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 reprogammable 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 terkendali, 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:

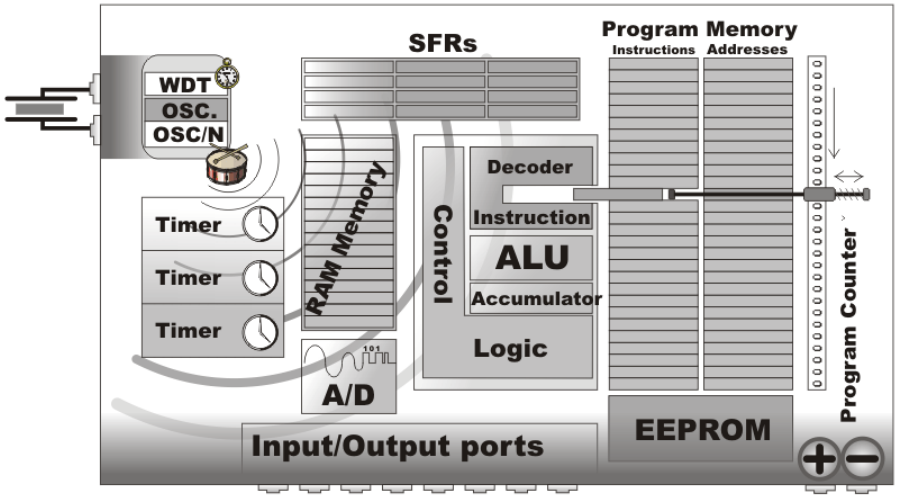
1. Dapat memperoleh informasi dari lingkungan (melalui sensor).
2. Dapat diprogram.
3. Dapat melaksanakan beberapa tugas yang berbeda.
4. Bekerja secara otomatis
5. Cerdas (*intelligent*)
6. Digunakan di industri. (*Membuat Robot Itu Gampang,* hal 5, 2006)

2.2 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. *Interupt Controller*



Gambar 2.1: 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 dalm 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 dugunakan untuk menyimpan sejumlah kecil parameter yang dapat berubah dari waktu ke waktu. Jenis memori ini bekerja relatif pelan, dan kemampuan untuk dihapus/tulis nya juga terbatas.

5. *FLASH (EPROM)*

FLASH meberikan 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*

a. *UART* (*Universal Asynchronous Receiver Transmitter*) adalah adapter serial port adapter untuk komunikasi serial asinkron.
b. *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 click yang lebih tinggi dibanding asinkron.
c. *SPI* (*serial peripheral interface*) merupakan port komunikasi serial sinkron.
d. *SCI* (*serial communications interface*) merupakan enhanced *UART* (asynchronous serial port)
e. *I2C* bus (*Inter-Integrated Circuit bus*) merupakan antarmuka serial 2 kawat yang dikembangkan oleh Philips. Dikembangkan untuk aplikasi 8 bit dan banyak digunakan pada konsumem elektronik, otomotif dan indistri. *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.
f. *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.*

g. *DAC* (*Digital to Analog Converters*) Kebalikan dar *ADC* yaitu merubah besaran digital (biasanya tegangan) ke bilangan analog.

7. *Interupt Controller*

Interupt merupakan metode yang efisien bagi mikrokontroler untuk memproses periperalnya, mikrokontroler hanya bekerja memproses periperal tersebut hanya pada saat terdapat data diperiperal tersebut. Pada saat terjadi interupt, mikrokontroler menunda operasi yang sedang dilakukan kemudian mengidentifikasi interupsi yang datang dan menjalankan rutin pelayanan interupsi. Rata-rata mikrokontroler memiliki setidak-tidaknya sebuah interupsi eksternal, interupsi yang dimiliki bisa dipicu oleh "edge" atau "level". Edge triggered interupt bekerja tidak tergantung pada pada waktu terjadinya interupsi, tetapi interupsi bisa terjadi karena glitch. Sedangkan Level triggered interupt 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.

2.2.1 IC ATMEGA 32

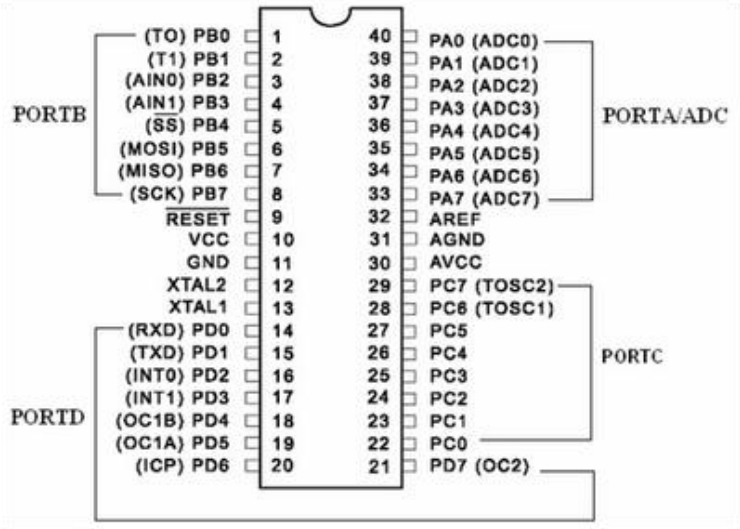
ATMega32 adalah mikrokontroler CMOS 8 *bit* daya rendah berbasis arsitektur RISC. Instruksi dikerjakan pada satu siklus *clock,* ATMega32 mempunyai *throughput* mendekati 1 MIPS per MHz, hal ini membuat ATMega32 dapat bekerja dengan kecepatan tinggi walaupun dengan penggunaan daya rendah.

Mikrokontroler ATMega32 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. Memori *Flash* sebesar *8kb* dengan kemampuan *read while write*
4. Unit Interupsi *Internal* dan *External*
5. *Port* antarmuka SPI untuk men-*download* program ke *flash*
6. EEP*ROM* sebesar 512 *byte* yang dapat diprogram saat operasi
7. Antarmuka komparator *analog*
8. *Port* USART untuk komunikasi serial.

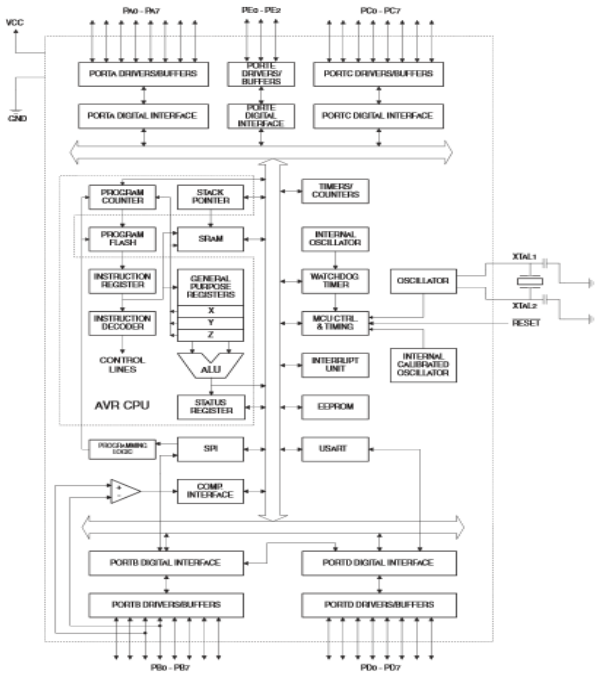
1. Konfigurasi *Pin* ATMega32

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 2.2 : Diagram Pin 3 ATMega32

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



Gambar 2.3 : Blog diagram ATMega32

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

Berikut ini adalah tabel penjelasan mengenai pin yang terdapat pada mikrokontroler ATMega32:

Tabel 2.1 : Tabel Penjelasan pin pada mikrokontroler ATMega32

Vcc	Tegangan suplai (*5 volt*)
GND	*Ground*
RESET	*Input reset level* rendah, pada *pin* ini selama lebih dari panjang pulsa *minimum* akan menghasilkan *reset* walaupun *clock* sedang berjalan. RST pada *pin* 9 merupakan *reset* dari AVR. Jika pada *pin* ini diberi masukan *low* 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 *low pass filter*
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

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

Berikut ini adalah penjelasan dari *pin* mikrokontroler ATMega32 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 2.2

Tabel 2.2 : Tabel Penjelasan pin pada port A

Pin	Keterangan
PA.7	ADC7 (ADC *Input Channel* 7)
PA.6	ADC6 (ADC *Input Channel* 6)

PA.5	ADC7 (ADC *Input Channel* 5)
PA.5	ADC4 (ADC *Input Channel* 4)
PA.3	ADC3 (ADC *Input Channel* 3)
PA.2	ADC2 (ADC *Input Channel* 2)
PA.1	ADC1 (ADC *Input Channel* 1)
PA.0	ADC0 (ADC *Input Channel* 0)

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

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 2.6.

Tabel 2.3 : Tabel Penjelasan pin pada port B

Pin	Keterangan
PB.7	SCK (SPI *Bus Serial Clock*)
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*)

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

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 2.3.

Tabel 2.4 : Tabel Penjelasan pin pada port C

Pin	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*)

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

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 2.4. (*Sistem Mikrokontroler*, hal 5, 2007)

Tabel 2.5 : Tabel Penjelasan pin pada port D

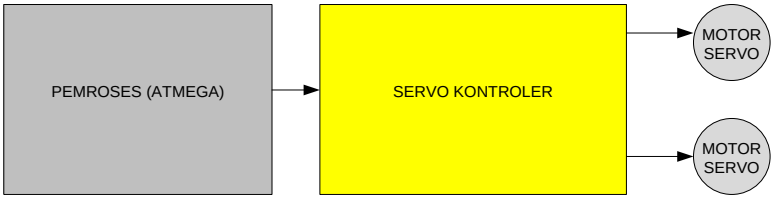
Pin	Keterangan
PD.0	RDX (*UART input line*)
PD.1	TDX (*UART output line*)
PD.2	INT0 (*external interrupt 0 input*)
PD.3	INT1 (*external interrupt 1 input*)
PD.4	OC1B (*Timer/Counter1 output compareB match output*)
PD.5	OC1A (*Timer/Counter1 output compareA match output*)
PD.6	ICP (*Timer/Counter1 input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

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

2.3 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 kontroller 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).



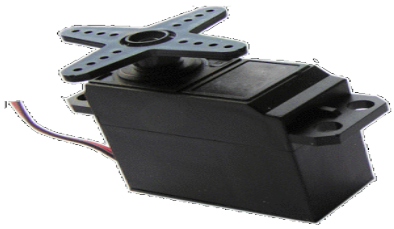
Gambar 2.4 : Struktur Diagram Servo Kontroler

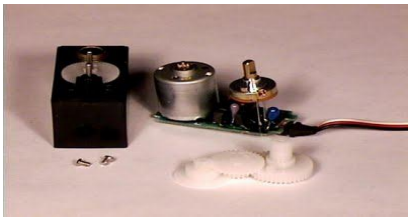
(Sumber : Milik Pribadi)

2.3.1 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 standard 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 2 Mei 2015)

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, sebernarnya 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. (Wikipediaindonesia.co.id. 2012)

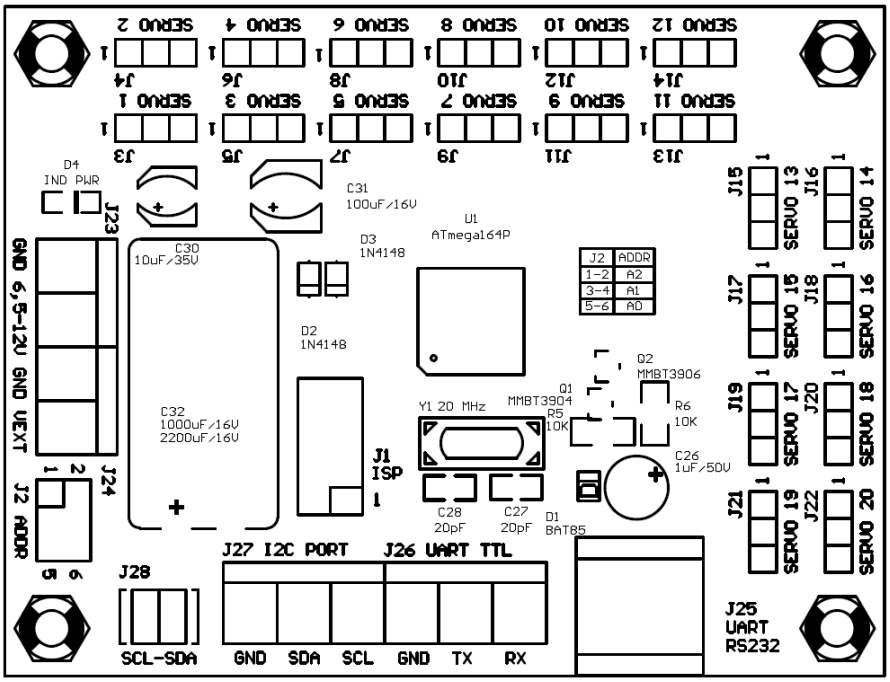
2.3.2 Modul SPC Servo Motor *Kontroller*

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 elekronik, servo kontroler ini dapat berkomikikasi melalui I2C dan Serial, servo kontroler ini dilengkapi dengan 20 chanel output untuk mengontrol servo.

Spesifikasi Modul 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 Modul SPC SERVO MOTOR CONTROLLER terpisah dengan catu daya untuk motor servo.

4. Catu daya untuk Modul 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 μs.
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, Modul SPC SERVO MOTOR CONTROLLER dapat di-*cascade* hingga 8 modul. (*Manual Book SPC Servo Motor Kontroller*. Hal 3.2011)



Gambar 2.6 : Tata Letak SPC Servo Motor Kontroller

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

2.3.3 Hitec HS-7954 SH

Hitec HS-7954 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 :
   6V dengan torsi 24 Kg
   7,4 deengan torsi 29Kg
3. Kecepatan
   6V = 0,15 sec/60°
   6.0V = 0,12 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 800u sec
   0° dengan pulsa 1500u sec
   +90° dengan pulsa 2200u sec (www.servodatabase.com, 2007)

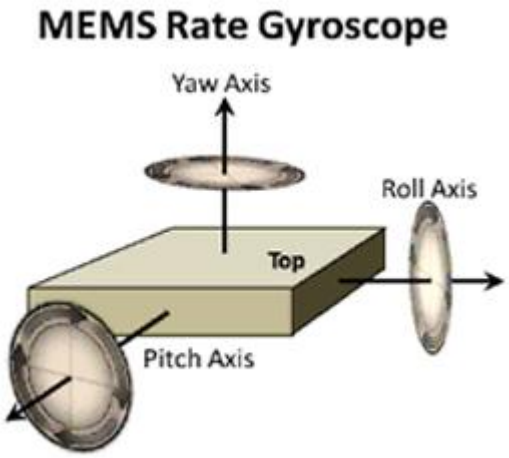
Gambar 2.7 : Bentuk Fisik Servo Hitec HS-7954 SH
(Sumber : http:// Datasheet Servo Hitec HS-7954 SH.2007)

2.4 Sensor Gyroscope dan Accelerometer

Gyroscope berfungsi untuk mengukur/menentukan orientiasi suatu benda berdasarkan pada ketetapan momentum sudut. Dari pengertian lain *gyroscope* berfungsi untuk menentukan gerakan sesuai dengan gravitasi yang dilakukan oleh pengguna. *Gyroscope* ini memiliki peranan yang sangat penting dalam hal mempertahankan keseimbangan suatu benda seperti penggunaannya pada pesawat terbang yang dapat menentukan kemiringan pada sumbu x,y, dan z.

Output yang dihasilkan oleh *gyroscope* berupa kecepatan sudut yang pada sumbu x akan menjadi *phi* (Φ), sumbu y menjadi *theta* (θ), dan sumbu z menjadi *psi* (Ψ). Sebelum digunakan biasanya *gyroscope* di kalibarasi terlebih dahulu dengan menggunakan bandul yang fungsinya untuk menentukan nilai faktor ataupun dapat juga melihat pada datasheet sensor yang digunakan.

Gyroscope adalah suatu alat berupa sensor *gyro* untuk menentukan orientasi gerak dengan bertumpu pada roda atau cakram yang berotasi dengan cepat pada sumbu.



Gambar 2.8 : Prinsip Kerja Gyroscope
(Sumber : http://id.wikipedia.org/wiki/Giroskop diakses pada tanggal 10 Mei 2015)

Gyroscope memiliki output yang peka terhadap kecepatan sudut dari arah sumbu x yang nantinya akan menjadi sudut *phi* (roll), dari sumbu y nantinya menjadi sudut *theta* (pitch), dan sumbu z nantinya menjadi sudut *psi* (yaw).

Penggunaan *gyroscope* dalam kehidupan sehari belum banyak. *Gyroscope* lazimnya digunakan pada pesawat terbang, kapal, helikopter, dll. Hal tersebut untuk mengurangi getaran yang ditimbulkan mesin agar keseimbangan tidak goyah. *Gyroscope* baru-baru ini juga diterapkan pada mainan Helikopter dengan tujuan menyeimbangkan saat diterbangkan.

Accelerometer adalah sebuah perangkat yang mengukur percepatan yang tepat . Hal ini tidak selalu sama dengan percepatan koordinat (perubahan kecepatan dari perangkat dalam ruang), tetapi agak jenis percepatan terkait dengan fenomena berat badan yang dialami oleh massa uji yang berada dalam kerangka acuan dari perangkat *accelerometer* . Untuk contoh di mana jenis percepatan berbeda, accelerometer akan mengukur nilai saat duduk di tanah, karena massa ada bobot, meskipun mereka tidak mengubah kecepatan. Namun, *accelerometer* di gravitasi jatuh bebas ke arah pusat bumi akan mengukur nilai nol karena, meskipun kecepatan meningkat, berada dalam kerangka acuan di mana ia ringan .

Dengan mengukur berat badan, *accelerometer* mengukur percepatan jatuh bebas kerangka referensi (kerangka acuan inersial) relatif terhadap dirinya

sendiri (*accelerometer*). *Accelerometer* kebanyakan tidak menampilkan nilai mereka mengukur, tetapi pasokan ke perangkat lain. *Accelerometer* nyata juga memiliki keterbatasan praktis dalam seberapa cepat mereka menanggapi perubahan dalam percepatan , dan tidak dapat merespon perubahan atas perubahan frekuensi tertentu.

Model tunggal dan multi-sumbu *accelerometer* yang tersedia untuk mendeteksi besar dan arah percepatan yang tepat (atau g-force), sebagai vektor kuantitas, dan dapat digunakan untuk orientasi akal (karena arah perubahan berat badan), percepatan koordinat (asalkan menghasilkan g-force atau perubahan g-force), getaran, guncangan , dan jatuh (kasus di mana perubahan percepatan yang tepat, karena cenderung menuju nol). Micromachined *accelerometer* semakin hadir di perangkat elektronik portabel dan video pengendali permainan, untuk mendeteksi posisi perangkat atau memberikan masukan permainan.

Pasangan *accelerometer* diperpanjang atas wilayah ruang dapat digunakan untuk mendeteksi perbedaan (gradien) dalam percepatan yang tepat dari frame referensi yang terkait dengan poin. Alat ini disebut gradiometers gravitasi , karena mereka mengukur gradien di medan gravitasi. Pasangan seperti *accelerometer* dalam teori juga dapat mendeteksi gelombang gravitasi.

Prinsip *accelerometer* adalah *accelerometer* mengukur percepatan yang tepat , yang merupakan percepatan itu pengalaman relatif terhadap terjun bebas dan percepatan yang dirasakan oleh orang-orang dan benda-benda. Dengan kata lain, pada setiap titik dalam ruang-waktu dengan prinsip kesetaraan menjamin keberadaan lokal kerangka inersia , dan *accelerometer* mengukur percepatan relatif terhadap frame yang. Percepatan tersebut populer diukur dalam hal g-force.

Sebuah *accelerometer* pada saat istirahat relatif terhadap permukaan bumi akan menunjukkan sekitar 1 atas g, karena setiap titik di permukaan bumi adalah percepatan ke atas relatif terhadap kerangka inersia lokal (bingkai benda jatuh bebas di dekat permukaan). Untuk mendapatkan percepatan gerak sehubungan dengan Bumi, "gravitasi offset" ini harus dikurangkan dan koreksi untuk efek yang disebabkan oleh rotasi bumi ke frame inersia.

Alasan munculnya offset gravitasi adalah prinsip kesetaraan Einstein , yang menyatakan bahwa efek gravitasi pada objek yang bisa dibedakan dari percepatan. Ketika dipertahankan tetap dalam medan gravitasi, misalnya, menerapkan kekuatan darat reaksi atau dorong ke atas setara, kerangka acuan untuk *accelerometer* (casing sendiri) mempercepat atas sehubungan dengan kerangka acuan jatuh bebas. Dampak dari percepatan ini yang bisa dibedakan dari percepatan lain dialami oleh instrumen, sehingga *accelerometer* tidak dapat mendeteksi perbedaan antara duduk di sebuah roket di landasan peluncuran, dan berada di roket yang sama di dalam ruang sementara itu menggunakan mesin untuk mempercepat pada 1 g. Untuk alasan serupa, *accelerometer* akan membaca nol dalam setiap jenis jatuh bebas . Hal ini termasuk penggunaan dalam pesawat ruang angkasa meluncur di luar angkasa jauh dari massa, sebuah pesawat ruang angkasa yang mengorbit bumi, pesawat dalam busur parabola "nol-g", atau jatuh bebas-dalam kekosongan. Contoh lain adalah bebas jatuh di ketinggian yang cukup tinggi bahwa efek atmosfer dapat diabaikan.

Namun ini tidak termasuk penurunan (non-bebas) di mana hambatan udara menghasilkan tarik kekuatan yang mengurangi percepatan, sampai konstanta kecepatan terminal tercapai. Pada kecepatan terminal *accelerometer* akan menunjukkan 1 ke atas g percepatan. Percepatan diukur dalam SI satuan meter per detik per detik (m / s 2), dalam cgs satuan gal (Gal), atau yang populer dalam hal g-force (g).

Untuk tujuan praktis untuk menemukan percepatan benda sehubungan dengan bumi, seperti untuk digunakan dalam sistem navigasi inersia , gravitasi pengetahuan lokal diperlukan. Hal ini dapat diperoleh dengan mengkalibrasi perangkat beristirahat, atau dari model yang dikenal gravitasi pada posisi saat ini perkiraan. (id.wikipedia.org, 2014)

2.4.1 Modul MPU-6050 6-Axis Gyroscope+Accelerometer Module

Sensor ini adalah modul sensor gerak terpadu yang merupakan *gyroscope* 3 sumbu sekaligus pengukur percepatan (*accelerometer*) 3 sumbu / *3-axis MEMS gyroscope + 3-axis MEMS accelerometer*. Sensor ini sangat akurat, dengan ADC

(*analog-to-digital converter*) internal beresolusi 16-bit.Modul ini menggunakan IC InvenSense MPU-6050 yang merupakan komponen elektronika pertama dan satu-satunya di dunia yang memadukan fungsi *gyroscope* dan *accelerometer* dalam satu sirkit terpadu (IC / *integrated circuit*).

Gambar 2.9 Gambar Fisik Modul MPU-6050 6-Axis *Gyroscope+Accelerometer* (Sumber http://www.vcc2gnd.com/2014/02/mpu-6050-6-axis-gyroscopeaccelerometer_73.html)

MPU-6050 menerapkan teknologi MotionFusion™ dan *run-time calibration firmware* yang menjamin kinerja optimal bagi pengguna. Dengan adanya Digital Motion Processor™ modul ini dapat diintegrasikan dengan magnetometer atau sensor lainnya lewat antarmuka I2C untuk memproses algoritma gerakan yang kompleks secara internal tanpa membebani kerja mikroprosesor / mikrokontroler utama.

Gambar 2.10 : Diagram Modul MPU 6050 *Gyroscope* dan *Accelerometer* (Sumber http://www.vcc2gnd.com/2014/02/mpu-6050-6-axis-gyroscopeaccelerometer_73.html)

Produsen IC ini mendukung pengembangan perangkat keras dan perangkat lunak untuk menggunakan chip ini, antara lain dengan merilis API (*Application Programming Interface*) yang dapat diakses dari situs web resmi Invensense. Berikut ini adalah fitur-fitur dan spesifikasi Modul MPU 6050 *Gyroscope* dan *Accelerometer*.

1. Catu daya IC dari 2,375 V ~ 3,46 V namun modul ini sudah dilengkapi dengan LDO / *low drop-out voltage regulator* (anda dapat menyambungkan pin Vcc dari modul ini dengan pin 5V pada sistem minimum anda)
2. Antarmuka kendali dan pengumpulan data lewat protokol I^2C berkecepatan tinggi (*Fast Mode*, 400 kHz), pada modul ini sudah dipasangkan *pull-up resistor* 2K2 sehingga Anda bisa menyambungkan pin SDA dan SCL dari modul ini dengan mikrokontroler / Arduino Board tanpa resistor eksternal tambahan
3. Pilihan rentang skala *gyroscope*: 250° (sensitivitas 13,1), 500° (65,6), 1000° (32,8), 2000° (16,4) per detik; sensitivitas dalam satuan LSB/°/detik
4. Pilihan rentang skala *accelerometer*: ±2g (sensitivitas 16384), ±4g (8192), ±8g (4096), ±16g (2048); sensitivitas dalam LSB/g
5. Data keluaran MotionFusion sebanyak 6 atau 9 sumbu dalam format matriks rotasi, quaternion, sudut Euler, atau data mentah (*raw data format*).
6. Memori penampung data (*buffer memory*) sebesar 1 Kb, FIFO (*First-In-First-Out*)
7. Dengan digabungkannya *accelerometer* dan *gyroscope* dalam satu sirkut terpadu menyebabkan pendeteksian gerakan menjadi lebih akurat (*reduced settling effects and sensor drift*) karena faktor kesalahan penyesuaian persilangan sumbu antara *accelerometer* dan *gyroscope* dapat dihilangkan
8. DMP™ Engine mengambil alih komputasi rumit dari prosesor utama sehingga sistem tidak terbebani kalkulasi yang kompleks (*red*: sebelum adanya IC ini, perancang rangkaian elektronika biasanya menggunakan chip PLD eksternal untuk mengerjakan komputasi semacam ini karena

perhitungan matematika dalam kalkulasi gerak sangatlah kompleks dan terlalu membebani kerja mikrokontroler yang biasanya bertenaga terbatas).

9. Tersedia platform pengembangan perangkat lunak MotionApps™ untuk sistem operasi Android, Linux, dan Windows
10. Algoritma untuk menghitung bias dan kalibrasi kompas sudah terpasang dan siap digunakan, tidak perlu intervensi dari pemakai
11. Interupsi yang dapat diprogram untuk mendeteksi pengenal gestur (*gesture recognition)*, pergeseran (*panning*), *zooming, scrolling,* dan *shake detection*
12. Konsumsi arus *gyroscope* hanya sebesar 3,6 mA, *gyroscope* + *accelerometer* hanya 3,8 mA (tenaga penuh, 1 kHz *sample rate*).
13. Moda siaga hemat daya hanya mengkonsumksi arus sebesar 5µA
14. Dapat menoleransi guncangan hingga 10000g
15. Modul dengan PCB berkualitas dengan *gold immersion welding* untuk menjamin kualitas
16. Akses sangat mudah menggunakan pin standar dengan *pitch* 0,1" / 2,54 mm. (www.vcc2gnd.com, 2014)

2.5 Definisi Bahasa Basic

BASIC, adalah singkatan dari *Beginners' All-purpose Symbolic Instruction Code* adalah sebuah kelompok bahasa pemrograman tingkat tinggi. Secara harfiah, BASIC memiliki arti "kode instruksi simbolis semua tujuan yang dapat digunakan oleh para pemula". Memang, istilah "Bahasa BASIC" di sini juga bisa diartikan menjadi bahasa untuk pemula, atau dengan kata lain, disebut sebagai bahasa dasar, tapi hal tersebut dirasa kurang tepat, mengingat BASIC dapat juga digunakan oleh para pemrogram ahli.

BASIC pertama kali dikembangkan pada tahun 1963 oleh John George Kemeny dan Thomas Eugene Kurtz yang berasal dari Dartmouth College, untuk mengizinkan akses terhadap komputer bagi para mahasiswa jurusan selain jurusan ilmu eksakta. Pada waktu itu, hampir semua komputer membutuhkan perangkat lunak, dan waktu itu belum ada perangkat lunak yang dijual secara bebas,

sehingga hanya orang-orang tertentulah yang dapat menggunakan komputer, yakni para matematikawan dan ilmuwan, karena mereka dapat membangun perangkat lunak sendiri. Bahasa BASIC, setelah diciptakan menjadi menjamur dan banyak dimodifikasi. Bahasa BASIC menjadi bahasa yang paling populer digunakan pada komputer mikro pada akhir tahun 1970-an dan komputer rumahan pada tahun 1980-an. Dan hingga saat ini, menjadi bahasa yang dialeknya beberapa kali berevolusi. (id.wikipedia.org, 2014)

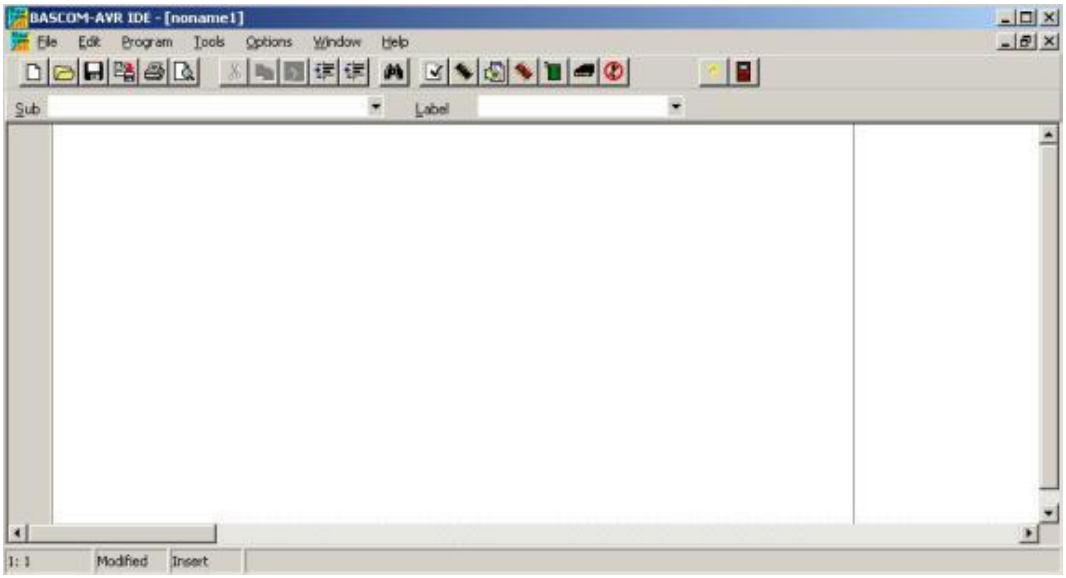
2.6 Definisi Bascom-AVR

BASCOM-AVR adalah program basic compiler berbasis windows untuk mikrokontroler keluarga AVR merupakan pemrograman dengan bahasa tingkat tinggi ” BASIC ” yang dikembangkan dan dikeluarkan oleh MCS elektronika sehingga dapat dengan mudah dimengerti atau diterjemahkan.

Dalam program BASCOM-AVR terdapat beberapa kemudahan, untuk membuat program software ATMEGA 32, seperti program simulasi yang sangat berguna untuk melihat, simulasi hasil program yang telah kita buat, sebelum program tersebut kita download ke IC atau ke mikrokontroler.

Gambar 2.11: Logo Bascom-AVR
(Sumber http://www.robotics-university.com/2013/04/cara-menggunakan-bascom-avr.html)

Ketika program BASCOM-AVR dijalankan dengan mengklik icon BASCOM-AVR, maka jendela berikut akan tampil seperti pada gambar 2.12.



Gambar 2.12 Tampilan Jendela Program BASCOM-AVR
(Sumber http://www.robotics-university.com/2013/04/cara-menggunakan-bascom-avr.html)

BASCOM-AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan pergerakan LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika kita membuat aplikasi yang berhubungan dengan LCD.

2.6.1 Kontruksi bahasa BASIC pada BASCOM-AVR

Setiap bahasa pemprograman mempunyai standar penulisan program. Konstruksi dari program bahasa BASIC harus mengikuti aturan sebagai berikut:

```
$regfile = “header”
’inisialisasi
’deklarasi variabel
’deklarasi konstanta
Do
’pernyataan-pernyataan
Loop
end
```

2.6.2 Pengarah preprosesor

$regfile = “m16def.dat” merupakan pengarah pengarah preprosesor bahasa BASIC yang memerintahkan untuk meyisipkan file lain, dalam hal ini adalah *file* m16def.dat yang berisi deklarasi register dari mikrokonroller ATmega 16, pengarah preprosesor lainnya yang sering digunakan ialah sebagai berikut:

```
$crystal = 12000000  ‘menggunakan crystal clock 12 MHz
$baud = 9600         ‘komunikasi serial dengan baudrate 9600
$eeprom              ‘menggunakan fasilitas eeprom
```

2.6.3 Tipe Data

Tipe data merupakan bagian program yang paling penting karena sangat berpengaruh pada program. Pemilihan tipe data yang tepat maka operasi data menjadi lebih efisien dan efektif.

Tabel 2.6 : Tipe Data pada BASCOM AVR

TIPE DATA	UKURAN (BYTE)	RENTANG
Bit	1/8	0 atau 1
Byte	1	0 sampai 255
Integer	2	-32768 sampai +32767
Word	2	0 sampai 65535
Long	4	-2147483648 sampai +2147483647
Single	4	
String	Maksimum 254 byte	

(Sumber : Byron (1991). *Pemrograman dengan Basic*.Penerbit Erlangga:Jakarta)

2.6.4 Konstanta dan Variabel

Konstanta merupakan suatu nilai dengan tipe data tertentu yang tidak dapat diubah-ubah selama proses program berlangsung. Konstanta harus didefinisikan terlebih dahulu diawal program. Contoh : *Kp = 35, Ki=15, Kd=40.*

Variabel adalah suatu pengenal (*identifier)* yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program yang dapat diubah-ubah sesuai dengan kebutuhan. Nama dari variable terserah sesuai dengan yang diinginkan namun hal yang terpenting adalah setiap variabel diharuskan :

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf, max 32 karakter.
2. Tidak boleh mengandung spasi atau symbol-simbol khusus seperti : *$, ?, %, #, !, &, *, (,),* -, +, = dan lain sebagainya kecuali *underscore*.
3. Deklarasi

Deklarasi sangat diperlukan bila akan menggunakan pengenal (identifier) dalam suatu program.

2.6.5 Deklarasi Variabel, Konstanta, Fungsi dan Buatan

Bentuk umum pendeklarasian suatu variable adalah Dim *nama_variabel* AS *tipe_data.* Contoh : *Dim x* As *Integer* '*deklarasi x bertipe integer*. Sedangkan dalam konstanta di deklarasikan langsung. Contohnya : *S = "Hello world" 'Assign string.*

Deklarasi fungsi merupakan bagian yang terpisah dari program dan dapat dipanggil di manapun di dalam program. Fungsi dalam Bahasa Basic ada yang sudah disediakan sebagai fungsi pustaka seperti print, input data dan untuk menggunakannya tidak perlu dideklarasikan.

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah Sub *Test* (byval *variabel* As *type*). Contohnya : Sub *Pwm(*byval *Kiri* As *Integer* , Byval *Kanan As Integer)*

2.6.6 Operator

Operator digunakan untuk melakukan operasi terhadap bilangan. Pada Bascom operator dibedakan menjadi operator aritmetik, operator relasional, dan operator logika. Operator aritmatik adalah operator yang digunakan dalam kalkulasi, yaitu + (penjumlahan), - (pengurangan), * (perkalian, / (pembagian), \ (pembagian

integer), MOD (modulo = sisa dari pembagian). Operator relasional digunakan untuk membandingkan dua nilai, yang memberikan hasil benar (1) atau salah (0) dan dapat digunakan untuk membuat keputusan.

Tabel 2.7 : Operator-Operator Relasi Pada BASCOM AVR

Operator	**Relasi**	**Pernyataan**
=	Sama dengan	X = Y
< >	Tidak sama dengan	X < > Y
<	Lebih kecil	X < Y
>	Lebih besar	X > Y
<=	Lebih kecil atau sama dengan	X <= Y
>=	Lebih besar atau sama dengan	X >= Y

(Sumber : Byron (1991). *Pemrograman dengan Basic*.Penerbit Erlangga:Jakarta)

Operator logika digunakan untuk menguji suatu pola bit tertentu, manipulasi bit atau operator Boolean. Misal operator AND dapat digunakan untuk mengabaikan semua bit dalam suatu byte kecuali satu bit untuk memantau status bit tersebut.

Tabel 2.8 : Operator-Operator Logika Pada Bascom AVR

Operator	**Makna**
NOT	Komplemen/Inverter
AND	Konjungsi (dan)
OR	Disjungsi (atau)
EXOR	Exclusive OR

(Sumber : Byron (1991). *Pemrograman dengan Basic*.Penerbit Erlangga:Jakarta)

2.6.7 Pernyataan Kondisional

Pernyataan ini digunakan untuk melakukan pengambilan keputusan terhadap dua buah bahkan lebih kemungkinan untuk melakukan suatu blok

pernyataan atau tidak. Konstruksi penulisan pernyatan IF-THEN-ELSE-END IF pada bahasa BASIC ialah sebagai berikut:

IF *pernyataan kondisi 1* THEN
‘blok pernyataan 1 yang dikerjakan bila kondisi 1 terpenuhi
IF *pernyataan kondisi 2* THEN
‘blok pernyataan 2 yang dikerjakan bila kondisi 2 terpenuhi
IF *pernyataan kondisi 3* THEN
‘blok pernyataan 3 yang dikerjakan bila kondisi 3 terpenuhi

Setiap penggunaan pernyataan IF-THEN harus diakhiri dengan perintah END IF sebagai akhir dari pernyatan kondisional. (www.robotics-university.com, 2013)

2.7 Pengontrol PID

Sistem Kontrol PID *(Proportional–Integral–Derivative controller*) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tesebut (Feed back).

Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan

2.7.1 Kontrol Proporsional

Kontrol P jika G(s) = kp, dengan k adalah konstanta. Jika u = G(s) • e maka u = Kp • e dengan Kp adalah Konstanta Proporsional. Kp berlaku sebagai Gain (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang

sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya rise time dan settling time.

2.7.2 Kontrol Integratif

Jika G(s) adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [integrale(t)dT]Ki$ dengan Ki adalah konstanta Integral, dan dari persamaan diatas, G(s) dapat dinyatakan sebagai $u = Kd.[deltae / deltat]$ Jika e(T) mendekati konstan (bukan nol) maka u(t) akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika e(T) mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan Ki yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan Ki yang sangat tinggi justru dapat menyebabkan output berosilasi karena menambah orde system

2.7.3 Kontrol Derivatif

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s.Kd$ Dari persamaan di atas, nampak bahwa sifat dari kontrol D ini dalam konteks "kecepatan" atau rate dari error. Dengan sifat ini ia dapat digunakan untuk memperbaiki respon transien dengan memprediksi error yang akan terjadi. Kontrol Derivative hanya berubah saat ada perubahan error sehingga saat error statis kontrol ini tidak akan bereaksi, hal ini pula yang menyebabkan kontroler Derivative tidak dapat dipakai sendiri

Untuk mendapatkan aksi kontrol yang baik diperlukan langkah coba-coba dengan kombinasi antara P, I dan D sampai ditemukan nilai Kp, Ki dan Kd seperti yang diiginkan.

1. Memahami cara kerja system.
2. Mencari model sistem dinamik dalam persamaan differensial.
3. Mendapatkan fungsi alih sistem dengan Transformasi Laplace.
4. Memberikan aksi pengontrolan dengan menentukan konstanta Kp, Ki dan Kd.

5. Menggabungkan fungsi alih yang sudah didapatkan dengan jenis aksi pengontrolan.
6. Menguji sistem dengan sinyal masukan fungsi langkah, fungsi undak dan impuls ke dalam fungsi alih yang baru.
7. Melakukan Transformasi Laplace balik untuk mendapatkan fungsi dalam kawasan waktu.
8. Menggambar tanggapan sistem dalam kawasan waktu. (putraekapermana.wordpress.com, 2013)

2.8 Penalaran (*Tuning*)

Penalaran sebuah pengontrol untuk suatu aplikasi tertentu meliputi langkah-langkah pemilihan mode kontrol yang akan digunakan serta pengaturan mode kontrol. Ini berarti penentuan apakah kontrol proposional, kontrol proposional plus derivatif, proposional plus integral, ataupun proposional plus integral plus derivatif yang akan digunakan, dan pemilihan nilai *Kp, Kd* dan *Ki* yang bersesuaian. Langkah ini akan menentukan bagaimana sistem bereaksi terhadap sebuah gangguna atau perubahan nilai pengaturan, seberapa cepat sistem menanggapi perubahan yang terjadi, berapa lama waktu yang dibutuhkan untuk mencapai keadaan tunak setelah terjadi gangguan atau perubahan nilai pengaturan, serta apakah terdapat *error* keadaan tunak atau tidak.

Terdapat beberapa jenis-jenis respons yang dapat muncul untuk berbagai mode kontrol yang berbeda ketika diberikan suatu masukan sinyal tangga (sinyal *step*) dimana terjadi perubahan tiba-tiba menjadi niali pengaturan konstan lainnya, atau mungkin pula gangguan konstans yang tiba-tiba. Kontrol proposional memberi respons yang cepat dengan osilasi yang menghilang dengan cukup cepat tetapi meninggalkan *error* keadaan tunak. Kontrol proposional plus integral tidak memiliki *error* keadaan tunak tetapi memperlihatkan adanya gejala *osilasi* yang lebih panjang sebelum mencapai keadaan tunak. Kontrol proposional plus derivatif juga tidak memperlihatkan adanya *error* keadaan tunak, karena adanya elemen integral, serta memiliki *osilasi* yang lebih pendek dibandingkan dengan

kontrol proposional plus integral. Dilibatkannya kontrol derivatif akan mereduksi osilasi yang terjadi.

Istilah penalaran (*tuning*) dugunakan untuk menggambarkan metode-metode yang digunakan untuk memilih pengaturan pengontrol terbaik untuk mendapatkan bentuk unjuk kerja tertentu, misalnya suatu komponen dimana sebuah sinyal *error* akan melibatkan variable yagn dikontrol berosilasi di sekitar nilai diinginkan dengan osilasi yang meluruh secara cepat sedemikian rupa sehingga setiap amplitudo yang berturutan bernilai seperempat dari amplitudo sebelumnya. Dua metode yang banyak digunakan adalah metode reaksi proses dan metode *ultimate cycle* yang ditemukan oleh *Ziegler* dan *Nichols.*

Pada laporan ini penulis menentukan batasan PID yaitu dari 800 – 2200, sedangkan batasan sensornya yaitu dari (-7) – (+7). Batasan nilai Kp, Kd dan Ki adalah 255. Kemudian dengan metode penalaran *ultimate cycle* menetukan nilai *Kp* terlebih dahulu. Dengan metode ini, aksi integral dan derivatif pertama-tama direduksi menjadi nilai efektif terkecilnya. Konstanta propsional *Kp* selanjutnya disetel rendah dan secara bertahap ditingkatkan hingga osilasi pada variabel yang dikontrol mulai terjadi. (*Sistem Instrumentasi dan Sistem Kontrol,* hal 116, 2006)