BAB 2 TINJAUAN PUSTAKA

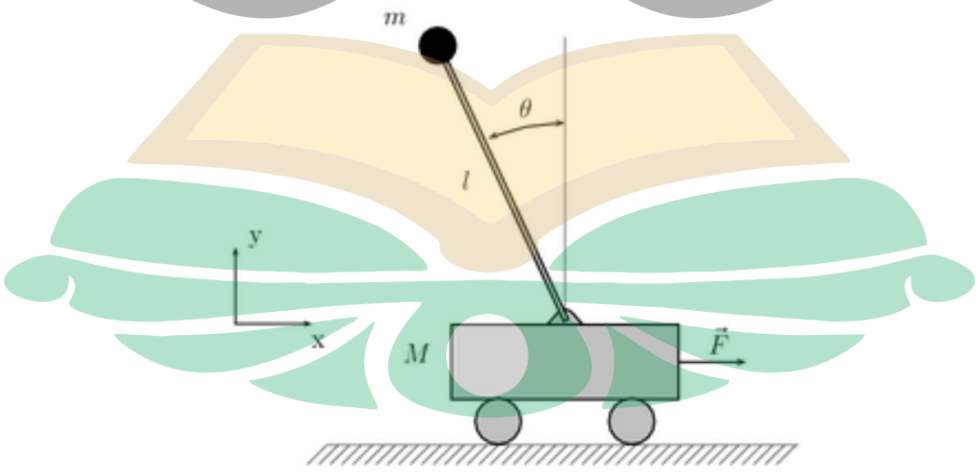
Pada bab tinjauan pustaka ini berisi tentang dasar teori dari penelitian yang akan dilakukan.

2.1. Robot *Self-Balancing*

Robot *self-balancing* beroda dua merupakan suatu robot *mobile* yang memiliki dua buah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Robot *self-balancing* beroda du aini merupakan pengembangan dari model pendulum terbalik yang diletakkan di atas kereta beroda.

Robot keseimbangan (*self-balancing robot*) beroda dua merupakan suatu robot *mobile* yang memiliki dua buah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. *Self-balancing* ini merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*)**.**

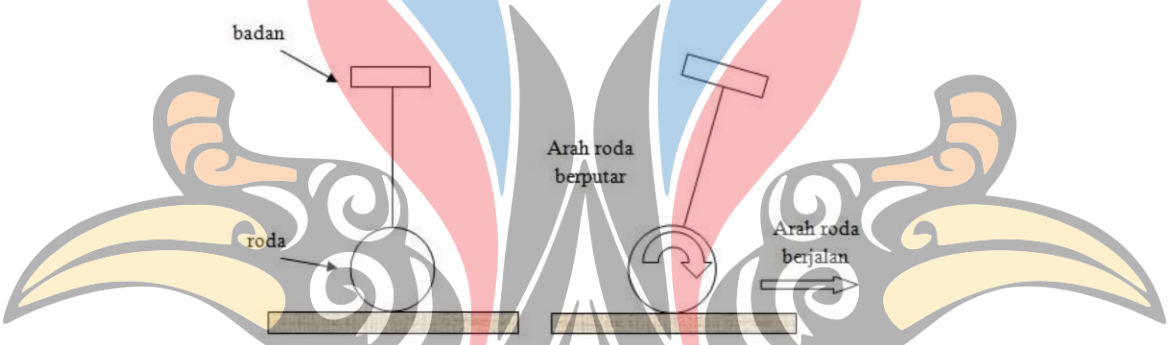
Inverted pendulum adalah pendulum yang terengsel ke kereta beroda yang dapat bergerak maju dan mundur pada bidang *horizontal* di sepanjang lintasan. Penerapan konsep pendulum terbalik dalam dunia robotika bisa dilihat pada robot keseimbangan, yaitu robot dengan dua roda yang roda tersebut diasumsikan sebagai kereta beroda dan badan robot diasumsikan sebagai pendulum. Sistem ini tidak stabil karena ketika kereta beroda diberi gangguan dari luar maka pendulum akan jatuh. Ketika pendulum atau *balancing robot*, mempertahankan agar pendulum tidak terjatuh dibutuhkan sebuah kendali suatu kendali khusus.



Gambar 2. 1 Pendulum terbalik di atas kereta beroda (Laksana,2012)

Pendulum terbalik yang diatur agar pada keadaan awal yang tegak, akan mulai membentuk sudut θ dan lama kelamaan akan jatuh karena adanya gaya gravitasi. Untuk mempertahankan posisi pendulum pada suatu titik, diperlukan sebuah gaya yang dapat menahan pergerakan pendulum. Cara menghasilkan gaya tersebut adalah dengan membuat kereta tersebut maju ke arah dengan arah kemana pendulum tersebut condong atau akan jatuh.

Kendali yang baik akan membuat pendulum tetap seimbang dengan cara mengendalikan pergerakan poros putar atau kereta beroda dimana pendulum tersebut terpasang. Dasar untuk membuat robot beroda dua dapat seimbang adalah dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas robot tersebut. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat seimbang.



Gambar 2. 2 Robot *self-balancing* beroda dua menyeimbangkan diri (Bobby,2015)

Robot *self-balancing* roda dua akan condong ke depan atau miring ke kanan, maka Tindakan yang perlu dilakukan adalah motor akan memutar roda searah jarum jam sehingga robot *self-balancing* beroda dua berjalan ke arah depan, sehingga robot akan kembali tegak lurus dengan permukaan bidang datar. Gaya yang digunakan untuk menyeimbangkan dihasilakan dari putaran roda. Putaran roda ini berasal dari torsi yang dihasilkan oleh motor (Laksana, 2012).

.

2.2 Hubungan Gaya, Percepatan Sudut, Percepatan Linear, Kecepatan Sudut dan Kecepatan Linear

Hubungan gaya, percepatan sudut, percepatan linear, kecepatan sudut dan kecepatan linear digunakan untuk menentukan nilai gaya dari robot *self-balancing.*

2.2.1 Gaya dengan Percepatan Sudut dan Kecepatan Sudut

Gaya ketika sebuah benda bergerak melingkar, maka benda tersebut akan mengalami sebuah gaya tarik yang arahnya menuju pusat lingkaran yang disebut dengan gaya sentripental. Gaya sentripental inilah yang selalu menahan benda agar tetap berada pada lintasan geraknya dan tidak terlepas. Gaya sentripental berbanding lurus dengan hasil kali *massa* dengan kuadrat kecepatan dan berbanding terbalik dengan jari-jari lintasan (Mikrajudin, 2016).

$$F = m.a_s = m.\omega^2.R = \frac{mv^2}{R} \quad (2.1)$$

Keterangan persamaan 2.1:

F = Gaya (N)

a = percepatan linear benda (m/s^2)

v = kecepatan linear benda (m/s^2)

ω = kecepatan sudut (rad/s)

R = jari-jari lintasan (m)

2.2.2 Percepatan Linear dan Percepatan Sudut

Percepatan linear atau percepatan tangensial pada gerak melingkar didefinisikan sebagai perubahan kecepatan linear dalam selang waktu tertentu dimana arah percepatan linear selalu menyinggung lintasan gerak benda yang berbentuk lingkaran (Mikrajuddin, 2016).

$$\alpha = \frac{\Delta v}{\Delta t} \quad (2.2)$$

Keterangan persamaan 2.2:

a = percepatan (m/s^2)

Δv = perubahan kecepatan (m/s)

Δt = selang waktu (s)

Percepatan sudut adalah perbandingan perubahan kecepatan sudut (ω) terhadap waktu (t). Dalam satuan internasional, percepatan sudut diukur dalam radian per detik kuadrat (rad/s^2) (Mikrajudin, 2016).

$$\alpha = \frac{d\omega}{dt} \quad (2.3)$$

Keterangan persamaan 2.3:

$d\omega$ = kecepatan sudut (rad/s)

dt = waktu (s)

Hubungan antara percepatam linear dan percepatan sudut dinyatakan dengan persamaan (2.4)

$$\alpha_t = \alpha R \quad (2.4)$$

Keterangan persamaan 2.4:

a_t = percepatan tangensial (m/ s^2)

α = percepatan angular (rad/ s^2)

R = jari-jari lingkaran (m)

2.2.3 Kecepatan Linear dan Kecepatan Sudut

Benda yang bergerak melingkar akan memiliki dua kecepatan, yaitu kecepatan linear dan kecepatan sudut. Kecepatan linear dapat dinyatakan pada persamaan (2.5).

$$v = \frac{2\pi R}{f} \quad (2.5)$$

Keterangan persamaan 2.5:

v = kecepatan linear gerak melingkar (m/ s)

T =periode putaran (s)

R = jari-jari lingkaran (m)

Arah kecepatan linear ini selalu berubah-ubah sesuai dengan arah lintasan yang ditempuhnya. Sedangkan dalam hal ini yang tidak berubah (konstan) adalah nilai kecepatan linear (Mikrajudin, 2016). Kecepatan sudut dinyatakan sebagai sudut yang ditempuh dibagi waktu tempuhnya.

$$\omega = \frac{2\pi}{T} \quad (2.6)$$

Keterangan persamaan 2.6:

ω = kecepatan linear gerak melingkar (m/ s)

T =periode putaran (s)

Besar kecepatan linear dan kecepatan sudut dihubungkan melalui persamaan yang ditunjukkan pada (2.7).

$$v = \omega . R \tag{2.7}$$

Keterangan persamaan 2.7:

ω = kecepatan sudut (rad/ s)

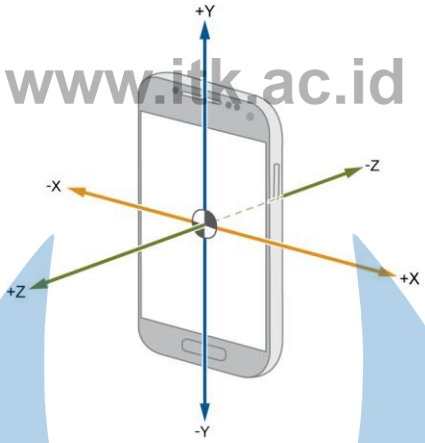
v = kecepatan linear (m/s)

R = jari-jari putaran(m

2.3 *Accelerometer*

Accelerometer merupakan sensor yang berfungsi untuk mengukur percepatan. Percepatan merupakan perubahan kecepatan terhadap waktu tertentu. Kecepatan yang berubah bisa semakin cepat ataupun lambat. Percepatan memiliki besaran dan arah, sehingga percepatan merupakan besaran vektor (Rakhman, 2015).

Accelerometer untuk saat ini adalah perangkat elektromekanis yang dapat mengukur percepatan statis dan linear. Semua benda di bumi dipengaruhi percepatan statis, sehingga selalu ditarik ke titik pusat bumi. Nilai percepatan statis tersebut bernilai sebesar 9,80665 m/s². Sedangkan percepatan linear adalah percepatan ketika benda bergerak. Contohnya, percepatan bola yang ditendang atau kendaraan yang sedang melaju di jalan. Dengan memanfaatkan percepatan statis ini, sensor dapat digunakan untuk mendeteksi nilai kemiringan dari sebuah benda. Pengaplikasian pengubahan mode orientasi pada layar *smartphone* juga memanfaatkan sensor *accelerometer*. Sedangkan pada percepatan linear, sensor pada aplikasinya dapat digunakan sebagai bagian dari peralatan navigasi. Saat ini terdapat 3 macam sensor *accelerometer*, 1 sumbu, 2 sumbu (x-y), dan 3 Sumbu (x-y-z). (Rakhman., 2015).

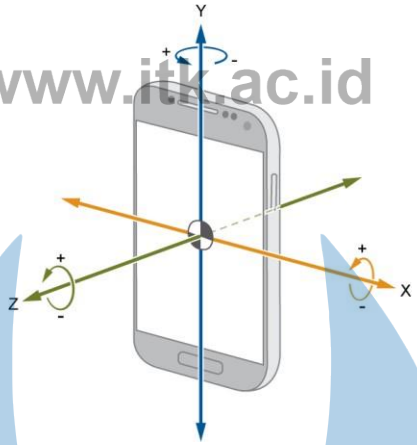


Gambar 2. 3 Arah sumbu *accelerometer* (mathworks.com)

Pendeteksian percepatan ini berdasarkan 3 sumbu yang berada pada bagian kanan-kiri (*Ax*), atas-bawah (*Ay*), dan depan-belakang (*Az*). Sensor ini dapat beroperasi untuk pengukuran kecepatan mesin, getaran mesin, getaran pada bangunan, dan kecepatan yang disertai pengaruh gravitasi bumi. Prinsip kerja dari *accelerometer* yaitu berdasarkan pada medan magnet yang digerakkan pada suatu konduktor pada medan magnet dan nantinya akan timbul induksi elektromagnetik pada konduktor tersebut. Contoh pengoperasian sensor accelerometer adalah pada penggunaan *Smartphone*, dan juga *Safety Installation* pada kendaraan mobil.

2.4 Gyroscope

Gyroscope berfungsi untuk mengukur atau menentukan orientiasi suatu benda berdasarkan pada ketetapan momentum sudut. Dari pengertian lain *gyroscope* berfungsi untuk menentukan gerakan sesuai dengan gravitasi yang dilakukan oleh pengguna. *Gyroscope* ini memiliki peranan yang sangat penting dalam hal mempertahankan keseimbangan suatu benda seperti penggunaannya pada pesawat terbang yang dapat menentukan kemiringan pada sumbu x,y, dan z. *Output* yang dihasilkan oleh *gyroscope* berupa kecepatan sudut yang pada sumbu x akan menjadi phi (Φ), sumbu y menjadi theta (θ), dan sumbu z menjadi psi (Ψ). Sebelum digunakan biasanya *gyroscope* dikalibarasi terlebih dahulu dengan menggunakan bandul yang fungsinya untuk menentukan nilai faktor ataupun dapat juga melihat pada datasheet sensor yang digunakan.



Gambar 2. 4 Arah sumbu *gyroscope* (mathworks.com)

Prinsip kerja dari *gyroscope* ini adalah pada saat *gyroscope* berotasi maka *gyroscope* akan memiliki nilai keluaran. Apabila *gyroscope* berotasi searah dengan jarum jam pada sumbu Z maka tegangan *ouput* yang dihasilkan akan mengecil sedangkan jika *gyroscope* berotasi berlawan arah dengan jarum jam pada sumbu Z maka tegangan output yang dihasilkan akan membesar. Pada saat *gyroscope* tidak sedang berotasi atau berada pada keadaan diam maka tegangan *ouput* akan sesuai dengan nilai *offset gyrosensor* tersebut (Rif'an, 2012).

2.5 IMU MPU6050

Dengan menggunakan kombinasi *accelerometer* dan *gyroscope* pada suatu sistem maka a*ccelerometer* dapat memberikan pengukuran sudut saat sistem berada pada kondisi diam. Sedangkan pada saat sistem berotasi *accelerometer* tidak bisa bekerja secara maksimal karena memiliki respon yang lambat. Kelemahan inilah yang dapat diatasi oleh *gyroscope* karena *gyroscope* dapat membaca kecepatan sudut yang dinamis. Namun *gyroscope* juga memiliki kelemahan yaitu proses perpindahan kecepatan sudut dalam jangka waktu yang panjang menjadi tidak akurat karena ada efek bias yang dihasilkan oleh *gyroscope*.

Contoh aplikatif kombinasi *accelerometer* dan *gyroscope* yaitu pada perangkat *iPhone* yang mengkombinasikan 2 sensor tersebut. Hal tersebut sangat membuat nyaman para pengguna dalam hal pendeteksian sensitivitas gerakan. Dari kombinasi *accelerometer* dan *gyroscope* didapatkan 6 sumbu pendeteksian yaitu 3 sumbu rotasi (x,y,z) dan 3 sumbu linier (atas-bawah, kanan-kiri, depan-belakang). *Output* dari kombinasi sensor ini berupa gambar yang sangat detail dan halus

gerakannya dibandingkan dengan *smartphone* yang hanya menggunakan *accelerometer* saja.

Salah satu IC kombinasi *accelerometer* dan *gyroscope* adalah IC MPU 6050. MPU 6050 merupakan kombinasi sensor antara *accelerometer* dan *gyroscope* meskipun pada dasarnya ada sensor temperaturnya. Akses sensor ini menggunakan fitur *I2c microcontroller*.

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya.

Sensor *three-axis accelerometer* ini digunakan sebagai sensor untuk mendeteksi kemiringan *payload* terhadap sumbu x (*roll*) dan sumbu y (*roll*). Untuk mencari sudut pitch dan roll menggunakan Persamaan (2.8) dan (2.9).

$$Pitch = artan(\frac{ax}{\sqrt{(ax)^2 + (az)^2}} \times (\frac{180}{\pi}) \quad (2.8)$$

$$Roll = artan(\frac{ax}{\sqrt{(ay)^2 + (az)^2}} \times (\frac{180}{\pi}) \quad (2.9)$$

Gambar 2. 5 Modul sensor MPU6050 (Invensense.com)

Adapun *pinout* pada MPU 6050 adalah sebagai berikut :

1. **VCC** : Tegangan untuk sensor, dapat dikoneksikan dengan tegangan 5 volt atau 3,3 volt
2. **GND** : untuk *grounding*

3. **SCL** : *Serial Clock Line* untuk I2C
4. **SDA** : *Serial Data Line* untuk I2C
5. **XDA** : *Auxilary Data*
6. **XCL** : *Auxilary Clock*
7. **AD0** : Ketika *pinout* ini di set menjadi rendah, I2C *address* menjadi 0×68, ketika di set menjadi tinggi, I2C *address* menjadi 0×69
8. **INT** : *Interrupt Digital Output*

Adapun fitur-fitur MPU 6050 ini antara lain:

1. Skala pengukuran *Accelerometer* yang dapat dipilih mulai 2/4/8 sampai 16 g.
2. Skala pengukuran *Gyroscope* yang dapat dipilih mulai 250/500/1000 sampai 2000 *degrees*/s.
3. *Range* 16 bit untuk kedua sensor
4. Sensitivitas percepatan linier dari *Gyroscope* 0,1 derajat/s

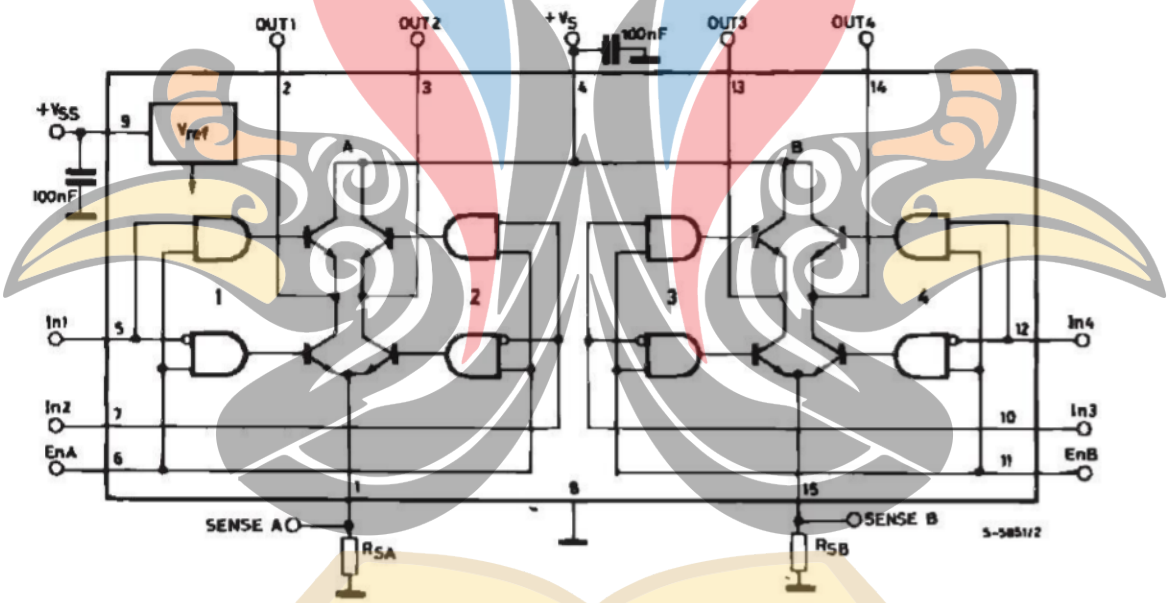
Data *rate output* hingga 1000 Hz, dilengkapi *digital low pass filter* dan memiliki frekuensi sudut maksimum 256 Hz (Irawan, 2015).

2.6 Driver Motor DC H-Bridge

Motor DC bergerak ketika *output* pin mikrokontroler dihubungkan langsung dengan motor DC, namun *output* arus dari mikrokontroler harus memenuhi kebutuhan motor DC. Jika kebutuhan arus motor besar, maka tidak dapat langsung dihubungkan dengan mikrokontroler karena *output* arus dari pin-pin mikrokontroler sangat kecil. Salah satu metode rangkaian motor DC adalah *H-Bridge*. Rangkaian *driver* motor DC *H-Bridge* merupakan suatu metode rangkaian transistor yang disusun menyerupai huruf H, rangkaian ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode *Pulse Width Modulation* (PWM) maupun metode sinyal logika dasar TTL (*high*) dan (*low*). Pengendalian motor DC dilakukan dengan memberikan PWM pada rangkaian *driver* motor DC sehingga kecepatan putaran motor DC dapat dikendalikan dengan baik. Apabila menggunakan metode logika TTL 0 dan 1 maka rangkaian ini hanya dapat mengendalikan arah putaran motor DC saja dengan kecepatan putaran motor DC maksimum. Rangkaian *driver* motor DC H-Bridge ini menggunakan rangkaian

jembatan transistor empat unit dengan proteksi tegangan induksi motor DC berupa dioda yang dipasang paralel dengan masing- masing transistor secara bias mundur.

Driver Motor DC dengan metode logika TTL (0 dan 1) atau *high* dan *low* hanya dapat mengendalikan arah putar motor DC dalam 2 arah tanpa pengendalian kecepatan putaran (kecepatan maksimum). Untuk mengendalikan motor DC dalam 2 arah dengan rangkaian *driver* motor DC *H-Bridge* pada Gambar 4.4 konfigurasi kontrol pada jalur *input* adalah dengan memberikan *input* berupa logika TTL ke jalur *input* A dan B. Untuk mengendalikan arah putar searah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur *input* A dan logika TTL 0 (*low*) pada jalur *input* B. Untuk mengendalikan arah putar berlawanan arah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur *input* B dan logika TTL 0 (*low*) pada jalur *input* A (Khakim, 2012).

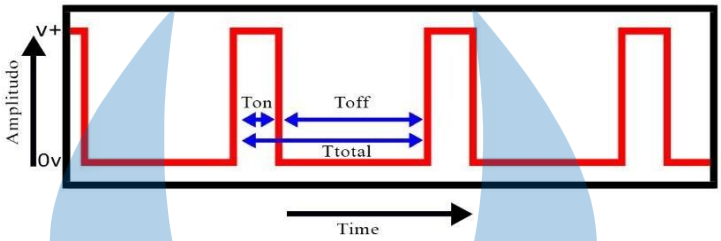


Gambar 2. 6 Rangkaian *driver* motor DC *H-Bridge* transistor (Antoni,2008)

2.7 Pulse Width Modulation

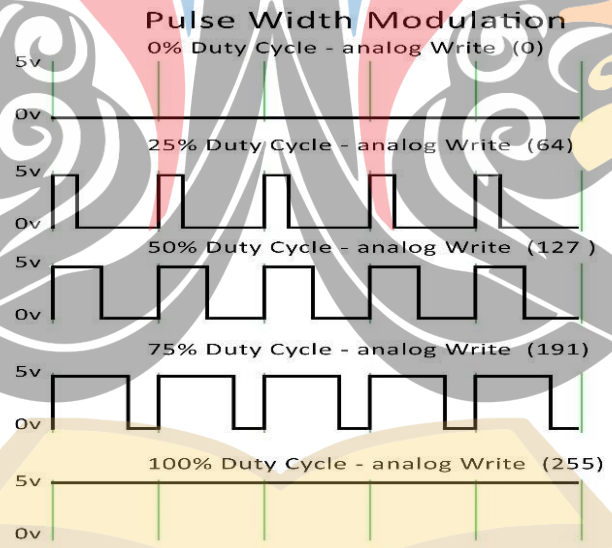
Pulse Width Modulation (PWM) secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam satu periode, untuk mendapatkan tegangan rata-rata yang berbeda. PWM juga merupakan sebuah mekanisme untuk membangkitkan sinyal *output* yang periodenya berulang antara *high* dan *low* dimana dapat mengontrol durasi sinyal (0 dan 1) atau *High* dan *Low*

sesuai dengan yang kita inginkan. *Duty cycle* merupakan prosentase periode sinyal *high* (1) dan periode sinyal *low* (0), prosentase *duty cycle* akan berbanding lurus dengan tegangan rata-rata yang dihasilkan (Marzuki, 2014).



Gambar 2. 7 Pulse width modulation (Marzuki,2014)

Sinyal PWM pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun memiliki lebar pulsa yang bervariasi. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. Sinyal PWM memiliki frekuensi gelombang yang tetap namun *duty cycle* bervariasi antara 0% hingga 100%.



Gambar 2. 8 Pembangkitan sinyal PWM secara digital (Khakim, 2015)

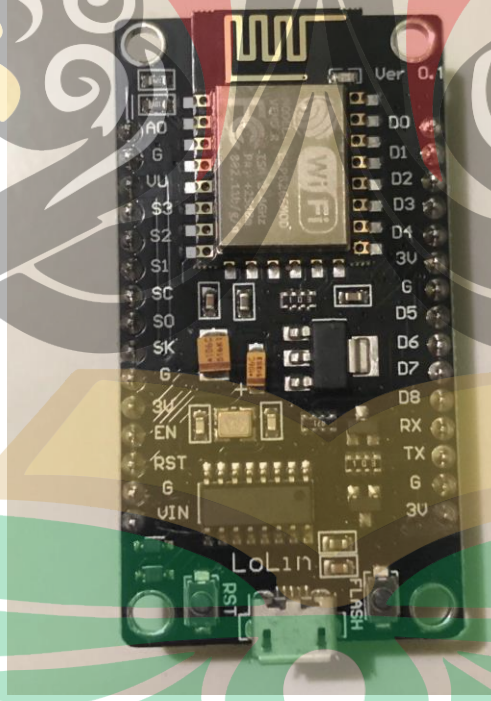
PWM merupakan salah satu teknik untuk mendapatkan sinyal analog dari sebuah piranti digital. Sinyal PWM dapat dibangkitkan dengan banyak cara, secara analog menggunakan IC *op-amp* atau secara digital. Secara analog setiap perubahan PWM-nya sangat halus, sedangkan secara digital setiap perubahan PWM dipengaruhi oleh resolusi PWM itu sendiri. Resolusi adalah jumlah variasi perubahan nilai dalam PWM tersebut. Misalkan suatu PWM memiliki resolusi 8

bit, berarti PWM ini memiliki variasi perubahan nilai sebanyak 256 variasi mulai dari 0 – 225 perubahan nilai yang mewakili *duty cycle* 0% – 100% dari output PWM tersebut (Khakim, 2015).

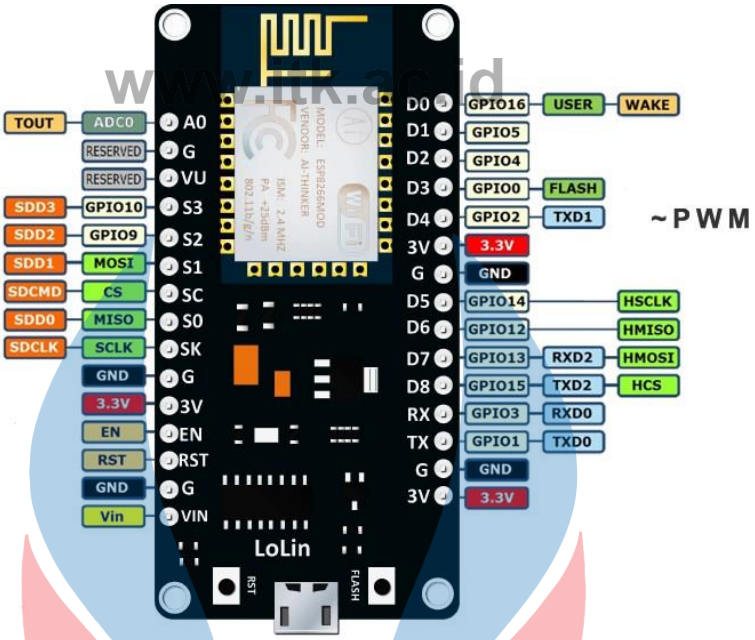
2.8 NodeMcu ESP8266

NodeMCU ESP8266 merupakan modul mikrokontroler yang didesain dengan ESP8266 di dalamnya. ESP8266 berfungsi untuk konektivitas jaringan Wifi antara mikrokontroler itu sendiri dengan jarigan Wifi. NodeMCU berbasis bahasa pemograman luar namun dapat juga menggunakan Arduino IDE untuk pemogramannya. NodeMCU ESP8266 memiliki pin I/O yang memadai dan dapat mengakses jaringan internet unuk mengirim atau mengambil data melalui koneksi WiFi (Septama,2018). Susunan kaki-kaki board NodeMCU ESP8266 diperlihatkan pada Gambar 2.10.

Modul NodeMCU ESP8266 diperlihatkan pada Gambar 2.9



Gambar 2. 9 NodeMcu ESP8266 (Espressif.com)



Gambar 2. 10 Pinout NodeMCU ESP8266 (Espressif.com)

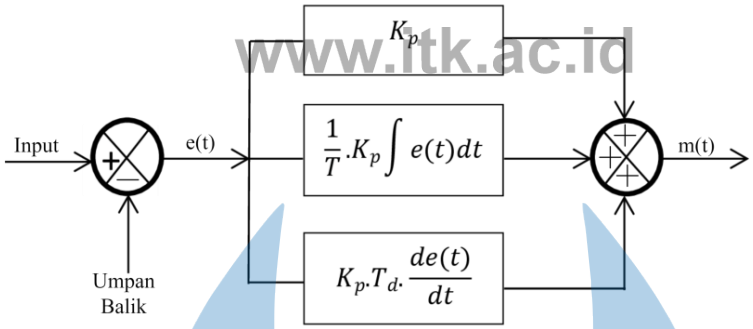
2.9 Kendali *Proportional Integral Derivative* (PID)

Dengan menggabungkan tiga pengontrol P, I dan D secara paralel menjadi pengendali PID, kekurangan dan kelebihan masing-masing dapat dikompensasikan satu sama lain. Kombinasi kendali proporsional, *integral* dan *derivative* dalam sistem kendali PID memiliki tujuan tertentu. Kendali proporsional memiliki waktu naik cepat yang sangat baik, kendali integral dapat menghilangkan *error steady state*, dan kontrol *derivative* dapat mengurangi *overshoot*. Jika digunakan dalam kombinasi, maka akan diperoleh hasil kontrol, dengan sifat menghilangkan *error*, mengurangi *rise time*, mengurangi *settling time* dan mengurangi *overshoot*. Karakteristik dari masing-masing parameter sistem kendali PID dapat dilihat pada Tabel 2.1 (Sighn, 2013).

Tabel 2. 1 Karakteristik Parameter PID

Penguatan	*Rise Time*	*Overshoot*	*Settling Time*	*Error Steady State*
K	Berkurang	Bertambah	Sedikit Berubah	Berkurang
Ki	Berkurang	Bertambah	Bertambah	Hilang Sedikit
Kd	Sedikit Berubah	Berkurang	Berkurang	Berubah

*)Yuan&Liu, 201



Gambar 2. 11 Blok diagram kontroler PID (Sighn, 2013)

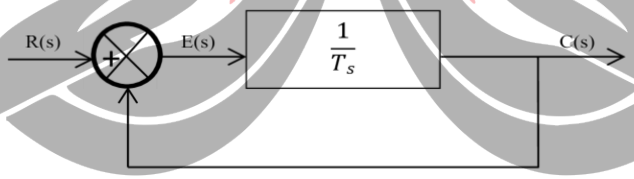
Model matematis dan struktur paralel sistem kendali PID pada (Gambar 2.11) diatas dapat dituliskan pada persamaan 2.10

$$G(s) = K_p\left[1 + \frac{1}{T_1 s} + T_D s\right] = K_p + \frac{K_1}{s} + K_D s \qquad (2.10)$$

2.10 Respon Transien

2.10.1 Respon Transien *Unit Step* Orde 1

Mengacu pada Gambar 2.12 hubungn antara input dan output dapat diberikan oleh persamaan 2.11



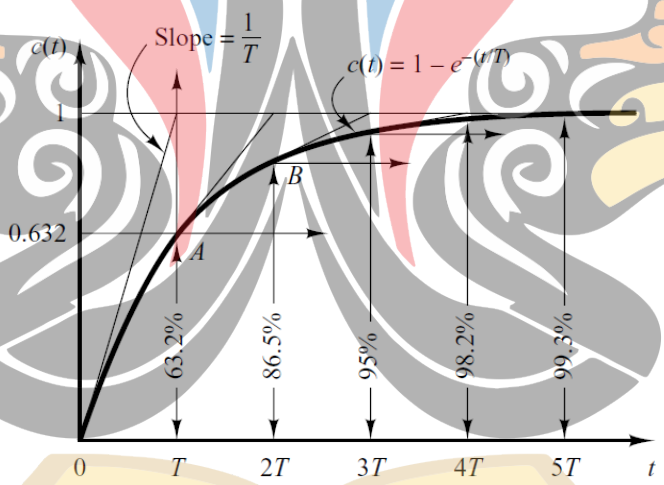
Gambar 2. 12 Blok diagram sistem orde 1(Ogata, 2010)

$$\frac{C(s)}{R(s)} = \frac{1}{T_s + 1} \qquad (2.11)$$

Karena transformasi *laplace* dari fungsi *unit-step* adalah 1 / s, dengan mengganti R (s) = 1 / s ke dalam Persamaan 2.11, maka dapat diperoleh persamaan 2.12(Ogata, 2010)

$$\frac{C(s)}{R(s)} = \frac{1}{T_s + 1}\frac{1}{s} \tag{2.12}$$

Kurva respons eksponensial C(t) yang diberikan oleh persamaan 2.3 ditunjukkan pada Gambar 2.11. Dalam konstanta waktu, kurva respons eksponensial berubah dari nilai akhir 0 menjadi 63,2%. Dalam dua konstanta waktu, responsnya adalah 86,5% dari nilai akhir. Pada t = 3T, 4T dan 5T, tanggapan masing-masing adalah 95%, 98,2% dan 99,3% dari nilai akhir. Oleh karena itu, untuk t ≥ 4T, respons tetap dalam 2% dari level akhir. Kondisi mapan dapat diperoleh secara matematis hanya setelah waktu yang tak terbatas. Namun, pada kenyataannya, perkiraan waktu respons yang wajar adalah lamanya waktu yang dibutuhkan kurva respons mencapai dan tetap berada dalam garis 2% (atau empat konstanta waktu) dari nilai akhir (Ogata, 2010).

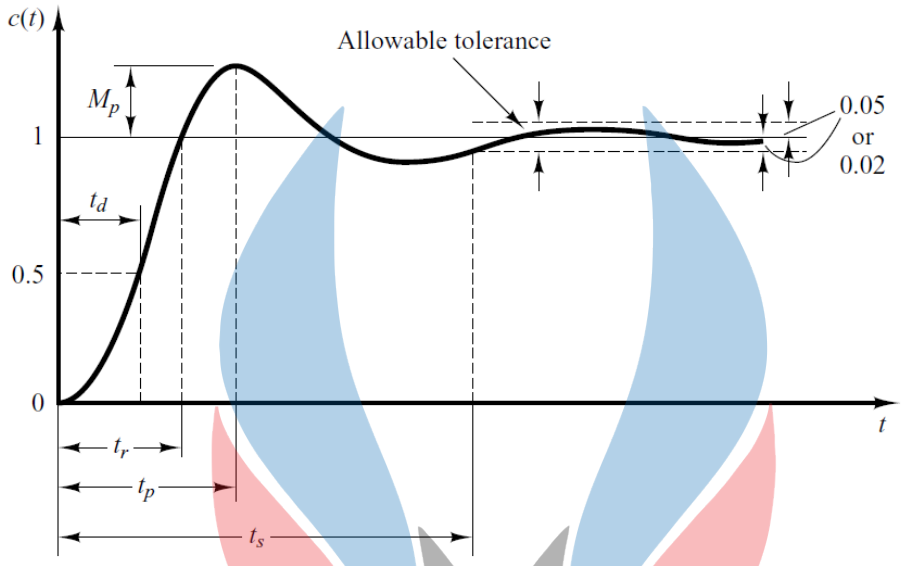


Gambar 2. 13 Kurva respon eksponensial (Ogata, 2010)

2.10.2 Respon Transien Unit Step Orde 2

Sistem kendali umumnya dirancang memiliki faktor redaman lebih kecil dari satu, misalnya pada respon berupa *osilasi* dari masukan sinyal *step*. Pada sistem kendali orde yang lebih tinggi lagi biasanya memiliki *pole-pole* konjugate kompleks dengan faktor redaman lebih besar dari satu yang cenderung melampaui *pole* yang lain. Oleh karena itu, waktu respon sistem kendali orde dua dan orde lebih tinggi dengan *input* sinyal *step* biasanya berbentuk redaman osilasi alami.

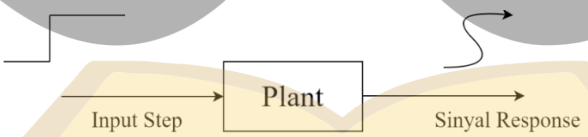
Seperti terlihat pada gambar, responnya memiliki "*overshoot*" dan "*undershoot*", dan hampir tidak ada waktu akhir (Ogata, 2010).



Gambar 2. 14 Grafik spesifikasi respon waktu (Ogata, 2010)

2.11 Metode *Cohen-Coon*

Cohen-Coon bisa di pakai untuk mendesain PID *Controller* dan *Cohen-Coon* bisa dipakai untuk plant yang memiliki *deadtime* besar (delay besar). Pada *Cohen-Coon* plant diberi *input step*, lalu respon dibiarkan sampai *steady state* seperti Gambar 2.13 :



Gambar 2. 15 Respon 1 *steady state* pada *Cohen-Coon* (Simamora, K. 2015)

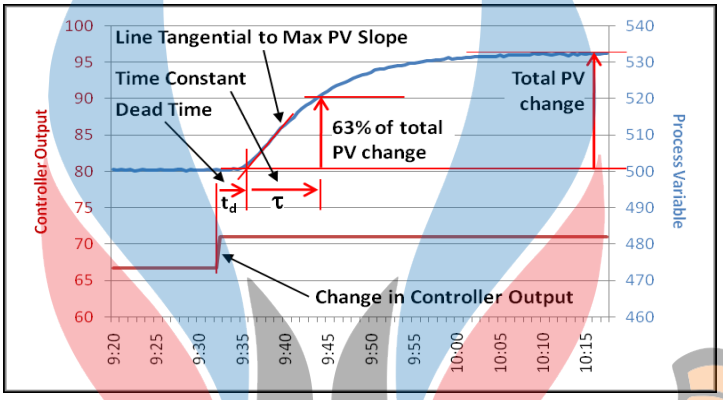
Input step di tambah lalu respon dibiarkan sampai mencapai *steady-state*. Sinyal perubahan inilah yang akan di pakai untuk mendesain kendali seperti Gambar 2.16

Gambar 2. 16 Respon 2 *steady state* pada *Cohen-Coon* (Simamora, K.2015)

Pada desain CC ini kita harus mencari nilai Gp (*gain process*), td (*time delay*), dan tau (*time constant*), dengan persamaan sebagai berikut :

$$Gp = \frac{Perubahan\ PV}{Perubahan\ Co} \tag{2.4}$$

Perubahan PV = y atas (steady state gelombang ke 2) – y bawah (saat step ke 2 naik) τ = perubahan PV x 0.63 Seperti penjelasan Gambar 2.17 :



Gambar 2. 17 Penjelasan perhitungan dengan metode *Cohen-Coon* (G.H. Cohen and G.A. Coon, 1953)

Setelah didapatkan nilai Gp, td dan tau, maka dapat dihitung nilai Kp, Ti dan Td dengan rumus mencari parameter PID dengan metode *cohen*. Tabel 2.2 menunjukkan tabel parameter PID untuk *Cohen-Coon*. (G.H. Cohen and G.A. Coon, 1953)

Tabel 2. 2 Parameter PID untuk *Cohen-Coon*

	Controller Gain	*Integral Time*	*Derivative Time*
P *Controller*	$K_c = \frac{1}{rK}\left\{1 + \frac{r}{3}\right\}$		
PI *Controller*	$K_c = \frac{1}{rK}\left\{0.9 + \frac{r}{12}\right\}$		
PD *Controller*	$K_c = \frac{1.24}{g_p}\left\{\frac{\tau}{t_d} + 0.129\right\}$	$T_1 = t_d \frac{30 + 3r}{9 + 20r}$	
PID *Controller*	$K_c = \frac{1}{rK}\left\{\frac{4}{3} + \frac{r}{4}\right\}$	$T_1 = t_d \frac{32 + 6r}{13 + 8r}$	$T_1 = 0.37 t_d \frac{\tau}{\tau + 0.185 t_d}$

Apabila harga Ti dan Td sudah diketahui, maka konstanta Ki dan Kd dapat ditentukan dengan rumus sebagai berikut :

$$K_i = \frac{K_c}{T_i} \quad (2.5)$$

$$K_d = K_p \times T_d \quad (2.6)$$

2.12 Penelitian Terdahulu

Pada Tabel 2.3 adalah rangkuman hasil penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang telah dilakukan.

Tabel 2. 3 Penelitian Terdahulu

No	Nama dan Tahun Publikasi	Hasil
1.	Congying Qiu 2015	Metode : *Proportional Integral Derivative dan Fuzzy Adaptive* Spesifikasi : *TMS320LF2407 DSP, Mio-x AHRS Module,* Matlab Hasil : Peningkatan struktur sistem kendali bisa mempersingkat waktu settling robot dan mengurangi overshoot sistem. Kp=500; Ki=50; dan Kd=40
2	Bagus Rio Rynaldo 2018	Metode : *Proportional Integral* Spesifikasi : IMU MPU6050, Matlab, Arduino Uno, Motor Driver, Modul Radio nRF24L01 Hasil : Kp=67,5; Ki=83,509
3	Efrain Mendez 2020	Metode : *Proportional Integral* dan *Derivative* Spesifikasi : MK64FN1M0VLL12, FXOS8700CQ, Matlab Hasil : Kp=67,5; Ki=83,509

No	Nama dan Tahun Publikasi	Hasil
4	Nuril Bahroin 2021	Metode: *Proportional Integral* dan *Derivative* dengan *tuning Cohen-Coon* Spesifikasi : IMU MPU6050, NodeMcu ESP8266, Motor Driver L298N Hasil :