

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Ruang Tunggu Otomatis

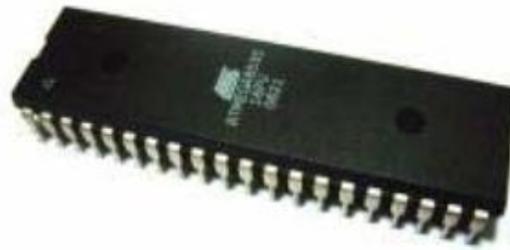
Ruang tunggu adalah ruang yang digunakan sebagai tempat menunggu bagi orang. Ruang tunggu otomatis adalah sebuah ruangan dimana memiliki teknologi canggih yang membuat orang merasa nyaman dan aman berada di ruangan tersebut. Ruang tunggu otomatis ini dilengkapi dengan pintu yang terbuka dan tertutup secara otomatis apabila terdeteksi orang yang akan masuk atau keluar, LCD yang berfungsi untuk menampilkan jumlah orang yang ada di dalam ruangan serta jumlah tempat duduk yang tersedia, serta sistem pencahayaan dalam ruang yang disesuaikan dengan jumlah orang dalam sehingga dapat menghemat energi listrik pada penggunaan lampu.

(Sumber : Hadibowo : 2010, Hal 15)

2.2 Mikrokontroler ATmega 8535

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

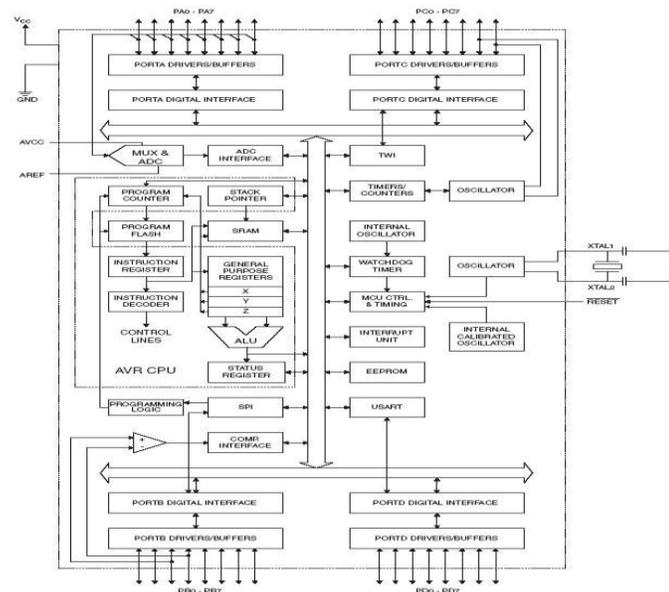
(Sumber : Ardianto : 2010, Hal 4)



Gambar 2.1 Mikrokontroler AVR ATmega8535

2.2.1 Diagram Blok ATmega 8535

Ada 3 jenis tipe AVR yaitu AT Tiny, AVR klasik, AT Mega. Perbedaannya hanya pada fasilitas dan I/O yang tersedia serta fasilitas lain seperti ADC, EEPROM dan lain sebagainya. ATmega 8535 merupakan salah satu tipe AVR yang memiliki teknologi RISC dengan kecepatan maksimal 16 MHz membuat ATmega 8535 lebih cepat bila dibandingkan dengan varian MCS 51. Dengan fasilitas yang lengkap tersebut menjadikan ATmega8535 sebagai mikrokontroler yang *powerfull*.



Gambar 2.2 Diagram Blok ATmega 8535

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

(Sumber : Ardhiyanto : 2010, hal 4)

2.2.2 Fitur ATMega 8535

Mikrokontroler ATMega8535 memiliki beberapa fitur diantaranya sebagai berikut :

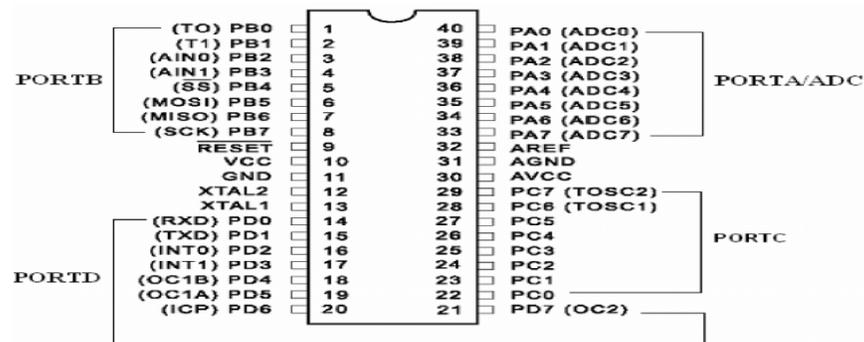
1. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 Mhz.
2. Kapabilitas memori flash 8KB, SRAM sebesar 512 byte, dan EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte.
3. ADC internal dengan fidelitas 10 bit sebanyak 8 saluran.
4. Portal komunikasi serial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

(Sumber : Ardhiyanto : 2010, Hal 4)

2.2.3 Konfigurasi Pin ATMega8535

IC Atmega 8535 ada 2 jenis yaitu jenis PDIP (berbentuk balok) dan jenis TQFP/MLF (berbentuk kotak) yang pada dasarnya memiliki fasilitas yang sama,

hanya saja memiliki bentuk yang berbeda sehingga letak kaki-kaki IC berbeda mengikuti bentuknya. Gambar 2.3 berikut ini merupakan konfigurasi pin mikrokontroler ATmega8535.



Gambar 2.3 Konfigurasi Pin ATmega8535

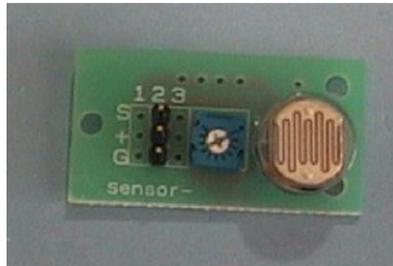
Dari Gambar 2.3 diatas dapat dilihat ada 40 pin IC yang Secara fungsional konfigurasi pin tersebut sebagai berikut :

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

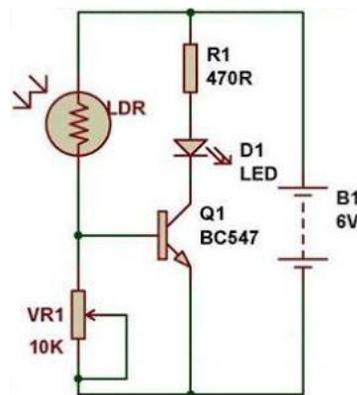
(Sumber : Ardhiyanto : 2010, Hal 4)

2.3 Sensor *Light Dependent Resistor* (LDR)

Sensor *Light Dependent Resistor* (LDR) adalah resistor yang besar resistansinya bergantung terhadap intensitas cahaya yang menyelimuti permukaannya. LDR, dikenal dengan banyak nama: foto-resistor, foto-konduktor, sel foto-konduktif, atau hanya foto-sel, yang sering digunakan dalam literatur adalah foto-resistor atau foto-sel. Sebuah Foto-resistor atau LDR adalah komponen yang menggunakan foto-konduktor di antara dua pin-nya, bentuk fisik sensor LDR seperti pada gambar 2.4. Saat permukaannya terpapar cahaya akan terjadi perubahan resistansi di antaranya.



Gambar 2.4 Sensor LDR



Gambar 2.5 Rangkaian Sensor LDR

Mekanisme di balik fotoreistor atau LDR adalah fotokonduktivitas, yaitu suatu peristiwa perubahan nilai konduktansi bahan semikonduktor saat energi foton dari cahaya diserap olehnya. Ketika digunakan sebagai Foto-resistor atau LDR, bahan semikonduktor hanya digunakan sebagai elemen resistif dan tidak ada koneksi PN-

nya seperti pada gambar 3.4. Dengan demikian, Foto-resistor atau LDR adalah murni komponen pasif. Sensor ini menghasilkan *keluaran* berupa nilai analog yang dibaca menggunakan fitur ADC (*Analog to Digital Converter*) pada mikrokontroler.

(Sumber : Sukendar dkk, Vol 1, No 1: 2013)

2.4 Sensor PIR (*Passive Infra Red*)

Sensor gerak PIR (*Passive Infra Red*) adalah sensor yang berfungsi untuk pendeteksi gerakan yang bekerja dengan cara mendeteksi adanya perbedaan/perubahan suhu sekarang dan sebelumnya. Sensor gerak menggunakan modul pir sangat simpel dan mudah diaplikasikan karena Modul PIR hanya membutuhkan tegangan input DC 5V cukup efektif untuk mendeteksi gerakan hingga jarak 5 meter. Ketika tidak mendeteksi gerakan, keluaran modul adalah LOW. Dan ketika mendeteksi adanya gerakan, maka keluaran akan berubah menjadi HIGH. Adapun lebar pulsa HIGH adalah $\pm 0,5$ detik. Sensitifitas Modul PIR yang mampu mendeteksi adanya gerakan pada jarak 5 meter memungkinkan kita membuat suatu alat pendeteksi gerak dengan keberhasilan lebih besar.

(Sumber : Hartika, Vol 6, No 1: 2013)



Gambar 2.6 Sensor PIR

2.5 LCD 2x16

LCD merupakan singkatan dari *Liquid Crystal Display* yang dapat menampilkan karakter dua baris dengan tiap baris 16 karakter. Terdapat dua jenis LCD yaitu LCD karakter dan LCD grafik. LCD karakter digunakan pada proyeksi-

proyek elektronika namun sekarang, LCD grafik mulai banyak digunakan oleh masyarakat. LCD karakter yang umumnya digunakan yaitu LCD 2x16 karakter.

(Sumber : Syarifuddin, Vol 2, No 1: 2009)



Gambar 2.7 Tampilan LCD

2.6 Motor Servo

Motor Servo merupakan motor yang digunakan sebagai sumber bergerak dalam sistem servo, dengan umpan balik (*feedback*) berupa posisi dan kecepatan untuk setiap aksi pengontrolan. Motor Servo dapat bekerja dengan tepat mengikuti instruksi yang diberikan, meliputi posisi dan kecepatan dengan karakteristik sebagai berikut :

1. Berputar dengan mantap pada daerah kecepatan yang diberikan.
2. Mengubah kecepatan dengan cepat, dan membangkitkan torka yang besar dari ukuran yang kecil.

Motor servo merupakan suatu alat untuk mengubah energi listrik menjadi energi mekanik. Magnet permanent motor DC servo mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Salah satu medan dihasilkan oleh magnet permanent dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Motor servo merupakan sebuah motor DC kecil yang diberi sistem gear dan potensio meter sehingga dia dapat menempatkan horn servo pada posisi yang dikehendaki. Saat motor berputar, arus pada kumparan motor

menghasilkan torsi yang nilainya konstan. Gambar 2.10 merupakan gambar motor servo.

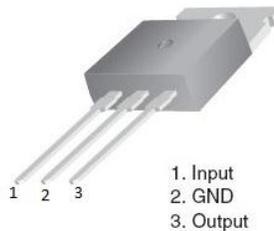
(Sumber : Yohannes, Vol 9, No 2: 2011)



Gambar 2.8 Motor Servo

2.7 IC LM7805

IC LM7805 merupakan komponen yang digunakan untuk meregulasi tegangan untuk memperoleh keluaran sebesar ± 5 volt. IC ini memiliki 3 buah kaki yaitu kaki *input*, *common*, dan *output*.



Gambar 2.9 IC LM7805

Meskipun IC LM7805 dirancang terutama untuk output tegangan tetap sebesar 5 volt, IC ini juga memungkinkan untuk digunakan oleh komponen *eksternal* agar mendapatkan output tegangan DC sebesar 5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, 20V, dan 24V.

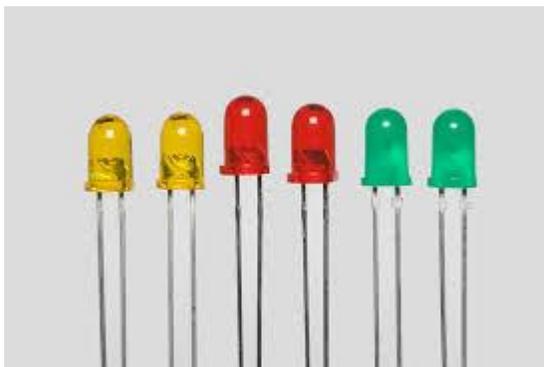
(Sumber : Ridwan : 2006, Hal 1)

2.8 LED (*Light Emitting Diode*)

LED atau singkatan dari Light Emitting Diode adalah salah satu komponen elektronik yang tidak asing lagi di kehidupan manusia saat ini. LED saat ini sudah banyak dipakai, seperti untuk penggunaan lampu permainan anak-anak, untuk rambu-rambu lalu lintas, lampu indikator peralatan elektronik hingga ke industri, untuk lampu emergency, untuk televisi, komputer, pengeras suara (speaker), hard disk eksternal, proyektor, LCD, dan berbagai perangkat elektronik lainnya sebagai indikator bahwa sistem sedang berada dalam proses kerja, dan biasanya berwarna merah atau kuning. LED ini banyak digunakan karena konsumsi daya yang dibutuhkan tidak terlalu besar dan beragam warna yang ada dapat memperjelas bentuk atau huruf yang akan ditampilkan. dan banyak lagi

Pada dasarnya LED itu merupakan komponen elektronika yang terbuat dari bahan semi konduktor jenis dioda yang mampu memancarkan cahaya. LED merupakan produk temuan lain setelah dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N. Untuk mendapatkna emisi cahaya pada semikonduktor, doping yang pakai adalah galium, arsenic dan phosporus. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula.

(Sumber : Yohannes, Vol 9, No 2: 2011)



Gambar 2.10 LED

2.9 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*. Fuzziness 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 *boolean* 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 *boolean* 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 melalui proses pelatihan.
6. Dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Didasarkan pada bahasa alami.

(Sumber : Sudradjat : 2008, Hal 23)

2.9.1 Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A[x]$ memiliki 2 kemungkinan (Kusumadewi dkk : 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 interpretasi 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 benar 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.

Himpunan *fuzzy* memiliki 2 atribut yaitu :

- Linguistik yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alamai misalnya muda, paruh baya, tua.
- Numeris yaitu suatu nilai (angka) yang menunjukkan ukuran suatu variabel misalnya 40, 25, 30 dan lain sebagainya.

(Sumber : Sudradjat : 2008, Hal 12)

2.10 Bahasa Pemrograman C

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

Program Bahasa C tidak mengenal aturan penulisan dikolom tertentu sehingga bisa dimulai dari kolom manapun. Namun demikian untuk mempermudah pembacaan program dan keperluan dokumentasi sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak di baca. Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukkan main (). Program yang dijalankan berada dalam tubuh

program dan dimulai dengan tanda kurung buka { dan diakhiri dengan kurung tutup }. Semua yang tertulis di dalam tubuh program disebut blok.

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

(Sumber : Heryanto dkk : 2008, Hal 19)

a) Tipe Data

Tipe data merupakan bagian program paling penting karena mempengaruhi setiap instruksi yang akan dilaksanakan oleh komputer. Tipe-tipe data tersebut dapat dilihat pada Tabel 2.1

Tabel 2.1 Tipe-tipe data

Tipe Data	Ukuran	Jangkauan Nilai
Char	1 byte	-128 s/d 225
Unsigned Char	1 byte	0 s/d 225
Signed Char	1 byte	-128 s/d 127
Int	2 byte	-32.768 s/d 32.767
Short Int	2 byte	-32.768 s/d 32.767
Unsigned Int	2 byte	0 s/d 65.535
Signed Int	2 byte	-32.768 s/d 32.767
Long Int	4 byte	-2.147.483.648 s/d 2.147.483.647
Unsigned Long Int	4 byte	0 s/d 4.294.967.295

Signed Long Int	4 byte	-2.147.483.648 s/d 2.147.483.647
-----------------	--------	-------------------------------------

b) Konstanta

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

c) Variabel

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

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

d) Deklarasi Variabel

Bentuk umum pendeklarasian suatu variabel adalah: Nama_tipe nama_variabel.

Contohnya :

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

e) Deklarasi Constanta

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

Contohnya :

```
#define PH 3.14
```

```
#define nama "muttakin"
```

f) Deklarasi Fungsi

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

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

Tipe_fungsi nama_fungsi(parameter_fungsi); contohnya:

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

g) Operasi Aritmatika

Bahasa C menyediakan lima operator aritmatika yaitu :

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

h) Operasi Logika

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

1. && : Logika AND (dan)
2. || : Logika OR (atau)
3. ! : Logika NOT (ingkaran)

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

i) Komentar Program

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

```
// program ini dibuat oleh.....
```

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

Contoh kedua:

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

Bentuk ini berguna kalau pernyataannya berupa kalimat panjang sampai beberapa baris (Bejo, Hal 30: 2008).

2.11 CodeVisionAVR C Compiler v1.24.7e

Code Vision AVR C Compiler (CVAVR) merupakan kompilasi bahasa C untuk AVR. Kompilasi ini cukup memadai untuk belajar AVR, karena selain mudah penggunaannya juga didukung beberapa fitur yang sangat membantu dalam pembuatan *software* untuk keperluan pemrograman AVR.

CVAVR ini dapat berjalan dibawah sistem operasi Windows 9x, Me NT 4, 2000 dan XP. CVAVR ini dapat mengimplementasikan hampir semua instruksi bahasa C yang sesuai dengan arsitektur AVR, terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifik dari AVR. Hasil kompilasi objek CVAVR bisa digunakan sebagai *source debug* dengan AVR studio *debugger* dari ATMEL

Beberapa kelebihan yang dimiliki oleh *CodeVisionAVR* antara lain

1. Menggunakan *Integrated Development Environment* (IDE),
2. Mampu membangkitkan kode program secara otomatis dengan menggunakan fasilitas *CodeWizardAVR*,
3. Memiliki fasilitas *programming downloader*,
4. Memiliki fasilitas *debugger* sehingga dapat menggunakan *software compiler* untuk mengecek *assembler code* nya,

(Sumber : Heryanto dkk : 2008, Hal 9)