

BAB II

TINJAUAN PUSTAKA

II.1. Konsep Dasar

II.1.1. Perancangan

Perancangan dalam arti yang sederhana dapat dijelaskan sebagai suatu proses mempersiapkan hal-hal yang akan dikerjakan pada waktu yang akan untuk mencapai suatu tujuan yang telah ditetapkan terlebih dahulu. Suatu definisi mengenai perancangan memang diperlukan agar dalam uraian selanjutnya tidak terjadi kesimpangsiuran definisi pada umumnya merupakan suatu pintu gerbang untuk memasuki pengertian-pengertian yang ada kaitannya dengan istilah yang dipakai, dalam hal ini perancangan. Sudah sejak awal pelita I istilah ini digunakan secara luas baik di kalangan pendidikan maupun di luar lingkungan pendidikan namun belum ditetapkan suatu definisi secara resmi. Hingga kini perancangan itu sendiri belum merupakan suatu disiplin ilmu tersendiri, namun secara garis besar berbagai definisi yang dibuat oleh para ahli sesuai bidang keahliannya, ternyata mempunyai persamaan arti yang pokok, walaupun dalam susunan kalimatnya atau penekanannya terdapat perbedaan dari definisi tadi tidak benar.

Umumnya definisi mengenai perancangan mengenai perancangan tidak disusun berdasarkan teori semata-mata, melainkan diangkat dari pengalaman peraktek. Ada yang mendefinisikan perancangan sebagai suatu alat untuk mengatur system, penyesuaiannya kebutuhan dan aspirasi seseorang atau masyarakat (Prof. Dr. Jusuf Enoch M.A: 1992 : 1).

II.1.2. Kecepatan

Kecepatan adalah laju perubahan tempat yang diperoleh dari pengukuran jarak yang ditempuh ΔS dan selang waktu Δt yang diperlukan untuk menempuh jarak tersebut, yaitu

$$\overline{v} = \Delta s / \Delta t$$

Pada umumnya kecepatan dapat berubah-ubah, tidak tetap sehingga kecepatan yang didefinisikan menurut persamaan di atas adalah kecepatan rata-rata selama selang waktu Δt tersebut. Kecepatan disuatu tempat pada suatu saat, yaitu kecepatan setempat atau sesaat, pada hakikatnya tidak ada karena tidak mungkin diukur. Yang terukur adalah kecepatan di sekitar sesuatu tempat atau sesaat, pada hakikatnya tidak ada karena tidak mungkin diukur. Yang terukur adalah kecepatan di sekitar suatu tempat, sepanjang jarak tempuh yang amat pendek di sekitar suatu titik (Dr. Peter Soedjo, B.Sc: 2005 : 1).

II.1.3. Mikrokontroler

Mikrokontroler adalah sistem komputer yang dikemas dalam sebuah *IC*. *IC* tersebut mengandung semua komponen pembentuk komputer seperti *CPU*, *RAM*, *ROM*, *Port IO*. Berbeda dengan *PC* yang dirancang untuk kegunaan umum (general purpose), Mikrokontroler digunakan untuk tugas atau fungsi yang khusus (special purpose) yaitu mengontrol sistem tertentu.

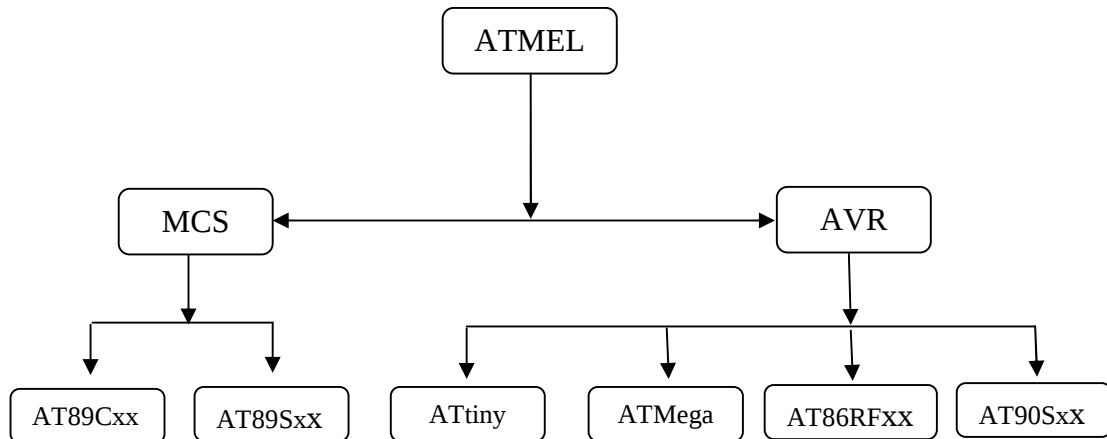
Mikrokontroler sering juga disebut sebagai *Embedded Microcontroller* yang berarti bahwa ia merupakan bagian dari *embedded system* menjadi satu bagian dari perangkat sistem atau sistem yang lebih besar. Secara fisik, kerja dari

sebuah *Mikrokontroller* dapat dijelaskan sebagai siklus pembacaan instruksi yang tersimpan di dalam memori. *Mikrokontroller* menentukan alamat dari memori program yang akan dibaca, dan melakukan proses baca data di memori. Data yang dibaca diinterpretasikan sebagai instruksi. Alamat instruksi disimpan oleh *Mikrokontroller* diregister, yang dikenal sebagai program counter (Sigit Firmansyah : 2005 : 277).

II.2. Mikrokontroler ATmega8535

Meurut Bagus Hari Sasongko : 2012 : 1. teknologi *Mikroprosesor* telah mengalami perkembangan. Hal sama telah terjadi pada *Mikrokontroller*. Jika pada *Mikroprosesor* terdahulu menggunakan teknologi *CISC* seperti prosesor intel 386/486 maka pada *Mikrokontroller* produksi *ATMEL* adalah jenis *MCS* (*AT89C51*, *AT89S51* dan *AT89S52*). Setelah mengalami perkembangan, teknologi *Mikroprosesor* dan *Mikrokontroller* mengalami peningkatan yang terjadi pada kisaran tahun 1996 s/d 1998 *ATMEL* mengeluarkan teknologi *Mikrokontroller* terbaru berjenis *AVR* (*Alf and Vegard's Risc processor*) yang menggunakan teknologi *RISC* (*Reduce Instruction Set Computer*) dengan keunggulan lebih banyak dibandingkan pendahulunya, yaitu *Mikrokontroller* jenis *MCS*. *Mikrokontroller* jenis *MCS* memiliki kecepatan kerja 1/12 kali frekuensi osilator yang digunakan sedangkan pada kecepatan frekuensi *AVR* sama dengan kecepatan kerja frekuensi osilator yang digunakan. Jadi apabila menggunakan frekuensi osilator yang sama, maka *AVR* mempunyai kecepatan 12 kali lebih cepat dibandingkan dengan *MCS*.

Adapun diagram blok perkembangan *Mikrokontroller ATMEL* dapat dilihat pada gambar II.1 pada gambar di bawah ini.

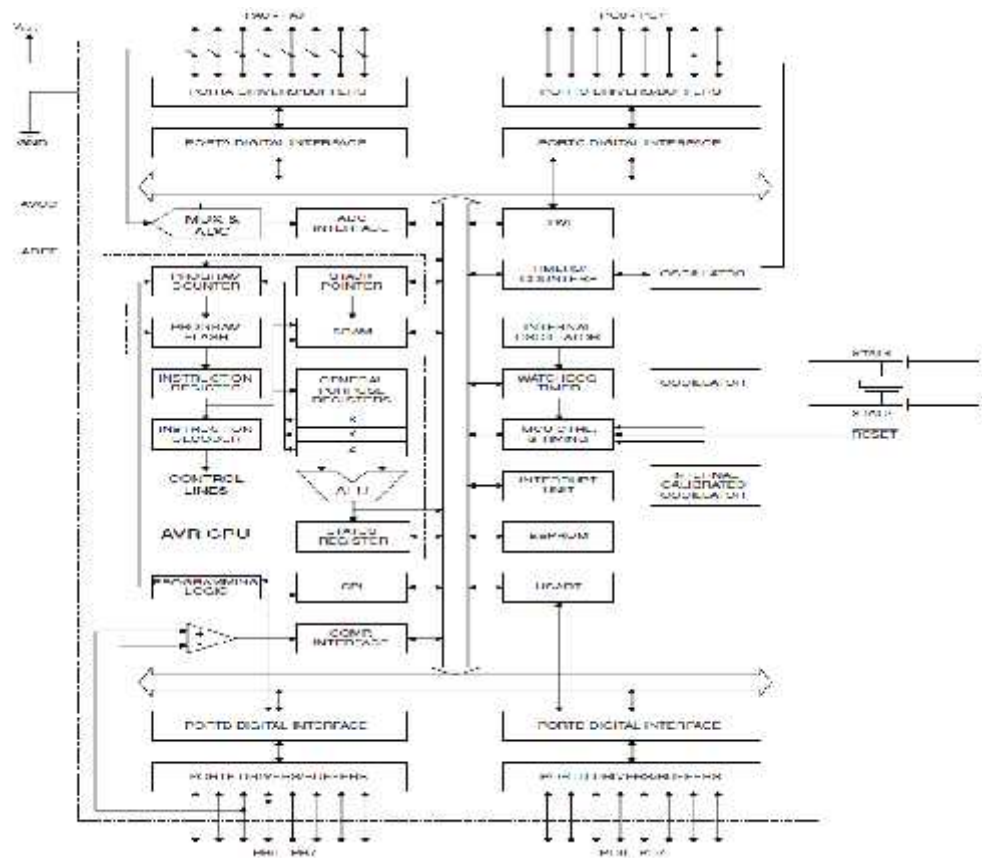


Gambar II.1 Diagram Blok Contoh Perkembangan Mikrokontroller ATMEL
(Sumber : Bagus hari Sasongko 2012 : 1)

ATMega8535 merupakan salah satu *Mikrokontroller* 8 bit buatan *Atmel* untuk keluarga *AVR* yang diproduksi secara masal pada tahun 2006. Karena merupakan keluarga *AVR*, maka *ATMega8535* juga menggunakan arsitektur *RISC* (*Reduced Instruction Set Computing*). *AVR* atau sebuah kependekan dari *Alf and Vegard's Risc Processor* merupakan *chip Mikrokontroller* yang diproduksi oleh *Atmel*, yang secara umum dapat dikelompokkan ke dalam 4 kelas :

1. *ATtiny*
2. *ATMega*
3. *AT90Sxx*
4. *AT86RFxx*

II.1.1. Diagram Blok Fungsional ATmega8535



Gambar II.2 Diagram Blok Fungsional ATmega8535
(Sumber : Bagus hari Sasongko 2012 : 1)

Keterangan gambar II.2 diatas memperlihatkan bahwa *ATmega8535*

memiliki bagian sebagai berikut:

Saluran *I/O* sebanyak 32 bit buah, yaitu *PortA*, *PortB*, *PortC*, dan *Port D*.

1. *ADC* 10 bit sebanyak 8 Saluran.
2. Tiga buah *Timer/Counter* dengan kemampuan pembandingan.
3. *CPU* yang terdiri atas 32 buah *register*.
4. *Watchdog Timer* dengan osilator *internal*.
5. *SRAM* sebesar 512 byte.
6. *Memori Flash* sebesar 8 Kb dengan kemampuan *Read While Write*.

7. Unit interupsi *internal* dan *external*.
8. Port antarmuka *SPI*.
9. *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte yang deprogram saat operasi.
10. Antarmuka kompurator *Analog*.
11. Port *USART* untuk komunikasi serial dengan kecepatan maksimal 12,5 Mbps.
12. System *Mikroposeor* 8 bit berbaris *RISC* dengan kecepatan maksimal 16 MHz.

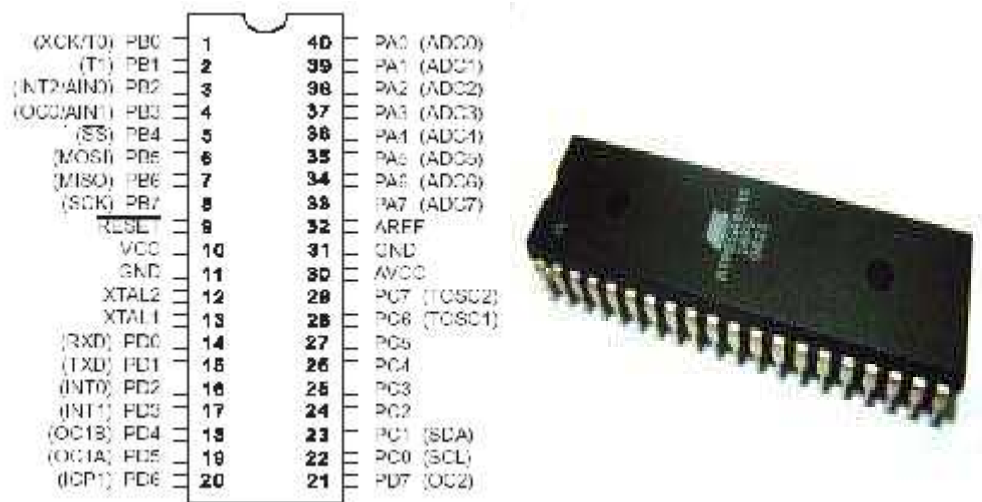
II.2.2. Konfigurasi Pin ATmega8535

Berikut adalah konfigurasi *PIN* dari *Mikrokontroler ATmega8535* :

- a. *VCC* merupakan *pin* yang berfungsi sebagai *pin* masukan catudaya
- b. *GND* merupakan *pin* *Ground*
- c. *PortA* (*PA0...PA7*) merupakan *pin I/O* dan *pin* masukan *ADC*
- d. *PortB* (*PB0...PB7*) merupakan *pin I/O* dan *pin* yang mempunyai fungsi khusus yaitu *Timer/Counter*, komparator *Analog* dan *SPI*
- e. *PortC* (*PC0...PC7*) merupakan *port I/O* dan *pin* yang mempunyai fungsi khusus, yaitu komparator *analog* dan *Timer Oscillator*
- f. *PortD* (*PD0...PD1*) merupakan *port I/O* dan *pin* fungsi khusus yaitu komparator *analog* dan *interrupt eksternal* serta komunikasi serial
- g. *RESET* merupakan *pin* yang digunakan untuk mereset *Mikrokontroler*
- h. *XTAL1* dan *XTAL2* merupakan *pin* masukan *clock eksternal*
- i. *AVCC* merupakan *pin* masukan untuk tegangan *ADC*

j. *AREF* merupakan *pin* masukan tegangan referensi untuk ADC

Adapun bentuk dan konfigurasi *PIN* Mikrokontroler *ATMega8535* dapat dilihat pada gambar II.3 dibawah ini.



Gambar II.3 Bentuk dan Konfigurasi PIN Mikrokontroler ATMega8535
(Sumber : Bagus hari Sasongko 2012 : 1)

Adapun keterangan tiap-tiap *pin* pada *ATMega8535* dapat di lihat pada tabel II.1 di bawah ini.

Tabel II.1 Keterangan pin – pin ATMega8535

No. Pin	Nama	Fungsi
1	PBO(XCK/TO)	Port B.0 / Counter / clock eksternal untuk USART (xck)
2	PB1 (T1)	Port B.1 / Counter 1
3	PB2 (INT2/AIN0)	Port B.2 / Input (+) Analog Komparator (AIN0) dan interupsi eksternal 2 (INT2)
4	PB3 (OC0/AIN1)	Port B.3 / Input (-) Analog Komparator (AIN1) dan output PWM 0
5	PB4 (SS)	Port B.4 / SPI Slave Select Input (SS)
6	PB5 (MOSI)	Port B.5 / SPI bus master Out Slave In
7	PB6 (MISO)	Port B.6 / SPI bus master In Slave Out
8	PB7 (SJK)	Port B.7 / Sinyal Clock Serial SPI
9	RESET	Me-Reset Mikrokontroler
10	VCC	Catu Daya (+)
11	GND	Sinyal Ground terhadap catu daya

12-13	XTAL2 – XTAL1	Sinyal Input Clock eksternal (kristal)
14	PDO (RXD)	Port D.0 / penerima data serial
15	PD1 (TXD)	Port D.1 / pengirim data serial
16	PD2 (INT0)	Port D.2 / Interupsi eksternal 0
17	PD3 (INT1)	Port D.3 / Interupsi eksternal 1
18	PD4 (OC1)	Port D.4 / Pembanding Timer-Counter 1
19	PD5 (OC1A)	Port D.5 / Output PWM 1A
20	PD6 (ICP1)	Port D.6 / Timer-Counter 1 Input
21	PD7 (OC2)	Port D.7 / Output PWM 2
22	PC0 (SCL)	Port C.0 / Serial bus clock line
23	PC1 (SDA)	Port C.1 / Serial bus data input-output
No. Pin	Nama	Fungsi
24-27	PC2 – PC5	Port C.2 – Port C.5
28	PC6 (TOSC1)	Port C.6 / Timer osilator 1
29	PC7 (TOSC2)	Port C.7 / Timer osilator 2
30	AVCC	Tegangan ADC
31	GND	Sinyal Ground ADC
32	AREFF	Tegangan Referensi ADC
33-40	PA0 (ADC0) – PA7 (ADC7)	Port A.0 – Port A.7 dan input untuk ADC (8 chanel : ADC0 – ADC7)

(Sumber : Bagus hari Sasongko 2012 : 1)

II.2.3. Deskripsi pin-pin pada mikrokontroler ATmega8535 :

1. Port A

Merupakan 8-bit *directional port I/O*. Setiap pinnya dapat menyediakan *internal pull-up resistor* (dapat diatur per bit). *Output buffer PortA* dapat memberi arus 20 mA dan dapat mengendalikan display *LED* secara langsung. Data *Direction Register portA (DDRA)* harus disetting terlebih dahulu sebelum *PortA* digunakan. *Bit-bit DDRA* diisi 0 jika ingin memfungsikan *pin-pin portA* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, kedelapan *pin portA* juga digunakan untuk masukan sinyal *analog* bagi *A/D converter*

2. PortB.

Merupakan 8-bit *directional port I/O*. Setiap pinnya dapat menyediakan *internal pull-up resistor* (dapat diatur per bit). *Output buffer PortB* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. Data *Direction Register portB (DDRB)* harus disetting terlebih dahulu sebelum *PortB* digunakan. Bit-bit *DDRB* diisi 0 jika ingin memfungsikan *pin-pin portB* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. *Pin-pin portB* juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel berikut.

Adapun fungsi-fungsi pin *portB* yang dimiliki *Mikrokontroler ATmega8535* dapat dilihat pada tabel II.2 dibawah ini.

Tabel II.2 Fungsi Pin Port B

Port Pin	Fungsi Khusus
PB0	T0 – timer/counter 0 external counter input
PB1	T1 – timer/counter 0 external counter input
PB2	AIN0 = analog comparator positive input
PB3	AIN1 – analog comparator negative input
PB4	SS – SPI slave select input
PB5	MOSI – SPI bus master output / slave input
PB6	MISO – SPI bus master input / slave output
PB7	SCK – SPI bus serial clock

(Sumber : Bagus hari Sasongko 2012 : 31)

3. Port C

Merupakan 8-bit *directional port I/O*. Setiap pinnya dapat menyediakan *internal pull-up resistor* (dapat diatur per bit). *Output buffer PortC* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. Data *Direction Register portC (DDRC)* harus disetting terlebih dahulu sebelum *PortC* digunakan. Bit-bit *DDRC* diisi 0 jika ingin memfungsikan *pin-pin portC* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, dua *pin port C (PC6 dan PC7)* juga memiliki fungsi alternatif sebagai *oscillator* untuk *timer/counter 2*.

4. PortD

Merupakan 8-bit *directional port I/O*. Setiap pinnya dapat menyediakan *internal pull-up resistor* (dapat diatur per bit). *Output buffer PortD* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. Data *Direction Register portD (DDRD)* harus disetting terlebih dahulu sebelum *PortD* digunakan. Bit-bit *DDRD* diisi 0 jika ingin memfungsikan *pin-pin portD* yang bersesuaian sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin port D* juga memiliki untuk fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel II.3.

Tabel II.3 Fungsi Pin Port D

Port Pin	Fungsi Khusus
PD0	RDX (UART input line)
PD1	TDX (UART output line)
PD2	INT0 (external interrupt 0 input)
PD3	INT1 (external interrupt 1 input)
PD4	OC1B (Timer/Counter1 output compareB match output)
PD5	OC1A (Timer/Counter1 output compareA match output)
PD6	ICP (Timer/Counter1 input capture pin)
PD7	OC2 (Timer/Counter2 output compare match output)

(Sumber : Bagus hari Sasongko 2012 : 32)

5. RESET

RST pada pin 9 merupakan reset dari AVR. Jika pada pin ini diberi masukan low selama minimal 2 machine cycle maka system akan di-reset.

6. XTAL1

XTAL1 adalah masukan ke inverting oscillator amplifier dan input ke internal clock operating circuit.

7. XTAL2

XTAL2 adalah output dari inverting oscillator amplifier.

8. AVcc

Avcc adalah kaki masukan tegangan bagi A/D Converter. Kaki ini harus secara eksternal terhubung ke Vcc melalui lowpass filter.

9. AREF

AREF adalah kaki masukan referensi bagi *A/D Converter*. Untuk operasionalisasi *ADC*, suatu level tegangan antara *AGND* dan *Avcc* harus dibeikan ke kaki ini.

10. *AGND*

AGND adalah kaki untuk *analog ground*. Hubungkan kaki ini ke *GND*, kecuali jika *board* memiliki *analog ground* yang terpisah.

II.3. Perangkat Keras Pendukung (*Support Hardware*)

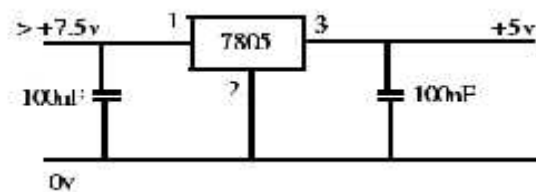
II.3.1. Rangkaian *Regulator IC 7805*

Menurut Ade Kurniawan : 2011 : 67. *Ragulasi voltase* untuk catu daya seringkali dibutuhkan dalam rangkaian *elektronika*, maka tersedia berbagai jenis *IC* yang memenuhi kebutuhan ini. Salah satu *IC* adalah seri 78xx, dimana xx adalah menunjukkan *voltase* keluaran dari *IC* tersebut. Terdapat xx = 05 untuk 5V, xx = 75 untuk 7.5v, xx = 09 untuk 9v, xx = 12 untuk 12V, xx = 15 untuk 15V dan juga terdapat *voltase* yang lebih tinggi. Untuk skripsi ini penulis memakai *IC Regulator 7805* sebagai catu daya dengan *voltase* 5V.

IC 78xx mempunyai tiga kaki, satu untuk V_{in} satu untuk V_{out} dan satu lagi untuk *GND*. Sambungan tersebut diperlihatkan dalam gambar II.4. dalam *IC* ini selain rangkaian *regulasi voltase* juga terdapat rangkaian pengaman yang melindungi *IC* dari arus atau daya yang terlalu tinggi. Terdapat pembatasan arus yang mengurangi *voltase* keluaran kalau batas arus terlampaui. Besar dari batas arus ini tergantung dari *voltase* pada *IC* sehingga arus maksimal lebih kecil kalau selisih *voltase* antara V_{in} dan V_{out} lebih besar. Juga terdapat pengukuran suhu yang mengurangi arus maksimal kalau suhu *IC* menjadi terlalu tinggi. Dengan

rangkaian – rangkaian pengaman ini IC terlindung dari kerusakan sebagai akibat beban yang terlalu besar.

Adapun Perangkat Keras Rangkaian *Regulator 7805* dapat dilihat pada gambar II.4 dibawah ini.



Gambar II.4 Rangkaian Regulator 7805
(Pengarang : Ade Kurniawan 2011: 67)

II.3.2. Servo Motor

Servo motor dilengkapi dengan motor *DC* untuk mengendalikan posisi sebuah robot. *Rotor* motor dapat diputar/diposisikan hingga 180 derajat. *Servo* motor *continous* dapat berputar hingga 360 derajat. *Servo* motor biasa digunakan untuk mengendalikan gerak dari toys (mainan) seperti model mobil, pesawat, perahu dan helikopter. *Servo* motor adalah *DC* motor kualitas tinggi yang memenuhi syarat untuk digunakan pada aplikasi servo seperti *closed control loop*. Motor tersebut harus dapat menangani perubahan yang cepat pada posisi, kecepatan dan percepatan, serta harus mampu menangani *intermittent torque*. *Servo* adalah *DC* motor dengan tambahan elektronika untuk kontrol *PW* dan digunakan untuk tujuan hobbyist, pada pesawat terbang model, mobil atau kapal. *Servo* mempunyai 3 kabel, yaitu *Vcc*, *Ground* dan *PV input*. Tidak seperti *PWM* pada *DC* motor, *input* sinyal untuk servo tidak digunakan mengatur kecepatan, tetapi digunakan untuk mengatur posisi dari putaran *servo*.

Sinyal *PW* yang digunakan untuk *servo* mempunyai frekuensi 50 Hz sehingga pulsa dibuat setiap 20ms. Sebagai contoh, sebuah pulsa 0.7ms akan memutar disk *servo* ke posisi kiri, dan pulsa 1.7ms akan memutar disk ke posisi kanan. Kekurangan *servo* adalah ia tidak menyediakan *feedback*. Umumnya kita membeli *servo continous* karena dapat berputar 360 derajat, namun anehnya harganya lebih murah dibandingkan *servo* standar yang derajat putarannya terbatas (Widodo Budiharto : 2009 : 7).

II.3.3. Resistor

Resistor adalah komponen elektrik yang berfungsi memberikan hambatan terhadap aliran arus listrik. Setiap benda adalah *resistor* karena dasarnya tiap benda dapat memberikan hambatan listrik. Dalam rangkaian listrik dibutuhkan *resistor* dengan spesifikasi tertentu, seperti besar hambatan, arus maksimum yang boleh dilewatkan dan karakteristik hambatan terhadap suhu dan panas.

Pada peralatan seperti radio dan *amplifier* (model lama) sering dijumpai terdapat pengatur volume atau nada yang menggunakan tombol yang dapat diputar, tombol tersebut adalah salah satu contoh *resistor* variabel, yaitu *resistor* yang dapat diubah-ubah nilai hambatannya. Perubahan resistansi akan mengubah besar arus yang menggerakkan membran speaker. (Sigit Firmansyah : 2005 : 30).

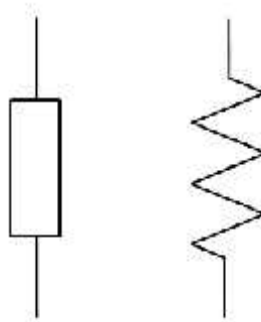
Adapun Perangkat Keras Rangkaian *Resistor* Variabel dapat dilihat pada gambar II.5 dibawah ini.



Gambar II.5 Resistor Variabel
(Sumber : Sigit Firmansyah 2005 : 30)

Pada skematik rangkaian, *resistor* disimbolkan sebagai garis zig-zag atau kotak dengan garis di kanan dan kirinya.

Adapun skematik rangkaian, *resistor* disimbolkan sebagai garis zig-zag dapat dilihat pada gambar II.6 dibawah ini.



Gambar II.6 Resistor Variabel
Sumber : Sigit Firmansyah (2005 : 33)

Bentuk *resistor* yang umum adalah silinder dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang kode warna untuk mengenali besar *resistansi*, sehingga kita dapat mengetahui besar *resistansinya* tanpa harus mengukur dengan *ohm* meter. Kode warna tersebut ditetapkan oleh standar manufaktur yang dikeluarkan oleh *EIA (Electronic Industries Association)*.

Adapun Nilai Hambatan Tiap Warna skematik rangkaian pada resistor dapat dilihat pada tabel II.4 dibawah ini.

Tabel II.4 Nilai Hambatan Tiap Warna Pada Resistor

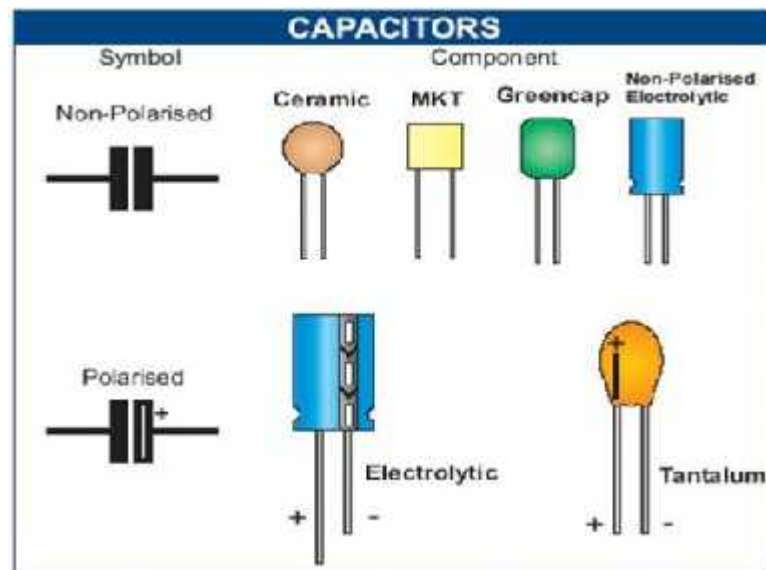
Warna	Pita pertama	Pita kedua	Pita ketiga (pengali)	Pita keempat (toleransi)	Pita kelima (koefisien suhu)
Hitam	0	0	$\times 10^0$		
Coklat	1	1	$\times 10^1$	$\pm 1\%$ (F)	100 ppm
Kuning	2	2	$\times 10^2$		
Biru	3	3	$\times 10^3$		10 ppm
Hijau	4	4	$\times 10^4$		25 ppm
Biru	5	5	$\times 10^5$	$\pm 0.5\%$ (D)	
Pink	6	6	$\times 10^6$	$\pm 0.25\%$ (C)	
Ungu	7	7	$\times 10^7$	$\pm 0.1\%$ (B)	
Abu-abu	8	8	$\times 10^8$	$\pm 0.05\%$ (A)	
Putih	9	9	$\times 10^9$		
Emas			$\times 10^{-1}$	$\pm 5\%$ (J)	
Perak			$\times 10^{-2}$	$\pm 10\%$ (K)	
Kosong				$\pm 20\%$ (M)	

(Sumber : Sigit Firmansyah 2005: 33)

II.3.4. Kapasitor

Kapasitor adalah komponen elektrik yang berfungsi menyimpan muatan listrik. Salah satu jenis *kapasitor* adalah *kapasitor* keping sejajar. *Kapasitor* ini terdiri atas dua buah keping metal sejajar yang dipisahkan oleh *isolator* yang disebut dielektrik. Bila kapasitor dihubungkan ke baterai *kapasitor* terisi hingga beda potensial antara kedua terminalnya sama dengan tegangan baterai. Jika baterai dicabut, muatan-muatan listrik akan habis dalam waktu yang sangat lama, terkecuali bila sebuah konduktor dihubungkan pada kedua terminal *kapasitor*. (Ade Kurniawan : 2011 : 20).

Adapun rangkain perangkat keras pada rangkaian elektronik *Kapasitor polar* dan *Non-Polar* dapat dilihat pada gambar II.7 pada gambar di bawah ini.



Gambar : II.7 Kapasitor polar dan Non-Polar
(Sumber : Ade Kurniawan 2011 : 20)

II.3.5. Jenis – jenis Kapasitor

Kapasitor biasanya diberi label sesuai dengan jenis dielektrik yang digunakan. Misalnya, *kapasitor* keramik, *kapasitor* mika, *kapasitor* polistiren, dan banyak jenis lainnya. *Kapasitor* jenis ini termasuk *kapasitor fixed*, artinya telah memiliki nilai yang konstan. Selain itu, dikenal *kapasitor* yang dapat diubah nilai kapasitansinya dengan memutar bagian tertentu dari *kapasitor*. *Kapasitor* jenis ini biasanya disebut *kapasitor* variabel.

Kapasitor pada umumnya tidak memiliki polaritas, namun ada kelas khusus dari *kapasitor* yang memiliki polaritas. Artinya, pemasangan tidak boleh terbalik karena memiliki kutub positif dan negatif. *Kapasitor* jenis ini biasanya disebut *kapasitor elco* (elektrolit). Biasanya *kapasitor* jenis ini memiliki resistansi yang besar dalam orde mikro Farad. (Ade Kurniawan : 2011 : 20).

II.3.6. LCD (*Liquid Crystal Display*)

LCD (Liquid Crystal Display atau dapat di bahasa Indonesia-kan sebagai tampilan kristal cair) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. *LCD* dapat memunculkan gambar atau dikarenakan terdapat banyak sekali titik cahaya (piksel) yang terdiri dari satu buah kristal cair sebagai sebuah titik cahaya. Walau disebut sebagai titik cahaya, namun kristal cair ini tidak memancarkan cahaya sendiri. Sumber cahaya di dalam sebuah perangkat *LCD* adalah lampu neon berwarna putih di bagian belakang susunan kristal cair tadi. Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan inilah yang membentuk tampilan citra. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya akan hanya membiarkan beberapa warna diteruskan sedangkan warna lainnya tersaring. Dalam menampilkan karakter untuk membantu menginformasikan proses dan *control* yang terjadi dalam suatu program robot kita sering menggunakan *LCD* juga. *LCD* yang digunakan untuk pembuatan skripsi ini adalah *LCD* dengan banyak karakter 16x2, maksudnya 16 menyatakan kolom dan 2 menyatakan baris.

Adapun dari *LCD* (Liquid Crystal Display) rangkaian perangkat keras dapat dilihat pada gambar II.8 pada gambar dibawah ini.



Gambar II.8 LCD 16x2

(Sumber : Arif Setiawan 2011 : 24)

Adapaun keterangan tiap-tiap pin pada *LCD* dapat di lihat pada tabel II.5 di bawah ini.

Tabel II.5 pin-pin Pada LCD

PIN	Nama Pin	Fungsi
1	VSS	<i>Ground voltage</i>
2	VCC	+5V
3	VEE	<i>Contrast Voltage</i>
4	RS	<i>Register Select</i> 0 = <i>write mode</i> 1 = <i>read mode</i>
5	R/W	<i>Read / Write, to choose write or read mode</i> 0 = <i>write mode</i> 1 = <i>read mode</i>
6	E	<i>Enable</i> 0 = <i>start to lacht data to LCD character</i> 1 = <i>disable</i>
7	DB0	<i>Data bit ke-0 (LSB)</i>
8	DB1	<i>Data bit ke-1</i>
9	DB2	<i>Data bit ke-2</i>
10	DB3	<i>Data bit ke-3</i>
11	DB4	<i>Data bit ke-4</i>
12	DB5	<i>Data bit ke-5</i>
13	DB6	<i>Data bit ke-6</i>
14	DB7	<i>Data bit ke-7</i>
15	BPL	<i>Back Plane Light</i>
16	GND	<i>Ground voltage</i>

(Sumber : Arif Setiawan 2011 : 24)

Keunggulan *LCD* adalah hanya menarik arus yang kecil (beberapa *mikro ampere*), sehingga alat atau system menjadi portable karena dapat menggunakan catu daya yang kecil. Keunggulan lainnya adalah tampilan yang diperlihatkan dapat dibaca dengan mudah dibawah terang sinar matahari. Di bawah sinar cahaya yang remang-remang atau dalam kondisi gelap, sebuah lampu (berupa *LED*) harus dipasang di belakang layar tampilan.

LCD yang digunakan adalah jenis *LCD* yang menampilkan data dengan 2 baris tampilan pada *display*. Keuntungan dari *LCD* ini adalah :

1. Dapat menampilkan karakter *ASCII*, sehingga dapat memudahkan untuk membuat program tampilan.
2. Mudah dihubungkan dengan *port I/O* karena hanya menggunakan 8 *bit* data dan 3 *bit* kontrol.
3. Ukuran modul yang proposional.
4. Daya yang digunakan relative sangat kecil.

II.3.7. Buzzer

Buzzer adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Pada umumnya *buzzer* digunakan untuk alarm, karena penggunaannya cukup mudah yaitu dengan memberikan tegangan input maka *buzzer* akan mengeluarkan bunyi. Frekuensi suara yang di keluarkan oleh *buzzer* yaitu antara 1-5 KHz.

Adapun bentuk dari perangkat keras rangkain elektronik *Buzzer* dapat dilihat pada gambar II.1 pada gambar di bawah ini.



Gambar II.9 Buzzer
(Sumber : Arif Setiawan 2011 : 44)

II.3.8. Sensor

Dalam rangkaian elektronika untuk keperluan pengukuran atau deteksi, diperlukan suatu bagian yang disebut *sensor*. *Sensor* berfungsi untuk menubah besaran yang bersifat fisis atau suhu, tekanan, berat, atau intensitas cahaya menjadi besaran listrik (tegangan atau arus listrik).

Sensor memiliki suatu ukuran yang disebut sensitivitas. Sensitivitas menunjukkan seberapa besar pengaruh perubahan nilai besaran *fisis* yang diukur oleh sensor terhadap keluaran dari sensor tersebut. Misalnya, sebuah sensor suhu yang tegangan keluarannya berubah 0,1 V jika terjadi perubahan suhu sebesar 1°C . Maka sensor suhu tersebut dapat dikatakan memiliki sensitivitas sebesar $0,1\text{V}/1^{\circ}\text{C}$. (Endra Pitowarno : 2006 : 56)

Sensor yang baik memiliki ciri-ciri sebagai berikut:

1. Peka terhadap besaran yang akan diukur.
2. Tidak peka terhadap besaran lain yang tidak akan diukur.
3. Keberadaan sensor tidak mempengaruhi besaran yang akan diukur.

II.3.9. Sinar Laser

Laser adalah singkatan dari *Light Amplification by Stimulated Emission and Radiation*. Telah diperkenalkan pada tahun 1958. Laser banyak dipakai dalam bidang-bidang *optic*, fisika, teknologi dan kedokteran. *Laser* digunakan untuk menghasilkan suatu berkas cahaya tampak yang intensitasnya kuat, monokromatis dan koheren(dengan arah tunggal arah tunggal).

Untuk mengerti cara kerjanya *laser*, maka terlebih dahulu harus mengerti istilah-istilah seperti emisi stimulasi/pancaran terangsang (*stimulated emission*). *Optical pumping*, *inverse populasi* (*population inversion*)

Laser yang sering digunakan adalah laser dari Helium-neon(*He-Ne*). *Laser* tersebut terdiri dari tabung yang berisi Helium dan Neon dalam perbandingan 7:1 pada tekanan *mmHg* (1torr). Atom-atom Helium adalah dalam keadaan tereksitasi pada *state*-metastabil akibat atom He yang tereksitasi bertumbukan dengan atom-atom neon yang tidak tereksitasi dan mentransfer tenagannya sehingga menghasilkan suatu *inverse populasi* dalam atom-atom Neon. Sinar *laser* terjadi pada atom-atom neon ini dengan transisi yang bersesuaian dengan panjang gelombang 6328 Å, 11523 Å dan 3,39 μm.(Prof. Dr. Mulyono:andi: 2005 :87-91).

Adapun bentuk dari perangkat keras rangkaian elektronik *Laser* (*Amplification by Stimulated Emission and Radiation*) dapat dilihat pada gambar II.10 pada gambar di bawah ini.



Gambar II.10 Sensor Laser
(Sumber : Prof. Dr. mulyono 2005 : 2011)

II.4. Perangkat Lunak (Software)

II.4.1. Pemrograman C

Pada suatu pengontrolan alat, program yang digunakan adalah pemrograman bahasa C. Untuk itu diperlukan juga pemahaman tentang pemrograman tersebut. Berikut adalah penjelasan dasar-dasar dari pemrograman bahasa C :

1. Tipe Data

Berikut ini adalah tipe-tipe data yang ada dalam bahasa C dan yang dikenali oleh *CodeVisionAVR* :

Adapun Bentuk Tipe Data pada pemrograman C dapat dilihat pada gambar II.6 tabel ini.

Tabel II.6 Tipe Data

Tipe Data	Ukuran	Jangkauan Nilai
Bit	1 bit	0 atau 1
Char	1 byte	-128 s/d 225
Unsigned Char	1 byte	0 s/d 225
Signed Char	1 byte	-128 s/d 127
Int	2 byte	-32.768 s/d 32.767
Short Int	2 byte	32.768 s/d 32.767
Unsigned Int	2 byte	0 s/d 65.535
Signed Int	2 byte	-32.768 s/d 32.767
Long Int	4 byte	-2.147.483.648 s/d 2.147.483.647
Unsigned Long Int	4 byte	0 s/d 4.294.967.295
Signed Long Int	4 byte	-2.147.483.648 s/d 2.147.483.647
Float	4 byte	1.2×10^{-38} s/d 3.4×10^{-38}
Double	4 byte	1.2×10^{-38} s/d 3.4×10^{-38}

2. Konstanta dan Variabel

Konstanta dan variabel merupakan sebuah tempat untuk menyimpan data yang berada di dalam memori. Konstanta berisi data yang nilainya tetap dan tidak dapat diubah selama program dijalankan, sedangkan variabel berisi data yang bisa berubah nilainya saat program dijalankan. Untuk membuat sebuah konstanta atau variabel maka kita harus mendeklarasikannya lebih dahulu, yaitu dengan sintaks berikut :

Const [tipe_data][nama_konstanta]=[nilai]

Contoh :

Const char konstantaku=0x10;

Deklarasi variabel :

[tipe_data][nama_variabel]=[nilai_awal]

Contoh :

Char variabelku;

```
Char variabelku=0x20;
```

```
Bit variabel_bit;
```

```
Bit variabel_bit=1;
```

Pada deklarasi variabel, [nilai_awal] bersifat operasional sehingga boleh diisi dan boleh tidak diisi. Nilai_awal merupakan nilai *default* variabel tersebut dan jika tidak diisi maka nilai *defaultnya* adalah 0 (nol). Beberapa variabel dengan tipe yang sama dapat dideklarasikan dalam satu baris seperti contoh berikut :

```
Char data_a, data_b, data_c;
```

3. Komentar

Komentar adalah tulisan yang tidak dianggap sebagai bagian dari tubuh program. Komentar digunakan untuk memberikan penjelasan, informasi ataupun keterangan-keterangan yang dapat membantu mempermudah dalam memahami kode program baik bagi si pembuat program maupun bagi orang lain yang membacanya. Komentar yang hanya satu baris ditulis dengan diawali `'//'` sedangkan komentar yang lebih dari satu baris diawali dengan `'/*'` dan diakhiri dengan `'*/'`.

Contoh :

```
// Ini adalah komentar satu baris
```

```
/* Sedangkan yang ini adalah komentar yang lebih dari satu baris*/
```

Selain digunakan untuk memberikan keterangan program, komentar juga dapat digunakan untuk membantu dalam pengujian program yaitu dengan menonaktifkan dan mengaktifkan kembali bagian program tertentu selama proses pengujian.

4. Pengarah Preprocessor

Pengarah *preprocessor* digunakan untuk mendefenisikan *processor* yang digunakan, dalam hal ini adalah untuk mendefenisikan jenis *Mikrokontroller* yang digunakan. Dengan pengarah preprosesor ini maka pendeklarasian *register-register* dan penamaannya dilakukan pada *file* yang lain yang disisipkan dalam program utama dengan sintaks sebagai berikut :

```
# include <nama_preprocessor>
```

Contoh :

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

5. Pernyataan

Pernyataan adalah satu buah instruksi lengkap berdiri sendiri.

```
PORTC = 0x0F;
```

Pernyataan diatas merupakan sebuah instruksi untuk mengeluarkan data 0x0F ke *PortC*.

Contoh sebuah blok pernyataan :

```
{
PORTA=0x00; // pernyataan_1
PORTB=0x0F; // pernyataan_2
PORTC=0xFF; // pernyataan_3
}
```

6. Operator Aritmatika

Operator aritmatika adalah beberapa operator yang digunakan untuk melakukan perhitungan aritmatika.

Adapun Bentuk operator Aritmatika pada pemerograman C dapat dilihat pada trabel II.7 tabel ini.

Tabel II.7 Operator Aritmatika

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

7. Operator Logika

Operator logika digunakan untuk membentuk suatu logika atas dua buah kondisi atau lebih. Berikut ini adalah tabel operator logika :

Adapun Bentuk Loika Aritmatika pada pemerograman C dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.8 Operator Logika

Operator	Keterangan
&&	Operator untuk logika AND
	Operator untuk logika OR
!	Operator untuk logika NOT

Contoh :

```
if ( (a==b) && (c!=d) ) = 0xFF;
```

Pernyataan diatas terdiri dari 2 buah kondisi yaitu $a==b$ dan $c!=d$ yang keduanya dihubungkan dengan logika && (AND). Jika logika yang dihasilkan benar maka perintah $PORTC = 0xFF$ akan dikerjakan dan jika salah tidak akan dikerjakan.

8. Operator Penambahan dan Pengurangan

Operator penambahan dan pengurangan adalah operator yang digunakan untuk menaikkan atau menurunkan nilai sebuah variabel dengan selisih 1.

Adapun Bentuk Operator Penambahan dan Pengurangan pada pemrograman C dapat dilihat pada tabel II.1 dibawah ini.

Tabel II.9 Operator Penambahan dan Pengurangan

Operator	Keterangan
++	Operator untuk penambahan nilai variabel dengan 1
--	Operator pengurangan nilai variabel dengan 1

Contoh :

```
a = 1; b = 5;
```

```
a++; b--;
```

Maka operator `a++` akan mengubah variable `a` dari 1 menjadi 2 dan operator `b--` akan mengubah *variable* `b` dari 5 menjadi 4.

9. Pernyataan If

Pernyataan *If* digunakan untuk melakukan pengambilan keputusan terhadap dua buah kemungkinan yaitu mengerjakan suatu *blok* pernyataan atau tidak. Bentuk pernyataan *If* adalah :

```
if (kondisi)
{
// blok pernyataan yang akan dikerjakan
// jika kondisi if terpenuhi }
```

Contoh :

```

if (PINA>0x80)
{
Dataku=PINA;
PORTC=0xFF;}

```

Pernyataan if diatas akan mengecek apakah data yang terbaca pada *PortA* (*PINA*) nilainya lebih dari 0x80 atau tidak, jika ya maka variable data di isi dengan nilai *PINA* dan data 0xFF dikeluarkan ke *PORTC*. Apabila dalam *blok* pernyataan hanya terdapat satu pernyataan saja maka tanda { dan } dapat dihilangkan seperti :

```

if (PINA>0x80) PORTC=0xFF;

```

(Sumber:<http://repository.usu.ac.id/bitstream/123456789/26400/4/Chapter%20II.pdf>).

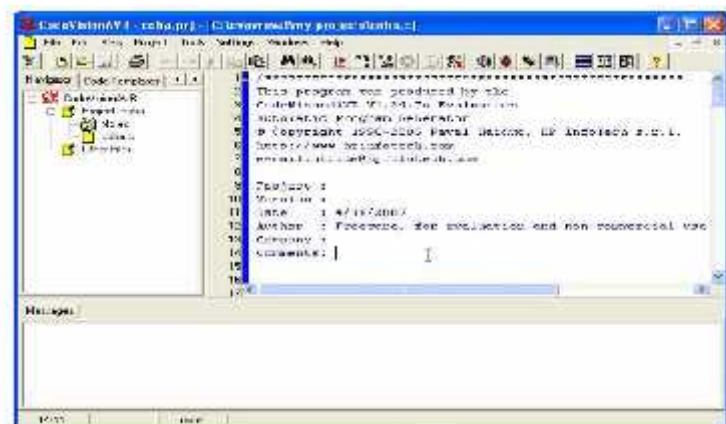
II.4.2 Codevision AVR

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 system operasi *Windows 95, 98, Me, NT4, 2000, dan XP*. *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 sistem *embedded*. *File object COFF* hasil kompilasi dapat digunakan untuk keperluan *debugging* pada tingkatan *C*, dengan pengamatan variabel, menggunakan debugger *Atmel AVR Studio*. Dari beberapa *software kompilator C*

yang pernah digunakan, *CodeVisionAVR* merupakan yang terbaik jika dibandingkan dengan kompiler-kompiler yang lain karena memiliki beberapa kelebihan yang dimiliki oleh *CodeVisionAVR* antara lain :

1. Menggunakan *IDE (Integrated Development Environment)*
2. Fasilitas yang disediakan lengkap (mengedit program, *mengkompile* program, *mendownload* program) serta tampilannya terlihat menarik dan mudah dimengerti
3. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas *CodeVisionAVR*.
4. Memiliki fasilitas untuk mendownload program langsung dari *CodeVision AVR* dengan menggunakan *hardware* khusus seperti *Atmel STK500*, *Kanda System STK200+/300* dan beberapa *hardware* lain yang telah didefinisikan oleh *CodeVisionAVR*.
5. Memiliki fasilitas debugger sehingga dapat menggunakan *software compiler* lain untuk mengecek kode assemblernya, contoh *AVRStudio*.
6. Memiliki terminal komunikasi serial yang terintegrasi dalam *CodeVisionAVR* sehingga dapat digunakan untuk membantu pengecekan program yang telah dibuat khususnya yang menggunakan fasilitas komunikasi serial *USART*.

(Sumber: Pemograman Bahasa C untuk *Mikrokontroller* : 8).

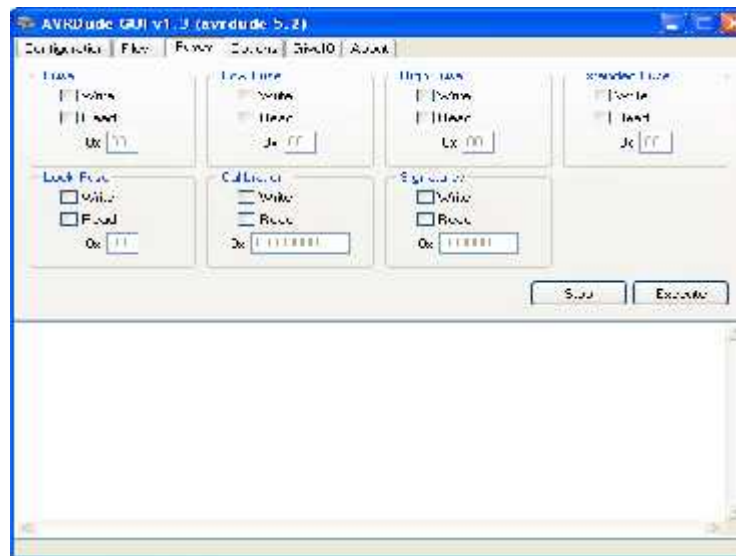


Gambar II.11 Tampilan Codevision AVR
(Sumber: Pemograman Bahasa C untuk Mikrokontroller : 9)

II.4.3. AVRdude

AVRdude adalah program untuk *upload/download* kode *hexa* ke *Mikrokontroler Atmel AVR ISP*. *Avrdude* jalan di sistem operasi *Linux* dan *windows*. Beragam *device programmer* dapat digunakan melalui *avrdude*, salah satunya *usbasp*. Sedangkan, *CodeVisionAVR* merupakan *IDE* untuk menulis kode, *mendeploy* dan mengelola *software*, menggunakan bahasa C dan sebagai perantaranya untuk *mendownload* kode yang sudah dibuat sebelumnya menggunakan *USBasp Downloader*.

Adapun tampilan *AVRdude* dapat dilihat pada gambar II.12 pada gambar di bawah ini.



Gambar II.12 Tampilan AVRdude
(Sumber: Pemograman Bahasa C untuk Mikrokontroler : 13)

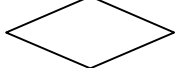
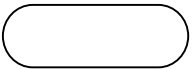
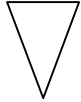
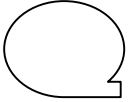
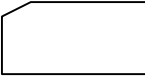
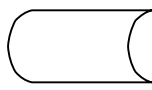
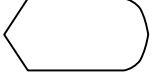
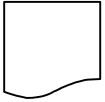
II.5. Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah prosedur dari suatu program dan alat.

Adapun tabel simbol *flowchart* pada Tabel II.10 dapat dilihat pada gambar dibawah ini.

Tabel II.10 Tabel Simbol Flowchart

No	Simbol	Arti	Keterangan
1		<i>Symbol Off-line Connector</i>	Simbol untuk keluar/masuk <i>procedure</i> atau proses dalam lembar/halaman yang lain.
2		<i>Symbol Connector</i>	Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama.
3		<i>Symbol Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer
4		<i>Symbol Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak

			dilakukan oleh komputer.
5		<i>Symbol Decision</i>	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
6		<i>Symbol Predefined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i>
7		<i>Symbol Terminal</i>	Simbol untuk permulaan atau akhir dari suatu program
8		<i>Symbol Off-line Storage</i>	Simbol yang menunjukkan bahwa data di dalam symbol ini akan disimpan
9		<i>Data Input Reader Operation</i>	Simbol operasi dengan membaca data input dari sistem atau bagian lain
10		<i>Symbol magnetic-tape unit</i>	Simbol yang menyatakan input berasal pita magnetic atau output disimpan ke pita <i>magnetic</i>
11		<i>Symbol punched card</i>	Simbol yang menyatakan input berasal dari kartu atau output ditulis ke kartu
12		<i>Symbol disk and on-line storage</i>	Simbol untuk menyatakan input berasal dari disk atau output disimpan ke <i>disk</i>
13		<i>Symbol display</i>	Simbol yang menyatakan peralatan output yang digunakan yaitu layar, <i>plotter</i> , printer, speaker dan sebagainya
14		<i>Symbol transmittal tape</i>	Simbol untuk menyatakan input berasal dari mesin jumlah/hitung
15		<i>Symbol document</i>	Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas

(Sumber: Pemograman Basis Data : 9)