

BAB II TINJAUAN PUSTAKA

2.1 Definisi Robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Istilah robot berawal bahasa Cheko “*robota*” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput.

Saat ini hampir tidak ada orang yang tidak mengenal robot, namun pengertian robot tidaklah dipahami secara sama oleh setiap orang. Sebagian membayangkan robot adalah suatu mesin tiruan manusia (*humanoid*), meski demikian *humanoid* bukanlah satu-satunya jenis robot, jenis robot yang lain yaitu robot *mobile*, robot *manipulator* (tangan), robot berkaki, robot *flying*, robot *under water*. Pada kamus *Webster* pengertian robot adalah : *An automatic device that performs function ordinarily ascribed to human beings* (sebuah alat otomatis yang melakukan fungsi berdasarkan kebutuhan manusia). Dari kamus *Oxford* diperoleh pengertian robot adalah: *A machine capable of carrying out a complex series of actions automatically, especially one programmed by a computer*. (Sebuah mesin yang mampu melakukan serangkaian tugas rumit secara otomatis, terutama yang diprogram oleh komputer).

Pengertian dari *Webster* mengacu pada pemahaman banyak orang bahwa robot melakukan tugas manusia, sedangkan pengertian dari *Oxford* lebih umum, beberapa organisasi di bidang robot membuat definisi tersendiri. *Robot Institute*

of America memberikan definisi robot sebagai: *A reprogrammable multifunctional manipulator designed to move materials, parts, tools or other specialized devices through variable programmed motions for the performance of a variety of tasks* (Sebuah *manipulator* multifungsi yang mampu diprogram, didesain untuk memindahkan material, komponen, alat, atau benda khusus lainnya melalui serangkaian gerakan terprogram untuk melakukan berbagai tugas) *International Organization for Standardization (ISO 8373)* mendefinisikan robot sebagai: *An automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either Fixed in place or mobile for use in industrial automation applications* (Sebuah *manipulator* yang terkontrol, multifungsi, dan mampu diprogram untuk bergerak dalam tiga aksis atau lebih, yang tetap berada di tempat atau bergerak untuk digunakan dalam aplikasi otomasi industri). Dari beberapa definisi di atas, kata kunci yang ada yang dapat menerangkan pengertian robot adalah:

1. Dapat memperoleh informasi dari lingkungan (melalui sensor).
2. Dapat diprogram.
3. Dapat melaksanakan beberapa tugas yang berbeda.
4. Bekerja secara otomatis
5. Cerdas (*intelligent*)
6. Digunakan di industri

2.2 Definisi Sensor

Sensor adalah komponen yang digunakan untuk mendeteksi suatu besaran fisik menjadi besaran listrik sehingga dapat dianalisa dengan rangkaian listrik tertentu. Sensor sangat berperan penting dalam dunia robotika yang berfungsi sebagai input navigasi suatu robot, adapun jenis jenis sensor yaitu: sensor suara, sensor cahaya, sensor tekanan sensor api, sensor suhu, sensor kelembapan, sensor ultrasonic, sensor magnet. Pada dasarnya sensor dengan penting di dunia robotik tergantung kebutuhan dari robot itu sendiri.

2.2.1 Sensor Suara

Sensor suara adalah sebuah alat yang mampu mengubah gelombang *Sinusioda*, suara menjadi gelombang sinus energi listrik (*Alternating Sinusioda Electric Current*). Sensor suara berkerja berdasarkan besar/kecilnya kekuatan gelombang suara yang mengenai membran sensor yang menyebabkan Bergeraknya membran sensor yang juga terdapat sebuah kumparan kecil di balik membran tadi naik & turun. Oleh karena kumparan tersebut sebenarnya adalah ibarat sebuah pisau berlubang-lubang, maka pada saat ia bergerak naik-turun, ia juga telah membuat gelombang magnet yang mengalir melewatinya terpotong-potong. Kecepatan gerak kumparan menentukan kuat-lemahnya gelombang listrik yang dihasilkannya.

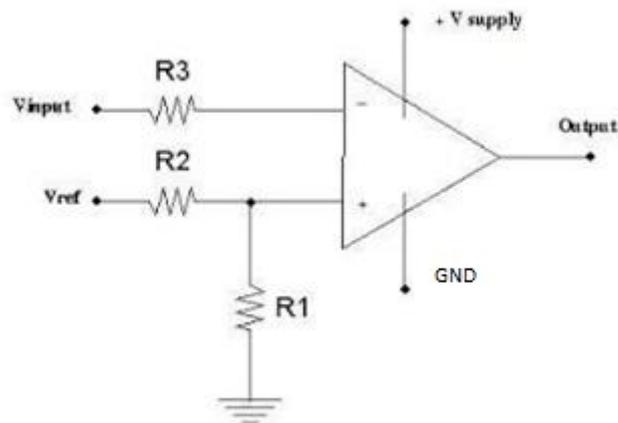
Komponen yang termasuk dalam Sensor suara yaitu *electric condenser microphone* atau mic kondenser. Intensitas suara adalah ukuran dari "aliran energi melewati satuan luas per satuan waktu" dan unit pengukuran adalah W/m² Probe intensitas suara mikrofon ini dirancang untuk menangkap intensitas suara bersama dengan unit arah aliran sebagai besaran vektor. Hal ini dicapai dengan menggabungkan lebih dari satu mikrofon di probe untuk mengukur aliran energi suara. mikrofon konvensional dapat mengukur tekanan suara (unit: Pa), yang mewakili intensitas bunyi di tempat tertentu (satu titik), tetapi dapat mengukur arah aliran. Mikrofon intensitas bunyi Oleh karena itu digunakan untuk sumber suara memeriksa dan untuk mengukur kekuatan suara.

2.3 Definisi *Comparator Operational Amplifier*

Pada umumnya *Operational Amplifier* atau penguat operasional digunakan sebagai *Comparator* (Komparator) atau pembanding. Komparator adalah komponen elektronik yang berfungsi membandingkan dua nilai masukan kemudian memberikan hasil keluaran dari satu keadaan dimana keluaran yang lebih besar atau lebih kecil.

1. Komparator Catu Tunggal

Komparator catu tunggal adalah rangkaian pembandingan yang hanya menggunakan catuan positif. Tidak seperti halnya komperator yang lain menggunakan dua catuan , yaitu tegangan positif dan tegangan negatif



Gambar 2.1 : Konfigurasi Komparator Catu daya Tunggal

Penjelasan dari gambar diatas adalah apabila tegangan pada input inverting (V_{input}) lebih positif dari pada input non inverting (V_{ref}), maka outputnya akan sama dengan ($-V_{supply}$) dan bila tegangan pada input inverting (V_{input}) lebih negatif dari pada input non inverting (V_{ref}), maka outputnya akan sama dengan ($+V_{supply}$). V_{ref} di hubungkan ke $+V$ supply, kemudian $R1$ dan $R2$ digunakan sebagai pembagi tegangan, sehingga nilai tegangan yang di referensikan pada masukan $+ op-amp$ dapat dicari dengan persamaan berikut:

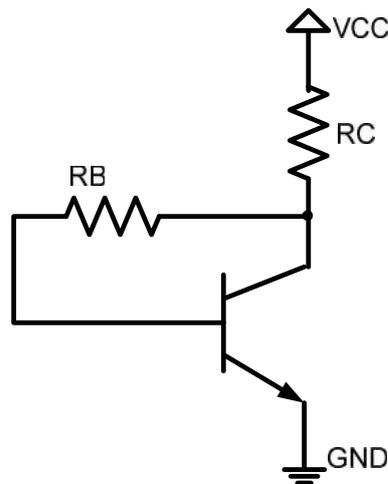
$$V_{ref} = \left(\frac{R1}{R1+R2} \right) \times (+V_{supply}) \quad (\text{Persamaan 2.1.Malvino.1991,hal.164})$$

2.4 *Preamplifier* Umpan Balik Kolektor

pada perancangan rangkaian *preamplifier* nanti, penulis disini menggunakan rangkaian *preamplifier* umpan balik kolektor, karena rangkaian prategangan umpan balik-kolektor memiliki beberapa kelebihan dibandingkan dengan rangkaian prategangan jenis lain, adapun kelebihan prategangan umpan-

balik kolektor yang penulis kutip dari buku prinsip-prinsip elektronika jilid1 (Malvino Barmawi, 1984, hal.149) yaitu :

- a. Transistor tidak dapat jenuh
- b. Tanggapan terhadap frekuensi yang lebih baik
- c. Tangkaian yang sederhana (hanya ada dua tahanan)



Gambar 2.2 : Rangkaian Prategangan Umpan-Balik Kolektor

(Sumber : Malvino Bermawi.1984)

Yang diperlukan dari rangkaian prategangan umpan-balik kolektor diatas agar dapat bekerja dengan baik adalah titik kerja Q yang pas dengan menggambar garis beban DC rangkain tersebut, adapun persamaan untuk rangkaian prategangan umpan balik kolektor yang penulis kutip dari buku Malvino Bermawi, Prinsip-Prinsip Konponen Elektronika jilid 1, yaitu :

$$R_b = \beta_{dc} \cdot R_c \quad (\text{Persamaan 2.2.Malvino,1984,hal.150})$$

$$I_c (\text{Max}) = \frac{V_{cc}}{R_c} \quad (\text{Persamaan 2.3.Malvino,1984,hal.127})$$

$$I_c = \frac{V_{cc} - V_{be}}{R_c + R_b / \beta_{dc}} \quad (\text{Persamaan 2.4.Malvino,1984,hal.127})$$

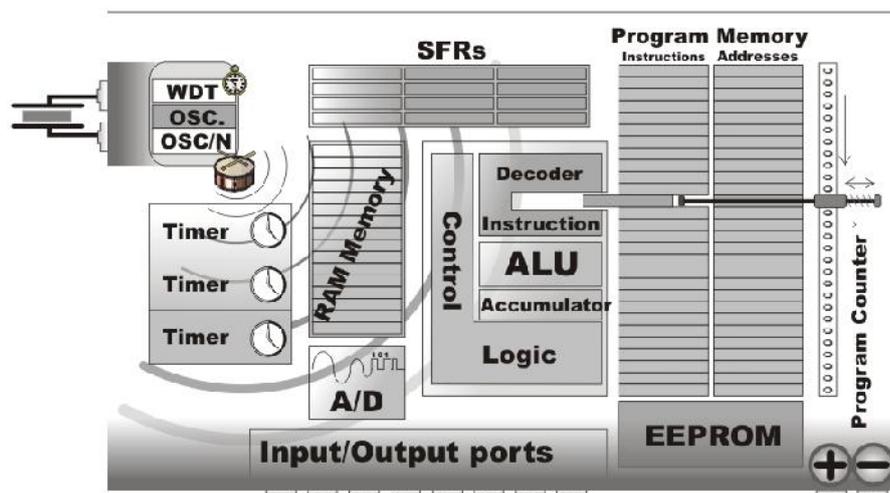
$$V_{ce} = V_{cc} \quad (\text{Persamaan 2.5.Malvino,1984,hal.127})$$

2.5 Definisi Mikrokontroler

Mikrokontroler merupakan suatu IC yang didalamnya berisi *CPU*, *ROM*, *RAM*, dan *I/O* yang dapat diprogram yang dapat disimpan didalam memory

sehingga dapat mengendalikan perintah secara otomatis. Mikrokontroler terdiri dari beberapa bagian yaitu :

1. *CPU (Central Processing Unit)*
2. *RAM (Random Access Memory)*
3. *EEPROM/EPROM/PROM*
4. *I/O, Serial & Parallel*
5. *Timer*
6. *Interrupt Controller*



Gambar 2.3: Struktur Dari Mikrokontroler

(Sumber : Hendrawan (2007). *Sistem Mikrokontroler*. Makalah Teknik Elektro Politeknik Batam)

1. *Central Processing Unit (CPU)*

CPU merupakan bagian utama dalam suatu mikrokontroler. *CPU* pada mikrokontroler ada yang berukuran 8 bit ada pula yang berukuran 16 bit. *CPU* ini akan membaca program yang tersimpan di dalam *ROM* dan melaksanakannya.

2. *Read Only Memory (ROM)*

ROM merupakan suatu memori (alat untuk mengingat) yang sifatnya hanya dibaca saja. Dengan demikian *ROM* tidak dapat ditulisi. Dalam dunia mikrokontroler *ROM* digunakan untuk menyimpan program bagi mikrokontroler

tersebut. Program tersimpan dalam format biner ('0' atau '1'). Susunan bilangan biner tersebut bila telah terbaca oleh mikrokontroler akan memiliki arti tersendiri.

3. *Random Acces Memory (RAM)*

Berbeda dengan *ROM*, *RAM* adalah jenis memori selain dapat dibaca juga dapat ditulis berulang kali. Tentunya dalam pemakaian mikrokontroler ada semacam data yang bisa berubah pada saat mikrokontroler tersebut bekerja. Perubahan data tersebut tentunya juga akan tersimpan ke dalam memori. Isi pada *RAM* akan hilang jika catu daya listrik hilang.

4. *Electrically Erasable Programmable Read Only Memory (EEPROM)*

Beberapa mikrokontroler memiliki *EEPROM* yang terintegrasi pada chipnya. *EEPROM* ini digunakan untuk menyimpan sejumlah kecil parameter yang dapat berubah dari waktu ke waktu. Jenis memori ini bekerja relatif pelan, dan kemampuan untuk dihapus/tulisnya juga terbatas.

5. *FLASH (EPROM)*

FLASH memberikan pemecahan yang lebih baik dari *EEPROM* ketika dibutuhkan sejumlah besar memori non-volatile untuk program. *FLASH* ini bekerja lebih cepat dan dapat dihapus/tulis lebih sering dibanding *EEPROM*.

6. *Input/Output*

- *UART (Universal Asynchronous Receiver Transmitter)* adalah adapter serial port adapter untuk komunikasi serial asinkron.
- *USART (Universal Synchronous/Asynchronous Receiver Transmitter)* merupakan adapter serial port untuk komunikasi serial sinkron dan asinkron. Komunikasi serial sinkron tidak memerlukan *start* atau *stop* bit dan dapat beroperasi pada clock yang lebih tinggi dibanding asinkron.
- *SPI (serial peripheral interface)* merupakan port komunikasi serial sinkron.
- *SCI (serial communications interface)* merupakan enhanced *UART* (asynchronous serial port)
- *I2C bus (Inter-Integrated Circuit bus)* merupakan antarmuka serial 2 kawat yang dikembangkan oleh Philips. Dikembangkan untuk aplikasi 8 bit dan banyak digunakan pada konsumen elektronik, otomotif dan industri. *I2C bus* ini berfungsi sebagai antarmuka jaringan multi-master, multi-slave dengan

deteksi tabrakan data. Jaringan dapat dipasangkan hingga 128 titik dalam jarak 10 meter. Setiap titik dalam jaringan dapat mengirim dan menerima data. Setiap titik dalam jaringan harus memiliki alamat yang unik.

- *ADC (Analog to Digital Conversion)* Fungsi *ADC* adalah merubah besaran analog (biasanya tegangan) ke bilangan digital. Mikrokontroler dengan fasilitas ini dapat digunakan untuk aplikasi-aplikasi yang memerlukan informasi analog (misalnya voltmeter, pengukur suhu dll). Terdapat beberapa tipe dari *ADC* yaitu *Succesive Approximation A/D converters*, *Single Slope A/D converters*, *Delta-Sigma A/Ds converters*, *Flash A/D*.
- *DAC (Digital to Analog Converters)* Kebalikan dari *ADC* yaitu merubah besaran digital (biasanya tegangan) ke bilangan analog.

7. *Interrupt Controller*

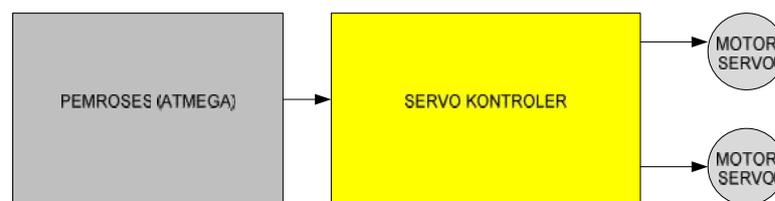
Interrupt merupakan metode yang efisien bagi mikrokontroler untuk memproses periperalnya, mikrokontroler hanya bekerja memproses periperall tersebut hanya pada saat terdapat data diperiperall tersebut. Pada saat terjadi interrupt, mikrokontroler menunda operasi yang sedang dilakukan kemudian mengidentifikasi interupsi yang datang dan menjalankan rutin pelayanan interupsi. Rata-rata mikrokontroler memiliki setidaknya-tidaknya sebuah interupsi eksternal, interupsi yang dimiliki bisa dipicu oleh "edge" atau "level". Edge triggered interrupt bekerja tidak tergantung pada waktu terjadinya interupsi, tetapi interupsi bisa terjadi karena glitch. Sedangkan Level triggered interrupt harus tetap pada logika high atau low sepanjang waktu tertentu agar dapat terjadi interupsi, interupsi ini tahan terhadap glitch.

8. *Timer*

Timer /Counter adalah suatu peripheral yang tertanam didalam microcontroller yang berfungsi pewaktu. Dengan peripheral ini pengguna microcontroller dapat dengan mudah menentukan kapan suatu perintah dijalankan (delay), tentu saja fungsi timer tidak hanya untuk penundaan perintah saja, timer juga dapat berfungsi sebagai oscilator, PWM, ADC, dan lain-lain.

2.6 Definisi Servo Kontroler

Servo Kontroler adalah sebuah perangkat yang menerima sinyal perintah dari sistem kontrol pemrosesan atau mikrokontroler, dan mengirimkan arus listrik ke motor servo untuk menghasilkan gerak sebanding dengan sinyal diperintahkan. Biasanya sinyal perintah merupakan kecepatan yang diinginkan dan juga dapat mewakili torsi atau posisi yang diinginkan. Servo controller memiliki tujuan utama yaitu untuk mempermudah dalam mengendalikan servo yang jumlahnya lebih dari satu, tujuan lain dari servo kontroler yaitu dapat memperkecil penggunaan port pada chip pemrosesan atau mikrokontroler, karena servo kontroler terhubung dengan cara I2C pada chip pemrosesan (mikrokontroler).



Gambar 2.4 : Struktur Diagram Servo Kontroler

2.6.1 Motor Servo

Motor servo adalah motor yang mampu bekerja dua arah (*CW* dan *CCW*) dimana arah dan sudut pergerakan rotornya dapat dikendalikan dengan memberikan variasi lebar pulsa (*duty cycle*) sinyal *PWM* pada bagian pin kontrolnya. Motor servo memiliki dua tipe yaitu servo standar dan servo *rotation* (*continuous*). Dimana biasanya untuk tipe standar hanya dapat melakukan pergerakan sebesar 180° sedangkan untuk tipe *continuous* dapat melakukan rotasi atau 360° .

Pada dasarnya motor servo tersusun dari motor DC, rangkaian kontrol, *gearbox* dan potensiometer. Tampak seperti gambar motor servo beserta komponen internal motor servo dibawah ini.



Gambar 2.5 : Bentuk Motor Servo (Kiri), Komponen Internal Motor Servo (kanan)

(Sumber : <http://Wikipediaindonesia.co.id/Motor-Servo/2012.html>. diakses pada tanggal 01 Juni 2014)

Terlihat jelas bahwa motor DC yang digunakan sangat kecil sehingga motor servo memiliki dimensi yang cukup kecil jika dibandingkan dengan motor DC pada umumnya. Rangkaian kontrol pada motor servo digunakan untuk mengontrol motor DC yang ada pada motor servo, dikarenakan untuk mengakses motor servo kita harus memberikan pulsa-pulsa kepada sinyal kontrol tersebut. Gearbox berfungsi untuk meningkatkan torsi dari motor servo, sebenarnya terdapat dua macam bahan penyusun *gearbox* yang digunakan untuk motor servo yaitu metal gear (biasanya untuk torsi yang sangat besar) dan nylon gear (berwarna putih seperti gambar diatas). Dan potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirimkan.

2.7 Definisi Bahasa C

Bahasa C pertama kali digunakan di komputer *Digital Equipment Corporation PDP-11* yang menggunakan sistem operasi *UNIX*. C adalah bahasa yang standar, artinya suatu program yang ditulis dengan bahasa C tertentu akan dapat dikonversi dengan bahasa C yang lain dengan sedikit modifikasi. Standar bahasa C yang asli adalah standar dari *UNIX*. Patokan dari standar *UNIX* ini diambil dari buku yang ditulis oleh Brian Kerninghan dan Dennis Ritchie berjudul "*The C Programming Language*", diterbitkan oleh Prentice-Hall tahun 1978. Deskripsi C dari Kerninghan dan Ritchie ini kemudian kemudian dikenal secara umum sebagai "K dan R C"

2.7.1 Penulisan Bahasa C

Program Bahasa C tidak mengenal aturan penulisan di kolom tertentu, jadi bias dimulai dari kolom manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan bahasa C diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut contoh penulisan Program Bahasa C:

```
#include <mega8535.h>
#include <delay.h>
main ()
{
.....
.....
}
```

Program dalam bahasa C selalu berbentuk fungsi seperti ditunjukkan dalam `main ()`. Program yang dijalankan berada di dalam tubuh program yang dimulai dengan tanda kurung buka `{` dan diakhiri dengan tanda kurung tutup `}`. Semua yang tertulis di dalam tubuh program ini disebut dengan blok.

Tanda `()` digunakan untuk mengapit argumen suatu fungsi. Argumen adalah suatu nilai yang akan digunakan dalam fungsi tersebut. Dalam fungsi `main` diatas tidak ada argumen, sehingga tak ada data dalam `()`. Dalam tubuh fungsi antara tanda `{` dan tanda `}` ada sejumlah pernyataan yang merupakan perintah yang harus dikerjakan oleh prosesor. Setiap pernyataan diakhiri dengan tanda titik koma `;`.

Baris pertama `#include <...>` bukanlah pernyataan, sehingga tak diakhiri dengan tanda titik koma `(;)`. Baris tersebut meminta kompiler 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*. File ini digunakan semacam perpustakaan bagi pernyataan yang ada di tubuh program.

`#include` merupakan salah satu jenis pengarah praprosesor (*preprocessor directive*). Pengarah praprosesor ini dipakai untuk membaca file yang di antaranya berisi deklarasi fungsi dan definisi konstanta. Beberapa file judul disediakan dalam C. File-file ini mempunyai ciri yaitu namanya diakhiri dengan ekstensi `.h`.

Misalnya pada program `#include <stdio.h>` menyatakan pada kompiler agar membaca file bernama `stdio.h` saat pelaksanaan kompilasi.

Bentuk umum `#include: #include "namafile"` Bentuk pertama (`#include <namafile>`) mengisyaratkan bahwa pencarian file dilakukan pada direktori khusus, yaitu direktori file `include`. Sedangkan bentuk kedua (`#include "namafile"`) menyatakan bahwa pencarian file dilakukan pertama kali pada direktori aktif tempat program sumber dan seandainya tidak ditemukan pencarian akan

dilanjutkan pada direktori lainnya yang sesuai dengan perintah pada sistem operasi.

2.7.2 Tipe Data

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap instruksi yang akan dilaksanakan oleh computer. Misalnya saja 5 dibagi 2 bisa saja menghasilkan hasil yang berbeda tergantung tipe datanya. Jika 5 dan 2 bertipe integer maka akan menghasilkan nilai 2, namun jika keduanya bertipe float maka akan menghasilkan nilai 2.5000000. Pemilihan tipe data yang tepat akan membuat proses operasi data menjadi lebih efisien dan efektif.

Tabel 2.1 : Tabel Penjelasan Tipe Data Bahasa C

No	Tipe Data	Ukuran	Range (Jangkauan)
1	Char	1 byte	-128 s/d 127
2	Int	2 byte	-32768 s/d 32767
3	Unsigned int	2 byte	0 s/d 65535
4	Long Int	4 byte	-2147483648 s/d 2147483648
5	Unsigned Long int	4 byte	0 s/d 4294967296
6	Float	4 byte	-3.4E-38 s/d 3.4E+38
7	Double	8 byte	1.7E-308 s/d 1.7E+308
8	Long Double	10 byte	3.4E-4932 s/d 1.1E+4932
9	Char	1 byte	-128 s/d 127
10	Unsigned char	1 byte	0 s/d 255

2.7.3 Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Konstanta nilainya selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai integer, pecahan, karakter dan string. Contoh konstanta : 50; 13; 3.14; 4.50005; 'A'; 'Bahasa C'.

2.7.4 Variabel

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

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C bersifat *case-sensitive* artinya huruf besar dan kecil dianggap berbeda.
2. Tidak boleh mengandung spasi.
3. Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk simbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, *, (,), -, +, = dsb
4. Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.

2.7.5 Deklarasi

Deklarasi diperlukan bila kita akan menggunakan pengenalan (identifier) dalam program. Identifier dapat berupa variabel, konstanta dan fungsi. Deklarasi terbagi 3 jenis, yaitu ;

1. Deklarasi variabel : bentuk umum pendeklarasian suatu variabel adalah : Nama_tipe nama_variabel; Contoh : `int x; // Deklarasi x bertipe integer.`
2. Deklarasi konstanta : dalam bahasa C konstanta dideklarasikan menggunakan preprocessor `#define`. Contohnya : `#define PHI 3.14 #define nim "0111500382"`
3. Deklarasi fungsi : fungsi merupakan bagian yang terpisah dari program dan dapat diaktifkan atau dipanggil di manapun di dalam program. Fungsi dalam

bahasa C ada yang sudah disediakan sebagai fungsi pustaka seperti *printf()*, *scanf()*, *getch()* dan untuk menggunakannya tidak perlu dideklarasikan. 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 : float luas_lingkar(int jari); void tampil(); int tambah(int x, int y);

2.7.6 Operator

Operator dalam bahasa C terbagi menjadi 7 jenis yaitu :

1. Operator penugasan : operator penugasan (*Assignment operator*) dalam bahasa C berupa tanda sama dengan (“=”).
2. Operator aritmatika : Bahasa C menyediakan lima operator aritmatika, yaitu :
 - a. `*` : untuk perkalian
 - b. `/` : untuk pembagian
 - c. `%` : untuk sisa pembagian (modulus)
 - d. `+` : untuk penjumlahan
 - e. `-` : untuk pengurangan
3. Operator hubungan : operator hubungan digunakan untuk membandingkan hubungan antara dua buah *operand* /sebuah nilai atau *variable*. Operasi majemuk seperti pada tabel dibawah ini:

Tabel 2.2 : Tabel Operator Hubungan

Operator	Arti	Contoh	
<	Kurang dari	X<Y	Apakah X kurang dari Y
<=	Kurang dari sama dengan	X<=Y	Apakah X Kurang dari sama dengan Y
>	Lebih dari	X>Y	Apakah X Lebih dari Y
>=	Lebih dari sama dengan	X==Y	Apakah X Lebih dari sama dengan Y
==	Sama dengan	X==Y	Apakah X Sama dengan Y
!=	Tidak sama dengan	X!= Y	Apakah X Tidak sama dengan Y

4. Operator logika : Jika operator hubungan membandingkan hubungan antara dua buah operand, maka operator logika digunakan untuk membandingkan

logika hasil dari operator-operator hubungan. Operator logika ada tiga macam, yaitu :

- a. `&&` : Logika AND (DAN)
- b. `||` : Logika OR (ATAU)
- c. `!` : 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. Sedangkan operasi NOT menghasilkan nilai benar jika ekspresinya bernilai salah, dan akan bernilai salah jika ekspresinya bernilai benar.

5. Operator bitwise digunakan untuk memanipulasi bit-bit dari nilai data yang ada di memori. Operator bitwise terbagi menjadi 8 jenis yaitu :
 - a. `<<` : Pergeseran bit ke kiri
 - b. `>>` : Pergeseran bit ke kanan
 - c. `&` : Bitwise AND
 - d. `^` : Bitwise XOR (exclusive OR)
 - e. `|` : Bitwise OR
 - f. `~` : Bitwise NOT
 - g. Pertukaran Nibble dan Byte
 - h. Mengambil Bit yang paling Berbobot
6. Operator Unary merupakan operator yang hanya membutuhkan satu operand saja. Dalam bahasa C terdapat beberapa operator unary seperti didalam table dibawah ini :
7. Operator majemuk : Operator majemuk terdiri dari dua operator yang digunakan untuk menyingkat penulisan. Operasi majemuk seperti pada tabel dibawah ini :

Tabel 2.3 : Tabel Operator Majemuk

Operator	Contoh	Kependekan Dari
<code>+=</code>	<code>Counter+=1</code>	<code>Counter = Counter +1</code>
<code>-=</code>	<code>Counter-=1</code>	<code>Counter = Counter -1</code>
<code>*=</code>	<code>Counter *=</code>	<code>Counter = Counter *1</code>
<code>/=</code>	<code>Counter /=1</code>	<code>Counter = Counter/1</code>

<code>%=</code>	<code>Counter%=1</code>	<code>Counter = Counter%1</code>
<code><<=</code>	<code>Counter<<=1</code>	<code>Counter = Counter<<1</code>
<code>>>=</code>	<code>Counter>>=1</code>	<code>Counter = Counter>>1</code>
<code>&=</code>	<code>Counter&=1</code>	<code>Counter = Counter&1</code>
<code> =</code>	<code>Counter =1</code>	<code>Counter = Counter 1</code>
<code>^=</code>	<code>Counter^=1</code>	<code>Counter = Counter^1</code>
<code>~=</code>	<code>Counter~=1</code>	<code>Counter = Counter~1</code>

2.7.7 Penyelesaian Kondisi

Penyeleksian kondisi digunakan untuk mengarahkan perjalanan suatu proses. Penyeleksian kondisi dapat diibaratkan sebagai katup atau kran yang mengatur jalannya air. Bila katup terbuka maka air akan mengalir dan sebaliknya bila katup tertutup air tidak akan mengalir atau akan mengalir melalui tempat lain. Fungsi penyeleksian kondisi penting artinya dalam penyusunan bahasa C, terutama untuk program yang kompleks. Penyeleksian kondisi terbagi menjadi 3 jenis, yaitu :

1. Struktur Kondisi “If”

Struktur *if* dibentuk dari pernyataan *if* dan sering digunakan untuk menyeleksi suatu kondisi tunggal. Bila proses yang diseleksi terpenuhi atau bernilai benar, maka pernyataan yang ada di dalam blok *if* akan diproses dan dikerjakan. Bentuk umum struktur kondisi *if* adalah :

if(kondisi)

{ pernyataan ; }

2. Struktur Kondisi “If, Else”

Dalam struktur kondisi *if....else* minimal terdapat dua pernyataan. Jika kondisi yang diperiksa bernilai benar atau terpenuhi maka pernyataan pertama yang dilaksanakan dan jika kondisi yang diperiksa bernilai salah maka pernyataan yang kedua yang dilaksanakan. Bentuk umumnya adalah sebagai berikut :

if(kondisi)

{ pernyataan-1 }

else

```
{ pernyataan-2 }
```

3. Struktur Kondisi “*Switch, Case, Default*”

Struktur kondisi *switch...case...default* digunakan untuk penyeleksian kondisi dengan kemungkinan yang terjadi cukup banyak. Struktur ini akan melaksanakan salah satu dari beberapa pernyataan ‘*case*’ tergantung nilai kondisi yang ada di dalam *switch*. Selanjutnya proses diteruskan hingga ditemukan pernyataan ‘*break*’. Jika tidak ada nilai pada *case* yang sesuai dengan nilai kondisi, maka proses akan diteruskan kepada pernyataan yang ada di bawah ‘*default*’. Bentuk umum dari struktur kondisi ini adalah :

switch(kondisi)

```
{
  case 1 : pernyataan-1;
  break;
  case 2 : pernyataan-2;
  break;
  ....
  case n : pernyataan-n;
  break;
  default : pernyataan-m
}
```

2.7.8 Perulangan

Dalam bahasa C tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan kita. Misalnya saja, bila kita ingin menginput dan mencetak bilangan dari 1 sampai 100 bahkan 1000, tentunya kita akan merasa kesulitan. Namun dengan struktur perulangan proses, kita tidak perlu menuliskan perintah sampai 100 atau 1000 kali, cukup dengan beberapa perintah saja. Struktur perulangan dalam bahasa C terbagi 3 macam, yaitu :

1. Struktur Perulangan “*While*”

Perulangan *while* banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar dan akan

berhenti bila kondisinya bernilai salah. Bentuk umum dari struktur kondisi ini adalah:

```

While (ekspresi)
{
    Pernyataan_1
    Pernyataan_2
}

```

2. Struktur Perulangan “*DO, While*”

Pada dasarnya struktur perulangan *do,while* sama saja dengan struktur *while*, hanya saja pada proses perulangan dengan *while*, seleksi berada di *while* yang letaknya di atas sementara pada perulangan *do....while*, seleksi *while* berada di bawah batas perulangan. Jadi dengan menggunakan struktur *do...while* sekurang-kurangnya akan terjadi satu kali perulangan. Bentuk umum dari struktur kondisi ini adalah:

```

Do
{
    Pernyataan_1
    Pernyataan_2
}
While (ekspresi)

```

3. Struktur Perulangan “*For*”

Struktur perulangan *for* biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan *for* tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Bentuk umum perulangan *for* adalah sebagai berikut :

```

for (inisialisasi; syarat; penambahan)
    pernyataan;

```

Keterangan:

Inisialisasi : pernyataan untuk menyatakan keadaan awal dari variabel kontrol.

syarat : ekspresi relasi yang menyatakan kondisi untuk keluar dari perulangan.

penambahan : pengatur perubahan nilai variabel kontrol.

2.7.9 Fungsi

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilmnya. Fungsi merupakan elemen utama dalam bahasa C karena bahasa C sendiri terbentuk dari kumpulan fungsi-fungsi. Dalam setiap program bahasa C, minimal terdapat satu fungsi yaitu fungsi `main()`. Fungsi banyak diterapkan dalam program-program C yang terstruktur. Keuntungan penggunaan fungsi dalam program yaitu program akan memiliki struktur yang jelas (mempunyai *readability* yang tinggi) dan juga akan menghindari penulisan bagian program yang sama.

```
Tipe_Nilai_Balik nama_fungsi(argumen1, argumen2)
{
    Pernyataan1;
    Pernyataan1;
    return(ekspresi);
}
```

2.8 Definisi Code Vision AVR

CodeVisionAVR pada dasarnya merupakan perangkat lunak pemrograman microcontroller keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: Compiler C, IDE dan Program generator.



Gambar 2.6: Logo Code Vision AVR

(Sumber : *Modul Training Mikrokontroler AVR* (2010). Workshop HME ITB)

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 LM75, 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 dapat digunakan untuk mentransfer kode mesin hasil kompilasi kedalam sistem memori mikrokontroler AVR yang sedang deprogram. CodeVisionAVR adalah suatu kompilator berbasis bahasa C, yang terintegrasi untuk memprogram dan sekaligus compiler aplikasi AVR (*Alf and Vegard's Risc processor*) terhadap mikrokontroler dengan sistem berbasis window. CodeVisionAVR ini dapat mengimplematasikan hampir semua interuksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifikasi dari CodeVisionAVR yaitu hasil kompilasi studio *debugger* dari ATMEL.

Integrated Development Environment (IDE) telah diadaptasikan pada chip AVR yaitu *In-System Programmer software*, memungkinkan programmer untuk mentransfer program ke chip mikrokontroler secara otomatis setelah proses assembly/kompilasi berhasil. *In-System Programmer software* didesign untuk bekerja dan dapat berjalan dengan perangkat lunak lain seperti AVR Dragon,

AVRISP, Atmel STK500, dan lain sebagainya. Disamping *library* standar C, CodeVisionAVR C compiler memiliki librari lain untuk:

1. Modul LCD *Alpanumerik*
2. *Delays*
3. *Protokol Semikonduktor Maxim/Dallas*, Dan lainnya.

CodeVisionAVR juga memiliki CodeWizardAVR sebagai generator program otomatis, yang memungkinkan kita untuk menulis, segala bentuk pengaturan Chip dalam waktu singkat, dan semua kode yang dibutuhkan untuk mengimplementasikan fungsi-fungsi seperti:

1. Pengaturan akses *External Memory*

Untuk chip-chip AVR yang memungkinkan koneksi memori eksternal SRAM, dapat juga mengatur ukuran memori dan wait state (tahap tunggu) dari memori ketika memori tersebut diakses.

2. Identifikasi *chip reset source*

Adalah suatu layanan dimana kita dapat membuat kode secara otomatis yang dapat mengidentifikasi kondisi yang menyebabkan *chip di reset*.

3. Inisialisasi *port input/output*

Pengaturan port-port yang kan dijadikan gerbang masukan dan keluaran dapat secara otomatis *digenerate* codenya. Yang kita lakukan hanya memilih *port-port* yang akan digunakan sebagai input atau output.

4. Inisialisasi *Interupsi external*

Pengaturan *interupsi eksternal* yang nantinya akan digunakan untuk menginterupsi program utama

5. Inisialisasi *timers/counters*

Pengaturan *timers* yang berfungsi untuk mengatur frekuensi yang nantinya digunakan pada interupsi.

6. Inisialisasi *timer watchdog*

Pengaturan *timers* yang berfungsi untuk mengatur frekuensi yang nantinya digunakan pada *interupsi*, sehingga *interupsi* akan dilayani oleh suatu fungsi *wdt_timeout_isr* .

7. Inisialisasi *UART (USART)* dan komunikasi serial

Pengaturan komunikasi serial sebagai penerima atau pengirim data.

8. Inisialisasi komparasi analog

Pengaturan yang berkaitan dengan masukan data yang digunakan dalam aplikasi yang membutuhkan komparasi pada ADC nya.

9. Inisialisasi *ADC*

Pengaturan *ADC (Analog-Digital Converter)* yang berfungsi untuk merubah format analog menjadi format digital untuk diolah lebih lanjut.

10. Inisialisasi antarmuka *SPI*

Pengaturan chip yang berkaitan dengan *Clock rate, Clock Phase*, dan lainnya.

11. Inisialisasi antarmuka *Two Wire BUS*

Pengaturan Chip yang berhubungan dengan pola jalur komunikasi antara register yang terdapat pada chip AVR.

12. Inisialisasi antarmuka *CAN*

Pengaturan chip yang lebih kompleks, yang dapat mengatur *interupsi*, transmisi data, *timers*, dan lainnya.

13. Inisialisasi sensor temperatur, thermometer, dan lainnya

Pengaturan yang berhubungan dengan sensor temperatur *one wire bus*, memiliki fungsi-fungsi yang ada pada librari CodeVisionAVR.

14. Inisialisasi *one wire bus*

Pengaturan yang berhubungan dengan sensor temperatur yang memiliki fungsi-fungsi yang ada pada librari CodeVisionAVR. Seperti Maxim/Dallas Semiconductor.

15. Inisialisasi modul *LCD*

Pengaturan port-port yang kan digunakan sebagai penghubung dengan *LCD alphanumeric*.

Untuk lebih jelas tampilan pengaturan yang disediakan oleh AVR dapat dilihat pada gambar dibawah ini :



Gambar 2.7 : Tampilan Pegaturan CodeVisionAVR

(Sumber : *Modul Training Mikrokontroler AVR* (2010). Workshop HME ITB)

2.9 Komponen – Komponen

Dalam pembuatan robot penari *humanoid* ini terdapat beberapa jenis komponen yang digunakan, komponen-komponen tersebut terdapat pada bagian : komponen sensor suara, komponen pemroses, dan komponen motor dan driver motor.

2.9.1 Komponen Sensor Suara

Sensor suara yang penulis rancang adalah menggunakan prinsip komparator yang berfungsi membandingkan sinyal output dari mikrofon dengan tegangan referensi, sehingga mendapatkan logika 1 atau 0 pada output komparator. Sensor suara tersebut tersusun dari beberapa komponen, seperti : mikrofon kondensator, transistor, kapasitor, resistor, dioda, dan IC LM358.

2.9.1.1 Mikrofon

Mikrofon adalah komponen elektronika dimana cara kerjanya yaitu membran yang digetarkan oleh gelombang suara akan menghasilkan sinyal listrik. Mic dapat diklasifikasikan menjadi beberapa jenis dasar antara lain; dinamis, carbon, dan kondensator.

1. Mikrofon Dinamis

Mikrofon dinamis adalah mikrofon yang menggunakan prinsip kerja induksi (mikrofon menjadi sumber listrik induksi). Prinsip kerja : Getaran suara yang masuk menggerakkan membran; getaran membran menggerakkan *moving coil*; getaran *moving coil* yang berada dalam membran magnet akan menyebabkan

timbulnya aliran listrik. Aliran listrik yang berupa gelombang listrik seirama dengan getaran suara yang diterima.

2. Mikrofon karbon

Mikrofon karbon adalah mikropon yang menggunakan prinsip kerja tahanan (resistansi) yang berubah-ubah, biasanya adalah resistor arang. Prinsip kerja : Getaran suara yang masuk menggetarkan membran. Getaran membran ini menyebabkan kerenggangan dan kerapatan arang berubah-ubah. Hal ini menyebabkan bervariasinya nilai resistansi arus listrik yang melewati kumparan primer. Arus listrik pada kumparan primer akan terinduksi pada gulungan sekunder dan besar kecilnya arus ini tergantung dari getaran membran yang disebabkan oleh getaran suara yang diterima.

3. Mikrofon Kondensor

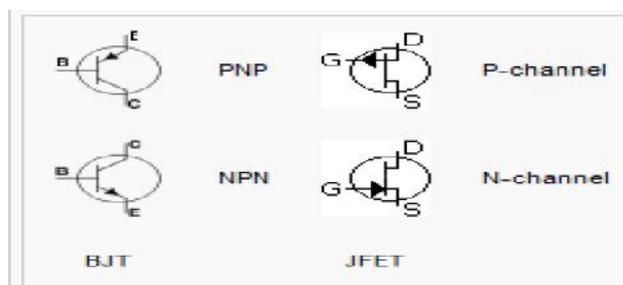
Mikrofon kondensor adalah mikropon yang dalam kerjanya menggunakan kondensator. Prinsip kerja : Getaran suara yang masuk menggetarkan membran. Getaran membran ini mengakibatkan gerakan maju dan mundur lempengan penghantar pada kondensator. Dengan perubahan ini, nilai kondensator pun berubah seiring dengan perubahan getaran. Perubahan kapasitansi ini menyebabkan terjadinya getaran listrik. Selanjutnya getaran listrik ini diperkuat oleh Preamp. Pada mikropon jenis ini memerlukan tegangan phantom dari preamp sebesar 48 volt, tetapi untuk aplikasi sehari-hari biasanya mikropon kondensor cukup menggunakan baterai 1,5 volt.

2.9.1.2 Transistor

Transistor adalah komponen semikonduktor yang dipakai sebagai penguat, sebagai sirkuit pemutus dan penyambung (switching), stabilisasi tegangan, modulasi sinyal atau sebagai fungsi lainnya. Transistor dapat berfungsi semacam kran listrik, dimana berdasarkan arus inputnya (BJT) atau tegangan inputnya (FET), memungkinkan pengaliran listrik yang sangat akurat dari sirkuit sumber listriknya.

Pada umumnya, transistor memiliki 3 terminal, yaitu Basis (B), Emitor (E) dan Kolektor (C). Tegangan yang di satu terminalnya misalnya Emitor dapat

dipakai untuk mengatur arus dan tegangan yang lebih besar daripada arus input Basis, yaitu pada keluaran tegangan dan arus output Kolektor. Transistor memiliki 2 tipe yaitu PNP dan NPN, dapat dilihat pada gambar di bawah ini:



Gambar 2.8 : Simbol Transistor

(Sumber : <http://id.wikipedia.org/wiki/Transistor>. Diakses tanggal 3 juni 2014.)

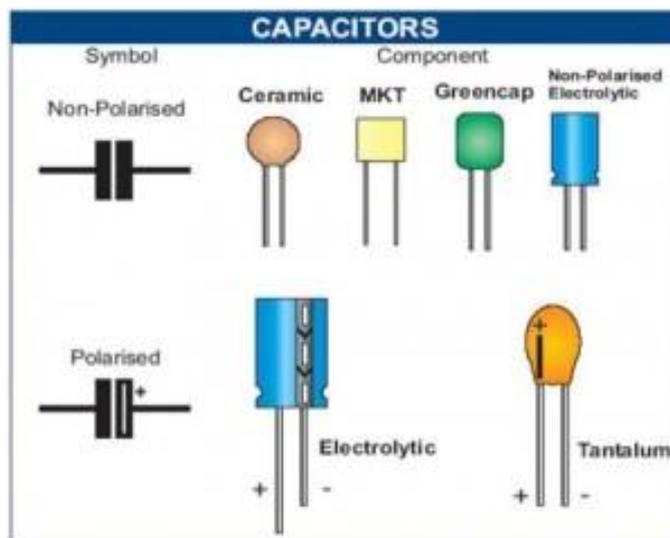
Menurut Owen : (2004:68) Bahwa semua transistor memiliki 3 buah kaki atau sambungan, transistor daya rendah dibuat dengan kemasan dari bahan plastik atau logam, kemasan yang terbuat dari bahan plastik ini memiliki salah satu sisi yang permukaan yang datar, sedangkan yang terbuat dari bahan logam memiliki sebuah tonjolan (*tag*) pada pinggiran bawah (*rim*), fitur fitur ini dimaksudkan untuk membantu pemakaian mengidentifikasi kaki-kaki terminal.

2.9.1.3 Kapasitor

Kapasitor adalah perangkat yang digunakan untuk menyimpan muatan listrik. Sebagai akibatnya, kapasitor merupakan suatu tempat penampungan (*reservoir*) di mana muatan dapat disimpan dan kemudian diambil kembali. Satuan kapasitansi adalah *farad* (F). Sebuah kapasitor dikatakan memiliki kapasitansi 1 F jika arus sebesar 1 A mengalir di dalamnya ketika tegangan yang berubah-ubah dengan kecepatan 1 V/s diberikan pada kapasitor tersebut. (Blocher, Richard. 2003:30) Arus yang mengalir di dalam sebuah kapasitor karenanya akan sebanding dengan hasil kali kapasitor (C) dengan kecepatan perubahan tegangan yang diberikan. Berdasarkan kapasitas dari suatu kapasitor, maka kapasitor dapat dibedakan dalam 2 jenis, yaitu :

1. Kapasitor Tetap

Kapasitor tetap adalah kapasitor yang memiliki Kapasitor tetap adalah kapasitor yang memiliki kapasitansi tetap dan tidak dapat diubah-ubah. Pada kategori kapasitor tetap, terdapat 2 jenis kapasitor yang dapat dibedakan berdasarkan polaritas elektrodanya.



Gambar 2.9: Simbol dan Jenis Kapasitor

(Sumber : <http://rangkaianelektronika.info/pengertian-fungsi-kapasitor/>. Diakses tanggal 6 juni 2014.)

a. Kapasitor Polar

Kelompok kapasitor elektrolit terdiri dari kapasitor-kapasitor yang bahan dielektriknya adalah lapisan metal-oksida. Umumnya kapasitor yang termasuk kelompok ini adalah kapasitor polar dengan tanda + dan – di badannya. Kapasitor ini dapat memiliki polaritas, adalah karena proses pembuatannya menggunakan elektrolisa sehingga terbentuk kutup positif anoda dan kutup negatif katoda.



Gambar 2.10 : Kapasitor Elektrolit

(Sumber : <http://rangkaianelektronika.info/pengertian-fungsi-kapasitor/>. Diakses tanggal 6 juni 2014.)

b. Kapasitor Non-Polar



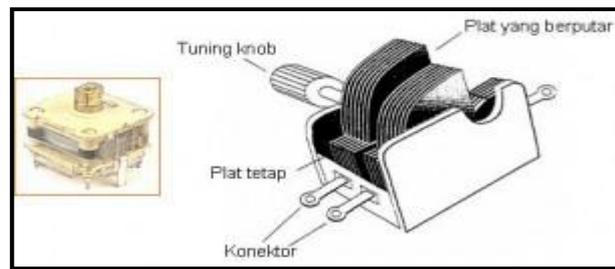
Gambar 2.11: Kapasitor Non-Polar

(Sumber : <http://rangkaianelektronika.info/pengertian-fungsi-kapasitor/>. Diakses tanggal 6 juni 2014.)

Kapasitor non polar adalah kelompok kapasitor yang dibuat dengan bahan dielektrik dari keramik, film dan mika. Keramik dan mika adalah bahan yang populer serta murah untuk membuat kapasitor yang kapasitansinya kecil. Tersedia dari besaran pF sampai beberapa uF, yang biasanya untuk aplikasi rangkaian yang berkenaan dengan frekuensi tinggi. Termasuk kelompok bahan dielektrik film adalah bahan-bahan material seperti *polyester* (*polyethylene terephthalate* atau dikenal dengan sebutan *mylar*), *polystyrene*, *polypropylene*, *polycarbonate*, *metalized paper* dan lainnya.

2. Kapasitor Tidak Tetap / Kapasitor Variabel

Kapasitor tidak tetap atau kapasitor variabel adalah kapasitor yang nilai kapasitansinya dapat diubah atau kapasitansinya dapat diatur sesuai keinginan dengan batas maksimal sesuai yang tertera pada kapasitor tersebut. Contoh suatu kapasitor variabel (*Varco* atau *trimmer* kapasitor) tertulis 100pF maka kapasitansi kapasitor tersebut dapat diatur maksimal 100pF sampai mendekati 0 pF.



Gambar 2.12 : Kapasitor Variabel

(Sumber : <http://rangkaianelektronika.info/pengertian-fungsi-kapasitor/>. Diakses tanggal 6 juni 2014.)

Aplikasi dari kapasitor variabel ini dapat ditemukan pada rangkaian penerima radio atau pembangkit gelombang, kapasitor variabel ini juga dapat ditemui pada pemancar radio. Fungsi kapasitor variabel ini pada rangkaian tersebut adalah untuk mengatur nilai frekuensi resonansi yang dihasilkan dari rangkaian pembangkit gelombang, dan sebagai trimer impedansi pemancar dan antena pada pemancar radio.

2.9.1.4 IC LM358

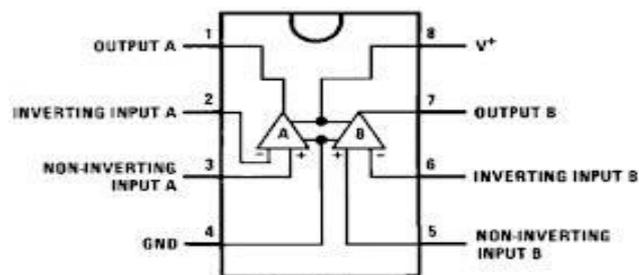
LM 358 merupakan rangkaian terintegrasi yang memiliki dua penguat operasional. Terdiri dari 4 masukan, memiliki faktor penguatan yang besar dan frekuensi internal yang berubah-ubah, yang mana di desain secara spesifik untuk beroperasi dari sebuah power supply melalui sebuah range tegangan. IC ini memiliki spesifikasi sebagai berikut :

1. Frekuensi internal yang dapat di ubah untuk penguatannya.
2. Penguatan tegangan yang besar (100dB).
3. Memiliki besar range tegangan antara 3V-32V.
4. Arus bias input rendah (20nA).
5. Arus offset input rendah (2nA).
6. Tegangan offset input rendah (2mV).
7. Tegangan output besar, berkisar 0 sampai (Vcc-1,5V).



Gambar 2.13 : Bentuk Fisik IC LM358

(Sumber : Datasheet LM358. 2013. *Semiconductor Componen Industries*)



Gambar 2.14: Diagram Penguat IC LM358

(Sumber : Datasheet LM358. 2013. *Semiconductor Componen Industries*)

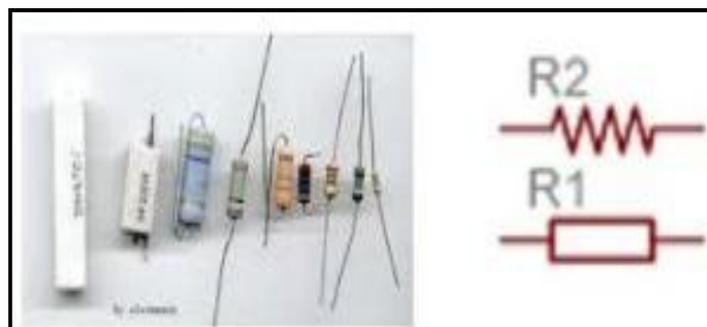
2.9.1.5 Resistor

Resistor merupakan sarana untuk mengontrol atau menahan arus dan tegangan yang bekerja dalam rangkaian-rangkaian elektronik. Spesifikasi untuk suatu resistor umumnya meliputi nilai resistansi (dinyatakan dalam ohm (Ω), kilohm (k Ω) atau megohm (M Ω), nilai ketepatan atau toleransi (dinyatakan sebagai penyimpangan maksimum yang diizinkan dari nilai yang tertera), dan rating daya (yang harus sama dengan atau lebih besar daripada disipasi daya maksimumnya) (Tooley, Michael. 2003:19).

Adapun fungsi dari resistor, yaitu menahan sebagian arus listrik agar sesuai dengan kebutuhan suatu rangkaian elektronika, menurunkan tegangan sesuai dengan yang dibutuhkan oleh rangkaian elektronika, membagi tegangan, bekerja sama dengan transistor dan kondensator dalam suatu rangkaian untuk membangkitkan frekuensi tinggi dan frekuensi rendah. Dilihat dari fungsinya, resistor dapat dibagi menjadi :

1. Resistor Tetap (*Fixed Resistor*)

Yaitu resistor yang nilainya tidak dapat berubah, jadi selalu tetap (*konstan*). Resistor ini biasanya dibuat dari karbon. Berfungsi sebagai pembagi tegangan, mengatur atau membatasi arus pada suatu rangkaian serta memperbesar dan memperkecil tegangan.



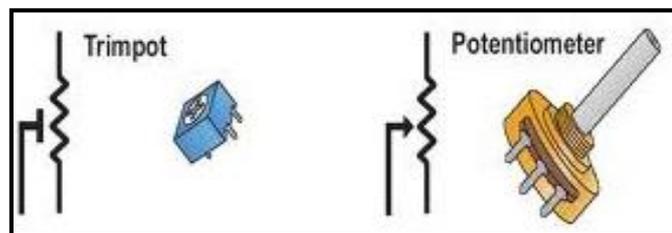
Gambar 2.15: Bentuk Fisik dan Lambang *Fixed Resistor*

(Sumber : <http://rangkaianelektronika.info/fungsi-dan-jenis-jenis-resistor/>).

Diakses tanggal 6 juni 2014.

2. Resistor Tidak Tetap (*Variable Resistor*)

Yaitu resistor yang nilainya dapat berubah-ubah dengan jalan menggeser atau memutar toggle pada alat tersebut, sehingga nilai resistor dapat kita tetapkan sesuai dengan kebutuhan. Berfungsi sebagai pengatur *volume* (mengatur besar kecilnya arus), tone control pada *sound system*, pengatur tinggi rendahnya nada (*bass* atau *treble*) serta berfungsi sebagai pembagi tegangan arus dan tegangan.



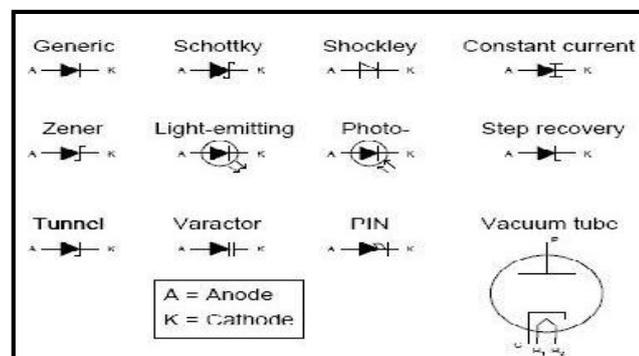
Gambar 2.16 Bentuk Fisik dan Lambang Variable Resistor

(Sumber : <http://rangkaiaelektronika.info/fungsi-dan-jenis-jenis-resistor/>).

Diakses tanggal 6 juni 2014.

2.9.1.6 Dioda

Dioda adalah semikonduktor yang terdiri dari persambungan (*junction*) P N. Sifat dioda yaitu dapat menghantarkan arus pada tegangan maju dan menghambat arus pada tegangan balik. (Sabrina. 2010:1)



Gambar 2.17: Simbol Dioda

(Sumber : Sabrina. 2010:1 <http://abisabrina.wordpress.com/2010/07/14/komponen-dasar-elektronika-induktor/> diakses pada tanggal 03 Juni 2014)

Dioda memiliki banyak yaitu :

1. Penyearah, contoh : *dioda bridge*
2. Penstabil tegangan (*voltage regulator*), yaitu dioda zener
3. Pengaman atau sekering
4. Sebagai rangkaian clipper, yaitu untuk memangkas atau membuang level sinyal yang ada di atas atau di bawah level tegangan tertentu.
5. Sebagai rangkaian *clamper*, yaitu untuk menambahkan komponen dc kepada suatu sinyal ac
6. Pengganda tegangan.
7. Sebagai indikator, yaitu LED (*light emitting diode*)
8. Sebagai sensor panas, contoh aplikasi pada rangkaian *power amplifier*
9. Sebagai sensor cahaya, yaitu dioda photo
10. Sebagai rangkaian VCO (*voltage controlled oscillator*), yaitu dioda *varactor*

Dari beragam fungsi diatas dioda terbagi dalam beberapa jenis yaitu :

1. Dioda Standar

Dioda jenis ini ada dua macam yaitu silikon dan germanium. *Dioda silicon* mempunyai tegangan maju 0.6V sedangkan dioda germanium 0.3V. Dioda jenis ini mempunyai beberapa batasan tertentu tergantung spesifikasi. Batasan batasan itu seperti batasan tegangan *reverse*, frekuensi, arus, dan suhu. Tegangan maju dari dioda akan turun 0.025V setiap kenaikan 1 derajat dari suhu normal. Sesuai karakteristiknya dioda ini bisa dipakai untuk fungsi-fungsi sebagai berikut:

- a. Penyearah sinyal AC
- b. Pemotong level
- c. Sensor suhu
- d. Penurun tegangan
- e. Pengaman polaritas terbalik pada dc input

Contoh dioda jenis ini adalah 1N400x (1A), 1N5392 (1.5A), dan 1N4148 (500mA).

2. LED (*Light Emitting Diode*)

Dioda jenis ini mempunyai lapisan fosfor yang bisa memancarkan cahaya saat diberi polaritas pada kedua kutubnya. LED mempunyai batasan arus

maksimal yang mengalir melaluinya. Diatas nilai tersebut dipastikan umur led tidak lama. Jenis led ditentukan oleh cahaya yang dipancarkan. Seperti led merah, hijau, biru, kuning, oranye, infra merah dan laser diode. Selain sebagai indikator beberapa LED mempunyai fungsi khusus seperti LED inframerah yang dipakai untuk transmisi pada sistem *remote control* dan opto sensor juga *laser diode* yang dipakai untuk *optical pick-up* pada sistem CD. Dioda jenis ini dibias maju (*forward*).

3. Dioda Zener

Fungsi dari dioda zener adalah sebagai penstabil tegangan. Selain itu dioda zener juga dapat dipakai sebagai pembatas tegangan pada level tertentu untuk keamanan rangkaian. Karena kemampuan arusnya yang kecil maka pada penggunaan dioda zener sebagai penstabil tegangan untuk arus besar diperlukan sebuah *buffer* arus. Dioda zener dibias mundur (*reverse*).

4. Dioda Photo

Dioda photo merupakan jenis komponen peka cahaya. Dioda ini akan menghantar jika ada cahaya yang mauk dengan intensitas tertentu. aplikasi dioda photo banyak pada sistem sensor cahaya (*optical*). Contoh pada *optocoupler* dan *optical pick-up* pada sistem CD. Dioda *photo* dibias maju (*forward*).

5. Dioda Varactor

Kelebihan dari dioda ini adalah mampu menghasilkan nilai kapasitansi tertentu sesuai dengan besar tegangan yang diberikan kepadanya. Dengan dioda ini maka sistem penalaan digital pada sistem transmisi frekuensi tinggi mengalami kemajuan pesat, seperti pada radio dan televisi. Contoh sistem penalaan dengan dioda ini adalah dengan sistem PLL (*Phase lock loop*), yaitu mengoreksi *oscilator* dengan membaca penyimpangan frekuensinya untuk kemudian diolah menjadi tegangan koreksi untuk *oscilator*. Dioda varactor dibias *reverse* (Sabrina.2010:1).

2.9.2 Komponen Mikrokontroler

Mikrokontroler merupakan tempat pemrosesan input sehingga menghasilkan output yang dapat bekerja secara otomatis berdasarkan program yang telah dituliskan dan tersimpan pada memori chip mikro. Adapun komponen

pendukung pada mikrokontroler yang penulis gunakan yaitu : IC ATmega 8535, dan Kristal.

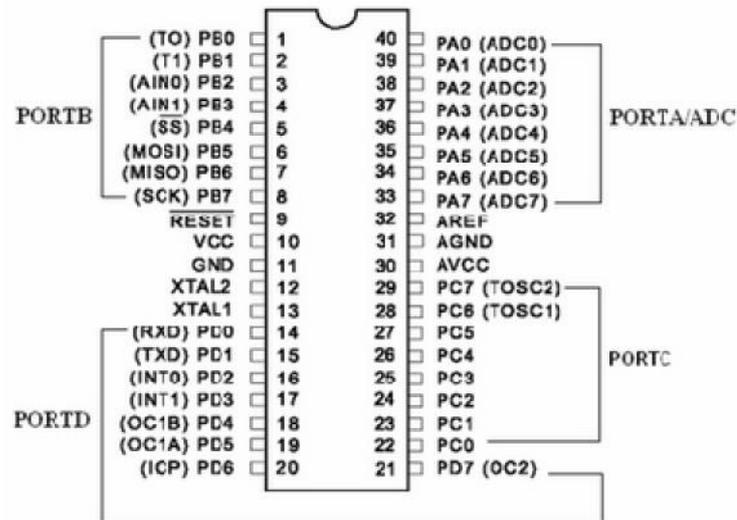
2.9.2.1 IC ATMEGA 8535

ATMega8535 adalah mikrokontroler CMOS 8 *bit* daya rendah berbasis arsitektur RISC. Instruksi dikerjakan pada satu siklus *clock*, ATMega8535 mempunyai *throughput* mendekati 1 MIPS per MHz, hal ini membuat ATMega8535 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah. Mikrokontroler ATmega8535 memiliki beberapa fitur atau spesifikasi yang menjadikannya sebuah solusi pengendali yang efektif untuk berbagai keperluan. Fitur-fitur tersebut antara lain:

1. Saluran I/O sebanyak 32 buah, yang terdiri atas *Port A, B, C* dan *D*
2. ADC (*Analog to Digital Converter*)
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan
4. CPU yang terdiri atas 32 *register*
5. *Watchdog Timer* dengan *osilator internal*
6. SRAM sebesar 512 *byte*
7. Memori *Flash* sebesar 8kb dengan kemampuan *read while write*
8. Unit Interupsi *Internal* dan *External*
9. *Port* antarmuka SPI untuk men-*download* program ke *flash*
10. EEPROM sebesar 512 *byte* yang dapat diprogram saat operasi
11. Antarmuka komparator *analog*
12. *Port* USART untuk komunikasi serial.

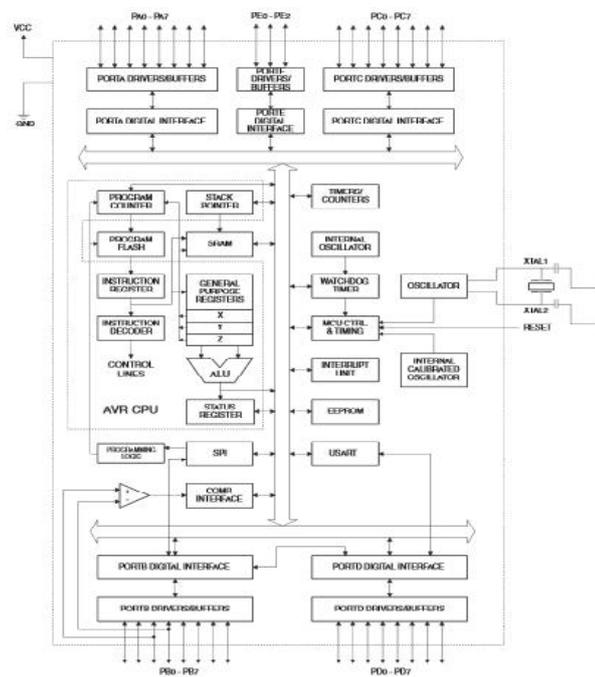
1. Konfigurasi *Pin* ATMega8535

Mikrokontroler AVR ATMega memiliki 40 *pin* dengan 32 *pin* diantaranya digunakan sebagai *port paralel*. Satu *port paralel* terdiri dari 8 *pin*, sehingga jumlah *port* pada mikrokontroler adalah 4 *port*, yaitu *port A, port B, port C* dan *port D*. Sebagai contoh adalah *port A* memiliki *pin* antara *port A.0* sampai dengan *port A.7*, demikian selanjutnya untuk *port B, port C, port D*. Diagram *pin* mikrokontroler dapat dilihat pada gambar berikut:



Gambar 2.18 : Diagram Pin ATMega8535

(Sumber : Hendrawan (2007). *Sistem Mikrokontroler*. Makalah Teknik Elektro Politeknik Batam)



Gambar 2.19 : Block diagram ATMEGA8535

(Sumber : Hendrawan (2007). *Sistem Mikrokontroler*. Makalah Teknik Elektro Politeknik Batam)

Berikut ini adalah tabel penjelasan mengenai pin yang terdapat pada mikrokontroler ATmega8535:

Tabel 2.4 : Tabel Penjelasan pin pada mikrokontroler ATmega8535

Vcc	Tegangan suplai (5 volt)
GND	Ground
RESET	Input reset level rendah, pada pin ini selama lebih dari panjang pulsa <i>minimum</i> akan menghasilkan reset walaupun clock sedang berjalan. RST pada pin 9 merupakan reset dari AVR. Jika pada pin ini diberi masukan <i>low</i> selama minimal 2 machine cycle maka sistem akan di-reset
XTAL 1	Input penguat osilator inverting dan input pada rangkaian operasi clock internal
XTAL 2	Output dari penguat osilator inverting
Avcc	Pin tegangan suplai untuk port A dan ADC. Pin ini harus dihubungkan ke Vcc walaupun ADC tidak digunakan, maka pin ini harus dihubungkan ke Vcc melalui <i>low pass filter</i>
Aref	pin referensi tegangan analog untuk ADC
AGND	pin untuk analog ground. Hubungkan kaki ini ke GND, kecuali jika board memiliki analog ground yang terpisah

Berikut ini adalah penjelasan dari pin mikrokontroler ATmega8535 menurut port-nya masing-masing:

A. Port A

Pin33 sampai dengan pin 40 merupakan pin dari port A. Merupakan 8 bit directional port I/O. Setiap pin-nya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer port A dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port A (DDRA) harus di-setting terlebih dahulu sebelum port A digunakan. Bit-bit DDRA diisi 0 jika ingin memfungsikan pin-pin port A yang disesuaikan sebagai

input, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin* pada *port A* juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel:

Tabel 2.5 : Tabel Penjelasan pin pada port A

<i>Pin</i>	Keterangan
PA.7	ADC7 (ADC <i>Input Channel</i> 7)
PA.6	ADC6 (ADC <i>Input Channel</i> 6)
PA.5	ADC7 (ADC <i>Input Channel</i> 5)
PA.5	ADC4 (ADC <i>Input Channel</i> 4)
PA.3	ADC3 (ADC <i>Input Channel</i> 3)
PA.2	ADC2 (ADC <i>Input Channel</i> 2)
PA.1	ADC1 (ADC <i>Input Channel</i> 1)
PA.0	ADC0 (ADC <i>Input Channel</i> 0)

B. *Port B*

Pin 1 sampai dengan *pin* 8 merupakan *pin* dari *port B*. Merupakan 8 bit *directional port I/O*. Setiap *pin*-nya dapat menyediakan *internal pull-up resistor* (dapat diatur per *bit*). *Output buffer port B* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port B* (DDRB) harus di-*setting* terlebih dahulu sebelum *port B* digunakan. *Bit-bit* DDRB diisi 0 jika ingin memfungsikan *pin-pin port B* yang disesuaikan sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin port B* juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel:

Tabel 2.6 : Tabel Penjelasan pin pada port B

<i>Pin</i>	Keterangan
PB.7	SCK (SPI <i>Bus Serial Clock</i>)

PB.6	VISO (SPI Bus Master Input/Slave Output)
PB.5	VOSI (SPI Bus Master Output/Slave Input)
PB.4	SS (SPI Slave Select Input)
PB.3	AIN1 (Analog Comparator Negative Input)OCC (Timer/Counter0 Output Compare Match Output)
PB.2	AIN0 (Analog Comparator Positive Input)INT2 (External Interrupt2 Input)
PB.1	T1 (Timer/Counter1 External Counter Input)
PB.0	T0 (Timer/Counter0 External Counter Input)XCK (JSART External Clock Input/Output)

C. Port C

Pin 22 sampai dengan *pin 29* merupakan *pin* dari *port C*. *Port C* sendiri merupakan *port input* atau *output*. Setiap *pin*-nya dapat menyediakan *internal pull-up resistor* (dapat diatur per *bit*). *Output buffer port C* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port C* (DDRC) harus di-*setting* terlebih dahulu sebelum *port C* digunakan. *Bit-bit* DDRC diisi 0 jika ingin memfungsikan *pin-pin port C* yang disesuaikan sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin port D* juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel II.6:

Tabel 2.7 : Tabel Penjelasan pin pada port C

<i>Pin</i>	Keterangan
PC.7	TOSC2 (Timer Oscillator Pin 2)
PC.6	TOSC1 (Timer Oscillator Pin 1)
PC.1	SDA (Two-Wire Serial Bus Data Input/Output Line)
PC.0	SCL (Two-Wire Serial Bus Clock Line)

4. Port D

Pin 14 sampai dengan *pin* 20 merupakan *pin* dari port D. Merupakan 8 bit *directional port I/O*. Setiap *pin*-nya dapat menyediakan *internal pull-up resistor* (dapat diatur per *bit*). *Output buffer port D* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port D* (DDRD) harus di-*setting* terlebih dahulu sebelum port D digunakan. *Bit-bit* DDRD diisi 0 jika ingin memfungsikan *pin-pin port D* yang disesuaikan sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin port D* juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel:

Tabel 2.8 : Tabel Penjelasan pin pada port D

<i>Pin</i>	Keterangan
PD.0	RDX (<i>UART input line</i>)
PD.1	TDX (<i>UART output line</i>)
PD.2	INT0 (<i>external interrupt 0 input</i>)
PD.3	INT1 (<i>external interrupt 1 input</i>)
PD.4	OC1B (<i>Timer/Counter1 output compareB match output</i>)
PD.5	OC1A (<i>Timer/Counter1 output compareA match output</i>)
PD.6	ICP (<i>Timer/Counter1 input capture pin</i>)
PD.7	OC2 (<i>Timer/Counter2 output compare match output</i>)

2.9.2.2 Kristal

Kristal lazimnya digunakan untuk rangkaian osilator yang menuntut stabilitas frekuensi yang tinggi dalam jangka waktu yang panjang. Alasan utamanya adalah karena perubahan nilai frekuensi kristal seiring dengan waktu, atau disebut juga dengan istilah faktor penuaan frekuensi (*frequency aging*), jauh lebih kecil dari pada osilator-osilator lain



Gambar 2.20 : Simbol Kristal (Kiri), Bentuk fisik Kristal (Kanan)
 (Sumber : <http://www.ElektronikaOnline.com/> diakses pada tanggal 28 Mei 2014)

2.9.3 Komponen Servo Kontroler Dan Motor Servo

Komponen yang penulis gunakan adalah servo kontroler merek SPC dan motor servo yang digunakan adalah merek towerpro MG996. Adapun spesifikasi masing masing komponen yaitu:

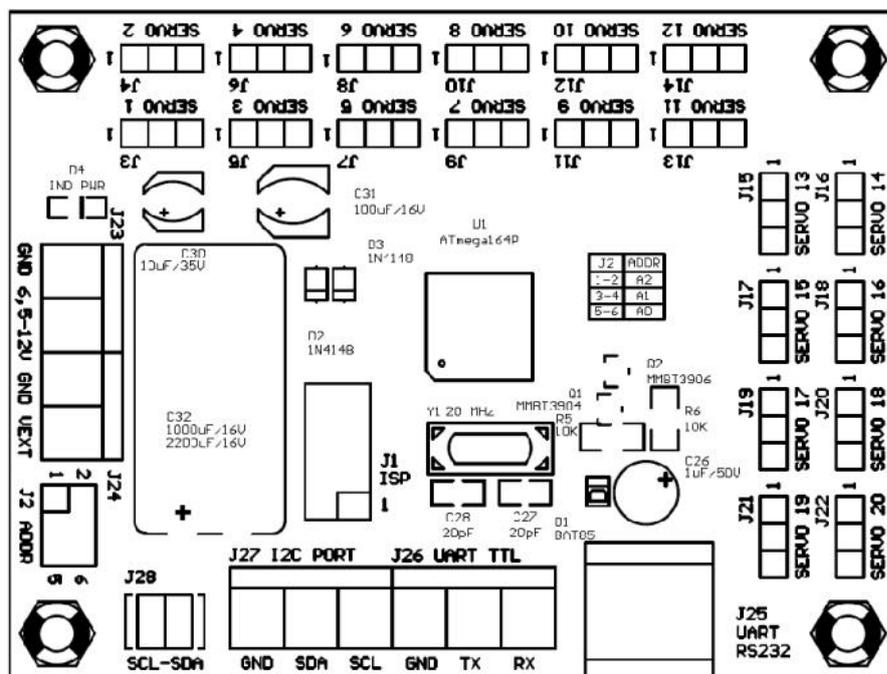
2.9.3.1 SPC Servo Motor *Kontroller*

Servo kontroler merupakan perangkat yang mempermudah dalam pengontrolan servo yang lebih dari satu, selain itu penggunaan servo kontroler dapat memper kecil penggunaan port pada mikrokontroler. Disini penulis menggunakan perangkat servo kontroler dengan merek SPC Servo Kontroller, buatan inovative elektronik, servo kontroler ini dapat berkomunikasi melalui I2C dan Serial, servo kontroler ini dilengkapi dengan 20 chanel output untuk mengontrol servo.

Spesifikasi SPC SERVO MOTOR CONTROLLER sebagai berikut:

1. Mampu mengendalikan hingga 20 motor servo secara serentak maupun sekuensial.
2. Mampu mengendalikan motor servo standar maupun kontinu.
3. Catu daya untuk SPC SERVO MOTOR CONTROLLER terpisah dengan catu daya untuk motor servo.
4. Catu daya untuk SPC SERVO MOTOR CONTROLLER dapat diperoleh dari sumber catu daya dengan tegangan 6,5 – 12 Volt.

5. Tiap modul SPC mampu mengendalikan 20 motor servo.
6. Resolusi pulsa kontrol servo sebesar 1 μ s.
7. Dilengkapi dengan kemampuan servo *ramping*.
8. Dilengkapi dengan kemampuan membaca pulsa kontrol (posisi) servo.
9. Dilengkapi dengan kemampuan *Enable* dan *Disable* servo.
10. Dilengkapi dengan kemampuan menyimpan dan menjalankan sampai dengan maksimal 32 sekuen gerakan.
11. Dilengkapi dengan kemampuan menyimpan dan kembali ke posisi *home* (*default*).
12. Dilengkapi dengan kemampuan menyimpan sekuen gerakan yang melibatkan beberapa motor servo.
13. Tersedia antarmuka UART RS-232, UART TTL, dan I2C.
14. Jika menggunakan I2C, SPC SERVO MOTOR CONTROLLER dapat di-*cascade* hingga 8 modul.



Gambar 2.21 : Tata Letak SPC Servo Motor Kontroller

(Sumber : Manual Book SPC Servo Motor Kontroller.2011.Inovative-Elektronik)

2.9.3.2 Towerpro MG996

Servo towerpro mg996 merupakan jenis motor servo DC standar yang dapat beroperasi 180° dari sudut -90° sampai dengan 90° derajat, adapun spesifikasi motor servo yaitu:

1. Jenis servo *standard*
2. Torsi :
4,8V dengan torsi 9.4 Kg
6V deengan torsi 11Kg
3. Kecepatan
4,8V = 0,17 sec/60°
6.0V = 0,14 sec/60°
4. Berat = 55,0 g
5. Dimensi
Lebar = 19,8 mm
Tinggi = 42,9 mm
Panjang =40,6 mm
4. Pulsa kontrol :
-90° dengan pulsa 600u sec
0° dengan pulsa 1500u sec
+90° dengan pulsa 2400u sec



Gambar 2.22 : Bentuk Fisik Servo Towerpro Mg996
(Sumber : [http:// Datasheet Servo Towerpro MG996.2007](http://Datasheet%20Servo%20Towerpro%20MG996.2007))