

## BAB II LANDASAN TEORI

### 2.1 Definisi

#### 2.1.1 Definisi Travel

Travel Agent merupakan usaha yang bergerak di bidang jasa yang memiliki tujuan untuk menyiapkan suatu perjalanan bagi seseorang yang merencanakan untuk mengadakannya (Nyoman,1994). Bisnis *travel agent* di Indonesia mengalami perkembangan sangat pesat. Dilihat dari kenaikan perusahaan *travel agent* pada tahun 2007 sampai 2011. Pada tahun 2007 perusahaan *travel agent* berjumlah 655, terjadi peningkatan 24,43 % menjadi 815 perusahaan di tahun 2008. Di tahun 2009 kembali meningkat yakni 16,81 % menjadi 952 perusahaan , terjadi peningkatan 17,23 % menjadi 1.116 perusahaan di tahun 2010, menurut (,2012) data terakhir diambil pada tahun 2011 dimana pada tahun tersebut kenaikan mencapai 0,36% dan jumlah menjadi 1120 perusahaan. Pada saat ini salah satu bisnis *travel agent* yang sedang berkembang pesat di DKI Jakarta adalah bisnis jasa transportasi yang mengusung konsep *shuttle service point-point* atau lebih dikenal sebagai *shuttle travel*. *Shuttle travel* mempunyai beberapa keunggulan antara lain jadwal tersedia hampir setiap waktu dan waktu perjalanan lebih singkat dibanding modal transportasi umum yang lain. Dari segi kenyamanan dan keamanan, *shuttle travel* menawarkan fasilitas yang lebih baik dibanding bus maupun kereta api,walaupun dari sisi biaya *shuttle travel* relatif lebih mahal. Peluang inilah yang ditangkap oleh para pelaku bisnis *shuttle travel* yang mulai menjamur di ibu kota Jakarta.

Di Kepulauan Bangka Belitung untuk perusahaan *travel agent* juga sudah mulai banyak yang membuka dengan peluang yang sangat besar mengingat banyaknya pariwisata, serta kota yang sudah mulai berkembang dengan begitu masyarakat mempunya daya minat dalam menggunakan layanan *travel agent*.

### **2.1.2 Definisi Sistem**

Terdapat dua kelompok pendekatan didalam mendefinisikan sistem, yaitu menekankan pada prosedurnya dan yang menekankan pada komponen atau yang elemennya. Pendekatan sistem yang lebih menekan pada prosedur mendefinisikan sistem sebagai berikut : “suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”

Menurut Jerry Fitzgerald Ardra F. Fitzgerald dan Warren D Stallings, Jr. mendefinisikan prosedur sebagai berikut: “suatu prosedur adalah urutan yang tepat dari tahapan tahapan intruksi yang menerangkan Apa yang harus dikerjakan, Siapa yang mengerjakan, Kapan dikerjakan dan Bagaimana mengerjakannya.

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut ini :

Sistem adalah kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Kedua kelompok definisi tersebut adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya. Pendekatan sistem yang merupakan kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang luas. Definisi ini lebih banyak diterima, karena kenyataanya suatu sistem dapat berdiri dari beberapa subsistem atau sistem bagian.

### **2.1.3 Definisi Infromasi**

Menurut Viktor Nicolas Nore, 2013, Mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini dan saat mendatang.

#### **2.1.4 Konsep dasar Sistem Informasi**

Menurut Viktor Nicolas Nore, 2013, Sistem informasi adalah suatu system didalam suatu organisasi yang mempertemukan kebutuhan pengelola transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan bagi pihak luar tertentu dengan laporan-laporan yang diperkirakan.

#### **2.1.5 SMS (SHORT MESSAGE SERVICE)**

##### **2.1.5.1 Definisi SMS**

Menurut Yudi Wiharto, 2011, *Short Message Service* (SMS) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam bentuk teks. SMS didukung oleh GSM (*Global System For Mobile Communication*), TDMA (*Time Division Multiple Access*), CDMA (*CodeDivision Multiple Access*) yang berbasis pada telepon seluler yang saat ini banyak digunakan. SMS (*Short Message Service*) adalah merupakan salah satu layanan pesan teks yang dikembangkan dan distandarisasi oleh suatu badan yang bernama ETSI (*European Telecommunication Standards Institute*) sebagian dari pengembangan GSM (*Global System for Mobile Communication*) Phase 2, yang terdapat pada dokumentasi GSM 03.40 dan GSM 03.38. Fitur SMS ini memungkinkan perangkat Stasiun Seluler Digital (*Digital Cellular Terminal*, seperti Ponsel) untuk dapat mengirim dan menerima pesan-pesan teks dengan panjang sampai dengan 160 karakter melalui jaringan GSM. SMS dapat dikirimkan ke perangkat stasiun seluler digital lainnya hanya dalam beberapa detik selama berada pada jangkauan pelayanan GSM. Lebih dari sekedar pengiriman pesan biasa, layanan SMS memberikan garansi SMS akan sampai pada tujuan meskipun perangkat yang dituju sedang tidak aktif yang dapat disebabkan karena sedang dalam kondisi mati atau berada di luar jangkauan layanan GSM. Dengan adanya *feature* seperti ini maka layanan SMS juga cocok untuk dikembangkan sebagai aplikasi-aplikasi seperti: *pager*, *e-mail*, dan notifikasi voice mail, serta layanan pesan banyak

pemakai (*multiple user*). Namun pengembangan aplikasi tersebut masih bergantung pada tingkat layanan yang disediakan oleh operator jaringan.

#### 2.1.5.2 Karakteristik SMS

Menurut Yudi Wiharto, 2011, Karakteristik utama SMS adalah SMS merupakan sebuah sistem pengiriman data dalam paket yang bersifat *out-of-band* dengan *bandwith* kecil. Dengan karakteristik ini, pengiriman suatu burst data yang sangat pendek dapat dilakukan dengan efisiensi yang sangat tinggi.

#### 2.1.5.3 Keuntungan SMS

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 *customer*.

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

#### 2.1.5.4 Cara Kerja SMS

Dalam sistem SMS, mekanisme utama yang dilakukan dalam suatu sistem adalah melakukan pengiriman short message dari satu terminal customer 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.

## 2.1.6 Sistem Kerja SMS

### 2.1.6.1 AT Command

Menurut Yudi Wiharto, 2011, 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 (Zvonar, 2000)*. 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 2.1: AT Command**

Perintah	Kegunaan
AT	
AT	Mengecek apakah <i>handphone</i> telah terhubung
AT+CMGF	Menetapkan format <i>mode</i> dan terminal
AT+CSCS	Menetapkan Jenis <i>encoding</i>

AT+CNMI	Mendetikasi pesan SMS yang baru masuk secara otomatis
AT+CMGL	Membuka daftar SMS yang ada pada SIM Card
AT+CMGS	Mengirim pesan SMS
AT+CMGR	Mengirim 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 : ATE1 digunakan untuk menanyakan *status port*. Output "OK" akan tampak dilayar jika kondisi port tersebut siap untuk berkomunikasi.

#### 2.1.6.2 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. 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 periode *validity* terpenuhi.

#### 2.1.6.3 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 :

1. Menggunakan Terminal Baik Berupa GSM Modem atau *Handphone* Cara ini adalah yang paling mudah tetapi memiliki kekurangan antara lain jumlah pesan yang dikirim permenit sangat terbatas(sekitar 6-10 pesan permenit). Untuk mengantisipasi hal ini biasanya digunakan lebih dari satu terminal.
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. Untuk melakukan koneksi ke SMSC diperlukan protokol penghubung. Protokol yang umum digunakan adalah UCP, SMPP, CIMD2, OIS dan TAP. Masing-masing operator GSM menyediakan tipe protokol yang berbeda-beda.
3. Menggunakan *Software* bantu Saat ini banyak vendor telekomunikasi menawarkan *software* bantu untuk melakukan koneksi ke SMSC, dari yang bersifat *freeware*, *open source* sampai dengan komersial. Pemilihan koneksi ke SMSC biasanya disesuaikan dengan jumlah pesan SMS yang akan dikirim.

#### 2.1.6.4 PDU Sebagai Bahasa SMS

Data yang mengalir ke atau dari SMSC harus berbentuk PDU (*Protocol Data Unit*). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri dari beberapa *header* yang berbeda pada saat kirim SMS ke SMSC dengan saat SMS diterima dari SMSC. PDU untuk mengirim SMS terdiri dari dari delapan *header*, sebagai berikut :

##### 1. Nomor SMSC

Header pertama ini terdiri dari tiga *subheader* yaitu :

- a. Jumlah pasangan heksadesimal SMSC dalam bilangan heksa.
- b. *National* atau *Internasional code*
- c. Nomor SMSC dalam pasangan heksa dibalik-balik, jika tertinggal satu angka yang tidak memiliki pasangan, angka tersebut akan dipasangkan

dengan huruf F didepannya. Berikut ini adalah daftar SMSC pada beberapa operator Indonesia dan cara penulisannya:

**Tabel 2.2 : Daftar SMSC**

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

**2. Tipe SMS**

Untuk send tipe SMS = 1. Jadi bilangan heksanya adalah 01.

**3. Nomor Referensi SMS**

Nomor referensi ini diartikan 0, jadi bilangan heksanya adalah 00. Nanti akan diberikan sebuah nomor referensi otomatis oleh ponsel atau alat SMS gateway.

**4. Nomor ponsel penerima**

Sama seperti cara penulisan PDU header untuk SMSC, header ini juga terbagi atas tiga bagian, yaitu :

- a. Jumlah pasangan heksadesimal nomor ponsel yang dituju dalam bilangan heksa.
- b. *National* atau *International code*
- c. Nomor ponsel yang dituju dalam pasangan heksa dibalik-balik.

**5. Bentuk SMS**

00 dikirim sebagai SMS, 01 dikirim sebagai telex, 02 dikirim sebagai fax.

**6. Skema encoding data I/O**

Yaitu skema 7 bit dan 8 bit

**7. Jangka waktu sebelum SMS Expired**



Jika bagian ini di-skip, berarti waktu berlakunya SMS tidak dibatasi, sedangkan jika diisi dengan suatu bilangan integer yang kemudian diubah ke pasangan heksa tertentu, bilangan tersebut akan mewakili jumlah waktu validitas SMS tersebut.

## 8. Isi SMS

Header ini terdiri dari dua *subheader*, yaitu :

- a. Panjang isi (jumlah huruf dari isi) Contoh : "hello" = 5 huruf ( 05 heksa ).
- b. Isi berupa pasangan bilangan heksa.

Sesungguhnya, terdapat dua mode untuk mengirim dan menerima SMS, yaitu *mode* teks dan *mode* PDU (*Protocol Data Unit*). Akan tetapi, sistem *mode* teks tidak didukung oleh semua operator GSM maupun terminal. Pada terminal, kita dapat mengecek menggunakan perintah "AT+CMGF=1". Jika hasilnya error, dapat dipastikan bahwa terminal Anda tidak mendukung *mode* teks.

### 1. Text Mode

*Mode* ini adalah cara termudah untuk mengirim pesan. Pada *mode* teks pesan yang kita kirim tidak dilakukan konversi. Teks yang dikirim tetap dalam bentuk aslinya dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Sesungguhnya, *mode* teks adalah hasil encode yang direpresentasikan dalam bentuk format PDU. Kelemahannya, kita tidak dapat menyisipkan gambar dan nada dering ke dalam pesan yang akan dikirim serta terbatasnya tipe encoding.

### 2. PDU (*Protocol Data Unit*)

*Mode* PDU *Mode* adalah format message dalam heksadesimal octet dan semi-decimal octet dengan panjang mencapai 160 (7 bit default alphabet) atau 140 (8 bit) karakter. Kelebihan menggunakan *mode* PDU adalah kita dapat melakukan encoding sendiri yang tentunya harus pula didukung oleh *Hardware* dan operator GSM, melakukan kompresi data, menambahkan nada dering dan gambar pada pesan yang akan dikirim. Beberapa tipe encoding yang umum digunakan adalah "PCCP437", "PCDN", "8859-1", "IRA" dan "GSM". Anda dapat mengeceknya dengan perintah "AT+CSCS". Kita juga dapat menambahkan header ke dalam pesan yang akan dikirim, seperti timestamp, nomor SMSC dan

meta-informasi lainnya. Keterangan lengkap mengenai mode ini dapat dilihat pada ETSI GSM 03.40 dan GSM 03.38. Aplikasi yang akan dibuat menggunakan *mode PDU dengan encoding 7 bit default alphabet*.

### 3 SMS Deliver PDU (Mobile Terminated)

SMS *Deliver* PDU ialah terminal menerima pesan yang datang atau masuk dari SMSC dalam format PDU. Gambar 2 dibawah merupakan gambar dari skema format SMS deliver PDU.

**Tabel 2.3: Skema Format SMS PDU Penerima**

SCA	PDUType	MR	DA	PID	DCS	VP	UDL	UD
-----	---------	----	----	-----	-----	----	-----	----

Contoh: Kita menerima pesan dari 628122888374 dengan isi pesan SMS adalah 'hellohello" pada tanggal 6 Januari 2004 pukul 16.22 wib. Maka format PDU adalah: 06912618010000040C912618228838470000401060612202820AE8329BFD4697D9EC37.

Dibawah ini merupakan penjelasan masing-masing format dari gambar yaitu sebagai berikut :

#### 1. *Service Center Address (SCA)*

SCA adalah alamat (nomor) dari SMSC. SCA memiliki tiga komponen utama yaitu, *type of number* dan *service center number*.

**Tabel 2.4 : Service Center Address dari PDU Penerima**

Octet	Keterangan	Nilai
<i>Len</i>	Panjang informasi SMSC dalam octet	06

<i>Type of number</i>	Format nomor dari SMSC  81 heksa = format lokal  91 heksa = format <i>international</i>	91
<i>Service center number</i>	Nomor SMSC dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F heksa.	2618010000

Pada contoh di atas nilai SCA adalah 0691261801000

## 2. PDU Type

Nilai *default* dari PDU Type untuk *SMS Deliver* adalah 04 heksa, yang memiliki arti 04 heksa = 00000100

**Tabel 2.5: PDU Type**

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	<none	<none	VPF	RD	MTI	MTI

## 3. Originator Address (OA)

OA adalah alamat (nomor) dari pengirim, yang terdiri dari atas panjangnya nomor pengirim (Len), format dari nomor pengirim (*Type Number*), dan nomor pengirim (*Originator Number*). Nilai dari OA pada contoh diatas adalah 0C91261822883847.

**Tabel 2.6: Originator Address**

<b>Octet</b>	<b>Keterangan</b>	<b>Nilai</b>
Len	Panjang nomor pengirim	0C
Type of number	Format dari nomor pengirim 81 heksa = format lokal 91 heksa = format international	91
Originator number	Nomor pengirim dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F heksa.	2618

Dibawah ini contoh format penulisan OA:

a. Nomor OA : ABCDEFGHIJKLMNOPQRST

Format dalam PDU : 14 81 BA DC FE HG JI LK NM PO RQ TS

b. Nomor OA : +ABCDEFGHIJKL

Format dalam PDU : 0C 91 BA DC FE HG JI LK

#### 4. **Coding Scheme Protocol Identifier (PID)**

Protocol Identifier adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari handphone pengirim. Misalnya tipe *Standard Text*, *Fax*, *E-mail*, *Telex*, *X400*, dan lain-lainnya. Nilai default dari PID adalah 00 = "*Standard Text*". Untuk contoh diatas nilai dari PID adalah 00, sehingga pesan yang diterima berupa text standard

#### 5. **Data Coding Scheme**

Data adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standar, Flash SMS atau bahkan Blinking SMS seperti terlihat pada tabel berikut ini.

**Tabel 2.7 : Penentuan Message Class**

<b>nNNilai Heksa</b>	<b>Character Coding</b>	<b>Message Class</b>
00	default ( 7 bit )	No class
F0	default ( 7 bit )	class 0 (immediate display)
F1	default ( 7 bit )	class 1 (Mobile Equipment-specific)
F2	default ( 7 bit )	class 2 (SIM specific message)
F3	default ( 7 bit )	class 3 (Terminate Equipment-specific)
F4	8-bit	class 0 (immediate

		display)
F5	8-bit	class 1 (Mobile Equipment-specific)
F6	8-bit	class 2 (SIM specific message)
F7	8-bit	class 3 (Terminate Equipment-specific)

Hal yang perlu diperhatikan di sini, pada beberapa *handphone* dengan message class 0 dengan *encoding* 7 bit berupa flash SMS, sedangkan dengan *encoding* 16 bit *Unicode* (ucs2), message yang didahului "0001" dengan class 0 berupa blinking flash SMS. Pada contoh di atas DCS adalah 00 yang berarti bahwa pesan yang diterima merupakan pesan *text standard*.

#### 6. *Service Center Time Stamp* (SCTS)

*Service Center Time Stamp* adalah waktu dari penerimaan pesan oleh SMSC penerima. SCTS terdiri dari tahun, bulan, tanggal, jam, menit dan detik, serta zona waktu. Nilai SCTS pada contoh diatas adalah 40106061220282.

**Tabel 2.8 : Service Center Time Stamp**

<b>Nama</b>	<b>Nilai</b>	<b>Hasil</b>
Tahun	40	04 (2004)
Bulan	10	05 (Mei)
Tanggal	60	06
Jam	61	16
Menit	22	22
Detik	02	20
Zona waktu	82	28, di mana 1 unit = 15 menit. Jadi $(15 \times 28) / 60 = 7$ jam. Sehingga menjadi GMT + 07.00 = WIB

Dari tabel diatas terlihat bahwa pesan diterima oleh SMSC pada tanggal 16 Januari 2004 pukul 16:22':20" WIB

**7. User Data Length (UDL)**

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

**8. User Data (UD)**

*User Data* adalah pesan yang diterima dalam format heksadesimal. Pada contoh di atas nilainya adalah E8329BFD4697D9EC37. pengkodean dari nilai heksadesimal menjadi teks standar dengan bantuan tabel kode ASCII dapat dilihat pada tabel berikut :

**Table 2.9 : User Data Pada PDU Penerima**

Nilai	Oktet (8 bit)	Septet (7 bit)	Decimal	Hasil
E8	1 1101000	1101000	104	h
32	00 110010	110010 1	101	e
9B	100 11011	11011 00	108	l
FD	1111 1101	1101 100	108	l
46	01000 110	110 1111	111	o
97	100101 11	11 01000	104	h
D9	1101100 1	1 100101	101	e
EC	1 1101100	1101100	108	l
37	00 110111	1101100	108	l
		110111 1	111	o

Dari tabel di atas terlihat bahwa nilai heksadesimal dari E8329BFD4697D9EC37 adalah "hellohello". Ini berarti pesan yang diterima adalah "hellohello".

**Table : 2.10 Kode ASCII**

Dec	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		

### 9. SMS Submit PDU (*Mobile Originated*)

SMS Submit PDU ialah pesan yang dikirim dari *handphone* ke terminal yang kemudian dikirimkan ke SMSC (Wesolowski 2002). Pada prinsipnya apabila kita mengirim pesan ke nomor tujuan, pesan itu akan melalui SMSC.

Pesan yang akan dikirimkan oleh terminal masih dalam bentuk teks, sedangkan dalam pengiriman ke SMSC harus dalam bentuk PDU. Untuk itu sebelum dikirim, terminal atau *handphone* akan melakukan perubahan dari format teks menjadi format PDU, proses ini disebut proses *encodec*. Adapun skema dari format SMS Submit PDU telah diatur dan ditetapkan oleh ETSI sebagai berikut :

**Tabel 11 : Skema Format SMS Submit PDU**

SCA	PDU	MR	DA	PID	DCS	VP	UDL	UD
	Type							

Misalnya kita mengirim pesan SMS ke nomor 628569976796 dengan isi pesan "pesan pendek" dengan batas waktu pengiriman (waktu penyimpanan pesan di SMSC, jika nomor tujuan tidak dapat menerima pesan) 5 hari. Maka format PDU adalah: 0011000C912618229888040000AB0CD0F23CEC06C1CB6E72790D

### 10. *Service Center Address (SCA)*

SCA adalah alamat (nomor) dari SMSC. SCA memiliki tiga komponen utama yaitu, *type of number*, dan *service center number*. Pada contoh di atas nilai dari SCA adalah 00.

**Tabel 2.11: Service Center Address**

Octet	Keterangan	Nilai
Len	Panjang informasi SMSC dalam octet	00
Type of number	Format nomor dari SMSC 81 heksa = format lokal 91 heksa = format international	<none>
Service center number	Nomor SMSC dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F heksa.	<none>

### 11. *PDU Type*

Nilai default dari PDU Type untuk SMS Submit adalah 11 heksa. Pada Contoh diatas, PDU Type adalah 11 yang memiliki arti:

**Tabel 2.12: PDU Type**

Bit no	7	6	5	4	3	2	1	0
--------	---	---	---	---	---	---	---	---

Nama	RP	UDH	SRI	VPF	VPF	RD	MTI	MTI
Nilai	0	0	0	1	0	0	0	0

0 0 → Jika pesan tidak sampai di SMSC.

1 0 → Format relatif (satu oktet).

0 1 → Format *enhanced* (tujuh oktet).

1 1 → Format absolut (tujuh oktet).

a. RD : *Reject Duplicates*. Parameter yang menandakan ya atau tidaknya Service Center akan menerima suatu pengiriman pesan SMS untuk suatu pesan yang masih disimpan dalam Service Center tersebut. Ia mempunyai MR dan DA yang sama sebagai pesan dikirimkan dari OA yang sama.

b. MTI : *Message Type Indicator*. Bit ini bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu SMS Deliver.

## 12. *Message Reference (MR)*

*Message Reference* adalah acuan dari pengaturan SMS. Untuk membiarkan pengaturan pesan SMS dilakukan sendiri oleh *handphone* tujuan, maka nilai yang diberikan adalah "00". Jadi pada *message Reference* hasilnya adalah 00.

## 13. *Destination Address (DA)*

DA adalah alamat (nomor) tujuan, yang terdiri atas panjang nomor tujuan (*Len*), format dari nomor tujuan (*Type Number*) dan nomor tujuan (*DestinationNumber*).

**Tabel 2.13: Destination Address dari PDU Pengirim**

Oktet	Nilai	Hasil
Len	12	OC
Type of number	Format Internasi onal	91
Destination Number	628122898840	261822988804

**14. Protocol Identifier (PID)**

Untuk contoh diatas nilai dari PID adalah 00.

**15. Data Coding Scheme**

Pada contoh di atas DCS adalah 00.

**16. Validity Period (VP)**

*Validity Period* adalah lama waktu pesan SMS disimpan di SMSC apabila pesan tersebut gagal diterima oleh *handphone* penerima.

Tabel 2.14: Validity Period

Waktu VP	Nilai VP
5 menit – 720 menit (12 jam)	$(\text{waktu VP} / 5) - 1$
12,5 jam – 24 jam	$143 + ((\text{waktu VP} - 12) * 2)$
2 – 30 hari	$166 + \text{waktu VP}$
Lebih dari 4 minggu	$192 + \text{waktu VP}$

Pada contoh di atas, waktu VP-nya 5 hari, maka nilai VP adalah  $166 + 5 = 171$  d = AB h. Jadi pada Validity Period hasilnya adalah AB.

**17. User Data Length (UDL)**

Pada contoh di atas nilai dari UDL adalah 0C, yang berarti pesan yang Dikirim adalah sebanyak 12 karakter.

**18. User Data (UD)**

User Data adalah pesan yang akan dikirim dalam format heksadecimal. Pada contoh ini pesan SMS yang dikirim adalah "Pesan pendek". Pengkodean dari nilai teks standar menjadi heksadesimal dilakukan dengan bantuan *Default Alphabet* yang dibekukan oleh ETSI GSM 03.38 (Tabel ASCII) (Park, 2003). Hal ini dapat dilihat pada tabel 15 berikut ini

**Tabel 2.15: User Data pada PDU Pengirim**

Nilai	Decimal	Septet (7 bit)	Oktet (8 bit)	Hasil
P	80	1010000	11010000	D0
E	101	1100101	11110010	F2
S	115	1110011	00111100	3C
A	97	1100001	11101100	EC
N	110	1101110	00000110	06
Spasi	32	0100000	11000001	C1
P	112	1110000	11001011	CB
E	101	1100101		
N	110	1101110	01101110	6E
D	100	1100100	01110010	72

E	101	1100101	01111001	79
K	107	1101011	00001101	0D

Dari tabel di atas terlihat bahwa hasil dari pengkodean adalah : D0F23CEC06C1CB6E72790D. Jadi pada User Data hasilnya adalah D0F23CEC06C1CB6E72790D. Dari penjelasan di atas di peroleh hasil unuk pengiriman SMS dalam format PDU unuk contoh tersebut adalah: 0011000C912618229888040000AB0CD0F23CEC06C1CB6E72790D

### 2.1.7 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. Pada akhirnya SMS menjadi layanan messaging yang populer dan digemari oleh customer telepon seluler. Layanan SMS dapat diintegrasikan dengan layanan GSM yang lain seperti voice, dan fax. Oleh karena itu pesan SMS selain digunakan untuk pengiriman pesan person to person juga digunakan untuk notifikasi voice dan fax mail yang datang kepada customer. Selain itu SMS juga berharga murah, bersifat sederhana dan personal.

### **2.1.8 Definisi SMS Gateway**

Menurut Rhyca Putri Ardy, Riske Atista, Antonius Wahyu Sudrajat, 2013, SMS Gateway Pada prinsipnya, SMS Gateway adalah sebuah perangkat lunak yang menggunakan bantuan komputer dan memanfaatkan teknologi seluler yang diintegrasikan guna mendistribusikan pesan-pesan yang di-generate lewat sistem informasi melalui media SMS yang di-handle oleh jaringan seluler. SMS Gateway ini memanfaatkan handphone untuk server pengiriman SMS. SMS memanfaatkan jaringan operator seluler untuk pengiriman sms dan database, mysql yang diintegrasikan dengan database.

#### **2.1.8.1 Keuntungan SMS Gateway**

Menurut Ariza Novianti, Ami Fauziah, 2009, Keuntungan SMS Gateway SMS Gateway merupakan pintu gerbang bagi penyebaran Informasi dengan menggunakan SMS. SMS Gateway dapat menyebarkan pesan ke ratusan nomor secara otomatis dan cepat yang langsung terhubung dengan database nomor nomor ponsel saja tanpa harus mengetik ratusan nomor dan pesan di ponsel karena semua nomor akan diambil secara otomatis dari database tersebut sehingga dapat menghemat waktu. Selain itu,

#### **2.1.8.2 Kebutuhan Sms Gateway**

Menurut Ariza Novianti, Ami Fauziah, 2009, Kebutuhan untuk SMS Gateway juga tidak terlalu berlebihan karena hanya menggunakan sebuah PC dengan menggunakan sebuah ponsel, kabel data, kartu GSM, dan sebuah program yang dapat dibangun sendiri khususnya menggunakan bahasa pemrograman JAVA yang berfungsi sebagai pengirim pesan. SMS Gateway dapat mengustomisasi pesan-pesan yang ingin dikirim. Dengan menggunakan program tambahan yang dapat dibuat sendiri, pengirim pesan dapat lebih fleksibel dalam mengirim berita karena biasanya pesan yang ingin dikirim berbeda-beda untuk masing-masing penerimanya.

## **2.1.9 Perangkat Lunak yang digunakan**

Bahasa Java tentunya sudah tidak asing lagi bagi kita, selain terkenal dengan motto-nya yaitu "*write once run everywhere*", penggunaan java juga telah semakin meluas dari aplikasi berbasis web sampai dengan peralatan-peralatan yang bersifat mobile, seperti handphone. Pada kesempatan ini kita juga menggunakan bahasa java untuk membuat aplikasi SMS server.

### **2.1.9.1 Pengenalan Java**

Menurut definisi SunMicrosystem, di dalam buku M. Shalahuddin dan Rosa A.S. (2010 : 1) Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (standalone) ataupun pada lingkungan jaringan. Java berdiri di atas sebuah mesin penterjemah (interpreter) yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca kode bit (bytecode) dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (platformindependent). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah "*write once, run everywhere*", yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (platform) manapun, tanpa harus melakukan perubahan kode program.



### 2.1.9.2 Karakteristik dan Kelebihan Java

Java memiliki beberapa karakteristik yang menjadikan java memiliki kelebihan dibandingkan bahasa pemrograman lain. Adapun karakteristik-karakteristik tersebut adalah :

#### 1. Sederhana

Java adalah suatu bahasa sederhana. Para perancang Java berusaha menciptakan suatu bahasa yang mudah dan cepat dipelajari, sehingga bahasa java dibuat dengan ukuran relatif kecil.

#### 2. Berorientasi objek

Bahasa pemrograman saat ini mengarah ke bahasa pemrograman berorientasi objek. Rancangan berorientasi objek merupakan suatu teknik yang memusatkan rancangan pada data (objek) dan antar muka. Fitur pemrograman berorientasi objek pada java pada dasarnya sama dengan C++. Perbedaan terletak pada kemampuan penurunan berganda (*multiple inheritance*) pada java lebih baik dibandingkan C++.

#### 3. Dapat didistribusi dengan mudah

Java memiliki *library* rutin yang lengkap untuk dirangkai pada protokol TCP/IP, seperti HTTP dan FTP dengan mudah. Kemampuan *networking* java lebih kuat dan mudah dipakai.

#### 4. Kuat

Suatu program yang dibuat dengan java dapat dipercaya dalam berbagai hal. Karena java banyak menekankan pada pengecekan awal untuk menghindari kemungkinan terjadi masalah, pengecekan pada saat *runtime* dan mengurangi kemungkinan timbulnya kesalahan (*error*).

#### 5. *Multiplatform*

Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan yang berbeda. Dengan slogan "*Write Once, Run Anywhere (WORA)*", java dapat dijalankan pada berbagai platform seperti Linux, Unix, *Windows*, *Solaris*, maupun *Mac*.

#### 6. *MultiThread*

*MultiThreading* adalah kemampuan suatu program komputer melakukan beberapa pekerjaan sekaligus, misalnya mencetak file sambil *browsing* internet. *MultiThreading* dalam java sangat mudah dikelola. *Thread* dalam java juga memiliki kemampuan untuk memanfaatkan kelebihan *multi-processor* jika sistem operasi yang digunakan mendukung multiprosesor

#### 7. *Interpeter*

*Interpreter* Java dapat mengeksekusi kode *byte* Java secara langsung pada setiap mesin yang terdapat interpreter dan setiap *runtime* java. Pada sistem Java, tahap *link* program adalah sederhana, bertahap dan ringan. Hal ini membuat siklus pengembangan menjadi sangat cepat.

#### 8. Kinerja yang tinggi

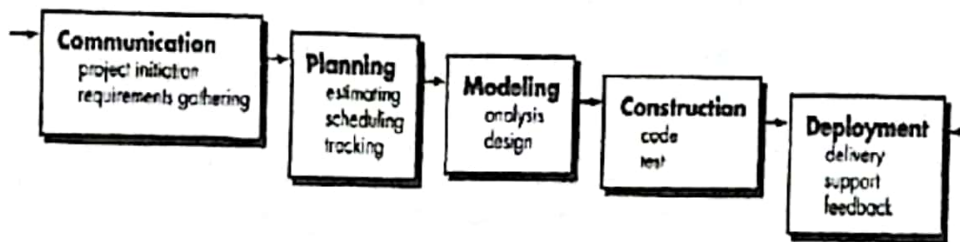
Java dapat mencapai performansi yang tinggi dengan cara mengadopsi sebuah skema yang memungkinkan *interpeter* dapat berjalan pada kecepatan penuh tanpa perlu memeriksa lingkungan *runtime*.

## 2.2 Definisi Metode Pengembangan Perangkat Lunak

### 2.2.1 Metode Waterfall

Menurut Pressman (2010, p39) model waterfall adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut ini ada dua gambaran dari waterfall model.

Fase-fase dalam model waterfall menurut referensi Pressman:



Gambar 2.1: waterfall Pressman

#### 1. Communication

Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan customer, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

#### 2. Planning

Proses planning merupakan lanjutan dari proses communication (analysis requirement). Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan software, termasuk rencana yang akan dilakukan.

#### 3. Modeling

Proses modeling ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement.

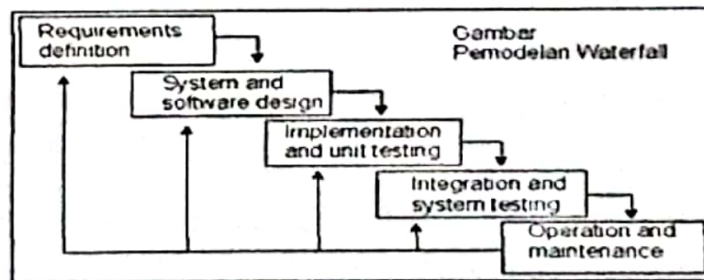
#### 4. Construction

Construction merupakan proses membuat kode. Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu software, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

#### 5. Deployment

Tahapan ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user. Kemudian software yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Sedangkan fase-fase model waterfall menurut referensi Sommerfile:



##### 1. Requirements Analysis and Definition

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh software yang akan dibangun. Hal ini sangat penting, mengingat software harus dapat berinteraksi dengan elemen-elemen yang lain seperti hardware, database, dsb. Tahap ini sering disebut dengan Project Definition.

##### 2. System and Software Design

Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software

engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dsb. Dari dua aktivitas tersebut (pencarian kebutuhan sistem dan software) harus didokumentasikan dan ditunjukkan kepada user. Proses software design untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk “blueprint” software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti dua aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari software.

### **3. Implementation and Unit Testing**

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

### **4. Integration and System Testing**

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer. Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).

### **5. Operation and Maintenance**

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsi-fungsi software harus diujicobakan, agar software bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya. Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

### **2.2.2 Kelebihan dari Model Waterfall**

Kelebihan dari model ini adalah selain karena pengaplikasian menggunakan model ini mudah, kelebihan dari model ini adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh, eksplisit, dan benar di awal proyek, maka Software Engineering (SE) dapat berjalan dengan baik dan tanpa masalah. Meskipun seringkali kebutuhan sistem tidak dapat didefinisikan se-eksplisit yang diinginkan, tetapi paling tidak, problem pada kebutuhan sistem di awal proyek lebih ekonomis dalam hal uang (lebih murah), usaha, dan waktu yang terbuang lebih sedikit jika dibandingkan problem yang muncul pada tahap-tahap selanjutnya.

### **2.2.3 Kekurangan dari Model Waterfall**

Kekurangan yang utama dari model ini adalah kesulitan dalam mengakomodasi perubahan setelah proses dijalani. Fase sebelumnya harus lengkap dan selesai sebelum mengerjakan fase berikutnya. Masalah dengan waterfall:

1. Perubahan sulit dilakukan karena sifatnya yang kaku.
2. Karena sifat kakunya, model ini cocok ketika kebutuhan dikumpulkan secara lengkap sehingga perubahan bisa ditekan sekecil mungkin. Tapi pada kenyataannya jarang sekali konsumen/pengguna yang bisa memberikan kebutuhan secara lengkap, perubahan kebutuhan adalah sesuatu yang wajar terjadi.
3. Waterfall pada umumnya digunakan untuk rekayasa sistem yang besar yaitu dengan proyek yang dikerjakan di beberapa tempat berbeda, dan dibagi menjadi beberapa bagian sub-proyek.

## **2.3 Definisi Tolls Pengembangan Perangkat Lunak**

### **2.3.1 Pengenalan Unified Modeling Language (UML)**

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan. UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya. UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2014:133).

### **2.3.2 Sejarah UML**

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep Object Modeling Technique(OMT) dari Rumbaugh dan Booch (1991), konsep The Classes, Responsibilities, Collaborators (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan Unified

Modeling Language (UML). Pada tahun 1996, Object Management Group (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Rosa A.S dan M. Shalahudin, 2014:138).

### **2.3.3. Diagram UML**

Rosa A.S dan M. Shalahudin (2014:140), pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. Structure diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Structure diagram terdiri dari class diagram, object diagram, component diagram, composite structure diagram, package diagram dan deployment diagram.
2. Behavior diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari Use case diagram, Activity diagram, State Machine System.
3. Interaction diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Interaction diagram terdiri dari Sequence Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram.

### **2.3.4 Use Case Diagram**

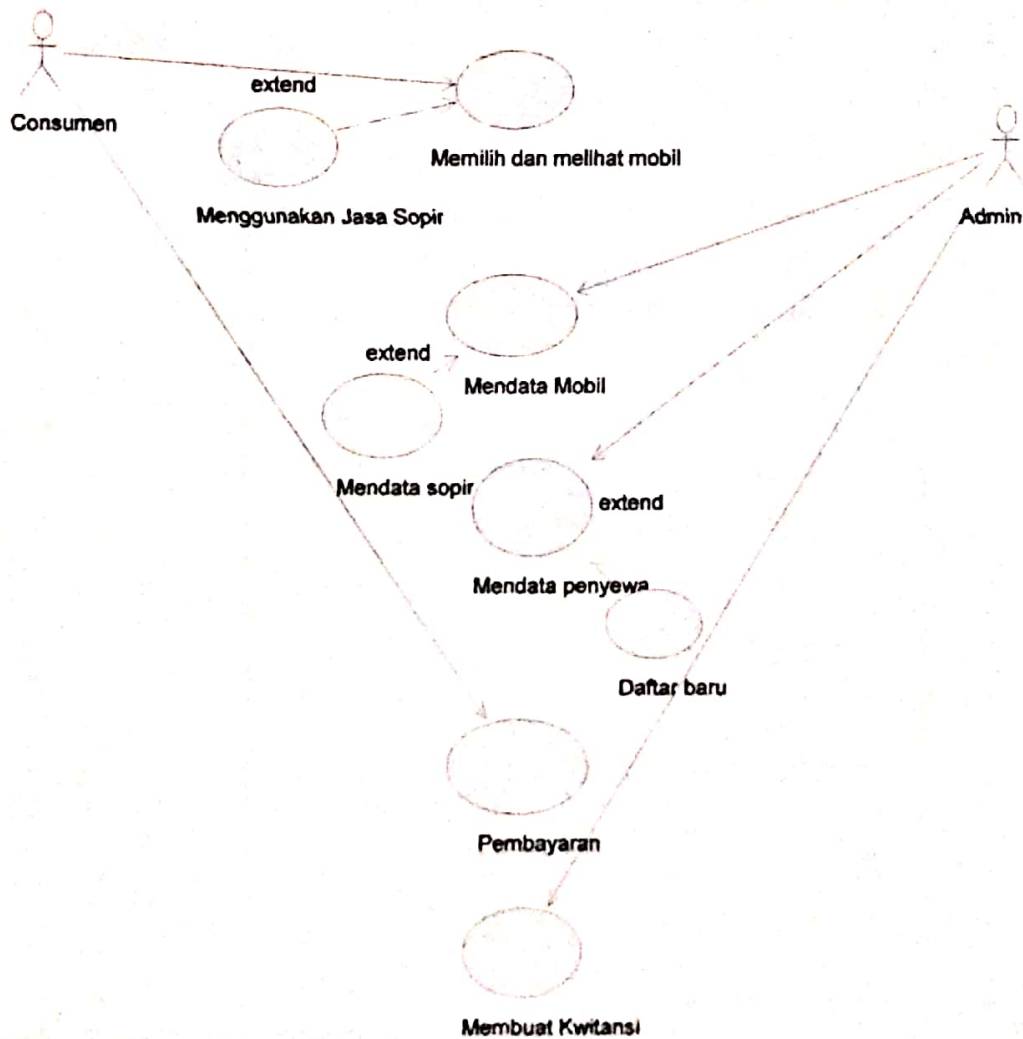
Rosa dan M. Shalahudin (2014:155), use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan



untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

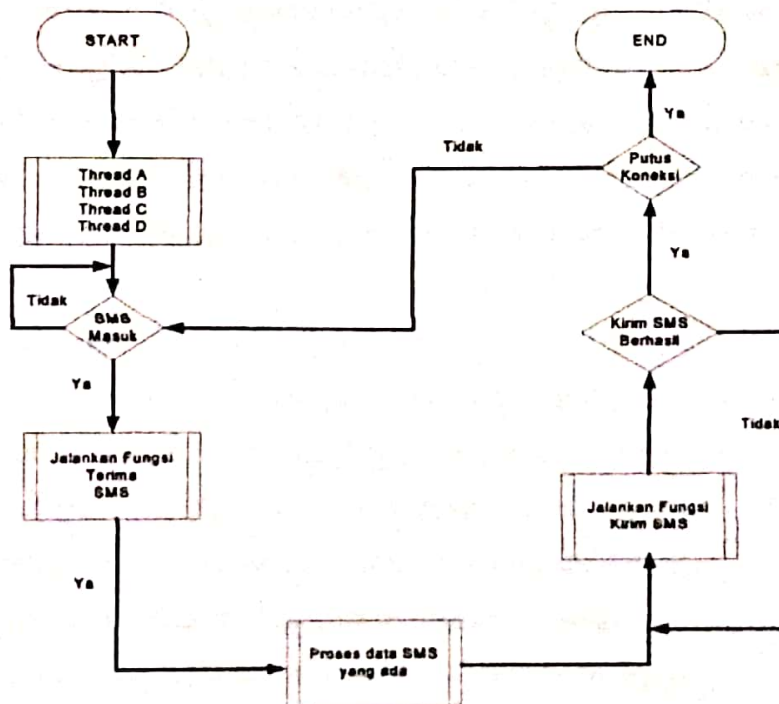
Berikut adalah simbol-simbol yang ada pada diagram use case:

**Table 2.16: simbol” diagram use case**



### 2.3.5 Flowchart

*Flowchart* adalah sekumpulan gambar-gambar tertentu untuk menyatakan alur dari suatu program yang akan diterjemahkan ke salah satu bahasa pemrograman. Kegunaan flowchart sama seperti halnya algoritma yaitu untuk menuliskan alur program tetapi dalam bentuk gambar atau symbol.



### 2.3.6 Perancangan sistem

Perancangan adalah proses penerapan berbagai teknik dan prinsip dengan tujuan untuk mentransformasikan hasil analisis ke dalam bentuk yang memudahkan pengimplementasian. Dalam perancangan sistem berorientasi objek, diagram-diagram UML yang digunakan adalah mengikuti standar UML dan untuk perancangan basis datanya dapat menggunakan basis data relational dengan pendekatan model data ERD. Diagram-diagram tersebut dijelaskan dibawah ini.

### 2.3.8.2 ERD (Entity Relationship Diagram)

Menurut Sutanta (2011:91) "Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek." Entity Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. Entity Relationship Diagram (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan Entity Relationship Diagram (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, Entity Relationship Diagram (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data didalamnya.

#### 1. Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut :

- a. Entitas dinyatakan dengan simbol persegi panjang.
- b. Nama entitas dituliskan didalam simbol persegi panjang.
- c. Nama entitas berupa kata benda, tunggal.
- d. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

#### 2. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

- a. Atribut digambarkan dengan simbol ellips.
- b. Nama atribut dituliskan didalam simbol ellips.
- c. Nama atribut merupakan kata benda, tunggal.

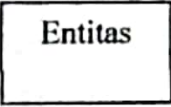



d. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

### 3. Relasi

Relasi Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut :

- a. Relasi dinyatakan dengan simbol belah ketupat.
- b. Nama relasi dituliskan didalam simbol belah ketupat.
- c. Nama relasi berupa kata kerja aktif.
- d. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

**Table 2.18: Komponen ERD**

Notasi	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai
	Relasi, menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda
	Atribut. Berfungsi mendeskripsikan karakter entitas (atribut yang berfungsi sebagai key diberi garis bawah)
	Garis, sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

#### 2.3.8.2 LRS (*Logical Record Structure*)

Menurut Sutanta (2011:91) LRS (*Logical Record Structure*) Adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil

antar himpunan entitas. Menentukan kardinalitas, jumlah table dan Foreign Key (FK).

#### **2.3.8.3 Spesifikasi Basis data**

Spesifikasi Basis Data menggambarkan struktur data fisik pada suatu sistem atau aplikasi. Spesifikasi basis data menyajikan bagaimana penyimpanan data dilakukan di *software* basis data.

#### **2.3.8.4 Rancangan Masukan**

Menurut Ferry Kristanto Hartono, 2011, Rancangan dokumen masukan merupakan dasar berupa formulir yang digunakan untuk memperoleh data yang telah terjadi. Data yang sudah dicatat dan tertera pada sistem informasi diolah lebih lanjut. Adapun rancangan masukan berisi :

1. Nama dokumen masukan
2. Sumber
3. Fungsi
4. Media
5. Rangkap
6. Frekuensi
7. Volume
8. Format dokumen ( lampiran)
9. Keterangan

#### **2.3.8.5 Rancangan Keluaran**

Menurut Ferry Kristanto Hartono, 2010, Rancangan dokumen keluaran merupakan hasil dari proses data-data informasi. Data-data yang diperoleh pada saat terjadi transaksi-transaksi yang dilakukan oleh organisasi merupakan bahan mentah untuk menghasilkan informasi. Adapun rancangan keluaran berisi :

1. Nama dokumen keluaran
2. Fungsi
3. Media

4. Distribusi
5. Rangkap
6. Frekuensi
7. Volume
8. Format dokumen ( lampiran)
9. Keterangan

#### 2.4 Penelitian terdahulu

Dalam penilitan ini akan digunakan delapan tinjauan yang nantinya Mendukung penelitian yang akan dilakukan, dimana tinjauan studi yang di ambil adalah bisa dilihat pada tabel dibawah ini: .

**Table 2.19: Penelitian Terdahulu**

No.	JUDUL	TAHUN	KESIMPULAN, PENELITIAN ,METODE HASIL	PENULIS
1	Aplikasi Short Message Service (SMS) Gateway Pembelian Tiket Pertandingan Klub Sepak bola Arema	2016	Kesimpulan program yang telah dibuat ini hanya untuk pembelian tiket pertandingan klub sepakbola arema indonesia, admin tidak perlu membalas sms satu persatu karena semua sms balasan dilakukan oleh server program. Penelitian ini bahwa proses pembelian tiket dengan sms gateway ini terkomputerisasikan dengan baik	Linda Suvi Rahmawati, Abi Yahya Al Anshori

2	Pengembangan Sistem Informasi SMS Gateway Dalam Meningkatkan Layanan Komunikasi Sekitar Akademika Fakultas Ilmu Komputer Unsri	2015	Kesimpulan program sms gateway dapat mempermudah pengiriman pesan disekitar akademi fasikom unsri. Penelitian ini terdiri dari Analisis permasalahan, mekanisme dan RPL, keluaran dan indikator keberhasilan. Hasil dari penelitian ini berupa perangkat lunak sms gateway untuk mendukung proses informasi yang ada difakultas ilmu komputer unsri.	Mira Afrina1, Ali Ibrahim2
3	Pembuatan Simulasi Aplikasi SMS Gateway Pemesanan Pakaian Muslimah Online Berbasis Web	2013	Metodolgi penelitian studi literatur, pengumpulan data yang berhubungan dengan penelitian, analisa sistem, implementasi sistem, pengujian sistem, dokumen sistem. Manfaat penelitian Untuk melakukan efisiensi transaksi pemesanan barang dan pelayanan, Untuk memberikan suatu kontribusi kepada masyarakat dalam hal ini transaksi dengan kecepatan dan ketelitian dan kelengkapan.	Kiki Aprillia
4	Pengembangan SHORT MESSAGE SERVICE (SMS) GATEWAY Layanan Informasi Akademik DI SMK YPPT	2012	Kesimpulan dari hasil penelitian pengembangan sms gateway dilakukan dengan menambahkan fasilitas pengiriman pesan pengumuman terhadap siswa berdasarkan pada	Isnawati Mulyani 1, Eri Satria2, Asep Deddy Supriatna3

	GARUT		kelas,jurusan dan subkelas.	
5	Implementasi SMS Gateway Pada Kantor PERUSTAKAAN DAN ARSIP KABUPATEN BATANG	2010	Metode penelitian dilakukan dengan pengumpulan data, observasi,wawancara dan studi pustaka. Kesimpulan dengan adanya sms gateway maka penyampaian informasi kepada anggota menjadi mudah.	Imas Panerang
6	Aplikasi Pelayanan Dan Pencarian TAXI Terdekat Dengan CELL ID Dan Pengiriman Pesanan Berbasis SMS Gateway	2012	Kesimpulan untuk simulasi pada proses mitigasi dan evakuasi yaitu posisi didapatkan mempunyai akurasi yang rendah. Hasil analisa cara mendapatkan informasi cell ID dari mobile.cara mengirimkan informasi cell ID ke server, pengiriman sms kepada 5 taxi terdekat.	Yuliana Setiowati1, Afrida Helen1, Istirokha2,
7	Sistem Informasi Pemesanan Tiket Bus Online Pada PT LORENA KARINA Transport Berbasis Web Dan SMS Gateway	2012	Kesimpulan PT LORENA EKA SARI membutuhkan sistem yang memberikan pelayanan kepada pelanggan berupa website yang dapat diakses 24 jam nonstop sehingga pelanggan dapat mengakses tanpa batasan jarak dan waktu.	Immanuel Cofrandy Suranta
8	Pembuatan Aplikasi Berbasis SMS Gateway Untuk Pemesanan Tiket Pesawat Menggunakan	2012	Kesimpulan Aplikasi SMS Gateway ini dapat melakukan autoreply untuk memberikan informasi sesuai dengan	Lathief Noor Astuwasito



	NetBeans IDE 6.8 Pada Gardoe Tiket		format pesan yang dikirimkan oleh calon penumpang. Hasil penelitian pengetesan aplikasi pada sisi server dan client	
--	------------------------------------	--	---	--