

BAB II TINJAUAN PUSTAKA

2.1. Pengertian Sistem Pengendali

Pengertian sistem menurut Jogianto (2005) adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, dan orang-orang yang betul-betul ada dan terjadi.

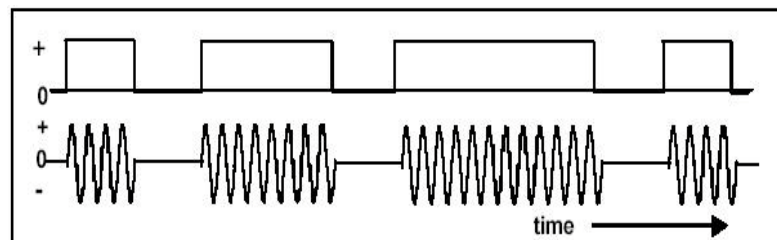
Sistem kendali merupakan hal yang di butuhkan oleh setiap manusia. Oleh karena itu, suatu alat (kumpulan alat) untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem dalam hal ini adalah alat pemanggang dengan kendali *remote control* dan mikrokontroler sebagai pusat pemrosesannya.

2.2. *Remote Control*

Remote Control adalah sebuah perangkat pengendali jarak jauh di mana perintah – perintahnya dikirimkan melalui media infra merah atau radio frekuensi (Budiharto, 2004), media infra merah atau radio frekuensi disebut juga sebagai pengirim (*transmitter*).

2.2.1 Metode Pengiriman Data

Dalam rangkaian ini penulis menggunakan radio frekuensi sebagai media pengiriman datanya. Cara kerja gelombang radio frekuensi adalah membawa sinyal-sinyal berupa pulsa yang nantinya akan dipisahkan kembali oleh rangkaian penerima agar dapat dijadikan nilai masukan pada mikrokontroler, dan menggunakan metode pengiriman data ASK (*Amplitude Shift Keying*).



Gambar 2.1 *Amplitudo Shift Keying* (ASK)

2.2.2 Pemancar

Pemancar (*transmitter*) adalah sebuah alat yang dapat memancarkan sinyal atau gelombang elektromagnetik dengan frekuensi tertentu. Dalam suatu pemancar terdapat dua buah sinyal/gelombang yang berbeda.

Gelombang pertama adalah gelombang pembawa (*carrier*), yang kedua adalah gelombang pemodulasi yang mempunyai frekuensi lebih rendah dari pada gelombang pembawa.

Modul pemancar dan penerima menggunakan modulasi digital. Salah satu contoh sistem komunikasi digital yang biasanya digunakan untuk memodulasi sinyal informasi ASK (*Amplitude Shift Keying*). ASK (*Amplitude Shift Keying*) adalah suatu modulasi di mana logika 1 diwakili dengan adanya sinyal dan logika 0 diwakili dengan adanya kondisi tanpa sinyal (Syamsudin, 2008).

Hasil ASK (*Amplitude Shift Keying*) diwakili oleh perbedaan amplitudo pada carrier. Dimana satu amplitudo adalah *zero*, ini menunjukkan kehadiran dan ketidakhadiran pada carrier yang digunakan. Sifat dari ASK antara lain:

- Rentan untuk pergantian *gain* tiba-tiba
- Tidak efisien
- Sampai dengan 1200bps pada voice grade line
- Digunakan pada *fiber optic*.

Modulasi ini merupakan suatu bentuk modulasi amplitudo dimana *carrier*-nya dimodulasi oleh sederetan pulsa digital. Modulasi ini terjadi diantara dua level amplitudo yang dilakukan dengan *switching carrier* on dan off, sehingga metode ini dikenal pula dengan nama OOK (*On Off Keying*).

Bentuk gelombang pemodulasi pada sistem komunikasi digital adalah sederetan *pulse* atau *square*. ASK adalah suatu bentuk modulasi amplitudo dimana *carrier* dimodulasi oleh sederetan pulsa. Modulasi ini terjadi antara dua level amplitudo atau lebih dan biasanya dengan melakukan *switching carrier* ON dan OFF, hal ini dikenal dengan ON-OFF ASK atau *ON-OFF Keying* (OOK). Output dari modulator ketika data dalam keadaan 0 adalah OFF atau tidak ada *output*.

2.2.3 Penerima

Penerima (*receiver*) adalah sebuah rangkaian yang dapat menerima gelombang yang mempunyai frekuensi yang sama dengan frekuensi yang dimilikinya. Penerima ini digunakan untuk menerima gelombang yang dipancarkan oleh pemancar. Pada gelombang RF yang telah diterima oleh penerima terdapat sinyal asli atau sinyal pemodulasi dari pembawa termodulasi dan nantinya akan digunakan sebagai nilai masukan pada mikrokontroler.

Demodulator ASK digunakan untuk mendemodulasi sinyal ASK yang berasal dari modulator agar diperoleh kembali sinyal informasi yang diinginkan. Pada penerima sinyal–sinyal ASK akan didemodulasi, dan dilakukan oleh rangkaian detektor, dimana frekuensi *carrier* akan dihilangkan dan menghasilkan *output* sesuai dengan data biner yang dikirim (Syamsudin, 2008).

Sinyal tersebut harus memiliki frekuensi dan fasa yang sama dengan *carrier* semula, yang mana akan digunakan sebagai referensi oleh *detector* sehingga dapat dideferensiasikan dua keadaan yang dikirim oleh pengirim dimana perkalian sinyal *carrier* membuat pergeseran spektrum sinyal pemodulasi $m(t)$ ke frekuensi *carrier* f_c .

Persamaan sinyal ASK (*Amplitude Shift Keying*) tampak pada Persamaan (1).

$$v(t) = Am(t)\cos(2f_c t) \dots\dots\dots (1)$$

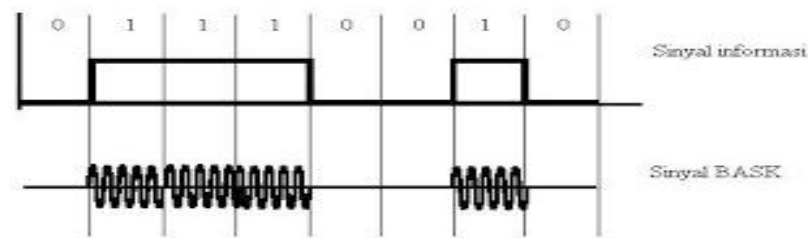
Keterangan:

- A = konstanta
- $m(t)$ = bernilai 1 atau 0
- f_c = frekuensi *carrier*
- t = durasi bit.

A adalah konstanta, $m(t)$ adalah sinyal data (sinyal pemodulasi) yang mempunyai nilai 0 atau 1, f_c adalah frekuensi putar dari sinyal pembawa, dan t adalah lebar dari satu bit.

Sebuah sinyal digital yang hanya mengandung 0 dan 1, dimodulasikan dengan *binary* ASK, maka kita hanya akan mengalikan sinyal pembawa dengan nilai 0 atau 1. Gambar 2.2 memperlihatkan modulasi *binary* ASK untuk sebuah sinyal digital yang diberikan 0 1 0 1 0 0 1 0. Seperti terlihat di gambar 2.2, sinyal–sinyal *binary* ASK bisa didapat dengan cara menyalakan dan mematikan (*on* dan

off) sinyal pembawa, tergantung apakah sinyal informasi (pemodulasi) bernilai 1 atau 0. *binary ASK* disebut juga *on-off keying* (OOK).



Gambar 2.2 *Binary ASK*

2.3. Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (*Personal Computer*) yang memiliki beragam fungsi. Perbedaan lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan mikrokontroler (Widodo, 2009).

2.3.1. Mikrokontroler ATmega8535

Beberapa tahun terakhir, mikrokontroler sangat banyak digunakan terutama dalam pengontrolan robot. Seiring perkembangan elektronika, mikrokontroler dibuat semakin kompak dengan bahasa pemrograman yang juga ikut berubah. Salah satunya adalah mikrokontroler AVR (*Alf and Vegard's Risc processor*) ATmega8535 yang menggunakan teknologi RISC (*Reduce Instruction Set Computing*) dimana program berjalan lebih cepat karena hanya membutuhkan satu *siklus clock* untuk mengeksekusi satu instruksi program. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu kelas ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama.

Teknologi yang digunakan pada mikrokontroler AVR berbeda dengan mikrokontroler seri MCS-51. AVR berteknologi RISC (*Reduced Instruction Set Computer*), sedangkan seri MCS-51 berteknologi CISC (*Complex Instruction Set Computer*). Mikrokontroler AVR dapat dikelompokkan menjadi empat kelas,

yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan keluarga AT89RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, kelengkapan peripheral dan fungsi-fungsi tambahan yang dimiliki (Wardhana, 2006).



Gambar 2.3 Mikrokontroler ATmega8535

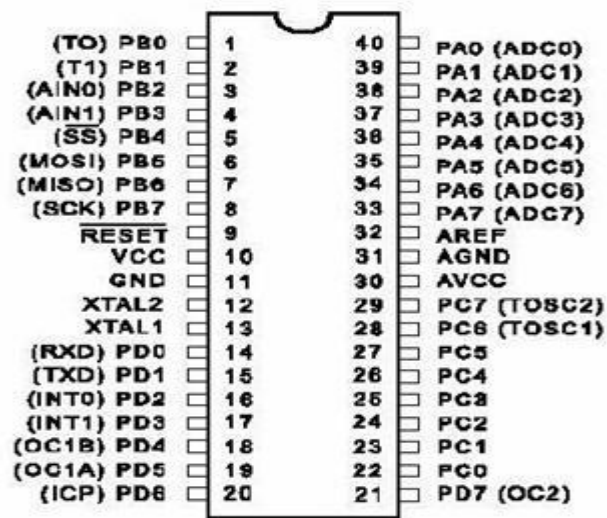
Pada pembuatan alat ini penulis menggunakan mikrokontroler ATmega8535 yang dapat dilihat pada gambar 2.2. Mikrokontroler ATmega8535 adalah mikrokontroler CMOS 8 bit daya rendah berbasis arsitektur RISC. Instruksi dikerjakan pada satu siklus *clock*, ATmega8535 mempunyai *throughput* mendekati 1 MIPS per MHz, hal ini membuat ATmega8535 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah.

2.3.2. Arsitektur Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 memiliki beberapa fitur atau spesifikasi yang menjadikannya sebuah solusi pengendali yang efektif untuk berbagai keperluan (Wardhana, 2006). Fitur-fitur tersebut antara lain:

1. Saluran I/O sebanyak 32 buah, yang terdiri atas *Port A*, *B*, *C* dan *D*
2. ADC (*Analog to Digital Converter*)
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan
4. CPU yang terdiri atas 32 *register*
5. *Watchdog Timer* dengan *osilator internal*
6. SRAM sebesar 512 *byte*
7. Memori *Flash* sebesar 8kb dengan kemampuan *read while write*
8. Unit Interupsi *Internal* dan *External*
9. *Port* antarmuka SPI untuk men-*download* program ke *flash*
10. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi
11. Antarmuka komparator *analog*
12. *Port* USART untuk komunikasi serial.

Mikrokontroler AVR ATmega 8535 memiliki 40 pin dengan 32 pin diantaranya digunakan sebagai port paralel. Satu port paralel terdiri dari 8 pin, sehingga jumlah port pada mikrokontroler adalah 4 port, yaitu port A, port B, port C dan port D. Sebagai contoh adalah port A memiliki pin antara port A.0 sampai dengan port A.7, demikian selanjutnya untuk port B, port C, port D (Wardhana, 2006). Diagram pin mikrokontroler dapat dilihat pada gambar 2.4.



Gambar 2.4 Konfigurasi Pin ATmega8535

Deskripsi port pada mikrokontroler ATmega8535 :

1. VCC (*Power Supply*) dan GND(*Ground*)
2. Port A (PA7...PA0)

Port A berfungsi sebagai *input* analog pada konverter A/D. Port A juga sebagai suatu Port I/O 8-bit dua arah, jika A/D konverter tidak digunakan. Pin - pin Port dapat menyediakan *resistor internal pull-up* (yang dipilih untuk masing-masing bit). Port A *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai *input* dan secara eksternal ditarik rendah, pin-pin akan memungkinkan arus sumber jika *resistor internal pull-up* diaktifkan. Pin Port A adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

3. Port B (PB7...PB0)

Port B adalah suatu port I/O 8-bit dua arah dengan *resistor internal pull-up* (yang dipilih untuk beberapa bit). Port B *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai *input*, pena Port B yang secara eksternal ditarik rendah akan arus sumber jika *resistor pull-up* diaktifkan. Pena Port B adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

4. Port C (PC7..PC0)

Port C adalah suatu Port I/O 8-bit dua arah dengan *resistor internal pull-up* (yang dipilih untuk beberapa bit). Port C *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai *input*, pena Port C yang secara eksternal ditarik rendah akan arus sumber jika *resistor pull-up* diaktifkan. Pena Port C adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

5. Port D (PD7..PD0)

Port D adalah suatu Port I/O 8-bit dua arah dengan *resistor internal pull-up* (yang dipilih untuk beberapa bit). Port D *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai *input*, pena Port D yang secara eksternal ditarik rendah akan arus sumber jika *resistor pull-up* diaktifkan. Pena Port D adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

6. *RESET* (*Reset input*)

7. XTAL1 (*Input Oscillator*)

8. XTAL2 (*Output Oscillator*)

9. AVCC adalah pena penyedia tegangan untuk Port A dan Konverter A/D.

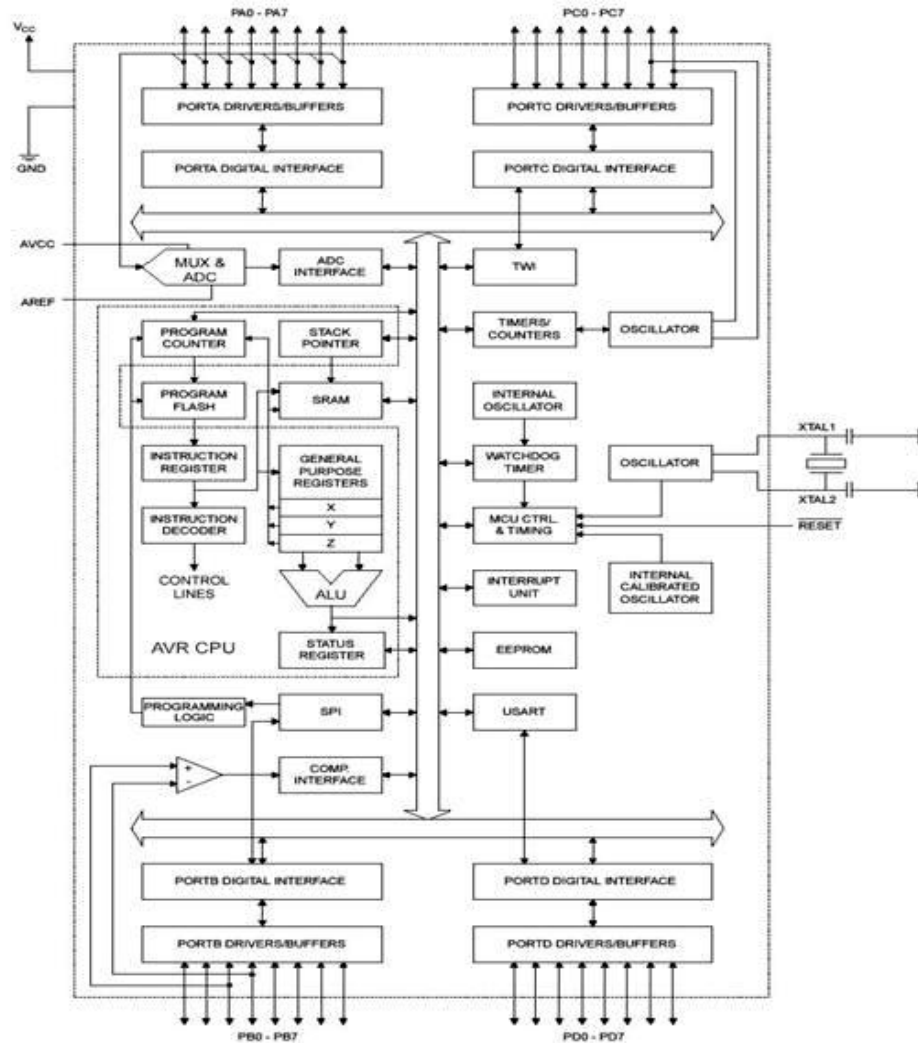
10. AREF adalah pena referensi analog untuk konverter A/D.

Berikut ini adalah tabel 2.1 penjelasan mengenai pin yang terdapat pada mikrokontroler ATmega8535 :

Tabel 2.1 Penjelasan pin pada mikrokontroler ATmega8535

Vcc	Tegangan suplai (5 volt)
GND	Ground
RESET	<i>Input reset level rendah, pada pin ini selama lebih dari panjang pulsamimum akan menghasilkan reset walaupun clock sedang berjalan. RST pada pin 9 merupakan reset dari AVR. Jika pada pin ini diberi masukan low selama minimal 2 machine cycle maka sistem akan di-reset</i>
XTAL 1	<i>Input penguat osilator inverting dan input pada rangkaian operasi clock internal</i>
XTAL 2	<i>Output dari penguat osilator inverting</i>
Avcc	<i>Pin tegangan suplai untuk port A dan ADC. Pin ini harus dihubungkan ke Vcc walaupun ADC tidak digunakan, maka pin ini harus dihubungkan ke Vcc melalui low pass filter</i>
Aref	<i>pin referensi tegangan analog untuk ADC</i>
AGND	<i>pin untuk analog ground. Hubungkan kaki ini ke GND, kecuali jika board memiliki analog ground yang terpisah</i>

Diagram blok ATmega8535 pada Gambar 2.5 digambarkan 32 *general purpose Working register* yang dihubungkan secara langsung dengan *Arithmetic Logic Unit (ALU)*. Sehingga memungkinkan dua *register* yang berbeda dapat diakses dalam satu siklus *clock* (Wardhana, 2006:2).



Gambar 2.5 Diagram blok mikrokontroler ATmega8535

2.3.3. Kapabilitas Mikrokontroler ATmega8535

Pada mikrokontroler ATmega8535 terdapat beberapa kelebihan adalah sebagai berikut (Wardhana, 2006) :

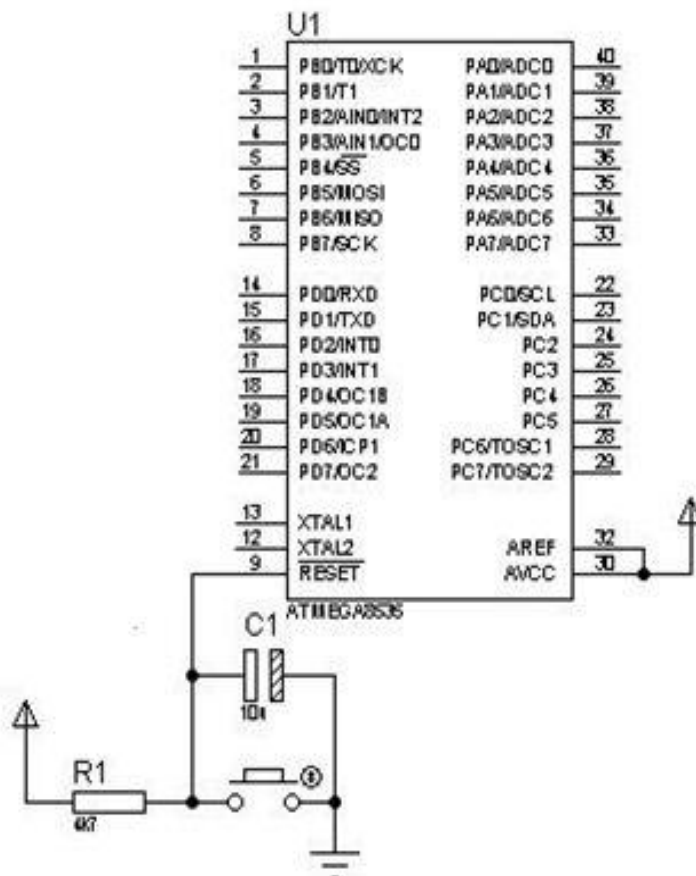
1. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
2. Kapabilitas memori *flash* 8Kb, *SRAM* sebesar 512 byte, dan *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte.
3. *ADC* internal dengan fidelitas 10 bit sebanyak 8 *channel*.
4. Portal komunikasi serial (*USART*) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan *mode sleep* menghemat penggunaan daya listrik.

2.3.4. Rangkaian Sistem Minimum ATmega8535

Sistem minimum mikrokontroler adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC mikrokontroler. Sistem minimum ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu. Di keluarga mikrokontroler AVR, seri 8535 adalah salah satu seri yang sangat banyak digunakan.

Untuk membuat rangkaian sistem minimum Atmel AVR 8535 diperlukan beberapa komponen yaitu :

1. IC mikrokontroler ATmega8535
2. 1 kapasitor eletrolit 100 uF (C3) 2 resistor yaitu 10 Kohm (R1)
3. 1 tombol reset pushbutton (PB1)



Gambar 2.6 Rangkaian Sistem Minimum ATmega8535

2.4. Motor DC

Motor DC (*direct current*) arus searah adalah peralatan elektronik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik yang disain awalnya dipekenalkan oleh Michael Faraday lebih dari seabad yang lalu (Pitowarno, 2006).



Gambar 2.7 Sebuah motor DC

Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaaan *torque* yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas. Gambar 2.7 memperlihatkan sebuah motor DC, dimana motor DC memiliki tiga komponen utama :

- ***Kutub medan.***

Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.

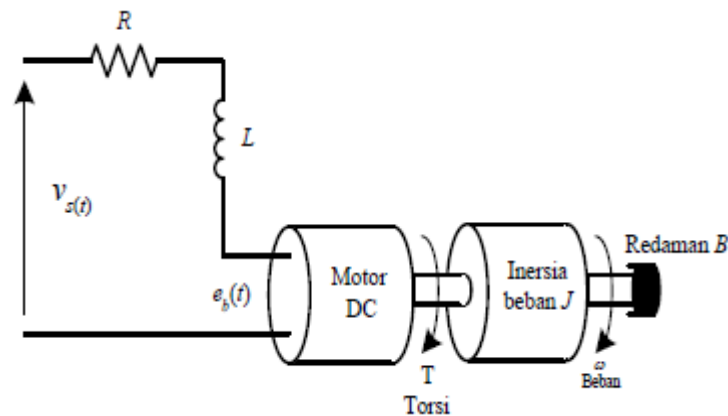
- ***Rotor.***

Bila arus masuk menuju rotor, maka arus ini akan menjadi elektromagnet. Rotor yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, rotor berputar dalam medan magnet yang dibentuk oleh kutub-

kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan rotor.

▪ **Commutator.**

Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.



Gambar 2.8 Model motor DC

Dari rangkaian pada gambar 2.8 dapat dibuat persamaan tegangan menurut hukum Kirchhoff tegangan seperti dinyatakan oleh persamaan (1) (Pitowarno, 2006).

$$v_s(t) = Ri_a(t) + L \frac{di_a(t)}{dt} + e_b(t) \quad \dots\dots\dots (1)$$

dengan

- R : tahanan belitan armatur (Ω),
- L : impedans belitan armatur (H),
- $i_a(t)$: arus armatur (A),
- $e_b(t)$: tegangan induksi di armatur (V),
- $v_s(t)$: tegangan terminal motor (V).

$e_b(t)$ adalah tegangan induksi yang tergantung pada putaran sudut, dinyatakan oleh persamaan (2).

$$e_b(t) = k_1 \phi \frac{60}{2\pi} \omega(t) \quad \dots\dots\dots (2)$$

dengan,

k_l : konstanta dimensi motor,

Φ : fluks magnet kutub motor (Wb),

$\frac{60}{2\pi} \omega(t)$: putaran rotor (rpm),

$\omega(s)$: kecepatan sudut rotor (rad/s).

Bila kutub motor adalah magnet permanen maka Φ konstan sehingga persamaan (2) dapat disusun kembali menjadi persamaan (3), dengan GGL lawan (*Back EMF*) sebanding dengan putaran.

$$e_b(t) = K\omega(t) \dots\dots\dots (3)$$

Untuk konstanta armatur $K = k_l \Phi \frac{60}{2\pi}$. Dengan mensubstitusikan persamaan (3) ke (1) diperoleh persamaan (4) sebagai berikut,

$$v_s(t) = R i_a(t) + L \frac{di_a(t)}{dt} + K\omega(t) \dots\dots\dots (4)$$

Untuk struktur mekanik yang mengacu pada hukum Newton, diperoleh persamaan torsi seperti persamaan (5),

$$\tau(t) = J \frac{d\omega(t)}{dt} + B\omega(t) = K i_a(t) \dots\dots\dots (5)$$

dengan:

J : momen inersia motor (kg.m²/s²),

B : konstanta redaman sistem mekanis (Nms),

Persamaan (4) dan (5) ditransformasikan dalam bentuk Laplace dengan asumsi semua kondisi awal sama dengan nol [6], diperoleh,

$$V_s(s) = (R + sL) I_a(s) + K\omega(s) \dots\dots\dots (6)$$

sehingga diperoleh $I_a(s)$ sebagai berikut,

$$I_a(s) = \frac{V_s(s) - K\omega(s)}{R + sL} \dots\dots\dots (7)$$

dan,

$$T(s) = (B + sJ)\omega(s) = K I_a(s) \dots\dots\dots (8)$$

Dari persamaan (8) diperoleh kecepatan sudut $\omega(s)$ sebagai berikut,

$$\omega(s) = \frac{KI_a(s)}{(B + sJ)} \dots\dots\dots (9)$$

Selanjutnya dengan mensubstitusi persamaan (9) ke (7) diperoleh fungsi alih $V_s(s)$ antara masukan tegangan dan keluaran arus armature $I_a(s)$ sebagai berikut,

$$\frac{I_a(s)}{V_s(s)} = \frac{Js + B}{(Js + B)(Ls + R) + K^2}$$

atau,

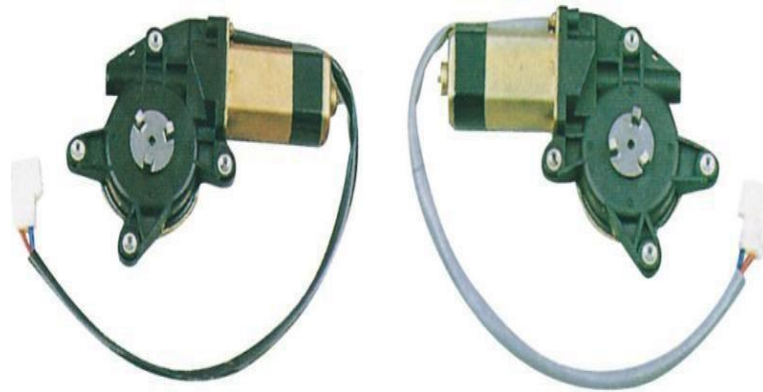
$$\frac{I_a(s)}{V_s(s)} = \frac{Js + B}{JLs^2 + (BL + JR)s + BR + K^2} \dots\dots\dots (10)$$

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

- Tegangan rotor – meningkatkan tegangan dinamo akan meningkatkan kecepatan
- Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan *rolling mills*, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya.

Salah satu jenis dari motor DC adalah motor DC *power window* yang menggunakan energi listrik dan energi magnet untuk menghasilkan energi mekanis dapat dilihat pada gambar 2.9. Operasi motor tergantung pada interaksi dua medan magnet. Secara sederhana dikatakan bahwa motor listrik bekerja dengan prinsip bahwa dua medan magnet dapat dibuat berinteraksi untuk menghasilkan gerakan. Tujuan motor adalah untuk menghasilkan gaya yang menggerakkan (torsion).



Gambar 2.9 Motor DC *power window*

Salah satu keistimewaan motor DC ini adalah kecepatannya dapat dikontrol dengan mudah. Sifat dari motor DC bila tenaga mekanik yang diperlukan cukup kecil maka motor DC yang digunakan cukup kecil pula. Motor DC untuk tenaga kecil pada umumnya menggunakan magnet permanen sedangkan motor listrik arus searah yang dapat menghasilkan tenaga mekanik besar menggunakan magnet listrik.

Motor ini bergerak kedepan dan kebelakang sesuai dengan pengoperasian *switch*. Arah putaran motor DC magnet permanen ditentukan oleh arah arus yang mengalir pada kumparan jangkar. Pembalikan ujung-ujung jangkar tidak membalik arah putaran. Kecepatan motor magnet permanen berbanding langsung dengan harga tegangan yang diberikan pada kumparan jangkar. Semakin besar tegangan jangkar, semakin tinggi kecepatan motor.

Motor DC *power window* memiliki beberapa bagian yaitu:

1. *Stator* motor DC

Stator merupakan bagian dari motor yang permanen atau tidak berputar. Bagian ini menghasilkan medan magnet, baik yang dihasilkan dari koil (elektromagnetik), maupun dari magnet.

2. *Rotor* atau Jangkar motor DC

Fungsi dari *rotor* atau jangkar yaitu untuk merubah energi listrik menjadi energi mekanik dalam bentuk gerak putar. Rotor terdiri dari poros baja dimana tumpukan keping-keping inti yang berbentuk silinder dijepit. Pada inti terdapat alur-alur dimana lilitan rotor diletakkan.

3. Komutator

Konstruksi dari komutator terdiri dari lamel-lamel, antar lamel dengan lamel lainnya diisolasi dengan mica.

4. Sikat (*Brush*)

Fungsi dari sikat-sikat adalah untuk jembatan bagi aliran arus dari lilitan jangkar beban, aliran arus tersebut akan mengalir dari sumber dan diterima oleh kontaktor.

5. Saklar atau *Switch*

Merupakan komponen utama dalam power window karena sebagai penyambung dan pemutus arus untuk menaikkan atau menurunkan jendela. Jendela dapat dinaikan atau diturunkan dengan cara membalik kutub tegangan pada saklar.

Dengan kata lain dua kabel yang keluar dari motor power window (umumnya berwarna hijau dan biru), digunakan dengan cara membolak-balikkan kutub tegangan masuknya. Untuk keperluan tersebut, perlu juga disediakan terminal – *female* sebanyak 5 unit dan *crimpler* untuk pemasangan terminal tersebut.

Spesifikasi Motor *Power Window* dan *Relay Power Window*

- Motor *Power Window*

<i>Rate voltage</i>	: DC 12 volt
<i>Operating Voltage Range</i>	: DC 10 – 16 volt
<i>Operating Temperature Range</i>	: - 300 C – (+) 800 C - 220 F – (+) 1760 F
<i>Speed</i>	: 40 ± 5 rpm
<i>Load</i>	: 4 N.m
<i>Power Window Relay</i>	: 200mA (coil load) 12 volt 10 A

2.5. *Transformator*

Transformator merupakan suatu peralatan listrik elektromagnetik statis yang berfungsi untuk memindahkan dan mengubah daya listrik dari suatu rangkaian listrik ke rangkaian listrik lainnya, dengan frekuensi yang sama dan perbandingan transformasi tertentu melalui suatu gandingan magnet dan bekerja berdasarkan

prinsip induksi elektromagnetis, dimana perbandingan tegangan antara sisi primer dan sisi sekunder berbanding lurus dengan perbandingan jumlah lilitan dan berbanding terbalik dengan perbandingan arusnya (Abdulrajak, 2001).

Dalam bidang teknik listrik pemakaian *transformator* dikelompokkan menjadi:

1. *Transformator* daya
2. *Transformator* distribusi
3. *Transformator* pengukuran; yang terdiri dari *transformator* arus dan *transformator* tegangan.

Transformator terdiri atas dua buah kumparan (primer dan sekunder) yang bersifat induktif. Kedua kumparan ini terpisah secara elektrik namun berhubungan secara magnetis melalui jalur yang memiliki reluktansi (*reluctance*) rendah. Apabila kumparan primer dihubungkan dengan sumber tegangan bolak-balik maka fluks bolak-balik akan muncul di dalam inti yang dilaminasi, karena kumparan tersebut membentuk jaringan tertutup maka mengalirlah arus primer. Akibat adanya fluks di kumparan primer maka di kumparan primer terjadi induksi (*self induction*) dan terjadi pula induksi di kumparan sekunder karena pengaruh induksi dari kumparan primer atau disebut sebagai induksi bersama (*mutualinduction*) yang menyebabkan timbulnya fluks magnet di kumparan sekunder, maka mengalirlah arus sekunder jika rangkaian sekunder di bebani, sehingga energi listrik dapat ditransfer keseluruhan (secara magnetisasi).



Gambar 2.10 *Transformator*

2.5.1. Prinsip Kerja *Transformator*

Prinsip kerja transformator adalah dimana alat ini terdiri dari sebuah susunan teras besi tertutup didalamnya terdapat gulungan kawat tembaga yang digulungkan pada kaki-kaki transformator yaitu (Anwar, 2008) :

1. Gulungan primer (P) adalah gulungan yang dipasangkan pada sumber arus.
2. Gulungan sekunder (S) adalah gulungan yang dipasang pada aliran listrik.

Bekerjanya *transformator* ini berdasarkan pembangkitan tegangan bolak-balik secara induksi didalam gulungan- gulungan kawat yang melingkari garis gaya (fluks) yang berubah. Tegangan bolak-balik E_1 mengalir I_1 dan aliran I_1 akan membangkitkan tegangan induksi E_2 pada klem-klem gulungan sekunder.

Bila tegangan V_1 yang berbentuk sinusoidal dihubungkan ke gulungan primer suatu *transformator* maka pada gulungan tersebut akan mengalir arus listrik I_0 yang membangkitkan fluks bolak-balik yang mengalir ke seluruh inti sehingga menimbulkan gaya gerak listrik induksi E_1 pada belitan primer dan E_2 pada belitan sekunder, karena pada belitan sekunder diberi beban Z maka akan timbul arus I_2 (arus beban) yang melingkari kumparan sekunder. Arus I_2 yang melalui kumparan menimbulkan fluks (F_2) sehingga mengurangi fluks yang pertama (F_m), sehingga fluks menjadi kecil dan E_1 juga turun begitu juga dengan E_2 , jadi $F = (F_m - F_2)$. Dengan turunnya E_1 berarti timbul I_2 untuk menyeimbangi (pelemahan F_m) E_2 sehingga E_1 dan E_2 tetap.

Besarnya tegangan induksi dituliskan dalam gulungan primer (N_1) dalam gulungan yang dapat dirumuskan sebagai berikut :

$$E_1 = f_v \cdot F \cdot N \cdot 10^{-8} \text{ Volt} \dots\dots\dots (1)$$

Karena: $f_v = 1,11$ (konstan)

$$\text{Jadi : } E_1 = 4,44 \cdot F \cdot 10^{-8} \text{ Volt}$$

Sedangkan untuk gulungan sekunder N_2 , maka tegangan induksi pada gulungan sekunder adalah :

$$E_2 = 4,44 \cdot F \cdot N_2 \cdot 10^{-8} \text{ Volt} \dots\dots\dots (2)$$

Dimana :

- E1 = tegangan primer
- E2 = tegangan sekunder
- N1 = gulungan primer
- N2 = gulungan sekunder

Perbandingan antara arus primer dengan arus sekunder adalah berbanding terbalik dengan perbandingan antara tegangan induksi primer dan tegangan induksi sekunder ini dapat dirumuskan sebagai berikut:

$$\frac{N1}{N2} = \frac{I2}{I1} = \frac{V1}{V2} \dots\dots\dots (3)$$

Perbandingan rumus diatas apabila transformator dianggap ideal dan perbandingan tersebut dinamakan perbandingan transformasi. Perlu diingat bahwa hanya tegangan listrik arus bolak-balik yang dapat ditransformasikan oleh *transformator*, sedangkan dalam bidang elektronika, *transformator* digunakan sebagai gandengan impedansi antara sumber dan beban untuk menghambat arus searah sambil tetap melakukan arus bolak-balik antara rangkaian.

Tujuan utama menggunakan inti pada *transformator* adalah untuk mengurangi reluktansi (tahanan magnetis) dari rangkaian magnetis (*common magnetic circuit*).

2.6. Sensor

Terdapat berbagai macam sensor yang digunakan dalam teknik robotik. Keberagaman ini juga termasuk dalam hal cara pengukuran dan cara *interfacing* ke kontroler.

Dari segi tipe *output* dan aplikasinya sensor dapat diklasifikasikan seperti pada tabel 2.2 berikut ini:

Tabel 2.2 Klasifikasi sensor berdasarkan tipe *output*

Output Sensor	Contoh aplikasi/ sensor
Biner (1/0)	Sensor <i>tactile (limit switch, TX-RX infra-merah)</i>
<i>Analog</i> , misal (0:5)V	Sensor temperatur, <i>acceserator</i>
Pulsa, misal PWM	<i>Giroskop (gyroscope)</i> digital
Data serial, misal RS232 atau USB	<i>Modul Global Positioning System (GPS)</i>

Dari sudut pandang robot, sensor dapat diklasifikasikan dalam dua kategori, yaitu sensor lokal (on-board) yang dipasang di tubuh robot, dan sensor global, yaitu sensor yang diinstal di luar robot tapi masih dalam lingkungannya (*environment*) dan data sensor global ini dikirim balik ke robot melalui komunikasi nirkabel. Dalam skala besar contoh sensor global ini adalah kamera yang terpasang pada satelit GPS yang mampu menangkap citra di lingkungan robot jauh dari atas (Pitowarno, 2006).

2.6.1. DF Robot (*Flame Sensor*)

Sensor DF ROBOT pada gambar 2.17 merupakan sensor api yang dapat mendeteksi nyala api dengan panjang gelombang 760 ~ 1100 nm, sensor ini dapat mendeteksi suhu panas berkisar 25 C – 85 C. Sensitivitas dari produk ini sudah teruji dengan baik melalui beberapa percobaan sehingga membuat penulis memilih sensor dari DF ROBOT ini sebagai sensor yang akan diintegrasikan dengan robotino sebagai salah satu sarana untuk mendeteksi suhu dari api.

Sensor ini dapat mendeteksi api dari jarak 100 cm dengan keluaran tegangan sebesar 0,5v, dan pada jarak 20 cm dengan objek sensor ini dapat mengeluarkan keluaran tegangan sebesar 5v (Robert, 2013).



Gambar 2.11 Sensor DF robot *flame sensor*

2.7. *Relay*

Relay adalah komponen yang menggunakan prinsip kerja medan magnet untuk menggerakkan saklar atau mengaktifkan switch. Saklar ini digerakkan oleh

magnet yang dihasilkan oleh kumparan di dalam relay yang dialiri arus listrik (Widodo, 2013).

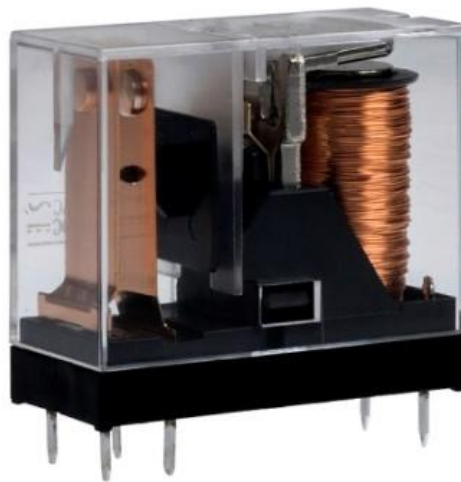
Sebuah *relay* tersusun atas kumparan, pegas, saklar (terhubung pada pegas) dan 2 kontak elektronik (*Normally Close* dan *Normally Open*).

1. *Normally Close (NC)*

Saklar terhubung dengan kontak ini saat *relay* tidak aktif atau dapat dikatakan saklar dalam kondisi terbuka.

2. *Normally Open (NO)*

Saklar terhubung dengan kontak ini saat *relay* aktif atau dapat dikatakan saklar dalam kondisi tertutup.



Gambar 2.12 *Relay*

Di dalam *relay* terdapat gulungan kawat tembaga (coil) dengan ujung-ujung kawat diberi nomor 85 dan 86, mekanisme saklar seperti gambar kawat terputus dengan ujung-ujungnya diberi nomor 30 dan 87. Gulungan atau coil digunakan untuk menciptakan medan magnet pada inti besi coil itu. seperti kita ketahui apa bila kita melilitkan kawat tembaga pada sebuah inti besi contoh paku, maka paku tersebut akan menjadi magnet apabila kawat tembaga itu kita aliri arus listrik. Hal ini juga digunakan pada relay, jika 85 diberi arus + dan 86 diberi arus atau sebaliknya maka akan tercipta medan magnet pada ujung inti besi coil itu (Widodo, 2013).

Jika medan magnet sudah terbentuk seperti pada gambar 2.18, maka mekanisme saklar yang terbuat dari besi akan tertarik oleh medan magnet, sehingga mekanisme saklar yang tadinya terbuka atau terputus menjadi tertutup

atau menyambung, sehingga 30 dan 87 menjadi satu kesatuan seolah-olah seperti seutas kawat ataupun seperti saklar yang sedang di aktifkan. Membuka dan menutupnya 30 dan 87 inilah yang digunakan untuk menghidupkan atau mematikan instrumen.

2.7.1. Prinsip Kerja Relay

Relay terdiri dari *coil* dan *contact*. Perhatikan gambar 2.12, *coil* adalah gulungan kawat yang mendapat arus listrik, sedang *contact* adalah sejenis saklar yang pergerakannya tergantung dari ada tidaknya arus listrik di *coil*. *Contact* ada 2 jenis, *Normally Open* (kondisi awal sebelum diaktifkan *open*), dan *Normally Closed* (kondisi awal sebelum diaktifkan *close*). Secara sederhana berikut ini prinsip kerja dari *relay* adalah ketika *coil* mendapat energi listrik (*energized*), akan timbul gaya elektromagnet yang akan menarik *armature* yang berpegas, dan *contact* akan menutup. Selain berfungsi sebagai komponen elektronik, *relay* juga mempunyai fungsi sebagai pengendali sistem. Sehingga *relay* mempunyai 2 macam simbol yang digunakan pada :

- Rangkaian listrik (*hardware*)
- Program (*software*)

2.7.2. Jenis-jenis Relay

Seperti saklar, *relay* juga dibedakan berdasar *pole* dan *throw* yang dimilikinya. Berikut definisi *pole* dan *throw* :

Pole : banyaknya *contact* yang dimiliki oleh *relay*

Throw : banyaknya kondisi (*state*) yang mungkin dimiliki *contact*

Berikut ini penggolongan relay berdasar jumlah pole dan throw :

1. SPST (*Single Pole Single Throw*)
2. DPST (*Double Pole Single Throw*)
3. SPDT (*Single Pole Double Throw*)
4. DPDT (*Double Pole Double Throw*)
5. 3PDT (*Three Pole Double Throw*)
6. 4PDT (*Four Pole Double Throw*)

2.8. *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah sinyal listrik menjadi sinyal suara. Proses mengubah sinyal ini dilakukan dengan cara menggetarkan komponennya yang berbentuk selaput. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loudspeaker*. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada alat (*alarm*). Fungsi dari *buzzer* adalah sama seperti pengeras suara, yaitu untuk menghasilkan suara dalam frekuensi tinggi dan rendah. (Prihono, 2009).



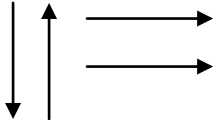
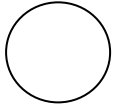
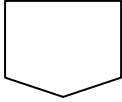


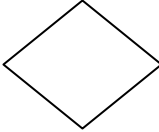


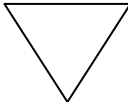
Gambar 2.13 *Buzzer*

2.9. *Flowchart*


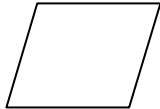

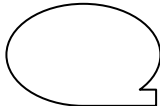
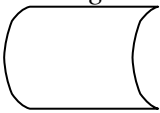


Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu sedangkan hubungan antara proses digambarkan dengan garis penghubung (Tosin, 1994).

Terdapat 2 macam *flowchart* yang menggambarkan proses dengan komputer yaitu *system flowchart* dan *program flowchart*. *System flowchart* adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file dalam media tertentu. Sedangkan *program flowchart* adalah bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file dalam media tertentu. Sedangkan *program flowchart* adalah bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Pada tabel 2.3 berikut menjelaskan tentang symbol dan fungsi pada *flowchart* :

Tabel 2.3a Tabel *Flowchart*

No	Simbol / Gambar	Keterangan
1.	<p style="text-align: center;"><i>Flow</i></p> 	Garis untuk menghubungkan arah tujuan symbol <i>flowchart</i> yang satu dengan yang lainnya.
2.	<p style="text-align: center;"><i>Connector</i></p> 	Simbol untuk keluar/masuk prosedur atau proses dalam/halaman yang sama.
3.	<p style="text-align: center;"><i>Off - Line Connector</i></p> 	Simbol untuk keluar/masuk prosedur atau proses dalam/halaman yang lain.
4.	<p style="text-align: center;"><i>Process</i></p> 	Menunjukkan pengolahan yang dilakukan oleh komputer.
5.	<p style="text-align: center;"><i>Manual Operation</i></p> 	Menunjukkan pengolahan yang tidak dilakukan oleh komputer.
6.	<p style="text-align: center;"><i>Decision</i></p> 	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
7.	<p style="text-align: center;"><i>Predefined Process</i></p> 	Simbol untuk mempersiapkan penyimpangan yang akan digunakan sebagai tempat pengolahan didalam <i>storage</i> .
8.	<p style="text-align: center;"><i>Terminal</i></p> 	Simbol untuk permulaan atau akhir dari suatu program.
9.	<p style="text-align: center;"><i>Off - Line Storage</i></p> 	Simbol yang menunjukkan bahwa data didalam simbol ini akan disimpan.

Tabel 2.3b Tabel *Flowchart*

No	Simbol / Gambar	Keterangan
10.	<p><i>Manual Input</i></p> 	Simbol untuk memasukkan data secara manual <i>on-line keyboard</i> .
11.	<p><i>Input/Output</i></p> 	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
12.	<p><i>Punched Card</i></p> 	Simbol yang menyatakan input berasal dari kartu atau <i>output</i> ditulis ke kartu.
13.	<p><i>Magnetic Tape Unit</i></p> 	Simbol yang menyatakan <i>input</i> berasal dari pita <i>magnetic</i> atau <i>output</i> disimpan ke pita <i>magnetic</i> .
14.	<p><i>Disk And On-Line Storage</i></p> 	Simbol yang menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .
15.	<p><i>Document</i></p> 	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.
16.	<p><i>Display</i></p> 	Simbol yang menyatakan peralatan <i>output</i> yang digunakan yaitu layar, <i>plotter</i> , <i>printer</i> dan sebagainya.

2.10. Pengenalan *Software* (Perangkat Lunak)

Perangkat lunak (*Software*) adalah program yang digunakan untuk menterjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman kedalam bahasa mesin supaya dapat dimengerti oleh computer (Jogiyanto, 1999). *Software* adalah penghubung antara manusia dengan perangkat keras komputer, berfungsi menerjemahkan bahasa manusia ke dalam bahasa mesin sehingga perangkat keras komputer memahami keinginan pengguna dan

menjalankan instruksi yang diberikan dan selanjutnya memberikan hasil yang diinginkan oleh manusia tersebut.

Untuk dapat menggunakan mikrokontroler, tentu diperlukan bahasa komunikasi yaitu bahasa pemrograman. Pemrograman yang tepat, mudah dan benar akan menjamin *embedded system* yang diciptakan beroperasi secara efisien dan efektif.

2.10.1. Penulisan Program Bahasa C

Bahasa pemrograman sendiri mengalami perkembangan, diawali dengan *Assembler* (bahasa tingkat rendah) sampai ADA (bahasa tingkat tinggi). Perkembangan bahasa tersebut secara detail adalah sebagai berikut : bahasa tingkat rendah meliputi *Assembler* dan *Macro-Assembler*, bahasa tingkat menengah meliputi FORTH, C, C++ dan Java, sedangkan bahasa tingkat tinggi meliputi BASIC, FORTRAN, COBOL, Pascal, Modula-2 dan ADA (Widodo, 2009).

Untuk dapat memahami bagaimana suatu program ditulis, maka struktur dari program harus dimengerti terlebih dahulu. Tiap bahasa komputer mempunyai struktur program yang berbeda. Jika struktur dari program tidak diketahui, maka akan sulit bagi pemula untuk memulai menulis suatu program dengan bahasa yang bersangkutan. Struktur dari program memberikan gambaran secara luas, bagaimana bentuk dari program secara umum. Selanjutnya dengan pedoman struktur program ini, penulis program dapat memulai bagaimana seharusnya program tersebut ditulis.

Dalam pembuatan program yang menggunakan fungsi atau aritmatika, Bahasa C menawarkan kemudahan dengan menyediakan fungsi – fungsi khusus, seperti: pembuatan konstanta, operator aritmatika, operator logika, operator *bitwise* dan operator *Assignment*. Selain itu, bahasa C menyediakan Program kontrol seperti: Percabangan (*if* dan *if...else*), Percabangan *switch*, *Looping* (*for*, *while* dan *do...while*), *Array*, serta fungsi – fungsi lainnya.

Struktur dari program C dapat dilihat sebagai kumpulan dari sebuah atau lebih fungsi-fungsi. Fungsi pertama yang harus ada di program C sudah ditentukan namanya, yaitu bernama *main()*. Suatu fungsi di program C dibuka dengan kurung kurawal ({) dan ditutup dengan kurung kurawal tutup (}). Diantara

kurung-kurung kurawal dapat dituliskan pernyataan- pernyataan program C. Struktur bahasa pemrograman C, antara lain:

- a. *Header File* adalah berkas yang berisi *prototype* fungsi definisi konstanta dan definisi variable. Fungsi adalah kumpulan kode C yang diberi nama dan ketika nama tersebut dipanggil maka kumpulan kode tersebut dijalankan.
- b. *Preprosesor Directive (#include)* adalah bagian yang berisi pengikutsertaan file atau berkas-berkas fungsi maupun pendefinisian *konstanta*.
- c. *Void* artinya fungsi yang mengikutinya tidak memiliki nilai kembalian (*return*).
- d. *Main ()* adalah fungsi yang pertama kali dijalankan ketika program dieksekusi, tanpa fungsi main suatu program tidak dapat dieksekusi namun dapat dikompilasi.
- e. *Statement* adalah instruksi atau perintah kepada suatu program ketika program itu dieksekusi untuk menjalankan suatu aksi. Setiap *statement* diakhiri dengan titik-koma.

Berikut ini gambaran penulisan struktur dari program C :

```

main ()
{
    Statemen-statemen
}
Fungsi_fungsi_lain()
{
    Statemen-statemen
}

```

The diagram illustrates the structure of a C program. It shows two function blocks. The first block is the `main ()` function, which contains `Statemen-statemen`. This block is labeled as the "fungsi utama" (main function). The second block is a general function `Fungsi_fungsi_lain()`, which also contains `Statemen-statemen`. This block is labeled as "fungsi-fungsi lain yang ditulis oleh pemrograman komputer" (other functions written by the computer programmer).

Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (*subroutine*). Fungsi-fungsi selain fungsi utama merupakan program-program bagian. Fungsi-fungsi ini dapat ditulis setelah fungsi utama atau diletakkan di file pustaka (*library*). Jika fungsi-fungsi diletakkan di file pustaka dan akan dipakai disuatu program, maka nama file judulnya (*header file*) harus dilibatkan di dalam

program yang menggunakannya dengan *preprocessor directive* #include (Hartono, 1992).

Tipe data merupakan bagian yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien. Tipe data pada bahasa C dapat dilihat pada Tabel 2.4.

Tabel 2.4 Tipe Data Bahasa C

Tipe Data	Ukuran (byte)	Format	Keterangan
<i>Char</i>	1	%c	Karakter / <i>String</i>
<i>Int</i>	2	%i %d	Bilangan Bulat (<i>integer</i>)
<i>Float</i>	4	%f	Bilangan Pecahan (<i>float</i>)
<i>Double</i>	8	%lf	Pecahan presisi ganda

2.10.2. Code Vision AVR

Pada laporan ini bahasa pemrograman yang digunakan adalah bahasa C sebagai sistem yang akan ditanamkan pada mikrokontroler. Program bahasa C tidak mengenal aturan penulisan di kolom tertentu, jadi bisa di mulai dari kolom manapun dan untuk mempermudah pembacaan program serta untuk keperluan dokumentasi. Sedangkan *compiler* yang digunakan adalah *CodeVisionAVR*.

CodeVisionAVR pada dasarnya merupakan perangkat lunak pemrograman mikrokontroler keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: *Compiler C*, IDE dan program *generator*. *CodeVisionAVR* dilengkapi dengan *source code editor*, *compiler*, *linker* dan dapat memanggil Atmel AVR *studio* dengan *debugger* nya (Andrianto, 2007).

Meskipun *CodeVision AVR* termasuk *software* komersial, namun kita tetap dapat menggunakannya dengan mudah karena terdapat versi evaluasi yang tersedia secara gratis walaupun dengan kemampuan yang dibatasi.

CodeVision AVR merupakan yang terbaik bila dibandingkan dengan kompilernya yang lain karena beberapa kelebihan yang dimiliki oleh *CodeVision AVR* antara lain :

1. Menggunakan IDE (*Integrated Development Environment*),
2. Fasilitas yang disediakan lengkap (mengedit program, mengkompilasi program, mendownload program) serta tampilannya sangat *user friendly*.
3. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas *CodeWizard AVR*.
4. Memiliki fasilitas untuk mendownload secara langsung menggunakan hardware khusus.
5. Memiliki fasilitas debugger sehingga dapat menggunakan software compiler lain untuk mengecek kode assemblynya, contoh AVRStudio.
6. Memiliki terminal komunikasi serial yang terintegrasi dalam CodeVision AVR sehingga dapat digunakan untuk membantu pengecekan program yang telah dibuat khususnya yang menggunakan fasilitas komunikasi serial UART.

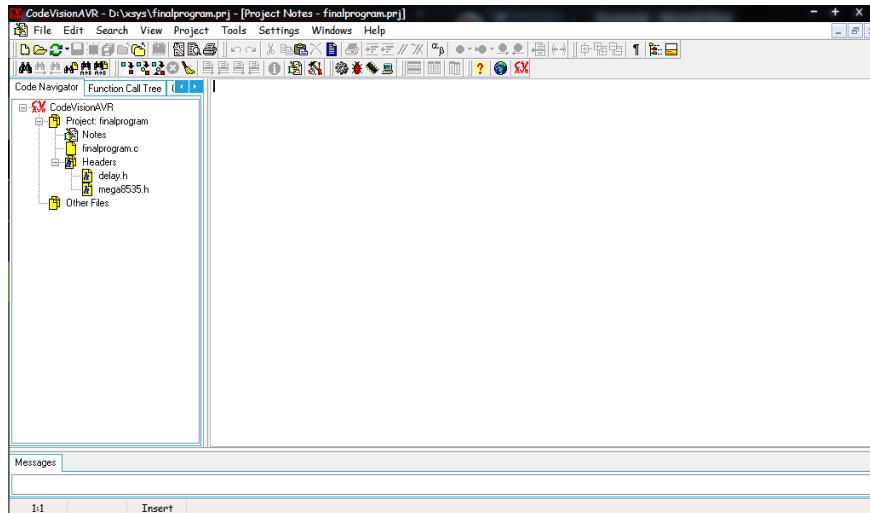
2.10.3. Menjalankan *CodeVisionAVR*

Ada beberapa program yang dapat digunakan sebagai *editor* dan *compiler* untuk mikrokontroler AVR, salah satunya yaitu *CodeVisionAVR*. Untuk menjalankan program tersebut terlebih dahulu menginstal program tersebut.

Setelah terinstal maka buka program *CodeVision* melalui menu *Start||All Program||CodeVision| CodeVisionAVR C Compiler* atau melalui desktop dengan mengklik lambang *codevision*.

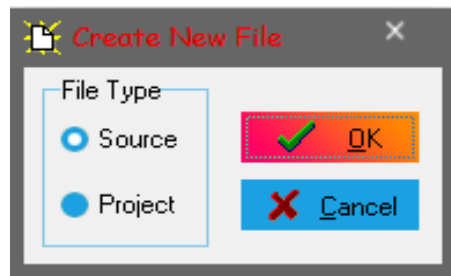


Gambar 2.14 Ikon *CodeVisionAVR*

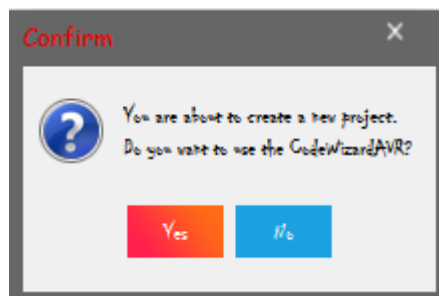


Gambar 2.15 Tampilan pertama kali *codevision* dijalankan

Lalu klik *File|New|* pilih *File Type* dan klik *Project* lalu klik OK,

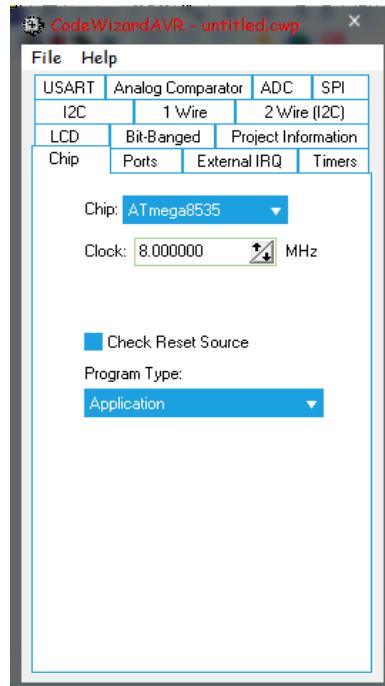


Gambar 2.16 Membuat *file project* baru



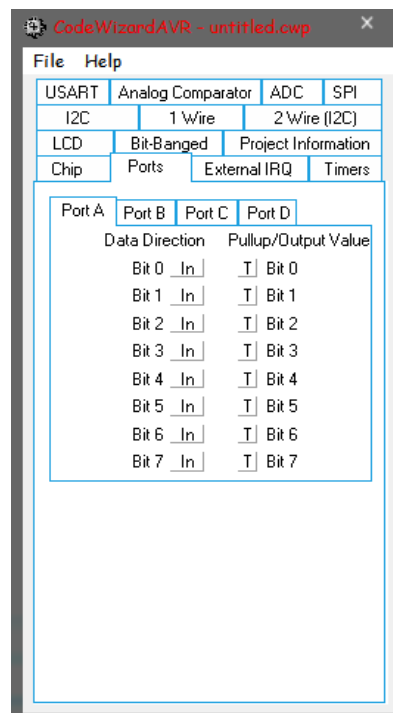
Gambar 2.17 *Project* baru menggunakan *CodeWizardAVR*

Pilih tampilan konfirmasi, dan menayakan apakah akan menggunakan *CodeWizard* untuk membuat *project* baru, pilih *yes*. Kemudian akan tampil konfigurasi USART, Analog *Comparator*, ADC, SPI, I2C, 1 *wire*, 2 *wire* (I2C), LCD, *Bit-Banged*, *Project Information*, *chip*, *Port*, *External IRQ*, *Timer*. Kemudian tinggal mengatur program yang akan dibuat melalui *CodeWizard* ini. Misalnya konfigurasi *chip* yang akan digunakan, pilih *chip*, lalu isi konfigurasi : *chip*: ATmega8535, *clock*: 8.000000 MHz.



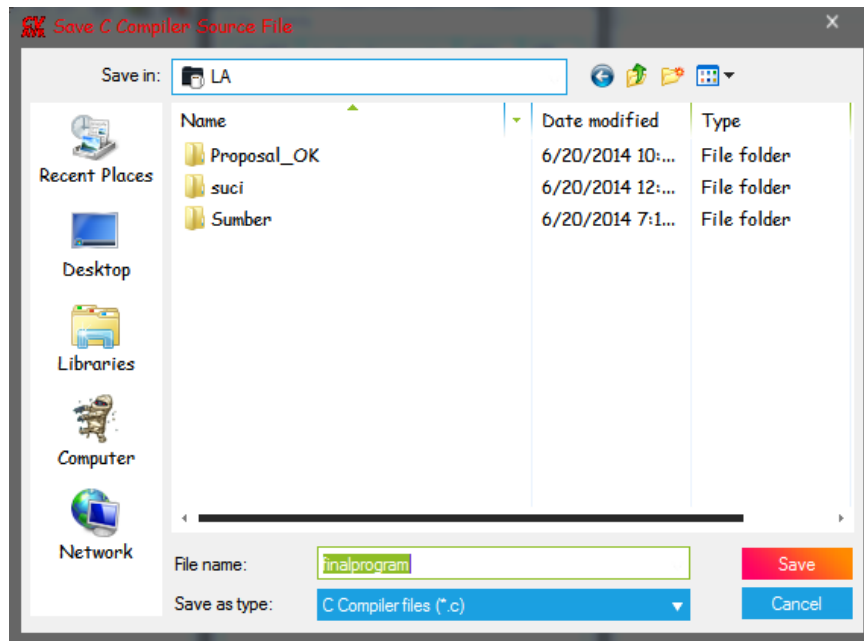
Gambar 2.18 Pengaturan *chip* pada *CodeVisionAVR*

Kemudian klik *PORT* untuk memilih dan mengatur *PORT* yang akan digunakan apakah sebagai *In* atau *Out*, atau memilih *PORTA*, *PORTB*, *PORTC*.

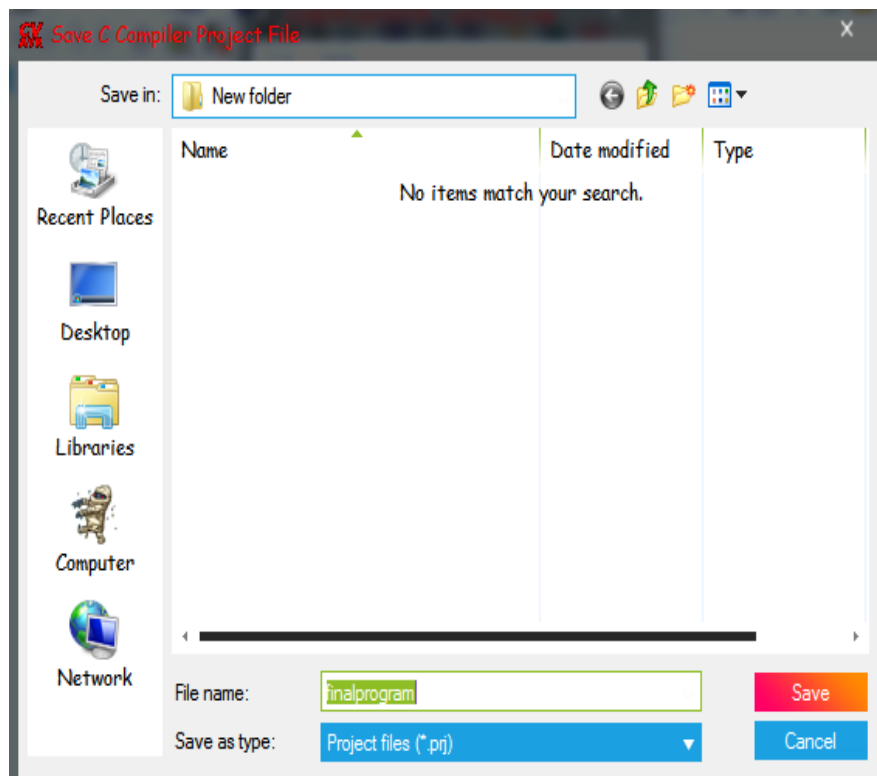


Gambar 2.19 Pengaturan *port* pada *CodeVisionAVR*

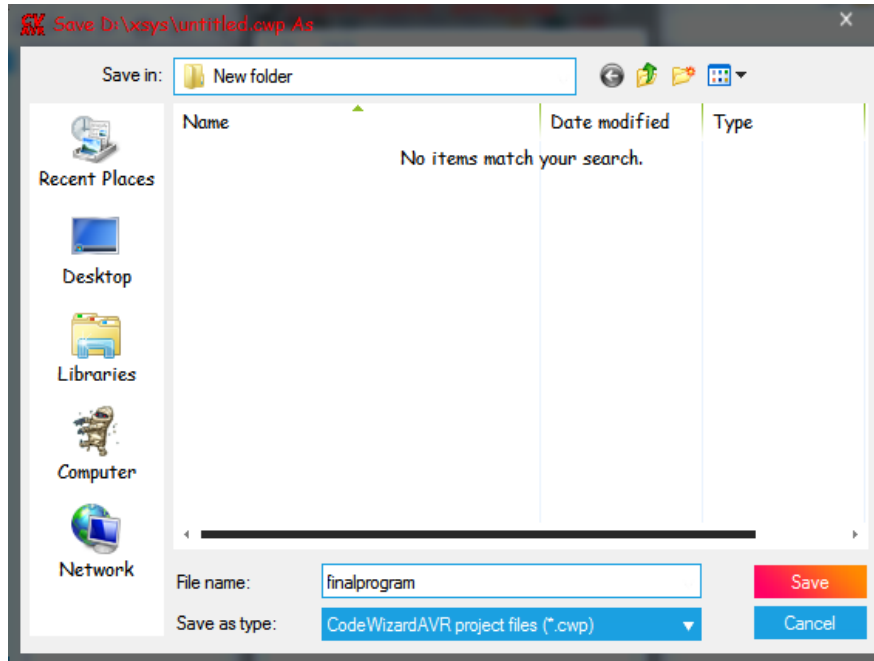
Kemudian beri nama *file* pertama lalu simpan. *File* ini bertipe *C Compiler (*.C)*

Gambar 2.20 Penyimpanan *file.C*

Kemudian beri nama lagi *file* yang kedua lalu simpan. *File* ini bertipe *ProjectFiles (*.PRJ)*

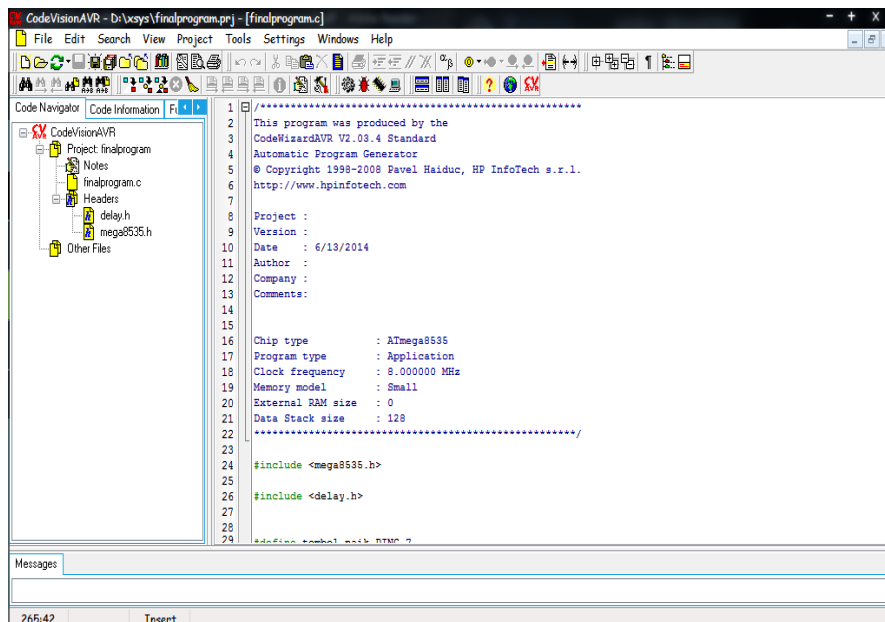
Gambar 2.21 Penyimpanan *file.PRJ*

Terakhir beri nama *file* lagi yang ketiga kemudian simpan. *File* ini bertipe *CodeWizardAVR Project file (*.cwp)*. Catatan : Usahakan nama Ketiga *File* sama agar lebih mudah mencari *file* tersebut.



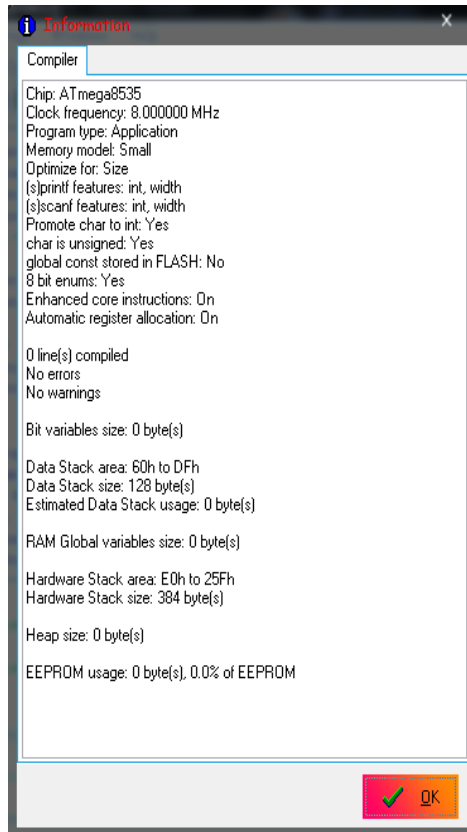
Gambar 2.22 Penyimpanan *file.cwp*

Tampilan awal setelah menggunakan *Code Wizard*



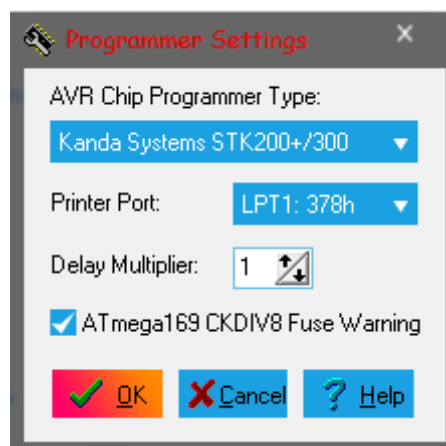
Gambar 2.23 Tampilan awal pada saat menggunakan *code wizard*

Jika sudah selesai membuat program, maka *compile* program, pilih *Project* klik *compile*.



Gambar 2.24 Hasil proses kompilasi

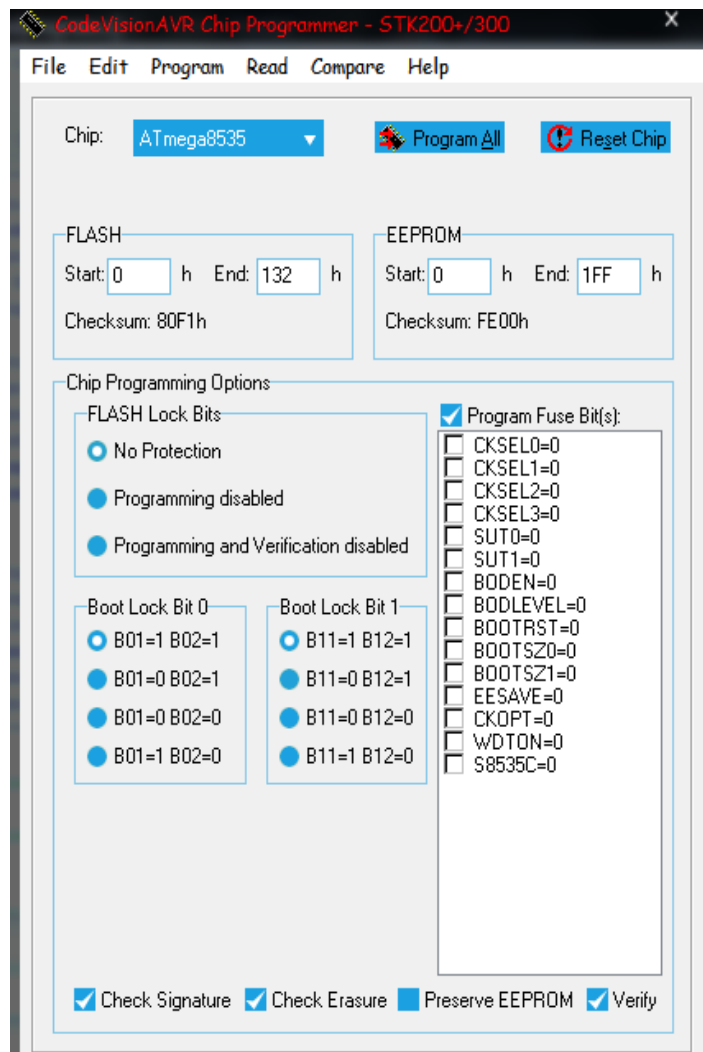
Untuk memasukkan program yang sudah dibuat ke *IC* mikrokontroler AVR ATmega8535, maka terlebih dahulu *setting programmer*, pilih *setting|| programmer*, pilih tanda sistem STK200+/300 untuk AVR *chip programmer type*. Pilih *printer port*=LPT1:378h, dan biarkan *Delay Multiplier* =1, dan pilih untuk Atmega169. Lalu klik tombol OK.



Gambar 2.25 *Setting programmer*

Sebelum memasukkan program *chip* mikrokontroler, *setting Fuse Bit* terlebih dahulu. Untuk *Flash Lock Bit* biarkan dalam keadaan “*No Protection*”.

Untuk *Boot Lock Bit 0*, *Boot Lock Bit 1*, *check Signature*, *check Erasure*, *Preserve EEPROM* dan *Verify* biarkan seperti awalnya, karena jika terdapat kesalahan dalam mengatur mengakibatkan *chip* mikrokontroler tidak dapat dibaca atau diprogram kembali dan harus dihapus menggunakan program khusus. *Fuse Bit* adalah bit yang mengatur konfigurasi dasar mikrokontroler, apabila tidak dilakukan pengaturan, maka penggunaan *Timer/Counter* atau komunikasi serial *USART* tidak dapat berjalan seperti yang diinginkan karena beberapa pengaturan menyangkut penggunaan Kristal.



Gambar 2.26 Pengaturan *chip programming options*

2.11. Analog To Digital Converter (ADC)

ATMega8535 menyediakan fasilitas ADC dengan resolusi 10 bit. ADC ini dihubungkan dengan 8 *channel Analog Multiplexer* yang memungkinkan

terbentuk 8 input tegangan *single-ended* yang masuk melalui pin pada Port A. ADC memiliki pin *supply* tegangan analog yang terpisah yaitu AVCC. Besarnya tegangan AVCC adalah $\pm 0.3V$ dari VCC.

Tegangan referensi ADC dapat dipilih menggunakan tegangan referensi internal maupun eksternal. Jika menggunakan tegangan referensi internal, bisa dipilih *on-chip* internal reference voltage yaitu sebesar 2.56 volt atau sebesar AVCC. Jika menggunakan tegangan referensi eksternal dapat dihubungkan melalui pin AREF.

ADC mengkonversi tegangan input analog menjadi data digital 8 bit atau 10 bit. Data digital tersebut akan disimpan didalam ADC data register yaitu ADCH dan ADCL. Sekali ADCL dibaca, maka akses ke data register tidak bisa dilakukan, dan ketika ADCH dibaca maka akses ke data register kembali *enable* (Heryanto,2008).

2.11.1. Inisialisasi ADC

Menurut Heryanto (2008) ada beberapa langkah yang harus dilakukan untuk inisialisasi ADC pada mikrokontroler ATMEGA8535, yaitu penentuan *clock*, tegangan referensi, format data input dan *mode* pembacaan. Inisialisasi ini dilakukan pada register- register berikut :

1. ADMUX

Pada register ini bertujuan untuk mengatur tegangan referensi yang digunakan ADC, format data output dan saluran ADC. Terlihat pada tabel 2.5 merupakan register- register dari ADMUX.

Tabel 2.5 Register ADMUX

REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
-------	-------	-------	------	------	------	------	------

a. REF₀₋₁ adalah bit- bit pengatur *mode* tegangan referensi ADC.

Tabel 2.6 REF₀₋₁

REFS ₁	REFS ₀	Mode Tegangan Referensi
0	0	Pin Vref
0	1	VCC
1	0	Tidak digunakan
1	1	Vref internal = 2,56 Volt

b. ADLAR adalah bit keluaran ADC.

Jika ADC telah selesai konversi maka data ADC akan diletakkan di 2 register, yaitu ADCH dan ADCL dengan format sesuai ADLAR. Pada tabel 2.7 dan tabel 2.8 format ADC dengan ADLAR dengan masing- masing nilai.

Tabel 2.7 Format data ADC dengan ADLAR = 0

-	-	-	-	-	-	ADC9	ADC8	ADH
ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ACL

Tabel 2.8 Format data ADC dengan ADLAR = 1

ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADH
ADC1	ADC0	-	-	-	-	-	-	ACL

c. MUX0-4 adalah bit-bit pemilihan saluran pembacaan ADC.

2. ADCSRA

ADCSRA adalah register 8 bit yang berfungsi untuk melakukan manajemen sinyal kontrol dan status ADC yang terlihat pada tabel 2.9.

Tabel 2.9 Register ADCSRA

ADEN	ADCS	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
------	------	-------	------	------	-------	-------	-------	--------

- ADEN merupakan bit pengatur aktivasi ADC. Jika bernilai 1 maka ADC aktif.
- ADCS merupakan bit penanda dimulainya konversi ADC. Selama konversi berlogika 1 dan akan berlogika 0 jika selesai konversi.
- ADATE merupakan bit pengatur aktivasi picu otomatis. Jika bernilai 1 maka konversi ADC akan dimulai pada saat tepi positif pada sinyal *trigger* digunakan.
- ADIF merupakan bit pengatur aktivasi interupsi. Jika bernilai 1 maka interupsi penandaan telah selesai, konversi ADC diaktifkan.
- ADPS₀₋₂ merupakan bit pengatur *clock* ADC. Pada tabel 2.10 merupakan tabel konfigurasi *clock* ADC.

Tabel 2.10 Konfigurasi *clock* ADC

ADPS₂₋₀	Clock ADC
000-001	$f_{osd} 2$
010	$f_{osd} 4$
011	$f_{osd} 8$
100	$f_{osd} 16$
101	$f_{osd} 32$
110	$f_{osd} 64$
111	$f_{osd} 128$

3. SFIOR

SFIOR adalah register 8 bit yang mengatur sumber pemacu ADC. Jika bit ADATE pada register ADCSRA bernilai 0 maka ADTS₀₋₂ tidak berfungsi.

Tabel 2.11 Register SFIOR

ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10	SFIOR
-------	-------	-------	---	------	-----	------	-------	-------

Tabel 2.12 Penjelasan register SFIOR

ADTS2	ADTS1	ADTS0	Sumber Pemacu
0	0	0	<i>Free Running Mode</i>
0	0	1	<i>Analog Comparator</i>
0	1	0	<i>Extenal Interupt Request 0</i>
0	1	1	<i>Timer/ Counter 0 Compare Match</i>
1	0	0	<i>Timer/ Counter 0 Overflow</i>
1	0	1	<i>Timer/ Counter 1 Compare Match B</i>
1	1	0	<i>Timer/ Counter 1 Overflow</i>
1	1	1	<i>Timer/ Counter 0 Capture Event</i>