

BAB 2

TEKNOLOGI PADA SISTEM PEMANTAUAN POSISI DAN TINGKAT PENCEMARAN UDARA BERGERAK

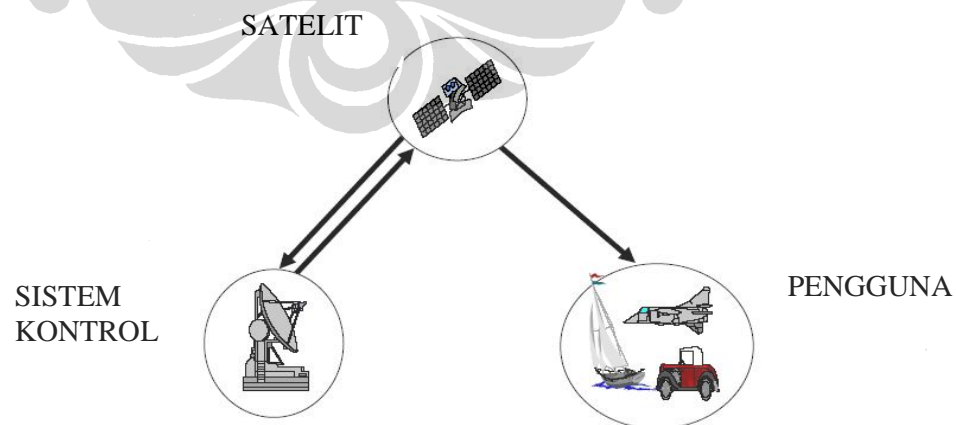
2.1 GLOBAL POSITIONING SYSTEM

GPS (*Global Positioning System*) adalah sistem satelit navigasi dan penentuan posisi menggunakan satelit. Nama formalnya adalah **NAVSTAR GPS**, kependekan dari “*NAVigation Satellite Timing and Ranging Global Positioning System*”. Sistem yang banyak digunakan oleh banyak orang sekaligus dalam segala cuaca ini, didesain untuk memberikan posisi dan kecepatan tiga dimensi yang teliti, dan juga informasi mengenai waktu, secara kontinu di seluruh dunia.

Arsitektur dari system GPS disetujui oleh Departemen Pertahanan Amerika Serikat pada tahun 1973. Satelit pertama diluncurkan pada tahun 1978, dan secara resmi system GPS dinyatakan operasional pada tahun 1994.[1]

2.1.1 Segmen Penyusun Sistem

Sistem GPS terdiri atas tiga segmen utama, yaitu segmen angkasa (*space segment*), segmen system control (*control system segment*), dan segmen pengguna (*user segment*). Ketiga segmen GPS tersebut digambarkan secara skematik pada Gambar 2.1.



Gambar 2.1 Sistem Penentuan posisi global (GPS).[1]

1) Segmen Angkasa

Segmen angkasa terdiri dari 24 buah satelit GPS yang secara kontinyu memancarkan sinyal – sinyal yang membawa data kode dan pesan navigasi yang berguna untuk penentuan posisi, kecepatan dan waktu. Satelit-satelit tersebut ditempatkan pada enam bidang orbit dengan periode orbit 12 jam dan ketinggian orbit 22.200 km diatas permukaan bumi. Keenam orbit tersebut memiliki jarak yang sama dengan berinklinasi 55° terhadap ekuator dengan masing-masing orbit ditempati oleh empat buah satelit dengan jarak antar satelit yang tidak sama.

2) Segmen Sistem Kontrol

Segmen sistem kontrol terdiri atas *Master Control Station* (MCS), *Ground Antenna Station* (GAS), *Prelaunch Compatibility Station* (PCS) dan beberapa *Monitor Station* (MS) yang berfungsi untuk mengontrol dan memonitor pergerakan satelit.

3) Segmen Pengguna

Segmen pengguna terdiri dari para pengguna satelit GPS yang ada di darat, laut maupun udara. Dalam hal ini *receiver* GPS dibutuhkan untuk menerima dan memproses sinyal-sinyal dari GPS untuk digunakan dalam penentuan posisi, kecepatan, dan waktu.

2.1.2 Penentuan Posisi Absolut Dengan GPS

Penentuan posisi dengan GPS adalah penentuan posisi dengan tiga dimensi yang dinyatakan dalam sistem koordinat kartesian (X,Y, Z) dalam datum WGS (*World Geodetic System*) 1984. Untuk keperluan tertentu, koordinat kartesian tersebut dapat dikonversi ke dalam koordinat geodetic (ϕ , λ , h). Titik yang akan ditentukan posisinya dapat diam (*static positioning*) maupun bergerak (*kinematic positioning*). Penentuan posisi absolut merupakan metode penentuan posisi yang paling mendasar dan paling banyak digunakan untuk aplikasi-aplikasi yang tidak memerlukan tingkat ketelitian posisi yang tinggi dan tersedia secara instan (*real-time*) seperti pada aplikasi navigasi wahana bergerak (darat, laut dan udara).

2.1.2.1. Prinsip Penentuan Posisi Absolut dengan GPS

Prinsip dasar penentuan posisi absolut dengan GPS adalah dengan reseksi jarak ke beberapa satelit GPS sekaligus yang koordinatnya telah diketahui [1]. Pada penentuan posisi absolut dengan data *pseudorange*, jarak pengamat (*receiver*) ke satelit GPS ditentukan dengan mengubah besarnya waktu tempuh sinyal GPS dari satelit *receiver* pengamat.

Waktu tempuh ditentukan dengan menggunakan teknik korelasi kode (*code correlation technique*) dimana sinyal GPS yang datang dikorelasikan dengan sinyal replika yang diformulasikan dalam *receiver*. Jarak dari *receiver* ke pengamat kemudian dapat ditentukan dengan mengalikan waktu tempuh dengan kecepatan cahaya. Karena ada perbedaan waktu dengan satelit dan jam *receiver*, maka data jarak yang diperoleh bukan merupakan jarak sebenarnya melainkan jarak *pseudorange* yang persamaannya dapat dirumuskan sebagai berikut[1]:

$$\rho = r + c(t_u - \delta_t) \quad (2.1)$$

$$r = c(T_u - T_s) = c\Delta t \quad (2.2)$$

dengan:

ρ = jarak *pseudorange*

r = jarak geometric

c = kecepatan cahaya

T_s = waktu GPS pada saat sinyal meninggalkan satelit

T_u = Perbedaan GPS pada saat sinyal mencapai *receiver*

t_u = Perbedaan waktu jam *receiver* dengan waktu GPS

δ_t = Perbedaan waktu jam satelit dengan waktu GPS

Untuk mendapatkan posisi tiga dimensi (X, Y, Z) maka terdapat empat parameter yang harus diestimasi / dipecahkan yaitu :

- Parameter kootdinat (X, Y, Z)
- Parameter kesalahan jam *receiver* GPS

Oleh sebab itu untuk memecahkan keempat parameter tersebut dibutuhkan pengamatan terhadap minimal empat buah satelit secara simultan yang dirumuskan dalam persamaan berikut [1].

$$\rho_1 = \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \quad (2.3)$$

$$\rho_2 = \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \quad (2.4)$$

$$\rho_3 = \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + ct_u \quad (2.5)$$

$$\rho_4 = \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + ct_u \quad (2.6)$$

dengan:

ρ = jarak *pseudorange*

x_i, y_i = koordinat satelit i

x_u, y_u = koordinat pengamat

ct_u = koreksi kesalahan jam *receiver*

2.1.2.2. Ketelitian Posisi Absolut

Ketelitian posisi absolut GPS sangat bergantung pada tingkat ketelitian data *pseudorange* serta geometri dari satelit pada saat pengukuran [1]

$$\text{Ketelitian posisi GPS} = \text{Geometri satelit} \times \text{Ketelitian Pseudorange}$$

Faktor-faktor yang mempengaruhi ketelitian penentuan posisi dengan GPS adalah sebagai berikut:

- Satelit, seperti kesalahan ephemeris (orbit) dan jam satelit,
- Medium propagasi, seperti bias ionosfer dan bias troposfer yang mempengaruhi kecepatan (memperlambat) dan arah perlambatan sinyal GPS
- *Receiver* GPS, seperti kesalahan jam *receiver*, kesalahan yang terkait dengan antenna dan noise (derau). Kesalahan-kesalahan ini bergantung pada kualitas dari *receiver* GPS dan berbanding lurus dengan harga dari *receiver* GPS, semakin tinggi harga *receiver*, semakin tinggi kualitasnya.
- Lingkungan sekitar *receiver* GPS, seperti *multipath* yaitu fenomena dimana sinyal GPS yang tiba diantena *receiver* GPS merupakan resultan dari sinyal yang langsung dari GPS dan satelit dipantulkan oleh benda-benda disekeliling *receiver* GPS.

Dalam kaitannya dengan ketelitian penentuan posisi dengan GPS, terdapat dua level ketelitian yang diberikan oleh GPS, yaitu SPS (*Standard Positioning Service*) dan PPS (*Precise Positioning Service*). SPS merupakan layanan standar yang diberikan oleh GPS kepada siapa saja tanpa dipungut biaya. Tingkat

ketelitian yang diberikan oleh layanan ini adalah ± 100 m pada saat kebijakan SA (*Selective Availability*) masih berlaku dan ± 20 m setelah kebijakan SA dihapus (1 Mei 2000, 00:00 EDT). Sedangkan PPS merupakan jenis layanan yang hanya dikhususkan untuk pihak militer Amerika dan pihak-pihak lain yang diijinkan dengan tingkat ketelitian yang lebih tinggi dari tingkat ketelitian SPS.

2.1.3 Format Data Keluaran GPS

Receiver GPS memiliki format keluaran sebanyak lima jenis, yaitu NMEA 0180, NMEA 0182, NMEA 0183, AVIATION, dan PLOTTING[2]. Format data tersebut ditetapkan oleh NMEA (*National Maritime Electronic Association*) dan dapat dikoneksikan ke komputer melalui port komunikasi serial dengan menggunakan kabel RS-232.

Data keluaran dengan format NMEA 0183 berbentuk kalimat (*string*) yang merupakan rangkaian karakter ASCII 8 bit. Setiap kalimat diawali dengan satu karakter '\$', dua karakter *Talker ID*, tiga karakter *Sentence ID*, dan diikuti oleh *data fields* yang masing-masing dipisahkan oleh koma serta diakhiri oleh *optional checksum* dan karakter *carriage return/line feed* (CR/LF). Jumlah maksimum karakter dihitung dari awal kalimat (\$) sampai dengan akhir kalimat (CR/LF) adalah 32 karakter.

Format data NMEA 0183:

```
$aacc,c—c*hh<CR><LF>
```

Keterangan:

- aa = *Talker ID*, menandakan jenis atau peralatan navigasi yang digunakan,
- ccc = *Sentence ID*, menandakan jenis informasi yang terkandung dalam kalimat,
- c—c* = *data fields*, berisi data-data navigasi hasil pengukuran,
- hh = *operational checksum*, untuk pengecekan kesalahan (*error*) kalimat,
- <CR><LF> = *carriage return/linefeed*, menandakan akhir dari kalimat.

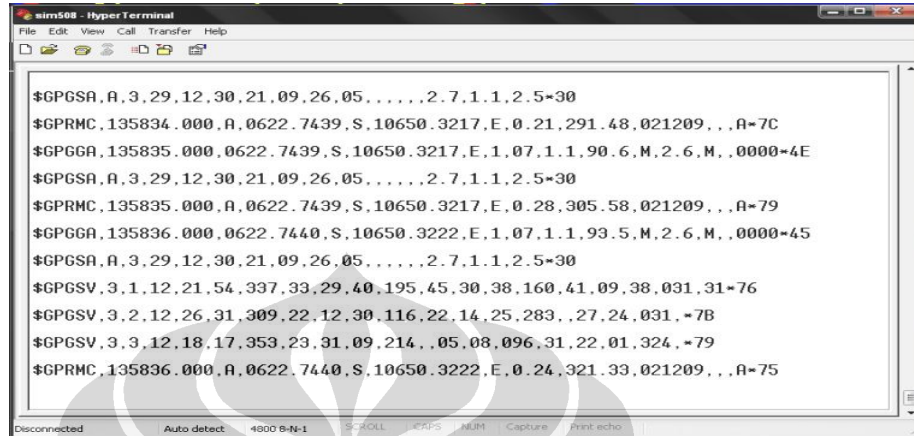
[Sumber:<http://Vancouver-webpages.com/peter/nmeafaq.txt>]

beberapa jenis *Talker ID* yang ada pada spesifikasi NMEA 0183 adalah:

- GP, untuk data keluaran GPS *receiver*,
- LC, untuk data keluaran Loran-C *receiver*,

- OM, untuk data keluaran Omega Navigation *receiver*.

Gambar 2.2 memperlihatkan contoh data keluaran GPS yang diterima melalui komputer dengan menggunakan perangkat lunak hyper terminal.



```

sim588 - Hyper Terminal
File Edit View Call Transfer Help
$GPGSA,A,3,29,12,30,21,09,26,05,,,,,2.7,1.1,2.5*30
$GPRMC,135834.000,A,0622.7439,S,10650.3217,E,0.21,291.48,021209,,A=7C
$GPGGA,135835.000,0622.7439,S,10650.3217,E,1.07,1.1,90.6,M,2.6,M,.0000*4E
$GPGSA,A,3,29,12,30,21,09,26,05,,,,,2.7,1.1,2.5*30
$GPRMC,135835.000,A,0622.7439,S,10650.3217,E,0.28,305.58,021209,,A=79
$GPGGA,135836.000,0622.7440,S,10650.3222,E,1.07,1.1,93.5,M,2.6,M,.0000*45
$GPGSA,A,3,29,12,30,21,09,26,05,,,,,2.7,1.1,2.5*30
$GPGSV,3,1,12,21,54,337,33,29,40,195,45,30,38,160,41,09,38,031,31*76
$GPGSV,3,2,12,26,31,309,22,12,30,116,22,14,25,283,,27,24,031,*7B
$GPGSV,3,3,12,18,17,353,23,31,09,214,,05,08,096,31,22,01,324,*79
$GPRMC,135836.000,A,0622.7440,S,10650.3222,E,0.24,321.33,021209,,A=75
Disconnected Auto detect 4000 Baud Scroll EOL NUM Capture Print echo

```

Gambar 2.2 Data keluaran dari *receiver* GPS yang ditampilkan pada komputer dengan menggunakan hyper terminal

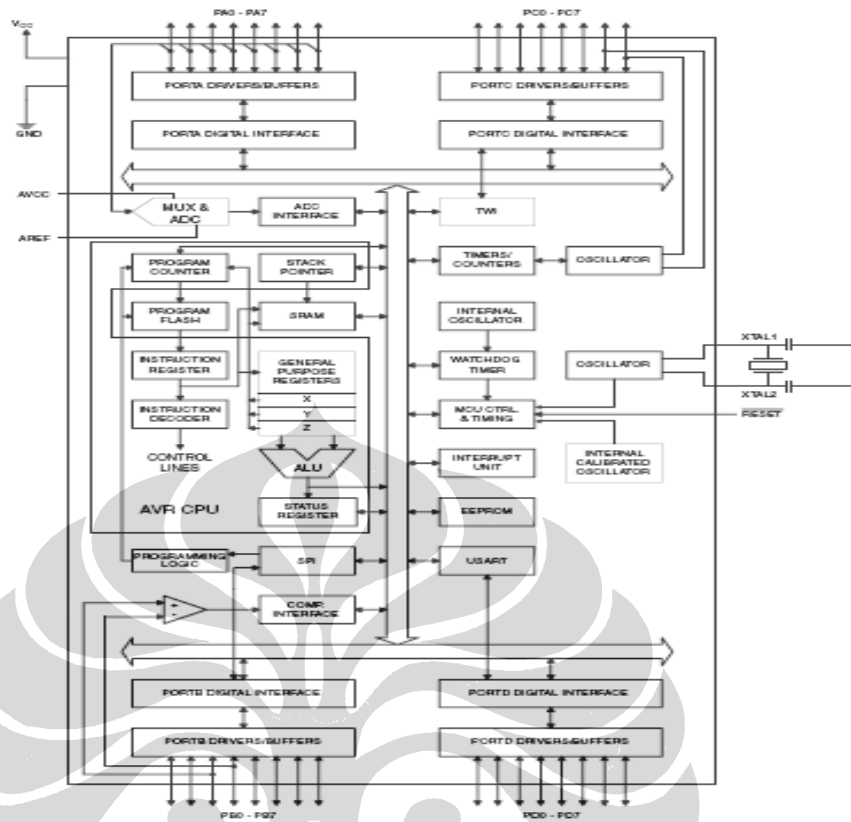
2.2 MIKROKONTROLER ATMEGA8535

2.2.1 Mikrokontroler AVR

Mikrokontroler AVR memiliki arsitektur RISC 8 bit dengan kecepatan maksimal 16 MHz, dimana semua instruksi dikemas dalam kode 16-bit (16-bit word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus klock, berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus klock. Tentu saja itu terjadi karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC (*Reduced Instruktion Set Computing*), sedangkan seri MCS51 berteknologi CISC (*Complex Instruktion Set Computing*). Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang dipergunakan, mereka bisa dikatakan hampir sama.

2.2.2 ARSITEKTUR ATMEGA8535

ATMega8535 mempunyai arsitektur seperti yang diperlihatkan pada Gambar 2.3 berikut



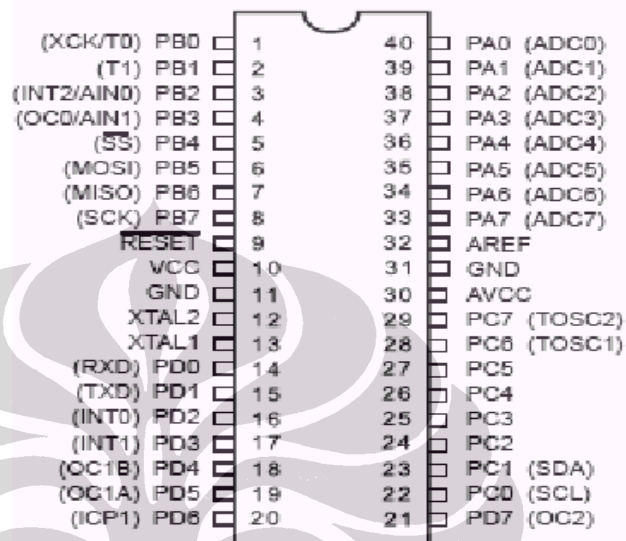
Gambar 2.3 Blok Diagram Fungsional ATmega8535 [5]

Dari Gambar 2.3 tersebut dapat dilihat bahwa ATmega8535 memiliki bagian sekaligus merupakan fitur sebagai berikut:

1. Saluran I/O sebanyak 32 buah yaitu Port A, Port B, Port C, dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog Timer* dengan osilator internal .
6. SRAM sebesar 512 byte.
7. Memori *Flash* sebesar 8 kb dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI
10. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial dengan kecepatan maksimal 2,5 Mbps..

2.2.3 KONFIGURASI PIN ATMEGA8535

Konfigurasi pin ATmega8535 bisa dilihat pada Gambar 2.4 Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega8535 sebagai berikut:



Gambar 2.4 Pin ATmega8535 [5]

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground*.
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu timer/counter, komparator analog dan SPI.
5. Pport C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan *timer Oscillator*.
6. Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

2.2.4 PETA MEMORI

AVR ATmega8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O, dan 512 byte SRAM *Internal*.

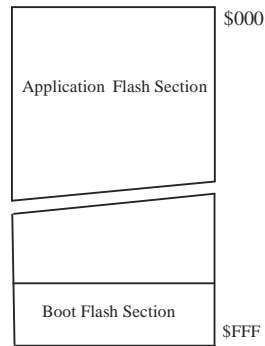
Register keperluan umum menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler mempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler, seperti control register, timer/counter, fungsi-fungsi I/O, dan sebagainya. Alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F. Konfigurasi memori data ditunjukkan pada Gambar 2.5 dibawah ini.

Register Umum	Alamat
R0	\$0000
R1	\$0001
....
R30	\$001E
R31	\$001F
Register I/O	
\$00	\$0020
\$01	\$0021
....
\$3E	\$005E
\$3F	\$005F
SRAM Internal	
	\$0060
	\$0061

	\$025E
	\$025F (RAMEND)

Gambar 2.5 Konfigurasi Memori Data AVR ATmega8535 [5]

Memori program yang terletak dalam Flash PEROM tersusun dalam word atau 2 byte karena setiap instruksi memiliki lebar 16-bit Flash PEROM dengan alamat mulai dari \$000 sampai \$FFF. AVR tersebut memiliki 12-bit *Program Counter* (PC) sehingga mampu mengamati isi Flash. Konfigurasi memori program diperlihatkan pada Gambar 2.6



Gambar 2.6 Memori Program AVR ATmega8535 [5]

Selain itu AVR ATmega8535 juga memiliki memori data berupa EEPROM 8-bit sebanyak 512 byte. Alamat EEPROM dimulai dari \$000 sampai \$1FF.

2.3 KOMUNIKASI SERIAL

2.3.1 Jenis komunikasi data serial

Ada dua macam cara pengiriman (transmisi) data secara serial. Kedua cara tersebut dibedakan oleh sinyal detak (*clock*) yang dipakai untuk mendorong data serial, yang pertama yaitu detak dikirim bersama-sama dengan data serial, cara tersebut dikatakan sebagai transmisi data serial secara *synchron*, sedangkan cara yang kedua yaitu disebut *asynchron* dimana detak dibangkitkan oleh masing-masing system pengirim maupun penerima.

2.3.2 Protokol Serial

Format populer untuk transmisi data kecepatan rendah adalah *asynchronous* protokol. Format tipikal untuk komunikasi *asynchronous* adalah 12 bit. Start bit menginisialisasikan rangkaian pewaktuan. Hal ini dideteksi oleh perubahan dari *High* ke *Low*. Berikutnya adalah delapan bit data, diawali dengan LSB (*Least Significant Bit*) diakhiri dengan MSB (*Most Significant Bit*). Kemudian diikuti oleh opsional bit parity. Terakhir, *line* menjadi *High* untuk satu atau dua bit (*stop bit*), menandakan karakter telah habis. Format ini ditunjukkan pada Gambar 2.7.



Gambar 2.7 Timing Format Asynchronous – UART

2.3.3 Pengaksesan port serial pada PC menggunakan pemrograman Visual Basic 6.0

2.3.3.1 Pengaksesan dengan Menggunakan Kontrol MSComm

Kontrol MSComm menyediakan fasilitas komunikasi antara program aplikasi yang dibuat dengan port serial untuk mengirim atau menerima data melalui port serial. Setiap MSComm hanya menangani satu port serial, sehingga jika ingin menggunakan lebih dari satu port serial, harus menggunakan MSComm sebanyak port serial yang dipakai.[3]

Properti MSComm

Jumlah properti pada MSComm sangat banyak beberapa properti yang sering dipakai adalah sebagai berikut:

- CommPort** : Digunakan untuk menentukan nomor port serial yang akan dipakai.
- Setting** : Digunakan untuk menset nilai baud rate, parity, jumlah bit data, dan jumlah bit stop.
- PortOpen** : Digunakan untuk membuka ataupun menutup port serial yang dihubungkan dengan MSComm ini.
- Input** : Digunakan untuk mengambil data string yang ada pada buffer penerima.
- Output** : Digunakan untuk menulis data string pada buffer kirim.

Berikut adalah contoh penggunaan properti tersebut untuk komunikasi menggunakan modem.

```
Private Sub Form_Load()
    MSComm1.CommPort = 1
    MSComm1.Settings = "4800,N,8,1"
    MSComm1.InputLen = 0
    MSComm1.PortOpen = True
    MSComm1.Output = "ATV1Q0" & Chr$(13)
Do
```

```

DoEvents
Buffer$ = Buffer$ & MSComm1,input
Loop Until InStr(Buffer$, "OK" & vbCRLF)
MSComm1.PortOpen = False
End Sub

```

Kode-kode program pada prosedur di atas melakukan aksi sebagai berikut:

- Port serial yang digunakan adalah COM1
- Setting MSComm adalah baud rate 4800, tanpa paritas, jumlah data 8 bit, dan jumlah bit stop adalah 1 bit.
- Memerintahkan kontrol MSComm membaca seluruh isi buffer ketika menggunakan perintah Input (MSComm1.InputLen = 0)
- Membuka port serial
- Mengirim perintah "ATV1Q0" diikuti ASCII 13 (enter) ke modem
- Menunggu modem mengirimkan jawaban "OK" ke komputer
- Menutup port serial

Even Pada MSComm

MSComm hanya mempunyai satu even saja, yaitu even OnComm. Even OnComm dibangkitkan jika nilai properti dari CommEvent berubah yang mengindikasikan telah terjadi even pada port serial baik even komunikasi maupun even error. Tabel 2.1 dan Tabel 2.2 berikut adalah nilai-nilai dari properti CommEvent. Nilai properti ini tidak tersedia pada saat *design time*, tetapi hanya dapat dibaca pada saat run time.

Table 2.1 Nilai-nilai Properti Even Error pada CommEvent [3]

Konstanta	Keterangan
comEventFrame	Hardware mendeteksi adanya kesalahan framing
comEventRxParity	Hardware mendeteksi adanya kesalahan pariti
comEventRxOver	Buffer penerima mengalami over flow, tidak ada ruang kosong lagi pada buffer penerima
comEventRxFull	Buffer kirim penuh
comEventOverrun	Port mengalami overrun
comEventBreak	Sinyal Break diterima
comEventDCB	Mendapatkan kembali Device Control Block (DCB) dari port serial

Table 2.2 Nilai-nilai Properti Even komunikasi pada *CommEvent* [3]

Konstanta	Keterangan
comEvSend	Jumlah karakter pada buffer kirim lebih sedikit daripada nilai property <i>Sthreshold</i> . Even ini akan dibangkitkan jika nilai pada property <i>Sthreshold</i> tidak diisi '0'.
comEvReceive	Telah diterima karakter sebanyak nilai property <i>Rthreshold</i> . Even ini akan dibangkitkan terus menerus sampai data diambil dari buffer penerima menggunakan perintah input. Even ini akan dibangkitkan jika nilai pada property <i>Rthreshold</i> tidak diisi '0'.
comEvCTS	Terjadi perubahan pada saluran Clear to Send
comEvDSR	Terjadi perubahan pada saluran Data Set Ready
comEvCD	Terjadi perubahan pada saluran Carrier Detect
comEvRing	Terdeteksi adanya sinyal Ring
comEvEOF	Karakter End of file diterima

Berikut adalah contoh penggunaan even *OnComm* untuk komunikasi menggunakan mikrokontroler. Akan dibaca hanya even **comEvReceive** saja, even yang lain diabaikan.

```

Private Static Sub MSComm1_OnComm ()
Dim Buffer As Variant
Slect Case MSComm1.CommEvent
Case comEvReceive
If MSComm1.InBufferCount >= 3 then
Buffer = CStr (MSComm1.Input)
If Mid(Buffer, 1, 1) = "O" Then
If Mid(Buffer, 2, 1) = "K" Then
StatusBar1.Panels("value").Text =
"Value:" & Asc(Mid(Buffer, 3, 1))
StatusBar1.Panels("Status").Text =
"satatus: Connect"
End If
End If
End Slect
End Sub

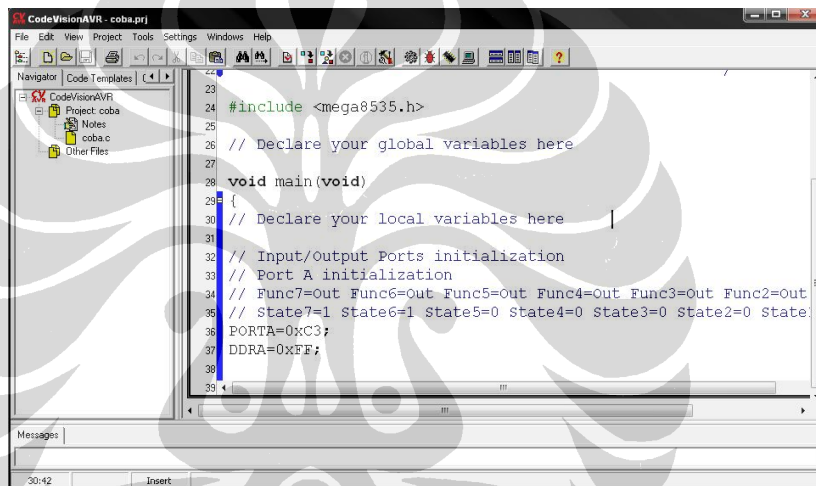
```

Kode-kode program pada prosedur diatas akan melakukan aksi sebagai berikut:

- Mendeteksi even **comEvReceive**, kemudian menentukan apakah sudah diterima tiga buah karakter pada buffer penerima
- Menentukan apakah karakter yang diterima adalah karakter 'OK'. Jika karakter yang diterima adalah karakter 'OK', maka akan diubah nilai 'Value' dan nilai 'Status' pada property panel control *StatusBar1*

2.4 CODEVISION AVR

CodeVisionAVR merupakan sebuah cross-compiler C, Integrated Development Environment (IDE), dan Automatic Program Generator yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, XP dan Vista. Cross-compiler C mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada system embedded. Gambar 2.8 berikut memperlihatkan tampilan CodeVisionAVR.

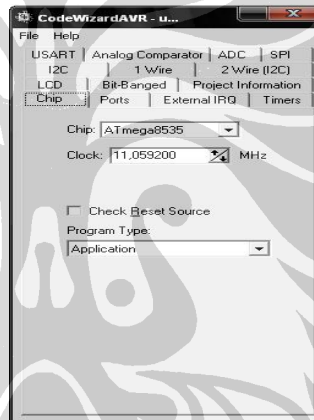


Gambar 2.8 Tampilan CodeVisionAVR

Selain library standar C, CodeVisionAVR juga mempunyai library tertentu untuk Modul LCD alphanumeric, Bus I2C dari Philips, Sensor Suhu LM75 dari National Semiconductor, Real-Time Clock: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas Semiconductor, Protokol 1-Wire dari Maxim/Dallas Semiconductor, Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas Semiconductor, Termometer/Termostat DS1621 dari Maxim/Dallas Semiconductor, EEPROM DS2430 dan DS2433 dari Maxim/Dallas Semiconductor, SPI, Power Management, Delay, dan Konversi ke Kode Gray

CodeVisionAVR juga mempunyai Automatic Program Generator bernama CodeWizardAVR, yang mengizinkan Anda untuk menulis, dalam hitungan menit,

semua instruksi yang diperlukan untuk membuat fungsi-fungsi: Set-up akses memori eksternal, Identifikasi sumber reset untuk chip, Inisialisasi port input/output, Inisialisasi interupsi eksternal, Inisialisasi Timer/Counter, Inisialisasi Watchdog-Timer, Inisialisasi UART (USART) dan komunikasi serial berbasis buffer yang digerakkan oleh interupsi, Inisialisasi Pembanding Analog, Inisialisasi ADC, Inisialisasi Antarmuka SPI, Inisialisasi Antarmuka Two-Wire, Inisialisasi Antarmuka CAN, Inisialisasi Bus I2C, Sensor Suhu LM75, Thermometer/Thermostat DS1621 dan Real-Time Clock PCF8563, PCF8583, DS1302, dan DS1307, Inisialisasi Bus 1-Wire dan Sensor Suhu DS1820, DS18S20, dan Inisialisasi modul LCD. Gambar 2.9 berikut memperlihatkan CodeVisionWizard yang dimiliki oleh CodeVisionAVR.



Gambar 2.9. CodeWizardAVR pada tab Chip

Untuk mendapatkan CVAVR dapat mendownload dari situs <http://www.hpinfotech.ro>.

2.5 INTEGRATED MAPPING VERSI MAPINFO

2.5.1 Teknik Pemrograman *Integrated Mapping*

Dengan teknik pemrograman “*Intergared mapping*”, setiap pengguna dapat mengendalikan mapinfo dengan menggunakan salah satu compiler bahasa pemrograman yang dikuasainya. Dengan implementasi teknik pemrograman ini, aplikasi “*integrated mapping*” terkait (secara operasional) pertama kali akan mengeksekusi sebaris pernyataan yang mengaktifkan (mengkoneksikan) Mapinfo

secara *background* melalui *OLE automation*. Selain itu, dengan “*integrated mapping*”, aplikasi pengguna dapat mengkombinasikan beberapa fungsionalitas pemetaan digital yang telah dimiliki oleh Mapinfo dengan kemampuan manipulasi yang telah dimiliki oleh bahasa pemrograman.[4]

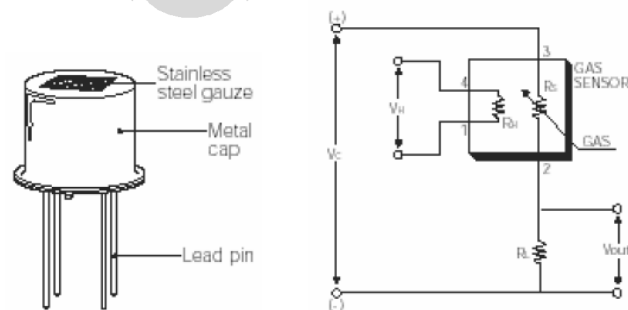
2.5.2 APLIKASI INTEGRATED MAPPING

Pengguna dapat segera membuat aplikasi *integrated mapping* yang sederhana menggunakan salah satu bahasa pemrograman komputer. Di dalam program ini terdapat sebaris kode yang akan mengaktifkan Mapinfo secara *background* tanpa memunculkan *splash screen* seperti biasanya. Kemudian, baris-baris program yang lain dapat memanipulasi Mapinfo dengan cara mengirimkan beberapa string yang merepresentasikan baris-baris kode Mapbasic (dengan menggunakan mekanisme *OLE automation*) ke Mapinfo. Akhirnya Mapinfo akan mengeksekusi string kode-kode MapBasic ini seolah pengguna mengetikannya baris demi baris di dalam MapBasic window.

2.6 GAS SENSOR FIGARO TGS 2600

2.6.1 Figaro TGS 2600

Figaro TGS 2600 adalah transducer utama yang digunakan dalam rangkaian ini, yang merupakan sebuah sensor kimia atau gas sensor. Sensor ini mempunyai nilai resistensi R_s yang akan merubah bila terkena gas dan juga mempunyai sebuah pemanas (*heater*) yang digunakan untuk membersihkan ruangan sensor dari kontaminasi udara luar. Struktur dari sensor terdapat pada Gambar 2.10.



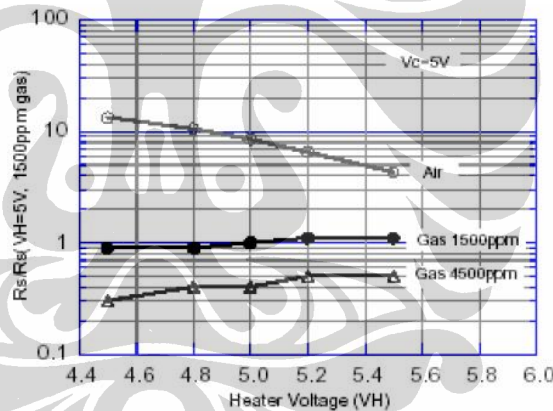
Gambar 2.10 Struktur Sensor Figaro TGS 2600 [6]

Output tegangan pada hambatan R_L (V_{out}) digunakan sebagai masukan pada mikroprosesor. Nilai resistansi R_L dipilih agar konsumsi daya dari sensor (P_S) di bawah batas 15 mW, nilai P_S akan meningkat pada waktu nilai resistansi nilai sensor R_S sama dengan resistansi R_L . nilai P_S dapat dicari berdasarkan persamaan berikut ini.

$$P_S = \frac{(V_c - V_{out})^2}{R_S} \dots\dots\dots(2.7)$$

$$R_S = \frac{(V_c - V_{out})}{V_{out}} \times R_L \dots\dots\dots(2.8)$$

Pemanas pada sensor memerlukan tegangan yang konstan (± 5 Volt DC) agar sinnyal output sensor dapat terjaga keseimbangannya. Karakteristik tegangan pemanas terhadap resistansi sensor terdapat pada Gambar 2.11.



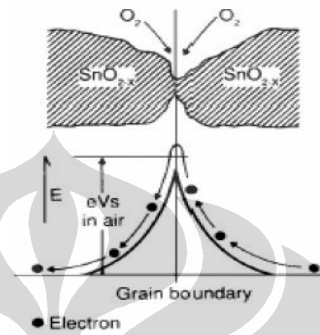
Gambar 2.11 Karakteristik tegangan pemanas terhadap resistansi sensor [6]

2.6.2 Cara Kerja Sensor Gas Figaro Secara Umum

Bahan detector gas dari sensor adalah metal oksida, khususnya senyawa SnO_2 . Ketika Kristal metal oksida (SnO_2) dihangatkan pada temperatur tertentu, oksigen akan diserap pada permukaan Kristal dan oksigen akan bermuatan negative, proses penyerapan oksigen oleh sensor dapat dilihat dari persamaan kimia berikut ini.

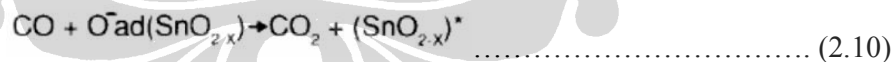


Hal ini disebabkan karena permukaan Kristal mendonorkan elektron pada oksigen yang terdapat pada lapisan luar, sehingga oksigen akan bermuatan negatif dan muatan positif akan terbentuk pada permukaan luar kristal. Tegangan permukaan yang terbentuk akan menghambat laju aliran electron seperti tampak pada ilustrasi Gambar 2.12.

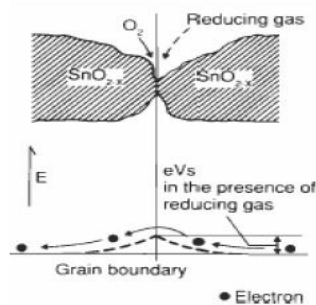


Gambar 2.12 Ilustrasi Penyerapan O₂ Oleh Sensor [6]

Di dalam sensor, arus elektrik mengalir melewati daerah *sambungan* (grain boundary) dari Kristal SnO₂. Pada daerah *sambungan*, penyerapan oksigen mencegah muatan untuk bergerak bebas. Jika konsentrasi gas menurun, proses deoksidasi akan terjadi, rapat permukaan dari muatan negatif oksigen akan berkurang, dan mengakibatkan menurunnya ketinggian penghalang dari daerah *sambungan*, misal terdapat adanya gas CO yang terdeteksi maka persamaan kimianya dapat digambarkan seperti tampak pada persamaan berikut ini.



Dengan menurunnya penghalang maka resistensi sensor akan juga ikut menurun.



Gambar 2.13 Ilustrasi Ketika Terdeteksi Adanya Gas [6]

Hubungan antara resistansi sensor dengan konsentrasi gas pada proses deoksidasi dapat ditunjukkan dengan persamaan:

$$R = A[C]^{-\alpha} \dots\dots\dots(2.11)$$

Dimana: R = Resistansi sensor

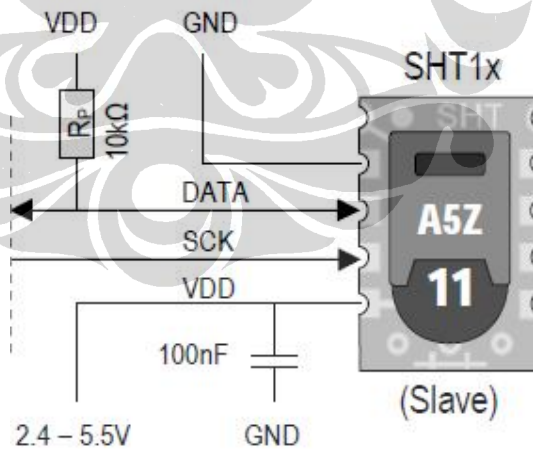
A, α = konstanta

[C] = konsentrasi gas

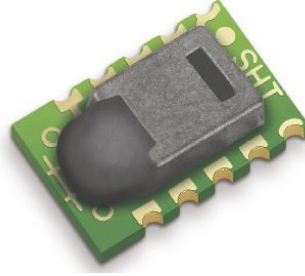
2.7 SENSOR SUHU DAN KELEMBABAN SHT11

SHT11 adalah sensor digital untuk temperatur sekaligus kelembaban pertama di dunia yang diklaim oleh pabrik pembuatnya, **Sensirion Corp.** Mempunyai kisaran pengukuran dari 0- 100% RH, dan akurasi RH absolute +/- 3% RH. Sedangkan akurasi pengukuran temperatur +/- 0.4°C @ 25°C. Sensor ini bekerja dengan interface 2-wire. Aplikasi sensor ini pada data logging, pemancar, automotive, perangkat instrumentasi dan lain sebagainya.

Untuk menghubungkan sensor 2 wire dengan mikrokontroler. Gambar 3.8 berikut merupakan rangkaian sensor kelembaban tipe SHT 11. Sensor SHT 11 ini dikoneksikan pada Port C.1 sebagai transfer data dan pada Port C.2 sebagai clock.



Gambar 2.14 Rangkaian SHT 11. [7]



Gambar 2.15 Bentuk Fisik SHT 11. [7]

2.8 AT (*ATTENTION*) COMMAND

AT Command adalah perintah-perintah yang digunakan untuk komunikasi dengan *handphone* melalui serial port pada komputer, mikrokontroler, dan sebagainya. Dengan penggunaan AT Command, dapat diketahui kondisi dari *handphone*, seperti mengirim pesan, membaca pesan, menambah *item* pada daftar telepon, dan sebagainya.

Perintah-perintah AT *Command* biasanya disediakan oleh *vendor* alat komunikasi yang dibeli atau *website* produsen alat tersebut. Pada tabel di bawah ini menunjukkan perintah AT Command dan keterangannya.

Tabel 2.3 Penulisan AT Command [9]

Perintah	AT Command	Keterangan
<i>Read Command</i>	AT+Cxxx?	Perintah ini digunakan untuk mengetahui nilai parameter yang sedang aktif pada terminal.
<i>Test Command</i>	AT+Cxxx=?	Perintah ini digunakan untuk mengetahui parameter-parameter dan jangkauan nilainya berdasarkan <i>write command</i> atau proses internal.
<i>Write Command</i>	AT+Cxxx=<...>	Perintah ini digunakan untuk mengubah nilai parameter.
<i>Execute Command</i>	AT+Cxxx	Perintah ini untuk mengetahui nilai parameter dari terminal yang tidak dapat diubah nilainya.

Berikut ini pada table 2.4 adalah beberapa perintah AT *Command* yang penting dalam SMS:

Tabel 2.4 AT Command SMS [9]

Command	Fungsi
AT+CMGS	Pengiriman SMS ke <i>handphone</i> lain
AT+CMGR	Membaca isi SMS yang ada pada <i>handphone</i>
AT+CMGD	Menghapus SMS yang ada pada <i>handphone</i>
AT+CGSN	Mengetahui nomor IMEI <i>handphone</i> .

Berikut ini adalah beberapa perintah AT Command yang berhubungan dengan panggilan (*dialing*):

Tabel 2.5 AT Command Dial [9]

Command	Fungsi
ATD<<str>;	Panggilan ke nomor <i>handphone</i> (<i>string</i>). Karakter ";" adalah tanda pengaktifan <i>voice utilty</i> .
ATD<<n>;	Panggilan ke nomor telepon dari <i>phone book</i> dalam lokasi nomor <n>.
ATD<<mem><n>;	Panggilan ke nomor telepon dari buku telepon <mem> lokasi nomor <n>.
ATDL	Panggilan ke nomor telepon panggilan terakhir.

2.9 SHORT MESSAGE SERVICE (SMS)

SMS merupakan sebuah layanan yang diaplikasikan pada jaringan komunikasi tanpa kabel (*wireless network*) yang memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric*. Teknologi yang digunakan adalah *store and forward service*. Pada teknologi ini, SMS yang terkirim akan disimpan sementara pada *server* SMSC (*Short Message Service Center*), kemudian oleh SMSC akan diteruskan ke nomor tujuan setelah dicek terlebih dahulu pada HLR (*Home Location Register*) untuk menentukan apakah ponsel aktif atau tidak. Apabila aktif, pesan SMS akan diteruskan ke nomor tujuan. Sedangkan apabila tidak aktif, SMS akan disimpan sementara di *server* sampai waktu tertentu.

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

2.9.1 Mode Penerimaan dan Pengiriman SMS

- **Mode Teks**

Mode teks merupakan mode pengiriman pesan yang termudah. Pada mode ini, pesan yang dikirim tidak dilakukan konversi. Teks yang dikirim tetap dalam bentuk aslinya.

- **Mode PDU (*Protocol Description Unit*)**

Mode PDU adalah format pesan dalam heksadesimal oktet dan semi-oktet dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Sehingga untuk mengirimkan pesan SMS, pesan yang ditulis dalam bentuk teks harus dikonversikan terlebih dahulu ke dalam format PDU dan semua SMS yang diterima oleh *handphone* agar dapat dibaca maka harus dikonversikan dari PDU format menjadi teks. Terdapat perbedaan format PDU mengirimkan SMS dan menerima SMS.

2.9.1.1 Format PDU untuk mengirimkan SMS

SMS PDU Pengirim adalah SMS Deliver PDU ialah terminal mengirim pesan ke SMSC (lihat topik Teknologi SMS) dalam format PDU. Skema format SMS *Deliver* PDU adalah :

SCA	PDU Type	MR	DA	PID	DCS	VP	UDL	UD
-----	----------	----	----	-----	-----	----	-----	----

Gambar 2.16 Format SMS *deliver* PDU

- **SCA (*Service Center Address*)**

SCA adalah informasi dari alamat (nomor) SMSC. SCA memiliki tiga komponen utama, yaitu: **Len** (*Length*) merupakan panjang informasi SMSC dalam oktet, **Type of Number** sebagai tipe alamat dari SMSC, nilai 81 h untuk format lokal dan nilai 91 h untuk format internasional dan **BSC Digits** berupa nomor SMSC. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal. Pada nomor SMSC IM3 6285500000 bernilai ganjil, maka mendapat tambahan F menjadi 6285500000F. Setelah itu dilakukan perpindahan tempat setiap 2 digit untuk mengubah Nomor SMSC ke Format PDU (*swapped nibble presentation*) dari 62 85 50 00 00 0F menjadi 26 58 05 00 00 F0.

Tabel 2.6 Format SCA pada operator GSM

Operator	Nomor SMSC	Format PDU
Satelindo	62816124	05 91 26 18 16 42
Exelcomindo	62818445009	07 91 26 18 48 54 00 F9
Telkomsel	6281100000	06 91 26 18 01 00 00
IM3	62855000000	05 91 26 58 05 00 00 F0

- **PDU Type**

Nilai *default* dari PDU untuk mengirimkan SMS adalah 04 heksadesimal, akan tetapi pada PDU Type diubah ke biner menjadi 00000100, dimana nilai "1" berada pada biner ke 2 (RD). Format PDU Type adalah :

RP	UDHI	SRR	VPF	VPF	RD	MTI	MTI
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Gambar 2.17. Format PDU type

RP (*Reply Path*) menunjukkan bahwa alur jawaban ada. UDHI (*User Data Header Indicator*) bernilai 1 jika data pengirim dimulai dengan suatu judul/tema. SRR (*Status Report Request*) bernilai 1 jika laporan status pengiriman diminta. VPF (*Validity Period Format*) merupakan Format dari batas waktu pengiriman jika pesan gagal diterima. RD (*Reject Duplicate*) menandakan ya atau tidaknya *service center* akan menerima suatu pengiriman pesan SMS untuk suatu pesan yang masih disimpan dalam *service center* tersebut. Ia memiliki MR dan DA yang sama sebagai pesan dikirimkan oleh OA yang sama. MTI (*Message Type Indicator*) Bernilai 0 menunjukkan SMS-DELIVER.

- **MR (Message Reference)**

Message Reference adalah acuan dari pengaturan SMS. Untuk membiarkan pengaturan SMS dilakukan sendiri oleh *handphone* tujuan, maka nilai yang diberikan adalah 00h. Jadi nilai *default* untuk kirim SMS adalah 00h.

- **DA (Destination Address)**

DA adalah format penulisan nomor tujuan SMS yang memiliki tiga komponen : Len (*length*) yaitu panjang nomor *Destination Address*, Tipe Penomoran (*Type of Number*) sebagai tipe alamat dari *Destination Address* Nilai 81h untuk format local dan Nilai 91h untuk format internasional, BSC *Digits* berupa nomor DA. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal.

Tabel 2.7 Contoh Format DA

format PDU	0C91261885980482
Len	0C
Tipe Penomoran	91
BSC Digits	26 18 85 98 04 82
Destination Address	+628158894028

- **PID (Protocol Identifier)**

Protocol identifier adalah tipe atau format dari cara pengiriman pesan, biasanya diatur oleh *handphone* pengirim. Misalnya tipe standard teks, fax, e-mail, telex dan lain-lain. Nilai *default* untuk kirim SMS adalah 00h. ini berarti tipe SMS yang dikirimkan adalah "*Standard-Text-SMS*".

- **DCS (Data Coding Scheme)**

Data Coding Scheme adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standard, flash SMS atau blinking SMS.

Tabel 2.8 Format DCS

Nilai	Karakter	Message
Heksa	Coding	Class
0	<i>default</i> (7 bit)	<i>no class</i>
F0	<i>default</i> (7 bit)	<i>class 0 (immediate display)</i>
F1	<i>default</i> (7 bit)	<i>class 1 (mobile equipment-specific)</i>
F2	<i>default</i> (7 bit)	<i>class 2 (SIM specific message)</i>
F3	<i>default</i> (7 bit)	<i>class 3 (Terminate Equipment-specific)</i>
F4	8 bit	<i>class 0 (immediate display)</i>
F5	8 bit	<i>class 1 (mobile equipment-specific)</i>
F6	8 bit	<i>class 2 (SIM specific message)</i>
F7	8 bit	<i>class 3 (Terminate Equipment-specific)</i>

Pada beberapa *handphone message* dengan *class 0* dengan *encoding 7 bit* berupa *flash SMS*. Sedangkan dengan *encoding 16 bit unicode (ucs2)*, *message* yang didahului "0001" dengan *class 0* berupa *blinking flash SMS*.

- **Validity Period (VD)**

Validity Period adalah validasi jangka waktu pengiriman SMS untuk disimpan pada SMSC jika *handphone* penerima tidak aktif. Jika

sudah melewati nilai *validity period* berarti SMSC tidak lagi melakukan pengiriman SMS ke penerima. Nilai yang diberikan berupa heksa desimal. Pada contoh format PDU VP bernilai AA heksadesimal = 170. berarti jangka waktu kadaluarsa pengiriman SMS adalah $170 - 166 = 4$ hari.

Tabel 2.9 Format Periode Validitas

Nilai VP	Nilai Validitas Periode
0 - 143	(VP+1) * 5 menit (interval 5 menit hingga 12 jam)
144 - 167	12 jam + ((TP-VP) * 30 menit)
168 - 196	(VP-166) * 1 hari
197 - 255	(VP-192) * 1 minggu

- **User Data Length (UDL)**

UDL adalah panjang karakter pesan teks. bila pesan yang diterima adalah "kirim sms", maka panjang karakter pesan teks adalah 9 desimal pada format PDU menjadi 09 heksadesimal.

- **UD (User Data)**

UD adalah pesan yang akan dikirim. Pesan "kirim<spasi>sms" dikodekan dalam 7 bit *default alphabet* agar bisa dikirimkan harus dikodekan menjadi 8 bit, hasil pengkodeannya yaitu "EBB43CDD06CDDDB73".

Tabel 2.10 Konversi Pesan Teks

Karakter	Septet (7 bit)	Nilai (hexa)	Oktet (8 bit)	Nilai (hexa)
k	1101011	6B	11101011	EB
I	1101001	69	10110100	B4
R	1110010	72	00111100	3C
I	1101001	69	11011101	DD
M	1101101	61	00000110	06
<spasi>	100000	20	11001101	CD
S	1110011	73	11011011	DB
m	1101101	61	01110011	73
s	1110011	73		

2.9.1.2 Format PDU untuk menerima SMS

SMS *receiver* PDU ialah terminal menerima pesan yang datang/masuk dari SMSC dalam format PDU. Skema format PDU untuk menerima SMS adalah :

SCA	PDU Type	OA	PID	DCS	SCTS	UDL	UD
-----	----------	----	-----	-----	------	-----	----

Gambar 2.18 Format PDU terima SMS

- **SCA (*Service Center Address*)**

SCA pada format PDU terima SMS sama dengan SCA pada format PDU pengiriman SMS.

- **Protocol Data Unit (PDU) Type**

RP	UDHI	SRI	<nn>	<nn>	MMS	MTI	MTI
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

Gambar 2.19. Format Tipe PDU terima

Nilai *default* dari PDU untuk menerima SMS adalah 04 heksadesimal, tetapi pada PDU *Type* diubah ke biner menjadi 00000100, dimana nilai "1" berada pada biner ke 2 (MMS).

RP (*Reply Path*) *Parameter* yang menunjukkan bahwa alur jawaban ada. UDHI (*User Data Header Indicator*) bernilai 1 jika data pengirim dimulai dengan suatu judul/tema. SRI (*Status Report Indication*) bernilai 1 jika status laporan akan dikembalikan ke SME. MMS (*More Message to Send*) bernilai 0 jika ada pesan lebih yang akan dikirim. MTI (*Message Type Indicator*) bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu SMS-deliver.

- **Originator Address (OA)**

Originator Address adalah Format penulisan nomor pengirim SMS yang memiliki 3 komponen utama: Len (*length*) yaitu panjang nomor *Originator Address*, *Type of Number* sebagai *type address* dari *Originator*

Address nilai 81h untuk format local dan nilai 91h untuk format Internasional, BSC Digits berupa nomor OA. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal. Sebagai contoh, Bila Nomor OA dalam format PDU = **0C91261885980482**, Maka Len = 0C, *Type of Number* = 91 (internasional) BSC *Digits* = 26 18 85 98 04 82 Nomor OA sebenar adalah +628158894028, yaitu nomor yang mengirim SMS ke penerima. Tanda +62 merupakan asal negara dimana 62 adalah negara Indonesia, sedangkan 8158894028 merupakan nomor *handphone* yang ditambah prefiks "0" (nol) menjadi 08158894028.

- **Protocol Identifier (PID)**

PID adalah tipe atau format cara pengiriman, yang biasa diatur dari *handphone* pengirim. Misalnya tipe standar teks, faks, e-mail, telex dan lain-lain. Nilai *default* untuk terima SMS adalah 00h. ini berarti tipe SMS yang diterima adalah "*Standard-Text-SMS*".

- **Data Coding Scheme (DCS)**

Data Coding Scheme adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standard, *flash SMS* atau *blinking SMS*. Hal yang perlu diperhatikan, pada beberapa *handphone message* dengan *class 0* dengan enkoding 7 bit berupa flash SMS. Sedangkan dengan enkoding 16 bit *Unicode (ucs2)*, message yang didahului "0001" dengan *class 0* berupa *blinking flash SMS*.

Tabel 2.11 Format DCS terima SMS

Nilai	Karakter	Pesan
Heksa	<i>Coding</i>	Kelas
0	<i>default (7 bit)</i>	<i>no class</i>
F0	<i>default (7 bit)</i>	<i>class 0 (immediate display)</i>
F1	<i>default (7 bit)</i>	<i>class 1 (mobile equipment-specific)</i>
F2	<i>default (7 bit)</i>	<i>class 2 (SIM specific message)</i>
F3	<i>default (7 bit)</i>	<i>class 3 (Terminate Equipment-specific)</i>
F4	8 bit	<i>class 0 (immediate display)</i>
F5	8 bit	<i>class 1 (mobile equipment-specific)</i>
F6	8 bit	<i>class 2 (SIM specific message)</i>
F7	8 bit	<i>class 3 (Terminate Equipment-specific)</i>

- **Service Center Time Stamp (SCTS)**

SCTS adalah waktu dari penerimaan pesan oleh SMSC penerima. SCTS terdiri atas tahun, bulan, tanggal, jam, menit dan detik serta zona waktu. Pada contoh format PDU SCTS bernilai 30503161403482, maka pemisahan tiap 2 digit untuk pembagian satuan waktu adalah 30 50 31 61 40 34 82. Dimana dilakukan *swapped nibble presentation*: Tahun = 30 menjadi 03 yaitu tahun 2003, Bulan = 50 menjadi 05 yaitu bulan Mei, Tanggal = 31 menjadi 13, Jam = 61 menjadi 16, Menit = 40 menjadi 04, Detik = 34 menjadi 43, Zona Waktu = 82 menjadi 28 unit : $15 \times 28 = 420$ menit (1 unit = 15 menit) $420 \text{ menit} / 60 = 7 \text{ jam}$.GMT + 07.00 merupakan zona jam di negara Indonesia, Thailand, Vietnam. Penerimaan SMS terjadi pada tanggal 13 Mei 2003 jam 16:04:43 waktu Indonesia (GMT+07:00)

- **User Data Length (UDL)**

UDL adalah panjang pesan yang diterima dalam bentuk teks standard. bila data yang diterima adalah "terima sms", maka panjang data adalah 10 desimal (karakter) pada format PDU menjadi 0A heksadesimal.

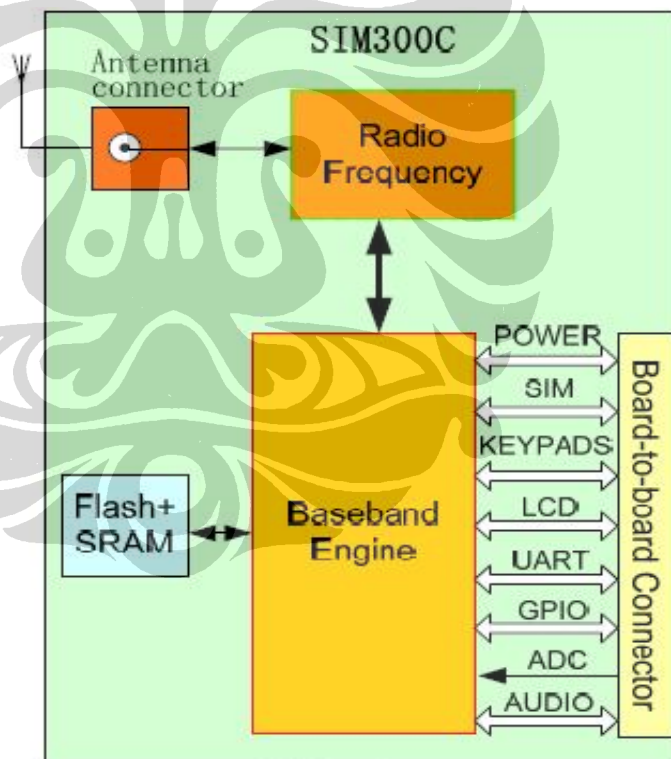
- **User Data (UD)**

UD adalah pesan yang akan dikirim. Pesan yang diterima dalam bentuk kode 8 bit "F4B23CDD0E83E6ED39" agar dapat diterjemahkan dalam bentuk teks. Pesan kode 8 bit tersebut harus dikonversikan terlebih dahulu ke dalam kode 7 bit *default alphabet* lalu dikonversikan ke dalam bentuk teks menggunakan referensi tabel 7 bit *default alphabet*.

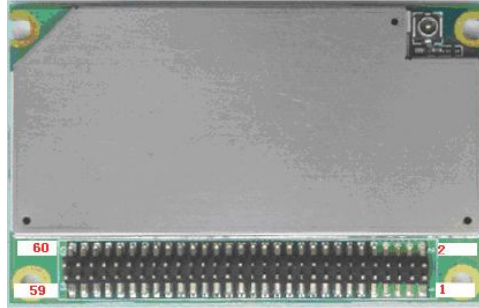
2.10 Modul SIM300C

SIM 300C merupakan modul Tri-band GSM/GPRS yang bekerja pada frekuensi EGSM 900 MHz, DCS 1800 Mhz, dan PCS 1900 MHz. SIM300C mempunyai kemampuan GPRS multi-slot class 10 atau class 8 dan support dengan coding schemes CS-1, CS-2, CS-3 dan CS-4.

Dengan ukuran kecil sekali yaitu 50mm x 33mm x 6,2 mm, SIM300C dapat digunakan dalam pembuatan aplikasi dengan ukuran yang relatif kecil seperti komunikasi bergerak dan lainnya. SIM300C memiliki 60 kaki I/O yang dapat dihubungkan dengan perangkat lainnya. Di dalam SIM300C ini sudah terdapat koneksi untuk keypad, LCD, port serial dan port debug, dua buah kanal audio untuk mikropon dan speaker, dan rangkaian pengisian baterai. Semua perangkat yang terhubung dengan konektor SIM300C dapat dikendalikan dengan AT-Command. Beberapa fungsi utama yang ada di dalam modul SIM300C yaitu modul Baseband GSM, memori Flash dan SRAM, perangkat RF GSM, antarmuka antenna, dan konektor board to board. Gambar 2.20 berikut memperlihatkan blok diagram dari SIM300C dan Gambar 2.21 memperlihatkan bentuk fisik SIM300C.



Gambar 2.20 Blok diagram SIM 300C [8]



Gambar 2.21. Bentuk fisik modul SIM300C [8]

Gambar 2.22 berikut memperlihatkan fungsi tiap-tiap kaki pada modul SIM300C

PIN NO.	PIN NAME	I/O	PIN NO.	PIN NAME	I/O
1	VBAT	I/O	2	GND	
3	VBAT	I/O	4	GND	
5	VBAT	I/O	6	GND	
7	VBAT	I/O	8	GND	
9	VBAT	I/O	10	GND	
11	VCHG	I	12	ADC0	I
13	TEMP_BAT	I	14	VRTC	I/O
15	VDD_EXT	O	16	NETLIGHT	O
17	PWRKEY	I	18	KBC0	O
19	STATUS	O	20	KBC1	O
21	GPIO0	I/O	22	KBC2	O
23	BUZZER	O	24	KBC3	O
25	SIM_VDD	O	26	KBC4	O
27	SIM_RST	O	28	KBR0	I
29	SIM_DATA	I/O	30	KBR1	I
31	SIM_CLK	O	32	KBR2	I
33	SIM_PRESENCE	I	34	KBR3	I
35	GPIO1	I/O	36	KBR4	I
37	DCD	O	38	DISP_CS	O
39	DTR	I	40	DISP_CLK	O
41	RXD	I	42	DISP_DATA	I/O
43	TXD	O	44	DISP_D/C	O
45	RTS	I	46	DISP_RST	O
47	CTS	O	48	DBG_RXD	I
49	RI	O	50	DBG_TXD	O
51	AGND		52	AGND	
53	SPK1P	O	54	MIC1P	I
55	SPK1N	O	56	MIC1N	I
57	SPK2P	O	58	MIC2P	I
59	SPK2N	O	60	MIC2N	I

Gambar 2.22. Fungsi tiap-tiap kaki konektor SIM300C [8]

2.11 Modul SIM508Z

SIM 300C merupakan modul Tri-band GSM/GPRS yang bekerja pada frekuensi EGSM 900 MHz, DCS 1800 Mhz, dan PCS 1900 MHz, dan sudah terdapat teknologi GPS untuk navigasi. SIM508Z mempunyai kemampuan GPRS multi-slot class 10 atau class 8 dan support dengan coding schemes CS-1, CS-2, CS-3 dan CS-4.

Dengan ukuran kecil sekali yaitu 55mm x 34mm x3.0mm, SIM508Z dapat digunakan dalam pembuatan aplikasi dengan ukuran yang relatif kecil seperti komunikasi bergerak, PDA, *Smart phone*, aplikasi AVL(*Automatic Vehicle Location*) dan aplikasi lainnya. SIM508Z memiliki 80 kaki I/O yang dapat di hubungkan dengan perangkat lainnya. Di dalam SIM508Z ini sudah terdapat koneksi untuk keypad, LCD, satu buah port serial dan dua buah port serial GPS, dua buah kanal audio untuk mikropon dan speaker, dan rangkaian pengisian batere. Semua perangkat yang terhubung dengan konektor SIM508Z dapat dikendalikan dengan AT-Command.

Gambar 2.23 berikut memperlihatkan bentuk fisik dari SIM508Z dan Gambar 2.24 memperlihatkan fungsi tiap-tiap kaki SIM508Z.



Gambar 2.23 Bentuk fisik SIM508Z [10]

PIN NO.	PIN NAME	PIN NO.	PIN NAME
2	GND	1	VBAT
4	GND	3	VBAT
6	GND	5	VBAT
8	GND	7	VBAT
10	GND	9	VBAT
12	ADC0	11	VCHG
14	VRTC	13	TEMP_BAT
16	NETLIGHT	15	NC
18	KBC0	17	PWRKEY
20	KBC1	19	STATUS
22	KBC2	21	GPIO0
24	KBC3	23	BUZZER
26	KBC4	25	SIM_VDD
28	KBR0	27	SIM_RST
30	KBR1	29	SIM_DATA
32	KBR2	31	SIM_CLK
34	KBR3	33	SIM_PRESENCE
36	KBR4	35	GPIO1
38	DISP_CS	37	DCD
40	DISP_CLK	39	DTR
42	DISP_DATA	41	RXD
44	DISP_D/C	43	TXD
46	DISP_RST	45	RTS
48	DBG_RXD	47	CTS
50	DBG_TXD	49	RI
52	AGND	51	AGND
54	MIC1P	53	SPK1P
56	MIC1N	55	SPK1N
58	MIC2P	57	SPK2P
60	MIC2N	59	SPK2N
62	GPS_WAKEUP	61	GPS_TIMEMARK
64	GPS_BOOTSEL	63	GPS_FREQ_XFER
66	GPS_M-RST	65	GPS_TIMERSYNC
68	GPS_TXB	67	NC
70	GPS_RXB	69	NC
72	GPS_TXA	71	NC
74	GPS_RXA	73	GPS_GPIO13
76	GPS_VCC_RF	75	GPS_GPIO14
78	GPS_VANT	77	GPS_VRTC
80	GPS_VCC	79	GPS_VCC

Gambar 2.24 Fungsi tiap-tiap kaki SIM508Z [10]