

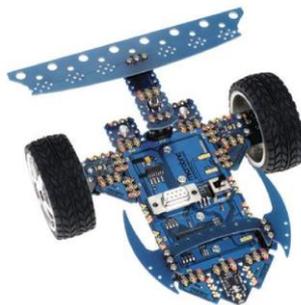
## BAB II

### LANDASAN TEORI

#### 2.1 Robot

Robot didefinisikan sebagai sebuah *automaton*, yakni suatu piranti mekanik yang cerdas yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu kecerdasan buatan (*artificial intelligence*). Kata robot berasal dari bahasa Czech, *robota* yang berarti pekerja, mulai menjadi 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). (Pitowarno, 2006: 1)

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 dan luar angkasa, pertambangan dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput.



Gambar 2. 1 Robot Beroda (Naili Suri, 2017)

Jika sebelumnya robot hanya dioperasikan di laboratorium ataupun dimanfaatkan untuk kepentingan industri, di negara-negara maju perkembangan robot mengalami

peningkatan yang tajam, saat ini robot telah digunakan sebagai alat untuk membantu pekerjaan manusia. Seiring dengan berkembangnya teknologi, khususnya teknologi elektronik, peran robot menjadi semakin penting tidak saja dibidang sains, tapi juga di berbagai bidang lainnya, seperti di bidang kedokteran, pertanian, bahkan militer. Secara sadar atau tidak, saat ini robot telah masuk dalam kehidupan manusia sehari-hari dalam berbagai bentuk dan jenis. Ada jenis robot sederhana yang dirancang untuk melakukan kegiatan yang sederhana, mudah dan berulang-ulang, ataupun robot yang diciptakan khusus untuk melakukan sesuatu yang rumit, sehingga dapat berperilaku sangat kompleks dan secara otomatis dapat mengontrol dirinya sendiri sampai batas tertentu. Robot memiliki berbagai macam konstruksi. Diantaranya adalah:

1. *Robot Mobile* (bergerak)
2. Robot Manipulator (lengan)
3. Robot Humanoid
4. *Flying Robot*
5. Robot Berkaki
6. Robot Jaringan
7. Robot Animalia

Pada gambar 2.1 robot beroda dapat didefinisikan sebagai sebuah alat mekanik yang dapat diprogram berdasarkan informasi dari lingkungan (melalui sensor) sehingga dapat melaksanakan beberapa tugas tertentu baik secara otomatis ataupun tidak sesuai program yang dimasukan berdasarkan logika.

## **2.2 Mikrokontroler**

Mikrokontroler merupakan alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data (Malik, 2009).

Keunggulan mikrokontroler adalah terdapatnya memori dan *port input* dan *output* dalam suatu kemasan IC yang kompak. Kemampuannya yang *programmable*, fitur yang lengkap (ADC internal, EEPROM internal, port I/O, komunikasi serial), dan juga harga yang terjangkau memungkinkan mikrokontroler digunakan di berbagai sistem elektronis, seperti pada robot, automasi industri, sistem alarm, peralatan telekomunikasi, hingga peralatan rumah tangga (Wardhana, 2006).

## 2.3 Implementasi Teknologi Digital

### 2.3.1 Arduino Nano

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, yang dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware* (perangkat keras)-nya memiliki prosesor Atmel AVR dan *software* (perangkat lunak)-nya memiliki bahasa pemrograman sendiri. *Open source IDE* yang digunakan untuk membuat aplikasi mikrokontroler yang berbasis platform arduino. Mikrokontroler *single-board* yang bersifat *open source hardware* dikembangkan untuk arsitektur mikrokontroler AVR 8 bit dan ARM 32 bit.

Dari pengertian di atas, dapat disimpulkan bahwa arduino adalah kit atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR. Mikrokontroler itu sendiri adalah *chip* atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output seperti yang diinginkan. Jadi mikrokontroler bertugas sebagai otak yang mengendalikan input, proses, dan output sebuah rangkaian elektronik.

Mikrokontroler terdapat pada perangkat elektronik sekelilingnya, misalnya *Handphone*, *MP3 Player*, *DVD*, *Televisi*, *AC*, dan lain-lain. Mikrokontroler juga dapat

mengendalikan robot, baik robot mainan maupun industri. Karena komponen utama arduino adalah mikrokontroler, maka arduino dapat diprogram menggunakan komputer sesuai kebutuhan.

Kelebihan arduino, antara lain:

1. Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer.
2. Sudah memiliki sarana komunikasi USB, sehingga pengguna laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.
3. Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* arduino. Contohnya *shield* GPS, *Ethernet*, dan lain-lain.

Komponen Arduino, antara lain:

1. Soket USB

Soket USB adalah soket kabel USB yang disambungkan ke komputer atau laptop, yang berfungsi untuk mengirimkan program ke arduino dan juga sebagai *port* komunikasi serial.

2. Input / Output Digital dan Input Analog

Input/output digital atau digital pin adalah pin-pin untuk menghubungkan arduino dengan komponen atau rangkaian digital, contohnya, jika ingin membuat LED berkedip, LED tersebut bisa dipasang pada salah satu pin input atau output digital dan *ground* komponen lain yang menghasilkan output digital atau menerima input digital bisa disambungkan ke pin ini.

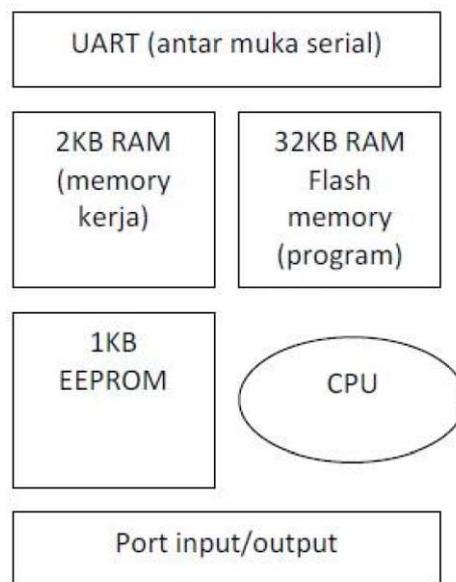
Input analog atau analog pin adalah pin-pin yang berfungsi untuk menerima sinyal dari komponen atau rangkaian analog, contohnya; potensiometer, sensor suhu, sensor cahaya, dan lain-lain.

3. Catu daya

Pin catu daya adalah pin yang memberikan tegangan untuk komponen atau rangkaian yang dihubungkan dengan arduino. Pada bagian catu daya ini pin V input dan Reset. V input digunakan untuk memberikan tegangan langsung kepada arduino tanpa melalui tegangan pada USB atau adaptor, sedangkan Reset adalah pin untuk memberikan sinyal reset melalui tombol atau rangkaian eksternal.

#### 4. Baterai / Adaptor

Soket baterai atau adaptor digunakan untuk menyuplai arduino dengan tegangan dari baterai/adaptor 9V pada saat arduino sedang tidak disambungkan ke komputer. Jika arduino sedang disambungkan ke komputer dengan USB, Arduino mendapatkan suplai tegangan dari USB, Jika tidak perlu memasang baterai atau adaptor pada saat memprogram arduino. Untuk memberikan gambaran mengenai apa saja yang terdapat di dalam sebuah mikrokontroler, gambar 2.2 memperlihatkan contoh diagram blok sederhana dari mikrokontroler Atmega 328 (dipakai pada Arduino Uno).



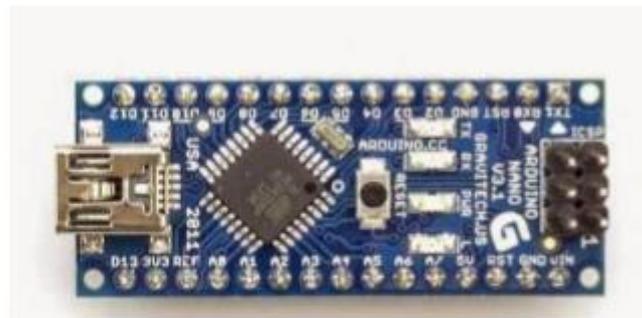
Gambar 2. 2 Diagram Sederhana Mikrokontroler Atmega 328

Blok-blok di atas dijelaskan sebagai berikut:

- a) *Universal Asynchronous Receiver/Transmitter* (UART) adalah antar muka yang digunakan untuk komunikasi serial seperti pada RS-232, RS-422 dan RS-485.
- b) 2KB RAM pada memori kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variabel-variabel di dalam program.
- c) 32KB RAM *flash memory* bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, *flash memory* juga menyimpan *bootloader*. *Bootloader* adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah *bootloader* selesai dijalankan, berikutnya program di dalam RAM akan dieksekusi.
- d) 1KB EEPROM bersifat *non-volatile*, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino.
- e) *Central Processing Unit* (CPU), bagian dari mikrokontroler untuk menjalankan setiap instruksi dari program.
- f) Port input/output, pin-pin untuk menerima data (input) digital atau analog, dan mengeluarkan data (output) digital atau analog.

#### Bagian – Bagian Papan Arduino

Pada gambar 2.3 dengan mengambil contoh sebuah papan Arduino tipe USB, bagian-bagiannya dapat dijelaskan sebagai berikut.



Gambar 2. 3 Bentuk Fisik Arduino Nano (Rangga, 2016)

- a) 14 pin input/output digital (0-13)

Berfungsi sebagai input atau output, dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog output dimana tegangan output-nya dapat diatur. Nilai sebuah pin output analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.

b) USB

Berfungsi untuk memuat program dari komputer ke dalam papan, komunikasi serial antara papan dan komputer dan memberi daya listrik kepada papan.

c) Sambungan SV1

Sambungan atau jumper untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.

d) Q1 = Kristal (*quartz crystal oscillator*)

Jika mikrokontroler dianggap sebagai sebuah otak, maka kristal adalah jantungnya karena komponen ini menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).

e) Tombol Reset S1

Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan mikrokontroler.

f) In = *Circuit Serial Programming (ICSP)*

Port ICSP memungkinkan pengguna untuk memprogram mikrokontroler secara langsung, tanpa melalui *bootloader*. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.

g) IC 1 = Mikrokontroler Atmega

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.

h) X1 = Sumber Daya Eksternal

Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan dc antara 9-12V.

i) 6 Pin Input analog (0-5)

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V (Arduino, 2012).

### 2.3.2 Gyroscope

*Rate-gyroscope* adalah sebuah sensor yang digunakan untuk mengukur rotasi benda, benda dikatakan berotasi jika benda tersebut bergerak pada sumbunya (Priswanto, 2009).

Prinsip kerja *rate-gyroscope* sama seperti gasing. Jika pusran (berputar pada sumbunya) gasing melambat, gasing itu miring ke samping. Gaya gravitasi menarik gasing ke bawah, tetapi gasing itu tidak jatuh. Gasing berpusar melambat di sekeliling titik keseimbangannya. Benda yang berpusar mengubah gaya vertikal gravitasi menjadi gerakan horisontal, fenomena seperti itu disebut dengan presisi (Kindersley, 1992)

Pengukuran kecepatan sudut membutuhkan alat yang dapat merubah kecepatan sudut menjadi besaran listrik sehingga dapat dianalisis besar kecepatan sudutnya, alat tersebut adalah sensor *rate-gyroscope*. Berdasarkan *datasheet*, persamaan yang digunakan untuk menghitung tegangan keluaran dari sensor *rate-gyroscope* sebagai berikut:

$$V_{out} = V_o + (Svx\omega)$$

$V_{out}$  adalah tegangan keluaran dalam satuan V,  $V_0$  adalah tegangan keluaran statis (saat kecepatan sudut nol) dalam V,  $S_v$  (*scale factor*) adalah faktor skala dalam satuan V/deg/s, dan  $\omega$  adalah kecepatan sudut atau *angular velocity* dalam satuan deg/s.

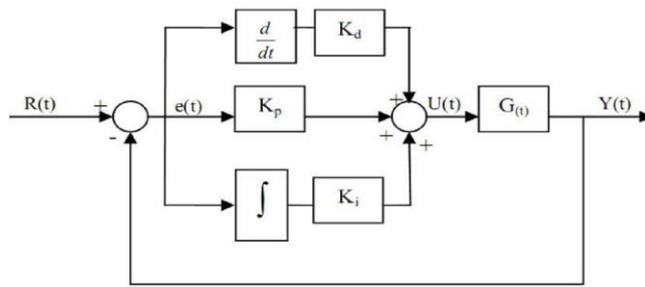
Perubahan temperatur mempengaruhi semua piranti elektronik dan dapat mengakibatkan perubahan arus *offset* dan tegangan *offset*, istilah ini disebut dengan geseran (*drift*). Keluaran dari sensor tidak murni nilai dari sensor tetapi ada *noise*, bias, dan *drift* dari sensor yang terbawa dan dikuatkan juga sampai pada output sistem. Tingkat eksitasi atau percepatan dari sensor tidak mempengaruhi bias. *Noise* yang disebabkan karena perubahan temperatur disebut dengan *drift noise* yang memiliki frekuensi rendah, sehingga untuk mengurangi *noise* tersebut digunakan *low pass filter* (Priwanto, 2009). Pengkondisi sinyal LPF (*Low Pass Filter*) yang berfungsi melewatkan sinyal pada frekuensi rendah agar keluaran sensor *rate-gyroscope* lebih stabil. Rangkaian LPF yang digunakan adalah rangkaian LPF orde 1. Perhitungan frekuensi *cut-off* untuk orde 1 (kutub tunggal) dihitung dengan menggunakan persamaan berikut:

$$f_c = \frac{1}{2\pi RC}$$

$f_c$  adalah frekuensi *cut-off*. R adalah nilai resistansi yang digunakan. C adalah nilai kapasitansi yang digunakan

### 2.3.3 Kontroler PID

Kontroler PID (*Proportional Integral Derivative controller*) merupakan kontroler yang terdiri dari pengendali Proporsional, Integral dan Turunan (*Derivative*). Dalam pengaplikasiannya, masing-masing pengendali dapat berdiri sendiri atau dapat melakukan pengkombinasian. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I dan D agar respon sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan (Royyan, 2015).



Gambar 2. 4 Diagram Block PID

Pada gambar 2.4 adalah Skema kontrol PID dinamai dari ketiga term pengendalinya, yang kemudian dijumlahkan menjadi variabel manipulasi. *Term* proporsional, integral, dan derivatif dijumlahkan untuk menghitung keluaran kontroler PID. Dengan  $u(t)$  mendefinisikan sebagai keluaran kontroler, bentuk akhir dari algoritme PID adalah:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

$u(t)$ : sinyal keluaran pengendali PID

$K_p$ : *Gain* proporsional, parameter tuning

$K_d$ : *Gain* Integral, parameter tuning

$K_i$ : *Gain* Derivatif, parameter tuning

$e$ : sinyal kesalahan (referensi-output)

$t$ : Waktu

$\tau$ : Variabel integrasi; nilainya diambil dari waktu nol sampai

### 2.3.3.1 Proporsional

Proporsional akan menghasilkan nilai keluaran yang berbanding lurus dengan nilai kesalahan. Responnya dapat diatur dengan mengalikan kesalahan (error) dengan konstanta  $K_p$ , disebut konstanta *gain* proporsional atau *gain* kontroler.

Term proporsional dirumuskan:

$$P_{out} = K_p e(t)$$

$$P_{out} = K_p (Y_{sp} - Y_m)$$

$P_{out}$  : Sinyal keluaran proporsional

$K_p$  : Gain proporsional, parameter tuning

$e$ : error

$Y_{sp}$  : Set point

$Y_m$  : Variable Proses

$t$ : Waktu

*Gain* yang besar menghasilkan perubahan yang besar pada keluaran untuk suatu nilai kesalahan tertentu. Namun, jika gain terlalu besar, sistem akan membutuhkan waktu yang cukup lama untuk mencapai kondisi *steady-state*. Sebaliknya, *gain* yang bernilai kecil maka respon keluaran juga kecil, sehingga kontroler menjadi kurang responsif/sensitif, hal ini akan mengakibatkan respon kontroler akan lebih lambat jika mendapatkan gangguan.

### 2.3.3.2 Integral

Peranan dari term integral berbanding lurus dengan besar dan lamanya error. Integral dalam kontroler PID adalah jumlahan error setiap waktu dan mengakumulasi *offset* yang sebelumnya telah dikoreksi. Error terakumulasi dikalikan dengan *gain* integral ( $K_i$ ) dan menjadi keluaran kontroler.

Term integral dirumuskan dengan:

$$I_{out} = K_i \int_0^t e(\tau) d\tau$$

$I_{out}$  : Sinyal keluaran integral

$K_i$  : Gain Derivatif, parameter tuning

$\tau$ : Variabel integrasi; nilainya diambil dari waktu nol sampai

Term integral mempercepat perpindahan proses menuju *setpoint* dan menghilangkan *steady-state error* yang muncul pada kontroler proporsional. Namun, karena integral

merespon terhadap eror terakumulasi dari sebelumnya, maka dapat menyebabkan *overshoot*.

### 2.3.3.3 Derivatif

Turunan eror pada proses dihitung dengan menentukan kemiringan eror setiap waktu dan mengalikan perubahan tiap waktu dengan *gain* derivatif  $K_d$ .

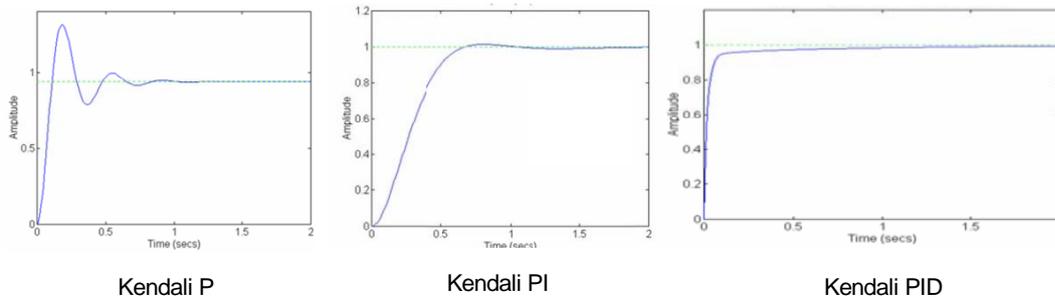
Term derivatif dirumuskan dengan:

$$D_{out} = K_d \frac{de(t)}{dx}$$

Aksi derivatif memprediksi perilaku sistem dan kemudian memperbaiki waktu tinggal dan stabilitas sistem. Aksi derivatif jarang digunakan pada industri- diperkirakan hanya 25% kontroler karena akibatnya pada stabilitas sistem pada aplikasi dunia nyata. Efek dari setiap pengontrol Proporsional, Integral dan Derivatif pada sistem lup tertutup disimpulkan pada tabel 2.1 dan pada gambar 2.5.

Tabel 2.1 Efek Dari Setiap Kontroler

<u>Respon Lup Tertutup</u>	<u>Rise Time</u>	<u>Overshoot</u>	<u>Settling Time</u>	<u>Steady-State Error</u>
Proporsional	Menurunkan	Meningkatkan	Perubahan kecil	Menurunkan/mengurangi
Integral	Menurunkan	Meningkatkan	Meningkatkan	Mengeliminasi
Derivatif	Perubahan Kecil	Menurunkan	Menurunkan	Perubahan kecil



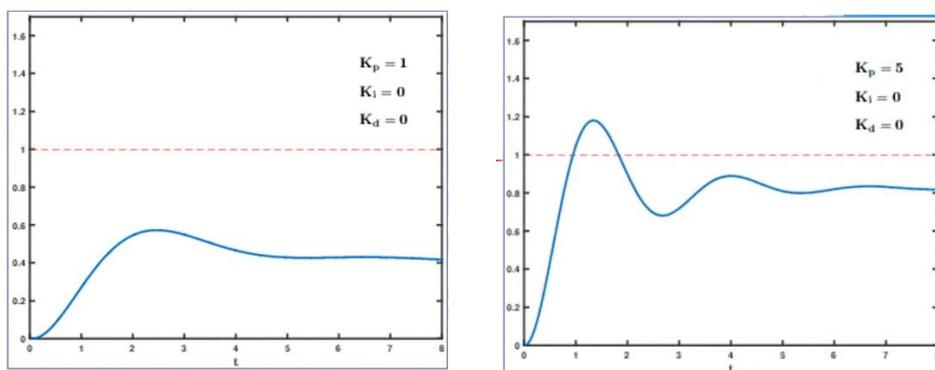
Gambar 2. 5 Respon Setiap Kendali

### 2.3.4 Metode Tuning PID

#### 2.3.4.1 Manual Tuning

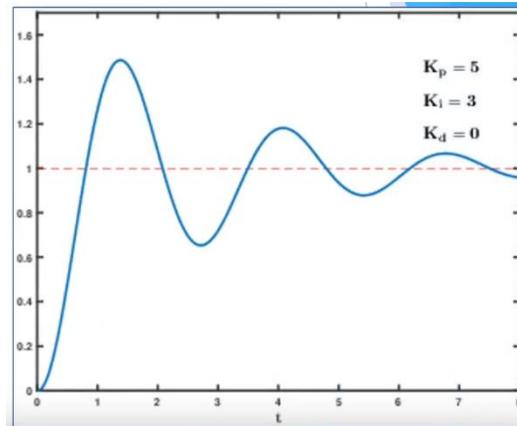
Penyetelan loop PID yang cepat biasanya melampaui sedikit untuk mencapai set-point lebih cepat namun, beberapa sistem tidak dapat menerima overshoot, dalam hal ini diperlukan sistem close-loop yang overdamped, yang akan memerlukan pengaturan  $K_p$  secara signifikan kurang dari setengah pengaturan  $K_p$ . pengaturan  $K_p$  yang menyebabkan osilasi.

Dalam metode ini salah satu cara penyetelan adalah dengan memasukkan nilai  $K_i$  dan  $K_d$  terlebih dahulu ke nol. tingkatkan  $K_p$  hingga output loop berosilasi, maka  $K_p$  harus disetel kira-kira setengah dari nilai itu, hasil grafik dari memberi nilai  $K_p$  dapat dilihat pada gambar 2.6.



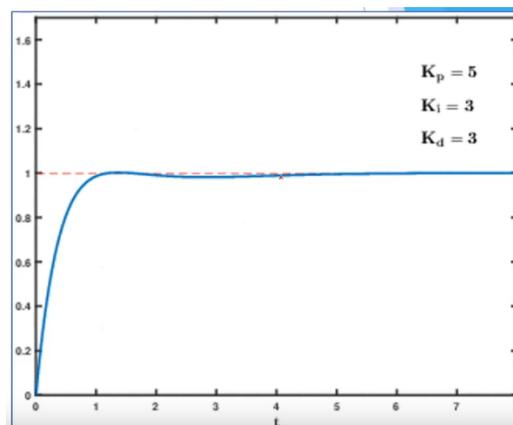
Gambar 2. 6 Memasukan Nilai  $K_p$

Kemudian masukan  $K_i$  sampai offset apa pun dikoreksi dalam waktu yang cukup untuk proses, namun terlalu banyak  $K_i$  akan menyebabkan ketidakstabilan hasil grafik dari memberi nilai  $K_i$  dapat dilihat pada gambar 2.7.



Gambar 2. 7 Memasukan Nilai  $K_i$

Masukan nilai  $K_d$  jika diperlukan, sampai loop dapat diterima dengan cepat untuk mencapai referensinya setelah gangguan, namun terlalu banyak  $K_d$  akan menyebabkan respon yang berlebihan dan overshoot, hasil grafik dari memberi nilai  $K_d$  dapat dilihat pada gambar 2.8.



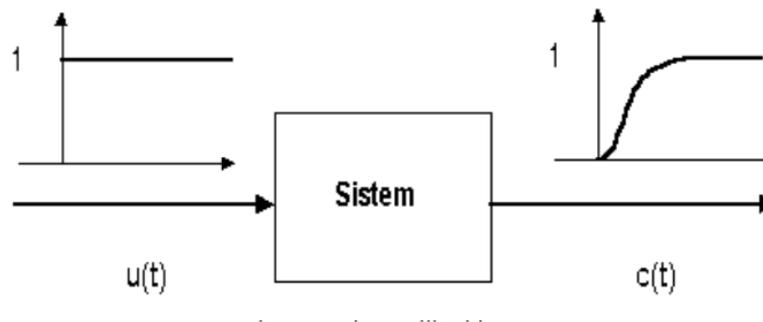
Gambar 2. 8 Memasukan Nilai  $K_d$

### 2.3.4.2 Metode Ziegler-Nichlos

Salah satu metoda mencari parameter-parameter PID adalah dengan metoda Ziegler-Nichols. Metoda ini memiliki 2 tipe, yaitu tipe 1 (*open loop*) dan tipe 2 (*closed loop*) seperti pada gambar 2.9.

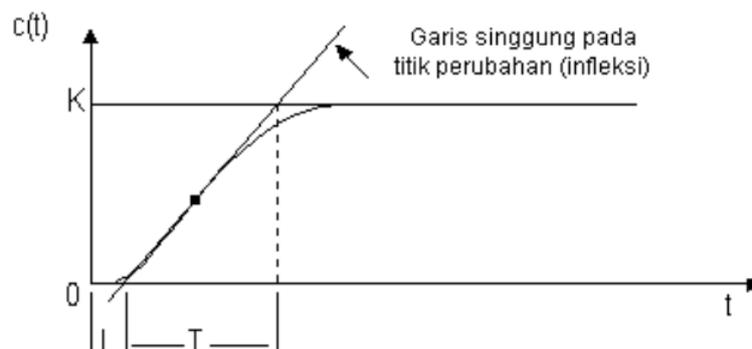
#### 1. Ziegler-Nichols tipe 1 (*open loop*)

Dalam metoda ini, digunakan sistem *open loop*. Sistem diberi input *step* sehingga respon *open loop* terbentuk.



Gambar 2. 9 Sistem Diberi Input Step

Dari respon *open loop* yang dihasilkan, parameter-parameter ZN tipe 1 (L dan T) dapat didapatkan. Proses desain menentukan parameter L dan T ditunjukkan pada gambar 2.10.



Gambar 2. 10 Proses Desain Penentuan Parameter L dan T

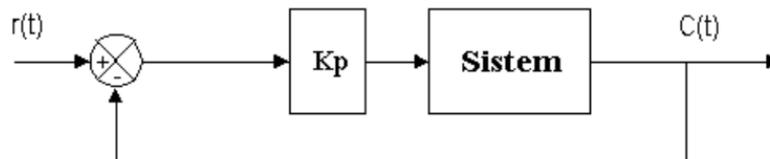
Setelah parameter  $L$  dan  $T$  didapatkan, nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  bisa dicari dengan menggunakan rumus-rumus parameter PID untuk metoda ZN tipe 1. Tabel 2.2 menunjukkan tabel parameter PID untuk ZN tipe 1.

Tabel 2. 2 Tabel Parameter PID Untuk ZN Tipe 1

Tipe kontrol	$K_p$	$T_i$	$T_d$
P	$\frac{T}{L}$	$\infty$	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

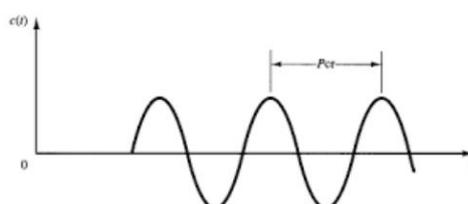
## 2. Ziegler-Nichols tipe 2 (*close loop*)

Pada gambar 2.11 Dalam metoda ZN tipe 2, digunakan sistem closed loop. Namun yang digunakan hanya  $K_p$  saja. Sistem dibuat hingga berosilasi terus menerus dengan mengatur besarnya nilai  $K_p$ .



Gambar 2. 11 Sistem closed loop dengan menggunakan  $K_p$  saja

Besarnya nilai  $K_p$  saat respon sistem berosilasi terus menerus merupakan nilai  $K_{cr}$ . Dari respon yang dihasilkan, parameter lain ZN tipe 2 selain  $K_{cr}$ , yaitu  $P_{cr}$  dapat dicari. Proses desain menentukan parameter  $P_{cr}$  ditunjukkan pada gambar 2.12.



Gambar 2. 12 Proses Desain Menentukan Parameter Pcr

Setelah parameter  $K_{cr}$  dan  $P_{cr}$  didapatkan, nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  bisa dihitung dengan menggunakan rumus-rumus parameter PID untuk ZN tipe 2. Tabel 2.3 menunjukkan tabel parameter PID untuk ZN tipe 2.

Tabel 2. 3 Tabel Parameter PID Untuk ZN Tipe 2

Tipe kontrol	$K_p$	$T_i$	$T_d$
P	$0.5K_{cr}$	$\infty$	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

#### 2.3.4.3 Metoda Cohen-Coon

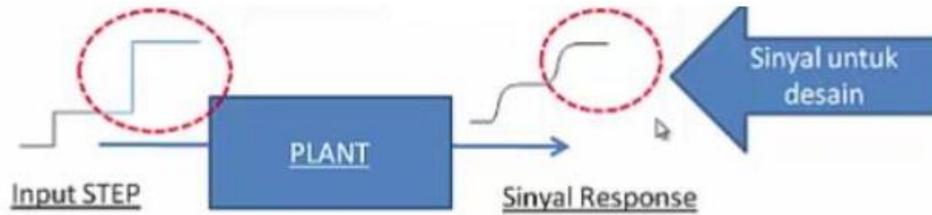
Pada gambar 2.13 dalam menentukan parameter-parameter PID tidak hanya dengan metoda Ziegler-Nichols saja. Ada metoda lain untuk menentukan parameter tersebut. Metoda tersebut adalah metoda Cohen-Coon. CohenCoon biasanya hanya digunakan untuk sistem *open loop*. Cohen-Coon bisa dipakai untuk plant yang memiliki *deadtime* yang besar.

Sistem *open loop* diberi input step sampai respon mencapai titik *steady-state*.



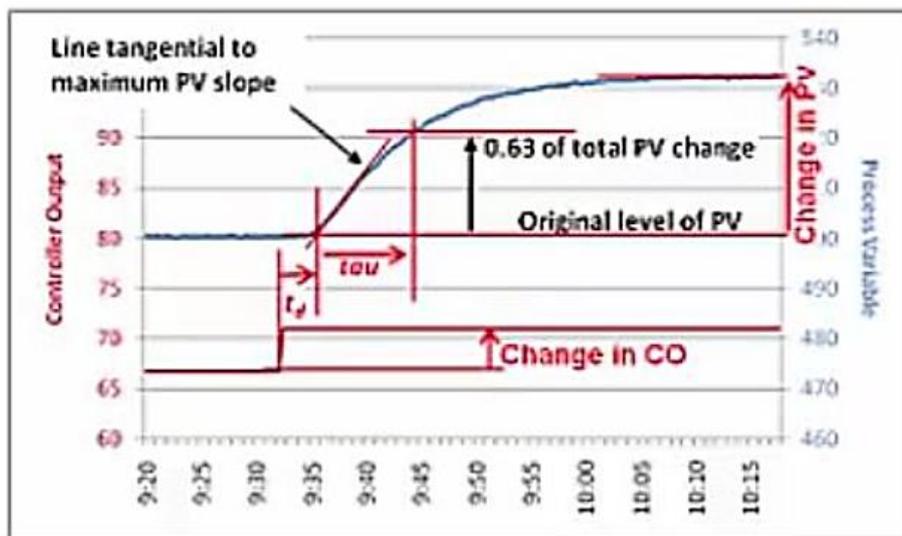
Gambar 2. 13 Sistem Open Loop Diberi Input Step

Setelah respon mencapai steady-state, respon kembali diberi input step. Sinyal perubahan yang dihasilkan adalah sinyal yang akan dipakai untuk mendesain.



Gambar 2. 14 Sinyal Steady-state Kembali Diberi Input Step

Pada gambar 2.14 dalam proses penentuan parameter PID dengan metoda Cohen-Coon, perubahan sinyal saat diberi input step yang kedua merupakan sinyal yang digunakan untuk proses desain. Gambar 2.15 menunjukkan proses penentuan parameter-parameter  $K_p$ ,  $T_i$ , dan  $T_d$ .



Gambar 2. 15 Proses Desain Menentukan Parameter  $k_p$ ,  $t_i$ , dan  $t_d$

Dimana  $K_p$  dicari dengan menggunakan rumus,

$$K_p = \frac{\text{Change in PV}}{\text{Change in CO}}$$

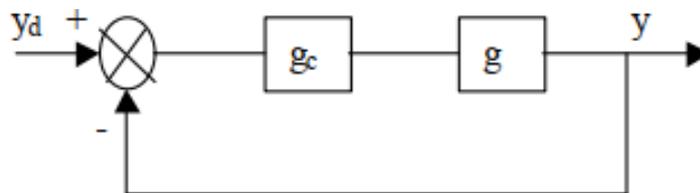
Setelah parameter-parameter  $K_p$ ,  $T_i$ , dan  $T_d$  didapatkan, nilai-nilai  $K_p$ ,  $T_i$ , dan  $T_d$  bisa dihitung dengan rumus-rumus parameter PID untuk CohenCoon. Tabel parameter PID untuk Cohen-Coon ditunjukkan pada tabel 2.4.

Tabel 2. 4 Tabel Parameter PID Untuk Cohen-Coon

<a href="http://www.opticontrols.com">www.opticontrols.com</a>	Controller Gain	Integral Time	Derivative Time
P Controller:	$K_c = \frac{1.03}{g_p} \left( \frac{\tau}{t_d} + 0.34 \right)$		
PI Controller:	$K_c = \frac{0.9}{g_p} \left( \frac{\tau}{t_d} + 0.092 \right)$	$T_i = 3.33 t_d \frac{\tau + 0.092 t_d}{\tau + 2.22 t_d}$	
PD Controller:	$K_c = \frac{1.24}{g_p} \left( \frac{\tau}{t_d} + 0.129 \right)$		$T_D = 0.27 t_d \frac{\tau - 0.324 t_d}{\tau + 0.129 t_d}$
PID Controller: (Noninteracting)	$K_c = \frac{1.35}{g_p} \left( \frac{\tau}{t_d} + 0.185 \right)$	$T_i = 2.5 t_d \frac{\tau + 0.185 t_d}{\tau + 0.611 t_d}$	$T_D = 0.37 t_d \frac{\tau}{\tau + 0.185 t_d}$

#### 2.3.4.4 Direct Synthesis

Metode ini terlebih dulu menentukan perilaku output yang diinginkan (*reference*) dengan membuat bentuk trayektorinya, dan model prosesnya (*plant*) digunakan untuk secara langsung mendapatkan persamaan kontroler yang sesuai dapat dilihat pada gambar 2.16.



Gambar 2. 16 Blok Diagram dari Sistem Kontrol Berumpan balik

#### 2.3.5 LCD

LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*.

LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. Material LCD (*Liquid Cristal Display*) adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang.

Pada gambar 2.17 dalam modul LCD (*Liquid Cristal Display*) terdapat *microcontroller* yang berfungsi sebagai pengendali tampilan karakter LCD. *Microntroller* pada suatu LCD dilengkapi dengan memori dan register. Memori yang digunakan *microcontroler* internal LCD adalah: DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM (*Character Generator Read Only Memory*).

Register *control* yang terdapat dalam suatu LCD (*Liquid Cristal Display*) diantaranya adalah register perintah yaitu register yang berisi perintah-perintah dari mikrokontroler ke panel LCD pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data. Register data yaitu register untuk menuliskan atau membaca data dari atau ke DDRAM (*Display Data Random Access Memory*). Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.



Gambar 2. 17 LCD (Angga Rusdinar, 2015)

### 2.3.6 *Bluetooth HC-05*

Pada Gambar 2.18 *Module bluetooth HC-05* adalah *module* komunikasi nirkabel via *bluetooth* yang dimana beroperasi pada frekuensi 2.4GHz dengan pilihan dua mode konektivitas. Mode 1 berperan sebagai *slave* atau *receiver* data saja, mode 2 berperan sebagai master atau dapat bertindak sebagai *transceiver*. Pengaplikasian komponen ini sangat cocok pada proyek elektronika dengan komunikasi nirkabel atau *wireless*. Aplikasi yang dimaksud antara lain aplikasi sistem kendali, *monitoring*, maupun gabungan keduanya.

Antarmuka yang dipergunakan untuk mengakses module ini yaitu serial TXD, RXD, VCC serta GND. Serta terdapat LED (built in) sebagai indikator koneksi *bluetooth* terhadap perangkat lainnya seperti sesama module, dengan smartphone android, dan sebagainya.

Jangkauan jarak efektif module ini saat terkoneksi dalam range 10 meter, dan jika melebihi dari range tersebut maka kualitas konektivitas akan semakin kurang maksimal



Gambar 2. 18 Bluetooth HC-05 (Asep Saefullah, dkk., 2015)

*Module* ini dapat digunakan sebagai mode *slave* (Rx), maupun mode master (TX) dan memiliki 2 metode konfigurasi yaitu AT Mode dan *Communication Mode*. Pada AT Mode berfungsi sebagai pengaturan konfigurasi dari HC-05, sedangkan pada *Communication Mode* berfungsi sebagai komunikasi nirkabel dengan perangkat atau piranti lainnya.

### 2.3.7 Sensor *Ultrasonic* HC-SR04

Pada Gambar 2.19 Sensor *Ultrasonic* adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor *ultrasonic* karena sensor ini menggunakan gelombang *ultrasonic* (bunyi *ultrasonic*). Gelombang *ultrasonic* adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi *ultrasonic* tidak dapat didengar oleh telinga manusia. Bunyi *ultrasonic* dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi *ultrasonic* bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi *ultrasonic* di permukaan zat padat hampir sama dengan reflektivitas bunyi

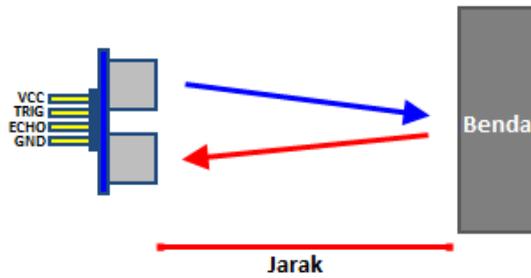
*ultrasonic* di permukaan zat cair. Akan tetapi, gelombang bunyi *ultrasonic* akan diserap oleh tekstil dan busa. Bentuk fisik dari sensor ini tampak seperti gambar berikut:



Gambar 2. 19 Sensor Ultrasonic HC-SR04 (Dwi Aji, 2017)

Berikut ini adalah beberapa aplikasi dari gelombang *ultrasonic*:

- Dalam bidang kesehatan, gelombang *ultrasonic* bisa digunakan untuk melihat organ-organ dalam tubuh manusia seperti untuk mendeteksi tumor, liver, otak dan menghancurkan batu ginjal. Gelombang *ultrasonic* juga dimanfaatkan pada alat USG (ultrasonografi) yang biasa digunakan oleh dokter kandungan.
- Dalam bidang industri, gelombang *ultrasonic* digunakan untuk mendeteksi keretakan pada logam, meratakan campuran besi dan timah, meratakan campuran susu agar homogen, mensterilkan makanan yang diawetkan dalam kaleng, dan membersihkan benda-benda yang sangat halus. Gelombang *ultrasonic* juga bisa digunakan untuk mendeteksi keberadaan mineral maupun minyak bumi yang tersimpan di dalam perut bumi.
- Dalam bidang pertahanan, gelombang *ultrasonic* digunakan sebagai radar atau navigasi, di darat maupun di dalam air. Gelombang *ultrasonic* digunakan oleh kapal pemburu untuk mengetahui keberadaan kapal selam, dipasang pada kapal selam untuk mengetahui keberadaan kapal yang berada di atas permukaan air, mengukur kedalaman palung laut, mendeteksi ranjau, dan menentukan posisi sekelompok ikan.



Gambar 2. 20 Cara kerja Sensor Ultrasonic HC-SR04 (Dwi Aji, 2017)

Pada gambar 2.20 cara kerja sensor *ultrasonic*, gelombang *ultrasonic* dibangkitkan melalui sebuah alat yang disebut dengan piezo elektrik dengan frekuensi tertentu. Piezo elektrik ini akan menghasilkan gelombang *ultrasonic* (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang *ultrasonic* menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.

Karena kecepatan bunyi adalah 340 m/s, maka rumus untuk mencari jarak berdasarkan *ultrasonic* adalah :

$$S = \left( \frac{340 \times t}{2} \right) \dots \dots \dots (2.2)$$

dimana S merupakan jarak antara sensor *ultrasonic* dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh *transmitter* dan waktu ketika gelombang pantul diterima *receiver*.

HC-SR04 merupakan sensor *ultrasonic* siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang *ultrasonic*. Alat ini bisa digunakan

untuk mengukur jarak benda dari 2cm - 4m dengan akurasi 3mm. Dengan demikian, untuk menghitung jarak yang hanya maksimal 4m maka rumus di atas harus dimodifikasi atau disesuaikan satuannya. Mikrokontroller bisa bekerja pada order mikrosekon ( $1s = 1.000.000 \mu s$ ) dan satuan jarak bisa kita ubah ke satuan cm ( $1m = 100$  cm). Oleh sebab itu, rumus di atas bisa diupdate menjadi:

$$S = \frac{\left(340 \left(\frac{100}{1000000}\right) \cdot t\right)}{2}$$

$$S = \frac{0.034 \cdot t}{2}$$

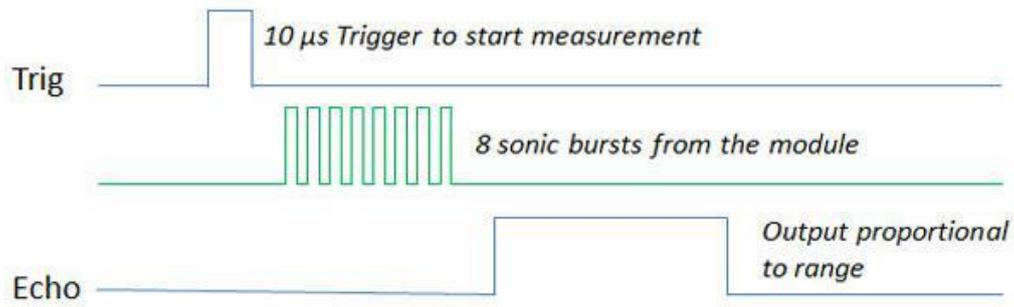
Alat ini memiliki 4 pin, pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk *ground*-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap sinyal pantul dari benda.

Cara kerja alat ini yaitu:

- Ketika kita memberikan tegangan positif pada pin Trigger selama  $10\mu s$ , maka sensor akan mengirimkan 8 step sinyal *Ultrasonic* dengan frekuensi 40kHz
- Selanjutnya, sinyal akan diterima pada pin Echo
- Untuk mengukur jarak benda yang memantulkan sinyal tersebut, selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut
- Rumus untuk menghitung jaraknya adalah  $S = (0.034 \cdot t) / 2$  cm.

Pada Gambar 2.21 merupakan hasil visualisasi dari sinyal yang dikirimkan oleh sensor

HC-SR04



Gambar 2. 21 Timing HC-SR04 (Dwi Aji, 2017)

### 2.3.8 Motor DC

Motor DC adalah salah satu jenis motor listrik yang banyak digunakan pada rangkaian elektronika, mulai dari robot *line following* sampai dengan *mobile robot* lainnya. Tak heran memang karena selain harganya murah, motor DC juga sangat mudah dikontrol dibandingkan dengan jenis motor listrik lainnya.

Motor DC adalah piranti elektronik yang mengubah energi listrik menjadi energi mekanik berupa gerak rotasi. Pada motor DC terdapat jangkar dengan satu atau lebih kumparan terpisah. Tiap kumparan berujung pada cincin belah (komutator). Dengan adanya insulator antara komutator, cincin belah dapat berperan sebagai saklar kutub ganda (*double pole, double throw switch*). Motor DC bekerja berdasarkan prinsip gaya Lorentz, yang menyatakan ketika sebuah konduktor beraliran arus diletakkan dalam medan magnet, maka sebuah gaya (yang dikenal dengan gaya Lorentz) akan tercipta secara ortogonal diantara arah medan magnet dan arah aliran arus.

Motor DC yang digunakan pada robot beroda umumnya adalah motor DC dengan magnet permanen. Motor DC jenis ini memiliki dua buah magnet permanen sehingga timbul medan magnet di antara kedua magnet tersebut. Di dalam medan magnet inilah jangkar/rotor berputar. Jangkar yang terletak di tengah motor memiliki jumlah kutub yang ganjil dan pada setiap kutubnya terdapat lilitan. Lilitan ini terhubung ke area kontak

yang disebut komutator. Sikat (*brushes*) yang terhubung ke kutub positif dan negatif motor memberikan daya ke lilitan sedemikian rupa sehingga kutub yang satu akan ditolak oleh magnet permanen yang berada di dekatnya, sedangkan lilitan lain akan ditarik ke magnet permanen yang lain sehingga menyebabkan jangkar berputar. Ketika jangkar berputar, komutator mengubah lilitan yang mendapat pengaruh polaritas medan magnet sehingga jangkar akan terus berputar selama kutub positif dan negatif motor diberi daya. Pengendalian kecepatan putar motor DC dapat dilakukan dengan mengatur kecepatan putaran. Pengguna hanya bisa mengatur putaran motor DC hanya dengan mengganti nilai input tegangannya saja

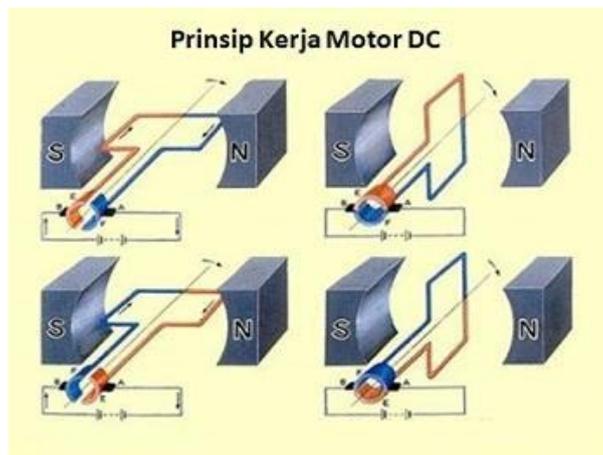


Gambar 2. 22 Motor DC (Hebi Jaya, 2015)

### 2.3.9 Prinsip kerja motor DC

Pada gambar 2.22 terdapat dua bagian utama pada sebuah motor listrik DC, yaitu *stator* dan *rotor*. *Stator* adalah bagian motor yang tidak berputar, bagian yang statis ini terdiri dari rangka dan kumparan medan. Sedangkan *rotor* adalah bagian yang berputar, bagian Rotor ini terdiri dari kumparan Jangkar. Dua bagian utama ini dapat dibagi lagi menjadi beberapa komponen penting yaitu diantaranya adalah *yoke* (kerangka magnet), *poles* (kutub motor), *field winding* (kumparan medan magnet), *armature winding* (kumparan jangkar), *commutator* (Komutator) dan *brushes* (kuas/sikat arang). Pada gambar 2.23 adalah prinsipnya motor listrik DC menggunakan fenomena

elektromagnet untuk bergerak, ketika arus listrik diberikan ke kumparan, permukaan kumparan yang bersifat utara akan bergerak menghadap ke magnet yang berkutub selatan dan kumparan yang bersifat selatan akan bergerak menghadap ke utara magnet. Saat ini, karena kutub utara kumparan bertemu dengan kutub selatan magnet ataupun kutub selatan kumparan bertemu dengan kutub utara magnet maka akan terjadi saling tarik menarik yang menyebabkan pergerakan kumparan berhenti.



Gambar 2. 23 Prinsip kerja motor DC (Risidian tata, 2013)

Untuk menggerakannya lagi, tepat pada saat kutub kumparan berhadapan dengan kutub magnet, arah arus pada kumparan dibalik. Dengan demikian, kutub utara kumparan akan berubah menjadi kutub selatan dan kutub selatannya akan berubah menjadi kutub utara. Pada saat perubahan kutub tersebut terjadi, kutub selatan kumparan akan berhadapan dengan kutub selatan magnet dan kutub utara kumparan akan berhadapan dengan kutub utara magnet. Karena kutubnya sama, maka akan terjadi tolak menolak sehingga kumparan bergerak memutar hingga utara kumparan berhadapan dengan selatan magnet dan selatan kumparan berhadapan dengan utara magnet. Pada saat ini, arus yang mengalir ke kumparan dibalik lagi dan kumparan akan berputar lagi karena adanya perubahan kutub. Siklus ini akan berulang-ulang hingga arus listrik pada kumparan diputuskan

### 2.3.10 Driver Motor L298

*Driver* motor L298 adalah komponen elektronik yang dipergunakan untuk mengontrol arah putaran motor DC. Satu buah L298 bisa dipergunakan untuk mengontrol dua buah motor DC. Selain bisa dipergunakan untuk mengontrol arah putaran motor DC, L298 ini pun bisa dipergunakan sebagai *driver motor* Stepper bipolar. IC *driver* L298 memiliki kemampuan menggerakkan motor DC sampai arus 2A dan tegangan maksimum 40volt DC untuk satu kanalnya. Pin *enable* A dan B untuk mengendalikan jalan atau kecepatan motor, pin input 1 sampai 4 digunakan untuk mengendalikan arah putaran. Pin output pada IC L298 13 dihubungkan ke motor DC yang sebelumnya melalui dioda yang disusun secara *H-bridge*. Pengaturan kecepatan motor digunakan teknik PWM (*pulse width modulation*) yang diinputkan dari mikrokontroler melalui pin *Enable*. PWM untuk kecepatan rotasi yang bervariasi level highnya.

Pada gambar 2.24 *Driver* motor L298N merupakan driver motor yang paling populer digunakan untuk mengontrol kecepatan dan arah pergerakan motor terutama pada robot *Line Follower / Line Tracer*. Kelebihan dari *driver* motor L298N ini adalah cukup presisi dalam mengontrol motor. Selain itu kelebihan *driver* motor L298N adalah mudah untuk dikontrol. Untuk mengontrol *driver* motor L298N ini dibutuhkan 6 buah Pin mikrokontroler dua buah untuk Pin *Enable* (satu buah untuk motor A dan satu buah yang lain untuk motor B karena *driver* motor L298N ini dapat mengontrol dua buah motor dc) 4 buah untuk mengatur kecepatan motor tersebut. Pada prinsipnya rangkaian *driver* motor L298N ini dapat mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat diatur (Yosua D. Widiarto, 2018)



Gambar 2. 24 Driver motor L298N (Yosua D. Widiarto, 2018)

### 2.3.11 Teknik Pengendalian Motor

Dasar teknik pengendalian motor DC sebagai berikut:

#### 1. Pengaturan Medan

Pengaturan ini dapat dilakukan dengan mengatur arus medan shunt dengan melemahkan dan menaikkan melalui pengaturan tahanan variabel yang dihubungkan seri dengan kumparan medan. Pengaturan kecepatan dengan cara ini mempunyai karakteristik sebagai berikut:

- kecepatan terendah didapat dengan membuat tahanan variabel sama dengan nol. Sedangkan kecepatan tertinggi akan dibatasi oleh perencanaan mesin dimana gaya sentrifugal maksimum tidak sampai merusak motor.
- Rugi panas yang timbul sangat rendah
- Sederhana dan mudah dalam perangkaiannya
- Pengaturan kecepatan seperti ini hanya bisa dilakukan pada motor DC *shunt* dan kompon

#### 2. Pengaturan Tahanan Jangkar

Pengaturan ini dapat dilakukan dengan mengatur tahanan jangkar dengan menghubungkan seri dengan tahanan variable. Dengan menyisipkan tahanan variabel secara seri terhadap tahanan jangkar, maka nilai  $I_a$   $R_a$  akan dapat dikontrol sehingga

nantinya harga  $n$  dapat dikontrol Cara ini jarang digunakan karena rugi panas yang timbul cukup besar

### 3. Pengaturan Tegangan

Pengaturan ini dilakukan dengan mengatur tegangan yang disuplai ke motor, dengan persamaan sebagai berikut.

$$V_t = E_a + I_a \cdot R_a$$

$$E_a = V_t - I_a \cdot R_a$$

$$E_a = c \cdot n \cdot \Phi_{sh}$$

$$n = \frac{V_t - I_a \cdot R_a}{c \cdot \Phi_{sh}}$$

dimana :

$n$  = Putaran kecepatan motor (rpm).

$V_t$  = Tegangan terminal (Volt).

$I_a$  = Arus jangkar (Ampere).

$c$  = Konstanta.

$\Phi_{sh}$  = Fluks medan shunt (Wb)

$I_{sh}$  = Arus shunt (Ampere).

$R_a$  = Hambatan jangkar (ohm)

#### 2.3.12 *Pulse Width Modulation (PWM)*

Metode *pulse width modulation* (PWM) adalah metode yang cukup efektif untuk mengendalikan kecepatan motor DC. PWM ini bekerja dengan membuat gelombang persegi yang memiliki frekuensi tetap, rasio (*duty cycle*) pulsa tinggi terhadap pulsa rendah yang telah ditentukan, yang bisa diatur dari 0% hingga 100%. Rasio pulsa tinggi terhadap pulsa rendah digunakan untuk mengatur penyambungan (*switch*) suplay tegangan terminal motor DC. Semakin besar lebar pulsa tinggi dari pada pulsa rendah

berarti penyambungan *suplay* tegangan terminal motor mendapat porsi waktu yang lebih lama, menyebabkan kecepatan motor semakin besar.

### **2.3.13 Modulasi lebar pulsa (*Pulse Width Modulation*)**

Modulasi lebar pulsa (*Pulse Width Modulation*) merupakan suatu teknik untuk mengendalikan tegangan input rangkaian analog dengan memanfaatkan sinyal output digital (Pulsa-pulsa tegangan digital). Pengaturan tegangan output dapat dilakukan dengan mengubah-ubah nilai *duty cycle*, dimana *duty cycle* merupakan presentase lebar pulsa pada kondisi aktif (level tinggi) per periode. Besar tegangan yang dihasilkan merupakan tegangan rata-rata, hasil perkalian nilai *duty cycle* dengan tegangan maksimum sumber.