

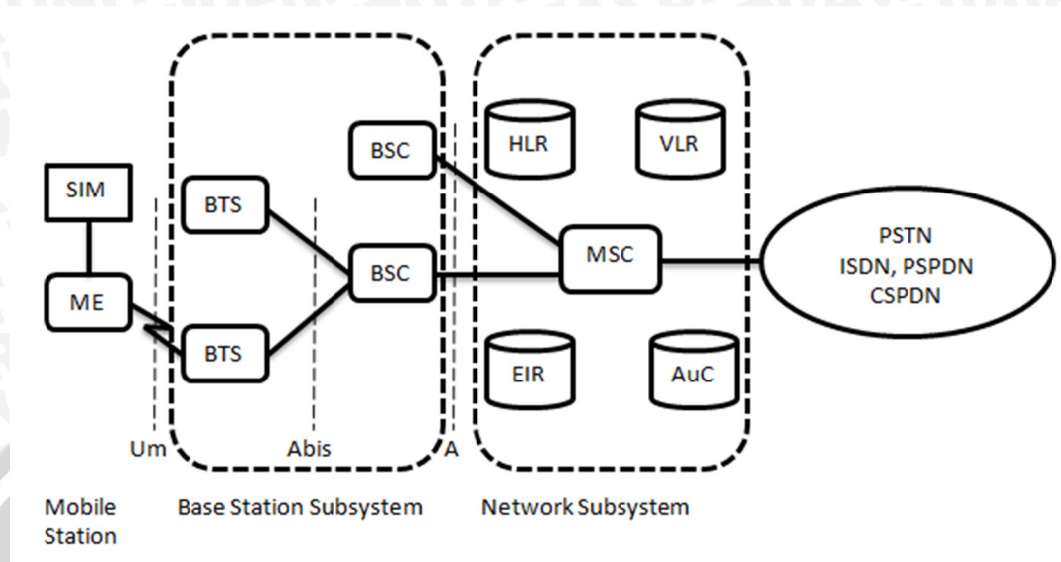
## BAB II DASAR TEORI

Pada bab ini dibahas tentang berbagai landasan teori yang akan sangat membantu untuk memahami sistem. Serta dapat digunakan sebagai acuan di dalam merencanakan suatu sistem. Dengan pertimbangan hal-hal tersebut, maka landasan teori merupakan bagian yang harus dipahami untuk pembahasan lebih lanjut. Dalam landasan teori ini akan dibahas teori dasar yang berhubungan dengan *Arsitektur jaringan GSM, Arsitektur Short Message Service (SMS), Smart Card tipe ACR30, Modul COM-USB, Mikrokontroler ATmega 32, Modul Smart Card, Delphi.*

### 2.1 Arsitektur jaringan GSM

GSM (*Global System For Mobile Telecommunication*) merupakan sistem telekomunikasi yang menggunakan sistem selular digital dengan menggunakan sinyal digital dalam transmisi datanya sehingga membuat kualitas data maupun *bitrate* yang dihasilkan menjadi lebih baik dibandingkan dengan sistem analog.

Keistimewaan dari GSM yang tidak terdapat pada sistem analog maupun pada *American Digital Cellular (ADC)* adalah adanya standardisasi *interface* antar masing-masing sub sistem. Dengan demikian, GSM menjanjikan suatu sistem yang tidak harus dimonopoli oleh satu merek. Dalam arti bahwa *Switching, Base Station*, dan *Out Station* dapat berasal dari merek/pemasok yang berbeda. Kondisi ini jelas sangat menguntungkan pihak operator, karena tidak ada ketergantungan sama sekali terhadap satu *supplier*.



**Gambar 2.1** Arsitektur jaringan GSM

(sumber [www.mobilecomms-technology.com](http://www.mobilecomms-technology.com))

Arsitektur jaringan GSM terdiri dari 3 komponen utama yakni:

1. Mobile Station
2. Base Station Subsystem (BSS)
3. Network Subsystem (NSS)

### 2.1.1 ME (Mobile Equipment)

Entitas Mobile Station terdiri dari Mobile Equipment (ME) yakni perangkat keras & perangkat lunak untuk transmisi radio yang dikenal dengan istilah telepon seluler (ponsel) dan Subscriber Identification Module (SIM).

Mobile equipment (ME) secara unik diidentifikasi dalam format International Mobile Equipment Identity (IMEI). SIM card berisi International Mobile Subscriber Identity (IMSI) yang digunakan untuk indentifikasi pelanggan ke sistem, kunci rahasia (untuk autentifikasi) serta menyimpan informasi lainnya seperti phone book atau pesan sms. SIM card dapat diproteksi dari



penggunaan yang tidak terotorisasi dengan password atau personal identity number (PIN).

### 2.1.2 BSS (*Base Station Subsystem*)

*Base Station Subsystem* (BSS) terdiri dari *Base Transceiver System* (BTS) dan *Base Station Controller* (BSC). *Base Station Controllers* (BSC) mengontrol dan mengatur beberapa BTS. BSC bertanggung jawab untuk memelihara koneksi saat panggilan dan kepadatan lalu lintas panggilan pada areanya dan meneruskannya ke *Network Subsystem*. BSC juga menangani *setup* radio-channel, *frequency hopping*, serta proses handover. BTS merupakan alat *tranceivers* radio (transmitter receiver radio) pada suatu area didefinisikan sebagai sebuah cell dan menangani protokol radio-link dengan *Mobile Station* lewat Um interface yang juga dikenal dengan *air interface*.

### 2.1.3 NSS (*Network Subsystem*)

*Network Subsystem* terdiri dari *Mobile Switching Centres* (MSC) dan beberapa *database* yang terhubung dengannya seperti *Home Location Register* (HLR), *Visitor Location Register* (VLR), *Authentication Center* (AuC) serta *Equipment Identity Register* (EIR). *Mobile Switching Centers* (MSC) berfungsi untuk *switching* suatu panggilan telepon dari jaringan internal atau dari jaringan lain, *call routing* untuk pelanggan yang melakukan *roaming*, menyimpan informasi billing serta data base lain yang berisi informasi *subscriber ID* (IMSI), nomor ponsel pelanggan, beberapa layanan atau larangan yang berkaitan dengan pelanggan, *autentifikasi* serta informasi lokasi pelanggan.

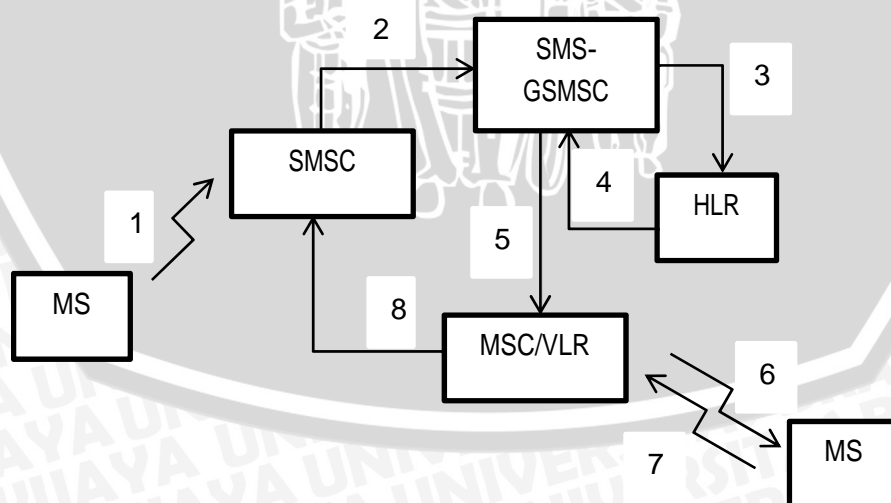
HLR dan VLR bersama dengan MSC menyediakan *call-routing* dan fungsi *roaming* dari GSM. HLR berisi semua informasi administrasi dari setiap pelanggan yang tersambung pada jaringan GSM. VLR berisi informasi administrasi terpilih dari HLR, yang penting untuk control panggilan (call control) dan provisi dari layanan pelanggan, dan control posisi setiap ponsel pada area geografis.

*Equipment Identity Register (EIR)* merupakan *database* yang berisi suatu daftar *valid mobile equipment* pada jaringan. Setiap *mobile station* diidentifikasi dengan *International Mobile Equipment Identity (IMEI)*. Pada kasus khusus sebuah IMEI ditandai/didaftarkan *invalid* bila ponsel dilaporkan dicuri/dirampas dari pemiliknya.

*Authentication Center (AuC)* merupakan *database proteksi* yang menyimpan salinan dari kunci rahasia (*secret key*) yang terdapat pada setiap *SIM card* pelanggan. *Proteksi* ini digunakan untuk *autentifikasi* dan *enkripsi* pada *channel radio*.

**2.2 Arsitektur SMS (Short Message Service)**

Suatu jaringan GSM terdiri dari sejumlah komponen entitas fungsional untuk layanan SMS. Komponen-komponen pada jaringan GSM yang berhubungan SMS yaitu *Short Message Entities (SME)*, *SMS-Center (SMSC)*, *Mobile Switching Center (MSC)*, *SMS Gateway-MSC (SMC-GMSC)*, *Home Location Register (HLR)*, *Visitor Location Register (VLR)*, *Base Station Sistem (BSS)* dan *Mobile Station (MS)*.



**Gambar 2.2** Diagram alir pengiriman SMS (*Short Message Service*)

(Sumber Gupta Punet, 2000)



1. MS mengirim pesan ke SMSC.
2. SMSC mengirimkan pesan ke SMS-GMSC.
3. SMS-GMSC menginterogasi HLR untuk informasi *routing*.
4. HLR membalas informasi *routing* ke SMS-GMSC.
5. SMS-GMSC meneruskan pesan ke MSC/VLR.
6. MS di-paging dan koneksi terbentuk antara MS dan *network*, sebagaimana dalam *setup* panggilan normal. (Dengan demikian posisi MS diketahui dan apakah MS boleh berada dalam *network* / proses otentikasi).
7. Jika otentikasi berhasil, MSC/VLR mengirim pesan sms tersebut ke MS. SMS dikirim melalui kanal signaling SDCCH.
8. Jika pengiriman berhasil, *delivery report* dikirim dari MSC/VLR ke SMSC. Namun jika tidak, MSC/VLR akan menginformasikan ke HLR, dan *failure report* dikirim ke SMS-C.

Pada kasus pengiriman yang gagal, HLR dan VLR akan mendapat informasi "*Messages waiting*" yang menunjukkan ada pesan di SMSC yang menunggu untuk dikirimkan ke MS. Informasi di HLR terdiri dari list SMSC pengirim pesan, sedangkan di VLR terdapat "*flag*" yang menunjukkan apakah list pesan dalam keadaan kosong atau tidak.

Jika MS *available* dan siap menerima pesan, maka HLR akan memberitahu SMSC.

### 1) MS (*Mobile System*)

*Mobile Station* (MS) merupakan bagian/tingkatan terendah dalam sistem GSM. MS berada di tingkat pelanggan dan bersifat *portabel*. MS terdiri dari terminal (*Mobile Equipment*) dan kartu SIM (*SIM-card*). Terminal

merupakan peralatan *mobile* yang digunakan pelanggan untuk mendapatkan layanan (*service*) dari jaringan. Terminal dapat berupa telepon seluler, PDA, laptop, dan sebagainya. Kartu SIM (*Subscriber Identitas Module*) adalah kartu yang memberikan mobilitas secara personal sehingga pengguna mempunyai akses ke layanan jaringan yang digunakannya. Kartu SIM untuk jaringan GSM hanya dapat digunakan pada terminal-terminal yang bekerja untuk jaringan GSM saja. Dengan bantuan kartu SIM, terminal dapat digunakan untuk menerima layanan operator jaringan, seperti membuat dan menerima panggilan, mengirim dan menerima SMS, atau menerima layanan yang disediakan layanan lain yang disediakan operator. Kartu SIM akan melindungi baik terminal maupun kartu SIM itu sendiri dari pengguna yang tidak bertanggung jawab dengan suatu PIN (*password*) dan nomor personal.

## 2) BSS (*Base Station System*)

*Base Station System* (BSS) melakukan pengiriman dan penerimaan panggilan atau pesan dari suatu MS. BSS bertugas untuk melaksanakan fungsi-fungsi yang berhubungan dengan *relay radio*, yaitu mengirimkan dan menerima informasi (suara atau pesan teks) melalui gelombang radio dari dan ke MS. BSS terdiri atas *Base Station Controller* (BSC) dan *Base Transceiver Station* (BTS).

### a. BSC (*Base Station Controller*)

*Base Station Controller* (BSC) terdiri dari beberapa BTS dan setiap BTS mempunyai sel area (jangkauan) pancar gelombang radio yang berbeda. Tidak jarang terjadi area yang *over lapping* antara BTS satu dengan BTS di dekatnya. Walaupun demikian kontinuitas komunikasi MS dengan infrastruktur jaringan seluler selalu terjaga. BSC digunakan untuk mengatur perpindahan MS dari satu sel area yang dijangkau BTS satu ke sel area lain yang dijangkau BTS lain.

### b. BTS (*Base Transceiver Station*)

*Base Transceiver Station* (BTS) yaitu untuk mengatur interkoneksi antara infrastruktur sistem seluler dengan MS. BTS selalu



memonitor MS yang masuk/keluar jangkauan sel BTS. Luas jangkauan BTS dipengaruhi antara lain oleh lingkungan, topografi, adanya bangunan-bangunan tinggi dan sebagainya.

### 3) NSS (*Network Switching System*)

*Network Switching System* (NSS) bertanggung jawab atas pemanggilan dan pengiriman pesan. NSS menyediakan semua fungsionalitas yang diperlukan untuk melayani pengguna bergerak seperti registrasi, otentifikasi, pembaruan lokasi (*location updater*), pindah tangan (*handover*), perutean (*routing*) panggilan/pengiriman ke pengguna yang sedang berada di luar jangkauan area. NSS terdiri dari beberapa unit fungsional, yaitu:

#### a. HLR (*Home Location Register*)

*Home Location Register* (HLR) adalah basis data tempat data dan informasi pelanggan disimpan secara permanen/tetap (tidak tergantung pada posisi pelanggan). Dalam proses pengiriman pesan, HLR juga dapat memberikan informasi ke SMSC bahwa jaringan *mobile* yang dituju telah dikenali dan dapat dilakukan pengaksesan, sehingga pesan dapat segera dikirim ke tujuan. Dalam hal ini, HLR bertindak sebagai pusat informasi pelanggan.

#### b. MSC (*Mobile Switching Center*)

*Mobile Switching Center* (MSC) bertugas melakukan *switching* ke sistem, selain itu MSC berperan sebagai interkoneksi hubungan pembicaraan, baik antar pelanggan seluler maupun antar pelanggan seluler dengan jaringan tetap telepon kabel (PTSN). MSC mengendalikan panggilan dari atau ke sistem telepon. Dalam kaitannya dengan pengiriman *short message*, MSC akan mengirimkan *short message* kepada pelanggan yang dituju dengan memilih stasiun utama (*base station*) yang tepat.

#### c. VLR (*Visitor Location Register*)

*Visitor Location Register* (VLR) berfungsi untuk menyimpan data dan informasi dimulai pada saat pelanggan memasuki area yang berada dalam wilayah suatu MSC VLR tersebut. Adanya informasi

tentang pelanggan dalam VLR memungkinkan MSC melakukan hubungan baik panggilan masuk maupun panggilan keluar. Berbeda dengan HLR, VLR bertindak sebagai basis data pelanggan yang bersifat dinamis karena selalu berubah setiap waktu menyesuaikan dengan pelanggan yang masuk atau berpindah wilayah suatu MSC. Data yang tersimpan dalam VLR akan selalu berubah mengikuti pergerakan pelanggan. Dengan adanya informasi yang di ambil dari VLR, MSC dapat mengalihkan pesan pendek ke BSS yang berhubungan.

**d. GMSC (*Gateway MSC*)**

*Gateway MSC* (GMSC) merupakan pintu gerbang jaringan bergerak yang dapat berhubungan dengan jaringan lain. Dalam pengiriman pesan, GMSC melakukan pengambilan data dari HLR untuk mengetahui informasi perjalanan pesan ke alamat yang dituju dan melakukan pengiriman pesan ke MSC yang tepat.

**e. SME (*Short Message Entitas*)**

*Short Message Entitas* (SME) merupakan entitas dari pengguna SMS, SME berupa MS atau jaringan tetap yang melakukan aktivitas pengiriman dan penerimaan SMS.

**f. AuC (*Authentication Center*)**

*Authentication Center* (AuC) menyimpan semua informasi yang diperlukan untuk memeriksa keabsahan pelanggan, sehingga usaha untuk mengandakan hubungan pembicaraan dengan pelanggan yang tidak sah dapat dihindarkan. Selain itu AuC juga berfungsi untuk menghindari adanya pihak ketiga yang secara tidak sah mencoba untuk menyadap pembicaraan pelanggan.

**g. EIR (*Equipment Identitas Register*)**

*Equipment Identitas Register* (EIR) adalah basis data yang berisi informasi tentang identitas peralatan bergerak yang *valid*. Setiap peralatan bergerak dilengkapi dengan IMEI (*International Mobile*



*Equipment Identitas*). EIR memuat data-data peralatan pelanggan yang dibagi atas tiga kategori yaitu,

- i. peralatan bergerak bebas mengadakan hubungan pembicaraan ke manapun.
- ii. peralatan bergerak hanya boleh mengadakan hubungan pembicaraan ke tujuan tertentu.
- iii. peralatan bergerak sama sekali tidak diizinkan untuk mengadakan hubungan (komunikasi).

**h. SMSC (SMS Center)**

SMS Center (SMSC) merupakan pengelola dan pentransmisi SMS. SMSC berupa *hardware* dan *software* yang berfungsi sebagai kantor pos dalam mengatur lalu lintas surat. Setiap jaringan GSM memiliki satu atau lebih SMSC untuk menyortir dan melakukan perutean (*routing*) pesan. SMSC bertugas melakukan pengecekan, pengaturan, dan pengiriman pesan kepada operator. Selain itu SMSC dapat juga mengirim dan menerima pesan antar jaringan GSM. Dalam pengiriman pesan terdapat dua metode, yang tergantung dari *interface* yang didukung oleh SMSC dalam jaringan GSM, yaitu secara langsung dan tidak langsung. Pengiriman pesan secara langsung dapat dilakukan oleh terminal yang telah terhubung dengan jaringan GSM melalui modem GSM. Hal ini berbeda dengan pengiriman pesan secara tidak langsung. Pengguna harus melakukan pemanggilan operator terlebih dahulu untuk meminta bantuan pengiriman pesan ke nomor yang dituju.

SMS Center harus mamapu melakukan

- i. Menerima SMS (pesan) dan mengirimkan SMS ke ponsel (*Mobile Station*) secara langsung jika ponsel (nomor) tujuan aktif (telah terdeteksi).
- ii. Menyimpan SMS tersebut sampai *Validity-Period* berakhir jika nomor tujuan tidak terdeteksi (tidak aktif).

- iii. Mengirimkan status pengiriman pesan ke nomor pengirim.

Kebutuhan fungsionalitas dari SMS *Center* tidak akan di bahas secara detail karena hal tersebut berbeda untuk masing-masing operator. Kebutuhan fungsionalitas dari SMS *Center* secara umum adalah:

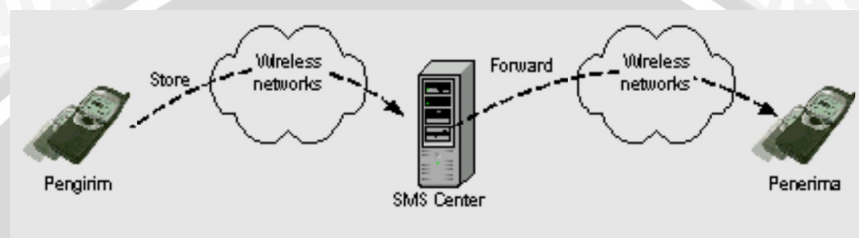
- i. Mengidentifikasi setiap SMS-SUBMIT dan SMS-DELIVER yang dikirim ke ponsel (MS).
- ii. Memberikan *stamp* yaitu waktu kedatangan SMS-SUBMIT di SMS Center dengan ketelitian waktu sampai satuan detik. Jika terdapat dua pesan atau lebih, untuk nomor tujuan yang sama, tiba di SMS Center dengan selisih waktu kedatangan yang kurang dari satu detik, maka SMS Center harus dapat mengatur agar semua pesan untuk nomor tujuan yang sama tapi memiliki waktu *stamp* yang berbeda.
- iii. SMS *Center* hanya mengirimkan satu SMS-DELIVER ke nomor tujuan pada jangka waktu tertentu. Dengan kata lain sebuah ponsel (MS) tidak akan menerima dua pesan atau lebih secara bersamaan dari SMS *Center*.

Hal ini berarti bahwa pengirim dan penerima SMS tidak perlu berada dalam status berhubungan (*connected/ online*) satu sama lain ketika akan saling bertukar pesan SMS. Pengiriman pesan SMS secara *store and forward* berarti pengirim pesan SMS menuliskan pesan dan nomor telepon tujuan dan kemudian mengirimkannya (*store*) ke server SMS (SMS-Center) yang kemudian bertanggung jawab untuk mengirimkan pesan tersebut (*forward*) ke nomor telepon tujuan. Proses ini terlihat pada gambar 2.3 dibawah ini. Hal ini mirip dengan mekanisme *store and forward* pada protocol SMTP yang digunakan dalam pengiriman e-mail internet.

Keuntungan mekanisme *store and forward* pada SMS adalah, penerima tidak perlu dalam status online ketika ada pengirim yang bermaksud mengirimkan



pesan kepadanya, karena pesan akan dikirim oleh pengirim ke SMSC yang kemudian dapat menunggu untuk meneruskan pesan tersebut ke penerima ketika ia siap dan dalam status online di lain waktu. Ketika pesan SMS telah terkirim dan diterima oleh SMSC, pengirim akan menerima pesan singkat (konfirmasi) bahwa pesan telah terkirim (*message sent*). Hal-hal inilah yang menjadi kelebihan SMS dan populer sebagai layanan praktis dari system telekomunikasi bergerak.



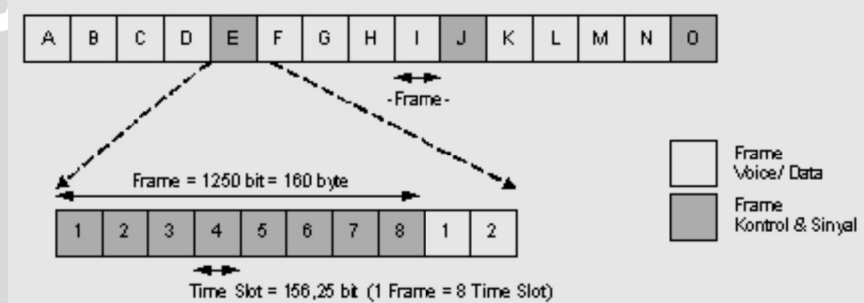
**Gambar 2.3** Mekanisme *store and forward* pada pengiriman pesan SMS  
(sumber *reference system komunikasi seluler*)

Keterbatasan SMS adalah pada ukuran pesan yang dapat dikirimkan, yaitu maksimal sebesar 160 byte. Keterbatasan ini disebabkan karena mekanisme transmisi SMS itu sendiri. SMS pada awalnya adalah layanan yang ditambahkan pada system GSM yang digunakan untuk mengirimkan data mengenai konfigurasi dari handset pelanggan GSM. SMS dikirimkan menggunakan *signalling frame* pada kanal frekuensi atau time slot frame GSM yang biasanya digunakan untuk mengirimkan pesan untuk kontrol dan sinyal setup panggilan telepon, seperti pesan singkat tentang kesibukan jaringan atau pesan CLI (*Caller Line identification*).

Frame ini bersifat khusus dan ada pada setiap panggilan telepon serta tidak dapat digunakan untuk membawa voice atau data dari pelanggan. Ukuran frame pada sistem GSM sendiri adalah sebesar 1250 bit (kurang lebih sama dengan 160 byte) Seperti terlihat pada gambar 2.3. Karena hanya menggunakan satu frame inilah pengiriman pesan SMS menjadi sangat murah, karena beban biaya hanya dihitung dari penggunaan satu frame melalui kanal rekuensi. Pengiriman SMS menggunakan frame pada kanal frekuensi adalah berarti SMS dikirim oleh

pengirim ke nomor telepon tertentu yang bertindak sebagai SMSC (SMS-Center) dan kemudian SMSC bertugas untuk meneruskannya ke penerima.

Pengiriman SMS berlangsung cepat karena, SMSC selain terhubung ke LAN aplikasi juga terhubung ke MSC (*Mobile Switching Network*) melalui SS7 (*Signaling System 7*) yang merupakan jaringan khusus untuk menangkap frame kontrol dan sinyal. Mekanisme pengiriman pesan singkat SMS yang serupa juga ditemukan dalam sistem jaringan lain seperti TDMA, PDC, dan cdmaOne. Beda antara sistem jaringan satu dengan yang lainnya adalah ukuran dari pesan SMS itu sendiri yang bergantung pada ukuran frame yang digunakan pada masing-masing sistem. Pada sistem TDMA dan PDC ukuran pesan SMS sama dengan sistem GSM, yaitu 160 byte, dan pada cdma-One ukuran pesan SMS sebesar 256 byte. Pada gambar 2.4 terlihat struktur time slot dan frame pada system GSM.



**Gambar 2.4** Struktur time slot dan frame pada system GSM  
(sumber reference system komunikasi seluler)

Pada akhirnya SMS menjadi layanan messaging yang populer dan digemari oleh pelanggan telepon seluler. Layanan SMS dapat diintegrasikan dengan layanan GSM yang lain seperti voice, data, dan fax, dan karena itu pesan SMS selain digunakan untuk pengiriman pesan *person to person* juga digunakan untuk notifikasi *voice dan fax mail* yang datang kepada pelanggan. Selain itu SMS juga berharga murah, bersifat simpel dan personal, serta dalam pengoperasiannya tidak terlalu mengganggu kesibukan pemakainya, karena mereka dapat mengirim atau menerima pesan pada waktu yang mereka kehendaki.



## 2.3 Short Message Service (SMS)

### 2.3.1 Pengkonversian 7 bit ( Septet ) ke 8 bit ( Oktet )

SMS yang dispesifikasikan oleh organisasi ETSI ( *European Telecommunication Standart Institute* ) memiliki panjang 160 karakter, bit tiap karakter adalah 7 bit ditunjukkan dalam tabel 7 bit *default* alphabet.

Dalam pengirimannya, SMS 7 bit ( septet ) diubah dalam oktet. Dijelaskan dalam contoh berikut : Pesan " hellohello " terdiri atas 10 karakter, tiap karakter diubah dalam 7 bit, pengubahan pesan ini ditunjukkan dalam tabel.

**Tabel 2.1** Karakter Dalam Septet Untuk Pesan hellohello

Pesan	7 Bit ( Septet )
h	1101000
e	1100101
l	1101100
l	1101100
o	1101111
h	1101000
e	1100101
l	1101100
l	1101100
o	1101111

( Sumber : Trik pemrograman SMS, 2002 : 15 )

Tabel 2.2 Tabel Skema 7 bit

b7	0	0	0	0	1	1	1	1					
b6	0	0	1	1	0	0	1	1					
b5	0	1	0	1	0	1	0	1					
b4	b3	b2	b1		0	1	2	3	4	5	6	7	
0	0	0	0	0	@	Δ	SP	0	-	P	..	p	
0	0	0	1	1	!	1	A	Q	a	q			
0	0	1	0	2	ς	Φ	"	2	B	R	b	r	
0	0	1	1	3		Γ	#	3	C	S	c	s	
0	1	0	0	4		Λ		4	D	T	d	t	
0	1	0	1	5		Ω	%	5	E	U	e	u	
0	1	1	0	6		Π	&	6	F	V	f	v	
0	1	1	1	7		Ψ		7	G	W	g	w	
1	0	0	0	8		Σ	(	8	H	X	h	x	
1	0	0	1	9		Θ	)	9	I	Y	i	y	
1	0	1	0	10	LF	Ξ	*	:	J	Z	j	z	
1	0	1	1	11			+	;	K	Ä	k	ä	
1	1	0	0	12			,	<	L	Ö	l	ö	
1	1	0	1	13	CR		-	=	M		m		
1	1	1	0	14		β	.	>	N	Ü	n	ü	
1	1	1	1	15			/	?	O		o		

( Sumber : Trik pemrograman SMS, 2002 : 15 )

Septet pertama ( h ) diubah dalam sebuah oktet dengan menambah satu bit paling kanan dari septet kedua agar jumlah bit menjadi 8. Bit ini dimasukkan pada bagian paling kiri dari 7 bit karakter 'h' sehingga 1 + 1101000 = 11101000 ( "E8" ) menjadi 8 bit. Bit paling kanan dari karakter kedua telah terpakai sehingga karakter kedua memerlukan 2 bit dari karakter



ketiga untuk membuat 8 bit. Demikian seterusnya hingga karakter terakhir, sehingga diperoleh format 8 bit untuk tiap karakter yang ditunjukkan dalam tabel. ( *Sumber : Trik pemrograman SMS, 2002 : 15* )

**Tabel 2.3** Hasil Pengubahan 7 bit default ke 8 bit

8 Bit Oktet	Heksadesimal
11101000	E8
00110010	32
10011011	9B
11111101	FD
01000110	46
10010111	97
11011001	D9
11101100	EC
00110111	37

( *Sumber : trik Pemrograman SMS, 2002 : 15* )

Jadi, hellohello akan dikirimkan dalam heksadesimal sebagai berikut :  
E8329BFD4697DEC37

### 2.3.2 PDU (Protocol Data Unit) untuk Kirim SMS ke SMS-Centre

#### Delapan header untuk kirim SMS.

PDU untuk mengirim SMS terdiri atas delapan header, sebagai berikut :

#### 1) Nomor SMS-Centre

Header pertama ini terbagi atas tiga subheader, yaitu :

- a. Jumlah pasangan heksadesimal SMS-Centre dalam bilangan heksa.
- b. National/International Code.
  - Untuk National, kode subheader-nya yaitu 81.
  - Untuk International, kode subheader-nya yaitu 91.
- c. No SMS-Centre-nya sendiri, dalam pasangan heksa dibalik-balik.

Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya.

Contoh : untuk nomor SMS-Centre Exelcom dapat ditulis dengan dua cara sebagai berikut : ( Sumber : trik Pemrograman SMS, 2002 : 9 )

### Cara 1 :

085258115588

- a. 06 → ada 6 pasang
- b. 81 → 1 pasang
- c. 80-81-44-05-90 → 5 pasang
- Total : 6 pasang

Digabung menjadi : **0685258115588**

### Cara 2 :

62818445009 diubah menjadi :

- a. 07 → ada 7 pasang
- b. 91 → 1 pasang
- c. 26-18-48-54-00-F9 → 6 pasang
- Total : 7 pasang

Digabung menjadi : **07912618485400F9**

## 2) Tipe SMS

Untuk SEND tipe SMS = 1. jadi bilangan heksanya adalah **01**.

## 3) Nomor Referensi SMS

Nomor referensi ini dibierkan dulu 0, jadi bilangan heksanya adalah **00**.

Nanti akan diberikan sebuah nomor referensi otomatis oleh ponsel/alat

SMS-gateway.



#### 4) Nomor Ponsel Penerima

Sama seperti cara menulis PDU *Header* untuk SMS-Center, *header* ini juga terbagi atas tiga bagian sebagai berikut :

- a. Jumlah bilangan decimal nomor ponsel yang dituju dalam bilangan heksa.
- b. National/International Code
  - Untuk National, kode subheade-nya : 81.
  - Untuk International, kode subheader-nya : 91.
  - Nomor ponsel yang dituju dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka trsebut dipasangkan dengan huruf F di depannya.

Contoh :

Untuk nomor ponsel yang dituju = 6285736061407 dapat ditulis dengan dua cara sebagai berikut :

**Cara 1** :085736061407 diubah menjadi :

- a. 0C → ada 12 angka
- b. 81
- c. 80-75-63-60-41-F7

Digabung menjadi : **0C818075636041F7**

**Cara 2** :6285736061407 diubah menjadi :

- a. 0D → ad13 angka
- b. 91
- c. 26-08-75-63-60-41-70

Digabung menjadi : **0D26087563604170**

5) **Bentuk SMS**, antara lain :

0 → 00 → dikirim sebagai SMS.

1 → 01 → dikirim sebagai telex.

2 → 02 → dikirim sebagai fax.

Dalam hal ini, untuk mengirim dalam bentuk SMS tentu saja memakai 00.

6) **Skema Encoding Data I/O**

Ada dua skema, yaitu :

a. Skema 7 bit → ditandai dengan angka 0 → 00.

b. Skema 8 bit → ditandai dengan angka lebih besar dari 0 → diubah ke heksa.

Kebanyakan ponsel/SMS Gateway yang ada di pasaran sekarang menggunakan skema 7 bit, sehingga menggunakan kode 00.

7) **Jangka Waktu Sebelum SMS Expired**

Jika bagian ini di-skip, itu berarti tidak membatasi waktu berlakunya SMS. Sedangkan jika diisi dengan suatu bilangan interger yang kemudian diubah ke pasangan heksa tertentu, bilangan yang kita berikan tersebut akan mewakili jumlah waktu validitas SMS tersebut.

Rumus untuk menghitung jangka waktu validitas SMS dapat dilihat pada table di bawah ini :

**Tabel 2.4** Tabel Rumus Perhitungan Jangka Waktu Validitas

Interger (INT)	Jangka Waktu Validitas SMS
0-143	(INT+1) x 5 menit (berarti : 5 menit s/d 12 jam)
144-167	12 jam + ((INT – 143) x 30 menit)



168-196	(INT -166) x 1 hari
197-255	(INT – 192) x 1 minggu

(Sumber : *Trik Pemrograman SMS, 2002 : 13*)

Agar SMS pasti terkirim sampai ke ponsel penerima, sebaiknya tidak memberika batasan waktu validitasnya.

### 8) Isi SMS

*Header* ini terdiri dari atas dua subheader, yaitu :

- a. Panjang isi (jumlah huruf dari isi)

Misalnya : untuk kata “hello” → ada 5 huruf → 05.

- b. Isi berupa pasangan bilangan heksa

Untuk ponsel/SMS *Gateway* berskema encoding 7 bit, jika mengetikkan suatu huruf dari keypad-nya, berarti telah membuat 7 angka I/O secara berurutan. PDU (Potocol Data Unit) untuk terima SMS dari SMS-Centre Delapan *header* untuk terima SMS.

*Header* untuk terima SMS hampir sama dengan *header* pada kirim SMS kecuali beberapa yang berbeda dijelaskan di bawah ini. (Sumber : trik Pemrograman SMS, 2002 :17)

- a. No SMS-Center (sama dengan format PDU pada kirim SMS).
- b. Tipe SMS.

Untuk terima SMS menggunakan kode 04.

- c. No ponsel pengirim (sama dengan format PDU pada kirim SMS).
- d. Bentuk SMS (sama dengan format PDU pada kirim SMS).
- e. Skema Encoding (sama dengan format PDU pada kirim SMS).
- f. Tanggal dan waktu SMS di SMS-Centre.

Diwakili oleh 12 bilangan heksa (6 pasangan) yang berarti :yy/mm/dd hh:mm:ss.

Contoh : 207022512380 → 02/07/22 15:32:08 → 22 juli 2002 15:32:08 WIB.

- g. Batas waktu validitas, jika tidak dibatasi ditulis dengan kode **00**.
- h. Isi SMS (diubah dari 7 bit menjadi 8 bit).

### 2.4 AT Command

AT Command bertugas mengirim dan menerima data dari dan menuju SMS. AT Command yang penting dalam sistem SMS adalah :

- **AT+CMGS** ( untuk mengirim SMS )

$AT+CMGS = n$  ( untuk mengirim SMS )

$n$  = jumlah pasangan heksa PDU SMS dimulai setelah nomor SMS *center* ( maksimal 140 )

**Tabel 2.5** Parameter command syntax AT+CMGS

Command	Respon
Jika mode PDU(+CMGF=0): +CMGS=<length><CR>PDU yang ada <ctrl Z/ESC>	Sukses:+CMGS:<mr> Gagal:+CMS:<error>
+CMGS=?S	

( Sumber :*Trik Pemrograman SMS, 2002 : 15* )

- **AT+CMGL** ( untuk memeriksa SMS )

$AT+CMGL = n?$  ( untuk memeriksa SMS ) antara lain :

- ❖  $n = 0$  untuk SMS baru di INBOX ( pesan belum terbaca )
- ❖  $n = 1$  untuk SMS lama di INBOX ( pesan telah terbaca )
- ❖  $n = 2$  untuk SMS Unset di OUTBOX ( pesan tidak terkirim )
- ❖  $n = 3$  untuk SMS sent di OUTBOX ( pesan terkirim )



- ❖ n = 4 untuk semua SMS

**Tabel 2.6** Parameter command syntax AT+CMGL

Command	Respon
+CMGL=[<stat at>]	Jika mode PDU(+CMGF=0) dan command sukses : +CMGL:<index>,<stat>,<length><CR><LF><pdu>[<CR><LF>] +CMGL:<index>,<stat>,<length><CR><LF><pdu>[...] Jika gagal :+CMS ERROR:<err>
+CMGL=?	+CMGL: ( status yang didukung <stat>s)

( Sumber : Trik Pemrograman SMS, 2002 :16 )

- **AT+CMGR** ( untuk menerima SMS )

**Tabel 2.7** Parameter command syntax AT+CMGR

Command	Respon
CMGR<index>	Sukses:CMGR:<stat>,<length><CR><LF><pdu> Gagal:+CMS:<error>
+CMGR=?	OK

( Sumber :Trik Pemrograman SMS, 2002 : 17)

- **AT+CMGD** ( untuk menghapus SMS )

n = nomor referensi SMS yang ingin dihapus.

**Tabel 2.8** Parameter command syntax AT+CMGD

Command	Respon
+CMGD = <index>	+CMS : <error>
+CMGD = ?	

( Sumber : Trik Pemrograman SMS, 2002 : 18)

- **AT+CMGF** ( untuk SMS format )

**Tabel 2.9** Parameter command syntax AT+CMGF

Command	Respon

+CMGF = [<mode>]	
+CMGF =?	+CMGF:<mode>
+CMGF=?	+CMGF: ( pilihan yang mendukung <mode>s)

( Sumber :*Trik Pemrograman SMS, 2002 : 19*)

### 2.5. Smartcard

Untuk jenis kartu yang digunakan dalam penyusunan skripsi ini adalah type ACR 30 dengan kapasitas memory kartu 1 Kilobyte.

Adapun perintah – perintah yang dikeluarkan reader dalam mengakses kartu ini yaitu:

#### 1. Reset

Di dalam melaksanakan reset pada saat kartu dimasukkan ke dalam reader harus dapat dilakukan dengan cara mengikuti format berikut :

Format perintah

**Tabel 2.10.** Format Reset.

Instruction Code	Data length
80 H	00 H

(*Sumber:Reference Manual ACR30,2005:19* )

#### 2. Power\_Off

Dalam mengaktifkan reader pada saat kartu dimasukkan dapat menggunakan format sebagai berikut :

Format perintah

**Tabel 2.11** Format Power\_of.

Instruction Code	Data length
81 H	00 H

(*Sumber:Reference Manual ACR30,2005:21* )





3. Read\_data

Pembacaan kartu dilakukan dengan jalan melakukan perintah pada saat kartu dimasukkan yaitu :

Format perintah

**Tabel 2.12** Format Read\_Data

Instruction Code	Data length	Data	
		ADDR	LEN
90 <sub>H</sub>	03 <sub>H</sub>		

(Sumber:Reference Manual ACR30,2005:21 )

ADDR : Alamat byte pada saat pembacaan kartu.

LEN : Jumlah data N yang dibaca dari kartu ( $0 < N \leq \text{MAX\_R}$ )

**2.6 Smartcard Reader**

Smartcard adalah sebuah kartu yang memiliki memory yang dapat menyimpan data-data. Kartu chip secara umum di golongan menjadi dua macam berdasarkan penggunaannya yaitu :

- Kartu chip yang hanya berupa *memory*.
- Kartu chip yang memiliki *memory (ROM, RAM, EEPROM)* sekaligus *microprocessor*.

Untuk jenis kartu chip pertama lebih sering digunakan sebagai kartu telpon, kartu internet dan fasilitas – fasilitas Prabayar yang tidak memerlukan informasi pengguna kartu. Kartu jenis ini fungsinya lebih ditekankan pada penyimpanan sejumlah kredit atau poin untuk transaksi-transaksi, contohnya pada kartu telpon chip. Pada beberapa jenis kartu yang baru telah dilengkapi dengan enkripsi atau pin identifikasi yang seperti ini biasa disebut dengan *smart memorycard*.



Sedangkan untuk jenis kartu chip kedua adalah kartu yang lebih canggih yang dilengkapi mikroprocessor untuk fungsi-fungsi tertentu, selain menyimpan identitas pemilik kartu juga dapat memproses sejumlah data. Kartu jenis ini juga memiliki memory berupa ROM, RAM dan EEPROM, data dilindungi dengan algoritma enkripsi. Jenis ini umum disebut dengan *smartcard* atau kartu cerdas (walau sebenarnya *memorycard* adalah juga merupakan *smartcard*).

Spesifikasi kartu telpon chip generasi pertama :

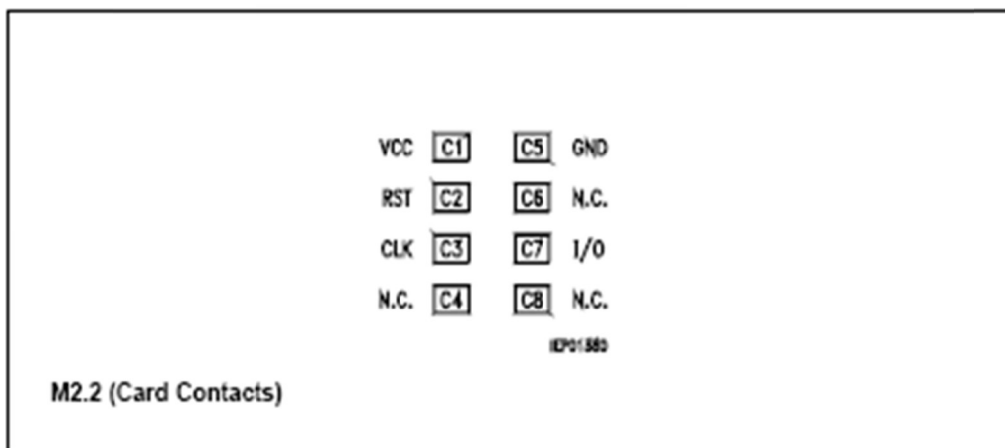
- Synchronous protocol
- Teknologi N-Mos atau CMOS untuk yang lebih baru.
- Organisasi memory 256 x 1 bit.
- 96 bit proteksi penulisan dengan clock out fuse.
- Pemakaian daya rendah 85 mW pada read mode.
- 21 volt programming voltage.
- Acces time 500 ms.
- Operating temperature range – 10°C sampai +70°C.
- Deteksi data hingga 10 thn.

Sedangkan untuk generasi kedua adalah

- ISO 7816 – ½ compatible protocol.
- Penggunaan tegangan supply tunggal 5 volt.
- Teknologi NMOS.
- Konsumsi daya rendah.

### 2.6.1 Pin pada Smartcard.

Posisi pin pada smartcard menggunakan standart ISO7816 dimana letaknya akan diperlihatkan pada gambar berikut :



**Gambar 2.5** Konfigurasi Pin Smartcard.

(Sumber : Datasheet SIEMENS SLE 4432,1995:6 )

**Tabel 2.13** Konfigurasi Pin Smartcard.

Nama	Deskripsi
Vcc	+ 5 volt
R/W	Read / write
CLK	Clock
GND	Ground
*	NC
I/O	Input / Output
*	NC

(Sumber: Datasheet SIEMENS SLE 4432,1995:6 )

### 2.6.2 System komunikasi Data Smartcard.

Komunikasi antara *Smart card* dengan interface device (reader) melalui beberapa proses yaitu :



- Kontak dan Aktivasi oleh reader.
- Reset dari smartcard.
- Answer to Reset (ATR) oleh *smart card*.
- Pertukaran data antara *smart card* dan reader.
- Deaktivasi kontak oleh reader.

#### 2.6.2.1 kontak dan aktivasi oleh reader.

Untuk menghindari kerusakan yang mungkin terjadi pada kartu yang disebabkan oleh reader pada saat kartu dimasukan maka sirkuit diharapkan tidak diaktifkan dahulu sebelum pin - pin pada kartu benar – benar terhubung dengan readernya. Adapun didalam mengaktifkan reader harus memperhatikan beberapa hal berikut yaitu :

- Reset berada pada status “low”.
- Vcc telah diberikan tegangan tertentu.
- I/O pada reader berada pada reception mode.
- CLK harus diberikan clock yang sesuai dan stabil.

#### 2.6.2.2 Reset Oleh Smartcard.

Setiap card reset dikenali oleh reader, dimana kartu harus merespon dengan ATR (Answer To Reset) seperti yang dijelaskan pada bagian berikut

Pada saat akhir aktivasi kontak oleh reader (RST berada pada state “low”, Vcc telah diberi tegangan tertentu, I/O pada reader berada pada *reception mode*, Vpp harus dinaikan pada *idle state*, CLK harus diberi clock yang sesuai dan stabil , maka kartu akan merespon secara asinkron bahwa card siap di reset.

Jika kartu merespon secara sinkron, seperti pada kartu kredit misalnya maka semua pin berada pada posisi “low” Vcc telah diberikan pada tegangan tertentu, Vpp diset pada *idle state*, CLK dan RST berada tetap pada “Low”, I/O

berada pada *reception mode*, Reset harus berada pada kondisi “High” sekurang – kurangnya 50  $\mu$ s sebelum kembali ke kondisi “Low”.

### 2.6.2.3 Answer To Reset (ATR).

Terdapat dua buah tipe transmisi pengiriman data pada saat answer to reset yaitu :

- Pengiriman secara asinkron
- Karakter yang dikirim melalui I/O (half duplex) secara asinkron dimana setiap karakter berupa Byte (8 bit).
- Pengiriman secara Sinkron.

Sekelompok bit yang dikirim secara half duplex melalui clock pada CLK.

Smartcard digunakan sebagai kartu GSM atau kartu telpon biasanya megunakan transmisi asinkron, sedangkan transmisi sinkron biasanya digunakan pada kartu kredit.

### 2.6.2.4 Deaktivasi Kontak oleh Reader.

Yaitu Jika pertukaran data putus atau dibatalkan yang disebabkan semisalnya oleh pengambilan kartu atau kartu yang tidak merespon maka kontak pada pin harus dinonaktifkan, adapun proses tersebut meliputi :

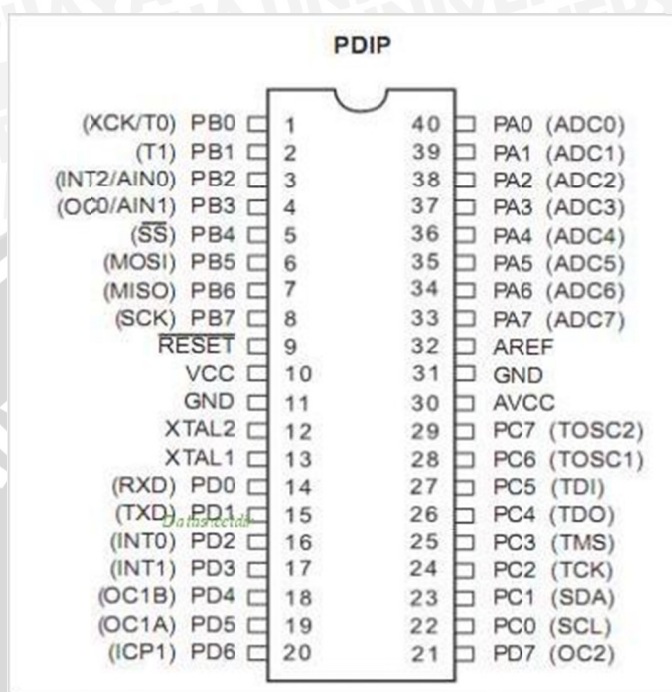
- RST pada kondisi “Low”.
- CLK pada Kondisi “Low”.
- Vpp tidak diberi tegangan.
- I/O berada pada kondisi “A”
- Vcc tidak diberi tegangan.

## 2.7 Mikrokontroler ATmega 32

Mikrokontroler ATMEGA32 adalah mikrokontroler yang diproduksi oleh Atmel. mikrokontroler ini memiliki clock dan kerjanya tinggi sampai 16 MHz,



ukuran flash memorinya cukup besar, kapasistas SRAM sebesar 2 KiloByte, 32 buah port I/O yang sangat memadai untuk berinteraksi dengan LCD dan keypad.



**Gambar 2.6** Pinout ATmega 32

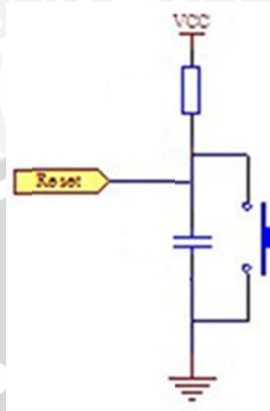
( Sumber : Datasheet ATmega 32 )

### 2.7.1 Program Memori Mikrokontroler ATmega 32

ATMEGA 32 memiliki 32 KiloByte flash memori untuk menyimpan program. Karena lebar intruksi 16 bit atau 32 bit maka flash memori dibuat berukuran 16K x 16. Artinya ada 16K alamat di flash memori yang bisa dipakai dimulai dari alamat 0 heksa sampai alamat 3FFF heksa dan setiap alamatnya menyimpan 16 bit instruksi.



## 2.8 Rangkaian Reset



**Gambar 2.7 Rangkaian Reset**

(Sumber : rangkaian dasar elektronika 2002)

Untuk me-*reset* mikrokontroler ATmega 32, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset* kapasitor dihubungkan dengan  $V_{CC}$  dan sebuah resistor yang dihubungkan ke *ground*.

Karena kristal yang digunakan mempunyai frekuensi sebesar 12 MHz, maka satu periode dapat dihitung dari persamaan :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{12MHz} s$$

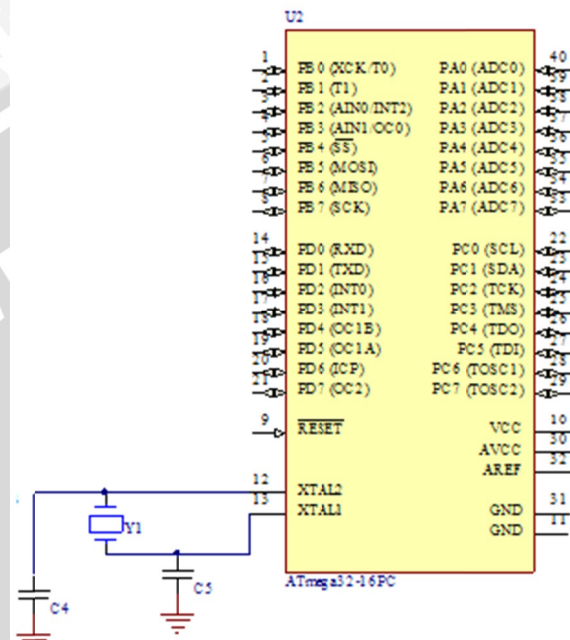
Untuk mendapatkan waktu minimal logika tinggi yang dibutuhkan untuk *mereset* mikrokontroler dapat dihitung menggunakan persamaan :

$$t_{reset(min)} = T \times \text{periode yang dibutuhkan}$$

Setelah mendapatkan Waktu minimal untuk mereset mikrokontroler maka menentukan nilai R dan C. Dari perhitungan dengan menggunakan persamaan :

$$f = \frac{1}{2\pi RC}$$

## 2.9 Rangkaian Clock



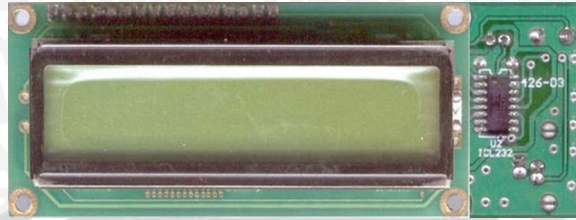
Gambar 2.8 Rangkaian Reset

(Sumber : datasheet ATmega 32)

Kecepatan proses yang diperlukan oleh mikrokontroler ATmega 32 ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Mikrokontroler ATmega 32 memiliki internal *clock* generator yang berfungsi sebagai sumber *clock* yang diperlukan. Untuk kristal *clock* dipasang Kristal dan resonator keramik yang berfungsi sebagai pembangkit *clock* osilator yang ada pada mikrokontroler.

Rangkaian ini terdiri dari dua buah kapasitor dan sebuah kristal. Untuk mengendalikan frekuensi osilatornya cukup dengan menghubungkan Kristal pada pin XTAL1 dan pin XTAL2 serta dua buah kapasitor ke *ground*. Dalam minimum kristal ini, menggunakan kristal 12 Mhz dan  $C_1 = C_2$  yaitu sebesar 33 pF.

### 2.10 Liquid Crystal Display (LCD)



**Gambar 2.9** LCD (*Liquid Central Display*)

(Sumber : perencanaan)

LCD yang digunakan merupakan LCD tipe karakter karena LCD ini dapat menampilkan data berupa karakter.

Keuntungan yang dapat diperoleh dengan menggunakan LCD adalah :

1. Dapat menampilkan karakter ASCII, sehingga memudahkan untuk membuat program tampilannya.
2. Mudah dihubungkan dengan port I/O karena hanya menggunakan 8 bit data dan 3 bit control.

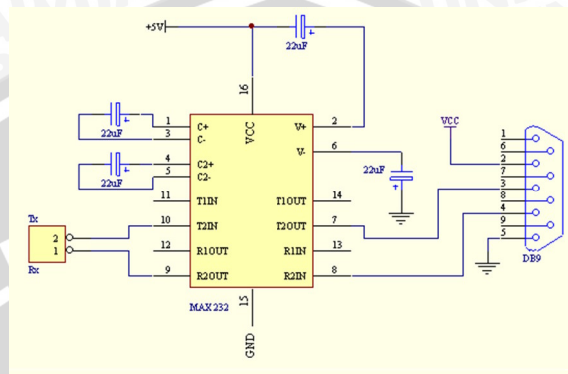
LCD yang digunakan dalam skripsi ini merupakan tipe karakter 16x2 baris, dan dapat menampilkan 16 karakter perbaris dan mempunyai 2 baris.

### 2.11 Rangkaian Interface

IC max 232 ini banyak digunakan untuk merubah level tegangan TTL (0 atau 5 Volt). Di dalam IC 232 mempunyai 2 *transmitter* (Tx) dan 2 *receiver* (Rx), tetapi yang digunakan dalam perancangan adalah 2 *transmitter* dan 1 *receiver* yang digunakan untuk merubah tegangan. IC ini meskipun hanya menggunakan catu daya +5 Volt, tetapi sanggup melayani level tegangan rs 232 yang -12 Volt sampai +12 Volt.



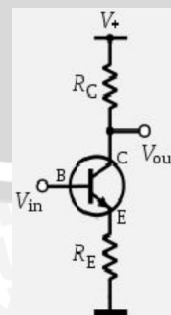
Pada IC max 232 terdapat 4 kapasitor yaitu C1, C2, C3, dan C4 yang masing-masing besarnya adalah 1µf. C1 dan C2 berguna untuk mengurangi impedansi output, sedangkan C3 dan C4 berfungsi untuk terjadinya ripple pada tegangan input.



**Gambar 2.10** Rangkaian RS 232

(Sumber : perencanaan)

Secara teoritis, sebuah motor AC tidak dapat digerakkan langsung oleh mikrokontroller. Arus dan tegangan yang dikeluarkan oleh mikrokontroller terlalu kecil untuk menggerakkan sebuah motor AC. Gerbang-gerbang Transistor Transistor Logic (TTL) mikrokontroller hanya mampu mengeluarkan arus dalam orde mili-ampere dan tegangan antara 2 sampai 2,5 Volt. Maka digunakan sebuah piranti tambahan yang memenuhi kebutuhan arus dan tegangan yang cukup besar. Rangkaian driver motor dengan komponen utama transistor, optocoupler dan Relay. Relay digunakan untuk menyalakan dan mematikan motor AC yang diaktifkan oleh transistor.



**Gambar 2.11** Rangkaian Common Emitter

(Sumber : Rangkaian Dasar Elektronika 2002)

Untuk mencari nilai  $V_{out}$ ,  $R_c$  dan  $R_e$  dapat dilakukan dengan menggunakan formula dibawah ini :

$$V_{out} = V_{in} \times A_v$$

$$R_c = R_{out} = \frac{V_{out}}{I_{out}}$$

$$R_e = \frac{R_c}{A_v}$$

### 2.11.1 Komunikasi Serial RS 232

RS 232 merupakan salah satu jenis *interface* dalam proses transfer data antar komputer dalam bentuk serial transfer. RS 232 merupakan kependekan dari **Recommended Standart Number 232**. RS 232 dibuat untuk *interface* antara peralatan terminal data dengan peralatan komunikasi data, dengan menggunakan data biner serial sebagai data yang ditransmisikan. Serial interface RS 232 memberi ketentuan *logic level* sebagai berikut :

- *Logic 1* disebut " mark " terletak antara -3 Volt hingga -15 Volt.
- *Logic 0* disebut " space " terletak antara +3 Volt hingga +15 Volt.

Daerah tegangan antara -3 Volt hingga +3 Volt adalah *invalid level*, yaitu daerah yang tidak memiliki *logic* sehingga daerah tersebut harus dihindari. Suatu saluran data RS 232 yang memberi keadaan ini berarti ada kesalahan. Demikian pula saluran daerah lebih negatif dari -15 volt dan daerah lebih positif dari +15 Volt. Pada saat pengiriman data, sebelum mengirim data bit 0 harus diawali dengan *start* bit terlebih dahulu kemudian baru mengirim bit 0. Setelah mengirim bit 7 masih harus diakhiri dengan stop bit.

### 2.11.2 Dasar-Dasar Serial Interface

Dasar-dasar serial *interface* telah diuraikan mengenai fungsi dari *interface* RS 232. Dalam pembahasan berikut ini, akan dijelaskan bagaimana dasar-dasar *interface* tersebut. Proses transfer secara serial menggunakan alat RS 232 yang dibuat oleh *Electronic Industry Assosiation* ( EIA ) antara

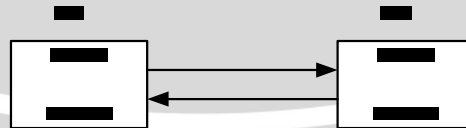
terminalnya, biasanya menggunakan DTE (*Data Terminal Equipment*) untuk masing-masing terminal. Kadang diperlukan seperangkat peralatan untuk kebutuhan komunikasi yang lebih kompleks, misalnya dengan memanfaatkan modem. Perangkat tersebut sering disebut dengan DCE (*Data Communication Equipment*).

Data yang ditransfer dari suatu terminal akan diterima oleh terminal lainnya, dan demikian pula sebaliknya melalui seperangkat peralatan di atas. Gambar di bawah menjelaskan konsep transfer antara DTE dengan DTE dan DTE dengan DCE. Jenis data yang akan ditransfer adalah dalam bentuk biner ( bit per bit transfer ) dengan satuan baut untuk kecepatannya ( bit per detik ). Dalam proses transfer ini harus terdapat suatu peralatan yang berfungsi sebagai *hand shake* ( jabat tangan ) yaitu sebagai pemantau status yang diterima atau ada untuk memberikan respon yng sesuai. Dalam merancang perangkat lunak komunikasi serial, *hand shake* disempurnakan dengan menambahkan karakter pengendali dalam deretan atau jumlah bit yang ditransfer, biasanya disebut *start bit* dan *stop bit*.



**Gambar 2.12** Blok Diagram Transfer Data DTE dengan DTE

(Sumber : Anonim1)



**Gambar 2.13** Blok Diagram Transfer Data DTE dengan DCE

(Sumber : Anonim2)



Secara sederhana dapat dijelaskan bagaimana konsep *interface* antara DTE dengan DCE yang dilakukan berulang-ulang sampai semua data selesai ditransfer, adalah sebagai berikut :

- Ketika DTE ingin mengirim data, sebuah protokol yaitu RTS dikirimkan untuk memberitahu DCE.
- Pada saat itu masukan RTS pada DCE menjadi aktif.
- Jika DCE mampu menerima balasan data, maka ia akan membalasnya dengan mengirim CTS.
- Begitu DTE menerima balasan, masukan CTS-nya diaktifkan.
- Pengiriman data dilakukan melalui TxD.
- Penerimaan data dilakukan melalui RxD.

### 2.11.3 Pin - Pin pada EIA RS 232 dan Kegunaannya

Secara praktis untuk kebutuhan transfer data cukup 9 pin yang digunakan. Adapun masing-masing adalah sebagai berikut :

- a) Pin 1 Protective Ground  
Pin ini berguna untuk menghindari kejutan karakteristik listrik karena kegagalan suatu daya. Dalam kasus bagaimanapun, pada standart RS 232 pin 1 bukan merupakan suatu keharusan.
- b) Pin 2 Transmitted Data ( TxD / TD )  
Berguna sebagai jalur pengiriman data dari DTE ke DCE.
- c) Pin 3 Received Data ( RxD / RD )  
Berguna sebagai jalur penerimaan data dari DCE ke DTE.
- d) Pin 4 Request to Send ( RTS )  
Berguna untuk memberitahu DCE bahwa DTE akan mengirim data. RTS merupakan sebuah protokol perangkat keras yang mendahului pengiriman data dari DTE ke DCE.
- e) Pin 5 Clear to Send ( CTS )

Berguna untuk memberitahu DTE bahwa DCE siap untuk menerima data. CTS merupakan sebuah protokol perangkat keras yang mendahului penerimaan data dari DTE ke DCE.

f) Pin 6 Data Set Ready ( DSR )

Berguna untuk memberitahu DTE bahwa DCE aktif dan siap untuk bekerja.

g) Pin 7 Signal Ground

Berguna sebagai referensi semua tegangan *interface*.

h) Pin 8 Data Carrier Detect ( DCD )

Berguna pada DTE untuk tidak memperbolehkan penerimaan data.

i) Pin 9 Data Terminal Ready ( DTR )

Berguna untuk memberitahu DCE bahwa DTE aktif dan siap untuk bekerja.

**Tabel 2.15** Konfigurasi Pin DB 9

Pin No	Name	Notes / Description
1	DCD	Data Carrier Detect
2	RD	Receive Data
3	TD	Transmit Data
4	DTR	Data Terminal Ready
5	SGND	Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

(Sumber : Datasheet DB9,2005: 12)

### 2.11.4 Protokol Komunikasi pada RS 232

Beberapa protokol dalam *interface* RS 232 adalah :

- **Start Bit**

Merupakan sebuah bit dengan logika '0', dimana bit ini yang menandakan bahwa akan ada karakter atau data yang mengikutinya. Bit ini langsung diberikan tanpa harus mensetnya terlebih dahulu.

- **Data Bit**

Merupakan bit yang mewakili dari karakter yang diikutinya. Data bit ini dapat diset antara 5 sampai 8 bit.

- **Parity Bit**

Merupakan bit yang digunakan sebagai *error checking* pada *receiver*, apabila terjadi kesalahan maka *receiver* akan menset *error flag* ( *parity error* ) pada spesial register. *Parity bit* ini menghitung jumlah data yang berlogika '1' pada data bit. Perhitungan jumlah data bit tersebut tergantung dari jenis *parity* yang diset. Untuk *parity 'event'* jumlah data bit yang berlogika '1' ditambah dengan *parity bit* akan menghasilkan jumlah yang ganjil. Sedangkan untuk *parity 'mark'* merupakan *parity bit* yang selalu berlogika '1' demikian pula pada *space parity bit* selalu berlogika '0' dan *parity 'none'* merupakan *parity bit* yang diabaikan.

- **Stop Bit**

Merupakan bit yang menandakan akhir dari suatu paket data ( biasanya 1 byte data ). Seperti pada start bit, bit langsung diberikan pada serial *device*. *Stop bit* ini dapat diset panjangnya menjadi satu bit, satu setengah atau dua bit.

- **Baud Rate**

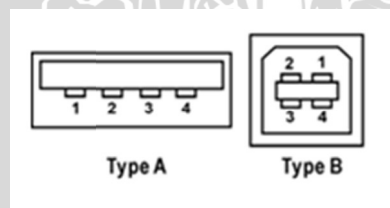
Sebenarnya baud rate berarti pergantian kondisi tiap detik, tetapi karena hanya ada dua kondisi pada komunikasi serial ( yaitu *logic '1'* dan *'0'* )



maka dapat juga digunakan untuk menunjukkan kecepatan dari transmisi bit ( *bit persecond* ).

## 2.12 Konektor USB

Di dalam menghubungkan modul COM-USB dengan komputer maka dibutuhkan kabel konektor USB dimana hanya terdapat ada 2 macam konektor yang digunakan dalam menghubungkan modul tersebut, yaitu konektor type A dan konektor type B seperti terlihat dalam *Gambar 2.12*. Konektor type A dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada bagian computer sedangkan Konektor type B dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada modul COM-USB sedangkan untuk peralatan USB yang sederhana, misalnya mouse, biasanya tidak pakai konektor B melainkan konektor tipe A hal ini dilakukan untuk menghemat biaya sehingga kabel langsung dihubungkan ke bagian dalam mouse.



**Gambar 2.12** Konektor USB

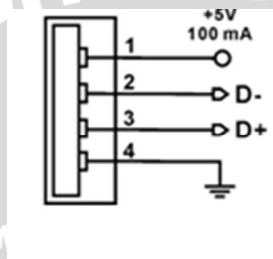
(sumber: *Serial DB 9: Joakim Ogren*)

Kabel USB terdiri dari 4 kabel ditambah konduktor pembungkus kabel seperti pelindung yang biasanya dijumpai dalam kabel audio.

Kabel nomor 1 dipakai untuk menyalurkan sumber daya dengan tegangan 5 Volt, jika diperlukan peralatan USB boleh mengambil daya dari saluran ini tidak lebih dari 100 mA. Komputer yang dilengkapi dengan kemampuan USB, wajib

menyediakan daya sebesar 500 mA untuk keperluan ini. Peralatan USB yang memerlukan daya lebih dari ketentuan tersebut di atas, harus menyediakan sendiri sumber daya untuk keperluan kerja peralatan tersebut.

Kabel nomor 4 adalah ground sebagai saluran balik sumber tegangan 5 Volt. Kabel nomor 2 dan nomor 3 dipakai untuk pengiriman sinyal. Kabel nomor 2 bernama D- dan kabel nomor 3 bernama D+, tegangan pada dua saluran ini berubah antara 0 Volt dan 3,3 Volt seperti terlihat pada *gambar 2.8* berikut:



**Gambar 2.13** Konektor Pin USB Tipe A

(sumber: *Universal Serial Bus (USB)* :Joakim Ogren)

### 2.13 Handphone Siemens C 45



**Gambar 2.14** Bentuk fisik dari Handphone Siemens M35

( *Sumber: Teori Dasar* )

Siemens C 45 boleh jadi merupakan salah satu produk sukses dari Siemens Divisi *Mobile*. Pada jamannya produk ini banyak yang mencari. Salah satu keberanian Siemens dengan produk ini adalah desainnya yang mungil. Fasilitasnya pun pada masa itu sudah dianggap cukup bagus. Selain fasilitas standard berupa SMS dan telepon terdapat juga fasilitas alarm sebagai pengingat ulang tahun atau *meeting*, *calculator*, *stop watch* dan juga *chronometer*.

*Phonebook*-nya hanya 50, tapi kita bisa menambahkan gambar tertentu pada nomor telepon seseorang. Gamenya ada empat yaitu Wayout, Reversi, Quattropoli, dan Minesweper. Dan yang menarik Siemens M 35 ini sudah mendukung internet.

Berikut ini adalah gambar kabel data Siemens M 35 dan konektor DB 9 dan konektor ke pin out HP.



**Gambar 2.15** Konektor DB9 dan konektor ke pinout HP

(Sumber: Teori Dasar )

Pin-pin yang terhubung antara *handphone* Siemens M35 dan PC akan diperlihatkan pada tabel 2.16



**Tabel 2.16** Pin-pin yang terhubung

	PC	HP48	
Shield	SHIELD	1	Shield
RxD	2	2	RxD
TxD	3	3	TxD
GND	5	4	GND

( Sumber: Serial DB 9: Joakim Ogren)

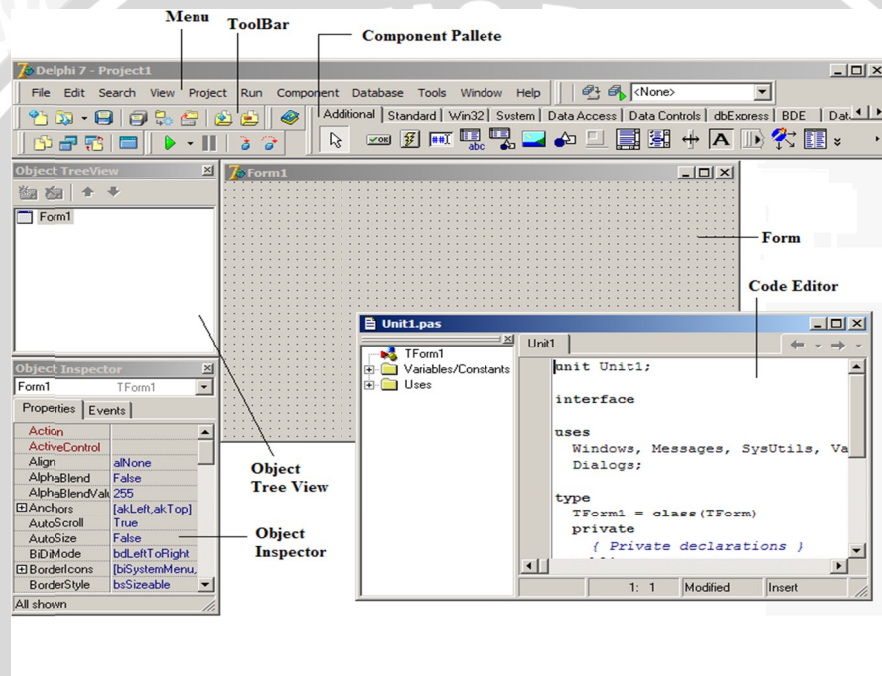
## 2. 11 Borland Delphi

Secara umum, Borland Delphi adalah sebuah program untuk membuat aplikasi – aplikasi berbasis Windows. Bahasa pengembangan yang digunakan oleh Delphi adalah bahasa Pascal. Turbo Pascal dikenal dengan kelebihan dalam kecepatan eksekusi dan kompilasi, dibandingkan dengan bahasa pemrograman lain yang berkembang saat ini. *Integrated Development Enviroment* (IDE) yang diperkenalkan dan diterapkan oleh Turbo Pascal sangat memudahkan para programmer merealisasikan program aplikasi mereka. Dengan IDE seorang programmer dapat dengan cepat dan mudah menulis kode program, melakukan kompilasi, melihat kesalahan (*error*) program, serta langsung menuju letak kesalahan dan memperbaiki kesalahan tersebut. Kemudian Turbo Pascal dirubah menjadi yang berorientasi obyek (*Object Oriented Programming*) berbasis tampilan visual yang menarik, dan dilengkapi kemampuan akses ke basis data. Inilah yang kemudian dikenal sebagai Delphi.

Delphi dapat digolongkan ke dalam bahasa tingkat tinggi (*High Type Language*) karena segala kemudahan ditawarkan untuk perancangan sebuah aplikasi.

### 2.11.1. IDE (*Integrated Development Enviroment*)

*IDE* adalah sebuah lingkungan yang berisi tool – tool yang diperlukan untuk desain, menjalankan dan mengetes sebuah aplikasi, disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. Di Delphi, *IDE* terdiri dari :



**Gambar 2.13.** IDE (*Integrated Development Enviroment*)

(*Sumber: Visual Borland Delphi 7, 2004*)

#### a. Main Window

Main Window adalah bagian utama dari *IDE*. Main Window mempunyai semua fungsi utama dari program – program Windows lainnya.

#### b. Menu Utama

Menu utama dipakai untuk membuka atau menyimpan file, memanggil wizard, menampilkan jendela lain, mengubah option dan lain sebagainya.

#### c. Toolbar

Dengan menu toolbar dapat melakukan beberapa operasi pada menu utama yang setiap tombol berisi informasi mengenai fungsi dari tombol tersebut.

#### d. Form Designer

Jendela kosong yang digunakan untuk merancang aplikasi Windows.

#### e. Code Editor

Merupakan bagian yang terpenting di lingkungan Delphi. Jendela ini dipakai untuk menuliskan program Delphi.

#### f. Code Explorer

Code explorer digunakan untuk memudahkan navigasi didalam file unit.

#### g. Object Treeview

Merupakan daftar dari komponen-komponen apa saja yang telah kita gunakan dan juga merupakan peta dari program yang kita buat.

### 2.11.2 Menu Borland Delphi

#### 1. Menu File

Berisi fasilitas untuk membuat Project baru, menyimpan Project, membuka Project, dan keluar dari IDE Delphi.

#### 2. Menu Edit

Berisi fasilitas untuk melakukan *editing* atau perubahan pada kode program, juga pengaturan form dan unit ( ukuran, penempatan, kontrol, dsb ).

#### 3. Menu Search

Berisi Fasilitas untuk melakukan pencarian atau penggantian kata dalam tubuh kode program ( unit ) dan juga mencari letak kesalahan program.



#### 4. **Menu View**

Berisi fasilitas untuk mengatur tampilan IDE Delphi. Misalnya Object Inspector, daftar komponen, pengaturan *Toolbar*, Form, dan Unit.

#### 5. **Menu Project**

Berisi fasilitas yang berkaitan dengan properti dari Project, misalnya menambahkan atau memisahkan Form dan Unit dari sebuah Project.

#### 6. **Menu Run**

Berisi fasilitas untuk Kompiler Delphi, yang terpenting adalah *Run* dan *Reset*

#### 7. **Menu Component**

Berisi fasilitas untuk mengatur properti *Component Pallete* dan instalasi komponen baru.

#### 8. **Menu Database**

Berisi fasilitas yang berkaitan dengan pembuatan aplikasi data.

#### 9. **Menu Tools**

Berisi fasilitas untuk melakukan pengaturan direktori, *library*, *path* penyimpanan file-file penting dalam Delphi, dan tools yang bekerjasama dengan Delphi.

#### 10. **Menu Window**

Berisi fasilitas untuk berpindah dari satu jendela kerja ke jendela kerja yang lain dalam IDE Delphi.

#### 11. **Menu Help**

Berisi fasilitas menerima bantuan atau keterangan tentang Delphi.

### 2.11.3 Komponen Visual dan Nonvisual

*Visual Component Library (VCL)* adalah herarki dari *class (object)* yang ditulis dengan *object Pascal* dan tergabung dalam *IDE Delphi*. Dengan *Component Palette* dan *Object Inspector*, kita dapat menempatkan komponen *VCL* pada form dan memanipulasi propertinya tanpa menuliskan program.

*VCL* adalah *Library Delphi* dalam lingkungan Windows. Delphi membagi komponen-komponen ke dalam dua kelompok besar berdasarkan visualisasinya, yaitu:

#### 1. Komponen Visual

Yaitu komponen yang akan ditampilkan di layar pada saat aplikasi dijalankan. Komponen visual dipakai untuk membangun antarmuka dengan pemakai. Komponen ini dapat diubah ukurannya dengan cara drag pada salah satu kotak kecil yang ada di sekeliling komponen. Contoh komponen visual adalah Button, Edit, Memo, dan lain-lain.

#### 2. Komponen Nonvisual

Yaitu komponen yang tidak ditampilkan di layar pada saat aplikasi dijalankan, tetapi diperlukan dalam aplikasi. Ukuran komponen ini tidak dapat diubah.