

BAB II

LANDASAN TEORI

2.1 Definisi Robot

Keunggulan dalam teknologi robotik telah lama dijadikan ikon kebanggaan Negara-negara maju dunia. Kecanggihan teknologi yang dimiliki, gedung-gedung tinggi yang mencakar langit, tingkat kesejahteraan yang tinggi, kota-kota yang modern, belum terasa lengkap tanpa popularitas kepiawaian dalam dunia robotik

Istilah “robot” berasal dari kata “robota” (bahasa Czech) yang berarti pekerja, mulai populer ketika seorang penulis berbangsa Czech (Ceko), Karl Capek, membuat pertunjukan dari lakon komedi yang ditulisnya pada tahun 1921 yang berjudul RUR (Rossum’s Universal Robot). Namun menurut kamus besar Webster memberikan definisi mengenai robot, yaitu “sebuah peralatan otomatis yang melakukan pekerjaan seperti yang dilakukan oleh manusia”. Lain halnya dengan Institute of America, menurut Institute of America definisi robot adalah “manipulator dengan fungsi ganda dan dapat diprogram kembali, didesain untuk dapat memindahkan komponen, peralatan-peralatan khusus melalui pergerakan yang diprogram agar dapat melakukan berbagai kegiatan”.

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, luar angkasa, pertambangan, dan pencarian kebocoran gas.

2.1.1 Klasifikasi Robot

Menurut stabilitasnya robot dibagi menjadi 2, yaitu:

- a. *Fixed Robot*, robot yang memiliki ruang kerja (*spatial space*) yang terbatas, dimana bagian dasarnya dilekatkan pada sebuah benda tetap seperti panel atau meja.
- b. *Mobile Robot*, robot yang memiliki ruang kerja yang luas, dimana bagian dasarnya pada sebuah alat gerak seperti roda atau kaki. Beberapa macam *mobile robot* antara lain robot beroda, robot yang bergerak dengan menggunakan roda. Robot berkaki, robot yang bergerak menggunakan kaki dalam perpindahannya. Pembahasan lebih lanjut akan berfokus pada jenis *mobile robot* dengan alat gerak roda, khususnya tipe *ackerman drive*. Untuk tipe *fixed robot* dan robot berkaki tidak akan dibahas lebih lanjut (Lehrbaum, 2008).

2.1.2 Robot Beroda

Robot beroda dalam pergerakannya menggunakan roda atau ban, sehingga dapat berpindah dari titik satu ke titik yang lain. Jenis-jenis robot beroda adalah *differential drive*, *skid steering*, *synchro drive*, *omni wheel*, *tricycle steering*, *ackerman steering*, dan *articulated Drive*. Jenis robot beroda di atas diklasifikasikan menurut pengendalian roda robot.

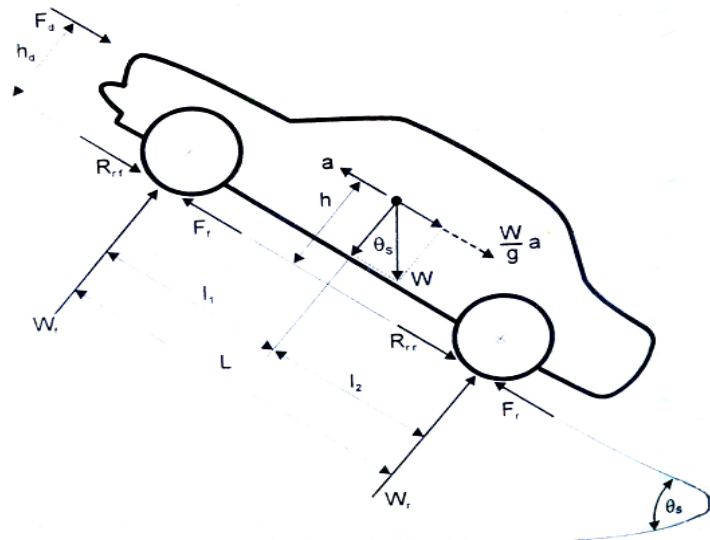


Gambar 2.1 Robot beroda (Lehrbaum,2008).

2.1.3 Kendaraan Bergerak Lurus (Sutantra, 2001)

2.1.3.1 Gaya Dorong dan Gaya Hambat pada Kendaraan

Kendaraan yang dapat digerakkan maju atau mundur perlu gaya dorong yang cukup dalam melawan hambatan yang dilalui. Gaya dorong pada suatu kendaraan terjadi pada roda penggerak kendaraan. Gaya dorong ditransformasikan dari torsi mesin kendaraan ke roda penggerak melalui sistem penggerak diantaranya kopling, transmisi, gigi differensial, dan poros penggerak. Gambar 2.2 merupakan diagram bodi bebas pada kendaraan bergerak maju dalam penggambaran gaya dorong pada kendaraan dengan dua buah poros penggerak. Hambatan angin dalam *rolling* berhubungan dengan gaya hambat.



Gambar 2.2 Diagram Bodi Bebas Kendaraan Bergerak Maju (Sutantra 2001).

Keterangan :

F_f = Gaya dorong pada roda depan.

F_r = Gaya dorong pada roda belakang.

R_{rr} = Gaya hambat *rolling* pada roda belakang.

R_{rf} = Gaya hambat *rolling* pada roda depan.

F_d = Gaya hambatan angin.

a = Percepatan kendaraan.

θ = Sudut tanjakan jalan

Gaya dorong pada roda yang ditransformasikan dari torsi mesin kendaraan dirumuskan sebagai

$$F = \frac{Me \cdot i_t \cdot i_d \cdot h_t}{r} \dots\dots\dots 2.1$$

Keterangan :

$F = F_f + F_r$ = Gaya dorong pada roda penggerak depan dan belakang. (Newton).

$F = F_f$ = Gaya dorong pada kendaraan dengan penggerak roda depan. (Newton).

$F = F_r$ = Gaya dorong pada kendaraan dengan penggerak roda belakang. (Newton).

M_e = Torsi keluaran dari mesin. (N.m)

r = Jari-jari roda. (m)

η_t = Efisiensi transmisi.

η_t = 0,88 – 0,92 pada mesin dengan letak memanjang.

η_t = 0,91 – 0,95 pada mesin dengan letak melintang.

i_t = Perbandingan gigi transmisi.

i_d = Perbandingan transmisi pada gardan

Gaya hambatan total pada kendaraan

$$F_R = F_d + R_r + R_g \dots\dots\dots 2.2$$

Keterangan :

$R_r = R_{rr} + R_n \equiv$ Total hambatan *rolling* (N).

$R_g = W \sin \theta \equiv$ Hambatan tanjakan (N).

Gaya dorong bersih dalam meningkatkan kecepatan kendaraan (F_n)

$$F_n = F - F_R \dots\dots\dots 2.3$$

Percepatan (a) yang dapat ditimbulkan melalui gaya dorong

$$a = \frac{F_n \cdot g}{k_m \cdot W} = \frac{(F - F_R)g}{k_m \cdot W} \dots\dots\dots 2.4$$

Kecepatan akhir kendaraan (V_t) setelah waktu t detik dihitung dengan menggunakan rumus

$$V_t = V_0 + \frac{(F - F_R)}{k_m \cdot W} g \cdot t \dots\dots\dots 2.5$$

2.1.3.2 Percepatan dan Perlambatan pada Kendaraan

Pada gerakan lurus, nilai percepatan dan perlambatan pada kendaraan berfungsi sebagai parameter kinerja laju kendaraan.

Pada kendaraan yang bergerak dari kondisi diam sampai kecepatan tertentu atau dari kecepatan tertentu sampai kendaraan berhenti, maka percepatan atau perlambatan (a), waktu (t) percepatan atau perlambatan, dan jarak (s) perlambatan atau percepatan dihitung dengan menggunakan rumus-rumus :

Kondisi Percepatan,

$$a = \frac{V_t^2}{2 \cdot s} = \frac{V_t}{t} = \frac{2 \cdot s}{t^2}$$

$$t = \frac{V_t}{a} = \frac{2 \cdot s}{V_t} = \sqrt{\frac{2 \cdot s}{a}} \dots\dots\dots 2.6$$

$$s = \frac{V^2}{2 \cdot a} = \frac{V_t \cdot t}{2} = \frac{a \cdot t^2}{2}$$

Keterangan :

V_t = Kecepatan akhir (meter/detik).

t = Waktu tempuh (detik).

s = Jarak tempuh (meter).

a = Percepatan (meter/detik²).

Kondisi Perlambatan,

$$a = \frac{V_0^2}{2 \cdot s} = \frac{V_0}{t} = \frac{2 \cdot s}{t^2}$$

$$t = \frac{V_0}{a} = \frac{2 \cdot s}{V_0} = \sqrt{\frac{2 \cdot s}{a}} \dots\dots\dots 2.7$$

$$s = \frac{V_0^2}{2 \cdot a} = \frac{V_0 \cdot t}{2} = \frac{a \cdot t^2}{2}$$

Keterangan :

V_0 = Kecepatan awal saat perlambatan (meter/detik).

t = Waktu perlambatan (detik).

s = Jarak perlambatan (meter).

a = Perlambatan (meter/detik²).

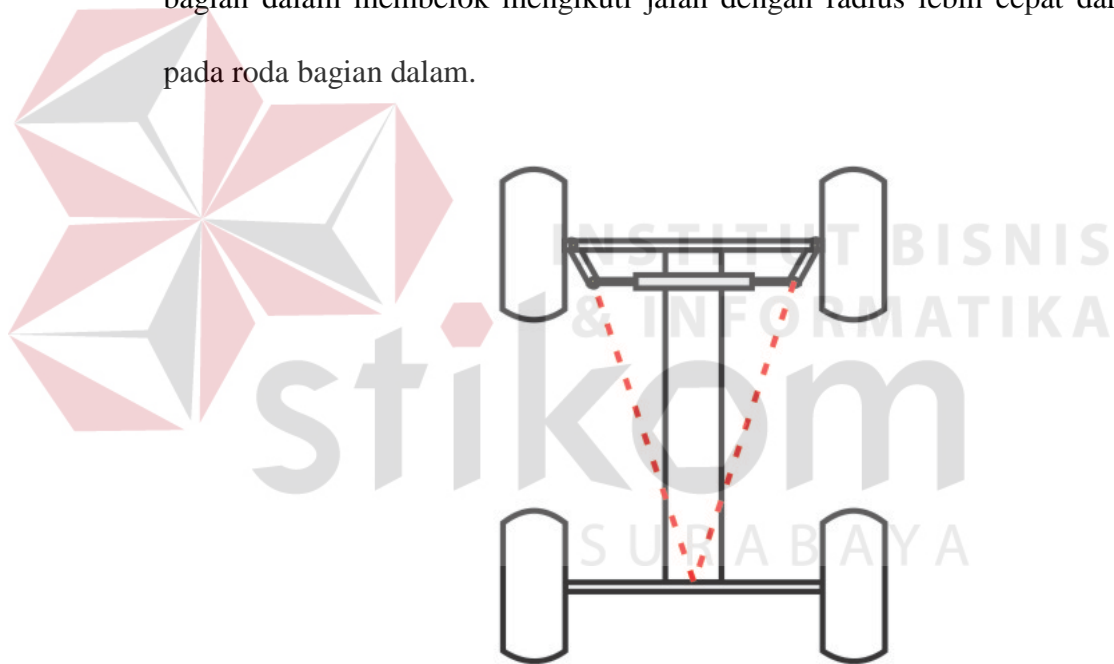
Dalam kondisi umum di mana kendaraan dapat dipercepat dari suatu kecepatan awal atau diperlambat sampai titik tertentu, percepatan atau perlambatan (a), waktu tempuh (t), dan jarak tempuh (s) dirumuskan sebagai.

$$a = \frac{V_t - V_0}{t} = \frac{V_t^2 - V_0^2}{2 \cdot s} = 2 \frac{s - V_0 \cdot t}{t^2} \dots\dots\dots 2.8$$

$$s = \frac{V_t^2 - V_0^2}{2 \cdot a} = \frac{V_t - V_0}{2} t = v_0 \cdot t + \frac{1}{2} a t^2 \dots\dots\dots 2.9$$

2.1.4 Pararell Steering

Pada lengan kemudi *Pararell steering* mempunyai sudut 90° , sumbu roda akan selalu sejajar satu sama lain kanan juga kiri dan sistem kendali ini biasanya banyak digunakan pada robot yang memiliki pola seperti roda mobil. Saat robot melaju pada bidang lurus maka tidak ada kelemahan pada sistem *Pararell steering*, serta sistem kendali ini sangat cocok digunakan pada robot pengiriman dari titik *object*. Namun seperti yang terlihat pada gambar 2.3 saat roda memutar melintasi belokan, roda bagian dalam membelok mengikuti jalan dengan radius lebih cepat dari pada roda bagian dalam.



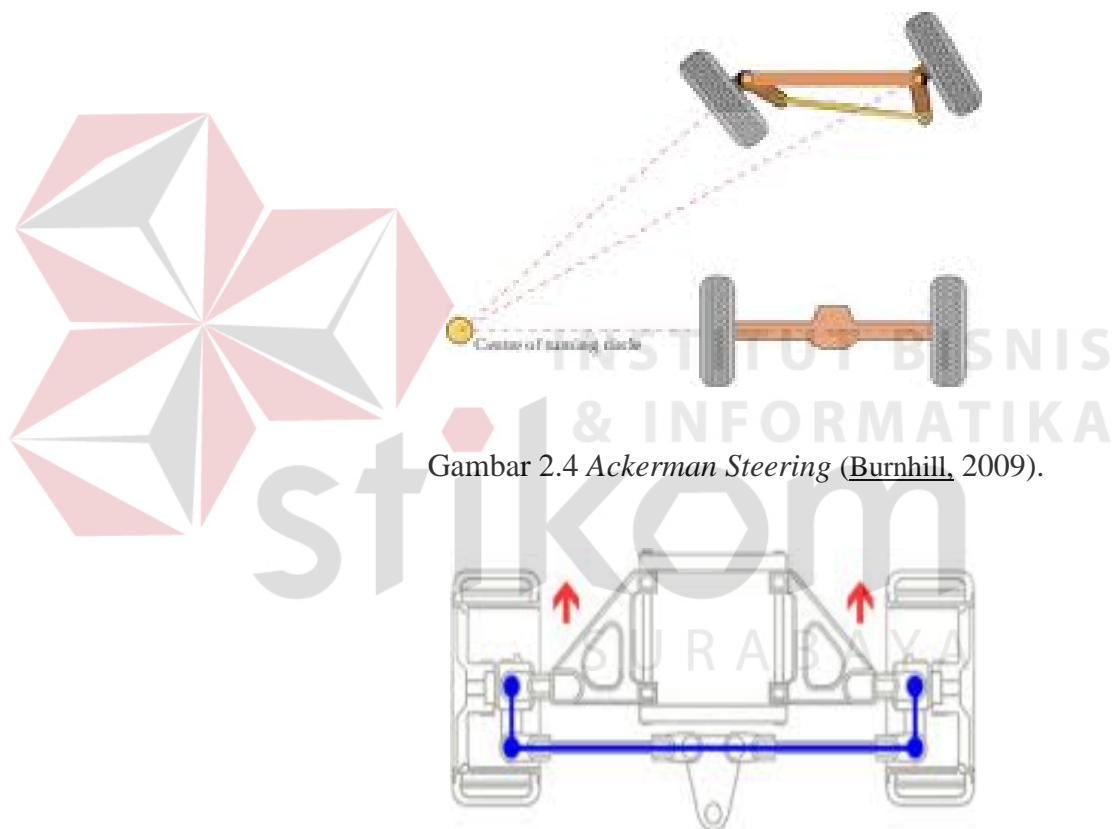
Gambar 2.3 Kemudi *Pararell steering* (makadir.sttnas.ac.id).

2.1.5 Ackerman Steering

Ackerman steering merupakan pengendalian arah gerak robot dengan menggerakkan sudut putar roda depan. Kinematika *ackerman* sangat mirip dengan mobil yang dikenal umum, sehingga dinamakan *car-like steering*. Kinematika *ackerman* pada dasarnya hampir sama dengan

kinematika *tricycle steering* dengan dua roda penentu arah di bagian depan. Penggunaan dua roda depan akan mempermudah pengendalian posisi.

Sebelum prinsip *ackerman steering* ini ditemukan, sebelumnya masih menggunakan prinsip *parallel steering*. Seperti yang ditunjukkan gambar 2.4 yang merupakan gambar *parallel steering*. *Parallel steering* lebih kaku dan sudut putar yang bisa diatur sangat terbatas.



Gambar 2.4 Ackerman Steering (Burnhill, 2009).

Gambar 2.5 Parallel steering (burnhill, 2009).

Ackermann steering merupakan proses pergerakan tuas antar roda kemudi, dengan menggunakan derajat. Sistem ini dirancang untuk memastikan gerak roda depan dalam diputar ke sudut yang sedikit lebih tajam dari luar roda saat membelok, sehingga menghilangkan *geomet*

rically disebabkan ban selip. Seperti yang terlihat pada gambar 2.6 sumbu dua roda depan berpotongan di satu titik yang sama dan terletak pada sumbu poros belakang. Dapat dirumuskan pada persamaan:

$$\cot \theta_i - \cot \theta_o = \frac{d}{l} \dots\dots\dots 2.10$$

Artinya:

θ_i = sudut kemudi relatif dari roda bagian dalam

θ_o = sudut kemudi relatif dari roda luar

l = pemisahan roda membujur

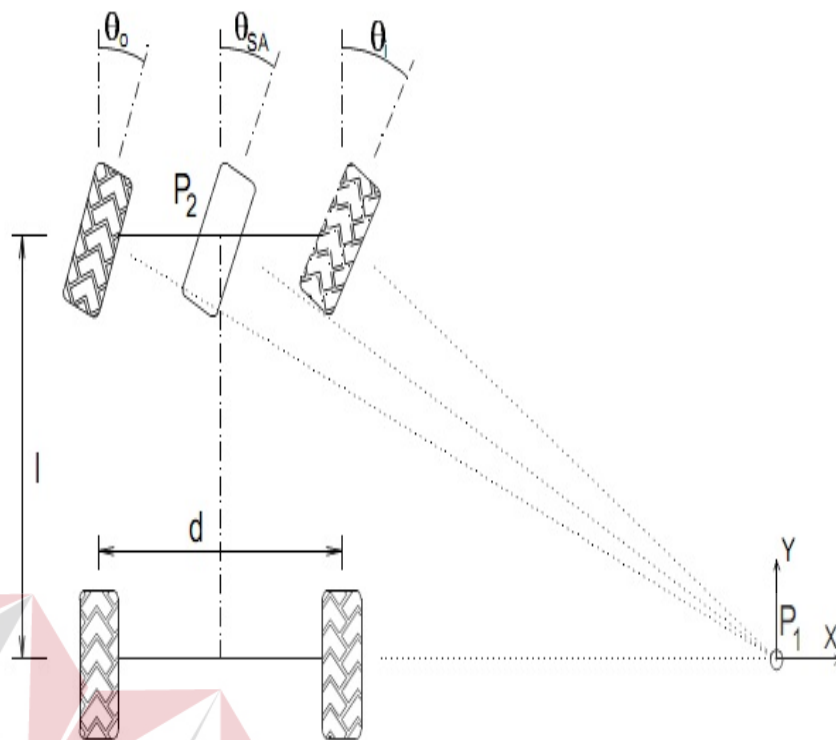
d = pemisahan roda lateral

Sudut kemudi kendaraan 2 dapat dianggap sebagai sudut (*relatif SA kendaraan*) dengan roda pusat imajiner yang terletak di titik acuan P sebagai ajuan ditunjukkan pada gambar 3.10, dinyatakan dalam sudut *steering* atau di luar *steering SA*

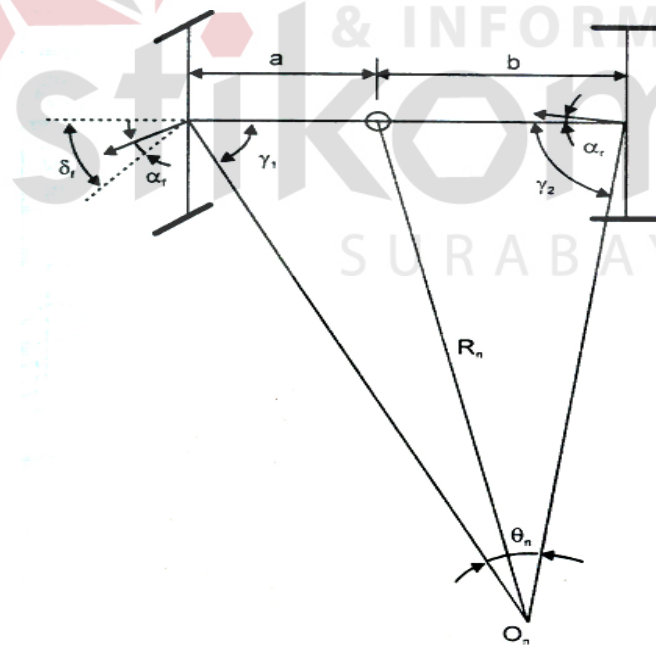
$$\cot \theta_{SA} = \frac{d}{2l} + \cot \theta_i \dots\dots\dots 2.11$$

Atau ,

$$\cot \theta_{SA} = \cot \theta_o - \frac{d}{2l} \dots\dots\dots 2.12$$



Gambar 2.6 kemudi robot belok ke kanan (makadir.sttnas.ac.id).



Gambar 2.7 Kondisi Nyata Kendaraan Berbelok (Sutantra, 2001).

Keterangan :

α_f, α_r = Sudut slip rata-rata dari roda depan dan roda belakang.

q_n = Sudut belok nyata.

O_n = Pusat belokan nyata.

Gerakan kendaraan ketika belok secara sederhana seperti pada Gambar 3.11, di mana secara geometri dirumuskan sebagai

$$\theta_n = 180^\circ - \gamma_1 - \gamma_2 \dots\dots\dots 2.13$$

$$\gamma_1 = 180^\circ - 90^\circ - (\delta_f - \alpha_f) \dots\dots\dots 2.14$$

$$\gamma_1 = 90^\circ - \delta_f + \alpha_f$$

$$\gamma_2 = 90^\circ - \alpha_r \dots\dots\dots 2.15$$

Bila persamaan (b) dan (c) dimasukkan ke persamaan (a),

$$\theta_n = \delta_f - \alpha_f + \alpha_r \dots\dots\dots$$

2.16

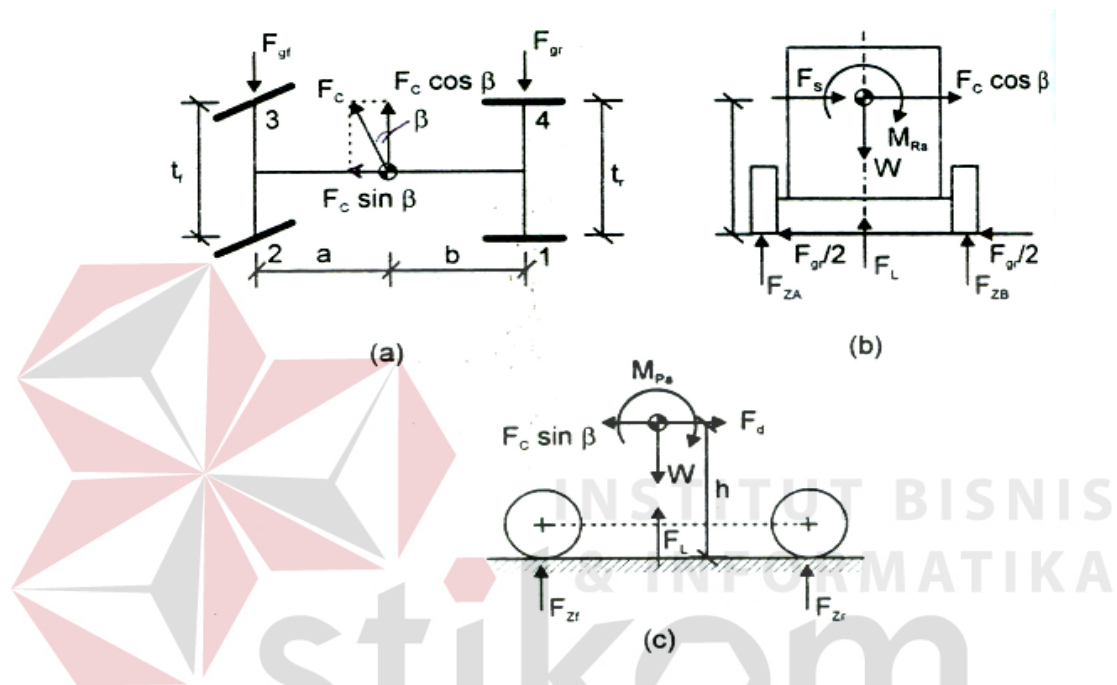
Radius belok nyata kendaraan dirumuskan sebagai,

$$R_n = \frac{a+b}{\theta_n^\circ} 57,29 = \frac{a+b}{\delta_f^\circ - \alpha_f^\circ + \alpha_r^\circ} 57,29 \dots\dots\dots 2.17$$

Bila nilai $\alpha_f = \alpha_r$, maka radius belok nyata (R_n) sama dengan radius belok ideal (R_i). Kondisi $\alpha_f = \alpha_r$ disebut kondisi netral. Bila kondisi $\alpha_f > \alpha_r$, maka $R_n > R_i$, kondisi ini disebut kondisi *understeer* sehingga kendaraan sulit dibelokkan.

2.1.5.1 Dinamika Kendaraan Berbelok

Pada gambar 2.8 merupakan diagram bodi bebas kendaraan ketika berbelok dipandang dari atas (a), dipandang dari belakang (b), dan dipandang dari samping (c).



Gambar 2.8 Gaya dan Momen pada Kendaraan Belok (Sutantra, 2001).

Keterangan :

F_c \equiv Gaya sentrifugal kendaraan.

F_{gf}, F_{gr} \equiv Gaya gesekan pada roda depan dan belakang.

F_s \equiv Gaya angin dari samping kendaraan.

M_{Ra} \equiv Momen *rolling* akibat angin.

F_z \equiv Gaya normal pada roda.

M_{pa} \equiv Momen angguk (*pitching*) akibat angin.

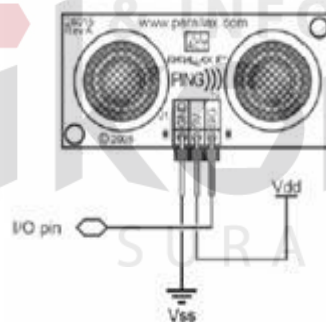
h \equiv Tinggi titik pusat massa kendaraan.

$$F_L \quad \equiv \quad \text{Gaya angkat angin.}$$

2.2 Sensor *Ultrasonic Distance*

Sensor *Ultrasonic Distance* adalah sensor yang dapat mendeteksi jarak obyek dengan cara memancarkan gelombang ultrasonik dengan frekuensi 40 KHz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas. Kelebihan sensor ini adalah hanya membutuhkan 1 sinyal (SIG) selain jalur 5V dan ground.

Untuk penjelasan atau prinsip aksesnya sama kok ma srf 04, hanya saja untuk sensor *Ultrasonic Distance* hanya memakai 3 pin, pin trigger sama echo digunakan dalam 1 pin, sehingga dengan menggunakan sensor *Ultrasonic Distance* kita dapat menghemat penggunaan I/O mikrokontroler. Konfigurasi pin sensor *Ultrasonic Distance* sbagai berikut:

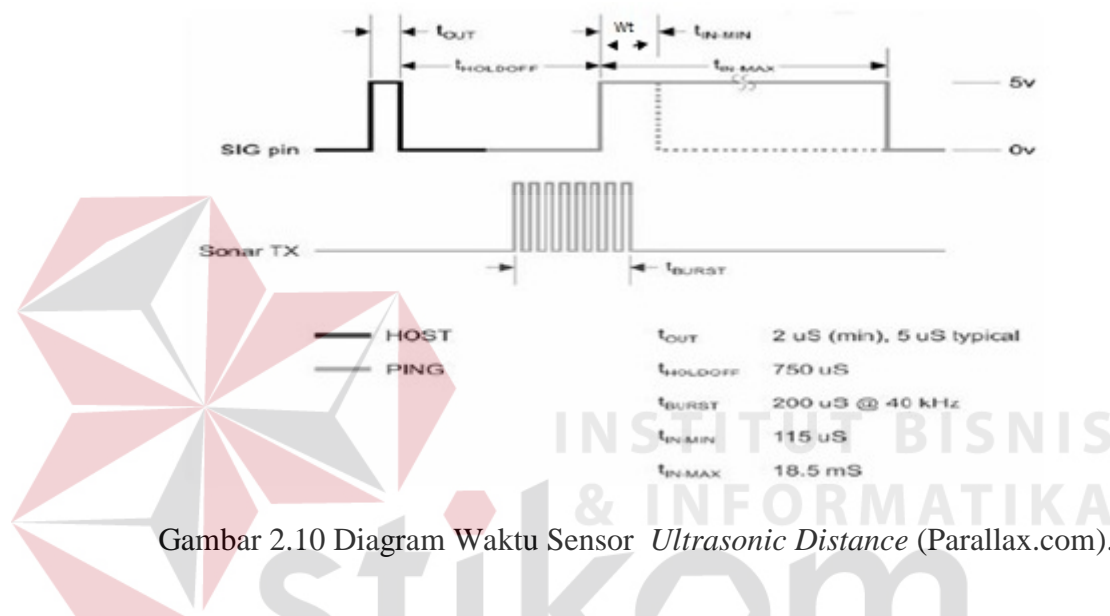


Gambar 2.9 Timing Sensor *Ultrasonic Distance* (Parallax.com).

Sensor PING mendeteksi jarak objek dengan cara memancarkan gelombang ultrasonik (40 KHz) selama $t = 200 \mu s$ kemudian mendeteksi pantulannya. Sensor *Ultrasonic Distance* memancarkan gelombang ultrasonik sesuai dengan kontrol dari *microcontroller* pengendali (pulsa *trigger* dengan tout min $2 \mu s$).

Spesifikasi sensor *Ultrasonic Distance* :

1. Kisaran pengukuran 3 cm – 3 m.
2. Input trigger – positive TTL pulse, 2 μ s min, 5 μ s tipikal.
3. Echo hold off 750 us dari of trigger pulse.
4. Delay before next measurement 200 μ s.
5. Burst indikator LED menampilkan aktivitas sensor.



Gambar 2.10 Diagram Waktu Sensor *Ultrasonic Distance* (Parallax.com).

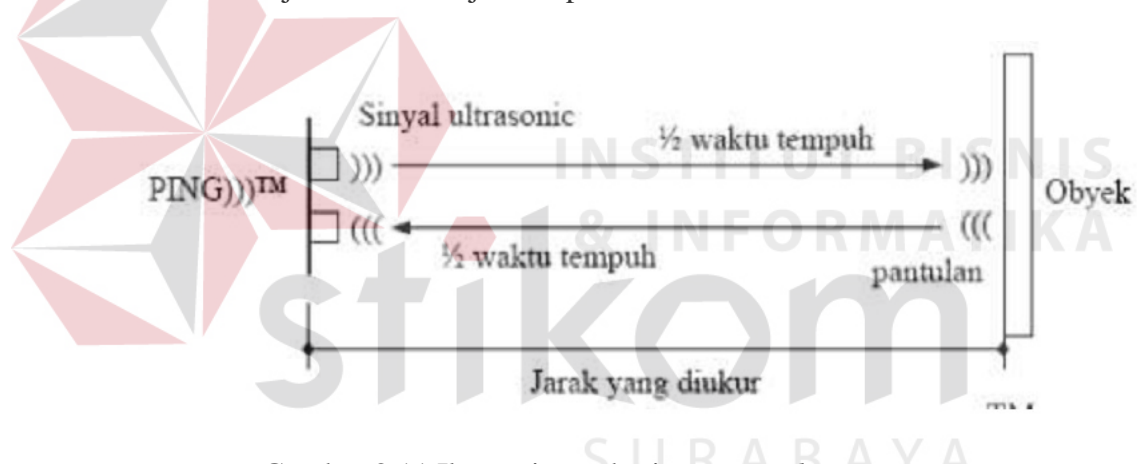
Keluaran dari sensor *Ultrasonic Distance* berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 μ S sampai 18,5 ms. Pada dasarnya sensor *Ultrasonic Distance* terdiri dari sebuah chip pembangkit sinyal 40 KHz, sebuah *speaker ultrasonik* dan sebuah mikropon *ultrasonik*. *Speaker ultrasonik* mengubah sinyal 40 KHz menjadi suara sementara mikropon *ultrasonik* berfungsi untuk mendeteksi pantulan suaranya.

Pin signal dapat langsung dihubungkan dengan *microcontroller* tanpa tambahan komponen apapun. sensor *Ultrasonic Distance* hanya akan mengirimkan suara *ultrasonik* ketika ada pulsa *trigger* dari *microcontroller* (Pulsa high selama 5 μ S). Suara *ultrasonik* dengan *frekuensi* sebesar 40 KHz akan

dipancarkan selama 200 μ S. Suara ini akan merambat di udara dengan kecepatan 344.424 m/detik (atau 1cm setiap 29.034 μ S), mengenai objek untuk kemudian terpantul kembali ke sensor. Selama menunggu pantulan, sensor akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (*low*) ketika suara pantulan terdeteksi oleh sensor. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara Ping dengan objek.

Agar sensor ini dapat digunakan untuk mengukur jarak dibutuhkan sebuah mikrokontroler untuk mengukur waktu tempuh sinyal ultrasonik dari sensor, memantul pada benda yang akan diukur, dan diterima kembali oleh sensor.

Ilustrasi cara kerja sensor ditunjukkan pada Gambar 2.11.



Gambar 2.11 Ilustrasi cara kerja sensor *Ultrasonic Distance* (Parallax.com).

Besarnya jarak yang diukur dapat dihitung dengan menggunakan persamaan :

$$S = v.t / 2$$

s = dengan :

s = jarak yang diukur (meter).

v = kecepatan suara (344 m/detik).

t = waktu tempuh (detik).

2.3 Obstacle Avoidance

Pada bagian *Obstacle Avoidance* menjelaskan tentang metode penghindar rintangan yang relevan yaitu deteksi ujung tepi object, tahap stabil dari obstacle avoidance, metode yang support terhadap relevansi obstacle avoidance.

2.3.1 Analisa Ujung Tepi Object

Salah satu kendala Metode *obstacle avoidance* yang sangat populer didasarkan pada analisa ujung tepi deteksi *obstacle*. Dalam metode ini, algoritma mencoba untuk menentukan posisi dari sisi kendala vertikal dan kemudian mengarahkan robot pada salah satu sisi ujung tepi *obstacle*. Garis yang menghubungkan dua sisi dalam dan luar dianggap mewakili salah satu dari batas-batas penghalang. Metode ini digunakan dalam penelitian, serta beberapa tulisan lainnya semua menggunakan sensor jarak meliputi sensor shap juga sensor ultrasonik untuk mendeteksi *obstacle*. Kelemahan dengan implementasi saat ini dari metode ini adalah bahwa robot berhenti di depan hambatan untuk mengumpulkan informasi sensor. Namun, ini bukan Keterbatasan yang melekat ujung tepi dari *obstacle* dengan metode pendeteksian, dimungkinkan untuk mengatasi masalah ini dengan komputer yang lebih cepat dalam implementasi mendatang. Dalam pendekatan lain tepi-deteksi (menggunakan sensor seperti shap dan ultrasonik), robot tetap diam saat mengambil scan panorama lingkungannya.

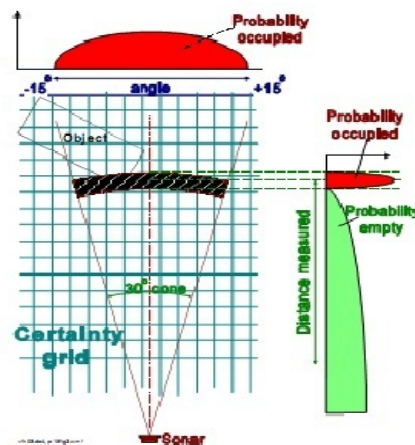
Setelah aplikasi tertentu *line-fitting algorithms*, perencana ujung tepi *obstacle* berbasis global dilembagakan untuk merencanakan robot bisa

berjalan. Posisi terendah untuk membatasi benda dengan sensor pada posisi spasial tepi *obstacle* 10-50 cm, tergantung pada jarak kendala dan sudut antara permukaan benda.

2.3.2 Kesetabilan dari kendala Obstacle

Metode untuk representasi probabilistik hambatan dalam dunia model kotak-jenis telah dikembangkan di Carnegie-Mellon University (CMU). Model nyata, yang disebut jaringan kepastian, sangat cocok untuk akomodasi data sensor akurat seperti rentang pengukuran dari sensor ultrasonik.

Dalam area kerja robot diwakili oleh array dua dimensi persegi elemen, dilambangkan sebagai sel. Setiap sel berisi nilai kepastian yang menunjukkan ukuran juga dimensi bahwa kendala ada dalam wilayah sel. Dengan metode CMU, memperhitungkan kesetabilan nilai yang dipengaruhi oleh fungsi probabilitas yang memperhitungkan karakteristik yang diberikan sensor. Karakteristik sensor ultrasonik ini misalnya, memiliki bidang kerucut pandang. Sebuah sensor ultrasonic mengembalikan ukuran radial dari jarak ke *obstacle* terdekat dalam kerucut, namun tidak menentukan lokasi sudut *obstacle*. (Gambar menunjukkan area A di mana sebuah *obstacle* harus berada dalam range untuk menghasilkan pengukuran jarak d). Jika *obstacle* terdeteksi oleh sensor ultrasonik, maka *obstacle* lebih dekat ke sumbu titik tengah sensor dari pada pinggiran bidang kerucut. Untuk alasan ini, CMU fungsi probabilistik meningkatkan kestabilan pada sel yang lebih dekat dengan sumbu titik tengah.



Gambar 2.12 sensor mengenali benda (Borenstein, 1990).

Dua-dimensi proyeksi kerucut bidang pandang sebuah sensor ultrasonik. Berbagai membaca d menunjukkan adanya *obstacle* di suatu tempat dalam daerah yang diarsir A (metode Carnegie Mellon).

2.4 Serial Peripheral Interface

Mikrokontroler dapat disebut sebagai single pada chip komputer yang meliputi jumlah peripheral seperti RAM, EEPROM, Timer, yang dibutuhkan untuk melakukan beberapa tugas yang telah ditetapkan.

Ada beberapa macam mikrokontroler yang digunakan dalam aplikasi yang berbeda sesuai kemampuan dan fungsi untuk melakukan tugas yang diinginkan, yang paling sering kali dipergunakan adalah 8.051, AVR dan PIC *microcontoler*. *Microcontoler* AVR tersedia dalam tiga kategori:

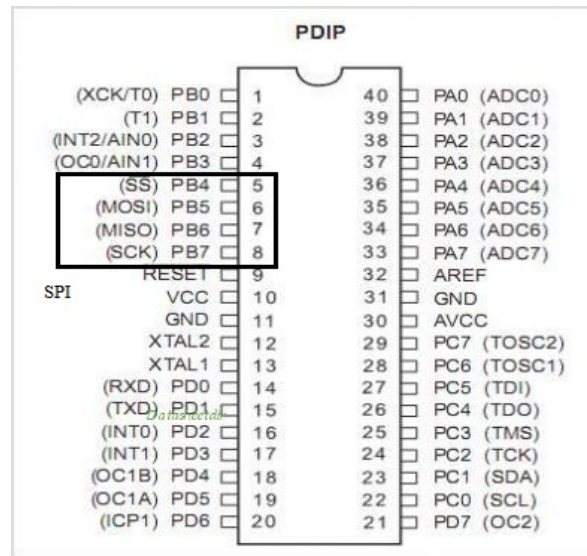
1. TinyAVR – memori kecil, ukuran kecil, hanya cocok untuk aplikasi sederhana.
2. Mega AVR adalah Avr yang biasanya digunakan karena memiliki memori (upto 256 KB), kapasitas yang lebih tinggi dari *peripheral inbuilt* dan cocok untuk moderat untuk aplikasi yang kompleks .

3. XmegaAVR- Digunakan secara komersial untuk aplikasi yang kompleks, yang membutuhkan memori program yang besar dan kecepatan tinggi. Tabel berikut membandingkan AVR seri yang disebutkan di atas dari mikrokontroler.

Series Name	Pins	Flash Memory	Special Feature
TinyAVR	6-32	0.5-8 KB	Small in size
MegaAVR	28-100	4-256KB	Extended peripherals
XmegaAVR	44-100	16-384KB	DMA , Event System included

Gambar2.13 Klasifikasi Atmega (Atmel 2011).

Minimum sistem merupakan suatu perangkat sistem yang dapat digunakan untuk belajar *microcontoler*. Ada berbagai jenis *minimum sistem* yaitu Minsis ATMega 8, Minsis Atmega 16 dan ATMega 32. *Minimum sistem* yang saya buat ini tidak dilengkapi dengan catu daya, namun hanya sistem minimum biasa. IC ATMega yang kompatibel dengan sistem minimum ini hanya IC tipe ATMega 8 saja. Dalam ATMega 8 terdapat Port B, Port C, dan Port D. Ada juga pin MISO, MOSI, SCK beserta RESET,VCC, dan GND yang dapat langsung dihubungkan ke downloader atau USB ASP.



Gambar2.14 Konfigurasi Pin SPI Atmega 32 (Atmel 2011).

Serial Peripheral Interface (SPI) adalah komunikasi microcontroller dengan device lain diluar microcontroller atau komunikasi antara microcontroller dengan microcontroller lain. Komunikasi ini merupakan salah satu metode komunikasi serial secara Synchronous yang memiliki kecepatan tinggi. Komunikasi ini membutuhkan 3 jalur yaitu MOSI, MISO, SCK.



Gambar 2.15 Konfigurasi Pin *Serial Peripheral Interface (SPI)* (Atmel 2011).

2.5 Motor DC

Motor *Brushless* adalah motor yang mempunyai kekuatan yang konsisten dan kinerja dari *run* untuk menjalankan sekaligus mudah dalam pemeliharaan, serta terdapat sistem pengontrol kecepatan listrik. Dalam motor *brushless* terdapat tiga komponen diantaranya adalah.

1. Kutub medan secara sederhana digambarkan interaksi dua kutub magnet akan menyebabkan putaran pada motor dc. Motor memiliki kutub medan yang stasioner dan dinamo yang menggerakkan bearing pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.

2. Dinamo, bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.

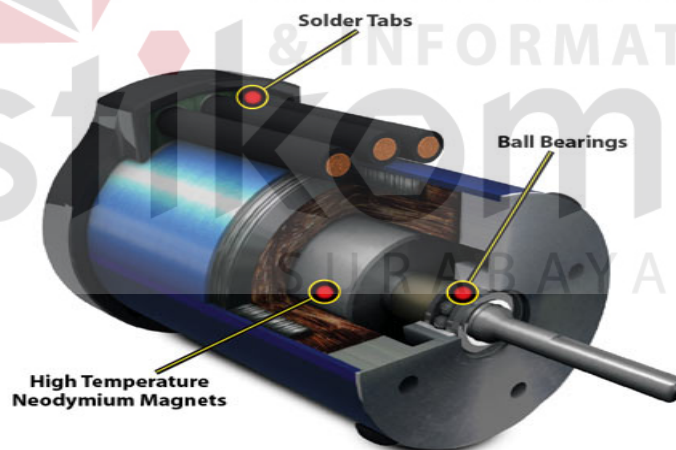
3. *Commutator* Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur tegangan dinamo meningkatkan tegangan dinamo akan meningkatkan kecepatan Arus medan menurunkan arus medan akan meningkatkan kecepatan. Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya

dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan *rolling mills*, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya. Motor DC juga relatif mahal dibanding motor AC. Antara kecepatan, *flux* medan dan tegangan dinamo ditunjukkan dalam persamaan berikut.

Gaya elektromagnetik $E = K\Phi N$ Torque: $T = K\Phi I_a$ dimana

E = gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt) Φ = *flux* medan yang berbanding lurus dengan arus medan N = kecepatan dalam RPM (putaran per menit) T = *torque elektromagnetik* I_a = arus dinamo K = konstanta persamaan.

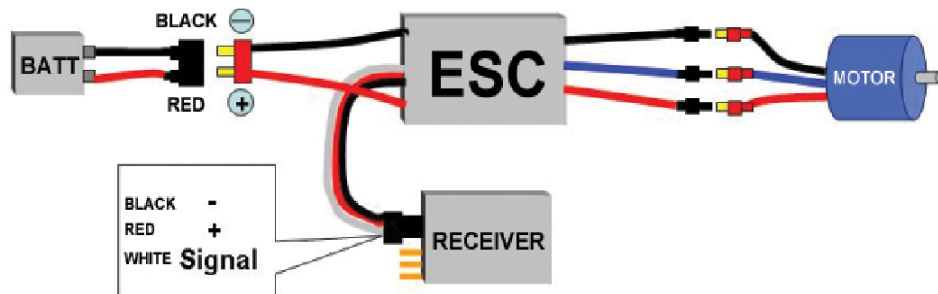


Gambar 2.16 motor *Brushless* (www.hpiracing.com/ Brushless).

2.6 *Electronic Speed Control (ESC)*

Motor brushless memiliki sebuah *Electronic Speed Control (ESC)* yang berfungsi sebagai pengatur kecepatan motor, selain itu juga berfungsi

untuk menaikkan jumlah arus yang diperlukan oleh motor. Kecepatan untuk motor yang keluar dari ESC diatur melalui pulsa dari mikrokontroler.



Gambar 2.17 *Electronic Speed Control (ESC)* (Hobbyking.com/esc).

2.7 Motor Servo

Motor servo adalah sebuah motor dengan sistem closed feedback di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar dengan pulsa 1.5 ms pada periode selebar 2 ms maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam.

Motor servo biasanya hanya bergerak mencapai sudut tertentu saja dan tidak kontinyu seperti motor DC maupun motor *stepper*. Walau demikian, untuk beberapa keperluan tertentu, motor servo dapat dimodifikasi agar bergerak kontinyu. Pada robot, motor ini sering digunakan untuk bagian kaki, lengan atau

bagianbagian lain yang mempunyai gerakan terbatas dan membutuhkan torsi cukup besar.

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya.

Motor Servo merupakan sebuah motor DC yang memiliki rangkaian control elektronik dan internal gear untuk mengendalikan pergerakan dan sudut angularnya. Motor servo adalah motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena internal gearnya.

Lebih dalam dapat digambarkan bahwa sebuah motor servo memiliki :

- a. jalur kabel : power, ground, dan control.
- b. Sinyal control mengendalikan posisi.
- c. Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Konstruksi didalamnya meliputi internal gear, potensiometer, dan feedback control (Ratih Kusuma Dewi, 2011).

2.7.1 Jenis-jenis Motor Servo

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).

2.8 Sistem Kendali Mobile Robot

2.8.1 Proportional Integral Derivative (PID) controller

Proportional Integral Derivative (PID) controller merupakan pengendali untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengendali PID merupakan gabungan dari tiga sistem kendali yang bertujuan untuk mendapatkan keluaran dengan *rise time* yang tinggi dan kecil. Seperti yang kita ketahui bahwa sistem kendali *proporsional* memiliki keunggulan yaitu *rise time* yang cepat tetapi sangat rentan dengan *overshot/undershot*, sistem kendali *integral* memiliki keunggulan untuk meredam galat, sedangkan sistem kendali *diverensial* memiliki keunggulan untuk memperkecil *delta error* atau meredam *overshot/undershot*. PID berdasarkan implementasinya dibedakan menjadi analog dan digital, PID analog diimplementasikan dengan komponen elektronika *resistor*, *capacitor*, dan *operational amplifier*, sedangkan PID digital diimplementasikan secara program (Embedded Robotic, 2006).

a. Pengendali Proporsional (P)

Pengaruh pada sistem :

1. Menambah atau mengurangi kestabilan dengan menambah atau mengurangi nilai konstanta K_P (Kontrol Proporsional).

2. Dapat memperbaiki respon transien khususnya : *rise time*, *settling time*.
3. Mengurangi *error steady state* (ESS). Untuk mengurangi ESS, dibutuhkan K_p besar, yang akan membuat sistem lebih tidak stabil. Kendali proporsional memberi pengaruh langsung (sebanding) pada *error*. Semakin besar *error*, semakin besar sinyal kendali yang dihasilkan pengendali.

b. Pengendali Integral (I)

Pengaruh pada sistem :

1. Mengurangi *error steady state* (ESS).
2. Respon lebih lambat (dibandingkan dengan P).
3. Dapat menambah ketidakstabilan (karena menambah orde pada sistem).

Perubahan sinyal kontrol sebanding dengan perubahan *error*. Semakin besar *error*, semakin cepat sinyal kontrol bertambah atau berubah.

c. Pengendali Derivatif (D)

Pengaruh pada sistem :

1. Memberikan efek redaman pada sistem yang berosilasi sehingga bisa memperbesar pemberian nilai K_p .
2. Memperbaiki respon transien, karena memberikan aksi saat ada perubahan *error*.
3. hanya berubah saat ada perubahan *error*, sehingga saat ada *error* statis D tidak beraksi. Sehingga D tidak boleh digunakan sendiri.

Besarnya sinyal kontrol sebanding dengan perubahan *error* (e). Semakin cepat *error* berubah, semakin besar aksi kontrol yang ditimbulkan (Wicaksono, 2004).

Sistem pengendali PID pada robot *obstacle Avoidance* robot menggunakan parameter jarak sebagai inputan dari *sensor ultrasonik distance*. Input jarak akan diproses untuk memberi respon kecepatan ke motor. Pada jarak yang jauh terhadap halangan *Sensor ultrasonik distance* akan mengirimkan data dalam *microcontroller* yang akan diolah ke PID untuk merespon motor melaju kencang. Pada titik halangan dekat respon motor akan langsung diturunkan sedikit hingga mendekati *set poin* yang telah ditentukan pada *obstacle*.

Dalam waktu kontinyu, sinyal keluaran pengendali PID dapat dirumuskan sebagai berikut.

$$u(t) = K_p + e(t) + K_i \int_0^1 e(t) dt + K_d \frac{de(t)}{dt} \dots\dots\dots 2.18$$

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proporsional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant.

1. Kontrol Proporsional

Kontrol P jika $G(s) = k_p$, dengan k adalah konstanta. Jika $u = G(s) \cdot e$ maka $u = K_p \cdot e$ dengan K_p adalah Konstanta Proporsional. K_p 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. Kontrol Integratif

Jika $G(s)$ adalah kontrol I maka u dapat dinyatakan sebagai $u(t) = [\int e(t)dt]$ K_i dengan K_i adalah konstanta Integral, dan dari persamaan diatas, $G(s)$ dapat dinyatakan sebagai $u = K_i \cdot e / s$

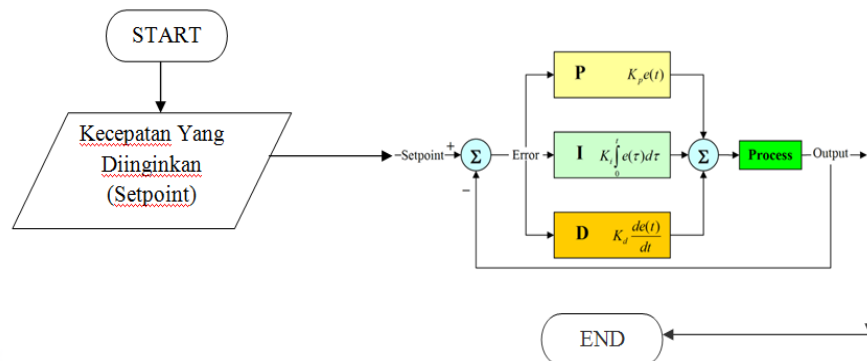
Jika $e(t)$ mendekati konstan (bukan nol) maka $u(t)$ akan menjadi sangat besar sehingga diharapkan dapat memperbaiki error. Jika $e(t)$ mendekati nol maka efek kontrol I ini semakin kecil. Kontrol I dapat memperbaiki sekaligus menghilangkan respon steady-state, namun pemilihan K_i yang tidak tepat dapat menyebabkan respon transien yang tinggi sehingga dapat menyebabkan ketidakstabilan sistem. Pemilihan K_i yang sangat tinggi justru dapat menyebabkan *output* berosilasi karena menambah *orde system*.

3. Kontrol Derivatif

Sinyal kontrol u yang dihasilkan oleh kontrol D dapat dinyatakan sebagai $G(s) = s \cdot K_d$ 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. Gambar blok diagram kontroler PID.

Diagram Blok pengendali PID dapat dilihat pada diagram berikut



Gambar 2.18 Diagram Blok pengendali PID (Fahmi 2011).

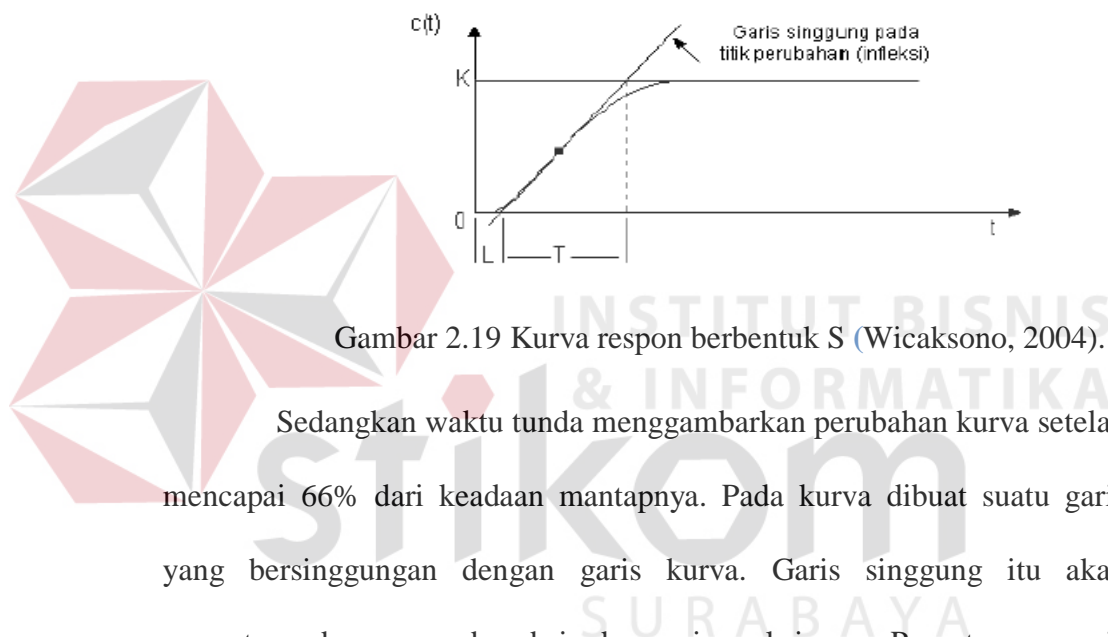
2.8.2 PID

Aspek yang sangat penting dalam desain kendali PID ialah penentuan parameter kendali PID supaya sistem kalang tertutup memenuhi kriteria performansi yang diinginkan (Wicaksono, 2004). Adapun metode *tuning* kendali PID yang sudah banyak dan sering digunakan adalah Ziegler-Nichols dan Cohen-Coon.

a. Metode Ziegler-Nichols

Ziegler-Nichols pertama kali memperkenalkan metodenya pada tahun 1942. Metode ini memiliki dua cara yaitu metode osilasi dan kurva reaksi. Kedua metode ditujukan untuk menghasilkan respon sistem dengan lonjakan maksimum sebesar 25%. Metode kurva reaksi didasarkan terhadap reaksi sistem kalang terbuka. Plant sebagai kalang terbuka dikenai sinyal *step function*. Kalau plant minimal tidak mengandung unsur

integrator ataupun *pole-pole* kompleks, reaksi sistem akan berbentuk S. Gambar 1 menunjukkan kurva berbentuk S tersebut. Kelemahan metode ini terletak pada ketidakmampuannya untuk menangani plant *integrator* maupun plant yang memiliki *pole* kompleks. Kurva berbentuk S mempunyai dua konstanta, waktu mati (*dead time*) L dan waktu tunda T . Dari Gambar, terlihat bahwa kurva reaksi berubah naik setelah selang waktu L .



Gambar 2.19 Kurva respon berbentuk S (Wicaksono, 2004).

Sedangkan waktu tunda menggambarkan perubahan kurva setelah mencapai 66% dari keadaan mantapnya. Pada kurva dibuat suatu garis yang bersinggungan dengan garis kurva. Garis singgung itu akan memotong dengan sumbu absis dan garis maksimum. Perpotongan garis singgung dengan sumbu absis merupakan ukuran waktu mati, dan perpotongan dengan garis maksimum merupakan waktu tunda yang diukur dari titik waktu L . Tabel 2.1 merupakan rumusan penalaan parameter PID berdasarkan cara kurva reaksi.

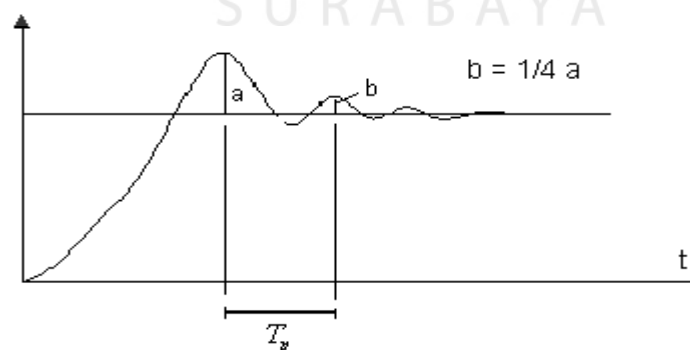
Tabel 2.1 Penalaan parameter PID dengan metode kurva reaksi

Type Kendali	P	I	D
I	$/L$		
D	$,9 T/L$	$/0,3$	
ID	$,2 T/L$	L	$,5L$

(Sumber : Wicaksono, 2004)

b. Metode Cohen-Coon

Karena tidak semua proses dapat mentolerir keadaan osilasi dengan amplitudo tetap, Cohen-Coon berupaya memperbaiki metode osilasi dengan menggunakan metode quarter amplitude decay. Respon loop tertutup sistem, pada metode ini, dibuat sehingga respon berbentuk *quarter amplitude decay*. *Quarter amplitude decay* didefinisikan sebagai respon transien yang amplitudonya dalam periode pertama memiliki perbandingan sebesar seperempat ($1/4$), untuk lebih jelasnya dapat dilihat pada Gambar 2.20.



Gambar 2.20 Kurva respon *quarter amplitude decay* (Wicaksono, 2004).

Pada kendali proporsional K_p ditala hingga diperoleh tanggapan *quarter amplitude decay*, periode pada saat tanggapan ini disebut T_p dan

parameter T_i dan T_d dihitung dari hubungan K_P dengan T_P . Sedangkan penalaan parameter kendali PID adalah sama dengan yang digunakan pada metode Ziegler-Nichols. Selain cara tersebut, metode Cohen-Coon ini bisa dihitung dengan aturan praktis yang parameter-parameter plantnya diambil dari kurva reaksi yang terdapat pada tabel 2.2.

Tabel 2.2 Penalaan parameter PID dengan metode Cohen-Coon

Tipe Kendali	K_p	T_i	T_d
P	$\frac{1}{K} \left(\frac{T}{L} \right) \left[1 + \frac{1}{3} \left(\frac{L}{T} \right) \right]$	-	-
PI	$\frac{1}{K} \left(\frac{T}{L} \right) \left[0,9 + \frac{1}{12} \left(\frac{L}{T} \right) \right]$	$L \left[\frac{30 + 3 \left(\frac{L}{T} \right)}{9 + 20 \left(\frac{L}{T} \right)} \right]$	-
PD	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{5}{4} + \frac{1}{6} \left(\frac{L}{T} \right) \right]$	-	$L \left[\frac{6 - 2 \left(\frac{L}{T} \right)}{22 + 3 \left(\frac{L}{T} \right)} \right]$
PID	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{4}{3} + \frac{1}{4} \left(\frac{L}{T} \right) \right]$	$L \left[\frac{32 + 6 \left(\frac{L}{T} \right)}{13 + 8 \left(\frac{L}{T} \right)} \right]$	$L \left[\frac{4}{11 + 2 \left(\frac{L}{T} \right)} \right]$

(Sumber : Wicaksono, 2004)