

## BAB II

### LANDASAN TEORI

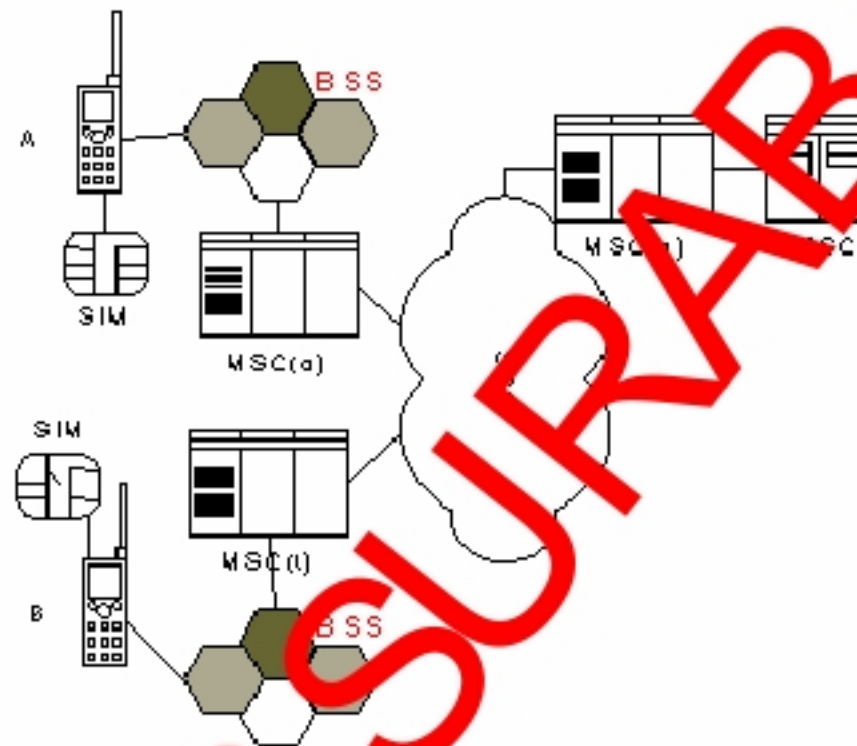
#### 2.1 SMS (*Short Message Service*)

SMS merupakan sebuah metode pengiriman berita singkat melalui jalur komunikasi wireless dengan perangkat bantu *HandPhone*. Dengan *Short Message Service* (SMS), pengguna *handphone* dapat mengirim dan menerima pesan/message singkat kurang lebih sebanyak 160 karakter. Pesan yang dikirim dapat berupa text, kata, nomor ataupun paduan dari alphanumerik. Umumnya apabila SMS yang dikirim berupa huruf latin maka sms dapat memuat sebanyak 160 karakter, apabila pesan yang dikirim berupa huruf Arab atau Cina maka jumlah karakter dapat mencapai 70 karakter.

Cara kerja SMS secara umum bisa dijelaskan sebagai berikut. Seluruh *operator GSM (Global Service Mobile) Network* mempunyai sebuah *message center*, dimana *message center* ini bertanggung jawab terhadap pengoperasian sekaligus manajemen dari pesan atau berita yang ada. Apabila seseorang mengirimkan sebuah pesan kepada orang lain melalui *handphone*, maka pesan ini harus melewati *message center* dari *operator network* tersebut, dan *message center* ini dengan segera dapat memunculkan tujuan/penerima pesan tersebut. *Message center* ini akan menambahkan pesan tersebut dengan tanggal, waktu dan nomor *handphone* pengirim, kemudian mengirimkan pesan tersebut kepada nomor tujuan atau penerimanya. Apabila *handphone* penerima sedang tidak aktif, maka *message center* akan menyimpan pesan

tersebut dan akan mengirimkannya lagi kepada penerima apabila *handphone* penerima aktif kembali atau terhubung dengan *network*.

Secara teknis yang diperlukan dalam penyelenggaraan SMS sesuai dengan diagram berikut :



Gambar 2.1 Diagram fungsi transaksi SMS

Bila kita perhatikan diagram tersebut diatas, dapat diamati ada beberapa komponen utama yaitu :

- Handphone* (A dan B), sering disebut sebagai *mobile terminal*, yang dilengkapi dengan SIM (*subscriber identity module*) card.
- BSS (*base station sub-system*) yang juga dikenal sebagai subsistem radio berfungsi untuk menyediakan dan mengatur jalur transmisi radio antara *handphone* dengan *Mobile Switching Center* (MSC). BSS ini juga berfungsi untuk mengatur *interface* radio antara *mobile stations* dan subsistem lain dalam GSM *network* lazim dikenal sebagai *radio network*.

- c. SSS, *signalling sub-system*, untuk mudahnya dikenal sebagai kumpulan dari MSC.
- d. SMSC, *short message service center*, yang berfungsi sebagai SMS server. SMSC memiliki nomor khusus (*global title*), yang dikenal secara global, baik dalam jaringan Telkomsel maupun jaringan *roaming partner*. Untuk Excelcom nomor ini adalah +62818445009. Tanda '+' merupakan kode akses international yang bersifat universal di seluruh jaringan GSM. Dengan menggunakan tanda '+', kita tetap dapat menggunakan SMS, walaupun tengah melakukan *roaming* internasional.

Untuk pelanggan GSM, pada *handphone* dapat melakukan dua jenis transaksi SMS, yaitu SMS-MO (*mobile originating*, mengirim SMS) dan SMS-MT (*mobile terminating*, menerima SMS). Sebagai ilustrasi, pelanggan A mengirimkan SMS ke pelanggan B. Maka, alur transaksi yang terjadi adalah sebagai berikut:

1. Pelanggan A, menulis pesan SMS pada *handphone*, lalu memberikan perintah pengiriman.
2. Pesan akan melalui BSS tempat pelanggan berada, lalu diterima oleh *serving MSC* (MSC(o) dimana pelanggan berada pada saat pengiriman), dicatat oleh MSC sebagai CDR (*call detail record*) SMS-MO, pada *message envelope* tercatat *calling number* adalah A sedangkan *called number* adalah SMSC. Nomor tujuan, B, disimpan didalam *message body*.
3. Melalui SSS, pesan akan disampaikan ke MSC (MSC(g), yang dalam hal ini bertindak sebagai *gateway* untuk SMSC) yang berhubungan dengan SMSC.



4. SMS lalu disimpan di dalam *database* transaksi SMS yang ada di SMSC.

Untuk setiap transaksi SMS yang ditangani oleh SMSC, akan tersimpan *transaction log*, yang berfungsi mirip dengan CDR pada MSC.

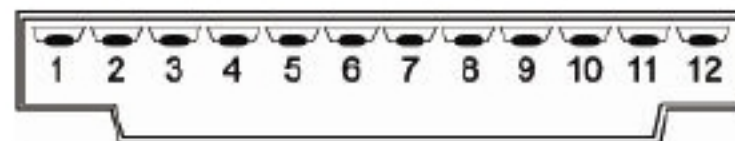
Alur diatas menggambarkan sebuah transaksi SMS-MO. Lantas, bagaimana dengan SMS-MT, yang dalam hal ini diwakili oleh pelanggan B. Setelah SMS diterima oleh SMSC, langkah-langkah berikut dilakukan:

1. SMSC memulai proses pencarian lokasi pelanggan melalui HLR (*home location register*) dimana pelanggan terdaftar, langkah ini dikenal dengan nama *sms\_send\_routing info*.
2. HLR akan menginformasikan status lokasi pelanggan saat ini, sesuai dengan informasi yang tercatat di HLR.
3. SMSC selanjutnya akan mengirimkan SMS ke lokasi pelanggan (dalam hal ini MSC tujuan) sesuai dengan informasi dari HLR, *sms\_forward\_short\_message*. MSC tujuan akan mencatat CDR SMS-MT, pada *message envelope* tercatat *calling number* adalah SMSC sedangkan *called number* adalah B. Seperti halnya dengan SMS-MO, nomor pengirim, A, disimpan didalam *message body*.

## 2.2 Interface Siemen C45

Siemens C45 memiliki konektor komunikasi serial standar yang digunakan untuk berkomunikasi, seperti yang terlihat pada gambar 2.2 sedangkan fungsi dari pin-pin konektor dapat dijelaskan pada tabel 2.1





Gambar 2.2 Konektor Siemens C45

Tabel 2.1 PinOut Konektor Siemens C45

Pin	Nama	Fungsi	In / Out
1	GND	Ground	
2	SELF SERVICE	Recognition / control battery charge	In / Out
3	LOAD	Charging Voltage	In
4	BATTERY	Battery	Out
5	DATA OUT	Data Sent	Out
6	DATA IN	Data Received	In
7	Z_CLK	Recognition / control accessories	
8	Z_DATA	Recognition / control accessories	
9	MICG	Ground for microphone	In
10	MIC	Microphone Input	
11	AUD	Low speaker	Out
12	AUDG	Ground for external speaker	

Untuk dapat berkomunikasi dengan PC ataupun piranti lain diperlukan port serial, dimana komunikasi serial tersebut mempunyai baudrate 19200, 8 bit, Odd Parity, 1 Stop Bit.

## 2.3 Protokol Siemens C45

### 2.3.1. Perintah AT Command

Untuk dapat berkomunikasi, selain memerlukan *interface siemens* juga memerlukan protokol komunikasi, dalam hal ini *siemens* memiliki standar protokol yang sama yang dimiliki dengan PC, yaitu *AT Command* & *AT Command* sebenarnya sama dengan perintah  $>$  (prompt) pada DOS (*Disk Operating System*). Perintah – perintah yang dimasukkan ke dalam Port Serial di awal dengan **AT** lalu diikuti oleh karakter lainnya, yang memiliki fungsi sendiri. Standar yang sama juga digunakan oleh modem untuk berkomunikasi dengan PC. Perintah-perintah *AT Command* biasanya disediakan oleh vendor dari alat komunikasi yang kita beli. Dengan *AT Command* ini maka kita mampu berkomunikasi dengan *HandPhone*. Beberapa *AT Command* penting yang digunakan untuk SMS yaitu :

1. *AT+CMGL=n ?*, yaitu untuk memeriksa SMS, dimana :
  - a.  $n = 0$  untuk memeriksa SMS baru di *Inbox*.
  - b.  $n = 1$  untuk memeriksa SMS lama di *Inbox*.
  - c.  $n = 2$  untuk memeriksa SMS *Unsent* di *Outbox*.
  - d.  $n = 3$  untuk memeriksa SMS *Sent* di *Outbox*.
  - e.  $n = 4$  untuk memeriksa semua SMS.
2. *AT+CMGD=n ?*, yaitu untuk menghapus SMS, dimana **n** adalah referensi index SMS yang akan dihapus.
3. *AT+CMGS*, yaitu untuk mengirim SMS, khusus untuk perintah ini diikuti dengan data I/O yang diwakili oleh apa yang disebut dengan PDU (Protokol Data Unit).

Biasanya *AT Command* tersebut diikuti oleh perintah enter untuk mengakhiri sebuah perintah, tetapi ada beberapa *AT Command* yang tidak menggunakan enter sebagai akhir perintah, hal ini tentu saja disesuaikan dengan fungsi dan *AT Command* yang terdapat pada tabel 2.2..

Tabel 2.2 Fungsi AT Command

Perintah	Fungsi
AT+CACM	Akumulasi panggilan meter
AT+CALM	Peringatan tipe suara
AT+CAMM	Maksimum panggilan meter terakumulasi
AT+CAOC	Jumlah yang perlu diisi
AT+CBC	Pengisi baterai
AT+CBST	Pemilihan pembawa tipe servis
AT+CCFC	Panggilan ke depan
AT+CCLK	Jam
AT+CCWA	Panggilan ditunggu
AT+CEER	Query alasan untuk memutuskan hubungan panggilan terakhir
AT+CGACT	PDP pengisi aktif atau non aktif
AT+CGANS	Respons manual ke sebuah permintaan jaringan untuk PDP mengaktifkan konteks
AT+CGATT	GPRS attach atau detach
AT+CGAUTO	Respons otomatis ke sebuah permintaan jaringan untuk PDP mengaktifkan konteks
AT+CGCLASS	GPRS mobile station class
AT+CGDATA	Tipe pemasukan data
AT+CGDCONT	Pentuan isi PDP
AT+CGERE	Laporan kejadian GPRS
AT+CGMI	Ida manufaktur kode ID
AT+CGMM	Isu model ID
AT+CGMR	Keluaran GSM versi telepon
AT+CGPADDR	Petunjuk alamat PDP
AT+CGQMIN	Kualitas dari servis profile (penerimaan minimum)
AT+CGQREQ	Kualitas dari servis profile(diminta)
AT+CGREG	Jaringan pendaftaran status GPRS
AT+CGSMS	Pilihan servis untuk pesan MO SMS
AT+CGSN	Pengeluaran nomer serial(IMEI)
AT+CHLD	Panggilan di - hold dan multiparty
AT+CHUP	Panggilan terminal
AT+CIMI	Keluaran dari IMSI



Perintah	Fungsi
AT+CKPD	Keypad kontrol
AT+CLCC	Daftar panggilan sekarang
AT+CLCK	Switch pengunci on dan of
AT+CLIP	Penampilkan nomer telepon pemanggil
AT+CLIR	Memilih modus incognito(pembatasan panggilan line identifikasi)
AT+CLVL	Level volume pengeras suara
AT+CMEE	Perluasan pesan error menurut GSM 07.07
AT+CMUT	Mute kontrol
AT+COLP	Terkoneksi baris lampiran identitas
AT+COPN	Membaca nama operator
AT+COPS	Perintah mengenai pilihan operator jaringan
AT+CPAS	Query status telepon
AT+CPBR	Membaca sebuah pemasukan buku telepon
AT+CPIN	Memasukkan pin dan kunci query
AT+CPOL	Mengajukan data operator
AT+CPUC	Harga per unit dan tabel data uang
AT+CPWD	Mengubah password ke sebuah kunci
AT+CR	Kontrol servis telepon
AT+CRC	Hasil kode-kode seluler
AT+CREG	Pendaftaran jaringan
AT+CRLP	Memilih radio link protocol para meter untuk keaslian non – transparansi data panggilan
AT+CRSL	Level bunyi kring (ringer)
AT+CRSM	Keterbatasan akses SIM
AT+CSCS	Memilih set karakter TE
AT+CSQ	Kualitas sinyal keluaran
AT+CSSN	Supplementary service notifications
AT+CVIB	Modus getaran
AT+GSN	Pengeluaran nomer serial (IMEI)
AT+VTS	Pengiriman sebuah bunyi DTMF
AT+VTD	Menentukan lamanya DTMF tone
AT+WZC	Memilih jaringan wireless
AT+CMGF	Mengirim sebuah perintah SMS
AT+CMGD	Menghapus sebuah SMS dalam SMS Memory
AT+CMGF	SMS format
AT+CMGL	Daftar SMS
AT+CMGR	Membaca dalam sebuah SMS
AT+CMGS	Mengirim sebuah SMS
AT+CMGW	Mengisi sebuah SMS ke SMS Memory
AT+CMSS	Mengirim sebuah SMS dari SMS Memory
AT+CNMA	Pemberitahuan pesan singkat langsung pengeluarannya

Perintah	Fungsi
AT+CNMI	Menampilkan datangnya SMS baru
AT+CPMS	Preferred SMS message storage
AT+CSCA	Alamat dari pusat SMS servis
AT+CSCB	Memilih sel pesen-pesen broadcast
AT+CSMS	Pemilihan dari servis pesan
AT+GCAP	Pemintaan daftar kemampuan
AT+IPR	Terpasang DTE rate
AT+FBADLIN	Menentukan atau membaca nomer line-line yang rusak ( bad line)
AT+FBADMUL	Menentukan, membaca, atau tes nomer dan line-line yang rusak
AT+FBOR	Query urutan bit untuk modus penerimaan
AT+FCIG	Query atau set pemilikan lokal id
AT+FCLASS	Memilih baca atau tes fax servis class
AT+FCQ	Kontrol kualitas duplikasi
AT+FCR	Kemampuan untuk menerima
AT+FDCC	Memilih servis untuk pesan NO SMS
AT+FDFFC	Kompresi data penukaran format
AT+FDIS	Query atau set session parameter
AT+FDR	Memulai atau melanjutkan fase C penerimaan data
AT+FDT	Transmisi data
AT+FET	Akhir dari halaman atau dokumen
AT+FK	Menghentikan operasi seperti membatalkan fax
AT+FLID	Query or set session parameters
AT+FMDL	Pengelolaan model produk
AT+FMFR	Minta identifikasi manufaktur
AT+FOPT	Set up order secara indeapenden
AT+FPHCTO	DTE Phase C respons waktu – habis
AT+FREV	Mengidentifikasi prouk revisi
AT+FRH	Penerimaan data menggunakan bingkai HDLC
AT+FRM	Penerimaan data
AT+FRS	Penerimaan sepi
AT+FTH	Pemindahan data menggunakan bingkai HDLC
AT+FTM	Pemindahan data
AT+FTS	Penghentian transmisi dan tunggu
AT+VRF	Vertical resolution format conversion
AT+SACM	Keluaran ACM (terakumulasi panggilan meter) dan ACMmax
AT+SBNR	Pembacaan binary
AT+SBNW	Pengisian binary
AT+SCID	Keluaran kartu ID
AT+SCKS	Keluaran kartu SIM status
AT+SCNI	Keluaran nomor panggilan informasi



Perintah	Fungsi
AT+SDLD	Menghapus dial nomor terakhir di memory
AT+SDBR	Pembacaan database
AT+SGAUTH	Select Type of Authentication for PPP conection
AT+SICO	Kontrol gambar icon
AT+SLCK	Switch locks (including user defined locks) on dan off
AT+SLNG	Setting bahasa
AT+SMGO	SMS indikasi overflow
AT+SMGL	Daftar SMS (tanpa perubahan status dari yang tak membaca ke yang membaca)
AT+SMGR	Membaca SMS (tanpa status berubah dari tak terbaca ke yang baca)
AT+SMSO	Mengganti peralatan off
AT+SNFS	Memilih perangkat keras
AT+SNFV	Set volume
AT+SPBC	Melihat kemasukan pertama dari urutan buku telepon yang dimulai dari surat yang terpilih (atau keberadaan selanjutnya)
AT+SPBG	Membaca kemasukan dari urutan buku telepon melalui urutan indeks
AT+SPBS	Memilih sebuah buku telepon
AT+SPIC	Pengeluaran counter
AT+SPLM	Membaca PLMN
AT+SPLR	Membaca sebuah pemasukan dari operator yang ditunjuk
AT+SPLW	Menulis sebuah pemasukan ke operator yang ditunjuk
AT+SPST	Memastikan sinyal tone
AT+SPWD	Mengubah password ke sebuah kunci (termasuk kunci-kunci yang ditentukan pengguna)
AT+SRTC	Set bunyi ring (ringer)
AT+SSTK	Peraturan SIM
A/	Mengulang perintah penting sesudahnya
AT...	Awalan untuk semua perintah lainnya
ATA	Menerima panggilan (V.25ter, menurut [3])
ATB[n]	Perintah modem ini digunakan menset pembawa servis untuk data koneksi (cf.AT+CBST). <n> dapat menggunakan salah satu nilai berikut : 7     2400 bps, asynchronous, V. 22 bis 11    4800 bps, asynchronous, V. 32 13    9600 bps, asynchronous, V. 32 15    14400 bps, asynchronous, V. 34 25    2400 bps, asynchronous, V. 110 ISDN 27    4800 bps, asynchronous, V. 110 ISDN 29    9600 bps, asynchronous, V. 110 ISDN 31    14400 bps, asynchronous, V. 110 ISDN



Perintah	Fungsi
ATD<str>	Dial abjad <str> dengan utiliti suara valid mengubah dial : I Pembatas AT+CLIR i Meniadakan AT+CLIR untuk panggilan selanjutnya T Tone Dial P Pulse Dial dibiarkan Karakter terakhir ";" menentukan ke telefon yang panggilannya harus di-set up dengan utiliti suara. Jika tidak, sebuah uji coba yaitu membuat set up sebuah data panggilan, yang telefon untuk segera memberitahu "ERROR". Perintah Dial akan mengembalikan OK ke pengguna dengan segera sesudah memulai sebuah panggilan suara. Jika tidak, tanda # akan diurutkan dalam perintah dial, dan panggilan data tinggal tak berubah.
ATD><n>;	Dial nomor telepon dari buku telepon nomor lokasi <n>
ATD><mem><n>;	Dial nomor telepon dari buku telepon nomor lokasi <mem> <n>
ATDx[;]	Dial nomor telepon x I ISDN panggilan telepon akan membuat seperti sebuah panggilan ISDN. Sebuah koneksi ISDN ke sebuah V. 110 terminal adaptasi akan diselenggarakan. Kecepatan data transmisi sama seperti untuk sebuah panggilan analog (2400/4800/9600/14400 bps) PP Plus sama seperti + karakter
ATDL	Dial nomor telepon terakhir
ATE0	Pengaturan aktivasi perintah echo
ATE1	Mengaktifkan perintah echo
ATH[0]	Melepaskan koneksi yang ada
ATI[n]	Perintah modem sesuai [3] : Menampilkan kode produksi : 0 042 1 042 2 OK, (check firmware checksum) 8 menampilkan modus operasi yang di-support (lihat ATB) 9 modem identifikasi dan mobile phone
ATI[n]	Monitor kekerasan speaker (perintah modem menurut ke [3])
ATM[n]	Monitor speaker mode (perintah modem menurut ke [3])
ATO[n]	Switch kembali ke transparent mode setelah +++ interupsi (perintah modem menurut ke [3])
ATQ0	Menampilkan penerimaan (respons atau pesan)
ATQ1	Meniadakan pemberitahuan (respons atau pesan)
ATSn=x	Menulis nilai x ke S register n (perintah modem menurut [3])

Perintah	Fungsi
ATSn?	Memperlihatkan nilai S register n (perintah modem menurut [3]) Catatan: tipe mobile phone tidak mengijinkan nilai dari semua S register untuk ditampilkan dengan sebuah perintah single
ATV0	Menampilkan penerimaan seperti angka-angka
ATV1	Menampilkan penerimaan seperti naskah
ATX<n>	Laporan link dengan hanya CONNECT biarkan sinyal sibuk <n> dapat menggunakan salah satu nilai berikut : 1 Laporan link dengan CONNECT ditambah baudrat, biarkan sinyal sibuk 2 sama seperti ATX1 3 sama seperti ATX, tetapi laporan BUSY 4 sama seperti ATX, tetapi laporan BUSY
ATZ	Set ke konfigurasi semula
AT&C<n>	Sirkuit 109 (penerimaan garis sinyal terdeteksi/DCD) perilaku <n> dapat mengambil salah satu nilai berikut : 0 DCD selalu ON 1 DCD ON jika pembawa terdeteksi
AT&D<n>	Sirkuit 108 (Data Terminal Ready/DTR) perilaku Catatan : perintah AT+Den> dicirikan berikut sehingga tidak berpengaruh ke sirkuit 108 tidak mendukung dalam tipe mobile phone ini. <n> dapat mengambil salah satu nilai berikut 0 DTR ignored 1 On DTR ON to OFF : ke perintah Online mode, tidak dikoneksi 2 On DTR ON to OFF : tidak dikoneksi ke perintah mode. Menjawab otomatis yaitu dimatikan ketika DTR OFF.
AT&F<0>	Riset semua parameter sementara dari beberapa perintah AT untuk profile pabrik AT, ATQ, ATV, ATX, AT+CBST, AT+CRLP, AT+CRC, AT+CR, AT+CNMI, AT+CMEE, AT+CSMS, AT+SCKS, AT+SACM, AT+CREG, AT+CLIP - S Parameter - AT&D;AT&C;AT&S Beberapa koneksi yang dihadirkan akan terputus. Bukan perintah lain yang diterima pada baris perintah yang sama.
AT+N	Tidak ada aksi (\N2 - \N6)
ATQn	Pilihan aliran control lokal (DTE – DCE); dapat dicustom <n> dapat diambil satu dari nilai berikut : 0 Tanpa aliran control 1 XON – XOFF aliran software controll 2 Hanya CTS aliran controll 3 RTS/CTS aliran controll



Perintah	Fungsi
\V[n]	Perintah modem : No /REL or /RLP lampiran dengan pesan koneksi

Selanjutnya *Siemens* memiliki bahasa khusus sesuai dengan perintah AT *Command* yang kita kirim melalui *port serial*. Bahasa yang dimaksud dikenal dengan istilah PDU (*Protocol Data Unit*). PDU ini berisi bilangan – bilangan hexadesimal yang mencerminkan bahasa I/O. PDU terdiri dari atas beberapa *header* yang memiliki arti sesuai dengan perintah AT *Command* yang kita kirim melalui *port serial* PC yang terhubung dengan *HandPhone Siemens*. PDU untuk mengirim SMS ke SMS *Center* berbeda dengan SMS yang diterima dari SMS *Center*.

### 2.3.2. PDU Mengirimkan SMS ke SMS *Center*

Seperti telah diuraikan sebelumnya bahwa PDU merupakan protokol yang digunakan untuk mengirimkan SMS, penggunaannya setelah memberikan perintah AT *Command* (AT+CMGS=? ) yang diikuti dengan PDU untuk pengiriman SMS.

Sebagai contoh untuk mengirimkan kata “hello” ke nomor ponsel +6281931503530 melalui SMS *Center* Excelcom, tanpa memberi batasan jangka waktu validasi, maka PDU lengkapnya adalah :

07 12618485400F901000D91261839515035F0000005E8329BFD06



PDU untuk mengirim SMS tersebut apabila kita uraikan terdiri dari 8 *header* sebagai berikut :

### 1. Nomor SMS Center

Nomor SMS Center yang dimaksud adalah nomor SMS Center dari Operator pengirim itu sendiri. Adapun *header* pertama ini terbagi atas tiga sub *header* yaitu :

- a. Jumlah pasangan heksadesimal SMS Center dalam bilangan heksa.
- b. *National/International Code*
  - Untuk *National Code Subheader*-nya adalah 81
  - Untuk *International*, kode *subheader* nya adalah 91
- c. No SMS Center-nya sendiri, dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, maka angka tersebut akan dipasangkan dengan F di depannya.

Sebagai contoh untuk nomor SMS Center dari Excelcom adalah :

#### Cara 1 (menggunakan *national code*) :

Nomor SMS Center = 0818445009, maka akan diubah menjadi :

- a. 06 → karena memiliki 6 pasang heksa yaitu 5 pasang nomor SMS Center dan 1 pasang untuk *National Code*.
- b. 81 (*National Code*)
- c. 80 – 81 – 44 – 05 – 90 (No SMS Center yang dibalik-balik)

Sehingga apabila digabungkan menjadi **06818081440509**

**Cara 2 (menggunakan international code) :**

Nomor SMS Center = 62818445009, maka diubah menjadi :

- a. 07 → karena memiliki 6 pasang heksa yaitu 6 pasang nomor SMS Center dan 1 pasang untuk *National Code*.
- b. 91 (*National Code*)
- c. 26 – 18 – 48 – 54 – 00 – F9 (Nomor SMS Center yang dibalik-balik, dimana F merupakan penambahan karena satu angka tidak memiliki pasangan heksa)

Sehingga apabila digabungkan menjadi **07912618485400F9**

Berikut ini adalah Kode PDU untuk beberapa operator selular yang ada di Indonesia :

Tabel 2.3 PDU SMS Center dengan *National Code*

No.	Operator Seluler	No SMS Center	Kode PDU
1.	Telkomsel	0811000000	06818011000000
2.	Satelindo	0816124	0581806121F4
3.	Excelcom	0818445009	06818081440590
4.	Indosat	0855000000	06818055000000

Tabel 2.4 PDU SMS Center dengan *International Code*

No.	Operator Seluler	No SMS Center	Kode PDU
1.	Telkomsel	62811000000	07912618010000F0
2.	Satelindo	62816124	059126181642
3.	Excelcom	62818445009	07912618485400F9
4.	Indosat	62855000000	07912658050000F0

**2. Tipe SMS**

Untuk *SEND* type SMS = 1, jadi bilangan heksa adalah **01**

### 3. Nomor referensi SMS

Nomor referensi ini dibiarkan dahulu 0, jadi pasangan heksanya adalah **00** karena nanti akan diberikan sebuah nomor referensi secara otomatis oleh *handphone* atau alat SMS Gateway.

### 4. Nomor *handphone* penerima

Sama dengan cara menulis PDU header untuk nomor SMS Center, header ini juga terbagi atas tiga *subheader*, sebagai berikut :

- a. Jumlah bilangan desimal nomor *handphone* penerima, ditulis dalam bentuk heksadesimal.
- b. *National code* diberikan **81** sedangkan untuk *international code* diberikan bilangan **91**.
- c. Nomor *handphone* penerima, dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan angka **F** didepannya.

Sebagai contoh adalah nomor penerima **081931505530**

#### Cara 1 (*national code*) :

Nomor penerima **081931505530** akan dirubah menjadi :

- a. 00 karena terdiri dari 12 angka
- b. 81 (*National Code*)
- c. 80 - 91 - 13 - 05 - 55 - 03 (No SMS Center yang dibalik-balik)

Sehingga apabila digabungkan menjadi **0C81809113055503**

#### Cara 2 (*international code*)

Nomor penerima **6281931505530** akan dirubah menjadi :



- a. 0D, karena terdiri dari 13 angka
- b. 91 (*National Code*)
- c. 26 – 18 – 39 – 51 – 50 – 35 – F0 (No SMS Center yang dibalik-balik)

Sehingga apabila digabungkan menjadi **0D91261839515035F0**

## 5. Bentuk SMS

Bentuk sms yang diberikan dalam *header* ini adalah :

- 0 → 00 → dikirim sebagai SMS
- 1 → 01 → dikirim sebagai telex
- 2 → 02 → dikirim sebagai fax

maka dalam hal ini apabila kita mengirim pesan dalam bentuk SMS maka tentu saja *header* ini harus diisi dengan 00

## 6. Skema *encoding Data I/O*

*Header* ini merupakan bagian yang penting, dimana didalam *header* ini menjelaskan bahwa paket data yang dikirim dengan menggunakan salah satu dari dua metode yaitu skema 7 bit atau 8 bit. Dimana untuk skema 7 bit ditandai dengan 0 yang diubah kedalam pasangan heksa menjadi **00**, sedangkan untuk skema 8 bit biasanya ditandai dengan lebih besar dari 0 dan diubah kedalam pasangan heksa.

Kebanyakan *handphone* ataupun SMS Gateway yang beredar di pasaran saat ini menggunakan skema 7 bit, sehingga kita memberikan kode **00**.

## 7. Jangka waktu sebelum SMS *expired*

*Header* ini merupakan batas waktu validasi pengiriman SMS. Jika bagian ini diabaikan dalam arti memberikan nilai **00** berarti kita tidak menentukan batas waktu validasinya. Tetapi apabila dengan bilangan integer tertentu yang kemudian

diubah dalam pasangan heksa , maka bilangan yang kita berikan tersebut akan mewakili nilai validasi SMS. Rumus yang digunakan untuk menentukan batas waktu validasinya seperti pada tabel dibawah ini :

Tabel 2.5 Menentukan batas waktu validasi SMS

Integer (INT)	Jangka Waktu Validasi SMS
0 – 143	$(INT + 1) * 5$ menit (berarti : 5 menit – 1 jam)
144 – 167	12 jam + $((INT) - 143) * 30$ menit
168 – 196	$(INT - 166) * 1$ hari
197 - 255	$(INT - 166) * 1$ minggu

Agar SMS kita pasti terkirim sebaiknya kita tidak memberikan batas waktu validasinya.

#### 8. Isi SMS

*Header* ini terdiri atas dua *sub header*, yaitu :

- Panjang isi (jumlah huruf dari isi SMS), misalnya untuk kata “Hello” maka *header* ini adalah 05
- Isi, berupa pasangan bilangan heksa. Pada dasarnya untuk *Handphone* atau *SMS Gateway* berskema 7 bit, maka jika kita memasukkan angka atau karakter dari keypadnya, berarti kita telah membuat 7 angka (bit) I/O secara berurutan. Isi SMS tersebut harus dikonversikan terlebih dahulu dalam bentuk PDU, dengan cara :

- Mengubah per karakter isi SMS terlebih dahulu menjadi kode 7 bit (biner), kode 7 bit dapat dilihat dalam tabel 2.6.

Misalnya untuk kalimat “Hello”, maka susunannya menjadi



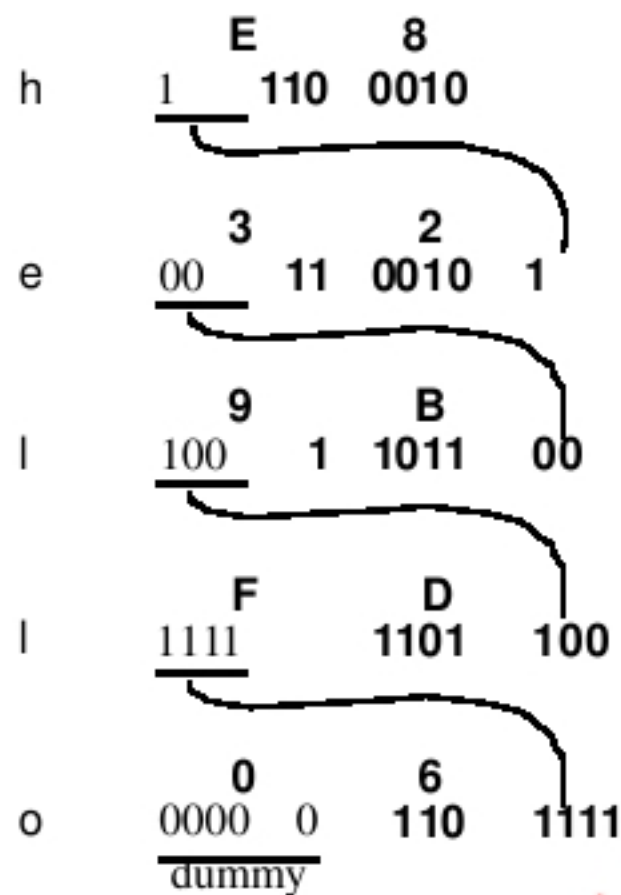
	Bit	7	6	5	4	3	2	1
<b>h</b>		1	1	0	1	0	0	0
<b>e</b>		1	1	0	0	1	0	1
<b>l</b>		1	1	0	1	1	0	0
<b>l</b>		1	1	0	1	1	0	0
<b>o</b>		1	1	0	1	1	1	1

- Kemudian dari kode 7 bit diubah menjadi 8 bit, yang diawali dalam pasangan heksa.

Tabel 2.6 Konversi 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			\$	0	-	P	"	p	
0	0	0	1	1			!	1	A	Q	a	q	
0	0	1	0	2			Φ	"	2	B	R	b	r
0	0	1	1	3			†	#	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			∏	&	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	A	LF	Ξ	*	:	J	Z	j	z	
1	0	1	1	B			+	;	K	Ä	k	ä	
1	1	0	0	C			,	<	L		l	ö	
1	1	0	1	D	CR		-	=	M	ü	m		
1	1	1	0	E		β	.	>	N		n	ü	
1	1	1	1	F			/	?	O		o		

Selanjutnya adalah menggabungkan skema 7 bit menjadi skema 8 bit,



Konversi diatas memerlukan *dummy*, yang didapatkan dengan menambahkan angka 0 didenpan sebanyak **n**. Dimana **n** dapat dicari dengan :

$$n = 8 - ((7 \text{ (panjang isi SMS)}) \text{ mod } 8) \quad (2.1)$$

Maka PDU yang didapatkan adalah E8329BFD06

Dengan demikian untuk *header* isi SMS PDU yang didapatkan **05E8329BFD06**

Setelah merinci ke delapan *header* diatas didapatkan kode PDU untuk masing-masing *header*-nya :

1. No. SMS center → **07912618485400F9**
2. Tipe SMS → **01**
3. Nomor referensi SMS → **00**
4. Nomor *handphone* penerima 6281931505530 → **0D91261839515035F0**
5. Bentuk SMS → **00**

6. Skema Encoding 7 bit → 00

7. Isi SMS “hello” → 05E8329BFD06

**PDU Lengkap 07912618485400F901000D91261839515035F0000005E8329BFD06**

Maka untuk melakukan pengiriman SMS “hello” melalui SMS Center Excelcom dengan nomor tujuan 6281931505530, maka AT Command yang diberikan melalui serial port Siemens adalah :

- AT+CMGS=19 akhiri perintah dengan karakter Chr\$13 (Enter). Angka 19 yang diberikan merupakan nilai yang didapat dari jumlah pasangan heksa dalam bentuk PDU dihitung setelah PDU nomor SMS Center.
- Setelah itu Siemens akan merespon dengan memberikan tanda “>”, ini dimaksudnya bahwa handphone siap menerima PDU Lengkap gabungan kedelapan header diatas, dalam hal ini adalah :

**07912618485400F901000D91261839515035F0000005E8329BFD06**

- Setelah itu akhiri perintah dengan karakter Chr\$ 26 (Ctrl+Z).
- Apabila Sukses handphone Siemens akan memberikan balasan +CMGS:n kemudian OK. Dimana n adalah nomor referensi kirim SMS yang diberikan secara otomatis oleh SMS gateway Siemens.





62818445009F, **F** merupakan *carry* yang ditambahkan karena terdapat satu angka yang tidak memiliki pasangan heksa. Jadi nomer SMS *Center* nya adalah **62818445009**, diketahui bahwa Operator seluler pengirimnya adalah Excelcom.

2. Tipe SMS, **04**. ini merupakan tipe SMS terima.
3. Nomer *handphone* pengirim, **0D91261808528105F7**.

Teknik Konversinya sama dengan kode PDU untuk *Send SMS*, jika diuraikan maka :

- a. *Sub header OD*, merupakan bentuk bilangan heksa jumlah (*length*) dari nomer *handphone* pengirim, OD berarti panjang nomer pengirimnya adalah 13 digit.
- b. Sub header **91**, menggunakan international code.
- c. Sub header **261808528105F7**, Nomer *handphone* pengirim yang dibalik-balik. Jika diterjemahkan nomer *handphone* pengirimnya adalah 6281802518507F, **F** merupakan *carry* yang ditambahkan karena terdapat satu angka yang tidak memiliki pasangan heksa. Jadi nomer *handphone* pengirimnya adalah **6281802518507**.

4. Bentuk SMS, **00**. Ini berarti pesan yang diterima dalam bentuk SMS
5. Skema pengkodean SMS, **00**. Ini berarti menggunakan skema pengkodean 7 bit.

SMS *Time Stamps*, **50408211104182**

- a. *Sub header 504082*, merupakan tanggal yang pada saat SMS ditandai oleh SMS *Center*. Format **yy/mm/dd hh:mm:ss**. PDU tersebut apabila

diterjemahkan menjadi 050428 (dibalik-balik). Jadi tanggalnya adalah 05/04/28 → 28 April 2005

- b. *Sub header 111041*, merupakan waktu yang pada saat SMS di undah oleh SMS Center. Format **hh:mm:ss**. PDU tersebut apabila diterjemahkan menjadi 110141 (dibalik-balik). Jadi waktunya adalah 11:01:41
- c. *Sub header 82*, merupakan Kode PDU untuk wilayah waktu GMT + 7.

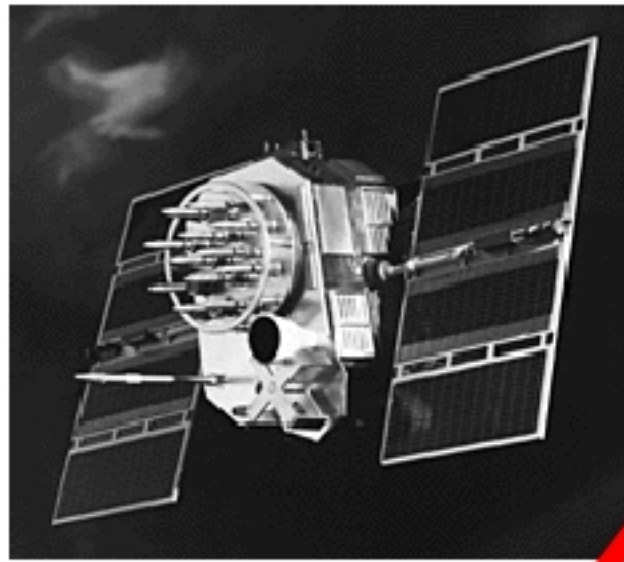
#### 7. Isi SMS, **05E8329BFD06**

- a. *Sub header 05* merupakan bilangan heksa dari panjang isi SMS (*length*) yang dikirim. Ini berarti terdapat 5 karakter dalam isi SMS
- b. *Sub header E8329BFD06*, merupakan PDU dari isi SMS yang dikirim. Kode ini apabila kita terjemahkan menjadi *binary code*, lalu kita pasangkan menjadi skema 7 bit berdasarkan referensi pada tabel 2.6 akan menjadi kalimat "hello" yang memiliki panjang karakter (*length*) sebanyak 5 digit.

## 2.4 GPS (*Global Positioning System*)

GPS adalah suatu sistem navigasi global yang melingkupi seluruh planet bumi ini. Sistem ini dibentuk oleh 24 satelit GPS dan stasiun-stasiun bumi. Fungsi dari satelit-satelit tersebut adalah untuk memberikan informasi-informasi pada seseorang yang berada di bumi seperti jarak antara orang tersebut dengan satelit, waktu internasional dan lain sebagainya. Sedangkan stasiun-stasiun bumi digunakan untuk memonitor data-data yang akan dipancarkan oleh satelit sekaligus mengoreksi adanya penyimpangan-penyimpangan yang terjadi.





Gambar 2.3. Satelit GPS

Secara sederhana cara kerja GPS dapat dijelaskan sebagai berikut. GPS penerima di bumi menangkap sinyal dari tiga satelit GPS. Dengan menggunakan rumus fisika sederhana,

$$\text{Jarak} = \text{Kecepatan Cahaya} \times \text{Waktu} \quad (2.2)$$

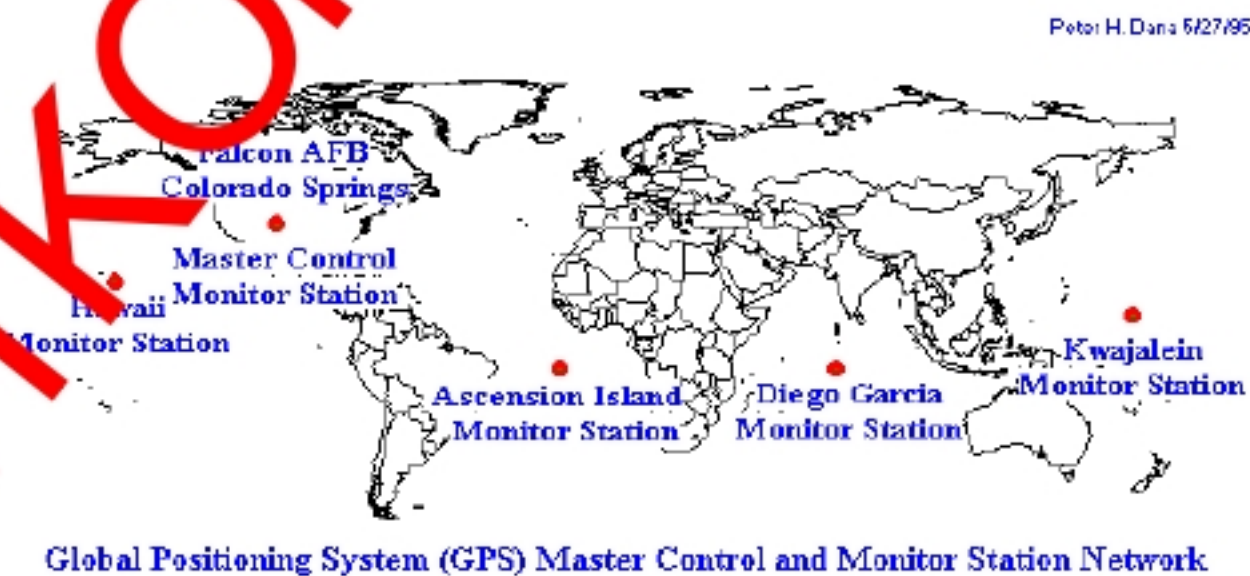
maka jarak masing-masing satelit ke penerima dapat diketahui. Dengan diketahui tiga jarak tersebut maka bila masing-masing jarak itu adalah jari-jari lingkaran akan didapatkan tiga buah lingkaran yang berpotongan di satu titik. Titik potong itulah yang merupakan posisi penerima yang sebenarnya di atas bumi.

Keterangan di atas merupakan cara kerja sederhana dari GPS apabila dianggap bahwa bumi itu datar. Untuk menghasilkan posisi yang baik secara 3 dimensi maka diperlukan minimal 4 buah satelit. Semakin banyak satelit yang dapat diterima oleh GPS *receiver* maka semakin akurat posisi yang didapatkan.

Selain peran dari satelit-satelit GPS juga harus terus menerus dimonitor oleh stasiun bumi. Hal ini perlu sekali dilakukan karena satelit-satelit tersebut letaknya jauh dari permukaan bumi sehingga kesalahan akan mungkin sekali terjadi.

Contoh dari kesalahan itu adalah bergesernya posisi satelit dari asalnya. Apabila hal ini terjadi, walaupun pergeserannya sangat kecil tetapi hal itu akan mengakibatkan perbedaan yang sangat besar ketika dibaca di bumi mengingat jarak antara penerima dengan satelit begitu jauh. Selain itu stasiun bumi juga akan mengoreksi referensi waktu yang dipakai dalam sistem GPS ini sehingga didapatkan suatu timing yang tepat. Jadi, dengan adanya stasiun-stasiun bumi ini maka faktor kesalahan masing-masing satelit dapat dipantau dan dikirimkan ke tiap-tiap penerima di bumi sehingga masing-masing penerima dapat mengoreksi kesalahan yang terjadi.

Pusat pengendali GPS oleh Amerika Serikat ditempatkan di pangkalan udara Falcon di Colorado. Selain itu juga di tempatkan stasiun-stasiun bumi yang lain, misalnya di Hawaii dan di tempat-tempat lain seperti pada gambar 2.4. Setiap stasiun pemantau akan selalu mengecek tiap-tiap satelit GPS di wilayahnya. Apabila suatu saat ternyata terjadi penyimpangan pada satelit maka stasiun itu akan segera melakukan pengkoreksian sistem ke satelit dengan jalan mengirimkan faktor error yang terjadi saat itu.



Gambar 2.4. Stasiun Bumi Pengendali Satelit-Satelit GPS



### 2.4.1. Cara kerja GPS

Gambaran singkat tentang sistem ini dalam memberikan informasi informasi posisi kepada pembawa GPS *receiver* yang berada di muka bumi. Dalam bab ini selanjutnya akan diterangkan secara lebih mendetail cara kerja dari sistem GPS itu sendiri.

Pada dasarnya cara kerja sistem GPS mempunyai 5 titik dasar. Adalah sebagai berikut :

1. Dasar dari GPS adalah "Pertemuan Titik Potong" dari satelit-satelit GPS.
2. Untuk menemukan titik potong dari satelit-satelit GPS tersebut, pesawat penerima harus mengukur jarak masing-masing satelit ke *receiver* dengan cara menghitung jarak tempuh sinyal radio dari satelit ke *receiver*.
3. Untuk menghitung jarak tempuh sinyal radio ke *receiver* maka GPS memerlukan sistem *timing* yang sangat akurat sehingga waktu pengiriman dengan waktu penerimaan dapat diketahui dengan tepat.
4. Selain itu untuk menghitung jarak maka harus mengetahui dengan pasti posisi masing-masing satelit sehingga dengan sedikit perhitungan matematika dapat ditentukan posisi titik potong itu yang tidak lain adalah posisi penerima.
5. Yang terakhir adalah pengkoreksian hasil yang diterima dengan faktor kesalahan yang terjadi selama gelombang radio dikirimkan, sebagai contoh *delay* yang terjadi akibat gelombang radio melewati *troposfer* dan faktor pembelokan saat gelombang radio melalui lapisan *ionosfer*.

Selanjutnya akan dijelaskan pengertian dari masing-masing titik yang telah dijabarkan di atas sehingga dapat membentuk untaian cara kerja GPS secara lengkap.



### 2.4.2. Mencari titik potong satelit

Apabila pertama kali dilihat bagaimana GPS dapat memberitahu posisi seseorang secara tepat di atas bumi, maka dirasakan bahwa sistem yang dipakai sangatlah rumit dan sulit untuk dimengerti. Akibatnya seseorang akan mengabaikan cara kerja dari GPS itu sehingga akhirnya orang itu hanya menjadi *user* saja tanpa tahu sedikitpun mengenai cara kerja dan ide dasar dari GPS tersebut.

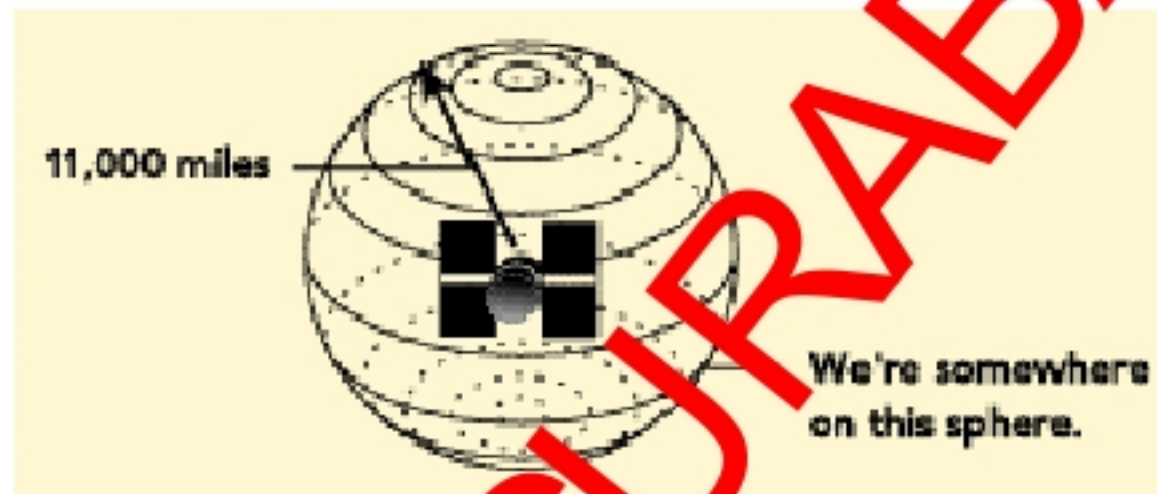


Gambar 2.5 Mencari Titik Potong di Bumi

Sebenarnya di balik semuanya itu, GPS hanya didasarkan pada satu ide saja, yaitu **“menjadikan satelit sebagai titik referensi untuk mengetahui posisi seseorang yang ada di atas bumi”**. Dengan demikian dapat diketahui jarak dan posisi masing-masing satelit terhadap pembawa GPS *receiver*. Setelah itu dengan mengamati data dari beberapa satelit dapat diperoleh titik potong dari jarak satelit-satelit tersebut sehingga diperoleh koordinat yang tepat seperti pada gambar 2.5.

Sebagai contoh, apabila suatu pesawat penerima dapat menerima sinyal dari 3 buah satelit GPS. Kemudian dengan berbagai perhitungan akhirnya didapatkan

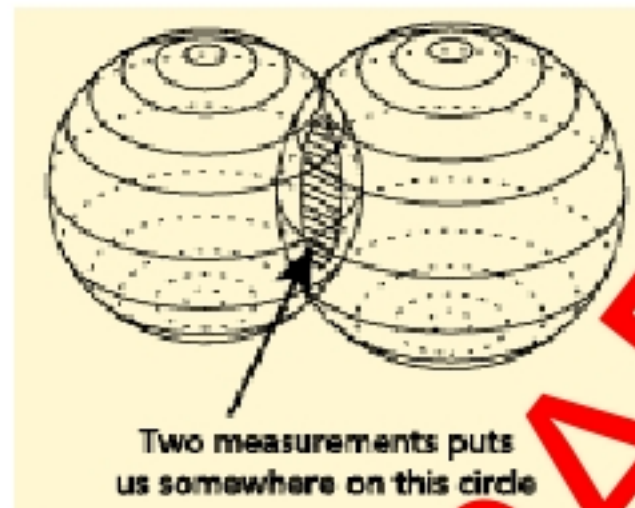
bahwa jarak penerima dengan satelit pertama adalah 11.000 mil. Jarak 11.000 mil akan mempunyai banyak sekali kemungkinan. Bila dianggap satelit pertama sebagai pusat dari bola dengan jari-jari 11.000 mil maka posisi penerima adalah sebuah titik di permukaan bola tersebut. Pada saat ini masih belum bisa didapatkan posisi yang sebenarnya karena faktor kemungkinannya masih terlalu luas.



Gambar 2.6 Satu Satelit Menghasilkan Kemungkinan Posisi Pada Permukaan Bola

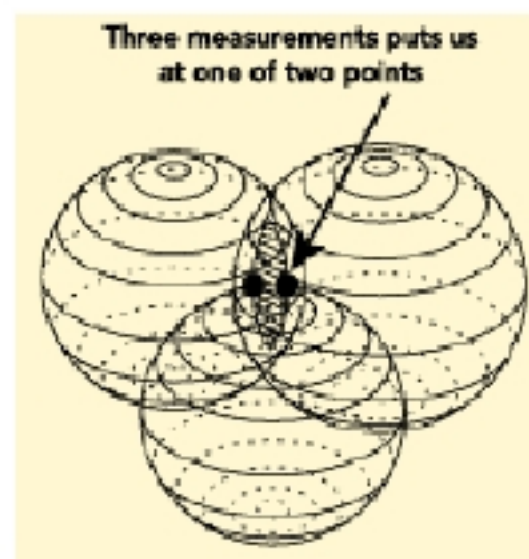
Selanjutnya, dari satelit kedua didapatkan bahwa jarak penerima dengan satelit kedua adalah 12.000 mil. Maka seperti pada satelit pertama pada gambar 2.6, akan didapatkan sebuah bola lagi dengan jari-jari 12.000 mil. Dan posisi penerima juga merupakan suatu titik di permukaan bola dengan jari-jari 12.000 mil tersebut. Dengan demikian telah didapatkan dua buah bola yang saling berpotongan seperti pada gambar 2.7. Perpotongan kedua bola tersebut akan menghasilkan sebuah lingkaran. Dengan demikian maka penerima adalah suatu titik di dalam lingkaran tersebut. Pada saat ini pun masih belum bisa didapatkan kepastian tentang letak posisi penerima yang sebenarnya karena faktor kemungkinannya masih terlalu luas. Namun

demikian faktor kemungkinan tersebut sudah lebih sempit dari faktor kemungkinan yang pertama.



Gambar 2.7 Dua Satelit Menghasilkan Kemungkinan Posisi Pada Tepian Lingkaran

Dan dari satelit ketiga misalnya didapatkan jarak 13.000 mil sehingga akan didapatkan sebuah bola lagi. Dengan demikian telah didapatkan 3 buah bola yang saling berpotongan seperti pada gambar 2.8. Dari gambar di bawah dapat dilihat bahwa perpotongan ketiga bola tersebut akan menghasilkan dua buah titik dan posisi penerima adalah di salah satu titik potong tersebut. Dengan demikian posisi penerima semakin jelas.



Gambar 2.8 Tiga Satelit Menghasilkan Kemungkinan Posisi Pada Dua Titik

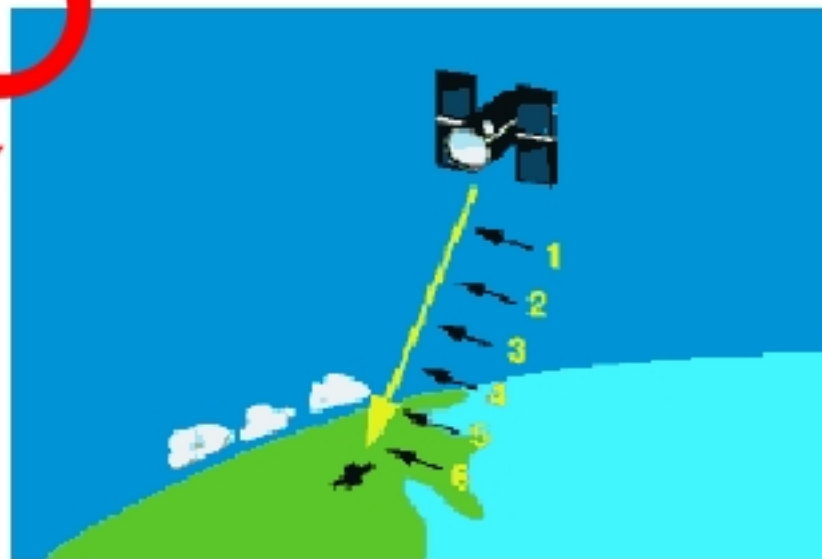


Apabila dilihat uraian di atas maka seharusnya bila diterima satu buah satelit lagi maka akan ditemukan titik posisi penerima yang sebenarnya. Namun hal itu tidak sesederhana itu karena adanya beberapa faktor-faktor yang mempengaruhi misalnya faktor error yang terjadi pada sistem GPS baik yang terjadi karena alam maupun yang memang dengan sengaja dibentuk oleh Amerika untuk kepentingan militer.

#### 2.4.3. Pengukuran jarak penerima dengan satelit

Dari titik nomor satu telah dimengerti bagaimana GPS dapat menentukan posisi seseorang secara tepat di bumi. Namun yang masih menjadi pertanyaan adalah bagaimana caranya mendapatkan jarak yang tepat antara penerima dengan satelit-satelit tersebut. Hal itu dapat dijelaskan sebagai berikut.

Sebagai contoh, apabila sebuah mobil berangkat dari kota X pada jam 6 pagi dan tiba di kota Y pada jam 7 pagi itu juga. Mobil berjalan dengan kecepatan rata-rata 100 km/jam. Maka waktu yang diperlukan adalah 1 jam. Jarak kota X dengan kota Y adalah 100 km/jam  $\times$  1 jam = 100 km/Jam.



Gambar 2.9 Mencari Jarak Antara Satelit Dengan Penerima

Prinsip kerja dari GPS juga seperti itu. Dalam hal ini kecepatan yang dipakai adalah kecepatan *microwave* di udara yang di anggap  $3 \cdot 10^8$  m/s yang merupakan konstanta kecepatan cahaya. Yang menjadi masalah di sini adalah mengenai waktunya. Untuk mengatasi masalah waktu ini, mula-mula satelit GPS mengirimkan sinyal informasi, dimana di dalam sinyal informasi tersebut terdapat data yang diperlukan untuk sistem GPS. Yang paling penting adalah informasi tentang kapan sinyal tersebut dikirimkan. Sedangkan di pesawat penerima akan mendecode sinyal informasi itu dan mengambil informasi tentang waktu pengiriman sinyal seperti pada gambar 2.9. Kemudian informasi waktu tersebut dikurangkan kepada waktu sekarang yang ada di dalam penerima GPS. Akibatnya didapatkan selang waktu pengiriman dari satelit ke penerima GPS.

Untuk dapat menghasilkan perhitungan yang tepat maka antara satelit GPS dan penerima GPS harus mempunyai sistem pewaktu yang tepat sehingga data yang dikirimkan dapat tepat saat ketika diterima. Agar mempunyai ketepatan waktu yang sangat tinggi maka pada masing-masing satelit GPS menggunakan jam Atom. GPS *receiver* yang ada di bumi akan mensinkronkan waktu yang dimiliki dengan waktu pada satelit. Dengan demikian maka ketepatan waktu GPS *receiver* dengan satelitnya menjadi tepat sehingga penghitungan selang waktu pengiriman dan penerimaan dapat diketahui dengan tepat pula.

#### 2.4.4. Mendapatkan waktu yang tepat

Dalam pembahasan sebelumnya telah dikatakan bahwa untuk dapat mengakses informasi yang dipancarkan oleh satelit GPS harus disinkronkan terlebih dahulu waktu penerimaan dengan waktu tiap-tiap satelit. Setelah waktu GPS *receiver* sinkron dengan satelit GPS maka untuk selanjutnya yang bekerja adalah waktu yang ada di dalam GPS *receiver* itu sendiri untuk mendecode sinyal informasi yang lain.

Untuk dapat menghitung jarak antara *receiver* dengan satelit GPS maka diperlukan waktu gelombang radio yang dipancarkan oleh satelit yang bersangkutan seperti pada persamaan 2.2. Untuk mendapatkan waktu yang sebenarnya maka harus dikurangi waktu yang diterima (waktu pengiriman) dengan waktu yang ada di dalam *receiver* saat itu. Dengan demikian ini menunjukkan bahwa GPS *receiver* memerlukan suatu sistem *timing* yang benar-benar tepat, karena perbedaan sedikit saja akan menimbulkan kesalahan sampai ratusan kilometer.

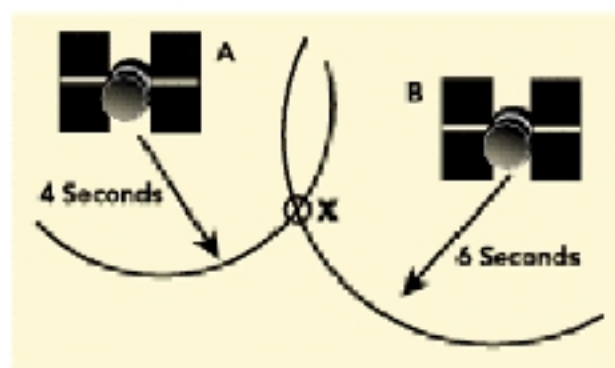


Gambar 2.10 Mendapatkan Waktu Yang Tepat



Masalah pewaktuan pada satelit GPS sudah tidak perlu lagi diragukan karena satelit GPS menggunakan jam atom sebagai sistem pewaktu-nya. Selain itu apabila sampai terjadi pergeseran (*offset*) maka akan di koreksi oleh stasiun yang ada di bumi. Untuk GPS *receiver* amat tidak mungkin menggunakan jam atom. Karena harga dari jam atom itu sendiri sangat mahal (USD 50.000 sampai USD 100.000). Oleh sebab itu sistem yang telah ada haruslah selalu mengadakan perhitungan terhadap error yang terjadi di setiap saat. Hasil yang didapatkan kemudian merupakan *offset* bagi GPS *receiver*. Selanjutnya *offset* tadi dikurangkan kepada perhitungan waktu yang sebelumnya., sehingga akan didapatkan sistem pewaktu yang setaraf dengan jam atom.

Sekarang akan dijelaskan terjadinya error seperti itu dan cara mengatasinya. Dianggap GPS bekerja secara 2 dimensi untuk mempermudah penjelasan. Suatu misal GPS *receiver* menerima sinyal informasi dari GPS A dan B. Dari satelit A sinyal diterima dalam waktu 4 detik dan dari satelit B sinyal diterima dalam waktu 6 detik. Kedua satelit akan membentuk dua buah lingkaran yang berpotongan di X yang mana X merupakan posisi penerima yang sebenarnya seperti pada gambar 2.11



Gambar 2.11 Titik Potong Yang Terjadi Oleh 2 Buah Satelit Dalam Sistem 2 Dimensi

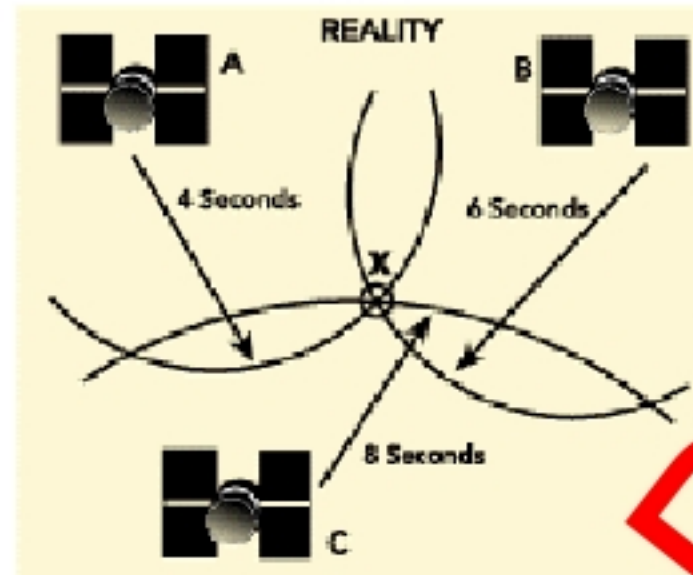
Bila terjadi terjadi pewaktu yang ada pada GPS *receiver* lebih lambat satu detik terhadap waktu GPS, akibatnya waktu yang dibutuhkan oleh sinyal dari satelit A menjadi 5 detik dan sinyal dari satelit B menjadi 7 detik. Hal itu tentu saja akan menyebabkan titik posisi yang dihitung akan bergeser. Misalkan saja titik bergeser ke XX.

Dari gambar 2.12 nampak *offset* yang terjadi adalah jarak dari X ke XX. Dengan menggunakan perhitungan matematika dapat dilakukan dengan cara mengurangi hasil yang dihitung pertama dengan *offset* yang didapatkan sehingga diperoleh sebuah titik X.



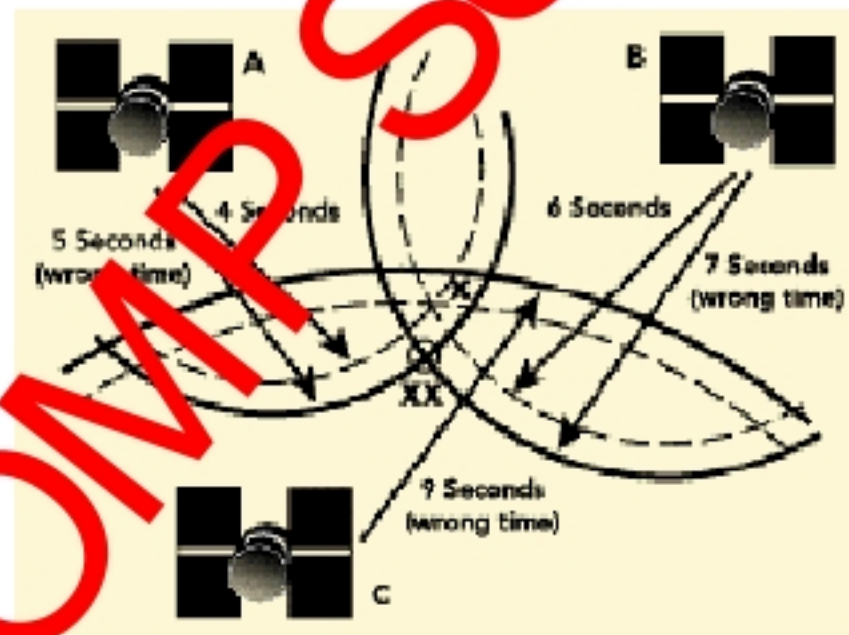
Gambar 2.12 Bergesernya Posisi Dari X ke XX Akibat Perhitungan Waktu Yang Salah

Kemudian diambil contoh lagi apabila perhitungan dilakukan dengan menggunakan tiga buah satelit, satelit A, satelit B dan satelit C. Dengan adanya ketiga satelit tersebut maka akan didapatkan tiga buah lingkaran yang berpotongan di satu titik. Seperti pada gambar 2.13 misalnya sebut titik itu sebagai X. Titik X akan menjadi titik potong yang sebenarnya, artinya secara ideal (tanpa gangguan) maka X merupakan titik koordinat dimana penerima berada saat itu.



Gambar 2.13 Titik Potong Yang Terjadi Oleh 3 Buah Satelit Dalam Sistem 2 Dimensi

Bila terjadi kesalahan offset sebesar satu detik pada masing-masing satelit maka akan didapatkan gambar seperti dibawah ini :



Gambar 2.14 Pergeseran Posisi Dari X ke XX Akibat Perhitungan Waktu Yang Salah

Dampak bahwa terjadi offset sebesar X sampai XX yang selanjutnya dipakai untuk mengoreksi hasil perhitungan yang sedang dilakukan. Untuk selanjutnya garis error tebal pada gambar di atas disebut sebagai "pseudorange" yang digunakan dalam dunia GPS untuk mengatakan pengukuran yang mengandung kesalahan.



Titik XX merupakan perpotongan antara lingkaran A dan B. Dengan demikian lingkaran C tidak dapat memotong titik itu. Keadaan yang seperti inilah yang dapat digunakan sebagai *error indicator* bagi komputer bahwa kesalahan telah terjadi dan untuk selanjutnya dikoreksi. Pengkoreksian dapat dilakukan dengan berbagai cara menurut teori-teori matematika yang ada. Pengkoreksian dapat dilakukan dengan cara mengurangi nilai yang salah dengan *offset* yang ada sehingga didapatkan hasil yang benar.

#### 2.4.5. Menentukan posisi satelit di angkasa

Dalam pembahasan sebelumnya telah diketahui bagaimana posisi seseorang di atas bumi ini dapat dicari, yaitu dengan mencari titik potong – titik potong lingkaran yang dibuat oleh satelit-satelit GPS terhadap posisi orang itu berada. Namun dalam semuanya itu akan timbul pertanyaan, “Bagaimana dapat dicari titik potong satelit-satelit tersebut apabila tidak diketahui posisi satelit-satelit tersebut?”. Pertanyaan itu tepat sekali karena dengan hanya mengetahui jarak seseorang dengan satelit-satelit GPS maka hal itu akan menjadi sesuatu yang percuma karena tidak akan diperoleh posisi orang itu hanya dengan data-data tersebut. Jawaban atas pertanyaan itu dapat dijelaskan pada paragraf-paragraf dibawah ini.

Satelit GPS untuk pertama kali diluncurkan oleh Departemen Pertahanan Amerika Serikat pada bulan Maret 1994 sebanyak 24 satelit ke dalam orbitnya masing-masing. Jarak orbit satelit-satelit tersebut adalah 11.000 mil dari bumi dan telah diatur sedemikian rupa sehingga dalam posisi manapun di bumi, minimal akan terlihat 5 buah satelit.

Amerika menempatkan satelit-satelit tersebut pada suatu koordinat. Dimana posisi dari satelit-satelit tersebut telah diprogramkan pada setiap penerima GPS di bumi. Sehingga bila mengetahui nomor serial dari satelit yang tertangkap dapat diketahui juga posisi dari satelit-satelit tersebut. Dengan demikian posisi penerima dapat semakin nyata dan dapat dicari.

Selain itu, untuk menjamin kebenaran informasi yang diterima pemerintah Amerika juga melakukan pengecekan secara kesinambungan terhadap posisi satelit tersebut dari stasiun-stasiun pemantau di bumi.



Gambar 2.15 Pemantauan Satelit Oleh Stasiun Bumi

Dengan menggunakan radar yang sangat teliti maka bujur, lintang dan ketinggian satelit-satelit tersebut dapat selalu dipantau. Kesalahan atau penyimpangan-penyimpangan yang terjadi dalam sistem ini disebut *ephemeris error*, karena hal itu akan mempengaruhi posisi orbit dari satelit-satelit tersebut.

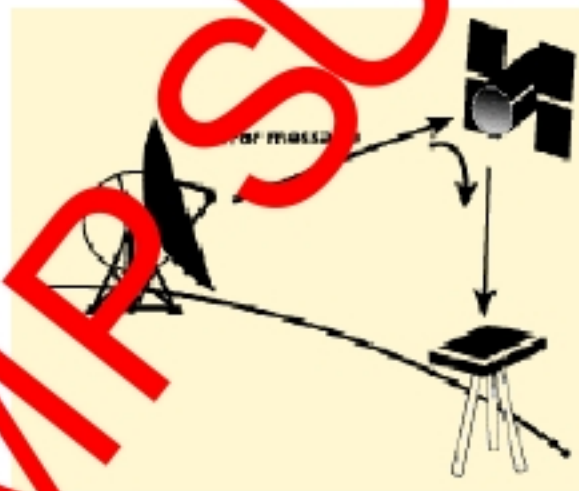
Penyimpangan orbit satelit GPS dapat terjadi karena beberapa faktor antara lain :

- Adanya gaya tarik yang dimiliki oleh bulan dan bumi.

- Adanya radiasi dari matahari yang akhirnya menyebabkan satelit-satelit GPS ini menyimpang sedikit dari orbitnya.

Sebenarnya penyimpangan ini tidak terlalu besar, tetapi jika menginginkan posisi yang tepat sekali maka faktor error tersebut harus diperhitungkan.

Setelah faktor kesalahan itu diketahui dan dihitung maka stasiun pemantau di bumi harus memberitahu tiap-tiap GPS *receiver* di bumi sehingga mereka dapat mengoreksi sinyal informasi yang didapat. Cara paling mudah yang dapat dilakukan oleh stasiun pemantau, yaitu mengirimkan kembali informasi error itu kepada satelit GPS dan kemudian informasi itu dikirimkan secara otomatis ke bumi dan diterima oleh tiap-tiap *receiver* di bumi seperti pada gambar 2.16.



Gambar 2.16 Pengiriman Kembali Informasi Error ke Satelit GPS

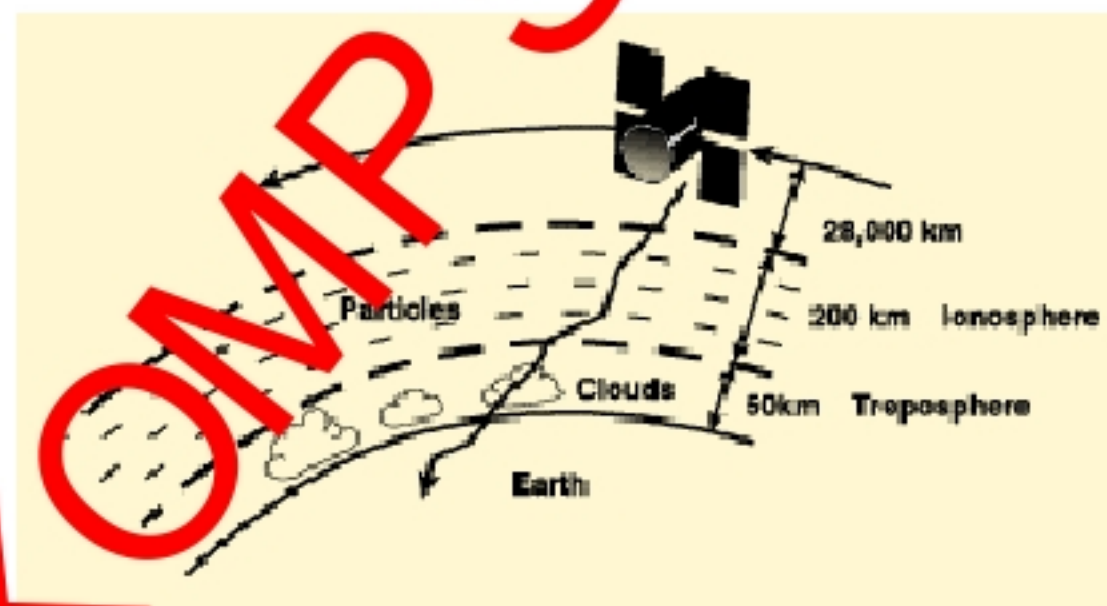
Dengan demikian, satelit GPS tidak lagi semata-mata merupakan suatu konstelasi yang memancarkan sinyal *pseudorandom* untuk perhitungan waktu yang tepat saja melainkan juga memancarkan sinyal-sinyal informasi navigasi dan error-error yang terjadi selama satelit berada di orbitnya.



#### 2.4.6. Error dalam GPS

Dalam pembahasan-pembahasan di atas telah dipelajari cara kerja GPS dan ternyata sistem GPS kelihatan sangat sederhana. Hal itu memang benar secara teoretis yaitu apabila segalanya berjalan secara sempurna tanpa ada hambatan dan gangguan. Selain itu juga telah diketahui bahwa untuk mengukur jarak penerima dengan suatu satelit GPS dapat dilakukan dengan persamaan 2.2. Rumus itu hanya akan berlaku jika sinyal GPS berjalan melewati suatu medium hampa udara sehingga tidak ada hambatan sama sekali. Sedangkan dalam kenyataannya hal itu tidak demikian karena untuk mengirimkan sinyal tersebut, sinyal harus melewati lapisan atmosfer yang berlapis-lapis, misalnya sinyal harus melewati lapisan *Troposfer* yang banyak mengandung uap air. Hal ini tentu saja akan menghambat perjalanan sinyal, akibatnya kesalahan pun terjadi. Error yang lain dapat disebabkan juga oleh keadaan alam pada lapisan *Ionosfer* yang berada pada ketinggian 50 sampai 500 km dari permukaan bumi, dan mengandung banyak sekali partikel yang ter-ionisasi. Karena yang dikandung adalah partikel-partikel bermuatan maka lapisan ini akan menjadi masalah yang cukup berat untuk sinyal GPS, karena lapisan ini dapat menyebabkan sinyal GPS membelok-belok tidak menentu yang ditimbulkan oleh medan magnet dari lapisan *Ionosfer*. Error lain bisa disebabkan juga oleh lapisan *Troposfer* yang merupakan lapisan paling bawah dari atmosfer bumi. Lapisan ini yang bersentuhan langsung dengan permukaan bumi. Pada lapisan ini semua peristiwa cuaca terjadi seperti awan, hujan, badai, angin dan lain sebagainya. Oleh karena itu lapisan ini banyak sekali mengandung uap air. Dengan adanya uap air dan awan maka perjalanan sinyal GPS menjadi lebih terhambat lagi. Akibatnya *delay* yang terjadi akan semakin besar.

Penyebab *error* lainnya disebut dengan *multipath error*, yakni *error* yang disebabkan karena pantulan-pantulan sinyal GPS dengan benda-benda yang ada di bumi sehingga penerima menjadi sulit untuk membedakan mana sinyal yang datang secara langsung dari satelit ke *receiver* dengan sinyal yang datang ke *receiver* melalui pantulan seperti pada gambar 2.17. Hal ini mirip dengan efek *ghosting* pada sistem televisi. GDOP (*Geometric Delusion of Point*) merupakan suatu faktor yang dapat menyebabkan kesalahan sebagai akibat dari struktur geometri yang terjadi antar satelit-satelit GPS terhadap pesawat penerima di bumi. *Error* yang lain dapat juga terjadi karena kesalahan pada satelit sendiri ataupun kemampuan dari satelit yakni *Selective Availability* (SA) yang sengaja diciptakan oleh Amerika agar pihak-pihak lain tidak dapat memanfaatkan teknologi ini..



Gambar 2.17 Error GPS Karena Lapisan Atmosfer

## 2.5 Protokol Komunikasi Garmin *Text Out*

Kehadiran GPS dalam dunia navigasi terus berkembang sejalan dengan perkembangan teknologi. Dengan adanya teknologi mikroprosesor dan otomatisasi peralatan semakin berkembang dan diperbaharui setiap saat. Demikian pula dengan *GPS receiver*, yang merupakan suatu alat yang dapat digunakan sebagai alat sistem navigasi. Dengan demikian muncul suatu ide, yaitu menggabungkan peralatan ini dengan peralatan-peralatan lain, sebagai contoh *auto pilot* pada pesawat terbang. Dengan menggunakan autopilot maka pesawat yang dioperasikan sudah tidak perlu lagi dikemudikan oleh seorang pilot.

Dengan beraneka ragamnya peralatan-peralatan yang dapat dihubungkan dengan sebuah *GPS receiver* maka harus dibentuk suatu protokol komunikasi yang standart sehingga suatu *GPS receiver* dapat berhubungan dengan peralatan-peralatan yang lain, dan suatu peralatan dapat berhubungan dengan berbagai macam *GPS receiver – GPS receiver* yang ada di pasaran. Salah satunya dikembangkan oleh Garmin dimana *GPS Receiver* ini mampu memberikan output berupa *text out* dimana didalamnya terdapat beberapa informasi yang diperlukan.

Sistem ini bekerja pada 4800 bps dengan sistem pengiriman secara data paket. Selain itu semua data yang dikirimkan sudah bukan berupa data byte langsung melainkan berupa kode-kode ASCII teks yang dapat dicetak. Berikut adalah format dari protokol komunikasi ini :

@050615081410S0715340E11242672g044+00053E0000N0000U0000



Dari format diatas dapat dilihat bahwa informasi sebuah lokasi diwakilkan dalam pasangan data S (*South*) atau lintang utara dan E (*East*) atau bujur timur. Sehingga dapat kita diterjemahkan sebagai sebuah lokasi yang berada pada  $07^{\circ}15.140$  lintang utara dan  $112^{\circ}42.672$  bujur timur.

## 2.6 Microcontroller AT89C51



Gambar 2.18 Diagram Pin AT89C51

Atmel AT89C51 merupakan bagian dari keluarga *microcontroller* MCS-51 dengan 4 K byte Flash PEROM (*Programmable and Erasable Read Only Memory*).

AT89C51 dengan teknologi *nonvolatile memory*, yaitu isi memory tersebut dapat

diisi ulang ataupun dihapus berkali-kali. *Microcontroller* AT89C51 mempunyai 40

pin dimana fungsi antara pin yang satu berbeda dengan pin yang lainnya seperti pada

gambar 2.18.

*Microcontroller AT89C51* yang mempunyai beberapa karakteristik

sebagai berikut :

- *Oscilator internal* dan rangkaian pewaktu.
- RAM *internal* 128 byte (on chip).
- Empat buah *programmable port I/O*, masing-masing terdiri atas 8 buah jalur I/O.
- 8 input kontrol : T0, T1, INT0, INT1, RESET, Tx, Rx dan EA.
- 4 output kontrol : RD, WR, ALE dan PSEN (aktif low kecuali ALE).
- Dua buah *timer/counter* 16 bit.
- Enam buah jalur interupsi (3 buah interupsi eksternal dan 3 buah interupsi internal).
- Kemampuan melaksanakan operasi perbandingan, pembagian dan operasi Boolean.
- Kecepatan pelaksanaan instruksi per siklus 1 mikrodetik pada frekuensi 12 MHz.
- Memiliki 3 (tiga) level penguncian pada program di memory.

Tabel 2.7 Penjelasan pin-pin AT89C51

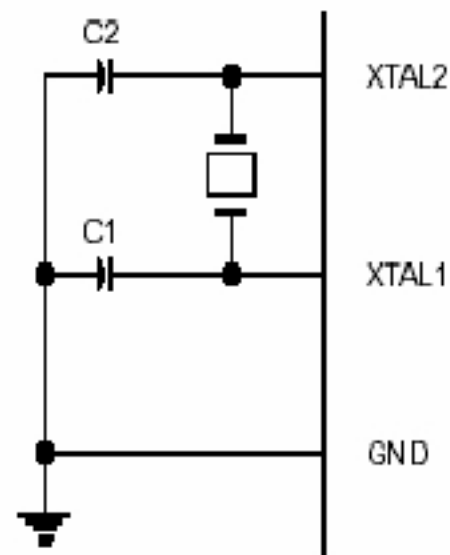
ALE/P	Pulsed output <i>Address Latch Enable</i> , untuk menahan alamat memory eksternal selama pelaksanaan instruksi
EA	<i>External akses active low</i> , untuk mengakses memory program. Bila diberi logika high, <i>microcontroller</i> akan melaksanakan instruksi dari ROM / EPROM ketika isi <i>program counter</i> kurang dari 8192
INT0	Interrupt interupsi eksternal 0, aktif low
INT1	Interrupt instruksi eksternal 1, aktif low
PSEN	<i>Program Stroke Enable</i> , merupakan sinyal pengontrol yang membolehkan program memory eksternal masuk ke dalam bus selama proses pemberian / pengambilan instruksi (fetching)
P0.0 – P0.7	Merupakan port paralel 8 bit <i>open drain</i> dua arah. Bila digunakan untuk mengakses Memory luar, port ini akan memultipleks alamat

	Memory dengan data
P1.0 – P1.7	Merupakan port paralel 8 bit dua arah ( <i>bidirectional</i> ) yang dapat digunakan untuk berbagai keperluan ( <i>general purpose</i> )
P2.0 – P2.7	Merupakan port paralel 2 (P2) selebar 8 bit dua arah ( <i>bidirectional</i> ). Port 2 ini mengirim byte alamat bila dilakukan pengaksesan Memory eksternal
P3.0 – P3.7	Sebagai I/O seperti P1 dan P2 dan dapat berfungsi sebagai spesial port seperti : RXD(P3.0), TXD(P3.1), INT0(P3.2), INT1(P3.3), T0(P3.4), T1(P3.5), WR(P3.6), RD (P3.7)
RXD (P3.0)	Pin penerima data serial
TXD (P3.1)	Pin pengiriman data serial
INT0 (P3.2)	Pin External Interrupt 0
INT1 (P3.3)	Pin External Interrupt 1
T0 (P3.4)	Pin External Timer 0 <i>Input</i>
T1 (P3.5)	Pin External Timer 1 <i>Input</i>
WR (P3.6)	<i>Write Strobe</i> , untuk mengakses Memory data eksternal
RD (P3.7)	<i>Read Strobe</i> , untuk mengakses Memory data eksternal
XTAL1	<i>Input</i> ke rangkaian <i>internal Oscillator</i>
XTAL2	<i>Output</i> rangkaian <i>internal Oscillator</i>
RESET	<i>Reset input, active high</i>

### 2.6.1. Pewaktuan CPU

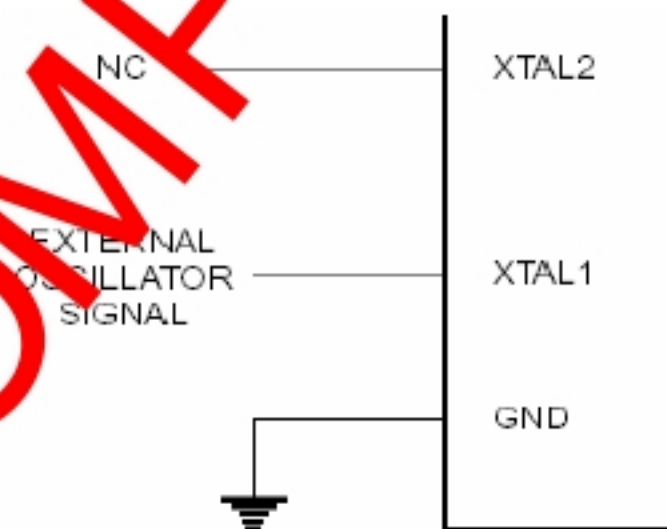
Microcontroller AT89C51 memiliki *oscillator internal (on-chip isolator)* yang dapat digunakan sebagai sumber pewaktu bagi CPU. Untuk menggunakan *oscillator internal* tersebut diperlukan sebuah kristal atau *resonator* keramik antara pin XTAL1 dan pin XTAL2 dan sebuah kapasitor ke ground.





Gambar 2.19 Penggunaan *oscillator* pada *microcontroller* AT89C51

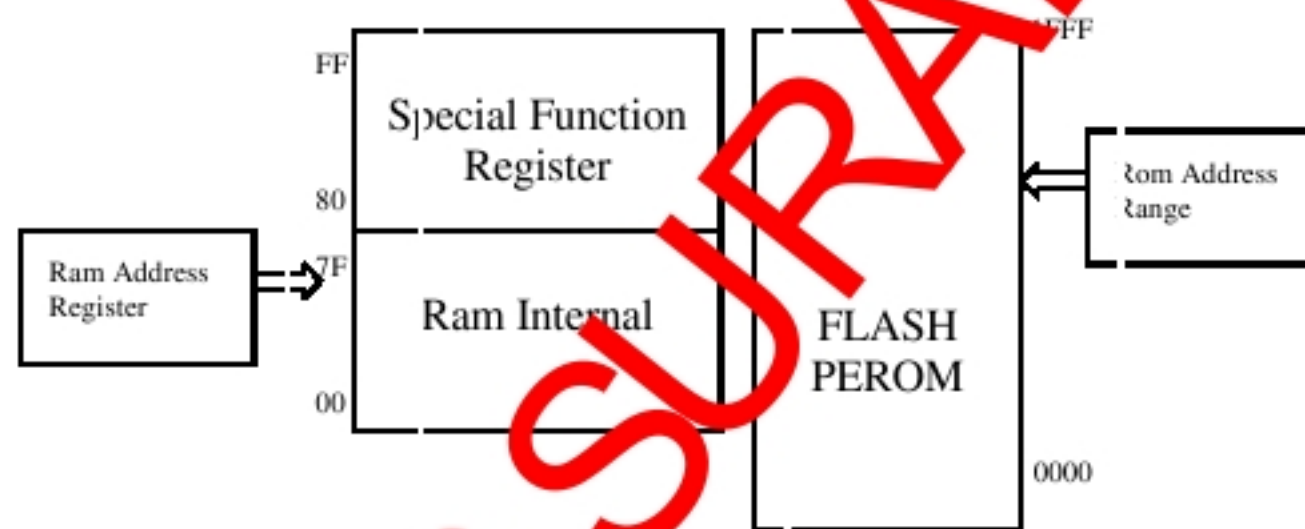
Untuk kristalnya dapat menggunakan frekuensi dari 6 sampai 12 MHz, sedangkan untuk kapasitor yang diperlukan bernilai 27 pF sampai 33 pF seperti pada gambar 2.19. Bila menggunakan *clock eksternal*, maka rangkaianya dihubungkan seperti pada gambar dibawah ini



Gambar 2.20 Penggunaan *Sumber Clock Eksternal* *Microcontroller* AT89C51

### 2.6.2. Struktur Memory AT89C51

*Microcontroller* AT89C51 mempunyai struktur Memory yang terpisah antara RAM *Internal* dan Flash PEROM (*Programmable and Erasable Read Only Memory*). Seperti tampak pada gambar 2.21. RAM *Address Register* terletak pada alamat RAM *internal* sedangkan *Program Address Register* terletak pada alamat Flash PEROM seperti pada gambar dibawah ini.



Gambar 2.21 Alamat RAM Internal dan Flash PEROM

Dalam aplikasi tertentu membutuhkan suatu Memory yang mempunyai kapasitas cukup besar sehingga Memory yang terdapat dalam *microcontroller* ini tidak cukup, sehingga diperlukan suatu memory *eksternal* (RAM atau EEPROM).

AT89C51 memisahkan proses pengalamatan antara Memory program dengan Memory data. Proses pembacaan program dari Memory program ditentukan oleh

sinyal yang keluar dari pin  $\overline{PSEN}$ . Sedangkan proses baca tulis data pada Memory

data ditentukan oleh sinyal yang dikeluarkan dari pin  $\overline{RD}$  dan  $\overline{WR}$ .

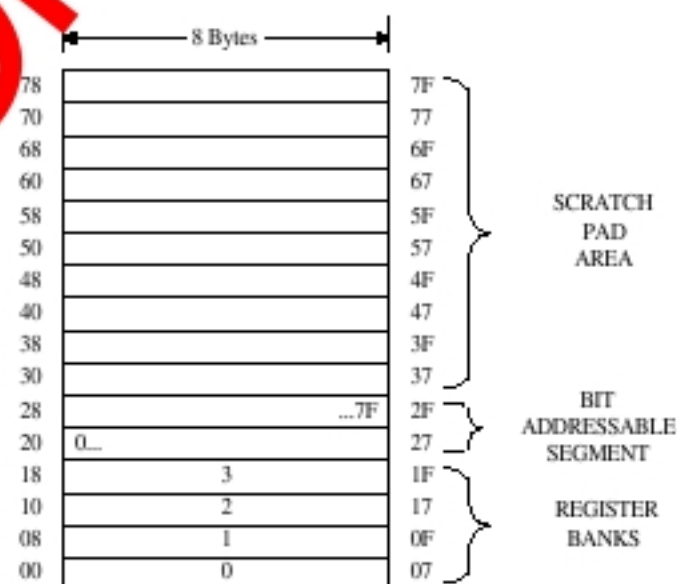
### 2.6.3. RAM internal

Pada *Microcontroller* AT89C51 memiliki 128 byte RAM internal (chip RAM) dengan pengalamatan 128 byte RAM ini dapat dilakukan secara *direct* maupun *indirect* ditambah dengan sejumlah *Special Function Register* (SFR) yang hanya dapat melakukan proses *addressing* secara *direct*.

Pada *internal* Memory terdapat lokasi yang mempunyai fungsi khusus yang dapat digambarkan sebagai berikut :

- Alamat 00H – 1FH merupakan kumpulan dari 4 *bank register* umum yang dapat dipakai dalam program sebagai R0 – R7. Pemilihan *bank register* umum yang digunakan dapat diatur melalui register PSW pada SFR.
- Alamat 20H – 2FH merupakan bagian *internal* Memory yang dialamati secara bit (*Bit Addressable*).
- Sedangkan sisanya mulai alamat 30H - 7FH merupakan Memory bebas.

Untuk memperjelas pemetaan Memory internal data pada MCS-51 dapat dilihat pada gambar berikut.



Gambar 2.22 Pemetaan *Memory Internal* Data



#### 2.6.4. Flash PEROM

AT89C51 mempunyai 4 Kb *Flash PEROM*, yaitu ROM yang dapat ditulis ulang atau dihapus menggunakan sebuah perangkat *programmer*. *Flash PEROM* dalam AT89C51 menggunakan *Atmel's High-Density Non Volatile Technology* yang mempunyai kemampuan untuk ditulis ulang hingga 1000 kali dan berurutan perintah standart MCS51.

Program yang ada pada *Flash PEROM* akan dijalankan pada saat sistem di-reset, pin EA/VP berlogika satu sehingga *microcontroller* aktif berdasarkan program yang ada pada *Flash PEROM*. Namun, jika pin EA/VP berlogika nol, *microcontroller* aktif berdasarkan program yang ada pada *Memory eksternal*.

Untuk keamanan program yang ada dalam *Flash PEROM*, AT89C51 mempunyai fasilitas *Lock Bit Protection* yang terdiri atas :

- a. *Lock Bit 1*, instruksi *MOVX* yang dijalankan dari *Memory eksternal* untuk membaca isi *Flash PEROM* tidak dapat dilakukan.
- b. *Lock Bit 2*, sama dengan *Lock Bit 1*, tetapi isi dari *Flash PEROM* tidak dapat di *verify* oleh AT89C51 *programmer*.
- c. *Lock Bit 3*, sama dengan *Lock Bit 2*, tetapi akses ke *Memory eksternal* tidak dapat dilakukan.

### 2.6.5. SFR

Merupakan bagian *Memory* yang mempunyai fungsi khusus terhadap MCS-51, baik sebagai pengendali perangkat keras maupun perangkat lunak. Register ini menempati alamat internal Memory mulai 80H – FFH. Dari 128 byte lokasi Memory tersebut tidak semuanya merupakan SFR. Lokasi yang lain dapat digunakan sebagai tempat menyimpan data dengan kecepatan proses baca tulis yang lebih cepat dibanding dengan *eksternal Memory*. SFR ini dibagi dalam dua bagian besar yaitu yang dapat diakses per bit dan yang tidak dapat diakses per bit. SFR dapat diakses per bit artinya adalah pengaksesan register tersebut dapat dimanipulasi per bit. Sedangkan SFR yang tidak dapat diakses per bit harus dimanipulasi datanya secara keseluruhan. Pemetaan dari SFR ini dapat dilihat pada gambar dibawah ini.

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PWS							D7
C8								CF
C0								C7
B8	IP							BF
B0								B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90								97
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH			PCON	87

↑  
Bit Addressable

Gambar 2.23 Pemetaan SFR MCS-51

Dari setiap bit register tersebut mempunyai nilai atau fungsi tersendiri yang dapat dijelaskan sebagai berikut

a. *Register PSW (Program Status Word)*

CY	AC	F0	RS1	RS0	OV	-	P
----	----	----	-----	-----	----	---	---

CY : PSW.7 *Carry flag.*

AC : PSW.6 *Carry flag untuk operasi BCD (Auxiliary Carry Flag).*

F0 : PSW.5 *Flag untuk kegunaan umum.*

RS1 : PSW.4 Untuk memilih *bank register* umum bit ke 1.

RS0 : PSW.3 Untuk memilih *bank register* umum bit ke 0.

OV : PSW.2 *Overflow flag.*

- : PSW.1 Tidak digunakan.

P : PSW.0 *Parity flag* yang proses set-resetnya dilakukan oleh perangkat keras yang menunjukkan data pada akumulator bernilai ganjil atau genap.

*Register ini dapat diakses secara bit (Bit Addressable). PSW.3 dan PSW.4 atau RS0 dan RS1 merupakan kombinasi bit yang menentukan bank register yang dipakai. Kombinasi tersebut dapat dilihat pada tabel dibawah ini.*

Tabel 2.8. Kombinasi RS0 dan RS1 sebagai pemilih *Bank Register*

RS1	RS0	Bank Register	Alamat
0	0	0	00H – 07H
0	1	1	08H - 0FH
1	0	2	10H – 17H
1	1	3	18H - 1FH



b. Register PCON (*Power Control Register*)

SMOD	-	-	-	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

SMOD : Pengali dua kecepatan transmisi data serial (*Baud Rate*).

- : Tidak digunakan.

- : Tidak digunakan

- : Tidak digunakan

GF1 : *Flag* untuk kegunaan umum.

GF0 : *Flag* untuk kegunaan umum.

PD : Untuk mengaktifkan mode *Power Down*.

IDL : Untuk mengaktifkan *mode Idle*.

c. Register IE (*Interrupt Enable Register*)

EA	-	-	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

EA : IE.7 Untuk mengaktifkan semua interupsi.

- : IE.6 Tidak digunakan.

- : IE.5 Tidak digunakan.

ES : IE.4 Mengaktifkan interupsi komunikasi serial.

ET1 : IE.3 Mengaktifkan interupsi timer 1.

EX1 : IE.2 Mengaktifkan interupsi eksternal 1.

ET0 : IE.1 Mengaktifkan interupsi timer 0.

EX0 : IE.0 Mengaktifkan interupsi eksternal 0.

Register ini dapat diakses secara bit (*bit addressable*).

d. Register IP (*Interrupt Priority Register*)

-	-	-	PS	PT1	PX1	PT0	PX0
---	---	---	----	-----	-----	-----	-----

- : IP.7 Tidak digunakan.
- : IP.6 Tidak digunakan.
- : IP.5 Tidak digunakan.
- PS : IP.4 Menentukan prioritas interupsi komunikasi serial.
- PT1 : IP.3 Menentukan prioritas interupsi timer 1.
- PX1 : IP.2 Menentukan prioritas interupsi eksternal 1.
- PT0 : IP.1 Menentukan prioritas interupsi timer 0.
- PX0 : IP.0 Menentukan prioritas interupsi eksternal 0.

Register ini dapat diakses secara bit (*bit addressable*).

e. Register TCON (*Timer/Counter Control Register*)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

- TF1 : TCON.7 Penunjuk jika terjadi interupsi timer 1.
- TR1 : TCON.6 Pengontrol timer 1 dari *software*.
- TF0 : TCON.5 Penunjuk jika terjadi interupsi timer 0.
- TR0 : TCON.4 Pengontrol timer 0 dari *software*.
- IE1 : TCON.3 Penunjuk jika terjadi interupsi eksternal 1.
- IT1 : TCON.2 Pengontrol interupsi eksternal 1 dari *software*.
- IE0 : TCON.1 Penunjuk jika terjadi interupsi eksternal 0.
- IT0 : TCON.0 Pengontrol interupsi eksternal 0 dari *software*.

Register ini dapat diakses secara bit (*bit addressable*).

f. Register TMOD (*Timer/Counter Mode Control Register*)

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

GATE : Untuk menentukan *input* interupsi dari *hardware* atau *software*.

C/T : Untuk memilih *Counter* atau *Timer*.

M1 : Untuk memilih *mode timer*.

M0 : Untuk memilih *mode timer*.

M1 dan M0 merupakan kombinasi bit yang menentukan *mode timer* yang dipakai.

Tabel 2.9 Kombinasi M0 dan M1 Sebagai Pemilih *Mode Timer*

M1	M0	Mode
0	0	13 bit <i>Timer</i>
0	1	16 bit <i>Timer / Counter</i>
1	0	(Timer 0) 8 bit <i>Timer / Counter</i> . TL0 dikontrol Timer 0, TH0 dikontrol Timer 1
1	1	(Timer 0) <i>Timer / Counter</i> 1 berhenti

g. Register SCON (*Serial Port Control Register*)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0 : SCON.7 Untuk memilih mode komunikasi serial.

SM1 : SCON.6 Untuk memilih mode komunikasi serial.

SM2 : SCON.5 Untuk komunikasi multi prosesor.

REN : SCON.4 Pengontrol penerima data dari *software*.

TB8 : SCON.3 Tempat data ke 9 yang akan dikirim.

RB8 : SCON.2 Tempat data ke 9 yang akan diterima.

TI : SCON.1 Penunjuk jika terjadi pengiriman data.



RI : SCON.0 Penunjuk jika terjadi penerimaan data.

### 2.6.6. Interupsi

AT89C51 mempunyai lima macam interupsi yang dapat diatur kegunaannya pada register IE, masing-masing interupsi dapat dibagi menjadi 3 bagian yaitu :

#### a. *Eksternal* Interupsi

*Eksternal* interupsi dapat dilakukan melalui pin  $\overline{INT0}$  dan  $\overline{INT1}$ . Jika signal pada pin  $\overline{INT0}$  dan  $\overline{INT1}$  aktif maka *hardware* akan mengeset bit IE0 dan IE1 pada *register* TCON.

#### b. *Timer* Interupsi

Timer Interupsi dapat dilakukan melalui pin T0 dan T1. Jika signal pada pin T0 dan T1 aktif maka *hardware* akan mengeset bit TF0 dan TF1 pada *register* TCON.

#### c. *Serial Port* Interupsi

*Serial port* Interupsi dapat dilakukan melalui pin TXD dan RXD. Jika signal pada pin TXD dan RXD aktif maka *hardware* akan mengeset bit TI dan RI pada *register* SCON. Bit TI dan RI pada *register* SCON dapat dimanipulasi dengan *software*.

Tabel 2.10 Level dan lokasi interupsi pada AT89C51

Level	Interupsi	Lokasi
1	IE0	0003H
2	TF0	000BH
3	IE1	0013H
4	TF1	001BH
5	RI + TI	0023H

### 2.6.7. Serial Port MCS-51

MCS-51 mempunyai rangkaian serial yang terpadu dengan memiliki pin TXD digunakan untuk mengirim data secara serial dan pin RXD digunakan untuk menerima data secara serial. Serial port pada MCS-51 mempunyai sistem transmisi data serial dengan tipe *Full Duplex* yaitu dapat mengirim dan menerima data pada waktu yang bersamaan.

Proses penerimaan dan pengiriman data secara *software* dilakukan melalui *register* SBUF. Pengiriman data dilakukan jika *register* SBUF diisi dengan data yang akan dikirim sedangkan penerimaan data akan diletakkan pada *register* SBUF juga.

a. Serial port mempunyai 4 mode pengoperasiannya yaitu :

*Mode 0 :*

Pada *mode 0* data yang dikirim mempunyai lebar 8 bit data. Data yang dikirim pertama kali adalah bit data yang terkecil. Komunikasi secara serial ini dilakukan melalui pin RXD dan pin TXD. Sedangkan *baud rate* yang digunakan adalah 1/12 osilasi dari frekuensi kristal yang digunakan.

*Mode 1 :*

Pada *mode 1* data yang dikirim mempunyai lebar 10 bit. Bit ke-0 adalah *start bit*, kemudian disusul 8 *bit* data, terakhir adalah *stop bit*. Pengiriman data yang pertama kali adalah *bit* data yang terkecil. Pada saat penerimaan, *bit* yang terakhir yaitu *bit stop* akan diletakkan pada bit RB0 pada *register* SCON. *Baud Rate* yang digunakan pada mode ini dapat diubah-ubah.

*Mode 2 :*

Pada *mode 2* data yang dikirim mempunyai lebar 11 bit. Bit ke-0 adalah *start bit*, diikuti dengan 8 *bit* data, 1 bit tambahan data yang biasa disebut *bit* data ke-9, bit yang terakhir adalah *bit stop*. Pada proses pengiriman *bit* data ke-9 diletakkan pada bit TB8 pada *register* SCON. Bit TB8 pada *register* SCON dapat diisi dengan angka 1 atau 0. Biasanya bit data ke-9 menunjukkan *parity check* dari 8 *bit* data. Pada saat penerimaan bit data yang ke-9 diletakkan pada bit RB8 pada *register* SCON. *Baud rate* pada mode ini adalah 1/32 atau 1/64 osilasi dari frekuensi kristal yang digunakan.

*Mode 3 :*

Pada *mode 3* mempunyai kesamaan pada *mode 2*. Data yang dikirim mempunyai lebar 11 *bit*. *Bit* ke-0 adalah *start bit*, kemudian data sebesar 8 *bit*, ditambah satu *bit* data ke-9, terakhir 1 *bit stop bit*. *Baud rate* yang digunakan dapat diubah-ubah.

#### 1. *Baud Rate Serial Port*

*Baud rate* menunjukkan laju kecepatan pengiriman data digital secara seri. Laju *baud rate* sama dengan banyaknya *bit* yang dikirim setiap detik. Pada MCS-51



pengiriman secara serial ditentukan oleh *mode* yang telah ditentukan. *Baud rate* yang digunakan juga bersesuaian dengan mode yang digunakan. *Mode 0* *Baud Rate* yang digunakan adalah :

$$\text{Baud\_Rate} = \frac{\text{Osilasi\_Frekuensi}}{12} \quad (2.3)$$

*Baud Rate* yang digunakan pada *mode 2* tergantung dari bit SMOD yang ada pada *register* PCON. Jika SMOD = 0 *Baud Rate* yang digunakan adalah  $K = \frac{1}{64}$ . Jika SMOD = 1 *Baud Rate* yang digunakan  $K = \frac{1}{32}$ .

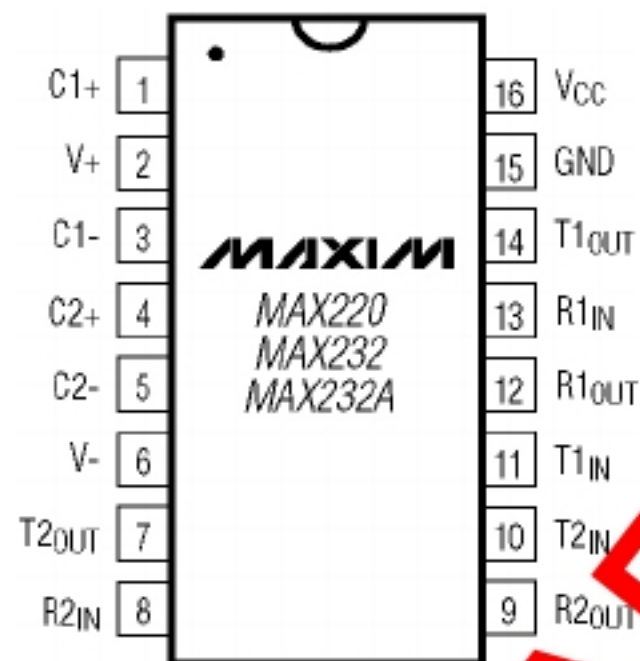
$$\text{Baud\_Rate} = K \times \text{Osilasi\_Frekuensi} \quad (2.4)$$

*Baud rate* yang digunakan untuk *mode 1* dan *3* dapat ditentukan melalui *register* *Timer 1* dan juga bit SMOD pada *register* PCON. Jika SMOD = 0 *Baud Rate* yang digunakan adalah  $K=1$ . Jika SMOD = 1 *Baud Rate* yang digunakan  $K=2$ .

$$\text{Baud\_Rate} = \frac{K}{32} \times \frac{\text{Osilasi\_Frekuensi}}{12 \times (256 - (TH1))} \quad (2.5)$$

## 2.7 IC MAX232

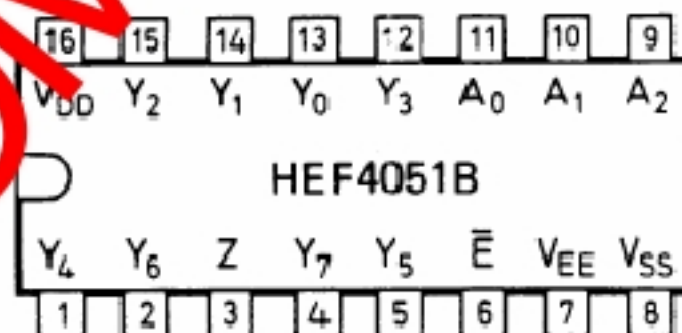
IC MAX232 mempunyai 16 pin dimana fungsi pin yang satu berbeda dengan pin yang lainnya seperti pada gambar 2.24. IC MAX232 digunakan sebagai pengubah tegangan pada *level* RS232 menjadi tegangan pada *level* TTL dan sebaliknya. Pada RS232, 1 bit direpresentasikan antara -3 sampai -25 volt dan untuk 0 bit direpresentasikan antara +3 sampai +5 volt sedangkan antara -3 sampai +3 volt tidak didefinisikan. Dan pada TTL untuk 0 bit dan 1 bit direpresentasikan 0 volt dan 5 volt.



Gambar 2.24 Pin IC MAX232

## 2.8 Multiplexer 4051

HEF 4051 adalah *multiplexer* analog yang memiliki 3 *input address* sebagai selektor dan 8 *output*. IC ini dapat digunakan sebagai selektor apabila kita ingin memilih 1 dari 8 *output* yang kita inginkan, fungsi pin dan tabel kebenaran dapat dilihat pada gambar dibawah ini (PHILIPS Semiconductors DataSheet, 1995)



Gambar 2.25 Pin IC HEF 4051

Tabel 2.11 Tabel kebenaran MUX 4051

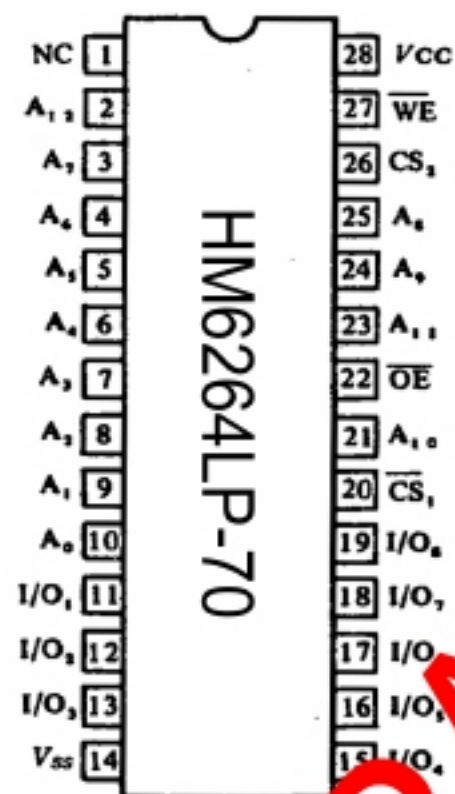
Inputs			Channel ON
A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	
0	0	0	Y <sub>0</sub>
0	0	1	Y <sub>1</sub>
0	1	0	Y <sub>2</sub>
0	1	1	Y <sub>3</sub>
1	0	0	Y <sub>4</sub>
1	0	1	Y <sub>5</sub>
1	1	0	Y <sub>6</sub>
1	1	1	Y <sub>7</sub>

## 2.9 RAM (Random Access Memory) 6264

RAM adalah *Memory* semikonduktor *volatile* (Memory yang akan hilang datanya apabila daya dimatikan) yang serbenergi digunakan dalam mikrokomputer modern untuk menyimpan data dan program.

RAM 6264 dengan *Memory* yang menyimpan 65.536 bit yang tersusun dalam 8.192 *byte*, yang masing-masing berisi 8 bit. Waktu pengaksesan data pada RAM 6264 adalah 35-70 ns. Fungsi dari masing-masing pin dan tabel kebenaran seperti pada gambar di bawah ini (Suwito, 1996)





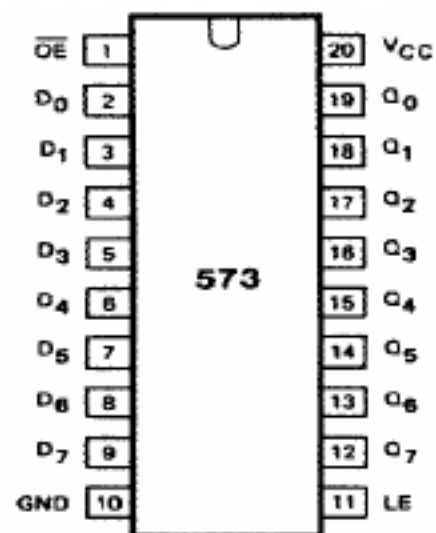
Gambar 2.26 Pin RAM 6264

Tabel 2.12 Tabel kebenaran RAM 6264

MODE	CE#	OE#	WE#	OPERASI I/O
Standby	H	X	X	High - Z
Output disable	L	H	H	High - Z
Read	L	L	H	Data Out
Write	L	X	L	Data In

### 2.10 IC 74HC573 (Octal D Latch)

IC 74HC573 adalah IC *latching* (pengunci / penahan) 8 bit dengan 3 kondisi output seperti pada gambar 2.27. IC 74HC573 digunakan untuk pengalamatan Memory pada sistem mikrokomputer dengan menggunakan sistem pengalamatan secara bergantian (*multiplex addressing*) yaitu memultiplex data dan byte alamat rendah (*low byte addressing*). Fungsi pin dan tabel kebenaran dari IC 74HC573 dapat dilihat pada gambar dibawah ini ( S, Suwito. 1996 )



Gambar 2.27 Pin IC 74HC573

Tabel 2.13 Tabel Kebenaran IC 74HC573

OUTPUT Enable	Latch Enable	O	Output $\bar{O}$
L	H	H	H
L	H	L	L
L	X	X	$Q_0$
H	X	X	Z

L = Low State, H = High State, X = Don't Care  
 Z = High Impedance State  
 $Q_0$  = Previous Condition of O

## 2.11 Komunikasi Data Serial

Komunikasi data berarti pengiriman data antara dua komputer yaitu antara sebuah komputer dengan terminal, atau antara terminal dengan terminal yang lain.

Komunikasi data dapat dilakukan dengan dua cara yaitu paralel dan serial.

Dalam *transfer* data paralel, sering 8 atau lebih jalur (konduktor kabel) digunakan

untuk men-transfer data ke suatu *device* yang berjarak hanya beberapa kaki. Contoh

*transfer* paralel adalah printer dan *hard disk* yang menggunakan kabel dengan banyak

jalur. Meskipun dalam kasus-kasus seperti ini banyak data bisa ditransfer dalam waktu singkat dengan menggunakan banyak kabel yang disusun paralel, tetapi jaraknya tidak bisa jauh dan biaya relatif lebih mahal. Untuk men-transfer data ke suatu *device* yang terletak lebih jauh maka, digunakan metode serial. Dalam komunikasi serial, data dikirim satu *bit* dalam suatu waktu, berbeda dengan komunikasi paralel, dimana data dikirim satu *byte* atau lebih dalam suatu waktu. Kelebihan metode serial ini adalah selain ia dapat digunakan dalam jarak yang jauh, ia juga memerlukan biaya yang lebih murah jika dibandingkan dengan metode paralel.

Komunikasi data serial menggunakan metode, *asinkronous* dan *sinkronous*. Metode *sinkronous* men-transfer suatu blok data (karakter) pada suatu waktu sedangkan *asinkronous* men-transfer suatu *byte* tunggal pada suatu waktu. Adalah mungkin untuk membuat *software* untuk digunakan dengan metode-metode di atas, tetapi programnya bisa membosankan dan panjang. Karena itu, ada chip-chip khusus yang dibuat oleh banyak pabrik untuk komunikasi data serial. Chip-chip ini secara umum dikenal sebagai UART (*universal asynchronous receiver-transmitter*) dan USART (*universal synchronous-asynchronous receiver-transmitter*). Pada komputer IBM PC *compatible port* serial atau sering dikenal sebagai COM menggunakan standart UART 8250.

### 11.1. Transmisi *half-* dan *full-duplex*

Dalam transmisi data jika data bisa ditransmisikan dan diterima, itu disebut transmisi *duplex*. Ini lain dari transmisi *simplex* seperti printer, yang mana computer hanya mengirim data. Transmisi *duplex* bisa *half* atau *full duplex*, tergantung pada



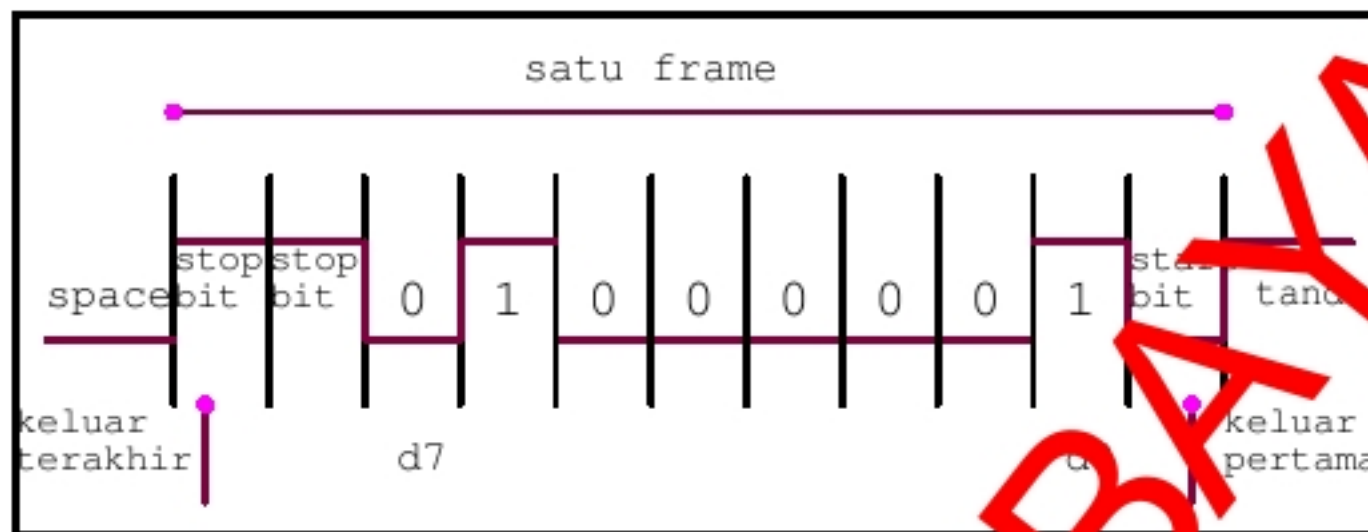
apakah transfer datanya bisa sekaligus atau tidak. Jika data ditransmisikan satu arah pada suatu waktu, ini disebut *half duplex*. Jika data bisa melewati dua arah pada waktu yang sama, itulah *full duplex*. *Full duplex* membutuhkan dua kabel konduktor (selain ground), satu untuk transmisi dan satu untuk penerimaan, agar bisa mentransfer dan menerima data secara sekaligus.

### 2.11.2. Komunikasi serial *asinkronous* dan data framing

Data yang masuk pada akhir penerimaan dari jalur data dalam transfer data serial semuanya dalam kode 0 dan 1; sangatlah sulit untuk memahami data kecuali pengirim dan penerima menyetujui seperangkat peraturan yaitu sebuah protokol yang dipakai, data yang dikirimkan, banyak *bit constitute* dalam sebuah karakter, dan kapan awal dan akhir dari data.

### 2.11.3. *Start bit* dan *stop bit*

Komunikasi data serial *asinkronous* digunakan secara luas untuk transmisi berorientasi-karakter dan transfer data berorientasi-blok menggunakan metode *sinkronous*. Dalam metode *asinkronous*, setiap karakter diletakkan antara *start* dan *stop bit* yang disebut *framing*. *Start* dan *stop bit* mengirimkan data *framing* untuk komunikasi *asinkronous* data seperti ASCII karakter, dipaketkan di antara sebuah *start bit* dan sebuah *stop bit*. *Start bit* terdiri dari satu *bit* tetapi *stop bit* bisa satu atau dua *bit*. *Start bit* selalu '0' (*low*) dan *stop bit* adalah '1' (*high*). Sebagai contoh, lihat Gambar 2.28 dimana karakter ASCII "A", biner 0100 0001, di-frame di antara sebuah *start bit* dan sebuah *single stop bit*. Dalam hal ini LSB dikirim terlebih dahulu.



Gambar 2.28 Framing ASCII "A" (11D)

Pada Gambar 2.28, bila tidak ada transfer, levelnya bernilai 1 (*high*), yang disebut sebagai *mark* dan bernilai 0 (*low*) disebut sebagai *space*. Perhatikan bahwa transmisi mulai dengan *start bit* yang diikuti oleh D0 (*LSB*), kemudian sisa bit-bit sampai MSB (D7), dan akhirnya, satu *stop bit* yang menunjukkan akhir dari karakter "A".

Dalam komunikasi serial *asinkronous*, *chip-chip peripheral* dan modem dapat diprogram untuk data selebar 7 atau 8 *bit*. Ini sebagai tambahan dari jumlah *stop bit*, 1 atau 2. Sementara dalam sistem yang lebih lama karakter-karakter ASCII adalah 7-bit, karena adanya karakter *extended ASCII*, bit ke 8 menjadi *common*.

Dalam beberapa sistem lama, disebabkan kelambatan peralatan mekanik yang menerima, 2 *stop bit* digunakan untuk memberikan peralatan tersebut cukup waktu untuk mengorganisasi dirinya sendiri sebelum transmisi dari *byte* berikutnya. Tetapi, dalam PC-PC modern penggunaan 1 *stop bit* adalah umum. Dengan asumsi bahwa kita mentransfer file text dari karakter ASCII menggunakan 2 *stop bit*, kita punya total 11 *bit* untuk tiap karakter yaitu 8 *bit* untuk kode ASCII-nya, dan 1 dan 2 *bit*

masing-masing untuk *start* dan *stop bit*. Oleh karena itu, untuk setiap karakter 8-bit ada 3 bit ekstra, atau lebih dari 30%.

Dalam beberapa sistem untuk menangani integritas data, *bit parity* dan *byte* karakter dimasukkan dalam *data frame*. Ini berarti bahwa untuk tiap karakter (7- atau 8-bit, tergantung pada sistem) kita punya *bit parity* tunggal sebagai tambahan dari *start bit* dan *stop bit*. *Bit parity* adalah ganjil atau genap. Dalam kasus *bit parity*-ganjil jumlah bit data, termasuk *bit parity*, punya *number* ganjil dari 1-an. Serupa dengan itu, dalam sebuah *bit parity*-genap jumlah total bit termasuk *bit parity*, adalah genap. Sebagai contoh, karakter ASCII "A" (biner 01000001), punya 0 untuk bit *parity*-genap. Chip-chip UART memungkinkan pemrograman bit *parity*-ganjil, *parity*-genap, dan pilihan tanpa-*parity* sebagaimana yang akan kita lihat dalam pembahasan berikutnya. Jika suatu sistem membutuhkan *parity*, *bit parity* ditransmisikan setelah MSB dan diikuti oleh *stop bit*.

#### 2.11.4. Laju transfer data

Laju transfer data dalam komunikasi data serial dinyatakan dalam *bps* (bits per second). Istilah lain yang digunakan untuk *bps* adalah *baud rate*. Bagaimanapun, *baud* dan *bps rate* tidaklah benar-benar sama. Ini disebabkan fakta bahwa *baud rate* adalah istilah modem dan didefinisikan sebagai jumlah perubahan sinyal per detik.

Dalam modem, ada kejadian-kejadian ketika suatu perubahan tunggal dari sinyal mentransfer beberapa bit data. Sejauh kabel konduktor yang menjadi fokus bahasan, *baud rate* dan *bps* adalah sama, dan dengan alasan ini dalam hal ini kita bisa saling mempertukarkan *term bps* dan *baud*.



Laju sistem data dari suatu sistem computer tergantung pada port-port komunikasi yang ada dalam sistem itu. Sebagai contoh, IBM PC/XT awal bisa mentransfer data pada laju 100 sampai 9600 bps. Tetapi dalam tahun-tahun terakhir, PC, PS, dan computer kompatibel 80x86 mentransfer data pada laju sebesar 19.200 bps. Harus dicatat bahwa dalam komunikasi data serial asinkronous, baud rate biasanya dibatasi sampai 100.000 bps.

#### 2.11.5. RS232 dan Standart-Standart I/O Serial Lain

Untuk memungkinkan kompatibilitas di antara peralatan komunikasi data yang dibuat oleh berbagai pabrik, standart *interfacing* yang disebut RS232 telah ditetapkan oleh EIA (*Electronics Industries Association*) pada tahun 1960. Pada tahun 1963 standart itu dimodifikasi dan dikenal sebagai RS232A. RS232B dan RS232C masing-masing diperkenalkan pada 1965 dan 1969. Dalam buku ini secara sederhana kita menyebutnya sebagai RS232. Dewasa ini, RS232 adalah standart *interfacing I/O* serial yang paling banyak digunakan. Tetapi, karena standart itu ditetapkan lama sebelum kemunculan keluarga TTL logic, level tegangan *input* dan *outputnya* tidak kompatibel TTL. Dalam RS232, '1' direpresentasikan dengan -3 s/d -25 V, sedangkan bit 0 adalah +3 s/d +25 V, sehingga -3 sampai +3 tidak didefinisikan. Karena itu, untuk menghubungkan suatu RS232 ke sistem berbasis mikroprosesor kita harus menggunakan konverter tegangan seperti MAX232 dan MAX232 untuk mengkonversi level logika TTL menjadi level tegangan RS232, dan sebaliknya. ( Bergsman. 1994 )