BAB III

PEMBAHASAN

1.1 Analisis Sistem

Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau menyelesaikan suatu sasaran tertentu[6]. Sedangan analisis sistem merupakan penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dari kebutuhan yangdiharapkan sehingga dapat diusulkan perbaikan-perbaikannya[6].

Hal-hal yang dilakukan pada tahap analisis sistem adalah analisis masalah, prosedur yang sedang berjalan, dokumen yang terlibat, analisis pengkodean, analisis kebutuhan fungsional dan kebutuhan nonfungsional meliputi analisis perangkat keras, analisis perangkat lunak serta analisis pengguna.

1.1.1 Analisis masalah

Mengidentifikasi masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat dicapai. Oleh karena itu langkah pertama yang harus dilakukan pada tahap ini adalah mengidentifikasi terlebih dahulu masalah masalah yang terjadi (*identify*).

Berdasarkan hasil observasi dan wawancara dengan pihak MIS-KA PT. Dirgantara Indonesia, didapat kesimpulan permasalahan dalam sistem pengolahan data *Sales Contract*, yang nantinya akan dijadikan landasan usulan perancangan sistem yang baru. Adapun permasalahan tersebuat yaitu:

1. Pengolahan data *Sales Contract* masih berbasis *desktop*.
2. Pada pengolahan modul *Sales Contract* masih banyak aplikasi yang harus di tambahkan dan diperbaiki.

Berdasarkan permasalahan yang telah diuraikan, penelitian ini dimaksudkan untuk menganalisis dan implemetasi modul Sales *Contract* yang sudah ada (berbasis desktop) ke modul Sales *Contract* berbasis web. Kegiatan difokuskan untuk mengembangkan modul lama (*dekstop application*) dengan keadaan modul yang berjalan saat ini yang akan dibuat berbasis web. Selanjutnya membuat dokumentasi dan implementasi (*coding*) dengan fungsi dan framework yang sudah ditentukan.

1.2 Gambaran Awal Modul

Gambaran awal modul merupakan *bussiness process* dari *Integrated Resource Planning* (IRP) yang awalnya adalah sebuah aplikasi *desktop*. Aplikasi tersebut diimplementasi sesuai dengan fungsi-fungsi yang ada dalam modul *Sales Contract* dengan penjelasan tentang bagaimana sirkulasi kegiatan-kegiatan mulai dari proses kontrak, perencanaan barang, pengadaan barang sampai *service-service* yang ada. Para Aktor yang nantinya menjadi *User* dijelaskan satu-persatu sesuai tugasnya.

1.3 Analisis Kebutuhan Non-Fungsional

Tahap ini meliputi elemen-elemen apa saja yang dibutuhkan untuk sebuah sistem yang akan dibangun, spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan sampai dengan sistem tersebut diimplementasikan. Kebutuhan non-fungisonal terbagi menjadi tiga yaitu analisis perangkat keras, perangkat lunak dan analisis pengguna.

1.3.1 Analisis Perangkat Keras

Perangkat keras atau *hardware* merupakan salah satu hal yang sangat penting, karena tanpa *hardware* yang memenuhi syarat, aplikasi yang akan dibangun tidak akan bisa berjalan. Berikut spesifikasi minimal perangkat keras yang ada pada MIS-KA PT. Dirgantara Indonesia yaitu :

a. Prosesor Pentium 166MHz atau yang lebih tinggi.
b. System operasi Windows 98, Windows 98 Se, Windows ME, Windows 2000, Windows 2003, dan Windows XP
c. Monitor
d. Hardisk minimal size yang kosong sebesar 500 Mb
e. Memory/RAM 256 Mb
f. VGA atau resolusi monitor 1024x768
g. CD ROM Drive
h. Keyboard
i. Mause
j. Perangkat Jaringan :

- Switch minamal 4 port (untuk menghubungkan 4 Host PC Client)
- Access Point (Untuk penggunaan koneksi Wirreless)
- Cable UTP Secukupnya dan RJ45

1.3.2 Analisis Perangkat Lunak

Perangkat lunak merupakan hal yang penting dalam mendukung kinerja sebuah sistem dan spesifikasi minimal perangkat lunak yang digunakan di MIS-KA PT. Dirgantara Indonesia adalah sebagai berikut :

1. Sistem Operasi Windows.
2. Browser menggunakan Mozilla Firefox.
3. Aplikasi CMS (Content Management System) Open Source.
4. Database PostgreSQL 8.4.
5. Virtual Webserver menggunakan Xampp.
6. Development Tools menggunakan Notepad++.
7. Programming Environment menggunakan Zend FrameWork.

1.3.3 Analisis Pengguna

Pengguna yang akan menggunakan aplikasi yang akan dibangun ini yaitu pekerjaan operasional pegawai bagian Sales (admin) dengan sumsi pengguna yang sudah terbiasa menggunakan sistem berbasis *web*, dengan karakteristik sebagai berikut :

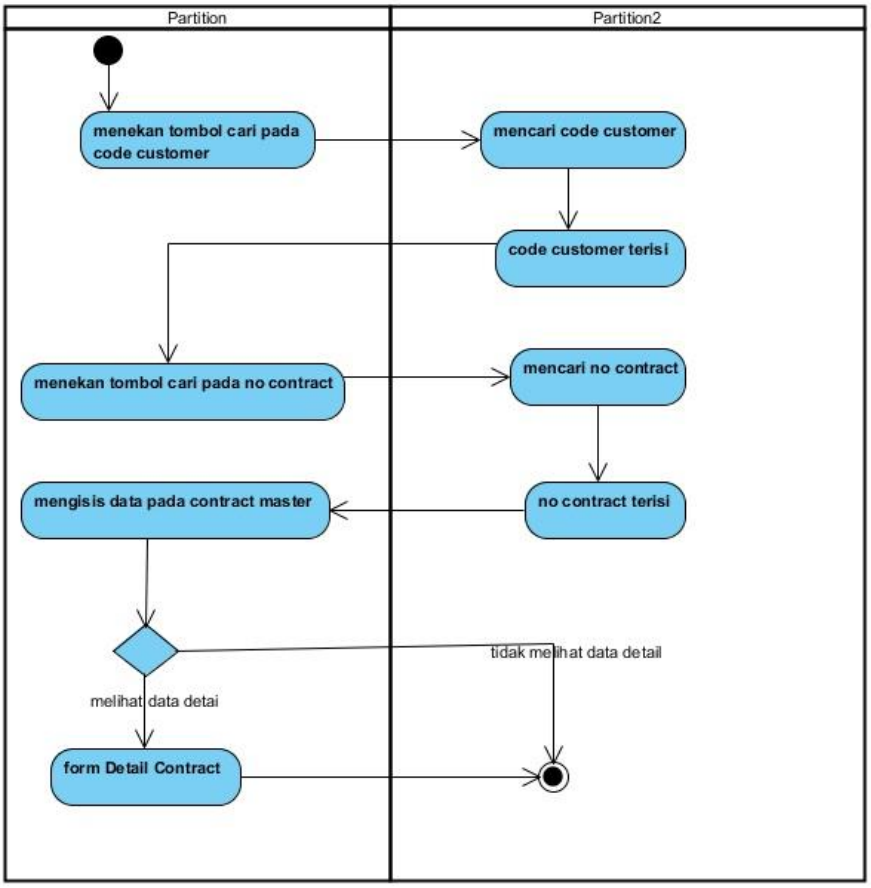
1. Mempunyai kemampuan dasar di bidang komputer.
2. Dapat mengoperasikan sistem operasi *Microsoft Windows XP*.

3. Memiliki pengalaman lebih lama dalam menggunakan komputer.
4. Memiliki latar belakang pendidikan S1.

1.4 Analisis Sismtem Yang Sedang Berjalan

Analisis sistem yang berjalan berisi tentang pemaparan sistem pengolahan data *Sales Contract* yang ada saat ini. Analisis ini dimaksudkan agar perangkat lunak yang dikembangkan tidak keluar dari cakupan sistem pengolahan data *Sales Contract* yang ada. Adapun prosedur pengolahan data *Sales Contract* yang ada di PT. Dirgantara Indonesia adalah :

1. Admin melakukan pengisian data petugas.
2. Admin mencari kode *Customer* dengan menekan tombol cari pada label code customer.
3. Admin mencari nomor *Contract Customer* dengan menekan tombol cari pada label no *Contract*.
4. Setelah terisi Admin dapat menambah data cutomer pada pada *Contract Master*.
5. Lalu Admin dapat melihat data yang telah ada pada form *Contract Detail.*



Gambar I.1 Analisis yang sedang berjalan

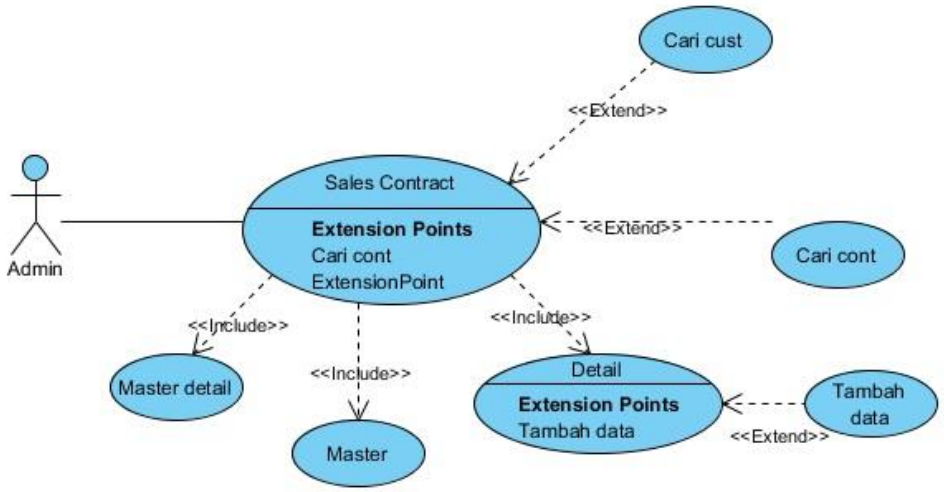
1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk mengetahui proses informasi yang mengalir melalui perangkat lunak. Untuk menggambarkan proses informasi secara umum menggunkan tools UML, adapun tahapan analisis sistem menggunakan UML adalah *use case diagram, use case skenario, activity diagram, squence diagram,* dan *class diagram* .

1.5.1 Use Case Diagram

Use case diagram merupakan konstruksi untuk mendeskripsikan hububungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat

pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pengembang. Dari analisis pengguna aplikasi yang ada maka *use case* diagram untuk aplikasi modul *Sales Contract* dapat dilihat dalam Gambar III.2.



Gambar I.2 Use Case

1.5.2 *Use Case* Skenario

Use Case Skenario mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilikakukan oleh sistem terhadap aktor. Berdasarkan *use case diagram* pada gambar maka *use case* skenario untuk aplikasi modul *Sales Contract* akan dijelaskan sebagai berikut :

1. *Use Case* Skenario Sales *Contract*

Tabel I.1 *Use Case* Skenario Sales *Contract*

Identifikasi	
Nomor	1
Nama	Sales *Contract*
Tujuan	Mengolah data Sales *Contract*
Deskripsi	Proses ini untuk mengolah data Sales *Contract*
Aktor	Admin
Use case yang berkaitan	Cari cust, Cari cont
Skenario Utama	
Kondisi Awal	Menampilkan Halaman Sales *Contract*
Aksi Aktor	**Reaksi Sistem**
1. Mengisi label yang ada pada halaman Sales *Contract*	
	2. Menampilkan Halaman yang telah terisi

2. *Use Case* Skenario Cari cust

Tabel I.2 *Use Case* Skenario Cari cust

Identifikasi	
Nomor	2
Nama	Cari cust
Tujuan	Mencari data cust

Deskripsi	Proses mencari data customer
Aktor	Admin
Use case yang berkaitan	-
Skenario Utama	
Kondisi Awal	Halaman list data cust ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol cari pada halaman Sales *Contract*	
	2. Menampilkan halaman list data cust
3. Memilih code cust	
	4. Menampilkan code cust yang dicari pada label code cust

3. *Use Case* Skenario Cari cont

Tabel I.3 *Use Case* Skenario Cari cont

Identifikasi	
Nomor	3
Nama	Cari cont
Tujuan	Mencari data *Contract*
Deskripsi	Proses mencari data *Customer* yang telah mempunyai *Contract*
Aktor	Admin
Use case yang berkaitan	-
Skenario Utama	

Kondisi Awal	Halaman list data cont ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Menekan tombol cari pada halaman Sales *Contract*	
	2. Menampilkan halaman list data cont
3. Memilih code cont yag telah mempunyai *Contract*	
	4. Menampilkan no dan nama cont

4. *Use Case* Skenario *Master Detail*

Tabel I.4 *Use Case* Skenario *Master Detail*

Identifikasi	
Nomor	4
Nama	*Master Detail*
Tujuan	Mengolah data Sales *Contract* secara *Detail*
Deskripsi	Proses ini untuk mengolah data Sales *Contract* secara *Detail*
Aktor	Admin
Use case yang berkaitan	-
Skenario Utama	
Kondisi Awal	Menampilkan Halaman *Contract Master*

Aksi Aktor	Reaksi Sistem
1. Mengisi label yang kosong	
	2. Menampilkan Halaman yang telah terisi
3. Menekan tombol simpan	
	4. Menampilkan pesan bahwa data telah tersimpan
	5. Data telah tersimpan
Skenario Alternatif (Proses Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Salah memasukkan data	
2. menekan tombol gagal	
	3. Menampilkan pesan bahwa data gagal disimpan
Kondisi Akhir	Admin dapat mengisi kembali data Sales *Contract* secara *Detail*

5. *Use Case* Skenario *Master*

Tabel I.5 *Use Case* Skenario *Master*

Identifikasi	
Nomor	5
Nama	*Master*
Tujuan	Menampilkan halaman *Contract Master Detail*
Deskripsi	Proses ini menampilkan halaman

	Contract Master Detail
Aktor	Admin
Use case yang berkaitan	-
Skenario Utama	
Kondisi Awal	Halaman *Contract Master Detail* ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Membuka Halaman *Contract Master Detail*	
	2. Menampilkan Halaman *Contract Master Detail*

6. *Use Case* Skenario *Contract Detail*

Tabel I.6 *Use Case* Skenario *Contract Detail*

Identifikasi	
Nomor	6
Nama	*Contract Detail*
Tujuan	Menampilkan halaman *Contract Detail*
Deskripsi	Proses ini menampilkan halaman *Contract Detail*
Aktor	Admin
Use case yang berkaitan	Tambah data
Skenario Utama	
Kondisi Awal	Halaman *Contract Detail* ditampilkan
Aksi Aktor	**Reaksi Sistem**

1. Membuka Halaman *Contract Detail*	
	2. Menampilkan Halaman *Master Detail*

7. *Use Case* Skenario Tambah Data

Tabel I.7 Use Case Skenario Tambah Data

Identifikasi	
Nomor	7
Nama	Tambah Data
Tujuan	Menambah data *Contract*
Deskripsi	Proses ini untuk menambahkan data customen yang telah memiliki *Contract*
Aktor	Admin
Use case yang berkaitan	-
Skenario Utama	
Kondisi Awal	Menampilkan Halaman *Contract Detail*
Aksi Aktor	**Reaksi Sistem**
1. Klik tulisan "klik untuk masukkan data"	
	2. Menampilkan Halaman Entry *Contract* Item
3. Menambahkan data *Contract*	
4. Menekan tombol tambahkan	
	5. Data telah ditambahkan dan tersimpan
Skenario Alternatif (Proses Gagal)	

Aksi Aktor	Reaksi Sistem
1. Klik tulisan "klik untuk masukkan data"	
	1. Data pada Sales *Contract* belum terisi
	2. Menampilkan pesan Cont. no belum terisi
	3. Kembali ke Halaman Sales Contrract
Kondisi Akhir	Admin dapat mengisi data Sales *Contract* terlebih dahulu

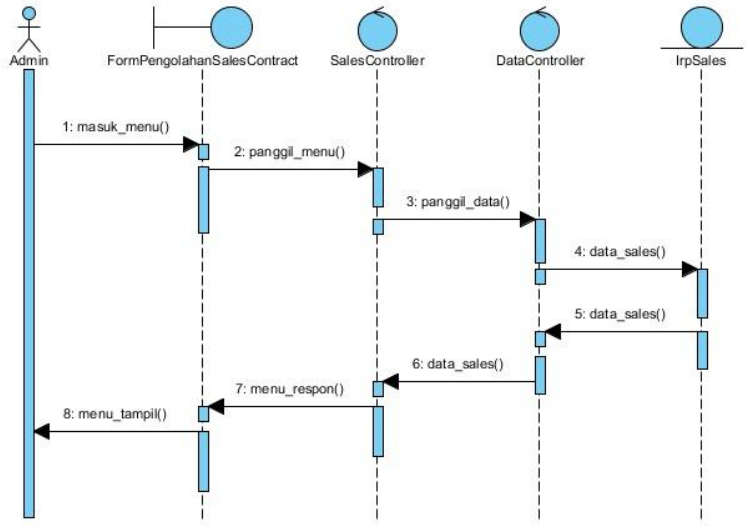
1.5.3 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar masing-masing objek pada setiap *use case* dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. *Sequence Diagram* pada aplikasi untuk aplikasi modul *Sales Contract* terdiri dari *Sequence Diagram* Sales *Contract, Sequence Diagram* cari cust, *Sequence Diagram Master Detail, Sequence Diagram Master* dan *Sequence Diagram Detail*. Berikut ini penjelasan dari masing-masing *Sequence Diagram* :

1.5.3.1 Sequence Diagram Sales Contract

Sequence diagram Sales *Contract* menunjukkan langkah awal dalam pengolahan data *Sales Contract. Sequence diagram* Sales *Contract* (Gambar III.3) terdiri dari *Sequence diagram* Cari Cust (Gambar III.4) dan *Sequence diagram*

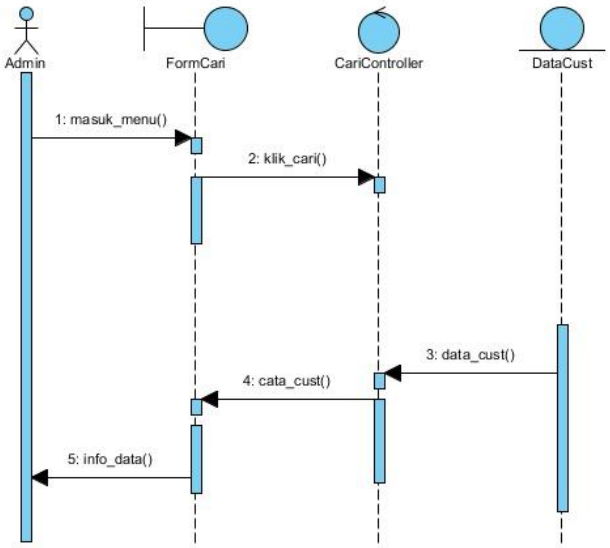
Cari Cont (Gambar III.5). *Sequence diagram* Sales *Contract* dapat dilihat pada Gambar III.3.



Gambar I.3 *Sequence Diagram Sales Contract*

1. *Sequence Diagram* Cari cust

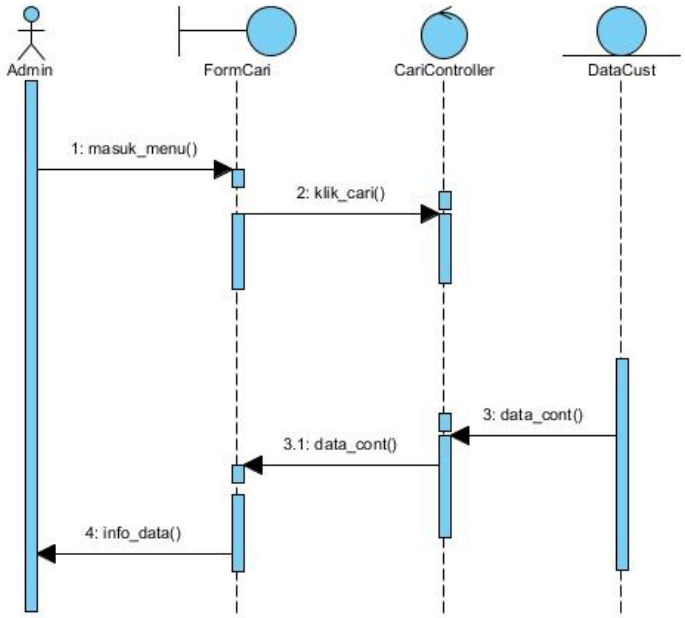
Sequence Diagram Cari Cust marupakan form pencarian data customer. Sequence Diagram Cari Cust dapat dilihat pada Gambar III.4.



Gambar I.4 *Sequence Diagram* Cari cust

2. *Sequence Diagram* Cari Cont

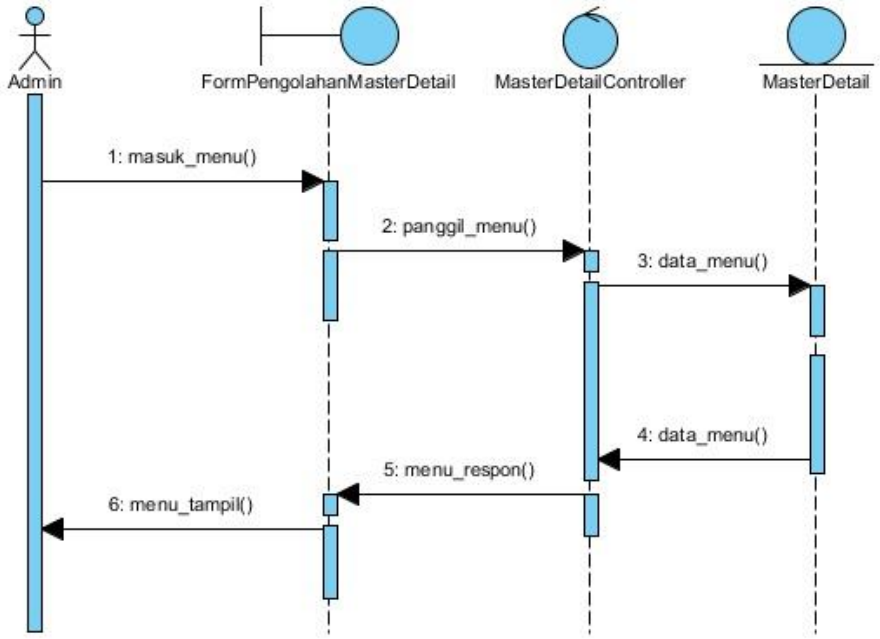
Sequence Diagram Cari Cont marupakan form pencarian data *Contract* customer. Sequence Diagram Cari Cont dapat dilihat pada Gambar III.5.



Gambar I.5 *Sequence Diagram* Cari Cont

1.5.3.2 *Sequence Diagram Master Detail*

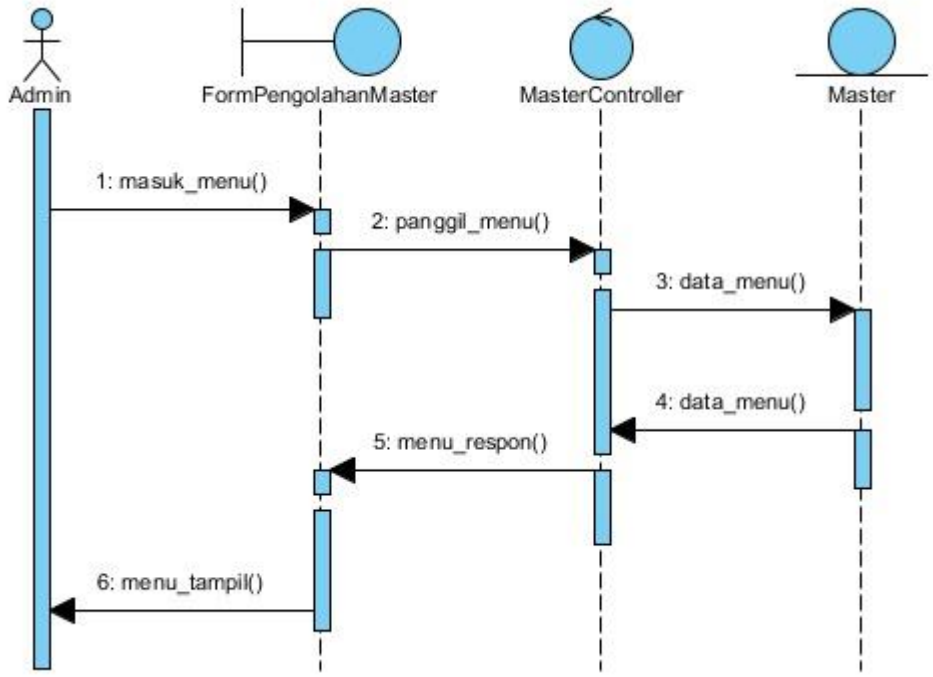
Sequence diagram Master Detail merupakan penambahan data *Customer* yang telah memiliki *Contract*. *Sequence diagram Master Detail* dapat dilihat pada Gambar III.6.



Gambar I.6 *Sequence Diagram Master Detail*

1.5.3.3 *Sequence Diagram Master*

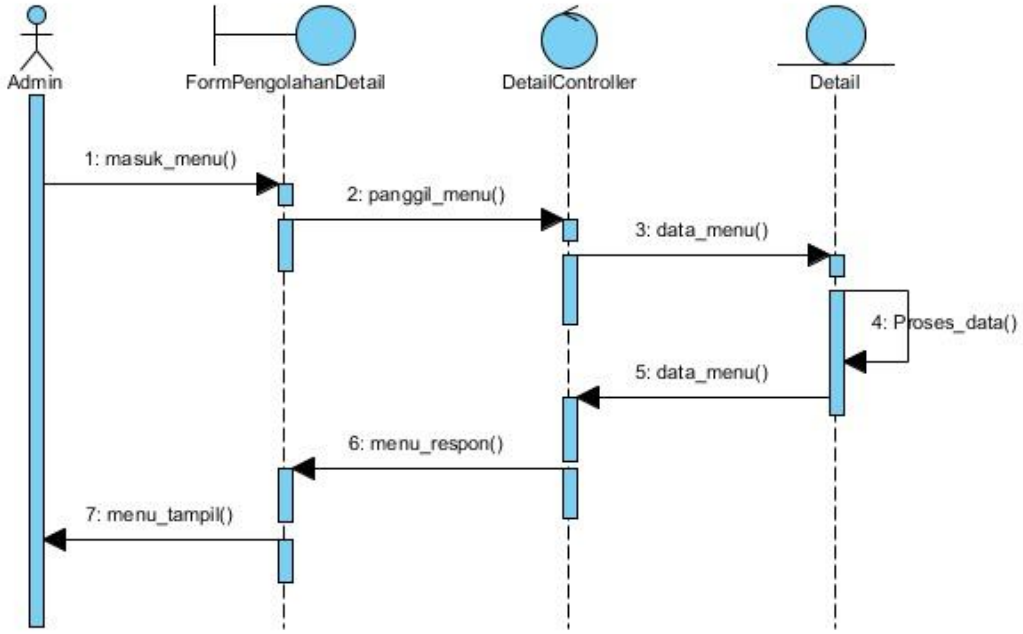
Se*quence diagram Master* merupaha menu *Contract Master Detail* dimana berisi table-tabel rinci data Sales *Contract*. Sequence Diagram *Master* dapat dilihat pada Gambar III.7.



Gambar I.7 *Sequence Diagram Master*

1.5.3.4 *Sequence Diagram Detail*

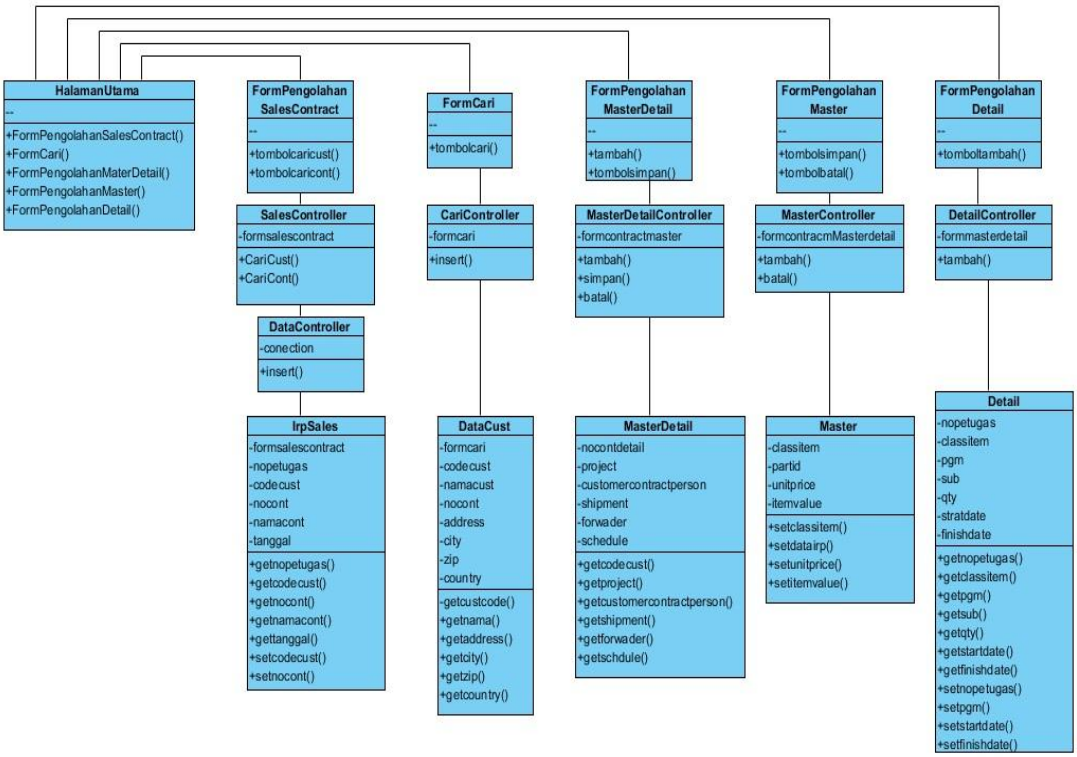
Sequence diagram Detail merupakan menu rinci Sales *Contract* yang bertujuan untuk menu pemanbahan *Contract* untuk *Customer* yang telah memiliki *Contract*. *Sequence diagram Detail* dapat dilihat pada Gambar III.8.



Gambar I.8 *Sequence diagram Contract Detail*

1.5.4 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut dan *method* yang ada pada masing-masing *class*. Hubungnnya dari masing-masing *class* yang ada tersebut digambarkan dengan menggunakan pewarisan dan generalisasi seperti terlihat pada Gambar III.9.



Gambar I.9 *Class Diagram*

1.6 Perancangan Sistem

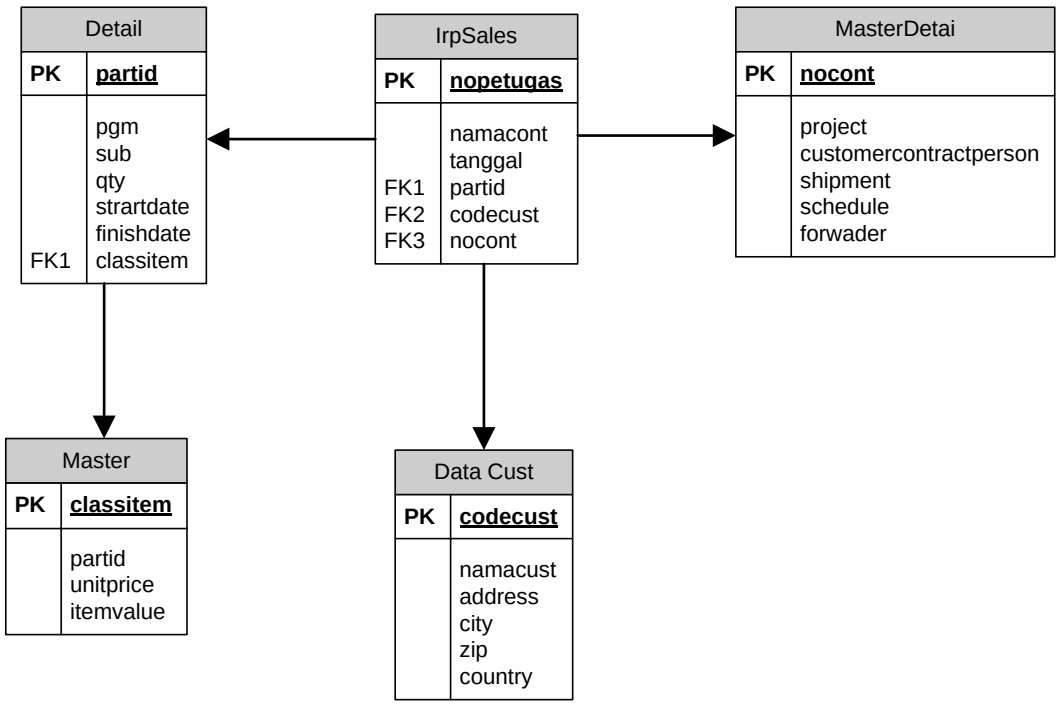
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi.

1.6.1 Perancangan Data

Perancangan data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan basis data terbagi menjadi dua yaitu skema relasi dan perancangan struktur tabel.

1.6.1.1 Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Gambar III.10 merupakan penjelasan rangkaian basis data pada modul *Sales Contract*.



Gambar I.10 Skema Relasi

1.6.1.2 Struktur Tabel

Tabel merupakan tempat penyimpanan informasi dari sebuah aliran data dalam sebuah sistem. Berikut merupakan struktur dari beberapa tabel sistem yang akan dibangun.

1. IrpSales yaitu menyimpan data-data *Customer Contract*

Tabel I.8 IrpSales

Nama Field	Tipe Data	Panjang	Null	Kunci

nopetugas	integer	-	No	*Primary Key*
nocont	integer	-	No	*Foreign Key*
partid	integer	-	No	*Foreign Key*
namacont	varchar	20	No	-
codecust	varchar	6	No	*Foreign Key*
tanggal	datetime	-	No	-

2. *Master Detail* yaitu menyimpan data *Detail Sales Contract*

Tabel I.9 *Master Detail*

Nama Field	Tipe Data	Panjang	Null	Kunci
nocont	integer	-	No	*Primary Key*
project	varchar	80	No	-
customerconractperson	varchar	19	No	-
shipment	varchar	20	Yes	-
schedule	datetime	-	No	-
forwader	varchar	10	No	-

3. *Detail* yaitu menyimpan data *Contract*

Tabel I.10 *Detail*

Nama Field	Tipe Data	Panjang	Null	Kunci
Partid	integer	-	No	*Primary Key*
Pgm	varchar	20	No	-
Classitem	varchar	10	No	*Foreign Key*
sub	varchar	15	Yes	-
Qty	Integer	-	No	-
startdate	Datetime	-	No	-
finishdate	datetime	-	No	

4. *Master* yaitu untuk menyimpan data-data cutomer yang memiliki banyak *Contract*

Tabel I.11 Master

Nama Field	Tipe Data	Panjang	Null	Kunci
classitem	varchar	10	No	*Primary Key*
partid	varchar	6	No	-
unitprice	varchar	15	No	-
itemvalue	varchar	20	Yes	-

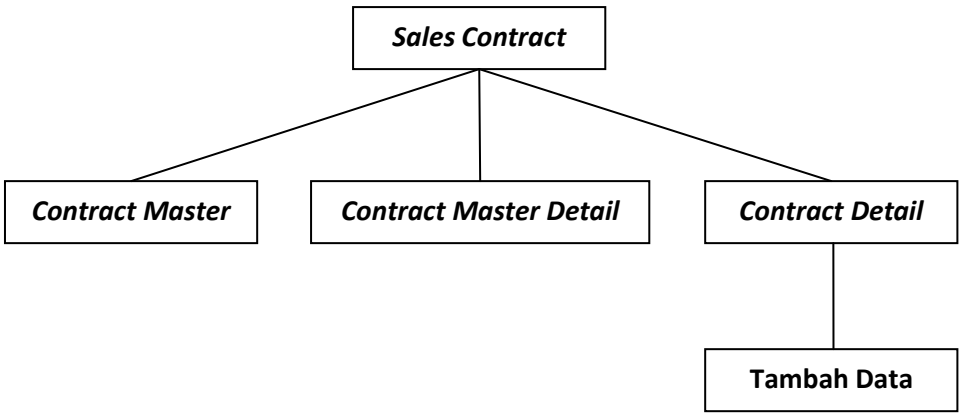
5. Data Cust yaitu untuk mengambil code *Customer* diman amenyimpan data *Customer*

Tabel I.12 Data Cust

Nama Field	Tipe Data	Panjang	Null	Kunci
codecust	varchar	6	No	*Primary Key*
Namacust	varchar	10	No	-
Address	varchar	40	No	-
city	varchar	15	Yes	-
zip	Integer	-	Yes	-
country	varchar	20	No	-

1.6.2 Struktur Menu

Perancangan struktur menu merupakan gambaran jalur pemakaian aplikasi sehingga aplikasi yang dikembangkan mudah dipahami dan mudah digunakan. Perancangan struktur menu dari aplikasi modul Sales *Contract* dapat dilihat pada Gambar III.10.



Gambar I.11 Struktur Menu

1.6.3 Perancangan Antarmuka

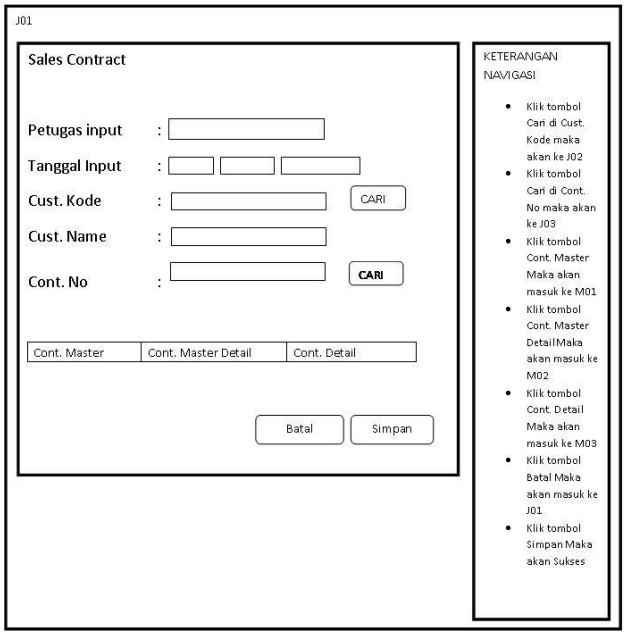
Perancangan antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun. Sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi yang user friendly.

1.6.3.1 Perancangan Form

Perancangan sistem informai Modul *Sales Contract* yang akan dibuat sebagai berikut :

1. Perancangan antar muka sales

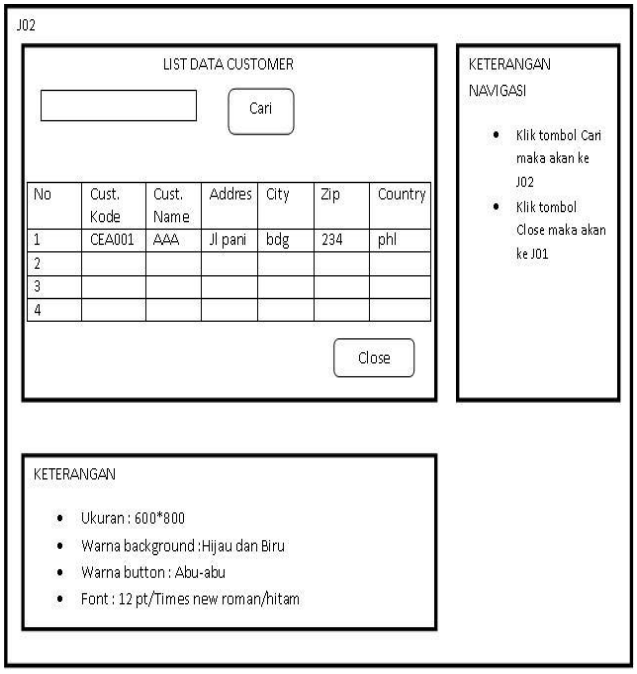
   Perancangan ini menjelaskan tentang isi form dari monioring sales *Contract*, yang nanti nya akan melakukan pencarian nama-nama sales *Contract*.



Gambar I.12 Perancangan antar muka sales

2. Perancangan antar muka data *customer/* cust.kode

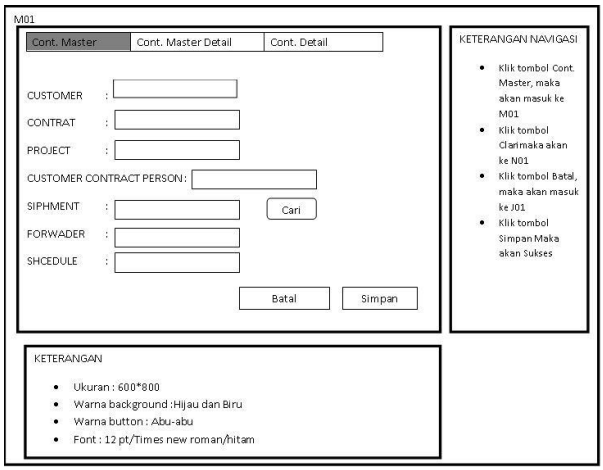
Perancangan ini menjelaskan tentang data-data list data customer, yang ada di form sales *Contract*.



Gambar I.13 Perancangan antar muka data *customer*/cust.kode

3. Perancangan antar muka cont. *Master*

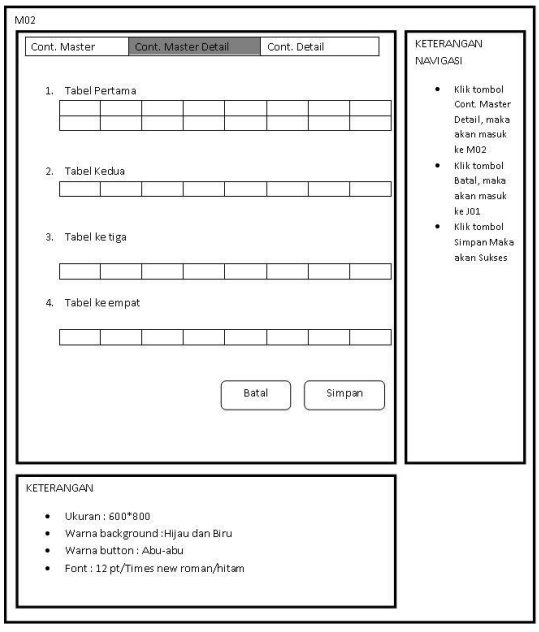
Perancangan ini menjelaskan tentang data *Master*, yang akan menyewa sperpat pesawat.



Gambar I.14 Perancangan antar muka cont. *Master*

4. Perancangan antar muka data cont.*Master Detail*

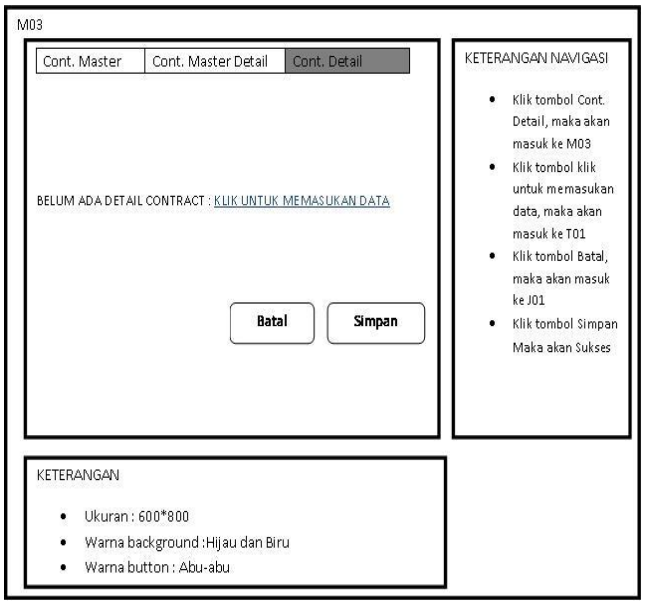
Perancangan ini menjelaskan tentang data *Contract Master Detail*, yang akan menyewa sperpat pesawat.



Gambar I.15 Perancangan antar muka data cont. *Master Detail*

5. Perancangan antar muka cont.*Detail*

Perancangan ini menjelaskan tentang *Contract Detail* , yang akan menyewa sperpat pesawat dan menambahkan bila ada kekurangan.



Gambar I.16 Perancangan antar muka cont.Detail

1.6.4 Implementasi

Setelah sistem dianalisis dan didesain secara rinci maka selanjutnya akan menuju tahap implementasi. Tujuan implementasi adalah untuk menjelaskan tentang manual modul kepada semua user yang akan menggunakan sistem. Sehingga user tersebut dapat merespon apa yang ditampilkan di sistem dan memberikan masukan kepada pembuat sistem untuk dilakukan perbaikan agar sistem lebih baik lagi.

1.6.4.1 Implementasi Basis Data

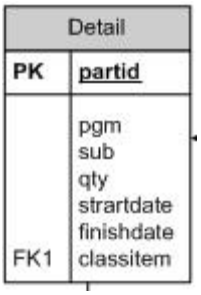
Database Management Sistem (DBMS) yang digunakan untuk mengimplementasikan data modul *Sales Contract* di PT. Dirgantara Indonesia ini

adalah PostgreSQL 8.4. Implementasi *database* dalam aplikasi yang telah dirancang adalah sebagai berikut :

1. **Tabel Detail**

Berikut *syntax* hasil *generate* tabel Detail ke dalam *database* MySQL :

Tabel I.13 Tabel *Detail*

Tabel	***Syntax* MySQL**
	CREATE TABLE `detail` (`partid` int(10) unsigned NOT NULL, `pgm` varchar(19) NOT NULL, `sub` varchar(80) NOT NULL, `qty` varchar(19) NOT NULL, `startdate` varchar(20) DEFAULT NULL, `finishdate` varchar(19) NOT NULL, PRIMARY KEY (`partid`) KEY 'classitem' FOREIGN KEY (`classitem`) REFERENCES `Master` (`classitem') WITH (OIDS=FALSE); ALTER TABLE irp.tmcontritem OWNER TO postgres;

2. **Tabel Master**

Berikut *syntax* hasil *generate* tabel Mater ke dalam *database* MySQL :

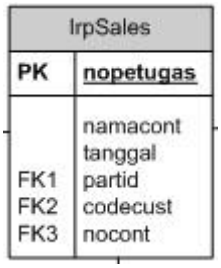
Tabel I.14 Tabel *Master*

Tabel	***Syntax* MySQL**
	CREATE TABLE `Master` (`classitem` int(10) unsigned NOT NULL, `partid` varchar(19) NOT NULL, `unitprice` varchar(80) NOT NULL, `itemvalue` varchar(19) NOT NULL, PRIMARY KEY (`partid`) WITH (OIDS=FALSE); ALTER TABLE irp.tmcontritem OWNER TO postgres;

3. Tabel IrpSales

Berikut *syntax* hasil *generate* tabel IrpSales ke dalam *database* MySQL :

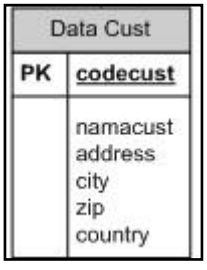
Tabel I.15 Tabel IrpSales

Tabel	*Syntax* MySQL
	CREATE TABLE `IrpSales` (`nopetugas` int(10) unsigned NOT NULL, `namacount` varchar(19) NOT NULL, `tanggal` varchar(80) NOT NULL, `itemvalue` varchar(19) NOT NULL, PRIMARY KEY (`nopetugas`) KEY `partid` FOREIGN KEY (`partid`) REFERENCES `detail` (`partid`), KEY 'codecust' FOREIGN KEY (`codecust`) REFERENCES `Datacust` (`codecust`), KEY 'nocont' FOREIGN KEY (`nocont`) REFERENCES `Masteretil` (`nocont`) WITH (OIDS=FALSE); ALTER TABLE irp.tmcontritem OWNER TO postgres;

4. Tabel Data Cust

Berikut *syntax* hasil *generate* tabel Data Cust ke dalam *database* MySQL :

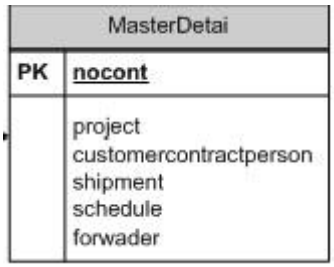
Tabel I.16 Tabel Data Cust

Tabel	*Syntax* MySQL
	CREATE TABLE `DataCust` (`codecust` int(10) unsigned NOT NULL, `namacust` varchar(19) NOT NULL, `address` varchar(80) NOT NULL, `city` varchar(19) NOT NULL, `zip` varchar(19) NOT NULL, `country` varchar(19) NOT NULL, PRIMARY KEY (`codecust`) WITH (OIDS=FALSE); ALTER TABLE irp.tmcontritem OWNER TO postgres;

5. Tabel Master Detail

Berikut *syntax* hasil *generate* tabel Master Detail ke dalam *database* MySQL :

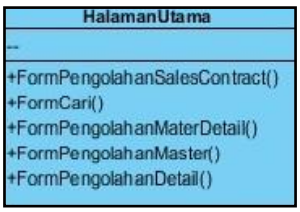
Tabel I.17 Tabel *Master Detail*

Tabel	***Syntax* MySQL**
	CREATE TABLE `MasterDetail` (`nocont` int(10) unsigned NOT NULL, `project` varchar(19) NOT NULL, `customercontacperson` varchar(80) NOT NULL, `shipment` varchar(19) NOT NULL, `schedule` varchar(19) NOT NULL, `forwader` varchar(19) NOT NULL, PRIMARY KEY (`nocont`) WITH (OIDS=FALSE); ALTER TABLE irp.tmcontritem OWNER TO postgres;

1.6.4.2 Implementasi Kelas

1. Kelas Halaman Utama

Kelas *form* utama merupakan kelas yang digunakan untuk menghubungkan semua kelas *form* yang ada di sales *Contract*.



Gambar I.17 Kelas Halaman Utama

A. Deskripsi Atribut

Tidak memilik atribut

B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. FormPengolahanSales*Contract*(), *method* untuk menghubungkan *form* pengolahan sales *contract* dengan *form* utama.
2. FormPengolahanCari(), *method* untuk menghubungkan *form* pengolahan Cari dengan *form* utama.
3. FormPengolahan*MasterDetail*(), *method* untuk menghubungkan *form* pengolahan *Master Detail* dengan *form* utama.
4. FormPengolahan *Master* (), *method* untuk menghubungkan *form* pengolahan *Master* dengan *form* utama.
5. FormPengolahan*Detail* (), *method* untuk menghubungkan *form* pengolahan *Detail* dengan *form* utama.

2. Kelas *Form* Pengolahan Sales *Contract*

Kelas *Form* Pengolahan Sales *Contract* merupakan kelas *interface* untuk pengolahan data sales *Contract*.



Gambar I.18 Kelas Form Pengolahan Sales *Contract*

A. Deskripsi Atribut

Tidak memilik atribut

B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. TombolCariCust(), *interface* untuk tombol cari cust
2. TombolCaricont(), *interface* untuk tombol Cari Cont

3. Kelas Sales *Controller*

Kelas *Form* Pengolahan Sales *Controller* merupakan kelas *interface* untuk pengolahan data sales *Controller*.



Gambar I.19 Kelas Sales *Controller*

A. Deskripsi Atribut

Tabel I.18 Deskripsi Atribut Kelas Sales Controller

Nama	Tipe	Panjang	Range	Nilai Tetap
FormSales*Contract*	int	-	-	-

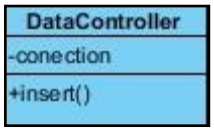
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. CariCust(), *interface* untuk tombol cari cust
2. Tambah(),*interface* untuk tombol cari cust

4. Kelas Data *Controller*

Kelas Data *Controller* merupakan kelas *interface* untuk pengolahan data *Controller*



Gambar I.20 Kelas Data *Controller*

A. Deskripsi Atribut

Tabel I.19 Deskripsi Atribut Kelas Data *Controller*

Nama	Tipe	Panjang	Range	Nilai Tetap
Connection	-	-	-	-

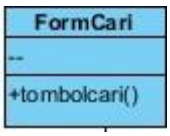
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. *Insert*(), *interface* untuk tombol *Insert*

5. Kelas *Form* Cari

Kelas *Form* Cari merupakan kelas *interface* untuk pengolahan data Cari



Gambar I.21 Kelas Form Cari

A. Deskripsi Atribut

Tidak memilik atribut

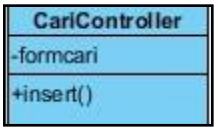
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. TombolCari(), *interface* untuk tombol cari

6. Kelas Cari *controller*

Kelas *Form* Cari ***controller*** merupakan kelas *interface* untuk pengolahan data Cari *controller*



Gambar I.22 Kelas Cari *Controller*

A. Deskripsi Atribut

Tabel I.20 Deskripsi Atribut Kelas Cari *Controller*

Nama	Tipe	Panjang	Range	Nilai Tetap
formcari	-	-	-	-

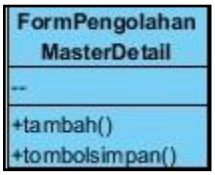
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. *Insert*(), *interface* untuk tombol *Insert*

7. Kelas *form* Pengolahan *Master Detail*

Kelas *form* Pengolahan *Master Detail* merupakan kelas *interface* untuk pengolahan data *Master Detail*



Gambar I.23 Kelas form Pengolahan *Master Detail*

A. Deskripsi Atribut

Tidak memilik atribut

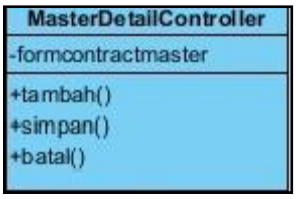
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. Tomboltambah(), *interface* untuk memanggil tombol tambah
2. Tombolsimpan(), *interface* untuk menyimpan data

8. Kelas *Master detail Controller*

Kelas *Master detail Controller* merupakan kelas *interface* untuk pengolahan data *Master Detail Controller*



Gambar I.24 Kelas Master detail Controller

A. Deskripsi Atribut

Tabel I.21 Deskripsi Atribut Kelas *Master Detail Controller*

Nama	Tipe	Panjang	Range	Nilai Tetap
Formcontractmaster	-	-	-	-

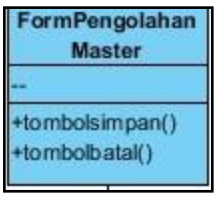
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. Tomboltambah(), *interface* untuk memanggil tombol tambah
2. Tombolsimpan(), *interface* untuk menyimpan data
3. Tombolbatal(), *interface* untuk tombol membatalkan data

9. Kelas *Form* Pengolahan *Master*

Kelas *Form* Pengolahan *Master* merupakan kelas *interface* untuk pengolahan data *Master*



Gambar I.25 Kelas Form Pengolahan *Master*

A. Deskripsi Atribut

Tidak memilik atribut

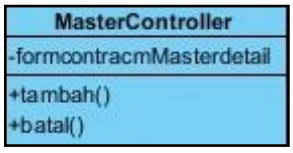
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. Tombolsimpan(), *interface* untuk menyimpan data
2. Tombolbatal(), *interface* untuk tombol membatalkan data

10. Kelas *Master Controller*

Kelas *Form Master controller* merupakan kelas *interface* untuk pengolahan data *Master controlerr*



Gambar I.26 Kelas *Master Controller*

A. Deskripsi Atribut

Tabel I.22 Deskripsi Atribut Kelas *Master Controller*

Nama	Tipe	Panjang	Range	Nilai Tetap
Formcontractmasterdetail	-	-	-	-

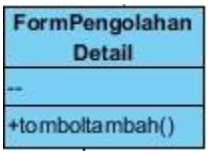
B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. Tomboltambah(), *interface* untuk memanggil tombol tambah
2. Tombolbatal(), *interface* untuk tombol membatalkan data

11. Kelas *Form* Pengolahan *Detail*

Kelas *Form* pengolahan *Detail* merupakan kelas *interface* untuk pengolahan data *detail*



Gambar I.27 Kelas Form Pengolahan Detail

A. Deskripsi Atribut

Tidak memilik atribut

B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. TombolTambah(), *interface* untuk menmbahkan Data

12. Kelas *Detail Controller*

Kelas *Form* pengolahan *Detail controlerr* merupakan kelas *interface* untuk pengolahan data *detail controlerr*



Gambar I.28 Kelas *Detail Controller*

A. Deskripsi Atribut

Tabel I.23 Deskripsi Atribut Kelas *Detail Controller*

Nama	Tipe	Panjang	Range	Nilai Tetap
Formmasterdetail	-	-	-	-

B. Deskripsi Layanan

Operasi – operasi yang ada pada kelas ini adalah :

1. Tomboltambah(), *interface* untuk memanggil tombol tambah

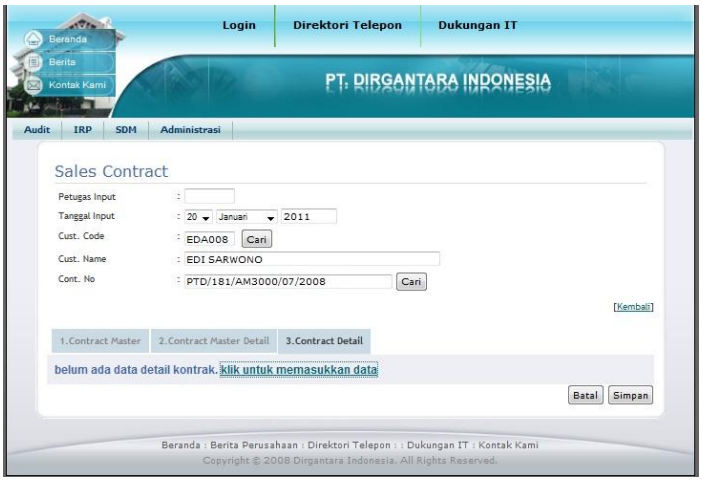
1.6.4.3 Implementasi Antarmuka Pemakai

Berikut adalah antarmuka yang terdapat dalam Modul Sales *Contract* ini:

1. Impelmentasi antar muka sales

   Dalam Header ini admin diminta untuk memasukan atau mencari *Customer* kode. Apabila no *Customer* ditemukan maka secara otomatis pada tab *Cust*. Name akan meng-generate nama *customer*. Jika *costumer* tersebut telah memiliki kontrak yang sedang berjalan maka pada tab *Cont*. No, akan memunculkan no kontrak yang sedang

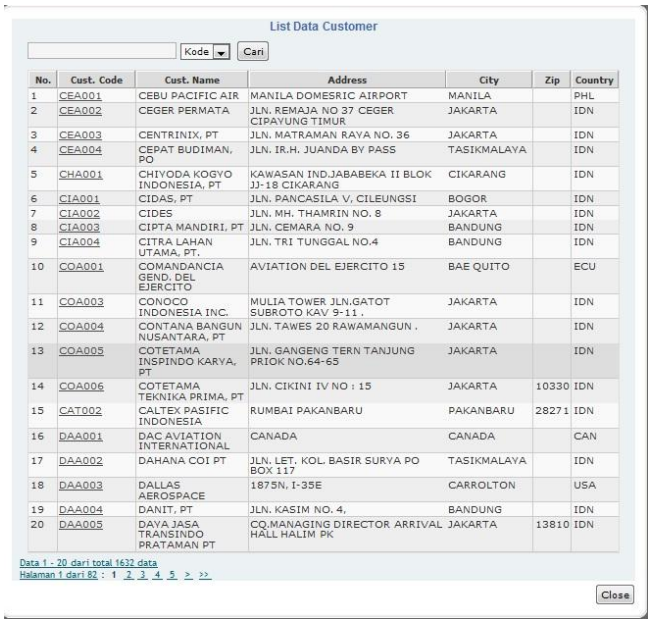
dilaksanakan. Tetapi apabila tidak ada kontrak, maka tab tersebut akan kosong.



Gambar I.29 Implementasi antar muka sales

2. Implementasi antar muka data *customer*/ *cust*.kode

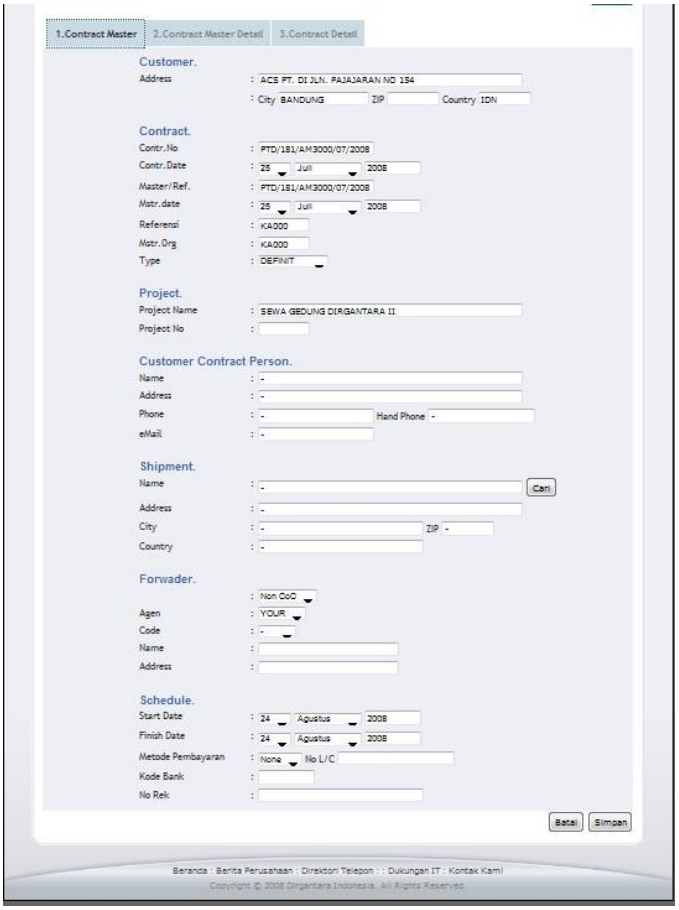
Implementasi ini menjelaskan tentang data-data list data *customer*, yang ada di form sales *Contract*.



Gambar I.30 Implementasi antar muka data *customer*/ cust.kode

3. Implementasi antar muka *cont*. *Master*

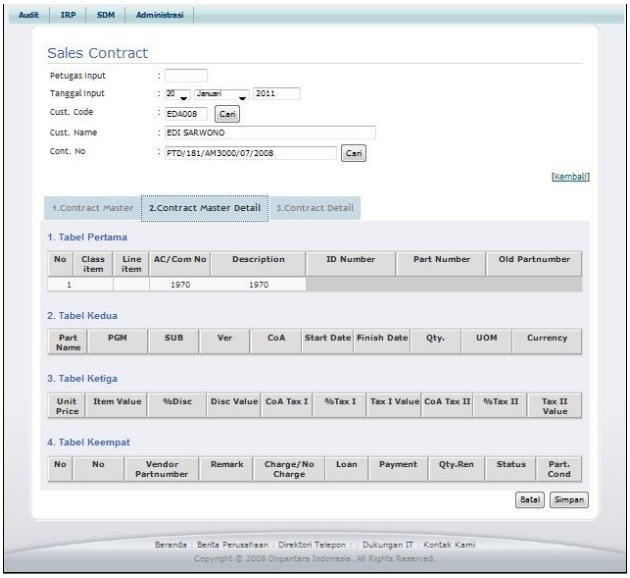
Dalam *Contract Master* ini terdapat field-field yang harus diisi oleh admirp, seperti tanggal dimulai dan berakhirnya kontrak, setelah kontrak disepakati.



Gambar I.31 Implementasi antar muka cont. *Master*

4. Implementasi antar muka data *cont.Master Detail*

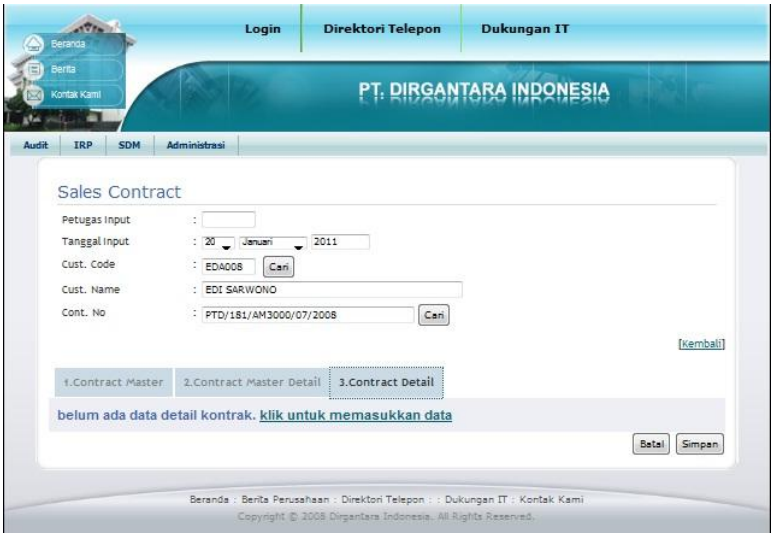
Pada bagian kontrak *Master Detail* akan ditampilkan beberapa tabel yang nantinya merefer kepada kontrak *Master* sebelumnya. Desini ditampilkan lebih rinci mengenai produk yang akan disepakati dalam kontrak.



Gambar I.32 Implementasi antar muka data cont.Master Detail

5. Implementasi antar muka *cont*.*Detail*

Pada bagian kontrak *Detail* akan menampilkan beberapa tabel yang nantinya merefer kepada kontrak *Master Detail* sebelumnya.



Gambar I.33 Implementasi antar muka cont.Detail