

BAB II

TINJAUAN PUSTAKA

Dalam bagian ini akan dijelaskan mengenai teori-teori yang akan menunjang perancangan Sistem Identifikasi Kontrol Kecepatan Motor DC dengan metode PID *self tuning* menggunakan MATLAB. Pengetahuan yang mendukung perencanaan dan realisasi alat meliputi sebagai berikut:

- Motor *Direct Current* (DC)
- *Pulse Width Modulation* (PWM)
- Mikrokontroler Arduino Uno
- Sensor *Photo-Electric*
- Kontroler PID *self tuning*
- *Driver* motor DC

2.1 Motor DC

Prinsip kerja motor DC sesuai dengan hukum kemagnetan Lorentz, yaitu membangkitkan fungsi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul GGL induksi. Setiap arus yang mengalir melalui sebuah konduktor akan timbul medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

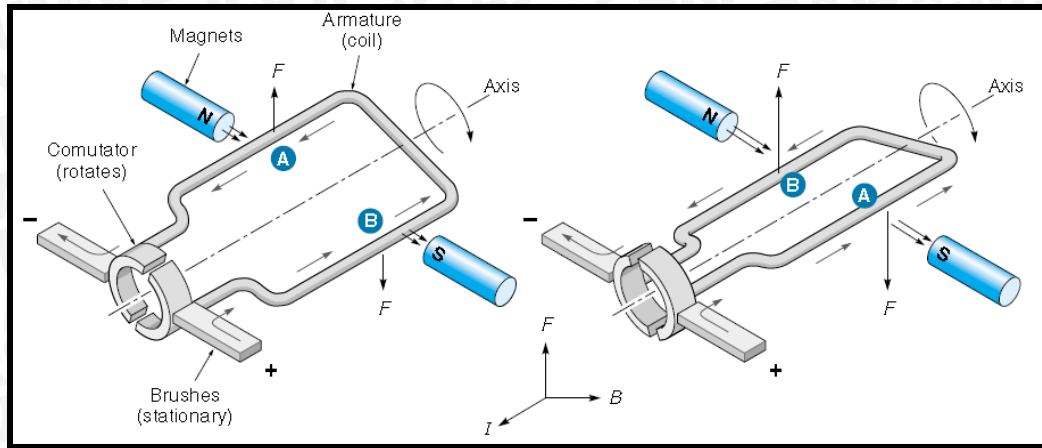
Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah medan putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B.I.L \text{ (Newton)} \quad (2.1)$$

Dimana :

B = kerapatan fluks magnet (weber), L = panjang konduktor (meter), I = arus listrik (ampere).

(Soemarwanto, 1999)



Gambar 2.1 Motor DC

Sumber : Kilian, 1996

Gambar 2.1 adalah ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat A-B berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (armature).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*commutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

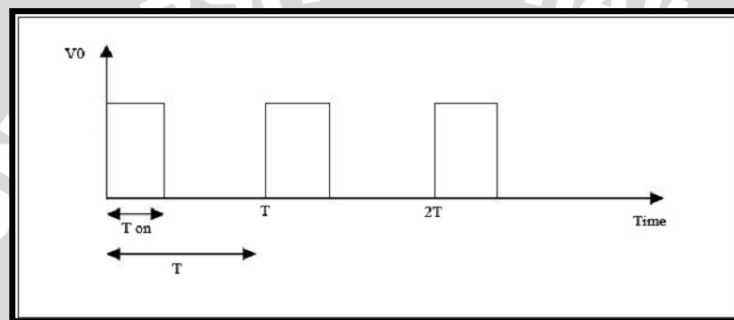
Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat A dan B secara bergantian berada di kiri dan kanan. Karena itu arah arus di A dan arah arus di B selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

Bagian *commutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Dalam gambar 2.1 arah fluks magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar B adalah ke atas, sementara gaya yang bekerja pada penghantar A adalah ke bawah. Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar 180° arah arus berbalik, pada saat itu penghantar A dan B bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

2.2 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. *Duty cycle* adalah rasio atau perbandingan lama waktu suatu sinyal berada dalam kondisi *high* dengan lama waktu suatu sinyal tersebut dalam kondisi (*high+low*).

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM secara umum dapat dilihat dalam Gambar 2.2



Gambar 2.2 Sinyal PWM Secara Umum

Sumber: www.electronics-scheme.com

$$\text{Duty cycle} = \frac{T_{\text{on}}}{T} \times 100\% \dots (\%) \quad (2.2)$$

Dengan :

T_{on} = Periode logika tinggi

T = Periode keseluruhan

$$V_{dc} = \text{Dutycycle} \times V_{cc} \dots (V) \quad (2.3)$$

Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut

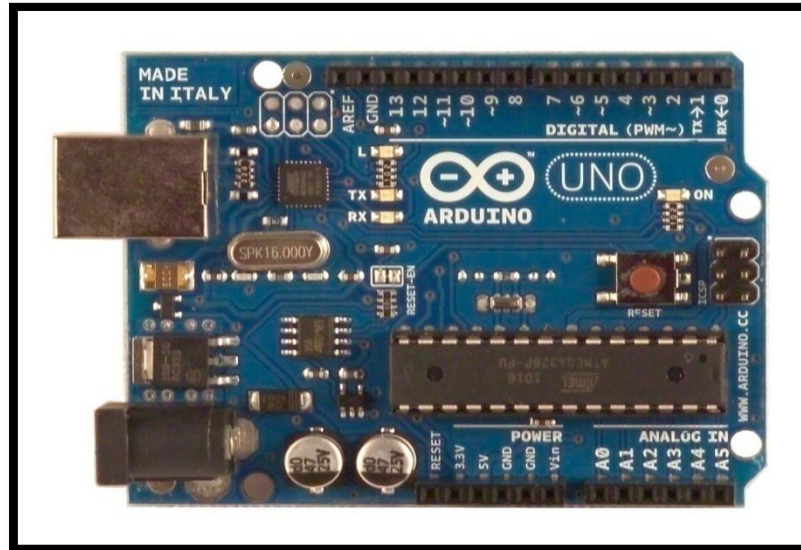
$$f_{\text{on}} = \frac{f_{\text{clk I/O}}}{N.256} \dots (\text{Hz}) \quad (2.4)$$

2.3 Arduino Uno

Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi *Universal Serial Bus* (USB), *jack power*, *In-circuit Serial Programming* (ICSP) *header*, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan

Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.

Uno berbeda dengan semua *board* sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-serial berbeda dengan *board* sebelumnya yang menggunakan chip FTDI driver USB-to-serial. Gambar 2.3 menunjukkan tampak depan dari *board* arduino uno.



Gambar 2.3 Tampak Depan Arduino Uno

Sumber: www.arduino.cc

Nama “Uno” berarti *satu* dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0. Uno dan versi 1.0 akan menjadi versi referensi dari Arduino. Uno adalah yang terbaru dalam serangkaian *board* USB Arduino, dan sebagai model referensi untuk *platform* Arduino, untuk perbandingan dengan versi sebelumnya.

2.3.1 Daya

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal (otomatis). Eksternal (non-USB) daya dapat berasal baik dari AC-ke adaptor-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug jack* pusat-positif ukuran 2.1mm konektor POWER. Ujung kepala dari baterai dapat dimasukkan kedalam Gnd dan Vin pin *header* dari konektor POWER. Kisaran kebutuhan daya yang disarankan untuk *board uno* adalah 7 sampai dengan 12 volt, jika diberi daya kurang dari

7 volt kemungkinan pin 5v *uno* dapat beroperasi tetapi tidak stabil kemudian jika diberi daya lebih dari 12V, regulator tegangan bisa panas dan dapat merusak *board uno*.

Pin listrik adalah sebagai berikut:

N. Tegangan masukan kepada *board* Arduino ketika itu menggunakan sumber daya eksternal (sebagai pengganti dari 5 volt koneksi USB atau sumber daya lainnya) 5V. Catu daya digunakan untuk daya mikrokontroler dan komponen lainnya. Sebuah pasokan 3,3 volt dihasilkan oleh regulator on-board.

GND. *Ground pin*.

2.3.2 Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader), 2 KB dari SRAM dan 1 KB EEPROM

2.3.3 Input dan Output

Masing-masing dari 14 pin digital di Uno dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode ()*, *digitalWrite ()*, dan *digitalRead ()*, beroperasi dengan daya 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki internal *pull-up* resistor (secara *default* terputus) dari 20-50 kOhm. Selain itu, beberapa pin memiliki fungsi khusus:

- a. *Serial*: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini dihubungkan ke pin yang berkaitan dengan chip Serial ATmega8U2 USB-to-TTL.
- b. *Eksternal menyela*: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai. *PWM*: 3, 5, 6, 9, 10, dan 11. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite ()*.
- c. *SPI*: 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan *SPI library*.
- d. *LED*: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin bernilai nilai HIGH, LED on, ketika pin bernilai LOW, LED off.

- e. Uno memiliki 6 masukan analog, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda). Selain itu, beberapa pin memiliki fungsi khusus:
 1. *I2C: A4 (SDA) dan A5 (SCL)*. Dukungan I2C (TWI) komunikasi menggunakan perpustakaan *wire*.
 2. *Aref*. Tegangan referensi (0 sampai 5V saja) untuk *input* analog. Digunakan dengan fungsi *analogReference()*.
 3. *Reset*. Bawa baris ini LOW untuk me-reset mikrokontroler.

2.3.4 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) untuk komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX). Sebuah ATmega8U2 sebagai saluran komunikasi serial melalui USB dan sebagai port virtual com untuk perangkat lunak pada komputer. *Firmware* '8 U2 menggunakan *driver* USB standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada Windows diperlukan, sebuah file '.inf'. Perangkat lunak Arduino terdapat monitor serial yang memungkinkan digunakan memonitor data tekstual sederhana yang akan dikirim ke atau dari *board* Arduino. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah *SoftwareSerial library* memungkinkan untuk berkomunikasi secara serial pada salah satu pin digital pada *board* Uno's. ATmega328 juga mendukung I2C (TWI) dan komunikasi SPI. Perangkat lunak Arduino termasuk perpustakaan Kawat untuk menyederhanakan penggunaan bus I2C.

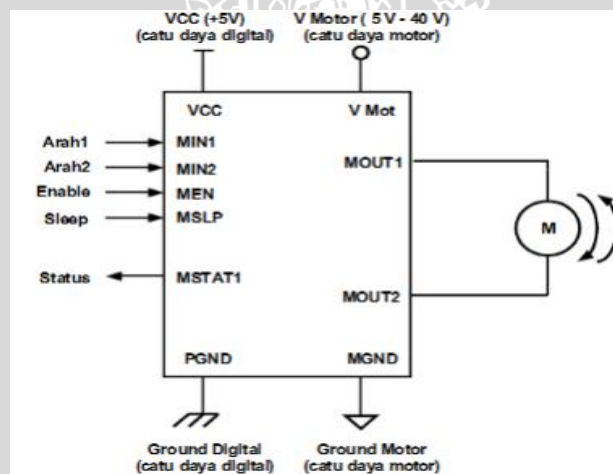
2.4 Modul Pengendali Motor DC EMS 5A H-Bridge

Modul pengendali motor DC yang digunakan adalah modul EMS 5A *H-Bridge* ditunjukkan pada Gambar 2.4. Secara garis besar, fungsi modul pengendali motor adalah untuk mengendalikan arah dan kecepatan putaran motor DC sesuai instruksi kendali dari

mikrokontroler pengguna. Gambar koneksi dari modul pengendali motor DC EMS 5A *H-bridge* ditunjukkan pada Gambar 2.5.



Gambar 2.4 Pengendali motor DC menggunakan EMS 5A *H-Bridge* Koneksi Modul EMS 5A *H-Bridge*

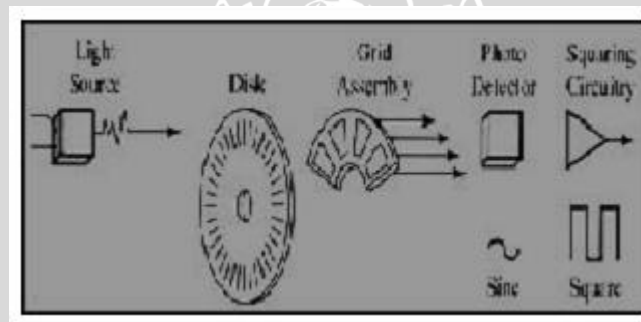


Gambar 2.5 Koneksi Modul EMS 5A *H-Bridge*

2.5 Sensor Rotary Encoder

Rotary encoder adalah elektromekanik yang dapat mendeteksi atau memonitor gerakan dan posisi. *Rotary encoder* biasanya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat dijadikan gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder*.

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 yanglain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dariLED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringanmenentukan akurasi *rotary encoder* tersebut.



Gambar 2.6 Blok Penyusun Rotary Encoder

Rangkaian penghasil pulsa ditunjukkan pada Gambar 2.6 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 :

$$f_c = \frac{s_w n}{60} \quad (2.5)$$

f_c = Frekuensi *cut-off filter*, s_w adalah kecepatan piringan dan n adalah jumlah slot pada piringan.

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.

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.

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.

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).

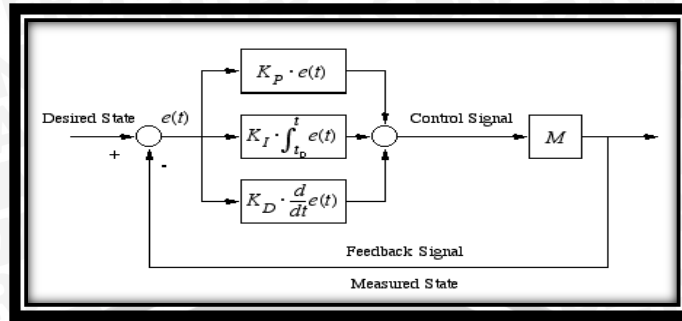
2.6.1 Kontroler PID (Proportional Integral Derivative)

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). Kontroler PID memiliki diagram kendali seperti yang ditunjukkan dalam Gambar 6.5.

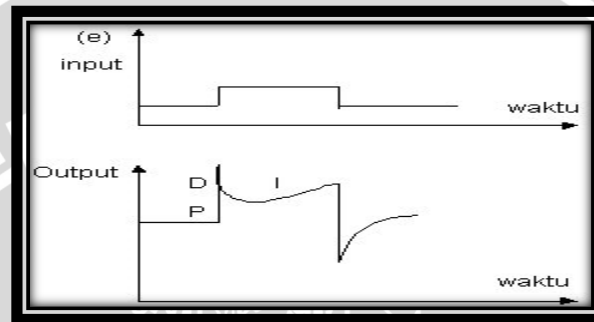
Aksi kontrolnya dinyatakan sebagai:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.6)$$

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



Gambar 2.7 Diagram blok kontroler PID



Gambar 2.8 Hubungan fungsi waktu antara sinyal keluaran dan sinyal masukan kontroler PID

Sumber: Gunterus, 1994:8-11

Keluaran kontroler PID merupakan penjumlahan dari keluaran kontroler proporsional, integral dan differensial. Gambar diatas menunjukkan hubungan tersebut. 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 konstribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

2.6.2 Kontroler Self Tuning PID (Proportional Integral Derivative) Metode Backward Rectangular

Versi Ideal dari *continuous-time* PID kontroler adalah sebagai berikut:

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_1^t e(\tau) d\tau + T_d \frac{de(t)}{dt}] \quad (2.7)$$

Untuk Mendapatkan versi *digital* dari PID kontroler, diperlukan untuk mendiskritisasi komponen *integral* dan *derivative* dari persamaan diatas. Untuk mendiskritisasi

komponen *integral*, digunakan metode *forward rectangular*, metode *backward rectangular* dan metode *trapezoidal rectangular*. Dalam skripsi ini metode yang digunakan adalah metode *backward rectangular*. Sedangkan untuk mendiskritisasi komponen *derivative* yaitu dengan cara mengganti *derivative* dengan differensial orde 1 (dua poin differensial)

$$\frac{de}{dt} \approx \frac{e(k) - e(k-1)}{T_0} = \frac{\Delta e(k)}{T_0} \quad (2.8)$$

$e(k)$ adalah nilai *error* pada waktu sampling- k , pada waktu $t = kT_0$.

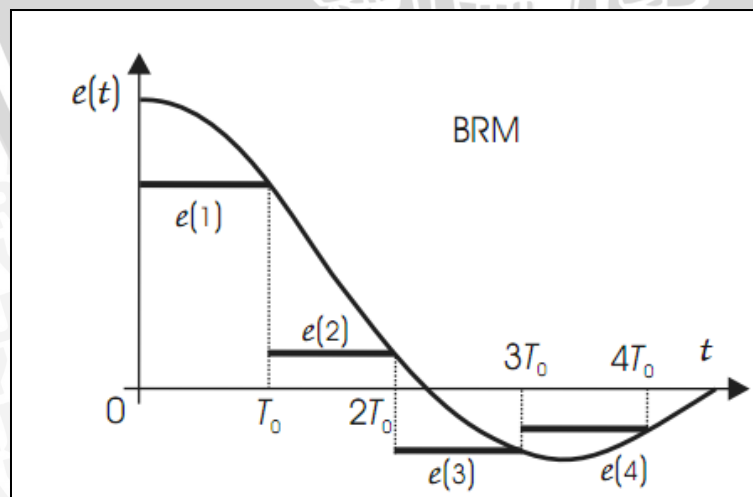
Untuk mendiskritisasi *integral*, jika *continuous-time* sinyal di diskritisasi rekursif menggunakan fungsi *step*, maka disebut metode *backward rectangular*, dan didapat persamaan

$$\int_1^t e(\tau) d\tau \approx T_0 \sum_{i=1}^k e(i) \quad (2.9)$$

Jadi persamaan diskrit PID kontroler adalah

$$u(k) = K_P \{e(k) + \frac{T_0}{T_I} \sum_{i=1}^k e(i) + \frac{T_D}{T_0} [e(k) - e(k-1)]\} \quad (2.10)$$

Metode *backward rectangular* untuk mendiskritisasi *integral* dari *continuous-time error* $e(t)$ dalam nilai $e(kT_0)$, dimana $k = 1, 2, \dots$. Diilustrasikan pada Gambar 2.9



Gambar 2.9 Diskritisasi Komponen Integral Metode Backward Rectangular

Algoritma persamaan beda menghitung nilai aktual dari *output* kontroler $u(k)$ dari nilai sebelumnya $u(k-1)$ dan perubahan $\Delta u(k)$ terlihat lebih praktis digunakan. Dengan menggunakan persamaan (2.10), didapat dari metode *backward rectangular*, untuk *step* k dan $k-1$, didapatkan persamaan beda

$$u(k) = \Delta u(k) + u(k-1) \quad (2.11)$$

$$\Delta u(k) = K_p \left\{ e(k) - e(k-1) + \frac{T_0}{T_I} e(k) + \frac{T_D}{T_0} [e(k) - 2e(k-1) + e(k-2)] \right\} \quad (2.12)$$

atau dalam bentuk umum

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1) \quad (2.13)$$

Dimana

$$q_0 = K_p \left(1 + \frac{T_0}{T_I} + \frac{T_D}{T_0} \right) \quad (2.14)$$

$$q_1 = -K_p \left(1 + 2 \frac{T_D}{T_0} \right) \quad (2.15)$$

$$q_2 = K_p \frac{T_D}{T_0} \quad (2.16)$$

Bentuk umum fungsi alih *plant* pada *self tuning* PID adalah:

$$G(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_2 z^{-1} + a_1 z^{-2}} \quad (2.17)$$

dan vektor dari parameter proses yang terestimasi adalah $[a_1, a_2, b_1, b_2]^T$.