

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Untuk memulai perancangan alat pendeteksi kerusakan pada ic berbasis ATmega 8, maka perlu dilakukan pengumpulan data referensi berdasarkan penelitian – penelitian yang pernah dilakukan sebelumnya dengan tujuan agar didapatkan perbandingan kelebihan dan kekurangan masing – masing perancangan yang sudah ada. Berikut ini adalah tabel perbandingan yang berhasil dikumpulkan.

Tabel 2.1 Tabel Data Penelitian Pertama

1	Nama Peneliti	Aris Kurniawan
2	Judul	Rancang Bangun Perangkat Sistem Pengujian IC Digital
3	Intisari	Pada skripsi ini dilakukan perancangan program aplikasi pengujian IC digital yang bertujuan untuk mendeteksi keadaan / kondisi dari IC digital khususnya IC TTL dan IC CMOS. Sistem dibangun dengan menggunakan dua komponen utama yang terdiri dari perangkat lunak dan perangkat keras. Pada perangkat lunak menggunakan program Visual Basic sebagai alat bantu dalam program aplikasi pengujian IC. Pada perangkat keras menggunakan rangkaian Mikrokontroler AVR ATmega 32
4	Komponen	1. ATmega32 2. IC 7486 3. PC
5	Design	1. Berbasis PC 2. Menggunakan listrik PLN 3. IC yang di deteksi berupa IC digital khususnya IC TTL dan CMOS

Tabel 2.2 Tabel Data Penelitian Kedua

1	Nama Peneliti	Aida Agustin S
2	Judul	Pengetes IC TCA 785 Dengan Tampilan Digital
3	Intisari	<p>Pengetes IC TCA 785 dengan Tampilan Digital ini berfungsi untuk mengetahui kondisi IC TCA 785 dalam keadaan baik atau rusak. Pengetes IC TCA 785 dengan Tampilan Digital ini terdiri dari 6 (enam) bagian utama yaitu rangkaian utama, rangkaian stabilisasi, rangkaian ADC, rangkaian sistem minimum AT89C51, rangkaian tampilan dan rangkaian catu daya. Rangkaian Utama (Blok I) berfungsi untuk mendeteksi IC TCA 785 yang akan dites. Rangkaian Stabilisasi (Blok 2) menerima input dari rangkaian utama. Rangkaian Stabilisasi berfungsi untuk menyearahkan dan menstabilkan tegangan output dari rangkaian utama. Rangkaian ADC (Blok 3) menggunakan IC ADC 0804. Rangkaian ADC 0804 berfungsi mengkonversi sinyal analog menjadi sinyal digital yang diumpankan ke sistem minimum AT89C51. Rangkaian ADC 0804 mengubah hasil output analog dari rangkaian stabilisasi menjadi bentuk biner.</p>
4	Komponen	<p>4. Mikrokontroler AT89C51</p> <p>5. ADC 0804</p>
5	Design	<p>4. Output dengan tampilan digital menggunakan seven segment</p> <p>5. IC yang di deteksi berupa IC TCA 785</p>

Tabel 2.3 Tabel Data Penelitian Ketiga

1	Nama Peneliti	<ol style="list-style-type: none"> 1. Moh. Iqbal, ST 2. Endang Supriyati, S.Kom
2	Judul	Rancang Bangun Alat Pendeteksi Kerusakan IC Digital Gerbang Dengan Komputer
3	Intisari	<p>Dengan menggunakan metode penelitian, meliputi literatur, perancangan, pembuatan dan pengujian alat, akhirnya didapat hasil penelitian berupa alat pendeteksi kerusakan IC Digital beserta <i>software</i>-nya, yang telah diuji untuk beberapa tipe/seri IC Digital Gerbang dan dapat bekerja dengan baik sesuai harapan. IC-IC tersebut diantaranya adalah IC 7408 dan IC 7432. Meskipun demikian, alat ini juga bisa digunakan untuk IC-IC Digital Gerbang yang lain yang mempunyai kemasan 14 pin DIL (<i>dual in-line</i>).</p>
4	Komponen	<ol style="list-style-type: none"> 1. Mikrokontroler AT89S51 2. Menggunakan Delphi sebagai bahasa pemograman
5	Design	<ol style="list-style-type: none"> 1. Output dengan tampilan PC 2. IC yang di deteksi berupa IC 7432 dan IC 7408

Dari referensi tabel penelitian sebelumnya, maka didapatkan tabel perbandingan yang akan dijadikan rujukan perbandingan dengan alat yang akan dibuat, sesuai dengan spesifikasi pada tabel berikut ini.

Tabel 2.4 Tabel Data Penelitian Yang Akan Dibuat

1	Nama	Tedi Darmawan
2	Judul	Alat Pendeteksi Kerusakan Pada IC Berbasis Mikrokontroler ATmega 8
3	Intisari	Pada Laporan Akhir ini dibuat alat pendeteksi kerusakan pada ic berbasis mikrokontroler ATmega 8 yang bertujuan untuk mendeteksi keadaan / kondisi dari IC mikrokontroler khususnya mikrokontroler jenis AVR. Sistem dibangun dengan menggunakan dua komponen utama yang terdiri dari perangkat lunak dan perangkat keras. Pada perangkat lunak menggunakan program Code Vision AVR sebagai alat bantu dalam program pengujian IC. Pada perangkat keras menggunakan rangkaian Mikrokontroler ATmega 8 dengan penampil data menggunakan LCD.
4	Komponen	<ol style="list-style-type: none"> 1. ATmega8 2. IC 7805 dan 7812 3. LCD
5	Design	<ol style="list-style-type: none"> 1. Berbasis Mikrokontroler ATmega8 2. Menggunakan transformator sebagai power supply 3. IC yang di deteksi berupa IC Mikrokontroler khususnya IC Mikrokontroler jenis AVR

Dari ke semua tabel penelitian diatas, dapat diketahui perbedaan berdasarkan fungsi dan kelebihan masing – masing perancangan alat. Pada tabel jurnal yang dijadikan referensi (Tabel data penelitian pertama, tabel data penelitian kedua, tabel data penelitian ketiga) penggunaan alat digunakan untuk mengecek kondisi ic digital TTL dan CMOS dengan pengoperasian menggunakan PC sebagai interface. Alat ini bersifat *fix desktop*, karena pengoperasiannya menggunakan komputer sehingga tidak bisa digunakan secara portable. Adapun untuk perancangan yang akan dibuat. Secara fungsi, alat bekerja untuk mikrokontroler jenis AVR dengan *interface* melalui tampilan *LCD* 16 x 2 dan seperangkat tombol pengaturan. Alat ini dirancang dengan menggunakan sumber listrik dari listrik PLN yang dialirkan melalui *trasformator* (trafo).

2.2 Mikrokontroler AVR (*Alf and Vegaard's Risc Processor*)

Mikrokontroler jenis AVR adalah prosesor yang sekarang ini paling banyak digunakan dalam membuat aplikasi sistem kendali bidang instrumentasi, dibandingkan dengan mikrokontroler keluarga MCS51 seperti AT 89C51/52.

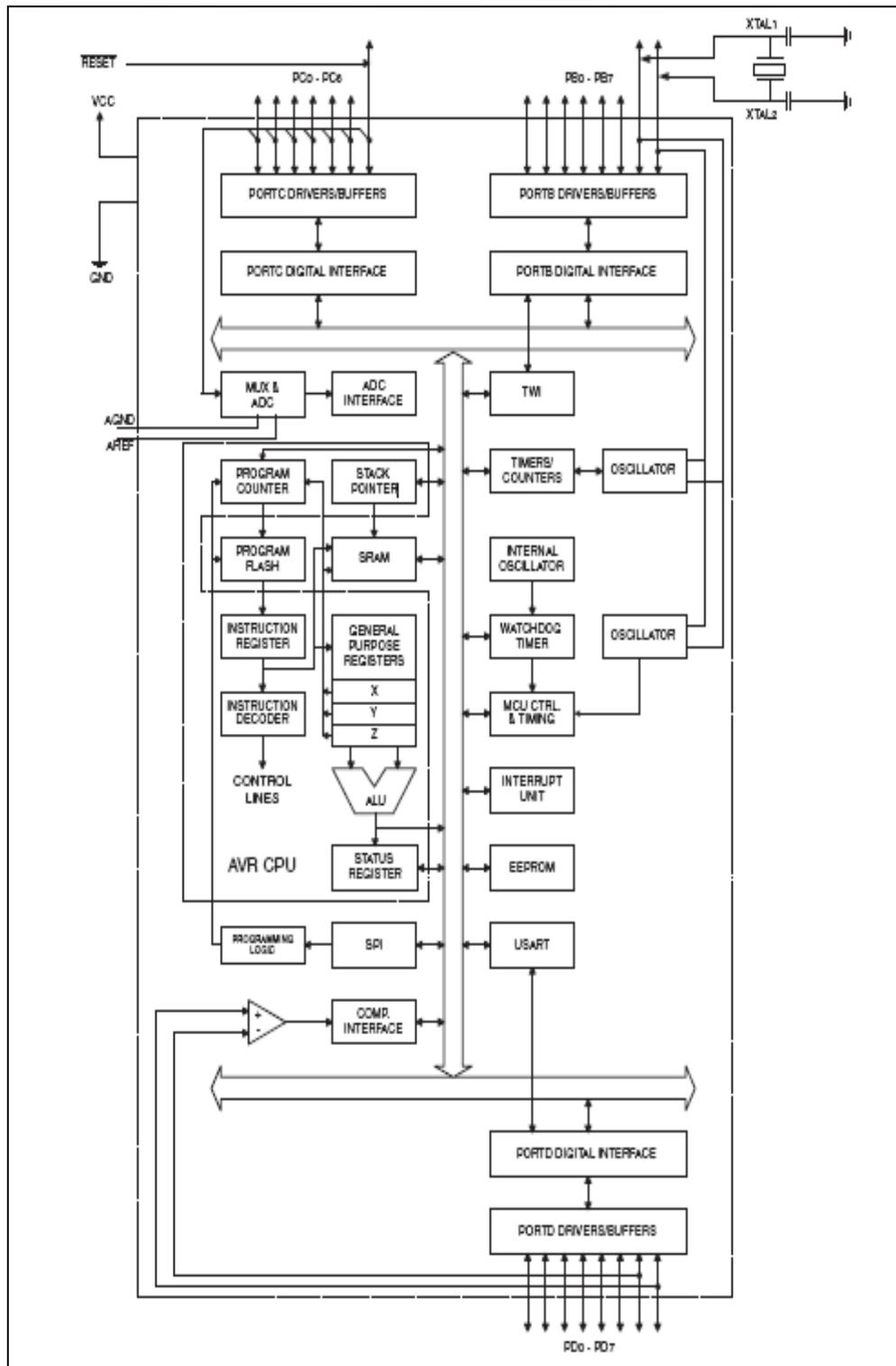
Mikrokontroler seri AVR pertama kali diperkenalkan ke pasaran sekitar tahun 1997 oleh perusahaan Atmel, yaitu sebuah perusahaan yang sangat terkenal dengan produk mikrokontroler seri AT89S51/52-nya yang sampai sekarang masih banyak digunakan di lapangan. Keterbatasan pada mikrokontroler tersebut (resolusi, memori, dan kecepatan) menyebabkan banyak orang beralih ke mikrokontroler AVR. Hal ini karena ada beberapa kelebihan dari tipe AVR ini yaitu diantaranya ADC, DAC, *Counter*, *Timer*, I2C, USART, dan sebagainya.

Mikrokontroler AVR standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit dan sebagian besar instruksi dieksekusi dalam satu situs *clock*, berbeda dengan instruksi MCS51 yang membutuhkan 12 situs *clock* (Widodo Budiharto dan Gamayel Rizal, 2007:28). Hal ini karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC(*Reduce Insruction Set Computing*), sedangkan seri MCS51 berteknologi CISC(*Complex Instruction Set Computing*). AVR dapat dikelompokkan menjadi empat kelas yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFFxx. Perbedaan dari masing - masing keluarga AVR tersebut adalah memori, peripheral, dan fungsinya. (Slamet dan Muhammad Munir, 2010 : 7)

2.2.1 Mikrokontroler ATmega 8

AVR ATmega8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16MHz. Jika dibandingkan dengan ATmega8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATmega8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk ATmega8 hanya dapat bekerja pada tegangan antara 4,5 – 5,5 V. (Wrespati, Putri Kartika Ratri)

2.2.2 Arsitektur Mikrokontroler ATmega8



Gambar 2.1 Blok Diagram Fungsional ATmega 8

2.2.3 Fitur ATmega8

Fitur – fitur yang terdapat pada mikrokontroler ATmega8 antara lain adalah sebagai berikut :

1. *High-Performance, Low-Power AVR 8-bit RISC microcontroller*
2. *Advanced RISC Architecture*
 - a. *130 Powerful Instructions – Most Single-clock Execution*
 - b. *32 x 8 General Purpose Working Registers*
 - c. *Fully Static Operation*
 - d. *Up to 16 MIPS Throughput at 16MHz*
 - e. *On-chip 2-cycle Multiplier*
3. *High-Endurance Non-Volatile Memory segments*
 - a. *8K Bytes In-System Self-programmable Flash Program Memory*
 - b. *512 Bytes EEPROM*
 - c. *1K Bytes of Internal SRAM*
 - d. *Write/Erase Cycles: 10,000 Flash / 100,000 EEPROM*
 - e. *Data Retention: 20 years at 85`C / 100 years at 25`C*
 - f. *Opitonal Boot Code Section with Independent Lock Bits*
 - g. *In-System Programming by On-chip Boot Program*
 - h. *True Read-While-Write Operation*
 - i. *Programming Lock for Software Security*
4. *Peripheral features*
 - a. *Two 8-bit Timers/Counters with Separate Prescaler, one Compare Mode*
 - b. *One 16-bit Timer/Counter with Separate Prescaler, Compare Mode,*

and Capture Mode

c. Real Time Counter with Separate Oscillator

d. Three PWM Channels

e. 6-channel ADC with 10-bit Accuracy

f. Byte-oriented Two-wire Serial Interface

g. Programmable Serial USART

h. Master/Slave SPI Serial Interface

i. Programmable Watchdog Timer with Separate On-Chip Oscillator

j. On-Chip Analog Comparator

5. Special Microcontroller features

a. Power-On Reset and Programmable Brown-out Detection

b. Internal Calibrated RC Oscillator

c. External and Internal Interrupt Sources

d. Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power down, and Standby

6. I/O and Packages

a. 23 Programmable I/O Lines

b. 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF

7. Operating Voltages

a. 2.7 – 5.5V (ATmega8L)

b. 4.5 – 5.5V (ATmega8)

8. Speed Grades

a. 0 – 8MHz (ATmega8L), 0 – 16MHz (ATmega8)

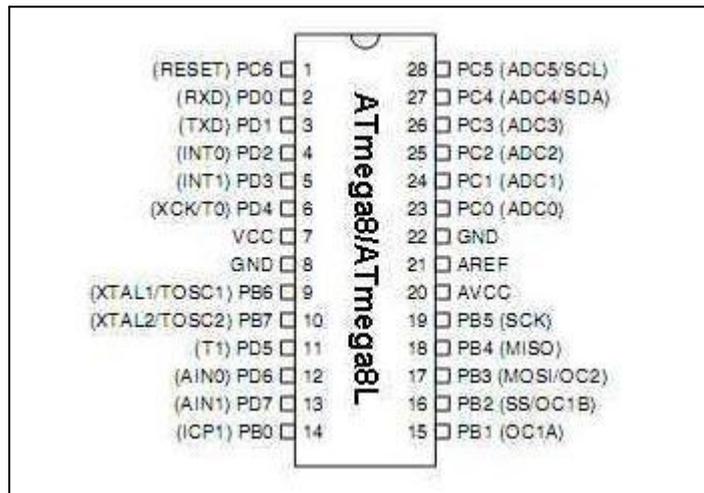
9. Power Consumption at 4MHz, 3V, 25°C

a. Active: 3.6 mA

b. *Idle Mode*: 1.0 mA

c. *Power-Down Mode*: 0.5 uA

2.2.4 Konfigurasi Pin ATmega8



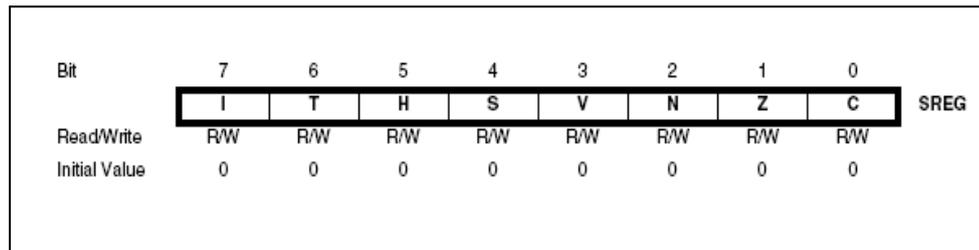
Gambar 2.2 Konfigurasi Pin ATmega 8

Berikut ini adalah susunan pin / kaki dari Mikrokontroler ATmega8.

1. VCC adalah merupakan pin masukan positif catu daya.
2. GND sebagai pin Ground.
3. PORT B (B.0-B.5) merupakan pin I/O dua arah dan pin fungsi khusus yaitu *Timer/Counter*, dan SPI.
4. PORT C (C.0-C.6) merupakan pin I/O dua arah dan dapat diprogram sebagai pin ADC.
5. PORT D (D.0-D.4) merupakan pin I/O dua arah dan pin fungsi khusus yaitu interupsi eksternal dan komunikasi serial.
6. Reset merupakan pin yang digunakan untuk me-reset mikrokontroler.
7. XTAL1 dan XTAL2 sebagai pin masukan *clock* eksternal. Suatu mikrokontroler membutuhkan sumber detak(*clock*) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi kristalnya, semakin cepat kerja mikrokontroler tersebut.
8. AVCC sebagai pin supply tegangan untuk ADC.
9. AREF sebagai pin masukan tegangan referensi untuk ADC.

2.2.5 Status Register

Status register adalah register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. SREG merupakan bagian dari inti CPU mikrokontroler. Berikut ini adalah status register dari ATmega8 beserta penjelasannya. (Slamet dan Muhammad Munir, 2010 : 11)



Gambar 2.3. Status Register ATmega8

1. Bit 7 (I)

Merupakan bit *Global Interrupt Enable*. Bit ini harus di-set supaya semua perintah interupsi dapat dijalankan. Untuk fungsi interupsi individual akan dijelaskan pada bagian lain. Jika bit ini di-set, maka semua perintah interupsi baik yang individual maupun secara umum akan diabaikan. Bit ini akan dibersihkan atau *cleared* oleh *hardware* setelah sebuah interupsi dijalankan dan akan di-set kembali oleh perintah RETI. Bit ini juga dapat di-set dan di-reset melalui aplikasi dengan instruksi SEI dan CLI.

2. BIT 6 (T)

Merupakan bit *Copy Storage*. Instruksi bit *Copy Instructions* BLD (Bit Load) dan BST (Bit Store) menggunakan bit ini sebagai asal atau tujuan untuk bit yang telah dioperasikan. Sebuah bit dari sebuah register dalam *Register File* dapat disalin ke dalam bit ini dengan menggunakan instruksi BST, dan sebuah bit di dalam bit ini dapat disalin ke dalam sebuah bit di dalam register pada *Register File* dengan menggunakan perintah BLD.

3. BIT 5 (H)

Merupakan bit *Half Carry Flag*. Bit ini menandakan sebuah *Half Carry* dalam beberapa operasi aritmatika. Bit ini berfungsi dalam aritmatik BCD.

4. BIT 4 (S)

Merupakan *Sign* bit. Bit ini selalu merupakan sebuah eksklusif diantara *Negative Flag (N)* dan *Two's Complement Overflow Flag(V)*.

5. BIT 3 (V)

Merupakan bit *Two's Complement Overflow Flag*. Bit ini menyediakan fungsi – fungsi aritmatika dua komplemen.

6. BIT 2 (N)

Merupakan bit *Negative Flag*. Bit ini mengindikasikan sebuah hasil negatif di dalam sebuah fungsi logika atau aritmatika.

7. BIT 1 (Z)

Merupakan bit *Zero Flag*. Bit ini mengindikasikan sebuah hasil nol “0” dalam sebuah fungsi arimatika atau logika.

8. BIT 0 (C)

Merupakan bit *Carry Flag*. Bit ini mengindikasikan sebuah *cary* atau sisa dalam sebuah fungsi aritmatika atau logika.

2.3 USBasp

USBasp adalah salah satu downloader yang support untuk atmel mikrokontroller dan yang pada penggunaannya memakai jalur komunikasi melalui “**port usb**”. Dahulu kala pada era tahun 2000 ke bawah(era 90'an) proses mendownload suatu program(firmware) ke mikrokontroller masih menggunakan / melalui “**port parallel(DB25)**” tepatnya yang biasa digunakan untuk mesin printer pada era tersebut. USBasp merupakan rangkaian USB yang digunakan untuk memprogram mikrokontroler AVR. Untuk memasukkan program dari pc/laptop ke suatu mikrokontroler harus menggunakan USBasp ini.

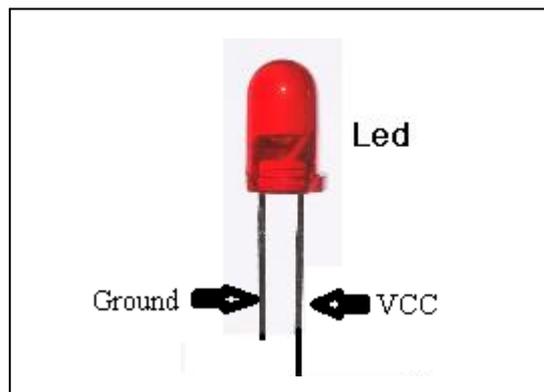


Gambar 2.4 USBasp

2.4 LED Indikator

Lampu LED atau kepanjangannya Light Emitting Diode adalah suatu lampu indikator dalam perangkat elektronika yang biasanya memiliki fungsi untuk menunjukkan status dari perangkat elektronika tersebut. Misalnya pada sebuah komputer, terdapat lampu LED power dan LED indikator untuk processor, atau dalam monitor terdapat juga lampu LED power dan power saving.

Lampu LED terbuat dari plastik dan dioda semikonduktor yang dapat menyala apabila dialiri tegangan listrik rendah (sekitar 1.5 volt DC). Bermacam-macam warna dan bentuk dari lampu LED, disesuaikan dengan kebutuhan dan fungsinya.

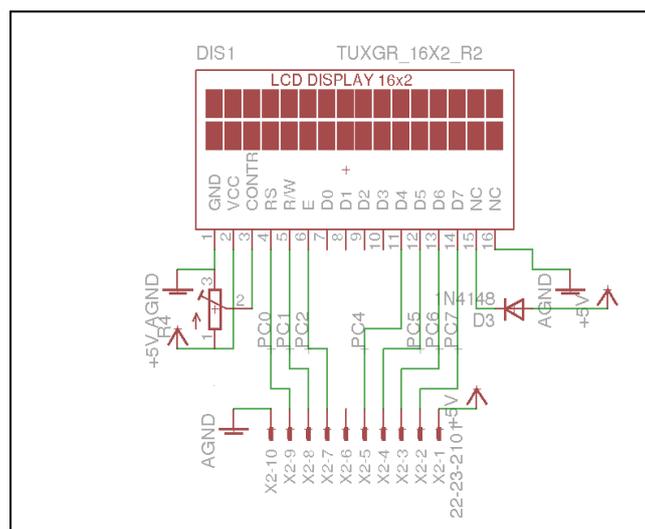


Gambar 2.5 LED

2.5 LCD (Liquid Crystal Display)

LCD merupakan singkatan dari *liquid crystal display*. LCD adalah satu bagian dari modul peraga yang menampilkan karakter yang diinginkan. Untuk dapat mengontrol tampilan ini diperlukan karakter generator yaitu bentuk-bentuk karakter yang dapat ditampilkan, urutan dan posisi dari karakter yang akan ditampilkan dan pergantian ke display harus disimpan dan digabungkan di RAM. Semua pengontrol tampilan ini telah dibentuk dalam satu IC.

LCD merupakan output yang akan menampilkan suatu tulisan, dimanadalam rangkaian LCD ini terdiri dari 14 kaki, kaki 1 adalah GND, sedangkan kaki 2 adalah VCC dan kaki 4, 6, 11, 12, 14 dihubungkan ke P1 pada mikrokontroler, sedangkan kaki 7, 8, 9, 10 diabaikan. Dikaki 2, 3 dipasang tripot 10 K Ω dimana tripot ini berfungsi untuk mengatur kontras display pada LCD (Wulandari, 2014 : 15). Gambar 2.12 dibawah ini merupakan gambar rangkaian LCD.



Gambar 2.6 Rangkaian LCD

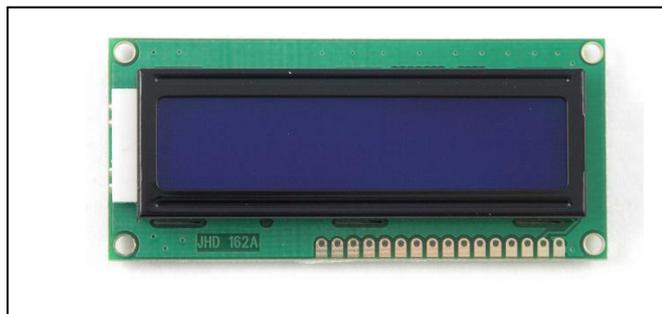
Kegunaan LCD banyak sekali dalam perancangan suatu sistem dengan menggunakan mikrokontroler. LCD (Liquid Crystal Display) dapat berfungsi untuk menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler. Pada praktek proyek ini, LCD yang

digunakan adalah LCD 16 x 2 yang artinya lebar display 2 baris 16 kolom dengan 16 Pin konektor.

Adapun konfigurasi dan deskripsi dari pin-pin LCD antara lain:

1. VCC (pin 1)
Merupakan sumber tegangan +5V
2. GND 0V (pin 2)
Merupakan sambungan ground
3. VEE (pin 3)
Merupakan input tegangan Kontras LCD
4. RS Register Select (pin 4)
Merupakan register pilihan 0 = Register Perintah, 1 = Register Data
5. R/W (pin 5)
Merupakan read select , 1 = Read, 0 = Write
6. Enable Clock LCD (pin 6)
Merupakan masukan logika 1 setiap kali pengiriman atau pembacaan data
7. D0 – D7 (pin 7 – pin 14)
Merupakan Data Bus 1 – 7 ke port
8. Anoda (pin 15)
Merupakan masukan Tegangan positif backlight
9. Katoda (pin 16)
Merupakan masukan Tegangan negatif backlight

Gambar LCD dapat di tunjukkan pada gambar berikut:



Gambar 2.7 LCD 16 x 2

Display karakter pada LCD diatur oleh pin EN, RS dan RW. Jalur EN dinamakan *Enable*. Jalur ini digunakan untuk memberitahu LCD bahwa sebuah data sedang dikirimkan. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika low “0” dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set EN dengan logika “1” dan tunggu dan berikutnya di set.

2.5.1 Modul LCD (*Liquid Crystal Display*) M1632

M1632 merupakan modul LCD matrix dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya dibentuk oleh 8 baris pixel dan 5 kolom pixel (1 baris pixel terakhir adalah kursor). HD44780 ini sudah tersedia dalam Modul M1632 yang dikeluarkan oleh Hitachi, Hyundai dan modul-modul M1632 lainnya.

HD44780 sebetulnya merupakan mikrokontroler yang dirancang khusus untuk mengendalikan LCD dan mempunyai kemampuan untuk mengatur proses *scanning* pada layar LCD yang terbentuk oleh 16 COM dan 40 SEG sehingga mikrokontroler/perangkat yang mengakses modul LCD ini tidak perlu lagi mengatur proses *scanning* pada layar LCD. Mikrokontroler ini hanya mengirimkan data-data yang merupakan karakter yang akan ditampilkan pada LCD atau perintah yang mengatur proses tampilan pada LCD saja.

2.5.2 Struktur Memori LCD

Modul LCD M1632 memiliki beberapa jenis memori yang digunakan untuk menyimpan atau memproses data-data yang akan ditampilkan pada layar LCD. Setiap jenis memori mempunyai fungsi-fungsi tersendiri.

a. DDRAM

DDRAM merupakan memori tempat karakter yang ditampilkan berada. Contohnya, karakter “A” atau 41h yang ditulis pada alamat 00 akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40h, karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.

b. CGRAM

CGRAM adalah memori untuk menggambarkan pola sebuah karakter dan bentuk karakter dapat diubah-ubah sesuai keinginan. Akan tetapi isi memori akan hilang saat power supply tidak aktif sehingga pola karakter akan hilang.

a. CGROM

CGROM adalah memori untuk menggambarkan pola sebuah karakter dan pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi. Oleh karena ROM bersifat permanen, pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif.

2.6 Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya, prinsip kerja buzzer hampir sama dengan loudspeaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik kedalam 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 biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm). (Aulia, Eko Nugroho, 2012 :10)



Gambar 2.8 Buzzer

2.7 Relay

Relay adalah saklar elektronik yang dapat membuka atau menutup rangkaian dengan menggunakan kontrol dari rangkaian elektronik lain. 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.

Berdasarkan pada prinsip dasar cara kerjanya, relay dapat bekerja karena adanya medan magnet yang digunakan untuk menggerakkan saklar. Saat kumparan diberikan tegangan sebesar tegangan kerja relay maka akan timbul medan magnet pada kumparan karena adanya arus yang mengalir pada lilitan kawat. Kumparan yang bersifat sebagai elektromagnet ini kemudian akan menarik saklar dari kontak NC ke kontak NO. Jika tegangan pada kumparan dimatikan maka medan magnet pada kumparan akan hilang sehingga pegas akan menarik saklar ke kontak NC.

Relay yang digunakan pada rangkaian ini memiliki spesifikasi SRU 12 VDC-SL-C. Jumlah pin pada relay ada 5 dan bertegangan kerja 12 VDC. Kemampuan arus yang dapat dilewatkan kontaktor adalah 10A pada tegangan 250VAC, 15A pada tegangan 120VAC, dan 10A pada tegangan 30VDC. (Slamet dan Muhammad Munir, 2010 : 22)



Gambar 2.9 Bentuk Fisik Relay SRU-12VDC-SL-C

2.8 Resistor

Resistor adalah komponen dasar elektronika yang digunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian (Slamet dan Muhammad Munir, 2010 : 24). Sesuai dengan namanya resistor bersifat resistif dan umumnya terbuat dari bahan karbon. Dari hukum Ohms diketahui, resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya. Satuan resistansi dari suatu resistor disebut Ohm atau dilambangkan dengan simbol Ω .

Tipe resistor yang umum adalah berbentuk tabung dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang kode warna untuk memudahkan pemakai mengenali besar resistansi tanpa mengukur besarnya dengan Ohmmeter. Kode warna tersebut adalah standar manufaktur yang dikeluarkan oleh EIA (*Electronic Industries Association*). (Slamet dan Muhammad Munir, 2010 : 7)

Tabel 2.5 Nilai- Nilai Gelang Warna Resistor

Warna	Nilai	faktor pengali	Toleransi
Hitam	0	1	
Coklat	1	10	1%
Merah	2	100	2%
Jingga	3	1.000	
Kuning	4	10.000	
Hijau	5	100.000	
Biru	6	10^6	
Violet	7	10^7	
Abu-abu	8	10^8	
Putih	9	10^9	
Emas	-	0.1	5%
Perak	-	0.01	10%
Tanpa warna	-	-	20%

2.9 Kapasitor

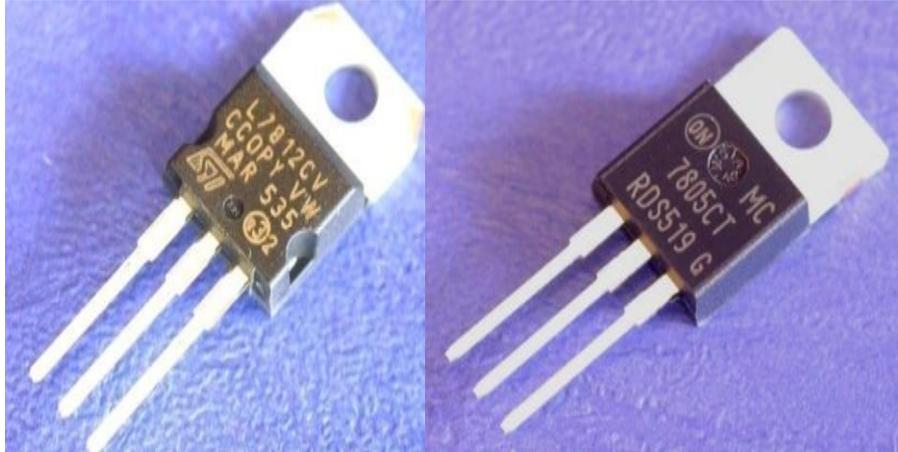
Kapasitor adalah komponen elektronika yang dapat menyimpan muatan listrik . Struktur sebuah kapasitor terbuat dari 2 buah plat metal yang dipisahkan oleh suatu bahan dielektrik. Bahan-bahan dielektrik yang umum dikenal misalnya udara vakum, keramik, gelas dan lain-lain. Jika kedua ujung plat metal diberi tegangan listrik, maka muatan-muatan positif akan mengumpul pada salah satu kaki (elektroda) metalnya dan pada saat yang sama muatan-muatan negatif terkumpul pada ujung metal yang satu lagi. Muatan positif tidak dapat mengalir menuju ujung kutup negatif dan sebaliknya muatan negatif tidak bisa menuju ke ujung kutup positif, karena terpisah oleh bahan dielektrik yang non-konduktif. Muatan elektrik ini "tersimpan" selama tidak ada konduksi pada ujung-ujung kakinya. Di alam bebas, fenomena kapasitor ini terjadi pada saat terkumpulnya muatan-muatan positif dan negatif di awan (Slamet dan Muhammad Munir, 2010 : 26).



Gambar 2.10 Kapasitor

2.10 IC Regulator

Untuk menstabilkan tegangan DC (+) dan tegangan DC (-) dari catu daya utama sebelum mensuplay rangkaian maka perlu digunakan regulator dengan memasang IC regulator tipe 78xx dan 79xx agar tegangan outputnya sesuai dengan kebutuhan rangkaian.



Gambar 2.11 IC Regulator

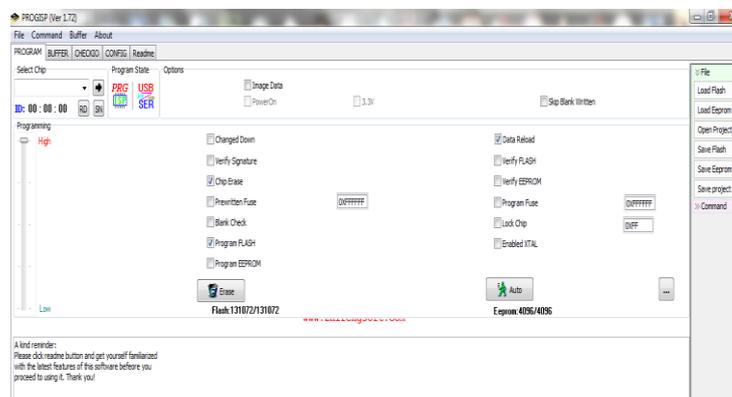
IC regulator memiliki berbagai macam jenis yang tergantung dari besar output keluarannya. Tidak semua nilai tegangan dapat diwujudkan dengan menggunakan IC regulator. Produsen IC regulator sudah menetapkan berbagai jenis IC regulator berdasarkan outputnya yang sampai sekarang ini banyak digunakan dalam rangkaian elektronik (Slamet dan Muhammad Munir, 2010 : 28). Berbagai tipe IC regulator beserta hasil keluaran outputnya dapat dilihat pada tabel 2.6.

Tabel 2.6 Tipe IC Regulator

No	Tipe	Output	Tipe	Output
1	L7805C	5V	L7905C	-5V
2	L7852C	5.2V	L7952C	-5.2V
3	L7806C	6V	L7906C	-6V
4	L7808C	8V	L7908C	-8V
5	L7809C	9V	L7909C	-9V
6	L7812C	12V	L7912C	-12V
7	L7815C	15V	L7915C	-15V
8	L7818C	18V	L7918C	-18V
9	L7820C	20V	L7920C	-20V
10	L7822C	22V	L7922C	-22V
11	L7824C	24V	L7924C	-24V

2.11 ProgISP

ProgISP merupakan program yang digunakan untuk menuliskan program ke dalam mikrokontroler ATmega8. Adapun caranya adalah sebagai berikut. Pertama mikrokontroler dihubungkan dengan kabel downloader dengan port paralel / port usb pada komputer. Lalu tekan tombol LoadFlash untuk membuka file Hex yang akan didownload pada mikrokontroler. Setelah itu tekan tombol auto tunggu sampai tulisan selesai, maka program sudah ditulis pada mikrokontroler, dan mikrokontroler siap untuk digunakan. Berikut adalah gambar tampilan ProgISP.



Gambar 2.12 Tampilan Software ISP

2.12 Pengenalan *Software* (Perangkat Lunak)

Perangkat lunak (*Software*) adalah program yang digunakan untuk menterjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman ke dalam bahasa mesin supaya dapat dimengerti oleh computer (Wulandari, 2014 : 31). *Software* adalah penghubung antara manusia dengan perangkat keras komputer, berfungsi menterjemahkan 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.

2.12.1 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 (Wulandari, 2014 : 31).

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, mengkompilernya, 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 assemblernya, contoh AVRStudio.

- 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 USART.

2.12.2 Menjalankan CodeVisionAVR

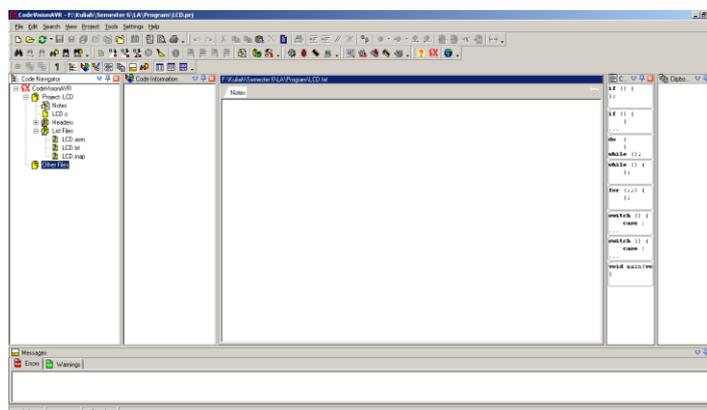
Ada beberapa program yang dapat digunakan sebagai *editor* dan *compiler* untuk mikrokontroler AVR, salah satunya yaitu *CodeVisionAVR*. Untuk menjalankan program tersebut terlebih dahulu menginstal program tersebut.

Setelah terinstal maka buka program *CodeVision* melalui menu *Start||All Program||CodeVision| CodeVisionAVR C Compiler* atau melalui desktop dengan mengklik lambang *codevision*. Gambar 2.13 dibawah ini adalah gambar Ikon *CodeVisionAVR*:



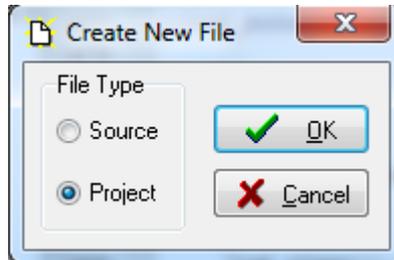
Gambar 2.13 Ikon *CodeVisionAVR*

Berikut adalah tampilan pertama kali *codevision* dijalankan.



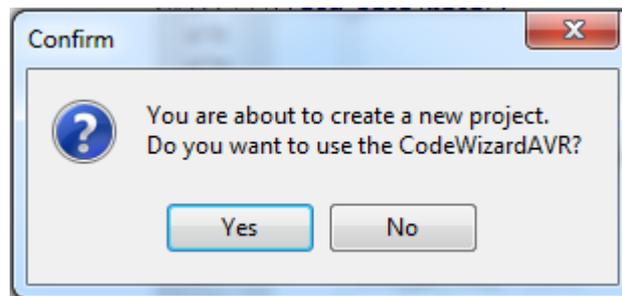
Gambar 2.14 Tampilan Pertama Kali *Codevision* Dijalankan

Lalu klik *File|New|* pilih *File Type* dan klik *Project* lalu klik OK. Berikut adalah gambar yang muncul jika ingin membuat file project baru.



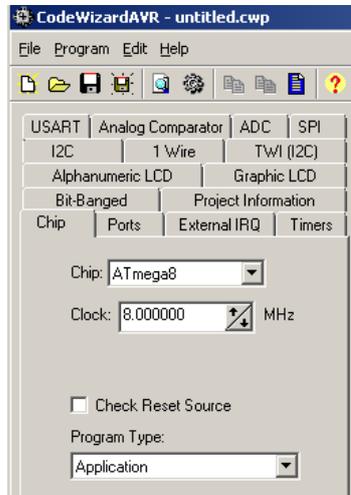
Gambar 2.15 Membuat *File Project* Baru

erikut adalah tampilan project baru menggunakan *Code Wizard AVR*.



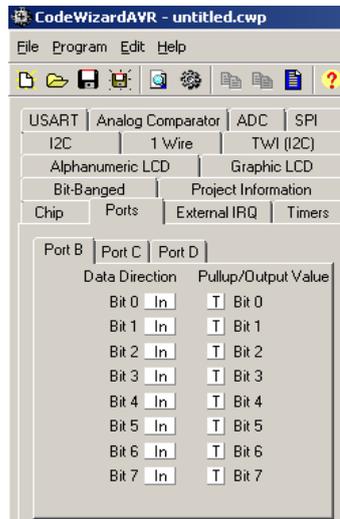
Gambar 2.16 *Project* Baru Menggunakan *CodeWizardAVR*

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: ATmega8*, *clock: 8.000000 MHz*. Gambar dibawah ini merupakan tampilan Pengaturan *chip* pada *CodeVisionAVR*.



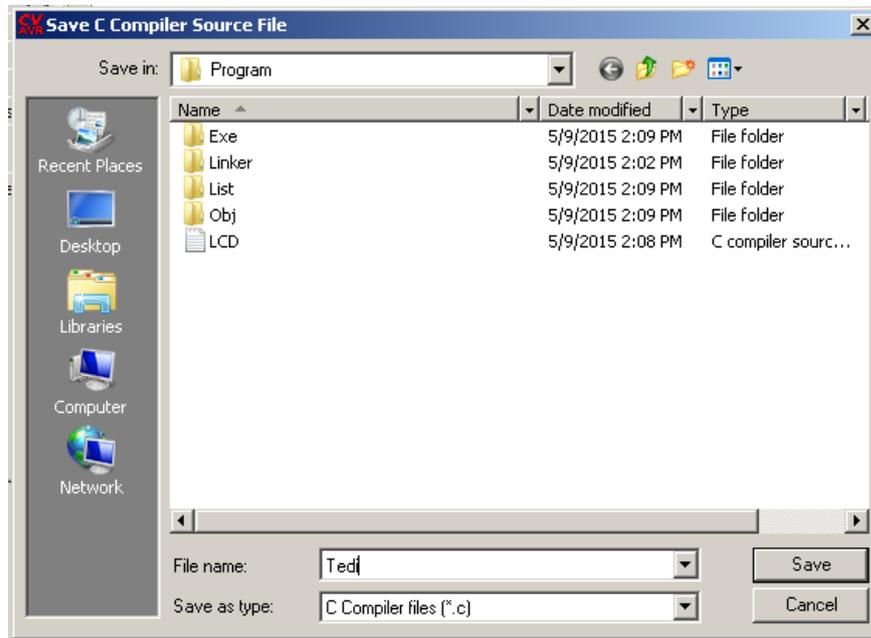
Gambar 2.17 Pengaturan *Chip* Pada *CodeVisionAVR*

Kemudian klik *PORT* untuk untuk memilih dan mengatur *PORT* yang akan digunakan apakah sebagai *In* atau *Out*, atau memilih *PORTB*, *PORTC*, *PORTD*. Gambar dibawah adalah tampilan pengaturan *port* pada *CodeVisionAVR*.



Gambar 2.18 Pengaturan *Port* Pada *CodeVisionAVR*

Kemudian beri nama *file* pertama lalu simpan. *File* ini *bertype* *C Compiler(*.C)*. berikut ini adalah tampilan penyimpanan file C.

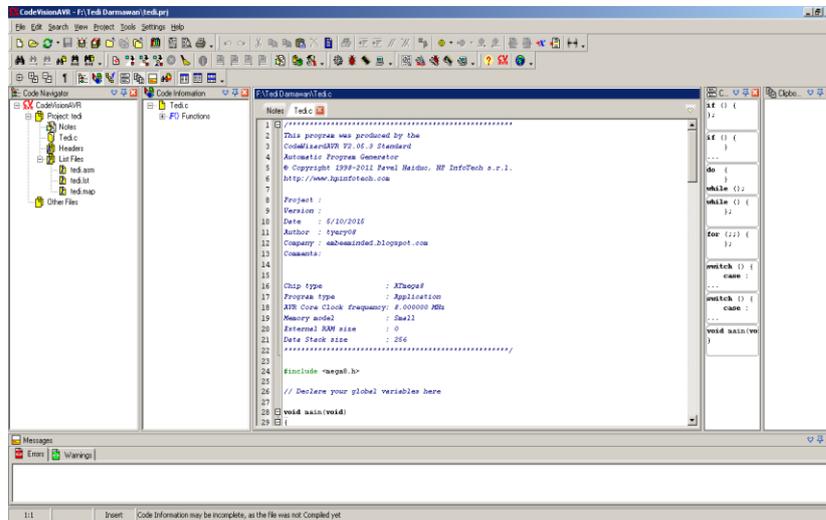


Gambar 2.19 Penyimpanan *File.C*

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

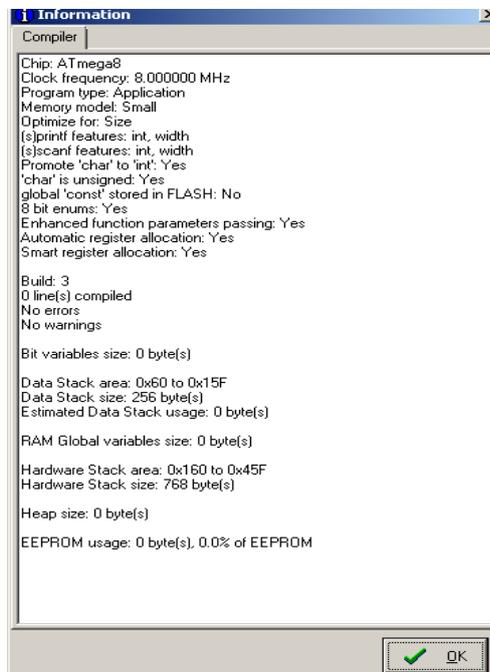
Terakhir beri nama *file* lagi yang ketiga kemudian simpan. *File* ini *bertype* *CodeWizardAVR Project file (*.cwp)*. Catatan : Usahakan nama Ketiga *File* sama agar lebih mudah mencari *file* tersebut.

Gambar dibawah ini adalah tampilan awal setelah menggunakan *Code Wizard*



Gambar 2.20 Tampilan Awal Pada Saat Menggunakan *Code Wizard*

Jika sudah selesai membuat program, maka *compile* program, pilih *Project* klik *compile*. Gambar dibawah ini adalah hasil proses kompilasi



Gambar 2.21 Hasil Proses Kompilasi

2.12.3 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 (Wulandari, 2014 : 39).

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.

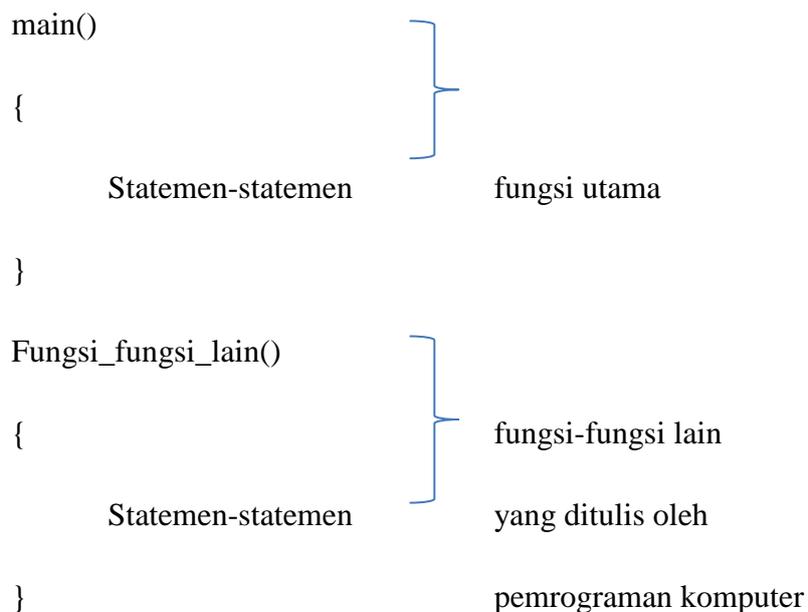
Dalam pembuatan program yang menggunakan fungsi atau aritmatika, Bahasa C menawarkan kemudahan dengan menyediakan fungsi – fungsi khusus, seperti: pembuatan konstanta, operator aritmatika, operator logika, operator *bitwise* dan operator *Assignment*. Selain itu, bahasa C menyediakan Program kontrol 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. *Preprocessor Directive (#include)* adalah bagian yang berisi pengikut sertaan 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 :



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*.

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 dibawah ini.

Tabel 2.7 Tipe data

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

2.13 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 gandengan 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 (Wulandari, 2014 : 18).

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 dibawah ini adalah gambar dari transformator:



Gambar 2.22 *Transformator*

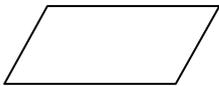
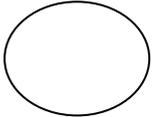
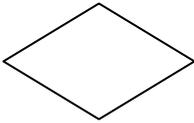
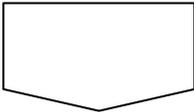
2.13 Flowchart

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 antar proses digambarkan dengan garis penghubung.

Flowchart ini merupakan langkah awal pembuatan program. Dengan adanya flowchart, urutan proses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah flowchart selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

Flowchart disusun dengan simbol – simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program.

Tabel 2.8 Flowchart

No	Simbol	Keterangan
1		Menunjukkan sebuah proses pengerjaan langkah program
2		Menunjukkan input yang dibutuhkan atau output yang dihasilkan oleh program
3		Memberikan keterangan pada alur program
4		Perulangan (<i>looping</i>)
5		Awal atau akhir program
6		Penghubung pada satu halaman
7		Pengujian pada suatu kondisi tertentu atau disebut juga dengan percabangan
8		Penghubung pada halaman lain
9		Arah alur Program