

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SISTEM INFORMASI PERPUSTAKAAN

untuk:

Telkom University

Dipersiapkan oleh :

1. Ridho Akbar (1301180290)
2. Daniel Septyadi (1301180009)
3. Mohammad Fajra S. (1301180370)
4. Gerald Ergi BN. (1301180133)
5. Mgs. M. Fakhri A. (1301180039)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-05 <xx:no grp>* | | *<35>* |
| Revisi | *<nomor revisi>* | *Tgl: 22-Apr-2020* |

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**](#_gjdgxs) **5**

[Tujuan Penulisan Dokumen](#_30j0zll) [6](#_30j0zll)

[Lingkup Masalah](#_1fob9te) [6](#_1fob9te)

[Definisi dan Istilah](#_3znysh7) [6](#_3znysh7)

[Referensi](#_2et92p0) [6](#_2et92p0)

[Sistematika Pembahasan](#_tyjcwt) [6](#_tyjcwt)

[**Deskripsi Perancangan Global**](#_3dy6vkm) **6**

[Rancangan Lingkungan Implementasi](#_1t3h5sf) [7](#_1t3h5sf)

[Deskripsi Arsitektural](#_4d34og8) [7](#_4d34og8)

[Deskripsi Komponen](#_2s8eyo1) [7](#_2s8eyo1)

[**Perancangan Rinci**](#_3rdcrjn) **8**

[Realisasi Use Case](#_26in1rg) [8](#_26in1rg)

[Use Case <nama use case 1>](#_lnxbz9) [8](#_lnxbz9)

[Identifikasi Kelas](#_35nkun2) [8](#_35nkun2)

[Sequence Diagram](#_1ksv4uv) [8](#_1ksv4uv)

[Diagram Kelas](#_z337ya) [8](#_z337ya)

[Perancangan Detil Kelas](#_3j2qqm3) [8](#_3j2qqm3)

[Kelas <nama kelas>](#_1y810tw) [8](#_1y810tw)

[Kelas <nama kelas>](#_4i7ojhp) [9](#_4i7ojhp)

[Diagram Kelas Keseluruhan](#_2xcytpi) [9](#_2xcytpi)

[Algoritma/Query](#_1ci93xb) [9](#_1ci93xb)

[Diagram Statechart](#_1pxezwc) [9](#_1pxezwc)

[Perancangan Antarmuka](#_3whwml4) [9](#_3whwml4)

[Perancangan Representasi Persistensi Kelas](#_2bn6wsx) [10](#_2bn6wsx)

[**Matriks Kerunutan**](#_qsh70q) **10**

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## 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 Perangcangan 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 refrensi untuk mengetahui bagaimana sistem bekerja lengkap dengan spesifikasi dan kebutuhan perangkat.

## 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 member (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 diperpustakaan, membuat akun pustakawan dan pustakawan akan melakukan pendataan, transaksi peminjaman dan pengembalian buku yang dilakukan oleh member.

## Definisi dan Istilah

|  |  |  |
| --- | --- | --- |
| No. | Definisi | Keterangan |
|  | SKPL | Spesifikasi Kebutuhan Perangkat Lunak dan merupakan spesifikasi dari perangkat lunak yang akan dikembangkan. |
|  | DPPL | Deskripsi Perancangan Perangkat Lunak dan merupakan dokumen sebagai penjelasan dan gambaran tentang rancangan yang dibuat. |
|  | Pustakawan | Merupakan seseorang yang mengelola data buku dan data member. |
|  | Member | Merupakan seseorang yang melakukan transaksi peminjaman dan pengembalian di perpustakaan. |
|  | DBMS | Singkatan dari “*Database Management System*” yaitu sistem pengorganisasian dan sistem Database pada Komputer. |
|  | MySQL | Perangkat lunak sistem manajemen basis data SQL atau DBMS |
|  | PHP, HTML, JavaScript | merupakan bahasa skrip yang digunakan untuk membuat Sistem Informasi. |
|  | XAMPP | berfungsi sebagai server lokal untuk mengampu berbagai jenis data website yang sedang dalam proses pengembangan |

## Referensi

Dokumen acuan yang digunakan dalam penulisan dokumen ini adalah :

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

## 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 :

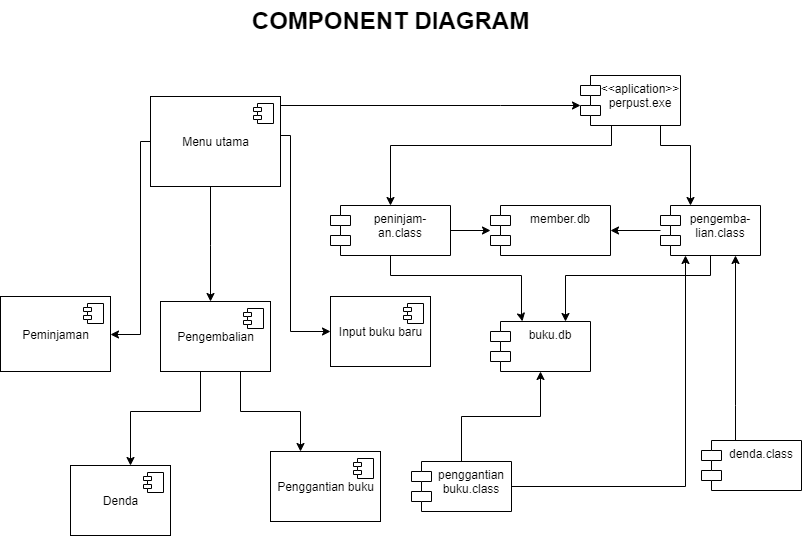
1. Bab 1 Pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah yang ditangani dalam perangkat lunak yang akan dibangun serta deskripsi umum dokumen.
2. Bab 2 Deskripsi Perancangan Global, berisi penjelasan perangkat lunak yang akan diimplementasikan di lingkungan pengguna secara global dan bagaimana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

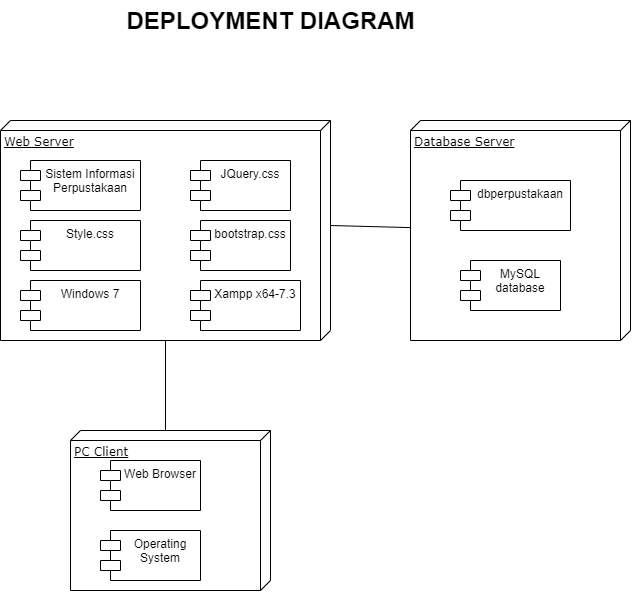
# Deskripsi Perancangan Global

## Rancangan Lingkungan Implementasi

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

## Deskripsi Arsitektural





## 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 diekseskusi |
| 8. | peminjaman.class | Fungsi peminjaman |
| 9. | member.db | Database member |
| 10. | pengembalian.class | Fungsi pengembalian |
| 11. | buku.db | Database buku |
| 12. | penggantianbuku.class | Fungsi penggantian buku |
| 13. | denda.class | Fungsi denda |

# Perancangan Rinci

## Realisasi Use Case

### Use Case

### 

* + 1. **Usecase Scenario**

1. Nama Use case : Login

Aktor : Member,Pustakawan, Admin

Pre-condition : Aktor belum masuk ke sistem

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

Deskripsi : Aktor melakukan login ke dalam sistem

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
|  | 1. Menampilkan laman login |
| 2. Menginput data akun |  |
|  | 3. Memvalidasi data login |
| 4. Sudah masuk ke dalam sistem |  |

2. Nama Use case : List Data Buku

Aktor : Member

Pre-condition : Member belum melihat Data buku

Post-condition : Member sudah melihat Data Buku

Deskripsi : Aktor telah melihat info/data buku dan mendapatkan list data buku

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
|  | 1. Menampilkan laman 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 : Member

Pre-condition : Member belum meminjam buku

Post-condition : Member sudah meminjam Data Buku

Deskripsi : Aktor telah meminjam buku yang dipilih

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
|  | 1. Menampilkan laman berisi form peminjaman |
| 2. Mengisi form peminjaman buku |  |
|  | 3. Menampilkan laman berdasarkan kode buku |
| 4. Menginputkan kode buku tsb. |  |
|  | 5. Berhasil menambahkan buku |

4. Nama Use case : Kembali Buku

Aktor : Member

Pre-condition : Member belum mengembalikan Buku

Post-condition : Member sudah mengembalikan Buku

Deskripsi : Aktor telah mengembalikan buku yang dipinjam

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
|  | 1. Menampilkan laman berisi form pengembalian |
| 2. Mengisi form pengembalian buku |  |
|  | 3. Menampilkan form laman pengembalian berdasarkan kode buku |
| 4. Menginputkan kode buku yg telah dipinjam. |  |
|  | 5. Buku berhasil dikembalikan |

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 disistem |

7. Nama Use case : Update 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 *database* yang ada |

8. Nama Use case : Input data Peminjaman

Aktor : Pustakawan

Pre-condition : Pustakawan belum menginputkan data peminjam buku

Post-condition : Pustakawan sudah menginputkan data peminjam buku

Deskripsi : Aktor telah menginputkan data peminjam buku

Tabel skenario

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

9. Nama Use case : Input data Pengembalian

Aktor : Pustakawan

Pre-condition : Pustakawan belum menginputkan data pengembalian buku

Post-condition : Pustakawan sudah menginputkan data pengembalian buku

Deskripsi : Aktor telah menginputkan data pengembalian buku

Tabel skenario

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

9. Nama Use case : Delete data peminjaman

Aktor : Pustakawan

Pre-condition : Pustakawan belum menghapus data peminjaman

Post-condition : Pustakawan sudah menghapus data peminjaman

Deskripsi : Pustakawan menghapus data peminjam yang sudah selesai atau lewat masa batas

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Mengakses data peminjaman |  |
|  |  |
| 2. Menghapus data peminjaman |  |
|  | 3. data peminjam tidak ada disistem |

10. Nama Use case : Update data pengembalian

Aktor : Pustakawan

Pre-condition : Komponen data belum berubah

Post-condition : Komponen data berhasil diubah

Deskripsi : Pustakawan memperbaharui data peminjaman yang sudah selesai atau lewat masa batas yg ditentukan

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Memilih menu data pengembalian |  |
|  | 2. Menampilkan informasi data pengembalian |
| 3. Memilih opsi edit |  |
|  | 4. Menampilkan form data pengembalian yang ingin diubah |
| 5. Mengedit bagian-bagian data pengembalian yang akan dirubah berupa info pengembalian |  |
| 6. Menyimpan hasil perubahan |  |
|  | 7. Memperbarui *database* yang ada |

11. Nama Use case : Input akun member

Aktor : Pustakawan

Pre-condition : Pustakawan belum me-registrasi data member baru

Post-condition : Pustakawan sudah me-registrasi data member baru

Deskripsi : Aktor telah menginputkan data member baru

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
|  | 1. Menampilkan laman berisi form data registrasi akun member |
| 2. Mengisi form data akun member sesuai profile user. |  |
|  | 3. Merekam proses penginputan data member lalu tersimpan ke database |
| 4. Menerima feedback dari sistem, data member berhasil diregistrasi. |  |

12. Nama Use case : Input data pustakawan

Aktor : Admin

Pre-condition : Admin belum me-registrasi data akun pustakawan

Post-condition : Admin sudah me-registrasi data akun pustakawan

Deskripsi : Aktor telah me-registrasi data akun pustkawan.

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
|  | 1. Menampilkan laman berisi form data registrasi akun pustakawan. |
| 2. Mengisi form data akun pustakawan |  |
|  | 3. Merekam proses penginputan data pustakawan lalu tersimpan ke database |
| 4. Menerima feedback dari sistem, data pustakawan berhasil diregistrasi. |  |

13. Nama Use case : Update data pustakawan

Aktor : Admin

Pre-condition : Komponen akun belum berubah

Post-condition : Komponen akun berhasil diubah

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

Tabel skenario

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

14. Nama Use case : Blokir data akun

Aktor : Admin

Pre-condition : Admin belum mendapatkan user yang tidak resign atau tidak sedang cuti

Post-condition : Admin mendapatkan user yang resign ato sedang cuti

Deskripsi : Aktor telah memblokir data akun pustakawan

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Membuka menu admin |  |
| . | 2. Menampilkan pengelolaan data akun sesuai role |
| 3. Menginputkan profile user sesuai role yang ingin diblokir |  |
|  | 4. Menerima feedback dari sistem, data akun berhasil diblokir |

15. Nama Use case : UnBlokir data akun

Aktor : Admin

Pre-condition : Admin mendapatkan user yang resign atau sedang cuti

Post-condition : Admin belum mendapatkan user yang tidak resign atau tidak sedang cuti

Deskripsi : Aktor telah me-unblokir data akun pustakawan dari sistem

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Membuka menu admin |  |
| . | 2. Menampilkan pengelolaan data akun pustakawan |
| 3. Menginputkan profile user yang ingin *diunblock* |  |
|  | 4. Menerima feedback dari sistem, data akun pustakawan berhasil *diunblock* |

16. Nama Use case : Rekap laporan peminjaman

Aktor : Admin

Pre-condition : Admin belum menerima laporan peminjaman

Post-condition : Admin sudah menerima laporan peminjaman

Deskripsi : Aktor telah menerima rekap laporan peminjaman perpustakaan tsb.

Tabel skenario

|  |  |
| --- | --- |
| Aktor | Sistem |
| 1. Membuka Menu Perpustakaan |  |
|  | 2. Menampilkan pengelolaan laporan data perpustakaan |
| 3. Klik Laporan Peminjaman |  |
|  | 4. Disimpan berupa format file |
| 5. Data rekap berhasil disimpan dikomputer. |  |

#### 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:*

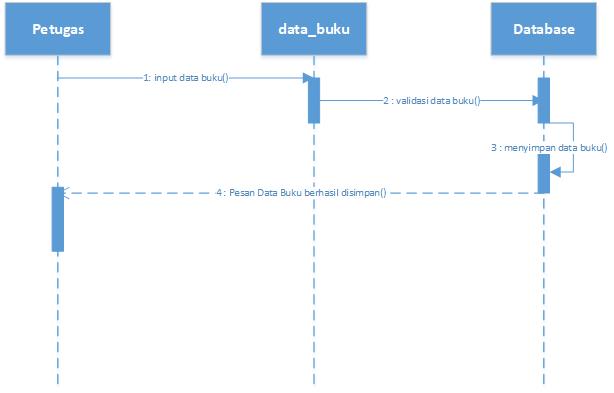
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| *1* | *Member* | *Controller* |
| *2* | *Pustakawan* | *Controller* |
| *3* | *Admin* | *Controller* |
| *4* | *Buku* | *Controller* |
| *5* | *Laporan* | *Controller* |
| *6* | *Denda* | *Controller* |

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

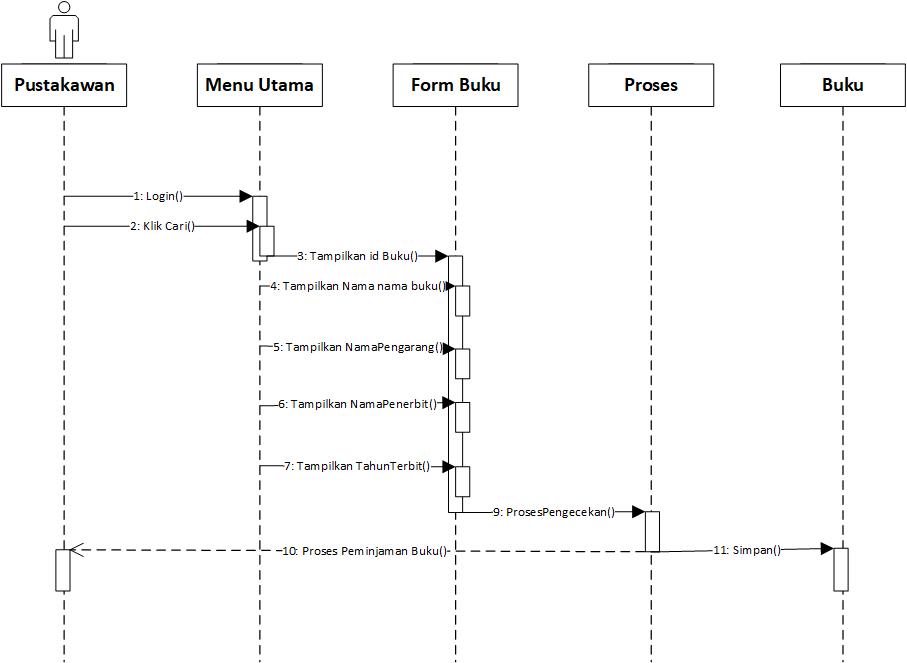
#### Sequence Diagram

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

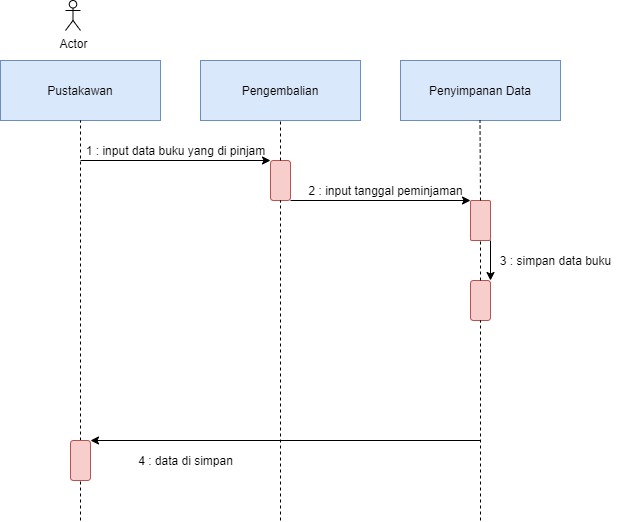
1. **Buku**



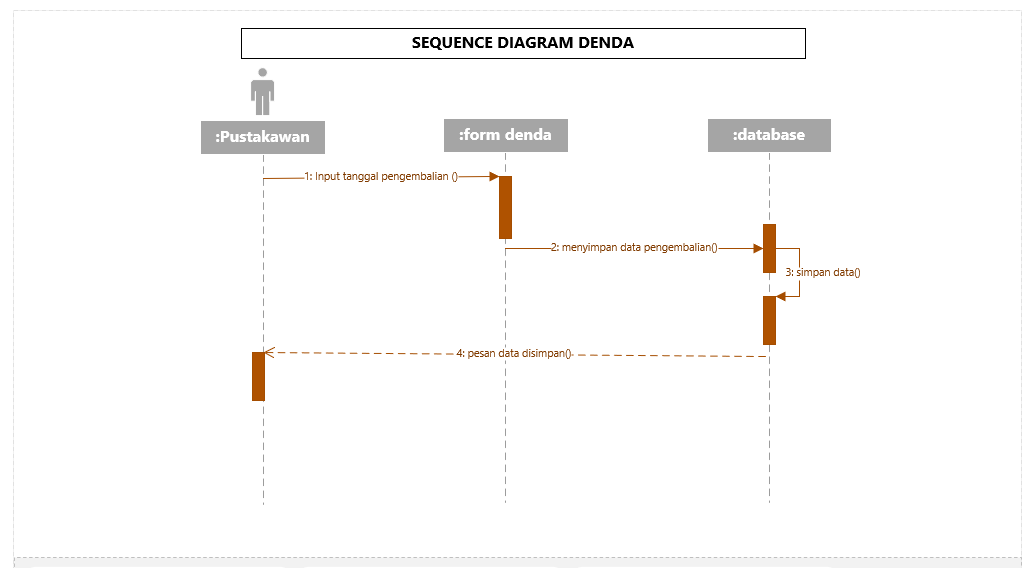
1. **Peminjaman Buku**



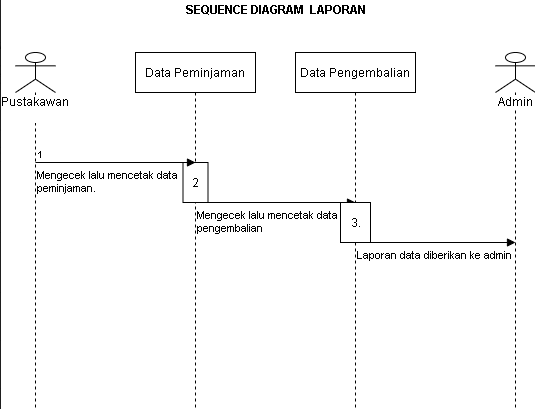
1. **Pengembalian Buku**



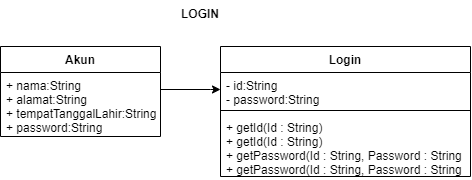
1. **Denda**

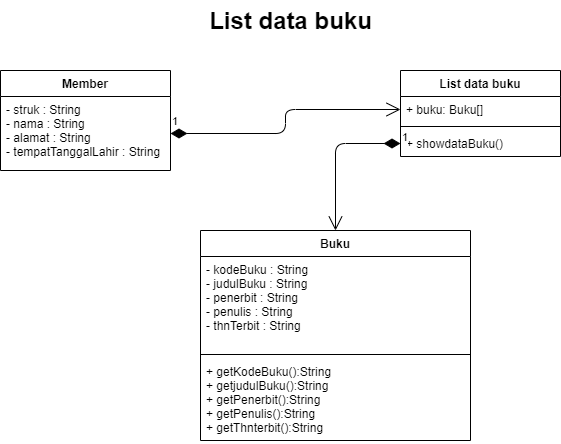
****

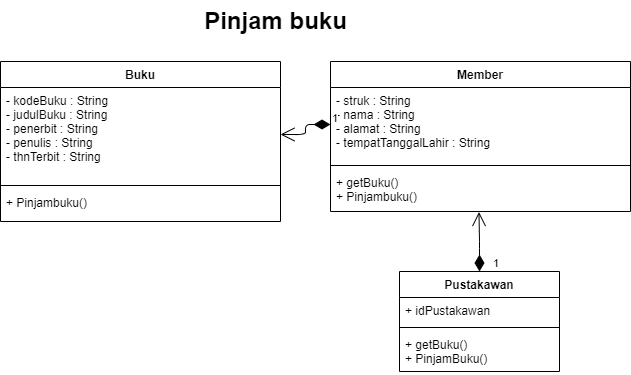
1. **Laporan**

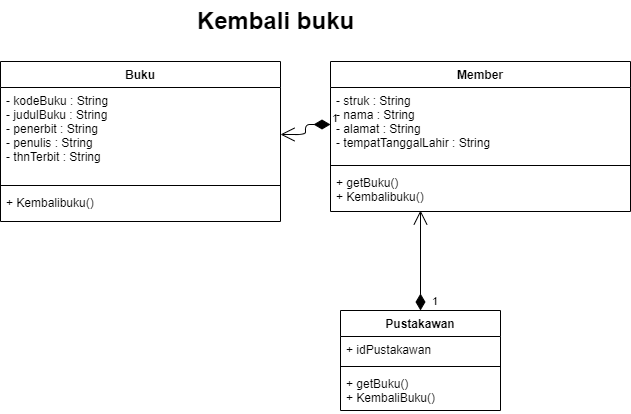
****

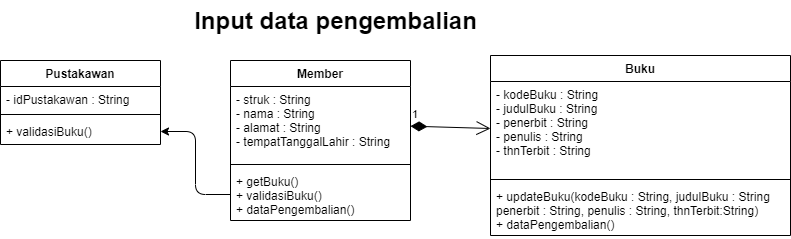
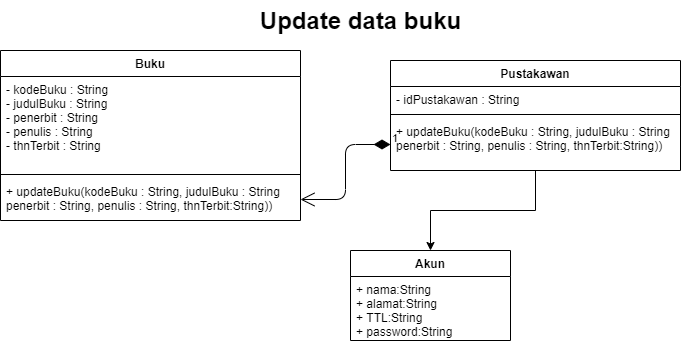
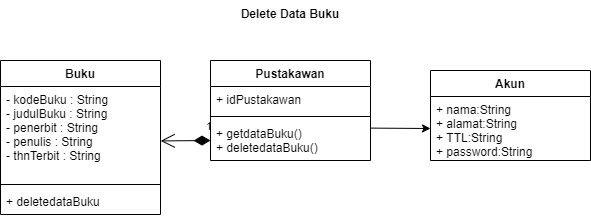
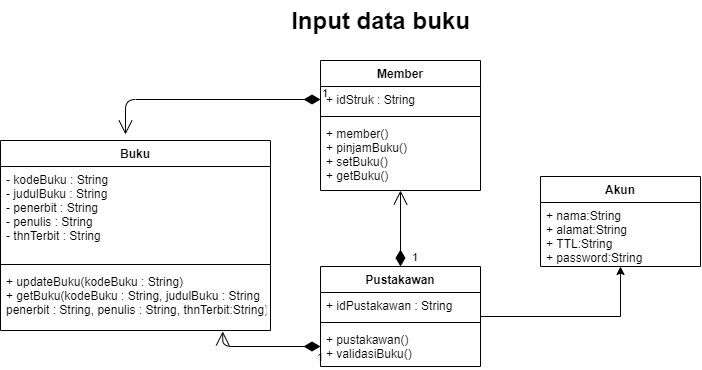
#### Diagram Kelas

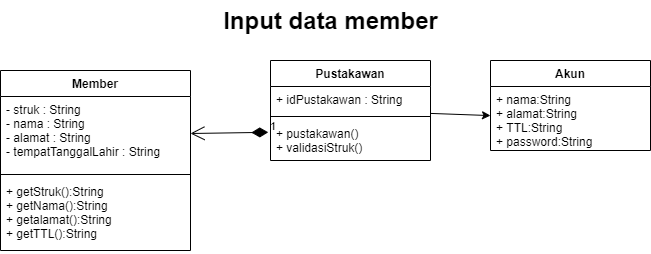
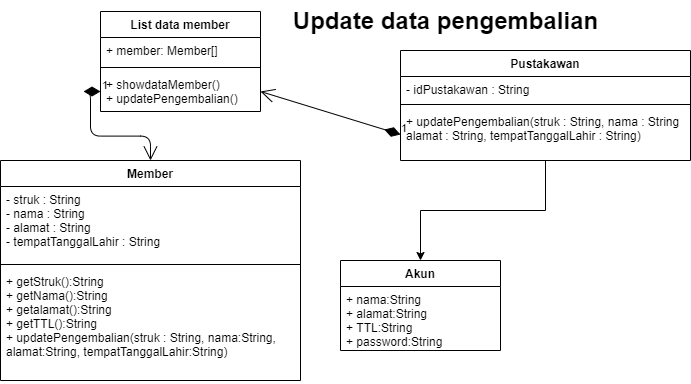
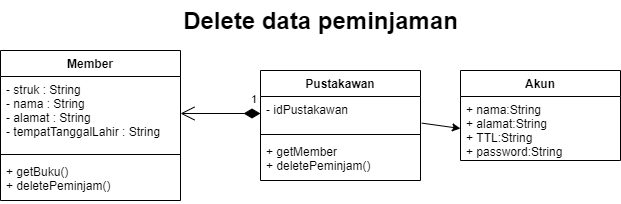
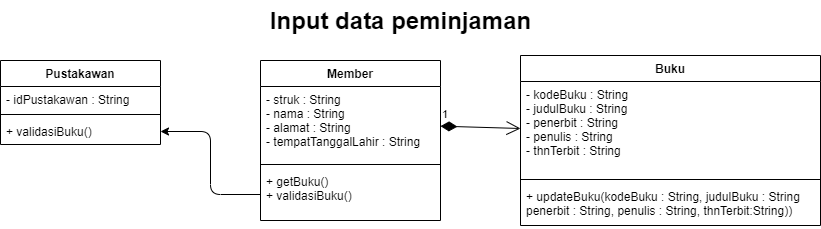


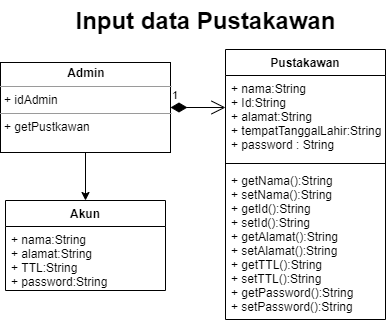


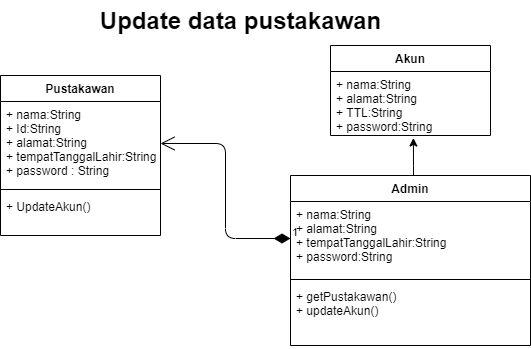


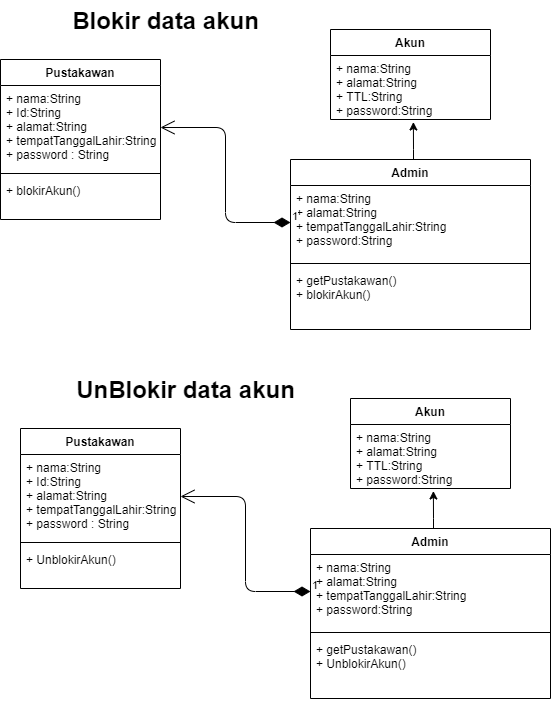


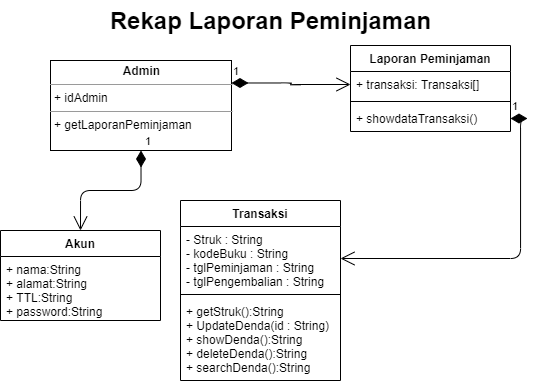












## Perancangan Detil Kelas

*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 | Member | Member |
| 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*

### 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* |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Nama | Private | String |
| Alamat | Private | String |
| tempatTanggalLahir | Private | String |
| Password | Private | String |

### Kelas <ADMIN>

Nama Kelas : Admin

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| idAdmin | Private | String |

**3.2.3 Kelas <PUSTAKAWAN>**

Nama Kelas : Pustakawan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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. |
| deleteMember() | Public | Method ini di gunakan pustakawan untuk menghapus data member. |
| updateMember() | Public | Method ini di gunakan pustakawan untuk mengubah/menambah data member. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| idPustawakan | Private | String |

**3.2.4 Kelas <MEMBER>**

Nama Kelas : Member

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Member | Public |  |
| getId() | Public | Method ini di gunakan untuk memanggil Id member. |
| getNama() | Public | Method ini di gunakan untuk memanggil nama member. |
| getTTL() | Public | Method ini di gunakan untuk memanggil tempat tanggal lahir member. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Nama | Private | String |
| Alamat | Private | String |
| tempatTanggalLahir | Private | String |

**3.2.5 Kelas <LAPORAN>**

Nama Kelas : Laporan

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| showLaporan() | Public | Method ini di gunakan untuk menampilkan laporan. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| dataBuku | Private | String |
| dataPustakawan | Private | String |
| dataMember | Private | String |
| dataTransaksi | Private | String |

**3.2.6 Kelas <BUKU>**

Nama Kelas : Buku

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| showBuku() | Public | Method ini di gunakan untuk menampilkan data buku. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| 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

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| 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. |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Struk | Private | String |
| kodeBuku | Private | String |
| tglPeminjaman | Private | String |
| tglPengembalian | Private | String |

## Diagram Kelas Keseluruhan

## 

## 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 :* Transaksi

*Nama Operasi :* Peminjaman Buku

*Algoritma :(Algo-001)*

1. Setelah member mendapat buku yang akan dipinjam, member menentukan tanggal peminjaman dan pengembalian buku, lalu melapor kepada pustakawan

2. Pustakawan akan menginputkan (kode\_buku) dan (nama\_member), peminjaman akan di insert ke database.

3. Sistem akan memberikan tanggal peminjaman berdasarkan tanggal member melakukan transaksi peminjaman tersebut.

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | Insert into tbTransaksi (Struk, kodeBuku, tgl\_peminjaman, tgl\_kembali) VALUES ($Struk, $kodeBuku, $tgl\_peminjaman, $tgl\_pengembalian); | *Menginputkan data peminjaman buku ke database* |
|
|

*Nama Kelas :* Transaksi

*Nama Operasi :* Pengembalian Buku

*Algoritma :(Algo-002)*

1. Setelah jatuh tempo tanggal pengembalian, member akan mengembalikan buku kepada pustakawan.

2. Pustakawan akan menerima buku dari member dan akan menginputkan (tgl\_peminjaman) dan (nama\_member).

3. Sistem akan menyimpan data dan mengeluarkan struk.

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-002* | Insert into tbTransaksi (Struk, kodeBuku, tgl\_pengembalian) VALUES ($Struk, $kodeBuku, $tgl\_pengembalian); | *Menginputkan data pengembalian buku ke database* |
|
|

*Nama Kelas :* Buku

*Nama Operasi :* Tambah Data Buku

*Algoritma :(Algo-003)*

1. Pustakawan akan menambahkan buku ke dalam database.

2. Pustakawan akan menginputkan (kodeBuku, judulBuku, penerbit, pengarang, tahun ) akan di insert ke database.

3. Sistem akan menyimpan ke dalam database buku

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-003* | Insert into tbBuku (kodeBuku, judulBuku, penerbit, pengarang, tahun) VALUES ($kodeBuku, $judulBuku, $penerbit, $pengarang, $tahun); | *Menginputkan data buku ke dalam database.* |
|
|

*Nama Kelas :* Buku

*Nama Operasi :* Edit Data Buku

*Algoritma :(Algo-004)*

1. Pustakawan akan mengedit data buku yang ada di database buku.

2. Pustakawan akan mengedit salah satu atau beberapa data yang ingin di ubah (kodeBuku, judulBuku, penerbit, pengarang, tahun ) akan di update ke database.

3. Sistem akan menyimpan ke dalam database buku

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-004* | Update tbBuku (kodeBuku, judulBuku, penerbit, pengarang, tahun) VALUES ($kodeBuku, $judulBuku, $penerbit, $pengarang, $tahun); | *Mengupdate data buku ke dalam database.* |
|
|

*Nama Kelas :* Buku

*Nama Operasi :* Delete Data Buku

*Algoritma :(Algo-005)*

1. Pustakawan akan menginput data buku yang akan dihapus pada database buku.

2. Pustakawan akan mensearch dan menghapus search berdasarkan (kode\_buku), setelah kode\_buku ketemu maka akan muncul semua data buku.

3. Sistem akan menghapus data buku pada database.

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-005* | Select kodebuku FROM buku;  Delete tbBuku (kodeBuku, judulBuku, penerbit, pengarang, tahun) VALUES ($kodeBuku, $judulBuku, $penerbit, $pengarang, $tahun); | *Menghapus data buku dari database.* |
|
|

*Nama Kelas :* Admin

*Nama Operasi :* Menambahkan data Pustakawan

*Algoritma :(Algo-006)*

1. Admin akan menambahkan data pustakawan ke dalam database pustakawan.
2. Data Pustkawakan akan di insert ke database dengan (id\_pustakawan, nama, TTL, password)
3. Sistem akan menyimpan data pada database pustakawan.

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-006* | Insert into tbPustakawan (id\_pustakawan, nama, TTL, password) VALUES ($id\_pustakawan, $nama, $TTL, $password); | *Menginput data pustakawan ke database* |
|
|

*Nama Kelas :* Admin

*Nama Operasi :* Edit data Pustakawan

*Algoritma :(Algo-007)*

1. Admin akan mengedit data pustakawan berdasarkan data yang ada di database pustakawan.
2. Data pustakawan yang akan di edit oleh admin adalah berdasarkan salah satu atau beberapa variabel (misal : nama)
3. Sistem akan mengedit dan menyimpan data pada database pustkawan.

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-007* | Update tbPustakawan (id\_pustakawan, nama, TTL, password) VALUES ($id\_pustakawan, ‘bambang’, $TTL, $password); | *Menginput data pustakawan ke database* |
|
|

*Nama Kelas :* Admin

*Nama Operasi :* Delete data Pustakawan

*Algoritma :(Algo-008)*

1. Admin akan menghapus data pustakawan.
2. Admin akan menghapus data pustakwan berdasarkan (id\_pustakawan)
3. Sistem akan menghapus dan menyimpan data pada database pustakawan.

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-008* | Select (id\_pustakwan) FROM Pustakawan;  Delete tbPustakawan (id\_pustakawan, nama, TTL, password) VALUES ($id\_pustakawan, ‘bambang’, $TTL, $password); | *Menginput data pustakawan ke database* |
|
|

*Nama Kelas :* Laporan

*Nama Operasi :* Laporan Data Buku

*Algoritma :(Algo-009)*

1. Setelah transaksi sudah dilakukan, data transaksi akan masuk ke database
2. Pustakawan dapat melihat laporan data transaksi peminjaman dan pengembalian buku

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-009* | Select \* from tbLaporan; | *Menampilkan laporan* |
|
|

*Nama Kelas :* Buku

*Nama Operasi :* Update data Buku

*Algoritma :(Algo-010)*

1. Pustakawan mengubah data buku , sesuai dengan privilege member & poin-poin yang ingin dirubah  
2. Pustakawan memilih opsi edit yang sudah disediakan berupa form data buku  
3. Pustakawan mengedit bagian-bagian data yang ingin diubah & menyimpan hasil perubahan ke dalam database

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-010* | Update tbBuku (kodeBuku, judulBuku, penerbit, pengarang, tahun) VALUES ($kodeBuku, $judulBuku, $penerbit, $pengarang, $tahun); | *Mengupdate data buku ke databasea* |
|
|

*Nama Kelas :* Buku

*Nama Operasi :* Update data Pengembalian

*Algoritma :(Algo-011)*

1. Pustakawan mengubah data pengembalian , sesuai dengan privillege member/poin-poin yang ingin dirubah  
2. Pustakawan memilih opsi edit yang sudah disediakan berupa form data pengemba  
3. Pustakawan mengedit bagian-bagian data peminjaman yang ingin diubah & menyimpan hasil perubahan ke dalam database

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-011* | Update tbPengembalian (kodeBuku, judulBuku, penerbit, pengarang, tahun) VALUES ($kodeBuku, $judulBuku, $penerbit, $pengarang, $tahun); | *Mengupdate data pengembalian ke database* |
|
|

*Nama Kelas :* Pustakawan

*Nama Operasi :* Input akun member

*Algoritma :(Algo-012)*

1. Pustakawan akan menambahkan data member ke dalam database member.
2. Data Member akan di insert ke database dengan (id\_Struk, nama, TTL, password)
3. Sistem akan menyimpan data pada database pustakawan.

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-012* | Insert into tbMember (id\_Struk, nama, TTL, password) VALUES ($id\_Struk, $nama, $TTL, $password); | *Membuat identitas member berdasarkan struk, nama, dan tanggal lahir yang akan diinputkan ke database.* |
|
|

*Nama Kelas :* Pustakawan

*Nama Operasi :* Delete data peminjaman

*Algoritma :(Algo-013)*

1. Pustakawan akan menghapus data peminjaman.
2. Pustakawa akan menghapus data pustakwan berdasarkan (id\_Struk) jika sudah melewati batas peminjaman atau kadaluarsa
3. Sistem akan menghapus dan menyimpan data pada database pustakawan.

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-013* | Delete into tbPeminjaman (id\_Struk, nama, TTL, password) VALUES ($id\_Struk, $nama, $TTL, $password); | *Menghapus data peminjaman yang sudah melewati batas peminjaman ke database.* |
|
|

*Nama Kelas :* Transaksi

*Nama Operasi :* Show denda

*Algoritma :(Algo-014)*

1. Denda akan muncul pada saat member telah melewati batas jatuh tempo pengembalian buku.

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-014* | Select (tgl\_peminjaman, namamember) FROM transaksi;  if (tgl\_pengembalian > bataspengembalian) {  denda = jumlahhariterlewat \* totaldenda;  } else (mengembalikan lebih awal dari tanggal jatuh tempo / batas pengembalian)  {  denda = 0;  }  } | *show data denda berdasarkan data peminjaman buku dari database* |
|
|

*Nama Kelas : Pustakawan*

*Nama Operasi :* Edit data peminjaman

*Algoritma :(Algo-015)*

1. Pustakawan akan mengedit data member jika
2. Pustakawa akan menghapus data pustakwan berdasarkan (id\_Struk) jika sudah melewati batas peminjaman atau kadaluarsa
3. Sistem akan menghapus dan menyimpan data pada database pustakawan.

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-015* | Select (id\_pustakawan) FROM Pustakawan;  Update tbPeminjaman (id\_Struk, nama, alamat, tglPeminjaman) VALUES ($id\_Struk, $nama, $tglPeminjaman); | *Mengedit data member yang kurang tepat , jika terjadi kesalahan penginputan data ke dalam database* |
|
|

*Nama Kelas :* Admin

*Nama Operasi :* Blokir data akun

*Algoritma :(Algo-016)*

1. Admin mendapati bahwa pustakawan cuti atau resign
2. Admin memblokir akun pustakawan tersebut dan menyimpan ke dalam database
3. Admin memperbaharui database pustakawan .

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-016* | Select (id\_pustakawan) FROM Pustakawan;  Delete tbPustakawan (id\_pustakawan, nama, TTL, password) VALUES ($id\_pustakawan, ‘bambang’, $TTL, $password); | *Memblokir data akun pustakawan sesuai dengan keterangan pustakawan yang dibuat* |
|
|

*Nama Kelas :* Admin

*Nama Operasi :* UnBlokir data akun

*Algoritma :(Algo-017)*

1. Admin mendapati bahwa pustakawan tidak cuti atau tidak resign
2. Admin memblokir akun pustakawan tersebut dan menyimpan ke dalam database
3. Admin memperbaharui database pustakawan .

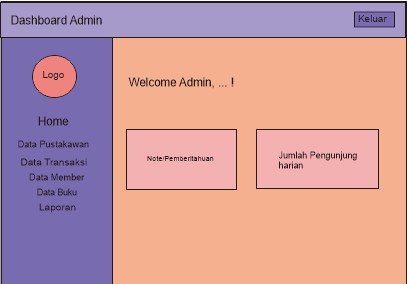
*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-017* | Select (id\_pustakawan) FROM Pustakawan;  Insert tbPustakawan (id\_pustakawan, nama, TTL, password) VALUES ($id\_pustakawan, ‘bambang’, $TTL, $password); | *Menublokir data akun pustakawan jika pustakawan tidak resign atau cuti* |
|
|

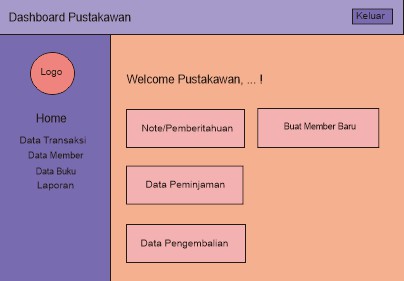
## Perancangan Antarmuka

Dashboard Admin :

**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| NAV 1 | Link | Home Link | Link untuk menuju halaman home |
| NAV 2 | Link | Data Pustakawan | Link untuk menuju ke kumpulan data pustakawan |
| NAV 3 | Link | Data Transaksi | Link untuk menuju ke halaman data transaksi |
| NAV 4 | Link | Data Member | Link untuk menuju ke halaman data member |
| NAV 5 | Link | Data Buku | Link untuk menuju ke halaman data buku |
| NAV 6 | Link | Laporan | Link untuk menuju ke halaman laporan |
| CARD 1 | Card | Note/Pemberitahuan | Menunjukan pemberitahuan yang masuk |
| CARD 2 | Card | Jumlah Pengunjung Harian | Menunjukan jumlah pengunjung harian |

Dashboard Pustakawan :



|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| NAV 1 | Link | Home Link | Link untuk menuju ke halaman home |
| NAV 2 | Link | Data Transaksi | Link untuk menuju ke halaman data transaksi |
| NAV 3 | Link | Data Member | Link untuk menuju ke halaman data member |
| NAV 4 | Link | Data Buku | Link untuk menuju ke halaman data buku |
| NAV 5 | Link | Laporan | Link untuk menuju ke halaman laporan |
| CARD 1 | Card | Note/Pemberitahuan | Menunjukan pemberitahuan yang masuk |
| CARD 2 | Card | Data Peminjaman | Menunjukan halaman data peminjaman |
| CARD 3 | Card | Data Pengembalian | Menunjukan halaman data pengembalian |
| CARD 4 | Card | Buat Member | Menunjukan halaman untuk membuat member baru |

## Perancangan Representasi Persistensi Kelas

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

# Matriks Kerunutan

*Mapping use case dengan kelas-kelas terkait*

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Usecase Terkait** | **Kelas** |
| FR-01 | Input data buku | Pustakawan |
| FR-02 | Input akun member | Pustakawan |
| FR-03 | Input data akun | Admin |
| FR-04 | Pinjam Buku, Kembali Buku | Pustakawan, Member |

# 