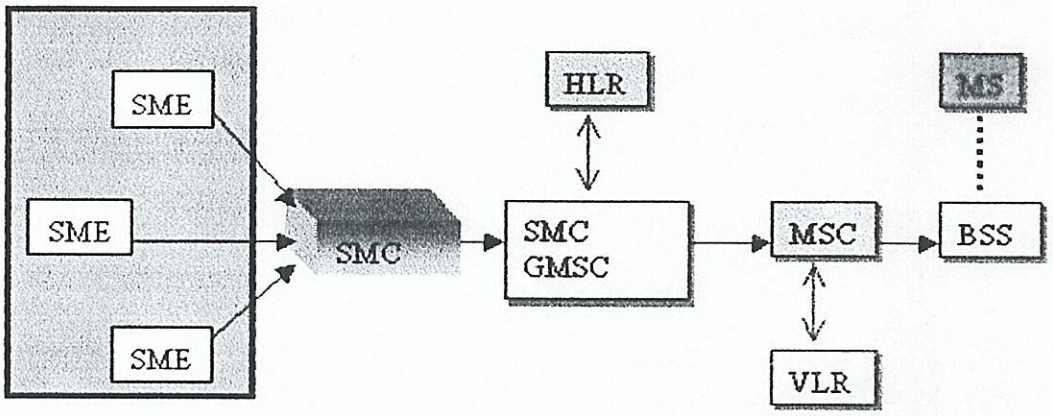
2.1.1 Cara Kerja SMS



Gambar 2. 1 Cara Kerja SMS

SMC berfungsi untuk melakukan penyimpanan dan pengiriman pesan ke dan dari mobile station. SME (Short Message Entity) yang terdapat dalam jaringan tetap atau mobile station, berfungsi untuk menerima dan mengirim SMS.

SMS GMSC (SMS gateway MSC) merupakan pintu gerbang MSC yang juga bisa menerima pesan pendek. Pintu gerbang MSC merupakan titik koneksi jaringan bergerak dengan jaringan lain. Dalam menerima pesan pendek dari SMC, GMSC menggunakan jaringan SS7 untuk menanyai lokasi mobile station dari HLR (Home Location Register).

HLR adalah database utama dari jaringan bergerak yang menyimpan informasi profil pelanggan dan juga tentang informasi rute bagi pelanggan, yaitu area dimana mobile sekarang berada.

MSC (Mobile Switching Center) berfungsi untuk mengerjakan penggantian koneksi antara mobile station atau antara mobile station dengan jaringan tetap.

VLR (Visitor Location Register) berhubungan dengan setiap MSC and memiliki informasi sementara tentang mobile, informasi seperti identifikasi mobile dan cell (atau grup cell) dimana mobile sekarang berada. Menggunakan informasi dari VLR, MSC dapat mengirim informasi (pesan pendek) ke BBS (Base Station System, BSC + BTS) yang cocok untuk dikirimkan pada mobile. BSS terdiri dari transceivers, yang digunakan untuk mengirim dan menerima informasi dari media udara, ke dan dari mobile station. Informasi ini dikirimkan melalui saluran signal jadi mobile dapat menerima pesan walaupun panggilan suara atau data sedang digunakan.

2.2 AT Command

Kontrol operasi dari telepon bergerak GSM dijalankan melalui antarmuka serial (bisa melalui koneksi kabel data atau infra merah), dimana perintah AT+C berdasarkan spesifikasi **ETSI GSM 07.07** dan **GSM 07.05**.

Garis pedoman modem V.25ter mempunyai serangkaian perintah antarmuka. Berdasarkan pedoman tersebut, perintah harus didahului oleh kata "**AT**" dan diakhiri dengan "**<CR>**" (tombol enter). Setiap perintah akan dijawab dengan menampilkan kata **Acknowledgment** yaitu "**OK**" atau "**ERROR**".

Setiap perintah yang sedang diproses akan dihentikan jika ada tambahan karakter yang dijalankan. Ini artinya jangan menjalankan perintah baru sebelum perintah yang lama mendapatkan jawaban, jika tidak perintah sebelumnya akan dihentikan.

2.2.1 Perintah Hayes-Standard

Berikut daftar perintah Hayes-Standard dapat dijalankan pada semua modem yang kompatibel dengan perintah AT Hayes.

Table 2. 1 Perintah Hayes-Standard

Perintah	Fungsi
A/	Ulangi perintah terakhir
AT...	Awalan untuk semua perintah yang lain
ATA	Menerima panggilan
ATD<str>;	Panggil nomor yang terdapat pada kata <str> menggunakan fungsi suara. Karakter ";" adalah penting, digunakan untuk memberitahu telepon bahwa panggilan menggunakan fungsi suara bukan data.
ATDL	Panggil nomor telepon terakhir.
ATE0	Non aktifkan gema perintah
ATE1	Aktifkan gema perintah
ATH[0]	Koneksi terpisah
ATQ0	Tampilkan Acknowledgement
ATQ1	Jangan Tampilkan Acknowledgement
ATV0	Tampilkan Acknowledgment sebagai angka
ATV1	Tampilkan Acknowledgment sebagai text
AT&F[0]	Kembalikan ke seting pabrik
ATZ	Kembalikan ke seting awal

AT+GCAP	Daftar kemampuan

2.2.2 Acknowledgment Untuk Komunikasi Data Normal

Table 2. 2 Acknowledgment Komunikasi Data Normal

Jawaban	Numerik	Arti
OK	0	Perintah berhasil dijalankan, tidak ada kesalahan
RING	2	Ring terdeteksi
NO CARRIER	3	Hubungan tidak terjadi atau terputus
ERROR	4	Salah perintah atau perintah terlalu panjang
NO DIALTONE	6	Tidak ada nada hubung, hubungan tidak memungkinkan
BUSY	7	Nomor yang dihubungi sedang sibuk

2.2.3 SMS AT Command

Di bawah ini akan dijelaskan spesifikasi perintah SMS AT Command berdasarkan referensi dari Brodersen (*GSM Modem UCM-9x AT Command, 2000*).

2.2.3.1 Definisi Parameter

Table 2. 3 Definisi Parameter SMS AT Command

Parameter	Arti
<da>	Alamat tujuan.

<dcs>	Skema pengkodean data.
<dt>	Waktu pelaksanaan Format: yy/MM/dd, hh:mm:ss:zz (tahun/bulan/tanggal, jam:menit:detik dan zona waktu).
<fo>	Oktet pertama, kode seperti oktet pertama pada SMS-SUBMIT, nilai default adalah 17 untuk SMS-SUBMIT.
<index>	Tempat/alamat pada memori
<length>	Pada mode text (+CMGF=0) menandakan jumlah karakter. Pada mode PDU (+CMGF=1) menandakan panjang unit data TP dalam oktet.
<mem1>	Memori yang digunakan untuk melihat daftar, membaca, dan menghapus pesan (+CMGL, +CMGR, +CMGD)
<mem2>	Memori yang digunakan untuk menulis dan mengirimkan pesan (+CMGW, +CMSS).
<mid>	Identifikasi pesan CBM
<mr>	Referensi pesan
<oa>	Alamat pengirim
<pid>	Identifikasi protokol
<pdu>	Untuk SMS berisikan alamat service center diikuti TPDU dalam format heksadesimal. Untuk CBS berisikan TPDU dalam format heksadesimal.
<ra>	Alamat penerima
<sca>	Alamat service center
<scts>	Waktu service center dalam format :

	'yy/MM/dd, hh:mm:ss+zz' (tahun/bulan/tanggal, jam:menit:detik+zonawaktu)
<sn>	Nomor serial CBM
<st>	Status dari SMS-STATUS-REPORT
<stat>	Status pesan pada memori
<tooa>	Tipe alamat dari <oa>
<tosca>	Tipe alamat dari <sca>
<total1>	Jumlah pesan pada memori <mem1>
<total2>	Jumlah pesan pada memori <mem2>
<used1>	Jumlah total lokasi pesan yang terdapat pada <mem1>
<used2>	Jumlah total lokasi pesan yang terdapat pada <mem2>
<vp>	Periode validitas pesan pendek, nilai default adalah 167

2.2.3.2 Daftar SMS AT Command

Table 2. 4 SMS AT Command Servis Pesan +CSMS

<table>
<tr><td>+CSMS</td><td>Servis yang dilayani adalah original GSM (SMS-MO) dan pesan pendek (SMS-MT), servis Cell Broadcast Message (SMS-CB)</td></tr>
<tr><td>Sintaks</td><td>AT+CSMS=<service>
<table><tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr><tr><td>AT+CSMS=0</td><td>+CSMS:1,1,1</td></tr><tr><td>AT+CSMS=1</td><td>+CMS ERROR:301</td></tr></table></td></tr>
</table>

	AT+CSMS=1		+CMS ERROR:301
	AT+CSMS? Ctt: Nilai sekarang?		+CSMS:0,1,1,1 OK
	AT+CSMS=? Ctt: Sevis yang ada		+CSMS:{0} OK
Definisi Nilai	<service>	0:	SMS AT Command kompatible dengan GSM 07.05 fase 2 versi 4.7.0
		1:	SMS AT Command kompatible dengan GSM 07.05 fase 2 versi +.

Table 2. 5 SMS AT Command Tempat Penyimpanan Pesan +CPMS

+CPMS	Perintah ini membolehkan untuk menentukan tempat penyimpanan pesan yang digunakan untuk membaca dan menulis.
Sintaks	AT+CPMS=<mem1>, [<mem2>]

Perintah	**Kemungkinan Jawaban**
AT+CPMS=?	+CPMS: (("SM","BM"),"SM") OK
AT+CPMS? Cat: Baca Tempat	+CPMS: "SM",3,10,"SM",3,10 OK
AT+CPMS="AN" Cat: Salah pilih	+CMS ERROR: 302 OK

	tempat penyimpanan		
	AT+CPMS="BM" Ctt: Pilih tempat penyimpanan pesan CBM		+CPMS: 2,20,3,10 OK Cat: Baca, daftar, hapus CBM dari RAM, 2 CBM tersimpan di RAM
	AT+CPMS? Cat: Baca tempat penyimpanan		+CPMS:"BM",2,20,"SM", 3, 10 OK Cat: Baca, daftar, hapus CBM dari RAM, tulis SMS ke SIM
Definisi Nilai	<mem1>	: "SM" "BM"	Memori yang digunakan untuk daftar, baca, dan hapus. Nilai yang bisa diisikan: Tempat penyimpanan SMS pada SIM Tempat penyimpanan CBM pada RAM
	<mem2>	: "SM"	Memori yang digunakan untuk menulis dan mengirim pesan Tempat penyimpanan SMS pada SIM
	Jika perintah yang dijalankan benar, maka akan menghasilkan pesan indikasi: +CPMS: <used1>, <total1>, <used2>, <total2> Jika <mem1> dipilih, maka semua perintah +CMGL, +CMGR, dan +CMGD akan berhubungan dengan jenis pesan yang disimpan pada memori ini.		

Table 2. 6 SMS AT Command Format Pesan +CMGF

+CMGF	Jenis format yang tersedia adalah format text dan format

<table>
<tr><td></td><td>
PDU. Dalam format PDU, pesan SMS yang lengkap, termasuk semua informasi header diberikan dalam bentuk string binari (dalam format heksadesimal, jadi hanya terdiri dari karakter {‘0’,’1’,’2’,’3’,’4’,’5’,’6’, ’7’,’8’,’9’,’A’,’B’,’C’,’D’,’E’,’F’}), setiap pasang karkater dikonversi menjadi sebuah byte (contoh: ‘41’ dikonversi menjadi karakter ASCII ‘A’, dimana mode ASCII adalah 0x41 atau 66).
Pada mode text, semua perintah dan jawaban diberikan dalam bentuk karakter ASCII.
Pilihan format akan disimpan pada EPROM dengan perintah +CSAS.
</td></tr>
<tr><td>Sintaks</td><td>
AT+CMGF
<table>
<tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr>
<tr><td>AT+CMGF?
Cat: Format pesan yang sedang</td><td>+CMGF:1
OK
Cat: Mode Text</td></tr>
<tr><td>AT+CMGF=?</td><td>+CMGF(0,1)
OK
Cat: Tersedia mode text atau PDU</td></tr>
</table>
Contoh untuk mengirim SMS dalam mode PDU
<table>
<tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr>
<tr><td>AT+CMGF=0
Cat: Format pesan PDU</td><td>OK
Cat: Mode PDU berhasil</td></tr>
<tr><td>AT+CMGS=14<CR></td><td>+CMGS:4</td></tr>
</table>
</td></tr>
</table>

	0001030691214365000004C9E9340B Cat: Kirim pesan lengkap dalam format PDU, pakai SC default	OK Cat: Pesan terkirim, <mr> diterima
Definisi Nilai	Pesan <pdu> tersusun dari alamat SC (00 berarti tidak diberikan alamat SC, gunakan alamat SC default yang dapat dibaca dengan dengan perintah +CSCA) dan pesan TPDU. Panjang oktet dari buffer TPDU adalah 14, yg dikodekan sebagai GSM 03.40. Pada kasus ini TPU adalah: 0x01 0x03 0x06 0x91 0x21 0x43 0x65 0x00 0x04 0xC9 0xE9 0x34 0x0B, dimana berdasarkan GSM 03.40 mempunyai arti: <fo> 0x01 (SMS-SUBMIT, tidak ada periode validitas <mr> (TP-MR) 0x03 (referensi pesan) <da> (TP-DA) 0x06 0x91 0x21 0x43 0x65 (alamat tujuan +123456 <pid> (TP-PID) 0x00 (Identifikasi protokor) <dcs> (TP-DCS) 0x00 Skema pengkodean data (alfabet 7 bit) <length> (TP-UDL) (Panjang data, 4 text karakter) TP-UD 0xC9 0xE9 0x34 0x0B (Data: ISSY) TPDU dalam format heksadesimal harus dikonversi kedalam dua karakter ASCII, contoh oktet dengan heksadesimal 0x2A di tampilkan sebagai dua karakter	

	"2".

Table 2. 7 SMS AT Command Menyimpan Setting +CSAS

+CSAS	Semua seting yang terdapat pada perintah +CSCA dan +CSMP disimpan dalam EPROM jika kartu SIM merupakan kartu tahap 1 atau dalam kartu SIM jika kartu SIM ada pada tahap 2.
Sintaks	AT+CSAS **Perintah** / **Kemungkinan Jawaban** AT+CSAS Cat: Simpan parameter +CSAS dan +CSMP OK Cat: Perameter tersimpan

Table 2. 8 SMS AT Command Mengembalikan Setting +CRES

+CRES	Semua setting yang terdapat pada perintah +CSCA dan +CSMP dikembalikan ke nilai semula yang terdapat pada EPROM jika kartu SIM merupakan kartu tahap 1 atau dari kartu SIM jika kartu SIM ada pada tahap 2.
Sintaks	AT+CRES **Perintah** / **Kemungkinan Jawaban** AT+CRES Cat: Kembalikan nilai awal parameter +CSCA dan +CSMP OK Cat: Nilai telah dikembalikan

Table 2. 9 SMS AT Command Menunjukkan Parameter Mode Text +CSDH

<table>
<tr><td>+CSDH</td><td>Perintah ini menghasilkan informasi yang lebih banyak kode pada mode text. Informasi ini berada pada golongan perintah +CMTI, +CMT, +CDS, +CMGR, +CMGL.</td></tr>
<tr><td>Sintaks</td><td>AT+CSDH
<table><tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr><tr><td>AT+CSDH?
Cat: Nilai Sekarang</td><td>+CSDH:0
OK
Cat: Jangan tampilkan nilai header</td></tr></table></td></tr>
</table>

Table 2. 10 SMS AT Command Indikasi Pesan Baru +CNMI

<table>
<tr><td>+CNMI</td><td>Perintah ini untuk memilih prosedur yang akan digunakan dalam hal cara untuk menerima pesan baru.</td></tr>
<tr><td>Sintaks</td><td>AT+CNMI=<mode>, <mt>, <bm>, <ds>, <bfr>
<table><tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr><tr><td>AT+CNMI=2,1,0,0,0
Cat: <mt>=1</td><td>OK
+CMTI:"SM",1
Cat: Pesan diterima</td></tr><tr><td>AT+CNMI=2,2,0,0,0
Cat: <mt>=2</td><td>OK
+CMT: "123456", "03/25/05, 12:30:00+00", 129, 4, 32, 240, "15379", 129, 5<CR><LF>Pesan yang diterima</td></tr></table></td></tr>
</table>

		Cat: Pesan diterima
	AT+CNMI=2,0,0,0,1 Cat: <ds>=1 AT+CMGS="+628129356812' <CR>Pesan yang akan dikirim <crtl+Z> Cat: Kirim pesan dalam mode text	OK +CMGS: 7 OK Cat: Transmisi berhasil +CDS: 2, 116, "+6281293568 12", 145, "03/25/05, 12:30: 07+04", "03/25/05, 12:30: 15+00" Cat: Pesan telah berhasil dikirim
Definisi Nilai	<mode>: Mengontrol proses terhadap kode hasil tak diminta. **Hanya <mode>=2 yang bisa digunakan.** Nilai lain untuk <mode> (0,1 atau 3) bisa diterima (kode jawaban "OK"), tapi proses terhadap kode hasil tak diminta akan sama dengan <mode>=2. <mode> 0: Buffer kode hasil tak diminta pada TA. Jika 1: Abaikan indikasi dan tolak kode hasil tak diminta	

	Sebaliknya kirimkan secara langsung ke TE. 2: Buffer kode hasil tak diminta pada TA saat jalur TE-TA sedang digunakan dan kirim buffer pada TE setelah jalur bisa digunakan. Sebaliknya kirim langsung pada TE. 3: Kirim langsung kode hasil tak diminta pada TE. Spesifik jalur TA-TE digunakan untuk menyimpan hasil kode dan data saat TA pada mode data online. <mt>: Untuk mengarahkan kode hasil indikasi untuk SMS-DELIVER. Nilai default 0. <mt> 0: Tidak ada indikasi SMS-DELIVER yang dikirimkan 1: SMS-DELIVER dikirimkan melalui kode tak diminta: +CMTI:"SM",<index> 2: SMS-DELIVER (kecuali pesan kelas 2) dikirimkan melalui kode tak diminta : +CMT: [<alpha>,] length, <CR><LF><pdu> (mode PDU) atau +CMT: <oa>, [<alpha>,]<scts> [,<tooa>, <fo>, <pid>, <dcs>, <sca>, <tosca>, <length>] <CR><LF><data> (mode text). 3: SMS-DELIVER kelas 3 dikirimkan langsung menggunakan kode pada <mt>=2. Hasil indikasi pada pesan dengan kelas lainnya menggunakan kode <mt>=1. <bm>: Untuk mengatur aturan untuk menyimpan tipe CBM (Cell Broadcast Message) yang diterima berdasarkan skema pengkodean, pilihan tipe CBM (perintah +CSCB) dan <bm>. Nilai default 0.

	<bm> 0: Tidak ada indikasi CBM yang dikirimkan ke TE. CBM disimpan. 1: CBD disimpan dan indikasi lokasi memori dikirimkan ke aplikasi pelanggan melalui kode hasil tak diminta: +CBMI:"BM",<index>. 2: CBM baru dikirimkan langsung ke TE menggunakan kode hasil tak diminta +CBM: <length><CR><LF><pdu> (mode PDU) atau +CBM: <sn>, <mid>, <dcs>, <page>, <pages> (mode text) <CR><LF> <data> 3: CBM kelas 3 seperti <bm>=2, kelas lain seperti <bm>=1. <ds>: Untuk laporan status SMS <ds> 0: Tidak ada SMS-STATUS-REPORT yang dikirimkan. 1: SMS-STATUS-REPORT dikirimkan menggunakan kode hasil tak diminta: +CDS: <length> <CR> <LF> <pdu> (kode PDU) atau +CDS: <fo>, <mr> [,<ra>] [,<tora>] , <scts>, <dt>, <st> (mode text). <bfr>: Nilai default 0. <bfr> 0: Buffer TA untuk kode hasil tak diminta diterapkan dalam perintah ini dikirimkan ke TE saat <mode> 1-3 dipilih. 1: Buffer TA untuk kode hasil tak diminta diterapkan dalam perintah ini dihapus saat <mode> 1-3 dipilih.

Table 2. 11 SMS AT Command Membaca Pesan +CMGR

<table>
<tr><td>+CMGR</td><td>Perintah ini digunakan untuk membaca pesan yang tersimpan.</td></tr>
<tr><td>Sintaks</td><td>AT+CMGR=<index>
Sintak jawaban yang diberikan:
<table><tr><td>Mode Text:</td><td>+CMGR: <stat>, <oa>, [<alpha>,] <scts> [,<tooa>, <fo>, <pid>, <dcs>, <sca>, <tosca>, <length>] <CR> <LF> <data> (untuk SMS-DELIVER).
+CMGR: <stat>, <da>, [<alpha>,] [,<toda>, <fo>, <pid>, <dcs>, [<vp>], <sca>, <tosca>, <length>] <CR> <LF> <data> (untuk SMS-SUBMIT).</td></tr><tr><td>Mode PUD:</td><td>+CMGR: <stat>, [<alpha>,] <length> <CR> <LF> <pdu></td></tr></table>Setiap pesan yang dibaca dengan status “REC UNREAD” akan diupdate pada memori dengan status “REC READ” karena sudah pernah dibaca
Contoh:.
<table><tr><th><u>Perintah</u></th><th><u>Kemungkinan Jawaban</u></th></tr><tr><td></td><td>AT+CMTI: “SM”,1
Cat: Pesan baru diterima.</td></tr><tr><td>AT+CMGR=1
Cat: Baca pesan</td><td>+CMGR: “REC UNREAD”, “0146290800”, “03/05/25, 18:22:11+00”, <CR><LF></td></tr></table></td></tr>
</table>

		ABCdefGHI OK
	AT+CMGR=1 Cat: Baca pesan lagi	+CMGR: "REC UNREAD", "0146290800", "03/05/25, 18:22:11+00", <CR><LF> ABCdefGHI OK
	AT+CMGR=2 Cat: Baca pesan dengan index salah	+CMS ERROR: 321 Cat: Error=Index salah
	AT+CMGF=0;+CMGR=1 Cat: Dalam mode PDU	+CMGR: 2, <length> <CR> <LF> <pdu> OK Cat: Pesan disimpan tapi belum terkirim, tidak ada field <alpha>

Table 2. 12 SMS AT Command Daftar Pesan +CMGL

+CMGL	Perintah ini membolehkan aplikasi untuk membaca pesan yang tersimpan, dengan mengindikasikan jenis pesan yang akan dibaca.	
Sintaks	AT+CMGL=<stat> Sintak jawaban:	
	Mode Text:	+CMGL: <index>, <stat>[,<alpha>], [<scts>, <tooa/toda>, <length>] <CR><LF> <data> (untuk SMS-DELIVER dan SMS-SUBMIT, mungkin akan diikuti oleh <CR><LF>+CMGL
	Mode PUD:	+CMGL: <index>, <stat>, [<alpha>,] <length>
		<CR> <LF> <pdu> (Untuk SMS-DELIVER dan SMS-SUBMIT, mungkin akan diikuti oleh <CR><LF>+CMGL:<index> yang lain

Mode PUD:	+CMGL: <index>, <stat>, [<alpha>,] <length> <CR> <LF> <pdu> (Untuk SMS-DELIVER dan SMS-SUBMIT, mungkin akan diikuti oleh <CR><LF>+CMGL:<index> yang lain	

Perintah	**Kemungkinan Jawaban**
AT+CMGL="REC UNREAD" Cat: Daftar pesan yang belum terbaca dalam mode text	+CMGL: 1, "REC UNREAD", "0146290800", <CR><LF> pesan +CMGL: 3, "REC UNREAD", "46290800",<CR> <LF> pesan berikutnya OK Cat: 2 buah pesan belum terbaca, pesan tersebut statusnya akan berubah menjadi "REC READ" (+CSDH=0)
AT+CMGL="REC READ" Cat: Daftar pesan yang pernah dibaca dalam mode text	+CMGL: 2, "REC READ", "0146290800" <CR> <LF> Test OK
AT+CMGL="STO SENT" Cat: Baca, simpan dan kirim pesan	OK Cat: Tidak ada pesan yang diketemukan.

Definisi Nilai			
	<stat>	0:	"REC UNREAD" (Pesan diterima yang belum
		1:	"REC READ" (Pesan diterima yang sudah
		2:	"STO UNSENT" (Pesan tersimpan yang

			belum pernah terkirim)
		3:	"STO SENT" (Pesan tersimpan yang sudah terkirim)
		4:	"ALL" (Semua pesan)

Table 2. 13 SMS AT Command Mengirim Pesan +CMGS

+CMGS	Untuk mengirimkan pesan, harus diakhiri oleh karakter <crtl-Z> (ASCII 26). Pada penulisan pesan dapat menggunakan semua karakter yang ada kecuali <crtl-Z> dan <ESC> (ASCII 27). Perintah ini dapat dibatalkan dengan menggunakan karakter <ESC> pada saat penulisan. Dalam mode PDU hanya karakter heksadesimal yang dapat digunakan (1..9, A..F).
Sintaks	Sintak perintah: Mode Text: AT+CMGS=<da>[,<toda>] <CR> isi pesan <crtl-Z/ESC> Mode PUD: AT+CMGS= <length> <CR> isi PDU <crtl-Z/ESC> **Perintah** — **Kemungkinan Jawaban** AT+CMGS="+628129356812"<CR> — +CMGS: <mr> AT+CMGS=<length><CR><pdu> — +CMGS: <mr>

	Cat: Mengirim pesan dalam mode PDU		Cat: Pengiriman berhasil
	Referensi pesan <mr> yang dikembalikan ke aplikasi di berikan dari modul GSM. Angka dimulai dari 0 dan bertambah 1 untuk setiap pesan keluar (baik yang berhasil maupun tidak), angka ini besarnya hanya 1 byte jadi maksimum angka yang dikembalikan sampai 255. Cat: Angka ini bukan merupakan angka penyimpanan. Pesan keluar tidak disimpan.		
Definisi nilai	<da>		Alamat tujuan dalam format string
	<toda>		Jenis alamat tujuan.
	<length>		Panjang data dalam bentuk oktet.

Table 2. 14 SMS AT Command Menulis Pesan Kedalam Memori +CMGW

+CMGW	Perintah ini menyimpan sebuah pesan ke dalam memori penyimpanan (SMS-SUBMIT atau SMS-DELIVER). Lokasi penyimpanan <index> akan dikembalikan jika proses penyimpanan berhasil. Isi dari text atau PDU sama dengan pada perintah pengiriman pesan +CMGS	
Sintaks	Sintak perintah:	
	Mode Text:	AT+CMGW= <oa/da> [,<tooa> <toda> [, <stat>]] isi pesan <crtl-Z/ESC>
	Mode PUD:	AT+CMGW= <length> [,<stat>] <CR> isi PDU <crtl-Z/ESC>

<table>
<tr><td></td><td colspan="3">Sintak jawaban: +CMGW: <index> atau
+CMS ERROR:<err> jika penyimpanan gagal.

<table><tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr><tr><td>AT+CMGW="+628129356812"
<CR> Halo apa kabar? <ctrl-Z>
Cat: Menulis pesan dalam mode text</td><td>+CMGW:4
OK
Cat: Pesan tersimpan dengan indek 4</td></tr><tr><td>AT+CMGW=<length><CR><pdu>
<ctrl-Z>
Cat: Menulis pesan dalam mode PDU</td><td>+CMGW:<index>
OK
Cat: Pesan tersimpan pada indek <index></td></tr></table></td></tr>
<tr><td>Definisi Nilai</td><td><oa/da></td><td></td><td>Alamat asal atau tujuan dalam format string</td></tr>
<tr><td></td><td><tooa/toda></td><td></td><td>Jenis alamat asal atau tujuan.</td></tr>
<tr><td></td><td><stat></td><td></td><td>Jenis integer dalam mode PDU (nilai default 2) atau jenis string pada mode text (nilai default "STO UNSENT"). Ini menunjukan status pesan pada memori. Jika <stat> tidak digunakan, maka pesan yang disimpan akan diperlakukan sebagai pesan yang akan dikirim.</td></tr>
<tr><td></td><td><stat></td><td>0:</td><td>"REC READ"</td></tr>
<tr><td></td><td></td><td>1:</td><td>"REC UNREAD"</td></tr>
<tr><td></td><td></td><td>2:</td><td>"STO UNSENT"</td></tr>
<tr><td></td><td></td><td>3:</td><td>"STO SENT"</td></tr>
<tr><td></td><td><length></td><td></td><td>Panjang data dalam bentuk oktet.</td></tr>
</table>

Table 2. 15 SMS AT Command Mengirim Pesan Dari Memori +CMSS

<table>
<tr><td>+CMSS</td><td>Printah ini mengrimkan pesan berdasarkan lokasi yang terdapat pada tempat penyimpanan (memori).</td></tr>
<tr><td>Sintaks</td><td>Sintak perintah: AT+CMSS=<index> [<da> [,<toda>]]
Sintak jawaban: +CMSS: <mr> atau +CMS ERROR: <err> jika pengiriman gagal.
Jika alamat tujuan baru diberikan, maka alamat yang ada pada memori akan diabaikan.
<table>
<tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr>
<tr><td>AT+CMGW=08129356812<CR>
Hari ini ulang tahunku<ctrl-Z>
Cat: Simpan pesan ke memori</td><td>+CMGW:5
OK
Cat: Pesan disimpan pada indek 5</td></tr>
<tr><td>AT+CMSS=5, 08568053746
Cat: Kirim dengan nomor yang berbeda</td><td>+CMSS:<mr>
OK
Cat: Pengiriman berhasil</td></tr>
</table>
</td></tr>
</table>

Table 2. 16 SMS AT Command Set Parameter Mode Text +CSMP

<table>
<tr><td>+CSMP</td><td>Perintah ini berfungsi untuk memilih nilai pada parameter <vp>, <pid>, dan <dcs>.</td></tr>
<tr><td>Sintaks</td><td>AT+CSMP=<fo>, <vp>, <pid>, <dcs>
<table>
<tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr>
<tr><td>AT+CSMP?
Cat: Nilai sekarang</td><td>+CSMP:1,1,1,1
OK</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Cat: Tidak ada periode validitas</td></tr>
<tr><td></td><td>AT+CMPS=17,23,64,244
Cat: <vp>=23 (2 jam, format relatif), <dcs>= Alpabet GSM 8 bit</td><td>OK
Cat: Perintah berhasil</td></tr>
<tr><td>Definisi Nilai</td><td><fo></td><td>Byte terdiri dari 6 buah field yang berbeda:
<table>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td>RP</td><td>UDHI</td><td>SRR</td><td colspan="2">VPF</td><td>RD</td><td colspan="2">MTI</td></tr>
</table>
<table>
<tr><td>RP:</td><td>Reply Path (alamat jawaban), tidak digunakan pada mode text</td></tr>
<tr><td>UDHI:</td><td>User Data Header Information (Informasi header data pemakai), b6=1 jika diawal field data pemakai</td></tr>
<tr><td>SRR:</td><td>Status Report Request (Permintaan status laporan), b5=1 jika status laporan diinginkan.</td></tr>
<tr><td>VPF:</td><td>Validity Period Format (format periode validitas).
b4=0 dan b3=0 -> filed <vp> tidak</td></tr>
</table></td></tr>
</table>

			b4=1 dan b3=0 -> field <vp> ditampilkan dalam format relatif. Format lain (absolut dan enhanced) tidak dapat dilakukan.
			RD: Tolak duplikasi, b=2 untuk menerintahkan SC menolak sebuah SMS-SUBMIT untuk SM yang masih terdapat pada SC dimana mempunyai <rm> dan <da> yang sama dengan SM yang dikirim sebelumnya dari <oa> yang sama.
			MTI: Messge Type Indicator (Indikasi jenis pesan). b1=0 dan b0=0 -> SMS-DELIVER (dari SC ke MS). b1=0 dan b0=1 -> SMS-SUBMIT (dari MS ke SC)
			<vp> hanya bisa pada format 'relatif'. Nilai asal adalah 167 (24 jam). Ini menandakan satu oktet bisa menggambarkan beberapa nilai yang berbeda:

Nilai VP	Nilai periode validitas
0-143	(VP+1)*5 menit
144-167	12 jam + ((VP-143)*30 menit)
168-196	(VP-166)*1 hari
197-255	(VP-192)*1 minggu

	<pid>		Untuk menandakan protokol layer tertinggi yang sedang dipergunakan atau menandakan kerja

			antara beberapa alat telematika. Sebagai contoh 0x22 digunakan untuk grup telefax, 0x24 untuk telepon suara, 0x25 untuk ERMES.
	<dcs>		Digunakan untuk menentukan cara me-encode informasi. Alpabet UCS2 dan text yang terkompres tidak bisa di-encode. Hanya alpabet GSM dan data 8 bit saja yang bisa di-encode.

Table 2. 17 SMS AT Command Hapus Pesan +CMGD

+CMGD	Perintah ini digunakan untuk menghapus pesan yang tersimpan.
Sintaks	AT+CMGD=<index>

Perintah	**Kemungkinan Jawaban**
	+CMTI:"SM", 3 Cat: Sebuah pesan baru diterima
AT+CMGR=3 Cat: Baca pesan	+CMGR: "REC UNREAD", "08129356812",, "03/05/25, 18:19:20+00" <CR><LF> Isi Pesan
AT+CMGD=3 Ctt: Hapus pesan tersebut	+CMGS:1 OK Cat: Penghapusan berhasil

Table 2. 18 SMS AT Command Alamat Service Center +CSCA

+CSCA	Perintah ini digunakan untuk memilih Service Center

<table>
<tr><td></td><td>mana yang akan digunakan untuk mengirim pesan.
Modul GSM tidak mempunyai nilai default untuk alamat ini. Jika aplikasi mencoba mengirimkan pesan tanpa alamat service center maka pesan kesalahan akan tampil. Jadi aplikasi harus menetapkan alamat pada saat awal. Alamat ini berlaku sepanjang waktu. Aplikasi bisa merubahnya jika diinginkan.</td></tr>
<tr><td>Sintaks</td><td>AT+CSCA=<SC address>
<table><tr><th>Perintah</th><th>Kemungkinan Jawaban</th></tr><tr><td>AT+CMGS='+628129356812"
<CR> Halo apa kabar?<crtl-Z>
Cat: Kirim pesan</td><td>+CMS:ERROR: 330
Cat: Secvice Center tidak diketahui</td></tr><tr><td>AT+CSCA="6281100000"
Cat: Menetapkan Service Center</td><td>OK</td></tr><tr><td>AT+CMGS='+628129356812"
<CR> Halo apa kabar?<crtl-Z>
Cat: Kirim pesan</td><td>+CMGS:1
OK
Cat: Pesan terkirim</td></tr></table></td></tr>
</table>

Table 2. 19 SMS AT Command Pilih Tipe Cell Broadcast Message +CSCB

+CSCB	Perintah ini digunakan untuk memilih tipe CBM yang akan diterima oleh ME. Perintah ini bisa digunakan untuk menrima pesan dalam format text atau PDU.

Sintaks	AT+CSCB=<mode> [,<mids>[,<dcss>]] Test perintah (AT+CSCB ?) tidak dapat dilakukan. Aktifasi penerimaan CBM (<mode>=0) hanya dapat memilih identifikasi pesan tertentu (daftar pada <mids>) untuk bahasa tertentu (daftar pada <>dcss), tetapi menonaktifkan berarti memberhentikan semua penerimaan CBM (AT+CSCB=1). Identifikasi pesan (parameter <mids>) menunjukan tipe identifikasi pesan apa yang harus didengar oleh ME.

2.3 Borland Delphi

Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai jenis aplikasi dapat dibuat dengan menggunakan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database, web, dan interface dengan hardware.

Secara umum, kemampuan Delphi adalah menyediakan kompnen-komponen dan bahasa pemrograman yang andal, sehingga memungkinkan untuk membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Delphi menyediakan fasilitas yang luas mulai dari fungsi untuk membuat form hingga fungsi untuk menggunakan beberapa format file basis data yang populer (dBASE, Paradok, dll).

2.3.1 Kelebihan Delphi

Kelebihan-kelebihan Delphi antara lain :

- Komponen yang dapat dipakai ulang dan dapat dikembangkan.

  Delphi mempermudah pembuatan program bagi komponen komponen windows seperti label, button, dan bahkan dialog.

  Komponen visual dan non-visual sudah tersedia dalam Delphi seperti komponen button, komponen-komponen *database*, komponen menu, dan dialog.

- Dapat mengakses VBX

  Delphi dapat mengakses objek-objek VBX secara langsung. Dalam Delphi VBX dianggap sebagai kumpulan komponen yang dapat digunakan langsung untuk membuat aplikasi.

- Template aplikasi dan Template Form

  Template tersebut dapat dipakai untuk membuat semua aplikasi dengan lebih cepat.

- Lingkungan pengembangan Delphi yang lebih produktif
- Program yang terkompilasi

  Di dalam Delphi program yang dihasilkan benar-benar program yang terkompilasi tanpa *interpreter* dan *pcode* sehingga dapat berjalan lebih cepat.

- Kemampuan mengakses data dalam bermacam format.

2.4.2 Komponen Delphi

Komponen-komponen Delphi yang dipakai dalam membuat program aplikasi antara lain :

1. Project

   Project adalah sekumpulan form, unit, dan beberapa hal lain dalam program aplikasi. Singkatnya project adalah program aplikasi itu sendiri.

2. Form

   Form adalah suatu object yang dipakai sebagai tempat bekerja program aplikasi. Form berbentuk jendela dan dapat dibayangkan sebagai kertas atau meja kerja yang dapat digambari atau diletakkan object-object lain diatasnya.

3. Unit

   Unit adalah modul kode program. Satu project bila mempunyai satu unit atau lebih. Dalam Delphi, ada unit yang tak terpisah dari form yang disebut unit form. Setiap kali dibuat satu form, maka otomatis pula dibuat satu unit.

   Manfaat penggunaan unit dalam program aplikasi antara lain seperti berikut ini :

   - Membagi program aplikasi yang besar dalam beberapa unit sehingga dapat mengedit unit-unit tertentu saja. Hal ini sangat memudahkan dalam perancangan, pengubahan, dan penyelesaian program aplikasi.
   - Membuat *library* (daftar pustaka) berupa *function* dan *procedure*, sehingga memudahkan sharing antar program.
   - Unit dapat dikompilasi terpisah dari program aplikasi, sehingga program aplikasi lain yang memerlukan unit serupa dapat menggunkannya tanpa harus menulis kembali kode programnya.

4. Program

   Program dibangun dari satu unit atau lebih. Pada Delphi, program akan otomatis dibuat dan di update oleh Delphi pada saat menambahkan, mengedit, atau menghapus form dan unit.

5. Property

   Property digunakan untuk mendefinisikan atribut atau setting suatu *object*. Setting property akan menentukan cara kerja dari object yang bersangkutan saat project dijalankan, misalnya menentukan warna object, ukuran object, font yang dipakai, *database* yang diambil, dll.

6. Event.

   Event adalah peristiwa atau kejadian yang diterima oleh suatu object, misalnya klik, drag, tunjuk, dll.

7. Method

   Method adalah procedure atau perintah yang melekat pada suatu object.

2.4 SQL (Structured Query Language)

SQL adalah kependekan dari *Structured Query Language*. Bahasa ini digunakan untuk mengakses basis data relational.

Statemen SQL secara garis besar di bagi menjadi dua kategori :

- Data Definition Language (DDL) statement

  Statement DDL merupakan bahasa definisi data yang digunakan untuk membuat, mengubah, dan menghapus struktur data asli (hold data), yang berkaitan dengan database itu sendiri, table, dan elemen lain yang merupakan bagian dari *database* seperti domain, index, triggers, strored procedure, role,

dan shadow. Semua definisi objek akan dilakukan dengan statement DDL yang dikenal dengan *metadata*. Statement DDL dimulai dengan keyword CREATE (membuat struktur data), ALTER (memodifikasi struktur data), dan DROP (menghapus struktur data). Sebagai contoh :

✓ Pembuatan tabel dengan menggunakan pernyataan CREATE TABLE.

CREATE TABLE nama_tabel (

Nama_kolom_1 tipe_data ([*ukuran*]),

Nama_kolom_2 tipe_data ([*ukuran*]),

……,

Nama_kolom_n tipe_data (*[ukuran*]);

✓ Memodifikasi tabel dengan mengguankan pernyataan ALTER TABLE

ALTER TABLE nama_table operational [, operational]

✓ Menghapus tabel dengan menggunkan DROP TABLE

DROP TABLE nama_table

- Data Manipulation Language (DML) statement

Statement DML merupakan bahasa manipulasi data yang digunakan untuk memanipulasi data dengan data struktur. Empat pokok statement DML yaitu INSERT (menambah data), UPDATE (memodifikasi data yang sudah ada), DELETE (menghapus data) dan SELECT (melihat data), sebagai contoh :

✓ Memasukan data ke sebuah tabel dengan menggunakan pernyataan INSERT.

INSERT INTO nama_tabel [*nama_kolom*,] VALUES (value)

✓ Mengupdate data pada sebuah tabel dengan menggunakan pernyataan UPDATE.

UPDATE Tabel

Set Kolom = Value

Where kondisi

✓ Menghapus data pada sebuah tabel dengan menggunakan pernyataan DELETE.

DELETE FROM tabel

WHERE kondisi

✓ Melihat data pada sebuah tabel dengan menggunakan pernyataan SELECT

*SELECT * FROM tabel*

2.5 MySQL

MySQL adalah suatu server database SQL yang multi user dan multithread (*MySQL Reference Manual for version 3.23.11-alpha, www.mysql.com, 2000*). SQL (Structured Query Language) adalah bahasa standar database yang paling populer di dunia. SQL adalah bahasa yang distandarisasi untuk memudahkan menyimpan, mengupdate, dan mengakses informasi. Sebagai contoh, SQL dapat digunakan untuk mengambil informasi suatu produk dan menyimpan informasi pelanggan untk sebuah web site. MySQL juga sangat cepat dan cukup fleksibel untuk menyimpan log dan gambar.

Tujuan utama MySQL adalah kecepatan, kekuatan, dan kemudahan penggunaan. MySQL pada awalnya dikembangkan karena dibutuhkan sebuah server SQL yang bisa menangani database yang sangat besar dan cepat dibandingkan produk lain dengan hardware yang tidak murah. MySQL telah dicobakan pada suatu server dengan 40 database yang terdiri dari 10.000 table

dimana 500 table lebih berisi lebih dari 7.000.000 data/record. Hal tersebut menunjukkan bahwa MySQL dapat menampung data lebih dari 100 Giga Byte.

Cara mengucapkan MySQL dalam ejaan bahasa inggris adalah "My Ess Que Ell" bukan "My-Sequel".

2.5.1 Karakteristic MySQL

- Mendukung multi threading menggunakan kernel thread, yang artinya dapat menggunakan beberapa CPU.
- Dapat bekerja di berbagai platform (Sistem Operasi).
- Mendukung banyak tipe field, seperti signed/unsigned integers dengan panjang 1, 2, 3, 4, dan 8 byte, `FLOAT`, `DOUBLE`, `CHAR`, `VARCHAR`, `TEXT`, `BLOB`, `DATE`, `TIME`, `DATETIME`, `TIMESTAMP`, `YEAR`, `SET`, dan `ENUM`.
- Fungsi join yang sangat cepat.
- Mendukung penggunaan fungsi dan operator pada perintah SQL untuk SELECT dan WHERE. Contoh:

  'SELECT CONCAT(first_name, " ", last_name) FROM tbl_name

  WHERE income/dependents > 10000 AND age > 30'
- Fungsi SQL dibuat dengan menggunakan librari class yang sudah optimal, sehingga hasilnya akan sangat cepat.
- Dapat men-Query beberapa table dari database yang berbeda.
- Untuk password dan hak akses sangat fleksibel dan aman, serta membolehkan verifikasi berdasarkan host. Password untuk koneksi akan aman, karena sudah diekripsi.
- Dapat menggunakan ODBC untuk koneksi menggunakan Microsoft Window.

- Satu table dapat mempunyai maksimal 16 indeks. Setiap indeks bisa terdiri dari 1 sampai 16 kolom.
- Dapat menangani database besar. Sudah ada beberapa database yang memiliki 50.000.000 record.
- Dibuat dengan bahasa C dan C++.
- Sistem alokasi memori thread_based yang sangat cepat.
- Nama kolom boleh sama dengan nama fungsi.
- Server dapat menghasilkan pesan kesalahan kepada client dalam banyak bahasa.
- Alias/sinonim pada table atau kolom sesuai dengan standar SQL92.
- Perintah DELETE, INSERT, REPLACE, dan UPDATE, menghasilkan pesan berapa record yang terpengaruh.
- Koneksi Clients ke server MySQL, bisa menggunakan TCP/IP, soket unix atau named pipes pada Microsft Windows NT.

2.5.2 Kekurangan MySQL

MySQL menjadi bahan perbincangan pada komunitas pengembang aplikasi. Banyak pengembang aplikasi menyukai MySQL karena gratis, tidak banyak memakan resource, sangat cepat, dan sangat mudah dipelajari dibandingkan dengan database lain seperti Oracle dan Sybase. Bagaimanapun MySQL mendapatkan kecepatannya tanpa beberapa fasilitas umum yang terdapat pada database lain dan kekurangan ini yang menyebabkan banyak yang belum memakai/menggantikan databasenya dengan MySQL. Tetapi walaupun memiliki beberapa kekurangan, banyak pula pengembang aplikasi yang tidak mempermasalahkannya.

Berikut beberapa fasilitas yang tidak/belum didukung oleh MySQL:

- Referential Integrity
- Transaction
- Stored Procedure

Berdasarkan informasi di internet MySQL masih terus dikembangkan untuk mengatasi kekurangan-kekurangan tersebut.

2.6 Data Flow Diagram (DFD)

Dataflow diagram merupakan alat pemodelan yang menggambarkan sebuah sistem sebagai sebuah jaringan dari proses fungsional, terbubung dengan yang lainnya melalui "pipelines" dan tempat penyimpanan data (*Edward Yourdon, Modern Structure Analysis, 1989).*

Komponen DFD terdiri dari:

- Proses, berbentuk bulatan. Proses menandakan bagian dari sistem yang mentransformasikan input menjadi output.
- Flow, berbentuk panah yang menuju ke atau keluar dari proses. Digunakan untuk menjelaskan pergerakan informasi dari bagian sistem yang satu menuju bagian sistem yang lain.
- Store, berbentuk dua buah garis pararel. Digunakan untuk memodelkan sekumpulan paket data.
- Terminator, berbentuk segiempat. Terminator menggambarkan eksternal entiti yang mana berkomunikasi dengan sistem. Biasanya terminator adalah seorang atau grup orang.

2.7 Pemodelan Data

Pemodelan data menjawab berbagai pertanyaan yang berhubungan dengan berbagai aplikasi pemrosesan data. Objek data utama apa yang akan diproses oleh sistem? Komposisi apa yang terdapat pada setiap objek data dan atribut apa yang menggambarkan objek tersebut? Dimana objek itu terletak? Apa hubungan antara objek yang satu dengan yang lain dan apa hubungan antara objek dan proses yang merubahnya?.

Untuk menjawab pertanyaan tersebut metode pemodelan data menggunakan Entity Relationship Diagram (ERD) (*Roger S. Pressman, Software Engineering, 2001*). ERD memungkinkan software engineer untuk mengidentifikasi objek-objek data dan hubungan antar objek menggunakan notasi grafis. Dalam konteks analisis terstruktur, ERD menggambarkan semua data yang diinput, disimpan, berubah, dan dihasilkan pada aplikasi.

Entity Relationship Diagram berfokus semata-mata hanya pada data, menggambarkan sebuah "jaringan data" yang ada pada suatu sistem. ERD sangat berguna untuk aplikasi yang mana data dan relasi yang mengatur data tersebut cukup komplek.

2.7.1 Objek Data, Atribut, dan Relasi

Model data terdiri dari 3 buah informasi yang saling berhubungan: objek data, atribut yang menggambarkan objek data dan relasi yang menghubungkan objek data dengan yang lainnya.

- Objek Data

  Objek data mewakili hampir semua gabungan informasi yang harus dimengerti oleh software. Gabungan informasi, maksudnya adalah sesuatu yang memiliki beberapa properti dan atribut yang berbeda.

  Objek data bisa disebut sebagai ekternal entity (misalnya apapun yang menghasilkan atau mengambil informasi), benda (misalnya laporan atau tampilan), kejadian (misalnya panggilan telepon) atau peristiwa (misalnya alarm), peran (misalnya seorang sales), sebuah unit organisasi (misalnya departemen akuntansi), sebuah tempat (misalnya gudang) atau sebuah struktur (misalnya file). Sebagai contoh, seseorang dan mobil dapat dilukiskan sebagai objek data dalam pengertian bahwa keduanya dapat didefinisikan dalam bentuk suatu atribut. Deskripsi objek data menggabungkan objek data dan semua atributnya.

  Objek data dapat digambarkan sebagai sebuah tabel. Header tabel mencerminkan atribut dari objek. Isi tabel menggambarkan instance khusus dari objek data.

- Atribut

  Atribut menggambarkan properti dari objek data dan membawa satu dari tiga karakteristik yang berbeda. Atribut bisa digunakan untuk (1) nama instance dari objek data, (2) menggambarkan instance atau (3) membuat referensi ke instance lain dari tabel lain. Satu atau lebih atribut harus ditetapkan sebagai *identifier.* Atribut identifier menjadi sebuah "key" untuk memudahkan mencari sebuah instance dari objek data.

- Relasi

  Objek data terhubung dengan objek data yang lain dalam cara yang berlainan. Misalkan ada dua buah objek data, **buku** dan **tokobuku**. Hubungan terjadi antara **buku** dan **tokobuku** karena kedua objek saling berkaitan. Relasi yang relevan dari kedua objek tersebut adalah:

  - Tokobuku memesan buku
  - Tokobuku memperlihatkan buku
  - Tokobuku menyimpan buku
  - Tokobuku menjual buku
  - Tokobuku mengembalikan buku

2.7.2 Cardinality dan Modality

Elemen dari pemodelan data memberikan dasar untuk mengetahui masalah daerah informasi. Bagaimanapun tambahan informasi yang berhubungan dengan elemen dasar ini harus juga dipahami.

Pasangan objek/relasi yang menunjukkan **objek x** berhubungan dengan **objek y** tidak menyediakan informasi yang cukup untuk tujuan software engineering. Pemahaman mengenai berapa banyak kejadian dari objek x yang menghubungkan dengan berapa banyak kejadian dari objek y sangat diperlukan. Hal ini yang menggiring kedalam konsep pemodelan data yang disebut *cardinality.*

- Cardinality

Model data harus mampu untuk menggambarkan jumlah objek kejadian dalam relasi yang telah diberikan. Tillmann mendefinisikan *cardinality* dari suatu pasangan objek/relasi dalam cara berikut:

Cardinality merupakan spesifikasi dari jumlah kejadian dari satu [objek] yang bisa berhubungan dengan jumlah kejadian dari yang lain [objek]. Cardinality biasanya digambarkan dengan 'one' or 'many'. Berdasarkan kombinasi dari 'one' dan 'many', dua objek dapat berhubungan sebagai:

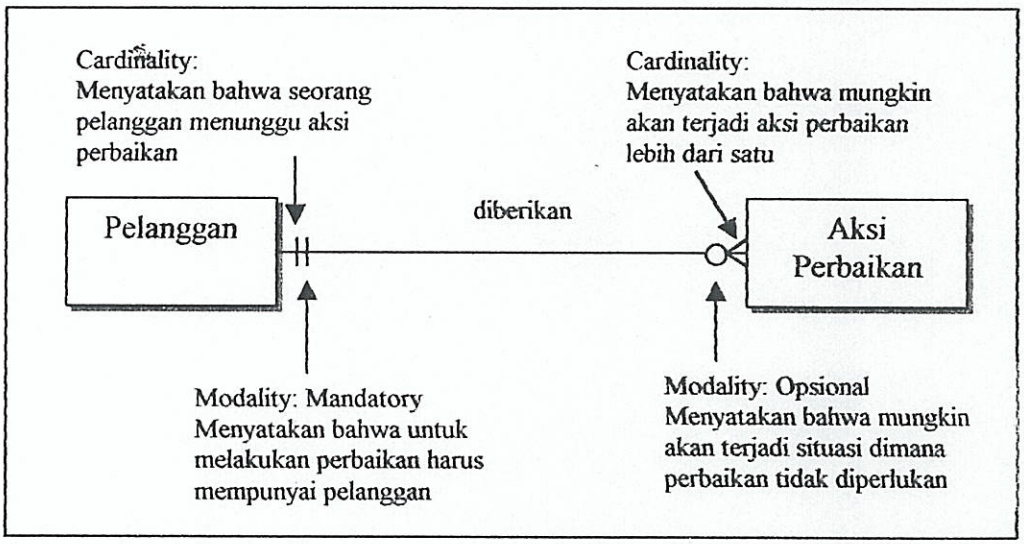
- One-to-one (1:1) – Suatu kejadian dari objek 'A' dapat berhubungan dengan dan hanya satu kejadian dari objek 'B', dan kejadian pada objek 'B' hanya bisa berhubungan dengan satu objek 'A'.
- One-to-many (1:N) – Satu kejadian dari objek 'A' dapat berhubungan dengan satu atau lebih kejadian pada objek 'B', tapi kejadian pada objek 'B' hanya bisa berhubungan dengan satu kejadian pada objek 'A'.
- Many-to-many (M:N) – Suatu kejadian pada objek 'A' dapat berhubungan dengan satu atau lebih kejadian pada objek 'B', sementara kejadian pada objek 'B' dapat berhubungan dengan satu atau lebih kejadian pada objek 'A'.

Cardinality menggambarkan "jumlah maksimum objek yang dapat berpartisipasi dalam relasi". Itu tidak mengindikasikan apakah suatu objek data harus ikut serta dalam suatu hubungan. Untuk menentukan informasi ini, model data menambahkan modality pada pasangan objek/relasi.

- Modality

  Modality suatu relationship adalah 0 jika tidak ada kebutuhan untuk terjadinya relasi atau relasi merupakan opsional. Modality adalah 1 jika terjadinya suatu

relasi adalah suatu mandatory (keharusan). Sebagai ilustrasi, anggap software yang digunakan oleh suatu perusahaan telepon lokal untuk memproses permintaan servis lapangan. Pelanggan menyatakan bahwa telah terjadi suatu masalah. Jika masalah yang terjadi relatif ringan, maka hanya satu perbaikan yang dilakukan. Akan tetapi bila masalahnya cukup komplek maka aksi perbaikan yang dilakukan lebih dari satu. Di bawah ini digambarkan relasi, cardinality dan modality antara objek data **pelanggan** dan **aksi perbaikan**.



Gambar 2. 2 Cardinality dan Modality

Berdasarkan gambar di atas, sebuah relasi one to many terjadi. Seorang pelanggan bisa diberikan kosong atau banyak aksi perbaikan. Simbol pada garis relasi dekat dengan kotak objek data melambangkan cardinality. Garis vertikal menggambarkan one dan tiga gigi garpu menggambarkan many. Garis vertikal kedua dari kiri melambangkan bahwa harus ada pelanggan untuk terjadinya suatu aksi perbaikan. Bulatan melambangkan bahwa kemungkinan tidak terjadinya aksi perbaikan untuk masalah yang dilaporkan oleh pelanggan.

2.7.3 Entity Relationship Diagram

Pasangan objek/relasi yang telah diterangkan di atas merupakan landasan model data. Pasangan ini dapat digambarkan secara grafis menggunakan Entity Relationship Diagram.

2.8 Kamus Data

Kamus data merupakan daftar tentang semua elemen yang berhubungan dengan sistem dan terdefinisi dengan tepat, sehingga user dan sistem analis mempunyai pengertian yang sama tentang input dan output.

Kamus data terdiri dari kamus data komposit dan elementer. Kamus data komposit adalah daftar tentang semua elemen data yang berhubungan dengan sistem dimana elemen data komposit dapat terdiri dari dua atau lebih elemen data elementer.

Sedangkan kamus data elementer adalah daftar tentang semua data yang berhubungan dengan sistem sehingga data yang mengalir dapat didefinisikan dan tersimpan secara lengkap.

2.9 State Transition Diagram (STD)

State Transition Diagram (STD) merupakan salah satu model untuk menggambarkan sifat ketergantungan terhadap waktu suatu sistem.

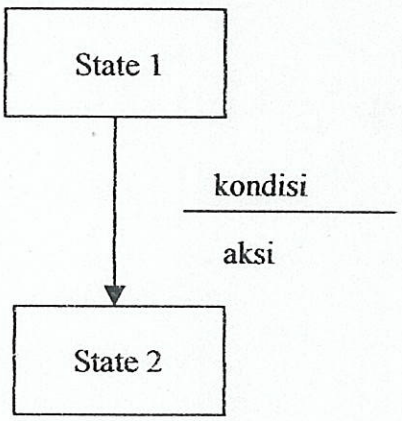
Sebelumnya model dari sistem ketergantungan terhadap waktu hanya penting untuk sistem dengan kategori spesial yang dikenal dengan sistem real time. Contoh dari sistem ini adalah kontrol proses, sistem pengaturan telepon, sistem penerimaan high-speed data, perintah kemiliteran, dan sistem kontrol.

Beberapa sistem tersebut bersifat pasif, yaitu tidak melakukan kontrol terhadap lingkungannya tetapi lebih kepada bereaksi terhadap lingkungan atau mengambil data dari lingkungan tersebut. Banyak dari sistem penerimaan data high-speed termasuk dalam kategori ini (contoh sistem pengambilan data ilmiah dari satelit). Sistem real time yang lain lebih bersifat aktif, yang berarti sistem tersebut lebih melakukan kontrol terhadap beberapa aspek dalam lingkungannya. Sistem kontrol proses termasuk pada kategori ini.

2.9.1 Notasi STD

Komponen utama dari STD adalah state dan panah yang menggambarkan perubahan state. State dilambangkan dengan kotak yang merupakan kumpulan keadaan atau atribut yang mencirikan sesuatu pada waktu tertentu dan keadaan tertentu.

Perubahan dari satu state terhadap state yang lain digambarkan dengan adanya perubahan suatu kondisi yang menyebabkan perubahan state dan aksi yang diterima sistem saat perubahan state. Kondisi dan aksi digambarkan disebelah panah.



Gambar 2. 3 Notasi STD