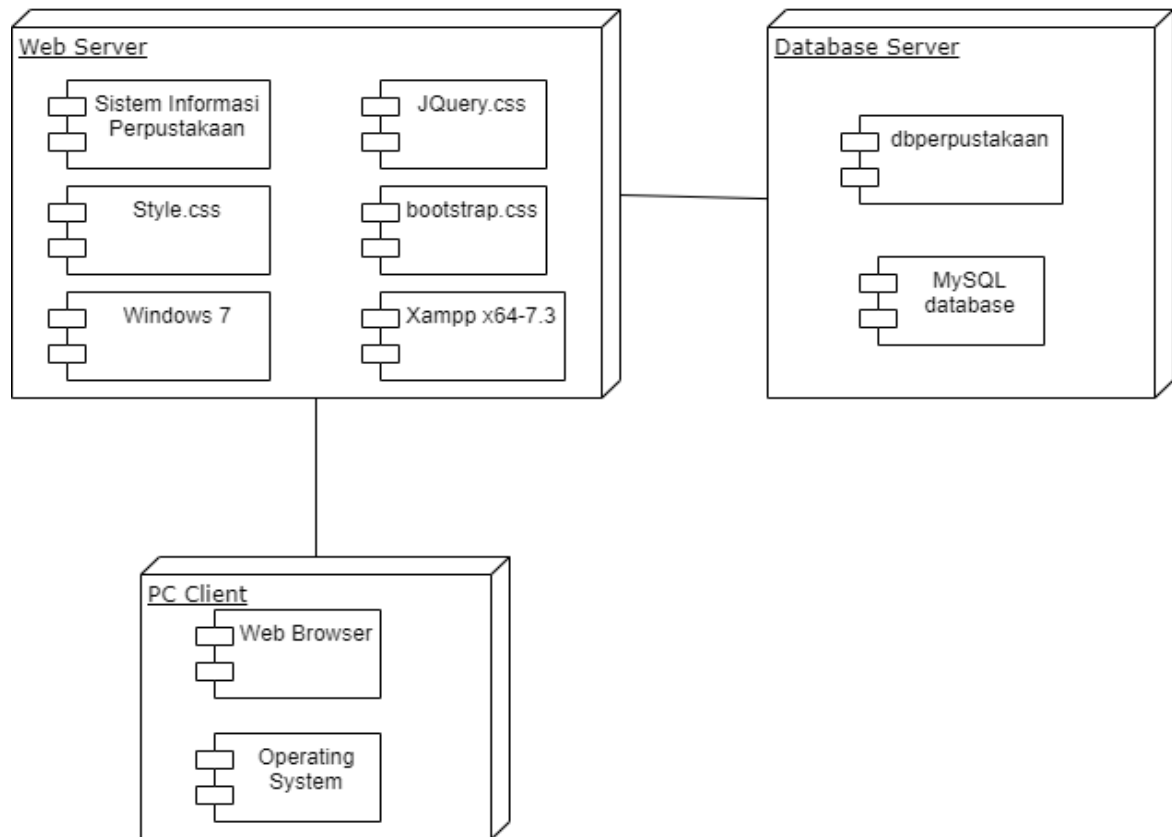
2.1 Rancangan Lingkungan Implementasi

Jenis Sistem	Spesifikasi
Sistem Operasi	Windows 7, 8, 10
DBMS	MySQL
Bahasa Pemrograman	HTML, Javascript, PHP
Development Tools	XAMPP, PHPmyAdmin, Sublime Text 3, Google Chrome

2.2 Deskripsi Arsitektural



DEPLOYMENT DIAGRAM



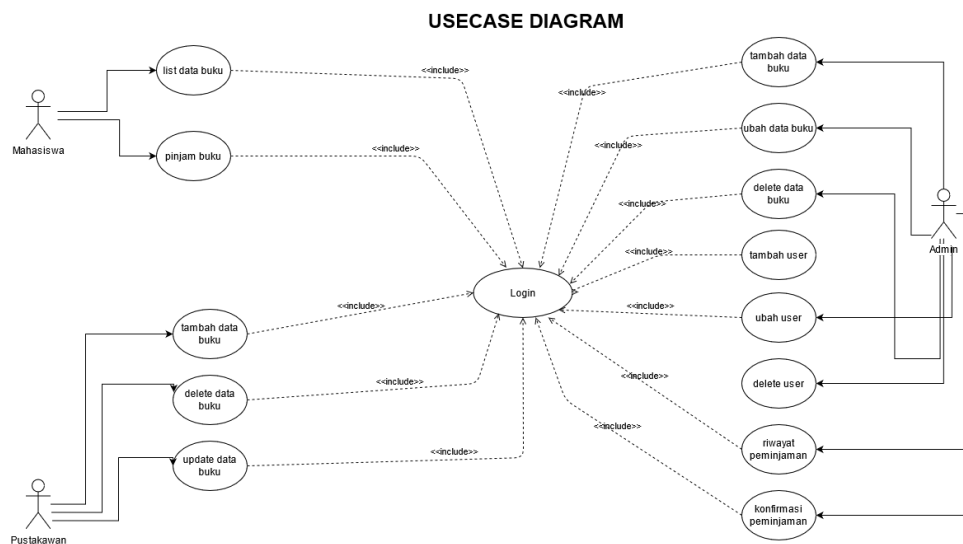
2.3 Deskripsi Komponen

No	Nama Komponen	Keterangan
1.	Menu Utama	Komponen Menu Utama
2.	Peminjaman	Komponen Peminjaman
3.	Pengembalian	Komponen Pengembalian
4.	Input buku baru	Komponen Input buku baru
5.	Denda	Komponen Denda
6.	Penggantian buku	Komponen Penggantian buku
7.	perpust.exe	Program perpust yang dapat dieksekusi
8.	peminjaman.class	Fungsi peminjaman
9.	Mahasiswa.db	Database Mahasiswa
10.	pengembalian.class	Fungsi pengembalian
11.	buku.db	Database buku
12.	penggantianbuku.class	Fungsi penggantian buku

3 Perancangan Rinci

3.1 Realisasi Use Case

3.1.1 Use Case



3.1.2 Use Case Scenario

- Nama Use case : Login

Aktor : Mahasiswa, Pustakawan, Admin

Pre-condition : Aktor belum masuk ke sistem

Post-condition masing-masing : Aktor masuk ke sistem dengan identitas

Deskripsi : Aktor melakukan login ke dalam sistem

Tabel skenario

Aktor	Sistem
	1. Menampilkan halaman login
2. Menginput data akun	
	3. Memvalidasi data login

4. Sudah masuk ke dalam sistem	
--------------------------------	--

2. Nama Use case : List Data Buku
- Aktor : Mahasiswa
- Pre-condition : Mahasiswa belum melihat Data buku
- Post-condition : Mahasiswa sudah melihat Data Buku
- Deskripsi : Aktor telah melihat info/data buku dan mendapatkan list data buku
- Tabel skenario

Aktor	Sistem
	1. Menampilkan halaman berisi daftar buku
2. Mengakses data buku yang sesuai	
	3. Menampilkan laman info/data buku yang telah dipilih
4. Mendapatkan info/data buku	

3. Nama Use case : Pinjam Buku
- Aktor : Mahasiswa
- Pre-condition : Mahasiswa belum meminjam buku
- Post-condition : Mahasiswa sudah meminjam Data Buku
- Deskripsi : Aktor telah meminjam buku yang dipilih

Tabel skenario

Aktor	Sistem
	1. Menampilkan laman list data buku
2. Memilih buku apa yang akan dipinjam di list	
	5. Buku berhasil dipinjam

5. Nama Use case : Input data Buku
- Aktor : Pustakawan
- Pre-condition : Pustakawan belum menginputkan data Buku
- Post-condition : Pustakawan sudah menginputkan Buku
- Deskripsi : Aktor telah menginputkan data buku yang dipinjam

Tabel skenario

Aktor	Sistem
	1. Menampilkan laman berisi form data buku
2. Mengisi form data buku	
	3. Merekam proses penginputan data buku lalu tersimpan ke database

4. Menerima feedback dari sistem, data berhasil ditambahkan	
---	--

6. Nama Use case : Delete data Buku
- Aktor : Pustakawan
- Pre-condition : Pustakawan belum menghapus data Buku
- Post-condition : Pustakawan sudah menghapus data Buku
- Deskripsi : Pustakawan menghapus data buku yang sudah dipinjam
- Tabel skenario

Aktor	Sistem
1. Mengakses data buku	
2. Menghapus data buku	
	3. data buku tidak ada di sistem

7. Nama Use case : Ubah data buku
- Aktor : Pustakawan
- Pre-condition : Komponen data belum berubah
- Post-condition : Komponen data berhasil diubah
- Deskripsi : Aktor mengubah poin-poin dalam data sesuai *privilege* aktor
- Tabel skenario

Aktor	Sistem
1. Memilih menu data buku	
	2. Menampilkan informasi data buku
3. Memilih opsi edit	
	4. Menampilkan form data yang ingin diubah
5. Mengedit bagian-bagian data buku yang akan dirubah	
6. Menyimpan hasil perubahan	
	7. Memperbarui <i>database</i> yang ada

9. Nama Use case : Input data buku
- Aktor : Admin
- Pre-condition : Admin belum menginputkan data buku
- Post-condition : Admin sudah menginputkan data buku
- Deskripsi : Admin sudah menambahkan data buku
- Tabel skenario

Aktor	Sistem
	1. Menampilkan laman berisi form data buku

2. Mengisi form data buku	
	3. Merekam proses penginputan data buku lalu tersimpan ke database
4. Menerima feedback dari sistem, data berhasil ditambahkan	

10. Nama Use case : Delete data buku
- Aktor : Admin
- Pre-condition : Admin belum me-registrasi data buku baru
- Post-condition : Admin sudah me-registrasi data buku baru
- Deskripsi : Aktor telah menginputkan data buku baru
- Tabel skenario

Aktor	Sistem
1. Mengakses data buku	
2. Menghapus data buku	
	3. data peminjam tidak ada di sistem

13. Nama Use case : Ubah data buku
- Aktor : Admin

Pre-condition : Komponen buku belum berubah

Post-condition : Komponen buku berhasil diubah

Deskripsi : Aktor mengubah poin-poin dalam akunnya sesuai *privilege* aktor

Tabel skenario

Aktor	Sistem
1. Memilih menu buku	
	2. Menampilkan informasi buku
3. Memilih opsi edit	
	4. Menampilkan form yang dapat diubah
5. Mengedit bagian-bagian profil yang akan dirubah	
6. Menyimpan hasil perubahan	
	7. Memperbarui <i>database</i> yang ada

Nama Use case : Tambah user

Aktor : Admin

Pre-condition : Admin belum registrasi data user

Post-condition : Admin sudah registrasi data user

Deskripsi : Admin registrasi data user role.

Tabel skenario

Aktor	Sistem
	1. Menampilkan laman berisi form data user
2. Mengisi form data buku	
	3. Merekam proses penginputan data buku lalu tersimpan ke database
4. Menerima feedback dari sistem, data berhasil ditambahkan	

Nama Use case : Ubah user

Aktor : Admin

Pre-condition : Komponen user belum berubah

Post-condition : Komponen user berhasil diubah

Deskripsi : Aktor mengubah poin-poin dalam akunnya sesuai *privilege* aktor

Tabel skenario

Aktor	Sistem
1. Memilih menu kelola user	
	2. Menampilkan informasi list user

3. Memilih user mana yang ingin di ubah	
	4. Menampilkan form yang dapat diubah
5. Mengedit bagian-bagian profil yang akan dirubah	
6. Menyimpan hasil perubahan	
	7. Memperbarui <i>database</i> yang ada

Nama Use case : Delete user

Aktor : Admin

Pre-condition : Admin belum menghapus user role

Post-condition : Admin sudah menghapus user role

Deskripsi : Aktor telah menghapuskan user role yang telah dipilih

Tabel skenario

Aktor	Sistem
1. Mengakses user role	
2. Menghapus user role	
	3. data user role tidak ada di sistem

3.1.2.1 Identifikasi Kelas

Identifikasi kelas yang terkait dengan use case tersebut. Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:

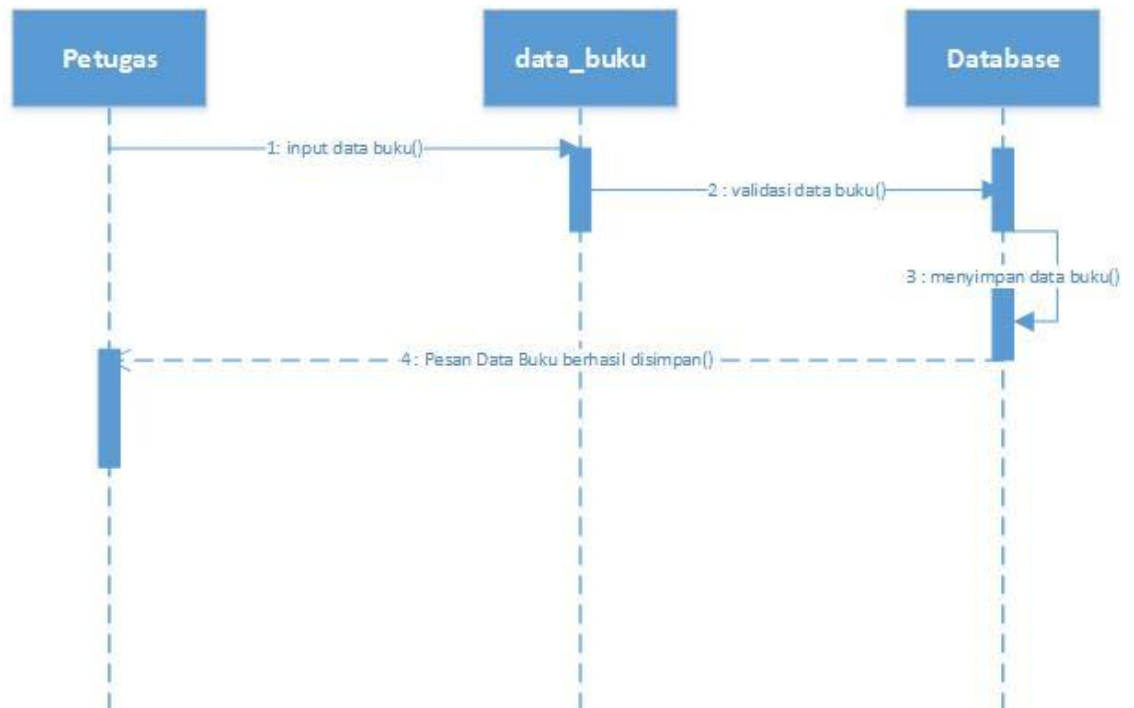
No	Nama Kelas Perancangan	Tipe Kelas
1	Mahasiswa	Controller
2	Pustakawan	Controller
3	Admin	Controller
4	Buku	Controller
5	Laporan	Controller
6	Denda	Controller

**Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

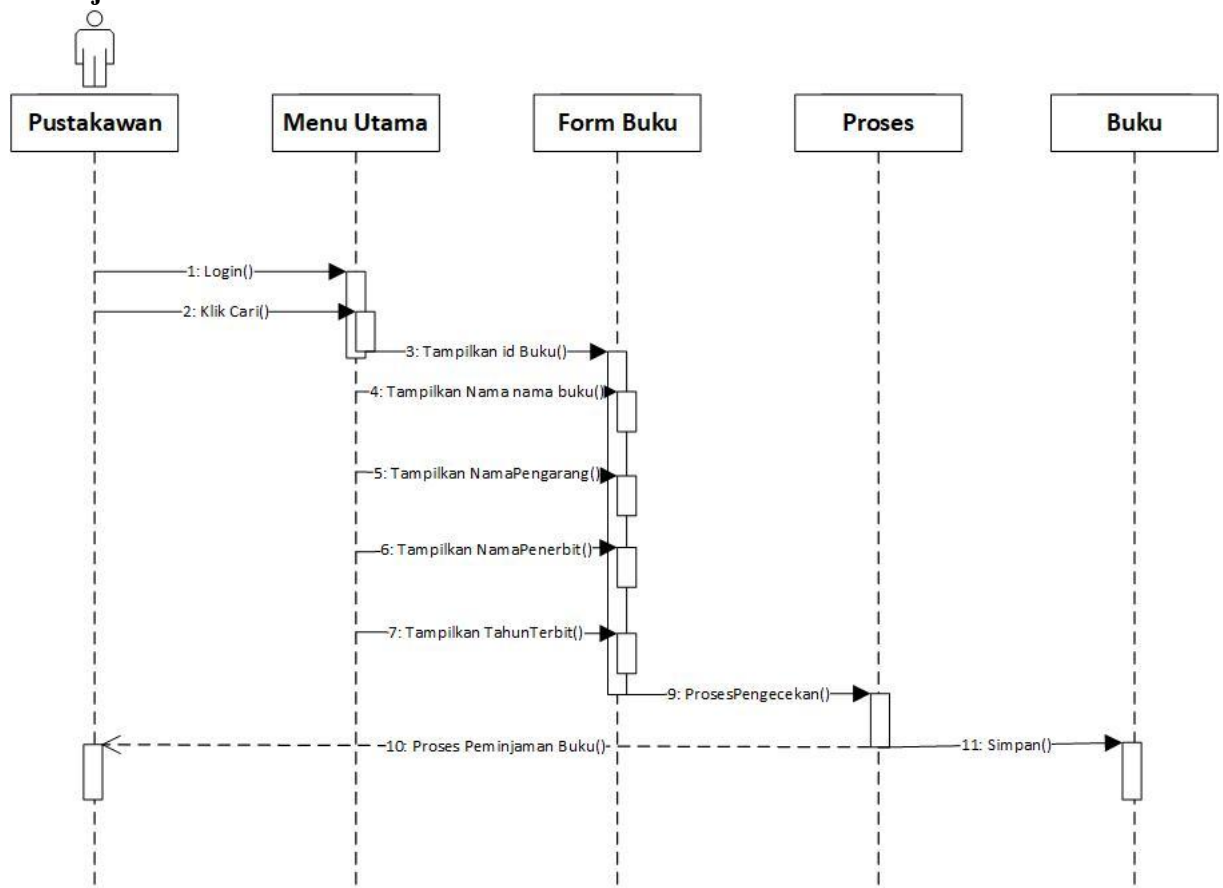
3.1.2.2 Sequence Diagram

Buatlah diagram sequence untuk setiap skenario use case. Skenario melibatkan kelas-kelas yang sudah diidentifikasi.

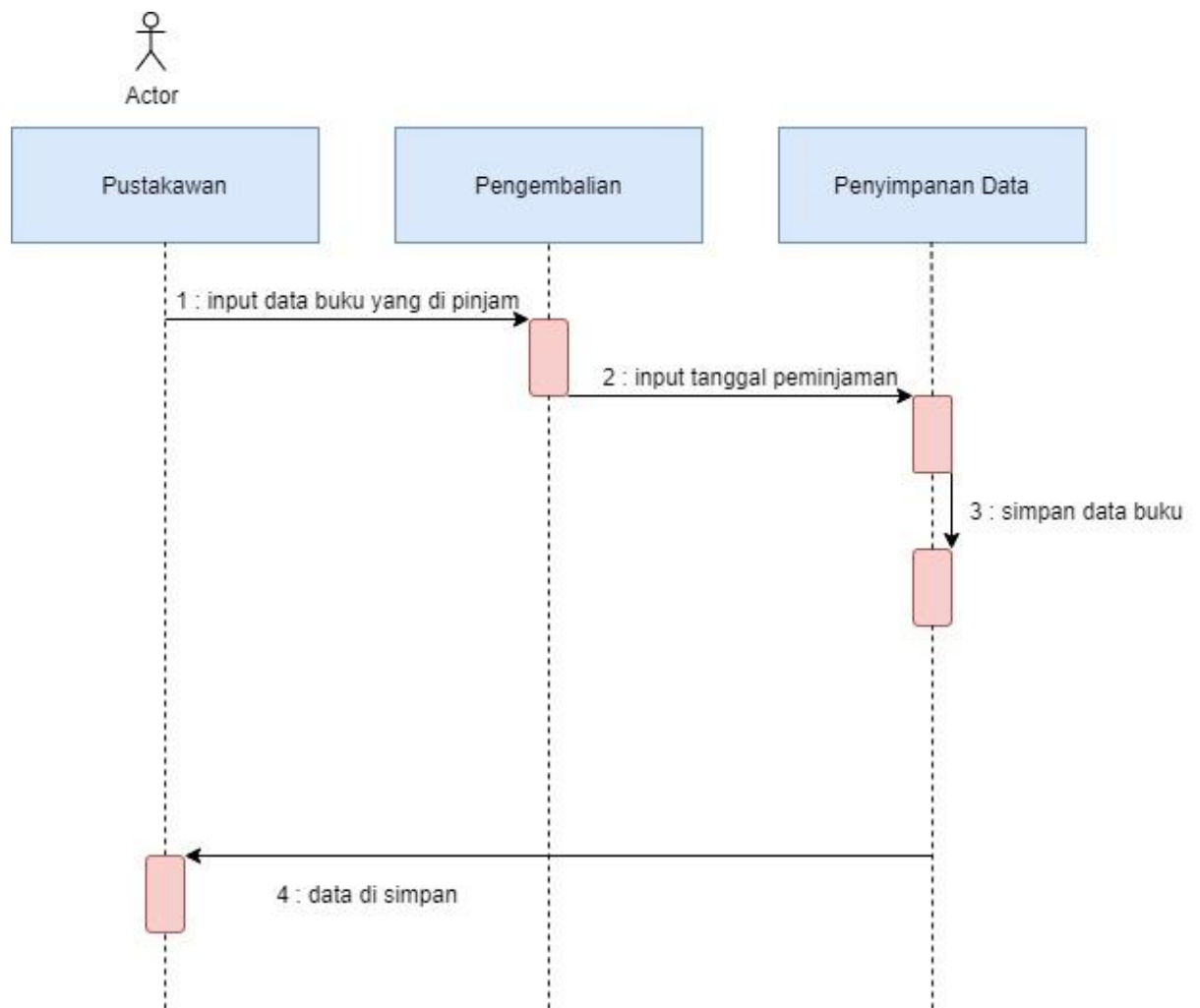
1. Buku



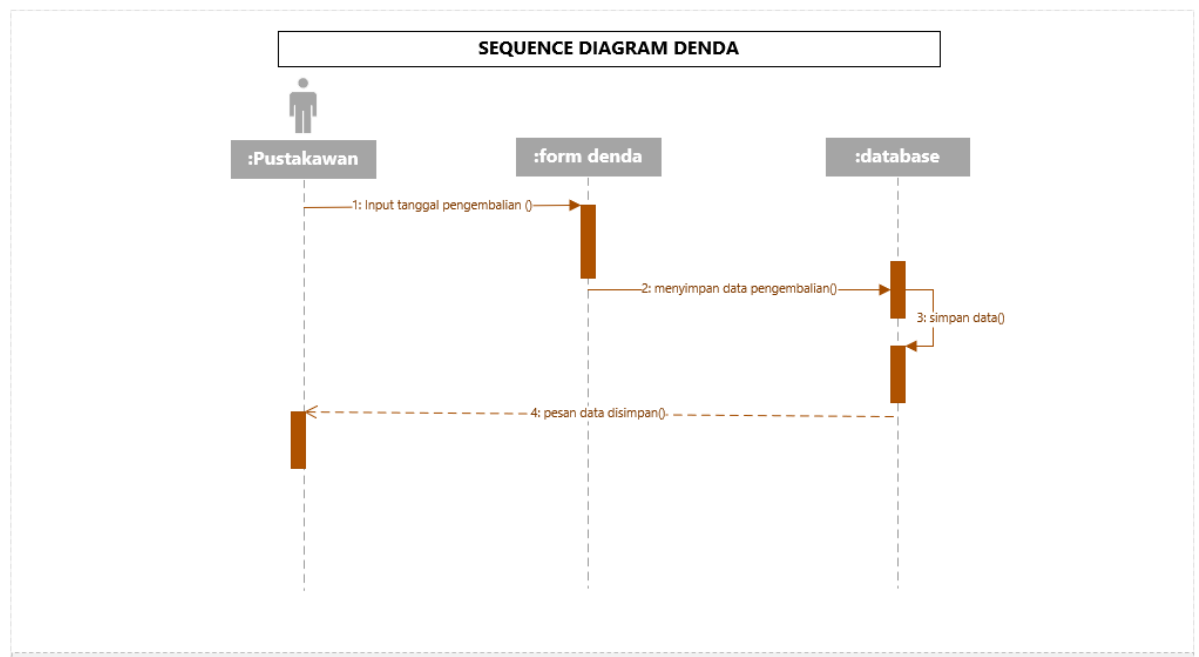
2. Peminjaman Buku



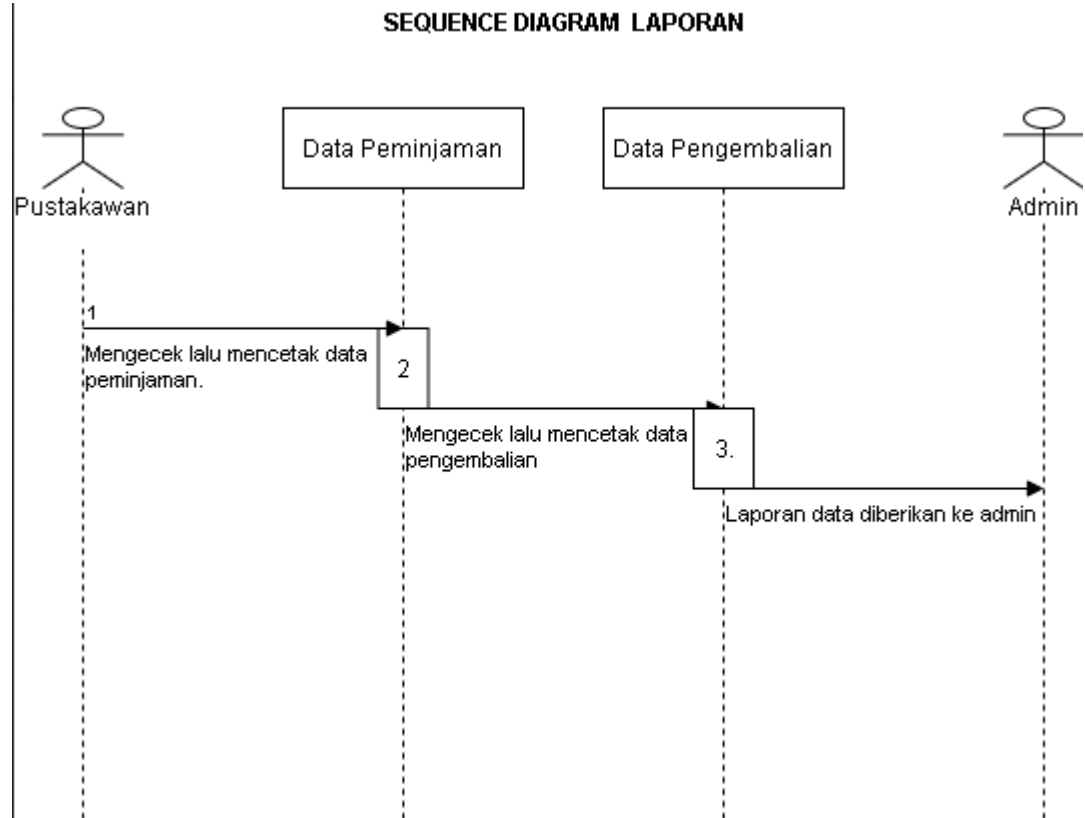
3. Pengembalian Buku



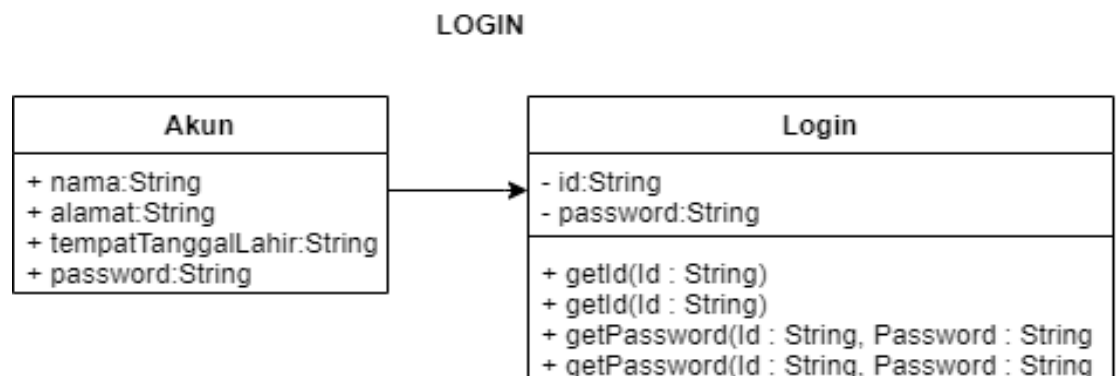
4. Denda



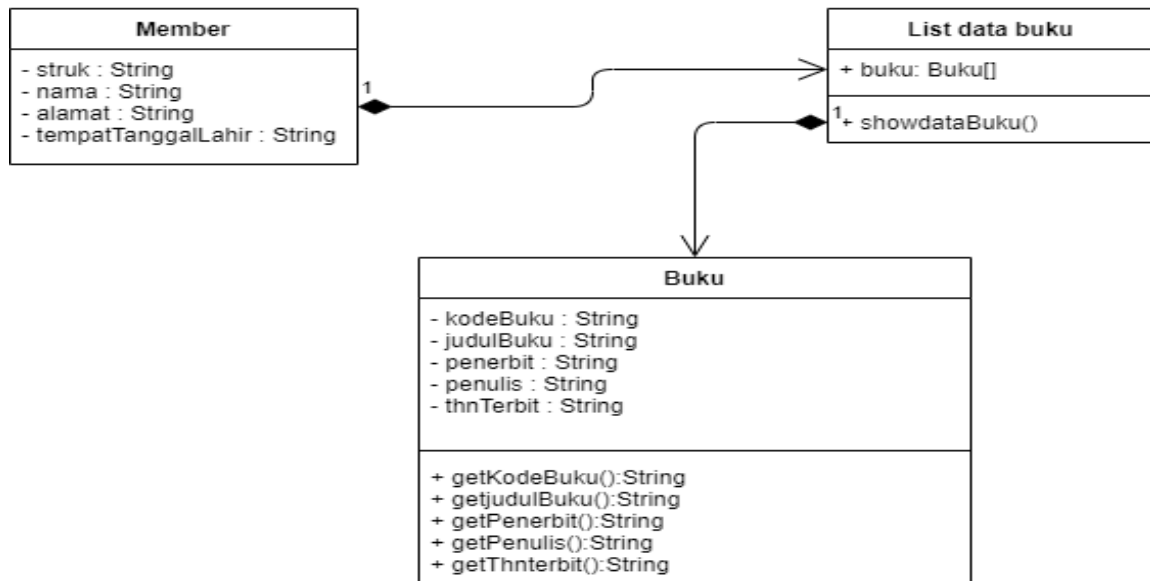
5. Laporan



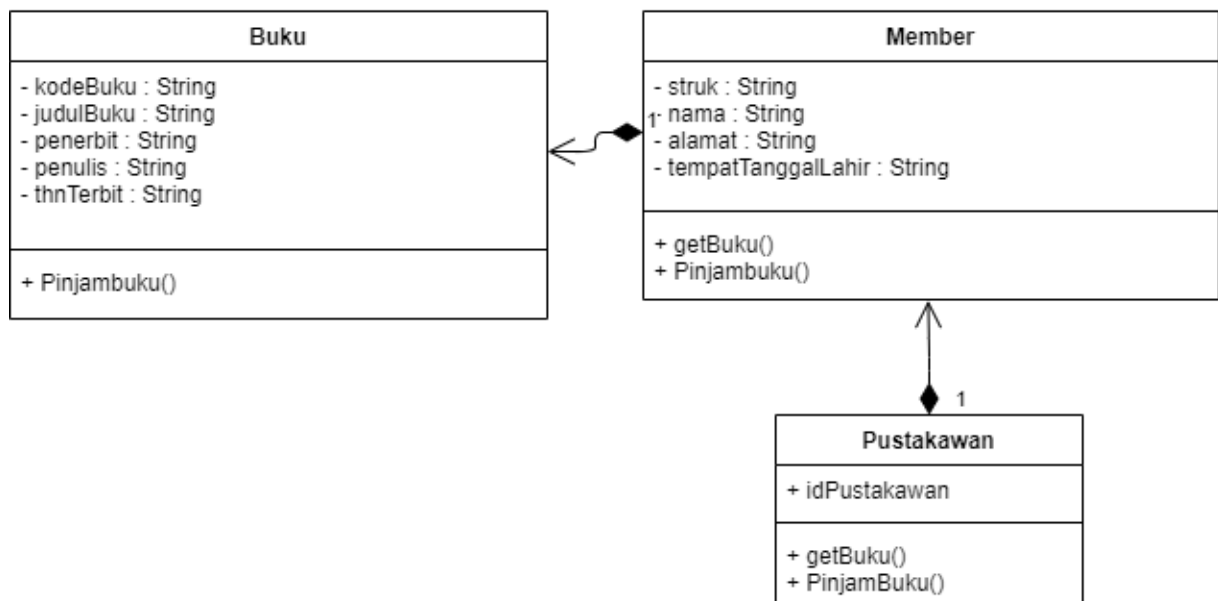
3.1.2.3 Diagram Kelas



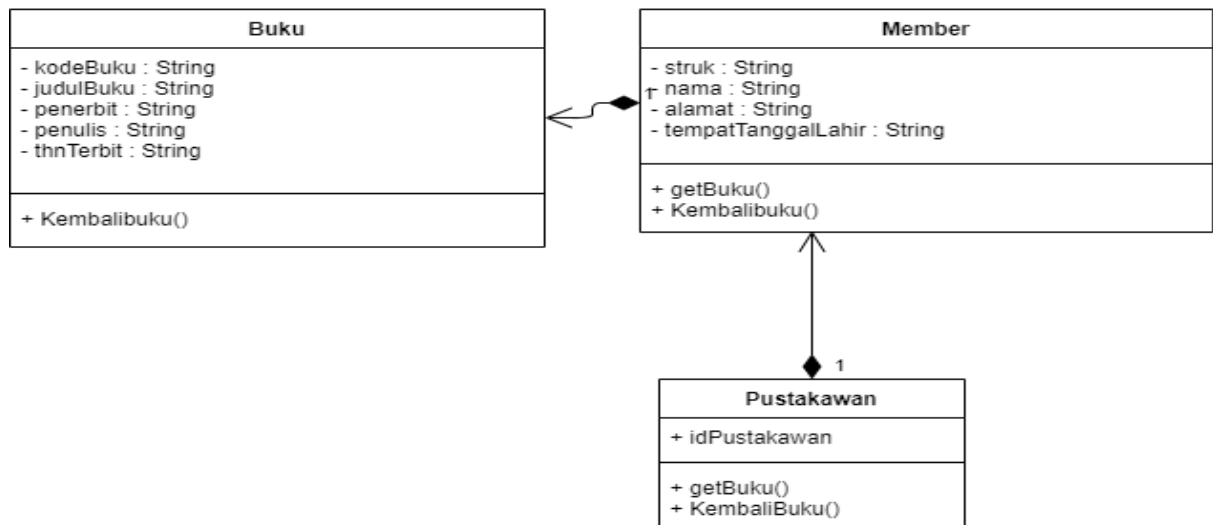
List data buku



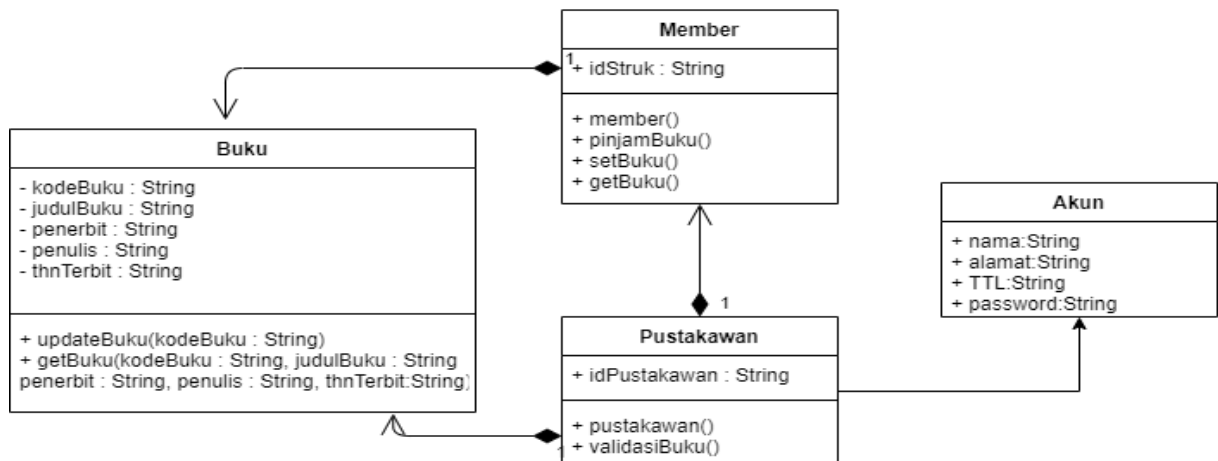
Pinjam buku



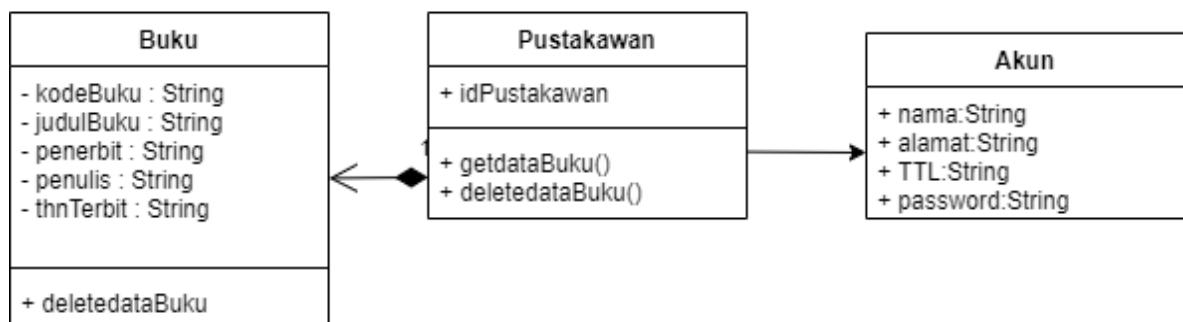
Kembali buku



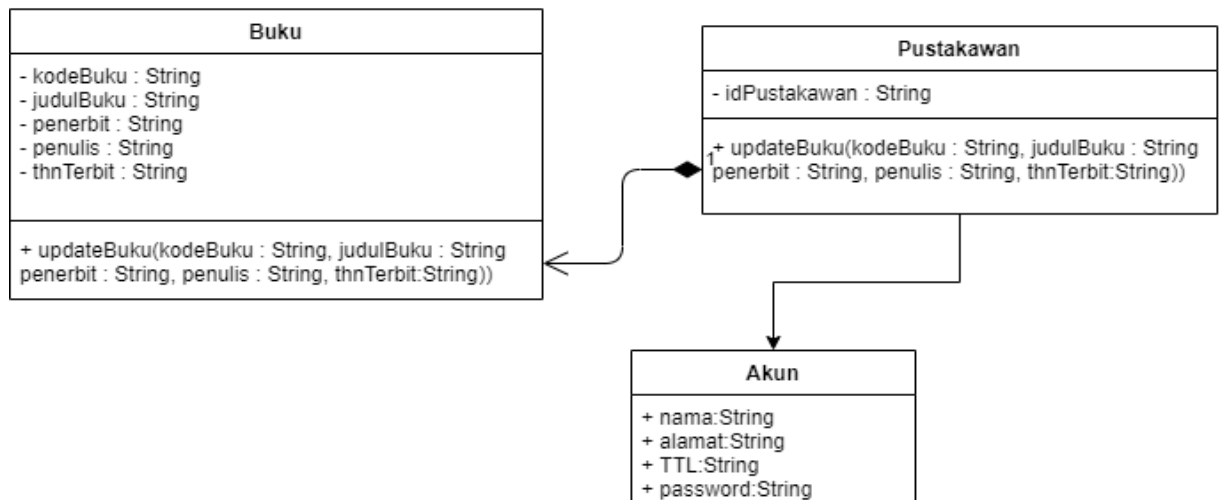
Input data buku



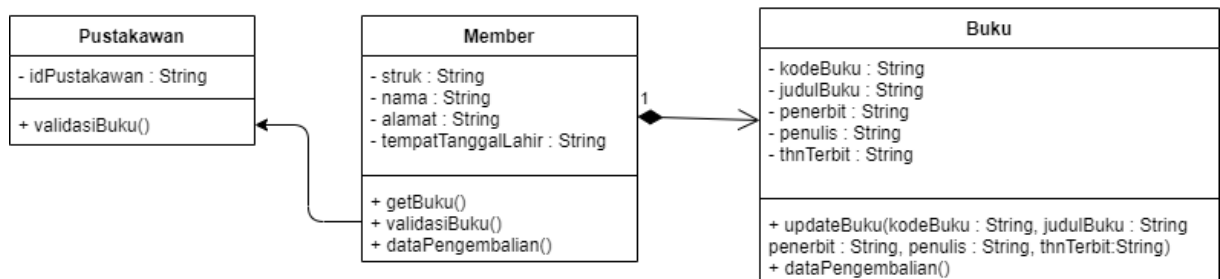
Delete Data Buku



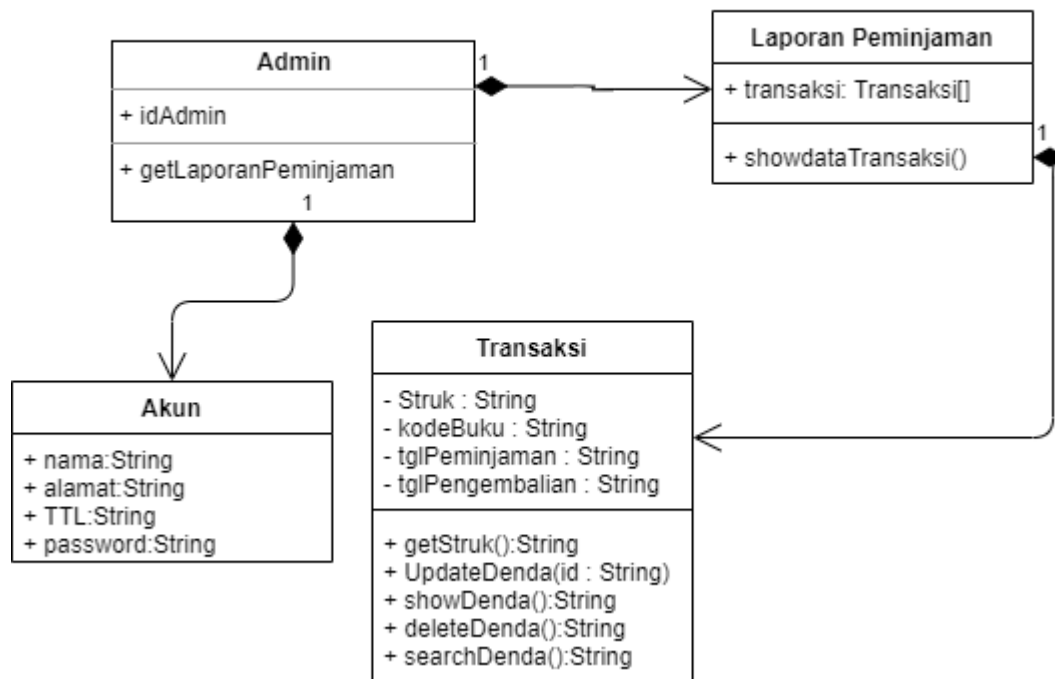
Update data buku



Input data pengembalian



Rekap Laporan Peminjaman



3.2 Perancangan Detail Kelas

Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Akun	Akun
2	Admin	Admin
3	Pustakawan	Pustakawan
4	Mahasiswa	Mahasiswa
5	Laporan	Laporan
6	Buku	Buku
7	Transaksi	Transaksi

Untuk setiap kelas:

- identifikasi operasi (mengacu pada tanggung-jawab kelas), termasuk visibility-nya
- identifikasi atribut, termasuk visibility-nya

3.2.1 Kelas <AKUN>

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

Nama Kelas : Akun

Nama Operasi	Visibility (private, public)	Keterangan
Diisi dengan signature operasi		

<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Nama	Private	String
Alamat	Private	String
tempatTanggalLahir	Private	String
Password	Private	String

3.2.2 Kelas <ADMIN>

Nama Kelas : Admin

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
Admin	Public	
getId()	Public	Method ini di gunakan untuk memanggil id admin.
getNama()	Public	Method ini di gunakan untuk memanggil nama admin.
getTTL()	Public	Method ini di gunakan untuk memanggil tempat tanggal lahir admin.
getPass()	Public	Method ini di gunakan untuk memanggil password admin.
deletePustakawan()	Public	Method ini di gunakan admin untuk menghapus data pustakawan.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
idAdmin	Private	String

3.2.3 Kelas <PUSTAKAWAN>

Nama Kelas : Pustakawan

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
Pustakawan	Public	
getId()	Public	Method ini di gunakan untuk memanggil Id pustakawan.
getNama()	Public	Method ini di gunakan untuk memanggil nama pustakawan.
getTTL()	Public	Method ini di gunakan untuk memanggil tempat tanggal lahir pustakawan.
getPass()	Public	Method ini di gunakan untuk memanggil password pustakawan.
deleteMahasiswa()	Public	Method ini di gunakan pustakawan untuk menghapus data Mahasiswa.
updateMahasiswa()	Public	Method ini di gunakan pustakawan untuk mengubah/menambah data Mahasiswa.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
idPustawakan	Private	String

3.2.4 Kelas <Mahasiswa>

Nama Kelas : Mahasiswa

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
Mahasiswa	Public	
getId()	Public	Method ini di gunakan untuk memanggil Id Mahasiswa.
getNama()	Public	Method ini di gunakan untuk memanggil nama Mahasiswa.
getTTL()	Public	Method ini di gunakan untuk memanggil tempat tanggal lahir Mahasiswa.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Nama	Private	String
Alamat	Private	String
tempatTanggalLahir	Private	String

3.2.5 Kelas <LAPORAN>

Nama Kelas : Laporan

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
showLaporan()	Public	Method ini di gunakan untuk menampilkan laporan.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
dataBuku	Private	String
dataPustakawan	Private	String
dataMahasiswa	Private	String
dataTransaksi	Private	String

3.2.6 Kelas <BUKU>

Nama Kelas : Buku

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
showBuku()	Public	Method ini di gunakan untuk menampilkan data buku.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
idBuku	Private	String
judulBuku	Private	String
kodeBuku	Private	String
penerbit	Private	String
penulis	Private	String
thnTerbit	Private	String

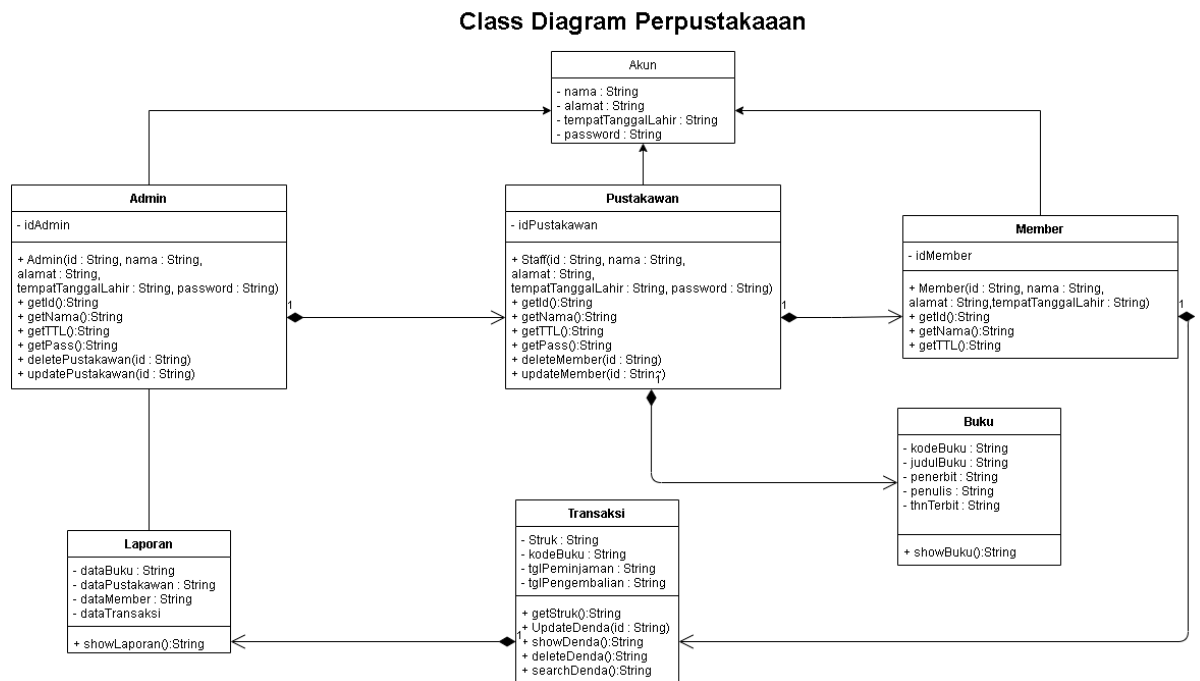
3.2.7 Kelas <TRANSAKSI>

Nama Kelas : Transaksi

<i>Nama Operasi</i>	<i>Visibility (private, public)</i>	<i>Keterangan</i>
getStruk()	Public	Method ini di gunakan untuk memanggil Struk.

UpdateDenda()	Public	Method ini di gunakan untuk memanggil denda.
showDenda()	Public	Method ini di gunakan untuk memperlihatkan denda.
deleteDenda()	Public	Method ini di gunakan untuk menghapus denda.
searchDenda()	Public	Method ini di gunakan untuk mencari denda.
<i>Nama Atribut</i>	<i>Visibility (private, public)</i>	<i>Tipe</i>
Struk	Private	String
kodeBuku	Private	String
tglPeminjaman	Private	String
tglPengembalian	Private	String

3.3 Diagram Kelas Keseluruhan



3.4 Algoritma/Query

Bagian ini hanya diisi untuk kerangka algoritma untuk **method-method** dari **Class** yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.

Contoh:

Nama Kelas :

Nama Operasi :

Algoritma :

(Algo-xxx)

{Jika mengacu query tertentu, lengkapi tabel query di bawah}

Query :

No Query	Query	Keterangan
Q-xxx		Tuliskan fungsi dari querynya

3.5 Perancangan Antarmuka

Bagian ini diisi dengan versi awal prototipe antarmuka.

Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya, misalnya seperti di bawah ini:

Antarmuka : {diisi dengan no. layar atau no gambar rancangan antarmuka}

Id_Objek	Jenis	Nama	Keterangan
		Diisi dengan string yg tampil pd layar	Diisi dengan penjelasan reaksi sistem, misalnya membuka layar apa, link kemana. Jika menyangkut suatu kode yang cukup rumit, acu algoritma yang telah diuraikan di atas.
Button1	Button	OK	Jika diklik, akan mengaktifkan Proses AlgoXXX.
RTF1	RTF Box		Isi Teks yang disimpan pada File xxx

3.6 Perancangan Representasi Persistensi Kelas

Bagian ini diisi dengan rancangan skema basisdata dan traceability-nya terhadap kelas entity. (PEMBUATAN SKEMA RELASI)

4 Matriks Kerunutan

Mapping use case dengan kelas-kelas terkait

<i>Requirement</i>	<i>Usecase Terkait</i>	<i>Kelas</i>
<i>FR-01</i>		
<i>FR-02</i>		