

BAB 2

TINJAUAN TEORITIS

2.1 Sensor Optocoupler

Optocoupler adalah suatu piranti yang terdiri dari 2 bagian yaitu transmitter dan receiver, yaitu antara bagian cahaya dengan bagian deteksi sumber cahaya terpisah. Biasanya optocoupler digunakan sebagai saklar elektrik, yang bekerja secara otomatis.

Pada dasarnya Optocoupler adalah suatu komponen penghubung (coupling) yang bekerja berdasarkan picu cahaya optic. Optocoupler terdiri dari dua bagian yaitu:

1. Pada transmitter dibangun dari sebuah LED infra merah. Jika dibandingkan dengan menggunakan LED biasa, LED infra merah memiliki ketahanan yang lebih baik terhadap sinyal tampak. Cahaya yang dipancarkan oleh LED infra merah tidak terlihat oleh mata telanjang.
2. Pada bagian receiver dibangun dengan dasar komponen Photodiode. Photodiode merupakan suatu transistor yang peka terhadap tenaga cahaya. Suatu sumber cahaya menghasilkan energi panas, begitu pula dengan spektrum infra merah. Karena spektrum inframerah mempunyai efek panas yang lebih besar dari cahaya tampak, maka Photodiode lebih peka untuk menangkap radiasi dari sinar infra merah.

Oleh karena itu Optocoupler dapat dikatakan sebagai gabungan dari LED infra merah dengan fototransistor yang terbungkus menjadi satu chips. Cahaya infra

merah termasuk dalam gelombang elektromagnetik yang tidak tampak oleh mata telanjang. Sinar ini tidak tampak oleh mata karena mempunyai panjang gelombang , berkas cahaya yang terlalu panjang bagi tanggapan mata manusia. Sinar infra merah mempunyai daerah frekuensi 1×10^{12} Hz sampai dengan 1×10^{14} GHz atau daerah frekuensi dengan panjang gelombang $1\mu\text{m} - 1\text{mm}$.

LED infra merah ini merupakan komponen elektronika yang memancarkan cahaya infra merah dengan konsumsi daya sangat kecil. Jika diberi bias maju, LED infra merah yang terdapat pada optocoupler akan mengeluarkan panjang gelombang sekitar 0,9 mikrometer.

Proses terjadinya pancaran cahaya pada LED infra merah dalam optocoupler adalah sebagai berikut. Saat dioda menghantarkan arus, elektron lepas dari ikatannya karena memerlukan tenaga dari catu daya listrik. Setelah elektron lepas, banyak elektron yang bergabung dengan lubang yang ada di sekitarnya (memasuki lubang lain yang kosong). Pada saat masuk lubang yang lain, elektron melepaskan tenaga yang akan diradiasikan dalam bentuk cahaya, sehingga dioda akan menyala atau memancarkan cahaya pada saat dilewati arus. Cahaya infra merah yang terdapat pada optocoupler tidak perlu lensa untuk memfokuskan cahaya karena dalam satu chip mempunyai jarak yang dekat dengan penerimanya. Pada optocoupler yang bertugas sebagai penerima cahaya infra merah adalah fototransistor. Fototransistor merupakan komponen elektronika yang berfungsi sebagai detektor cahaya infra merah. Detektor cahaya ini mengubah efek cahaya menjadi sinyal listrik, oleh sebab itu fototransistor termasuk dalam golongan detektor optik.

Fototransistor memiliki sambungan kolektor–basis yang besar dengan cahaya infra merah, karena cahaya ini dapat membangkitkan pasangan lubang elektron. Dengan diberi bias maju, cahaya yang masuk akan menimbulkan arus pada kolektor.

Fototransistor memiliki bahan utama yaitu germanium atau silikon yang sama dengan bahan pembuat transistor. Tipe fototransistor juga sama dengan transistor pada umumnya yaitu PNP dan NPN. Perbedaan transistor dengan fototransistor hanya terletak pada dindingnya yang memungkinkan cahaya infra merah

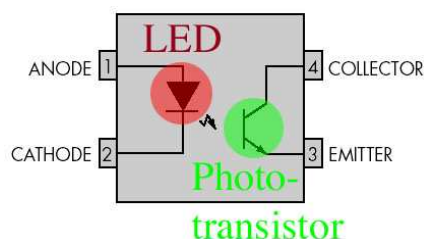
mengaktifkan daerah basis, sedangkan transistor biasa ditempatkan pada dinding logam yang tertutup.

Ditinjau dari penggunaannya, fisik optocoupler dapat berbentuk bermacam-macam. Bila hanya digunakan untuk mengisolasi level tegangan atau data pada sisi transmitter dan sisi receiver, maka optocoupler ini biasanya dibuat dalam bentuk solid (tidak ada ruang antara LED dan Photodiode). Sehingga sinyal listrik yang ada pada input dan output akan terisolasi. Dengan kata lain optocoupler ini digunakan sebagai optoisolator jenis IC.

Prinsip kerja dari optocoupler adalah :

- a. Jika antara Photodiode dan LED terhalang maka Photodiode tersebut akan off sehingga output dari kolektor akan berlogika high.
- b. Sebaliknya jika antara Photodiode dan LED tidak terhalang maka Photodiode dan LED tidak terhalang maka Photodiode tersebut akan on sehingga output-nya akan berlogika low.

Sebagai piranti elektronika yang berfungsi sebagai pemisah antara rangkaian power dengan rangkaian control. Komponen ini merupakan salah satu jenis komponen yang memanfaatkan sinar sebagai pemicu on/off-nya. Opto berarti optic dan coupler berarti pemicu. Sehingga bisa diartikan bahwa optocoupler merupakan suatu komponen yang bekerja berdasarkan picu cahaya optic opto-coupler termasuk dalam sensor, dimana terdiri dari dua bagian yaitu transmitter dan receiver. Dasar rangkaian dapat ditunjukkan seperti pada gambar dibawah ini:



Gambar 2.1 Optocoupler

Sebagai pemancar atau transmitter dibangun dari sebuah led infra merah untuk mendapatkan ketahanan yang lebih baik daripada menggunakan led biasa.

Sensor ini bisa digunakan sebagai isolator dari rangkaian tegangan rendah kerangkaian tegangan tinggi. Selain itu juga bisa dipakai sebagai pendeteksi adanya penghalang antara transmitter dan receiver dengan memberi ruang uji dibagian tengah antara led dengan photo transistor. Penggunaan ini bisa diterapkan untuk mendeteksi putaran motor atau mendeteksi lubang penanda disket pada disk drive computer. Tapi pada alat yang penulis buat optocoupler untuk mendeteksi putaran.

Penggunaan dari optocoupler tergantung dari kebutuhannya. Ada berbagai macam bentuk, jenis, dan type. Seperti MOC 3040 atau 3020, 4N25 atau 4N33 dan sebagainya. Pada umumnya semua jenis optocoupler pada lembar datanya mampu dibebani tegangan sampai 7500 Volt tanpa terjadi kerusakan atau kebocoran. Biasanya dipasaran optocoupler tersedianya dengan type 4NXX atau MOC XXXX dengan X adalah angka part valuenya. Untuk type 4N25 ini mempunyai tegangan isolasi sebesar 2500 Volt dengan kemampuan maksimal led dialiri arus forward sebesar 80 mA. Namun besarnya arus led yang digunakan berkisar antara 15mA - 30 mA dan untuk menghubungkan-nya dengan tegangan +5 Volt diperlukan tahanan sekitar 1K ohm

2.2 Motor

Motor merupakan perangkat elektromagnetik yang dapat mengubah energi listrik menjadi energi mekanik. Perubahan ini dilakukan dengan merubah tenaga listrik menjadi magnet yang disebut sebagai elektromagnet. Sebagaimana kita ketahui bahwa : kutub-kutub dari magnet yang senama akan tolak-menolak dan kutub-kutub tidak senama, tarik-menarik. Maka kita dapat memperoleh gerakan jika kita menempatkan sebuah magnet pada sebuah poros yang dapat berputar, dan magnet yang lain pada suatu kedudukan yang tetap. Dengan cara inilah energi listrik dapat diubah menjadi energi mekanik.

Energi mekanik ini digunakan untuk, misalnya memutar impeler pompa, fan atau blower, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga dirumah (mixer, bor listrik, afan angin) dan industri. Motor listrik

kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban total industri.

Secara umum motor listrik dapat dibagi menjadi motor ac dan motor dc, pembagian ini berdasarkan pada arus listrik yang digunakan untuk menggerakannya. Namun penulis pada bagian ini kita hanya membahas mengenai motor dc

2.3 Motor Stepper

Motor stepper dibedakan menjadi dua macam berdasarkan magnet yang digunakan, yaitu tipe *permanen magnet* dan *variabel reluktansi*. Pada umumnya motor stepper yang digunakan saat ini adalah motor stepper yang mempunyai variabel reluktansi. Cara yang paling mudah untuk membedakan antara tipe motor stepper di atas adalah dengan cara memutar rotor dengan tangan ketika tidak dihubungkan ke suplay.

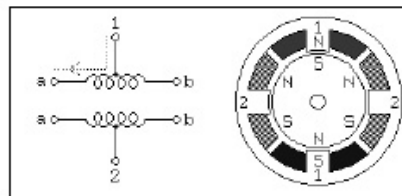
Pada motor stepper yang mempunyai permanen magnet maka ketika diputar dengan tangan akan terasa lebih tersendat karena adanya gaya yang ditimbulkan oleh magnet permanen. Tetapi ketika menggunakan motor dengan variabel reluktansi maka ketika diputar akan lebih halus karena sisa reluktansinya cukup kecil.

2.3.1 Tipe Motor Stepper

a. Motor Stepper Unipolar

Motor stepper unipolar terdiri dari dua lilitan yang memiliki center tap. Center tap dari masing masing lilitan ada yang berupa kabel terpisah ada juga yang sudah terhubung didalamnya sehingga center tap yang keluar hanya satu kabel. Untuk motor stepper yang center tapnya ada pada masing – masing lilitan kabel inputnya ada 6 kabel. Namun jika center tapnya sudah terhubung di dalam kabel inputannya hanya 5 kabel. Center tap dari motor stepper dapat dihubungkan ke pentanahan atau ada juga yang

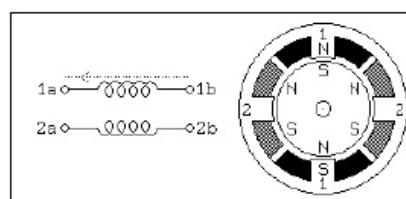
menghubungkannya ke +VCC hal ini sangat dipengaruhi oleh driver yang digunakan. Sebagai gambaran dapat dilihat konstruksi motor stepper unipolar pada gambar berikut:



Gambar 2. 2 : Motor stepper Unipolar

b. Motor Stepper Bipolar

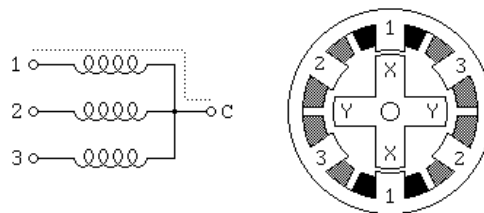
Motor stepper bipolar memiliki dua lilitan perbedaaan dari tipe unipolar adalah bahwa pada tipe bipolar lilitannya tidak memiliki center tap. Keunggulan tipe bipolar yaitu memiliki torsi yang lebih besar jika dibandingkan dengan tipe unipolar untuk ukuran yang sama. Pada motor stepper tipe ini hanya memiliki empat kabel masukan. Namun ntuk menggerakan motor stepper tipe ini lebih rumit jika dibandingkan dengan menggerakan motor stepper tipe unipolar. Sebagai gambaran dapat dilihat konstruksi motor stepper bipolar pada gambar berikut :



Gambar 2.3 : Motor stepper Bipolar

2.3.2 Variabel Reluktansi Motor

Pada motor stepper yang mempunyai variabel reluktansi maka terdapat 3 buah lilitan yang pada ujungnya dijadikan satu pada sebuah pin common. Untuk dapat menggerakkan motor ini maka aktivasi tiap-tiap lilitan harus sesuai urutannya. Gambar 2.4 merupakan gambar struktur dari motor dengan variabel reluktansi dimana tiap stepnya adalah 30° . Mempunyai 4 buah kutub pada rotor dan 6 buah kutub pada statornya yang terletak saling berseberangan.



Gambar 2.4 Variabel Reluktansi Motor

Jika lilitan 1 dilewati oleh arus, lilitan 2 mati dan lilitan 3 juga mati maka kumparan 1 akan menghasilkan gaya tolakan kepada rotor dan rotor akan berputar sejauh 30° searah jarum jam sehingga kutub rotor dengan label Y sejajar dengan kutub dengan label 2.

Jika kondisi seperti ini berulang terus menerus secara berurutan, lilitan 2 dilewati arus kemudian lilitan 3 maka motor akan berputar secara terus menerus. Agar dapat berputar sebanyak 21 step maka perlu diberikan data dengan urutan seperti pada data berikut :

```
Lilitan 1: 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1  
Lilitan 2: 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0  
Lilitan 3: 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0
```

Gambar 2.5 Urutan Data Motor Stepper Dengan Tipe Variabel Reluktansi

Dari data di atas '1' diartikan bahwa lilitan yang bersangkutan dilewati arus sehingga menghasilkan gaya tolak untuk rotor. Sedangkan '0' diartikan lilitan dalam kondisi off, tidak mendapatkan arus.

2.4 Mikrokontroller Jenis AVR (*Alf and Vegard RISC*)

Mikrokontroller merupakan alat pengolahan data digital dan analog (fitur ADC pada seri AVR) dalam level tegangan maksimum 5V. Keunggulan mikrokontroller dibanding microprocessor yaitu lebih murah dan didukung dengan software compiler yang sangat beragam seperti software compailer C/C++, basic, pascal, bahkan assembler. Sehingga penggunaan dapat memilih program yang sesuai dengan kemampuannya.

Pada mikrokontroller AVR membutuhkan sedikit komponen pendukung, tidak seperti mikrokontroller yang sistem pendukungnya terpisah atau terbentuk secara parsial, seperti RAM, ROM, dan Mikroprocessor sendiri.

Keunggulan AVR dibanding dengan mikrokontroller lain, memiliki kecepatan eksekusi yang lebih cepat karena sebagian besar instruksi dibentuk dalam satu siklus clock, jika dibandingkan dengan mikrokontroller jenis MCS51 yang mengeksekusi satu instruksi dengan 12 siklus clock.

AVR memiliki fitur yang lengkap (ADC internal, EEPROM internal, Timer/counter, Watchdog timer, PWM, Port I/O, komunikasi serial, komperator, TWI, dll), sehingga dengan fasilitas ini programer dan disigner dapat menggunakannya untuk berbagai sistem elektronika seperti robot, automasi industri, peralatan telekomunikasi dan berbagai keperluan lain.

Beberapa produsen mikrokontroller mengeluarkan jenis mikrokontroller yang memiliki fitur-fitur yang sangat beragam seperti AVR jenis ATtiny, ATmega, dan AT90. dari segi jumlah pin dan memori dapat kita lihat perbedaan jenis mikrokontroller jenis AVR ini, seperti tabel dibawah ini.

Tabel 2.1 Jenis-Jenis AVR

Mikrontroller		Memori		
Tip	Jumlah Pin	Flash	EEPROM	SRAM
TinyAVR	8 – 32	1 – 2K	64 – 128	0 – 128
AT90XX	20 – 44	1 – 8K	128 – 512	0 – 1K
ATmega	32 – 64	8 – 128K	512 – 4K	512 – 4K

Pemrograman mikrokontroler AVR dengan menggunakan bahasa C lebih tangguh bila dibanding dengan jenis pemrograman lainnya. Karena bahasa program C independent terhadap hardware, lain halnya dengan menggunakan bahasa assembler yang mana setiap jenis mikrokontroler berbeda jenis pemrogramannya sehingga orang yang ingin memprogram jenis mikrokontroler yang lainnya harus belajar bahasa assembler nya dulu. Keunggulan lainnya dengan menggunakan bahasa C adalah penyusunan program yang besar dapat dilakukan dengan mudah, dan program yang sudah jadi dapat digunakan ke jenis AVR lain dengan hanya mengubah fungsi-fungsi register dan portnya.

Beberapa faktor pertimbangan penting untuk memilih mikrokontroler jenis AVR antara lain:

- a. Harga mikrokontroler yang lebih murah dibanding mikroprocessor;
- b. Ukuran memori mikrokontroler yang cukup besar dan untuk menambah memorinya dapat digunakan memori eksternal;
- c. Fitur ADC, timer, PWM, USART, dan fasilitas lainnya yang memudahkan Disigner dalam merancang sistem;
- d. Kecepatan eksekusi program dimana intruksi dieksekusi dalam 1 siklus clock sementara mikrokontroler jenis MCS51 hanya dapat mengeksekusi intruksi dalam 12 clock.
- e. Adanya clock internal sehingga mikrokontroler dapat digunakan tanpa menggunakan crystal;
- f. Software pendukung yang sangat beragam dan penggunaanya jauh lebih mudah karena software menyediakan fitur yang memudahkan dalam memprogramnya seperti CodeVisionAVR dan BASCOM AVR yang

menyediakan fitur design LCD pada BASCOM AVR dan fitur penghasil program pada CodeVisionAVR.

2.4.1 Fitur ATmega8535

Berikut ini adalah fitur – fitur yang dimiliki oleh Atmega 8535

- a. 130 macam intruksi yang hampir semuanya dieksekusi dalam satu siklus clock
- b. 32 x 8-bit register serba guna
- c. Kecepatan mencapai 16 MIPS dengan clock 16 MHz
- d. 8 Kbyte Flash Memori, yang memiliki fasilitas In system Programming
- e. 512 Byte internal EEPROM
- f. 512 Byte SRAM
- g. Programming Lock, Fasilitas untuk mengamankan kode program
- h. 2 buah timer/counter 8-bit dan 1 buah timer/ counter 16-bit
- i. 4 channel output PWM
- j. 8 channel ADC 10-bit
- k. Serial USART
- l. Master / Slave SPI serial interface
- m. Serial TWI atau I2C
- n. On-Chip Analog Comparator

2.4.2 Fungsi Pin-Pin AVR ATmega8535 yang digunakan

a. Fungsi VCC

VCC merupakan pin yang berfungsi sebagai masukan catu daya

b. Fungsi GND

GND merupakan pin ground untuk catu daya digital

c. Fungsi PORTA

PORTA memiliki fungsi input dan juga digunakan sebagai input analog yang akan dikonversikan menjadi data-data digital (ADC), seperti yang ditunjukkan pada tabel di bawah. Jika salah satu PORTA dikonfigurasi sebagai output ketika PORTA difungsikan sebagai ADC maka hasil pengkonversian analog ke digital akan

menghasilkan data error. Untuk mengaktifkan fungsi ADC pada PORTA dengan CodeVision AVR akan dibahas di lab selanjutnya tentang teknik pemrograman C pada ATmega8535 dengan menggunakan software CodeVision AVR. Fungsi PORTA dapat dilihat pada tabel.

Tabel 2.2 PORTA

PORT	Keterangan Fungsi PORT
PA.0	Keypad (Baris ke-1)
PA.1	Keypad (Baris ke-1)
PA.2	Keypad (Baris ke-2)
PA.3	Keypad (Baris ke-2)
PA.4	Keypad (Baris ke-3)
PA.5	Keypad (Baris ke-3)
PA.6	Keypad (Baris ke-4)
PA.7	Keypad (Baris ke-4)

d. Fungsi PORTB

PORTB merupakan pin Input dan Output 8 bit dua arah dan memiliki pin fungsi khusus yaitu Timer/ Counter, Komparator analog dan SPI dan Fungsi **PORTB** yang digunakan pada rangkaian dapat dilihat pada tabel dibawah ini,

Tabel 2.3 PORTB

PORT	Keterangan Fungsi PORT
PB.0	Driver Motor Stepper
PB.1	Driver Motor Stepper
PB.2	Driver Motor Stepper
PB.3	Driver Motor Stepper
PB.7	Buzzer

e. Fungsi PORTC

PORTC merupakan pin input dan output dua arah dan memiliki pin fungsi khusus yaitu TWI, Komparator analog dan Timer Oscilator dan Fungsi **PORTC** yang digunakan dapat dilihat pada tabel dibawah ini,

Tabel 2.4 PORTC

PORT	Keterangan Fungsi PORT
PC.0	D4 (LCD PIN 11)
PC.1	D5 (LCD PIN 12)
PC.2	D6 (LCD PIN 13)
PC.3	D7 (LCD PIN 14)
PC.4	Free
PC.5	RS (LCD PIN 4)
PC.6	RW (LCD PIN 5)
PC.7	CLK (LCD PIN 6)

f. Fungsi PORTD

PORTD merupakan pin input dan output dua arah dan pin fungsi khusus yaitu komparator analog, interupsi eksternal dan komunikasi serial. Fungsi **PORTD** dapat dilihat pada tabel dibawah ini,

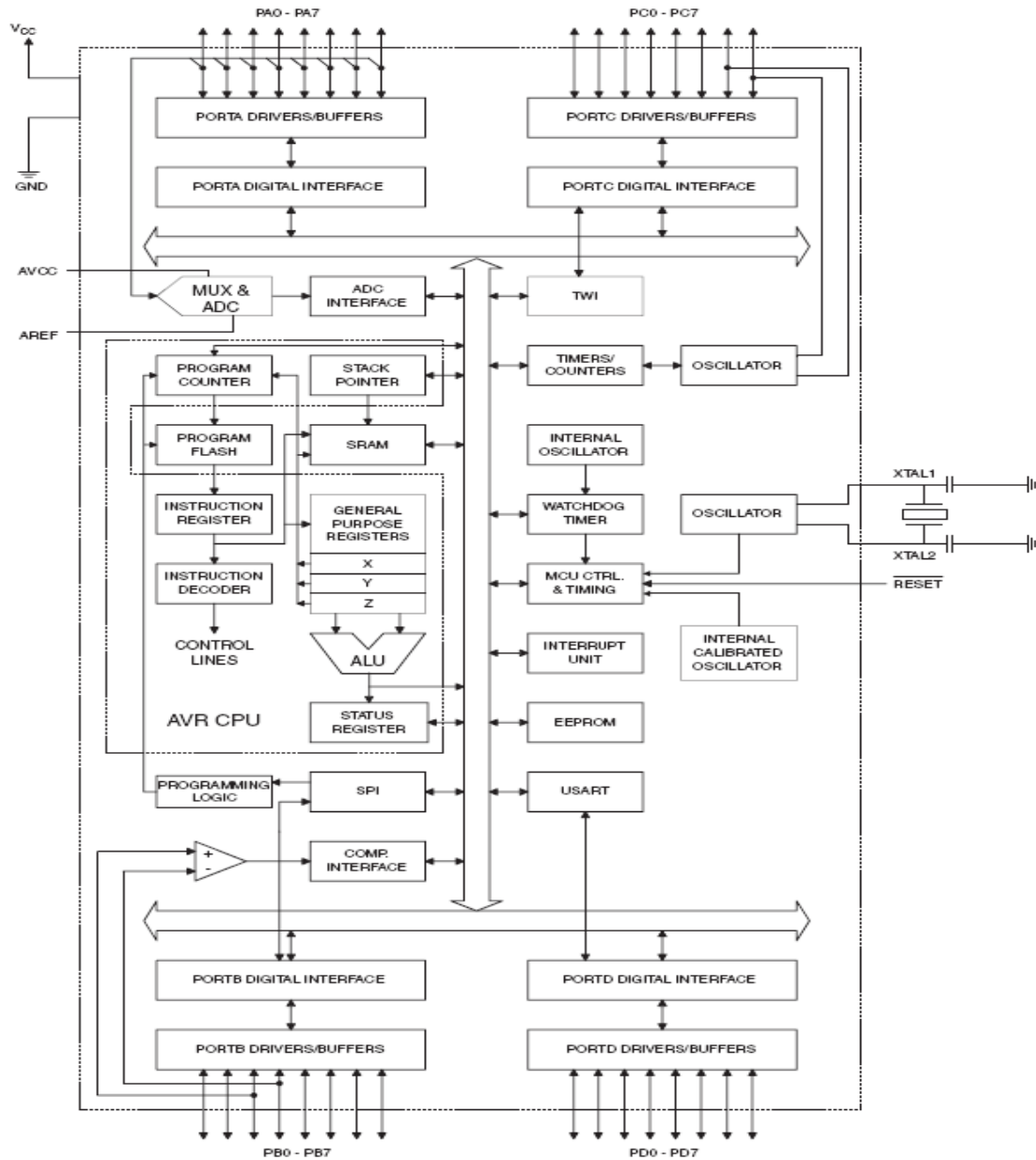
Tabel 2.5 PORTD

PORTIN	Keterangan Fungsi PORT
PIND.7	Input sensor Optik

2.4.3 Arsitektur ATmega8535

ATmega8535 adalah mikrokontroler 8-bit CMOS dengan menggunakan daya yang rendah dan menjalankan intruksi dalam satu clock. Mikrokontroler ATmega8535 dikombinasikan dengan 32 register. Semua register terhubung langsung ke Arithmetic Logic Unit (ALU), membiarkan 2 register untuk di akses di dalam satu instruksi dieksekusi dalam satu clock. Status dan kontrol berfungsi untuk menyimpan instruksi aritmatika yang baru saja dieksekusi. Informasi ini berguna untuk mengubah alur program saat mengeksekusi operasi kondisional. Instruksi dijemput dari flash memory. Setiap byte flash memory memiliki alamat masing-masing. Alamat instruksi yang akan dieksekusi senantiasa disimpan Program Counter. Ketika terjadi interupsi atau pemanggilan rutin biasa, alamat di Program Counter disimpan terlebih dahulu di

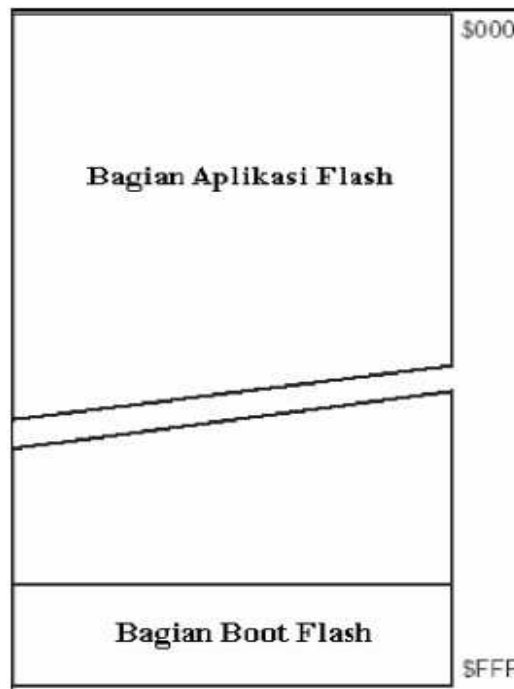
stack. Alamat interupsi atau rutin kemudian ditulis ke Program Counter, instruksi kemudian dijemput dan dieksekusi. Ketika CPU telah mengeksekusi rutin interupsi atau rutin biasa, alamat yang ada di stack dibaca dan ditulis kembali ke Program



Gambar 2.6 Arsitektur ATmega8535

2.4.4 Program Memori

ATmega8535 berisi 8K bytes On-Chip di dalam sistem Memori flash Reprogrammable untuk penyimpanan program. Karena semua AVR instruksi adalah 16 atau 32 bits lebar, Flash adalah berbentuk 4K x16. Untuk keamanan perangkat lunak, Flash Ruang program memori adalah dibagi menjadi dua bagian, bagian boot program dan bagian aplikasi program dengan alamat mulai dari \$000 sampai \$FFF. Flash Memori mempunyai suatu daya tahan sedikitnya 10,000 write/erase Cycles. ATmega8535 Program Counter (PC) adalah 12 bitebar, alamat ini 4K lokasi program memori.



Gambar 2.7 Program Memori

2.4.5 Memori Data (SRAM)

Memori data terbagi menjadi 3 bagian, yaitu 32 register umum, 64 buah register I/O, dan 512 byte SRAM. Dalam data memori terdapat 608 lokasi address data. 96 lokasi address digunakan untuk register file dan I/O memori sementara 512 lokasi

address lainnya digunakan untuk internal data SRAM. Register file terdiri dari 32 general purpose working register, I/O register terdiri dari 64 register

2.4.6 EEPROM

ATmega8535 memiliki EEPROM sebesar 512 byte untuk menyimpan data. Lokasinya terpisah dengan sistem address register, data register dan control register yang dibuat khusus untuk EEPROM. EEPROM dapat diprogram, dibaca dan dihapus sebanyak 100.000 kali.

Data yang tersimpan di EEPROM akan tetap tersimpan walaupun mikrokontroler ATmega8535 tidak diberi catu daya.

2.5 Perangkat Lunak (*Software*)

2.5.1 ATmega8535 dengan Bahasa C

C adalah bahasa pemrograman yang dapat dikatakan berada antara bahasa tingkat rendah (bahasa yang berorientasi pada mesin) dan bahasa tingkat tinggi (bahasa yang berorientasi pada manusia). Seperti yang diketahui, bahasa tingkat tinggi mempunyai kompatibilitas antara platform. Karena itu, amat mudah untuk membuat program pada berbagai mesin. Berbeda halnya dengan menggunakan bahasa mesin, sebab setiap perintahnya sangat bergantung pada jenis mesin

Pembuat bahasa C adalah Brian W. Kernighan dan Dennis M. Ritchie pada tahun 1972. C adalah bahasa pemrograman terstruktur , yang membagi program dalam bentuk block. Tujuannya untuk memudahkan dalam pembuatan dan pengembangan program. Program yang ditulis dengan bahasa C mudah sekali dipindahkan dari satu jenis program ke bahasa program lain. Hal ini karena adanya

standarisasi bahasa C yaitu berupa standar ANSI (American National Standar Institute) yang dijadikan acuan oleh para pembuat kompuler.

2.5.1.1 Pengenalan pada Bahasa C

Pengenalan merupakan sebuah nama yang didefinisikan oleh program untuk menunjukkan sebuah konstanta, variabel, fungsi, label, atau tipe data khusus.

Pemberian pengenalan pada program harus memenuhi syarat-syarat dibawah ini:

- a. Karakter pertama tidak menggunakan angka;
- b. Karakter kedua berupa huruf, angka, garis bawah;
- c. Tidak menggunakan spasi;
- d. Bersifat case sensitive, yaitu huruf kapital dan huruf kecil dianggap berbeda;
- e. Tidak boleh menggunakan kata-kata yang merupakan sintaks atau operator

Contoh menggunakan pengenalan yang diperbolehkan:

1. Nama
2. _nama
3. Nama2
4. Nama_pengenal

Contoh penggunaan pengenalan yang tidak diperbolehkan:

1. 2nama
2. Nama+2
3. Nama pengenal

2.5.1.2 Tipe Data

Pemberian signed dan unsigned pada tipe data menyebabkan jangkauan dari tipe berubah. Pada unsigned menyebabkan tipe data akan selalu bernilai positif sedangkan signed menyebabkan nilai tipe data bernilai negatif dan memungkinkan data bernilai positif. Perbedaan nilai tipe data dapat kita lihat pada tabel dibawah ini:

Tabel 2.6 Tipe Data

Pemodifikasi Tipe	Persamaan	Jangkauan Nilai
Signed char	Char	-128 s/d 127
Signed int	Int	-32.768 s/d 32.767
Signed short int	Short, signed, short	-32.768 s/d 32.767
Signed long int	Long, long int, signed long	-2.147.483 s/d 2.147.647
Unsigned char	Tidak ada	0 s/d 255
Unsigned int	Unsigned	0 s/d 65.535
Unsigned short int	Unsigned short	0 s/d 65.535
Unsigned long int	Unsigned long	0 s/d 4.294.967.295

Contoh program yang menunjukkan pengaruh signed dan unsigned pada hasil program,

```
#include <mega.8535.h>
```

```
#include <delay.h>
```

```
Void main (void)
```

```
{
```

```
    Int a, b;        //pengenal
```

```
    Unsigned d, c;
```

```
    a = b0;
```

```
    b = 40;
```

```
    d = 50;
```

```
    e = 40;
```

```
    PORTC = 0x00;
```

```
    DDRC = 0xff      //set PORTC sebagai output
```

```
    PORTB = 0x00;
```

```
    DDRB = 0xff      //set PORTB sebagai output
```

```

        While(1)
        {
            PORTB = a b;
            PORTC = d - e;
            Delay_ms (100);
        };
    }

```

Program di atas akan memberikan data di PORTB – 10 (desimal) sedangkan PORTIB = -10 (desimal) karena PORT mikrokontroler tidak dapat mengeluarkan nilai negatif maka PORTIB dan PORTC akan memiliki keluaran 0x0A tapi pada kenyataannya PORTC lebih banyak memakan memori karena tanda negatif tersebut disimpan dalam memori.

Pada program di atas terdapat tulisan //set PORTB sebagai output yang berguna sebagai komentar yang mana komentar ini tidak mempengaruhi hasil dari program. Ada dua cara penulisan komentar pada pemrograman bahasa C, yaitu dengan mengawali komentar dengan tanda “// “ (untuk komentar yang hanya satu baris) dan mengawali komentar dengan tanda “ /* “ dan mengakhiri komentar dengan tanda “ */ “.

Contoh:

```
// ini adalah komentar
```

```
/* ini adalah komentar
```

Yang lebih panjang

Dan lebih panjang lagi */

2.5.1.3 Header

Header digunakan untuk menginstruksikan kompiler untuk menyisipkan file lain. Di dalam file header ini tersimpan deklarasi, fungsi, variabel, dan jenis mikrokontroler

yang kita gunakan (pada software Code Vision AVR). File-file yang berakhiran **.h** disebut file header.

File header yang digunakan untuk mendefinisikan jenis mikrokontroler yang digunakan berfungsi sebagai pengarah yang mana pendeklarasian register-register yang terdapat program difungsikan untuk jenis mikrokontroler apa yang digunakan (pada software Code Vision AVR).

Contoh:

```
# include <mega8535. h>
```

```
# include <delay. h>
```

```
# include <stdio. h>
```

2.5.1.4 Operator Aritmatika

Operator aritmatika digunakan untuk melakukan proses perhitungan matematika. Fungsi-fungsi matematika yang terdapat pada bahasa C dapat dilihat pada tabel di bawah ini:

Tabel 2.7 Operasi Aritmatika

Operator	Keterangan
+	Operator untuk penjumlahan
-	Operator untuk pengurangan
*	Operator untuk perkalian
/	Operator untuk pembagian
%	Operator untuk sisa bagi

Contoh penggunaan operator aritmatika dapat dilihat di bawah ini,

```
# include < mega8535. h>
```

```
# include < delay. h>
```

```
void main ( void )
```

```

{

Unsigned char a, b;
a = 0x03 ;
b = 0x05 ;

        DDRC 0xFF ; // PORTC di gunakan sebagai output
        While ( 1 )
        {
        PORTC = ( a * b ) ;
        delay_ms ( 500 ) ;
        }
}

```

2.5.1.5 Operator Pembandingan

Operator pembandingan digunakan untuk membandingkan 2 data atau lebih. Hasil operator akan di jalankan jika pernyataan benar dan tidak di jalankan jika salah. Operator pembandingan dapat kita lihat pada tabel di bawah ini:

Tabel 2.8 Operator Pembandingan

Operator	Contoh	Keterangan
==	x == y	Benar jika kedua data bernilai sama
!=	x != y	Bernilai benar jika kedua data tidak sama
>	x > y	Bernilai benar jika nilai x lebih besar dari pada y
<	x < y	Bernilai jika x lebih kecil dari y
>=	x >= y	Bernilai jika x lebih besar atau sama dengan y
<=	x <= y	Bernilai benar jika x lebih kecil atau sama dengan y

2.5.1.6 Operator Logika

Operator logika digunakan untuk membentuk logika dari dua pernyataan atau lebih. Operator logika dapat dilihat pada tabel di bawah ini:

Tabel 2.9 Operator Logika

Operator	Keterangan
&&	Logika AND
	Logika OR
	Logika NOT

Contoh program:

```
# include < mega32 . h>
```

```
# include < delay . h>
```

```
void main ( void )
```

```
{
```

```
    DDRC = 0XFF ; // sebagai output
```

```
    DDRA = 0X00 ; // sebagai input
```

```
    while ((1 )
```

```
    {
```

```
        if ( PINA . 0 = 1 ) || ( PINA . 1 = 1 ) {
```

```
            PORTC = 0XFF ;
```

```
            delay_ms ( 500 ) ;
```

```
            PORTC = 0X00 ;
```

```
            delay_ms ( 500 ) ;
```

```
        }
```

```
        else {
```

```
            PORTC = 0x00 ;
```

```
            delay ms ( 500 ) ;
```

```
        }
```

```
    }
```

```
}
```

Penjelasan program:

Apabila PINA.0 atau PINA.1 diberi input logika 1 maka PORTC akan mengeluarkan logika 0xff kemudian logika 0x00 secara bergantian dengan selang waktu 0,5 s. dan apabila bukan PINA.1 atau PINA.0 diberi logika 1 maka PORTC akan mengeluarkan logika 0x00

2.5.1.7 Operator Bitwise

Operator logika ini bekerja pada level bit. Perbedaan operator bitwise dengan operator logika adalah pada operator logika akan menghasilkan pernyataan benar atau salah sedangkan pada operator bitwise akan menghasilkan data biner. Operator bitwise dapat dilihat pada tabel di bawah ini:

Tabel 2. 10 Operator Bitwise

Operator	Keterangan
&	Operator AND level bit
	Operator OR level bit
^	Operator XOR level bit
~	Operator NOT level bit
>>	Operator geser kanan
<<	Operator geser kiri

Contoh program:

```
# include < mega32 . h>
```

```
# include < delay . h>
```

```
void main ( void )
```

```
{
```

```
    Unsigned char a, b, c;
```

```
    DDRC = 0XFF ; // sebagai output
```

```

DDRA = 0X00 ; // sebagai input

while ((1 )

{
    a = 0x12;
    b = 0x34;
    c = a & b;
    PORTC = c;
    Delay_ms (500);

};

}

```

Penjelasan program:

```

a = 0x12 = 0001 0010
b = 0x32 = 0011 0000
-----
a & b = 0001 0000

```

2.5.1.8 Operator penugasan dan operator majemuk

Operator ini digunakan untuk memberikan nilai atau manipulasi data sebuah variabel. Operator penguasa dapat kita lihat pada tabel di bawah ini:

Tabel 2. 11 Operator Penugasan

Operator	Keterangan
=	Memberikan nilai variable
+=	Menambahkan nilai variable

-=	Mengurangi nilai variable
*=	Mengalikan nilai variable
/=	Membagi nilai variable
%=	Memperoleh sisa bagi

Contoh :

$a +- 2$; artinya nilai variabel a berubah menjadi $a - a + 2$

$a *= 4$; artinya nilai variabel b berubah menjadi $b = b* 4$

selain operator penugasan diatas juga ada operator penugasan yang berkaitan dengan operator bitwise seperti pada tabel dibawah ini:

Tabel 2. 12 Operator Penugasan

Operator	Contoh	Arti
&=	$x \&= 1$	Variabel x di AND kan dengan 1
=	$x = 1$	Variabel x di OR kan dengan 1
~=	$x ~= 1$	$x = \sim (1)$; $x = 0xFE$
^=	$x ^= 1$	Variabel x di XOR kan dengan 1
<<=	$x <<= 1$	Variabel x digeser kiri 1 kali
>>=	$x >>= 1$	Variabel x digeser kanan 1 kali

2.5.1.9 Operator Penambahan dan Pengurangan

Operator ini digunakan untuk menaikkan atau menurunkan nilai suatu variabel dengan selisih 1. operator ini dapat dilihat pada tabel di bawah ini:

Tabel 2. 13 Operator Penugasan

Operator	Keterangan
++	Penambahan 1 pada variable
--	Pengurangan

Contoh :

```
a = 1;  
b = 2;  
a ++;  
b --;
```

Penjelasan:

Maka operator a++ akan mengubah variabel a dari satu menjadi 2, sedangkan operator B akan mengubah variabel b dari 2 menjadi 1.

2.5.1.10 Pernyataan If dan If Bersarang

Pernyataan if digunakan untuk pengambilan keputusan terhadap 2 atau lebih pernyataan dengan menghasilkan pernyataan benar atau salah. Jika pernyataan benar maka akan di jalankan intruksi pada blok nya, sedangkan jika pernyataan tidak benar maka intruksi yang pada blok lain yang di jalankan (sesuai dengan arah programnya).

Bentuk pernyataan IF adalah sebagai berikut:

1. Bentuk sederhana

```
If (kondisi) {  
    Pernyataan_1;  
    Pernyataan_2;  
    .....;  
}
```

2. Pernyataan else

```
If (kondisi)  
{
```

```

        Pernyataan_1;
        .....;
    }
Else
{
    Pernyataan_2;
    .....;
}

```

3. If di dalam if

Pernyataan ini sering disebut nested if atau if bersarang. Salah satu bentuknya adalah sebagai berikut:

```

If (kondisi1)
    Pernyataan_1;
Else if (kondisi2)
    Pernyataan_2;
Else if (kondisi3)
    Pernyataan_3;
Else
    Pernyataan;

```

Contoh program

```

if (dtkey == 0x0a)
{
}
else
    Data = dtkey;
    sprintf(buf, "%x ", Data);
    lcd_puts(buf);
}

```

2.5.1.11 Pernyataan Switch

Pernyataan switch di gunakan untuk melakukan pengambilan keputusan terhadap banyak kemungkinan. Bentuk pernyataan switch adalah sebagai berikut:

```
Switch (ekspresi)
{
    Case nilai_1 : pernyataan_1;break;
    Case nilai_2 : pernyataan_2;break;
    Case nilai_3 : pernyataan_3;break;
    ...
    Default      : pernyataan_default;break;
}
```

Pada pernyataan switch, masing-masing pernyataan (*pernyataan_1 sampai dengan pernyataan_default*) dapat berupa satu atau beberapa perintah dan tidak perlu berupa blok pernyataan. *Pernyataan_1* akan dikerjakan jika *ekspresi* bernilai sama dengan *nilai_1*, *pernyataan_2* akan dikerjakan jika *ekspresi* bernilai sama dengan *nilai_2*, *pernyataan_3* akan dikerjakan jika *ekspresi* bernilai sama dengan *nilai_3* dan seterusnya. *Pernyataan_default* bersifat opsional, artinya boleh dikerjakan apabila nilai *ekspresi* tidak ada yang sama satupun dengan salah satu *nilai_1*, *nilai_2*, *nilai_3* dan seterusnya. Setiap akhir dari pernyataan harus diakhiri dengan *break*, karena ini digunakan untuk keluar dari pernyataan switch.

Contoh:

Switch (PINA)

```
{
    case 1: dtkey=0x1;break;
    case 2: dtkey=0x4;break;
};
```

Pernyataan di atas berarti membaca port A, kemudian datanya (*PINA*) akan disesuaikan dengan nilai *CASE*. Jika *PINA* bernilai 1 maka data 0x00 akan

dikeluarkan ke port C kemudian program keluar dari pernyataan switch tetapi jika *PINA* bernilai 2 maka data 0x4 akan dikeluarkan ke port C kemudian program keluar dari pernyataan switch.

2.5.1.12 Pernyataan While

Pernyataan `while` digunakan untuk mengulangi sebuah pernyataan atau blok kenyataan secara terus menerus selama kondisi tertentu masih terpenuhi. Bentuk pernyataan `while` adalah sebagai berikut:

```
While (kondisi)
{
// sebuah pernyataan atau blok pernyataan
}
```

Jika pernyataan yang akan diulangi hanya sebuah pernyataan saja maka tanda `{` dan `}` bisa dihilangkan.

Contoh:

```
Unsigned char a = 0
```

```
...* *
```

```
While (a < 10)
```

```
{
PORT=a;
a++;
}
```

Pernyataan di atas akan mengeluarkan data `a` ke port C secara berulang-ulang. Setiap kali pengulangan nilai `a` akan bertambah 1 dan setelah nilai `a` mencapai 10 maka pengulangan selesai.

2.5.2 Software Code Vision AVR

Code Vision AVR merupakan salah satu software kompilasi yang khusus digunakan untuk mikrokontroler keluarga AVR, dan dari beberapa kompilasi C yang digunakan, Code Vision AVR merupakan yang terbaik dibandingkan kompilasi-kompilasi yang lain karena beberapa kelebihan yang dimiliki oleh CodeVision AVR antara lain :

- a. Menggunakan IDE (*Integrated Development Environment*)
- b. Fasilitas yang disediakan lengkap (mengedit program, mendownload program, mengkompilasi program)
- c. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas CodeWizardAVR
- d. Memiliki fasilitas untuk mendownload program langsung dari CodeVisionAVR dengan menggunakan hardware khusus seperti AtmelSTK500, Kanda System STK200+/300 dan beberapa hardware lain yang telah didefinisikan oleh CodeVisionAVR
- e. Memiliki fasilitas debugger sehingga dapat menggunakan software compiler lain untuk mengecek kode Assembler nya contoh AVRStudio
- f. Memiliki terminal komunikasi serial yang terintegrasi dalam CodeVisionAVR sehingga dapat digunakan untuk membantupengecekan program yang telah dibuat khususnya yang menggunakan fasilitas komunikasi serial UART

Adapun tahap – tahap dalam menggunakan Code Vision AVR

- a. Buka program Code Vision AVR dengan cara start menu → All Program → CodeVision AVR→CodeVision AVR C Compiler
- b. Untuk membuat Project Baru gunakan File → New maka akan muncul 2 pilihan yaitu source atau project. Karena kita akan membuat project dulu maka → pilih Project
- c. Setelah itu akan muncul konfirmasi apakah kita akan menggunakan fasilitas CodeWizardAVR atau tidak. Pilih Yes jika ingin menggunakan fasilitas CodeWizardAVR atau No jika tidak. Fasilitas CodeWizardAVR digunakan untuk membangkitkan kode program secara otomatis sesuai menu-menu yang

telah tersedia. Untuk project pertama ini kita akan menuliskan kode program sendiri sehingga kita pilih No

- d. Tuliskan dan simpan nama file project Anda. Sebagai contoh kita simpan dengan nam projectku.prj
- e. Pilih Tab C Compiler kemudian pilih Chip sesuai dengan IC mikrokontroler yang kita pakai serta atur Clock sesuai dengan frekuensi kristal yang digunakan
- f. Sampai disini Anda telah selesai membuat file project baru dengan nama projectku.prj. Langkah selanjutnya adalah Anda membuat file baru untuk menuliskan listing program yaitu dengan cara : File → New → pilih Source. Setelah itu akan muncul jendela baru yang dapat kita gunakan untk menulis program. Sebagai contoh dalam program pertama ini kita akan tuliskan listing program

```
#include <mega8535.h>
void main ( void )
{
    DDRC=0xFF
    PORTC=0x0F
}
```

Kemudian setelah itu kita simpan dengan cara File → Save dan diberi nama Led.c.

- g. Selanjutnya kita harus memasukan file program ini(led.c) kedalam file project yang telah kita buat sebelumnya (projectku.prj) dengan cara : Project → Configure → pilih tab Files → Add → pilih file led.c → Open, sekarang file program led.c yang anda tuliskan telah masuk dalam project.
- h. Langkah berikutnya adalah proses kompilasi. Untuk mengkompilasi program yang telah kita tuliskan tadi dilakukan dengan cara : Project → Make
Informasi diatas akan muncul pada akhir proses kompilasi dari informasi diatas kita dapat mengetahui Apabila dalam proses kompilasi ini terdapat beberapa eror, maka kita harus mencari sumber error tersebut kemudian membetulkannya dan mengkompilanya lagi sampai tidak terdapat error lagi, jika proses kompilasi berjalan dengan baik maka akan muncul beberapa file baru antar lain file led.asm, led.obj, led.hex dan file nled.eep. dan dalam project ini

file dengan ekstensi *.hex inilah yang akan didownload ke dalam mikrokontroler. Langkah berikut dan seterusnya adalah proses download program ke mikrokontroler

- i. Buka Program CodeVisionAVR Chip Programmer, caranya adalah tools I Chip Programmer, jangan lupa pastikan setting programmernya pada modul STK200
- j. Buka file program yang akan didownload ke mikrokontroler. Caranya adalah File I Load FLASH kemudian pilih nama file led.hex
- k. Download program ke mikrokontroler. Caranya adalah program I All
- l. Sekarang Anda bisa amati bagaimana menyala lampu LED yang terhubung ke PORTC, jika 4 buah menyala dan 4 buah mati maka project pertama Anda telah berhasil.