**DPPL-xx**

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

<Nama Perangkat Lunak>

untuk:

<Nama User>

Dipersiapkan oleh:

<Nomor Grup & Anggota>

Program Studi Teknik Informatika

FIF – Telkom University

Jl. Telekomunikasi, Bandung 40257

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TU-logo-primer-memusat | **Prodi S1- Teknik Informatika**  **Universitas Telkom** | Nomor Dokumen | | Halaman |
| *DPPL-xx <xx:no grp>* | | *<#>/<jml #* |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

DAFTAR PERUBAHAN

|  |  |
| --- | --- |
| Revisi | Deskripsi |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G |  |

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

Daftar Halaman Perubahan

|  |  |  |  |
| --- | --- | --- | --- |
| Halaman | Revisi | Halaman | Revisi |
|  |  |  |  |

Daftar Isi

1. Pendahuluan 5

1.1 Tujuan Penulisan Dokumen 5

1.2 Lingkup Masalah 5

1.3 Definisi dan Istilah 5

1.4 Aturan Penamaan dan Penomoran 5

1.5 Referensi 5

1.6 Ikhtisar Dokumen 5

2 Deskripsi Perancangan Global 6

2.1 Rancangan Lingkungan Implementasi 6

2.2 Deskripsi Arsitektural 6

2.3 Deskripsi Komponen 6

3 Perancangan Rinci 7

3.1 Realisasi Use Case 7

3.1.1 Use Case <nama use case 1> 7

3.1.1.1 Identifikasi Kelas 7

3.1.1.2 Sequence Diagram 7

3.1.1.3 Diagram Kelas 7

3.2 Perancangan Detil Kelas 7

3.2.1 Kelas <nama kelas> 7

3.2.2 Kelas <nama kelas> 8

3.3 Diagram Kelas Keseluruhan 8

3.4 Algoritma/Query 8

3.5 Diagram Statechart 8

3.6 Perancangan Antarmuka 8

3.7 Perancangan Representasi Persistensi Kelas 9

4 Matriks Kerunutan 9

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini dijadikan sebagai acuan dalam pembuatan perangkat lunak. Dokumen ini akan digunakan oleh pengembang perangkat lunak. Tujuan dari proyek ini adalah untuk mendeskripsikan dan memberi gambaran bagaimana sebuah Sistem Penjualan Produk Otomotif (SPPO) dikerjakan secara bertahap. Proyek ini dikatakan berhasil, jika proyek ini dapat menangani setiap kebutuhan pengguna.

## Lingkup Masalah

Sistem Penjualan Produk Otomotif (SPPO) ini merupakan sistem penjualan produk otomotif berbasis web. Sistem ini dibuat untuk membantu pelanggan dalam melakukan pembelian produk otomotif. Seringkali dengan waktu yang sangat sibuk, beberapa orang tidak punya waktu untuk datang ke *dealer* dalam membeli produk otomotif. Maka dengan adanya masalah tersebut, dibuatkanlah suatu sistem yang dapat memudahkan pelanggan dalam melakukan pembelian produk otomotif.

## Definisi dan Istilah

### Pelanggan

Pelanggan adalah orang atau organisasi yang memesan dan membayar produk, dan biasanya (tidak harus) ia yang akan memutuskan kebutuhannya. Dalam sistem ini, pelanggan memesan produk otomotif.

### Perusahaan

Perusahaan adalah orang atau organisasi yang menghasilkan produk untuk pelanggan. Dalam system ini, perusahaan menghasilkan produk otomotif.

### Supplier

*Supplier* adalah orang atau organisasi yang menghasilkan bagian bagian yang melengkapi produk. Dalam sistem ini, *Supplier* menghasilkan *sparepart* untuk produk otomotif.

|  |  |
| --- | --- |
| **Kata Kunci atau Frase** | **Definisi dan atau Akronim** |
| Server | Komputer atau program komputer yang mengelola akses ke sumber daya atau layanan terpusat dalam suatu jaringan. |
| Database | Satu set data terstruktur yang disimpan di komputer, terutama yang dapat diakses dengan berbagai cara. |
| MySQL | Data base yang di gunakan dalam web dan sebagai penyimpan data yang sudah di terima oleh sistem |
| Apache | Berfungsi sebagai server pada Sistem Penjualan Produk Otomotif (SPPO) ini |
| Sequence Diagram | Diagram yang menggambarkan interaksi antara actor dan system dalam suatu scenario *use case* |
| SKPL | Spesifikasi Kebutuhan Perangkat Lunak Dokumen hasil analisis yang berisi spesifikasi kebutuhan user |
| Class Diagram | Model statis yang menggambarkan struktur dan deskripsi class serta hubungannya antara class |
| Component Diagram | Diagram yang mendeskripsikan *physical* struktur dari sebuah system dan digunakan untuk menggambarkan bagaimana code program dibagi menjadi beberapa komponen serta menggambarkan hubungan antar komponen tersebut. |
| Deployment Diagram | Diagram yang menggambarkan arsitektur *hardware* dan *software* sebuah system, menjelaskan komponen, *processor*, dan *device* yang membangun system. |
| MVC Pattern | Sebuah pendekatan untuk menstruktur *software* agar code program bersifat *maintainable* dan *flexible*. |

## 

## Aturan Penamaan dan Penomoran

| **Hal/Bagian** | **Aturan Penomoran/Penamaan** |
| --- | --- |
| Komponen Fungsional | DPPL-Fxxxx |
| Komponen Non Fungsional | DPPL-NFxxx |
| Use Case | UC-xxx |
| Use Case Realization dari Use Case xx | UCR-xxx-yy |
| Class Diagram | CSD-xxx |
| Skenario Normal Use Case | SC-N-xx |
| Skenario Alternatif Use Case | SC-A-xx |
| Class Diagram | CLS-xxx |

## Referensi

1. IEEE Std 830-1993, IEEE Recommended Practice for Software Requirement Specifications.
2. IEEE Std 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology (ANSI).
3. Jurusan Teknik Informatika – Institut Teknologi Bandung Panduan GL01, Panduan Penggunaan dan Pengisian Spesifikasi Kebutuhan Perangkat Lunak.
4. SKPL Sistem Penjualan dan Pembelian Produk Otomotif

## Ikhtisar Dokumen

Dokumen Deskripsi Perancangan Perangkat lunak terdiri dari beberapa sebagai berikut :

1. Pendahuluan

Pada pendahuluan diberikan gambaran umum tentang dokumen yang berisikan :

* 1. Tujuan Penulisan Dokumen
  2. Lingkup Masalah
  3. Definisi dan Istilah
  4. Aturan Penamaan dan Penomoran
  5. Referensi
  6. Ikhtisar Dokumen

1. Deskripsi Perancangan Global

Pada bagian ini mendeskripsikan perancangan perangkat lunak secara global yang diperlukan di dalam pengembangan Sistem Penjualan Produk Otomotif (SPPO) yang berisikan :

* 1. Rancangan Lingkungan Implementasi
  2. Deskripsi Arsitektural
  3. Deskripsi Komponen

1. Perancangan Rinci

Pada bagian ini mendeskripsikan perancangan perangkat lunak secara rinci yang diperlukan di dalam pengembangan Sistem Penjualan Produk Otomotif (SPPO) yang berisikan :

* 1. Realisasi Use Case
  2. Perancangan Detil Kelas
  3. Diagram Kelas Secara Keseluruhan
  4. Algoritma / Query
  5. Perancangan Antarmuka

1. Matriks Kerunutan

Pada bagian ini dilakukan *mapping* use case dengan class class terkait.

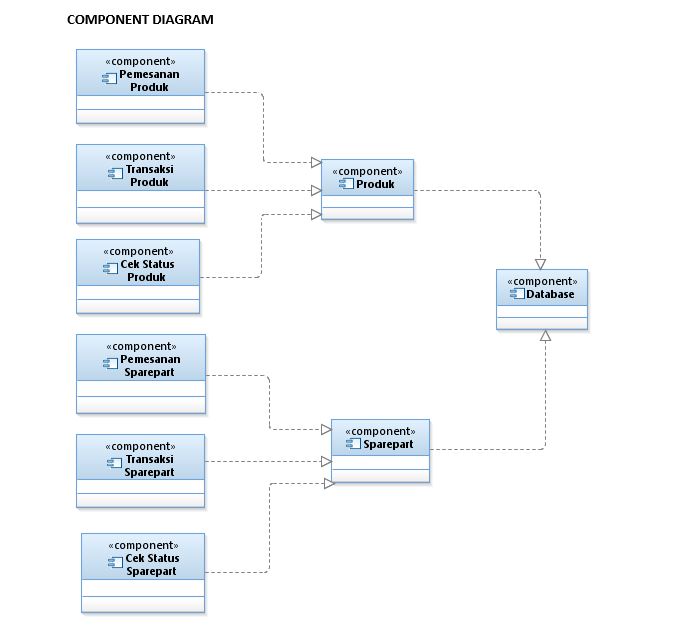
***2. Deskripsi Perancangan Global***

## 2.1. Rancangan Lingkungan Implementasi

|  |  |
| --- | --- |
| System Operasi | Windows |
| DBMS | MySQL |
| Bahasa Pemrograman | PHP |
| Development Tools | Notepad++ |

**Tabel 2.1** Rancangan Lingkungan Implementasi

## Deskripsi Arsitektural



**Gambar 2.1.** Diagram Komponen

## Deskripsi Komponen

| **DPPL - ID** | **Nama Komponen** | **Keterangan** |
| --- | --- | --- |
| DPPL-F001 | Registrasi Pemesanan | Modul ini hanya bisa dilakukan oleh pelanggan untuk melakukan pemesanan produk |
| DPPL-F002 | Transaksi Produk | Modul ini hanya bisa dilakukan oleh pelanggan untuk menyelesaikan proses pemesanan produk |
| DPPL-F003 | Input Data Pemesanan Produk | Modul ini hanya bisa dilakukan oleh perusahaan untuk melakukan pendataan pemesanan produk |
| DPPL-F004 | Pemesanan Sparepart | Modul ini hanya bisa dilakukan oleh perusahaan untuk melakukan pemesanan sparepart |
| DPPL-F005 | Transaksi Sparepart | Modul ini hanya bisa dilakukan oleh perusahaan untuk menyelesaikan proses pemesanan sparepart |
| DPPL-F006 | Input Data Pemesanan Sparepart | Modul ini hanya bisa dilakukan oleh supplier untuk melakukan pendataan pemesanan sparepart |
| DPPL-F007 | Cek Status Pengiriman Sparepart | Modul ini hanya bisa dilakukan oleh supplier untuk melakukan pengecekan terkait status pengiriman sparepart untuk perusahaan |
| DPPL-F008 | Cek Status Pengiriman Produk | Modul ini hanya bisa dilakukan oleh perusahaan untuk melakukan pengecekan terkait status pengiriman produk untuk pelanggan |

**Tabel 2.2** Deskripsi Komponen

# Perancangan Rinci

## Realisasi Use Case

### Use Case <nama use case 1>

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

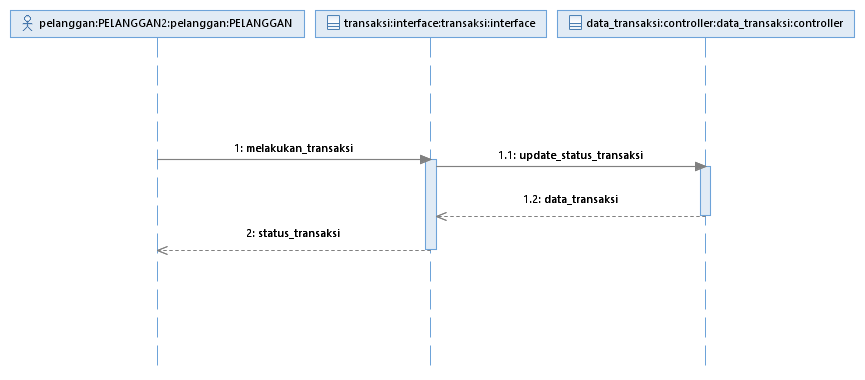
#### Identifikasi Kelas

*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:*

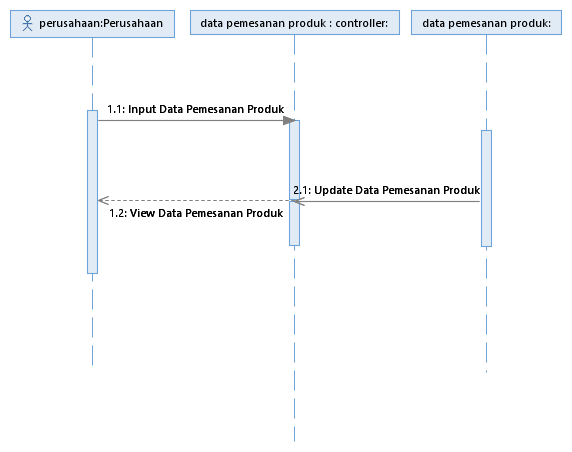
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### Sequence Diagram

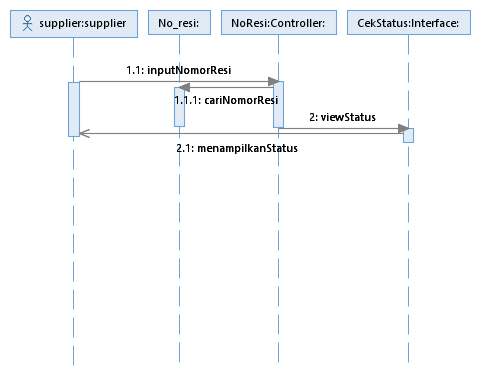
Use Case 1 : Pemesanan Produk



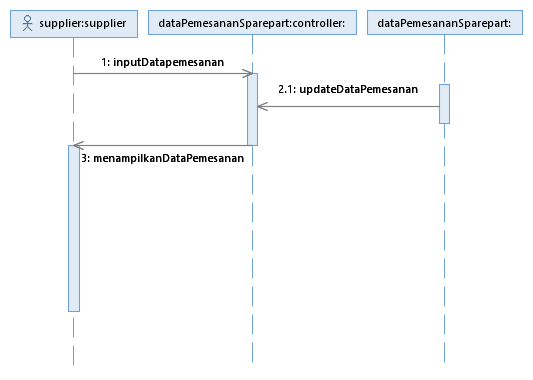
Use Case 2 : Input Data Pemesanan Produk



Use Case 3 : Melakukan Transaksi Produk

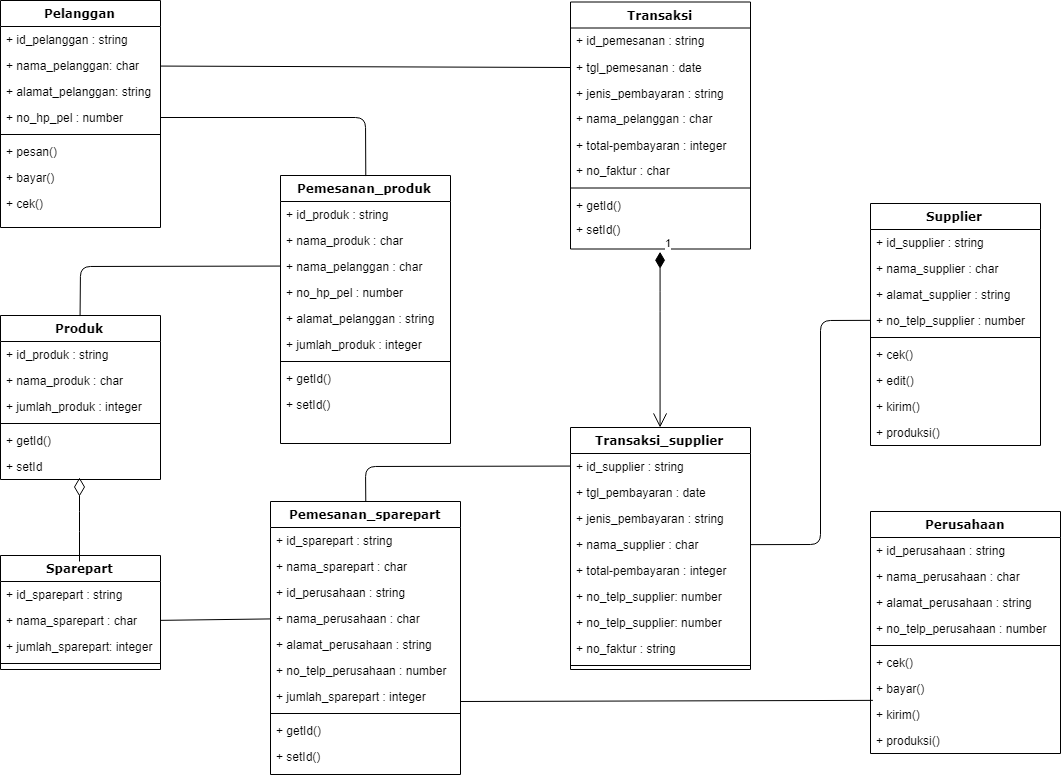


Use Case 4 : Cek Status Pengiriman Sparepart



Use Case 5 : Input Data Pemesanan Sparepart

#### Diagram Kelas (*Class Diagram*)



**Gambar 3.2.** Class Diagram

## Perancangan Detil Kelas

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

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

*Untuk setiap kelas:*

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

### Kelas <nama kelas>

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

*Nama Kelas : ……..*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *Diisi dengan signature operasi* |  |  |
|  |  |  |
|  |  |  |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *Diisi dengan nama atribut* |  | *Tuliskan tipenya sesuai dengan yang dikenal pada bahasa pemrograman yang digunakan* |
|  |  |  |
|  |  |  |

### Kelas <nama kelas>

## 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 :*

*Nama Operasi :*

*Algoritma : (Algo-xxx)*

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

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| *Q-xxx* |  | *Tuliskan fungsi dari querynya* |
|  |  |  |
|  |  |  |

## Diagram Statechart

*Bagian ini hanya diisi jika ada kelas yang kompleks. Perubahan status kelas tersebut harus digambarkan dalam bentuk diagram statechart. Boleh dibuat subba per kelas.*

## Perancangan Antarmuka

*Bagian ini diisi dengan versi awal prototipe antarmuka.*

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

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

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

Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan

## Perancangan Representasi Persistensi Kelas

*Bagian ini diisi dengan rancangan skema basisdata dan traceability-nya terhadap kelas entity.*

# Matriks Kerunutan

Mapping use case dengan kelas-kelas terkait

|  |  |  |  |
| --- | --- | --- | --- |
| Class/Use Case | <<Nama Use Case 1>> | <<Nama Use Case 2>> | <<Nama Use Case 3>> |
| <<Class 1>> | X |  |  |
| <<Class 2>> |  | X | X |
| <<Class 3>> | X | X |  |

# 