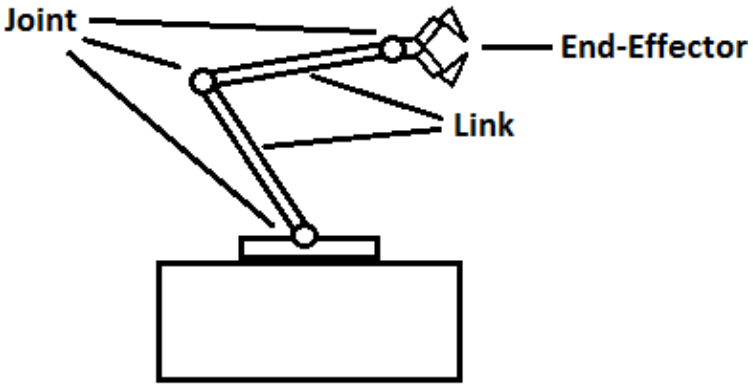
BAB II
TEORI PENUNJANG

2.1 Komponen Struktur Lengan

Komponen utama lengan secara keseluruhan terbagi menjadi tiga bagian, yaitu *manipulator*, sistem mekanik, dan sistem kontrol.

2.1.1 Manipulator

Manipulator adalah bagian mekanik yang berfungsi untuk memindahkan, mengangkat, dan memanipulasi benda bekerja.



Gambar 2.1. Diagram struktur Lengan

Fungsi alat tergantung dari jenis *end effector* yang dipasang pada *manipulator*. *End effector* bisa berupa perangkat pengelasan, pengecatan, peralatan mesin ataupun *gripper* yang memiliki rahang yang dapat bergerak membuka dan menutup.

2.1.2 Desain

Desain sistem mekanik menggambarkan bagaimana bentuk dari tangan dan jenis komponen yang harus digunakan.

a) Desain Mekanik

Desain mekanik merupakan asas yang menentukan keterampilan sebuah lengan tersebut. Terdapat 3 aspek dasar yang harus dipertimbangkan dalam mendesain sebuah lengan mekanik, yaitu: [3]

- Jumlah batang (*link*)
- Jumlah sambungan (*joint*)
- Ukuran dan pergerakan tiap batang (derajat kebebasan/DOF)

b) Klasifikasi Gerak

Menurut arah pergerakannya, gerak dapat dibedakan menjadi dua jenis gerak, yaitu: [3]

- Gerak Translasi atau *Linear*
- Gerak Rotasi

c) Transmisi Gerak

Secara umum ada dua tipe transmisi gerak, yaitu transmisi gerak rotasi ke rotasi dan transmisi gerak rotasi ke linear. [3]

- Transmisi gerak rotasi ke rotasi terkait pada aspek penting yaitu roda gigi ideal yang berarti roda gigi benar-benar bulatan penuh, berputar pada titik tengah, dan juga tanpa inersia (*inertialess*). Permukaan antar roda gigi tidak terjadi gaya gesek sehingga menimbulkan gerigi-gerigi.
- Transmisi gerak rotasi ke linear ada beberapa komponen yang dapat menghasilkan konversi gerak rotasi ke linear, salah satunya adalah mur dan baut (*screw*), dan *rack pinion*.

d) Sistem Penggerak

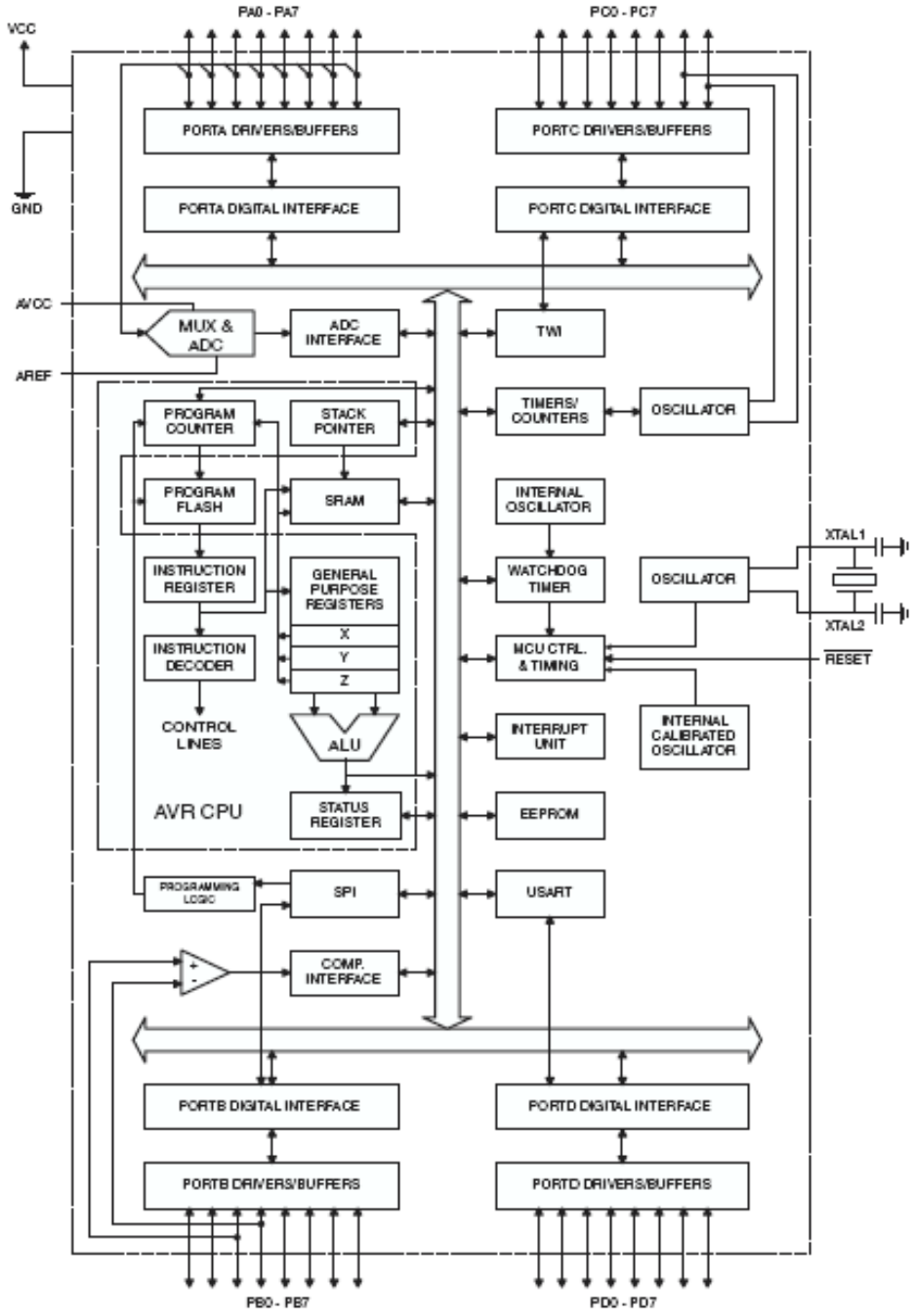
Actuator atau penggerak adalah komponen penggerak suatu sistem yang diklasifikasikan menjadi beberapa tipe sesuai dengan prinsip penghasil geraknya, yaitu penggerak elektrik (motor DC, motor servo, motor stepper, motor AC, dan lain-lain), penggerak pneumatic (berbasis kompresi gas: udara, nitrogen, dan lain-lain), dan penggerak hidrolik (berbasis kompresi benda cair: minyak pelumas, dan lain-lain). Terdapat beberapa kriteria yang perlu diperhatikan dalam pemilihan jenis *actuator*, antara lain: torsi, arus atau daya (*power supply*), ukuran, dan kecepatan respon *actuator*. [3]

2.2 Mikrokontroler ATMega16

Mikrokontroler merupakan keseluruhan sistem komputer yang dikemas menjadi sebuah *chip* dengan terdapat mikroprosesor, I/O *(Input atau Output)*, memori bahkan ADC *(Analog To Digital Converter)*, berbeda dengan

mikroprosesor yang berfungsi sebagai proses data. Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit dan sebagian besar instruksi dieksekusi dalam 1 siklus *clock* atau dikenal dengan teknologi RISC (*Reduced Instruction Set Computing*). Secara umum, AVR (*Alf and Vegard's RISC processor*) dapat dikelompokan ke dalam 4 kelas, yaitu keluarga AT90Sxx, keluarga ATMega dan AT86RFxx. Pada dasarnya yang membedakan masing-masing adalah kapasitas memori, *peripheral* dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan semuanya bisa dikatakan hampir sama.

ATmega16 adalah mikrokontroler AVR dengan fitur lengkap dan jumlah *port* I/O yang mencukupi. Mikrokontroler ini memiliki 40 kaki (*dual-inline Package*) tetap digunakan untuk diaplikasikan pada sistem yang membutuhkan jumlah *port* I/O yang banyak.

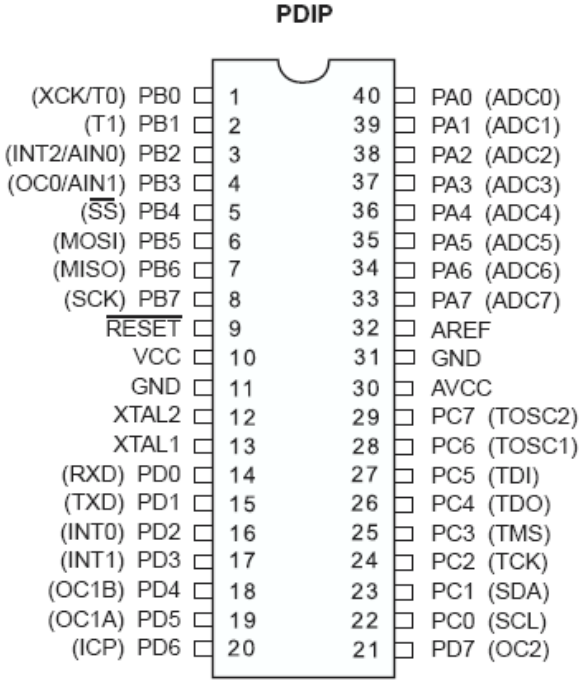


Gambar 2.2. Blok Diagram ATmega16 [1]

Pada gambar blok diagram mikrokontroler ATmega16 di atas terdapat fitur – fitur sebagai berikut :

- Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi dengan daya rendah.

- Arsitektur RISC dengan throughput mencapai 16 MIPS pada frekuensi 16MHz.
- Memiliki kapasitas *flash* memori 16 Kbyte, EEPROM 512 byte dan SRAM 1 Kbyte.
- Saluran I/O sebanyak 32 buah yaitu *PortA*, *PortB*, *PortC* dan *PortD*.
- CPU yang terdiri atas 32 buah register.
- Unit interupsi internal dan eksternal.
- Port USART untuk komunikasi serial.



Gambar 2.3. Konfigurasi Pin ATmega16 r [1]

Keterangan konfigurasi Pin ATMega16 adalah sebagai berikut : [1]

a) **Vcc :** *Power Supply*
b) **GND :** *Ground*
c) **PORTA (PA0 - PA7) :** *portA* digunakan sebagai sebuah *port* I/O bidireksional 8-bit *Output buffer PortA* dapat memberi arus 20 mA dan dapat menyalakan display LED secara langsung. *Data Direction Register portA* (DDRA) harus diatur terlebih dahulu sebelum *portA* digunakan. Bit-bit DDRA diberikan 0 jika akan memfungsikan *portA* sebagai *input*, atau

memberikan 1 jika sebagai *output*. Selain itu, kedelapan pin *portA* juga digunakan untuk masukan sinyal analog bagi *Analog/Digital converter*.

d) **PORTB (PB0 – PB7) :** *portB* digunakan sebagai sebuah *port* I/O bidireksional 8-bit *Output buffer PortB* dapat memberi arus 20 mA dan dapat menyalakan display LED secara langsung. *Data Direction Register portB* (DDRB) harus diatur terlebih dahulu sebelum *portB* digunakan. Bit-bit DDRB diberikan 0 jika akan memfungsikan *portB* sebagai *input*, atau memberikan 1 jika sebagai *output*. Khusus PB6 dapat digunakan sebagai masukan Kristal (*inverting oscillator amplifier*) sedangkan khusus PB7 dapat digunakan *output* Kristal (*output inverting oscillator amplifier*).

e) **PORTC (PC0 – PC5) :** *port* I/O (PC5 – PC0) digunakan sebagai sebuah *port* I/O bidireksional *Output buffer PortC* dapat memberi arus 20 mA dan dapat menyalakan display LED secara langsung. *Data Direction Register portC* (DDRC) harus diatur terlebih dahulu sebelum *portC* digunakan. Bit-bit DDRC diberikan 0 jika akan memfungsikan *portC* sebagai *input*, atau memberikan 1 jika sebagai *output*. Selain itu, dua pin *portC* (PC6 dan PC7) juga memiliki fungsi alternatif sebagai *oscillator* untuk *timer/counter 2*.

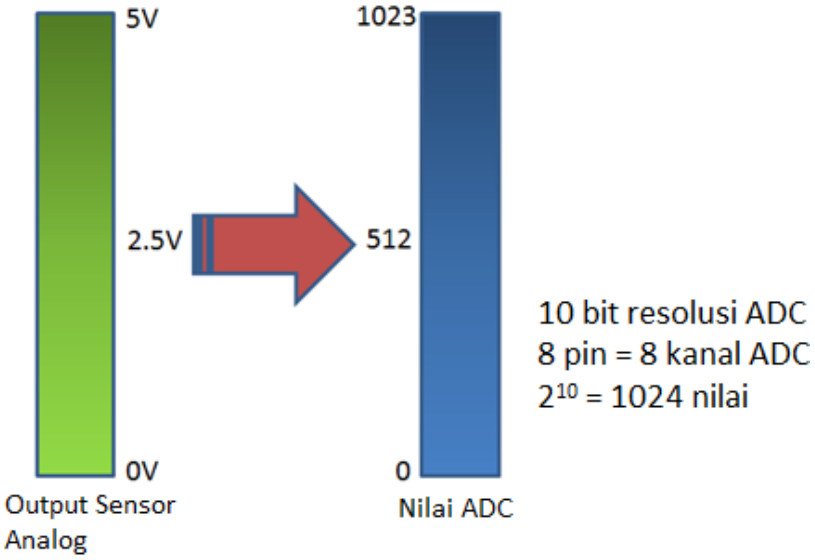
f) **PORTD (PD0 - PD7) :** *port* I/O 8-bit *Output buffer PortD* dapat memberi arus 20 mA dan dapat menyalakan display LED secara langsung. *Data Direction Register portD* (DDRD) harus diatur terlebih dahulu sebelum *portD* digunakan. Bit-bit DDRD diberikan 0 jika akan memfungsikan *portD* sebagai *input*, atau memberikan 1 jika sebagai *output*. Selain itu PD0 dan PD1 juga memiliki fungsi sebagai pin komunikasi serial antara mikrokontroler dengan perangkat lain, PD0 sebagai penerima data dan PD1 sebagai pengirim data.

Pada mikrokontroler ATmega16 ini terdapat beberapa fungsi yang digunakan untuk perancangan, yaitu : pin-pin ADC, *portC*, *portD* dan *portB*.

2.2.1 ADC Mikrokontroler

Pin ADC digunakan sebagai pembaca data *input* dari sensor yang kemudian akan diolah data tersebut menjadi nilai ADC. Langkah langkah yang harus dipahami dalam menggunakan ADC ini adalah: [2]

1. Memahami fitur spesifikasi yang tersedia pada mikrokontroler ATmega16 ini pada *datasheet.* Berikut ini adalah fitur spesifikasi ADC pada mikrokontroler ATmega16: [1]
   - Lebar resolusi ADC 10-bit
   - *Integral Non-Linearity* ADC sebesar 0.5 LSB
   - Akurasi absolut pada LSB ±2 LSB
   - Waktu konversi sebesar 13 – 260 µs
   - Memiliki 7 *Input* Kanal yang berbeda
   - Memiliki 2 *Input* Kanal dengan pilihan tambahan 10x dan 200x
   - Rentang tegangan *Input* ADC antara 0 - Vcc (tegangan *input*)
   - Pemilihan tegangan referensi ADC 2.56V
   - *Free Running or Single Conversion Mode*
   - ADC mulai mengkonversi otomatis setelah mendapat pemicu tegangan
2. Penjelasan tentang Resolusi ADC pada mikrokontroler ATmega16 terdapat pada gambar di bawah ini :



Gambar 2.4. Resolusi ADC mikrokontroler [2]

Sepuluh bit resolusi yaitu nilai ADC memiliki rentang nilai 2^{10} = 1024, artinya nilai ADC akan memiliki nilai antara 0 – 1023. Jika nilai analog yang menjadi *input* adalah 0 – 5 volt, maka konversi analog ke digital pada ADC akan memiliki nilai yang linier sebagai berikut: [2]

- 0 volt = 0 nilai ADC

- 2.5 volt = 511 nilai ADC
- 5 volt = 1023 nilai ADC

3. Pengaturan ADC *prescaler* yaitu untuk menentukan frekuensi yang digunakan ADC pada saat melakukan konversi. Nilai frekuensi yang digunakan ADC tergantung pada frekuensi dari XTAL yang digunakan pada mikrokontroler. Nilai *prescaler* ADC yaitu 2, 4, 8, 16, 32, 64 dan 128. Umumnya ADC bekerja pada rentang frekuensi antara 50KHz sampai 200KHz. Untuk mendapatkan rentang frekuensi tersebut, maka frekuensi XTAL mikrokontroler dibagi oleh *prescaler* maka akan menghasilkan frekuensi kerja ADC. Seperti contoh mikrokontroler yang memiliki frekuensi XTAL 11 MHz, kemudian dipilih nilai *prescaler* 64 maka frekuensi ADC didapat:

   $\frac{11\ MHz}{64} = 171.875\ KHz$

   Semakin rendah nilai frekuensi ADC yang didapat, maka akan semakin tinggi tingkat akurasi nilai hasil konversi yang dihasilkan. [2]

2.2.2 Port I/O

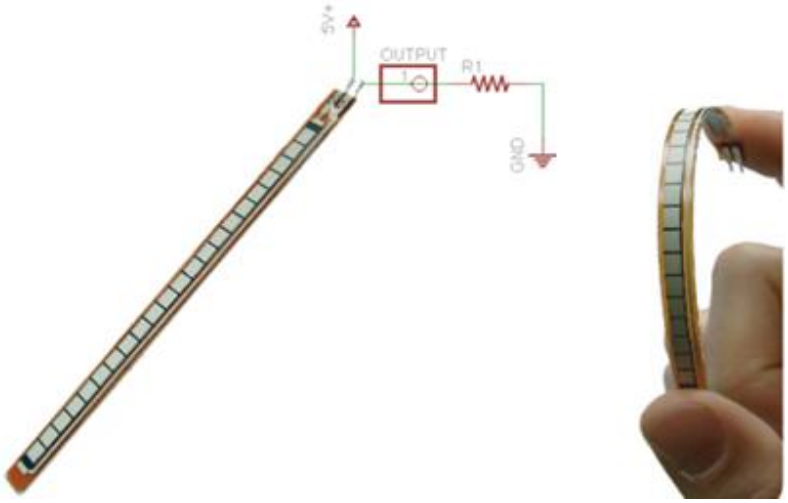
Port input/output digunakan sebagai jalur keluar masuk data mikrokontroler. *Port* yang digunakan sebagai *input* adalah *portA*, yaitu: PA0 – PA7 yang membaca *input* dari sensor *flex*. *Port* yang digunakan sebagai *output* adalah *portB*, *portC* dan *portD*. Cara menggunakan *port output* ini pada program utama menuliskan sintaks deklarasi "*Config Portx.n = Output"*. Keterangan sintaks tersebut adalah: *"x"* sebagai keterangan *port* yang digunakan (B, C atau D) dan *"n"* sebagai keterangan nomor *port* (0 – 7). Setelah deklarasi *port* tersebut, maka *port output* dapat digunakan. [1]

2.2.3 Port Komunikasi Serial

PortD digunakan sebagai *output* dan *input* data mikrokontroler untuk komunikasi data serial antara dua mikrokontroler. PD0 sebagai Rx atau penerima data dan PD1 sebagai Tx atau pengirim data. Komunikasi serial pada mikrokontroler ini disebut dengan USART (*Universal Synchronous Asynchronous Receiver Transmitter).* Komunikasi serial ini bertujuan agar mikrokontroler dapat berkomunikasi secara serial dengan perangkat lain, seperti mikrokontroler dapat

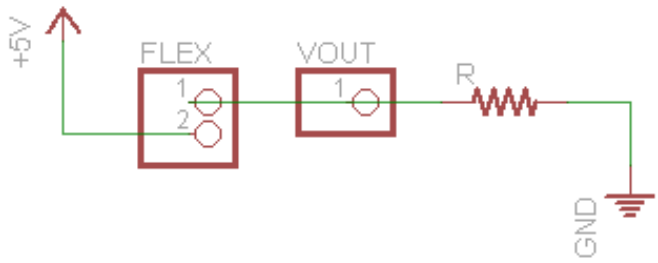
berkomunikasi dengan mikrokontroler lain, dengan perangkat komputer dan masih banyak perangkat lainnya. [1]

2.3 Sensor Flex



Gambar 2.5. Sensor Flex

Sensor *flex* adalah sensor gerak dengan cara kerja mengeluarkan perubahan resistansi akibat adanya perubahan lekukan pada kontur sensor. Sensor *flex* yang digunakan berukuran 4,5 inchi memiliki 2 kaki pin, dengan bentuk fisik tipis memanjang dan lentur. Sensor ini mengeluarkan *output* berupa resistansi. Dua pin kaki sensor tersebut, jika salah satu pin diberikan tegangan +5 volt maka pin yang lainnya sebagai *output* serta tegangan 0 volt. Prinsip kerja sensor *flex* ini mirip dengan variabel resistor. Sensor *flex* akan memberikan resistansi kepada mikrokontroler melalui rangkaian pembagi tegangan. [3]



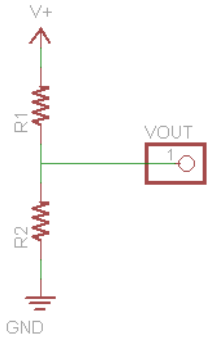
Gambar 2.6. Rangkaian Pembagi Tegangan Sensor Flex

Adapun spesifikasi yang terdapat pada sensor *flex* yaitu: [4]

- Resistansi *flat* 10Kohm ± 30 %
- Rentang lengkungan kontur antara 45 KOhm – 125 KOhm

2.4 Pembagi Tegangan

Rangkaian pembagi tegangan berfungsi untuk membagi tegangan *input* pada bagian *output* tertentu. Persamaan pembagi tegangan adalah sebagai berikut: [5]



Gambar 2.7. Rangkaian Pembagi Tegangan

$$V_{out} = \frac{R_2}{R_1+R_2} \times V_s$$

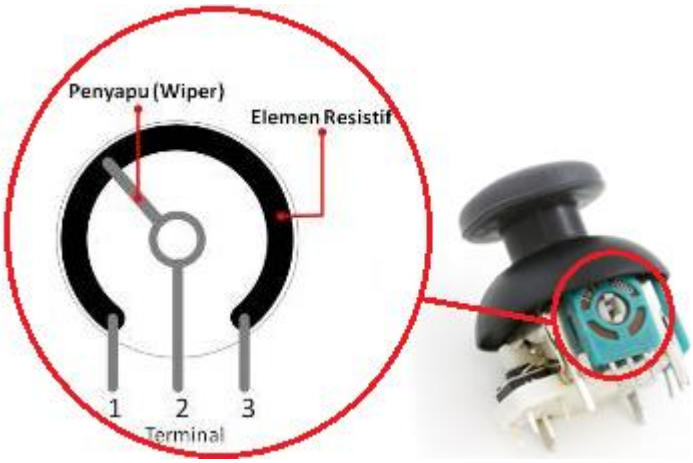
Dimana : V_{out} = *output* tegangan yang dicari

R_1 = resistor 1

R_2 = resistor 2

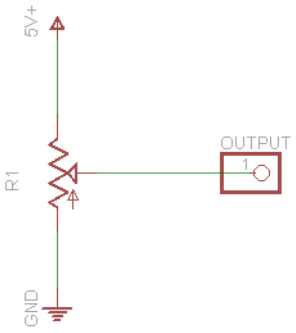
V_s = tegangan *input*

2.5 Potensiometer (*joystick*)



Gambar 2.8. Potensiometer [4]

Potensiometer adalah salah satu jenis resistor dimana nilai resistansinya dapat diatur sesuai dengan kebutuhan rangkaian elektronika. Potensiometer termasuk kedalam jenis variabel resistor karena nilai *output* resistansinya dapat diubah-ubah. Struktur bentuknya terdiri dari 3 buah kaki pin dan tuas sebagai pengatur nilai resistansinya. Komponen elektronika ini jarang digunakan untuk mengendalikan daya tinggi secara langsung (> 1 watt). Biasanya hanya digunakan untuk menyetel sinyal tegangan analog dan pengendali *input* untuk rangkaian elektronik.



Gambar 2.9. Rangkaian Potensiometer

Potensiometer terdiri dari sebuah elemen resistif yang membentuk lingkaran dengan terminal pada kedua ujungnya. Elemen resistif ini umumnya terbuat dari bahan campuran metal dan keramik ataupun bahan karbon. Sedangkan terminal lainnya berupa *wiper* atau tuas kecil yang digunakan untuk menentukan

pergerakan pada jalur elemen resistif tadi. Pergerakan *wiper* pada jalur resistif inilah yang akan mengatur besar kecilnya nilai resistansi *output* pada sebuah potensiometer.

2.6 Motor Servo



Gambar 2.10. Motor Servo [6]

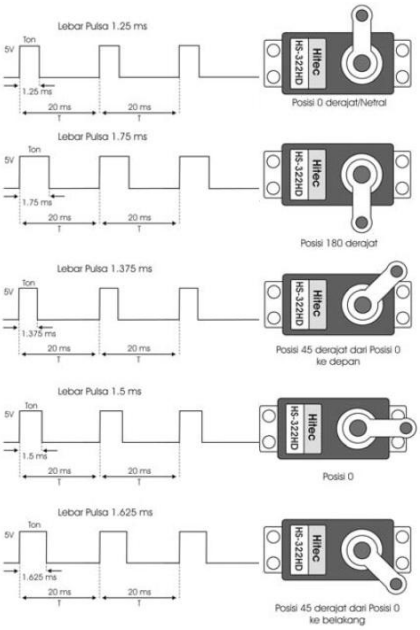
Motor Servo adalah sebuah motor dengan sistem *closed feedback* dimana posisi dari motor akan diinformasikan kembali pada rangkaian kontrol yang ada di dalam motor servo. Struktur perangkatnya terdiri dari sebuah motor, rangkaian gigi, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran yang dihasilkan. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirimkan melalui kaki sinyal yang telah disediakan. Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak bergerak kontinyu seperti motor DC maupun motor *stepper*. Tetapi motor servo dapat dimodifikasi agar dapat bergerak kontinyu. Pada sebuah robot, motor servo ini sering digunakan untuk bagian kaki, lengan dan bagian lainnya yang bergerak terbatas dan sangat membutuhkan torsi cukup besar. Motor servo bergerak dua arah, yaitu searah jarum jam dan berlawanan arah jarum jam. Arah dan sudut dari pergerakan rotornya dapat dikendalikan dengan memberikan pengaturan sinyal *pulseout* dari mikrokontroler pada pin bagian sinyal motor servo tersebut.

Motor servo termasuk kedalam jenis motor DC yang memiliki rangkaian kontrol elektronik dan internal *gear* untuk mengendalikan pergerakan sudutnya. Putaran motor servo relatif lambat, tetapi memiliki torsi cukup kuat karena memiliki rangkaian internal *gear*.

Berikut spesifikasi utama sebuah motor servo: [6]

- Terdapat tiga kabel, yaitu: *power*, *ground*, *control*
- Sinyal kontrol mengendalikan posisi motor
- Rangkaian internal terdapat *gear*, potensiometer, dan *feedback control*.
- Operasional motor motor servo dikendalikan oleh pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms sampai dengan 2 ms merupakan rentang sudut maksimum.

Operasional motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari kisaran sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan 90^o, sedangkan apabila diberikan pulsa kurang dari 1.5 ms maka posisi mendekati 0^o dan apabila diberikan pulsa lebih dari 1.5 ms maka posisi mendekati 180^o. [6]

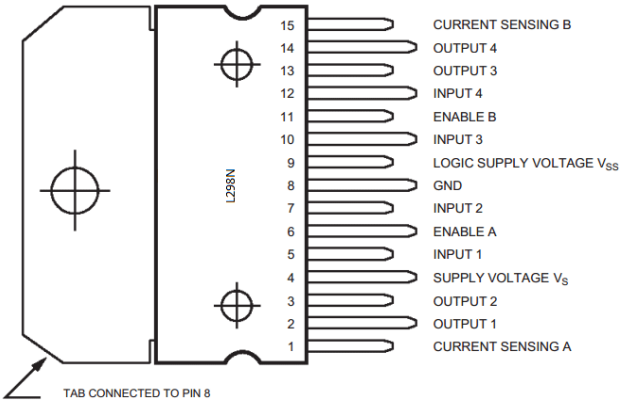


Gambar 2.11. Kendali Motor Servo [6]

Motor Servo akan bekerja dengan baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *Ton duty cycle* 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0°/ netral). Pada saat *Ton duty cycle* dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar

berlawanan arah jarum jam (*Counter Clock wise, CCW*) dengan membentuk sudut yang besarnya linier terhadap besarnya *Ton duty cycle*, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika *Ton duty cycle* dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (*Clock Wise, CW*) dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle*, dan bertahan pada posisi tersebut. [6]

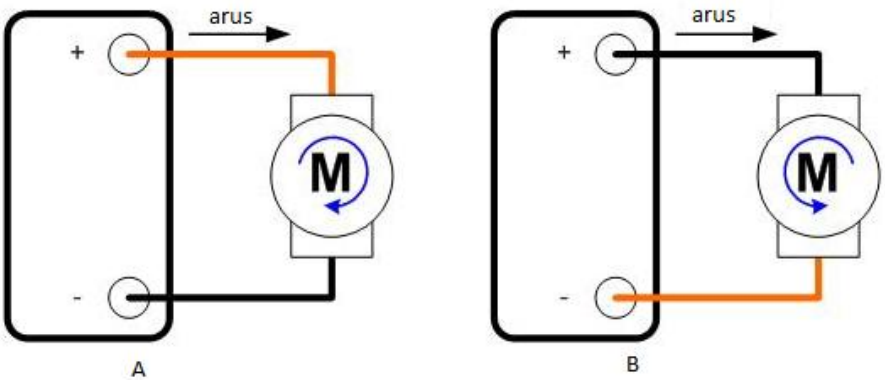
2.7 Driver Motor DC L298



Gambar 2.12. IC driver motor L298 [7]

L298 adalah jenis IC driver motor yang dapat mengendalikan arah putaran dan kecepatan motor DC. IC ini mampu mengeluarkan *output* tegangan untuk motor DC dan motor stepper sebesar 50 volt. IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang *NAND* yang memudahkan dalam menentukan arah putaran suatu motor DC. IC ini dapat mengendalikan 2 untuk motor DC masing-masing dapat menghantarkan arus sebesar 2A. Selain itu, IC ini juga dapat menggerakkan beban induktif seperti solenoid dan motor stepper. IC ini memiliki 2 buah pin *enable* untuk mengaktifkan atau mematikan sinyal *output* secara independen. [7]

2.8 Motor DC



Gambar 2.13. Motor DC [8]

Motor DC adalah Motor jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Polaritas dari tegangan yang diberikan pada dua kaki tegangannya menentukan arah putaran motor. Pada gambar A dengan memberikan polaritas positif, maka motor akan berputar searah jarum jam. Bila polaritas dari tegangan tersebut dibalik seperti pada gambar B, maka arah putaran motor akan terbalik berlawanan arah jarum jam. Kecepatan putaran motor akan ditentukan oleh besarnya tegangan yang diberikan. Komponen motor DC memiliki 2 bagian dasar, yaitu bagian yang tetap/*stasioner* yang disebut *stator*. *Stator* ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen, sedangkan bagian yang berputar disebut rotor, yaitu berupa sebuah koil dimana arus listrik mengalir. Gaya elektromagnet pada motor DC timbul saat ada arus yang mengalir pada penghantar yang berada dalam medan magnet. Medan magnet itu sendiri ditimbulkan oleh megnet permanen. Garis-garis gaya magnet mengalir diantara dua kutub magnet dari kutub utara ke kutub selatan. [8]

2.9 Modul wireless xbee pro S2

Modul *wireless* Xbee pro atau yang sering disebut dengan modul Zigbee merupakan modul *transceiver*. Pengirim dan penerima frekuensi radio ini berfungsi untuk komunikasi secara *full duplex*. Salah satu modul komunikasi *wireless* dengan frekuensi 2,4 GHz adalah Xbee Pro OEM Zigbee/IEEE 802.15.4

2,4 GHZ. Radio frekuensi *transceiver* ini merupakan sebuah modul yang terdiri dari RF *transmitter* dan RF *receiver* dengan sistem *interface* serial UART. [9]



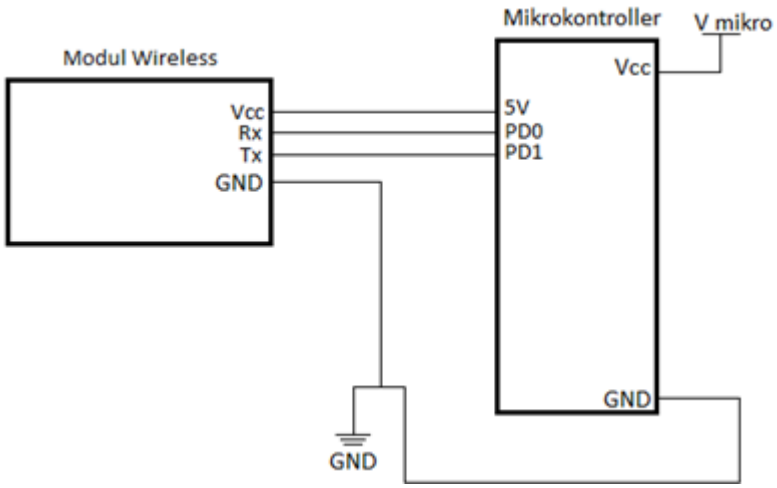
Gambar 2.14. Modul Wireless [9]

Modul Zigbee ini mempunyai 20 pin (kaki) dengan koneksi minimum agar modul ini dapat bekerja adalah pin VCC (pin 1), GND (pin 10), DOUT (pin 2),dan DIN (pin 3). Tegangan VCC yang digunakan adalah 3,3 volt. Pin Tx dan Rx dari mikrokontroler dapat dikoneksikan langsung ke pin DIN dan DOUT pada Zigbee. Data yang masuk ke Zigbee melalui DIN akan disimpan terlebih dahulu di DI *Buffer* dan RF TX *Buffer* sebelum ditransmisikan melalui *port* antena menuju Zigbee lainnya. Begitu juga sebaliknya dengan data yang diterima melalui *port* antena. [9]

Fitur-fitur yang terdapat pada Xbee Seri 2 antara lain : [9]

- Xbee Pro S2 kabel antena
- Xbee modul
- Tegangan minimum 3,3 volt @ 40 mA
- Mempunyai pin *output* yang sama dengan Xbee seri 1
- Data rate maksimum 250 Kbps
- Antar muka standart UART : 9600, 8-N-1
- 2mW *output* (+3 dBm)
- Jarak jangkauan nirkabel sebesar 400ft (120m)
- 6 pin *input* @ 10-bit ADC
- 8 pin I/O digital
- 128-bit enkripsi.

Berikut ini adalah gambaran mengenai penggunaan modul *wireless* Zigbee, yaitu:



Gambar 2.15. Modul Wireless

Gambar di atas adalah ilustrasi bagaimana cara penggunaan modul *wireless* pada salah satu mikrokontroler untuk mengirimkan data. Terdapat 4 pin yang dihubungkan dari modul *wireless* pada mikrokontroler, pin Vcc modul *wireless* dihubungkan pada pin +5 volt mikrokontroler, pin Rx modul *wireless* dihubungkan pada pin PD0 mikrokontroler, pin Tx modul *wireless* dihubungkan pada pin PD1 mikrokontroler dan pin *ground* modul *wireless* dihubungkan pada *ground* mikrokontroler.

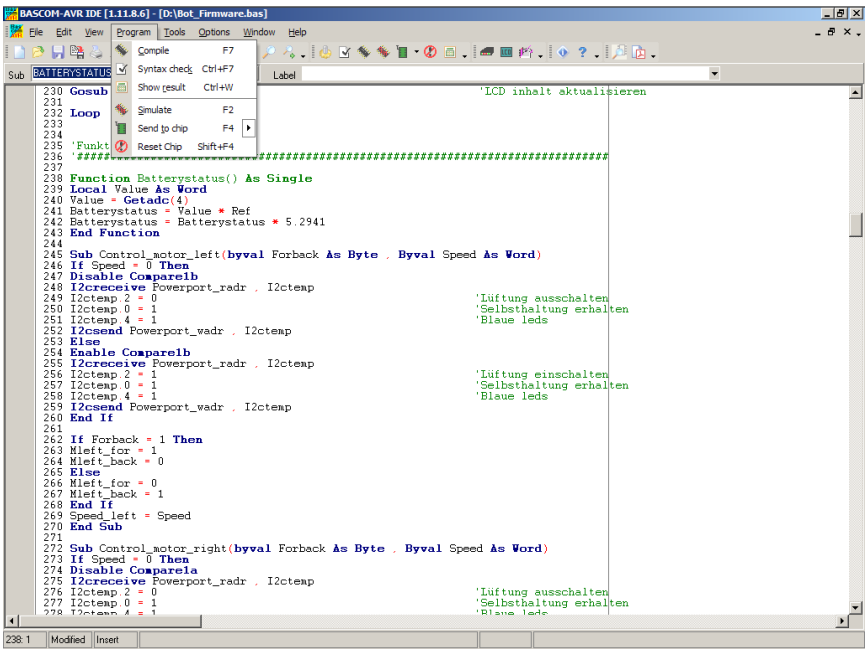
2.10 Pemrograman BASCOM

Bascom dikembangkan oleh MCS *Electronics* dan merupakan *Basic Compiler*. Program yang dibuat dalam bahasa *Basic* akan dikompilasi menjadi *machine code* untuk kemudian dimasukkan ke dalam mikrokontroler melalui sebuah pemrograman. Saat ini, sesuai dengan referensi dari situs web MCS *elektronics*, *bascom* baru mendukung mikrokontroler keluarga MCS51 (BASCOM-8051) dan keluarga AVR (BASCOM-AVR), dimana keduanya merupakan produk dari *Atmel Corporation*. Untuk chip kelas AT89S51/51 dapat diprogram secara ISP *(in System Programming)*.



Gambar 2.16. Tampilan awal BASCOM AVR [10]

BASCOM AVR merupakan salah satu *tool* untuk pengembangan atau pembuatan program untuk kemudian ditanamkan dan dijalankan pada mikrokontroler, terutama mikrokontroler keluarga AVR. BASCOM AVR disebut juga sebagai IDE *(Integrated Development Environment)* yaitu lingkungan kerja yang terintegrasi, karena selain tugas utamanya (melakukan kompilasi kode program menjadi file HEX / bahasa mesin), BASCOM AVR juga memiliki kemampuan atau fitur lain yang berguna sekali dimana sebagai terminal (monitoring komunikasi serial) dan sebagai programmer (untuk menanamkan program yang sudah dikompilasi ke mikrokontroler).



Gambar 2.17. Contoh Source Code BASCOM AVR [10]

Pada Pembuatan sebuah program secara umum, dapat dibagi menjadi 3 bagian penting, yaitu :

1. *Header* Program

   Header ditulis paling awal *listing* program yang dibuat. Bagian ini menentukan mikrokontroler jenis dan tipe apa yang digunakan, menentukan *baud rate* yang akan digunakan dan mencantumkan nilai frekuensi XTAL yang digunakan pada mikrokontroler prosesor. Contoh sintaks sebagai berikut:

   ```
   ‘$regfile = "m16def.dat"
   ‘$baud = 9600
   ‘$crystal = 8000000
   ```

2. Deskripsi *variable* program

   Pada bagian deskripsi *variable* harus mencantumkan keterangan *port* apa saja yang akan digunakan sebagai *output*, menentukan konfigurasi fitur apa yang akan digunakan pada mikrokontroler, serta mencantumkan *variable* dengan tipe data yang akan digunakan. Contoh sintaks sebagai berikut:

   ```
   ‘Config Servos = 2 , Servo1 = Portc.0 , Servo2 = Portc.1 , Reload = 10
   ‘Config Portc.0 = Output
   ‘Config Portc.1 = Output
   ‘Config Adc = Single , Prescaler = Auto , Reference = Avcc
   Dim Jr1 As Word , Jr2 As Word
   ```

3. Program Utama

   Pada bagian program utama mengetikkan sintaks program yang akan dieksekusi untuk dimasukkan ke dalam mikrokontroler. Langkah-langkah pengetikan sintaks dalat dilihat pada menu *help* yang akan membatu dalam proses pembuatan program secara keseluruhan. Contoh sintaks sebagai berikut:

   ```
   ‘Do
   ‘a = Getadc(0)
   ‘If a > 4500 Then
   ‘a = 4500
   ‘End If
   ‘Print “  ” ; a
   ‘Waitms 10
   ‘Loop
   ‘End
   ```