

**PERANCANGAN APLIKASI SMS GATEWAY UNTUK  
PEMESANAN BARANG PADA PT.WAJAH BARU  
ELEKTRONIK**

**Laporan Tugas Akhir**

Diajukan untuk melengkapi persyaratan  
mencapai gelar sarjana strata satu (S1)  
di Program Studi Sistem Informasi

Disusun oleh :

**DADI PRIYANTO**

**4180401-027**



**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MERCU BUANA**

**JAKARTA**

**2008**

## DAFTAR ISI

LEMBAR PENGESAHAN.....	i
LEMBAR PERNYATAAN.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penulisan.....	2
1.3 Batasan Masalah.....	2
1.4 Ruang Lingkup.....	3
1.5 Metode Penulisan.....	3
1.6 Sistematika penulisan.....	4
BAB II LANDASAN TEORI.....	6
2.1 SMS ( <i>Short Message service</i> ).....	6
2.1.1 Karakteristik SMS.....	7
2.1.2 Keuntungan SMS.....	8
2.1.3 Cara Kerja SMS.....	8

2.2 AT Command.....	10
2.3 SMS Center.....	12
2.3.1 Koneksi Ke SMSC.....	12
2.3.2 Koneksi Langsung Ke SMSC.....	13
2.4 Layanan Aplikasi SMS.....	15
2.5 Pendekatan Pengembangan Dengan Metode Waterfall.....	16
2.6 Teory Perancangan Antarmuka UML.....	18
2.6.1 Pengertian Use Case.....	20
2.6.2 Notasi Use Case.....	21
2.6.3 Relasi Use Case.....	22
2.6.4 Diagram Aktivitas (Activity Diagram).....	22
2.6.5 Sequence Diagram (Diagram Sequence).....	22
2.7 Class Diagram.....	24
2.8 Entity Relationship Diagram.....	25
2.9 Normalisasi.....	29
2.10 Bahasa Pemrograman Java (Jdk1.5_08).....	31
2.10.1 Tool Utama dari Java 2 SDK 1.5.....	32
2.11 Database MySQL.....	33
<b>BAB III ANALISA DAN PERANCANGAN SISTEM.....</b>	<b>35</b>
3.1 Permasalahan Dan Strategi Pemecahan Masalah .....	34
3.1.1 Pemecahan Masalah.....	36
3.2 Rancangan Aplikasi .....	36
3.2.1 Analisa Aplikasi Usulan .....	36

3.2.2 Metode Kerja Sistem Usulan.....	38
3.3 Diagram Use Case .....	41
3.4 Diagram Activity .....	43
3.5 Diagram Sequence.....	45
3.6 Class Diagram.....	50
3.7 Perancangan DataBase.....	51
3.7.1 Entity Relationship.....	51
3.7.2 Normalisasi.....	52
3.7 Rancangan Tabel.....	55
3.8 Rancangan Layar.....	59
BAB IV IMPLEMENTASI DAN PENGUJIAN.....	67
4.1 Cara pengoperasian.....	67
4.2 Tampilan Layar Menu Utama.....	67
4.3 Pemrosesan pesan dan Tampilan Pada Handphone.....	74
4.4 Metode Pengujian.....	80
4.4.1 Skenario Pengujian .....	81
4.4.2 Hasil Pengujian.....	83
4.4.3 Analisis Hasil Pengujian.....	84
BAB V PENUTUP.....	86
5.1 Kesimpulan.....	86
5.2 Saran.....	86

DAFTAR PUSTAKA.....	88
Lampiran A Surat Keterangan Riset.....	89
Lampiran B Listing Program.....	L1

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dengan meningkatnya mobilitas dan dinamika masyarakat, semakin dirasakan kebutuhan komunikasi yang tidak bergantung tempat, waktu dan keadaan. Maka timbul pemikiran untuk mengembangkan konsep tentang teknologi bergerak. Teknologi ini telah menciptakan kondisi bahwa komunikasi tidak lagi diasosiasikan dengan lokasi (perangkat), tetapi diasosiasikan dengan personal, yang bisa berada dimana saja di wilayah layanan. Seiring perkembangannya sekarang ini teknologi telepon bergerak tidak hanya digunakan untuk berkomunikasi suara dengan lawan bicara, tetapi juga dapat digunakan untuk saling berkiriman pesan dengan orang lain melalui teks atau yang lebih populer disebut dengan SMS.

Media SMS sekarang ini sedang marak digunakan oleh para pengguna telepon bergerak dikarenakan biaya yang dikeluarkan murah dan tingkat kegagalan penerimaannya rendah. Maraknya penggunaan SMS sekarang ini, membuat berkembangnya layanan-layanan berbasis SMS, salah contohnya adalah pemesanan barang elektronik. pengoperasiannya pun cukup mudah misalnya untuk mengetahui pesanan barang yang kita pesan, apa yang ada di toko tersebut, cukup ketikkan nama barang yang dipesan, kemudian mengirimkannya ke nomor tertentu, maka tidak berselang lama SMS yang berisi informasi yang diminta akan diperoleh.

Tidak terkecuali, dalam hal pemesanan barang yang dilakukan pelanggan kepada sebuah toko tersebut. Aktivitas seorang Pelanggan yang tidak hanya di satu tempat, mengakibatkan sulitnya dalam pemesanan barang. Jika jumlah pelanggan meningkat, menandakan bahwa toko tersebut memiliki kualitas yang baik di masyarakat dan dapat dipercaya dapat memuaskan pelanggan. Dengan menggunakan pemanfaatan media SMS, maka aplikasi ini diharapkan dapat membantu untuk pelanggan mendapatkan barang di sebuah toko.

Berdasarkan hal-hal yang telah dipaparkan diatas maka penulis mengajukan dengan judul perancangan aplikasi sms gateway untuk pemesanan barang pada PT. wajah baru elektronik.

## **1.2 .Tujuan Penulisan**

Tujuan dari penulisan tugas akhir ini adalah:

1. Mempublikasikan PT. Wajah Baru Elektronik kepada masyarakat
2. Merancang sebuah aplikasi yang mempercepat dalam mendapatkan barang yang diinginkan & mempermudah pelanggan dalam pemesanan barang dengan fasilitas sms, terutama pelanggan yang telah terdaftar sebagai anggota.

## **I.3. Batasan Masalah**

Dengan banyaknya hal yang termasuk di dalam pemesanan barang maka penulis membatasi skripsi ini pada :

1. Server pemesanan barang, *database* PT. Wajah Baru Elektronik. Selain itu, untuk penghitungan biaya barang telah dimiliki oleh toko itu. Semua pesanan dapat di lakukan oleh pelanggan yang telah terdaftar pada *database* Wajah Baru elektronik. Kemudian sistem ini akan mengirimkan balasan berupa balasan SMS

2. Aplikasi SMS ini bekerja dalam lingkungan sistem operasi Windows dengan bahasa Pemograman Java jdk1.5.0\_08 dan data base *Database* Mysql 4.13, sedangkan pengiriman dan penerimaan SMS oleh pelanggan akan dilakukan oleh operator telekomunikasi.
3. Arsitektur ini menggunakan tiga hardware yaitu PC/Laptop (SMS Server), *handphone* sebagai terminal, operator Indosat sebagai SMS center (SMSC), *Bluetooth* serta dibutuhkan juga *handphone* user sebagai pengirim.

#### **I.4 Ruang Lingkup**

1. Pelayanan yang diberikan hanya satu kota saja, misalnya hanya melayani kota tangerang dan sekitarnya.
2. Pelayanan operator yang dapat dilakukan adalah saat ini hanya *GSM* saja .
3. Pelayanan pemesanan barang akan diberikan kepada pelanggan yang telah melakukan registrasi.

#### **1.5 Metode Penulisan**

Beberapa metode penelitian yang digunakan untuk pengumpulan data dan informasi-informasi pelengkap untuk mempermudah analisa dan perancangan aplikasi SMS ini, antara lain:

1. Metode Kepustakaan (*Library Research*)

Metode ini dilakukan untuk mengumpulkan data dan informasi melalui buku-buku atau Sumber-sumber bacaan lainnya serta beberapa yang berkaitan dengan penyusunan skripsi dan aplikasi yang sedang dianalisa dan dirancang.

2. Metode Wawancara



Dengan cara terjun langsung ke lapangan dan melakukan wawancara dengan pihak yang melakukan pengelolaan data pemesanan barang di Wajah baru elektronik untuk mengetahui permasalahan yang dihadapi dan mendiskusikan aplikasi yang di harapkan dapat memecahkan permasalahan tersebut

### 3. Metode model rekayasa perangkat lunak

Metode yang digunakan adalah model waterfall, karena metode ini membutuhkan pendekatan sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dari tingkat sistem & kemajuan melalui analisis, desain, coding, testing dan pemeliharaan

## **1.6 Sistematika Penulisan**

Sistematika penulisan tugas akhir ini agar dapat terlihat jelas terkaitannya antara bab yang satu dengan bab lainnya, maka penulisannya disusun secara sistematis yang terdiri dari lima bab yaitu :

### **BAB I : PENDAHULUAN**

Dalam Bab ini dibahas mengenai latar belakang pembuatan skripsi, tujuan penulisan, batasan-batasan dari masalah yang dibahas, ruang lingkup, metode penulisan dan sistematika penulisan skripsi.

### **BAB II : LANDASAN TEORI**

Bab ini membahas tentang teori-teori dasar yang berkaitan dengan pembuatan aplikasi ini, mulai dari teori yang bersifat umum sampai teori yang membahas tentang perangkat lunak yang digunakan untuk pembuatan aplikasi SMS.

### **BAB III : ANALISA DAN PERANCANGAN SISTEM**

Dalam Bab ini dibahas tentang mengidentifikasi masalah dan pemecahannya berupa sistem yang diusulkan yaitu mengenai bagaimana aplikasi ini dikembangkan, analisa program yang dirancang beserta cara kerja dan *feature* yang ada, serta perancangan tampilan layar.

### **BAB IV : IMPLEMENTASI DAN PENGUJIAN**

Dalam Bab ini pembahasannya adalah mengenai implementasi, cara kerja program yang dikembangkan, pengujian terhadap sistem yang diusulkan serta pengembangan lebih lanjut dari program yang ada.

### **BAB V : PENUTUP**

Bab ini menuliskan mengenai kesimpulan yang didapat dari hasil analisa bab-bab sebelumnya serta saran yang dapat berguna bagi pengembangan aplikasi ini selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 SMS (Short Message Service)**

*Short Message Service (SMS)* merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan atau antara terminal pelanggan dengan sistem eksternal seperti email, voice mail, dan lain-lain.

SMS didukung oleh GSM (*Global System For Mobile Communication*), TDMA (*Time Division Multiple Access*), CDMA (*Code Division Multiple Access*) yang berbasis pada telepon seluler yang saat ini banyak digunakan. Karena layanan berbasis SMS telah banyak digunakan selama bertahun-tahun, maka penetrasi telepon seluler dan user yang banyak menggunakan layanan lain sudah membuat layanan berbasis SMS menjadi sebuah layanan yang menarik. Sekarang ini otomatis dalam layanan ini dimanfaatkan untuk berkomunikasi secara singkat pada para pelanggan suatu perusahaan.

SMS menjadi fenomena tersendiri, dalam waktu yang cukup singkat, tingkat pertumbuhannya sangat tinggi tanpa ada penurunan tarif yang berarti, bahkan dapat dikatakan tarifnya mengambil posisi *steady state*. Biasanya, bahkan dalam kasus layanan telepon bergerak, tarif akan turun seiring dengan meningkatnya penggunaan. Fakta lainnya adalah fasilitas SMS dalam telepon bergerak ternyata punya andil cukup, dalam menarik kaum muda masuk ke pasar telepon bergerak. Hal inilah yang meningkatkan perubahan SMS dimana dari sekedar untuk mentransfer data berupa teks, kemudian dikenal dengan ditambahi suara, SMS semacam ini dikenal dengan

nama EMS (*Enhanced Message Service*) dan bahkan yang lebih maju lagi, yaitu MMS (*Multimedia Message Service*) yang memungkinkan pengiriman tidak sekedar berupa teks dan suara, namun juga foto, gambar statis dan bahkan video.

Tidak diragukan lagi SMS sangat sukses dipasaran, di tempat kelahirannya sendiri, yaitu Eropa, trafik SMS mencapai lebih dari 3 miliar *message* per bulan meskipun tanpa ada program marketing yang proaktif dari operator seluler dan vendor pembuat perangkat komunikasi bergerak. Kesuksesan SMS dianggap sebagai kesuksesan yang tidak sengaja dan cukup mengejutkan bagi pihak-pihak yang terjun dalam industri telekomunikasi bergerak karena beberapa pihak yang berkompeten sebelumnya memprediksi bahwa SMS tidak akan laku karena penggunaannya cukup sulit dan materi untuk marketingnya sulit.

### **2.1.1 Karakteristik SMS**

Karakteristik utama SMS adalah SMS merupakan sebuah sistem pengiriman data dalam paket yang bersifat *out-of-band* dengan *bandwidth* kecil. Dengan karakteristik ini, pengiriman suatu *burst* data yang sangat pendek dapat dilakukan dengan efisiensi yang sangat tinggi.

Layanan SMS merupakan sebuah layanan yang bersifat *nonreal time* dimana sebuah *short message* dapat di-submit ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu *short message* hingga sampai ke tujuan. Kegagalan pengiriman yang bersifat sementara seperti tujuan tidak aktif akan selalu teridentifikasi sehingga pengiriman ulang *short message* akan selalu dilakukan

kecuali bila diberlakukan aturan bahwa *short message* yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

### **2.1.2 Keuntungan SMS**

Pada tingkat minimum keuntungan yang dapat diberikan oleh SMS bagi pemakai meliputi pengiriman notifikasi dan peringatan (*alert*), penyampaian pesan SMS yang terjamin, handal, mekanisme komunikasi dengan biaya rendah, kemampuan untuk menyaring pesan SMS dan menanggapi panggilan secara selektif sehingga meningkatnya produktifitas pelanggan.

Untuk fungsionalitas yang lebih canggih, SMS memberikan beberapa keuntungan tambahan bagi pelanggan yaitu pengiriman pesan SMS ke beberapa pelanggan sekaligus dalam waktu yang bersamaan, kemampuan menerima informasi yang beragam, dan integrasi dengan aplikasi lain yang berbasis internet dan data.

### **2.1.3 Cara kerja SMS**

Dalam sistem SMS, mekanisme utama yang dilakukan dalam suatu sistem adalah melakukan pengiriman *short message* dari satu terminal pelanggan ke terminal yang lain. Hal ini dapat dilakukan berkat adanya sebuah entitas dalam sistem SMS yang bernama *Short Message Service Center* (SMSC), disebut juga *Message Center* (MC). Pada saat pesan SMS dikirim dari *handphone* (*mobile orginated*) pesan tersebut tidak langsung dikirim ke *handphone* tujuan (*mobile terminated*), akan tetapi terlebih dahulu ke SMSC, baru kemudian pesan tersebut dikirimkan ke *handphone* tujuan.

SMSC merupakan sebuah perangkat yang melakukan tugas *store and forward* trafik *short message*. Di dalamnya termasuk penentuan atau pencarian rute tujuan.

akhir dari *short message*. Sebuah SMSC biasanya didesain untuk dapat menangani *short message* dari berbagai sumber seperti *Voice Mail System* (VMS), Web-based messaging, *Email Integration*, *External Short Message Entities* (ESME), dan lain-lain.

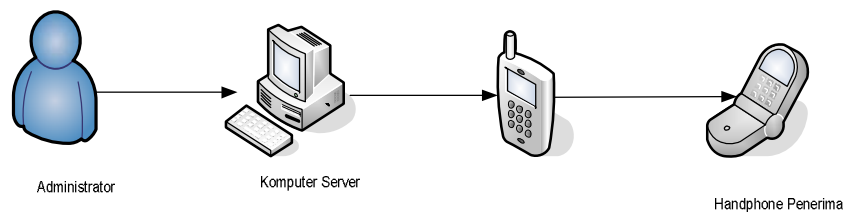
Untuk contoh sederhananya, pada saat kita mengirim pesan SMS dari *handphone*, pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan, akan tetapi dikirim terlebih dahulu ke SMSC, baru kemudian pesan tersebut diteruskan ke *handphone* tujuan. Dengan adanya SMSC ini kita dapat mengetahui status dari pesan SMS yang telah dikirim, apakah telah sampai atau gagal diterima oleh *handphone* tujuan. Apabila *handphone* tujuan dalam keadaan aktif dan dapat menerima SMS yang dikirim, ia akan mengirimkan kembali pesan konfirmasi ke SMSC yang menyatakan bahwa pesan telah diterima, kemudian SMSC mengirimkan kembali status tersebut ke si pengirim. Jika *handphone* tujuan dalam keadaan mati, pesan yang kita kirimkan akan disimpan pada SMSC sampai *period-validity* terpenuhi. Berikut ini adalah daftar SMSC pada beberapa operator GSM di Indonesia:

Tabel 2.1: Daftar SMSC di indonesia

<b>Operator GSM</b>	<b>Nomor SMSC</b>
Satelindo	62816124
Excelcomindo	62818445009
Telkomsel	6281100000
IM3	62855000000

Terminologi SMSC mengacu pada sesuatu yang berupa *hardware* dan *software*. Sebuah SMSC harus memiliki keandalan yang tinggi, kapasitas yang cukup, dan *throughput* yang memadai dalam trafik *short message*. Selain itu, sistem harus bersifat fleksibel dan *scalable* agar dapat mengakomodasi pertumbuhan permintaan layanan SMS. Faktor lain yang juga harus diperhatikan adalah aplikasi harus dapat dioperasikan dengan mudah, begitu juga pemeliharannya. Sebagai contoh adalah fleksibilitas untuk aktivasi layanan baru dan *upgrade software*.

Sedangkan tahapan-tahapan yang dilakukan pada aplikasi *broadcast* pemesanan barang melalui SMS adalah hampir sama seperti aplikasi SMS *Sever* hanya saja *user* tidak meminta atau mengirimkan pesan dalam format tertentu. Administrator akan melihat data pelanggan atau pimpinan di *database*. Setelah diketahui data mana yang ingin digunakan, admin tinggal mengirimkan ke nomor tujuan. Untuk lebih jelasnya dapat dilihat pada gambar berikut di bawah ini.



**Gambar 2.1** Skema Pengiriman Broadcast

([www.pengirimanbroadcast.com](http://www.pengirimanbroadcast.com))

## 2.2 AT Command

Dibalik teks SMS yang diterima dan dikirim pada sebuah telepon seluler sebenarnya adalah berupa perintah AT Command yang bertugas mengirim atau menerima data dari dan ke SMS *Center*.

Perintah AT Command tiap-tiap SMS device bisa berbeda-beda, setiap vendor biasanya memberikan referensi tentang daftar perintah AT yang tersedia atau bisa di

download di internet. AT Command digunakan untuk berkomunikasi dengan terminal melalui serial port pada komputer. Dengan menggunakan perintah AT, kita dapat mengetahui kekuatan sinyal dari terminal, mengirim pesan, menambahkan item pada buku alamat, mematikan terminal dan banyak fungsi lainnya. Salah satu software yang digunakan untuk mengetes perintah AT Command adalah windows HyperTerminal yang biasanya telah tersedia bersama windows installer, sehingga hanya perlu menambahkan software tersebut dari control panel.

Tidak semua perintah AT digunakan pada program, yang diambil hanya yang diperlukan saja, misal untuk mengirim, membaca, menghapus dan menerima pesan dari terminal. AT command yang umum digunakan adalah:

Tabel 2.2 : Tabel AT *Command*

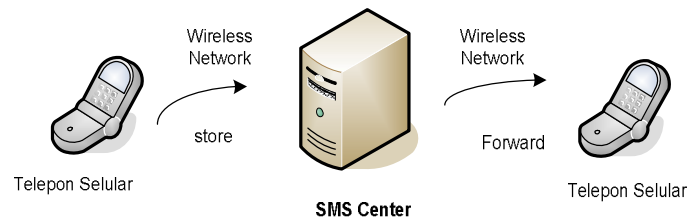
Perintah AT	Kegunaan
AT	Mengecek apakah <i>handphone</i> telah terhubung
AT+CMGF	Menetapkan format mode dan terminal
AT+CSCS	Menetapkan jenis encoding
AT+CNMI	Mendeteksi pesan SMS baru masuk secara otomatis
AT+CMGL	Membuka daftar SMS yang ada pada SIM Card
AT+CMGS	Mengirim Pesan SMS
AT+CMGR	Membaca Pesan SMS
AT+CMGD	Menghapus Pesan SMS

AT Command sebenarnya hampir sama dengan perintah ( prompt ) pada DOS. Perintah-perintah yang dimasukkan ke port dimulai dengan kata AT, lalu diikuti oleh karakter lainnya, yang memiliki fungsi unik. Contoh : AT digunakan untuk menanyakan status port. Output “OK” akan tampak dilayar jika kondisi port tersebut siap untuk berkomunikasi.



### 2.3 SMS Center ( SMSC )

Pada saat kita mengirim pesan SMS dari *handphone* (mobile originated) pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan ( mobile terminated ), akan tetapi dikirim terlebih dahulu ke SMS Center (SMSC), baru kemudian pesan tersebut diteruskan ke *handphone* tujuan. Gambar 2.2 dibawah ini merupakan skema cara kerja SMS.



**Gambar 2.2** Skema Cara Kerja SMS

([www.schema.cara.kerja.SMS.com](http://www.schema.cara.kerja.SMS.com))

Dengan adanya SMSC ini kita dapat mengetahui status dari pesan SMS yang dikirim, apakah telah sampai atau gagal diterima oleh *handphone* tujuan. Apabila *handphone* tujuan dalam keadaan aktif dan dapat menerima pesan SMS yang dikirim, *handphone* tersebut akan mengirimkan kembali pesan konfirmasi ke SMSC yang menyatakan bahwa pesan telah diterima. Kemudian SMSC mengirimkannya kembali status tersebut kepada si pengirim. Jika *handphone* tujuan dalam keadaan mati, pesan yang kita kirimkan akan disimpan pada SMSC sampai period-validity terpenuhi.

#### 2.3.1 Koneksi ke SMSC

untuk dapat mengirim dan menerima pesan, kita harus melakukan koneksi ke SMSC. Ada beberapa cara untuk melakukan koneksi ke SMSC antara lain :

- a) Menggunakan terminal baik berupa GSM modem atau *handphone*.

Cara ini adalah yang paling mudah tetapi memiliki kekurangan antara lain jumlah pesan yang dikirim per menit sangat terbatas (sekitar 6-10 pesan per menit). Untuk mengantisipasi hal ini biasanya digunakan lebih dari satu terminal.

### 2.3.2 Koneksi langsung ke SMSC

Dengan melakukan koneksi langsung ke SMSC kita dapat mengirim pesan dalam jumlah banyak, dapat mencapai sekitar 600 sms per menit bergantung pada kapasitas dari SMSC itu sendiri. Pemilihan koneksi ke SMSC biasanya disesuaikan dengan jumlah pesan SMS yang akan dikirim.

Tabel 2.3 : User Data

Nilai	Decimal	Septet (7 bit)	Oktet (8 bit)	Hasil
S	83	1010011	<b>1</b> 1010011	D3
A	97	1100001	<b>01</b> 110000	70
Y	121	1111001	<b>001</b> 111110	3E
A	97	1100001	<b>0000</b> 1100	0C

Dari penjelasan di atas di peroleh hasil unuk pengiriman SMS dalam format PDU untuk contoh tersebut adalah :

0010000D812618139070620000AA0AD3703E0C.

a) User Data Length (UDL)

*User Data Length* adalah panjang dari pesan yang diterima dalam bentuk teks standard. Pada contoh di atas nilai dari UDL adalah 0A, yang berarti pesan yang diterima adalah sebanyak 10 karakter.

b) User Data (UD)

*User Data* adalah pesan yang diterima dalam format *heksadecimal*. Pada contoh di atas nilainya adalah D3703E0C1AA6D9F2F0. pengkodean dari nilai *heksadecimal* menjadi teks standard dengan bantuan tabel kode ASCII (Tabel 2.16 ) dapat dilihat pada tabel di bawah ini :

Tabel 2.4 : User Data

Nilai	Oktet (8 bit)	Septet (7 bit)	Decimal	Hasil
D3	11010011	1010011	83	S
70	01110000	1100001	97	a
3E	00111110	1111001	121	y
0C	00001100	1100001	97	a

Dari tabel di atas terlihat bahwa nilai heksadesimal dari D3703E0C adalah “Saya”. Ini berarti pesan yang diterima adalah “Saya”

Tabel 2.5 : Kode ASCII

Decimal	0	1	2	3	4	5	6	7	8	9
0										
1	LF			CR						
2										
3			SP	!	“	#	\$	%	&	`
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	`	a	b	C
10	d	E	f	g	h	i	j	k	L	M
11	n	O	p	q	r	s	t	u	v	W
12	x	Y	z	{		}	~	DEL		

Tabel 2.6 : Tabel Default Alphabet 7 bit (septet)

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
B4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0		P		P
0	0	0	1	1	£	_	!	1	A	Q	a	Q
0	0	1	0	2	\$	Φ	"	2	B	R	b	R
0	0	1	1	3	¥	Γ	#	3	C	S	c	S
0	1	0	0	4	è	Λ		4	D	T	d	T
0	1	0	1	5	é	Ω	%	5	E	U	e	U
0	1	1	0	6	ù	Π	&	6	F	V	f	V
0	1	1	1	7	i	ψ	`	7	G	W	g	W
1	0	0	0	8	ò	Σ	(	8	H	X	h	X
1	0	0	1	9	ç	Θ	)	9	I	Y	i	Y
1	0	1	0	10	LF	Ξ	*	:	J	Z	j	Z
1	0	1	1	11	ø	1)	+	;	K	Ä	k	Ä
1	1	0	0	12	Ø	Æ	,	<	L	Ö	l	Ö
1	1	0	1	13	CR	æ	-	=	M	Ñ	m	Ñ
1	1	1	0	14	Å	β	.	>	N	Ü	n	Ü
1	1	1	1	15	å	É	/	?	O	Š¥	o	À

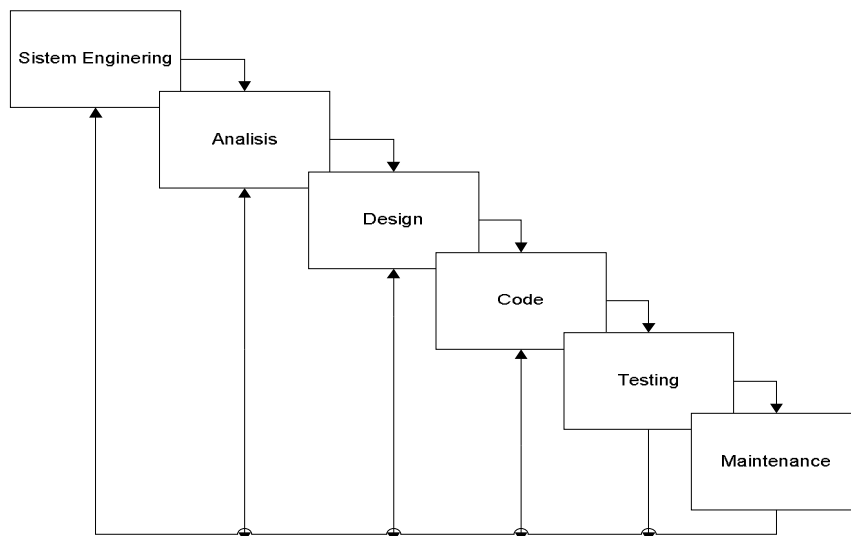
## 2.4 Layanan Aplikasi SMS

Layanan aplikasi SMS pada dasarnya memiliki karakteristik yang berbeda dengan aplikasi internet dan internet pada umumnya, yaitu layar monitor yang berukuran kecil, keterbatasan jumlah karakter yang dapat dituliskan, serta keterbatasan tombol pada ponsel untuk pengoperasian aplikasi. Tiga karakteristik tersebut selalu menjadi fokus yang mendasari pada pengembangan aplikasi tersebut, sehingga informasi yang disediakan singkat dan jelas dengan pengoperasian aplikasi mudah dan sederhana yang merupakan penggunaan tombol pada ponsel. Dengan demikian akan dapat dikenal aplikasi yang cocok untuk dikembangkan menjadi aplikasi berbasis SMS.

## 2.5 Pendekatan Pengembangan dengan metode Waterfall

Pengembangan sistem (system Engineering) dapat berarti menyusun atau membangun sebuah sistem yang baru untuk menggantikan sistem yang lama, baik secara keseluruhan maupun memperbaiki sistem yang telah ada sehingga segala masalah dapat diharapkan dapat diatasi.

Ada beberapa tahapan dalam pengembangan sistem, tahapan-tahapan tersebut merupakan daur hidup (classic Life Cycle) yang selalu berhubungan. Tahapan-tahapan tersebut adalah : Roger S Pressman, 1987:20)



**Gambar 2.3** Model Waterfall atau Daur Hidup Perangkat Lunak

Adapun penjelasan dari metode waterfall diatas yaitu :

### 1. System Engineering dan Analysis

Pertama, semua kebutuhan sistem disusun dan dialokasikan menjadi beberapa bagian kebutuhan perangkat lunak. Gambaran sistem adalah hal yang utama pada saat perangkat lunak di hubungkan dengan elemen lain seperti perangkat keras, manusia, dan basis data. Sistem engineering dan analysis meliputi kebutuhan sekumpulan

manusia pada tingkat sistem dengan jumlah kecil dari desain dan analisis tingkat tinggi.

## **2. Software requirement analysis**

Proses pengumpulan kebutuhan diintensikan dan difokuskan secara khusus pada perangkat lunak. Seorang analis harus mengerti informasi domain perangkat lunak sebelum menyusun program. Seperti fungsi bentuk penghubung (interface) yang dibutuhkan, kebutuhan-kebutuhan untuk sistem dan perangkat lunak perlu didokumentasikan dan diriview dengan konsumen.

## **3. Design**

Desain perangkat lunak adalah proses multistep yang berfokus pada tiga atribut : struktur data, arsitektur perangkat lunak, dan detail prosedur. Proses desain menerjemahkan kebutuhan ke sebuah representasi perangkat lunak yang dapat dinilai berkualitas sebelum pengkodean dimulai. Desain adalah dokumentasi dan menjadi bagian konfigurasi perangkat lunak.

## **4. Coding (pengkodean)**

Desain Harus diterjemahkan ke machine readable form, jika desain dibuat dalam bentuk pengkodean yang detail maka secara mekanis akan menjadi lebih cepat.

## **5. Testing (Test)**

Satu kode telah dihasilkan, maka pengetesan program dimulai. Test program difokuskan pada internal logic perangkat lunak. Semua statement dipastikan telah dites baik pada fungsi eksternal, tes dilakukan untuk memastikan input akan menghasilkan hasil yang diinginkan.

## **6. Maintenance (Pemeliharaan)**

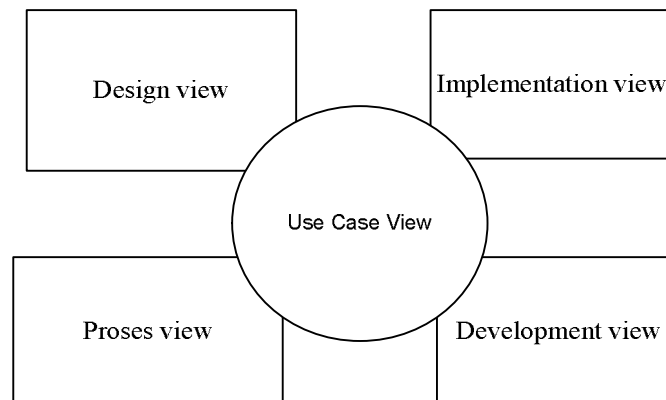
Perangkat lunak pasti mengalami perubahan setelah dikirim ke konsumen (pengecualian yang mungkin adalah perangkat lunak tambahan). Perubahan akan

terjadi karena kesalahan yang ditemukan, karena harus diadaptasikan untuk mengakomodasi perubahan pada lingkungan eksternal, atau karena konsumen membutuhkan peningkatan fungsi dan performanya. Pemeliharaan perangkat lunak selalu memakai daur hidup yang terdahulu untuk sebuah program yang akan dibandingkan dengan yang baru.

## 2.6 Teory Perancangan Antarmuka UML (Unfied Modeling Language)

UML adalah salah satu alat bantu yang sangat handal di dunia sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML di bangun atas model 4+1 view. Model ini di dasarkan pada fakta bahwa struktur sebuah sistem di deskripsikan dalam 5 view dimana salah satu diantaranya use case view. Use case view ini memegang peran khusus untuk mengintegrasikan content ke view lainnya.



**Gambar 2.4** Model 4 +1 View

Kelima view tersebut tidak berhubungan dengan diagram yang dideskripsikan di UML. Setiap view berhubungan dengan perspektif tertentu dimana sistem akan di uji. Penjelasan lengkap tentang sistem bisa di bentuk dengan menggabungkan informasi yang ada pada kelima view tersebut yaitu :

### **1. Use Case View**

Mendefinisikan perilaku eksternal sistem sistem. Hal ini menjadi daya tarik user bagi end user, analis dan tester. Pandangan ini mendefinisikan kebutuhan sistem karena mengandung semua view yang lain yang mendeskripsikan aspek-aspek tertentu dari rancangan sistem. Itulah sebabnya use case view menjadi pusat peran pengembangan perangkat lunak.

### **2. Design View**

Mendeskripsikan struktur logika yang mendukung fungsi-fungsi yang dibutuhkan use case. Design view ini berisi definisi komponen program, class-class utama bersama-sama dengan spesifikasi data, perilaku dan interaksinya. Informasi yang terkandung di view ini menjadi perhatian karena menjelaskan secara detail bagaimana fungsionalitas sistem akan diimplementasikan.

### **3. Implementation View**

Menjelaskan komponen-komponen fisik dari sistem yang akan dibangun. Hal ini berbeda dengan komponen logic yang dideskripsikan pada design view. Termasuk disini adanya file exe, library dan database.

#### **1. Proses View**

Berhubungan dengan hal-hal yang berkaitan dengan concurrency dalam sistem. Sedangkan deployment view menjelaskan bagaimana komponen-komponen fisik didistribusikan ke lingkungan fisik seperti jaringan komputer dimana sistem akan

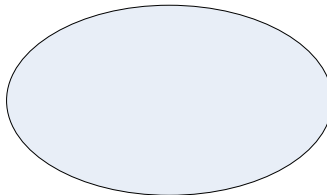


dijalankan . kedua view ini menunjukkan kebutuhan fungsional dari sistem seperti toleransi kesalahan dan hal yang berhubungan dengan kinerja sistem.

### 2.6.1 Pengertian Use Case

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan di inialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu.

Penamaan Use Case sesuai dengan tujuan yang di capai dari hasil interaksi dengan actor. Use Case biasanya menggunakan kata kerja misalnya.



**Gambar 2.5** Entry Pemesanan

Use Case adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Tidak terlalu mudah bagi pengguna untuk menyatakan bagaimana mereka bermaksud menggunakan sebuah sistem. Karena sistem pengembangan tradisional sering ceroboh dalam melakukan analisis, akibatnya pengguna sering kali susah menjawabnya tatkala dimintai masukan tentang sesuatu.

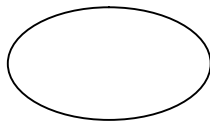
## 2.6.2 Notasi Use Case

Diagram use case menunjukan tiga aspek dari sistem yaitu: actor, use case dan sistem boundary. Actor mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan use case.



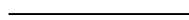
Actor

Sebuah actor mencirikan suatu bagian outside user atau susunan yang berkaitan dengan user yang berinteraksi dengan sistem.



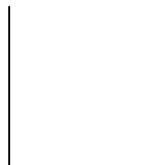
Use Case

Use Case adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah actor.



Arus

Menggambarkan data



Sistem Boundary

Sistem boundary menggambarkan batas suatu sistem

Untuk mengidentifikasi actor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Actor adalah abstraction dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem yang biasa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan use case tetapi tidak memiliki control atas use case.

### **2.6.3 Relasi Use Case**

Selain terdapat relasi-relasi antara aktor dan use case juga terdapat relasi use case-use case. Ada beberapa jenis relasi antara use case-use case, yaitu :

- (1) Include, digunakan untuk menggambarkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
- (2) Generalization, digunakan untuk menggambarkan variasi pada kondisi perilaku normal tetapi digambarkan dengan sedikit berbeda.
- (3) Extend, digunakan untuk menunjukkan bahwa satu use case merupakan tambahan fungsional dari use case yang lain jika kondisi atau syarat tertentu dipenuhi.

### **2.6.4 Diagram Aktivitas (Activity Diagram)**

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa.

### **2.6.5 Sequence Diagram (Diagram Sequence)**

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Pada diagram ini menunjukkan sejumlah contoh obyek dan message (pesan) yang diletakkan obyek-obyek ini didalam use case. Komponen utama sequence diagram atas obek yang dituliskan dengan kotak segiempat bernama.message diwakili dengan tanda panah dan waktu ditunjukkan dengan progress vertical.

Sequence diagram mempunyai :

### **A. Obyek/Participant**

Obyek yang diletakan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. mereka diatur dalam urutan guna menyederhanakan diagram. Pengertian obyek hanya ada di UML 1, sedangkan di UML 2 di ganti dengan participant.

### **B. Lifeline dan Aktivation.**

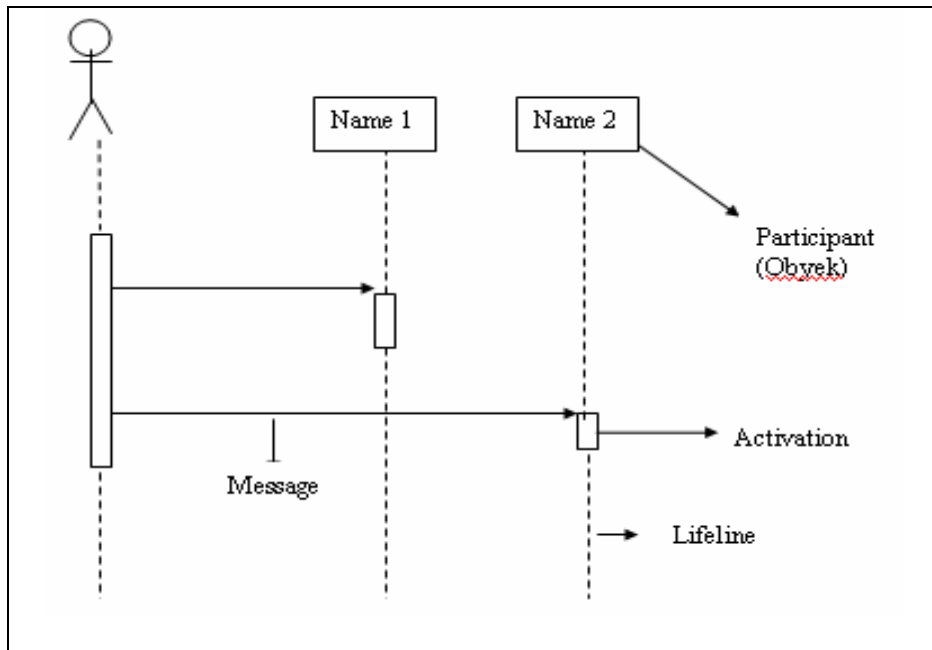
Setiap participant terhubung dengan garis titik-titik yang disebut lifeline. Sepanjang lifeline ada kotak yang disebut activation. Activation berbanding lurus dengan dengan durasi activation.

### **C. Message**

Sebuah message bergerak dari satu participant ke participant yang lain dari satu lifeline ke lifeline yang lain. Sebuah participant bisa mengirim sebuah mesagge kepada dirinya sendiri.

### **D. Time**

Time adalah diagram yang mewakili waktu pada arah vertical. Waktu dimulai dari atas kebawah. Message yang lebih dekat dari atas atas akan dijalankan terlebih dahulu disbanding message yang lebih dekat kebawah.



**Gambar 2.6** Simbol-simbol pada sequence diagram

## 2.7 Class Diagram

Sebuah Class Diagram menunjukkan struktur yang statis dari beberapa class dalam suatu sistem. *Class-class* merepresentasikan suatu keadaan (atribut/properti) dan yang akan dikerjakan oleh sistem (metoda/fungsi).

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

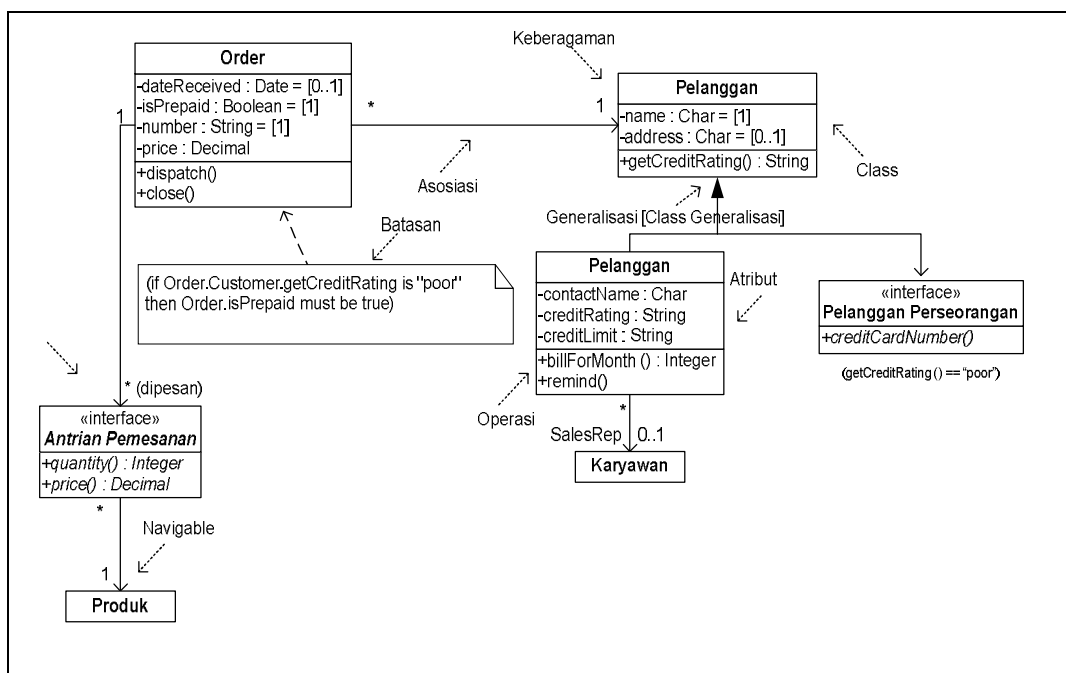
Atribut dan metoda dalam class diagram dapat memiliki salah satu sifat seperti berikut di bawah ini :

- *Private*, hanya dapat diakses oleh class itu sendiri.
- *Protected*, hanya dapat diakses oleh class itu sendiri dan turunan dari class tersebut.

- *Public*, dapat diakses oleh class selain dari class yang bersangkutan.

*Class* dapat direpresentasikan dalam sebuah interface atau sebaliknya merupakan implementasi dari sebuah interface yang berupa class abstrak yang hanya tidak memiliki attribute dan hanya memiliki metoda.

Berikut merupakan bentuk *class* diagram secara umum :



**Gambar 2.7** Class Diagram

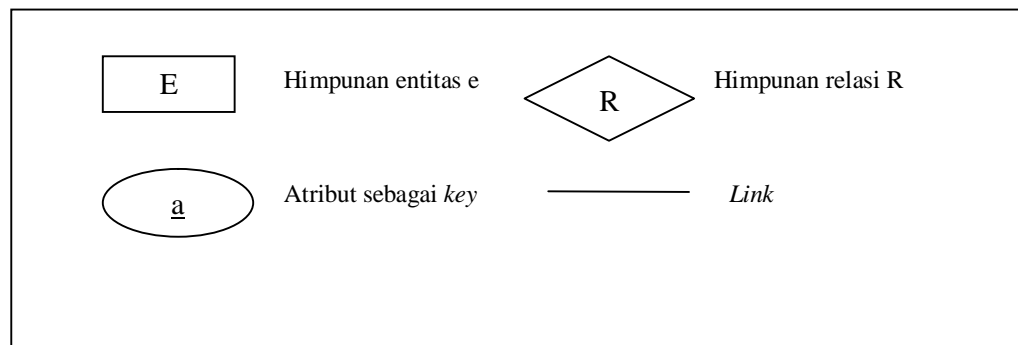
## 2.8 Entity Relationship Diagram (ERD)

Menurut Fathansyah (2001:70) Diagram Keterhubungan Entitas atau *Entity-Relationship Diagram*, selanjutnya disebut ERD, adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DAD). ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Model *entity-relationship* yang

berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang merepresentasikan seluruh fakta dari *real world* (dunia nyata) yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan ERD.

ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Notasi-notasi simbolik di dalam ERD yang dapat digunakan adalah:

1. Persegi panjang, menyatakan himpunan entitas.
2. Lingkaran atau elips, menyatakan atribut (atribut yang berfungsi sebagai key digarisbawahi).
3. Belah ketupat, menyatakan himpunan relasi.
4. Garis (*link*), sebagai penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.
5. Kardinalitas relasi dapat dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan N untuk relasi satu-ke-banyak atau N dan N untuk relasi banyak-ke-banyak).



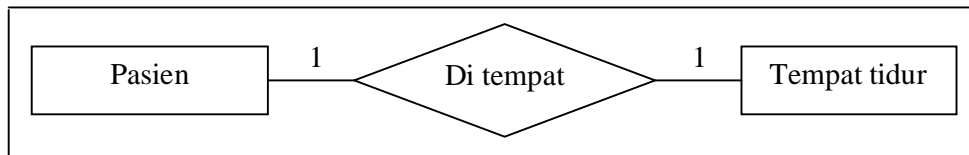
**Gambar 2.8** Notasi ERD

Entitas merupakan individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas.

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas dapat berupa:

1. Satu ke satu (*One to one*).

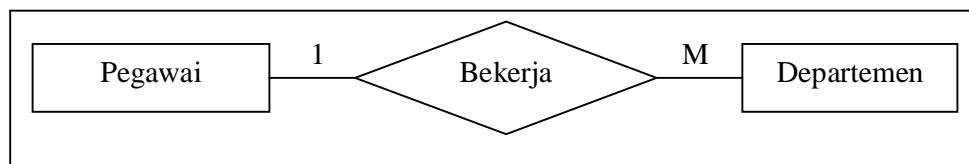
Setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas kedua, begitu juga sebaliknya.



**Gambar 2.9** Contoh kardinalitas satu ke satu (Linda Marlinda, S.Kom, 2004:20)

2. Satu ke banyak (*One to many*).

Setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas kedua, dan setiap entitas pada himpunan entitas kedua berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas pertama.

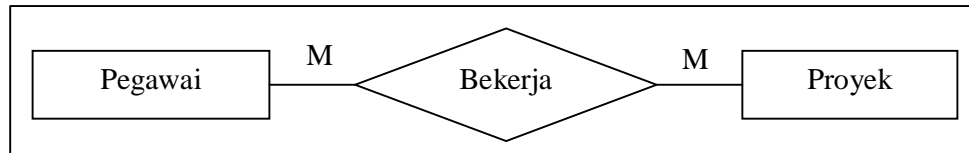


**Gambar 2.10** Contoh kardinalitas satu ke banyak

3. Banyak ke banyak (*Many to many*).



Setiap entitas pada himpunan entitas pertama dapat berhubungan dengan banyak entitas pada himpunan entitas kedua, dan demikian juga sebaliknya.



**Gambar 2.11** Contoh kardinalitas banyak ke banyak

Kardinalitas relasi satu ke banyak dan banyak ke satu dapat dianggap sama, karena tinjauan kardinalitas relasi selalu dilihat dari dua sisi (dari himpunan entitas pertama ke himpunan entitas kedua dan dari himpunan entitas kedua ke himpunan entitas pertama).

Dua kelompok pentahapan yang biasa ditempuh dalam pembuatan ERD, yaitu:

1. Tahap pembuatan ERD awal (*preliminary design*).

Mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau.

2. Tahap optimasi ERD (*final design*)

Melakukan koreksi terhadap hasil tahap pertama, berupa pendekomposisian himpunan entitas, penggabungan himpunan entitas, pengubahan derajat relasi, penambahan relasi baru hingga perubahan (penambahan dan pengurangan) atribut-atribut untuk masing-masing entitas dan relasi.

Langkah-langkah teknis yang dapat dilakukan untuk menghasilkan ERD awal adalah:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang terlibat.
2. Menentukan atribut-atribut *key* dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta kunci asingnya (*foreign-key*).

4. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (non *key*).

## 2.9 Normalisasi

Suatu desain database harus memenuhi kondisi dari struktur perancangan yang baik diantaranya yaitu tidak mengandung anomaly yaitu suatu kejanggalan dari suatu penempatan atribut dari suatu objek data. Untuk menghindari hal tersebut maka diperlukan proses normalisasi data. Normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entity dan relasinya.

Normalisasi adalah proses pengelompokan atribut-atribut dari suatu relasi sehingga membentuk sebuah relation yang jumlah kerangkapan datanya sedikit dan memberikan kemungkinan bagi user untuk melakukan insert, delete, modify terhadap garis-garis data pada relation tersebut yang tidak berakibat terjadinya error atau inconsistencies data yang disebabkan oleh operasi-operasi tersebut.

Berdasarkan aturan dari ketergantungan data dalam pembentukan normalisasi dilakukan langkah-langkah sebagai berikut :

1. Bentuk tidak normal pertama (Unnormalized Form)

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu. Dapat saja data tidak lengkap atau terdublikasi. Data dikumpulkan apa adanya sesuai dengan menginputnya.

2. Bentuk normal pertama (1NF/ First Normal Form)

Suatu relasi 1NF jika dan hanya jika sifat dari setiap relasi atribut bersifat atomic. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

Ciri-ciri 1NF :

- Setiap data dibentuk dalam flat file, data dibentuk dalam satu record demi satu record nilai dari field berupa atomic value.
- Tidak ada set atribut yang berulang atau bernilai ganda
- Tiap file hanya satu pengertian saja

Untuk bentuk normal pertama, data store hanya berisi elemen data sederhana dan setiap elemen hanya mempunyai satu nilai. Menurut Kowal (1988,p88-189) ada 3 cara untuk mengubah data store menjadi bentuk normal pertama :

- Gantilah semua elemen group data dengan elemen data sederhana yang sebanding
- Hapus semua group berulang.
- Perluaslah primary key untuk memasukan pengenalan yang unik dari setiap group berulang yang telah dihapus.

### 3. Bentuk normal kedua (2NF / Second Normal Form) ]

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukanlah kunci haruslah bergantung secara fungsi pada kunci utama / primary key, sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci field. Kunci field haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya. Menurut Kowal (1998,p190) untuk bentuk normal kedua, semua atribut bukan key (bukan primary key) harus mempunyai hubungan dari semua kunci dan tidak hanya sebagian dari kunci. Untuk mengubah data store menjadi 2NF maka diperlukan langkah-langkah sebagai berikut :

- jika primary key bukan merupakan composite key, key memiliki lebih dari satu elemen data dasar, data store secara otomatis menjadi 2NF meskipun dalam bentuk 1NF

- Jika primary key adalah komposite key, kita perlu untuk memeriksa data store untuk membenarkan bahwa setiap atribut bukan key menggambarkan keseluruhan key dan tidak sebagian dari key.

#### 4. Bentuk normal ketiga (3NF/third Normal Form)

Untuk menjadi bentuk ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif.

### **2.10 Bahasa Pemrograman Java (Jdk1.5\_08)**

Java merupakan bahasa pemrograman yang berorientasi objek. Tidak seperti bahasa pemrograman pendahulunya, C++, Java sepenuhnya menerapkan konsep Object Oriented Programming (OOP) dalam setiap program aplikasinya. OOP itu sendiri hanyalah merupakan sebuah konsep yang akan membawa pembaca ke sebuah paradigma baru dalam membuat program.

Metode Pemrograman Berorientasi Objek (*Object Oriented Programming* OOP) menawarkan suatu teknik pembuatan dan pengembangan program aplikasi dengan cara mudah dan menyenangkan. Berbagai program aplikasi berbasis windows dibangun menggunakan metode program ini.

Konsep yang digunakan dalam OOP berbeda dengan metode prosedural. Terdapat beberapa aspek yang harus diketahui sehubungan dengan metode OOP. Meski demikian, metode OOP bukannya tanpa masalah persoalan besar yang dihadapi dalam metode ini adalah sulitnya memahami konsep objek beserta aspek-aspek yang terkait misalnya class, inheritance dan lain-lain. Masalah besar lainnya adalah

besarannya sumber daya komputer yang diperlukan, misalnya memerlukan memori lebih banyak dibandingkan dengan jika membuat program tradisional. Dua buah objek yang identik akan memerlukan dua area memori berbeda walaupun dari sisi data dan proses keduanya memiliki jumlah dan jenis yang sama. Hal ini disebabkan karena data dan proses pada kedua objek tersebut dipisahkan oleh komputer.

### 2.10.1 Tool Utama dari Java 2 SDK 1.5

Java 2 SDK menyediakan tool-tool penting untuk kompilasi dan menjalankan program Java. Dalam Java 2 SDK ada dua tool utama yang digunakan dalam mengompilasi dan menjalankan program Java.

#### i. javac

Perintah ini digunakan untuk mengompilasi program Java. Perintah ini akan mengubah file berektensi **.java** menjadi byte code (file berektensi **.class**).

Cara menggunakannya:

```
C:>javac namafilejava.java
```

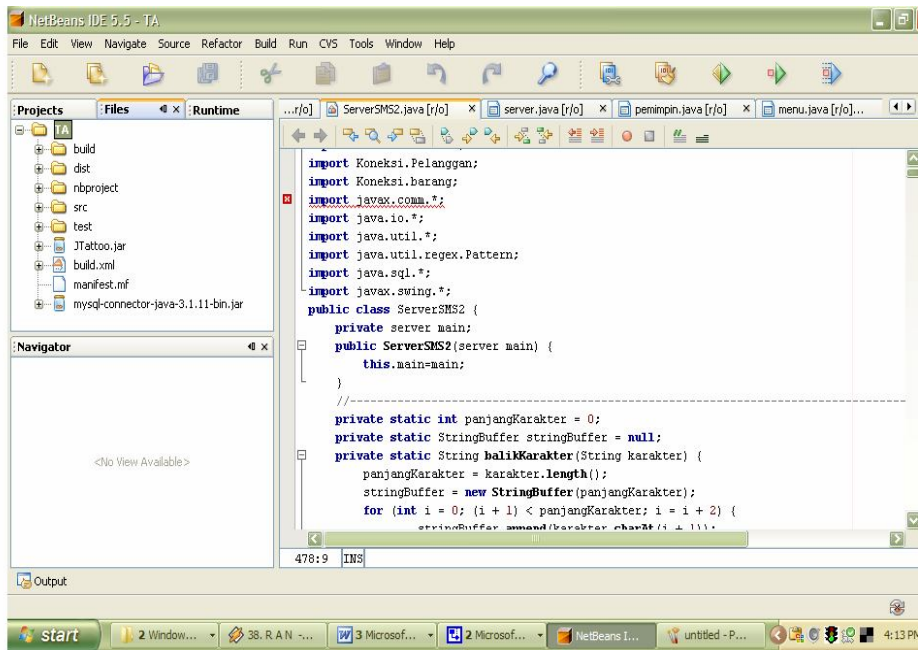
#### ii. Java

Perintah ini digunakan untuk menjalankan program java atau file hasil kompilasi (file berektensi **.class**).

Cara menggunakannya:

```
C:>javac nama class Atau
```

```
C:>javac -classpath namaclass.
```



Gambar 2.12 Lembar kerja java

## 2.11 Database MySQL

Database ini telah menyebut dirinya sendiri sebagai “Database *open source*” paling terkenal di dunia, yang jelaslah bukan keangkuhan diri dikarenakan sejak pertama kali dirilis pada tahun 1995 dimana internet *booming*. MySQL secara cepat menjadi database dalam pemrograman *web* yang dominan dikarenakan kecepatan, ukuran *file*-nya, kemudahan penggunaan yang komperatif selain juga dikarenakan hak lisensinya yang sejak pertama kali diluncurkan sudah bersifat *open source* dan bebas untuk digunakan untuk tujuan apa saja. Database ini juga sudah dikenali oleh beberapa bahasa pemrograman sebut saja *Java* dan *PHP*, sehingga dengan alasan-alasan tersebut penulis memilih untuk menggunakannya dalam pengembangan program aplikasi ini.

MySQL sediri dikembangkan, dijual, dipasarkan dan didukung oleh MySQL AB, suatu perusahaan Swedia. Lisensi perusahaan mempunyai 2 cara :

- a. Open source software: MySQL tersedia via GNU GPL (*General Public License*) untuk yang gratis. Siapa saja yang ingin menggunakan software ini secara cuma-cuma (*Free*).
- b. Commercial License: MySQL tersedia dengan lisensi komersial (*Commercial License*) bagi siapa saja yang menyukai GPL. Jika pengembangan ingin menggunakan MySQL sebagai bagian dari software produk baru dan ingin menjual produk, dibanding melepaskannya di bawah GPL, pengembang harus membeli lisensi komersial

Seperti sudah disebutkan beberapa alasan diatas menggunakan database MySQL karena memiliki beberapa keunggulan, diantaranya:

- a. Cepat.

Tujuan utama dari pengembangan MySQL adalah kecepatan, sebagai konsekuensi software yang dirancang dari awal untuk kecepatan.

- b. Tidak mahal.

MySQL adalah cuma-cuma di bawah lisensi GPL, open source, sementara pembiayaan untuk lisensi komersialnya sangatlah pantas.

- c. Mudah digunakan.

Anda dapat membangun dan berinteraksi dengan database MySQL hanya dengan menggunakan sedikit pernyataan sederhana di dalam bahas SQL, yang menjadi bahasa standar untuk komunikasi dengan RDBMS (*Rational Database Management System*).

## **BAB III**

### **ANALISA DAN PERANCANGAN SISTEM**

#### **3.1 Permasalahan Dan Strategi Pemecahan Masalah**

Toko adalah tempat pemesanan barang, merupakan kegiatan rutin setiap hari yang mereka lakukan untuk melayani pelanggan dalam membeli barang. Kegiatan ini menjadi sangat penting karena dapat dijadikan sebagai tolak ukur keberhasilan sebuah toko, dari sini dapat diketahui bagaimana pandangan masyarakat terhadap toko tersebut. Dan sebagaimana biasanya bila ingin memesan barang pelanggan ingin melihat Brosur barang terlebih dahulu. Selain itu Toko untuk setiap bulan akan membuat sebuah laporan yang akan diberikan kepada pihak yang memberi wewenang sebagai salah satu bentuk pertanggung jawaban.

Pada Wajah Baru Elektronik, kegiatan pemesanan barang biasanya dilakukan masih menggunakan telepon ataupun langsung datang ke toko tersebut. Tahapan-tahapan pemesanan barang melalui telepon yang harus dilalui pelanggan adalah pelanggan telepon sebutkan nama, no KTP, alamat dan setelah itu dapat melakukan pemesanan barang. Dalam penyerahan laporan, selama ini biasanya diberikan dalam bentuk tertulis (*print out*) yang diserahkan pada waktu satu bulan.

Laporan diserahkan setelah melalui proses yang panjang sehingga menimbulkan akibat ketidakefisienan pada waktu biaya yang harus dikeluarkan. Selain itu kendala lainnya adalah kesibukan pimpinan sebagai pihak yang menerima laporan yang tidak selalu berada di satu tempat karena beragam aktifitas yang harus dilakukan, sehingga laporan yang seharusnya dapat diterima dengan cepat menjadi



agak lama diterima. Mengingat biasanya laporan diberikan melalui mesin faks, tetapi belum dapat dipastikan apakah pimpinan dapat menerima dengan cepat atau tidak.

### **3.1.1 Pemecahan Masalah**

Melihat permasalahan di atas dibutuhkan suatu cara agar pimpinan dapat menerima laporan penjualan dimanapun berada dan kapan pun diperlukan. Untuk mengatasi masalah hal tersebut maka digunakan suatu sistem yang menggunakan SMS sebagai media penyimpanan. Keuntungan yang dapat diperoleh dan penggunaan aplikasi adalah mengurangi waktu dan biaya yang harus dikeluarkan dalam pembuatan sebuah laporan penjualan. Serta memudahkan pimpinan atasan untuk mendapatkan informasi dengan cepat serta mudahnya penggunaan aplikasi tersebut sebab SMS merupakan salah satu fasilitas yang umumnya selalu ada pada setiap telepon selular dan mudah dimengerti penggunaannya.

Selain laporan penjualan yang dapat diakses melalui SMS dapat juga melakukan pemesanan barang untuk pelanggan melalui SMS untuk memudahkan dalam melakukan pemesanan. Dan bila setiap pelanggan yang ingin memesan ingin melihat info tentang daftar barang maka toko menyediakan fasilitas info daftar barang, dan pelanggan bila ingin melakukan pemesanan pembeli harus melakukan registrasi terlebih dahulu pada toko.

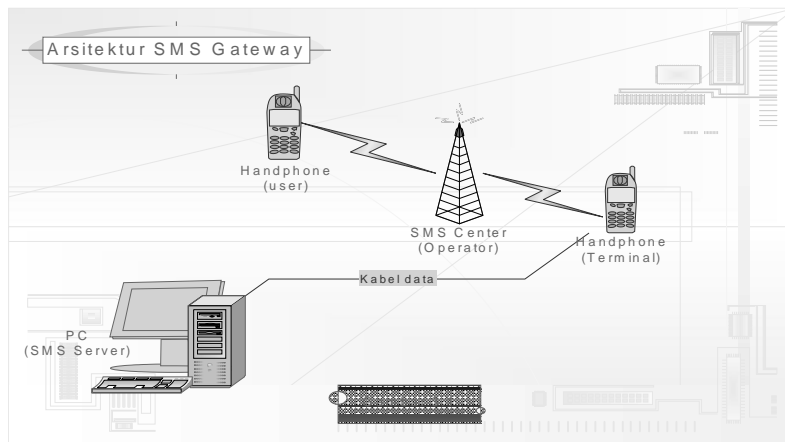
## **3.2 Rancangan Aplikasi**

### **3.2.1 Analisa Aplikasi Usulan**

Analisa aplikasi yang dimaksudkan adalah untuk memberikan pemecahan masalah mengenai ketidakefisienan dan kesulitan dalam pemesanan barang yang dilakukan pelanggan dan penyerahan laporan penerimaan penjualan kepada pimpinan

yang membutuhkan. Salah satu aplikasi yang dapat digunakan untuk mengatasi permasalahan pemesanan barang dan penyerahan laporan tersebut adalah menggunakan SMS yang bersifat interaktif. Dengan cara ini pelanggan dapat melakukan pemesanan barang di manapun mereka berada dan pimpinan dapat sebagai *user* dapat mengetahui laporan terakhir mengenai laporan penjualan. Hanya dengan mengirimkan SMS yang berisikan kode yang telah ditentukan formatnya, kemudian aplikasi ini akan mengirimkan SMS balasan yang berisi informasi sesuai dengan permintaan. Hal ini sangat dibutuhkan oleh para pelanggan maupun pimpinan yang mempunyai mobilitas sangat tinggi.

Sedangkan skema arsitektur SMS gateway yang digunakan untuk aplikasi ini dapat dilihat pada gambar 3.1. dibawah ini. Arsitektur ini menggunakan tiga hardware yaitu PC/Laptop (SMS Server), *handphone (terminal)*, operator Indosat sebagai SMSC, kabel data/*Bluetooth* serta dibutuhkan juga *handphone user* sebagai pengirim.



**Gambar 3.1** Skema Arsitektur SMS Gateway

([www.schema sms gateway.com/](http://www.schema_sms_gateway.com/)15september 2008)

Keterangan Hardware diatas adalah:

- PC/Laptop (SMS Server), untuk meletakkan aplikasi SMS Gateway dan Administrasi SMS yang akan dibangun
- *Handphone* (Terminal), untuk menerima ataupun mengirim SMS yang dikirimkan oleh *Handphone user*. Digunakan sebagai terminal
- *Handphone* (*User*), untuk mengirimkan ataupun menerima SMS dari atau ke terminal
- *Bluetooth* untuk mentransfer SMS (*input/output*) dari *Handphone* ke PC atau sebaliknya
- Operator **Telkomsel** sebagai *SMS Center*

### 3.2.2 Metode Kerja Sistem Usulan

Sistem aplikasi ini dijalankan dengan menggunakan *NetBeans 5.5* sebagai sistem dan *MYSQL* sebagai *database*. Sedangkan pengiriman dan penerimaan SMS dari *user* maupun dari pelanggan akan dilakukan oleh operator indosat untuk percobaan.

Di bawah ini adalah tahapan-tahapan umum proses yang dilakukan pada aplikasi SMS Pemesanan Barang adalah

- 1) Pemilik dan pelanggan sebagai *user* mengirim pesan SMS dengan format tertentu yang telah ditentukan dan diterima oleh *SMSC* (*SMS Center*).
- 2) SMS yang masuk di *SMSC* akan dikirimkan ke komputer di Wajah Baru Elektronik melalui *SMS center*. Setelah itu SMS yang masuk akan disimpan ke dalam *database*.
- 3) SMS ini kemudian akan diproses dan hasilnya akan dikirimkan kembali kepada *SMSC* melalui SMS kembali ke *user*.

Sedangkan tahapan-tahapan yang dilakukan hanya menggunakan *Internal* Komputer untuk menerima SMS yang masuk adalah:

- 1) Awalnya SMS Server akan membuka komunikasi pada *COM* untuk mendengarkan (*listen*) apakah ada permintaan atau tidak.
- 2) untuk mengambil permintaan, sistem memeriksa alamat *remote* IP, apakah sama dengan format yang ditentukan. Jika tidak sama maka SMS Server akan membalas kembali untuk mengatakan bahwa format SMS yang anda ketikkan salah.
- 3). Jika data yang diambil benar maka akan dipecah-pecah menjadi bagian-bagian kecil, yang berisikan data nomor *handphone*, service id, isi SMS permintaan, pada tabel telepon.
- 3) Selanjutnya adalah memeriksa data nomor *handphone* telah terdaftar sebagai nomor *handphone* yang berhak untuk mengakses aplikasi, pada tabel pelanggan dan pimpinan bila tidak maka SMS Server akan membalas bahwa anda belum dapat melakukan transaksi pada Toko tersebut.
- 4) Terakhir jika nomor *handphone* terdaftar, seluruh data yang telah dipecah-pecah tersebut akan proses ke proses Query yang telah ditentukan sesuai kode, untuk menggunakan sebagai informasi pada proses pengolahan data.

Untuk pengolahan SMS yang masuk, *system* aplikasi berbasis SMS dalam *system* usulan ini akan memanfaatkan *database* yang berisikan informasi pemesanan barang dan laporan penjualan untuk pimpinan toko melalui SMS. Untuk pengolahan, awalnya dilakukan dengan memeriksa status pemrosesan SMS pada tabel *outbox*. Jika SMS belum diproses, maka pengolahan data akan dilanjutkan. Informasi yang didapat dari hasil itu sesuai prosesQuery. Informasi inilah yang akan dikirimkan kembali kepada *user* yang meminta. Dalam penggunaan aplikasi ini, terdapat beberapa ketentuan yaitu:

- 1) Pemesanan barang dan laporan akan dikirimkan setelah *user* mengirimkan permintaan berdasarkan format tertentu dan akan menyimpan *request*.
- 2) Jika terjadi kesalahan format dalam pengiriman permintaan, SMS tersebut tidak akan diproses didalam proses Query dan akan dikirimkan pesan kesalahan.

Agar memudahkan pengolahan data dan menghindarkan terjadinya kesalahan pengiriman maka dalam aplikasi ini juga ditentukan format yang digunakan dalam pengiriman SMS.

**Tabel 3.1** Format SMS

<b>Format SMS</b>	<b>Keterangan</b>
REG	Untuk registrasi menjadi anggota
HELP	Untuk mengetahui format sms yang benar
UPDATE	Untuk merubah data pelanggan.
INFO	Untuk mengetahui spesifikasi data barang dan harganya.
PESAN	Untuk melakukan pemesanan.
BATAL	Untuk melakukan pembatalan pesanan
SARAN	Untuk menerima saran dari pelanggan
LAPORAN	Untuk pimpinan yang ingin mengetahui laporan perbulan.
UNREG	Untuk berhenti menjadi anggota.

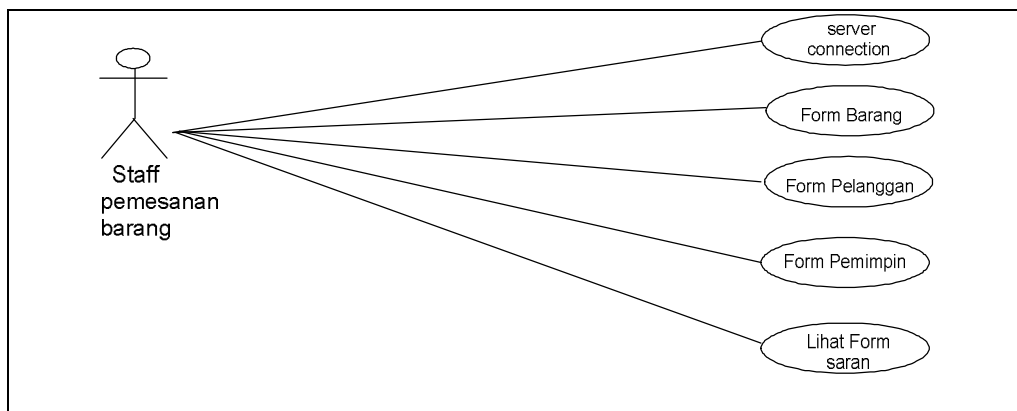
Ini adalah contoh format yang digunakan

- 1) **REG<spasi>Nama#No KTP#Alamat**
- 2) **HELP**
- 3) **UPDATE<spasi>Nama#NoKTP#Alamat**
- 4) **INFO<spasi>Kode Barang**
- 5) **PESAN<spasi> Kode Barang<spasi>Jml Barang#Alamat**
- 6) **BATAL<spasi>No Pesanan**
- 7) **SARAN<spasi>Isi saran**

- 8) LAPORAN<spasi>dd-mm-yyyy contoh:LAPORAN<spasi>19-03-2008  
ATAU LAPORAN <spasi>dd-mm-yyyy<spasi>dd-mm-yyyy  
contoh:LAPORAN<spasi>19-03-2008<spasi>19-04-2008
- 9) UNREG

### 3.3 Diagram Use Case

Sebuah use case mempresentasikan sebuah interaksi antara aktor dengan sistem. Use case diagram juga menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada di luar sistem (aktor). Rancangan hasil analisa tersebut adalah sebagai berikut :



**Gambar 3.2** Diagram Use Case Dalam aplikasi

**Tabel 3.2** Keterangan Diagram Entry Data Barang

Use case	Entry Data Barang
Primary Actor	Staff Pemesanan Barang
Main Success Scenario	Staff pemesanan barang menginput atau menambah data ke dalam form entry form barang , kemudian data yang telah ditambah atau diedit tersebut disimpan dalam database form barang.

**Tabel 3.3** Keterangan Diagram Entry Data pelanggan

Use case	Entry Data pelanggan
Primary Actor	Staff pemesanan Barang
Main Success Scenario	Staff pemesanan Barang menginput atau menambah data ke dalam form pelanggan , kemudian data yang telah ditambah atau diedit tersebut disimpan dalam database form pelanggan.

**Tabel 3.4** Keterangan Diagram Entry Data pemimpin

Use case	Entry Data pelanggan
Primary Actor	Staff pemesanan Barang
Main Success Scenario	Staff pemesanan Barang menginput atau menambah data ke dalam form pemimpin , kemudian data yang telah ditambah atau diedit tersebut disimpan dalam database form pemimpin.

**Tabel 3.5** Keterangan Diagram Server

Use case	server
Primary Actor	Staff pemesanan barang
Main Success Scenario	Untuk memberitahukan kepada pelanggan apabila ada potongan harga dengan mengirimkannya lewat sms.

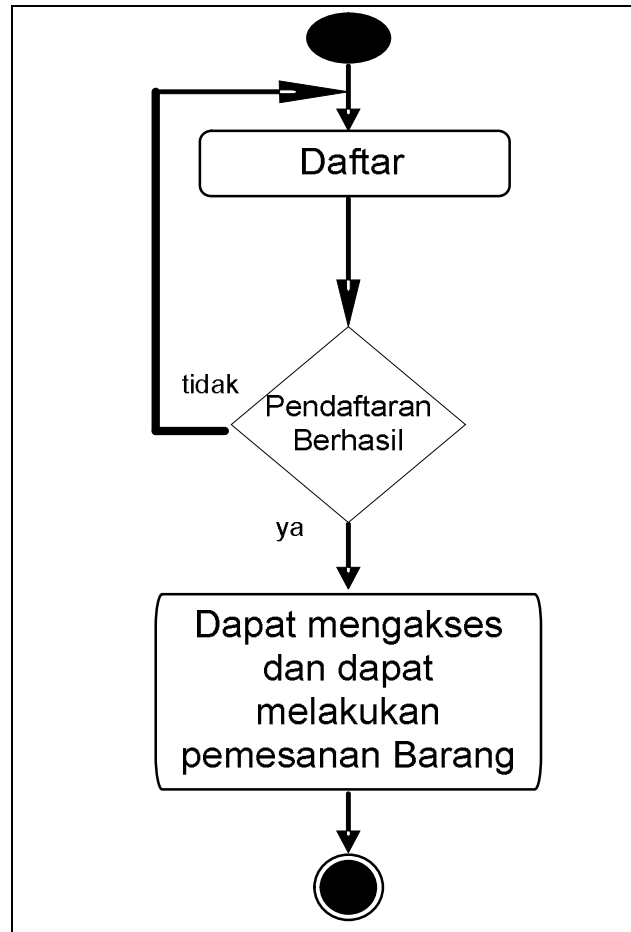
**Tabel 3.6** Keterangan Diagram Saran

Use case	Saran
Primary Actor	Staff pemesanan barang
Main Success Scenario	Staff pemesanan barang memperlihatkan Pesan kritik dan saran yang masuk ke tabel saran yang dikirimkan oleh pelanggan.

### 3.4 Diagram Activity

Berikut ini *activity diagram* yang terjadi dalam sistem pengiriman sms.

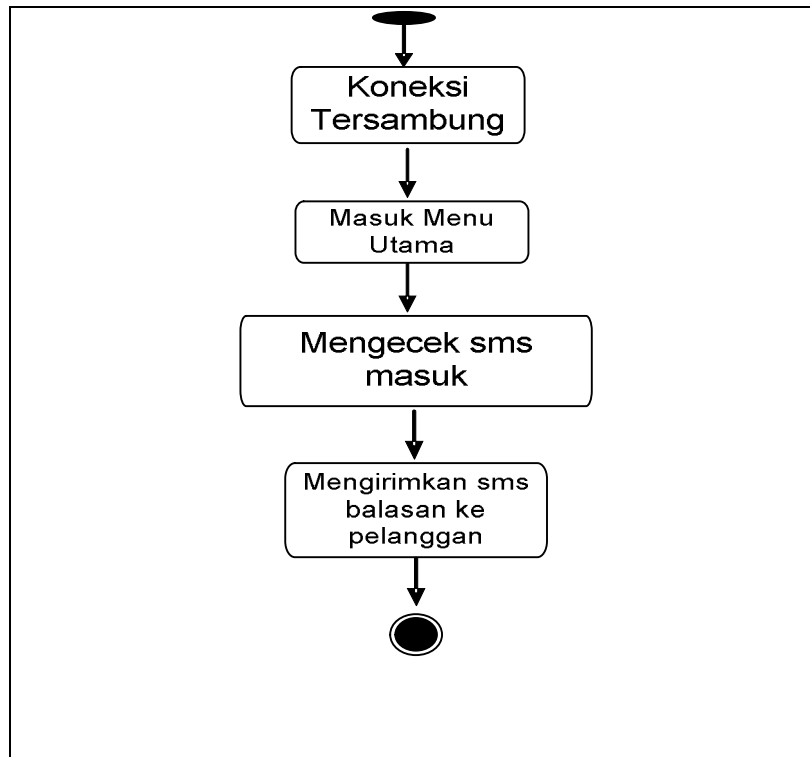
#### A. prosedur pengiriman sms



**Gambar 3.3** Diagram Activiti prosedur pengiriman sms



B. proses penerimaan sms masuk



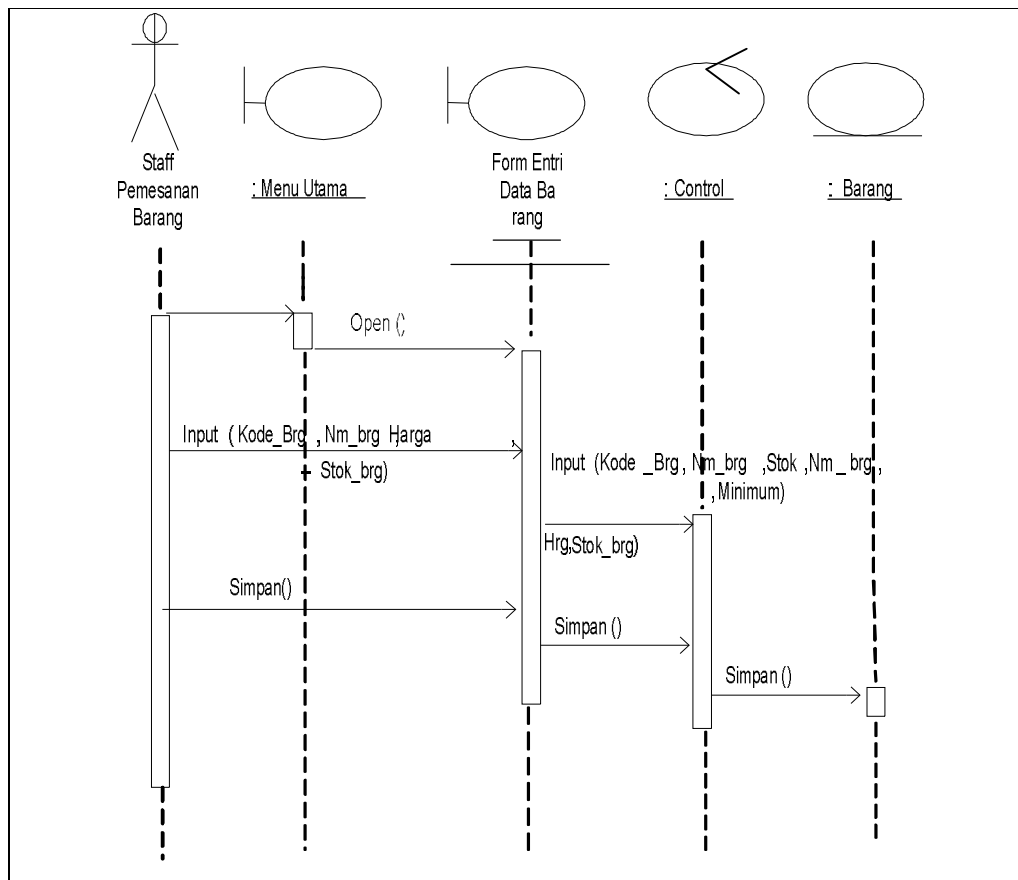
**Gambar 3.4** Diagram Activiti proses penerimaan sms masuk

### 3.5 Diagram Sequence

Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu.

#### a) Diagram Sequence Entry Data Barang

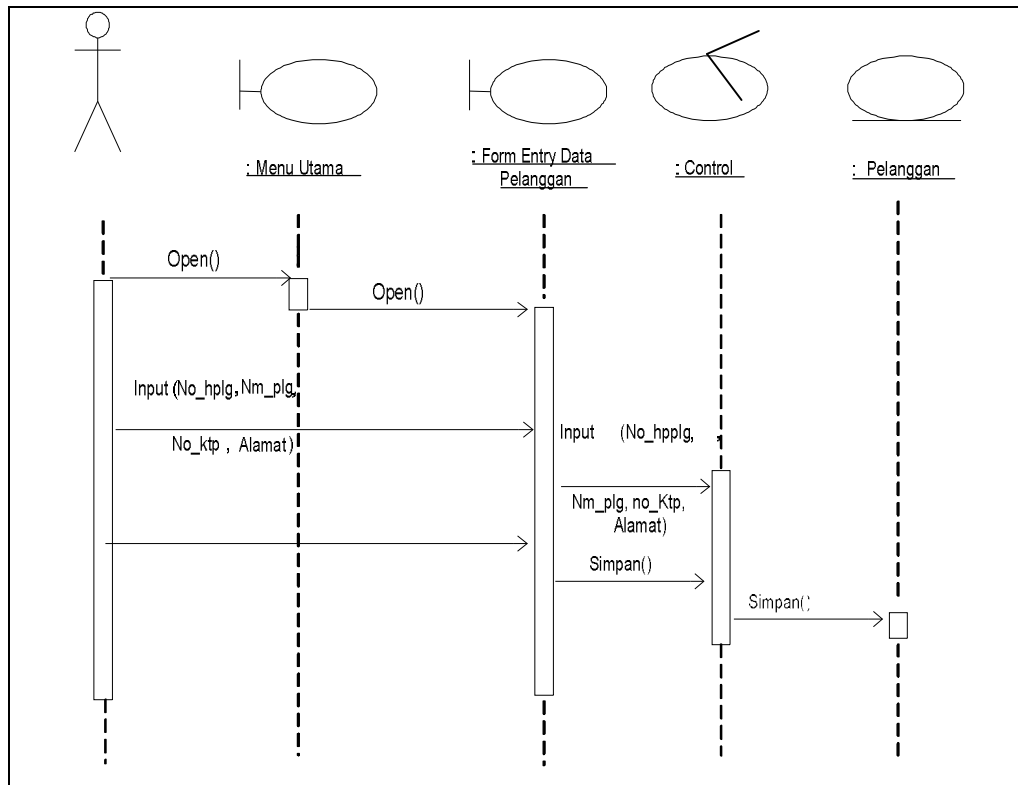
Diagram sequence entry data barang menerangkan kegiatan yang dilakukan sistem pada saat meng-entry data barang. Lihat gambar 3.5



**Gambar 3.5 :** Diagram Sequence Entry Data Barang

b) Diagram Sequence Entry Data Pelanggan

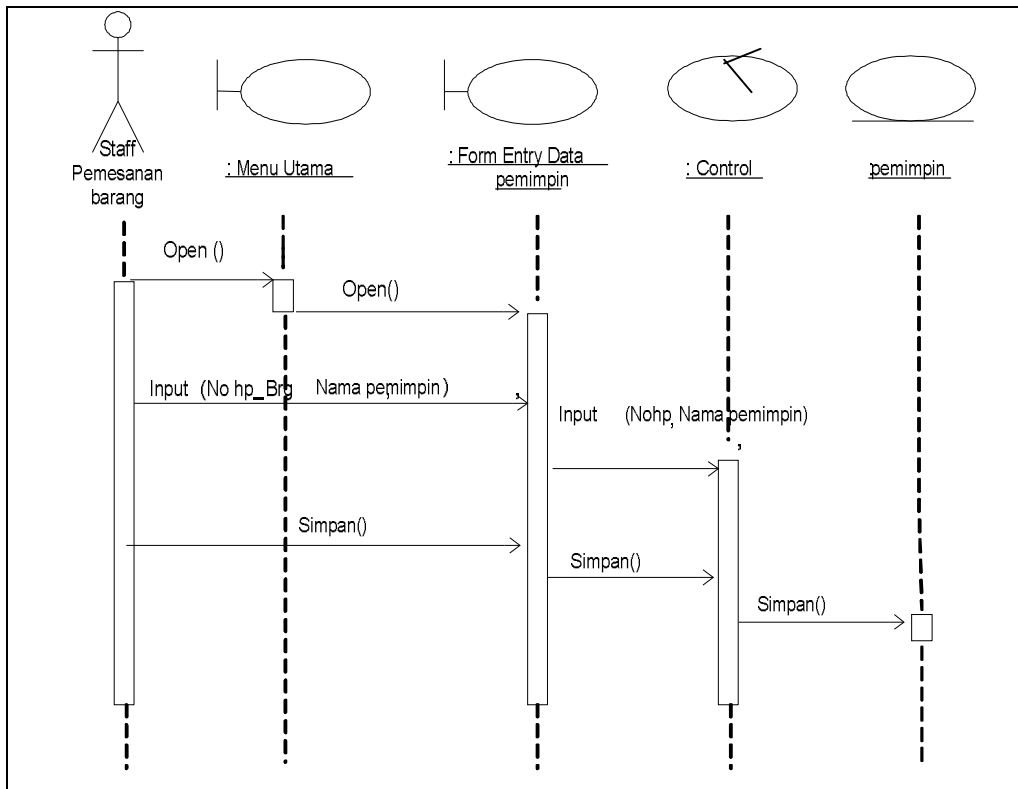
Diagram Sequence entry data barang menerangkan kegiatan yang dilakukan sistem pada saat meng-entry data pelanggan. Lihat gambar 3.6



**Gambar 3.6 :** Diagram Sequence Entry Data Pelanggan

c) Diagram Sequence Entry Data Pemimpin

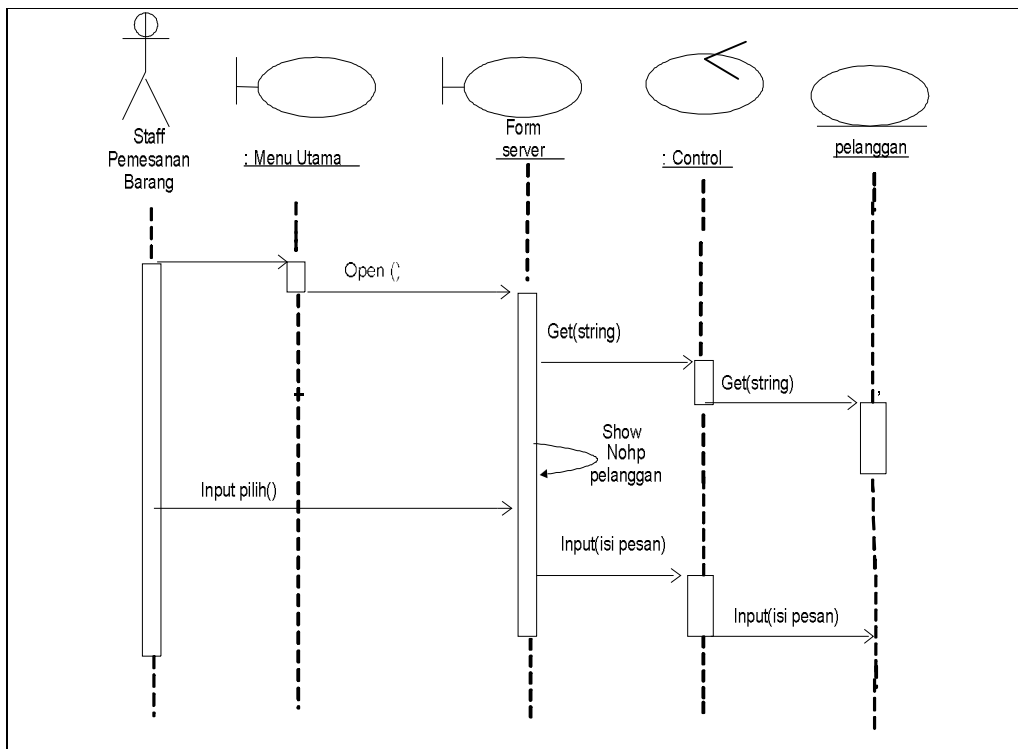
Diagram Sequence entry data pemimpin menerangkan kegiatan yang dilakukan sistem pada saat meng-entry data pemimpin. Lihat gambar 3.7



**Gambar 3.7** Diagram Sequence Entry Data Pemimpin

d) Diagram Sequence Server

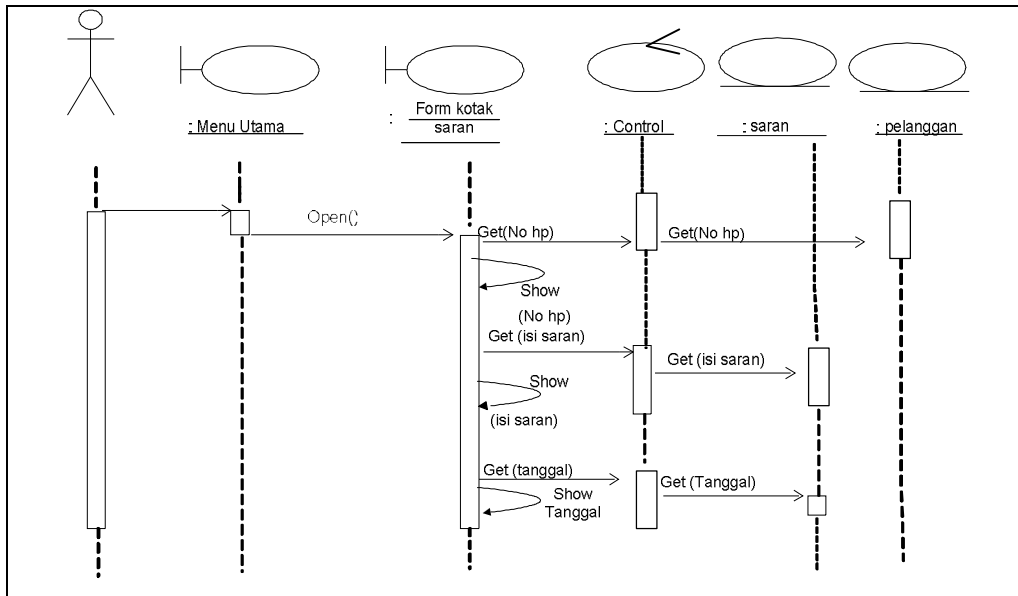
Diagram Sequence server menerangkan kegiatan yang dilakukan pada sistem untuk mengetahui, menerima dan membalas sms. Lihat gambar 3.8



**Gambar 3.8** Diagram Sequence Server

e) Diagram Sequence Saran

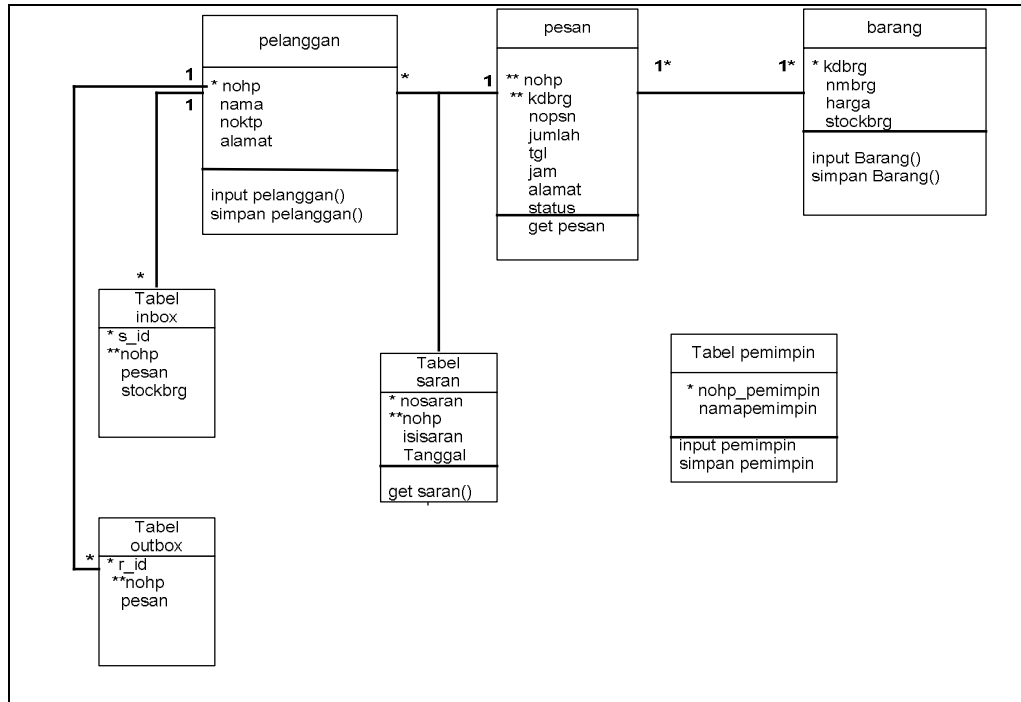
Diagram Sequence saran menerangkan kegiatan yang dilakukan sistem pada saat melihat saran. Lihat gambar 3.9



**Gambar 3.9** Diagram Sequence Saran

### 3.6 Class Diagram

Class Diagram menunjukkan struktur yang statis dari beberapa class dalam suatu sistem. Lihat gambar 3.10

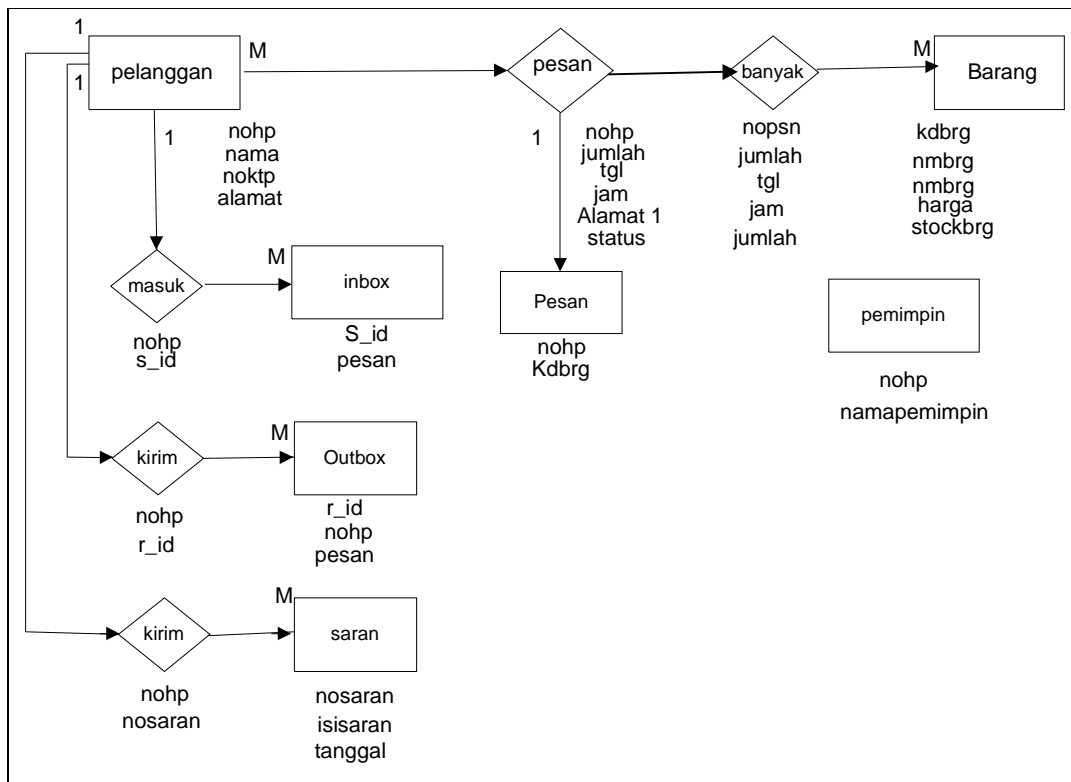


**Gambar 3.10** Class Diagram

### 3.7 Perancangan DataBase

#### 3.7.1 Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) keluaran dari kegiatan analisa terstruktur data yang memodelkan data apa yang ada, dan menggambarkan data dalam keadaan diam, dan bukan bagaimana proses terhadap data itu terjadi. Gambar ERD dapat dilihat pada Gambar 3.9



**Gambar 3.11** Entity Relationship Diagram

#### Penjelasan ERD

1. 1 pelanggan dapat memesan banyak barang
2. 1 pesan dapat mencatat banyak barang
3. 1 pelanggan dapat menjadi banyak inbox
4. 1 pelanggan dapat di simpan dalam banyak outbox



5. 1 pelanggan dapat mengisi banyak saran

### 3.7.2 Normalisasi

Pada keseluruhan tabel penulis ingin membuat tabel tersebut menjadi normal dengan melakukan normalisasi. Dibawah ini merupakan gambar 3.11 NF1.

#### A. Bentuk tidak normal

s_id *	kdbrg*
nohp	nmbrg
pesan	harga
r_id *	stockbrg
nohp	
pesan	
nosaran *	
nohp*	
isisaran	
Tanggal	
nohp*	
nama	
noktp	
alamat	
nohp	
kdbrg	
nopsn*	
jumlah	
tgl	
jam	
alamat1	
status	

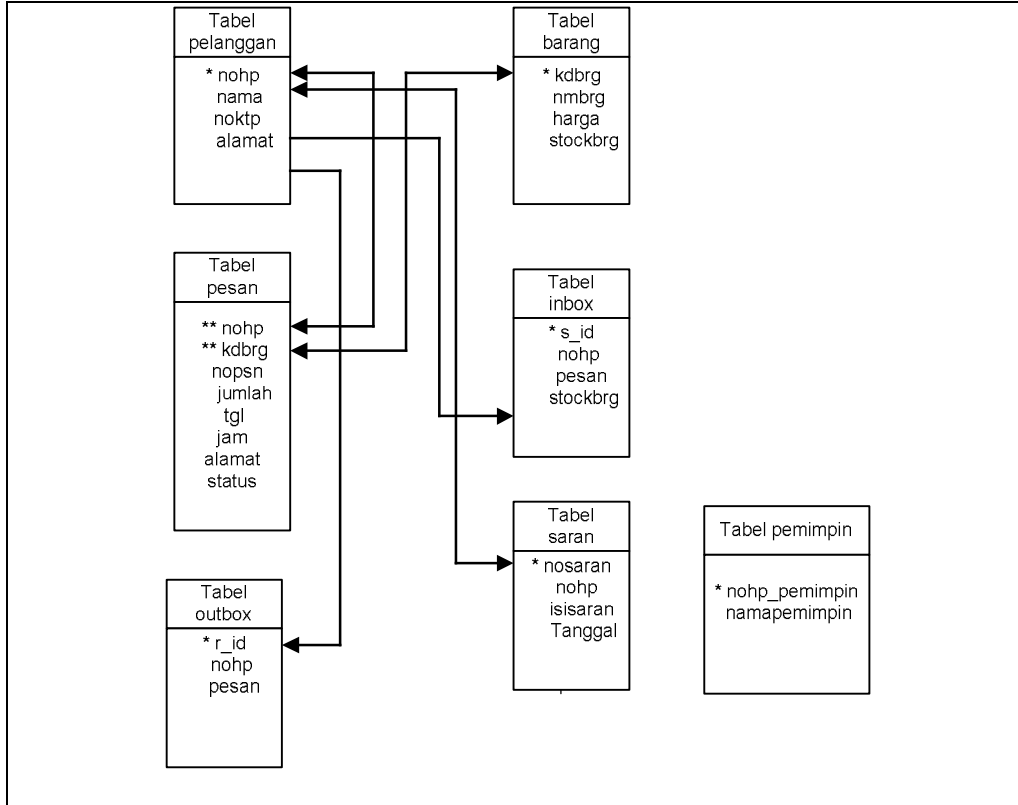
**Gambar 3.12** Bentuk tidak normal

## B. Bentuk 1 NF

*	s_id
	nohp
	Pesan
	r_id
	nosaran
	isisaran
	tanggal
	nama
	noktp
	alamat
*	kdbrg
	nopsn
	jumlah
	jam
	alamat
	status
	nmbrg
	harga
	stockbrg

**Gambar 3.13** Bentuk 1 NF

### C. Bentuk 2 NF



Gambar 3.14 Bentuk 2 NF

### 3.8 Rancangan Tabel

#### Tabel Pelanggan

Tabel pelanggan adalah tabel *master* yang digunakan untuk proses *insert*, *update*, *delete* dan *select* data pelanggan. Dan nohp adalah *primary key* yang digunakan sebagai penghubung ke tabel pesan.

Tabel 3.7 Spesifikasi Tabel Pelanggan

*Primary\_Key*: nohp

Field	Type	Length	Keterangan
Nohp	Varchar	15	No <i>Handphone</i> Pelanggan
Nama	Varchar	20	Nama Pelanggan
Noktp	int	20	No KTP Pelanggan
alamat	Varchar	160	Alamat Pelanggan

#### Tabel Pesan

Tabel pesan adalah tabel transaksi yang digunakan sebagai proses *insert*, *update* dan *select* data pesan. Kemudian nohp, kdbrg dan nopsn adalah *primary key* yang digunakan sebagai penghubung ke tabel pelanggan dan pesan.

Tabel 3.8 Spesifikasi Tabel Pesan

*Primary\_Key*: nohp,kdbrg,nopsn

Field	Type	Length	Keterangan
Nohp	Varchar	15	No <i>Handphone</i> Pelanggan
Kdbrg	Varchar	5	Kode Barang untuk Dipesan
Nopsn	Varchar	5	No Pesan untuk Konfirmasi
jumlah	Int	3	Jumlah Pesan
Tgl	Date	10	Tanggal Pesan
Jam	Time	8	Waktu Pesan
Alamat1	Vachar	160	Alamat Pemesanan
Status	Int	1	Status Pemesanan

#### Tabel Pemimpin

Tabel pimpinan adalah tabel *master* yang digunakan untuk proses *insert*, *update*, *delete* dan *select* data pimpinan.

Tabel 3.9 Spesifikasi Tabel Pemimpin

*Primary\_Key* : nohp

Field	Type	Length	Keterangan
nohp	Varchar	15	Nomor Handpone Pemimpin
namapemimpin	Varchar	50	Nama Pemimpin

### Tabel barang

Tabel barang adalah tabel *master* yang digunakan untuk proses *insert*, *update*, *delete* dan *select* data barang. Kemudian Kdbrg adalah *primary key* yang digunakan sebagai penghubung ke tabel pesan.

Tabel 3.10 Spesifikasi Tabel Barang

*Primary\_Key*: kdbrg

Field	Type	Length	Keterangan
kdbrg	Varchar	5	Kode Barang untuk Dipesan
nmbrg	Varchar	20	Nama Barang
harga	Int	10	Harga Barang
stockbrg	Int	4	Stock Barang

### File Pendukung

Selain tabel-tabel basis data diatas, digunakan tabel-tabel lain untuk keperluan proses terima SMS, kirim SMS, admin, dan saran. Berikut adalah tabel-tabel tersebut.

### Tabel Inbox

Tabel *inbox* adalah tabel pendukung yang digunakan untuk proses *insert* dan *select* pada data *inbox*. Kemudian nohp adalah *primary key* yang digunakan sebagai penghubung ke tabel-tabel lainnya.

Tabel 3.11 Spesifikasi Tabel Inbox

*Primary\_Key* : s\_id

Field	Type	Length	Keterangan
s_id	Int	6	No Id auto_Increment
Nohp	Varchar	15	No <i>Handphone</i> SMS terima
Pesan	Varchar	160	Pesan Masuk

Tabel Outbox

Tabel *outbox* adalah tabel pendukung yang digunakan untuk proses *insert* dan *select* data *outbox*. Kemudian *nohp* adalah *primary key* yang digunakan sebagai penghubung ke tabel-tabel lainnya.

Tabel 3.12 Spesifikasi Tabel Outbox

*Primary\_Key* : r\_id

Field	Type	Length	Keterangan
r_id	Int	6	No Id auto_Increment
nohp	Varchar	15	No <i>Handphone</i> Kirim SMS
pesan	varchar	160	Kirim Pesan

## Tabel Saran

Tabel saran adalah tabel pendukung yang digunakan untuk proses *insert* dan *select* pada data saran.

Tabel 3.13 Spesifikasi Tabel Saran

*Primary\_Key* : nosaran

Field	Type	Length	Keterangan
nosaran	Int	3	No saran
Nohp	Varchar	15	No <i>Handphone</i> Pelanggan
isisaran	Varchar	160	Isi Saran
Tanggal	Date	10	Tanggal Saran

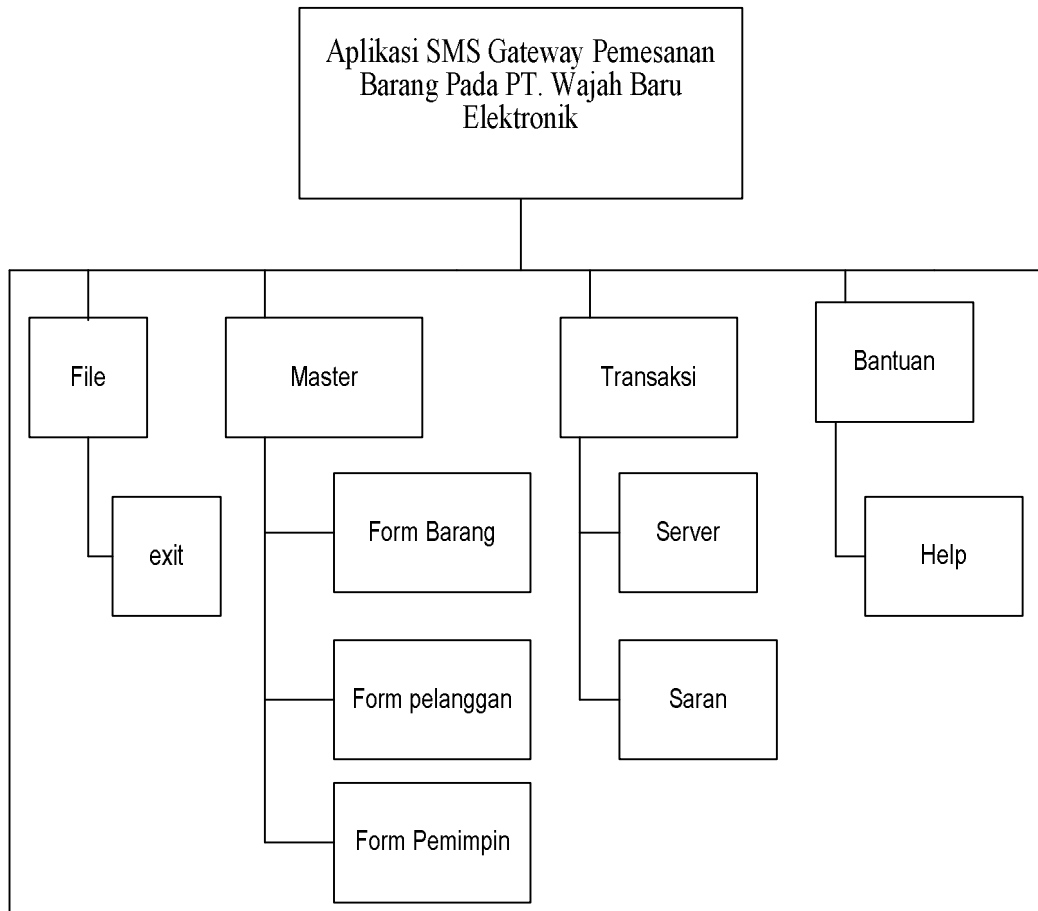
## 3.9 Rancangan layar

Rancangan layar (Interface) merupakan tampilan yang dirancang untuk sistem.

Ada tiga bentuk rancangan antar muka yang dirancang yaitu :

1. Rancangan layar
2. Rancangan Masukan
3. dan Rancangan Keluaran

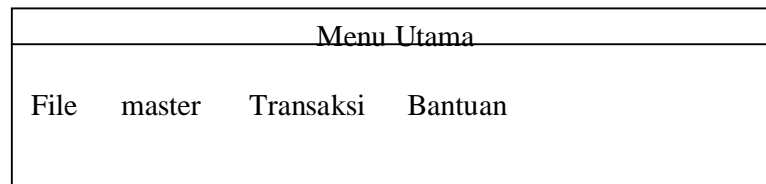




**Gambar 3.15** Struktur Tampilan

a) Rancangan Layar Menu Utama

Rancangan layar menu utama menerangkan tentang rancangan layar pada menu utama pada sistem pemesanan barang. Lihat gambar 3.16



**Gambar 3.16** Rancangan Layar Menu Utama

Pada rancangan Menu Utama terdapat File Master, Pendaftaran, Penyeleksian, Pembagian Kelas, Laporan dan Exit.

b) Rancangan Layar File

Rancangan layar file menerangkan tentang rancangan pada layar file pada program sistem pemesanan barang. Lihat gambar 3.17

Menu Utama			
File	Master	Transaksi	Bantuan
Exit			

**Gambar 3.17** Rancangan Layar File

Pada rancangan layar file terdapat Submenu Exit dimaksudkan untuk menutup semua program.

c) Rancangan Layar Master

Rancangan layar Master menerangkan tentang rancangan pada layar master pada program sistem pemesanan barang. Lihat gambar 3.18

Menu Utama			
File	Master	Transaksi	Bantuan
	Form Barang		
	Form pelanggan		
	Form pemimpin		

**Gambar 3.18** Rancangan Layar Master

Pada rancangan layar master terdapat submenu yang berisi form barang, form pelanggan, dan form pemimpin. Yang masing-masing akan menghubungkan pada layar yang akan dipilih oleh user.

d) Rancangan Layar Form Barang

Rancangan layar form barang menerangkan tentang rancangan pada layar form barang pada program pemesanan barang. Lihat gambar 3.19

The image shows a software interface titled "Entry Data Barang". It features several input fields and buttons. The input fields are labeled "Kd\_brg", "Nm\_brg", "Harga", and "Stock\_brg". There is a "cari" button next to the "Kd\_brg" field. Below the input fields, there are two groups of buttons: "Tombol Kontrol" containing "simpan", "ubah", "Bersih", and "Hapus"; and "Tombol Navigator" containing four arrow buttons for navigation. At the bottom, there is a table with four columns labeled "Kd\_brg", "Nm\_brg", "Harga", and "Stock\_brg".

Kd_brg	Nm_brg	Harga	Stock_brg

**Gambar 3.19** Rancangan Layar Form Barang

Rancangan layar form barang dimaksudkan untuk memasukan data pada barang.

e) Rancangan Layar Form Pelanggan

Rancangan layar form barang menerangkan tentang rancangan pada layar form barang pada program pemesanan barang. Lihat gambar 3.20

Entry Data pelanggan

Nohp\_plg

Nm\_plg

No\_ktp

Alamat

Tombol Kontrol

Tombol Navigator

Nohp\_plg Nm\_plg No\_ktp Alamat

Nohp_plg	Nm_plg	No_ktp	Alamat

**Gambar 3.20** Rancangan Layar Form Pelanggan

Rancangan layar form pelanggan dimaksudkan untuk memasukan data pelanggan.

f) Rancangan Layar Pemimpin

Rancangan layar form barang menerangkan tentang rancangan pada layar form barang pada program pemesanan barang. Lihat gambar 3.21

Entry Data Pemimpin

Input pemimpin

No\_hp

Nama pemimpin

Tombol Kontrol

**Gambar 3.21** Rancangan Layar Form Pemimpin

Rancangan layar form pemimpin dimaksudkan untuk memasukan data pemimpin.

g) Rancangan Layar Transaksi

Rancangan layar Transaksi menerangkan tentang rancangan pada layar Transaksi pada program sistem pemesanan barang. Lihat gambar 3.22

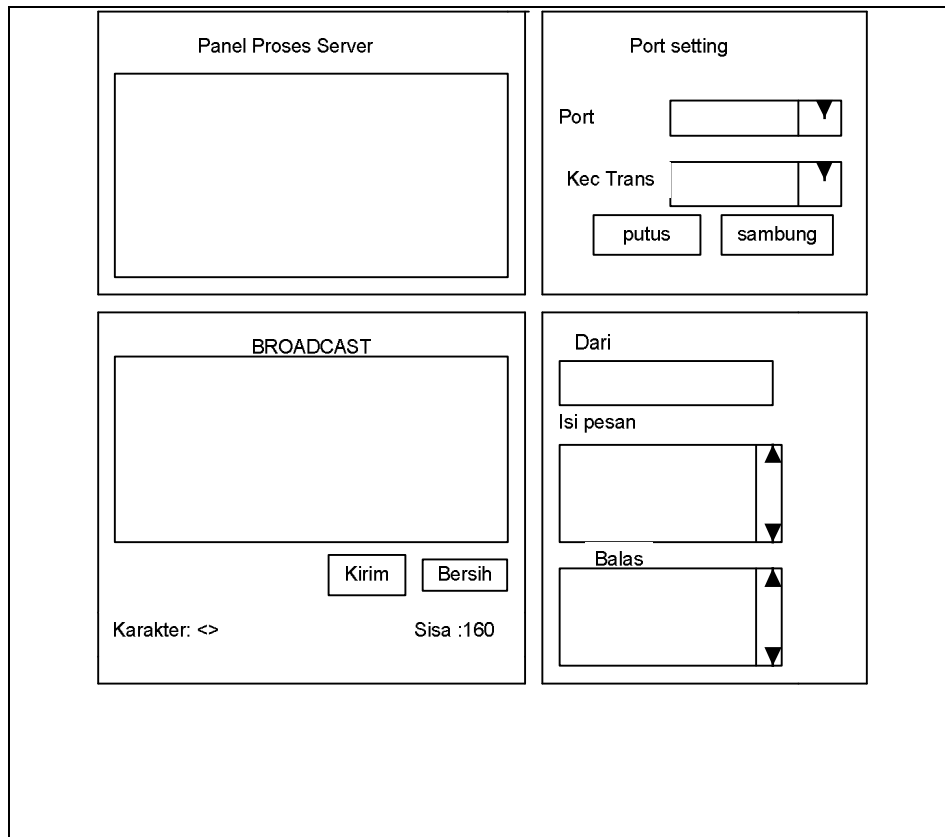
Menu Utama			
File	Master	Transaksi Server Saran	Bantuan

**Gambar 3.22** Rancangan Layar Transaksi

Pada rancangan layar transaksi terdapat submenu yang berisi server dan saran. Yang masing-masing akan menghubungkan pada layar yang akan dipilih oleh user.

h) Rancangan Layar Server

Rancangan layar server menerangkan tentang rancangan pada layar server.pada sistem pemesanan barang. Lihat gambar 3.23



**Gambar 3.23** Rancangan Layar Server

Rancangan layar server dan broadcast untuk mengetahui, menerima dan membalas sms kepada semua pelanggan.

i) Rancangan Layar Saran

Rancangan layar server menerangkan tentang rancangan pada layar server.

Pada sistem pemesanan barang. Lihat gambar 3.24

KOTAK SARAN		
Nohp	Isisaran	Tanggal

**Gambar 3.24** Rancangan Layar Saran

Rancangan layar saran adalah untuk mengetahui pesan kritik dan saran yang masuk ke tabel saran.

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

Tujuan dari pembuatan aplikasi pemanfaatan SMS untuk pemesanan barang elektronik pada Wajah Baru Elektronik adalah untuk memudahkan pelanggan memesan barang dimanapun dan kapanpun mereka butuhkan. Hal yang sebelumnya dilakukan melalui telepon dan memakan banyak biaya, kini dapat dilakukan dengan cepat dan murah.

#### **4.1 Cara Pengoperasian**

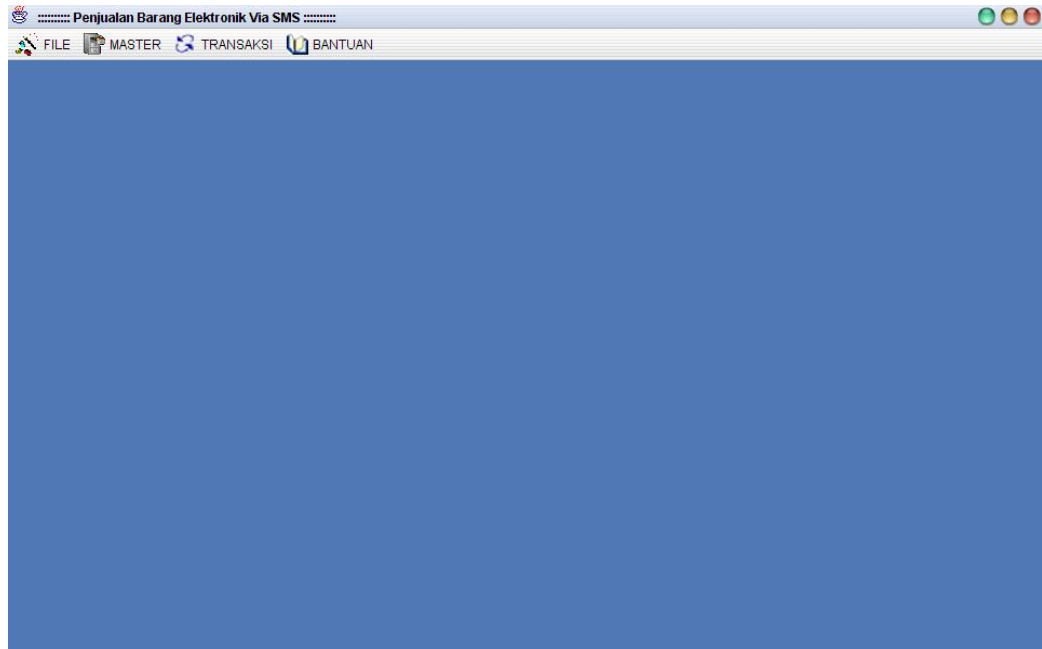
Pada prinsipnya aplikasi ini akan berjalan setelah SMS server di Wajah Baru Elektronik menerima SMS *request* yang masuk. SMS yang masuk ini kemudian diolah untuk mendapatkan informasi yang sesuai sesuai dengan isi permintaan. Setelah itu, secara otomatis aplikasi akan mengirimkan hasil yang diperoleh juga melalui SMS server.

#### **4.2 Tampilan Layar Menu utama**

Untuk dapat melakukan kegiatan yang lainnya akan muncul tampilan awal dari aplikasi ini. Pada menu utama ini terdapat empat menu pilihan diantaranya menu *File* yang terdiri menu pilihan *Exit*, menu yang kedua adalah *Master* terdapat menu pilihan *form* barang, *form* pelanggan, yang ketiga adalah menu transaksi yang terdapat menu pilihan *Server* dan Saran dan menu terakhir adalah menu *System* yang terdapat hanya



satu pilihan yaitu Bantuan, yang berisi panduan pemakaian program. Tampilannya seperti dibawah ini :



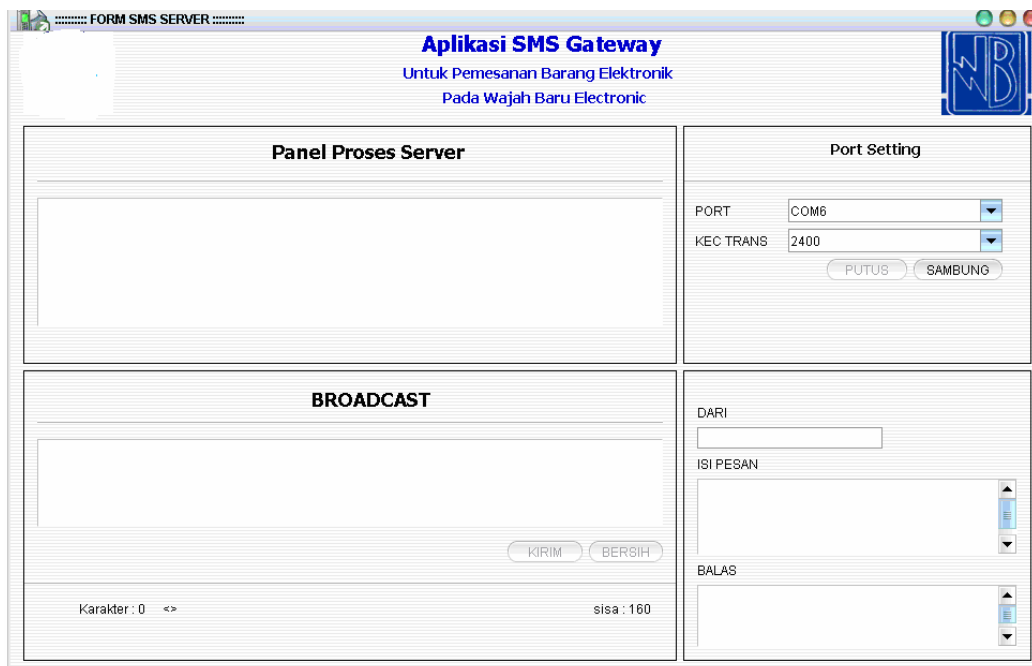
**Gambar 4.1** Layar Menu Utama

### **coding**

```
1. //Tampil layar menu utama
2. public javax.swing.JDesktopPane getDesktop() {
3.     return this.jd;
4. }
5. public void setProteksiMenu(boolean b) {
6.     masterbarang.setEnabled(b);
7.     masterpelanggan.setEnabled(b);
8.     server.setEnabled(b);
9.     help.setEnabled(b);
10.    saran.setEnabled(b);
11.    logon.setEnabled(!b);
12. }
13. // Pemanggilan method keluar atau exit
14. private void Exit(){
```

## A. Tampilan SMS Server

Informasi yang ditampilkan pada layar adalah pesan yang masuk dan keluar. Dengan tombol sambung untuk menyambungkan pada settingan *database*, settingan terminal untuk menerima SMS, list paling bawah untuk menampilkan data-data yang diterima dan dikirimkan.



**Gambar 4.2** Layar SMS Server

Jika SMS tersebut telah disimpan pada tabel *inbox*, maka pada saat program aplikasi masuk dijalankan akan ditampilkan seperti pada Gambar 4.6. berikut ini adalah penggalan program untuk menampilkan informasi SMS yang masuk.

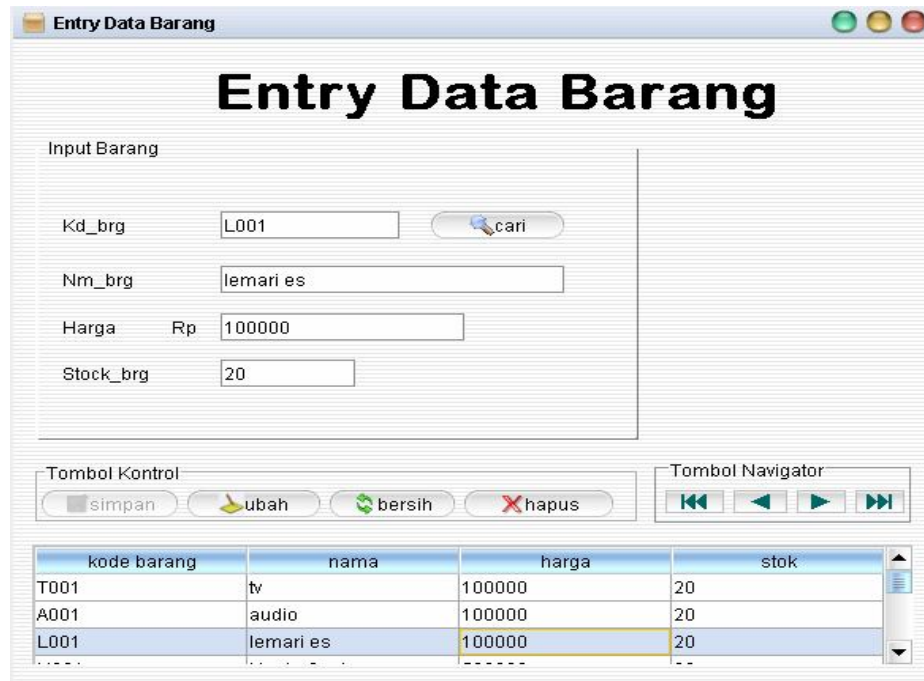
```

1. public void Sambung() {
2.     Thread t=new Thread() {
3.         public void run() {
4.             setProses("Mulai Server");
5. sms.setPort(Port2.getItemAt(Port2.
6. sms.setKecepatanTransper
7. (Trans.getSelectedIndex()).toString()));
8.         sms.setTerminal();
9.         lsProses.setModel(lsm);
10.     }
11. };

```

## **B. Tampilan Master**

Layar Form barang pada Gambar 4.8 untuk maintance Form barang. Tampilan tabel data dari master tersebut. Data yang dapat di tambah, ubah, hapus, bersih dan cari adalah data barang, pelanggan, admin, pimpinan. Pelanggan yang ingin menggunakan SMS untuk pemesanan Barang Elektronik harus terlebih dahulu mendaftarkan data diri mereka ke dengan datang langsung ke Wajah Baru Elektronik. Dibawah ini adalah potongan-potongan program untuk menambah, menghapus, mengubah data.



**Gambar 4.3** Layar Form Barang

Proses menampilkan data pada tabel barang

```

1. //Tampil tabel barang
2. private void tmplbrg() {
3.     DefaultTableModeltblModel=newDefaultTableModel(null,new
   Object[]
4.     {"kode barang","nama","harga","stok"});
5.     Object data[] = new Object[4];
6.     System.err.println(exc.getMessage());
7.     }
8.     }

```



**Gambar 4.4** Layar Form Pelanggan

Proses menampilkan data pada tabel pelanggan

```

1. //Tampil tabel pelanggan
2. private void tmpIPlg() {
3.     DefaultTableModel tblModel=new DefaultTableModel(null,new
4.     Object[]{"nohp","nama","noktp ","alamat"});
5.     tabelPlg.setModel(tblModel);
6. } catch(Exception exc) {
7.     System.err.println(exc.getMessage());
8. }
9. }

```



**Gambar 4.5** Layar Form Pemimpin

```















1. // proses query simpan form pemimpin
2.     private void simpan() {
3.         try {
4.             //untuk memanggil validasi
5.             if(!validasi()) return;
6.             Connection kn=Koneksi.Koneksi.bukaKoneksi();
7.             Statement stt =kn.createStatement();
8.             String sql="INSERT INTO pemimpin
9.             VALUES("+txtnohp.getText()+", "+txtnamapemimpin.getText()+")";
10.            stt.executeUpdate(sql);
11.            kn.close();
12.            JOptionPane.showMessageDialog(null,"Data Telah disimpan");
13.            bersih();

```



**Gambar 4.6** Layar Form Saran

### C. Tampilan Bantuan

Fungsi dari menu-menu SMS-GATEWAY WAJAH BARU ELECTORNIC	
MENU	FUNGSI
 FILE	Berisikan tampilan Log In, Log Off dan Keluar dari aplikasi ini
 MASTER	Berisikan tampilan master barang, master pelanggan dan master admin
 TRANSAKSI	Berisikan tampilan form sms server
 BANTUAN	Berisikan tampilan bantuan
 LOG IN	Tampilan form Log In
 LOG OFF	Tampilan form Log Off
 EXIT	Tampilan buat keluar dari aplikasi ini
 FORM BARANG	Tampilan buat memasukan data-data barang
 FORM PELANGGAN	Tampilan buat memasukan data-data pelanggan
 FORM ADMIN	Tampilan buat memasukan data-data admin
 FORM PEMIMPIN	Tampilan buat memasukan data-data pemimpin
 SERVER	Tampilan buat SMS Server
 SARAN	Tampilan buat Saran
 HELP	Tampilan buat HELP

**Gambar 4.7** Layar Bantuan

### 4.3 Pemrosesan Pesan dan Tampilan Pada Handphone

Pemrosesan pesan dilakukan pada modul-modul yang berbeda sesuai dengan isi permintaan. Pada tampilan layar SMS masuk dibawah terdapat 8 SMS:

#### 1) PESAN

SMS yang berisi permintaan “PESAN” akan diproses pada modul method pesan.

```

1. // Method pesan
2. public String Pesan() {
3. //Mengecek Pelanggan
4. if(!plg.isPelanggan()) {
5. return "Maaf Anda Belum Terdaftar Menjadi
6. Pelanggan";
7. }
8. if(_data.length!=2){
9. return "Format Pesan anda salah";
10. }

```



**Gambar 4.8** Tampilan SMS Balasan Pesan

**PESAN<spasi> Kode Barang<spasi>Jml Barang#Alamat**

## 2) BATAL

SMS yang berisi permintaan “BATAL” akan diproses pada modul method batal.

```

1. // Method batal
2. public String Batal() {
3. String
4. String kode_psn;
5. if(index == -1) {
6. kode_psn=potong.trim();
7. }

```





**Gambar 4.9** Tampilan SMS Balasan Batal

**BATAL<spasi>No Pesanan**

3) LAPORAN

SMS yang berisi permintaan "LAPORAN" akan diproses pada modul method laporan.

```

1. // Method laporan
2. //sql = "select SUM(Total)as Total from pesanan where
3. Tgl_Psn between '"+tgl2+" 00:00:00' and '"+tgl1+"
   23:59:59";
4. }
5. kn.close();
6. laporan = "Pendapatan Sebesar =Rp."+
7. }
8. }
9. return back;
10. }
  
```



**Gambar 4.10** Tampilan SMS balasan laporan

**LAPORAN<spasi>dd-mm-yyyy contoh:LAPORAN<spasi>19-03-2008**

#### 4) HELP

SMS yang berisi permintaan “HELP” akan diproses pada modul method help.

```
1.     else {
2.         return "pesanktkPESAN<sps>Kd Brg<sps>jumlah,Kd
3.         Brg<sps>jml#alm"+
4.             "infoktkINFO<sps>kdBrg" +
5.             "helpktkHELP"+
6.             "batalktk BATAL<sps>kd pesan"+
7.             "regktk<sps>nm#noktp#alm"+
8.             "berhenti unreg";
9.     }
10. }
```



**Gambar 4.11** Tampilan SMS balasan help

#### HELP

#### 5) SARAN

SMS yang berisi permintaan “SARAN” akan diproses pada modul method saran.

```

1. // Method Saran
2. public String Metodesaran(String isi,String nohpKu){
3. String SQL = "INSERT INTO saran (nohp, isisaran, tanggal)
4. ("yyyy-MM-d").format(new Date()+")";
5. back = "Saran Anda telah kami terima, terima kasih atas saran
        anda";
6.     }
7.     return back; }}

```



**Gambar 4.12** Tampilan SMS balasan saran

#### **SARAN<spasi>Isi saran**

##### 6) UPDATE

SMS yang berisi permintaan “UPDATE” akan diproses pada modul method update.

```

1. // Method update
2.     if(plg.isPelanggan()) {
3.         int jaw=plg.Ubah();
4.         return "Terimakasih Anda Merubah data anda";
5.     }
6.
7.     else return "Proses Gagal";
8.     }

```



**Gambar 4.13** Tampilan SMS balasan update

**UPDATE**<spasi>Nama#NoKTP#Alamat

7) UNREGISTER

SMS yang berisi permintaan “UNREGISTER” akan diproses pada modul method unregister.

```

1. // Method unregister
2.         if(jaw != -1) return "Terimakasih anda pernah menjadi
3.         pelanggan kami..";
4.         else return "Un Register GAGAL.";}

```



**Gambar 4.14** Tampilan SMS balasan unregister

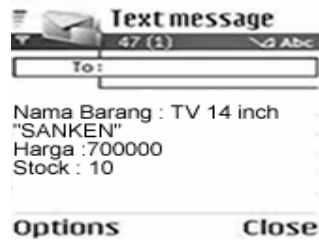
8) INFO

SMS yang berisi permintaan “INFO” akan diproses pada modul method info.

```

1. // Method info
2. public String Barang() {
3.     String kd_brg[]=pesan.split(" ");
4.     Koneksi.barang barang=new Koneksi.barang();
5.     barang.setKodeBarang(kd_brg[1]);
6.     }

```



**Gambar 4.15** Tampilan SMS balasan info

#### 10) INFO<spasi>Kode Barang

#### 4.4 Metode Pengujian

Metode *White Box* yaitu metode perancangan test case yang digunakan untuk menggunakan struktur kontrol dari perancangan prosedural untuk mendapatkan test case. Dengan menggunakan metode pengujian *white box test case* dapat melakukan :

Memberikan jaminan bahwa semua jalur *independent* pada suatu modul telah digunakan paling tidak satu kali. Menggunakan semua keputusan logis pada sisi *true* dan *false* Mengeksekusi semua *loop* pada batasan mereka dan batasan operasional mereka. Menggunakan struktur data internal untuk menjamin validitasnya (Roger S. Pressman Ph.D.2002:533).

Metode pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak, pengujian *black box* bukan merupakan alternatif dari teknik *white box* tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu menemui kesalahan daripada metode *white box*.

Pengujian *black box* dapat menemukan kesalahan dalam kategori sebagai

berikut :

1. Fungsi yang tidak benar atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Pengujian program aplikasi SMS Gateway ini dilakukan dengan menggunakan metode *Black box Testing*. Metode *Black Box Testing* ini merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *Black Box Testing* ini adalah untuk menemukan kesalahan fungsi pada program.

Pengujian dengan menggunakan metode *black box testing* dilakukan dengan cara memberikan sejumlah masukan (*input*) pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi SMS Gateway menghasilkan keluaran (*output*) yang diinginkan sesuai dengan fungsi dari program aplikasi SMS Gateway tersebut. Bila dari input yang diberikan proses menghasilkan output yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi yang penulis rancang berhasil tetapi bila output yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi tersebut.

#### **4.41 Skenario Pengujian**

Skenario pengujian dilakukan untuk menentukan langkah-langkah dalam melakukan pengujian. Pengujian dilakukan dengan menjalankan aplikasi pemesanan barang tersebut. Setelah aplikasi dijalankan selanjutnya adalah menguji tombol-

tombol yang terdapat dalam antar muka aplikasi tersebut apakah sesuai dengan tahap perancangan. Skenario pengujian aplikasi pemesanan barang dapat dilihat pada tabel

**Tabel 4.1** Tabel Skenario Pengujian Aplikasi SMS Gateway untuk Pemesanan Barang pada PT. Wajah Baru Elektronik

No	Antar Muka yang diuji	Bagian dari antar muka yang diuji	Status aplikasi	Skenario pengujian	Hasil yang diharapkan
1	Halaman utama	Master	Aplikasi belum terbuka secara keseluruhan	Klik menu transaksi	Form server terbuka
2	Sambung koneksi ke handphone	Menguji port setting dan kecepatan transmisi	Server sedang berjalan	Klik tombol sambung	Koneksi dapat tersambung
3	Mengirim pesan ke semua pelanggan melalui layar broadcast	Menguji tombol kirim broadcast	Server sedang berjalan	Untuk memberikan informasi kepada pelanggan	Pelanggan yang telah terdaftar dapat menerima sms tersebut.
4	Cari dan simpan data barang, kode barang, nama barang, harga barang, stok barang	Menguji tombol-tombol yang terdapat didalam antar muka tersebut	Form entry data barang sedang berjalan	Memasukan data-data sesuai dengan form tersebut	Tersimpan kedalam data base dan, tombol berfungsi sebagaimana fungsi masing-masing
5	Cari dan simpan data pelanggan, no hp pelanggan, nama pelanggan, no ktp, alamat pelanggan	Menguji tombol-tombol yang terdapat di dalam antar muka tersebut	Form entry data pelanggan sedang berjalan	Memasukan data-data sesuai dengan form tersebut	Tersimpan kedalam data base dan, tombol berfungsi sebagaimana fungsi masing-masing

6	Menambah, mengedit, cari dan delete data barang	Menguji tombol-tombol yang terdapat di dalam antar muka tersebut agar dapat merubah isi table setiap data tersebut	Form tersebut sedang berjalan	Mengganti, menambah data dan menyimpan kembali	Berubah sesuai data yang disimpan.
7	Ubah dan hapus data pelanggan	Menguji tombol-tombol yang terdapat didalam antar muka tersebut agar dapat merubah isi table setiap data tersebut	Form tersebut sedang berjalan	Mengganti data dan menyimpan kembali	Berubah sesuai data yang diubah.

#### 4.42 Hasil Pengujian

Pengujian yang penulis terapkan pada aplikasi ini dengan menggunakan teknik *black box*. Teknik *black box* adalah pengujian yang dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses yang diinginkan. Pengujian dilakukan untuk menguji fungsi-fungsi khusus dari aplikasi yang dirancang. Dari keluaran yang dihasilkan, kemampuan program dalam memenuhi kebutuhan pemakai dapat diukur sekaligus dapat diketahui kesalahan-kesalahannya. Di bawah ini merupakan hasil pengujian dengan menggunakan teknik *black box* :

**Tabel 4.2** Tabel hasil pengujian Aplikasi SMS Gateway untuk Pemesanan Barang pada PT. Wajah Baru Elektronik Dengan Metode *Black Box*

No	Antar Muka	Tujuan	Hasil pengujian
----	------------	--------	-----------------



	yang diuji		
1	Halaman utama	Master	Halaman master akan terbuka dalam 1 detik
2	Halaman master	Membuka form barang	form barang akan terbuka dalam 1 detik
3	form barang	Menyimpan data barang	Layar konfirmasi data telah di simpan dalam 1 detik
4	Halaman master	Membuka form pelanggan	form pelanggan akan terbuka dalam 1 detik
5	Form pelanggan	Menyimpan data pelanggan	Layar konfirmasi data telah di simpan dalam 1 detik
6	Halaman master	Membuka form pemimpin	Form pemimpin akan terbuka dalam 1 detik
7	Form pemimpin	Menyimpan data pemimpin	Layar konfirmasi data telah di simpan dalam 1 detik
8	Halaman transaksi	Membuka server	Server terbuka dalam 1 detik
9	Server	Mengatur port setting	Layar konfirmasi server dapat digunakan
10	Halaman saran	Melihat saran yang masuk	Layar konfirmasi saran dapat di lihat
11	Bantuan	membantu melihat fungsi menu	Layar konfirmasi dapat di lihat

#### 4.4.3 Analisis Hasil Pengujian

Setelah dilakukan pengujian terhadap semua menu sistem pada program pemesanan barang ini, maka dapat disimpulkan bahwa pengujian telah menunjukkan hasil keluaran (Output) dan proses yang sesuai dengan rancangan aplikasi program ini. Dan hasil dari pengujian dapat dikatakan bahwa program ini dapat berfungsi dengan baik dan benar, karena telah dibuktikan dari hasil skenario pengujian.

Bahwa program aplikasi SMS Gateway ini dapat digunakan sebagai kontrol dalam melakukan kegiatan pemesanan barang serta dapat menunjang proses pemesanan barang kepada masyarakat yang ingin melakukan transaksi pembelian barang elektronik pada PT. Wajah Baru Elektronik.

## **BAB V**

### **PENUTUP**

Berdasarkan analisa yang telah dilakukan terhadap permasalahan dari aplikasi yang dibuat, maka dapat ditarik kesimpulan dan saran yang mungkin diperlukan untuk pengembangan sistem ketahap yang lebih kompleks.

#### **1. Kesimpulan**

Dari hasil analisa terhadap permasalahan dan aplikasi yang dikembangkan, maka dapat ditarik kesimpulan:

1. Penulis ingin mempublikasikan PT. Wajah Baru Elektronik ke masyarakat
2. Penulis merancang sebuah aplikasi SMS Gateway
3. Memudahkan pelanggan dalam melakukan transaksi pemesanan barang
4. Dapat mengetahui info jumlah dan harga barang dengan mudah, cukup dengan melalui *handphone*.
5. Kesalahan-kesalahan yang disebabkan oleh keterbatasan seperti *human Error* ataupun ketidaksengajaan dapat dikurangi seperti kesalahan pada saat menginput data-data barang dan pelanggan dapat langsung dikoreksi dengan menekan tombol hapus yang terdapat dalam program

#### **2. Saran**

Selain menarik beberapa kesimpulan, dibawah ini diberikan saran-saran yang mungkin bisa dijadikan pertimbangan dalam pengembangan sistem. Saran-saran tersebut diantaranya adalah:

- a. Agar aplikasi dapat berjalan dengan benar dan mempunyai waktu proses yang cepat maka spesifikasi kebutuhan program harus dipenuhi.
- b. Perlu ditambahkan fitur-fitur pengiriman informasi yang lebih lengkap, sehingga mampu memenuhi kebutuhan yang lebih kompleks.
- c. Tanpa adanya perawatan dan pengawasan dari pihak yang bertanggung jawab dalam pemeliharaan sistem, maka sistem tidak akan dapat berjalan dengan baik dan lancar.
- d. Fitur yang disediakan masih terbatas pada pemesanan, informasi, laporan. Maka perlu ditambah informasi-informasi lain yang berhubungan dengan pemesanan daftar barang tersebut.

## DAFTAR PUSTAKA

Hendry.2008.*Belajar Otodidak Java dengan Netbeans*.Jakarta:Elex Media Komputindo.

Fowler,Martin.2005.*UML Distilled Edisi 3*.Yogyakarta:ANDI.

Gunawan Ferry.2003.*Membuat Aplikasi SMS Gateway Server dan Client dengan Java dan PHP*.Jakarta:Elex Media Komputindo.

Mulyana Y.B, S.Kom.2003.*Trik Membangun Situs Menggunakan PHP dan MYSQL*.Jakarta:Elex Media Komputindo.

Purnama Rangsang.2003.*Tuntunan Pemrograman Java Jilid 2*.Surabaya:Tim Prestasi Pustaka.

Purnama Rangsang.2003.*Tuntunan Pemrograman Java Jilid 3*.Surabaya:Tim Prestasi Pustaka.

Komputer Wahana.2005.*Pengembangan Aplikasi Sistem Akademik Berbasis SMS dengan Java*.Jakarta.Salemba Infotek.

Pressman,Roger.S.2007.*Rekayasa Perangkat Lunak Edisi 1*.Yogyakarta:ANDI.

Jogiyanto.2005.*Analisis dan Desain Sistem Informasi*.Yogyakarta:ANDI.

Ridwan Sanjaya.2003.*Membuat Aplikasi Windows Multiplatform dengan Java GUI*.Jakarta:Elex Media Komputindo.