

BAB II

TINJAUAN PUSTAKA

2.1. Radio

Radio adalah teknologi yang digunakan untuk pengiriman sinyal dengan cara modulasi dan radiasi elektro magnetik (gelombang elektromagnetik). Gelombang ini melintas dan merambat lewat udara dan bisa juga merambat lewat ruang yang hampa udara (Daryanto, 2001).

Gelombang radio adalah satu bentuk dari radiasi elektromagnetik, dan terbentuk ketika objek bermuatan listrik dari gelombang osilator (gelombang pembawa) dimodulasi dengan gelombang audio (ditumpangkan frekuensinya) pada frekuensi yang terdapat dalam frekuensi gelombang radio (RF) pada satu spektrum elektromagnetik, dan radiasi elektromagnetiknya bergerak dengan cara osilasi elektrik maupun magnetik. Gelombang elektromagnetik lain yang memiliki frekuensi diatas gelombang radio meliputi sinar gamma, sinar-X, inframerah, ultraviolet, dan cahaya terlihat. Ketika gelombang radio dikirim melalui kabel kemudian dipancarkan oleh antenna, osilasi dari medan listrik dan magnetik tersebut dinyatakan dalam bentuk arus bolak-balik dan voltase di dalam kabel. Dari pancaran gelombang radio ini kemudian dapat diubah oleh radio penerima (pesawat radio) menjadi signal audio atau lainnya yang membawa siaran dan informasi.

2.1.1 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.

Dalam sistem ini, sinyal radio tidak terus menerus dipancarkan tapi hanya dibangkitkan saat pengontrol mengirimkan perintah atas/bawah, itupun hanya merupakan frekuensi radio yang terputus-putus, sehingga merupakan pengiriman pulsa-pulsa frekuensi gelombang radio (Daryanto, 2001).

2.1.2 Pemancar Gelombang Radio Jenis PT2262

Pada dasarnya prinsip dari pemancar adalah menggabungkan sinyal informasi dengan sinyal pembawa, dimana sinyal informasi yang berupa pulsa-pulsa diperkuat dan difilterisasi untuk membatasi lebar pita, kemudian osilator menentukan frekuensi pembawa. Karena kestabilan frekuensi yang baik diperlukan dalam pemancaran, sehingga diperlukan osilator yang berfrekuensi tetap atau stabil.

Untuk menambahkan daya pada gelombang pembawa diperlukan sebuah penguat, dimana penguat tersebut dapat berfungsi untuk meningkatkan daya sinyal pembawa dari osilator yang diperlukan untuk masuk ke modulator. Pada modulator menggabungkan sinyal informasi dengan sinyal pembawa, frekuensi dari sinyal pembawa akan menghasilkan salah satu jenis gelombang termodulasi.

2.1.3 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 (Daryanto, 2001).

2.1.4 Penerima Gelombang Radio Jenis SC2272

Sebuah receiver akan menerima getaran-getaran elektromagnetik yang dipancarkan oleh suatu pemancar. Walaupun di wilayah itu terdapat bermacam-macam gelombang radio yang masuk ke antena penerima, namun pesawat penerima dapat memisahkan gelombang tersebut dengan menggunakan gelombang penala.

Umumnya dalam suatu penerima untuk keperluan tertentu, batas dari frekuensi yang akan diterima oleh pesawat ini telah ditetapkan, sehingga hanya dapat menerima jenis gelombang tertentu saja.

2.2 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 (Romy Budhi Widodo, 2009).

2.2.1 Mikrokontroler ATmega8

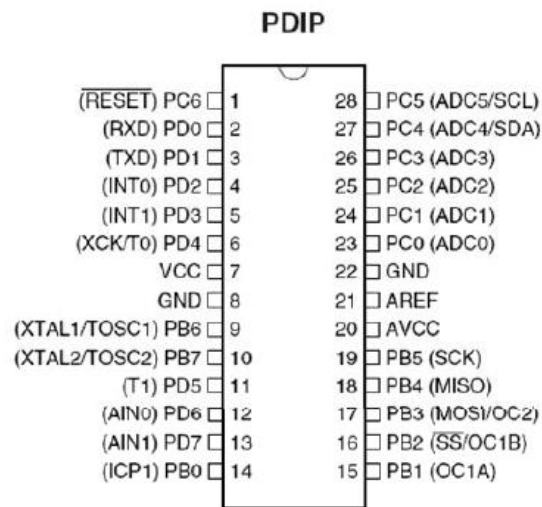
AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan *oscillator* eksternal karena di dalamnya sudah terdapat internal oscillator. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu ada tombol reset dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *byte* sampai dengan 512 *byte*.

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.



Gambar 2.1 Bentuk Fisik Mikrokontroler ATmega8

2.2.2 Konfigurasi Pin ATmega8



Gambar 2.2 Konfigurasi Pin ATmega8

Deskripsi port pada mikrokontroler ATmega8 :

1. VCC merupakan *supply* tegangan digital.
2. GND Merupakan ground untuk semua komponen yang membutuhkan grounding.
3. Port B (PB7...PB0)

Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai *input* maupun *output*. Port B merupakan sebuah 8-bit *bi-directional I/O* dengan internal pull-up resistor. Sebagai *input*, pin-pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor diaktifkan. Khusus PB6 dapat digunakan sebagai *input* Kristal (*inverting oscillator amplifier*) dan *input* ke rangkaian *clock* internal, bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Sedangkan untuk PB7 dapat digunakan sebagai *output* Kristal (*output oscillator amplifier*) bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Jika sumber *clock* yang dipilih dari *oscillator internal*, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *Asynchronous Timer/Counter2*

maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input timer*.

4. Port C (PC5...PC0)

Port C merupakan sebuah *7-bit bi-directional I/O port* yang di dalam masing-masing pin terdapat *pull-up* resistor. Jumlah pin nya hanya 7 buah mulai dari *pin C.0* sampai dengan *pin C.6*. Sebagai keluaran/output port C memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) ataupun mengeluarkan arus (*source*).

5. RESET/PC6

Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin I/O*. *Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port C* lainnya. Namun jika RSTDISBL *Fuse* tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika *level* tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak bekerja.

6. Port D (PD7...PD0)

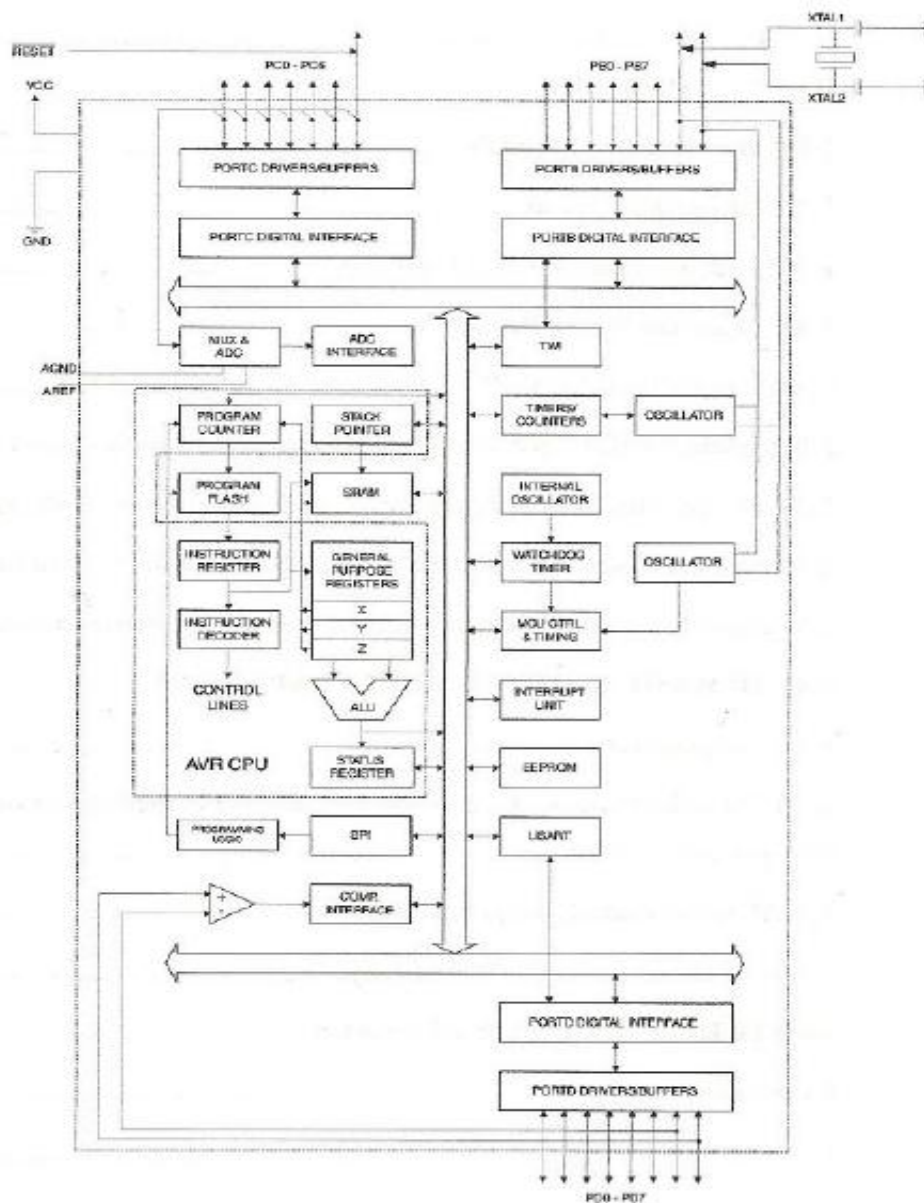
Port D merupakan *8-bit bi-directional I/O* dengan internal *pull-up* resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan *I/O*.

7. AVcc

Pin ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui *low pass filter*.

8. AREF

Merupakan pin referensi jika menggunakan ADC.



Gambar 2.3 Diagram Blok Mikrokontroler ATmega8

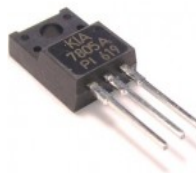
2.3 Baterai

Baterai adalah alat elektro kimia yang dibuat untuk mensuplai listrik ke sistem starter mesin, sistem pengapian, lampu-lampu dan komponen kelistrikan lainnya. Alat ini menyimpan listrik dalam bentuk energi kimia yang dikeluarkannya bila diperlukan dan mensuplainya ke masing-masing sistem kelistrikan atau alat yang memerlukannya. Karena di dalam proses baterai

kehilangan energi kimia, maka alternator mensuplainya kembali ke dalam baterai (yang disebut pengisian). Baterai menyimpan listrik dalam bentuk energi kimia. Siklus pengisian dan pengeluaran ini terjadi berulang kali dan terus menerus (Daryanto, 2001).

2.4 IC Regulator

IC regulator atau yang sering disebut sebagai regulator tegangan (*voltage regulator*) merupakan suatu komponen elektronik yang melakukan suatu fungsi yang penting dan berguna dalam perangkat elektronik baik digital maupun analog. Hal yang dilakukan oleh IC regulator ini adalah menstabilkan tegangan yang melewati IC tersebut. Setiap IC regulator mempunyai rating tegangannya sendiri-sendiri. Salah satunya IC regulator dengan nomor seri 7805 merupakan regulator tegangan sebesar 5 volt, yang artinya selama tegangan masukan lebih besar dari tegangan keluaran maka akan dikeluarkan tegangan sebesar 5 volt (Marethania, 2011). IC regulator 7805 ini mempunyai 3 buah kaki, yaitu kaki tegangan masukan yang biasa sering disebut V_{in} , kaki *ground* (0V) dan yang ketiga adalah kaki tegangan keluaran atau V_{out} (Surjati, 2008).



Gambar 2.4 IC Regulator 7805

2.5 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 komputer (Jogiyanto : 1999). *Software* sebagai penghubung antara manusia sebagai pengguna 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. Perangkat lunak menurut kategorinya digolongkan sebagai berikut :

1. Sistem Operasi (*Operating System*)

Sistem Operasi yaitu program yang berfungsi untuk mengendalikan sistem kerja yang mendasar sehingga mengatur kerja media input, output, tabel pengkodean, memori, penjadwalan prosesor, dan lain-lain. Sistem operasi berfungsi sebagai penghubung antara manusia dengan perangkat keras dan perangkat lunak yang akan digunakan. Contoh Sistem Operasi, misalnya : *Disk Operating System (DOS)*, *Microsoft Windows*, *Linux*, dan *Unix*.

2. Program Aplikasi (*Application Programs*)

Program Aplikasi adalah perangkat lunak yang dirancang khusus untuk kebutuhan tertentu, misalnya program pengolah kata, mengelola lembar kerja, program presentasi, design grafis, dan lain-lain. Atau dengan kata lain program yang di desain untuk membantu pengguna supaya dapat lebih produktif atau membantu dengan tugas- tugas khusus.

Beberapa perangkat lunak aplikasi dasar diantaranya adalah *browser*, pengolah kata seperti *Microsoft Word* , pengolah grafik, *game*, sistem manajemen, dll.

3. Bahasa Pemrograman (*Language Software*)

Perangkat lunak bahasa yaitu program yang digunakan untuk menerjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman ke bahasa mesin dengan aturan atau prosedur tertentu, agar diterima oleh komputer. Contoh *software* ini adalah : *Visual Basic*, *Delphi*, *Visual C++*.

4. Program Bantu (*Utility*)

Perangkat Lunak merupakan perangkat lunak yang berfungsi sebagai aplikasi pembantu dalam kegiatan yang ada hubungannya dengan komputer, misalnya memformat disket, mengopi data, mengompres file, dan lain-lain. Seperti *ScanDisk*, *PC Tools* dan *Norton Utility*.

Fungsi Perangkat Lunak antara lain :

1. Mengidentifikasi program
2. Menyiapkan aplikasi program sehingga tata kerja seluruh perangkat komputer terkontrol.

3. Mengatur dan membuat pekerjaan lebih efisien.
4. Mengatur input dan output dari komputer.
5. Menyediakan dan mengatur serta memerintah *hardware* agar dapat berjalan dengan baik.
6. Menjalankan perintah tertentu pada sebuah sistem komputer.
7. Mengatur dan membuat pekerjaan lebih efisien dan maksimal.

2.6 Bahasa Pemrograman Interface

2.6.1 Pengenalan Bahasa C

C merupakan bahasa universal dalam bidang pengembangan *software* dan banyak digunakan pada mesin-mesin dan komputer, banyak sekali *software* sistem yang dibuat dengan C karena bahasa C memiliki kemampuan untuk mengakses sistem dari komputer, mulai dari RAM yang sederhana, disk bahkan sampai yang sangat detail dan dalam seperti register dan port-port pada komputer, baik itu PC maupun *mini computer* dan *Mainframe*. Menulis Bahasa Pemrograman C tidaklah sesulit yang anda bayangkan. Saya asumsikan anda menggunakan sistem operasi *linux/unix* atau mungkin *cygwin* di *windows*, di mana sudah terinstall *compiler* yang tersedia untuk Bahasa Pemrograman C. Ada begitu banyak kompiller untuk bahasa c ini di berbagai platform seperti GCC, CC di *linux/unix*; *Miracle C*, *Turbo C*, *Microsoft Visual C++* di *linux/unix*. Kompiler yang akan kita pergunakan kali ini adalah GCC. Selanjutnya anda juga perlu mempersiapkan *text editor* apa saja. Bisa notepad jika anda menggunakan *windows* atau *vi*, *pico*, *vim*, *kwort*, *nano*, *gedit*, *emacs* dan lain sebagainya (Jogiyanto, 1992).

2.6.2 Penulisan Program Bahasa C

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.

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 statemen-statemen program C. Berikut ini adalah struktur dari program C.

```

main()
{
    Statemen-statemen
}

Fungsi_fungsi_lain()
{
    Statemen-statemen
}

```

fungsi utama

fungsi-fungsi lain
yang ditulis oleh
pemrograman komputer

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)

2.7 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam

pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman. (Adelia, Sistem Informasi : Vol.6, Tahun 2011)

Ada beberapa macam *flowchart* diantaranya :

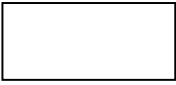
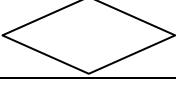
Sistem *flowchart* dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

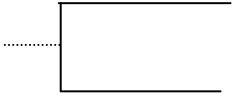
1. Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya.
2. Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarinya.
3. Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem.
4. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terinci (*detailed computer program flowchart*). Bagan alir logika

program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alat- logika program ini dipersiapkan oleh analis sistem. Gambar berikut menunjukkan bagan alir logika program. Bagan alir program komputer terinci (*detailed computer program flow-chart*) digunakan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Bagan alir ini dipersiapkan oleh pemrogram.

5. Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Tabel 2.1 Simbol-simbol *Flowchart*

Bagan	Nama	Fungsi
	<i>Terminator</i>	Awal atau akhir program
	<i>Flow</i>	Arah aliran program
	<i>Preparation</i>	inisialisasi/pemberian nilai awal
	<i>Process</i>	Proses/pengolahan data
	<i>Input/Output Data</i>	input/output data
	Sub Program	sub program
	<i>Decition</i>	Seleksi atau kondisi
	<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang sama

	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang berbeda
	<i>Comment</i>	Tempat komentar tentang suatu proses

2.8 Code Vision AVR

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, 2013).

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, *Compiler C* yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan *library* fungsi standar berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, *compiler C* untuk mikrokontroler ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (*embedded*). Khusus untuk *library* fungsi, disamping *library* standar (seperti fungsi- fungsi matematik, manipulasi *string*, pengaksesan memori dan sebagainya), *CodeVisionAVR* juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi *library* yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, IC RTC (*Real time Clock*), sensor suhu, SPI (*Serial Peripheral Interface*) dan lain sebagainya. Untuk memudahkan pengembangan program aplikasi, *CodeVisionAVR* juga dilengkapi IDE yang sangat *user friendly*. Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, *CodeVisionAVR* ini telah mengintegrasikan perangkat lunak *downloader* yang bersifat *In System*

Programmer yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi ke dalam sistem memori mikrokontroler AVR yang sedang diprogram (Widodo, 2013).

CodeVisionAVR juga menyediakan sebuah fitur yang dinamakan dengan *Code Generator* atau *CodeWizardAVR*. Secara praktis, fitur ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi *programmer* dalam peng-inisialisasian register-register yang terdapat pada mikrokontroler AVR yang sedang diprogram. Dinamakan *Code Generator*, karena perangkat lunak CodeVision ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela *CodeWizardAVR* selesai dilakukan (Widodo, 2013).

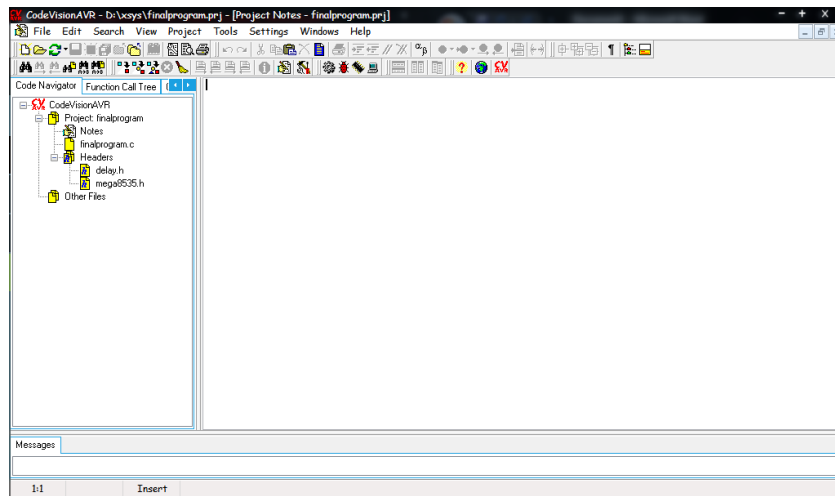
2.9 Menjalankan *CodeVisionAVR*

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

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

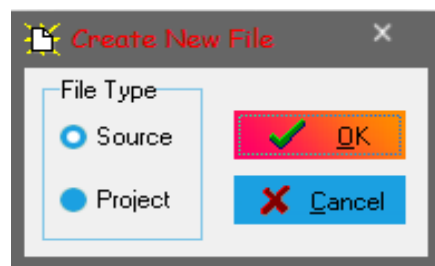


Gambar 2.5 Ikon CodeVisionAVR



Gambar 2.6 Tampilan pertama kali codevision dijalankan

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

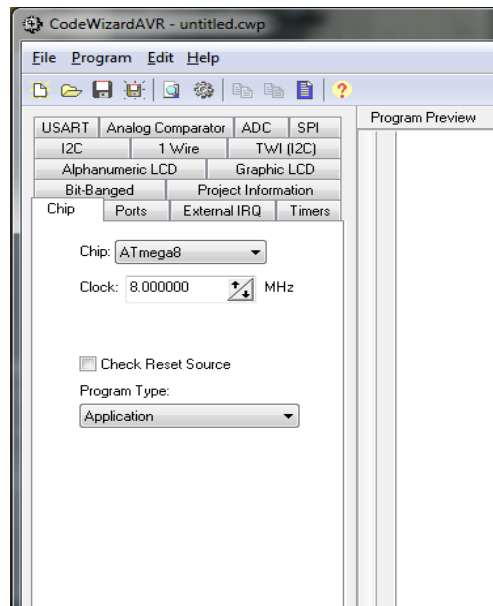


Gambar 2.7 Membuat file project baru



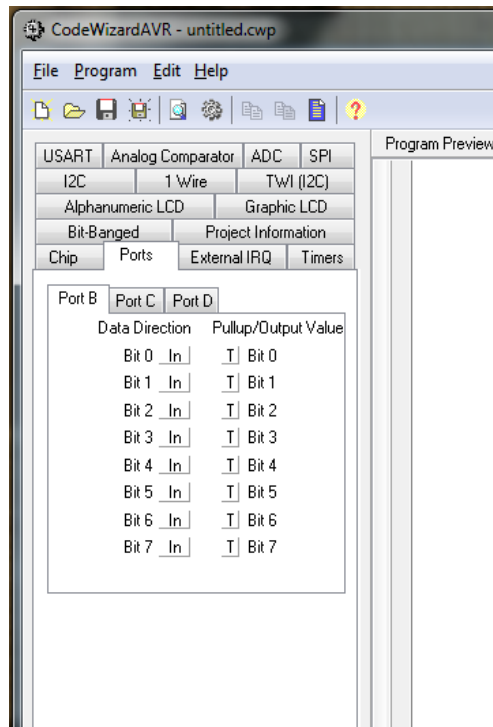
Gambar 2.8 Project baru menggunakan CodeWizardAVR

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.



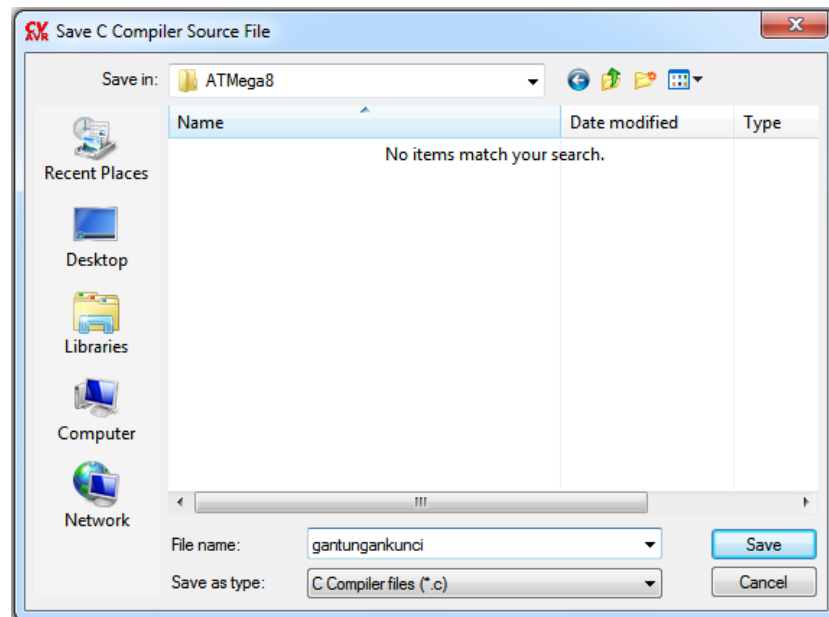
Gambar 2.9 Pengaturan chip pada CodeVisionAVR

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



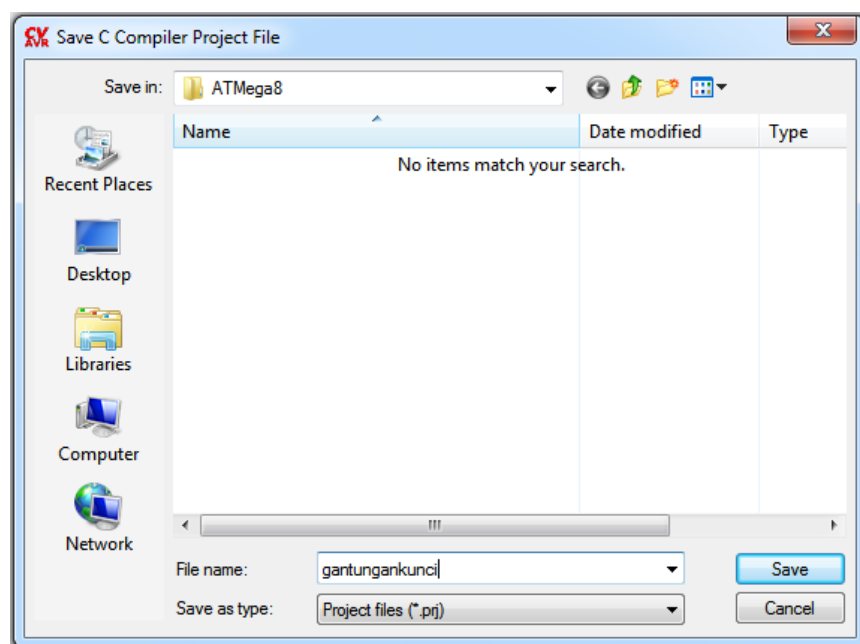
Gambar 2.10 Pengaturan port pada CodeVisionAVR

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



Gambar 2.11 Penyimpanan file.C

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



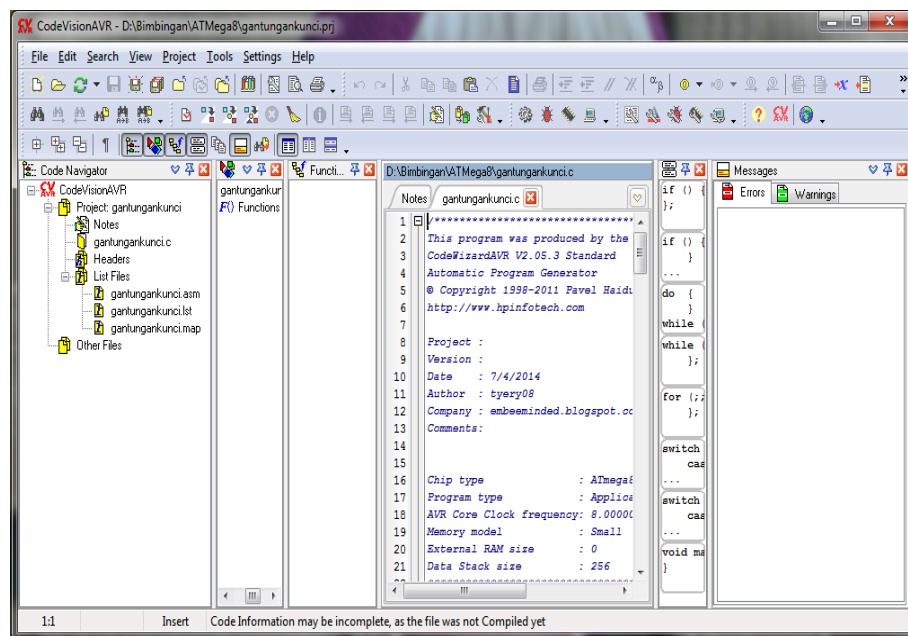
Gambar 2.12 Penyimpanan file.PRJ

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.



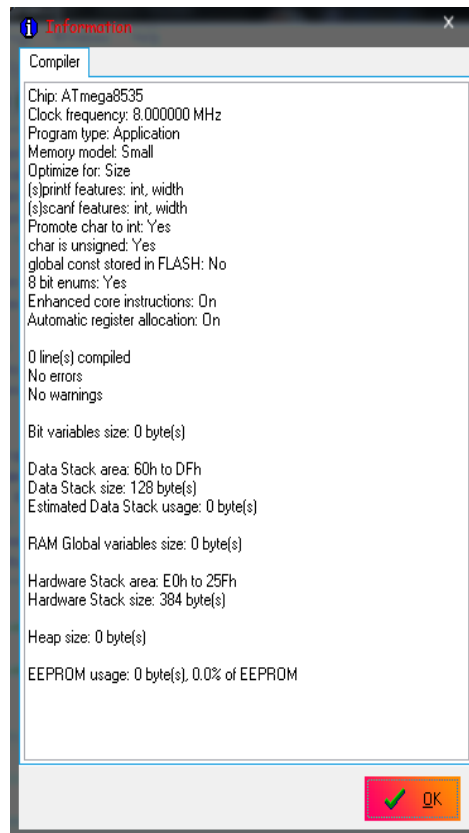
Gambar 2.13 Penyimpanan file.cwp

Tampilan awal setelah menggunakan *Code Wizard*



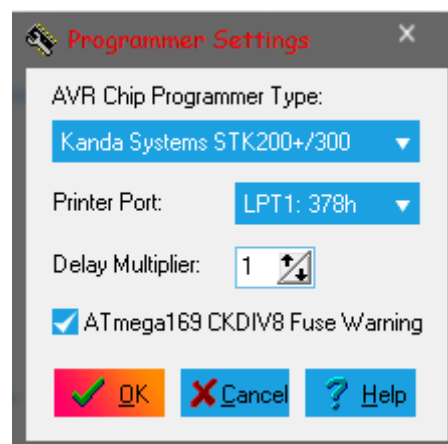
Gambar 2.14 Tampilan awal pada saat menggunakan code wizard

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



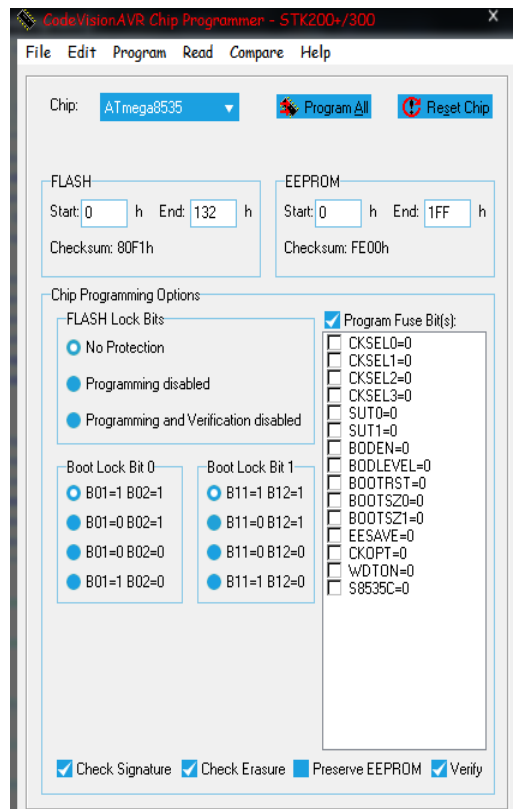
Gambar 2.15 Hasil proses kompilasi

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 2.16 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 *Verity* 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 2.17 Pengaturan chip programming options