**DPPL-02**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

APLIKASI KOPERASI SIMPAN PINJAM (KSP)

Dipersiapkan oleh:

Muhammad Rifki Fauzan (1301174078)  
Daffa Maulana Hibban (1301174098)  
Irsyad Rafi Diesta (1301170201)  
Muhammad Danil Muis (1301174433)

Program Studi S1 Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-02* | | *36* |
| Revisi | *1* | *Tgl: 19 November 2019* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | * Relasi di class diagram |
| 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 6](#_Toc25099932)

[1.1 Tujuan Penulisan Dokumen 6](#_Toc25099933)

[1.2 Lingkup Masalah 6](#_Toc25099934)

[1.3 Definisi dan Istilah 6](#_Toc25099935)

[1.4 Referensi 6](#_Toc25099936)

[1.5 Sistematika Pembahasan 6](#_Toc25099937)

[2 Deskripsi Perancangan Global 7](#_Toc25099938)

[2.1 Rancangan Lingkungan Implementasi 7](#_Toc25099939)

[2.2 Deskripsi Arsitektural 7](#_Toc25099940)

[2.3 Deskripsi Komponen 8](#_Toc25099941)

[3 Perancangan Rinci 9](#_Toc25099942)

[3.1 Realisasi Use Case 9](#_Toc25099943)

[3.1.1 Use Case Registrasi Anggota 9](#_Toc25099944)

[3.1.1.1 Identifikasi Kelas 9](#_Toc25099945)

[3.1.1.2 Sequence Diagram 9](#_Toc25099946)

[3.1.1.3 Diagram Kelas 9](#_Toc25099947)

[3.1.2 Use Case Registrasi Pengurus 10](#_Toc25099948)

[3.1.2.1 Identifikasi Kelas 10](#_Toc25099949)

[3.1.2.2 Sequence Diagram 10](#_Toc25099950)

[3.1.2.3 Diagram Kelas 10](#_Toc25099951)

[3.1.3 Use Case Login Anggota 11](#_Toc25099952)

[3.1.3.1 Identifikasi Kelas 11](#_Toc25099953)

[3.1.3.2 Sequence Diagram 11](#_Toc25099954)

[3.1.3.3 Diagram Kelas 11](#_Toc25099955)

[3.1.4 Use Case Login Pengurus 12](#_Toc25099956)

[3.1.4.1 Identifikasi Kelas 12](#_Toc25099957)

[3.1.4.2 Sequence Diagram 12](#_Toc25099958)

[3.1.4.3 Diagram Kelas 12](#_Toc25099959)

[3.1.5 Use Case Transaksi 12](#_Toc25099960)

[3.1.5.1 Identifikasi Kelas 12](#_Toc25099961)

[3.1.5.2 Sequence Diagram 13](#_Toc25099962)

[3.1.5.3 Diagram Kelas 13](#_Toc25099963)

[3.1.6 Use Case View Transaksi 13](#_Toc25099964)

[3.1.6.1 Identifikasi Kelas 13](#_Toc25099965)

[3.1.6.2 Sequence Diagram 14](#_Toc25099966)

[3.1.6.3 Diagram Kelas 14](#_Toc25099967)

[3.1.7 Use Case View Anggota 14](#_Toc25099968)

[3.1.7.1 Identifikasi Kelas 14](#_Toc25099969)

[3.1.7.2 Sequence Diagram 15](#_Toc25099970)

[3.1.7.3 Diagram Kelas 15](#_Toc25099971)

[3.2 Perancangan Detil Kelas 15](#_Toc25099972)

[3.2.1 Kelas Registrasi 15](#_Toc25099973)

[3.2.2 Kelas Login 16](#_Toc25099974)

[3.2.3 Kelas Anggota 16](#_Toc25099975)

[3.2.4 Kelas Pengurus 16](#_Toc25099976)

[3.2.5 Kelas Transaksi 17](#_Toc25099977)

[3.2.6 Kelas Peminjaman 17](#_Toc25099978)

[3.2.7 Kelas Penyimpanan 17](#_Toc25099979)

[3.2.8 Kelas Transfer 17](#_Toc25099980)

[3.2.9 Kelas Penarikan 17](#_Toc25099981)

[3.2.10 Kelas Pelunasan 18](#_Toc25099982)

[3.3 Diagram Kelas Keseluruhan 18](#_Toc25099983)

[3.4 Algoritma/Query 19](#_Toc25099984)

[3.5 Perancangan Antarmuka 23](#_Toc25099985)

[3.6 Perancangan Representasi Persistensi Kelas 29](#_Toc25099986)

[4 Matriks Kerunutan 29](#_Toc25099987)

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini ditulis untuk memenuhi tugas besar mata kuliah Implementasi dan Pengujian Perangkat Lunak, yang akan digunakan oleh kelompok kami sebagai dasar penilaian oleh dosen pengampu mata kuliah IMPAL.

## Lingkup Masalah

Aplikasi Koperasi Simpan Pinjam. Aplikasi berfungsi untuk membantu berjalannya koperasi simpan pinjam. Dengan adanya aplikasi ini diharapkan masyarakat yang menggunakan aplikasi ini ikut membangun dan mengembangkan dalam industri 4.0.

## Definisi dan Istilah

* Use case diagram: gambaran dari beberapa atau semua actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem.
* Sequence diagram: suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut.
* Diagram class: model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class.
* Algoritma: urutan atau langkah-langkah untuk penghitungan atau untuk menyelesaikan suatu masalah yang ditulis secara berurutan.
* Database: kumpulan data yang disimpan secara sistematis di dalam computer.
* Android: Suatu sistem operasi yang digunakan untuk handphone berteknologi canggih.
* MySQL server: suatu server database yang digunakan untuk menyimpasn suatu data.

## Referensi

Referensi berasal dari SKPL yang telah dibuat sebelumnya.

## Sistematika Pembahasan

Bab 1: Berisikan pendahuluan mengenai tujuan DPPL ini dibuat

Bab 2: Berisikan prancangan global yang berkaitan dengan tugas besar

Bab 3: Berisikan perincian rancangan dengan use case, sequence diagram, diagram class, algoritma dan tampilan aplikasi

Bab 4: Berisikan matriks kerunutan

# Deskripsi Perancangan Global

## Rancangan Lingkungan Implementasi

Sistem informasi pengelolaan keuangan universitas ini akan dikembangkan pada lingkungan dengan spesifikasi sebagai berikut :

1. Sistem operasi : Windows
2. Bahasa pemrograman : Javascript,PHP
3. Application Server : Google Cloud Platforms
4. Database : Mysql Server



## Deskripsi Arsitektural



## Deskripsi Komponen

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1 | Client | End-user yang menggunakan aplikasi dengan berbasis mobile android |
| 2 | Server | Melayani kebutuhan Client yang terkait dengan aplikasi |
| 3 | View | Bagian dari server yang menangani tampilan dari aplikasi |
| 4 | Controller | Bagian dari server yang menghubungkan View dan Model |
| 5 | Model | Bagian dari server yang berhubungan langsung dengan database |

# Perancangan Rinci

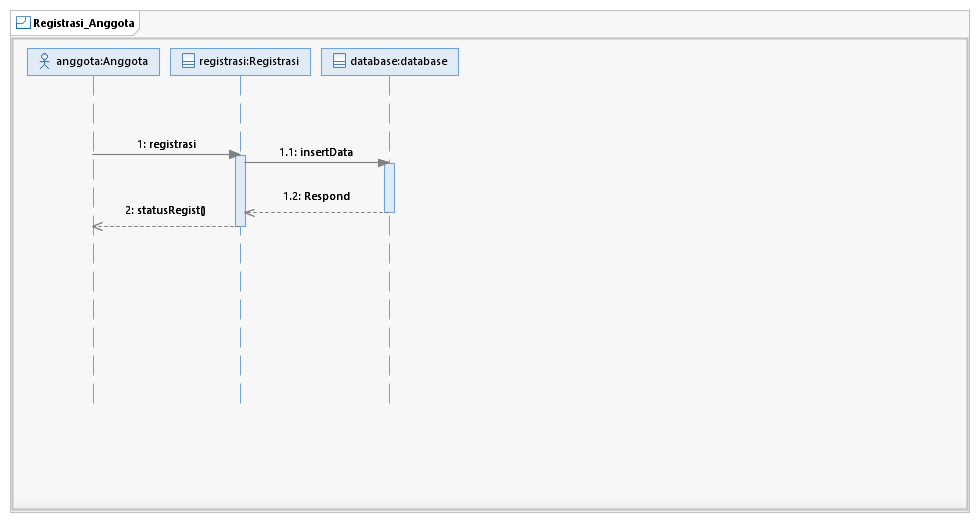
## Realisasi Use Case

### Use Case Registrasi Anggota

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| 2 | Anggota | Entity |
| 3 | pageRegistrasi | Boundary |

#### Sequence Diagram



#### Diagram Kelas

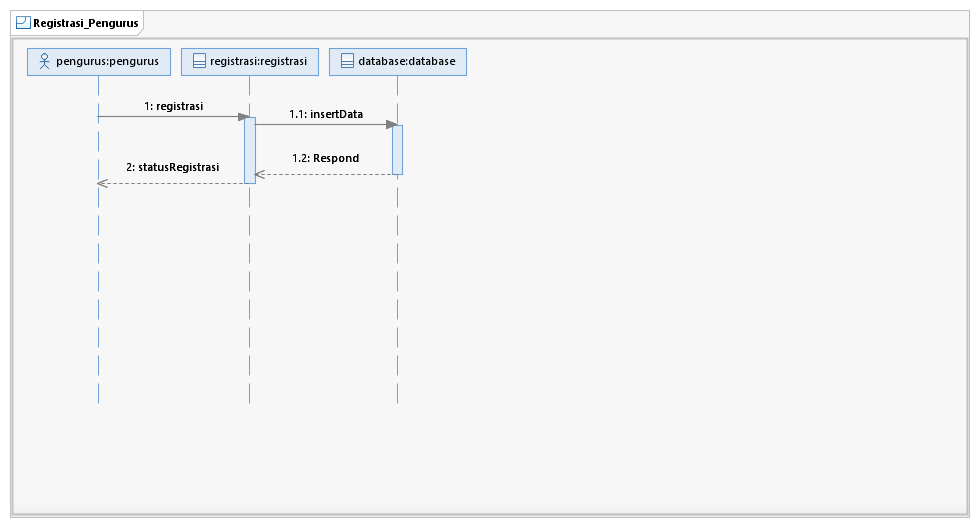


### Use Case Registrasi Pengurus

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| 2 | Pengurus | Entity |
| 3 | pageRegistrasi | Boundary |

#### Sequence Diagram



#### Diagram Kelas

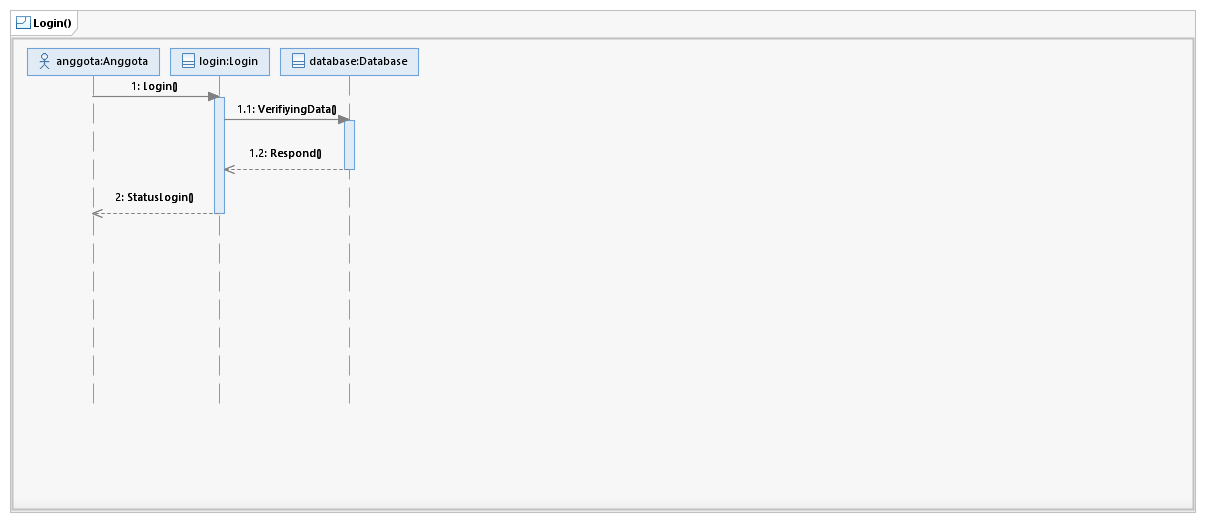


### Use Case Login Anggota

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| 2 | Anggota | Entity |
| 3 | pageLogin | Boundary |

#### Sequence Diagram



#### Diagram Kelas

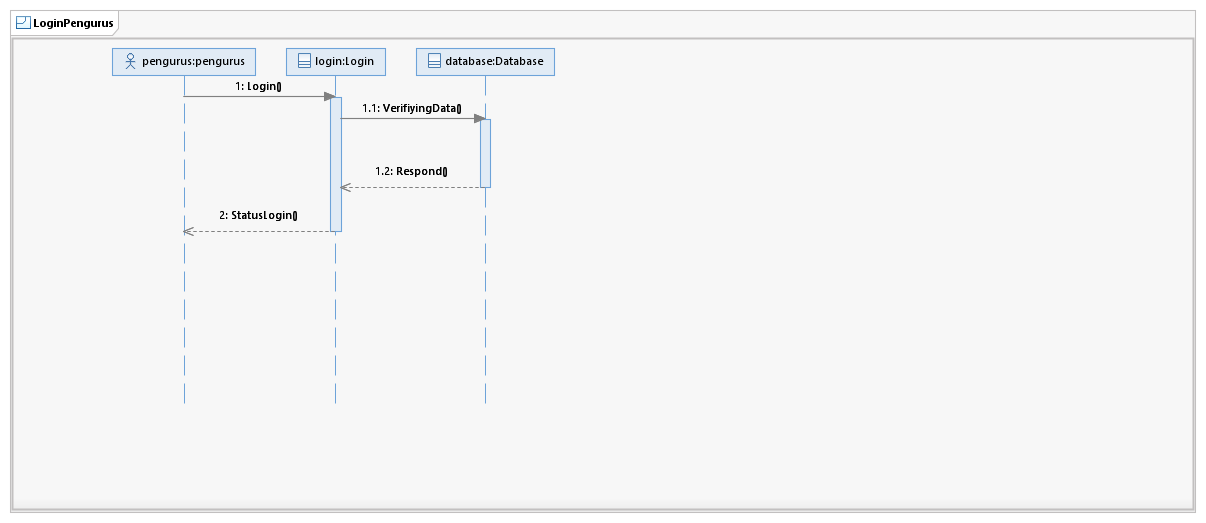


### Use Case Login Pengurus

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| 2 | Pengurus | Entity |
| 3 | pageLogin | Boundary |

#### Sequence Diagram



#### Diagram Kelas

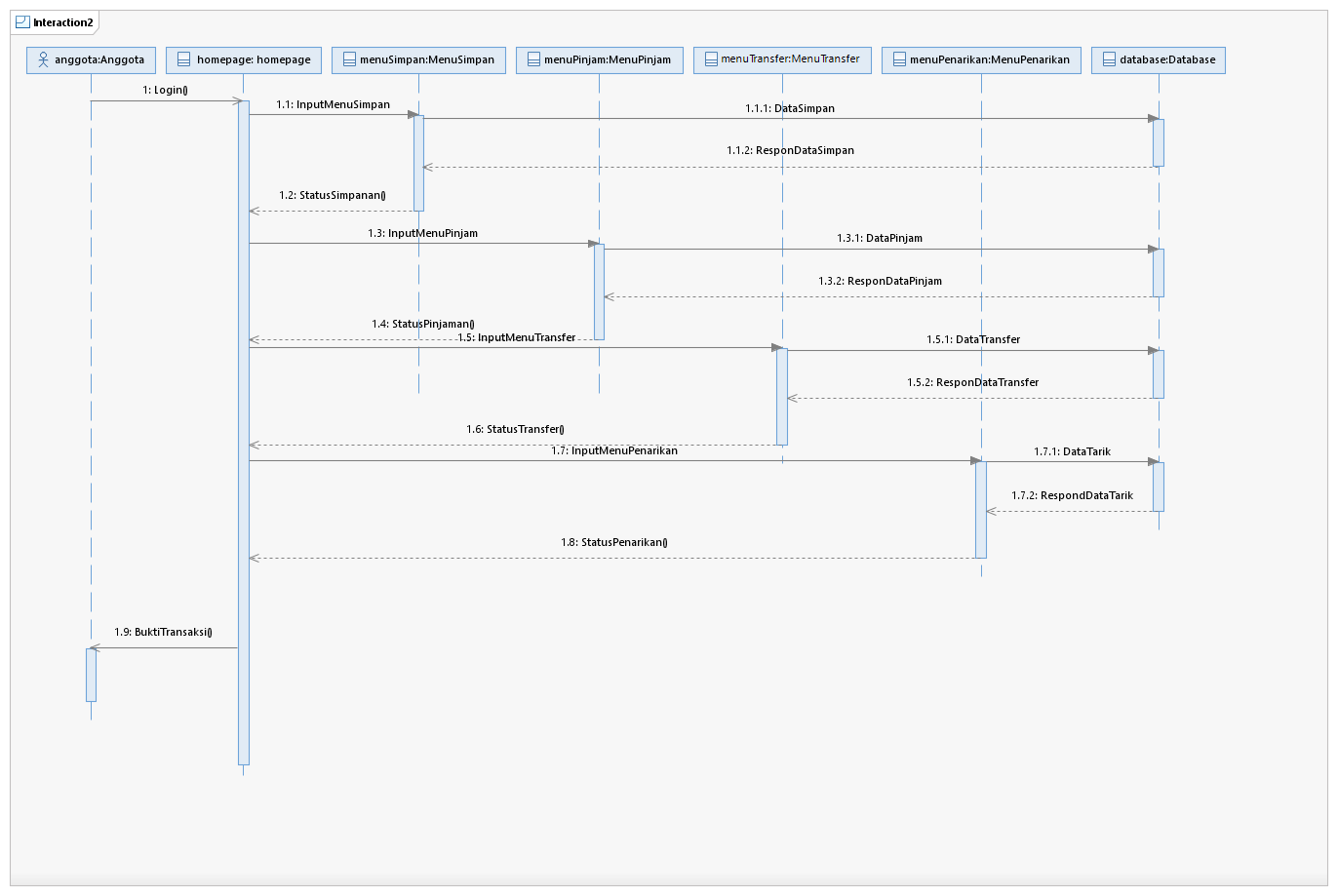


### Use Case Transaksi

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| *2* | *Transaksi* | *Entity* |
| *3* | *Memiliki* | *Entity* |
| *4* | *Akun* | *Entity* |
| *5* | *pageTransaksi* | *Boundary* |

#### Sequence Diagram



#### Diagram Kelas

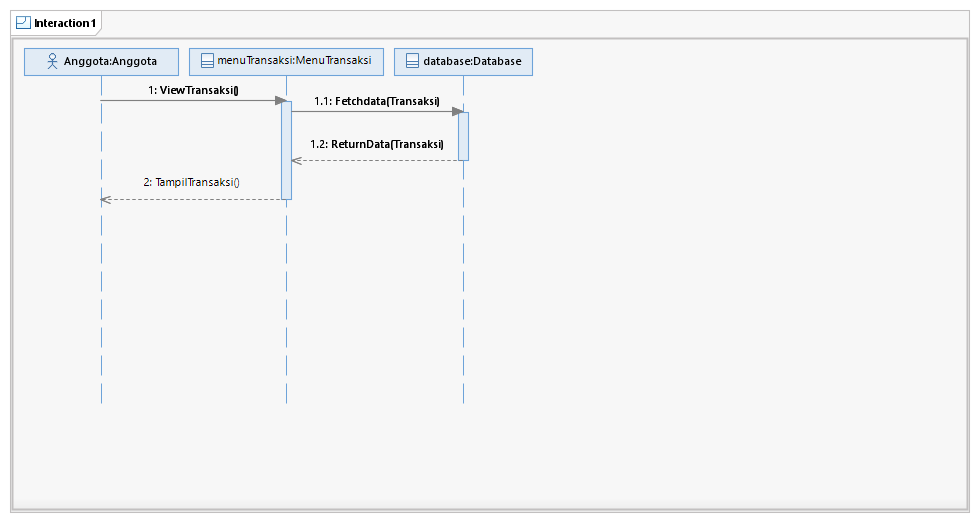


### Use Case View Transaksi

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| 2 | Akun | Entity |
| 3 | Transaksi | Entity |
| 4 | pageDaftarTransaksi | Boundary |

#### Sequence Diagram



#### Diagram Kelas

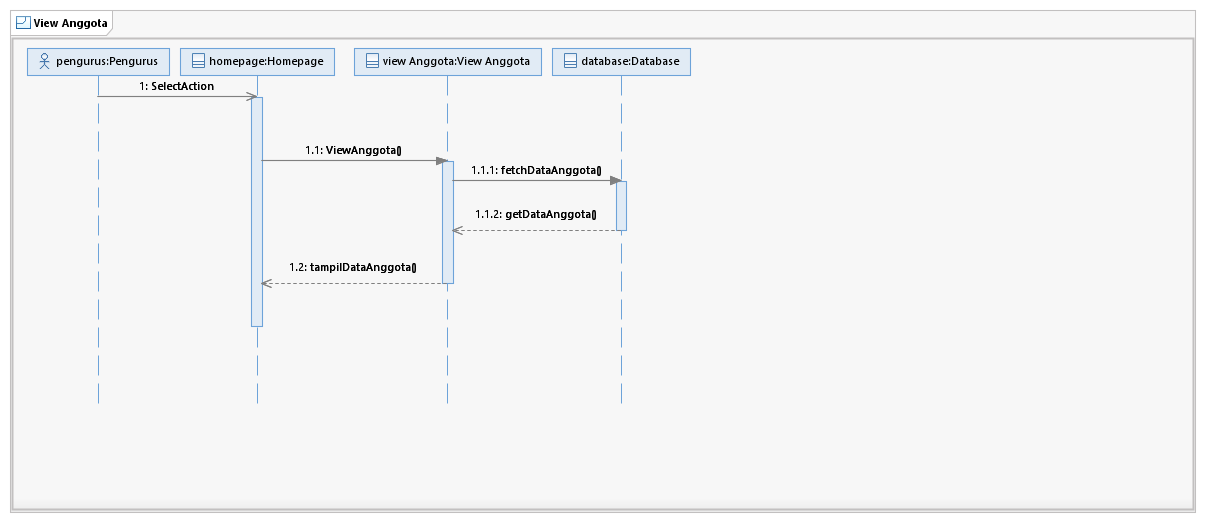


### Use Case View Anggota

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Tipe Kelas |
| 1 | Controller | Controller |
| 2 | Akun | Entity |
| 3 | Anggota | Entity |
| 4 | homepagePengurus | Boundary |

#### Sequence Diagram



#### Diagram Kelas



## Perancangan Detil Kelas

|  |  |  |
| --- | --- | --- |
| No | Nama Kelas Perancangan | Nama Kelas Analisis Terkait |
| 1 | Registrasi |  |
| 2 | Login | Anggota, Pengurus, Peringatan |
| 3 | Anggota | Login, Transaksi, Peminjaman, Pelunasan, Penyimpanan, Transfer, Penarikan |
| 4 | Pengurus | Login, Peringatan, Transaksi, Anggota |
| 5 | Transaksi | Peminjaman, Pelunasan, Penyimpanan, Transfer, Penarikan |
| 6 | Penyimpanan | Transaksi |
| 7 | Transfer | Transaksi |
| 8 | Penarikan | Transaksi |
| 9 | Peminjaman | Transaksi, Pelunasan |
| 10 | Pelunasan | Transaksi, Peminjaman |

### Kelas Registrasi

*Nama Kelas : Registrasi*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| InputRegistrasi | Public | Memasukkan data kepada aplikasi, lalu aplikasi memasukkan ke *database* apabila tidak terjadi duplikasi data |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| NIK | Private | String |
| NamaLengkap | Private | String |
| TTL | Private | String |
| Alamat | Private | String |
| Agama | Private | String |
| Username | Private | String |
| Password | Private | String |

### Kelas Login

*Nama Kelas : Login*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| InputLogin | Public | Memasukkan *username* dan *password* ke aplikasi untuk bisa menggunakan aplikasi tersebut |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| Username | Private | String |
| Password | Private | String |

### Kelas Anggota

*Nama Kelas : Anggota*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Bertransaksi | Public | Melakukan transaksi dengan memasukkan data yang diminta aplikasi dengan sesuai keinginan pengguna |
| ViewPeringatan | Public | Pengguna melihat peringatan yang dikirim oleh pengurus |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| IdAnggota | Private | String |
| Nama | Private | String |
| Kredit | Private | Long |
| Debit | Private | Long |

### Kelas Pengurus

*Nama Kelas : Pengurus*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| ViewAnggota | Public | Untuk melihat daftar seluruh anggota yang terdaftar |
| InputPeringatan | Public | Untuk memasukkan peringatan ke aplikasi yang akan ditujukan kepada salah satu anggota |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| IdPengurus | Private | String |
| Nama | Private | String |

### Kelas Transaksi

*Nama Kelas : Transaksi*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| ViewTransaksi | Public | Melihat hasil transaksi yang telah dilakukan |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| NoTransaksi | Private | String |
| JumlahUang | Private | Long |

### Kelas Peminjaman

*Nama Kelas : Peminjaman*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Meminjam | Public | Melakukan transaksi peminjaman |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

### Kelas Penyimpanan

*Nama Kelas : Penyimpanan*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Menyimpan | Public | Melakukan transaksi penyimpanan |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

### Kelas Transfer

*Nama Kelas : Transfer*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Mengirim | Public | Melakukan transaksi transfer |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

### Kelas Penarikan

*Nama Kelas : Penarikan*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Menarik | Public | Melakukan transaksi penarikan |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

### Kelas Pelunasan

*Nama Kelas : Pelunasan*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| Melunasi | Public | Melakukan transaksi pelunasan |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |

## Diagram Kelas Keseluruhan



## Algoritma/Query

*Nama Kelas :Anggota, Pengurus*

*Nama Operasi :InputRegistrasi*

*Algoritma : (Algo-001)*

*Input(NIK,NamaLengkap,TTL,Alamat,Agama,Username,Password)*

*If( tidak kosong (NIK,NamaLengkap,TTL,Alamat,Agama,Username,Password) ) then*

*If(Username tidak duplikat) then*

*Masukkan ke database*

*Else then*

*Print : Username sudah digunakan*

*Kembali memasukkan username*

*Else*

*Kembali memasukkan data yang masih kosong*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* from anggota/pengurus where username = ‘Username’* | *Mengecek jika ada username yang duplikat* |
| *Q-002* | *INSERT INTO anggota/pengurus VALUES (NIK,NamaLengkap,TTL,Alamat,Agama,Username,Password)* | *Memasukkan data ke dalam database* |

*Nama Kelas :Anggota, Pengurus*

*Nama Operasi :InputLogin*

*Algoritma : (Algo-002)*

*Input(Username,Password)*

*If (tidak kosong (Username,Password) ) then*

*If((Username,Password) ada di database) then*

*Masuk ke homepage*

*Else then*

*Print : Username atau password tidak sesuai*

*Kembali memasukkan username dan password*

*Else*

*Kembali memasukkan data yang masih kosong*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* from anggota/pengurus where username = ‘Username’ and password = ‘Password’* | *Mengecek username dan password di dalam database* |

*Nama Kelas :Anggota*

*Nama Operasi :Bertransaksi*

*Algoritma : (Algo-003)*

*If(tombol simpanan di klik) then*

*Input(JumlahUang)*

*If(valid(JumlahUang)) then*

*Masukkan ke database*

*Else*

*Print(jumlahUang salah)*

*Kembali memasukkan JumlahUang*

*Else If(tombol penarikan di klik) then*

*Input(JumlahUang)*

*If(valid(JumlahUang) and JumlahUang < debit) then*

*Masukkan ke database*

*Else*

*Print(jumlahUang salah)*

*Kembali memasukkan JumlahUang*

*Else if(tombol transfer di klik) then*

*Input(IdAnggota\_Tujuan)*

*If(valid(idAnggota\_Tujuan)) then #id Anggota ada di database  
 Input(JumlahUang)  
 if(valid(JumlahUang) and JumlahUang < debit) then*

*Masukkan ke database*

*Else*

*Print(jumlahUang salah)*

*Kembali memasukkan JumlahUang*

*Else*

*Print(idAnggota\_tujuan tidak ditemukan)*

*Kembali memasukkan idAnggota\_tujuan*

*Else if(tombol pinjaman di klik) then*

*If(kredit > 0 ) then*

*Tampil menu pelunasan*

*Input(jumlahUang) #untuk dilunaskan*

*If(valid(jumlahUang)) then*

*Masukkan ke database*

*Else*

*Print(jumlahUang salah)*

*Kembali memasukkan jumlahUang*

*Else*

*Tampil menu peminjaman*

*Input(jumlahUang) #untuk meminjam*

*If(valid(jumlahUang)) then*

*Masukkan ke database*

*Else*

*Print(jumlahUang salah)*

*Kembali memasukkan jumlahUang*

*Print Bukti Transaksi terkait*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *INSERT INTO transaksi values (‘jumlahUang’,’simpanan’);* | *Memasukkan data transaksi penyimpanan ke dalam database* |
| *Q-002* | *INSERT INTO melakukan values (‘NoTransaksi’);* | *Memasukkan data transaksi sesuai jenis transaksi ke dalam database* |
| *Q-003* | *UPDATE akun set debit = (debit+jumlahUang) WHERE idAnggota = id;* | *Update akun dengan menambahkan jumlah debit* |
| *Q-004* | *INSERT INTO transaksi values (‘jumlahUang’,’penarikan’);* | *Memasukkan data transaksi penarikan ke dalam database* |
| *Q-005* | *UPDATE akun set debit = (debit – JumlahUang) WHERE idAnggota = id;* | *Update akun dengan mengurangi jumlah debit* |
| *Q-006* | *SELECT \* FROM akun where id\_Anggota = idAnggota\_tujuan;* | *Mencari idAnggota yang dituju di dalam database(table akun)* |
| *Q-007* | *INSERT INTO transaksi values(‘jumlahUang’,transfer’,idAnggota\_tujuan);* | *Memasukkan data transaksi transfer ke dalam database* |
| *Q-008* | *UPDATE akun set debit = (debit+jumlahUang) where id\_anggota = idAnggota\_tujuan);* | *Update akun dengan menambahi jumlah debit (orang yang menerima transfer)* |
| *Q-009* | *UPDATE akun set debit = (debit-jumlahUang)*  *Where id\_anggota = id;* | *Update akun dengan mengurangi jumlah debit(orang yang melakukan transfer)* |
| *Q-010* | *INSERT INTO transaksi values (jumlahUang,peminjaman)* | *Memasukkan data transaksi peminjaman ke dalam database* |
| *Q-011* | *UPDATE akun set kredit = (jumlahUang) where id\_anggota = id;* | *Update akun dengan set nilai kredit sesuai jumlahUang* |
| *Q-012* | *INSERT INTO transaksi values(jumlahUang,pelunasan)* | *Memasukkan data transaksi pelunasan ke dalam database* |
| *Q-013* | *UPDATE akun set kredit = (kredit-jumlahUang) where id\_anggota = id;* | *Update akun dengan mengurangi jumlah kredit* |

*Nama Kelas :Anggota*

*Nama Operasi :ViewPeringatan*

*Algoritma : (Algo-004)*

*If(peringatan ada) then*

*Tampilkan dalam bentuk notifikasi*

*Else*

*Notifikasi kosong*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* from peringatan where id\_anggota = id* | *Mengecek apakah ada peringatan utk user yang sedang login* |

*Nama Kelas :Pengurus*

*Nama Operasi :InputPeringatan*

*Algoritma : (Algo-005)*

*Menekan tombol input peringatan untuk user yang terkait (hasil dari viewAnggota)  
input(isiPeringatan)*

*If(berhasil inputPeringatan) then*

*Print(‘Peringatan berhasil di inputkan untuk anggota (id) ‘)*

*Else*

*Print(‘Gagal menginputkan’)*

*Kembali menginput peringatan*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *INSERT INTO peringatan values(idAnggota,isi)* | *Memasukkan isi dari peringatan ke dalam database* |

*Nama Kelas :Pengurus*

*Nama Operasi :ViewAnggota*

*Algoritma : (Algo-006)*

*If(data anggota ada ) then*

*Tampilkan daftarAnggota yang ada beserta akunnya*

*Else*

*Print(‘data tidak ada’)*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* from anggota JOIN akun using (idAnggota)* | *Menampilkan seluruh anggota beserta data data yang ada* |

*Nama Kelas :Transaksi*

*Nama Operasi :ViewTransaksi*

*Algoritma : (Algo-007)*

*Anggota klik tombol lihat daftar Transaksi*

*If(transaksi ada) then*

*Tampilkan daftarTransaksi*

*Else*

*Print(‘data tidak ada’)*

*#untuk pengurus*

*Pengurus klik salah satu anggota dari daftarAnggota*

*If(transaksi ada) then*

*Tampilkan daftarTransaksi*

*Else*

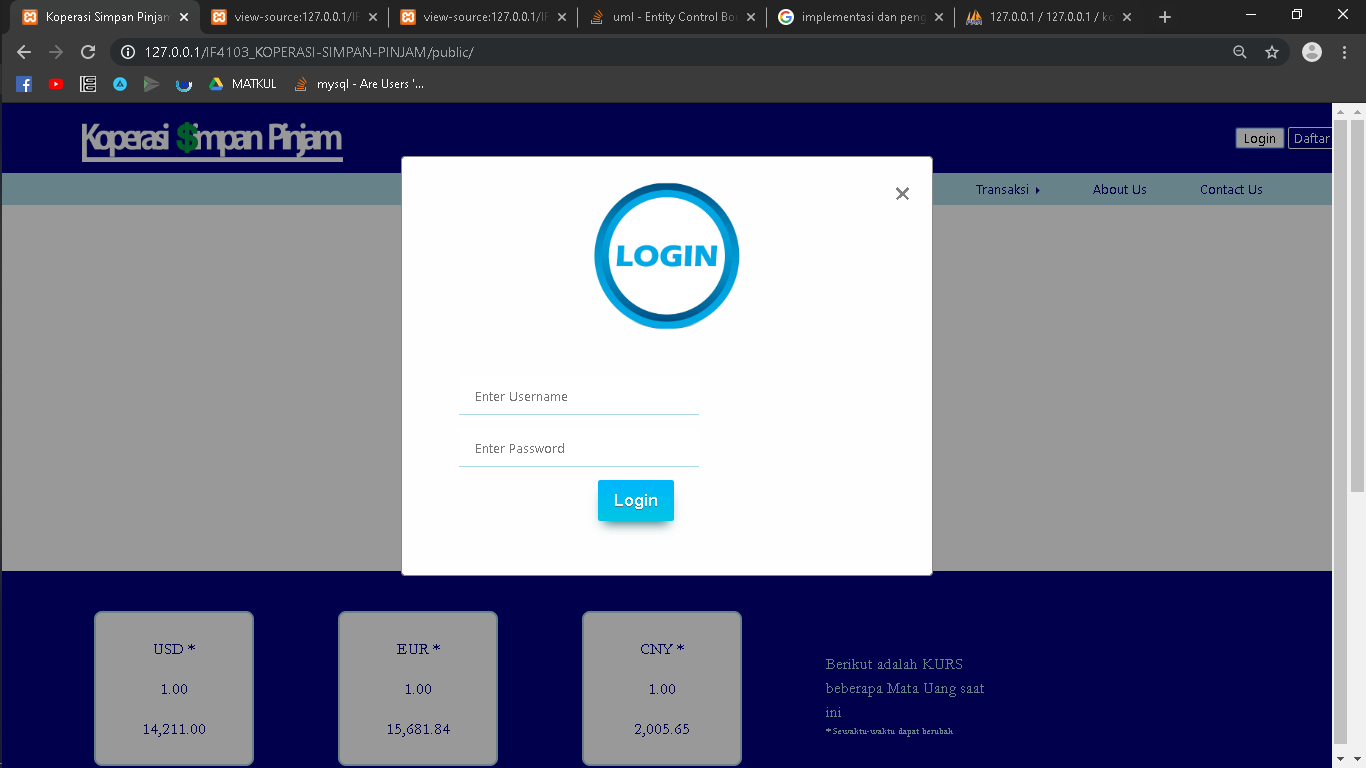
*Print(‘data tidak ada’)*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* from akun*  *JOIN melakukan using (idAnggota)*  *JOIN transaksi using(noTransaksi)*  *Where idAnggota = id* | *Menampilkan seluruh transaksi dari anggota yang terkait* |

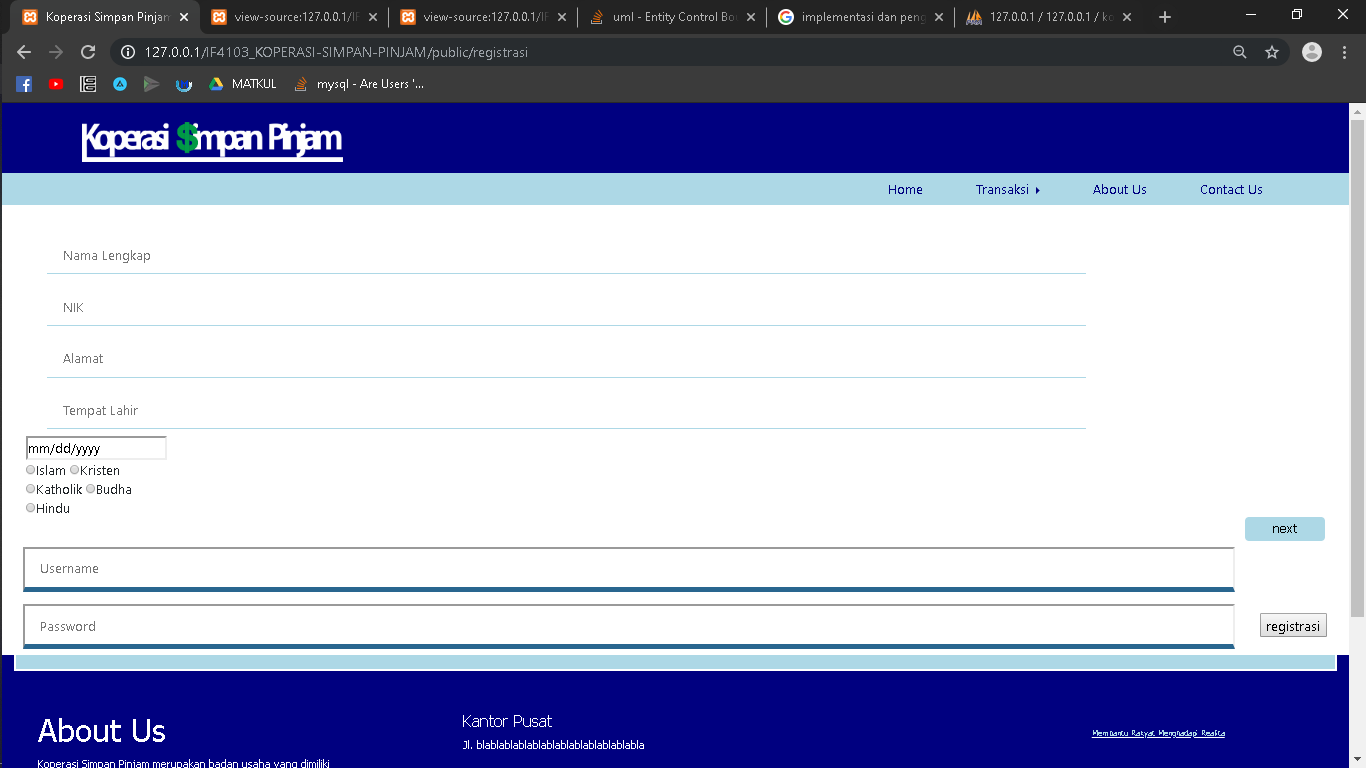
## Perancangan Antarmuka

*Antarmuka : Login*



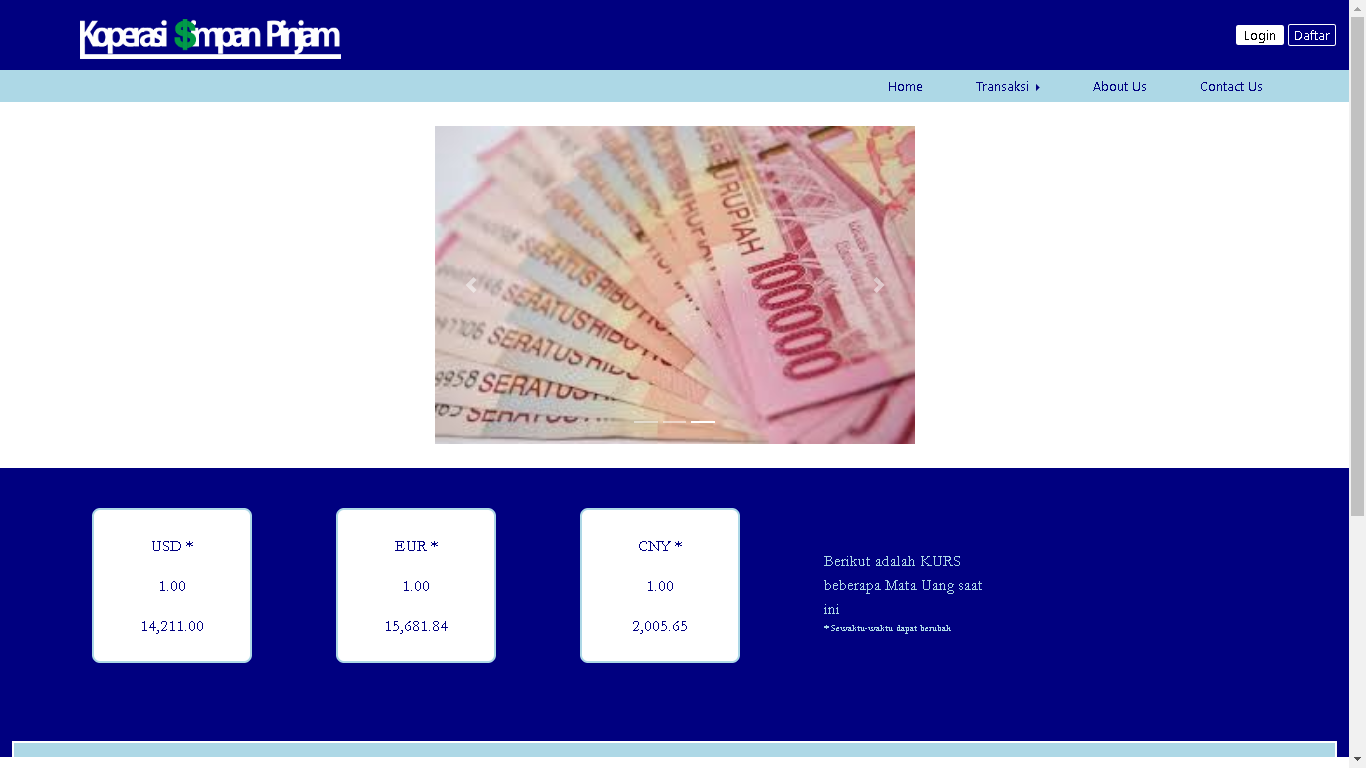
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Username*  *Password* | *Jika Username dan Password benar, maka akan masuk ke halaman home* |
|  |  | *Belum daftar?* | *Ketika menekan string tersebut, maka akan ditampilkan halaman registrasi* |
| *Button1* | *Button* | *Login* | *Jika ditekan, akan mengaktifkan proses Algo-002* |

*Antarmuka : Registrasi*



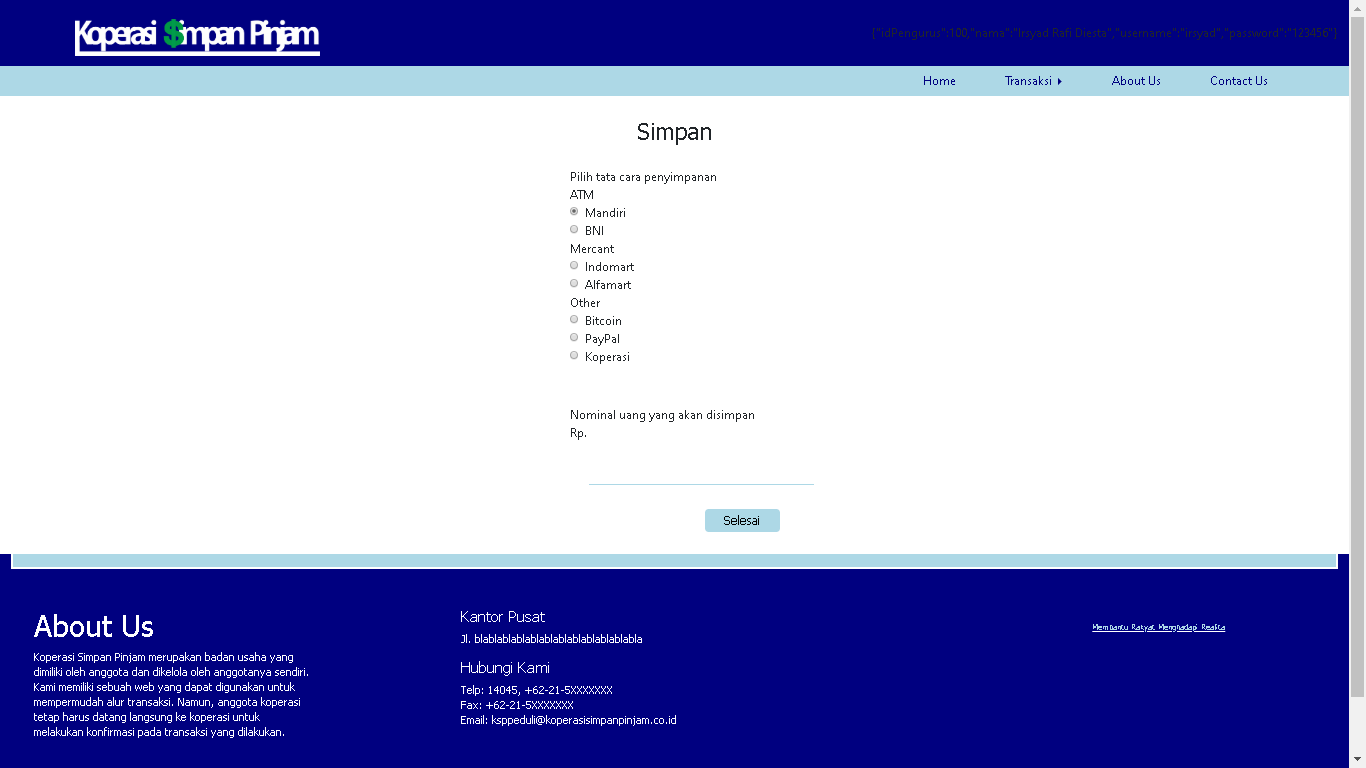
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *NIK*  *Nama Lengkap*  *TTL*  *Alamat*  *Agama*  *Password*  *Username* | *Merupakan data yang akan dimasukkan ke dalam database sebagai data anggota* |
| *Button1* | *Button* | *Selesai* | *Jika ditekan, maka akan mengaktifkan proses Algo-001* |

*Antarmuka : Home*



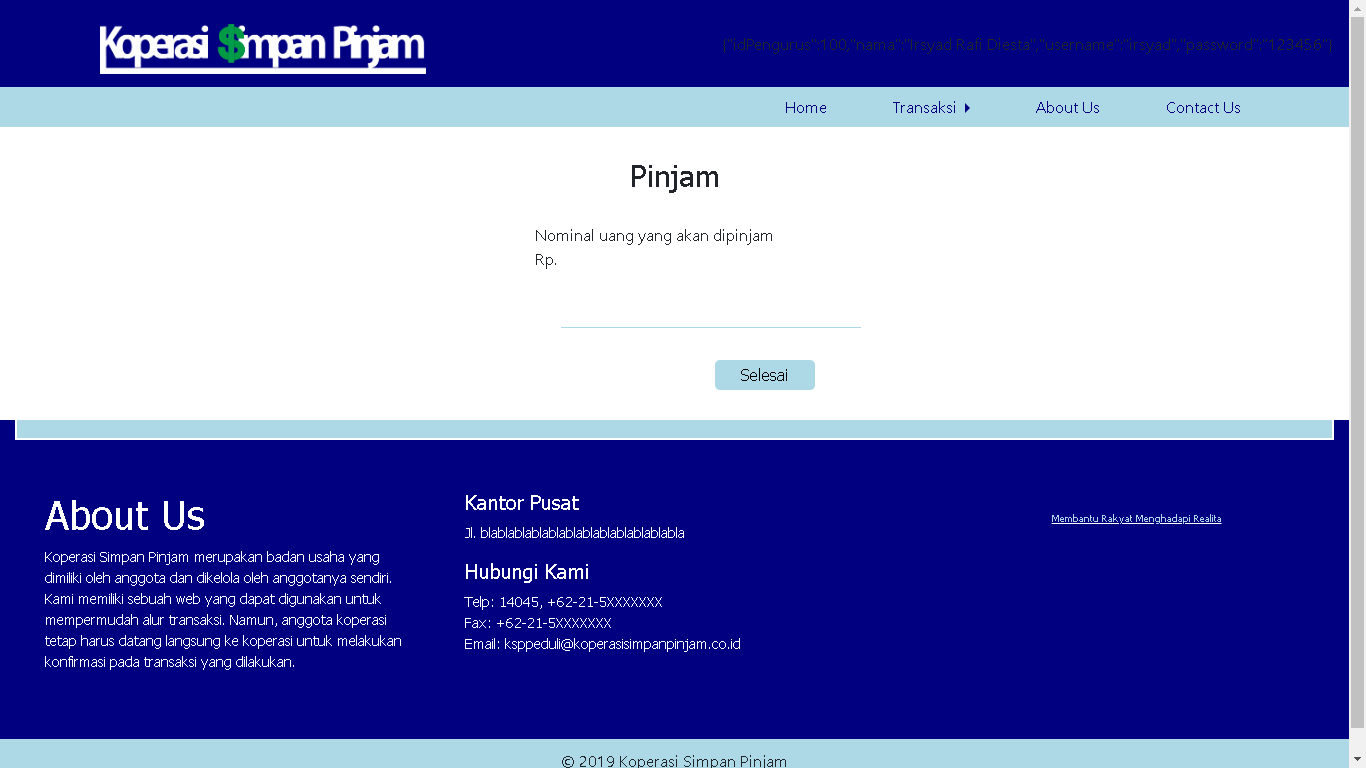
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| Button1 | Button | Simpan | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |
| Button2 | Button | Pinjam | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |
| Button3 | Button | Tarik | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |
| Button4 | Button | Transfer | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |
| Button5 | Button | (Simbol Lonceng) | *Jika ditekan, maka akan mengaktifkan proses Algo-004* |

*Antarmuka : Simpan*



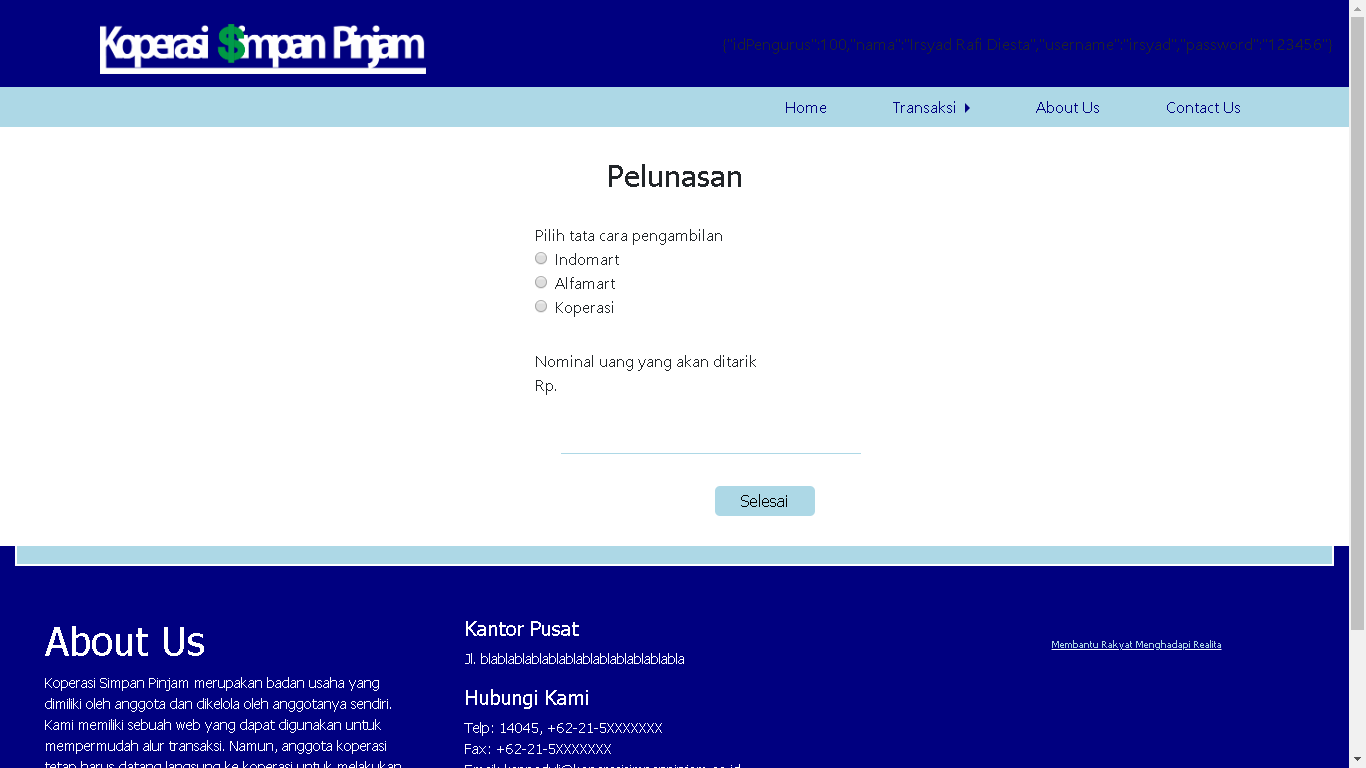
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Simpan dengan* | *Memilih metode simpan yang akan dilakukan* |
|  |  | *Masukkan nominal uang* | *Memasukkan nominal uang yang akan disimpan* |
| *Button1* | *Button* | *Lanjut* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |
| *Button2* | *Button* | *Selesai* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |

*Antarmuka : Pinjam*



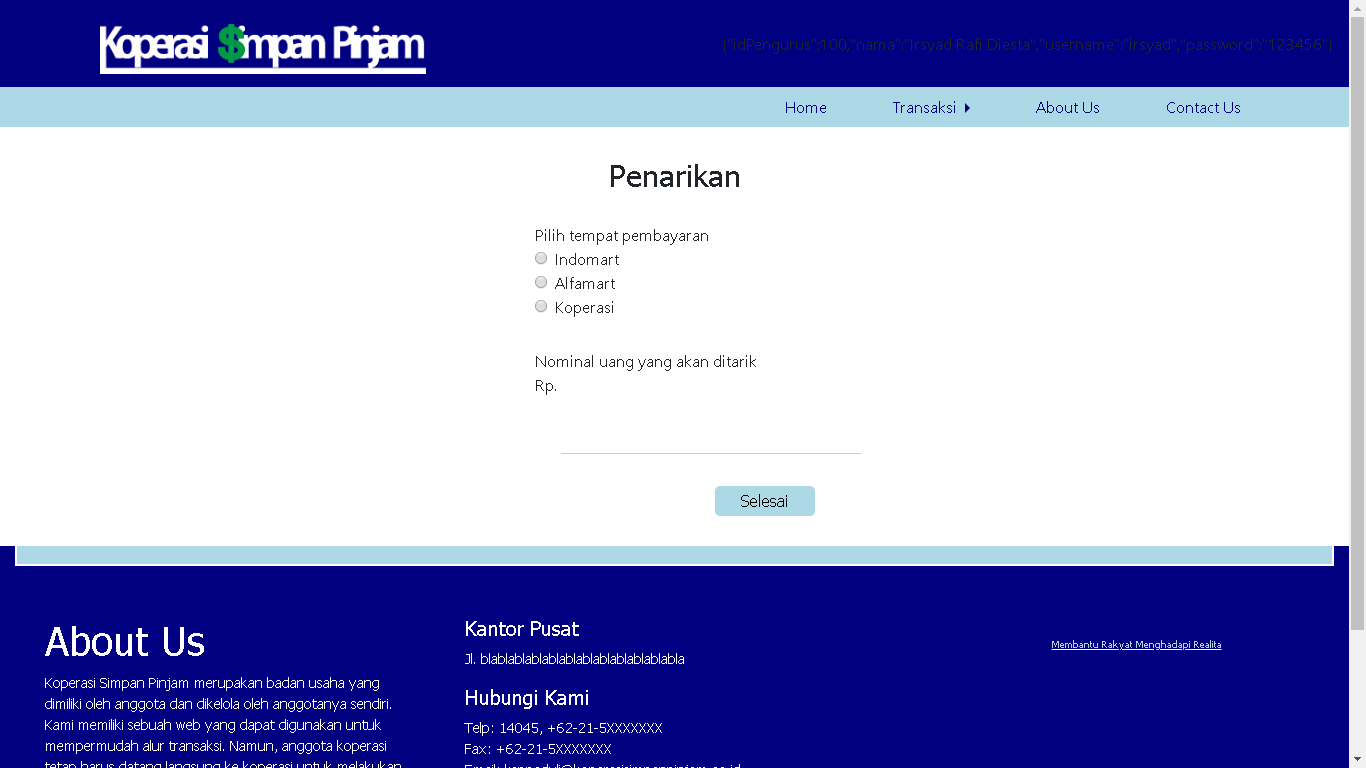
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Masukkan nominal uang* | *Memasukkan nominal uang yang akan dipinjam* |
| *Button1* | *Button* | *Selesai* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |

*Antarmuka : Pelunasan*



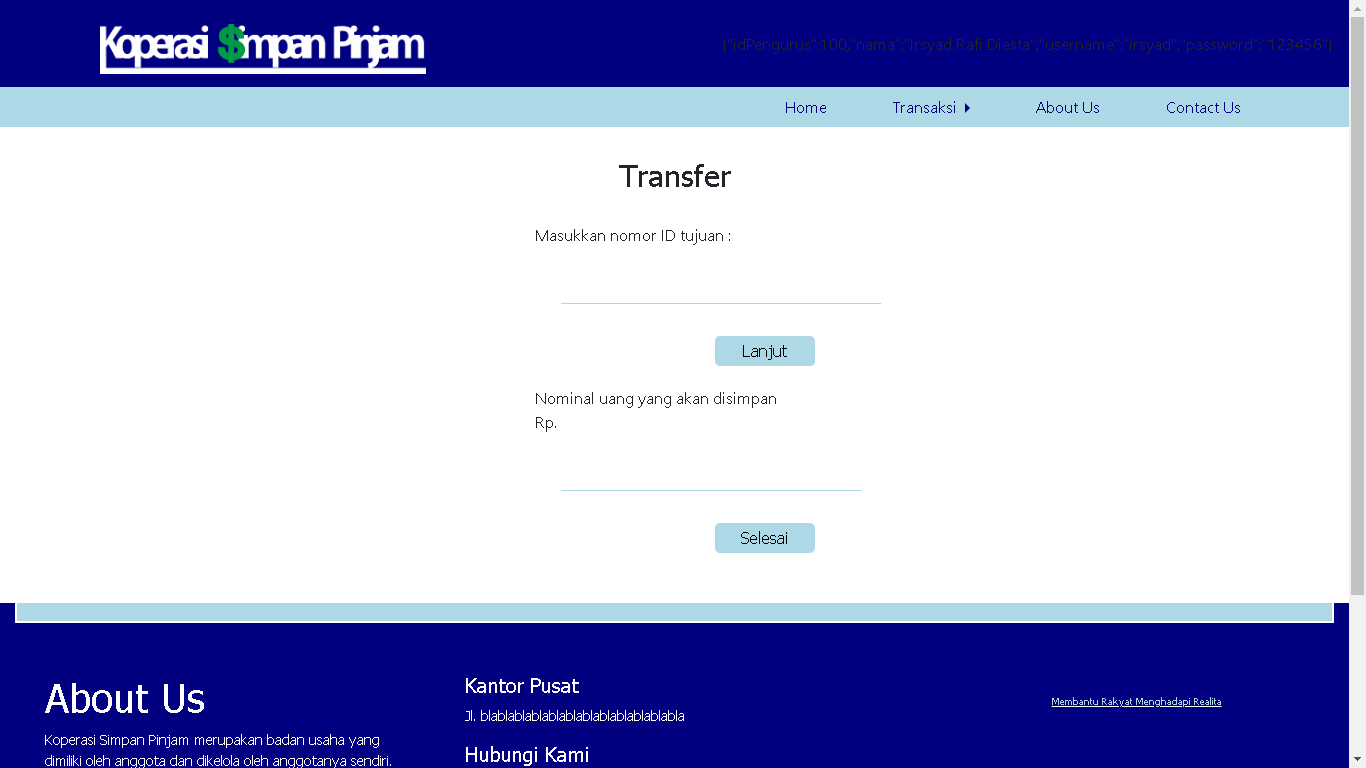
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Masukkan nominal uang* | *Memasukkan nominal uang yang akan dibayar untuk pelunasan* |
| *Button1* | *Button* | *Selesai* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |

*Antarmuka : Tarik*



|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Masukkan nominal uang* | *Memasukkan nominal uang yang akan diambil* |
| *Button1* | *Button* | *Selesai* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |

*Antarmuka : Transfer*



|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Masukkan nomor ID tujuan* | *Memasukkan nomor ID yang akan ditransfer, jika benar maka akan masuk ke halaman memasukkan nominal uangnya* |
|  |  | *Masukkan nominal uang* | *Memasukkan nominal uang yang akan ditransfer* |
| *Button1* | *Button* | *Lanjut* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |
| *Button2* | *Button* | *Selesai* | *Jika ditekan, maka akan mengaktifkan proses Algo-003* |

*Antarmuka : View Anggota*



|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *ID, Nama Lengkap, Username, Kredit, Debit* | *Berisikan data anggota* |

*Antarmuka : Bukti Transaksi*

**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
|  |  | *Nominal Uang*  *Jenis Transaksi*  *Waktu*  *No. Transaksi* | *Berisikan data-data ketika melakukan transaksi, yang isinya berupa nominal uang, jenis transaksi, waktu transaksi, dan nomor transaksi* |
| *Button1* | *Button* | *Selesai* | *-* |

## Perancangan Representasi Persistensi Kelas

Anggota(IdAnggota, Nama, Kredit, Debit);

Pengurus(IdPengurus, Nama);

Transaksi(NoTransaksi, JumlahUang, Jenis Transaksi);

# Matriks Kerunutan

*Mapping use case dengan kelas-kelas terkait*

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Usecase Terkait** | **Kelas** |
| FR-01 |  | Registrasi |
| FR-02 |  | Login |
| FR-03 | Registrasi, Login, View Transaksi, Transaksi | Anggota |
| FR-04 | Registrasi, Login, View Anggota, View Transaksi | Pengurus |
| FR-05 |  | Transaksi |
| FR-06 |  | Penyimpanan |
| FR-07 |  | Transfer |
| FR-08 |  | Penarikan |
| FR-9 |  | Peminjaman |
| FR-10 |  | Pelunasan |

# 