

BAB II

TINJAUAN PUSTAKA

Dough divider merupakan alat pembagi adonan roti yang pada prinsipnya bekerja seperti pembuatan bakso. Adonan roti yang sudah kalis dimasukkan dalam *screw conveyor* melalui lubang tempat masuknya bahan yang disebut *feeding hopper*. *Screw conveyor* berfungsi sebagai pendorong adonan melalui *screw* yang diputar oleh sebuah motor. Adonan yang keluar dari *screw conveyor* akan menyentuh sensor kapasitif. Sensor kapasitif ini berfungsi sebagai penentu besar kecilnya adonan yang diatur melalui jarak antara mulut keluarnya adonan dan sensor kapasitif. Pada saat adonan menyentuh sensor kapasitif, adonan dipotong oleh pisau yang digerakkan oleh sebuah motor DC. Adonan yang sudah dipotong ini jatuh pada *belt conveyor*. Namun seperti yang dijelaskan dalam batasan masalah, penulis hanya membahas tentang *screw conveyor*. Dalam merencanakan dan merealisasikan sebuah *screw conveyor*, maka dibutuhkan pemahaman tentang berbagai hal yang mendukung sistem ini. Pemahaman ini akan bermanfaat untuk merancang perangkat keras dan perangkat lunak sistem. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi sensor *rotary encoder*, mikrokontroler ATmega8535, motor DC serta kontroler PID.

2.1 Sekilas Tentang Roti

Roti awal masuk di Indonesia sekitar abad ke-16 dibawa oleh para pedagang Belanda. Roti yang tadinya dianggap sebagai makanan bangsawan bagi para bangsawan Belanda dijamin pen jajahan, kini sudah menjadi makanan pokok kedua setelah nasi. Kandungan gizi produk olahan dari tepung ini lebih unggul dibandingkan dengan mie instan. Bahkan ada jenis roti yang selain kaya serat, mengandung omega 3 yang berfungsi sebagai penangkal berbagai penyakit degenerative (penurunan fungsi organ tubuh yang umumnya terjadi pada usia tua) (Chendhawati, 2008: 9).

Dengan beragamnya jenis roti, kandungan gizinya pun sangat beragam. Kandungan gizi roti sangat ditentukan oleh bahan penyusun adonan dan cara pembuatan roti. Bahan baku roti dapat dibagi menjadi dua kategori, yaitu bahan utama dan bahan tambahan. Bahan baku utama berupa tepung terigu, ragi, garam, dan air. Bahan baku tambahan berupa gula pasir, shortening, telur, dan susu bubuk (Chendhawati, 2008: 9).

Pada prinsipnya roti dibuat dengan cara mencampurkan tepung dan bahan penyusun lainnya menjadi adonan kemudian difermentasikan dan dipanggang. Pembuatan roti dapat dibagi menjadi dua bagian utama, yaitu proses pembuatan adonan dan proses pembakaran. Kedua proses ini akan menentukan mutu hasil akhir. Pembuatan adonan meliputi proses pengadukan bahan dan pengembangan adonan sampai proses fermentasinya. Urutan proses pembuatan roti secara umum ditunjukkan dalam Gambar 2.1.

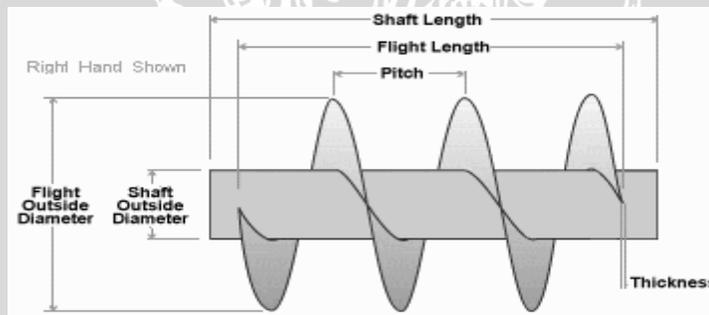


Gambar 2.1 Diagram Alir Proses Pembuatan Roti (UKM MR Bakery)

2.2 Screw Conveyor

Tujuan *screw conveyor* adalah untuk mendorong bahan umpan sepanjang tabung. Dorongan ini diakibatkan pengecilan jarak antar ulir sehingga menyebabkan volume antar ulir mengecil. Penggunaan *screw conveyor* tidak terbatas pada pengangkut material dalam arah horizontal, tetapi *screw conveyor* juga dapat digunakan untuk mengangkut material dengan sudut tertentu. Bahkan juga dapat digunakan untuk mengangkut atau memindahkan material dalam arah vertical.

Screw conveyor (Gambar 2.2) biasanya terdiri atas poros yang terpasang *screw* yang berputar dalam *trough* dan unit penggerak. Pada saat *screw* berputar, material dimasukkan melalui *feeding hopper* ke *screw* yang bergerak maju akibat daya dorong (*thrust*) *screw*. Poros dan *screw* berputar sepanjang lintasan *casing* berbentuk U (*U-shaped*) atau O (*O-shaped*). Material yang dipindahkan diisikan ke dalam *trough* oleh satu atau lebih cawan pengisi (*feed hopper*). Bahan dikeluarkan pada ujung *trough* atau bukaan bawah *trough*. Gambar 2.2 menunjukkan susunan dari *screw conveyor*.



Gambar 2.2 Susunan *Screw Conveyor* (U.S. Air Filtration Systems Engineer)

2.3. Motor DC

2.3.1. Prinsip Kerja Motor DC

Segulung kawat yang dialiri arus listrik dan ditempatkan di dalam suatu medan magnet akan mengalami gaya yang sebanding dengan arus dan kekuatan medan magnetnya. Gaya yang ditimbulkan disebut dengan Gaya *Lorentz* (Sukis Wariyono, 2008) yang dapat dirumuskan sebagai berikut:

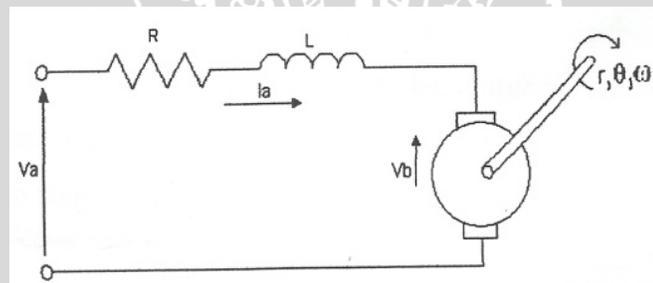
$$F = B I L \quad (2.1)$$

Dalam hal ini B adalah kerapatan *fluks* magnet (tesla), I adalah arus yang mengalir (ampere) dan L adalah panjang kawat (m).

Persamaan 2.1 merupakan prinsip dasar sebuah motor, dimana terjadinya proses perubahan arus listrik (I) akan menimbulkan gaya mekanik (F dalam satuan *newton*). Jika motor mempunyai jari-jari sebesar r (m), maka akan menimbulkan torsi Γ (Nm : *newton meter*) sebesar :

$$\Gamma = F.r = B.I.L.r \quad (2.2)$$

Pada saat dibangkitkan, konduktor akan bergerak di dalam medan magnet dan akan menimbulkan gaya gerak listrik (ggl) yang merupakan reaksi lawan terhadap tegangan penyebabnya. Proses konversi energi listrik menjadi energi mekanik dapat berlangsung jika tegangan sumber lebih besar dari gaya gerak listrik lawan.



Gambar 2.3 Rangkaian Dasar Motor DC (Pitowarno, E. 2006)

Motor dapat berputar jika tegangan masukan motor lebih besar dari ggl yang timbul. Hubungan antara tegangan sumber dan ggl lawan seperti ditunjukkan dalam Gambar 2.3, dapat dirumuskan sebagai berikut:

$$E_a = V_{in} - I_a.R_a \quad (2.3)$$

Dalam hal ini E_a adalah tegangan pada jangkar (V), V_{in} adalah tegangan masukan (volt), I_a adalah arus jangkar (ampere) dan R_a adalah tahanan jangkar (ohm), sedangkan induksi yang timbul adalah:

$$E_a = C n \Phi \quad (2.4)$$

Dengan C adalah konstanta mesin, n adalah kecepatan putaran motor (rpm), dan Φ adalah *fluks* magnetik (wb) yang besarnya sebanding dengan arus penguatan torsi.

Torsi pada motor juga sebanding dengan *fluks* magnetik dan arus. Hal ini ditunjukkan pada Persamaan 2.5 berikut:

$$\Gamma = C \Phi I_a \quad (2.5)$$

Dimana

Γ : torsi dengan satuan newton meter (Nm)

C : konstanta mesin

Φ : fluks yang dihasilkan oleh kumparan medan (wb)

I_a : arus jangkar dengan satuan ampere (A)

Jika diketahui kecepatan sudut ω adalah:

$$\omega = 2\pi \frac{n}{60} \quad (2.6)$$

Dimana

ω : kecepatan sudut

π : konstanta (3,14)

n : kecepatan putaran motor (rpm)

Maka hubungan torsi dan kecepatan motor adalah:

$$\Gamma = \frac{P}{\omega}$$

$$\Gamma = \frac{P}{2\pi \frac{n}{60}} \quad (2.7)$$

Dengan P adalah daya motor.

2.3.2. Pengaturan Motor DC

Apabila Persamaan 2.3 disubstitusikan dengan Persamaan 2.4 akan didapatkan rumus kecepatan motor (n) sebagai berikut:

$$n = \frac{V_{in} - I_a R_a}{C\Phi} \quad (2.8)$$

Dimana

V_{in} = Tegangan jangkar dalam satuan volt (V)

I_a = Arus Jangkar dalam satuan ampere (A)

R_a = Tahanan jangkar dalam satuan ohm (Ω)

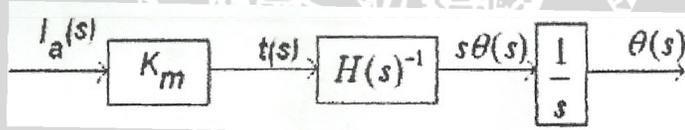
n = Putaran (RPM)

Φ = Fluks / kutub

C = Konstanta

Dari persamaan 2.8, dapat dilihat bahwa kecepatan putaran (n) motor DC dapat diatur dengan mengubah-ubah besarnya V_{in} (tegangan jangkar), R_a (Tahanan Jangkar), dan Φ (fluks magnet) (Azzahratunnisa, 2009). Jadi apabila tegangan masukan besar maka kecepatan motor akan cepat, demikian pula sebaliknya, jika tegangan masukan kecil maka kecepatan motor akan lambat.

Jika motor DC dianggap linier, yaitu torsi berbanding lurus dengan arus (motor ideal), maka model matematik dapat disederhanakan dengan memperhatikan proporsional konstanta motor (K_m) ditunjukkan dalam Gambar 2.4 dengan asumsi bahwa motor DC adalah dari jenis motor magnet permanen dan masukanya dinyatakan sebagai arus maka fungsi alih loop terbuka (*transfer function open loop*).



Gambar 2.4 Fungsi Alih Motor DC (Pitowarno, E. 2006)

2.3 Mikrokontroler ATmega8535

Secara umum, mikrokontroler berfungsi sama dengan komputer. Bedanya adalah mikrokontroler memiliki desain dalam sebuah *single chip* (IC). Mikrokontroler terdapat di hampir semua peralatan elektronik di sekeliling kita, didalam tape, TV, radio, telepon genggam (*Hand Phone*). Mikrokontroler memiliki kemampuan yang diperlukan untuk membuat keputusan berdasarkan sinyal dari luar dengan kata lain mikrokontroler merupakan otak dari sebuah perangkat elektronik (Widodo Budiharto dan Togu Jefri, 2003)

ATmega8535 merupakan salah satu mikrokontroler buatan ATMEL keluarga ATmega yang mempunyai 8 kbyte Flash PEROM (*Flash Programmable and Erasable Read Only Memory*), 512 byte SRAM, 32 pin I/O (4 buah port I/O bit) yang

mana tiap pin tersebut dapat diprogram secara paralel dan tersendiri, mempunyai dua buah *timer/counter* 8 bit dan satu buah *timer/counter* 16 bit, mempunyai 10 bit 8 channel ADC.

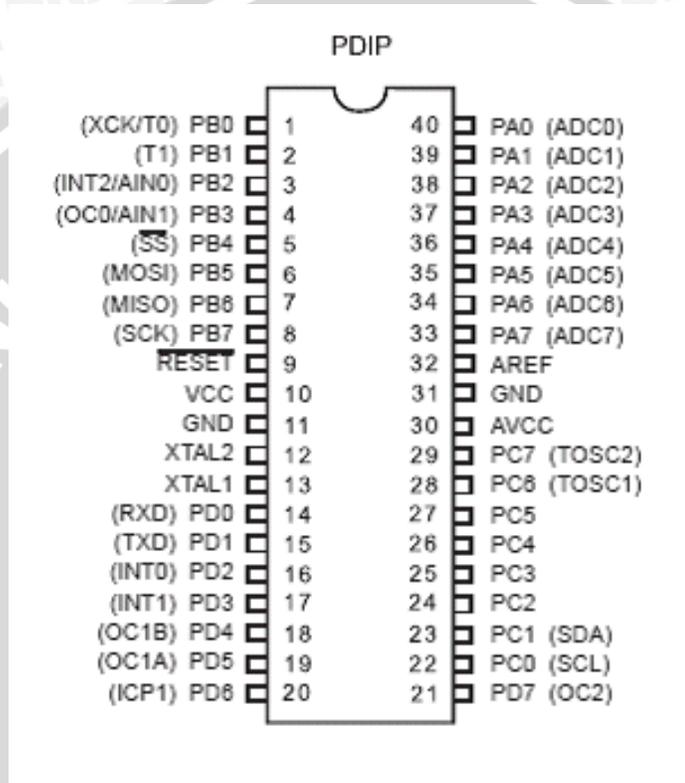
Pada dasarnya mikrokontroler adalah terdiri atas mikroprosesor, *timer*, dan *counter*, perangkat I/O dan internal memori. Mikrokontroler termasuk perangkat yang sudah didesain dalam bentuk chip tunggal. Mikrokontroler dikemas dalam satu chip (*single chip*). Mikrokontroler didesain dengan instruksi-instruksi lebih luas dan 8 bit instruksi yang digunakan membaca data instruksi dari internal memori ke ALU.

Sebagai suatu sistem kontrol mikrokontroler ATmega8535 bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor didalamnya tidak terdapat keduanya. Secara umum konfigurasi yang dimiliki mikrokontroler ATmega8535 adalah sebagai berikut :

- Sebuah CPU 8 bit dengan menggunakan teknologi dari Atmel.
- Memiliki memori baca-tulis sebesar 512 byte SRAM.
- Jalur dua arah (*bidirectional*) yang digunakan sebagai saluran masukan atau keluaran yang dikontrol oleh register DDR.
- Sebuah komunikasi serial USART yang dapat diprogram.
- Sebuah master/slave serial SPI yang dapat diprogram.
- Sebuah *Two Wire Serial Interface*.
- Dua buah *timer/counter* 8 bit dan sebuah *timer/counter* 16 bit.
- *Watchdog Timer* yang dapat diprogram.
- *Analog comparator* di dalam chip.
- Osilator internal dan rangkaian pewaktu.
- Flash PEROM yang besarnya 8 kbyte untuk memori program
- Kemampuan melaksanakan operasi perkalian, pembagian, dan operasi Boolean.
- Mampu beroperasi sampai 16 MHz.

ATmega8535 mikrokontroler mempunyai kompatibilitas instruksi dan konfigurasi pin dengan mikrokontroler ATmega.

Masing-masing kaki dalam mikrokontroler ATmega8535 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler ATmega8535, perancangan aplikasi mikrokontroler ATmega8535 akan lebih mudah. ATmega8535 mempunyai 40 pin, susunan masing-masing pin ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Konfigurasi Pin ATmega8535 (Datasheet ATmega8535)

Fungsi kaki-kaki ATmega8535 adalah :

- *Port A* (Pin A0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus *Port A* adalah sebagai ADC (input ADC channel 0..7).
- *Port B* (Pin B0..7), merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus *Port B* diantaranya adalah : *Port B0* (T0 (*timer/counter0* eksternal *counter input*) & XCK (USART eksternal *clock input/output*), *Port B1* (T1 (*timer/counter* eksternal *counter input*)), *Port B2* (AIN0 (*Analog comparator positive input*) & INT2 (Eksternal interrupt 2

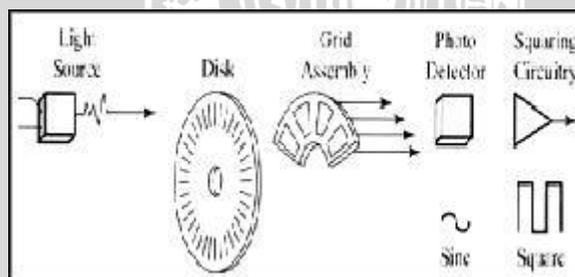
input)), Port B3 (AIN0 (*Analog comparator negative input*) & (OC0 (*Timer/counter0 output compare match output*)), Port B4 (SS (*SPI slave select input*)), Port B5 (MOSI (*SPI bus master output/slave input*)), Port B6 (MISO (*SPI bus master input/slave output*)), Port B7 (SCK (*SPI bus serial clock*)).

- *Port C (Pin C0..7)*, merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus dari Port C diantaranya adalah : Port C0 (*SCL (Two-Wire serial bus clock line)*), Port C1 (*SDA (Two-Wire serial bus data input/output line)*), Port C6 (*TOSC1 (Timer Oscillator pin1)*), Port C7 (*TOSC2 (Timer oscillator pin2)*).
- *Port D (Pin D0..7)*, merupakan saluran masukan/keluaran dua arah dan juga mempunyai fungsi khusus. Fungsi khusus dari Port D diantaranya adalah : Port D0 (*RXD (USART input pin)*), Port D1 (*TXD (USART output pin)*), Port D2 (*INT0 (Eksternal interrupt 0 input)*), Port D3 (*INT1 (Eksternal interrupt 1 input)*), Port D4 (*OC1B (Timer/counter 1 output compare B match output)*), Port D5 (*OC1A (Timer/counter 1 output compare A match output)*), Port D6 (*ICP (Timer/counter input capture pin)*), Port D7 (*OC2 (timer/counter 2 compare match output)*).
- Pin 9 RESET, merupakan saluran dua masukan untuk mereset mikrokontroler dengan cara memberi masukan logika rendah.
- Pin 10 VCC, merupakan saluran masukan untuk catu daya positif sebesar 5 volt DC.
- Pin 11 GND, merupakan Ground dari seluruh rangkaian.
- Pin 12 dan 13(XTAL2 dan XTAL1), merupakan saluran untuk mengatur pewaktuan sistem. Untuk pewaktuan dapat menggunakan pewaktuan internal maupun eksternal.
- Pin 32 AREF, merupakan Pin analog referensi untuk masukan ADC.
- Pin 33 GND, merupakan ground dari ADC.
- Pin 30 AVCC, merupakan catu untuk port A dan juga merupakan catu untuk ADC.

2.4 Rotary Encoder

Rotary encoder adalah divais elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. *Rotary encoder* umumnya digunakan pada pengendalian robot, motor drive, dsb.

Rotary encoder tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau divais berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Gambar 2.6 menunjukkan bagan skematik sederhana dari *rotary encoder*. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary encoder* tersebut.



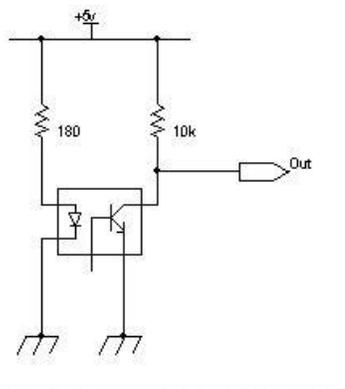
Gambar 2.6 Blok Penyusun *Rotary Encoder*

Rangkaian penghasil pulsa (Gambar 2.7) yang digunakan umumnya memiliki output yang berubah dari +5V menjadi 0.5V ketika cahaya diblok oleh piringan dan ketika diteruskan ke photo-transistor. Karena divais ini umumnya bekerja dekat

dengan motor DC maka banyak noise yang timbul sehingga biasanya output akan dimasukkan ke low-pass filter dahulu. Apabila low-pass filter digunakan, frekuensi cut-off yang dipakai umumnya ditentukan oleh jumlah slot yang ada pada piringan dan seberapa cepat piringan tersebut berputar, dinyatakan dengan:

$$f_c = \frac{s_w n}{60} \quad (2.9)$$

Dimana f_c adalah frekuensi cut-off filter, s_w adalah kecepatan piringan dan n adalah jumlah slot pada piringan.



Gambar 2.7 Rangkaian Tipikal Penghasil Pulsa pada Rotary Encoder

Terdapat dua jenis *rotary encoder* yang digunakan, *Absolute rotary encoder* dan *incremental rotary encoder*. Masing-masing *rotary encoder* ini akan dipaparkan pada bagian berikutnya.

2.4.1 Absolute Rotary Encoder

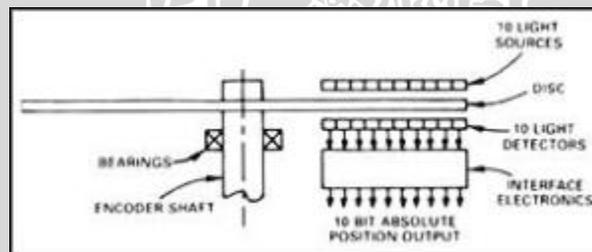
Absolute encoder menggunakan piringan dan sinyal optik yang diatur sedemikian sehingga dapat menghasilkan kode digital untuk menyatakan sejumlah posisi tertentu dari poros yang dihubungkan padanya. Piringan yang digunakan untuk absolut encoder tersusun dari segmen-segmen cincin konsentris yang dimulai dari bagian tengah piringan ke arah tepi luar piringan yang jumlah segmennya selalu dua kali jumlah segmen cincin sebelumnya. Cincin pertama di bagian paling dalam memiliki satu segmen transparan dan satu segmen gelap, cincin kedua memiliki dua segmen transparan dan dua segmen gelap, dan seterusnya hingga cincin terluar.

Sebagai contoh apabila absolut encoder memiliki 16 cincin konsentris maka cincin terluarnya akan memiliki 32767 segmen. Gambar 2.8 menunjukkan pola cincin pada piringan absolut encoder yang memiliki 16 cincin.



Gambar 2.8 Contoh Susunan Pola 16 Cincin Konsentris pada Absolut Encoder

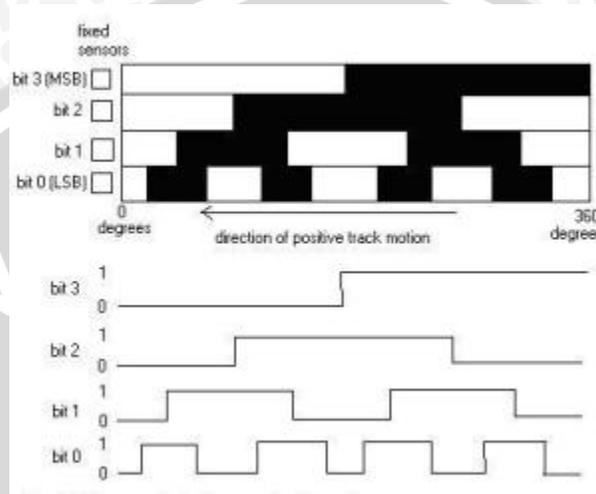
Karena setiap cincin pada piringan absolute encoder memiliki jumlah segmen kelipatan dua dari cincin sebelumnya, maka susunan ini akan membentuk suatu sistem biner. Untuk menghasilkan sistem biner pada susunan cincin maka diperlukan pasangan LED dan photo-transistor sebanyak jumlah cincin yang ada pada absolut encoder tersebut.



Gambar 2.9 Contoh Piringan dengan 10 Cincin dan 10 LED – Photo-Transistor untuk Membentuk Sistem Biner 10 bit.

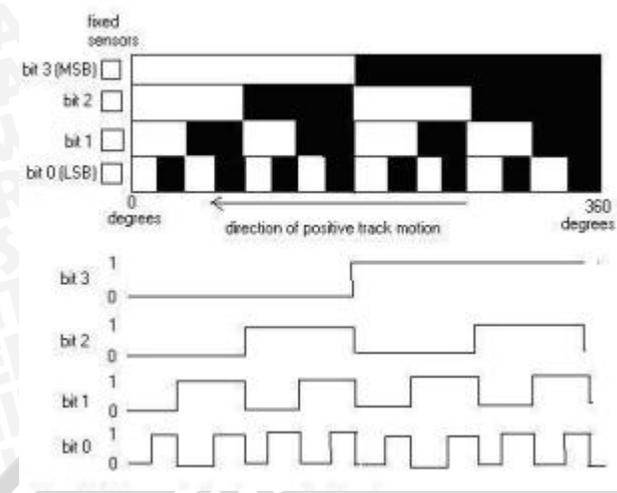
Gambar 2.9 merupakan contoh dari pembentukan sistem biner pada piringan 10 cincin. Sistem biner yang untuk menginterpretasi posisi yang diberikan oleh absolute encoder dapat menggunakan kode gray atau kode biner biasa, tergantung dari pola cincin yang digunakan. Untuk lebih jelas, kita lihat contoh absolut encoder yang hanya tersusun dari 4 buah cincin untuk membentuk kode 4 bit. Apabila encoder

ini dihubungkan pada poros, maka photo-transistor akan mengeluarkan sinyal persegi sesuai dengan susunan cincin yang digunakan. Gambar 2.10 dan Gambar 2.11 menunjukkan contoh perbedaan diagram keluaran untuk absolute encoder tipe gray code dan tipe binary code.



Gambar 2.10 Contoh Diagram Keluaran Absolut Encoder 4-bit Tipe Gray Code

Dengan absolute encoder 4-bit ini maka kita akan mendapatkan 16 informasi posisi yang berbeda yang masing-masing dinyatakan dengan kode biner atau kode gray tertentu. Tabel 2.1 menyatakan posisi dan output biner yang bersesuaian untuk absolute encoder 4-bit. Dengan membaca output biner yang dihasilkan maka posisi dari poros yang kita ukur dapat kita ketahui untuk diteruskan ke rangkaian pengendali. Semakin banyak bit yang kita pakai maka posisi yang dapat kita peroleh akan semakin banyak.



Gambar 2.11 Contoh Diagram Keluaran Absolut Encoder 4-bit Tipe Binary Code

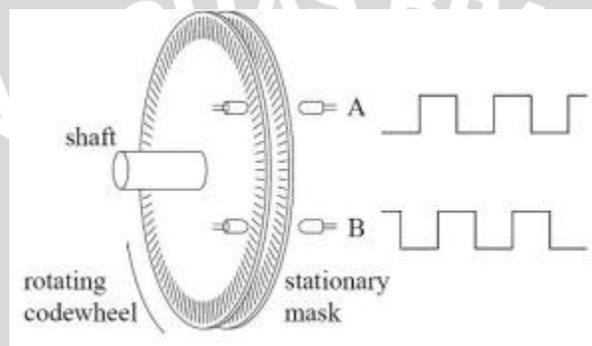
Tabel 2.1 Posisi dan Output Biner yang Bersesuaian pada Absolute Encoder 4-bit

DESIMAL	RENTANG PUTARAN	KODE BINER	KODE GRAY
0	0-22.5	0000	0000
1	22.5-45	0001	0001
2	45-67.5	0010	0011
3	67.5-90	0011	0010
4	90-112.5	0100	0110
5	112.5-135	0101	0111
6	135-157.5	0110	0101
7	157.5-180	0111	0100
8	180-202.5	1000	1100
9	202.5-225	1001	1101
10	225-247.5	1010	1111
11	247.5-270	1011	1110
12	270-292.5	1100	1010
13	292.5-315	1101	1011
14	315-337.5	1110	1001
15	337.5-360	1111	1000

2.4.2 INCREMENTAL ROTARY ENCODER

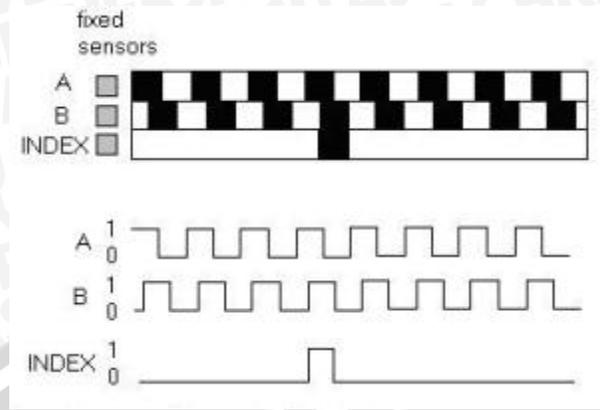
Incremental rotary encoder terdiri dari dua track atau single track dan dua sensor yang disebut channel A dan B (Gambar 2.12). Ketika poros berputar, deretan pulsa akan muncul di masing-masing channel pada frekuensi yang proporsional

dengan kecepatan putar sedangkan hubungan fasa antara channel A dan B menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur. Untuk mengetahui arah putaran, dengan mengetahui channel mana yang leading terhadap channel satunya dapat kita tentukan arah putaran yang terjadi karena kedua channel tersebut akan selalu berbeda fasa seperempat putaran (quadrature signal). Seringkali terdapat output channel ketiga, disebut INDEX, yang menghasilkan satu pulsa per putaran berguna untuk menghitung jumlah putaran yang terjadi.

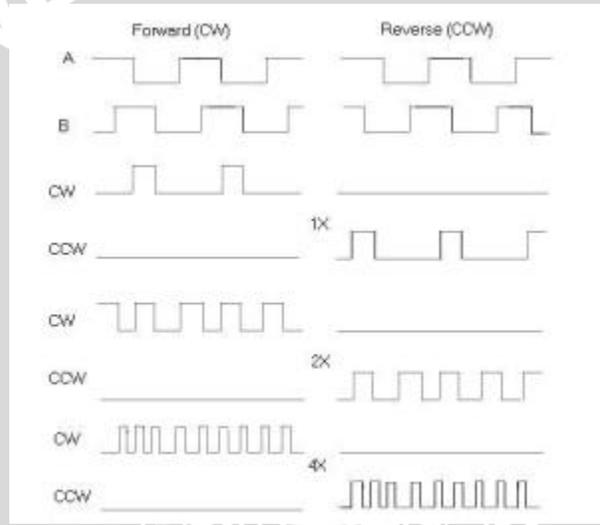


Gambar 2.12 Susunan Piringan untuk Incremental Encoder

Contoh pola diagram keluaran dari suatu incremental encoder ditunjukkan dalam Gambar 2.13. Resolusi keluaran dari sinyal quadrature A dan B dapat dibuat beberapa macam, yaitu 1X, 2X dan 4X. Resolusi 1X hanya memberikan pulsa tunggal untuk setiap siklus salah satu sinyal A atau B, sedangkan resolusi 4X memberikan pulsa setiap transisi pada kedua sinyal A dan B menjadi empat kali resolusi 1X. Gambar 2.14 contoh output arah putaran pada resolusi yang berbeda-beda. Arah putaran dapat ditentukan melalui level salah satu sinyal selama transisi terhadap sinyal yang kedua. Pada contoh resolusi 1X, $A = \text{arah bawah dengan } B = 1$ menunjukkan arah putaran searah jarum jam, sebaliknya $B = \text{arah bawah dengan } A = 1$ menunjukkan arah berlawanan jarum jam.



Gambar 2.13 Contoh Pola Keluaran Incremental Encoder



Gambar 2.14 Output dan Arah Putaran pada Resolusi yang Berbeda-beda

Pada incremental encoder, beberapa cara dapat digunakan untuk menentukan kecepatan yang diamati dari sinyal pulsa yang dihasilkan. Diantaranya adalah menggunakan frequencymeter dan periodimeter

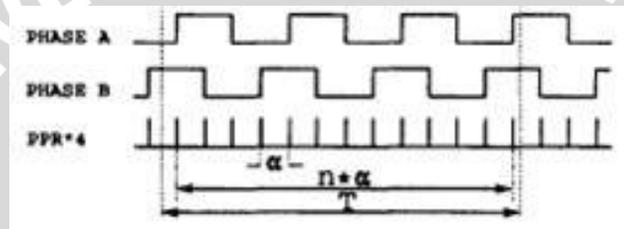
Cara yang sederhana untuk menentukan kecepatan dapat dengan frequencymeter, yakni menghitung jumlah pulsa dari encoder (n) pada selang waktu yang tetap (T) yang merupakan periode loop kecepatan (persamaan 2.11). Apabila α adalah sudut antara pulsa encoder, maka sudut putaran pada suatu periode adalah:

$$\alpha_f = n\omega \quad (2.10)$$

Sehingga kecepatan putar akan kita dapatkan sebagai:

$$\omega_r = \frac{\alpha_f}{T} \quad (2.11)$$

Kelemahan yang muncul pada cara ini adalah pada setiap periode sudut α_f yang didapat merupakan kelipatan integer dari α . Ini akan dapat menghasilkan error pada kecepatan yang ingin diukur. Dengan menggunakan frequencymeter dapat menghasilkan sinyal keluaran seperti Gambar 2.15.



Gambar 2.15 Sinyal Keluaran Encoder untuk Pengukuran Kecepatan dengan Frequencymeter

Cara yang lain adalah dengan menggunakan periodimeter. Dengan cara ini kita akan mengukur kecepatan tidak lagi dengan menghitung jumlah pulsa encoder tetapi dengan menghitung clock frekuensi tinggi (HF Clock) untuk sebuah pulsa dari encoder yaitu mengukur periode pulsa dari encoder (persamaan 2.13). Apabila α_p adalah sudut dari pulsa encoder, t adalah periode dari HF clock, dan n adalah jumlah pulsa HF yang terhitung pada counter. Maka waktu untuk sebuah pulsa encoder (T_p) adalah:

$$T_p = nt \quad (2.12)$$

Sehingga kecepatan yang akan kita ukur dapat kita peroleh dengan:

$$\omega_r = \frac{\alpha_p}{T_p} \quad (2.13)$$

Seperti halnya pada frequencymeter, disini juga muncul error karena waktu T_p akan selalu merupakan perkalian integer dengan t .

2.5 Liquid Crystal Display (LCD)

Liquid crystal display (LCD) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Jenis LCD yang dipakai pada alat ini adalah LCD M1632. LCD terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/ angka dua baris, masing – masing baris bisa menampung 16 huruf/ angka. Berikut merupakan tampilan dari LCD dalam Gambar 2.16



Gambar 2.16 LCD Character 2x16 (Irwansetyo, 2010)

Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempel dibalik pada panel LCD, berfungsi mengatur tampilan LCD. Dengan demikian pemakaian LCD M1632 menjadi sederhana, sistem lain cukup mengirimkan kode – kode ASCII dari informasi yang ditampilkan. Berikut merupakan spesifikasi LCD M1632 :

1. Tampilan 16 karakter 2 baris dengan matrik 5 x 7 + kursor.
2. ROM pembangkit karakter 192 jenis.
3. RAM pembangkit karakter 8 jenis (diprogram pemakai).
4. RAM data tampilan 80 x 8 bit (8 karakter).
5. Duty ratio 1/16.
6. RAM data tampilan dan RAM pembangkit karakter dapat dibaca dari unit mikroprosesor.
7. Beberapa fungsi perintah antara lain adalah penghapusan tampilan (*display clear*), posisi kursor awal (*crusor home*), tampilan karakter kedip (*display character blink*), penggeseran kursor (*crusor shift*) dan penggeseran tampilan (*display shift*).
8. Rangkaian otomatis reset saat daya dinyalakan.
9. Catu daya tunggal +5 volt.

2.6 Kontroler

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu sub sistem, yaitu kontroler (Widodo Budiharto, 2003)

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini sesuai dengan tujuan sistem kontrol dimana mendapat nilai sinyal keluaran sama dengan nilai yang diinginkan (referensi). Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran *plant* dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, 1996: 197-204). Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengamati perbedaan ini untuk segera menghasilkan sinyal keluaran untuk mempengaruhi *plant*. Dengan demikian sistem secara cepat mengubah keluaran *plant* sampai diperoleh selisih dengan nilai referensi sekecil mungkin.

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, maka sistem harus memiliki kestabilan secara relatif, yakni tolok ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata, K., 1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespons *input*, dan bagaimana peredaman terhadap adanya lonjakan (*over shoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan. Sistem dikatakan tidak stabil jika *outputnya* berosilasi terus

menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu *input*, tidak dapat mengikuti *input* secara serentak, tapi menunjukkan respons transien berupa suatu osilasi teredam sebelum mencapai *steady state* (Alam Azral,2008)

Dalam sistem pengendalian terdapat 2 macam *loop*:

1. Pengendalian dengan *loop* terbuka

Sistem kontrol *loop* terbuka adalah sistem kontrol yang keluarannya tidak berpengaruh pada aksi pengontrolan. Jadi pada sistem kontrol *loop* terbuka, keluaran tidak diukur atau diumpun balikkan untuk dibandingkan dengan masukan.

2. Pengendalian dengan *loop* tertutup

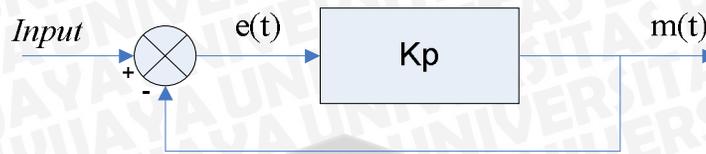
Sistem kontrol *loop* tertutup adalah sistem kontrol yang keluarannya mempunyai pengaruh langsung pada aksi pengontrolan. Disebut juga sistem kontrol yang menggunakan umpan balik untuk memperkecil kesalahan sistem.

2.7 Kontroler Proporsional Integral Diferensial (PID)

2.7.1 Kontroler Proporsional

Kontroler proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Gambar 2.17 menunjukkan diagram blok yang menggambarkan hubungan antara *input* (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal kesalahan (*error*) merupakan selisih antara besaran *setting* dengan besaran aktualnya.



Gambar 2.17 Diagram Blok Kontroler Proporsional (Ogata,K.,1995: 157)

Pada pengendali proporsional hubungan antara keluaran kontroler $m(t)$ dan sinyal kesalahan $e(t)$ adalah (Ogata,K., 1995: 157)

$$m(t) = Kp e(t) \quad (2.14)$$

atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = Kp \quad (2.15)$$

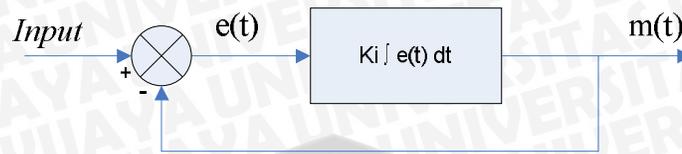
dengan Kp adalah penguatan proporsional. Keluaran $m(t)$ hanya tergantung pada Kp dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan. Penambahan Kp akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respons dan mengurangi kesalahan keadaan mantap.

Apapun wujud mekanisme yang sebenarnya dan apapun bentuk daya penggeraknya, kontroler proporsional pada dasarnya merupakan penguat dengan penguatan yang dapat diatur (Ogata, K., 1997).

2.7.2 Kontroler Integral

Kontroler integral berfungsi mengurangi kesalahan keadaan mantap yang dihasilkan pada kontroler proporsional sebelumnya. Kalau sebuah *plant* tidak memiliki unsur integrator ($1/s$), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantap nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Gambar 2.18 menunjukkan diagram blok kontroler integral (Ogata, K.,1997)



Gambar 2.18 Diagram Blok Kontroler Integral (Ogata, 1997: 158)

Nilai keluaran kontroler $m(t)$ sebanding dengan integral sinyal kesalahan $e(t)$,
Sehingga (Ogata, 1995: 157)

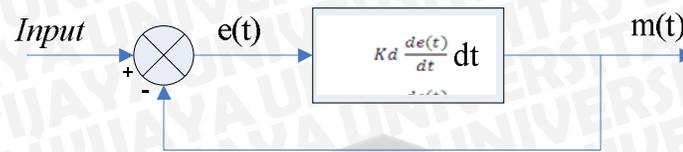
$$\frac{dm(t)}{dt} = K_i e(t) \quad (2.16)$$

$$m(t) = K_i \int_0^t e(t) dt \quad (2.17)$$

dengan K_i adalah konstanta integral. Jika sinyal kesalahan $e(t)=0$, maka laju perubahan sinyal kendali integral $\frac{dm(t)}{dt} = 0$ atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Jika harga $e(t)$ diduakalikan, maka harga $m(t)$ berubah dengan laju perubahan menjadi dua kali semula. Jika kesalahan penggerak nol, maka harga $m(t)$ tetap stasioner. Aksi kontrol integral seringkali disebut kontrol *reset* (Ogata,K.,1997). Aksi kontrol integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respons.

2.7.3 Kontroler Diferensial

Kontroler diferensial digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar *phase lead* terhadap penguatan kontrol dan mengurangi *phase lag* penguatan tersebut (Ogata,K.,1997). Kontroler diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.19 berikut menunjukkan diagram blok pada kontroler diferensial.



Gambar 2.19 Diagram Blok Kontroler Diferensial (Ogata, 1995: 177)

Nilai keluaran kontroler $m(t)$ sebanding laju sinyal kesalahan $\frac{de(t)}{dt}$. Hubungan ini dapat ditulis sebagai: (Ogata, 1995: 179)

$$m(t) = K_d \frac{de(t)}{dt} \quad (2.18)$$

Kontroler differensial akan memberikan sinyal kendali keluaran $m(t) = 0$, untuk sinyal kesalahan $e(t)$ yang konstan sehingga kontroler differensial tidak mempengaruhi keadaan mantap. Kontroler differensial digunakan untuk memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler differensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler differensial hanyalah efek dari lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler differensial tidak bisa digunakan tanpa ada kontroler lain.

Dari ketiga aksi kontrol dasar di atas dapat dibuat kombinasi dari ketiganya, yaitu Kontroler Proporsional Integral Diferensial (PID).

2.7.4 Kontroler Proporsional Integral Diferensial (PID)

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral differensial (PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10).

Aksi kontrolnya dinyatakan sebagai:

$$u(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot e(t)dt + K_p \cdot T_d \frac{de(t)}{dt} \quad (2.19)$$

dimana $\frac{1}{T_i} = K_i$ dan $T_d = K_d$

Bentuk integral dan differensial dapat ditulis dalam bentuk diskrit seperti pada persamaan

$$\int_0^t e(t)dt \approx T \sum_0^k e(k) \quad (2.20)$$

$$\frac{de(t)}{dt} \approx \frac{e_k - e_{k-1}}{T} \quad (2.21)$$

$$u(t) = K_p e(k) + K_i T \sum_0^k e(k) + \frac{1}{T} K_d (e_k - e_{k-1}) \quad (2.22)$$

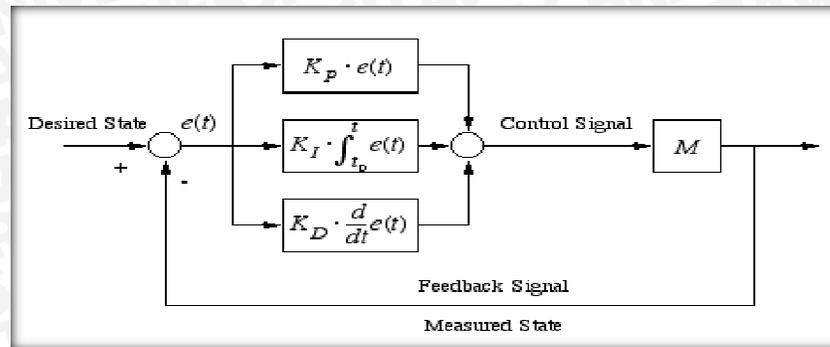
$$u = K_p \cdot xerror + K_i \cdot (error * dt + last_error) \cdot xTs + \frac{K_d}{Ts} (error - last_error) \quad (2.23)$$

Dimana:

- K_p ialah konstanta proportional
- K_i ialah konstanta integral
- K_d ialah konstanta diferensial
- $error$ ialah nilai kesalahan
- $last_error$ ialah nilai kesalahan sebelumnya
- T_s ialah *sampling time* (waktu cuplik)

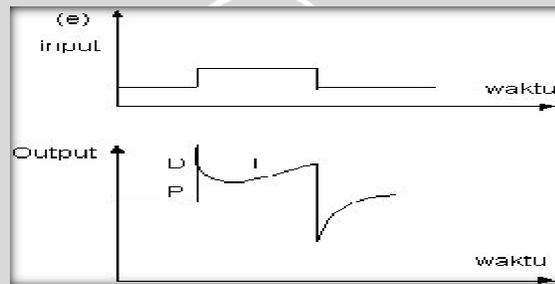
Jenis kontroler ini digunakan untuk memperbaiki kecepatan respons, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.

Gambar 2.20 menunjukkan diagram blok kontroler PID.



Gambar 2.20 Diagram Blok Kontroler PID

Keluaran kontroler PID merupakan penjumlahan dari keluaran komponen proporsional, integral dan diferensial. Hubungan tersebut ditunjukkan dalam Gambar 2.22.



Gambar 2.22 Fungsi Waktu antara Sinyal Keluaran dan Sinyal Masukan Kontroler PID

(Gunterus, 1994:8-11)

Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta K_p , T_i dan T_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah yang akan memberikan kontribusi pada respons sistem secara keseluruhan (Gunterus, 1994, 8-10).

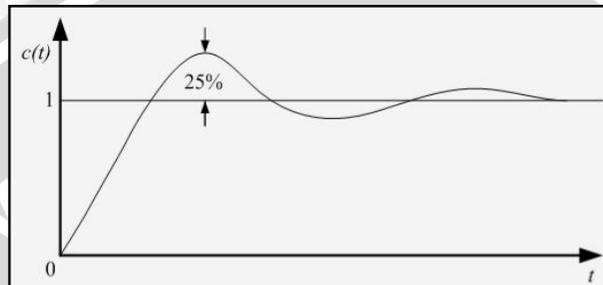
2.7.5 Metode Tuning PID

Metode Ziegler-Nichols

Ziegler dan Nichols mengemukakan aturan-aturan untuk menentukan nilai dari gain proporsional K_p , waktu integral T_i , dan waktu derivatif T_d berdasarkan

karakteristik respon transien dari *plant* yang diberikan. Penentuan parameter kontroler PID atau penalaan kontroler PID tersebut dapat dilakukan dengan bereksperimen dengan *plant* (Ogata, K., 1997).

Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respons *unit step*, seperti ditunjukkan dalam Gambar 2.23.



Gambar 2.23 Kurva Respons *Unit Step* yang Menunjukkan 25% *Maximum Overshoot* (Ogata,K. 1997)

a) Metode Pertama

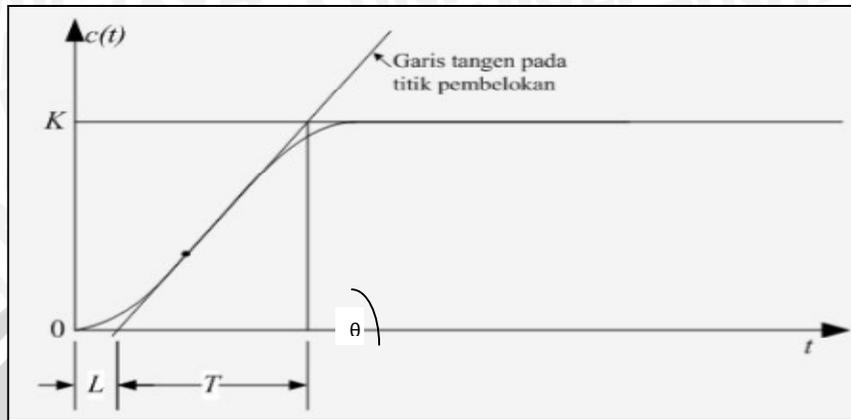
Metode pertama atau sering disebut metode kurva reaksi, respons dari *plant* dapat diperoleh secara *eksperimental* dengan masukan berupa *unit step*, seperti yang ditunjukkan dalam Gambar 2.24.



Gambar 2.24 Respons *Plant* Terhadap Masukan Berupa *Unit Step* (Ogata,K. 1997)

Jika dalam *plant* tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respons *unit step* berbentuk seperti huruf S. Kurva berbentuk S tersebut dapat dikarakteristikan menjadi dua konstanta yaitu waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar sebuah garis tangen pada titik pembelokan dari kurva S, dan menentukan perpotongan antara garis tangen dengan sumbu waktu t dan sumbu

$c(t) = K$, seperti yang telah ditunjukkan dalam Gambar 2.25. Jika respons tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997)



Gambar 2.25 Respon Plant Berbentuk S (Ogata,K. 1997)

Fungsi alih $C(s)/U(s)$ dapat dilakukan pendekatan dengan sistem orde satu dengan persamaan sebagai berikut:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts+1} \tag{2.24}$$

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari K_p , T_i dan T_d berdasarkan pada formula yang ditunjukkan dalam Tabel 2.2 (Ogata, K., 1997).

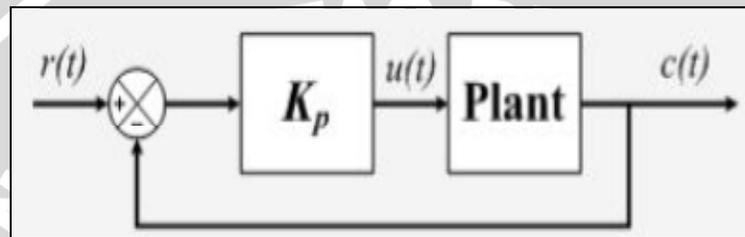
Tabel 2.2 Aturan Penalaan Ziegler-Nichols berdasarkan Respons Unit Step (Ogata, K. 1997)

Tipe Kontroler	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5 L$

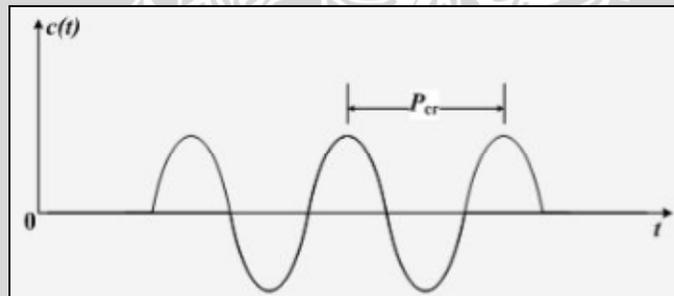
b) Metode Kedua

Dalam metode kedua Ziegler-Nichols, mula-mula yang dilakukan adalah membuat $T_i = 0$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol

proporsional, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki osilasi yang berkesinambungan (Jika keluaran tidak memiliki osilasi berkesinambungan untuk nilai K_p manapun yang telah diambil, maka metode ini tidak berlaku). Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Diagram blok sistem *loop* tertutup dengan kontroler proporsional dapat dilihat dalam Gambar 2.26 dan untuk osilasi berkesinambungan dengan periode P_{cr} dapat dilihat dalam Gambar 2.27.



Gambar 2.26 Sistem Loop Tertutup dengan Kontroler Proporsional (Ogata,K. 1997)



Gambar 2.27 Osilasi Berkesinambungan dengan Periode P_{cr} (Ogata,K. 1997)

Ziegler dan Nichols menyarankan penyetelan nilai parameter K_p, T_i, T_d dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.3. (Ogata, K., 1997)

Tabel 2.3 Aturan Dasar Ziegler-Nichols Berdasarkan *Critical Gain K_{cr}* dan *Critical Period P_{cr}* (Ogata,K. 1997)

Type Kontroler	K_p	T_i	T_d
P	$0.5 K_{cr}$	∞	0

PI	0.45 Kcr	$\frac{1}{12}$ Pcr	0
PID	0.60 Kcr	0.5 Pcr	0.125 Pcr

Respon transien dalam sistem kontrol praktis sering menunjukkan osilasi teredam sebelum mencapai keadaan steady state. Berikut adalah parameter karakteristik respon transien terhadap masukan *unit step*:

1. *Delay time* (td) adalah waktu yang diperlukan respon untuk mencapai setengah harga akhir yang pertama kali.
2. *Rise time* (tr) adalah waktu yang diperlukan respon untuk naik dari 10% sampai 90%, 5% sampai 95% atau 0% sampai 100% dari harga akhirnya.
3. *Peak time* (tp)) adalah waktu yang diperlukan respon untuk mencapai puncak lewatan yang pertama kali.
4. *Maximum overshoot* (Mp) adalah persen harga puncak maksimum dari kurva respon yang diukur dari satu. Jika harga keadaan tunak tidak sama dengan satu, maka biasanya digunakan persen lewatan maksimum yang didefinisikan seperti persamaan 2.25.

$$M_p = \frac{c(tp) - c(\infty)}{c(\infty)} \times 100\% \quad (2.25)$$

Dimana

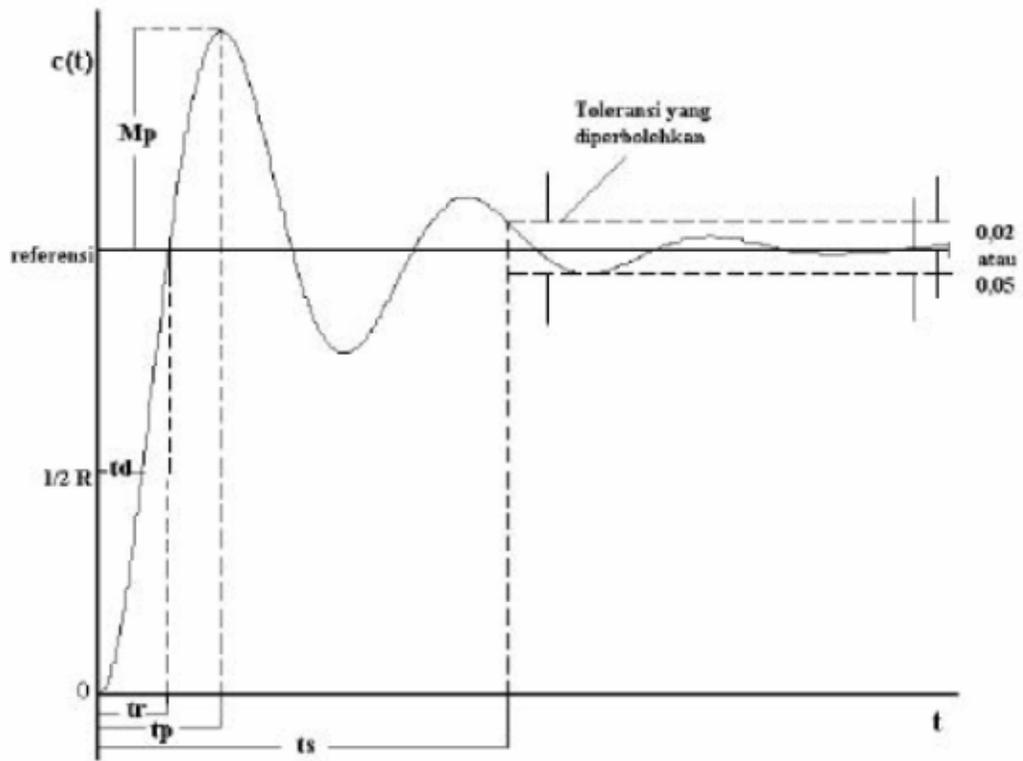
$c(\infty)$: overshoot

$c(tp)$: nilai set point

Besarnya (persen) lewatan maksimum secara langsung menunjukkan kestabilan relative sistem.

5. *Settling time* (ts) adalah waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah disekitar harga akhir yang ukurannya ditentukan dengan presentase mutlak dari harga akhir (biasanya 2% atau 5%).

Berikut merupakan grafik dari respon metode osilasi dalam Gambar 2.28



Gambar 2.28 Kurva Tanggapan Sistem dengan Karakteristik *Transient Respon*

Keterangan

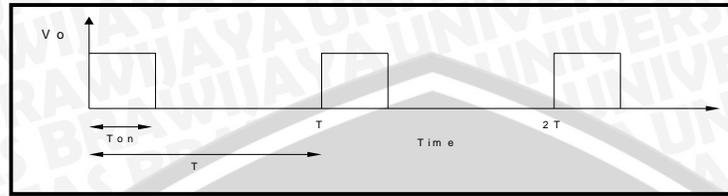
- Mp : maksimum overshoot
- td : delay time
- tr : rise time
- ts : settling time
- tp : peak time

2.8 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) digunakan untuk mengatur kecepatan dari motor DC. Kecepatan motor DC tergantung pada besarnya *duty cycle* yang diberikan pada motor DC tersebut.

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang

diinginkan. Pada AVR ATMEGA 32 *duty cycle* ditentukan dari *Output Compare Register 0* (OCR0). Sinyal PWM secara umum dapat dilihat dalam Gambar 2.29.



Gambar 2.29 Sinyal PWM secara Umum (Irwansetyo, 2010)

$$Duty\ cycle = \frac{T_{on}}{T} \times 100\% \quad (2.26)$$

$$V_{DC} = Duty\ cycle \times V_{CC} \quad (2.27)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus sebagai berikut:

$$f_{OCn} = \frac{f_{clk_I/O}}{N \cdot 256} \quad (2.28)$$

Timer/counter yang digunakan pada PWM ini yaitu *Timer/Counter0* (8-bit) dengan mode Fast PWM dan *Prescalers factor* (N) yaitu 256.