BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian sebelumnya yang dilakukan oleh (Ma'ful Wahyu Nurhadi dan Paulinus Yunawan Widiantoro, 2010). Dalam jurnal yang berjudul "Jemuran Pakaian Otomatis dengan Menggunakan Sensor Cahaya (LDR) dan Sensor Hujan". Pada jurnal ini dijelaskan bahwa rel jemuran yang akan bergerak untuk masuk atau keluar dari atap jemuran sesuai dengan hasil *input*an sensor cahaya (LDR) dan sensor hujan.

Kemudian pada penelitian yang dilakukan oleh (Monilia Sitophila, Heriyanto dan Samsul Hidayat, 2014) dalam jurnal yang berjudul "Rancang Bangun Atap Sirip Otomatis Menggunakan LDR dan Sensor Tetes Air Hujan Berbasis Mikrokontroler". Adapun cara kerjanya yaitu menutup dan membuka atap otomatis berdasarkan hasil dari *inputan* sensor cahaya (LDR) dan sensor hujan. Apabila terdeteksi hujan maka sensor hujan akan mengirim perintah ke mikrokontroler untuk mengambil tindakan menutup atap sirip agar jemuran pakaian terlindungi dari hujan. Dan sebaliknya apabila terdeteksi cuaca cerah dan tidak hujan maka atap sirip akan membuka.

2.1.1 Perbedaan dengan Penelitian Sebelumnya

Dalam laporan akhir yang berjudul "Rancang Bangun Atap Jemuran Otomatis Berdasarkan Pencahayaan Dengan Logika Fuzzy Berbasis Mikrokontroler" yang bertujuan untuk membuat suatu atap yang dapat membuka tutup otomatis berdasarkan dari hasil masukan sensor LDR (cahaya) dan juga sensor hujan. Pada rancang bangun alat ini terdapat alat untuk mengeringkan jemuran disaat atap menutup yaitu menggunakan kipas angin yang menggantikan panas matahari sebagai pengeringnya. Dengan menggunakan kipas angin ini tidak akan merusak warna maupun tekstur dari jemuran pakaian itu sendiri dan lebih aman untuk digunakan sebagai alat pengering jemuran pakaian. Di rancang bangun atap otomatis ini menggunakan sensor LDR (cahaya) yang difuzzy kan

dengan menghasilkan 3 kategori cuaca yaitu cerah, mendung dan gelap. Apabila sensor LDR mendeteksi cuaca cerah maka atap terbuka, apabila cuaca mendung maka atap akan menutup ½ dan kipas angin akan aktif, dan apabila cuaca gelap maka atap akan menutup keseluruhan jemuran dan kipas angin juga akan aktif, kipas angin berfungsi sebagai pengering jemuran disaat terjadi hujan dan cuaca mulai mendung dan gelap. Pada rancang bangun atap ini juga menggunakan 4 buah sensor hujan yang mendeteksi turunnya hujan atau tidak yang berfungsi untuk menjaga pakaian dari cuaca hujan.

2.2 Mikrokontroler ATMEGA 8535

Mikrokontroller adalah suatu kombinasi mikroprosesor, piranti I/O (*Input/Output*) dan memori, yang terdiri atas ROM (*Read Only Memory*) dan RAM (*Random Access Memory*), dalam bentuk keping tunggal (*single chip*). Mikrokontroller ATmega8535 adalah mikrokontroller 8 bit buatan ATMEL dengan 8 *KByte System Programable Flash* dengan teknologi memori tak sumirna (*nonvolatile*), kepadatan tinggi, dan kompatibel dengan pin *out* dan set instruksi standar industri MCS51 INTEL. Arsitektur yang digunakan dengan RISC (*Reduce Instruction set in singgle chip*). Mikrokontroller ATmega8535 memiliki karakteristik sebagai berikut :

1. Kompatibel dengan produk keluarga MCS51.

2. Dapat digunakannya bahasa C sebagai bahasa pemrogramannya.

3. *Programmable Flash Memory* sebesar 8 *KByte*.

4. Memiliki 512 *Bytes* EEPROM yang dapat diprogram.

5. Ketahanan (*endurance*) : 10.000 siklus tulis/hapus.

6. Jangkauan operasi : 4,5 – 5,5 Volt.

7. *Fully Static Operation* : 0 Hz – 16 MHz untuk ATmega8535.

8. Dua level Program *Memory Lock* yaitu *flash program* dan EEPROM data *seccurity*

9. RAM *Internal* 128 X 8 bit,

10. Memiliki 32 jalur I/O yang dapat diprogram,

11. Satu pencacah 8 bit dengan *separate prescaler*,

12. Satu pencacah16 bit dengan *separate prescaler*,

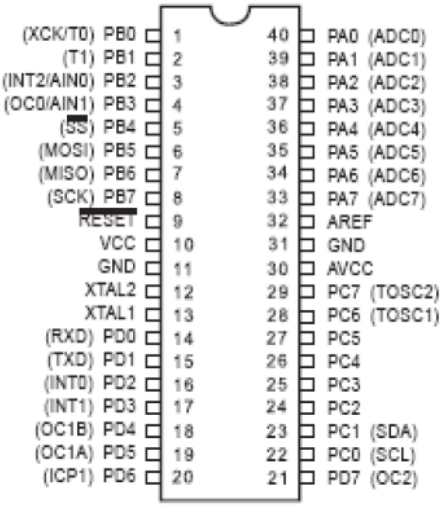
13. Sumber interupsi (*interrupt source*) eksternal dan internal,

14. Kanal pengirim-penerima tak serempak universal (UART - *Universal Asynchronous Receiver - Transmitter*) yang dapat diprogram.

15. *Low-power Idle* dan *Power-down*.

2.2.1 Susunan Pin MIKROKONTROLER ATmega8535

Bentuk kemasan dan susunan pin mikrokontroler dari ATmega8535 diperlihatkan seperti pada Gambar 2.1.



Gambar 2.1 Susunan pin pada ATmega8535 (Agus Bejo,2008:4)

Penjelasan dari masing-masing kaki adalah sebagai berikut :

1. VCC (kaki 40) dihubungkan ke Vcc

2. GND (kaki 20) dihubungkan ke ground.
3. PortA (PA7..PA0) (kaki 32-39) merupakan port 8 bit dua arah (bidirectional) I/O. Port ini berfungsi sebagai port data/alamat I/O ketika menggunakan SRAM eksternal.
4. Port B (PB7..PB0) (kaki 1-8) merupakan port 8 bit dua arah (bidirectional) I/O, untuk berbagai keperluan (multi purpose).
5. Port C (PC7..PC0) (kaki 21-28) adalah port 8 bit dua arah I/O, dengan internal pull-up resistor. Port C ini juga berfungsi sebagai port alamat ketika menggunakan SRAM eksternal.
6. Port D (PD7..PD0) (kaki 10-17) adalah port 8 bit dua arah I/O dengan resistor pull-up internal. Port D juga dapat berfungsi sebagai terminal khusus.
7. Reset (kaki 9) ketika kondisi rendah rendah yang lebih lama dari 50 nS mikrokontroler akan reset walaupun detak tidak berjalan.
8. XTAL1 (kaki 19) masukan bagi penguat osilator terbalik dan masukan bagi rangkaian operasi detak internal.
9. XTAL2 (kaki 18) keluaran dari penguat osilator terbalik.
10. ICP (kaki 31) adalah masukan bagi masukan fungsi Capture Timer/counter1.
11. OC1B (kaki 29) adalah kaki keluaran bagi fungsi Output CompareB keluaran Timer/Counter1.
12. ALE (Address Latch Enable) (kaki 30) digunakan ketika menggunakan SRAM eksternal. Kaki ini digunakan untuk mengunci 8 bit alamat bawah pada saat siklus akses pertama, dan berfungsi sebagai port data pada siklus akses kedua.

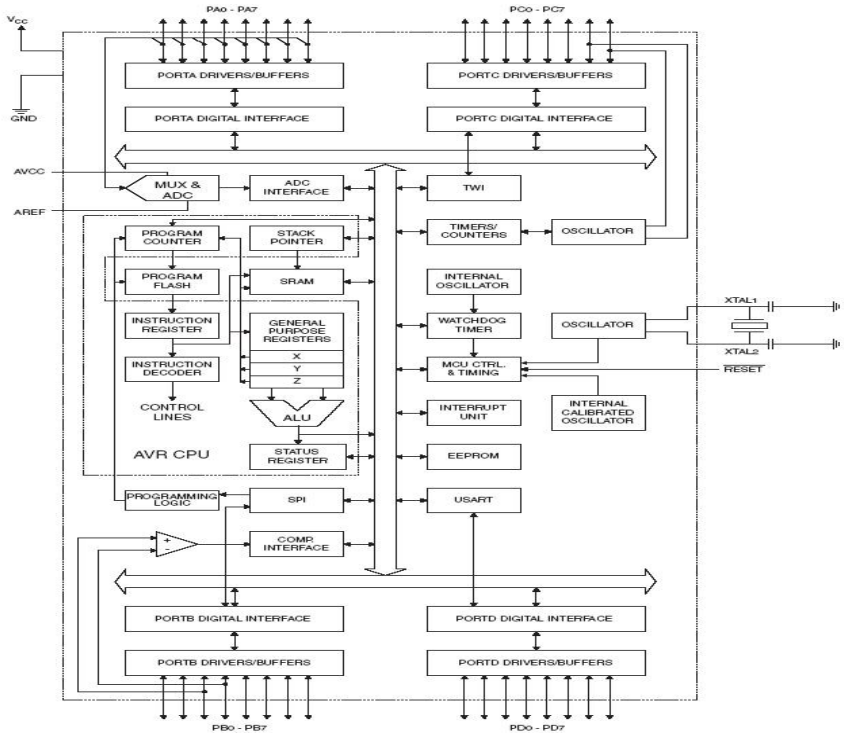
2.2.2 Fitur ATMega 8535

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.

2.2.3 Diagram Blok ATmega8535



Gambar 2.2 Diagram blok Mikrokontroler ATmega8535

Sumber : www.atmel.com

ATmega8535 mempunyai 32 *general purpose register* (R0..R31) yang terhubung langsung dengan *Arithmetic Logic Unit* (ALU), sehingga register dapat diakses dan dieksekusi hanya dalam waktu satu siklus *clock*. ALU merupakan tempat dilakukannya operasi fungsi aritmetik, logika dan operasi bit. R30 disebut juga sebagai Z-Register, yang digunakan sebagai register penunjuk pada pengalamatan tak langsung. Didalam ALU terjadi operasi aritmetik dan logika

antar register, antara register dan suatu konstanta, maupun operasi untuk register tunggal (*single register*). (Delta Agus,2008:24)

2.3 Logika Fuzzy

Logika fuzzy merupakan suatu cara atau metode untuk memetakan suatu ruang input kedalam suatu ruang output. Titik awal dari konsep modern mengenai ketidakpastian adalah *paper* yang dibuat oleh Lotfi A Zadeh (1965) dimana Zadeh memperkenalkan teori yang memiliki objek-objek dari himpunan fuzzy yang memiliki batasan tidak presisi dan keanggotaan himpunan fuzzy dan bukan dalam bentuk logika benar (*true)* atau salah (*false)* tapi dinyatakan dalam derajat (*degree).* Konsep ini dikenal dengan Fuzziness dan teorinya dinamakan Fuzzy *set theory.* Fuziness dapat didefinisikan sebagai logika kabur berkenaan dengan semantik dari suatu kejadian, fenomena atau pernyataan itu sendiri.

Contoh logika fuzzy yaitu sebagai berikut :

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidaknya pelayan yang diberikan;
3. Anda mengatakan pada saya seberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.
4. Penumpang taksi berkata pada sopir taksi seberapa cepat laju kendaraan yang diinginkan, sopir taksi akan mengatur pijakan gas taksinya.

Fuzzy *system* didasari atas konsep himpunan kabur yang memetakan domain *input* kedalam domain *output.* Perbedaan mendasar himpunan tegas dengan himpunan kabur adalah nilai keluarannya. Himpunan tegas hanya memiliki dua nilai yaitu 0 dan 1 sedangkan himpunan kabur memiliki banyak nilai keluaran yang dikenal dengan nilai derajat keanggotaan.

Logika Fuzzy adalah peningkatan dari logika *bolean* yang berhadapan dengan konsep kebenaran sebagian dimana logika klasik (*crisp)* menyatakan bahwa segala hal dapat diekspresikan dalam istilah *binary* (0 atau 1, ya atau tidak

serta hitam atau putih). Logika fuzzy menggantikan kebenaran *bolean* dengan tingkat kebenaran, logika fuzzy memungkinkan nilai 0 atau 1, tingkat keabuan dan juga hitam atau putih serta dalam bentuk *linguistic,* konsep tidak pasti seperti "sedikit", "lumayan" dan "kuat".

Ada beberapa alasan mengapa orang menggunakan logika fuzzy (Kusumadewi S, Purnomo H : 2010) antara lain :

1. Konsep logika fuzzy mudah dimengerti, konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
2. Logika fuzzy sangat fleksibel.
3. Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
4. Mampu memodelkan fungsi-fungsi *non linear* yang sangat kompleks.
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa melaui proses pelatihan.
6. Dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Didasarkan pada bahasa alami.

2.3.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan μA[x] memiliki 2 kemungkinan (Kusumadewi S, Purnomo H : 2010) yaitu :

1. Satu berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan himpunan fuzzy dengan probabilitas menimbulkan kerancuan. Keduanya memiliki *interval* [0,1], namun interprestasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan fuzzy memberikan suatu ukuran terhadap pendapat atau keputusan sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai

beanr dalam jangka panjang. Misalnya , jika nilai keanggotaan suatu himpunan fuzzy USIA adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas usia berarti 10% dari himpunan tersebut diharapkan tidak muda.

Adapun persamaan logika fuzzy representasi linier turun adalah :

$\mu(x) = (b-x) / (b-a)$; $a \leq x \leq b$.....(2.1)

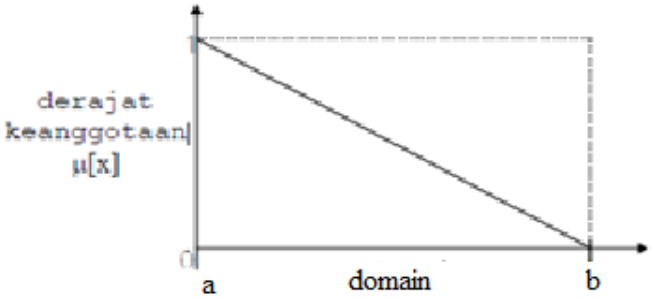
Keterangan :

$\mu(x)$ = hasil dari logika fuzzy

a = variabel pertama

b = variabel kedua

x= nilai di antara variabel pertama dan kedua.



Gambar 2.3 Representasi Linier Turun

Sumber : http://repo.pens.ac.id/434/1/878.pdf

Adapun persamaan logika fuzzy representasi linier naik adalah :

$\mu(x) = (x-c) / (d-c)$; $c \leq x \leq d$.....(2.2)

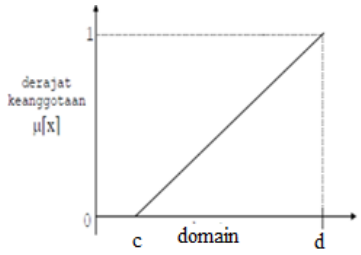
Keterangan :

$\mu(x)$ = hasil dari logika fuzzy

c = variabel pertama

d = variabel kedua

x= nilai di antara variabel pertama dan kedua.



Gambar 2.4 Representasi Linier Naik

Sumber : http://repo.pens.ac.id/434/1/878.pdf

2.3.2 Metode Fuzzy Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran Mamdani, hanya saja output (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Michio Sugeno mengusulkan penggunaan *singleton* sebagai fungsi keanggotaan dari konsekuen. *Singleton* adalah sebuah himpunan fuzzy dengan fungsi keanggotaan yang pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut.

Secara umum bentuk model *fuzzy* Sugeno Orde Nol adalah :

IF (x1 is A1) o (x2 is A2) o (x3 is A3) o… o (xN is AN) THEN z=k.....(2.3)

dengan Ai adalah himpunan *fuzzy* ke-I sebagai antesenden, dan k adalah suatu konstanta tegas sebagai konsekuen. (Saddam Rabbani,2013)

2.4 Sensor LDR (*Light Dependent Resistor*)

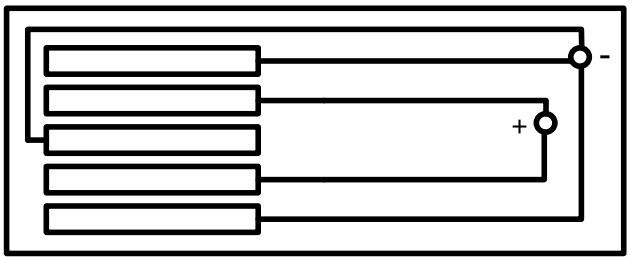
LDR atau light depending diode adalah suatu alat yang digunakan untuk mendeteksi intensitas cahaya. Prinsip kerja LDR sangat sederhana, bila intensitas cahaya yang diterima LDR kecil, maka resistansinya kecil. Sehingga arus yang mengalir akan besar. Begitu juga sebaliknya, bila intensitas cahaya yang diterima besar, maka resistansi pada LDR akan besar sehingga arus yang mengalir akan kecil. (Adi Wisaksono, 2011:3)

Gambar 2.5 Sensor Cahaya (LDR)

Sumber : http://elektronikadasar.info/sensor-cahaya.htm

2.5 Sensor Hujan

Sensor hujan berfungsi untuk mendeteksi adanya air yang berupa air hujan atau embun pada malam hari. Prinsip kerja plat konduktor sama seperti saklar. Sensor ini berupa dua buah lempeng konduktor yang akan terhubung bila terkena air. Air dapat menghantarkan arus listrik karena air merupakan salah satu konduktor walaupun bukan termasuk konduktor yang bagus. Berikut ini adalah tampilan dari sensor hujan dengan menggunakan papan PCB seperti pada gambar 2.6. (Adi Wisaksono, 2011:2)



Gambar 2.6 Sensor Hujan

2.6 Motor Servo

Motor servo adalah sebuah motor dengan sistem closed feedback di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor stepper. Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan motornya dapat dikendalikan hanya dengan memberikan pengaturan duty cycle sinyal PWM pada bagian pin kontrolnya. Motor Servo tampak pada gambar 2.7. (Ariel Yagusandri, 2011:8)

Gambar 2.7 Motor Servo

Sumber : https://www.futurlec.com/HardwareMain.shtml

2.7 *Relay*

Relay adalah komponen yang menggunakan prinsip kerja medan magnet untuk menggerakkan saklar atau mengaktifkan *switch*. Saklar ini digerakkan oleh magnet yang dihasilkan oleh kumparan di dalam *relay* yang dialiri arus listrik. (Ririn Noviyantika, 2010).

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

a. *Normally Close* (NC)

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

b. *Normally Open* (NO)

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

Gambar 2.8 *Relay*

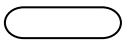
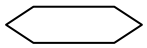
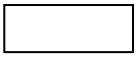
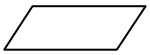
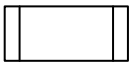
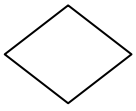
Sumber : http://www.tradekorea.com/product/detail/P288227/relay.html

2.8 Flowchart

Flowchart atau diagram alir adalah bagan-bagan yang mempunyai arus yang mempunyai langkah-langkah penyelesain suatu masalah. Dengan adanya flowchart, akan sangat membantu untuk memvisualisasikan isi dari setiap halaman tersebut. Flowchart adalah sekumpulan simbol-simbol yang menunjukan atau menggambarkan rangkaian kegiatan-kegiatan program dari awal hingga akhir, jasi flowchart juga digunakan untuk menggambarkan urutan langkah- langkah pekejaan disuatu algoritma. (Ahmad Wahyudi, 2011).

Berikut ini adalah beberapa gambaran simbol yang digunakan dalam menggambar suatu flowchart:

Tabel 2.1 Simbol-simbol Flowchart

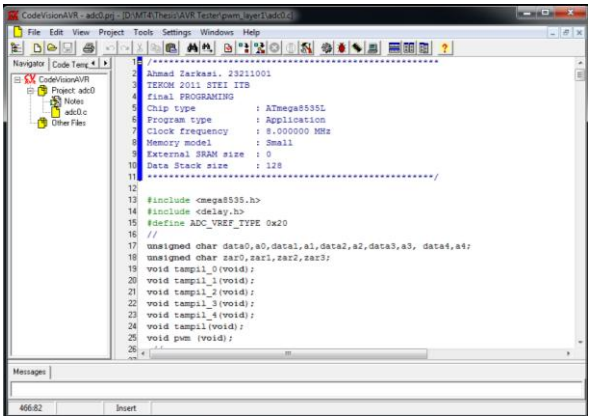
Simbol	**Nama**	**Fungsi**
	Terminator	Permulaan/akhir program
	Garis Alir (Flow Line)	Arah Aliran Program
	Preparation	Proses inisialisasi/pemberian harga awal
	Proses	Proses perhitungan/proses pengolahan data
	Input / Output Data	Proses input/output data, parameter, informasi.
	Predefined Process (sub program)	Permulaan sub program/proses menjalankan sub program
	Decision	Perbandingan pernyataan, penyelesaian data yang memberikan pilihan untuk langkah selanjutnya
	On Page Connector	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	Off Page Connector	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

2.9 *CodeVisionAVR* C Compiler

CodeVisionAVR merupakan salah satu *software* kompiler yang khusus digunakan untuk mikrokontroler keluarga AVR. Meskipun *CodeVisionAVR* termasuk *software* komersial, namun kita dapat menggunakannya secara langsung, karena terdapat versi evaluasi yang tersedia secara gratis walaupun dengan kemampuan yang dibatasi. Gambar 2.9 merupakan gambar tampilan dari *CodeVisionAVR*.

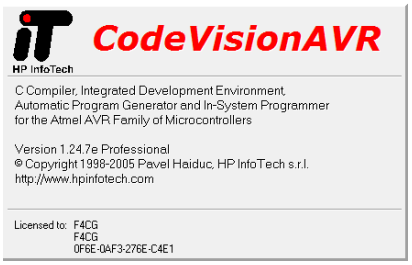
Beberapa kelebihan yang dimiliki oleh *CodeVisionAVR* antara lain:

1. Menggunakan *Integrated Development Enveronment* (IDE),
2. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas *CodeWizardAVR*,
3. Memiliki fasilitas *programing downloader*,
4. Memilik fasilitas *debugger* sehingga dapat menggunakan *software compiler* untuk mengecek *assembler code* nya,
5. Memiliki terminal komunikasi serial yang terintegrasi, sehingga dapat digunakan untuk membantu pengecekan program yang menggunakan fasilitas UART.



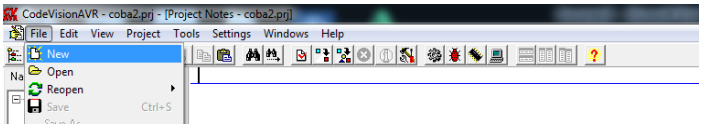
Gambar 2.9 Tampilan halaman program *CodeVisionAVR*.

1. Buat folder pada **drive (C:\)** atau drive yang lain, kemudian beri nama foldernya.
   Misal drive **D:\ Coba**
2. Klik **tampilan/shortcut CodevisionAVR** pada desktop komputer



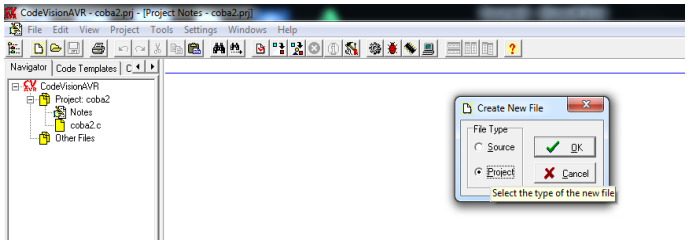
Gambar 2.10 Tampilan *shotcut* awal *CodeVision*AVR

3. Klik *new file* pada lembar kerja *codevision*AVR



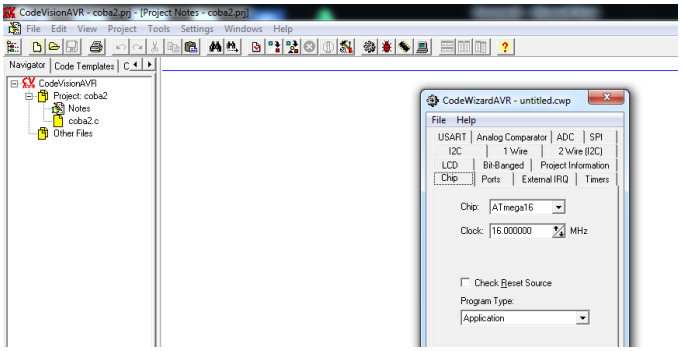
Gambar 2.11 Tampilan *toolbar* pada *CodeVision*AVR

4. Pada **Create New File**, pilih **Project → klik Ok, → klik Yes**



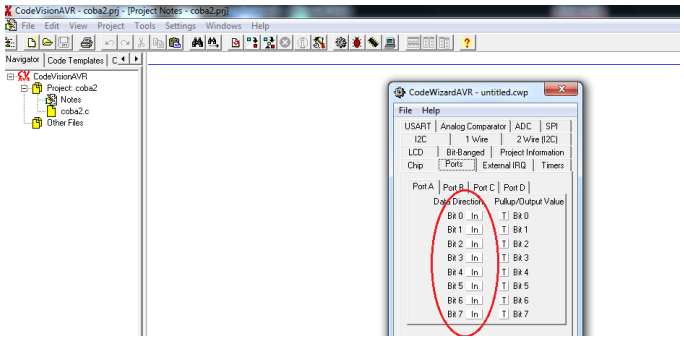
Gambar 2.12 Tampilan pembuatan *file* program baru

5. Setelah itu akan muncul jendela **CodeWizardAVR**. Jendela ini digunakan untuk mengatur (setting), peripheral internal yang akan digunakan. Misalnya jenis mikrokontroler, port-port yang digunakan, Timer dll.



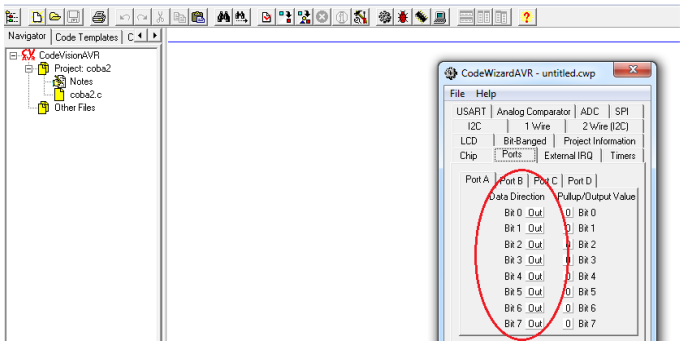
Gambar 2.13 Men*setting* peripheral *internal*

6. Misalnya kita akan mengatur PORT A dan PORT B sebagai INPUT, dan PORTC dan PORTD sebagai output. Maka klik **Ports → klik Port A**



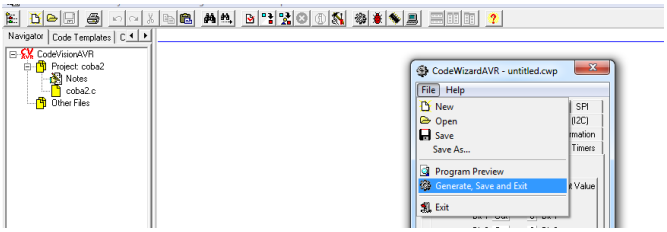
Gambar 2.14 *Setting Input* Port A dan Port B

Kemudian klik tombol (IN) sehingga berubah menjadi kondisi (OUT).



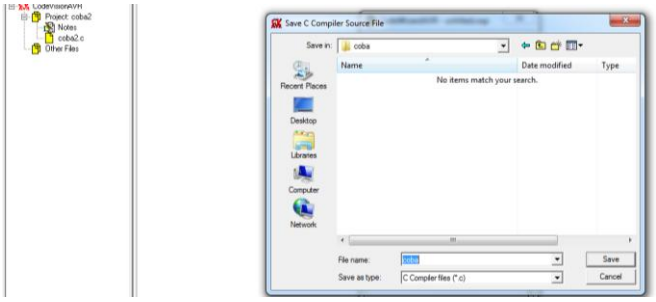
Gambar 2.15 *Setting Output* PORT C dan PORT D

7. Buka **File → klik Generate, Save and Exit**



Gambar 2.16 Penyimpanan program

Kemudian cari **folder D:/Coba**, pada **File name → tulis coba → klik save**

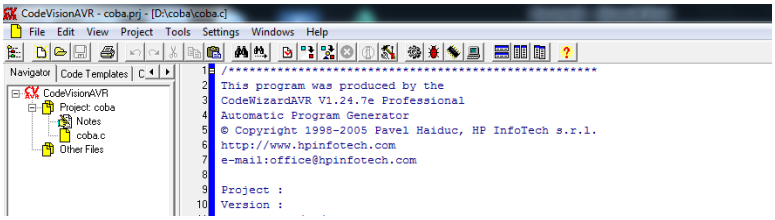


Gambar 2.17 Tampilan *Compiler*

Kemudian akan muncul jendela ***Save C Compiler Project File***, pada file name tulis **coba → klik save**

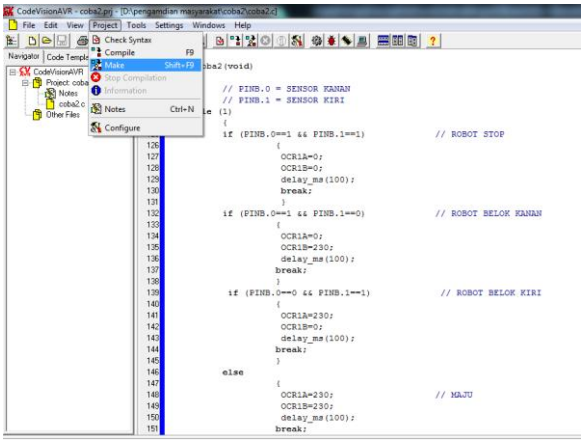
Kemudian akan muncul jendela **Save D:\coba\untitled.cwp As**, pada **file name tulis coba, → klik save.**

8. Langkah selanjutnya adalah akan muncul **workspace / area kerja** yang digunakan untuk membuat program yang diinginkan



Gambar 2.18 Tampilan halaman kerja program

9. Setelah program selesai dibuat, langkah selanjutnya adalah mengubah program tersebut kedalam Kode Hexadesimal **(.Hex)**. hal ini dikarenakan mikrokontroler hanya dapat menerima kode hexadesimal dalam memori programnya (EPROM). Caranya **Klik Project → klik BUILD**



Gambar 2.19 Proses *Build* Program

Jika tidak terdapat kesalahan, maka akan muncul tampilan jendela *no errors*. Kemudian **klik OK.** (Zuda Eka, 2010:3)