

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan analisis, desain dan implementasi perangkat lunak SPKPJP dan pembahasan pada bab-bab sebelumnya, maka kesimpulan yang diperoleh adalah sebagai berikut :

1. Metode *inferensi fuzzy mamdani* dalam memberikan solusi dalam permasalahan penentuan jumlah produksi pada Perusahaan Industri Tekstil King's Tex.
2. Sistem pendukung keputusan penentuan jumlah produksi kain berbasis *desktop* telah berhasil dikembangkan dengan menerapkan metode *inferensi fuzzy Mamdani*.

#### 6.2. Saran

Saran yang dapat diambil dari proses analisis sampai pada pengembangan perangkat lunak SPKPJP adalah : Perangkat lunak SPKPJP masih dapat dikembangkan lagi dalam hal penentuan jumlah produksi dengan menggunakan fungsi keanggotaan lainnya seperti fungsi keanggotaan *trapesium*, *gauss*, dan kombinasi dari fungsi keanggotaan yang ada untuk memperoleh hasil yang lebih optimal.

## DAFTAR PUSTAKA

- Abbasi, Ebrahim., dan Abouec, Amir. 2008. *Stock Price Forecast by Using Neuro-Fuzzy Inference System*. International Journal of Social and Human Sciences 2.
- Ayub, Mewati. 2006. *Alternatif Implementasi Basis Pengetahuan dalam Berbasis Aturan Fuzzy*. Jurnal Informatika UKM, Vol. II, No. 3, Juni 2006. Bandung : Jurusan Teknik Informatika, Universitas Kristen Maranatha.
- Badiru, B. Adedeji., dan Cheung, Y. John. 2002. *Fuzzy Engineering Expert System with Neural Network Applications*. Penerbit John Wiley & Sons. Canada.
- Djunaedi, Much., Setiawan, Eko, Andistas, Fajar Whedi. 2005. *Penentuan Jumlah Produksi Dengan Aplikasi Metode Fuzzy Mamdani*. *Jurnal Ilmiah Teknik Industri* 4: hal. 95-104. Surakarta: Jurusan Teknik Industri Universitas Muhammadiyah Surakarta.
- Eliyani., Pujiyanto, Utomo., dan Rosyadi, Didin. 2009. *Decision Support System untuk Pembelian Mobil menggunakan Fuzzy Database Model Tahani*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI 2009), Yogyakarta, ISSN : 1907-5022.
- Kusumadewi, S. 2003. *Artificial Intelegence Teknik dan Aplikasinya*. Graha Ilmu. Jogjakarta.
- MathWorks. 1995. *Fuzzy Logic Toolbox for Use with MATLAB*. The MathWorks Incorporations.

- Salim, Ja'far., dan Hanifah, Abu Imam. 2006. *Penentuan Jumlah Produksi Optimum berdasarkan Jumlah Mesin dan Tenaga Kerja pada Sistem Produksi Tak Sempurna dengan Aplikasi Metode Fuzzy*. Universitas Sultan Ageng Tirtayasa.
- Subakti, Irfan. 2002. *Sistem Pendukung Keputusan (Decision Support System)*. Institut Teknologi Sepuluh November Surabaya.
- Suprpto, Heri., dan Wulandari, Sri. 2006. *Decission Support System (DSS) dalam Prakuualifikasi Kontraktor. International Civil Engineering Conference "Towards Sustainable Civil Engineering Practice"*. Surabaya.
- Supryono. 2007. *Analisis Perbandingan Logika Fuzzy dengan Regresi Berganda Sebagai Alat Peramalan*. Seminar Nasional III SDM Teknologi Nuklir 2007 ISSN:1978-0176.
- Suryadi, Kadarsah., dan Ali, Ramdhani. 2002. *Sistem Pendukung Keputusan : Suatu Wacana Struktural Idealisasi dan Implementasi Konsep Pengambilan Keputusan*. Bandung : PT.Remaja Rosdakarya.
- Turban, Efraim., dan Arosan, E., Jay. 2000. *Decision Support and Intelligent Systems*. Penerbit Prentice Hall International.
- Wulandari, Fitri. 2005. *Pembuatan Sistem Pendukung Keputusan Berbasis Teori Fuzzy Untuk Mengembangkan Suatu Produk Baru*. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, UIN Suska, Vol. 2, No. 2, 2005 : 62-66. Pekanbaru

Zadeh, L.A. 1994. *Fuzzy Logic, Neural Network, and Soft Computing*. Communications of the ACM, Vol. 14, No.5.



# SKPL

## SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

### SPKPJP

(Pengembangan Sistem Pendukung Keputusan untuk  
Penentuan Jumlah Produksi Kain dengan Metode  
Inferensi Fuzzy Mamdani)

Untuk :

Perusahaan Industri Tekstil King'S Tex

Dipersiapkan oleh:

Gede Putra Dana / 080705616

Program Studi Teknik Informatika - Fakultas Teknologi  
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<b>SKPL- SPKPJP</b>		1/23
		Revisi		

## Daftar Isi

1	Pendahuluan .....	4
1.1	Tujuan .....	4
1.2	Lingkup Masalah .....	4
1.3	Definisi, Akronim dan Singkatan .....	5
1.4	Deskripsi umum (Overview) .....	5
2	Deskripsi Kebutuhan .....	6
2.1	Perspektif produk .....	6
2.2	Fungsi Produk .....	7
2.3	Karakteristik Pengguna .....	9
2.4	Batasan-batasan .....	10
2.5	Asumsi dan Ketergantungan .....	10
3	Kebutuhan khusus .....	10
3.1	Kebutuhan antarmuka eksternal .....	10
3.2	Kebutuhan fungsionalitas Perangkat Lunak .....	12
4	Spesifikasi Rinci Kebutuhan .....	12
4.1	Spesifikasi Kebutuhan Fungsionalitas .....	12
5	Entity Relationship Diagram (ERD) .....	23

## Daftar Gambar

1. Arsitektur Perangkat Lunak SPKPJP.....	9
2. Use Case Diagram .....	13
3. ERD .....	23



## 1. Pendahuluan

### 1.1. Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SPKPJP (Pengembangan Sistem Pendukung Keputusan untuk Penentuan Jumlah Produksi Kain Dengan Metode Inferensi Fuzzy Mamdani) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna) performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SPKPJP ini juga mendefinisikan batasan perancangan perangkat lunak.

### 1.2. Lingkup Masalah

Perangkat Lunak SPKPJP dikembangkan dengan tujuan :

1. Membantu dalam menentukan jumlah produk yang harus diproduksi berdasarkan jumlah permintaan *customer* dan persediaan di gudang pada perusahaan industri tekstil King's Tex.
2. Mengelola data permintaan produk dari *customer*.
3. Memberikan *report* dari permintaan *customer*.
4. Memberikan *report* dari hasil pengolahan permintaan dan penentuan jumlah produksi.
5. Sistem ini berjalan pada lingkungan berbasis dekstop.

Program Studi Teknik Informatika	SKPL – SPKPJP	4/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 1.3. Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-SPKPJP-XXX	Kode yang merepresentasikan kebutuhan pada SPKPJP (Sistem Pendukung Keputusan untuk Penentuan Jumlah Produksi Dengan Metode Inferensi Fuzzy Mamdani) dimana XXX merupakan nomor fungsi produk.
SPKPJP	Pengembangan Sistem pendukung keputusan untuk menentukan jumlah produksi kain berdasarkan permintaan <i>customer</i> dan persediaan produk di gudang.

### 1.4. Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak SPKPJP yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk

Program Studi Teknik Informatika	SKPL – SPKPJP	5/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak SPKPJP tersebut.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak SPKPJP yang akan dikembangkan.

## **2. Deskripsi Kebutuhan**

### **2.1. Perspektif produk**

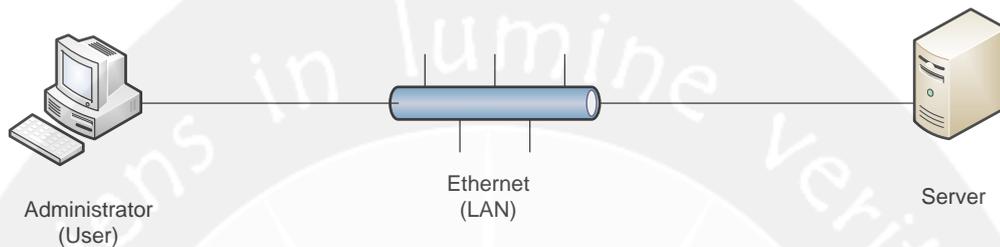
SPKPJP merupakan perangkat lunak yang dikembangkan untuk membantu dalam proses penentuan jumlah produk yang harus di produksi oleh perusahaan industri tekstil King's Tex berdasarkan permintaan *customer* dan persediaan produk di gudang. Sistem ini memudahkan pengelola perusahaan dalam menentukan jumlah produk yang harus diproduksi setiap bulannya dengan memasukkanc jumlah permintaan dan persediaan di gudang.

Perangkat lunak SPKPJP ini berjalan pada platform *Windows* berbasis desktop, dan dibuat menggunakan bahasa pemrograman *Microsoft Visual C#*. Sedangkan untuk lingkungan pemrogramannya menggunakan *Microsoft Visual Studio 2005*.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (*Graphical User Interface*). Pada sistem ini, seperti terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan berupa client server, di mana semua data disimpan di server.

Program Studi Teknik Informatika	SKPL – SPKPJP	6/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

Inputan data yang dimasukkan akan disimpan dalam basisdata server, sehingga jika ada pencarian data, maka data yang diinginkan akan dicari ke basisdata server yang selanjutnya dikirimkan ke client yang merequest melalui server.



**Gambar 1. Arsitektur Perangkat lunak SPKPJP**

## 2.2. Fungsi Produk

Fungsi produk perangkat lunak SPKPJP adalah sebagai berikut :

### 1. Fungsi *Pengelolaan Data Customer* (**SKPL-SPKPJP-001**)

Fungsi pengelolaan data *customer* merupakan fungsi yang digunakan oleh administrator untuk mengelola data *customer* pada perusahaan industri tekstil King's Tex.

Fungsi *Pengelolaan Data Customer* meliputi :

- a. Fungsi *Add Customer* (**SKPL-SPKPJP-001-01**).  
Fungsi *add customer* merupakan fungsi yang digunakan untuk menambahkan data *Customer* baru ke dalam basis data.
- b. Fungsi *Edit Customer* (**SKPL-SPKPJP-001-02**).  
Fungsi *edit customer* merupakan fungsi yang digunakan untuk mengubah data *Customer* yang ada di basis data.

c. Fungsi *Search Customer* (**SKPL-SPKPJP-001-03**)

Fungsi *search customer* merupakan fungsi yang digunakan untuk mencari data *customer* yang ada di basis data berdasarkan masukkan pengguna.

d. Fungsi *Delete Customer* (**SKPL-SPKPJP-001-03**)

Fungsi *delete customer* merupakan fungsi yang digunakan untuk menghapus data *customer* yang ada di basis data.

2. Fungsi *Pengelolaan Permintaan Produk* (**SKPL-SPKPJP-002**)

Fungsi *pengelolaan permintaan produk* merupakan adalah fungsi yang digunakan oleh administrator untuk mengelola data *permintaan produk customer*.

Fungsi *Pengelolaan Permintaan Customer* meliputi :

a. Fungsi *Add Permintaan Customer* (**SKPL-SPKPJP-002-01**) .

Fungsi *add permintaan customer* merupakan fungsi yang digunakan untuk menambahkan data *permintaan produk* dari *customer* setiap bulannya ke dalam basis data.

b. Fungsi *Edit Permintaan Customer* (**SKPL-SPKPJP-002-02**) .

Fungsi *edit permintaan customer* merupakan fungsi yang digunakan untuk mengubah data *permintaan customer* yang ada di basis data.

c. Fungsi *Delete Permintaan Customer* (**SKPL-SPKPJP-002-03**)

Fungsi *delete permintaan customer* merupakan fungsi yang digunakan untuk menghapus data permintaan *customer* yang ada di basis data.

3. Fungsi Penentuan Jumlah Produksi (**SKPL-SPKPJP-003**)

Fungsi *penentuan jumlah produksi* merupakan fungsi yang digunakan untuk menentukan jumlah produksi setiap bulannya berdasarkan masukkan total permintaan *customer* dan persediaan produk di gudang sebelum proses produksi dilakukan.

4. Fungsi Report Permintaan Customer (**SKPL-SPKPJP-004**)

Fungsi *report permintaan customer* merupakan fungsi yang digunakan untuk menerbitkan laporan dari seluruh permintaan *customer* terhadap masing-masing jenis produk berdasarkan parameter bulan transaksi dilakukan.

5. Fungsi Report History Produksi (**SKPL-SPKPJP-005**)

Fungsi *report history produksi* merupakan fungsi yang digunakan untuk menerbitkan laporan dari *history* produksi untuk masing-masing jenis produk setiap bulannya.

### 2.3. Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak SPKPJP adalah sebagai berikut :

1. Memahami pengoperasian PC.
2. Memahami pengoperasian sistem yang digunakan.

Program Studi Teknik Informatika	SKPL – SPKPJP	9/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## **2.4. Batasan-batasan**

Batasan-batasan dalam pengembangan perangkat lunak SIPS tersebut adalah :

1. Kebijakan Umum  
Berpedoman pada tujuan dari pengembangan perangkat lunak SPKPJP.
2. Keterbatasan perangkat keras  
Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

## **2.5. Asumsi dan Ketergantungan**

Sistem ini dapat dijalankan pada perangkat PC yang menggunakan sistem operasi Windows XP dan Windows 7.

## **3. Kebutuhan khusus**

### **3.1. Kebutuhan antarmuka eksternal**

Kebutuhan antarmuka eksternal pada perangkat lunak SPKPJP meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak.

#### **3.1.1. Antarmuka pemakai**

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk *form-form*.

#### **3.1.2. Antarmuka perangkat keras**

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SPKPJP adalah:

1. Perangkat PC (desktop).
2. Keyboard.
3. Mouse.

Program Studi Teknik Informatika	SKPL – SPKPJP	10/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

### 3.1.3. Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak SPKPJP adalah sebagai berikut :

1. Nama : SQL Server 2005  
Sumber : Microsoft

Sebagai *database management system* (DBMS) yang digunakan untuk penyimpanan data di sisi server.

2. Nama : Windows XP atau Windows 7  
Sumber : Microsoft.

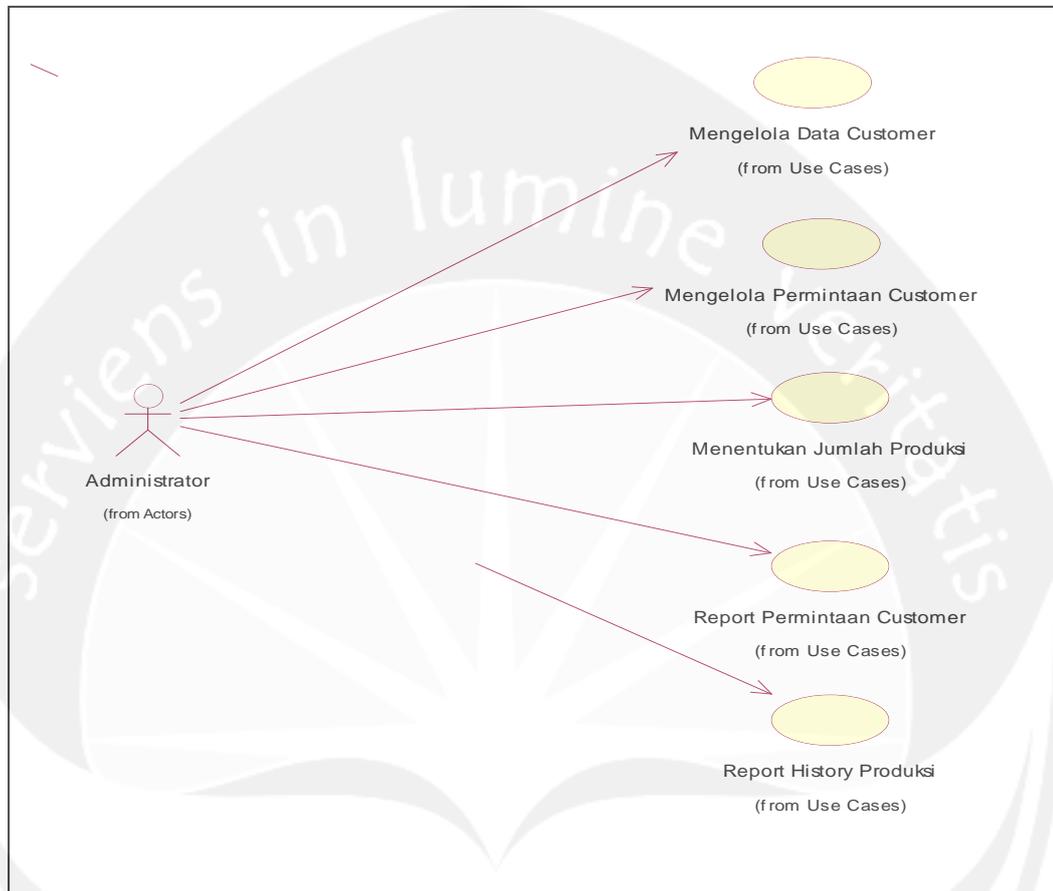
Sebagai sistem operasi untuk perangkat komputer.

3. Nama : Microsoft Visual Studio 2005  
Sumber : Microsoft.

Sebagai IDE (Integrated Development Environment).

## 3.2. Kebutuhan fungsionalitas Perangkat Lunak

### 3.2.1. Use Case Diagram



Gambar 2. Use Case Diagram SPKPJP

## 4. Spesifikasi Rinci Kebutuhan

### 4.1. Spesifikasi Kebutuhan Fungsionalitas

#### 4.1.1 Use case Specification : Pengelolaan Data Customer

##### 1. Brief Description

*Use case ini digunakan oleh aktor untuk mengelola data customer. Aktor dapat melakukan Add data customer, edit data customer, dan search data customer.*

Program Studi Teknik Informatika	SKPL – SPKPJP	12/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## 2. Primary Actor

1. Administrator

## 3. Supporting Actor

none

## 4. Basic Flow

1. *Use case* ini dimulai ketika aktor memilih untuk melakukan pengelolaan data *customer*.
2. Sistem memberikan pilihan untuk melakukan *add data customer*, *edit data customer*, *delete data customer*, atau *search data customer*.
3. Aktor memilih untuk melakukan *add data customer*.
4. Aktor memasukkan data *customer* melalui *textbox* yang telah tersedia, meliputi pengisian nama *customer*, alamat, kodepos, telepon, dan *contact person*.
5. Aktor meminta sistem untuk menyimpan data *customer* yang telah dimasukkan.
6. Sistem mengecek data *customer* yang telah dimasukkan.
7. Sistem menyimpan data *customer* ke basis data.
8. *Use case* ini selesai.

## 5. Alternative Flow

A-1 Aktor memilih untuk melakukan *edit data customer* (Setelah Basic Flow langkah ke 2)

1. Aktor memilih untuk melakukan *edit data customer*.
2. Sistem menampilkan data *customer* yang ada pada basis data melalui *data grid view*.

Program Studi Teknik Informatika	SKPL – SPKPJP	13/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Aktor mengubah data *customer* yang telah ditampilkan dengan menekan data *customer* yang ditampilkan pada *data grid view*.
4. Aktor meminta sistem untuk menyimpan data *customer* yang telah diubah.
5. Sistem melakukan pengecekan terhadap data *customer* yang telah diubah.
6. Sistem menyimpan data *customer* yang telah diubah ke basis data.
7. Berlanjut ke Basic Flow langkah ke 8.

A-2 Aktor memilih untuk melakukan *delete data customer*(Setelah Basic Flow langkah ke 2)

1. Aktor memilih untuk melakukan *delete data customer*.
2. Sistem menampilkan data *customer* yang ada pada basis data melalui *data grid view*.
3. Aktor menekan data *customer* pada *data grid view*.
4. Sistem menghapus data *customer* yang di pilih.
5. Berlanjut ke Basic Flow langkah ke 8.

A-3 Aktor memilih untuk melakukan *search data customer*(Setelah Basic Flow Langkah ke 2)

1. Aktor memilih untuk melakukan *search data customer*.
2. Sistem menampilkan *field* untuk memasukan data *customer* yang akan di cari.

3. Aktor memasukkan data *customer* yang dicari pada textbox yang disediakan, kemudian menekan *button* "search".
4. Sistem melakukan pengecekan di basis data untuk mencari nama *customer* yang dimasukkan oleh aktor
5. Sistem menampilkan hasil dari pencarian di dalam basis data.
6. Berlanjut ke Basic Flow langkah ke 8.

#### **6. Error Flow**

E-1 Data *customer* yang dimasukkan sudah ada pada basis data (Setelah Basic Flow langkah ke 6)

1. Sistem memberikan pesan peringatan bahwa "data yang diinputkan sudah ada di basisdata!".
2. Kembali ke Basic Flow Langkah ke 4

E-2 Data *customer* yang diubah oleh aktor salah pada *field* telepon bukan berupa angka (Setelah Alternative Flow 1 langkah ke 5)

1. Sistem memberikan pesan peringatan bahwa "data telepon harus berupa angka !".
2. Kembali ke Alternative Flow A-1 langkah ke 2.

#### **7. PreConditions**

1. Aktor telah memasuki sistem.

Program Studi Teknik Informatika	SKPL – SPKPJP	15/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## 8. PostConditions

1. Data *customer* di basis data telah diperbaharui.
2. Data *customer* berhasil ditampilkan.

### 4.1.2 Use case Spesification : Pengelolaan Permintaan Produk

#### 1. Brief Description

*Use case Pengelolaan Permintaan Produk* digunakan oleh aktor untuk mengelola permintaan produk dari *customer*. Pada *use case* ini aktor dapat melakukan *Add Permintaan Customer*, *Edit Permintaan Customer*, dan *Delete Permintaan Customer*.

#### 2. Primary Actor

1. Administrator

#### 3. Supporting Actor

none

#### 4. Basic Flow

1. Use case ini dimulai ketika aktor akan melakukan pengelolaan permintaan produk.
2. Sistem menampilkan form antarmuka untuk pengelolaan permintaan produk.
3. Aktor memasukkan dan memilih data untuk diproses sebagai transaksi permintaan *customer* yang terdiri dari memilih nama *customer* melalui *combobox* "nama customer", kemudian memilih salah satu dari 3 jenis produk yang akan diproses, menentukan tanggal permintaan, kemudian memasukkan jumlah permintaan produk kain tersebut.

Program Studi Teknik Informatika	SKPL – SPKPJP	16/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

4. Aktor meminta sistem untuk melakukan proses transaksi permintaan *customer* dan menyimpannya pada basis data dengan menekan *button* "proses".
5. Sistem memeriksa *field* yang berisi data yang dimasukkan Aktor.
6. Sistem menyimpan data permintaan *customer* ke basis data.
7. *Use case* ini selesai.

### 5. Alternative Flow

A-1 Aktor memilih untuk melakukan *edit Permintaan customer* (Setelah Basic Flow langkah ke 2)

1. Aktor memilih untuk melakukan *edit permintaan customer*.
2. Sistem menampilkan permintaan *customer* yang ada pada basis data melalui *data grid view* "data permintaan *customer*".
3. Aktor mengubah permintaan *customer* yang telah ditampilkan dengan menekan data permintaan *customer* yang ditampilkan pada *data grid view*.
4. Aktor meminta sistem untuk menyimpan data permintaan *customer* yang telah diubah.
5. Sistem melakukan pengecekan terhadap data permintaan *customer* yang telah diubah.
6. Sistem menyimpan data permintaan *customer* yang telah diubah ke dalam basis data.
7. Berlanjut ke Basic Flow langkah ke 7.

A-2 Aktor memilih untuk melakukan *delete permintaan customer*(Setelah Basic Flow langkah ke 2)

1. Aktor memilih untuk melakukan *delete permintaan customer*.
2. Sistem menampilkan data permintaan *customer* yang ada di dalam basis data melalui *data grid view*.
3. Aktor menekan data permintaan *customer* pada *data grid view*, kemudian untuk menghapus permintaan tersebut dilakukan dengan menekan *button "Delete"*.
4. Sistem menghapus data permintaan *customer* yang di pilih.
5. Berlanjut ke Basic Flow langkah ke 7.

## 6. Error Flow

E-1 Data permintaan *customer* yang dimasukkan sudah ada di dalam basis data (Setelah Basic Flow langkah ke 5)

1. Sistem memberikan pesan peringatan bahwa "*data permintaan customer yang diinputkan sudah ada pada basis data!*".
2. Kembali ke Basic Flow langkah ke 3.

## 7. PreConditions

1. Aktor telah memasuki sistem.

## 8. PostConditions

1. Aktor selesai mengisi form pengelolaan permintaan *customer*.

### 4.1.3 Use case Spesification : Penentuan Jumlah Produksi

#### 1. Brief Description

Use case ini digunakan untuk menentukan jumlah produksi berdasarkan masukkan total permintaan dan persediaan di gudang.

#### 2. Primary Actor

1. administrator

#### 3. Supporting Actor

none

#### 4. Basic Flow

1. *Use case* ini dimulai ketika aktor akan melakukan penentuan jumlah produksi berdasarkan jumlah permintaan dan persediaan.
2. Sistem menampilkan antarmuka untuk penentuan jumlah produksi.
3. Sistem menampilkan *field* pengisian data untuk menentukan jumlah produksi dan menampilkan total permintaan dari semua *customer*.
4. Aktor memasukkan total persediaan di gudang saat itu.
5. Aktor meminta sistem untuk melakukan pemrosesan terhadap masukkan aktor untuk memperoleh jumlah produk yang harus diproduksi dengan menekan *button* "Proses".
6. Sistem memeriksa *field* yang berisi data yang di pilih Aktor.

Program Studi Teknik Informatika	SKPL – SPKPJP	19/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

7. Sistem memproses inputan aktor dan menampilkan jumlah produk yang harus diproduksi dan menyimpan data tersebut ke dalam basis data.
8. Use case ini selesai.

#### **5. Alternative Flow**

none

#### **6. Error Flow**

none

#### **7. PreConditions**

1. Aktor telah memasuki sistem.

#### **8. PostConditions**

1. Penentuan jumlah produksi selesai dilakukan.

#### **4.1.4 Use case Spesification : Report Permintaan Customer**

##### **1. Brief Description**

*Use case ini digunakan oleh aktor untuk menerbitkan laporan dari permintaan customer untuk setiap bulannya.*

##### **2. Primary Actor**

1. Administrator

##### **3. Supporting Actor**

none

##### **4. Basic Flow**

1. Use case ini dimulai ketika aktor akan membuat laporan mengenai permintaan *customer* terhadap masing-masing jenis produk dalam kurun waktu tertentu.

Program Studi Teknik Informatika	SKPL – SPKPJP	20/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Sistem menampilkan antarmuka *report permintaan customer*.
3. Aktor memilih jenis produk, batas awal dan batas akhir waktu yang diinginkan untuk menerbitkan laporan dari permintaan customer.
4. Aktor meminta sistem untuk melakukan pemrosesan laporan sesuai dengan inputan aktor dengan menekan *button "Generete"*.
5. Sistem memproses laporan yang diinginkan oleh aktor.
6. Use case ini selesai.

#### **5. Alternative Flow**

none

#### **6. Error Flow**

none

#### **7. PreConditions**

1. Aktor telah memasuki sistem.

#### **8. PostConditions**

1. Aktor selesai menerbitkan laporan permintaan *customer*.

### **4.1.5 Use case Spesification : Report History Produksi**

#### **1. Brief Description**

*Use case* ini digunakan oleh aktor untuk menerbitkan laporan dari *history* produksi, yang meliputi total permintaan, persediaan di gudang, dan penentuan jumlah produksi.

Program Studi Teknik Informatika	SKPL – SPKPJP	21/ 23
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

## **2. Primary Actor**

1. Administrator

## **3. Supporting Actor**

none

## **4. Basic Flow**

1. Use case ini dimulai ketika aktor akan menerbitkan laporan mengenai *history* produksi dalam kurun waktu tertentu.
2. Sistem menampilkan antarmuka *report history produksi*.
3. Aktor memilih jenis produk, batas awal dan batas akhir waktu yang diinginkan untuk menerbitkan laporan *history* produksi.
4. Aktor meminta sistem untuk melakukan pemrosesan laporan sesuai dengan inputan aktor dengan menekan *button* "Generete".
5. Sistem memproses laporan yang diinginkan oleh aktor.
6. Use case ini selesai.

## **5. Alternative Flow**

none

## **6. Error Flow**

none

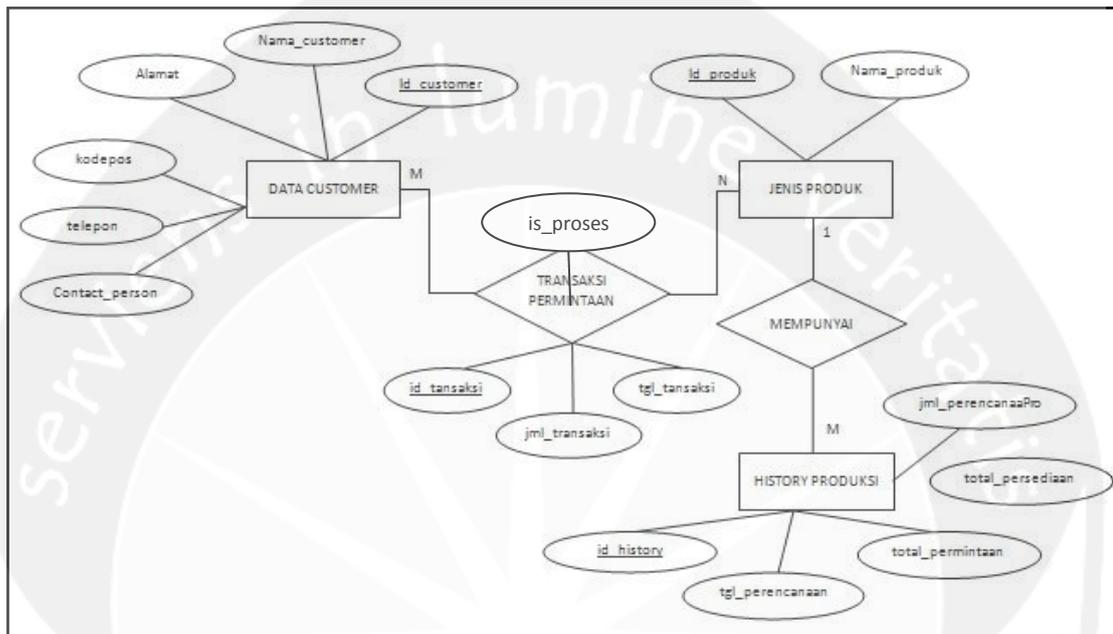
## **7. PreConditions**

1. Aktor telah memasuki sistem.

## 8. PostConditions

1. Aktor selesai menerbitkan laporan history produksi.

## 5. Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram SPKPJP

# DPPL

## DESKRIPSI PERANCANGAN PERANGKAT LUNAK SPKPJP

(Pengembangan Sistem Pendukung Keputusan untuk  
Penentuan Jumlah Produksi Kain dengan Metode  
Inferensi Fuzzy Mamdani)

Untuk :

Perusahaan Industri Tekstil King'S Tex

Dipersiapkan oleh:

Gede Putra Dana / 080705616

Program Studi Teknik Informatika - Fakultas Teknologi  
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika	Nomor Dokumen		Halaman
		<b>DPPL-SPKPJP</b>		1/37
		Revisi		

## 1. Pendahuluan

### 1.1. Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) ini bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen ini akan digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap berikutnya.

### 1.2. Ruang Lingkup

Perangkat Lunak SPKPJP dikembangkan dengan tujuan untuk :

5. Membantu dalam menentukan jumlah produk yang harus diproduksi berdasarkan jumlah permintaan *customer* dan persediaan di gudang pada perusahaan industri tekstil King's Tex.
6. Mengelola data permintaan produk dari *customer*.
7. Menerbitkan *report* dari permintaan *customer* untuk setiap produk.
8. Menerbitkan *report* dari *history* produksi untuk setiap jenis produk.

Perangkat Lunak SPKPJP berjalan pada lingkungan berbasis dekstop.

### 1.3. Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga <i>Software Design Description</i> (SDD). Merupakan deskripsi dari perancangan produk / perangkat lunak yang akan dikembangkan.
SPKPJP	Sistem pendukung keputusan untuk menentukan jumlah produksi kain berdasarkan permintaan <i>customer</i> dan persediaan produk di gudang.

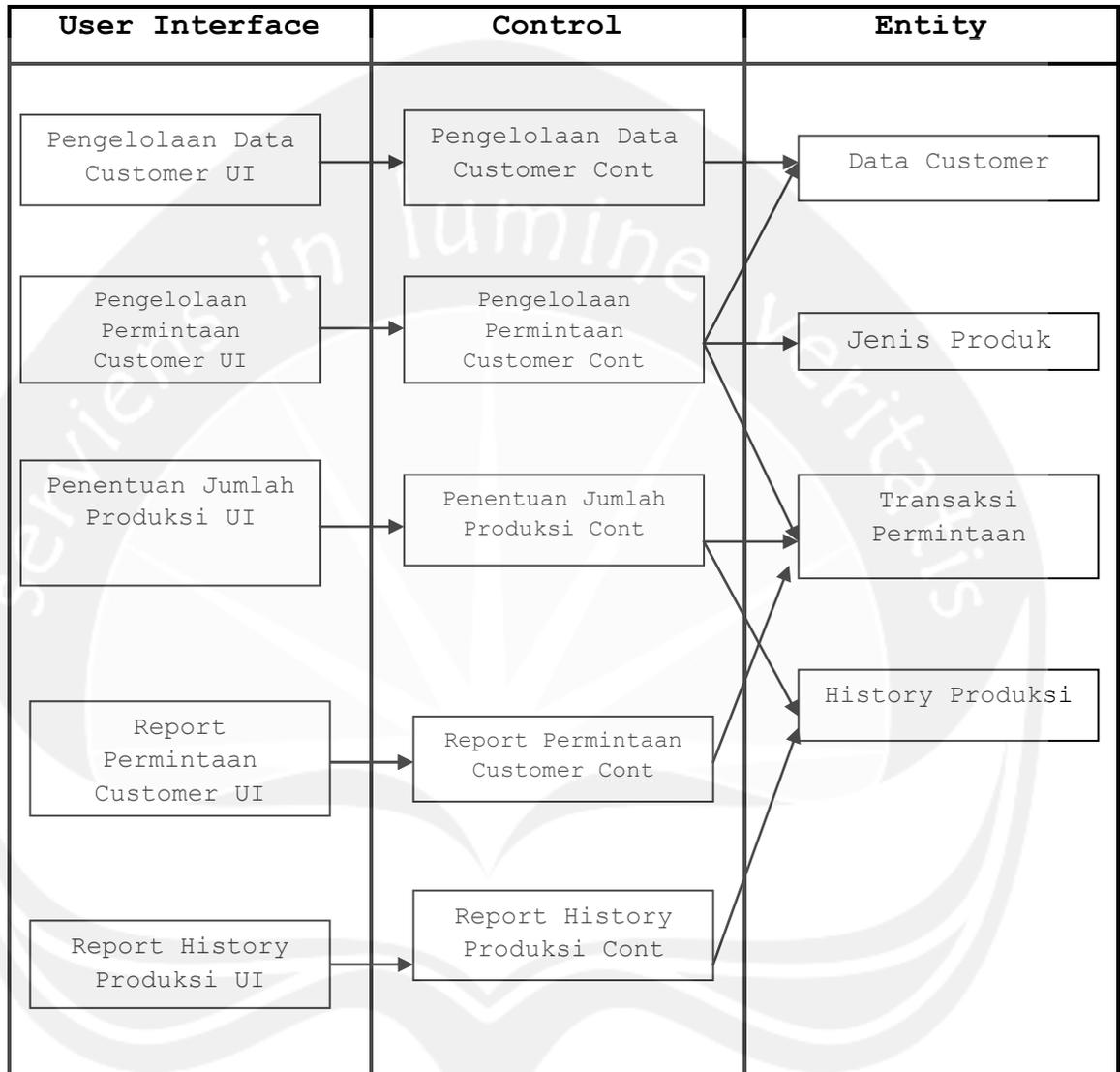
### 1.4. Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. MSDN Library-October 2005, Microsoft, 2005.
2. Dana, Gede, Putra. *Spesifikasi Kebutuhan Perangkat Lunak SPKPJP*. Universitas Atma Jaya Yogyakarta. 2011. Yogyakarta.

## 2. Perancangan Sistem

### 2.1. Perancangan Arsitektur

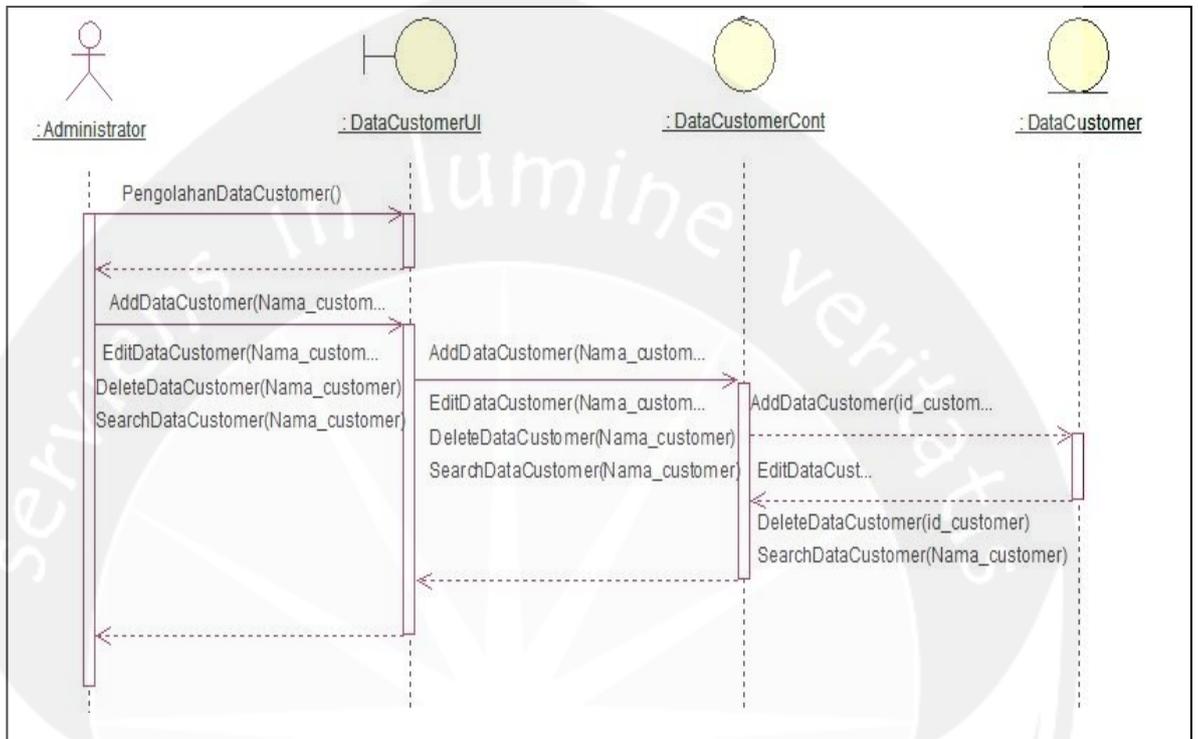


Gambar 2.1 Rancangan Arsitektur SPKPJP

## 2.2. Perancangan Rinci

### 2.2.1. Sequence Diagram

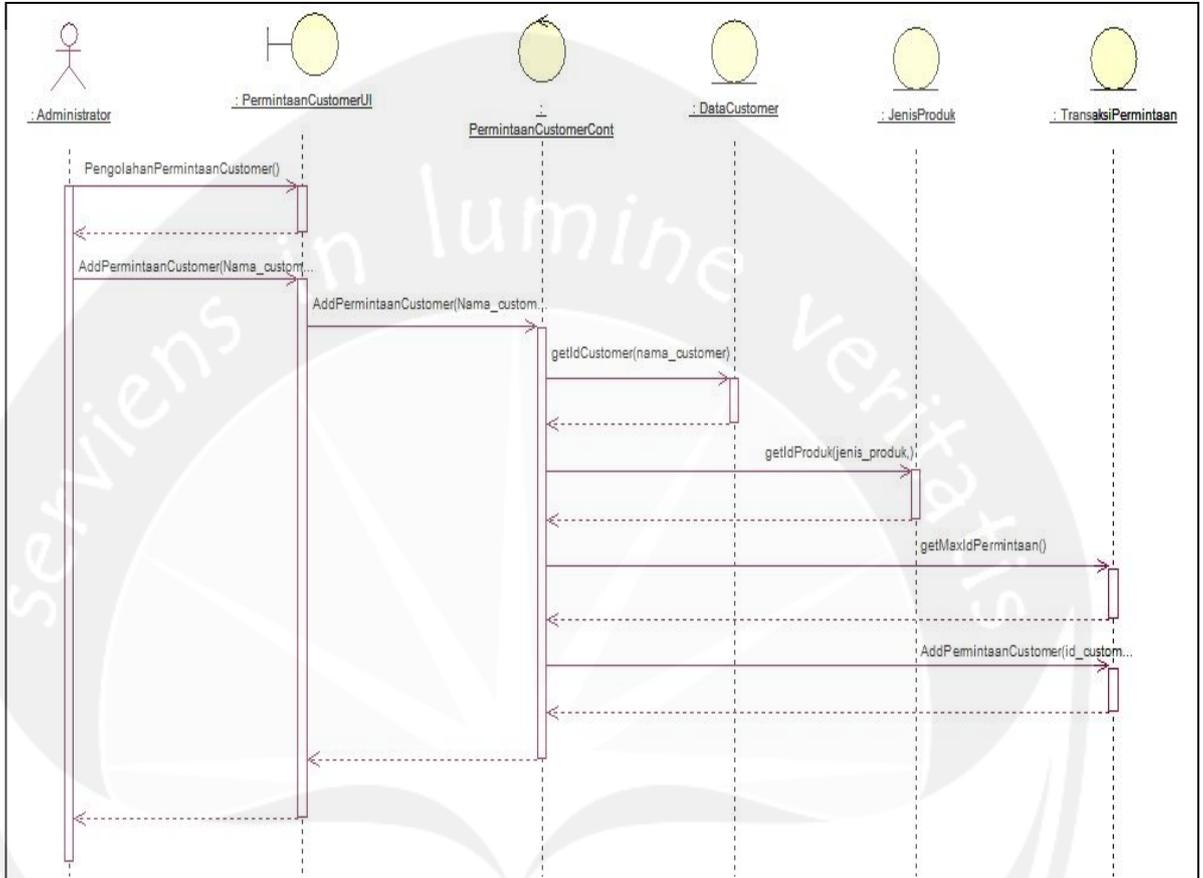
#### 2.2.1.1. Pengelolaan Data Customer



Gambar 2.2 Sequence Diagram : Pengelolaan Data Customer

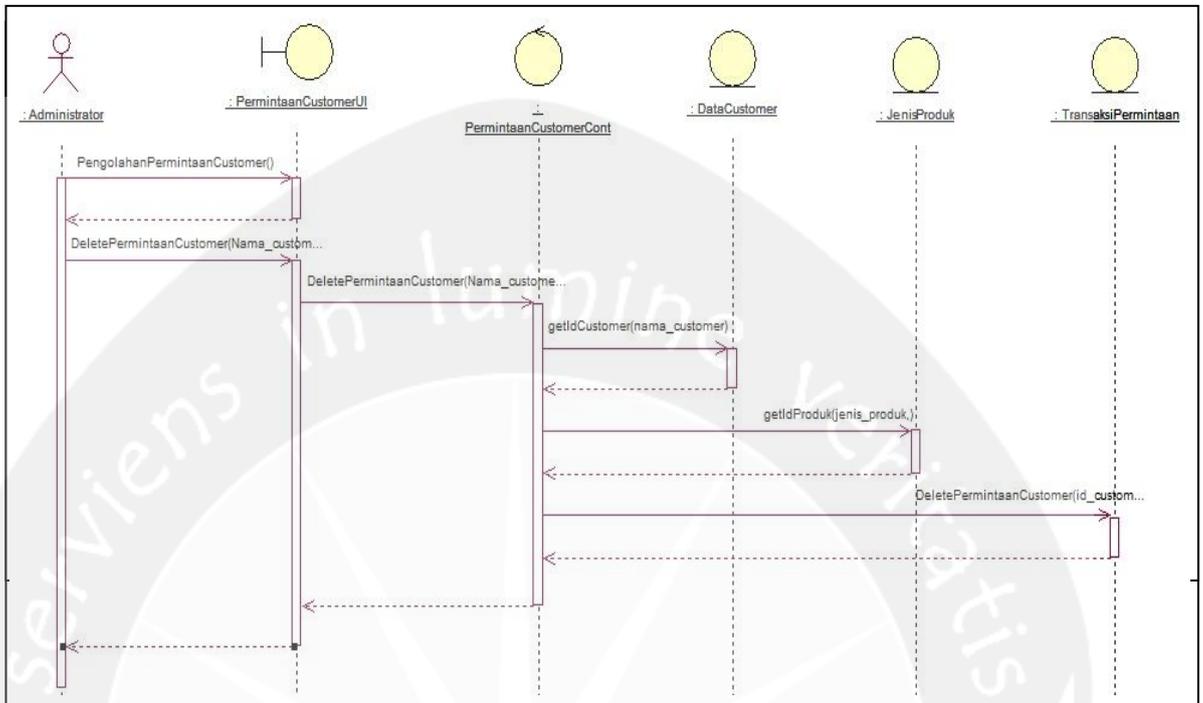
## 2.2.1.2. Pengelolaan Permintaan Customer

### 2.2.1.2.1. Add Permintaan Customer



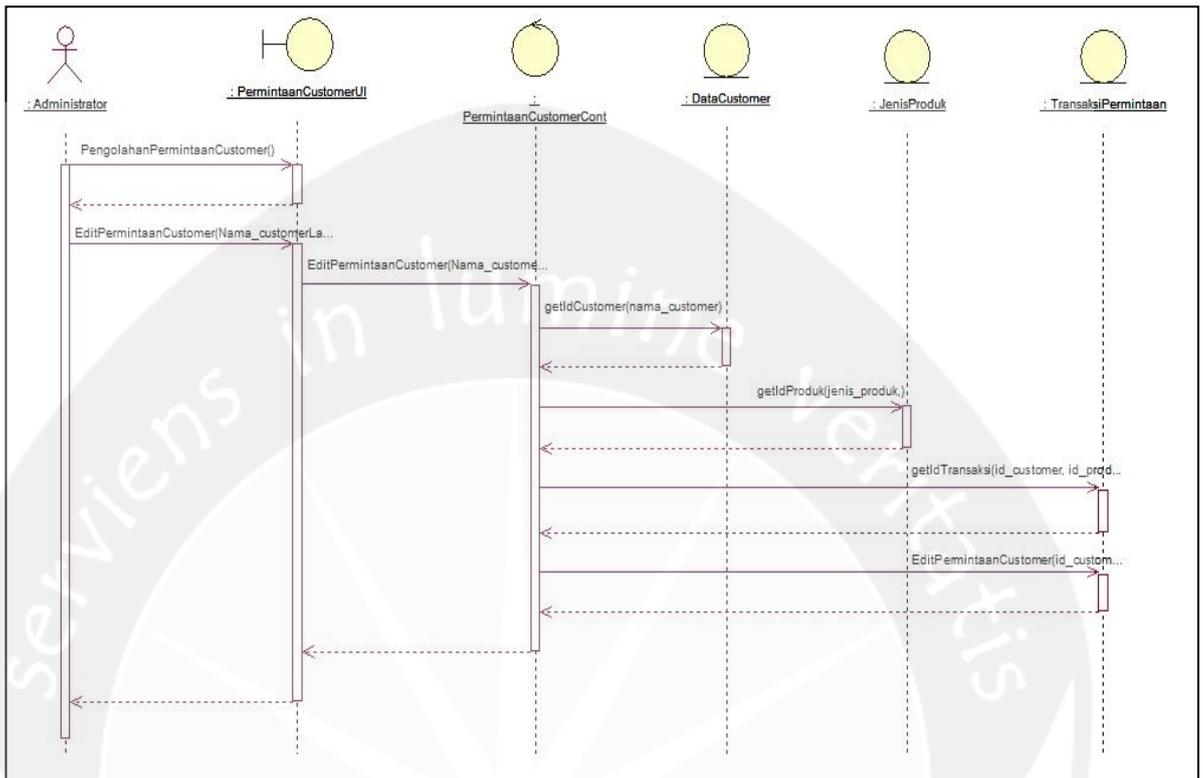
Gambar 2.3 Sequence Diagram : Add Permintaan Customer

### 2.2.1.2.2. Delete Permintaan Customer



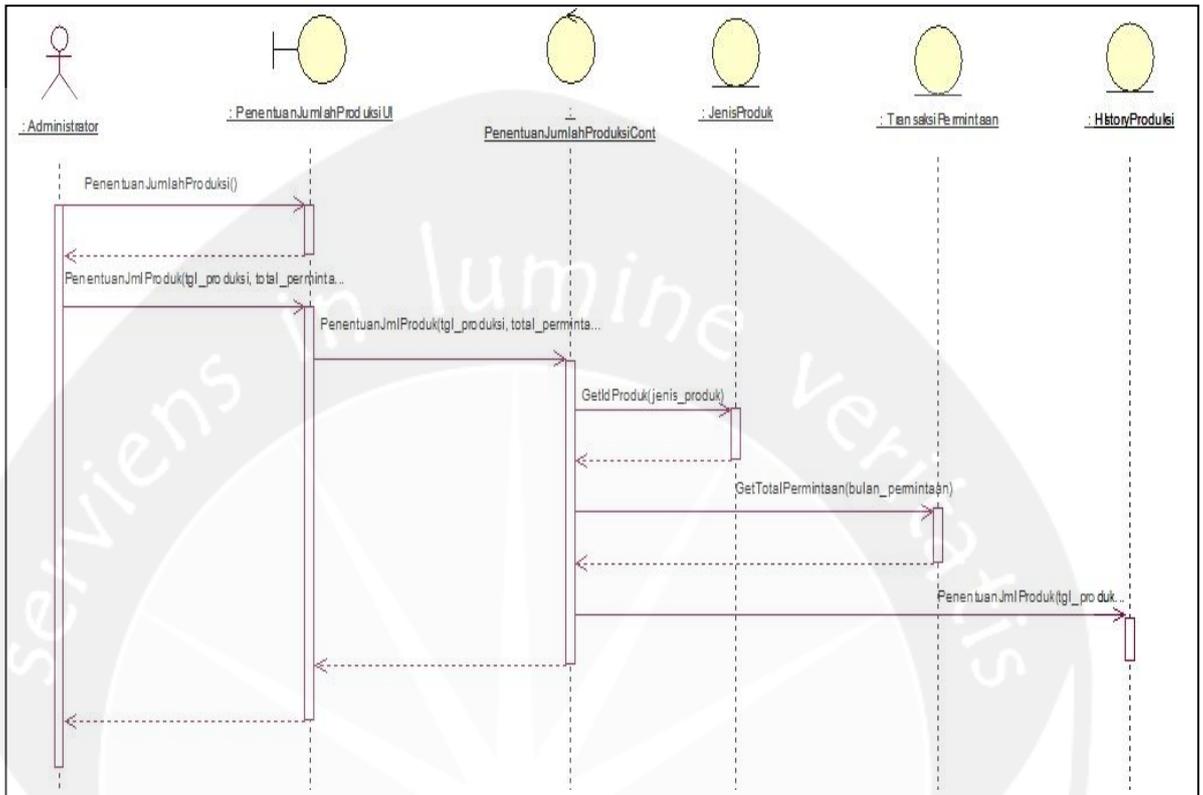
Gambar 2.4 Sequence Diagram : Delete Permintaan Customer

### 2.2.1.2.3. Edit Permintaan Customer



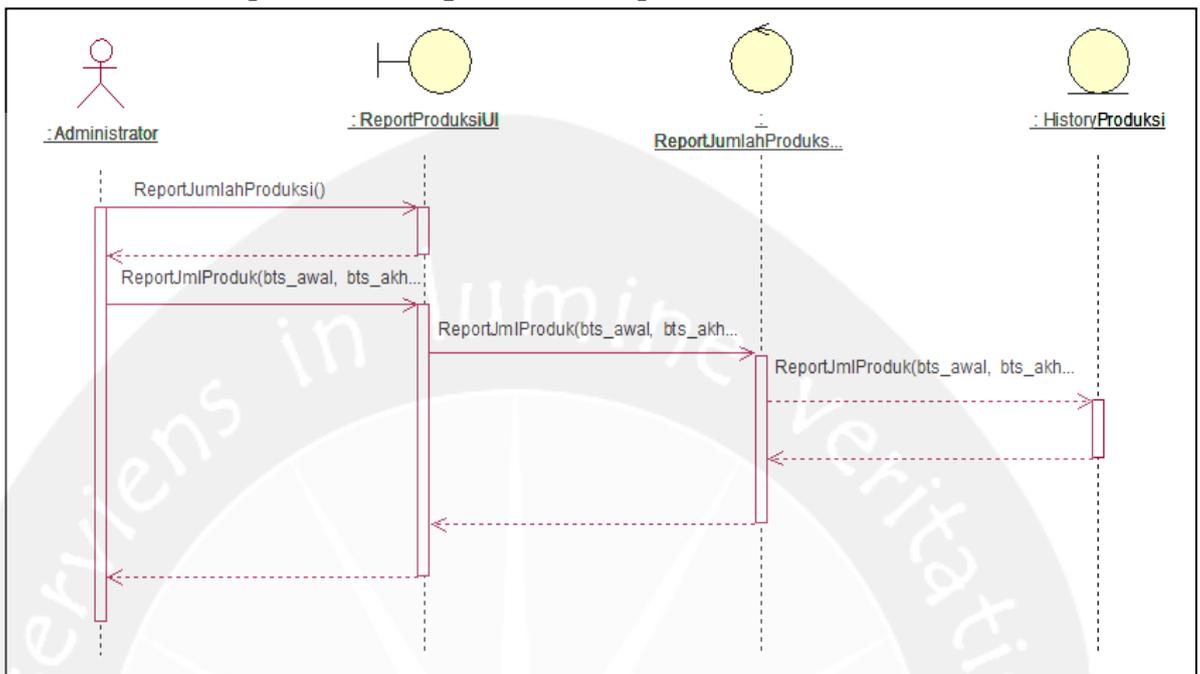
Gambar 2.5 Sequence Diagram : Edit Permintaan Customer

### 2.2.1.3. Penentuan Jumlah Produksi



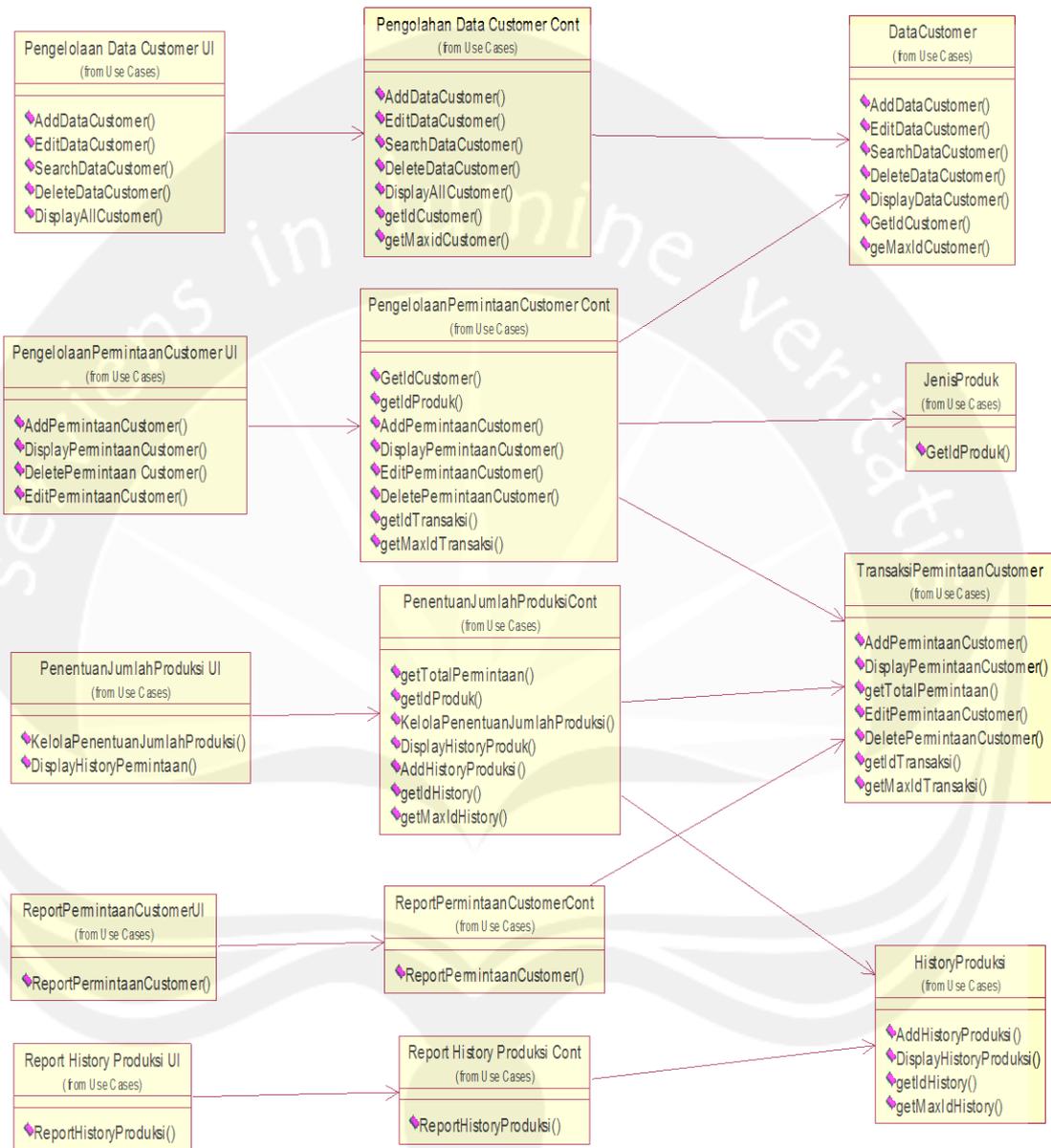
Gambar 2.6 Sequence Diagram : Penentuan Jumlah Produksi

#### 2.2.1.4. Report History Produksi produksi



Gambar 2.7 Sequence Diagram : Report History Produksi

### 2.2.2. Class Diagram



Gambar 2.6 Class Diagram fungsionalitas SPKPJP

### 2.2.3. Deskripsi Class

#### 2.2.3.1. Spesifikasi Design Kelas Pengelolaan Data Customer UI

PengelolaanDataCustomerUI	<<boundary>>
<pre>+PengelolaanDataCustomer()      Default konstruktor untuk inialisasi semua atribut dari kelas ini.  +AddDataCustomer(nama_customer, alamat, kodepos, telepon, contact_person):void      Operasi ini digunakan untuk melakukan penambahan data <i>customer</i> baru untuk disimpan ke dalam basis data sebagai data <i>Customer</i>. Dimana data yang di inputkan berupa <i>nama_customer</i>, <i>alamat</i>, <i>kodepos</i>, <i>telepon</i>, <i>contact_person</i>.  +cekEntry():void      Operasi ini digunakan sebagai <i>compare</i> atau pengecekan untuk menentukan apakah data yang diinputkan user tepat.  Selain itu terdapat juga fungsi :  +EditDataCustomer(nama_customerLama, nama_customerEdit, alamat, kodepos, telepon, contact_person): void      Operasi ini digunakan untuk melakukan perubahan pada data <i>customer</i> yang tersimpan di basis data dengan menggunakan parameter <i>nama customer lama</i> sebagai penentu di basis data.  + SearchDataCustomer(nama_customer): void      Operasi ini digunakan untuk mencari <i>data customer</i> yang ada pada data <i>customer</i> di basis data dengan menginputkan <i>nama customer</i> yang dicari.  + DeleteDataCustomer(nama_customer): void      Operasi ini digunakan untuk menghapus data <i>customer</i> yang ada pada data <i>customer</i> di basis data.</pre>	

### 2.2.3.2. Spesifikasi Design Kelas Pengelolaan Permintaan Customer UI

<b>PengelolaanPermintaanCustomer UI</b>	<<boundary>>
<pre> +PengelolaanPermintaanCutomer()    Default konstruktor untuk inisialisasi semua atribut dari kelas   ini.  +AddPermintaanCustomer(nama_customer, jenis_produk,   tgl_permintaan, jml_permintaan):void    Operasi ini digunakan untuk melakukan pengelolaan terhadap   permintaan customer pada setiap bulannya. Adapun atribut dalam   pengoperasian prosedur pengelolaan permintaan customer ini   yaitu nama_customer, jenis_produk yang dipilih melalui radio   button, tgl_permintaan berupa tanggal dimana customer meminta   produk, dan jml_permintaan untuk menampung total permintaan   customer untuk barang tersebut.  Selain itu terdapat juga fungsi :  + EditPermintaanCustomer(nama_customerLama, nama_customerBaru,   tgl_tansaksi, jenis_produkLama, jenis_produkBaru,   jml_permintaan): void    Operasi ini digunakan untuk melakukan perubahan pada data   permintaan customer yang tersimpan di basidata.  + DeleteDataPermintaanCustomer(nama_customer, tgl_permintaan,   jenis_produk, jml_permintaan): void    Operasi ini digunakan untuk menghapus data permintaan customer   yang ada pada basidata. </pre>	

### 2.2.3.3. Spesifikasi Design kelas Penentuan Jumlah Produksi UI

PenentuanJumlahProduksiUI	<<boundary>>
<pre> +PenentuanJumlahProduksi()  Default konstruktor untuk inisialisasi semua atribut dari kelas ini.  +PenentuanJmlProduksi(tgl_produk, total_permintaan, total_persediaan):void  Operasi ini digunakan untuk melakukan penentuan jumlah produksi untuk setiap produk berdasarkan total permintaan dan persediaan di gudang dengan inputan yang terdiri dari tgl_produk untuk menampung tanggal proses produksi dilakukan, total_permintaan untuk menampung total permintaan dari customer, dan total_persediaan yang ada di gudang.  +cekEntry():void  Operasi ini digunakan sebagai <i>compare</i> atau pengecekan untuk menentukan apakah data yang diinputkan user tepat.  Selain itu terdapat juga fungsi :  + DisplayHistoryProduksi (): void  Operasi ini digunakan untuk menampilkan seluruh history jumlah produksi berdasarkan total permintaan dan persediaan untuk setiap bulannya. </pre>	

#### 2.2.3.4. Spesifikasi Design kelas Report Permintaan CustomerUI

<b>ReportPermintaanCustomerUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre>+ ReportPermintaanCustomer(): void</pre> <p>Default konstruktor untuk inisialisasi semua atribut dari kelas ini.</p> <pre>+ ReportPermintaanCustomer(batas_awal, batas_akhir, jenis_produk): void</pre> <p>Operasi ini digunakan untuk menerbitkan report dari permintaan customer pada kurun waktu tertentu berdasarkan batasan inputan pengguna. batas_awal digunakan untuk menampung tanggal awal pencetakan report dan batas_akhir untuk menampung tanggal akhir dari penentuan produksi, sedangkan parameter jenis_produk digunakan untuk menampung jenis produk yang akan diterbitkan laporannya.</p>	

### 2.2.3.5. Spesifikasi Design kelas Report History Produksi UI

<b>ReportHistoryProduksiUI</b>	<b>&lt;&lt;boundary&gt;&gt;</b>
<pre> + ReportHistoryProduksi(): void      Default konstruktor untuk inisialisasi semua atribut dari     kelas ini.  + ReportHistoryProduksi(batas_awal, batas_akhir, jenis_produk):   void      Operasi ini digunakan untuk menerbitkan report dari total     permintaan customer, total persediaan di gudang, dan total     perencanaan produksi pada kurun waktu tertentu berdasarkan     batasan inputan pengguna. batas_awal digunakan untuk menampung     tanggal awal pencetakan report dan batas_akhir untuk menampung     tanggal akhir dari penentuan produksi, sedangkan parameter     jenis_produk digunakan untuk menampung jenis produk yang akan     diterbitkan laporannya.           </pre>	

### 2.2.3.6. Spesifikasi Design kelas Pengelolaan Data Customer Cont

PengelolaanDataCustomerCont	<<control>>
<pre> +getMaxIdCustomer() : int      Operasi ini digunakan untuk memperoleh id_customer terbesar yang nantinya digunakan untuk mengenerete secara otomatis id_customer selanjutnya.  +AddDataCustomer(nama_customer, alamat, kodepos, telepon, contact_person):void      Operasi ini digunakan untuk melakukan penambahan data customer baru untuk disimpan ke dalam database sebagai data Customer. Dimana data yang di inputkan berupa nama_customer, alamat, kodepos, telepon, contact_person.  Selain itu terdapat juga fungsi :  +EditDataCustomer(nama_customerLama, nama_customerEdit, alamat, kodepos, telepon, contact_person): void      Operasi ini digunakan untuk melakukan pengubahan pada data customer yang tersimpan di basidata dengan menggunakan parameter nama customer lama sebagai penentu di basidata.  + SearchDataCustomer(nama_customer): void      Operasi ini digunakan untuk mencari data customer yang ada pada data customer di basidata dengan menginputkan nama customer yang dicari.  + DeleteDataCustomer(nama_customer): void      Operasi ini digunakan untuk menghapus data customer yang ada pada data customer di basidata. </pre>	

### 2.2.3.7. Spesifikasi Design kelas Pengolahan Permintaan Customer Cont

<b>PengelolaanPermintaanCustomer Cont</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre> +getIdCustomer(nama_customer) : int  Operasi ini digunakan untuk memperoleh id_customer yang telah melakukan transaksi permintaan produk.  +getIdProduk(nama_produk) : int  Operasi ini digunakan untuk memperoleh id_produk yang di minta dalam transaksi permintaan produk.  +AddPermintaanCustomer(nama_customer, nama_produk,tgl_permintaan, jml_permintaan):void  Operasi ini digunakan untuk melakukan pengelolaan terhadap permintaan customer pada setiap bulannya. Adapun atribut dalam pengoperasian prosedur pengelolaan permintaan customer ini yaitu nama_customer, jenis_produk yang dipilih melalui radio button, tgl_permintaan berupa tanggal dimana customer meminta produk, dan jml_permintaan untuk menampung total permintaan customer untuk barang tersebut.  Selain itu terdapat juga fungsi :  + EditPermintaanCustomer(nama_customerLama, nama_customerBaru, tgl_tansaksi, jenis_produkLama, jenis_produkBaru, jml_permintaan): void  Operasi ini digunakan untuk melakukan perubahan pada data permintaan customer yang tersimpan di basidata.  + DeleteDataPermintaanCustomer(nama_customer, tgl_permintaan, jenis_produk, jml_permintaan): void  Operasi ini digunakan untuk menghapus data permintaan customer yang ada pada basidata. </pre>	

**2.2.3.8. Spesifikasi Design kelas Penentuan Jumlah Produksi Cont**

<b>PenentuanJumlahProduksiCont</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre> +getIdProduk(nama_produk) : int  Operasi ini digunakan untuk memperoleh id_produk yang di minta dalam transaksi permintaan produk.  +getTotalPermintaan(id_produk, bln_permintaan): int  Operasi ini digunakan untuk memperoleh total permintaan dari seluruh customer untuk setiap produk.  +PenentuanJmlProduksi(tgl_produk, total_permintaan, total_persediaan):void  Operasi ini digunakan untuk melakukan penentuan jumlah produksi untuk setiap produk berdasarkan total permintaan dan persediaan di gudang dengan inputan yang terdiri dari tgl_produk untuk menampung tanggal proses produksi dilakukan, total_permintaan untuk menampung total permintaan dari customer, dan total_persediaan yang ada di gudang.  +getMaxId_history():int  Operasi ini digunakan untuk memperoleh id_history terbesar. Sehingga dapat dilakukan proses generate id_history secara otomatis.  +AddHistoryProduksi(tgl_perencanaan, total_permintaan, total_persediaan, jml_perencanaanProduksi): void  Operasi ini digunakan untuk menyimpan data history produksi ke basisdata dengan inputan berupa tanggal perencanaan produksi, total permintaan customer, total persediaan di gudang, dan jumlah perencanaan / penentuan jumlah produksi untuk masing- masing produk. </pre>	

**2.2.3.9. Spesifikasi Design kelas Report Permintaan Customer Cont**

<b>ReportPermintaanCustomerCont</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ ReportPermintaanCustomer(batas_awal, batas_akhir,     jenis_produk): void</pre> <p>Operasi ini digunakan untuk menerbitkan report dari permintaan customer pada kurun waktu tertentu berdasarkan batasan inputan pengguna. batas_awal digunakan untuk menampung tanggal awal pencetakan report dan batas_akhir untuk menampung tanggal akhir dari penentuan produksi, sedangkan parameter jenis_produk digunakan untuk menampung jenis produk yang akan diterbitkan laporannya.</p>	

**2.2.3.10. Spesifikasi Design kelas Report History Produksi Cont**

<b>ReportHistoryProduksiCont</b>	<b>&lt;&lt;control&gt;&gt;</b>
<pre>+ ReportHistoryProduksi(batas_awal, batas_akhir, jenis_produk):     void</pre> <p>Operasi ini digunakan untuk menerbitkan report dari total permintaan customer, total persediaan di gudang, dan total perencanaan produksi pada kurun waktu tertentu berdasarkan batasan inputan pengguna. batas_awal digunakan untuk menampung tanggal awal pencetakan report dan batas_akhir untuk menampung tanggal akhir dari penentuan produksi, sedangkan parameter jenis_produk digunakan untuk menampung jenis produk yang akan diterbitkan laporannya.</p>	

### 2.2.3.11. Spesifikasi Design kelas JenisProduk

<b>JenisProduk</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>+ getIdProduk(nama_produk): int</pre> <p>Operasi ini digunakan untuk memperoleh id_produk berdasarkan nama_produk yang diinputkan.</p>	

### 2.2.3.12. Spesifikasi Design Kelas Pengelolaan Data Customer

<b>PengelolaanDataCustomer</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>+AddDataCustomer(id,nama_customer, alamat, kodepos, telepon, contact_person):void</pre> <p>Operasi ini digunakan untuk melakukan penambahan data customer baru untuk disimpan ke dalam database sebagai data Customer. Dimana data yang di inputkan berupa id_customer, nama_customer, alamat, kodepos, telepon, contact_person.</p> <p>Selain itu terdapat juga fungsi :</p> <pre>+EditDataCustomer(id_customer, nama_customerEdit, alamat, kodepos, telepon, contact_person): void</pre> <p>Operasi ini digunakan untuk melakukan perubahan pada data customer yang tersimpan di basidata dengan menggunakan parameter id_customer, nama_customerEdit, alamat, kodepos, telepon, contact_person.</p> <pre>+ SearchDataCustomer(nama_customer): void</pre> <p>Operasi ini digunakan untuk mencari data customer yang ada pada data customer di basidata dengan menginputkan nama customer yang dicari.</p> <pre>+ DeleteDataCustomer(nama_customer): void</pre> <p>Operasi ini digunakan untuk menghapus data customer yang ada pada data customer di basidata.</p> <pre>+ DisplayAllCutomer() : void</pre> <p>Operasi ini digunakan untuk menampilkan seluruh data customer yang ada di basidata pada saat program dijalankan.</p>	

### 2.2.3.13. Spesifikasi Design Kelas Transaksi Permintaan Customer

<b>TransaksiPermintaanCustomer</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>+AddPermintaanCustomer(id_transaksi, id_customer, id_produk, tgl_permintaan,jml_permintaan):void</pre> <p>Operasi ini digunakan untuk menyimpan transaksi permintaan customer ke dalam basisdata pada setiap bulannya. Adapun atribut dalam pengoperasian prosedur pengelolaan permintaan customer ini yaitu id_transaksi,id_customer, id_produk, tgl_permintaan berupa tanggal dimana customer meminta produk, dan jml_permintaan untuk menampung total permintaan customer untuk barang tersebut.</p> <p>Selain itu terdapat juga fungsi :</p> <pre>+ EditPermintaanCustomer(nama_customerLama, nama_customerBaru, tgl_tansaksi, jenis_produkLama, jenis_produkBaru, jml_permintaan): void</pre> <p>Operasi ini digunakan untuk melakukan perubahan pada data permintaan customer yang tersimpan di basisdata.</p> <pre>+ DeleteDataPermintaanCustomer(nama_customer, tgl_permintaan, jenis_produk, jml_permintaan): void</pre> <p>Operasi ini digunakan untuk menghapus data permintaan customer yang ada pada basisdata.</p> <pre>+ DisplayPermintaanCustomer(): void</pre> <p>Operasi ini digunakan untuk menampilkan seluruh permintaan customer untuk masing-masing produk.</p> <pre>+getTotalPermintaan(id_produk, bln_permintaan) : int</pre> <p>Operasi ini digunakan untuk memperoleh total permintaan customer perbulannya berdasarkan id_produk dan bulan permintaan produk.</p>	

### 2.2.3.14. Spesifikasi Design kelas History Produksi

<b>HistoryProduksi</b>	<b>&lt;&lt;entity&gt;&gt;</b>
<pre>+AddHistoryProduksi(id_history, id_produk, tgl_perencanaan, total_permintaan, total_persediaan, jml_perencanaanProduksi): void</pre> <p>Operasi ini digunakan untuk menyimpan data history produksi ke basisdata dengan parameter berupa id_history, id_produk, tanggal perencanaan produksi, total permintaan customer, total persediaan di gudang, dan jumlah perencanaan / penentuan jumlah produksi untuk masing-masing produk.</p>	

## 3. Perancangan Data

### 3.1. Dekomposisi Data

#### 3.1.1. Deskripsi Entitas Data MST\_CUSTOMER

Nama	Tipe	Panjang	Keterangan
Id_customer	Integer	5	Kode customer, primary Key
Nama_customer	Varchar	25	Nama dari customer
alamat	Varchar	50	Alamat dari customer
kodepos	Integer	5	Kodepos dari wilayah customer
telepon	Varchar	15	Telepon dari customer
Contact_person	Varchar	10	Contact person dari customer

### 3.1.2. Deskripsi Entitas Data MST\_JENISPRODUK

Nama	Tipe	Panjang	Keterangan
Id_produk	Integer	5	Kode dari jenis produk, primary Key
Nama_produk	Varchar	20	Nama dari produk

### 3.1.3. Deskripsi Entitas Data MST\_TRANSAKSIPERMINTAAN

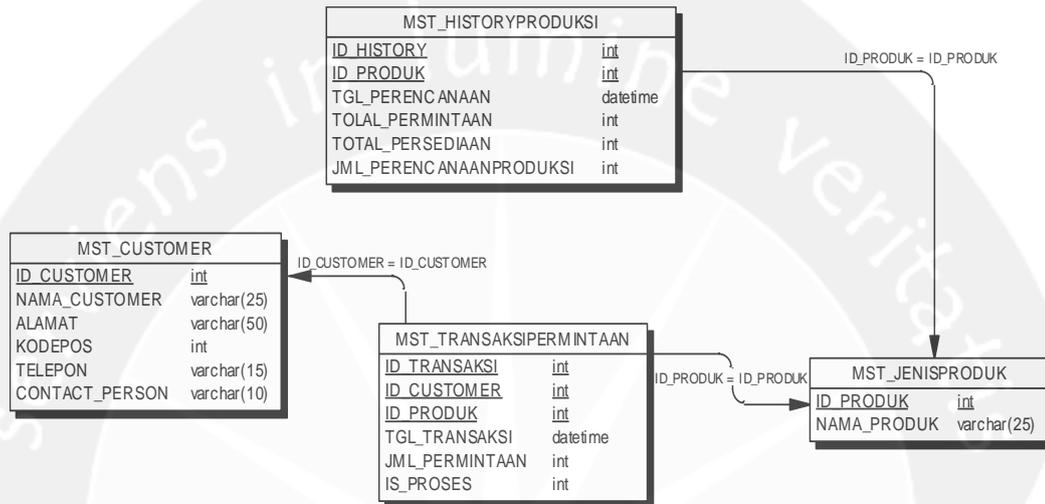
Nama	Tipe	Panjang	Keterangan
Id_transaksi	Integer	5	Kode transaksi, primary Key
Id_cutomer	Integer	5	Kode dari customer, primary key, foreign key (MST_CUSTOMER)
Id_produk	Integer	5	Kode dari produk, primary key, foreign key (MST_JENISPRODUK)
Tgl_transaksi	DateTime	-	Tanggal transaksi permintaan dilakukan
Jml_permintaan	Integer	5	Jumlah permintaan dari customer

3.1.4. Deskripsi Entitas  
MST\_HISTORYPRODUKSI

Nama	Tipe	Panjang	Keterangan
Id_history	Integer	5	Kode dari history produksi, primary Key
Id_produk	Integer	5	Kode dari produk, primary key, foreign key (MST_JENISPRODUK)
Tgl_perencanaan	DateTime	-	Tanggal perencanaan untuk bulan produksi
Total_permintaan	Integer	5	Total permintaan dari produk
Total_persediaan	Integer	5	Total persediaan di gudang

Jml_perencanaanProduksi	Integer	5	Jumlah produk yang direncanakan untuk diproduksi
-------------------------	---------	---	--

### 3.2. Physical Data Modeling



Gambar 2.7 Physical Data Modeling SPKPJP

#### 4. Perancangan Antar Muka

##### 4.1. Sketsa UI dan Deskripsinya

##### 4.1.1. Antarmuka Halaman Pengelolaan Data Customer

##### 4.1.1.1. Antarmuka Halaman Pengelolaan Add Data Customer

The screenshot shows a window titled "Pengolahan Data Customer" with the subtitle "tabel data customer". On the left side, there are five text input fields labeled "nama Customer", "Alamat", "kodepos", "telepon", and "contact person". Below these fields are two buttons labeled "save" and "close". On the right side, there is a large empty rectangular box representing a data table.

Gambar 4.1 Rancangan Antarmuka Add Data Customer

Antarmuka pada gambar 4.1 merupakan antarmuka untuk melakukan penambahan data customer baru ke dalam basis data. Hal tersebut dapat dilakukan dengan mengisi *field* pada antarmuka *Add Data Customer*, dimana *field* pada antarmuka *Add Data Customer* terdiri dari *nama\_customer*, *alamat*, *kodepos*, *telepon*, dan *contact\_person* yang berupa *textbox*. Setelah proses pengisian data pada *field Add Data Customer* dilakukan, untuk menyimpan data tersebut dilakukan dengan menekan *button "Save"*.

Apabila terjadi kesalahan pada saat pengisian data, misalnya untuk *field kodepos* dan *telepon* yang seharusnya berupa angka atau bilangan, maka sistem akan menampilkan instruksi untuk melakukan pengisian data kembali.

#### 4.1.1.2. Antarmuka Halaman Pengelolaan Search dan Edit Data Customer

The image shows a software interface titled "Pengolahan Data Customer". On the left side, there are several input fields: "search customer", "nama Customer", "Alamat", "kodepos", "telepon", and "contact person". Below these fields are two buttons: "Save" and "close". In the center, there is a "Search" button. To the right of the "Search" button is a large rectangular area labeled "tabel data customer", which is currently empty. The entire interface is overlaid on a watermark of a university logo with the motto "Luminae veritatis".

Gambar 4.2 Rancangan Antarmuka Search dan Edit Data Customer

Antarmuka pada gambar 4.2 merupakan antarmuka untuk melakukan pencarian dan pengubahan data *customer* yang ada di basis data. Untuk pencarian data dilakukan dengan memasukkan nama *customer* pada *field* "search customer", kemudian untuk memulai proses pencarian, dilakukan dengan menekan *button* "Search", maka masukkan pada *field search* nama *customer* akan dicocokkan pada semua nama *customer* yang ada di basis data, apabila nama ditemukan maka akan muncul data *customer* tersebut pada *dataGridView* yang ada di sebelah kanan *field search* nama *customer*.

Untuk melakukan pengubahan data *customer*, dilakukan dengan menekan data dari *customer* yang ditampilkan pada *dataGridView Edit Data Customer*, kemudian data *customer*

tersebut akan ditampilkan pada *field* masukkan *Edit Data Customer*. Setelah perubahan selesai dilakukan, untuk menyimpan data *customer* yang baru dapat dilakukan dengan menekan *button* "Save", maka secara otomatis data yang diubah akan ditampilkan pada *dataGridVied data customer* dan data hasil perubahan akan tersimpan di basis data.

#### 4.1.1.3. Antarmuka Halaman Pengelolaan Delete Data Customer

The screenshot displays a web interface titled "Pengolahan Data Customer". On the left side, there is a search section with a text input field labeled "search customer" and a "Search" button below it. Below the search section, there are several data fields with placeholder text: "nama Customer" with placeholder "<<nama customer>>", "Alamat" with "<<alamat>>", "kodepos" with "<<kodepos>>", "telepon" with "<<telepon>>", and "contact person" with "<<contact person>>". On the right side, there is a large empty rectangular area labeled "tabel data customer". At the bottom left of the interface, there are two buttons: "Delete" and "close".

**Gambar 4.3 Rancangan Antarmuka Delete Data Customer**

Antarmuka pada gambar 4.3 merupakan antarmuka untuk melakukan pencarian dan penghapusan data *customer* yang ada di basis data. Untuk pencarian data dilakukan dengan menginputkan nama *customer* pada *field* "search customer", kemudian untuk memulai proses pencarian, dilakukan dengan menekan *button* "Search", maka masukkan pada *field* akan dicocokkan pada semua nama *customer* yang ada di basis data, apabila nama ditemukan maka data *customer* tersebut akan

ditampilkan pada *dataGridView* data *customer* yang ada di sebelah kanan *field Delete Data Customer*.

Untuk melakukan penghapusan data *customer*, dilakukan dengan menekan data dari *customer* yang ditampilkan pada *dataGridView Delete Data Customer*, kemudian data *customer* tersebut akan ditampilkan pada *field* keterangan *Delete Data Customer*. Untuk melakukan penghapusan data *customer*, dapat dilakukan dengan menekan *button "Delete"*, maka secara otomatis data *customer* tersebut akan terhapus dari basis data.



#### 4.1.2. Antarmuka Halaman Pengelolaan Permintaan Customer

##### 4.1.2.1. Antarmuka Halaman Pengelolaan Add Permintaan Customer

The screenshot shows a web interface for adding customer requests. It includes a dropdown menu for selecting a customer name, a data grid for the selected customer, radio buttons to choose a product type (kain daster 160, kain daster 170, or kain Tedjo), input fields for the request date and quantity, another data grid for the request details, a dropdown for the request data table, a total request field, and buttons for processing and closing the form.

**Gambar 4.4 Rancangan Antarmuka Pengelolaan Add Permintaan Customer**

Antarmuka pada gambar 4.4 merupakan antarmuka untuk melakukan pengelolaan terhadap permintaan *customer*. Proses permintaan dilakukan dengan memilih nama *customer* melalui *combobox* "nama customer", kemudian data *customer* tersebut akan ditampilkan pada *dataGridView* "Nama customer". Kemudian permintaan dapat dilakukan dengan memilih *radio button* "jenis produk" dari jenis produk yang akan dipesan oleh *customer*, kemudian memasukkan tanggal permintaan dan jumlah permintaan. Untuk menyimpan permintaan *customer* tersebut ke basis data, dilakukan dengan menekan *button* "proses".

Untuk dapat melihat data dari permintaan *customer*, dapat dilakukan dengan menekan *combobox* "Permintaan Customer". pada *combobox* tersebut terdapat pilihan untuk menampilkan seluruh transaksi, transaksi yang belum diproses, dan transaksi yang sudah diproses.

#### 4.1.2.2. Antarmuka Halaman Pengelolaan Edit Permintaan Customer

The screenshot shows a web-based form for editing customer orders. On the left side, there are several input fields: a dropdown menu for 'Nama Customer', a group of radio buttons for 'jenis produk' with options 'kain daster 160', 'kain daster 170', and 'kain Tedjo', and two text input fields for 'tgl\_permintaan' and 'jml\_permintaan'. Below these fields are two buttons labeled 'Save' and 'close'. On the right side, there are two data grids. The top one is labeled 'tabel data customer' and the bottom one is labeled 'tabel data Permintaan customer'. Both grids are currently empty.

Gambar 4.5 Rancangan Antarmuka Pengelolaan Edit Permintaan Customer

Antarmuka pada gambar 4.5 merupakan antarmuka untuk melakukan perubahan data permintaan *customer* yang ada pada basis data. Untuk melakukan proses perubahan data permintaan *customer* dilakukan dengan menekan data permintaan pada *dataGridView* "data permintaan customer", kemudian data tersebut akan muncul pada *field editDataPermintaan*. Setelah proses perubahan data permintaan *customer* tersebut selesai

dilakukan, untuk menyimpan perubahan data permintaan tersebut dilakukan dengan menekan *button* "Save".

#### 4.1.2.3. Antarmuka Halaman Pengelolaan Delete Permintaan Customer

The screenshot shows a web form for deleting customer orders. On the left, there is a section titled 'jenis produk' with three radio buttons: 'kain daster 160', 'kain daster 170', and 'kain Tedjo'. Below this are four input fields: 'Nama Customer' with placeholder '<<nama\_customer>>', 'jenis produk' with placeholder '<<jenis\_produk>>', 'tgl\_permintaan' with placeholder '<<tgl\_permintaan>>', and 'jml\_permintaan' with placeholder '<<jml\_permintaan>>'. At the bottom left of the form are two buttons: 'Delete' and 'close'. On the right side, there are two data tables. The top one is 'tabel data customer' and the bottom one is 'tabel data Permintaan customer'. At the bottom right, there is a 'Total Permintaan' field.

Gambar 4.6 Rancangan Antarmuka Pengelolaan Delete Permintaan Customer

Antarmuka pada gambar 4.6 merupakan antarmuka untuk melakukan penghapusan data permintaan *customer* yang ada pada basis data. Untuk melakukan proses penghapusan data permintaan *customer*, dilakukan dengan memilih jenis produk melalui *radio button* "jenis produk" yang mana terdapat data yang akan dihapus. Kemudian untuk menampilkan keterangan dari data yang akan dihapus pada *field Delete Data Customer*, dilakukan dengan menekan data permintaan pada *dataGridView* "data permintaan customer", kemudian data tersebut akan muncul pada *field Delete Data Permintaan*. Untuk menghapus

data permintaan yang telah ditampilkan, dilakukan dengan menekan *button* "Delete".

#### 4.1.3. Antarmuka Halaman Penentuan jumlah Produksi

The interface contains the following elements:

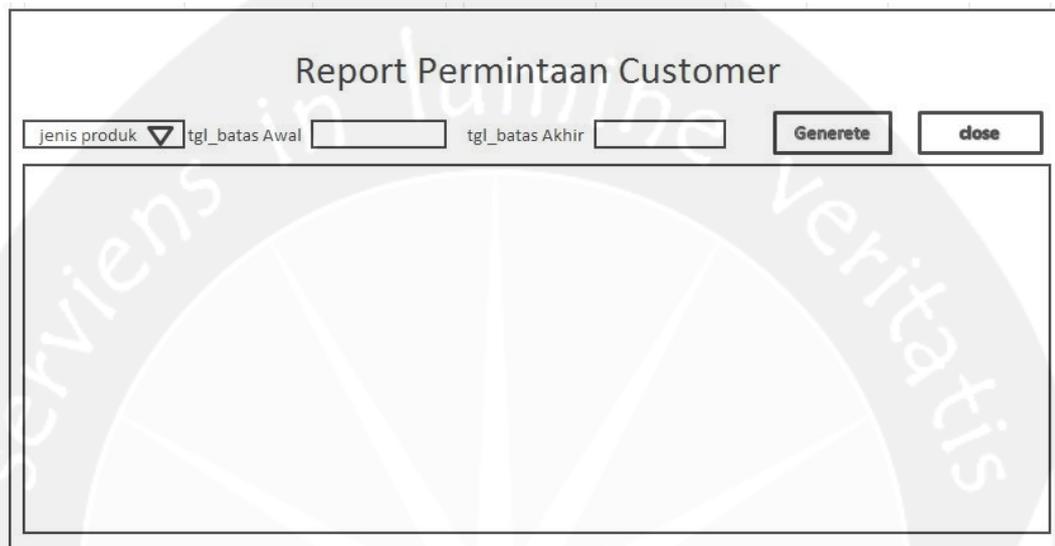
- Fungsi Keanggotaan:** A dropdown menu.
- Tgl Perencanaan Produksi:** A date input field.
- Jenis Produk:** Three radio buttons corresponding to "kain daster 160", "kain daster 170", and "kain Tedjo".
- Total Permintaan:** An input field.
- Total Persediaan:** An input field.
- Buttons:** "Proses" and "close".
- Tables:** Two empty table placeholders labeled "tabel history produksi" and "tabel data Permintaan customer".
- Keterangan:** A text area for notes.

**Gambar 4.7 Rancangan Antarmuka Penentuan Jumlah Produksi**

Antarmuka pada gambar 4.7 merupakan antarmuka untuk melakukan proses penentuan jumlah produksi berdasarkan total permintaan *customer* dan persediaan di gudang. Proses penentuan jumlah produksi dapat dilakukan dengan memilih fungsi keanggotaan yang akan digunakan yang terdiri dari fungsi *segitiga* dan fungsi *phi*, kemudian memasukkan tanggal perencanaan produksi, kemudian memilih jenis produk yang akan diproduksi, maka pada field "total permintaan" akan muncul total dari permintaan customer terhadap jenis produk yang telah dipilih. Kemudian untuk jumlah persediaan pada field "total persediaan" diisikan oleh user. Untuk memulai proses penentuan jumlah produksi, dilakukan dengan menekan

button "Proses", maka pada field "keterangan" akan muncul keterangan mengenai jumlah produk yang harus diproduksi.

#### 4.1.4. Antarmuka Halaman Report Permintaan Customer



**Gambar 4.8 Rancangan Antarmuka Report PermintaanCustomer.**

Antarmuka pada gambar 4.8 merupakan antarmuka untuk menerbitkan laporan mengenai jumlah permintaan produk dari masing-masing *customer*. Untuk menerbitkan laporan tersebut, dilakukan dengan memilih jenis produk melalui *combobox* "jenis produk", batas tanggal awal dari data yang akan dibuat laporannya, dan batas tanggal akhir. Kemudian untuk mulai menerbitkan laporan, dilakukan dengan menekan *button* "generate".

#### 4.1.5. Antarmuka Halaman Report History Produksi

Report History Produksi

jenis produk ▼ tgl\_batas Awal  tgl\_batas Akhir

Gambar 4.8 Rancangan Antarmuka Report History Produksi.

Antarmuka pada gambar 4.9 merupakan antarmuka untuk menerbitkan laporan mengenai total permintaan produk, total persediaan di gudang, dan total perencanaan produksi. Untuk menerbitkan laporan tersebut, dilakukan dengan memilih jenis produk melalui *combobox* "jenis produk", batas tanggal awal dari data yang akan dibuat laporannya, dan batas tanggal akhir. Kemudian untuk mulai menerbitkan laporan, dilakukan dengan menekan *button* "generate".