

## **BAB II**

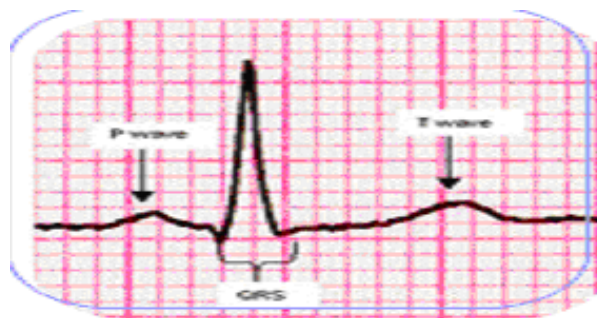
### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Empiris**

#### **2.2 Landasan Teori**

##### **2.2.1 Electrocardiogram ( EKG atau ECG)**

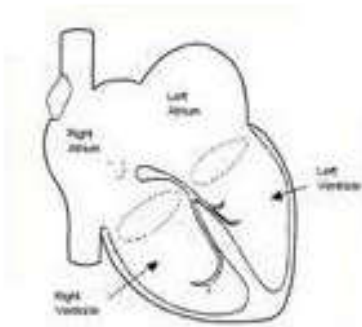
EKG atau ECG adalah suatu alat yang mengukur aktivitas elektrik denyut jantung. Aktivitas jantung menekan dan memompa darah menimbulkan gerakan / gelombang elektrik yang berjalan. Suatu ECG memberi dua macam informasi utama. Pertama, dengan mengukur interval waktu pada ECG, seorang dokter dapat menentukan berapa lama gelombang elektrik yang menerobos jantung dan melihat berapa lama suatu gelombang pada saat berjalan dari satu bagian dari jantung. Kedua, dengan mengukur jumlah aktivitas elektrik jantung (lambat, tidak beraturan, cepat, rapat atau normal) ahli jantung pediatric mampu menemukan bagian-bagian dari jantung yang terlalu besar atau terlalu banyak bekerja. Untuk lebih jelasnya tentang Electrocardiograms dapat dilihat pada gambar di bawah:



Gambar 2.1

Suatu denyut jantung normal pada ECG akan menunjukkan pemilihan waktu puncak dan ruang yang lebih rendah. Pada gambar diatas menunjukkan denyut jantung normal tunggal.

Disegala tempat atria atau kamar bagian atas membuat gelombang yang pertama menghubungi a “ P melambaikan”-- berikut suatu garis flat / kempes ketika dorongan / gerakan jantung yang elektrik pergi kepada kamar alas / bilik. Disegala tempat kamar alas / atau bilik jantung membuat gelombang itu menghubungi a “ QRS kompleks.” Gelombang yang akhir atau “ T melambaikan” menghadirkan kesembuhan elektrik atau kembali ke suatu beristirahat status untuk bilik jantung yang dapat dilihat pada gambar 2.6 dibawah ini:



Gambar 2.2

### 2.2.2 Komunikasi Data Serial

Dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial sinkron dan asinkron. Pada komunikasi data serial sinkron, clock dikirimkan secara bersama-sama dengan data serial, sedangkan pada komunikasi data serial asinkron, clock tidak dikirimkan bersama data serial, tetapi dibangkitkan secara

sendiri-sendiri baik pada sisi pengirim (transmitter) maupun pada sisi penerima (receiver). Pada IBM PC kompatibel port serialnya termasuk jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (Universal Asynchronous Receiver/Transmitter). IC UART dibuat khusus untuk mengubah data paralel menjadi serial dan menerima data serial yang kemudian diubah kembali menjadi data paralel.

Pada UART, kecepatan pengiriman data (baud rate) dan fase clock pada sisi transmitter dan pada sisi receiver harus sinkron. Untuk itu dilakukan sinkronisasi antara transmitter dan receiver. Hal ini dilakukan oleh bit “start” dan bit “stop”.

Kecepatan transmisi (baud rate) dapat dipilih bebas dalam rentang tertentu. Baud rate yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400 dan 9600 (bit/detik). Dalam komunikasi data serial, baud rate dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya, harus ditentukan panjang data (6, 7 atau 8 bit), paritas (genap atau ganjil atau tanpa paritas), dan jumlah bit “stop” (1,  $\frac{1}{2}$ , atau 2 bit).

#### **2.2.2.1 Karakteristik Sinyal Port Serial**

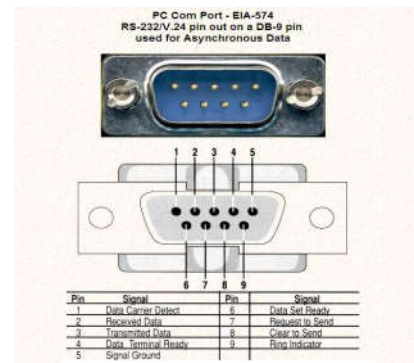
Standar sinyal komunikasi yang banyak digunakan adalah standar RS232 yang dikembangkan oleh Electronic Industry Association and The Telecommunications Industry Association (EIA/TIA) yang pertama kali dipublikasikan pada tahun 1962.

Standar sinyal serial RS232 memiliki ketentuan level tegangan sebagai berikut:

1. Logika “1” disebut “mark” terletak antara –3 volt hingga -25 volt.
2. Logika “0” disebut “space” terletak antara +3 volt hingga +25 volt.
3. Daerah tegangan antara –3 volt hingga +3 volt adalah invalid level, yaitu daerah tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga untuk level tegangan lebih negatif dari –25 volt atau lebih positif dari +25 volt juga harus dihindari karena tegangan tersebut dapat merusak line driver pada saluran RS232.

### 2.2.2.2 Konfigurasi Port Serial

Gambar disamping adalah gambar konektor DB-9 pada bagian belakang CPU. Pada komputer IBM PC kompatibel biasanya kita dapat menemukan dua konektor port serial DB-9 yang sering diberi nama COM1 dan COM2.



Gambar 2,3

Pin	Nama	Arah	Keterangan
1	DCD	In	Data Carrier Detect
2	RxD	In	Receive Data
3	TxD	Out	Transmit Data
4	DTR	Out	Data Terminal Ready
5	GND	-	Ground
6	DSR	In	Data Set Ready
7	RTS	Out	Request to Send
8	CTS	In	Clear to Send
9	RI	In	Ring Indicator

Tabel 2.1

Untuk dapat menggunakan port serial, kita perlu mengetahui alamatnya. Biasanya tersedia dua port serial yaitu COM1 dan COM2. Base address COM1 biasanya adalah 1016 (3F8H) dan COM2 biasanya 760 (2F8H). Alamat tersebut adalah alamat yang biasa digunakan, tergantung dari komputer yang digunakan. Tepatnya kita bisa melihat pada peta memori tempat menyimpan alamat tersebut, yaitu memori 0000.0400H untuk base address COM1 dan memori 0000.0402H untuk base address COM2.

Setelah kita mengetahui base addressnya, maka kita dapat menentukan alamat-alamat register-register yang digunakan untuk komunikasi port serial ini. Berikut adalah tabel register-register tersebut beserta alamatnya :

Nama Register	COM1	COM2
Tx Buffer	3F8H	2F8H
Rx Buffer	3F8H	2F8H
Baud rate Divisor Latch LSB	3F8H	2F8H
Baud rate Divisor Latch MSB	3F9H	2F9H
Interrupt Enable Register	3F9H	2F9H
Interrupt Identification Register	3FAH	2FAH
Line Control Register	3FBH	2FBH
Modem Control Register	3FCH	2FCH
Line Status Register	3FDH	2FDH
Modem Status Register	3FEH	2FEH

Tabel 2.2

### 2.2.2.3 Alasan Penggunaan Port Serial

Dibandingkan dengan menggunakan port paralel, penggunaan port serial terkesan lebih rumit. Berikut ini keuntungan-keuntungan penggunaan port serial dibandingkan penggunaan port paralel :

1. Pada komunikasi dengan kabel yang panjang, masalah cable loss tidak akan menjadi masalah besar daripada menggunakan kabel paralel. Port serial mentransmisikan “1” pada level tegangan –3 volt sampai –25 volt, sedangkan port paralel mentransmisikan “0” pada level tegangan 0 volt dan “1” pada level tegangan 5 volt.
2. Dibutuhkan jumlah kabel yang lebih sedikit, bisa hanya menggunakan 3 kabel, yaitu saluran transmit data, saluran receive data dan saluran ground.
3. Saat ini penggunaan mikrokontroler semakin populer. Kebanyakan mikrokontroler sudah dilengkapi dengan SCI (Serial Communication Interface) yang dapat digunakan untuk komunikasi dengan port serial komputer.

#### **2.2.2.4 Pengaksesan Port Serial Pada Visual Basic**

Untuk mengakses port serial pada Visual Basic, kita dapat mengaksesnya secara langsung melalui register UART atau menggunakan kontrol MSComm yang telah disediakan Visual Basic.

##### **1. Pengaksesan Langsung Melalui Register UART**

Saluran yang digunakan UART untuk komunikasi baik untuk pengiriman maupun penerimaan data adalah saluran RxD dan saluran TxD serta saluran-saluran untuk kontrol, yaitu saluran DCD, DSR, RTS, CTS, DTR dan RI. Saluran-saluran ini ada yang sebagai output dan ada yang sebagai input. Kecuali saluran RxD, saluran-saluran ini dapat diakses secara langsung melalui register UART. Berikut adalah tabel alamat dan lokasi bit saluran tersebut pada register UART.

Nama Pin	Nomor Pin Pada DB-9	COM1	COM2	Bit	Arah
TxD	3	3FBH	2FBH	6	Out
DTR	4	3FCH	2FCH	0	Out
RTS	7	3FCH	2FCH	1	Out
CTS	8	3FEH	2FEH	4	In
DSR	6	3FEH	2FEH	5	In
RI	9	3FEH	2FEH	6	In
DCD	1	3FEH	2FEH	7	In

Tabel 2.3

Untuk dapat mengaksesnya, kita dapat menggunakan fungsi Port\_Out dan fungsi Port\_In yang terdapat pada Port\_IO.dll dan untuk men-set atau meng-clear-kan bit-bit tertentu kita dapat menggunakan prosedur Set\_Bit atau prosedur Clear\_Bit.

Berikut adalah contoh penggunaannya. Kita akan mencoba untuk men-set bit DTR, yaitu membuat saluran DTR berlogika low yang dalam port serial IBM PC Kompatibel bertegangan +12 Volt. Alamat register pengontrol DTR adalah 3FCH untuk COM1 pada bit 0. Perintahnya adalah sebagai berikut :

```
Set_Bit (&H3FC, 0)
```

Untuk meng-clear-kannya, yaitu membuat saluran DTR berlogika high yang dalam port serial IBM PC Kompatibel bertegangan -12 Volt, kita gunakan perintah :

```
Clear_Bit (&H3FC, 0)
```

## 2. Pengaksesan Dengan Menggunakan Kontrol MSComm

Kontrol MSComm menyediakan fasilitas komunikasi antara program aplikasi yang kita buat dengan port serial untuk mengirim atau menerima data melalui port serial. Setiap MSComm hanya menangani satu port serial sehingga jika kita ingin menggunakan lebih dari satu port serial, kita juga harus menggunakan MSComm sebanyak port serial yang kita pakai.

### Properti MSComm

Properti-properti yang sering dipakai sebagai berikut :

CommPort : digunakan untuk menentukan nomor port serial yang akan dipakai.

Setting : digunakan untuk men-set nilai baud rate, parity, jumlah bit data, dan jumlah bit stop.

PortOpen : digunakan untuk membuka atau 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 = "9600,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
```

Keterangan :

- Port serial yang digunakan adalah COM1
- Setting MSComm adalah baud rate 9600, tanpa paritas, jumlah data 8 bit, dan jumlah bit stop adalah 1 bit.
- Memerintahkan kontrol MSComm membaca seluruh isi buffer ketika menggunakan perintah Input (MSComm.InputLen = 0)
- Membuka port serial
- Mengirim perintah "ATV1Q0" diikuti ASCII 13 (enter) ke modem
- Menunggu modem mengirimkan jawaban "OK" ke komputer
- Menutup port serial

### 2.3 Mikrokontroler ATmega 8535

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer), hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register general-purpose, timer/counter fleksibel dengan mode compare, interrupt internal dan eksternal, serial UART, programmable Watchdog Timer, dan mode power saving. Beberapa diantaranya mempunyai ADC dan PWM internal. AVR juga mempunyai In-System Programmable Flash on-chip yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. Chip AVR yang digunakan untuk praktikum adalah ATmega8535. ATmega8535 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan, kebanyakan instruksi dikerjakan pada satu siklus clock. ATmega8535 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses. Beberapa keistimewaan dari AVR ATmega8535 antara lain:

#### 1. Advanced RISC Architecture

- a. 130 Powerful Instructions – Most Single Clock Cycle Execution
- b. 32 x 8 General Purpose Working Registers
- c. Fully Static Operation
- d. Up to 16 MIPS Throughput at 16 MHz
- e. On-chip 2-cycle Multiplier

## **2. Nonvolatile Program and Data Memories**

- a. 8K Bytes of In-System Self-Programmable Flash, Endurance: 10,000 Write/Erase Cycles
- b. Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
- c. 512 Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- d. 512 Bytes Internal SRAM
- e. Programming Lock for Software Security

## **3. Peripheral Features**

- a. Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- b. One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- c. Real Time Counter with Separate Oscillator
- d. Four PWM Channels
- e. 8-channel, 10-bit ADC
  - 8 Single-ended Channels
  - 7 Differential Channels for TQFP Package Only
  - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
    - for TQFP Package Only

- f. Byte-oriented Two-wire Serial Interface
- g. Programmable Serial USART
- h. Master/Slave SPI Serial Interface
- i. Programmable Watchdog Timer with Separate On-chip Oscillator
- j. On-chip Analog Comparator

#### **4. Special Microcontroller Features**

- a. Power-on Reset and Programmable Brown-out Detection
- b. Internal Calibrated RC Oscillator
- c. External and Internal Interrupt Sources
- d. Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down,
- e. Standby and Extended Standby

#### **5. I/O and Packages**

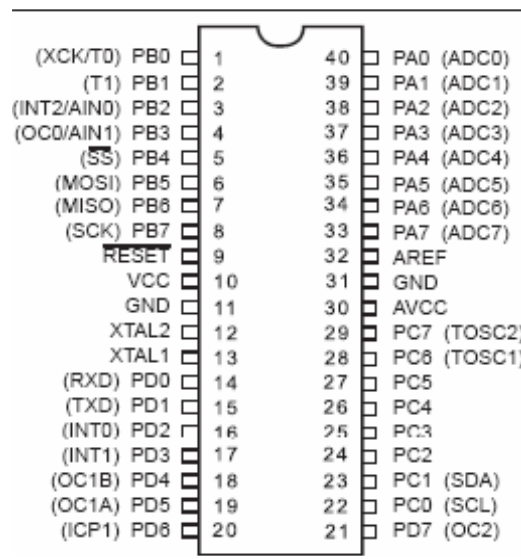
- a. 32 Programmable I/O Lines
- b. 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MFL

#### **6. Operating Voltages**

- a. 2.7 - 5.5V for ATmega8535L
- b. 4.5 - 5.5V for ATmega8535

#### **7. Speed Grades**

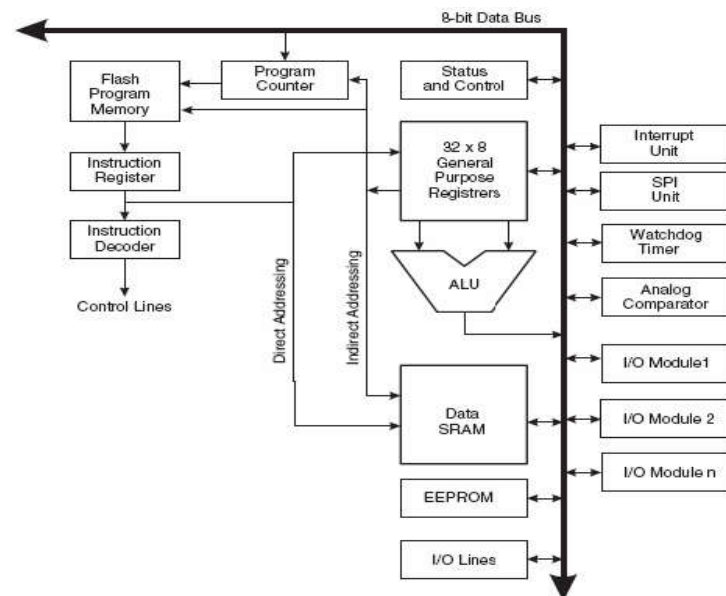
- a. 0 - 8 MHz for ATmega8535L
- b. 0 - 16 MHz for ATmega8535



Gambar 2.4

**Pin-pin ATmega 8535 kemasan 40-pin**  
**Sumber : (Oktavianto, 2005 : 3 )**

Pin-pin pada ATmega 8535 dengan kemasan 40-pin DIP (dual in-line package) ditunjukkan oleh gambar 2.3. Untuk memaksimalkan performa dan paralelisme, AVR menggunakan arsitektur Harvard (dengan memori dan bus terpisah untuk program dan data). Arsitektur CPU dari AVR ditunjukkan oleh gambar 2.5. Instruksi pada memori program dieksekusi dengan pipelining single level, selama sebuah instruksi sedang dikerjakan maka instruksi berikutnya diambil dari memori program.



Gambar 2.5

### Arsitektur CPU dari AVR

Sumber : (Oktavianto, 2005 : 4)

Program ditulis menggunakan tool CodeVision AVR. CodeVision AVR merupakan crosscompiler. Program cukup ditulis menggunakan bahasa-C.



Gambar 2.6

### Alur Pemrograman AVR menggunakan CodeVision AVR

Sumber : (Oktavianto, 2005 : 4)

ATmega 8535 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat port tersebut merupakan jalur bi-directional dengan pilihan internal pull-up. Tiap port mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari port sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam register DDRx (Data Direction Register) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin output. Bila DDxn diset 0 maka Px berfungsi sebagai pin input.

Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin input, maka resistor pull-up akan diaktifkan. Untuk mematikan resistor pull-up, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin output. Pin port adalah tri-state setelah kondisi reset. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 1. Dan bila PORTxn diset 0 pada saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 0.

Saat mengubah kondisi port dari kondisi *tri-state* (DDxn=0, PORTxn=0) ke kondisi *output high* (DDxn=1, PORTxn=1) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (DDxn=0, PORTxn=1) atau kondisi *output low* (DDxn=1, PORTxn=0). Biasanya, kondisi pull-up enabled dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah pull-up. Jika ini bukan suatu masalah, maka bit PUD pada register SFIOR dapat diset 1 untuk mematikan semua pull-up dalam semua port.

Peralihan dari kondisi *input dengan pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi tri-state ( $DDx_n=0$ ,  $PORTx_n=0$ ) atau kondisi output high ( $DDx_n=1$ ,  $PORTx_n=0$ ) sebagai kondisi transisi. Lebih detail mengenai port ini dapat dilihat pada manual datasheet dari IC ATmega 8535.

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Tabel 2.4

**Konfigurasi pin port**  
Sumber (Oktavianto, 2005 : 5 )

Bit	7	6	5	4	3	2	1	0	SFIOR
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10	
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 2 – PUD : Pull-up Disable Bila bit diset bernilai 1 maka pull-up pada port I/O akan dimatikan walaupun register DDxn dan PORTxn dikonfigurasi untuk menyalakan pull-up ( $DDx_n=0$ ,  $PORTx_n=1$ ).

Pada software Code Vision AVR telah disediakan beberapa rutin standar yang dapat langsung digunakan. Anda dapat melihat lebih detail pada manual dari Code Vision AVR.