**DPPL-xx**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

**DANGDAN**

**(Sistem Aplikasi Service Motor Online)**

untuk:

<Nama User>

Dipersiapkan oleh:

<Nomor Grup & Anggota>

1. Angga Agustira (1301174308)
2. Dicky Hidayat (1401174259)
3. Igo Novaldi Sandri (1301174140)
4. Bayu Anugrah (1301174045)

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-xxx* | | *35* |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | Perubahan class diagram menjadi bentuk mvc |
| 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](#_Toc7725845)

[1.1 Tujuan Penulisan Dokumen 5](#_Toc7725846)

[1.2 Lingkup Masalah 5](#_Toc7725847)

[1.3 Definisi dan Istilah 5](#_Toc7725848)

[1.3.1 DPPL 5](#_Toc7725849)

[1.3.2 SKPL 5](#_Toc7725850)

[1.3.3    ALGORITMA 6](#_Toc7725851)

[1.3.4    QUERY 6](#_Toc7725852)

[1.3.5    USE CASE DIAGRAM 6](#_Toc7725853)

[1.3.6    KELAS DIAGRAM 6](#_Toc7725854)

[1.3.7    SEQUENCE DIAGRAM 6](#_Toc7725855)

[1.4 Referensi 6](#_Toc7725856)

[1.5 Sistematika Pembahasan 7](#_Toc7725857)

[2 Deskripsi Perancangan Global 8](#_Toc7725858)

[2.1 Rancangan Lingkungan Implementasi 8](#_Toc7725859)

[2.2 Deskripsi Arsitektural 8](#_Toc7725860)

[2.3 Deskripsi Komponen 10](#_Toc7725861)

[3 Perancangan Rinci 11](#_Toc7725862)

[3.1 Realisasi Use Case 11](#_Toc7725863)

[3.1.1 Use Case <Sistem Aplikasi Service Motor Online> 11](#_Toc7725864)

[3.2 Perancangan Detil Kelas 22](#_Toc7725865)

[3.2.1 Kelas <Customer> 22](#_Toc7725866)

[3.2.2 Kelas <Montir> 23](#_Toc7725867)

[3.2.3 Kelas <Pemilik> 23](#_Toc7725868)

[3.2.4 Kelas <ControllerCustomer> 23](#_Toc7725869)

[3.2.5 Kelas <ControllerMontir> 24](#_Toc7725870)

[3.2.6 Kelas <ControllerPemilik> 24](#_Toc7725871)

[3.2.7 Kelas <ViewCustomer> 25](#_Toc7725872)

[3.2.8 Kelas <ViewMontir> 25](#_Toc7725873)

[3.2.9 Kelas <ViewPemilik> 26](#_Toc7725874)

[3.3 Diagram Kelas Keseluruhan 26](#_Toc7725875)

[3.4 Algoritma/Query 27](#_Toc7725876)

[3.5 Perancangan Antarmuka 30](#_Toc7725877)

[3.6 Perancangan Representasi Persistensi Kelas 35](#_Toc7725878)

[4 Matriks Kerunutan 35](#_Toc7725879)

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

*Tuliskan dengan ringkas tujuan dokumen DPPL ini dibuat, dan digunakan oleh siapa.*

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) merupakan dokumen spesifikasi kebutuhan perangkat lunak yang akan dikembangkan.Tujuan dokumen ini digunakan untuk mengembangkan perangkat lunak yang sedang dikembangkan ke tahap selanjutnya dan target orang yang menggunakan perangkat lunak ini adalah siapapun yang membutuhkan yang mana sudah paham tentang penggunakan perangkat lunak.

Tujuan pembuatan dokumen Deskripsi Perancangan Perangkat Lunak(DPPL) ini adalah untuk mengarahkan kemana perkembangan perangkat lunak tersebut dan agar lebih menarik kedepannya.

## Lingkup Masalah

*Tuliskan dengan ringkas nama aplikasi dan deskripsinya. Maksimal 1 paragraf. Sama dengan yang ditulis di SKPL.*

Dandang merupakan aplikasi service motor yang mana untuk membantu masyarakat untuk mempermudah perbaikan motor tanpa harus keluar rumah.

## Definisi dan Istilah

*Semua definisi dan singkatan yang digunakan dalam dokumen ini dan penjelasannya*

### 1.3.1 DPPL

Deskripsi Perangcangan Perancangan Perangkat Lunak (DPPL) merupakan mendeskripsikan dan menjabarkan secara tereperinci mengenai perancangan perangkat lunak yang dibuat.

### 1.3.2 SKPL

SKPL adalah Spesifikasi Kebutuhan Perangkat Lunak, atau dalam bahasa Inggris-nya

sering juga disebut sebagai Software Requirements Spesification (SRS), dan merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.

### 1.3.3    ALGORITMA

Merupakan langkah-langkah yang ditulis secara berurutan untuk menyelesaikan masalah pemrograman komputer.

### 1.3.4    QUERY

Bahasa untuk melakukan manipulasi terhadap database, yang telah distandarkan dan lebih dikenal dengan nama Structured Query Language (SQL).

### 1.3.5    USE CASE DIAGRAM

Suatu Diagram yang mendeskripsikan fungsionalitas suatu system berdasarkan sudut pandang pengguna.

### 1.3.6    KELAS DIAGRAM

Model statis yang menggambarkan struktur dan deskripsi kelas serta hubungannya antara kelas.

### 1.3.7    SEQUENCE DIAGRAM

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut.

## Referensi

*Dokumentasi PL yang dirujuk oleh dokumen ini, minimal SKPL*

*Buku, Panduan, Dokumentasi lain yang dipakai dalam dokumen ini (jarang sekali!).*

*-Masukkan SKPL sebagai referensi*

Referensi yang digunakan dalam pembuatan dokumen DPPL ini adalah sebagai berikut:

a. Dokumen Spesifikasi Kebutuhan Pearangkat Lunak (SKPL) Sistem Aplikasi Service Motor Online.

b. Modul Analisis dan Perancangan Perangkat Lunak Fakultas Informatika Telkom University.

## Sistematika Pembahasan

*Bagian ini merupakan deskripsi umum dokumen. Tuliskan sistematika pembahasan dokumen DPPL ini.*

*contoh: misalkan dalam bab1 berbicara tentang apa, bab 2 berbicara tentang apa dll*

Dokumen DPPL ini berisi deskripsi kebutuhan pengembangan perangkat lunak berbasis website secara rinci. Dokumen ini dibagi menjadi empat bagian utama, yaitu:

1. Bab 1 Pendahuluan, berisi penjelasan tentang dokumen DPPL yang mencakup tujuan pembuatan dokumen ini, lingkup masalah yang diselesaikan oleh perangkat lunak yang dikembangkan, definisi, referensi dan deskripsi umum.
2. Bab 2 Deskripsi Perancangan Global, berisi diagram dan spesifikasi kelas, komponen sistem dan arsitektur sistem dari Sistem Peminjaman Sarana Prasarana Universitas yang telah dispesifikasikan pada dokumen SKPL
3. Bab 3 Deskripsi Rinci Kebutuhan, berisi hasil analisis terhadap kebutuhan perangkat lunak berbasis website secara terperinci.
4. Bab 4 Matriks Kerutuntam, berisi mapping use case dengan kelas-kelas terkait.

# Deskripsi Perancangan Global

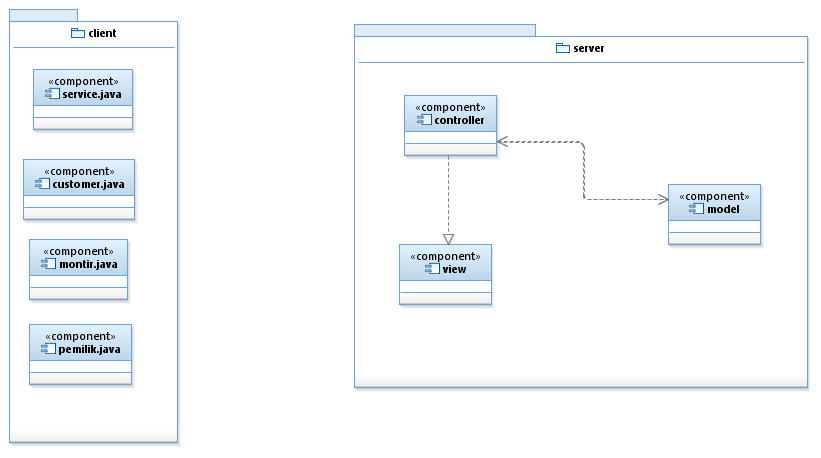
## Rancangan Lingkungan Implementasi

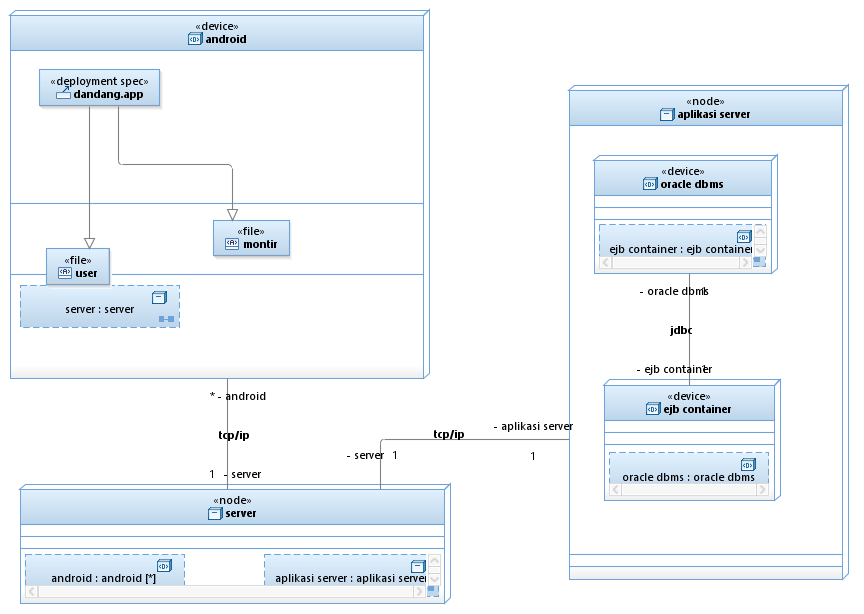
Sistem ini diimplementasikan dalam lingkungan sebagai berikut :

1. Sistem informasi : Web.
2. Bahasa Pemrograman : PHP.
3. Development Tools : XAMPP, Visual Studio Code.

## Deskripsi Arsitektural

Berikut ini merupakan komponen diagram yang digunakan untuk pengembangan sistem aplikasi service motor online :





## Deskripsi Komponen

*Diisi dengan daftar modul. Daftar modul bisa dalam bentuk tabel berikut:*

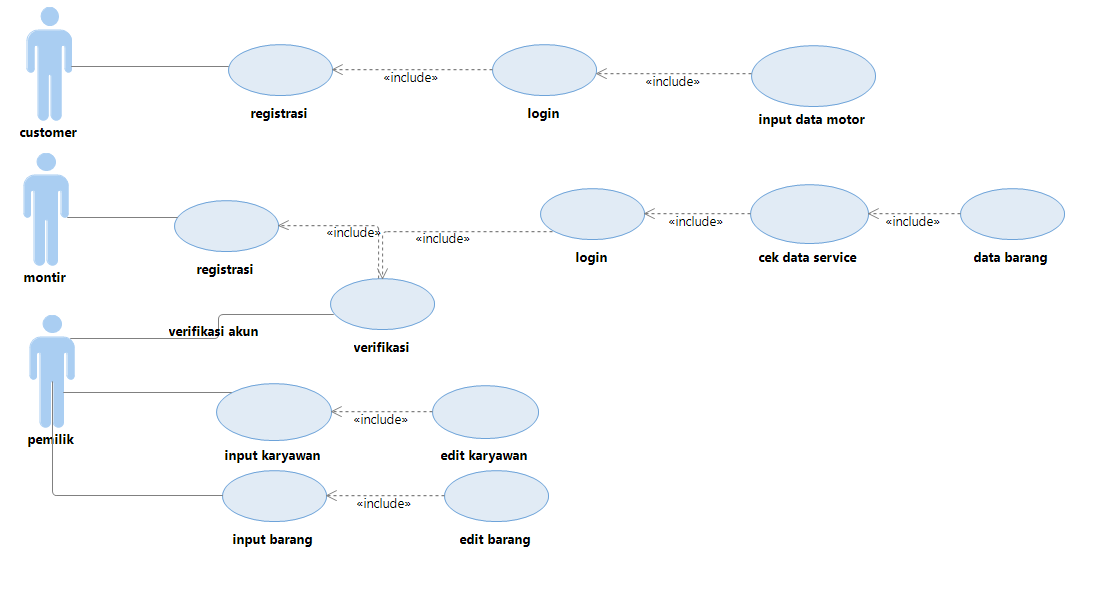
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1 | Login | Berfungsi sebagai pengalih halaman apabila login berhasil sesuai dengan tipe akun yang ada yaitu customer, montir dan pemilik |
| 2 | barang | Berfungsi sebagai modul yang berperan untuk mengatur informasi spare part termasuk pengadaan spare part |
| 3 | Security | Berfungsi sebagai modul yang berperan untuk memastikan jaminan keamanan barang (spare part ataupun motor customer) |
| 4 | Customer | Berfungsi sebagai modul yang berperan untuk mengatur keanggotaan pelanggan dalam melakukan service motor |
| 5 | Montir | Berfungsi sebagai modul yang berperan untuk memperbaiki atau memproses perbaikan dari motor yang diajukan customer dan memberitahukan persentase proses dari perbaikannya |
| 6 | Pemilik | Berfungsi sebagai modul yang berperan untuk memberikan persetujuan/verifikasi registrasi montir dan mengelola semua data barang dan data pegawai |
| 7 | Registrasi | Berfungsi sebagai pembuatan suatu akun baik customer ataupun montir yang nantinya digunakan untuk melakukan login dan memberikan sebuah akses bagi pengguna |

# Perancangan Rinci

## Realisasi Use Case

### Use Case <Sistem Aplikasi Service Motor Online>

Berikut merupakan rancangan usecase yang dibuat dalam pengembangan system aplikasi service motor online:



#### Identifikasi Kelas

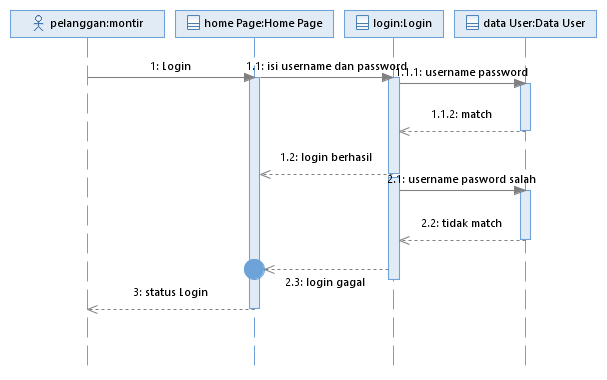
*Identifikasi kelas yang terkait dengan use case tersebut.Kelas di tahap perancangan dapat berbeda dengan dengan kelas di tahap analisis. Dapat menggunakan tabel di bawah:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| 1. | *Pemilik* | *- input karyawan*  *- edit karyawan*  *- input barang*  *- edit barang*  *- verifikasi* |
| *2.* | *Montir* | *- registrasi*  *- verifikasi*  *- login*  *- cek data service*  *- cek data barang* |
| *3.* | *Customer* | *- registrasi*  *- login*  *- input data motor* |

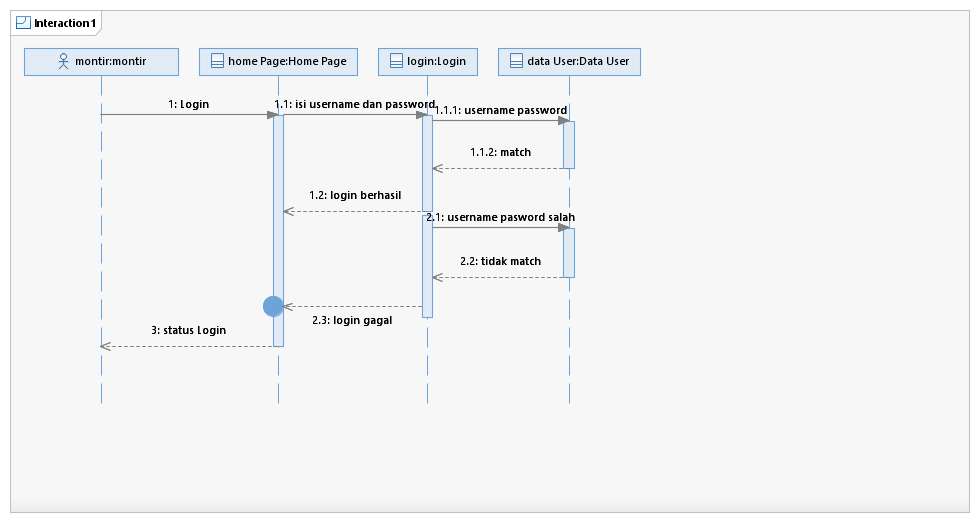
*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

#### Sequence Diagram

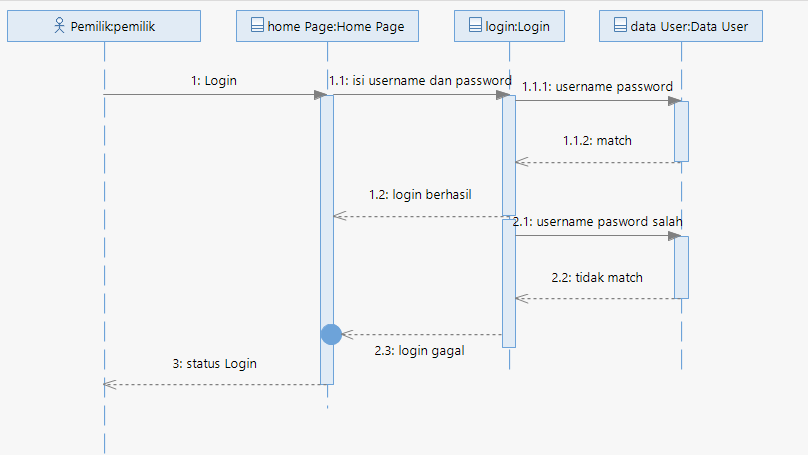
Login Pelanggan :



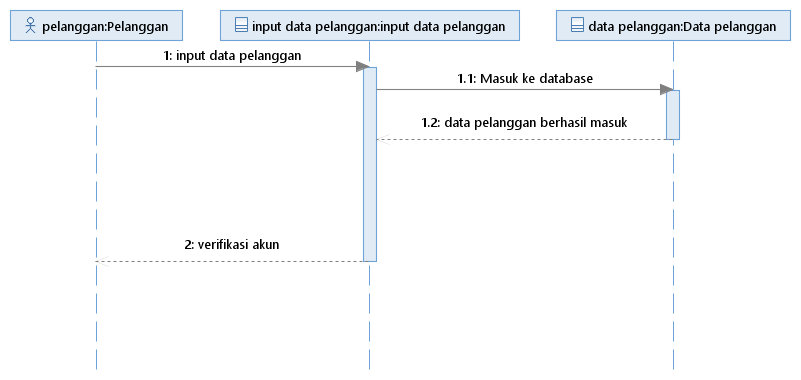
Login Montir :



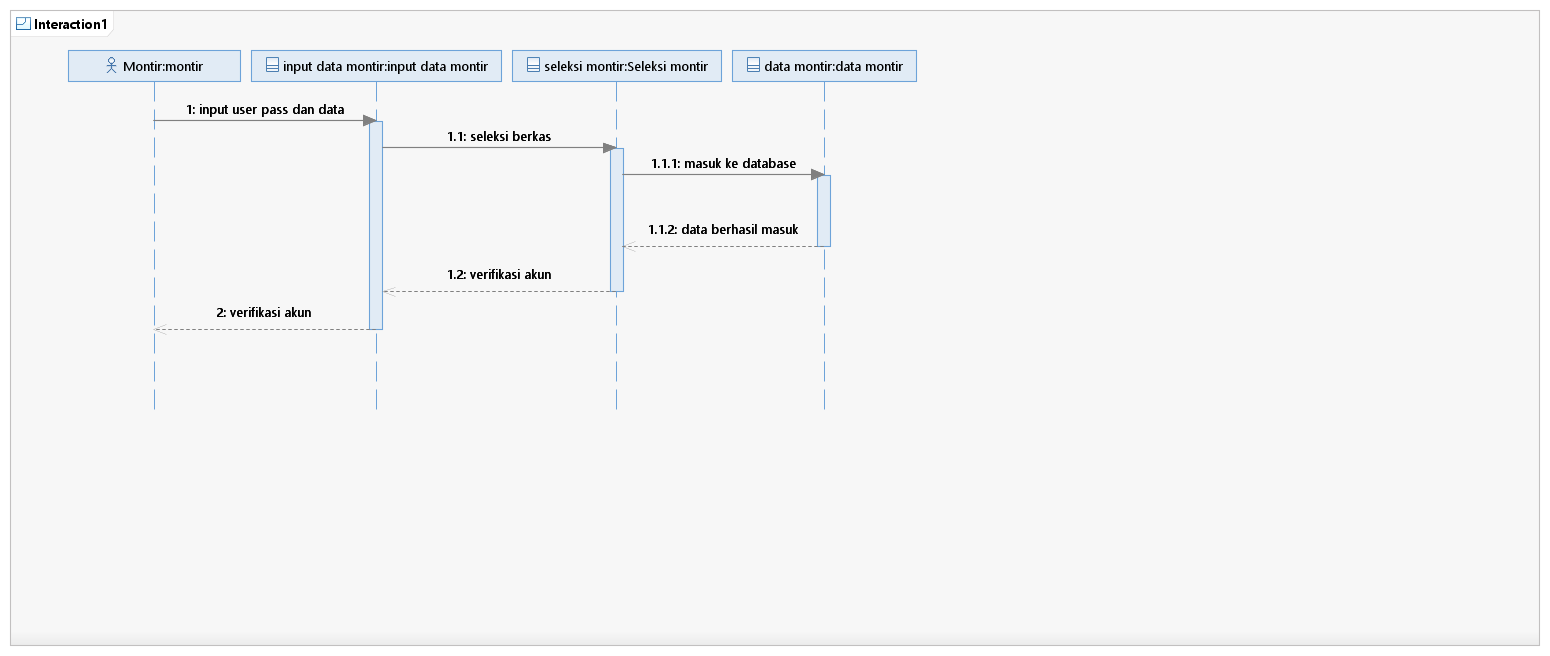
Login Pemilik :



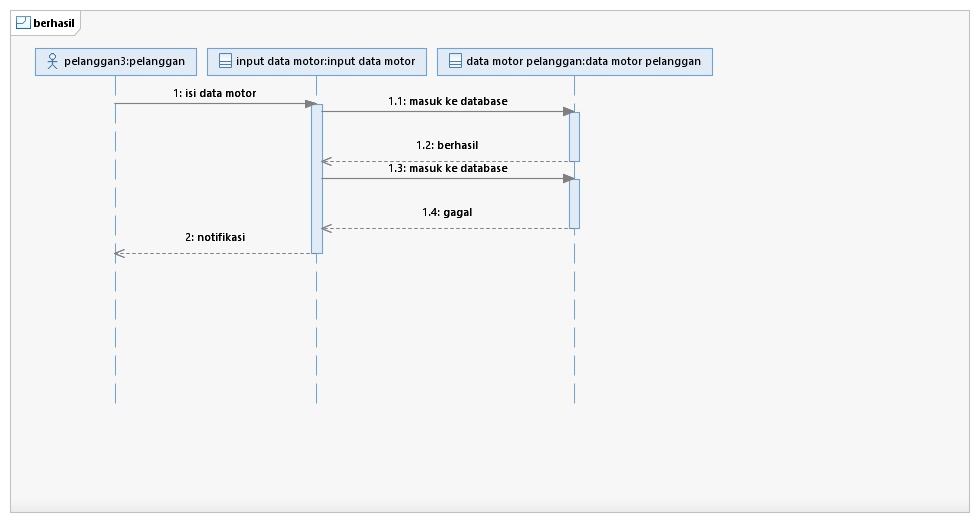
Registrasi Pelanggan



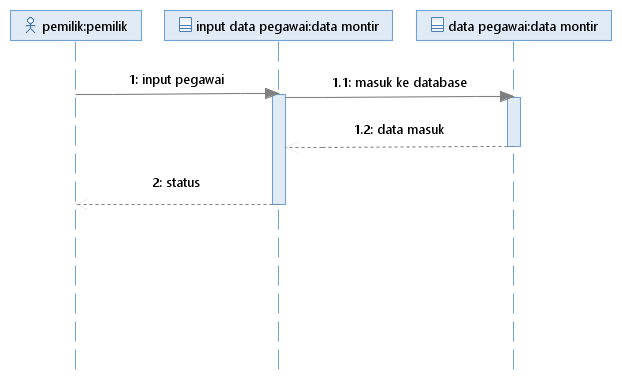
Registrasi Montir



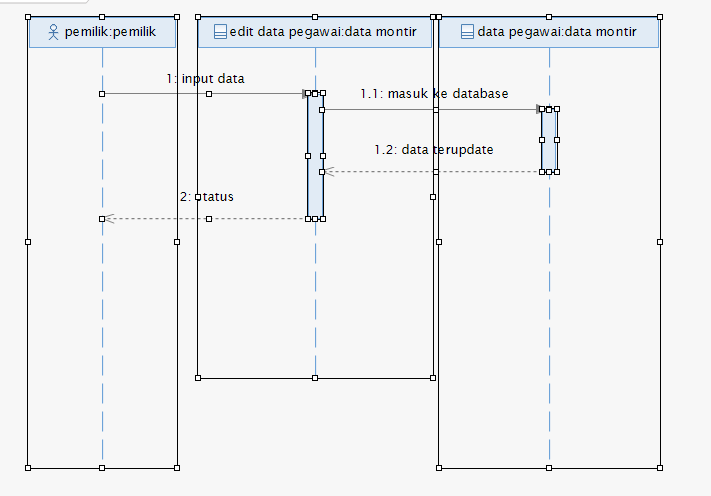
input data motor :



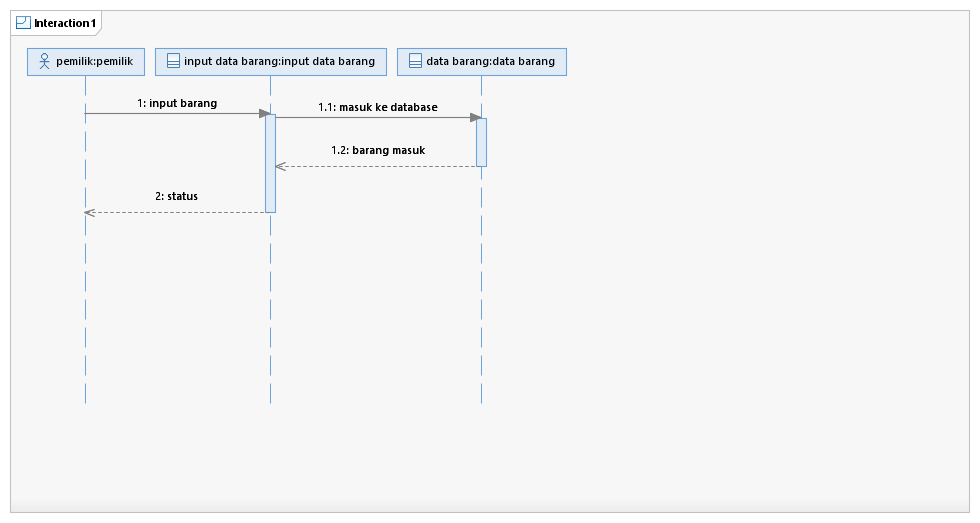
Input Karyawan



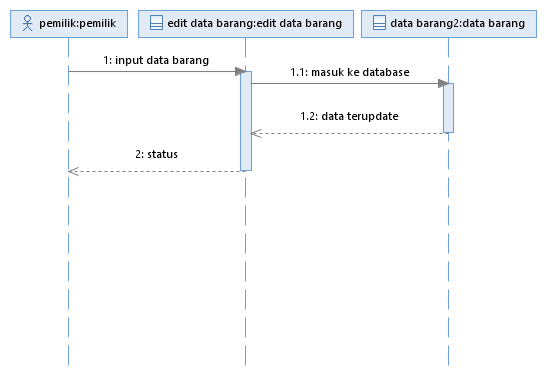
Edit data pegawai :



input data barang :



edit data barang :

**

#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut. buat class diagram* ***BUKAN KESELURUHAN, tapi PER USE CASE***

##### Use Case Login Customer



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Login | ViewCustomer |
| 2 | Proses Login | ControllerCustomer |
| 3 | Data Login | Customer |

*Untuk setiap kelas:*

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

##### Use Case Login Montir



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Login | ViewMontir |
| 2 | Proses Login | ControllerMontir |
| 3 | Data Login | Montir |

*Untuk setiap kelas:*

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

##### Use Case Login Pemilik



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Login | ViewPemilik |
| 2 | Proses Login | ControllerPemilik |
| 3 | Data Login | Pemilik |

*Untuk setiap kelas:*

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

##### Use Case Registrasi Customer



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Registrasi | ViewCustomer |
| 2 | Proses Registrasi | ControllerCustomer |
| 3 | Data Registrasi | Customer |

*Untuk setiap kelas:*

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

##### Use Case Input Data Motor



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Input Data Motor | ViewCustomer |
| 2 | Proses Input Data Motor | ControllerCustomer |
| 3 | Data Input Data Motor | Customer |

*Untuk setiap kelas:*

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

##### Use Case Cek Data Motor



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Cek Data Motor | ViewMontir |
| 2 | Proses Cek Data Motor | ControllerMontir |
| 3 | Data Cek Data Motor | Montir |

*Untuk setiap kelas:*

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

##### Use Case Data Barang



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Data Barang | ViewMontir |
| 2 | Proses Data Barang | ControllerMontir |
| 3 | Data Data Barang | Montir |

*Untuk setiap kelas:*

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

##### Use Case Input Karyawan



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Input Karyawan | ViewPemilik |
| 2 | Proses Input Karyawan | ControllerPemilik |
| 3 | Data Input Karyawan | Pemilik |

*Untuk setiap kelas:*

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

##### Use Case Edit Karyawan



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Edit Karyawan | ViewPemilik |
| 2 | Proses Edit Karyawan | ControllerPemilik |
| 3 | Data Edit Karyawan | Pemilik |

*Untuk setiap kelas:*

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

##### Use Case Input Barang



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Input Barang | ViewPemilik |
| 2 | Proses Input Barang | ControllerPemilik |
| 3 | Data Input Barang | Pemilik |

*Untuk setiap kelas:*

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

##### Use Case Edit Barang



###### **Perancangan Detil Kelas**

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Menu Edit Barang | ViewPemilik |
| 2 | Proses Edit Barang | ControllerPemilik |
| 3 | Data Edit Barang | Pemilik |

*Untuk setiap kelas:*

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

## Perancangan Detil Kelas

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| 1 | Customer | Customer |
| 2 | Montir | Montir |
| 3 | Pemilik | Pemilik |
| 4 | ControllerCustomer | ControllerCustomer |
| 5 | ControllerMontir | ControllerMontir |
| 6 | ControllerPemilik | ControllerPemilik |
| 7 | ViewCustomer | ViewCustomer |
| 8 | ViewMontir | ViewMontir |
| 9 | ViewPemilik | ViewPemilik |

*Untuk setiap kelas:*

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

### Kelas <Customer>

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

*Nama Kelas : Customer*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
|  |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Username* | *Private* | *String* |
| *Password* | *Private* | *String* |
| *Nama* | *Private* | *String* |
| *KTP* | *Private* | *String* |
| *Alamat* | *Private* | *String* |
| *Email* | *Private* | *String* |

### Kelas <Montir>

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

*Nama Kelas : Montir*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
|  |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Username* | *Private* | *String* |
| *Password* | *Private* | *String* |
| *Nama* | *Private* | *String* |
| *KTP* | *Private* | *String* |
| *Alamat* | *Private* | *String* |
| *Email* | *Private* | *String* |

### Kelas <Pemilik>

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

*Nama Kelas : Pemilik*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
|  |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Username* | *Private* | *String* |
| *Password* | *Private* | *String* |
| *Nama* | *Private* | *String* |
| *KTP* | *Private* | *String* |
| *Alamat* | *Private* | *String* |
| *Email* | *Private* | *String* |

### Kelas <ControllerCustomer>

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

*Nama Kelas :ControllerCustomer*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Login* | *Public* | *Untuk login ke sistem* |
| *Registrasi* | *Public* | *Mendaftar agar dapat melakukan service* |
| *inputDataMotor* | *Public* | *Memasukkan data motor yang akan diservice* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *noMotor* | *Private* | *String* |
| *namaMotor* | *Private* | *String* |
| *Cc* | *Private* | *String* |
| *Tahun* | *Private* | *String* |

### Kelas <ControllerMontir>

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

*Nama Kelas :ControllerMontir*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Login* | *Public* | *Untuk login ke sistem* |
| *inputStatusService* | *Public* | *Memasukkan Status service apakah sudah selesai atau belum* |
| *cekDataMotor* | *Public* | *Cek apakah data motor sesuai* |
| *cekDataBarang* | *Public* | *Cek apakah data barang masih tersedia* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *noMotor* | *Private* | *String* |
| *namaMotor* | *Private* | *String* |
| *Cc* | *Private* | *String* |
| *Tahun* | *Private* | *String* |

### Kelas <ControllerPemilik>

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

*Nama Kelas :ControllerPemilik*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Login* | *Public* | *Untuk login ke sistem* |
| *InputDataBarang* | *Public* | *Memasukkan Data barang* |
| *EditDataBarang* | *Public* | *Memperbarui data barang* |
| *InputDataKaryawan* | *Public* | *Memasukkan Data Karyawan* |
| *EditDataKaryawan* | *Public* | *Memperbarui data Karyawan* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *NamaBarang* | *Private* | *String* |
| *KodeBarang* | *Private* | *String* |
| *Stok* | *Private* | *Int* |
| *StatusBarang* | *Private* | *String* |
| *Harga* | *Private* | *Int* |
| *namaKar* | *Private* | *String* |
| *usernameKar* | *Private* | *String* |
| *passwordKar* | *Private* | *String* |
| *emailKar* | *Private* | *String* |
| *KTP\_Kar* | *Private* | *String* |

### Kelas <ViewCustomer>

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

*Nama Kelas :ViewCustomer*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ViewMenuUtama* | *Public* | *Menampilkan halaman utama aplikasi* |
| *ViewMenuLogin* | *Public* | *Menampilkan halaman login* |
| *ViewMenuRegistrasi* | *Public* | *Menampilkan halaman registrasi* |
| *ViewMenuService* | *Public* | *Menampilkan halaman service* |
| *ViewMenuBeliBarang* | *Public* | *Menampilkan halaman beli spare part* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |
|  |  |  |

### Kelas <ViewMontir>

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

*Nama Kelas :ViewMontir*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ViewMenuUtama* | *Public* | *Menampilkan halaman utama aplikasi* |
| *ViewMenuLogin* | *Public* | *Menampilkan halaman login* |
| *ViewMenuCekDataMotor* | *Public* | *Menampilkan halaman cek data motor* |
| *ViewMenuService* | *Public* | *Menampilkan halaman service* |
| *ViewDataBarang* | *Public* | *Menampilkan halaman daftar data barang* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |
|  |  |  |

### Kelas <ViewPemilik>

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

*Nama Kelas :ViewCustomer*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ViewMenuUtama* | *Public* | *Menampilkan halaman utama aplikasi* |
| *ViewMenuLogin* | *Public* | *Menampilkan halaman login* |
| *ViewMenuBarang* | *Public* | *Menampilkan halaman daftar barang* |
| *ViewInputBarang* | *Public* | *Menampilkan form input barang* |
| *ViewEditBarang* | *Public* | *Menampilkan form edit barang* |
| *ViewInputKaryawan* | *Public* | *Menampilkan form input karyawan* |
| *ViewEditKaryawan* | *Public* | *Menampilkan form edit Karyawan* |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
|  |  |  |
|  |  |  |

## Diagram Kelas Keseluruhan

*Bagian ini diisi dengan 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 : ControllerCustomer*

*Nama Operasi : Login()*

*Algoritma : (Algo-001)*

*Input (username, password)*

*Query(Q-001)*

*if(Username=true and password=true) then*

*output(“Login berhasil”)*

*masuk ke halaman utama*

*else*

*output (“Username atau Password salah”)*

*Nama Kelas : ControllerMontir*

*Nama Operasi : Login()*

*Algoritma : (Algo-002)*

*Input (username, password)*

*Query(Q-002)*

*if(Username=true and password=true) then*

*output(“Login berhasil”)*

*masuk ke halaman utama*

*else*

*output (“Username atau Password salah”)*

*Nama Kelas : ControllerPemilik*

*Nama Operasi : Login()*

*Algoritma : (Algo-003)*

*Input (username, password)*

*Query(Q-003)*

*if(Username=true and password=true) then*

*output(“Login berhasil”)*

*masuk ke halaman utama*

*else*

*output (“Username atau Password salah”)*

*Nama Kelas : ControllerCustomer*

*Nama Operasi : Registrasi*

*Algoritma : (Algo-004)*

*Input (username, password, nama, alamat, email, KTP)*

*Query(Q-004)*

*if(success) then*

*output(“Registrasi berhasil”)*

*masuk ke halaman login*

*else*

*output (“registrasi gagal”)*

*Nama Kelas : ControllerCustomer*

*Nama Operasi : inputDataMotor()*

*Algoritma : (Algo-005)*

*Input (noMotor, namaMotor, cc, tahun)*

*Query(Q-005)*

*if(success) then*

*output(“Data Motor berhasil ditambah”)*

*masuk ke halaman utama*

*else*

*output (“gagal”)*

*Nama Kelas : ControllerMontir*

*Nama Operasi : inputStatusServicer()*

*Algoritma : (Algo-006)*

*Input (statusService)*

*Query(Q-006)*

*if(Service selesai) then*

*statusService “Selesai”*

*else*

*output (“Belum Selesai”)*

*Nama Kelas : ControllerMontir*

*Nama Operasi : cekDataMotor()*

*Algoritma : (Algo-007)*

*Query(Q-006)*

*if(data ada) then*

*output(noMotor,namaMotor, cc, tahun )*

*else*

*output (“data tidak ada”)*

*Nama Kelas : ControllerMontir*

*Nama Operasi : cekDataBarang()*

*Algoritma : (Algo-008)*

*Query(Q-007)*

*if(data ada and stok tersedia) then*

*output(namaBarang, kodeBarang, stok, harga, statusBarang )*

*else*

*output (“data tidak ada”)*

*Nama Kelas : ControllerPemilik*

*Nama Operasi : inputDataBarang()*

*Algoritma : (Algo-009)*

*Query(Q-008)*

*if(data tidak ada) then*

*Input(namaBarang, kodeBarang, Stok ,harga, statusBarang)*

*else*

*output (“data sudah ada”)*

*Nama Kelas : ControllerPemilik*

*Nama Operasi : inputDataKaryawan()*

*Algoritma : (Algo-010)*

*Query(Q-009)*

*if(data tidak ada) then*

*Input(UsernameKar ,passwordKar, namaKar ,alamatKar, emailKar, KTP\_Kar)*

*else*

*output (“data sudah ada”)*

*Nama Kelas : ControllerPemilik*

*Nama Operasi : EditDataBarang()*

*Algoritma : (Algo-011)*

*Query(Q-010)*

*if(data ada) then*

*Input(namaBarang, kodeBarang, Stok ,harga, statusBarang)*

*else*

*output (“data tidak ada”)*

*Nama Kelas : ControllerPemilik*

*Nama Operasi : EditDataKaryawan()*

*Algoritma : (Algo-011)*

*Query(Q-010)*

*if(data ada) then*

*Input(UsernameKar ,passwordKar, namaKar ,alamatKar, emailKar, KTP\_Kar)*

*else*

*output (“data tidak ada”)*

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

*Query :*

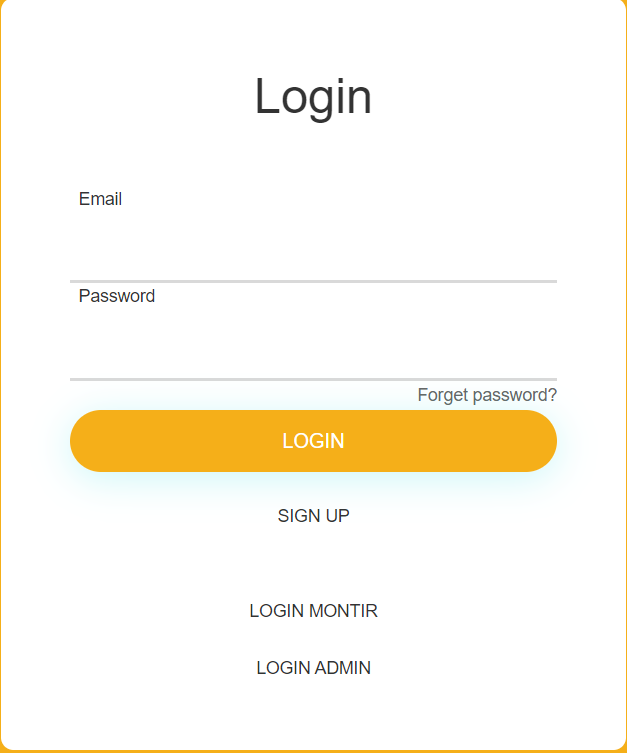
|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-001* | *SELECT \* FROM customer WHERE username=username and password = password* | *Login customer* |
| *Q-002* | *SELECT \* FROM montir WHERE username=username and password = password* | *Login montir* |
| *Q-003* | *SELECT \* FROM pemilik WHERE username=username and password = password* | *Login Pemilik* |
| *Q-004* | *INSERT INTO customer VALUES (Username, password, nama, alamat, email, KTP)* | *Registrasi customer* |
| *Q-005* | *INSERT INTO motor VALUES (noMotor, namaMotor, cc, tahun)* | *Input data motor* |
| *Q-006* | *SELECT \* FROM motor WHERE noMotor=noMotor* | *Cek data motor* |
| *Q-007* | *SELECT \* FROM barang* | *Cek data barang* |
| *Q-008* | *INSERT INTO barang VALUES (namaBarang, kodeBarang, Stok ,harga, statusBarang)* | *Input data barang* |
| *Q-009* | *INSERT INTO montir VALUES (UsernameKar ,passwordKar, namaKar ,alamatKar, emailKar, KTP\_Kar)* | *Input data karyawan* |
| *Q-010* | *UPDATE barang SET namaBarang = namaBarang, kodeBarang = kodeBarang, stok = stok, harga = harga, statusBarang = statusBarang* | *Edit data barang* |
| *Q-011* | *UPDATE montir SET usernameKar = usernameKar, passwordKar = passwordKar, namaKar = namaKar, alamatKar = alamatKar, emialKar = emailKar, KTP\_Kar = KTP\_Kar* | *Edit data karyawan* |

## Perancangan Antarmuka

*Bagian ini diisi dengan versi awal prototipe antarmuka.*

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

Login :



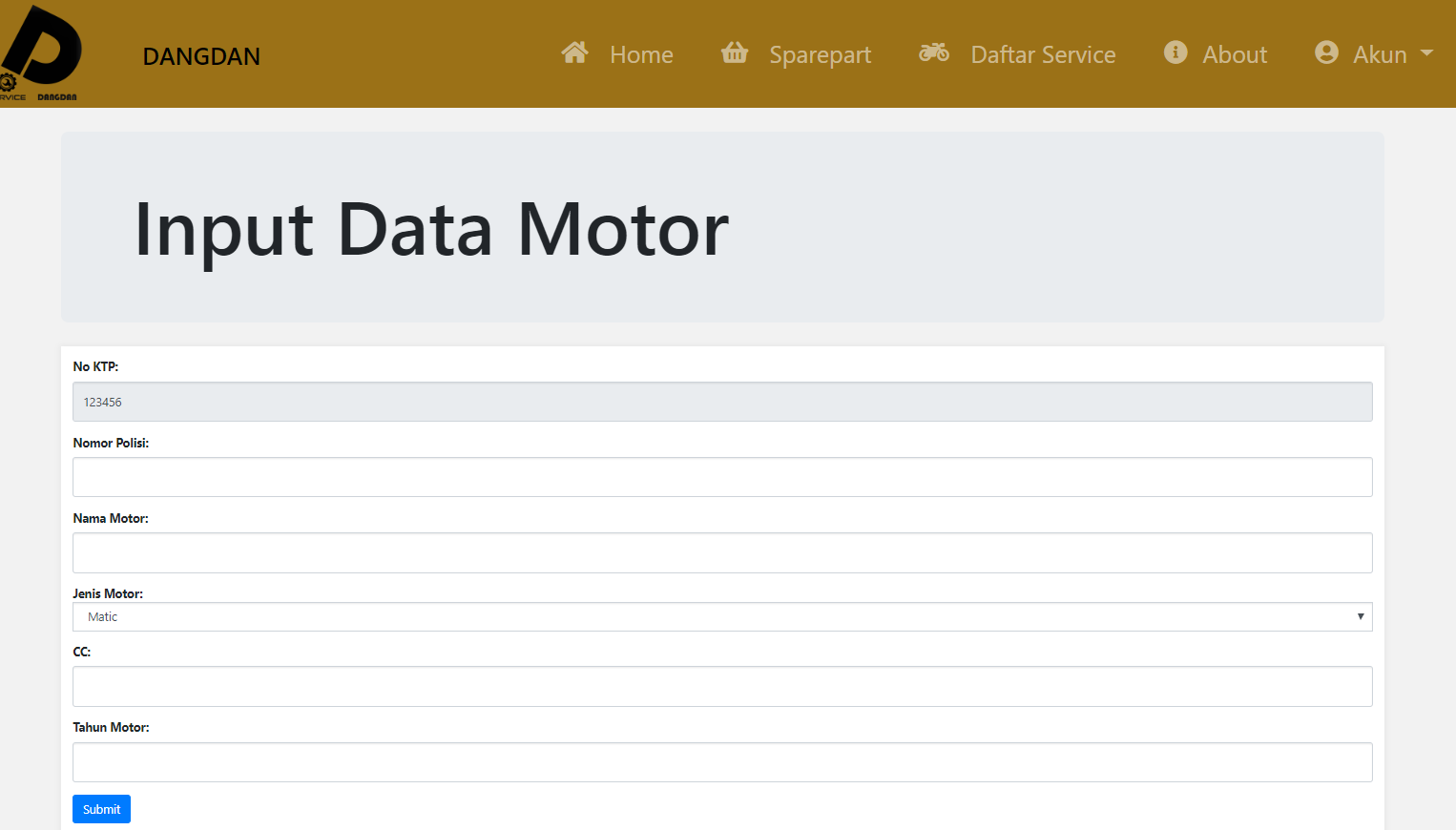
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *Input user* | *Input* | *EMAIL* | *Mengisi username yang sudah terdaftar pada database untuk melakukan login* |
| *Input pass* | *input* | *PASSWORD* | *Mengisi password yang sudah terdaftar pada database untuk melakukan login.* |
| *Link reg* | *Link* | *SIGN UP* | *Jika di klik maka akan pindah ke halaman register* |
| *Btn login* | *button* | *LOGIN* | *jika di klik dan username password tersedia di database maka akan pindah ke halaman utama* |
| *Link Montir* | *Link* | *LOGIN MONTIR* | *Jika di klik maka akan pindah ke halaman login montir* |
| *Link Admin* | *Link* | *LOGIN ADMIN* | *Jika di klik maka akan pindah ke halaman login admin* |

Register :



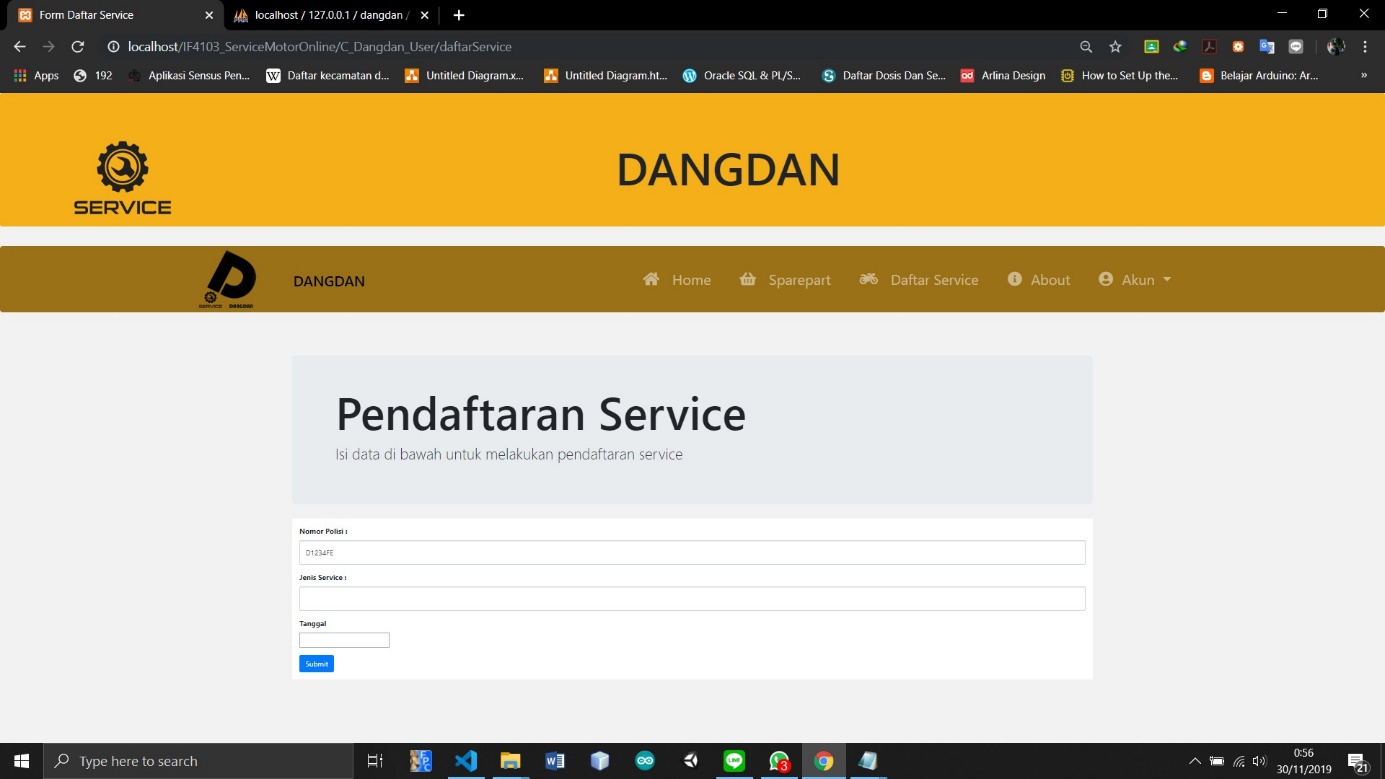
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *Input user* | *Input* | *NO-KTP* | *Mengisi username untuk dimasukkan ke database.* |
| *Input user* | *Input* | *NAMA* | *Mengisi username untuk dimasukkan ke database.* |
| *Input user* | *Input* | *USERNAME* | *Mengisi username untuk dimasukkan ke database.* |
| *Input pass* | *input* | *PASSWORD* | *Mengisi password untuk dimasukkan ke database.* |
| *Input email* | *Input* | *EMAIL* | *Mengisi email untuk dimasukkan ke database.* |
| *Input no-telp* | *Input* | *NO-TELP* | *Mengisi no-telp untuk dimasukkan ke database.* |
| *Input alamat* | *Input* | *ALAMAT* | *Mengisi alamat untuk dimasukkan ke database.* |
| *Link login* | *Link* | *LOGIN* | *Jika di klik maka akan menuju ke halaman login* |
| *Btn simpan* | *Button* | *SIMPAN* | *Jika di klik maka data yang sudah diisi akan disimpan pada database.* |

Input Data Motor :



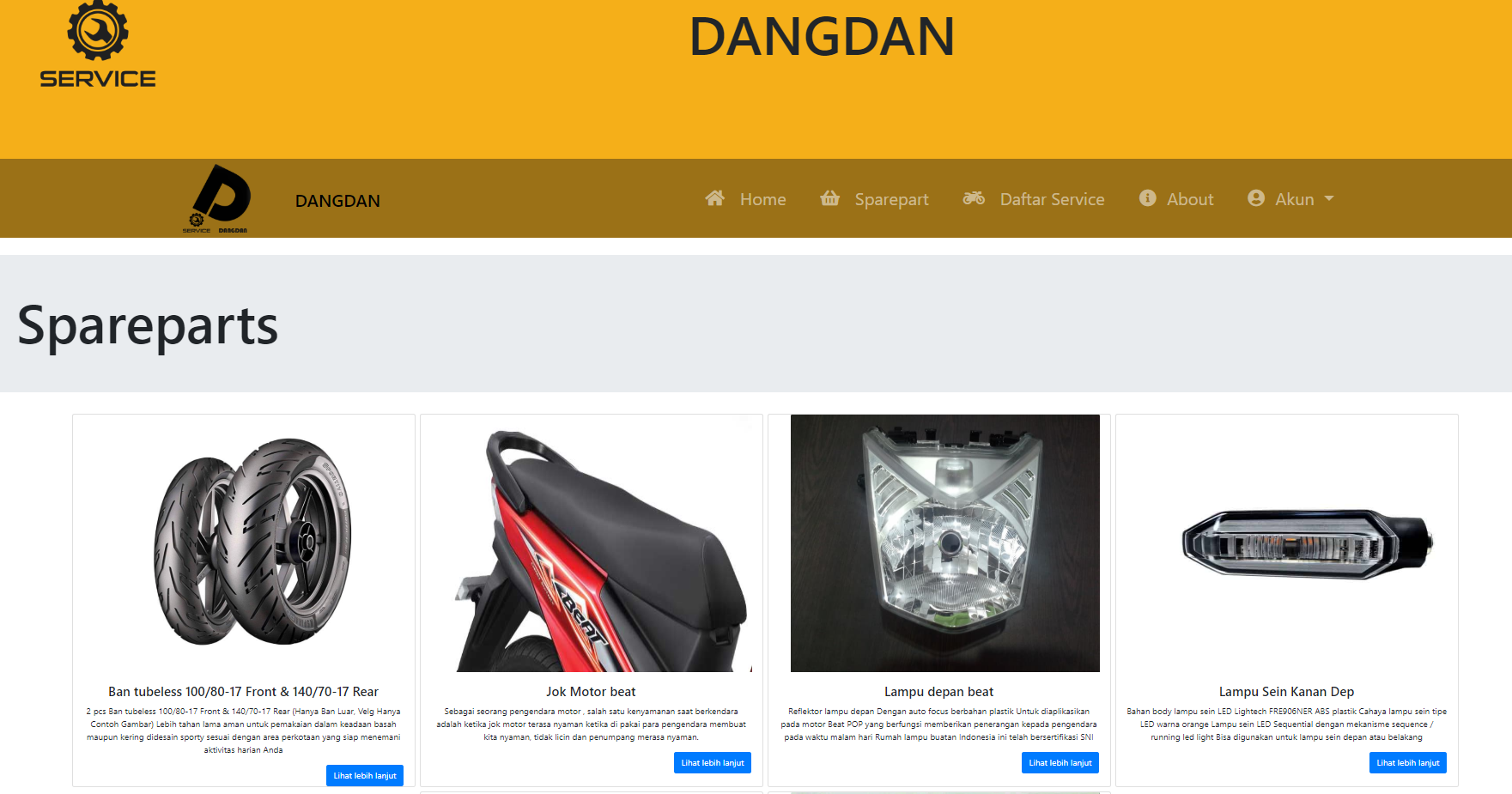
|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *Input ktp* | *Input* | *NO-KTP* | *Mengisi username untuk dimasukkan ke database.* |
| *Input nopol* | *Input* | *NOMOR POLISI* | *Mengisi username untuk dimasukkan ke database.* |
| *Input motor* | *Input* | *NAMA MOTOR* | *Mengisi username untuk dimasukkan ke database.* |
| *List motor* | *Dropdown list* | *JENIS MOTOR* | *Memilih jenis motor untuk dimasukkan ke database service.* |
| *Input cc* | *Input* | *CC* | *Mengisi username untuk dimasukkan ke database.* |
| *Input tahun* | *Input* | *TAHUN MOTOR* | *Mengisi username untuk dimasukkan ke database.* |
| *Btn submit* | *Button* | *SUBMIT* | *Jika di klik maka akan menyimpan data ke database* |

Menu Daftar Service :



|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *Input nopol* | *Input* | *NOMOR POLISI* | *Mengisi username untuk dimasukkan ke database.* |
| *Input service* | *Input* | *JENIS SERVICE* | *Mengisi username untuk dimasukkan ke database.* |
| *Input date* | *Input* | *TANGGAL* | *Mengisi username untuk dimasukkan ke database.* |
| *List tanggal* | *Drop down list* | *TANGGAL* | *Memilih tanggal pemesanan untuk dimasukkan ke database pesanan.* |
| *List jam service* | *Drop down list* | *JAM SERVICE* | *Memilih jam pemesanan untuk dimasukkan ke database pesanan.* |
| *List pilih service* | *Drop down list* | *PILIH SERVICE* | *Memilih jenis service yang akan dilakukan pada pensevicesan* |
| *Txt area catatan* | *Text area* | *CATATAN* | *Menginputkan catatan untuk penjual yang diinginkan oleh pembeli dan disimpan kedalam database pesanan.* |
| *Btn booking* | *Button* | *BOOKING* | *Jika di klik maka akan menyimpan pesanan ke dalam database dan kembali ke halaman utama.* |

Menu Beli Spareparts :



|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| *Button keterangan* | *Button* | *Lihat lebih lanjut* | *Melakukan pindah ke halaman keterangan sparepart.* |

## Perancangan Representasi Persistensi Kelas

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

A screenshot of a computer

Description automatically generated

# Matriks Kerunutan

*Mapping use case dengan kelas-kelas terkait*

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Usecase Terkait** | **Kelas** |
| FR-01 | Registrasi | Customer, ControllerCustomer, ViewCustomer |
| FR-02 | Login | Customer, Montir, Pemilik, ControllerCustomer, ControllerPemilik, ControllerMontir, ViewCustomer, ViewMontir, ViewPemilik |
| FR-03 | InputDataMotor | Customer, ControllerCustomer, ViewCustomer |
| FR-04 | CekDataService | Montir, ControllerMontir, ViewMontir |
| FR-05 | dataBarang | Montir, ControllerMontir, ViewMontir |
| FR-06 | InputKaryawan | Pemilik, ControllerPemilik, ViewPemilik |
| FR-07 | EditKaryawan | Pemilik, ControllerPemilik,ViewPemilik |
| FR-08 | inputBarang | Pemilik, ControllerPemilik,ViewPemilik |
| FR-09 | editBarang | Pemilik, ControllerPemilik,ViewPemilik |

# 