BAB 2
TINJAUAN PUSTAKA

2.1 Perangkat Keras (*Hardware*)

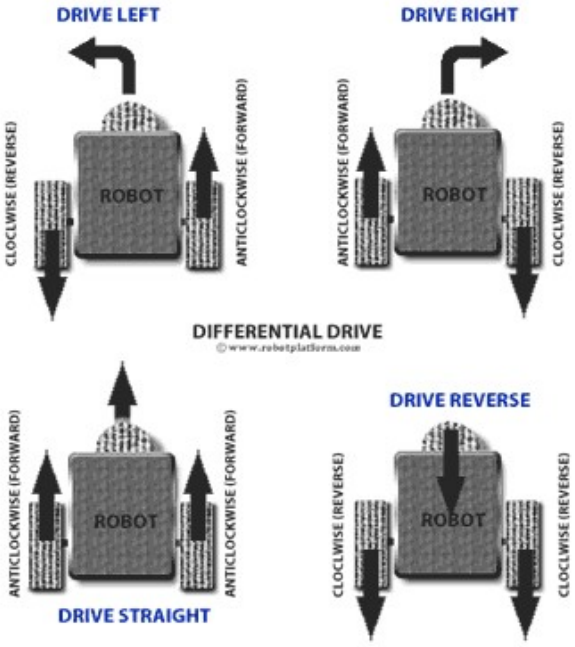
Dalam membangun sebuah robot pasti memerlukan perangkat keras yang terdiri atas bagian mekanik berupa fisik robot. Disamping itu dibutuhkan pula bagian elektronik berupa sensor-sensor serta komponen penunjang lainnya [4].

2.1.1 Sistem Kemudi *Mobile Robot*

Ada beberapa tipe sistem kemudi mobile robot. Berikut merupakan beberapa sistem kemudi *mobile* robot [5]:

1. *Differential Steering* (Sistem Kemudi Diferensial)

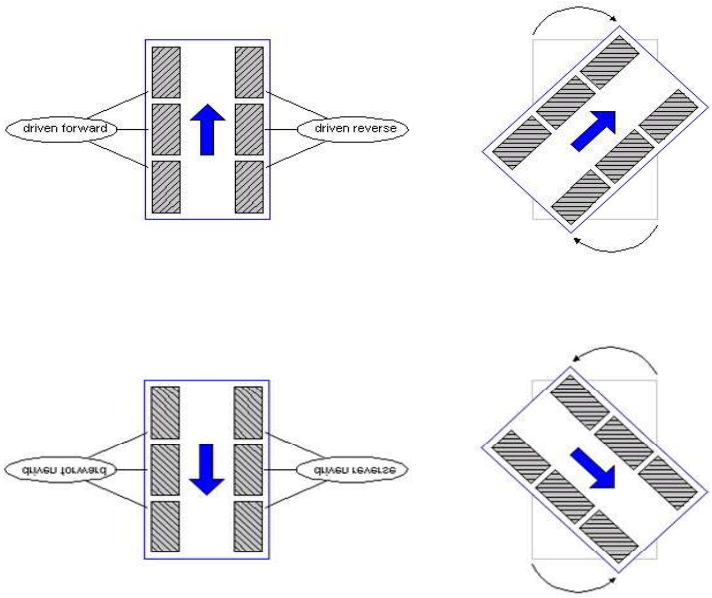
   Differential Steering merupakan metode kemudi dimana dua buah roda dapat bergerak berlawanan arah sehingga radius belokan lebih kecil. Hal ini meningkatkan kemampuan bermanuver robot lebih baik. Kelemahan dari tipe kemudi ini adalah menghasilkan gesekan pada roda kanan dan kiri saat melakukan gerakan berbelok. Sehingga akan bermasalah ketika robot bergerak di atas permukaan yang kasar atau tidak rata. Gambar 2.1 menunjukan sistem *Differential Steering*



Gambar 2.1 Sistem Differential Streering [6]

2. Skid Steering (Sistem Kemudi Selip)

   Pada dasarnya sistem kemudi ini sama seperti differential steering. Sistem kemudi ini menggunakan rantai roda atau susunan roda dengan posisi tetap. Kelemahannya adalah gesekan pada roda saat berbelok lebih besar daripada sistem differential steering. Gambar 2.2 menunjukan sistem kemudi selip

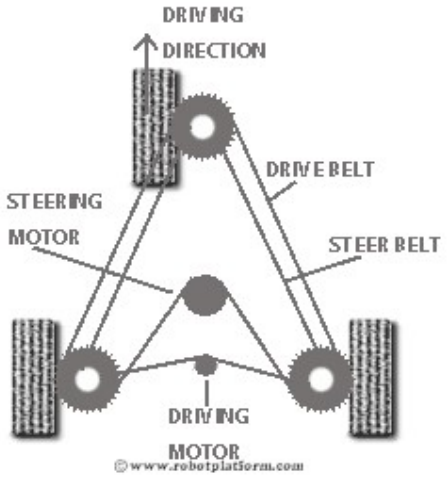


Gambar 2.2 Sistem Skid Steering

3. Car-Like Steering (Sistem Kemudi Tipe Mobil)

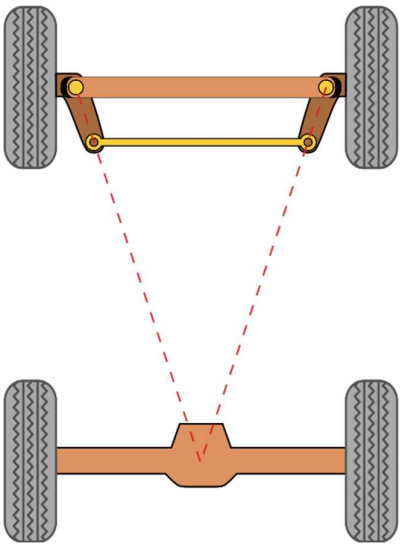
   Sistem kemudi ini merupakan konfigurasi dimana dua roda digunakan untuk mengubah arah robot. Biasanya roda depan yang digunakan meskipun tidak menutup kemungkinan untuk menggunakan roda belakang sebagai kemudinya. Sistem kemudi ini dibagi menjadi dua yaitu Kemudi Paralel dan Kemudi Ackermann.

   Kemudi Paralel merupakan sistem kemudi dimana roda yang digunakan untuk kemudi akan selalu sejajar satu sama lain tanpa dipengaruhi oleh sudut kemudi masing-masing roda. Permasalahan pada sistem kemudi ini yaitu pada saat berbelok, masing-masing roda bergerak pada poros yang berbeda. Hal ini mengakibatkan kedua roda mengalami peningkatan gesekan karena kedua roda berjalan pada jalur yang berbeda. Visualisasi sistem kemudi paralel seperti yang terlihat pada gambar 2.3.

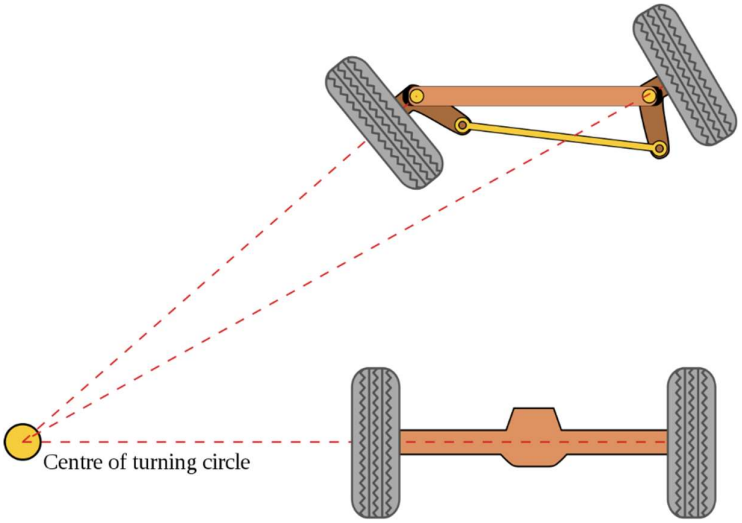


Gambar 2.3 Sistem Kemudi Paralel

Kemudi Ackermann mengatasi permasalahan pada kemudi paralel. Pada Sistem Kemudi Ackermann radius belok bisa kecil dan tidak terjadi selip. Kelemahan dari sistem car-like steering adalah tidak mampu berotasi terhadap sumbu vertikalnya. Selain itu, radius belok dibatasi oleh sudut belok roda depan dan jarak antara roda depan dan roda belakang. Visualisasi sistem kemudi Ackermann ditunjukkan oleh Gambar 2.4 Sistem Kemudi Ackermann.



Gambar 2.4 Sistem Kemudi Ackermann



Gambar 2.5 Alur Kemudi Ackerman

4. Multi Independent Steering (Sistem Kemudi Multi Independen)

   Sistem kemudi multi independent merupakan sistem kemudi dimana tiap roda dapat bergerak secara bebas. Dengan sistem kemudi ini robot dapat bergerak bebas ke segala arah dan merupakan tipe gerak

holomonic. Sebuah robot yang memiliki kemampuan gerak holomonic dapat melakukan semua jenis gerak belok tanpa harus mengubah posisi badan robot. Gambar 2.6 menunjukan sistem multi independent steering.



Gambar 2.6 Sistem Multi Independent Steering [6]

2.1.2 Microcontroller

Microcontroller adalah sebuah sistem komputer fungsional dalam sebuah *chip.* Fungsi dari *Microcontroller* adalah adalah mengerjakan instruksi-instruksi pada memori program, program merupakan imajinasi dari seorang *programmer*, dimana program merupakan instruksi-instruksi dari instruksi set dari CPU, program disimpan pada memori dan secara berurutan instruksi-instruksi tersebut dikerjakan oleh CPU. Sebuah *chip Microcontroller* umumnya memiliki fitur-fitur sebagai berikut:

a. *Central Processing Unit* (CPU)
b. *Input/output* antarmuka jaringan seperti serial port (UART)
c. Antarmuka komunikasi serial lainnya seperti seri*al peripheral interface* dan *controller area network*.
d. *Peripheral* seperti *timer* dan *watchdog*
e. *Random Access Memory* (RAM)
f. *Read Only Memory* (ROM), *Erasable Programmable Read Only Memory* (EPROM)

g. Pembangkit *clock*, biasanya berupa *resonator* rangkaian RC
h. Pembangkit analog ke digital

Microcontroller menggunakan *oscilator clock* yang berfungsi untuk memicu CPU dalam mengerjakan suatu instruksi ke instruksi selanjutnya dalam program yang berurutan. Setiap langkah operasi *microcontroller* memakan waktu beberapa clock untuk mengerjakan satu instruksi tergantung dari nilai dari *oscilator clock*.

Central Processing Unit (CPU) yang mengerjakan instruksi-instruksi yang diprogram oleh programmer, menginstruksikan CPU untuk membaca informasi dari piranti input, membaca informasi dan menulis informasi ke memori. Ada beberapa tipe dari memori di dalam microcontroller yang digunakan untuk beberapa tujuan yang berbeda, yang sering ditemui adalah *Random Access Memory* (RAM) dan *Read Only Memo*ry (ROM). RAM berfungsi sebagai tempat penyimpanan data sementara dan hasil kalkulasi selama proses operasi, ROM digunakan sebagai media penyimpanan data yang bersifat *volatile* yang berarti jika tidak ada tegangan pada microcontroller maka data tersebut tetap tersimpan [4].

2.1.3 Aktuator

Aktuator adalah bagian penggerak yang menghasilkan gerakan pada robot. Aktuator yang digunakan terdiri dari 2 jenis, yaitu : Motor DC dan motor servo.

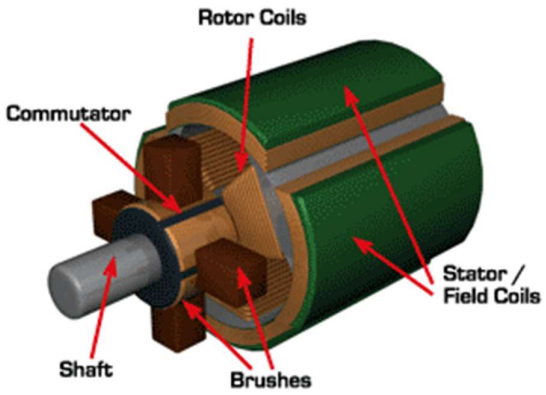
2.1.3.1 Motor DC

Motor DC merupakan alat yang mengubah pulsa listrik menjadi gerak, mempuyai prinsip dasar yang sama dengan motor stepper namun gerakannya bersifat kontinyu atau bekelanjutan. Motor DC dibagi menjadi 2 jenis yaitu ; Motor DC dengan sikat (mekanis komutasi), yaitu motor yang memiliki sikat karbon berfungsi sebagai pengubah arus pada kumparan sedemikian rupa sehingga arah tenaga putaran motor akan selalu sama [7].

Motor DC tanpa sikat, menggunakan semi konduktor untuk merubah maupun membalik arus sehingga layaknya pulsa yang menggerakkan motor tersebut. Biasa digunakan pada sisrem *servo,* karena mempunyai efisiensi tinggi,

umur pemakaian lama, tingkat kebisingan suara listrik rendah, karena putarannya harus seperti stepper namun putarannya terus-menerus tanpa adanya *step.* Terdapat dua bagian mendasar pada Motor DC, yaitu [7]:

1. Bagian yang tetap/stasioner yang disebut stator. Stator menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
2. Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir. Berikut contoh motor dc dapat dilihat pada Gambar 2.7 .



Gambar 2.7 Motor DC

2.1.3.2 Motor Servo

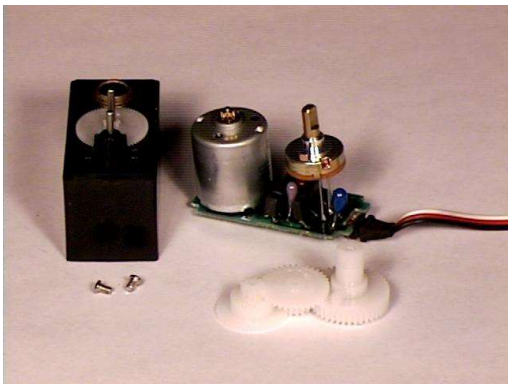
Motor servo adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feeback* yang terintegrasi dalam motor tersebut. Pada motor servo posisi putaran pada sumbu (*axis*) dari motor akan diinformasikan kembali ke

rangkaian kontrol yang ada di dalam motor servo [8]. Berikut adalah salah satu contoh merk servo ditunjukkan oleh Gambar 2.8



Gambar 2.8 Servo HITEC HS-5645MG

Motor servo disusun dari sebuah motor DC, *gearbox,* variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang ada pada pin kontrol motor servo [8].



Gambar 2.9 Kontruksi 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 Standar 180° Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°. Motor Servo Continuous Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu) [8].

2.1.4 Sensor Jarak (Ultrasonik)

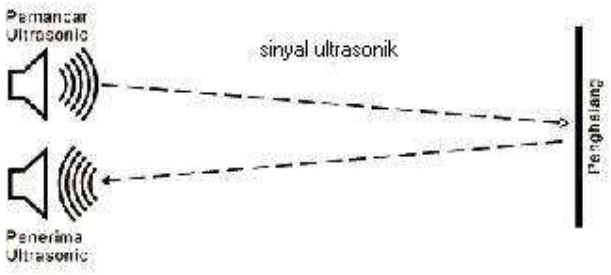
Dalam pengembangan sebuah robot, dibutuhkan perangkat keras sensor sebagai masukan untuk mengetahui lingkungan sekitar. Dalam lingkup robotika sensor memberikan kesamaan fungsi seperti indera pada manusia.

Sensor jarak digunakan untuk mendeteksi benda tanpa bersentuhan secara fisik, contoh dari sensor jarak adalah sensor ultrasonik. Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip dari pantulan gelombang suara dan digunakan untuk mendeteksi kebradaan suatu objek tertentu yang berada di depannya, frekuensi kerja nya berada pada daerah di atas gelombang suara yaitu dari 20KHz – 40KHz. Contoh sensor ultrasonik dapat dilihat pada Gambar 2.10.



Gambar 2.10 Sensor Ultrasonik

Pada dasarnya sensor ultrasonik terdiri dari dua bagian yaitu bagian pemancar dan penerima. Bagian pemancar mengirimkan gelombang ultrasonik dan bagian penerima menunggu pantulan (*echo*) dari gelombang ultrasonik yang telah dikirim sebelumnya. Prinsip kerja sensor ultrasonik dapat dilihat pada Gambar 2.11



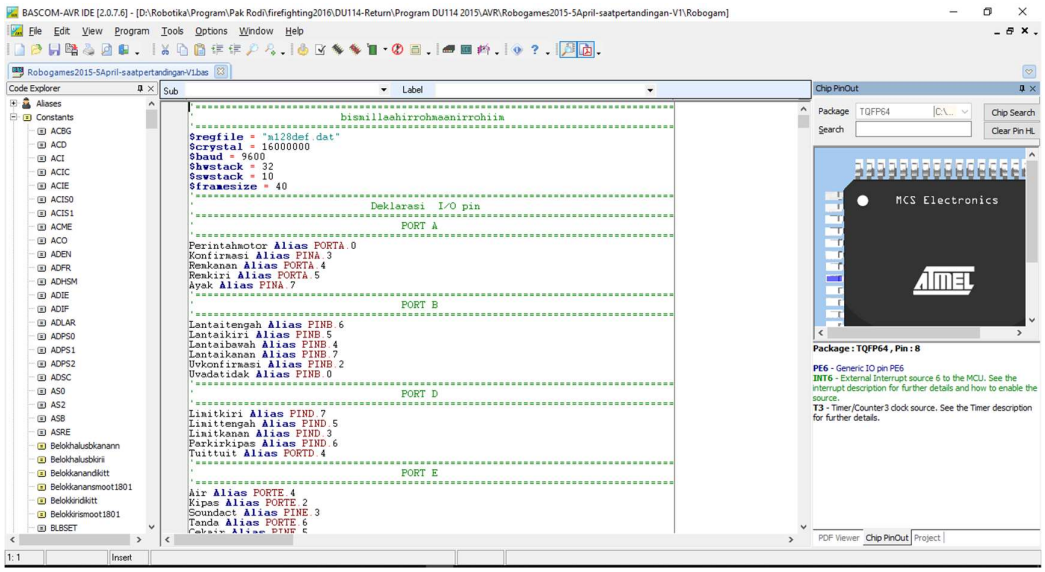
Gambar 2.11 Prinsip Kerja sensor Ultrasonik [9]

2.2 Perangkat Lunak

Perangkat lunak merupakan aplikasi pendukung yang digunakan untuk mengontrol seluruh perangkat keras yang telah di rancang dan diintegrasi. Perangkat lunak ini berupa algoritma gerak navigasi dalam bentuk *listing* program yang ditanamkan didalam *microcontroller*

2.2.1 BASCOM-AVR

BASCOM-AVR merupakan program *basic compiler* berbasis windows untuk *microcontroller* keluarga AVR. BASCOM-AVR merupakan pemrograman dengan bahasa tingkat tinggi, bahasa "*Basic*" yang dikembangkan dan dikeluarkan oleh MCS Electronics. Gambar 2.12 Menunjukan tampilan antarmuka Bascom AVR



Gambar 2.12 Tampilan Antarmuka Bascom AVR

Instruksi-instruksi yang digunakan pada program BASCOM-AVR adalah berbeda-beda pada setiap microcontroller keluarga AVR, namun untuk instruksi-instruksi dasar sama di setiap microcontroller keluarga AVR. Berikut ini merupakan instruksi-instruksi dasar program BASCOM-AVR. Contoh instruksi yang digunakan pada pemrograman BASCOM-AVR ditunjukan oleh Tabel 2.1

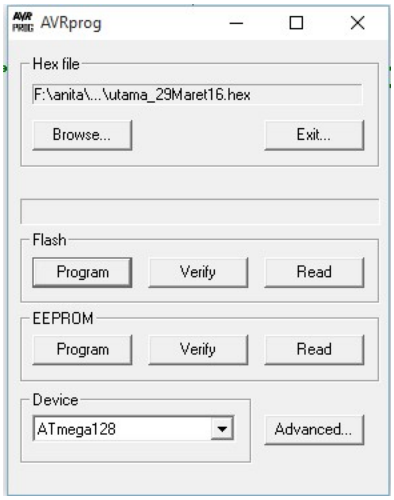
Tabel 2.1 Instruksi Dasar BASCOM-AVR [10]

Jenis Instruksi	Instruksi	Keterangan
Percabangan	IF...THEN	Mengeksekusi suatu kondisi atau percabangan berdasarkan evaluasi ekspresi boolean.
	IF...ELSE	
	IF...ELSE IF	
	SELECT...CASE	Mengeksekusi suatu *statement* dari beberapa *statement* tergantung nilai dari suatu ekspresi.

Perulangan	FOR...NEXT	Mengeksekusi *statement* dari beberapa blok *statement* berulang.
	DO...LOOP	Mengulang sebuah blok *statement* sampai kondisi terpenuhi.
	WHILE...WEND	Mengeksekusi *statement* berulang selama kondisi bernilai benar.
Lompatan	GOTO	Lompat ke label spesifik.
	CALL	Memanggil dan mengeksekusi sebuah *subroutine*.
	RETURN	Kembali dari sebuah *subroutine*.
	GOSUB	Bercabang ke dan mengeksekusi *subroutine*.
Penunda	DELAY	Menunda mengeksekusi program untuk waktu yang cepat (1000µs).
	WAIT	Menunda eksekusi program selama waktu yang diberikan dalam detik.
	WAITMS	Menunda eksekusi program selama waktu yang diberikan dalam satuan mili detik.
	WAITUS	Menunda eksekusi program selama waktu yang diberikan dalam satuan mikro detik.

2.2.2 AVRProg

AVRProg adalah perangkat lunak yang digunakan untuk memprogram mikrokontroler keluarga AVR. Perangkat lunak ini bekerja dengan cara mengunduh program yang telah ditulis dalam bentuk *.HEX ke dalam mikrokontroler. Berikut Gambar 2.13 menunjukan tampilan layar AVRProg.



Gambar 2.13 Tampilan Layar AVRProg

2.2.3 IDE Arduino

Untuk memulai program Arduino (untuk membuatnya melakukan apa yang kita inginkan) kita menggunakan IDE Arduino (Integrated Development Environtment), IDE Arduino adalah bagian *software opensource* yang digunakan untuk memprogram papan Arduino [11].



Gambar 2.14 Tampilan Antar Muka IDE Arduino

IDE Arduino menggunakan bahasa pemrograman tingakat tinggi sehingga memudahkan dalam pembuatan prorgram, bahasa pemrograman yang digunakan adalah bahasa C. IDE memungkinkan kita untuk menulis sebuah program secara *step by step* kemudian instruksi tersebut di upload ke papan Arduino. Gambar 2.14 menunjukan tampilan antar muka IDE Arduino.