BAB II
TINJAUAN PUSTAKA

2.1 Air

Air adalah zat atau materi atau unsur yang penting bagi semua bentuk kehidupan yang diketahui sampai saat ini di bumi, tetapi tidak di planet lain dalam Sistem Tata Surya dan menutupi hampir 71% permukaan bumi (Matthews, 2005). Wujudnya bisa berupa cairan, es (padat) dan uap/gas. Dengan kata lain karena air, maka Bumi menjadi satu-satunya planet dalam Tata Surya yang memiliki kehidupan (Parker, 2007).

Manusia dan semua makhluk hidup lainnya butuh air. Air merupakan material yang membuat kehidupan terjadi di muka bumi. Menurut dokter dan ahli kesehatan manusia wajib minum air putih 8 gelas per hari. Namun, air yang diminum sudah pasti harus air yang bebas dari bakteri dan kuman. Karena jika di dalam air tersebut masih terdapat bakteri dan kuman, maka akan membuat kesehatan manusia terganggu.

Pemanasan yang secukupnya akan membunuh kuman dan bakteri dalam air yang terkontaminasi pada suhu yang masih di bawah titik didih. Saat mencapai titik didih (100^0C), air yang dipanaskan sudah cukup panas agar kuman dan bakteri mati. Tidak perlu mendidihkan air selama 5 menit, 10 menit atau 20 menit seperti yang direkomendasikan oleh banyak sumber. (Sumber : Robert J. Kodoatie & Roestam Sjarief, 2010)

2.2 Beras

Beras adalah butir padi yang telah dibuang kulit luarnya (sekamnya) yang menjadi dedak kasar (Sediotama, 1989). (Astawan, 2004) Beras adalah gabah yang bagian kulitnya sudah dibuang dengan cara digiling dan disosoh menggunakan alat pengupas dan penggiling serta alat penyosoh. Berdasarkan artikel pada halaman web *http://id.wikihow.com/Memasak-Nasi-dengan-Rice-Cooker*, dalam mendeteksi tingat kematangan nasi adalah menggunakan sifat titik didih air yaitu 100^0 C. Dalam kasus

ini, sebuah *rice cooker* akan otomatis berhenti memasak nasi jika suhu sudah mencapai 100^0C, dan selanjutnya suhu akan berubah menjadi 70^0C untuk menghangatkan nasi yang berada di dalamnya. Berarti, ini dapat juga dikatakan bahwa nasi yang sedang dimasak, apabila suhunya sudah mencapai 100^0C maka sudah dinyatakan matang.

2.3 Sayuran

Sayuran adalah suatu makanan bervitamin yang bermanfaat bagi tubuh kita (Widagdo, 1992). Sayuran memiliki manfaat yang besar sebagai sumber gizi maupun sumber tenaga untuk melakukan aktivitas sehari-hari. Banyak sekali jenis-jenis sayuran namun terkadang masyarakat hanya mengkonsumsi sayuran yang itu-itu saja, padahal jika setiap hari sayuran yang dikonsumsi lebih beragam, maka tidak akan ada rasa jenuh untuk mengkonsumsi sayuran. Selain itu, vitamin yang didapat lebih banyak dari pada mengkonsumsi satu atau dua jenis sayuran saja. Dalam memasak sayuran tidak membutuhkan waktu yang lama dan suhu yang terlalu panas. Karena, suhu yang terlalu panas dan waktu memasak yang terlalu lama, dapat membuat vitamin yang terkandung didalam sayuran menjadi hilang dan rusak.

2.4 Kompor

Kompor adalah alat yang digunakan untuk memanaskan suatu benda (Sigit Pramuko, 2011). Di dalam rumah tangga, kompor merupakan alat memasak yang utama. Selain untuk memasak makanan, kompor juga digunakan untuk menghangatkan makanan yang sudah dimasak sebelumnya. Salah satu jenis kompor adalah kompor listrik. Kompor listrik memiliki cara pembangkitan dan sumber panas yang berbeda dengan kompor yang menggunakan bahan bakar gas ataupun minyak. Sumber panas pada kompor listrik berasal dari energi di dalam arus listrik, sedangkan pembangkitan panasnya tercipta ketika aliran arus listriknya dihambat. Gambar 2.1 memperlihatkan bentuk dari kompor listrik.

Gambar 2.1 Kompor Listrik

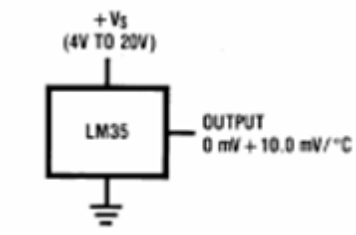
2.5 Suhu

Suhu adalah ukuran panas atau dinginnya suatu benda (Esvandiari, 2006). Definisi yang lebih tepat menyatakan suhu adalah ukuran kelajuan gerak partikel-partikel dalam suatu benda atau ukuran kinetik rata-rata partikel dalam suatu benda.

Dalam kehidupan sehari-hari dalam mengukur suhu, masyarakat cenderung menggunakan indera peraba. Namun, dalam dunia modern saat ini, dalam mengukur suatu suhu sudah dapat dilakukan dengan cara yang mudah, yaitu dengan menggunakan sensor. Salah satu sensor suhu yang sering digunakan adalah IC LM35.

2.5.1 IC LM35

Untuk mendeteksi suhu digunakan sebuah sensor suhu LM35 yang dapat dikalibrasikan langsung (AF. Nasution, 2011). LM35 ini difungsikan sebagai *basic temperature* sensor seperti pada gambar 2.2



Gambar 2.2 LM 35 *Basic Temperature* Sensor

Sensor LM35 memiliki 3 buah pin, yang masing-masing jika nampak dari depan memiliki fungsi yaitu pin 1 berfungsi sebagai sumber tegangan kerja dari senor LM35, pin 2 atau tengan digunakan sebagai tegangan keluaran atau Vout dan pin 3 berfungsi sebagai ground. Vout pada sensor LM35 memiliki tegangan keluaran yang terskala linier terhadap suhu terukur, yaitu 10mV per 1^0C. Jadi didapatkan rumus sebagai berikut :

Vout = Data Suhu x 10mV (Persamaan 2.1)

Fitur LM35 adalah sebagai berikut :

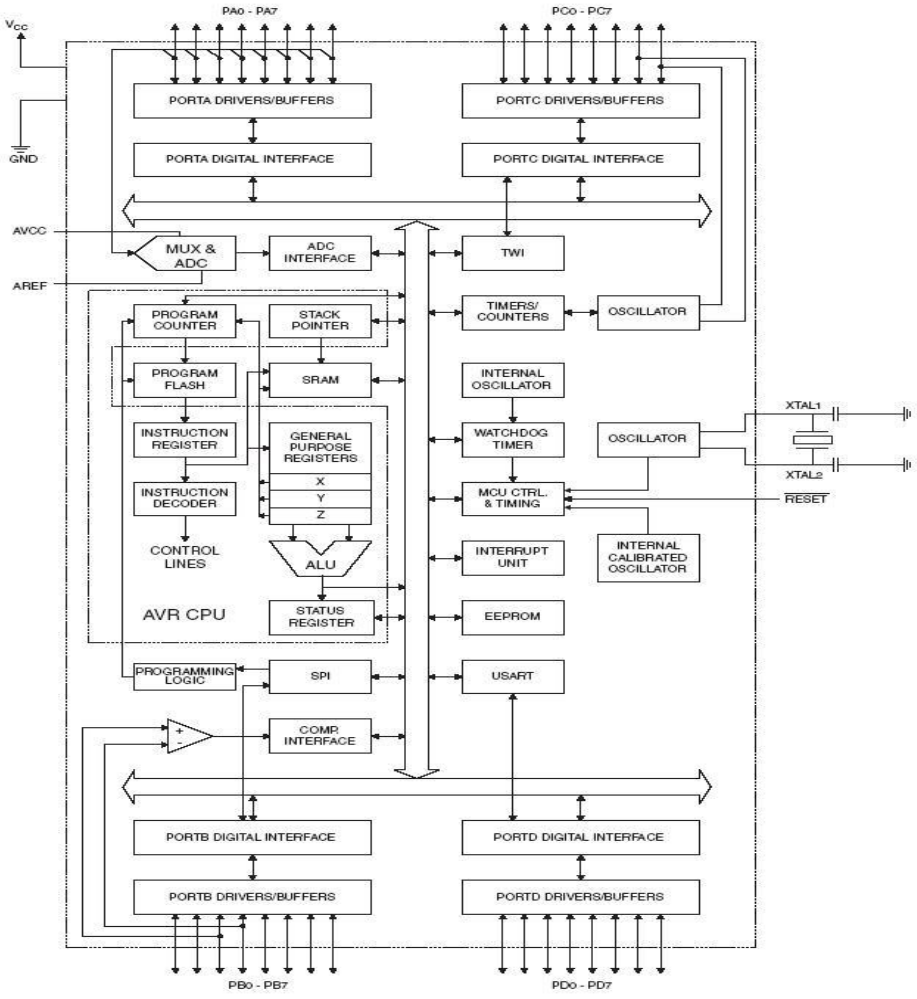
1. *Centigrade* atau *celcius*
2. Sensitivitas10 mV/ ^{0}C
3. Akurasi 0,5 C pada suhu 25 ^{0}C
4. Handal dalam pengukuran jarak jauh
5. Tegangan operasi 4 s/d 30 V
6. Konsumsi arus 60 uA
7. *Selt heating* rendah 0,08 ^{0}C
8. *Non linieritas* ± 0,25 ^{0}C
9. Impedansi keluaran 0,1 Ω untuk arus beban 1 mA.

2.6 Mikrokontroler AVR ATMega8535

Mikrokontroler AVR memiliki arsitektur RISC 8 Bit, sehingga semua instruksi dikemas dalam kode 16-bit (16-bits *word*) dan sebagian besar instruksi dieksekusi dalam satu siklus instruksi *clock*. Dan ini sangat membedakan sekali dengan instruksi MCS-51 (Berarsitektur CISC) yang membutuhkan siklus 12 *clock*. ATMega 8535 merupakan *chip* IC keluaran ATMEL yang termasuk golongan *single chip microcontroller,* dimana semua rangkaian termasuk I/O dan memori tergabung dalam satu bentuk IC. (Agus Bejo, 2008).

2.6.1 Diagram Blok ATMega 8535

Ada 3 jenis tipe AVR yaitu AT Tiny, AVR klasik, AT Mega. Perbedaannya hanya pada fasilitas dan I/O yang tersedia serta fasilitas lain seperti ADC, EEPROM dan lain sebagainya. ATMega 8535 merupakan salah satu tipe AVR yang memiliki teknologi RISC dengan kecepatan maksimal 16 MHz membuat ATMega8535 lebih cepat bila dibandingkan dengan varian MCS 51. Dengan fasilitas yang lengkap tersebut menjadikan ATMega8535 sebagai mikrokontroler yang *powerfull*. Adapun blok diagramnya adalah sebagai berikut seperti terlihat pada gambar 2.3 yang diambil dari pustaka (W.N. Riantiningsih, 2009).



Gambar 2.3 Diagram Blok ATMega8535

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

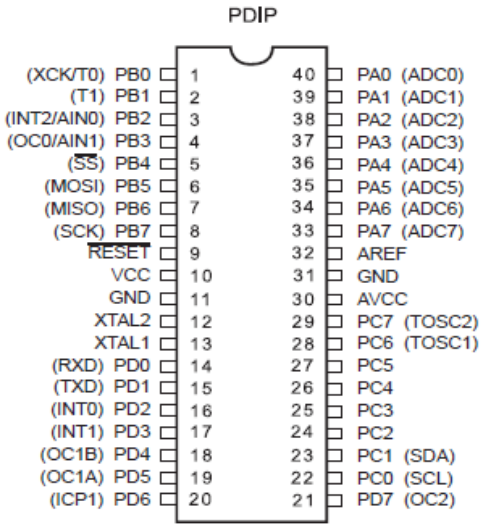
2.6.2 Fitur ATMega 8535

Mikrokontroler ATMega8535 memiliki beberapa fitur diantaranya sebagai berikut :

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 saluran.
4. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan *mode sleep* menghemat penggunaan daya listrik.

2.6.3 Konfigurasi Pin ATMega8535

IC Atmega 8535 ada 2 jenis yaitu jenis PDIP (berbentuk balok) dan jenis TQFP/MLF (berbentuk kotak) yang pada dasarnya memiliki fasilitas yang sama, hanya saja memiliki bentuk yang berbeda sehingga letak kaki-kaki IC berbeda mengikuti bentuknya. Gambar 2.4. berikut ini merupakan konfigurasi pin mikrokontroler ATMega8535 yang diambil dari pustaka (W.N. Riantiningsih, 2009).



Gambar 2.4 Konfigurasi Pin ATMega8535

Dari Gambar 2.4 diatas dapat dilihat ada 40 pin IC yang Secara fungsional konfigurasi pin tersebut sebagai berikut :

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

2.7 Motor Stepper

Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. (Wayan Supardi, 2012)

Keunggulannya antara lain adalah :

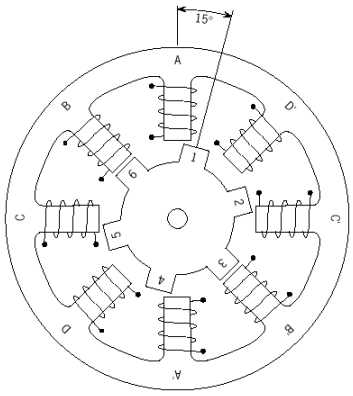
1. Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
2. Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak
3. Posisi dan pergerakan repetisinya dapat ditentukan secara presisi
4. Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran)
5. Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC
6. Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya
7. Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada range yang luas.

Pada dasaranya terdapat 3 tipe motor stepper yaitu:

1. Motor stepper tipe *Variable reluctance* (VR)

   Motor stepper jenis ini telah lama ada dan merupakan jenis motor yang secara struktural paling mudah untuk dipahami. Motor ini terdiri atas sebuah rotor besi lunak dengan beberapa gerigi dan sebuah lilitan stator. Ketika lilitan stator diberi energi dengan arus DC, kutub-kutubnya menjadi termagnetasi. Perputaran terjadi ketika gigi-gigi rotor tertarik oleh kutub-kutub stator.

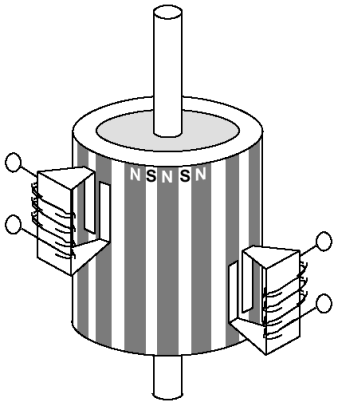
Berikut ini adalah penampang melintang dari motor stepper tipe *variable reluctance* (VR):



Gambar 2.5 Penampang melintang dari motor stepper tipe *variable reluctance* (VR)

2. Motor stepper tipe *Permanent Magnet* (PM)

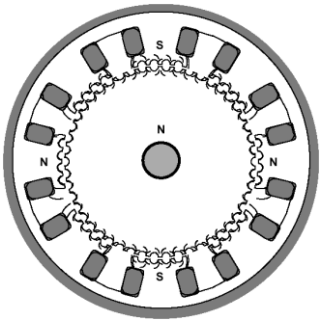
Motor stepper jenis ini memiliki rotor yang berbentuk seperti kaleng bundar (*tin can*) yang terdiri atas lapisan magnet permanen yang diselang-seling dengan kutub yang berlawanan (perhatikan gambar 2.9). Dengan adanya magnet permanen, maka intensitas fluks magnet dalam motor ini akan meningkat sehingga dapat menghasilkan torsi yang lebih besar. Motor jenis ini biasanya memiliki resolusi langkah (*step*) yang rendah yaitu antara $7{,}5^0$ hingga 15^0 per langkah atau 48 hingga 24 langkah setiap putarannya. Berikut ini adalah ilustrasi sederhana dari motor stepper tipe *permanent magnet*:



Gambar 2.6 Ilustrasi sederhana dari motor stepper tipe *permanent magnet* (PM)

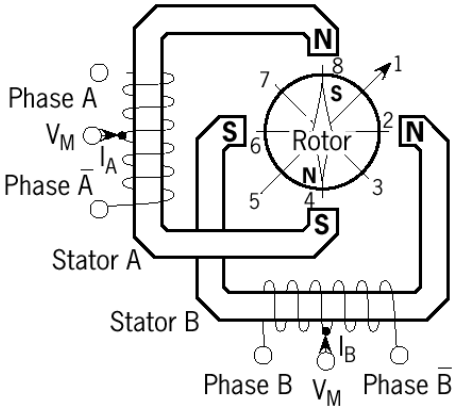
3. Motor stepper tipe *Hybrid* (HB)

   Motor stepper tipe hibrid memiliki struktur yang merupakan kombinasi dari kedua tipe motor stepper sebelumnya. Motor stepper tipe hibrid memiliki gigi-gigi seperti pada motor tipe VR dan juga memiliki magnet permanen yang tersusun secara aksial pada batang porosnya seperti motor tipe PM. Motor tipe ini paling banyak digunkan dalam berbagai aplikasi karena kinerja lebih baik. Motor tipe hibrid dapat menghasilkan resolusi langkah yang tinggi yaitu antara $3{,}6^0$ hingga $0{,}9^0$ per langkah atau 100-400 langkah setiap putarannya. Berikut ini adalah penampang melintang dari motor stepper tipe hibrid:



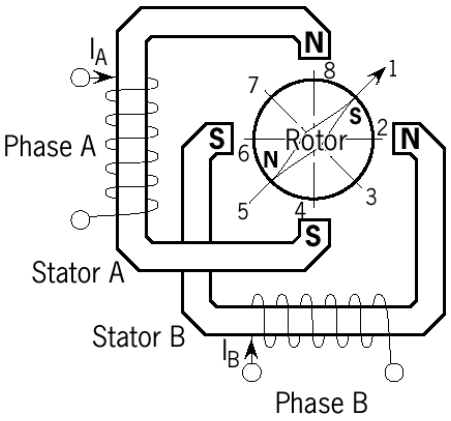
Gambar 2.7 Penampang melintang dari motor stepper tipe *hybrid*

Berdasarkan metode perancangan rangkain pengendalinya, motor stepper dapat dibagi menjadi jenis unipolar dan bipolar. Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (*ground*) pada salah satu terminal lilitan (*wound*) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (VM) pada bagian tengah (*center tap*) dari lilitan (perhatikan gambar 2.8).



Gambar 2.8 Motor stepper dengan lilitan unipolar

Untuk motor stepper dengan lilitan bipolar, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya (perhatikan gambar 2.9). Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor unipolar. Motor stepper bipolar memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama.

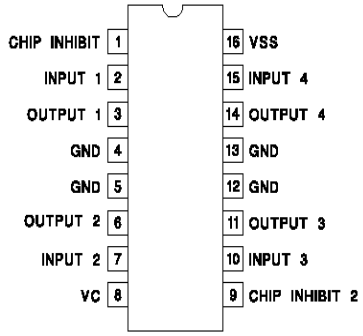


Gambar 2.9 Motor stepper dengan lilitan bipolar

2.8 Perangkat H – *Bridge*

Rangkaian dasar yang biasa digunakan untuk menggerakkan motor DC dikenal dengan nama rangkaian *H-Bridge*, rangkaian ini digunakan jika motor akan digerakkan dalam dua arah. *H-Bridge* adalah suatu rangkaian elektronik yang memungkinkan motor DC dapat berputar ke depan (*forward*) atau ke belakang (*backward*). *H-Bridge* tersedia dalam bentuk *Integrated Circuit* (IC) atau dapat dirangkai sendiri dari komponen-komponen elektronika. Saat ini IC penggerak motor DC sudah banyak beredar dipasaran, diantaranya L293D, L298, UCN 2998, dan lain-lain.

Pemilihan tipe IC (*chip*) yang digunakan berdasarkan nilai tegangan dan arus motor yang mampu dikendalikannya. Untuk motor kecil (< 600 mA), salah satu *driver* tesebut yaitu L293D, sedangkan kebutuhan arus motor yang lebih besar (< 4 A) dapat digunakan L298. IC L293D berisi empat *driver* H berarus tinggi. IC tersebut didesain guna menyediakan pengatur (*driver*) arus listrik secara dua arah (*bidirectional*) hingga mencapai lebih dari 1 A pada tegangan dari 4.5 V sampai dengan 36 V. Gambar skematis *pin-pin* pada IC L293D yang ditunjukkan pada gambar 2.10.



Gambar 2.10 Konfigurasi *Pin Driver* Motor DC (L293D)

Pada IC L293D dapat digunakan untuk mengendalikan dua motor sekaligus, yang dikendalikan yaitu kecepatan putar (dengan PWM) dan arah putar. Sinyal PWM yang berasal dari mikrokontroler menjadi masukkan kaki EN (*Enable*) L293D, sedangkan IN1 dan IN2 mendapat masukkan sumber tetap.

Namun, jika dikehendaki, dapat mengendalikannya dengan kontroler, maka perlu menghubungkan ke mikrokontroler. Jadi, secara ideal, sebuah motor membutuhkan 3 pin I/O mikrokontroler. Adapun tabel kebenaran L293D terdapat pada tabel 2.1.

Tabel 2.1 Tabel Kebenaran L293D

EN (*Enable*)	IN1	IN2	Fungsi
1	1	0	Putar Kanan
1	0	1	Putar Kiri
1	0	0	Henti Segera
1	1	1	Henti Segera
0	X	x	Henti Lambat

2.9 LCD (*Lycuid Crystal Display*)

Menurut (Wardhana, Lingga 2006). *Display* LCD 16x2 berfungsi sebagai penampil karakter yang di *input* melalui *keypad*. LCD yang digunakan pada alat ini mempunyai lebar *display* 2 baris 16 kolom atau biasa disebut sebagai LCD *Character* 16x2, dengan 16 pin konektor. Berikut gambar 2.11 yang menunjukan gambar LCD (*Lycuid Cristal Display*) yang digunakan pada alat ini.

Gambar 2.11 LCD (*Lycuid Cristal Display*)

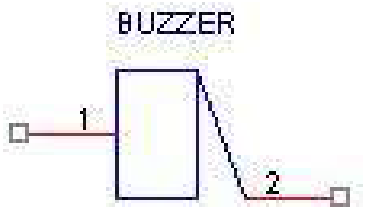
Berikut ini merupakan tabel yang menjelaskan mengenai konfigurasi pin dari LCD 16x2 :

Tabel 2.2 Konfigurasi Pin LCD 16 x 2

Pin	Simbol	Level	Tujuan	Fungsi
1	Vss	-	*Power Supply*	*Ground*
2	V_{DD}	-	*Power Supply*	Tegangan *Supply*
3	V_{LS}	-	*Power Supply*	*Power Supply* untuk men-*drive* LCD guna mengatur kontrasnya
4	RS	H/L	uC	H : Data; L : *Instruction Code*
5	R/W	H/L	uC	H : *Read*; L : *Write*
6	E	H/L	uC	*Enable*
7	DB0	H/L	uC	Data Bus Line
8	DB1	H/L	uC	
9	DB2	H/L	uC	
10	DB3	H/L	uC	
11	DB4	H/L	uC	
12	DB5	H/L	uC	
13	DB6	H/L	uC	
14	DB7	H/L	uC	

15	V+BL	-	*Back Ligh Supply*	Tegangan *Supply*
16	V-BL	-	*Back Ligh Supply*	*Ground*

2.10 *Buzzer*



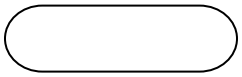
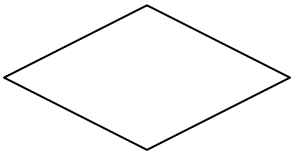
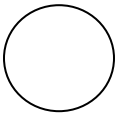
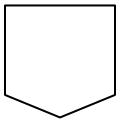
Gambar 2.12 Simbol *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* ini digunakan sebagai indikator *(alarm)*.

2.11 *Flowchart*

Flowchart merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan tipe operasi program yang berbeda. Sebagai *representasi* dari sebuah program. *Flowchart* maupun algoritma dapat menjadi alat bantu untuk memudahkan perancangan alur urutan logika suatu program, memudahkan pelacakan sumber kesalahan program, dan alat untuk menerangkan logika program. Berikut merupakan simbol-simbol yang sering digunakan dalam *flowchart* :

Tabel 2.3 Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
	*Terminator*	Permulaan/akhir program
	Garis alir	Arah alir program
	*Preparation*	Proses inisialisasi / pemberian harga awal
	Proses	Proses perhitungan/proses pengolahan data
	*Input / output data*	Proses *input/output* data, parameter, informasi
	*Predefined process* (sub program)	Permulaan sub program / proses menjalankan sub program
	*Decision*	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	*On page connector*	Penghubung ke bagian-bagian *flowchart* yang berada pada satu halaman
	*Off page connector*	Penghubung bagian-bagian *flowchart* yang berada pada halaman berbeda

2.12 Bahasa Pemrograman C

Bahasa program adalah suatu bahasa ataupun suatu tata cara yang dapat digunakan oleh manusia (*programmer*) untuk berkomunikasi secara langsung dengan komputer. Bahasa C adalah bahasa pemrograman yang dapat dikatakan berada di antara bahasa beraras rendah dan beraras tinggi. Bahasa beraras rendah artinya bahasa yang berorientasi pada mesin, sedangkan beraras tinggi berorientasi pada manusia. Bahasa beraras rendah misalnya assembler, ditulis dengan sandi yang hanya dimengerti oleh mesin sehingga hanya digunakan bagi yang memprogram mikroprosesor. Bahasa beraras tinggi relatif mudah digunakan karena ditulis dengan bahasa manusia sehingga mudah dimengerti dan tidak tergantung mesinnya. Bahasa beraras tinggi umumnya digunakan pada komputer.

Program Bahasa C tidak mengenal aturan penulisan dikolom tertentu sehingga bisa dimulai dari kolom manapun. Namun demikian untuk mempermudah pembacaan program dan keperluan dokumentasi sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak di baca. Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukan main ( ). Program yang dijalankan berada dalam tubuh program dan dimulai dengan tanda kurung buka {dan diakhiri dengan kurung tutup }. Semua yang tertulis di dalam tubuh program disebut blok.

Tanda () digunakan untuk mengapit *argument* suatu fungsi. Argumen adalah suatu nilai yang akan digunakan dalam fungsi tersebut. Dalam fungsi main tidak ada argument sehingga tak ada data dalam (). Yang merupakan perintah dan harus dikerjakan oleh prosesor. Setiap pernyataan diakhiri tanda titik koma ;. Baris pertama #include <….> bukanlah pernyataan sehingga tak diakhiri tanda titik koma (;). Baris tersebut meminta *compiler* untuk menyertakan file yang namanya ada di antara tanda <…> dalam proses kompilasi. File-file ini (berekstensi.h) berisi deklarasi fungsi ataupun *variable*. File ini disebut *header* dan digunakan semacam perpustakaan untuk pernyataan yang ada di tubuh program. (Agus Bejo, 2008).

a) Tipe Data

Tipe data merupakan bagian program paling penting karena mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Tipe-tipe data tersebut dapat dilihat pada Tabel 2.4.

Tabel 2.4 Tipe-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	1 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	2 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 $3.4*10^{+38}$

b) Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap dan harus didefenisikan terlebih dahulu di awal program. Kemudian konstanta dapat bernilai *integer*, pecahan, karakter, dan *string*.

c) Variabel

Variabel adalah suatu pengenalan yang digunakan untuk mewakili nilai tertentu dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai suatu variabel bisa di ubah-ubah sesuai kebutuhan. Namun suatu variabel dapat ditentukan sendiri oleh program dengan aturan sebagai berikut:

1. Terdiri atas gabungan huruf dan angka dengan karakter pertama harus berupa huruf.
2. Tidak boleh mengandung spasi
3. Tidak boleh mengandung simbol khusus kecuali garis bawah
4. Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai

d) Deklarasi Variabel

Bentuk umum pendeklarasian suatu variabel adalah: Nama_tipe nama_variabel. Contohnya :

1. Int x; // *Deklarasi x bertipe integer*
2. Char y, huruf, nim[10]; // *deklarasi variabel bertipe char*
3. Float nilai; // *deklarasi variabel bersifat float*
4. *Double* beta; // *deklarasi variabel bertipe double*
5. Int array[5][4]; // *deklarasi array bertipe integer*
6. Char *p; // *deklarasi pointer p bertipe char*

e) Deklarasi Konstanta

Dalam bahasa C konstanta dideklarasikan menggunakan *preprocessor* #define. Contohnya :

#define PH 3.14

#define nama “muttakin”

f) Deklarasi Fungsi

Fungsi merupakan bagian terpisah dari program dan dapat diaktifkan atau dipanggil di mana pun dalam program. Ada fungsi dalam bahasa C yang sudah disediakan sebagai fungsi pustaka, seperti printf(), scanf(), dang etch(). Kemudian fungsi tersebut tidak perlu dideklarasikan untuk menggunakannya.

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi ysng dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah:

Tipe_fungsi nama_fungsi(parameter_fungsi);

contohnya:

1. Float luas_lingkaran(int jari);
2. Void tampil();
3. Int tambahan(int x, int y);

g) Operasi Aritmatika

Bahasa C menyediakan lima operator aritmatika yaitu :

1. .* : untuk perkalian
2. / : untuk pembagian
3. % : untuk sisa pembagian (modulus)
4. + : untuk penjumlahan
5. - : untuk pengurangan

h) Operasi Logika

Jika operator hubungan membandingkan hubungan antara dua *operand* maka operator logika digunakan untuk membandingkan logika hasil operator-operator hubungan. Operator logika ada tiga macam yaitu:

1. && : Logika AND (dan)
2. || : Logika OR (atau)

3. ! : Logika NOT (ingkaran)

Operasi AND akan bernilai benar jika dua ekspresi bernilai benar. Operasi OR akan bernilai benar jika dan hanya jika salah satu eksresinya bernilai benar. Sementara operasi NOT menghasilkan nilai benar jika ekspresinya bernilai salah, dan akan bernilai salah jika ekspresinya bernilai benar.

i) Komentar Program

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan keterangan atau penjelasan program. Cara memberikan komentar atau penjelasan dalam bahasa C adalah menggunakan pembatas /* dan */ atau menggunakan tanda // untuk komentar yang hanya terdiri atas satu baris. Komentar program tidak akan ikut diproses dalam program (akan diabaikan).

Contoh pertama:

// program ini dibuat oleh……

Dibelakang tanda // tak akan diproses dalam kompilasi. Tanda ini hanya untuk satu baris kalimat.

Contoh kedua:

//* program untuk memutar motor DC atau motor stepper */

Bentuk ini berguna kalau pernyataannya berupa kalimat panjang sampai beberapa baris.

2.13 *CodeVisionAVR*

CodeVisionAVR merupakan sebuah *cross-compiler* C, *Integrated Development Environtment* (IDE), dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, 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.

IDE mempunyai fasilitas *internal* berupa *software* AVR *Chip In-System Programmer* yang memungkinkan Anda untuk melakukan transfer program kedalam *chip* mikrokontroler setelah sukses melakukan kompilasi/asembli secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 programmers/development boards.

Untuk keperluan debugging sistem embedded, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah Terminal. Selain library standar C, *CodeVisionAVR* juga mempunyai *library* tertentu untuk:

1. Modul LCD alphanumeric
2. Bus I2C dari Philips
3. Sensor Suhu LM75 dari National Semiconductor
4. Real-Time Clock: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas Semiconductor
5. Protokol 1-Wire dari Maxim/Dallas Semiconductor
6. Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas Semiconductor
7. Termometer/Termostat DS1621 dari Maxim/Dallas Semiconductor
8. EEPROM DS2430 dan DS2433 dari Maxim/Dallas Semiconductor
9. SPI
10. Power Management
11. Delay

12. Konversi ke Kode Gray

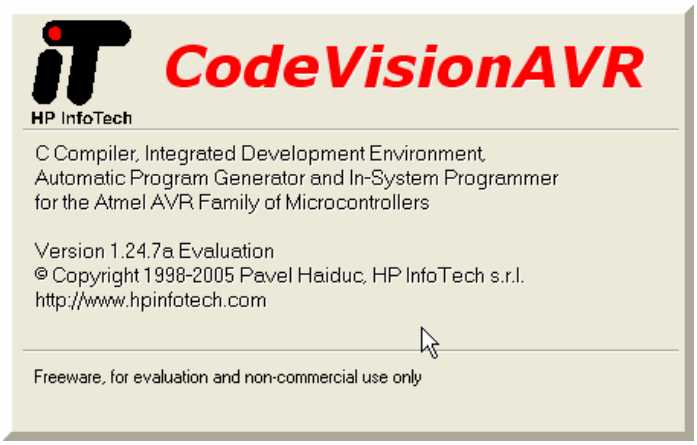
CodeVisionAVR juga mempunyai Automatic Program Generator bernama CodeWizardAVR, yang mengujinkan Anda untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

1. Set-up akses memori eksternal
2. Identifikasi sumber reset untuk chip
3. Inisialisasi port input/output
4. Inisialisasi interupsi eksternal
5. Inisialisasi Timer/Counter
6. Inisialisasi Watchdog-Timer
7. Inisialisasi UART (USART) dan komunikasi serial berbasis buffer yang digerakkan oleh interupsi
8. Inisialisasi Pembanding Analog
9. Inisialisasi ADC
10. Inisialisasi Antarmuka SPI
11. Inisialisasi Antarmuka Two-Wire
12. Inisialisasi Antarmuka CAN
13. Inisialisasi Bus I2C, Sensor Suhu LM75, Thermometer/Thermostat DS1621 dan Real-Time Clock PCF8563, PCF8583, DS1302, dan DS1307
14. Inisialisasi Bus 1-Wire dan Sensor Suhu DS1820, DS18S20
15. Inisialisasi modul LCD

2.13.1 Membuat Project dengan *CodeVisionAVR*

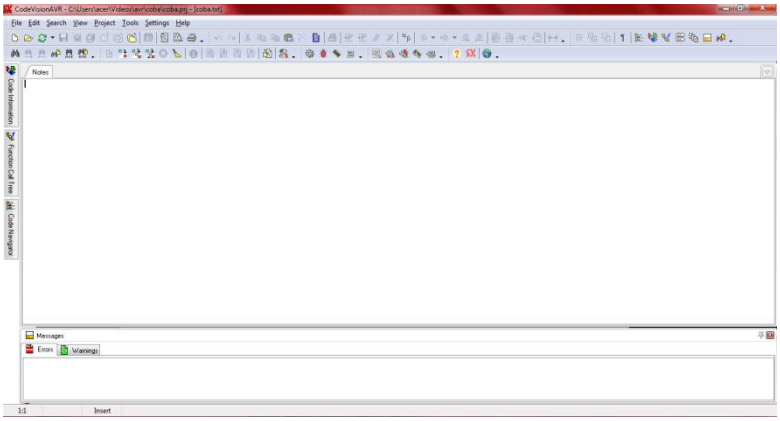
Jalankan aplikasi *CodeVisionAVR* dengan cara melakukan klik ganda pada *shortcut* ikon *CodeVisionAVR* yang terbentuk pada *Desktop*.

Sebuah *Splash Screen* akan muncul seperti ditunjukkan oleh Gambar 2.13 Informasi tentang versi yang dipakai dan keterangan *evaluation* akan terlihat.



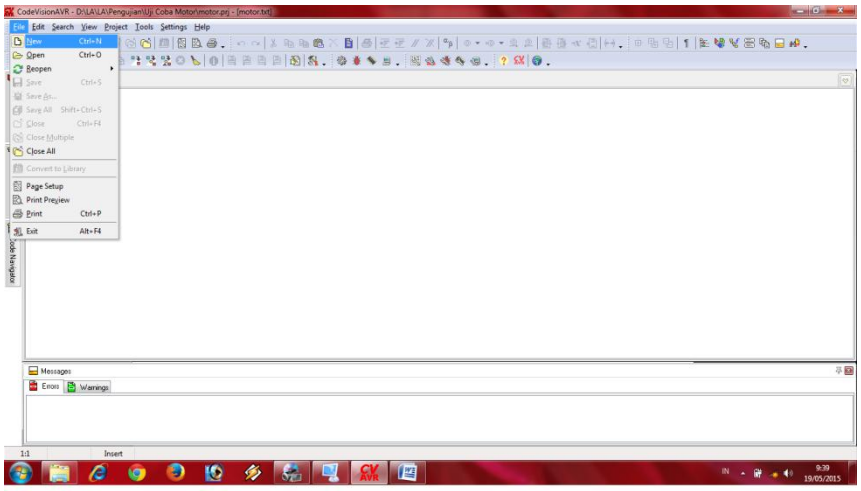
Gambar 2.13 Tampilan *Splash Screen*

Beberapa detik kemudian IDE dari *CodeVisionAVR* akan muncul seperti yang ditunjukkan oleh Gambar 2.14.



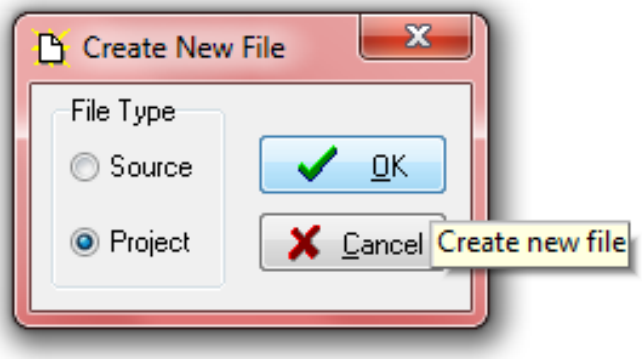
Gambar 2.14 IDE *CodeVisionAVR*

Untuk memulai membuat *project* baru, pada menubar, pilih *File* > *New*, seperti yang ditunjukkan oleh Gambar 2.15.



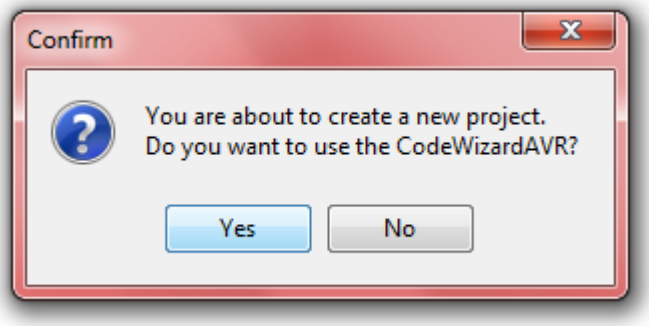
Gambar 2.15 Membuat *file* baru

Anda harus membuat sebuah project sebagai induk desain dengan memilih *Project*, lalu klik tombol OK seperti pada Gambar 2.16.



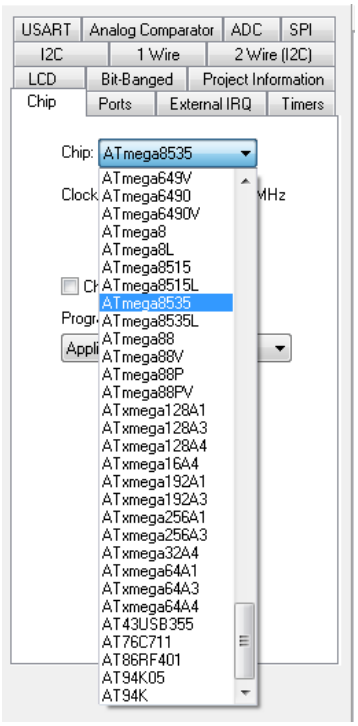
Gambar 2.16 Membuat *project* baru

Berikutnya Anda akan ditanya apakah akan menggunakan *CodeWizardAVR*. Tentu saja lebih menyenangkan bila Anda memilih jawaban "ya" dengan cara menekan tombol *Yes* seperti pada Gambar 2.17.



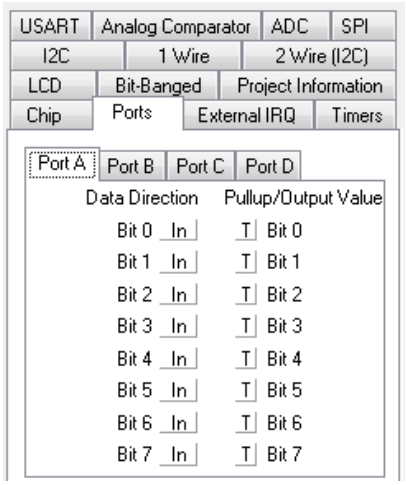
Gambar 2.17 Memilih untuk menggunakan *CodeWizardAVR*

Tampilan *CodeWizardAVR* yang sederhana namun lengkap ditunjukkan oleh Gambar 2.15 Pilih *Chip* dengan IC yang Anda gunakan. Sebagai contoh Anda memilih *Chip* ATmega8535. Tab-tab pada *CodeWizardAVR* menunjukkan fasilitas yang dimiliki oleh *chip* yang Anda pilih. Cocokkan pula frekuensi kristal yang Anda gunakan pada bagian *Clock*. Pengisian frekuensi *clock* digunakan oleh *software* untuk menghitung rutin-rutin seperti *delay* agar diperoleh perhitungan yang cukup akurat.

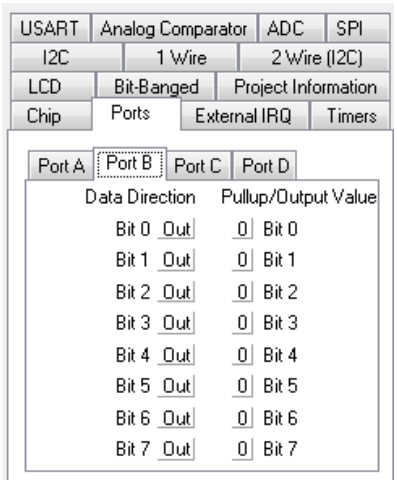


Gambar 2.18 *CodeWizardAVR* pada tab *Chip*

Berikutnya Anda akan menginisialisasi Port A tersambung dengan saklar sebagai modul input. Kemudian lakukan inisialisasi Port B seperti pada Gambar 2.19. Port B yang terhubung dengan LED. LED merupakan modul output.

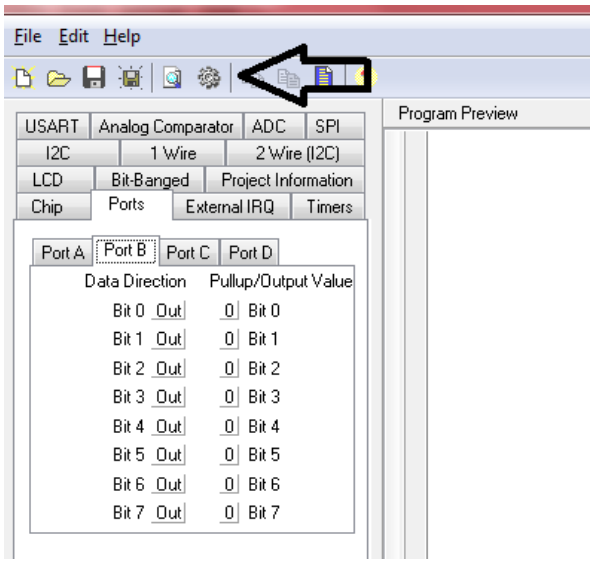


Gambar 2.19 Seting Port A sebagai pin input



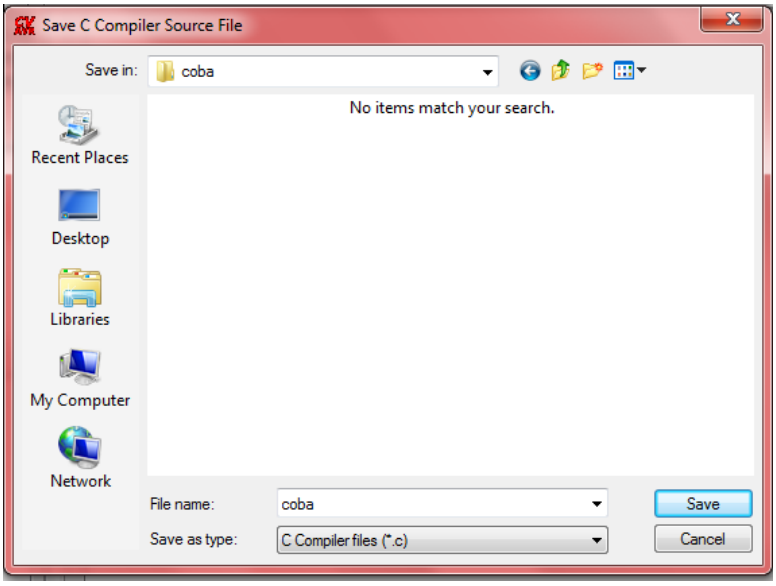
Gambar 2.20 Seting Port B sebagai pin output

Karena pada contoh ini tidak digunakan fasilitas lain maka seting *CodeWizardAVR* siap disimpan dalam file. Pada menu *CodeWizardAVR*, pilih *File* > *Generate, Save and Exit*, seperti yang ditunjukkan oleh Gambar 2.21.



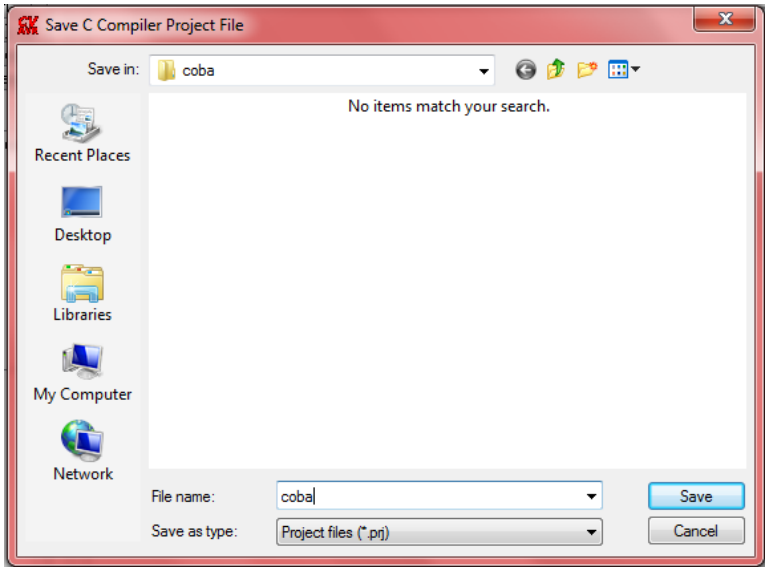
Gambar 2.21 Menyimpan *setting*

Agar file yang dihasilkan tidak berantakan, buatlah sebuah folder baru, misalnya folder bernama "coba". Kemudian masuk kedalam folder tersebut untuk menyimpan file-file yang dihasilkan oleh *CodeWizardAVR*. Yang pertama Anda diminta untuk memberikan nama file C yang dihasilkan. Misalnya beri nama "coba", lalu klik tombol *Save*. Lebih jelas pada Gambar 2.22 File tersebut nantinya akan mempunyai akhiran .C.



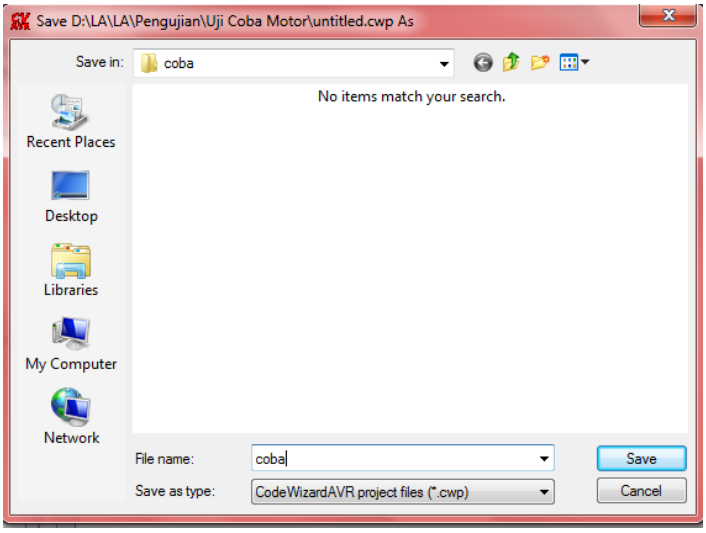
Gambar 2.22 Menyimpan *file* pertama

Yang kedua Anda diminta untuk memberikan nama file project yang dihasilkan. Misalnya beri nama “coba”, lalu klik tombol *Save*. Lebih jelas pada Gambar 2.23 *File* tersebut nantinya akan mempunyai akhiran .prj.



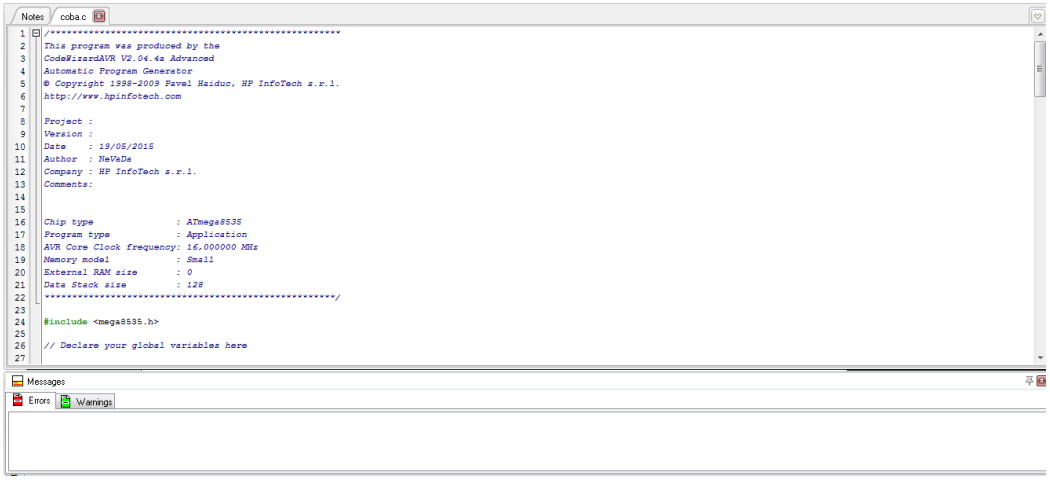
Gambar 2.23 Menyimpan *file* kedua

Yang terakhir Anda diminta untuk memberikan nama *file project CodeWizard* yang dihasilkan. Misalnya beri nama “coba”, lalu klik tombol *Save*. Lebih jelas pada Gambar 2.24 *File* tersebut nantinya akan mempunyai akhiran .cwp.



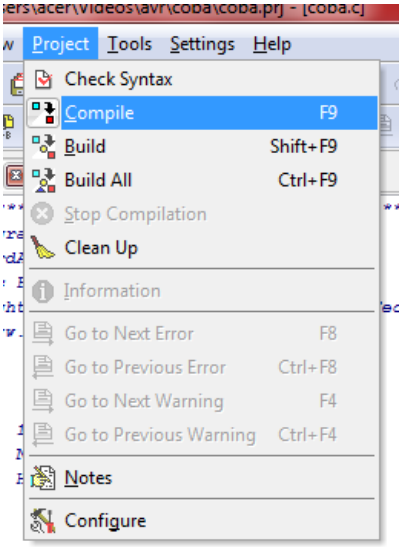
Gambar 2.24 Menyimpan *file* ketiga

Setelah ketiga *file* disimpan maka pada *Project Navigator* akan muncul nama *project* beserta *file* C-nya. Secara bersamaan isi file C akan dibuka pada jendela *editor* seperti ditunjukkan oleh Gambar 2.25.

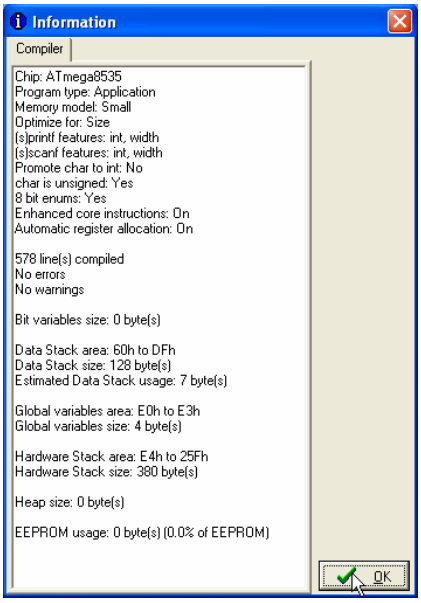


Gambar 2.25 *Project* baru telah siap dalam hitungan detik

Jika nanti saklar diaktifkan maka LED yang bersesuaian akan aktif pula. Kemudian pilih menu *Project* "Compile" untuk melakukan kompilasi seperti yang ditunjukkan oleh Gambar 2.26.



Gambar 2.26 Melakukan kompilasi



Gambar 2.27 Informasi hasil kompilasi