

BAB II

TINJAUAN PUSTAKA

2.1 Robot

Robot merupakan salah satu aplikasi ilmu pengetahuan modern yang fungsi umumnya dikontrol dari peralatan yang sudah terprogram. Kata robot diperkenalkan pertama kali oleh Karel Capek, kata robot berasal dari bahasa Czech (Ceko) “robot” yang artinya pekerja, Karel Capek adalah seorang penulis drama yang berasal dari Czech (Ceko), yang membuat pertunjukan komedi pada tahun 1921 yang berjudul Rossum’s Universal Robots (RUR). Karel Capek bercerita tentang mesin yang sangat mirip seperti manusia namun bedanya mesin ini mampu bekerja terus menerus tanpa lelah. Lalu pada tahun 1926 dibuat sebuah film robot pertama Jerman yang berjudul Metropolis, film ini mengisahkan tentang robot yang dapat berjalan mirip seperti manusia beserta hewan peliharaannya.

Penelitian dan pengembangan pertama yang menghasilkan produk robotik dimulai tahun 1940-an menurut fu[3] dalam bukunya yaitu oleh sebuah laboratorium di Amerika (Argonne National Laboratories) yang memperkenalkan sebuah mekanisme robotik yang diberi nama *master slave manipulator*. Alat ini digunakan untuk menangani material radioaktif. Lalu di tahun 1950-an diperkenalkan ke publik robot komersial pertama oleh perusahaan Unimation Cooperated Amerika. Lalu kemudian pada tahun 1960-an robot berkembang di seluruh dunia untuk mengikuti kebutuhan didalam dunia industri. Dan di tahun 1980-an teknologi *mobile robot* mulai dikembangkan secara meluas ke negara-negara maju dan berkembang.

Penggunaan istilah robot sendiri biasanya digunakan untuk menjelaskan berbagai jenis mesin yang memiliki kemampuan untuk bergerak dan dapat digunakan untuk melakukan pekerjaan fisik. Model gerak yang bisa dilakukan oleh robot dibedakan menjadi *holonomic* dan *non holonomic*. Gerak *holonomic* yaitu posisi badan robot dibuat tidak berpindah tempat diletakkan planar sejajar permukaan bumi, badan robot dan bagian ujung robot dapat bergerak bebas untuk menjangkau daerah kerja robot. Sedangkan yang dimaksud gerak *non holonomic* yaitu robot tidak dapat langsung bergerak ke segala arah, tetapi harus melakukan gerakan tertentu terlebih dahulu agar dapat menuju kearah yang dituju.

Dan berdasarkan struktur dan fungsi fisiknya robot diklasifikasikan menjadi *Mobile Robot* dan *Non Mobile Robot (Manipulator Robot)*, yang akan dijelaskan berikut ini.

2.1.1 Mobile Robot

Mobile Robot merupakan robot yang dapat melakukan gerakan berpindah tempat.

Struktur *Mobile Robot* yang bergerak dan dapat berpindah tempat terdiri dari :

- a. Sistem pengendali pergerakan merupakan gabungan antara algoritma program dan peralatan mekanik yang dibuat yang secara langsung memberi perintah kepada robot untuk bergerak sesuai dengan kondisi masukan dan umpan balik yang diterima.
- b. Sistem sensor yang merupakan bagian yang berfungsi untuk mengenali kondisi lingkungan yang akan menjadi informasi umpan balik pada pengendali robot.
- c. Sistem aktuator yang terhubung ke peralatan mekanik robot menjadi alat penggerak robot dan membentuk konstruksi fisik robot.

Mobile Robot diklasifikasikan menjadi robot beroda dan robot berkaki. Perbedaan keduanya adalah pada sistem pergerakannya, yaitu di bagian mekanik robot yang terhubung langsung dengan aktuator yang terkontrol. Untuk robot beroda, aktuator dihubungkan ke mekanik roda dan untuk robot berkaki, aktuator dihubungkan ke alat gerak yang bentuk mekaniknya didesain menyerupai kaki. Dalam perancangan *mobile robot* jenis robot beroda maka struktur dan bentuk fisiknya disesuaikan dengan alat gerak *mobile robot* berupa roda. Gerakan berpindah tempat yang bisa dilakukan oleh robot beroda termasuk ke dalam jenis gerak *non holonomic*.

Terdapat beberapa jenis konstruksi robot beroda yaitu:

- a. *Single Wheel Drive* adalah sistem kemudi roda robot yaitu menggunakan dua buah roda biasa pada bagian belakang robot dan satu roda yang dihubungkan dengan motor stepper untuk mengontrol kemudi gerak robot yang ditempatkan pada bagian depan tengah badan robot.
- b. *Differential Drive* adalah sistem kemudi roda robot dengan menggunakan dua buah roda yang masing-masing terhubung dengan motor stepper sebagai kontrol kemudi gerak robot dan dua buah roda bebas.
- c. *Ackerman Steering* adalah sistem kemudi roda robot dengan menggunakan dua buah roda yang masing-masing terhubung dengan motor stepper sebagai kontrol kemudi gerak robot dan dua buah roda yang saling terhubung yang terletak pada bagian depan badan robot.

2.1.2 Manipulator Robot

Manipulator Robot adalah robot yang sengaja dibuat dengan desain khusus untuk melakukan fungsi tertentu, sehingga robot dapat memanipulasi objek. *Manipulator Robot* biasa digunakan dalam aplikasi industri, pertanian dan kesehatan. Struktur *manipulator robot* terdiri dari :

- a. Sistem pengontrol sebagai pengendali pergerakan merupakan perpaduan antara algoritma program dan rangkaian elektronik berbasis mikroprosesor yang berfungsi secara langsung dapat memberi perintah kepada robot untuk bergerak sesuai dengan kondisi masukan yang diterima ketika melakukan fungsi kerja tertentu.
- b. Manipulator merupakan bagian mekanik yang bisa digunakan untuk memindah, mengangkat dan memanipulasi benda kerja.
- c. Sistem sensor berfungsi sebagai pemberi informasi tentang berbagai keadaan dari bagian-bagian manipulator.
- d. Sistem aktuator sebagai komponen penggerak fisik robot.

2.2 Sensor

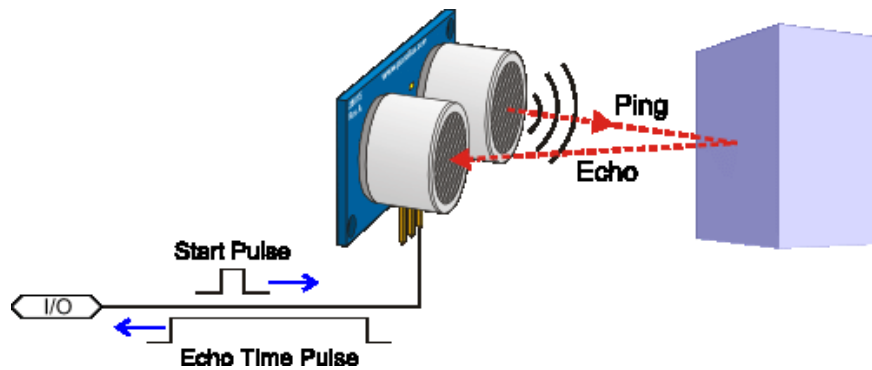
Sensor adalah alat/komponen elektronika yang berfungsi untuk mengukur kondisi fisik tertentu yang kemudian menjadi sinyal yang bisa dibaca oleh sistem pada robot[4]. Terdapat klasifikasi sensor berdasarkan tipe keluaran yang dihasilkan yaitu sensor biner yang akan menghasilkan keluaran 1 (on) atau 0 (off) saja, sensor analog yang keluarannya berbentuk analog dan sensor yang keluarannya menghasilkan pulsa seperti *gyroscope*.

Pada perancangan robot beroda, sebagai paramater untuk menentukan gerakan perpindahan robot selanjutnya adalah deteksi terhadap objek dinding/penghalang yang terdapat di sekitar robot. Sehingga sensor yang digunakan adalah jenis sensor biner, dimana sensor akan memberikan logika tertentu kepada prosessor pengontrol yaitu memberikan logika 1 (satu) dalam kondisi tidak terdeteksi adanya penghalang dan logika 0 (nol) jika dideteksi terdapat objek penghalang.

2.2.1 Sensor Ultrasonik

Sensor ultrasonik merupakan sensor yang mempunyai prinsip kerja menggunakan pantulan gelombang suara serta dapat digunakan untuk mendeteksi keberadaan suatu objek atau benda tertentu didalam cakupan frekuensi pada daerah diatas gelombang suara dari 20 kHz hingga 2 MHz[5]. Terdapat unit pemancar dan unit penerima pada sensor ultrasonik. Prinsip kerjanya sederhana yaitu sebuah kristal piezoelectric dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar tegangan bolak-balik yang mempunyai frekuensi kerja 20 kHz hingga 2 MHz. Keadaan kontraksi mengembang atau menyusut adalah penyebab dari struktur atom dari kristal piezoelectric, sebuah polaritas tegangan yang diberikan dan ini disebut dengan efek piezoelectric pada sensor ultrasonik.

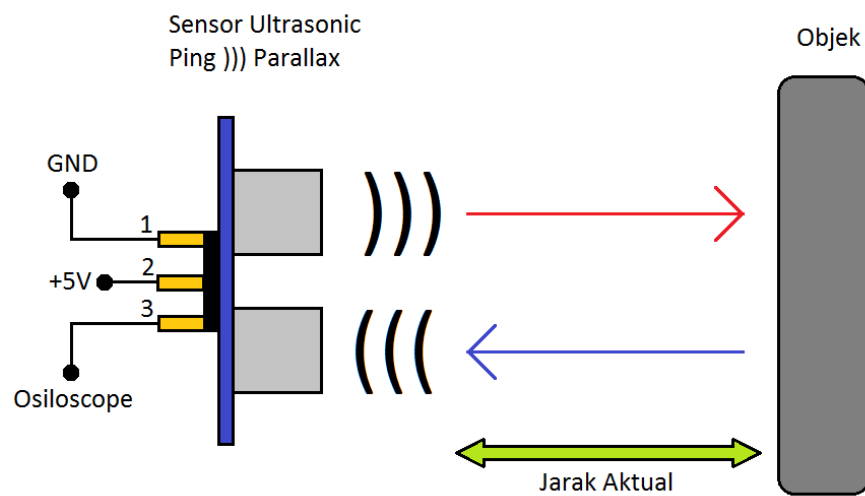
Ketika terdapat objek tertentu di depan sensor maka pantulan sinyal ultrasonik bereaksi dan pantulan sinyal ultrasonik akan diterima kembali oleh unit penerima[6]. Efek piezoelectric akan menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang sama dan unit penerima akan mengakibatkan diafragma penggetar menjadi bergetar. Prinsip kerja dari sensor ultrasonik dapat pada gambar 2.1 berikut ini :



Gambar 2.1 Prinsip Sensor Ultrasonik

Sumber: <https://images.app.goo.gl/aKjS9TJa4gWjvpmN9>

Besarnya amplitudo suatu sinyal elektrik yang dihasilkan oleh sensor penerima tergantung dari jauh dekatnya sebuah objek yang akan dideteksi. Proses pembacaan sensor yang dilakukan terhadap sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan objek sasaran. Untuk lebih jelasnya, prinsip pemantulan sensor ultrasonik dapat dilihat pada gambar 2.2 berikut ini:



Gambar 2.2 Prinsip Pemantulan Ultrasonik

Sumber: <https://images.app.goo.gl/vEy8oThwGk4t5ksaA>

2.2.2 Sensor Ultrasonik HCSRFB-04

Prinsip kerja dari sensor hcsrf-04 adalah transmitter mengirimkan gelombang ultrasonik kemudian diukur dengan waktu yang dibutuhkan sampai datangnya pantulan dari objek. Lamanya waktu ini sama dengan dua kali jarak sensor dengan objek, sehingga jarak sensor dengan objek dapat ditentukan melalui persamaan 1 berikut:

$$s = \frac{v \times t}{2} \dots \dots \dots (2.1)$$

Keterangan: s = jarak (meter)

v = kecepatan suara (344 m/detik)

t = waktu tempuh (detik)

Hcsrf-04 dapat berfungsi untuk mengukur jarak pada rentang antara 3cm–3m dengan keluaran panjang pulsa yang sama dengan jarak objek. Sensor ini hanya membutuhkan 2 pin I/O untuk dapat berkomunikasi dengan mikrokontroler, yaitu TRIGGER dan ECHO. HCSRFB-04 aktif apabila mikrokontroler mengirimkan gelombang positif melalui pin TRIGGER minimal selama 10 μ s, kemudian HCSRFB-04 akan mengirimkan gelombang positif melalui pin ECHO selama 100 μ s hingga 18 ms, yang sama dengan jarak objek. Spesifikasi dari sensor ultrasonik HCSRFB-04 adalah sebagai berikut:

- a. Dimensi Sensor: 24mm (P) x 20mm (L) x 17mm (T).
- b. Konsumsi Arus Sensor: 30 mA (rata-rata), 50 mA (max).

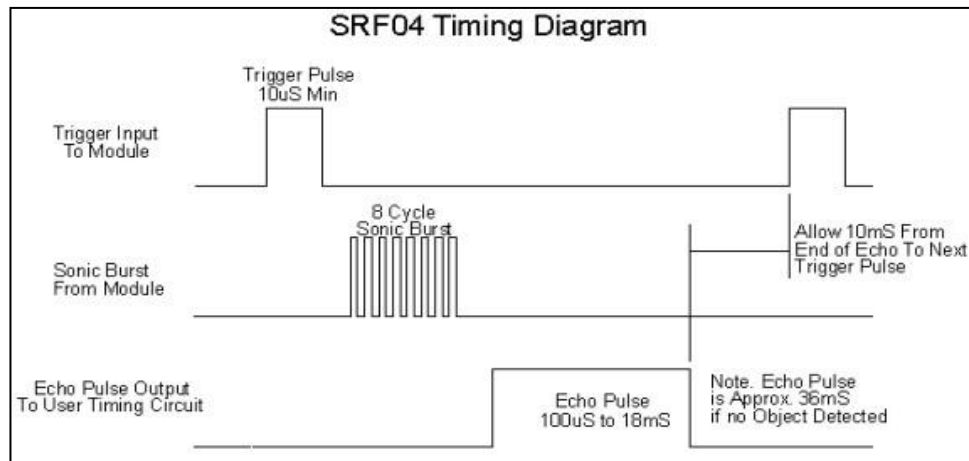
- c. Jangkauan Sensor: 3 cm–3 m.
- d. Sensitifitas Sensor: Mampu mendeteksi objek dengan diameter 3 cm pada jarak > 1m.



Gambar 2.3 Sensor Ultrasonik HCSRFB-04
Sumber: <https://images.app.goo.gl/sjR3KRMqKZxtxYaM8>

2.2.3 Prinsip Kerja Sensor Ultrasonik HCSRFB-04

HCSRFB-04 mempunyai prinsip kerja yaitu jika transmitter memancarkan sinyal ultrasonik (20 KHz), kemudian apabila didepan HCSRFB-04 terdapat objek padat maka receiver akan menerima pantulan sinyal ultrasonik tersebut. Objek yang dipantulkan dan selisih waktu pemancaran akan dibaca lebar gelombangnya (dalam bentuk PWM) oleh receiver. Jarak objek didepan sensor dapat diketahui dengan melakukan pengukuran tersebut. Perhatikan gambar 2.4 berikut:



Gambar 2.4 Timing Diagram Sensor Ultrasonik HCSR04
 Sumber: <https://images.app.goo.gl/e9PuyhnnvKTuYo27>

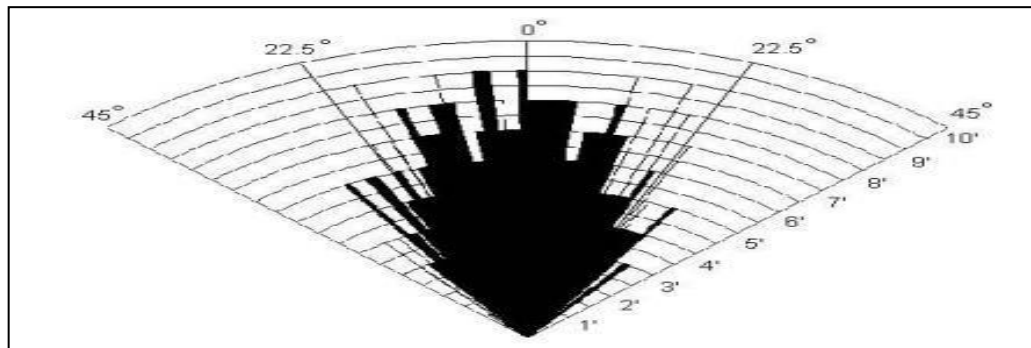
Pin trigger dan echo dihubungkan ke mikrokontroler. Untuk memulai pengukuran jarak mikrokontroler akan mengeluarkan output high pada pin trigger minimal selama 10 μs sinyal high yang masuk sehingga membuat sensor HCSR04 ini mengeluarkan gelombang suara ultrasonik. Suara tersebut akan diterima dan mengakibatkan output sinyal high pada pin echo setelah suara ultrasonik dipantulkan kembali ke sensor HCSR04 yang kemudian inputan pada mikrokontroler HCSR04 akan memberikan pulsa 100 μs - 18ms pada keluarannya tergantung pada informasi jarak pantulan objek yang diterima. Lamanya sinyal high dari pin echo inilah yang akan digunakan untuk menghitung jarak antara sensor HCSR04 dengan objek yang memantulkan bunyi yang berada didepan sensor.

Untuk menghitung lamanya sinyal high yang diterima mikrokontroler dari pin echo, maka bisa menggunakan fasilitas timer yang ada pada masing-masing mikrokontroler. Kemudian ketika ada perubahan dari low ke high dari pin echo maka hal tersebut akan mengaktifkan timer dan ketika ada perubahan dari high ke low dari

pin echo maka hal tersebut akan mematikan timer. Setelah itu selanjutnya adalah mengkonversi nilai timer dari yang satuannya dalam detik, menjadi ke dalam satuan jarak (inch/cm) dengan menggunakan rumus berikut:

- a. Jarak (inch) = waktu hasil pengukuran (us)/148
- b. Jarak (cm) = waktu hasil pengukuran (us)/58

Data perbandingan antara sudut pantulan dan jarak pada sensor ultrasonik Hcsr-04 dapat dilihat pada gambar 2.5 berikut.



Gambar 2.5 Pembeding Sudut Pantul

Sumber: <https://images.app.goo.gl/AZmvwW9kEYauGHhBA>

2.3 Mikrokontroler Arduino

Arduino adalah *platform* prototipe elektronik *open-source*, yang berdasarkan perangkat keras dan lunak yang fleksibel dan mudah digunakan[7]. Arduino diperuntukkan bagi seniman, desainer, hobiis dan siapa pun yang tertarik untuk membuat alat yang interaktif.

Arduino secara fisik adalah mikrokontroler. Arduino merupakan perangkat keras yang ukurannya kecil dan berbentuk rangkaian elektronik serta berfungsi sebagai

kontroler[8]. Dihubungkan dengan sensor yang akan memberikan informasi keadaan obyek atau lingkungan disekitarnya, kemudian mengolah informasi tersebut lalu menghasilkan suatu aksi. Proses ini akan dilakukan berulang-ulang. Dalam tugas akhir ini penulis menggunakan board mikrokontroler Arduino Mega 2560 R3.

2.3.1 Arduino Mega2560

Arduino mega 2560 adalah piranti mikrokontroler menggunakan Atmega2560. Modul ini memiliki 54 digital *input* dan *output*. Dimana 14 pin digunakan untuk PWM output dan 16 pin digunakan sebagai analog input, 4 pin untuk UART, 16 MHz osilator kristal, koneksi USB, *power jack ICSP header*, dan tombol reset. Modul ini memiliki segala yang dibutuhkan untuk memprogram mikrokontroler seperti kabel USB dan catu daya melalui adaptor atau baterai. Semua ini diberikan untuk mendukung pemakaian mikrokontroler Arduino, hanya terhubung ke komputer dengan kabel USB atau listrik dengan adaptor dari AC ke DC atau baterai untuk memulai pemakaian.



Gambar 2.6 Arduino Mega2560 R3

Sumber: <https://images.app.goo.gl/6HvLaJ754Dy6tNyA7>

Spesifikasi :

- Mikrokontroler : Atmega 2560
- *Operating Voltage* : 5V
- *Input Voltage (recommended)* : 7-12 V
- *Input Voltages (limits)* : 6-20 V
- *Digital I/O Pins* : 54 (15 PWM output)
- *Analog Input Pins* : 16
- *DC current for I/O pin* : 40 mA
- *DC current for 3.3 V pin* : 256 KB (8 KB digunakan untuk *bootloader*)
- *SRAM* : 8 KB
- *EEPROM* : 4 KB
- *Clock Speed* : 16 MHz

2.3.2 Bahasa Program Arduino

Pemrograman microcontroller arduino ini dilakaukan dengan menggunakan aplikasi arduino software (IDE). Bagian dari arduino development environment yaitu editor teks, sebuah area pesan, sebuah konsol, sebuah toolbar dan beberapa menu. Arduino Development Environment terhubung ke arduino board untuk memasukkan program dan juga untuk berkomunikasi dengan arduino board lainnya. Perangkat lunak yang penulisannya menggunakan Arduino Development Environment disebut sketch. Sketch disimpan dengan file berekstensi ino. Tombol yang ada pada *toolbar* digunakan untuk mengecek , *upload*, membuat, membuka, menyimpan dan menampilkan serial monitor.

Arduino menggunakan bahasa program dengan bahasa C. Bahasa pemrograman C tidak terlalu sulit, bahasa ini menggunakan fungsi-fungsi yang sederhana sehingga pemula bisa mempelajarinya dengan cukup mudah dan cepat. Penjelasan mengenai karakter bahasa C dan *software* Arduino berikut ini.

1. Struktur

Setiap pemrograman Arduino (*sketch*) memiliki dua buah *function* yang harus terdapat didalam setiap bahasa pemrograman pada Arduino.

- Void setup () {}

Semua kode yang berada didalam kurung kurawal akan dijalankan hanya satu kali pada saat program Arduino dijalankan untuk pertama kalinya.

- Void loop () {}

Function ini akan dijalankan ketika setup (*function void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan secara terus menerus samoaau catu daya dilepaskan.

2. Syntax

Elemen-elemen bahasa C yang dibutuhkan untuk format penulisan pemrograman Arduino Syntax

- // (komentar satu baris)

Kadang diperlukan untuk memberi catatan pada diri sendiri apa artu kode-kode yang dituliskan pada beberapa baris sebagai komentar. Apapun yang terletak setelah dua simbol tersebut yang berada satu baris akan diabaikan oleh program.

- `/* */` (komentar banyak baris)

Digunakan untuk membuat catatan yang banyak yang dapat dituliskan pada beberapa baris sebagai komentar. Apapun yang terletak berada di antara dua simbol tersebut akan diabaikan oleh program.

- `{ }` (kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai berakhir (digunakan juga pada fungsi dan pengulangan).

- `;` (titik koma)

Setiap baris kode pada program harus diakhiri dengan tanda titik koma (program tidak akan bisa dijalankan jika ada titik koma yang hilang).

3. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.

- `Int` (*integer*)

Berfungsi untuk menyimpan angka dalam 2 byte (16 bit). Tidak memiliki angka desimal dan menyimpan angka dari -32.768 dan 32.767.

- `Long` (*long*)

Digunakan ketika integer tidak mencukupi lagi. Menggunakan 4 byte (32 bit) dari memori (RAM) dan memiliki rentang angka dari -2.147.483.648 dan 2.147.483.646.

- `Boolean` (*boolean*)

Digunakan untuk menyimpan nilai *TRUE* (benar) atau *ELSE* (salah). Variabel ini hanya menggunakan 1 bit dari RAM.

- Float (*float*)

Digunakan untuk angka desimal (*floating point*). Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari $-3,4928235E+38$ dan $3,4028235E+38$.

- Char (*character*)

Menyimpan 1 karakter menggunakan kode program ASCII (misalnya „A”=65) memakai 1 byte (8 bit) saja dari RAM.

4. Operator Matematika

Operator matematika digunakan untuk manipulasi angka (bekerja seperti matematika yang sederhana).

- =

Menjadikan sesuatu menjadi sama dengan nilai yang lain (misalnya: $x=10*2$, maka x sama dengan 20).

- %

Menjadikan sisa dari hasil pembagian suatu nilai dengan nilai yang lain (contohnya: $12\% 10$, maka akan menghasilkan angka 2).

- +

Penjumlahan

- -

Pengurangan

- *

Perkalian

- /

Pembagian

5. Operator pembandingan

Digunakan untuk membandingkan nilai logika.

- ==

Sama dengan (contohnya: $12 == 10$ adalah *FALSE* (salah) atau $12 == 12$ adalah *TRUE* (benar)).

- !=

Sama dengan (contohnya: $12 != 10$ adalah *TRUE* (benar) atau $12 != 12$ adalah *FALSE* (salah)).

- <

Lebih kecil dari (contohnya: $12 < 10$ adalah *FALSE* (salah) atau $12 < 12$ adalah *FALSE* (salah) atau $12 < 14$ adalah *TRUE* (benar)).

- >

Lebih besar dari (contohnya: $12 > 10$ adalah *TRUE* (benar) atau $12 > 12$ adalah *FALSE* (salah)).

6. Struktur pengaturan

Program berjalan dengan bergantung pada pengaturan apa yang akan dijalankan setelahnya, berikut elemen dasar pengaturan.

1. if .. else, dengan format berikut ini:

2. `if (kondisi) { }`
3. `else if (kondisi) { }`
4. `else { }`

dengan susunan seperti diatas program akan menjalankan kode yang berada didalam kurung kurawal apabila kondisinya *TRUE*, dan apabila tidak *FALSE* ada kondisi pada else if akan diperiksa dan kode else yang akan dijalankan jika kondisinya *FALSE*.

1. for, dengan format seperti berikut ini:

```
for(int i = 0; i < #pengulangan; i++) { }
```

Digunakan bila ingin melakukan pengulangan kode didalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Untuk melakukan perhitungan ke atas menggunakan `i++` atau ke bawah menggunakan `i--`.

7. Digital

1. `pinMode(pin, mode)`

Dipakai untuk menetapkan pola dari suatu pin, pin yaitu nomor pin yang akan digunakan dari 0 – 19 (pin analog 0 – 5 adalah 14 – 19). Mode yang bisa digunakan adalah *input* atau *output*.

2. `digitalWrite(pin, value)`

Saat sebuah pin ditetapkan sebagai *output*, pin itu dapat dijadikan *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

3. `digitalRead(pin)`

Ketika sebuah pin ditetapkan sebagai *input* maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (ditarik 5 volts) atau *LOW* (diturunkan menjadi ground).

8. Analog

1. analogWrite(pin, value)

Beberapa pin pada Arduino yang dapat mendukung PWM (*pulse width modulation*) yaitu pin 3, 5, 6, 9, 10, 11. Pin ini bisa merubah pin hidup (*on*) atau mati (*off*) dengan cepat sehingga membuatnya dapat berfungsi sama halnya keluaran analog. Nilai pada pola kode tersebut yaitu angka antara 0 (0% *duty cycle* ~ 0V) dan 255 (100% *duty cycle* ~ 5V).

2. analogRead(pin)

Pada saat pin analog ditetapkan jadi *input* maka pin tersebut bisa membaca *output* voltasenya. Outputnya berupa angka antara 0 (untuk 0 volts) dan 1.024 (untuk 5 volts).

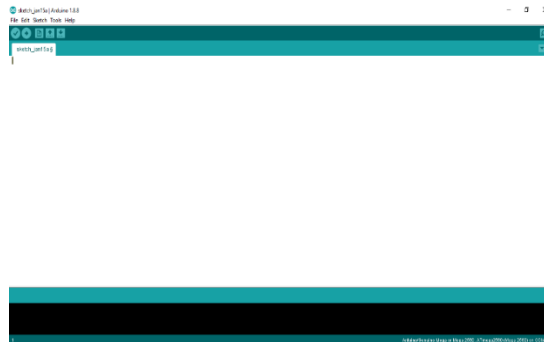
2.3.3 Software Arduino

Arduino diciptakan untuk seorang pemula dan sangat disarankan untuk yang tidak memiliki *basic* bahasa pemrograman sama sekali, karena pemrograman Arduino yang menggunakan bahasa C yang telah dipermudah oleh *library* (arduino.cc) sehingga membuat Arduino terbilang mudah untuk dipelajari.

Arduino menggunakan *software processing* yang dipakai untuk membuat program kedalam Arduino. Processing sendiri yaitu penggabungan antara bahasa C++ dan *java software*. *Software* Arduino ini dapat didownload diberbagai *operating*

system (OS) seperti: LINUX, Mac OS, dan Windows. Tiga bagian dari *Software* (Integrated Development Environment) IDE Arduino yaitu :

1. *Editor* program, yang berfungsi menulis dan mengedit program dalam bahasa *processing*.
2. *Compiler*, bagian yang berfungsi untuk mengubah bahasa *processing* (kode program) menjadi kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh *microcontroller*.
3. *Uploader*, modul yang berfungsi memasukan kode biner kedalam memori *microcontroller*.



Gambar 2.7 Software Arduino

2.4 Motor Servo

motor servo adalah jenis motor DC dengan sistem umpan balik tertutup yang terdiri dari sebuah motor DC, serangkaian gear, rangkaian kontrol, dan juga potensiometer yang di pasang agar motor servo dapat digerakan sesuai sudut. Potensiometer fungsinya untuk menentukan batas sudut dari putaran servo. Sementara sudut sumbu motor servo dapat disusun berdasarkan lebar pulsa yang dikirim oleh

kaki sinyal dari kabel servo itu sendiri. Sehingga motor servo dapat berputer searah dan berlawanan arah jarum jam.



Gambar 2.8 Motor Servo

Sumber: <https://images.app.goo.gl/A4JdwPnjQd5cAGqZ7>

Motor servo dapat menampilkan gerakan 0 derajat, 90 derajat, 180 derajat, hingga 360 derajat dengan mengatur pulsa yang di kirim ke motor servo tersebut. Motor servo juga mempunyai torsi yang besar sehingga dapat menopang beban yang cukup berat. Berikut jenis-jenis dari motor servo.

1. Motor servo Standard 180 derajat

Motor servo standar hanya mampu berputar 180 derajat. Motor servo jenis standard ini dapat berputar searah maupun berlawanan dengan arah jarum jam. Sesuai dengan namanya, sudut defleksinya hanya bisa mencapai 180 derajat, dengan perhitungan masing-masing sudut 90 derajat. Motor servo standar sering digunakan pada sistem robotika misalnya untuk membuat lengan robot.

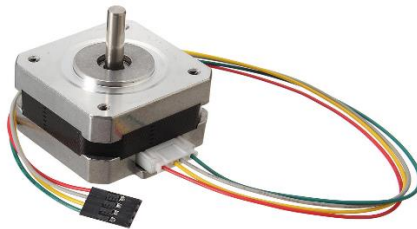
2. Motor Servo Continous

Motor servo continous dapat berputar 360 derajat. Motor servo ini bisa berputar searah maupun berlawanan arah jarum jam. Yang berbeda dengan motor servo standar yaitu sudut defleksi putarannya. Motor servo ini tidak mempunyai sudut

defleksi putaran melainkan berputar secara kontinyu. Motor servo continuous sering dipakai untuk Mobile Robot.

2.5 Motor Stepper

Motor stepper adalah sebuah motor listrik yang dapat mengubah pulsa listrik menjadi gerakan motor *discret* (terputus) yang bergerak dengan *step* (langkah)[9]. Satu putaran motor akan memerlukan 360° dengan jumlah langkah yang tertentu perderajatnya.



Gambar 2.9 Motor Stepper

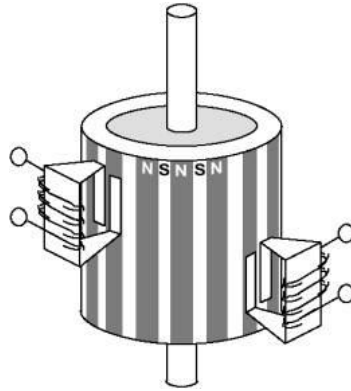
Sumber: <https://images.app.goo.gl/tXgvZ5mh82XKfmPw9>

Motor stepper akan bergerak berdasarkan pulsa yang diberikan kepada motor. Diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodic untuk menggerakkan motor stepper. Terdapat 3 tipe motor stepper yaitu :

1. Motor stepper tipe Permanent Magnet (PM)

Motor stepper tipe PM memiliki *rotor* yang membentuk kaleng bundar, motor ini terdiri dari lapisan magnet permanen yang disusun selang-seling dengan

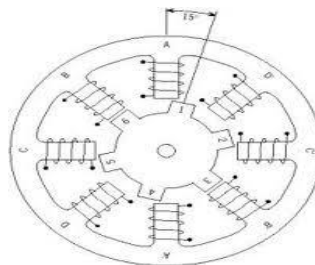
kutub yang berlawanan. Motor jenis ini biasanya mempunyai resolusi langkah (step) yang rendah yaitu antara $7,5^\circ$ hingga 15° per langkahnya atau 48 hingga 24 langkah setiap putarannya.



Gambar 2.10 Motor Stepper Permanent Magnet (PM)
 Sumber: <https://images.app.goo.gl/VW1JwphwFUNPbyWM9>

2. Motor stepper *Variable Reluctance* (VR)

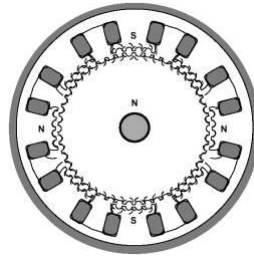
Motor stepper ini adalah motor stepper jenis lama dan merupakan jenis motor yang secara struktural paling mudah untuk dipahami. Motor tipe VR ini terdiri dari sebuah *rotor* besi lunak dengan beberapa gerigi dan sebuah lilitan *stator*. Dimana ketika lilitan *stator* diberikan energi dengan arus DC, kutub-kutub menjadi termagnetasi. Perputaran akan terjadi saat gigi-gigi rotor tertarik oleh kutub-kutub *stator*.



Gambar 2.11 Penampang Melintang Motor Stepper Variable Reluctance (VR)
 Sumber: <https://images.app.goo.gl/BPPwCw8PouCashMz9>

3. Motor Stepper tipe *Hybrid* (HB)

Motor tipe ini memiliki struktur yang merupakan kombinasi dari kedua tipe sebelumnya. Motor stepper tipe ini mempunyai gerigi seperti pada motor tipe VR dan juga mempunyai magnet permanen yang sudah terstruktur secara *aksial* pada batang porosnya sama seperti tipe motor tipe PM. Motor tipe ini dapat menghasilkan resolusi langkah yang tinggi yaitu antara $3,6^\circ$ hingga $0,9^\circ$ per langkah atau 100-400 langkah setiap putarannya.

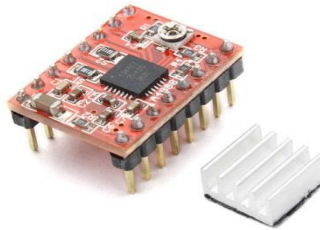


Gambar 2.12 Penampang Melintang Motor Stepper Hibrid
 Sumber: <https://images.app.goo.gl/XyRoKVxNAvH8Qdxt9>

2.6 Motor Driver Stepper

Motor driver IC A4988 dipilih sebagai motor driver pada perancangan alat ini karena tidak boros pin arduino dan tidak susah cara pengoperasiannya[10]. Motor driver pada umumnya hanya menaikkan tegangan output dari input arduino, hal ini sangat boros pin pada arduino mengingat tiap motor stepper bipolar memiliki 4 buah kabel. IC A4988 sangat mudah dipakai karena hanya terdapat dua buah input masukan yaitu DIR dan STEP. DIR berfungsi untuk mengubah arah putaran searah jarum jam dan berlawanan jarum jam. STEP berfungsi untuk mengatur kecepatan motor stepper dengan memberikan sinyal HIGH dan LOW dengan jedanya.

IC ini dirancang untuk mengendalikan berbagai macam motor stepper, baik motor stepper bipolar, penuh, setengah, atau seperempat. Dengan output hingga 35V dan $\pm 2A$. Bentuk IC A4988 dapat dilihat pada Gambar 2.6. IC A4988 memiliki 8 buah kaki yang terdapat pada sisi kanan dan sisi kirinya.



Gambar 2.13 Driver Motor A4988

Sumber: <https://images.app.goo.gl/XZhwZ8rtTvutxNvj7>

Pengaturan resolusi dari putaran motor terdapat pada pin MS1, MS2 dan MS3 yang dijelaskan pada Tabel 2.1.

Tabel 2.1 Pengaturan Resolusi Motor *Stepper*

MS1	MS2	MS3	Microstep Resolution	Stepper Revolution
L	L	L	Full Step	200
H	L	L	Half Step	400
H	H	L	Quarter Step	800
H	H	L	Eighth Step	1600
H	H	H	Sixteenth Step	3200