**DPPL**-xxxx

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

Pembelian Kereta Api

untuk:

Perusahaan Kereta Api

Dipersiapkan oleh:

Viebiyanty Prihatiningrum 1301164132  
Fauziah Giyanti 1301164352  
Dwi Kuncoro Aji 1301160253  
M. Arvinadia Arie K 1301164277

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Teknik Informatika  -  Fakultas Informatika | Nomor Dokumen | | Halaman |
| *DPPL-xxx* | | 33 |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A | Bab 1  Bab 2  Flowmap |
| B | Context Diagram + DFD (3.2.1)  ERD (3.3)  Perbaikan.....  ..... |
| C | Deskripsi proses (3.2.2)  Deskripsi Data (3.3)  Kebutuhan antarmuka eksternal (3.1)  Kebutuhan Non Fungsional (3.4)  Perbaikan........... |
| D | Atribut Kualitas Perangkat Lunak (3.5)  Batasan Perancangan (3.6)  Matriks keterunutan (4)  Melengkapi Lampiran ()  Perbaikan............. |
| 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

[Daftar Isi 4](#_Toc513106259)

[1. Pendahuluan 5](#_Toc513106260)

[1.1 Tujuan Penulisan Dokumen 5](#_Toc513106261)

[1.2 Lingkup Masalah 5](#_Toc513106262)

[1.3 Definisi dan Istilah 5](#_Toc513106263)

[1.4 Aturan Penamaan dan Penomoran 6](#_Toc513106264)

[1.5 Referensi 6](#_Toc513106265)

[1.6 Ikhtisar Dokumen 6](#_Toc513106266)

[2 Deskripsi Perancangan Global 8](#_Toc513106267)

[2.1 Rancangan Lingkungan Implementasi 8](#_Toc513106268)

[2.2 Deskripsi Arsitektural 8](#_Toc513106269)

[2.3 Deskripsi Komponen 9](#_Toc513106270)

[3 Perancangan Rinci 10](#_Toc513106271)

[3.1 Realisasi Use Case 10](#_Toc513106272)

[3.1.1 Use Case Registrasi 10](#_Toc513106273)

[3.1.2 Use Case Login 11](#_Toc513106274)

[3.1.3 Use Case Mengelola Data Pelanggan 12](#_Toc513106275)

[3.1.4 Use Case Pemesanan 14](#_Toc513106276)

[3.1.5 Use Case Pembelian Produk 15](#_Toc513106277)

[3.1.6 Use Case Mengelola Data Pemesanan 16](#_Toc513106278)

[3.1.7 Use Case Mendata Barang Sparepart 18](#_Toc513106279)

[3.1.8 Use Case Konfirmasi Designer 19](#_Toc513106280)

[3.2 Perancangan Detil Kelas 20](#_Toc513106281)

[3.2.1 Kelas *DataPelanggan\_View* 20](#_Toc513106282)

[3.2.2 Kelas *DataPelanggan\_Controller* 21](#_Toc513106283)

[3.2.3 Kelas *DataPelanggan\_database* 21](#_Toc513106284)

[3.2.4 Kelas *PemesananProduk\_View* 21](#_Toc513106285)

[3.2.5 Kelas *PemesananProduk\_Controller* 22](#_Toc513106286)

[3.2.6 Kelas *PemesananProduk\_database* 22](#_Toc513106287)

[3.2.7 Kelas *BarangSparepart\_View* 22](#_Toc513106288)

[3.2.8 Kelas *BarangSparepart\_Controller* 22](#_Toc513106289)

[3.2.9 Kelas](#_Toc513106290) *[BarangSparepart\_database](#_Toc513106290)* [22](#_Toc513106290)

[3.2.10 Kelas *DataDesign\_View* 23](#_Toc513106291)

[3.2.11 Kelas *DataDesign\_Controller* 23](#_Toc513106292)

[3.2.12 Kelas *DataDesign\_database* 23](#_Toc513106293)

[3.2.13 Kelas *Login\_View* 23](#_Toc513106294)

[3.2.14 Kelas *Login\_Controller* 23](#_Toc513106295)

[3.2.15 Kelas *Login\_database* 24](#_Toc513106296)

[3.2.16 Kelas *Registrasi\_View* 24](#_Toc513106297)

[3.2.17 Kelas *Registrasi\_Controller* 24](#_Toc513106298)

[3.2.18 Kelas *Registrasi\_database* 24](#_Toc513106299)

[3.2.19 Kelas *PembelianProduk\_View* 25](#_Toc513106300)

[3.2.20 Kelas *PembelianProduk\_Controller* 25](#_Toc513106301)

[3.2.21 Kelas *PembelianProduk\_database* 25](#_Toc513106302)

[3.2.22 Kelas *DataPemesanan\_View* 25](#_Toc513106303)

[3.2.23 Kelas *DataPemesanan\_Controller* 26](#_Toc513106304)

[3.2.24 Kelas *DataPemesanan\_database* 26](#_Toc513106305)

[3.3 Diagram Kelas Keseluruhan 26](#_Toc513106306)

[3.4 Algoritma/Query 27](#_Toc513106307)

[3.5 Diagram Statechart 27](#_Toc513106308)

[3.6 Perancangan Antarmuka 29](#_Toc513106309)

[3.7 Perancangan Representasi Persistensi Kelas 33](#_Toc513106310)

[4 Matriks Kerunutan 33](#_Toc513106311)

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak(DPPL) yang dibuat, digunakan sebagai landasan dalam proses pembuatan aplikasi Penjualan Kereta Api. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

## Lingkup Masalah

Nama aplikasi berbasis web yang kami rancang adalah buyTrain. Perangkat lunak yang kita kembangkan merupakan perangkat lunak yang bergerak di bidang transportasi khususnya kereta api, dimana perangkat lunak ini akan memudahkan konsumen dalam pembelian produk berupa kereta api sehingga produk yang diinginkan akan sesuai dengan tujuan perusahaan kereta api tersebut. Konsumen hanya perlu menyampaikan produk yang diinginkan sehingga nantinya akan di analisis oleh perusahaan kereta api tersebut dan diproduksilah produk yang diinginkan. Aplikasi yang kami kembangkan ini juga merupakan aplikasi baru sehingga diharapkan dapat sebagai acuan perusahaan startup lain. Selain itu, kami juga berharap bahwa web kami dapat diterima di kalangan perusahaan kereta api, sehingga dapat memudahkan perusahaan dalam transaksi jual-beli kereta api.

## Definisi dan Istilah

|  |  |  |
| --- | --- | --- |
| **NO** | **Istilah/Singkatan** | **Definisi** |
| 1. | DPPL | Dokumen deskripsi dari perancangan perangkat lunak yang akan dikembangkan dan bertujuan untuk memberikan landasan yang diperlukan dalam proses pengkodean aplikasi Q-AS. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya. |
| 2. | Sequence Diagram | salah satu dari diagram-diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object |
| 3. | Operating system | perangkat lunak sistem yang mengatur sumber daya dari perangkat keras dan perangkat lunak, serta sebagai jurik (daemon) untuk program komputer |
| 4. | DBMS | perangkat lunak (software) yang dipakai untuk membangun basis data yang berbasis komputerisasi |
| 5. | development tools | salah satu tipe proyek dalam bidang IT yang berfokus pada menciptakan atau mengembangkan sebuah software untuk memenuhi kebutuhan manusia, mendesain software, mengembangkan sebuah software aplikasi pada perusahaan, mengembangkan platfrom |
| 6. | filing system | suatu rangkaian kerja yang teratur yang dapat dijadikan pedoman untuk penyimpanan arsip sehingga saat diperlukan dapat ditemukan kembali dengan cepat dan tepat. Dengan menggunakan filing system yang tepat, penyimpanan dan penemuan kembali arsip dapat dilakukan dengan mudah tanpa membuang waktu. |
| 7. | Diagram komponen | diagram UML yang menampilkan komponen dalam system dan hubungan antara mereka. |
| 8. | Diagram statechart | suatu diagram yang menggambarkan daur hidup (behavior pattern) dari sebuah objek, dari awal objek tsb diinisialisasi sampai di-destroy. Menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu obyek pada sistem sebagai akibat dari stimulans yang diterima. |

## Aturan Penamaan dan Penomoran

Penamaan dan penomoran dilakukan untuk:

* Sequence Diagram

Sistem penamaan untuk Sequence diagram dibuat dengan format SQ-D-XX-YYXX merupakan nomor kode usecase,YY merupakan versi sequence diagram untuk satu usecase (satu usecase dapat terdiridari beberapa sequence diagram),Contoh : SQ-D-03-02, maka itu meruapakan suatu sequence diagram versi 2 untuk usecase no.3,-

* Diagram Kelas

Sistem penamaan untuk Sequence diagram dibuat dengan format CL-D-ZZZZ merupakan nomor kode usecase, tiap usecase terdiri dari satu diagram kelasContoh : CL-D-02, maka itu meruapakansuatu diagram kelas untuk usecase no.2,

## Referensi

Dalam pembuatan dokumen ini kami menggunakan dokumen-dokumen lain yang ada di internet sebagai acuan penulisan dokumen ini.

## Ikhtisar Dokumen

DPPL ini berisi penjabaran rancangan dari perangkat lunak yang akan dikembangkan, sehingga pada tahap pengimplementasian, perangkat lunak tersebut dapat diimplementasikan dengan spesifikasi yang jelas dengan tetap menjadikan spesifikasi kebutuhan perangkat lunak(SKPL) sebagai bahan acuannya.

|  |  |
| --- | --- |
| **BAB I** | **PENDAHULUAN** |
|  | Berisi overview perangkat lunak yang merupakan ringkasan terhadap deskripsi aplikasi secara umum yang meliputi tujuan, lingkup masalah, definisi dan istilah, aturan penamaan dan penomoran, referensi serta ikhtisar dokumen. |
| **BAB II** | **DESKRIPSI PERANCANGAN GLOBAL** |
|  | Berisi perancangan global dari perangkat lunak yang meliputi lingkungan di mana perangkatlunak diimplementasikan, deskripsi arsitektural maupun deskripsi komponen dari perangkatlunak yang dibangun. |
| **BAB III** | **DESKRIPSI PERANCANGAN RINCI** |
|  | Berisi deskripsi lengkap dan rinci dari kebutuhan perangkat lunak yang meliputi diagram usecase, sequence diagram tiap scenario, diagram kelas beserta detilnya, algoritma serta querypenting, statechart untuk use case penting, antarmuka, representasi persistensi kelas serta matrikskerunutan tiap use case. |
| **BAB IV** | **MATRIKS KERUNUTAN** |
|  | Mapping use case dengan kelas-kelas terkait |

# Deskripsi Perancangan Global

## Rancangan Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk pengembangan system Pembelian Kereta Api dengan menggunakan metode sebagai berikut :

Sistem Operasi : Windows

DBMS : MySQL

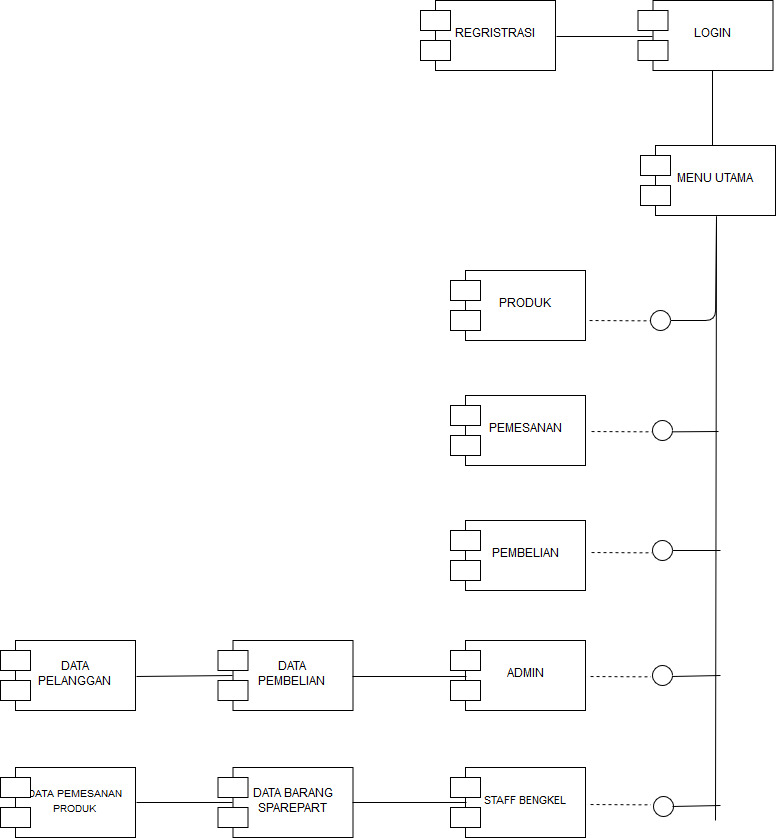
Development Tools : Sublime

Filing System :

Bahasa Pemrograman :HTML, PHP, CSS

## Deskripsi Arsitektural

Perangkat lunak ini dibangun dengan aplikasi berbasis web. Dengan menu utama sebagai fungsional system. Sebelum masuk ke menu utama, actor harus login dan jika belum bsa login mereka harus regristrasi terlebih dahulu.



## Deskripsi Komponen

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

| **No** | **Nama Komponen** | **Keterangan** |
| --- | --- | --- |
| 1 | Login | Syarat untuk masuk ke Menu Utama |
| 2 | Regristrasi | Syarat untuk bias login |
| 3 | Menu Utama | Antarmuka yang berisi kelas-kelas utama Aplikasi |
| 4 | Produk | Daftar produk tersedia maupun tidak tersedia (Bagian dari Menu Utama) |
| 5 | Pemesanan | Kelas Pemesanan (Bagian dari Menu Utama) |
| 6 | Pembelian | Kelas Pembelian (Bagian dari Menu Utama) |
| 7 | Admin | Kelas Admin (Bagian dari Menu Utama) |
| 8 | Data Pembelian | Didalam kelas admin berisi data pembelian |
| 9 | Data Pelanggan | Didalam kelas admin berisi data pelanggan |
| 10 | Staff Bengkel | Kelas Staff Bengkel (Bagian dari Menu Utama) |
| 11 | Data Barang Sparepart | Didalam kelas Staff Bengkel berisi data barang sparepart |
| 12 | Data Pemesanan Produk | Didalam kelas Staff Bengkel berisi data pemesanan produk |

# Perancangan Rinci

## Realisasi Use Case

### Use Case Registrasi

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

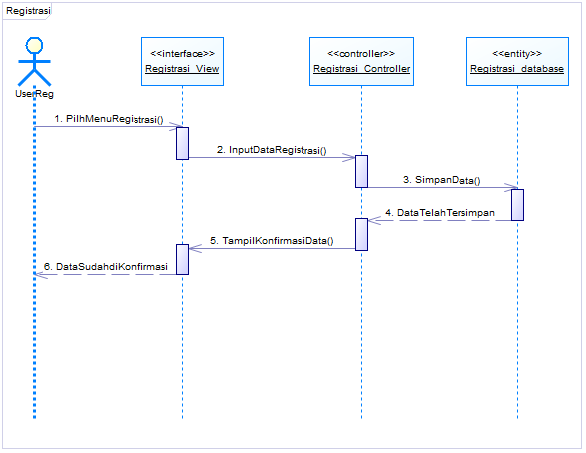
#### Identifikasi Kelas

*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* | *Nama Kelas Analisis Terkait* |
| *1.* | *Registrasi\_View* | *Registrasi* |
| *2.* | *Registrasi\_Controller* | *Registrasi* |
| *3.* | *Registrasi\_database* | *Registrasi* |

#### Sequence Diagram

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



#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

### Use Case Login

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 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* | *Nama Kelas Analisis Terkait* |
| *1.* | *Login\_View* | *Login* |
| *2.* | *Login\_Controller* | *Login* |
| *3.* | *Login\_database* | *Login* |

#### Sequence Diagram

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

**

#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

### Use Case Mengelola Data Pelanggan

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 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* | *Nama Kelas Analisis Terkait* |
| *1.* | *DataPelanggan\_View* | *Pelanggan* |
| *2.* | *DataPelanggan\_Controller* | *Pelanggan* |
| *3.* | *DataPelanggan\_database* | *Pelanggan* |

#### Sequence Diagram

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

**

#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

### Use Case Pemesanan

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 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* | *Nama Kelas Analisis Terkait* |
| *1.* | *PemesananProduk\_View* | *Produk* |
| *2.* | *PemesananProduk\_Controller* | *Produk* |
| *3.* | *PemesananProduk\_database* | *Produk* |

#### Sequence Diagram



#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

### Use Case Pembelian Produk

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 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* | *Nama Kelas Analisis Terkait* |
| *1.* | *PembelianProduk\_View* | *Transaksi* |
| *2.* | *PembelianProduk\_Controller* | *Transaksi* |
| *3.* | *PembelianProduk\_database* | *Transaksi* |

#### Sequence Diagram

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



#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

### Use Case Mengelola Data Pemesanan

Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

#### 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* | *Nama Kelas Analisis Terkait* |
| *1.* | *DataPemesanan\_View* | *Transaksi* |
| *2.* | *DataPemesanan\_Controller* | *Transaksi* |
| *3.* | *DataPemesanan\_database* | *Transaksi* |

#### Sequence Diagram



#### Diagram Kelas

**

### Use Case Mendata Barang Sparepart

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1.* | *BarangSparepart\_View* | *Sparepart* |
| *2.* | *BarangSparepart\_Controller* | *Sparepart* |
| *3.* | *BarangSparepart\_database* | *Sparepart* |

#### Sequence Diagram



#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

### Use Case Konfirmasi Designer

#### Identifikasi Kelas

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1.* | *DataDesign\_View* | *Produk* |
| *2.* | *DataDesign\_Controller* | *Produk* |
| *3.* | *DataDesign\_database* | *Produk* |

#### Sequence Diagram

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

**

#### Diagram Kelas

*Buatlah diagram kelas untuk use case tersebut.*

**

## Perancangan Detil Kelas

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

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
| *1.* | *Login\_View* | *Login* |
| *2.* | *Login\_Controller* | *Login* |
| *3.* | *Login\_database* | *Login* |
| *4.* | *Registrasi\_View* | *Registrasi* |
| *5.* | *Registrasi­\_Controller* | *Registrasi* |
| *6.* | *Registrasi\_database* | *Registrasi* |
| *7.* | *DataPelanggan\_View* | *Pelanggan* |
| *8.* | *DataPelanggan\_Controller* | *Pelanggan* |
| *9.* | *DataPelanggan\_database* | *Pelanggan* |
| *10.* | *PemesananProduk\_View* | *Produk* |
| *11.* | *PemesananProduk\_Controller* | *Produk* |
| *12.* | *PemesananProduk\_database* | *Produk* |
| *13.* | *BarangSparepart\_View* | *Sparepart* |
| *14* | *BarangSparepart\_Controller* | *Sparepart* |
| *15.* | *BarangSparepart\_databases* | *Sparepart* |
| *16.* | *DataDesign\_View* | *Produk* |
| *17.* | *DataDesign\_Controller* | *Produk* |
| *18.* | *DataDesign\_database* | *Produk* |
| *19.* | *PembelianProduk\_View* | *Transaksi* |
| *20.* | *PembelianProduk\_Controller* | *Transaksi* |
| *21.* | *PembelianProduk\_database* | *Transaksi* |
| *22.* | *DataPemesanan\_View* | *Transaksi* |
| *23.* | *DataPemesanan\_Controller* | *Transaksi* |
| *24.* | *DataPemesanan\_database* | *Transaksi* |

*Untuk setiap kelas:*

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

### Kelas *DataPelanggan\_View*

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

*Nama Kelas : DataPelanggan\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *TampilanMenuData()* | *public* | *menampilkan tampilan dari menu data.* |
| *PilihMenuEditData()* | *public* | *menampilkan tampilan dari menu edit data.* |
| *PilihDataPelanggan()* | *public* | *menampilkan pilihan data pelanggan yang akan dipilih oleh admin.* |
| *PilihHapusData()* | *public* | *Menampilkan tampilan ketika ingin menghapus data.* |
| *TampilPilihanPelanggan()* | *public* | *Menampilkan pilihan pelanggan.* |
| *TampilDataPelanggan()* | *public* | *menampilkan tampilan data dari para pelanggan* |

### Kelas *DataPelanggan\_Controller*

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

*Nama Kelas : DataPelanggan\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihPelanggan()* | *public* | *Memilih pilihan pelanggan.* |
| *PilihEditDataPelanggan()* | *public* | *Memilih edit data pelanggan.* |
| *InsertPilDataPelanggan()* | *public* | *Menerima input pilihan data pelanggan.* |
| *PilihHapusDataPelanggan()* | *public* | *Memilih hapus data pelanggan.* |

### Kelas *DataPelanggan\_database*

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

*Nama Kelas : DataPelanggan\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *GetDataPelanggan()* | *public* | *Memanggil data pelanggan.* |
| *EditDataPelanggan()* | *public* | *Mengedit data pelanggan yg disimpan di database* |
| *HapusDataPelanggan()* | *public* | *Menghapus data pelanggan dan disimpan di database* |
| *EditData()* | *public* | *Melakukan edit data pelanggan* |
| *HapusData()* | *public* | *Melakukan hapus data pelanggan.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Nama* | *private* | *String* |
| *Alamat* | *private* | *String* |
| *Tgl\_lahir* | *private* | *Date* |
| *Notelp* | *private* | *int* |

### Kelas *PemesananProduk\_View*

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

*Nama Kelas : PemesananProduk\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihMenuProduk()* | *public* | *menampilkan tampilan dari menu produk.* |
| *PilihProduk()* | *public* | *menampilkan pilihan Produk yang akan dipilih.* |
| *TampilKatalogProduk()* | *public* | *menampilkan katalog produk yang diminta oleh pelanggan.* |
| *TampilVerifikasiPesanan()* | *public* | *Menampilkan verifikasi pesanan yang sudah dipesan oleh pelanggan.* |

### Kelas *PemesananProduk\_Controller*

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

*Nama Kelas : :PemesananProduk\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ReqKatalogProduk()* | *public* | *Meminta katalog produk.* |
| *InsertPilProduk()* | *public* | *Menerima input dari pilih produk* |
| *PilihDataPelanggan()* | *public* | *menampilkan pilihan data pelanggan yang akan dipilih oleh admin.* |
| *PilihHapusData()* | *public* | *Menampilkan tampilan ketika ingin menghapus data.* |

### Kelas *PemesananProduk\_database*

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

*Nama Kelas : PemesananProduk\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *SetPilProduk()* | *public* | *Memasukkan pilihan produk.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *NamaProduk* | *private* | *String* |
| *JumlahProduk* | *private* | *int* |

### Kelas *BarangSparepart\_View*

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

*Nama Kelas : BarangSparepart\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihMenuSparepart()* | *public* | *Menampilkan menu sparepart.* |
| *TampilDaftarSparepart()* | *public* | *Menampilkan daftar sparepart yang diminta.* |
| *PilihDataSparepart()* | *public* | *Menampilkan data sparepart.* |

### Kelas *BarangSparepart\_Controller*

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

*Nama Kelas : BarangSparepart\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ReqDaftarSparepart()* | *public* | *Meminta daftar sparepart.* |
| *InputDataSparepart()* | *public* | *Menginput data sparepart.* |

### Kelas *BarangSparepart\_database*

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

*Nama Kelas : BarangSparepart\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *SaveDataSparepart()* | *public* | *Menyimpan data sparepart.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *NamaSparepart* | *private* | *String* |
| *JumlahSparepart* | *private* | *int* |

### Kelas *DataDesign\_View*

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

*Nama Kelas : DataDesign\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihMenuDesign()* | *public* | *Menampilkan menu design.* |
| *ValidasiDesign()* | *public* | *Menampilkan validasi design yang diambil.* |

### Kelas *DataDesign\_Controller*

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

*Nama Kelas : DataDesign\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ReqDataDesign()* | *public* | *Meminta data design.* |

### Kelas *DataDesign\_database*

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

*Nama Kelas : DataDesign\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *GetDataDesign()* | *public* | *Mengambil data design.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *NamaDesign* | *private* | *String* |
| *JumlahDesign* | *private* | *int* |

### Kelas *Login\_View*

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

*Nama Kelas : Login\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihLogin()* | *public* | *Menmpilkan tampilan login.* |
| *TampilLoginBerhasil()* | *public* | *Menampilkan pesan login berhasil.* |

### Kelas *Login\_Controller*

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

*Nama Kelas : Login\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *InputUserdanPass()* | *public* | *Menginput user dan password.* |

### Kelas *Login\_database*

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

*Nama Kelas : Login\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *CekDataUser()* | *public* | *Mengecek data inputan user di database.* |
| *ValidasiUser()* | *public* | *Memvalidasi apakah user valid atau tidak.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Username* | *private* | *String* |
| *Password* | *private* | *int* |

### Kelas *Registrasi\_View*

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

*Nama Kelas : Registrasi\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihMenuRegistrasi()* | *public* | *Menampilkan tampilan menu registrasi.* |
| *TampilKonfirmasiData()* | *public* | *Menampilkan konfirmasi data registrasi yang sudah disimpan.* |

### Kelas *Registrasi\_Controller*

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

*Nama Kelas : Registrasi\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *InputDataRegistrasi()* | *public* | *Menginputkan data registrasi.* |

### Kelas *Registrasi\_database*

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

*Nama Kelas : Registrasi\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *SimpanDataRegistrasi()* | *public* | *Menyimpan data registrasi yang sudah diinput oleh user.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Nama* | *private* | *String* |
| *Email* | *private* | *String* |
| *Notelp* | *private* | *int* |

### Kelas *PembelianProduk\_View*

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

*Nama Kelas : PembelianProduk\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihMenuBelanja()* | *public* | *Menampilkan menu belanja.* |
| *TampilValidasiPesanan()* | *public* | *Menampilkan validasi pesanan.* |
| *PilihMenuPembayaran()* | *public* | *Menampilkan menu pembayaran.* |
| *TampilPembayaranBerhasil()* | *public* | *Menampilkan tampilan pesan pembayaran berhasil.* |

### Kelas *PembelianProduk\_Controller*

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

*Nama Kelas : PembelianProduk\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *InputDataPesanan()* | *public* | *Menginputkan data pesanan.* |
| *InputPembayaran()* | *public* | *Melakukan pembayaran* |

### Kelas *PembelianProduk\_database*

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

*Nama Kelas : PembelianProduk\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *SaveDataPesanan()* | *public* | *Menyimpan data pesanan.* |
| *KonfirmasiPembayaran()* | *public* | *Memberikan bukti konfirmasi pembayaran dan disimpan di database.* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *KodePesanan* | *private* | *String* |
| *JumlahPesanan* | *private* | *Int* |
| *NamaPesanan* | *private* | *String* |

### Kelas *DataPemesanan\_View*

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

*Nama Kelas : DataPemesanan\_View*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *PilihMenuPesanan()* | *public* | *Menampilkan tampilan menu pesanan.* |
| *PilihDataPesanan()* | *public* | *Menampilkan tampilan data pesanan* |
| *TampilDaftarPesanan()* | *public* | *Menampilkan daftar pesanan* |
| *TampilDataPesanan()* | *public* | *Menampilkan data pesanan* |
| *PilihMenuEdit()* | *public* | *Menampilkan tampilan dari menu edit* |
| *PilihHapusPesanan()* | *public* | *Menampilkan tampilan hapus pesanan* |

### Kelas *DataPemesanan\_Controller*

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

*Nama Kelas : DataPemesanan\_Controller*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *ReqDaftarPesanan()* | *Public* | *Meminta daftar pesanan* |
| *InsertPilDataPesanan()* | *Public* | *Menerima inputan pilih data pesanan* |
| *PilihEditPesanan()* | *Public* | *Menampilkan tampilan ketika akan edit pesanan* |
| *PilihHapusPesanan()* | *Public* | *Menampilkan tampilan ketika akan hapus pesanan* |

### Kelas *DataPemesanan\_database*

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

*Nama Kelas : DataPemesanan\_database*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *GetDataPesanan()* | *public* | *Mengambil data pesanan dari database* |
| *EditDataPesanan()* | *public* | *Mengedit data pesanan* |
| *HapusDataPesanan()* | *public* | *Menghapus data pesanan* |
| *EditData()* | *public* | *Melakukan edit data pesanan* |
| *HapusData()* | *public* | *Melakukan hapus data pesanan* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *KodePesanan* | *private* | *String* |
| *JumlahPesanan* | *private* | *Int* |
| *NamaPesanan* | *private* | *String* |

## Diagram Kelas Keseluruhan

*Bagian ini diisi dengan diagram kelas keseluruhan.*

**

## Algoritma/Query

*Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses 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 :Login\_database*

*Nama Operasi :ValidasiUser()*

*Algoritma : (Algo-001)*

*If(ValidasiUser=true)then*

*Output(“Login Berhasil”)*

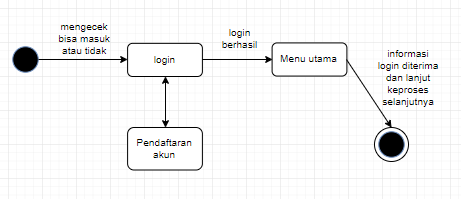
*Else*

*Output(“Username dan Password Salah”)*

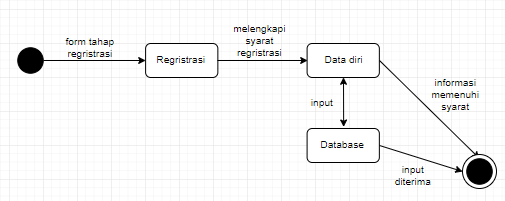
## Diagram Statechart

*Bagian ini hanya diisi jika ada kelas yang memiliki atribut dengan value yang dinamis (berubah-ubah sesuai state atau kondisi). Jika tidak ada kedinamisan value dari atribut kelas tersebut, diagram ini tidak perlu dibuat. Perubahan status kelas tersebut harus digambarkan dalam bentuk diagram statechart. Boleh dibuat subbab per kelas.*

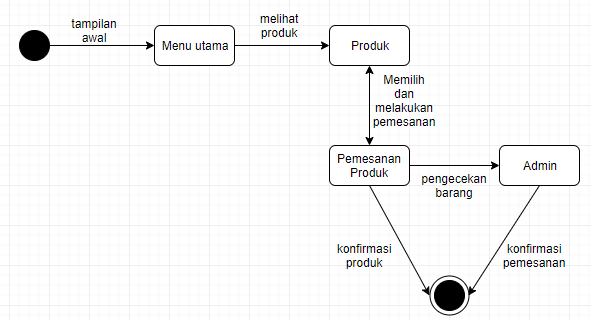
1. Proses login



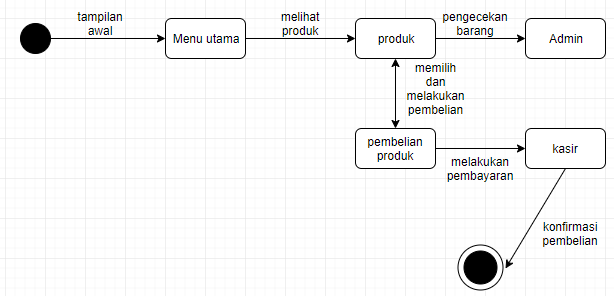
1. Proses Regristrasi



1. Proses Pemesanan Barang

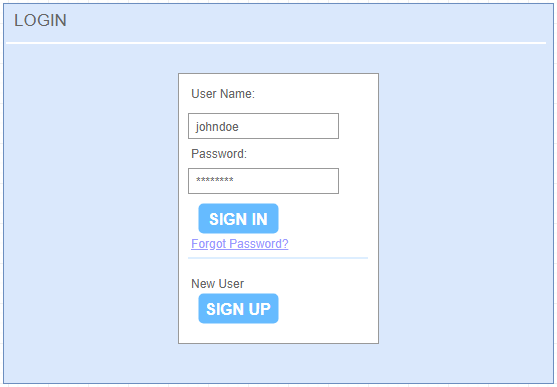


1. Proses Pembelian Barang



## Perancangan Antarmuka

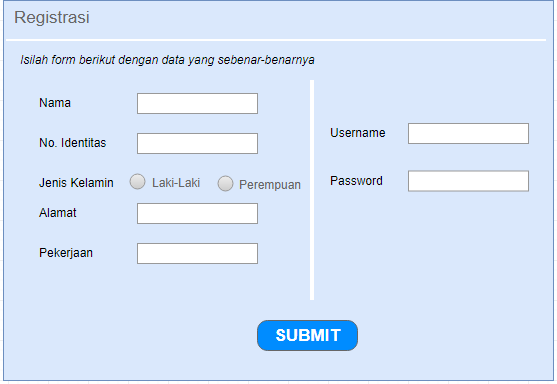
*Antarmuka : Login*

**

Gambar 1. Interface halaman login

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Input\_l1* | *text* | *username* | Text untuk menginputkan username pengguna |
| *Input\_l2* | Text | Password | Text untuk menginputkan password pengguna |
| *Btn\_l1* | *Button* | *Signin* | Tombol untuk mensubmitkan username dan password agar dapat memasuki halaman selanjutnya |
| *Btn\_l2* | *Button* | *Signup* | Tombol untuk menuju halaman registrasi apabila pengguna belum memiliki akun |
| *Link\_l* | *Link* | *Forgot Password?* | Link untuk menuju halamn lupa password apabila pengguna lupa password |

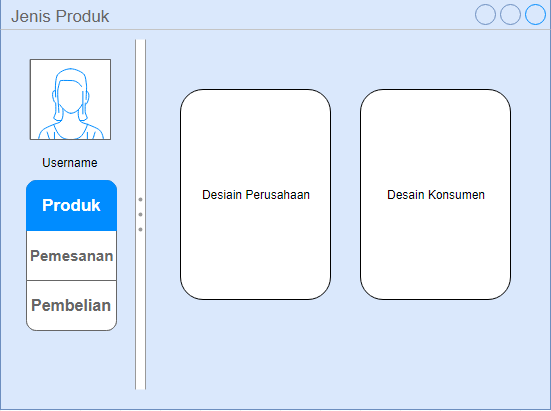
*Antarmuka : Registrasi*



Gambar 2. Interface halaman registrasi

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Input\_r1* | *text* | *nama* | Text untuk menginputkan nama pengguna |
| *Input\_r2* | Text | nip | Text untuk menginputkan no identitas pengguna |
| *Radio\_r1* | Radio button | lk | Text untuk menginputkan pilihan jenis kelamin laki-laki pengguna |
| *Radio\_r2* | Radio button | Pr | Text untuk menginputkan pilihan jenis kelamin perempuan pengguna |
| *Input\_r3* | Text | Alamat | Text untuk menginputkan alamat pengguna |
| *Input\_r4* | Text | Pekerjaan | Text untuk menginputkan pekerjaan pengguna |
| *Input\_r5* | Text | Username | Text untuk menginputkan username pengguna |
| *Input\_r6* | Text | Password | Text untuk menginputkan password pengguna |
| *Btn\_r* | *Button* | *Submit* | Tombol untuk mensubmitkan data pelanggan yang telah diisi |

*Antarmuka : Jenis Produk*



Gambar 3. Interface halaman jenis produk

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Img\_jp1* | *Image* | *Profil* | Menampilkan gambar profil |
| *Btn\_jp1* | Button | Produk | Tombol untuk menyambungkan ke halaman jenis produk |
| *Btn\_jp2* | *Button* | *Pemesanan* | Tombol untuk menyambungkan ke halaman jenis pemesanan |
| *Btn\_jp3* | *Button* | *Pembelian* | Tombol untuk menyambungkan ke halaman jenis pembelian |
| *Link\_jp1* | *Link* | *Desain perusahaan* | Link untuk menuju halaman desain perusahaan |
| *Link\_jp2* | *Link* | *Desain konsumen* | Link untuk menuju halaman desain konsumen |

*Antarmuka : Desain Perusahaan*



Gambar 4. Interface halaman desain produk perusahaan

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Img\_jp1* | *Image* | *Profil* | Menampilkan gambar profil |
| *Btn\_jp1* | Button | Produk | Tombol untuk menyambungkan ke halaman jenis produk |
| *Btn\_jp2* | *Button* | *Pemesanan* | Tombol untuk menyambungkan ke halaman jenis pemesanan |
| *Btn\_jp3* | *Button* | *Pembelian* | Tombol untuk menyambungkan ke halaman jenis pembelian |
| *Link\_jp1* | *Link* | *Desain perusahaan* | Link untuk menuju halaman desain perusahaan |
| *Link\_jp2* | *Link* | *Desain konsumen* | Link untuk menuju halaman desain konsumen |

*Antarmuka : Desain Konsumen*



Gambar 5. Interface halaman desain konsumen

| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| --- | --- | --- | --- |
| *Img\_jp1* | *Image* | *Profil* | Menampilkan gambar profil |
| *Btn\_jp1* | Button | Produk | Tombol untuk menyambungkan ke halaman jenis produk |
| *Btn\_jp2* | *Button* | *Pemesanan* | Tombol untuk menyambungkan ke halaman jenis pemesanan |
| *Btn\_jp3* | *Button* | *Pembelian* | Tombol untuk menyambungkan ke halaman jenis pembelian |
| *Link\_jp1* | *Link* | *Desain perusahaan* | Link untuk menuju halaman desain perusahaan |
| *Link\_jp2* | *Link* | *Desain konsumen* | Link untuk menuju halaman desain konsumen |

## Perancangan Representasi Persistensi Kelas

|  |  |
| --- | --- |
| **Entity Class** | **Basis Data** |
| Pelanggan | Pelanggan |
| Produk | Produk |
| Sparepart | Sparepart |
| Admin | Admin |
| Staff Bengkel | Staff Bengkel |

# Matriks Kerunutan

|  |  |
| --- | --- |
| **Kelas** | **Use Case Terkait** |
| Login\_view | Login |
| Login\_controller | Login |
| Registrasi\_view | Registrasi |
| Registrasi\_controller | Registrasi, pengelolaan data pelanggan |
| DataPemesanan\_view | Pengelolaan data pemesanan |
| DataPemesanan\_controller | Pengelolaan data pemesanan |
| PemesananProduk\_view | Pemesanan |
| PemesananProduk\_controler | Pemesanan |
| BarangSparepart\_view | Cek sparepart |
| BarangSparepart\_controler | Cek sparepart |

# 