

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Gas Engine*

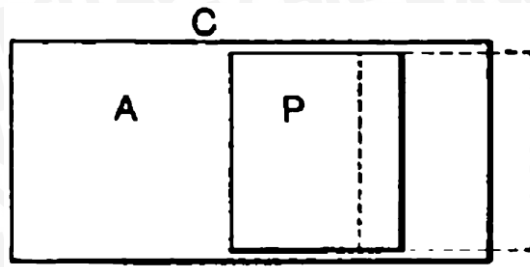
*Gas Engine* atau mesin pembakaran dalam, adalah sebuah mesin di mana bahan bakarnya dibakar langsung di dalam silinder. Setiap mesin pembakaran dalam adalah mesin bensin, tidak peduli apakah bahan bakar yang digunakan adalah gas atau cairan, karena dalam proses pembakarannya, bahan bakar cair awalnya akan diubah menjadi gas.

Klasifikasi umum dari *Gas Engine* adalah sebagai berikut:

1. Menurut bahan bakar: *Gas Engine* yang bekerja dengan menggunakan bahan bakar dalam bentuk gas, oli mesin, dan bahan bakar minyak yang lebih berat daripada bensin seperti minyak tanah, minyak bakar, minyak mentah, bensin, alkohol.
2. Menurut RPM ( Revolution per Minute ): Kecepatan tinggi misalnya, mesin mobil berjalan pada 1.200 RPM dan kecepatan lambat, mesin stasioner berjalan pada 100 RPM. Tetapi, kecepatan kaki piston per menitnya kemungkinan akan sama antara keduanya.
3. Menurut tak dalam siklus kerja: yaitu siklus 2 tak dan siklus 4 tak.

##### 2.1.1 Prinsip Kerja Umum

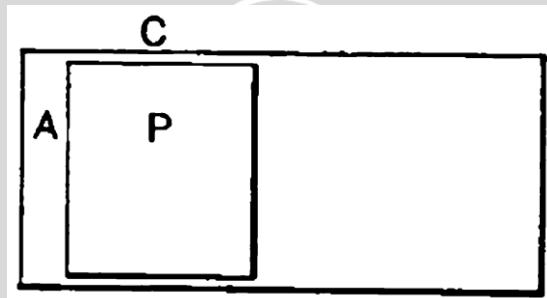
Pada gambar 2.1 terdapat silinder (C), piston (P) yang mudah dapat mendorong bolak-balik, tapi sangat erat dengan silinder sehingga udara tidak bisa bocor. Ruang (A) diisi dengan campuran udara dan uap dari beberapa bahan bakar cair, seperti bensin. Campuran ini terdapat pada tekanan udara dan saat dinyalakan akan menghasilkan pembakaran yang mana akan meningkatkan suhu, dan akibatnya tekanan di A menyebabkan campuran untuk mengembang dan mendorong piston keluar ke posisi yang ditunjukkan oleh garis putus-putus, atau sampai tekanan turun kembali ke kondisi normal.



Gambar 2.1 Blok Penampang *Gas Engine* dengan Memantik Api

Sumber: Mehrtens A. B. C., 1879: 1-2

Jika campuran udara dan bahan bakar awalnya dikompresi dengan menggerakkan piston dalam, seperti yang ditunjukkan pada gambar 2.2 dan kemudian menyala, piston akan didorong keluar dengan kekuatan yang jauh lebih besar dari sebelumnya dan bekerja lebih cepat lagi sesuai dengan banyaknya bahan bakar yang dikeluarkan. Saat ini, semua *Gas Engine* bekerja pada prinsip kompresi.



Gambar 2.2 Blok Penampang *Gas Engine* dengan Menggerakkan Piston

Sumber: Mehrtens A. B. C., 1909: 2-3

## 2.2 Kontroler

Dengan adanya kontroler dalam sebuah sistem kontrol sangat berperan penting terhadap seluruh perilaku yang terjadi pada sistem. Pada dasarnya semua itu disebabkan oleh komponen yang digunakan sebagai perancangan system tersebut. Artinya, karakteristik plant yang digunakan harus dapat diterima sebagaimana adanya, sehingga segala pergerakan dari sistem hanya dapat dilakukan dengan menambahkan subsistem yaitu kontroler.

Salah satu fungsi komponen kontroler adalah mengurangi sinyal kesalahan atau tingkat kesalahan sistem, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Dengan cara tersebut akan sesuai dengan tujuan sistem kontrol di mana mendapat nilai aktual atau sinyal keluaran sama dengan nilai yang diinginkan/referensi. Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Jika perbedaan antara nilai referensi dengan nilai keluaran selisihnya relatif besar, maka kontroler yang baik harus dapat mempengaruhi plant agar memperkecil selisih nilai keluaran *plant* dengan nilai referensi sekecil mungkin secara cepat dan tepat.

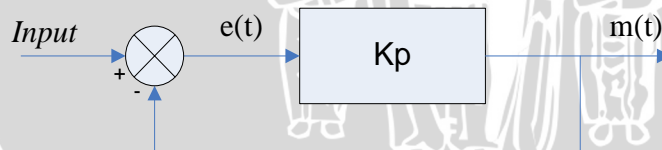
Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran plant dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, K., 1997).

### 2.3 Kontroler PID (Proporsional Integral Deferensial)

#### 2.3.1 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.

Pada Gambar 2.3 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. 3 Diagram Blok Kontroler Proporsional  
 Sumber: Ogata, K., 1997: 157

Pada pengendali proporsional hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan  $e(t)$  adalah:

$$m(t) = K_p e(t) \dots\dots\dots(2-1)$$

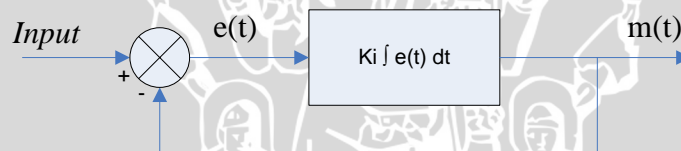
Sumber: Ogata, K., 1997: 157

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

### 2.3.2 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 2.4 menunjukkan diagram blok kontroler integral.



**Gambar 2. 4 Diagram Blok Kontroler Integral**  
 Sumber: Ogata, K., 1997: 158

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

$$\frac{dm(t)}{dt} = K_i \cdot e(t) \dots\dots\dots(2-2)$$

$$m(t) = K_i \int_0^t e(t)dt \dots\dots\dots(2-3)$$

Sumber: Ogata, K., 1997: 157

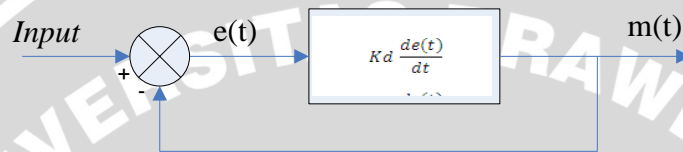
dengan  $K_i$  adalah konstanta integral. Jika sinyal kesalahan  $e(t)=0$ , maka laju perubahan sinyal kendali integral  $\frac{dm(t)}{dt} = 0$  atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi kontrol integral digunakan untuk



menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respons.

### 2.3.3 Kontroler Derferensial

Kontroler Derferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.5 berikut menunjukkan diagram blok pada kontroler Derferensial.



**Gambar 2. 5 Diagram Blok Kontroler Derferensial**

Sumber: Ogata, K., 1997: 177

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

$$m(t) = Kd \frac{de(t)}{dt} \dots\dots\dots(2-4)$$

Sumber: Ogata, 1997: 179

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

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

Dari ketiga aksi kontrol dasar di atas dapat dibuat kombinasi dari ketiganya, yaitu kontroler Proporsional Integral Derferensial (PID)



### 2.3.4 Kontroler Proporsional Integral Deferenensial (PID)

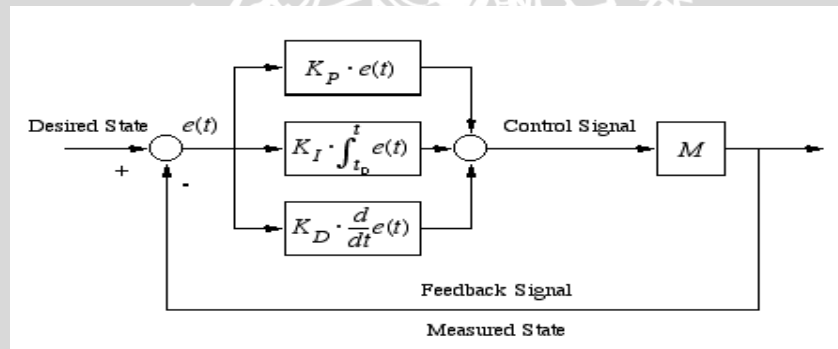
Setiap kekurangan dan kelebihan dari masing-masing kontroler Proporsional (P), Integral (I) dan Deferenensial (D) dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral Deferenensial (PID). Elemen-elemen kontroler Proporsional (P), Integral (I) dan Deferenensial (D) masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10). Kontroler Proporsional Integral Deferenensial (PID) memiliki diagram kendali seperti dalam Gambar 2.6.

Aksi kontrolnya dinyatakan sebagai:

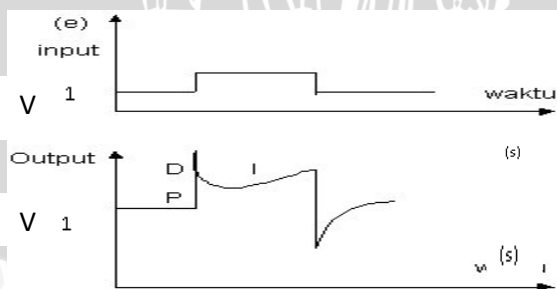
$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \dots \dots \dots (2-5)$$

Sumber: Ogata, K., 1997: 183

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



Gambar 2. 6 Diagram Blok Kontroler PID



Gambar 2. 7 Fungsi Waktu antara Sinyal Keluaran dan Sinyal Masukan Kontroler PID

Sumber: Gunterus, 1994:8-11

Keluaran kontroler Proporsional Integral Deferenensial (PID) merupakan penjumlahan dari keluaran kontroler proporsional, integral dan Deferenensial. Gambar 2.6 menunjukkan hubungan tersebut. Karakteristik kontroler Proporsional Integral



Deferensial (PID) sangat dipengaruhi oleh kontribusi besar dari ketiga parameter Proporsional (P), Integral (I) dan Deferensial (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.

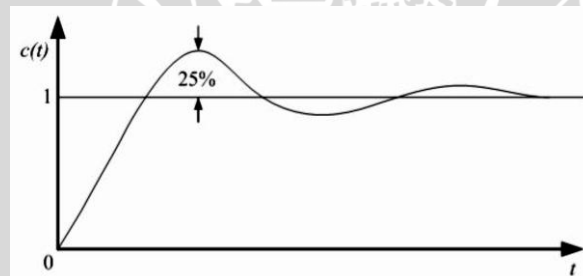
Konstanta yang menonjol itulah yang akan memberikan kontribusi pada respons sistem secara keseluruhan (Gunterus, 1994, 8-10).

### 2.3.5 Metode Perancangan Kontroler Proporsional Integral Deferensial (PID)

#### Menggunakan 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 plan. (Ogata, K., 1997)

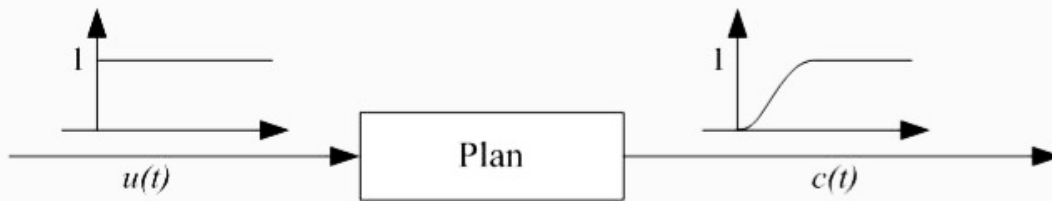
Terdapat dua metode yang disebut dengan aturan penalaan Ziegler-Nichols, pada kedua metode tersebut memiliki tujuan yang sama yaitu untuk mencapai 25% *maximum overshoot* pada respon unit step, seperti ditunjukkan dalam Gambar 2.8



Gambar 2.8 Kurva Respon Unit Step yang Menunjukkan 25% *Maximum Overshoot*  
Sumber: Ogata, K., 1997

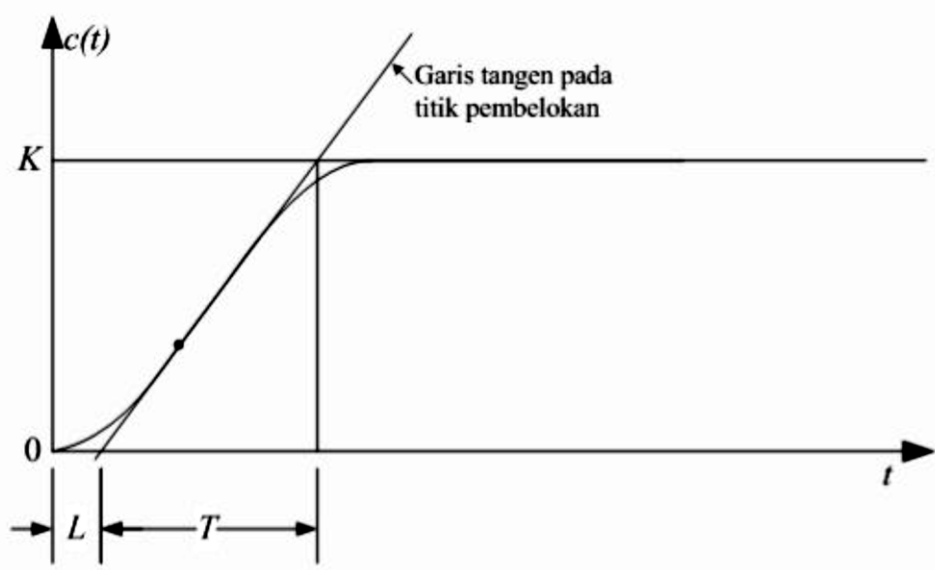
## a). Metode Pertama

Metode pertama atau sering disebut metode kurva reaksi, respon dari plan dapat diperoleh secara eksperimental dengan masukan berupa unit step, seperti yang ditunjukkan dalam Gambar 2.9



Gambar 2.9 Respon Plant Terhadap Masukan Berupa Unit Step  
Sumber: Ogata, K. 1997

Jika dalam plan tersebut terdapat integrator atau *dominan complex-conjugate poles*, maka kurva respon unit step berbentuk seperti huruf S, seperti dalam Gambar 2.10 jika respon tidak memberikan bentuk kurva S, maka metode ini tidak berlaku. (Ogata, K., 1997).



Gambar 2.10 Kurva Respon yang Berbentuk S  
Sumber: Ogata, K. 1997

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 2.10 Fungsi alih



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

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

Sumber: Ogata, K. 1997

Ziegler dan Nichols menyarankan untuk menentukan nilai-nilai dari  $K_p$ ,  $T_i$  dan  $T_d$  berdasarkan pada formula yang ditunjukkan dalam Tabel 2.1. (Ogata, K., 1997).

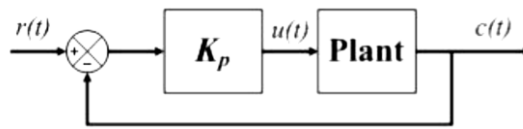
**Tabel 2.1. Aturan Penalaan Ziegler-Nichols Berdasarkan Respon Unit Step Dari Plan**

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

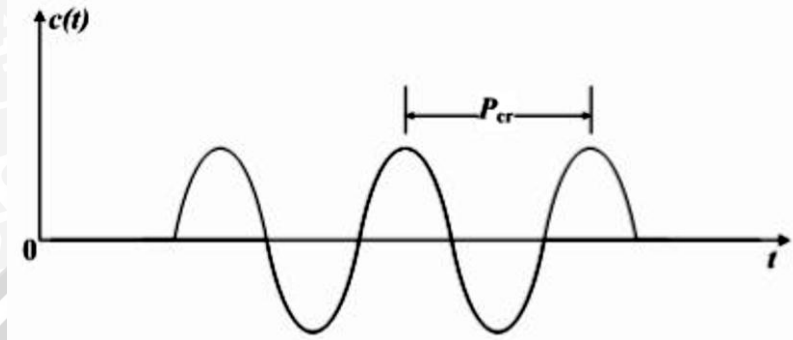
Sumber: Ogata, K. 1997

#### b). Metode Kedua

Dalam metode kedua ziegler-nichols, mula-mula yang dilakukan adalah membuat  $T_i = \infty$  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 2.11. dan untuk osilasi berkesinambungan dengan periode  $P_{cr}$  dapat dilihat dalam gambar 2.12. Ziegler dan Nichols menyarankan penyetelan nilai parameter  $K_p$ ,  $T_i$ ,  $T_d$  dan berdasarkan rumus yang diperlihatkan dalam Tabel 2.2 (Ogata, K., 1997).



Gambar 2.11 Sistem Loop Tertutup dengan Kontroler Proporsional  
 Sumber: Ogata, K., 1997



Gambar 2.12 Osilasi Berkesinambungan dengan Periode  $P_{cr}$   
 Sumber: Ogata, K., 1997

Tabel 2.2 Aturan Dasar Ziegler-Nichols Berdasarkan Critical Gain  $K_{cr}$  dan Critical Period  $P_{cr}$

Type Kontroler	$K_p$	$T_i$	$T_d$
P	$0.5 K_{cr}$	$\infty$	0
PI	$0.45 K_{cr}$	$\frac{1}{1.2} P_{cr}$	0
PID	$0.60 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

Sumber: Ogata, K., 1997

### 2.3.6 Hand Tuning Kontroler PID

Kontroler PID dapat di *tuning* dalam beberapa cara, antara lain Ziegler-Nichols *tuning*, *loop shaping*, metode analitis, optimisasi, *pole placement*, *auto tuning* dan *hand tuning* (Smith, 1979; Astrom & Hagglund, 1995). Pada skripsi ini digunakan cara *hand tuning* untuk menentukan besar  $K_p$ ,  $K_i$ , dan  $K_d$ . Hal ini dilakukan karena ada kendala untuk melakukan cara lain yang disebutkan diatas. Kendala tersebut adalah tidak dapat melihat respons motor secara langsung karena tidak digunakannya sensor untuk mengukur kecepatan motor saat sistem berjalan. Selain itu tidak adanya model matematis dari motor membuat cara analitis sulit untuk dilakukan.

Menurut Smith (1979), untuk melakukan *hand tuning* prosedur yang dilakukan adalah sebagai berikut:

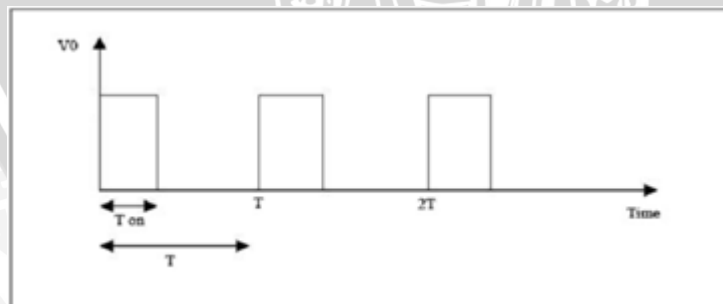
1. Melepaskan kontroler integral dan diferensial dengan memberikan nilai  $K_i = 0$  dan  $K_d = 0$ .
2. Mengatur nilai  $K_p$  hingga didapatkan respons yang diinginkan, dengan mengabaikan *offset* dari *setpoint*.
3. Dengan terus menaikkan nilai  $K_p$ , nilai dari  $K_d$  dinaikkan untuk mengurangi *overshoot* yang terjadi.
4. Naikkan nilai  $K_i$  untuk mengurangi *offset*.

Keuntungan dari *hand tuning* adalah prosedur diatas dapat dilakukan dengan segera, *online* dan dapat melihat dengan cepat respons sistem setelah perubahan  $K_p$ ,  $K_i$  dan  $K_d$ . Kerugian dari cara ini adalah kesulitan untuk melihat apakah *setting* akhir dari kontroler merupakan nilai optimal atau tidak (Jantzen, 2001).

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

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

Pada sinyal PWM, frekuensi sinyal konstan sedangkan *duty cycle* bervariasi dari 0%-100%. Dengan mengatur *duty cycle* akan diperoleh keluaran yang diinginkan. Sinyal PWM (*Pulse Width Modulation*) secara umum dapat dilihat dalam Gambar 2.13 berikut:



Gambar 2.13 Gambar Sinyal PWM Secara Umum

Sumber: electronics-scheme.com

$$Dutycycle = \frac{T_{on}}{T} \times 100\% \dots (\%) \dots \dots \dots (2-13)$$

Dengan:

- Ton = Periode logika tinggi
- T = Periode keseluruhan

$$V_{dc} = Dutycycle \times V_{cc} \dots (V) \dots \dots \dots (2-14)$$

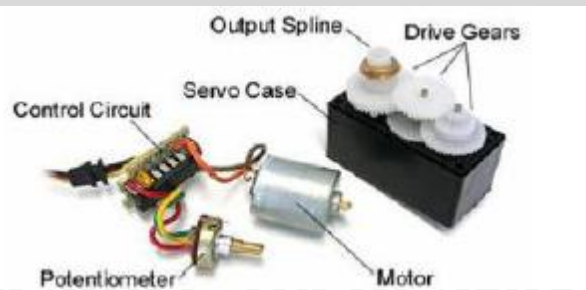
Sedangkan frekuensi sinyal dapat ditentukan dengan rumus berikut:

$$f_{0n} = \frac{f_{clk} I/O}{N.256} \dots (Hz) \dots \dots \dots (2-15)$$

### 2.4 Motor Servo

Motor *servo* adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor *servo*. Motor ini terdiri dari sebuah motor DC, serangkaian gear, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran *servo*. Sedangkan sudut dari sumbu motor *servo* diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Seperti yang kita tahu bahwa *servo* terdiri dari rangkaian pengontrol, gear, potensiometer dan DC motor. Potensiometer terhubung dengan gear demikian pula DC motor. Ketika DC motor diberi signal oleh rangkaian pengontrol maka dia akan bergerak demikian pula potensiometer dan otomatis akan mengubah resistansinya. Rangkaian pengontrol akan mengamati perubahan resistansi dan ketika resistansi mencapai nilai yang diinginkan maka motor akan berhenti pada posisi yang diinginkan.



Gambar 2.14 Bentuk Motor Servo  
Sumber: Am R., 2011

## 2.6 Mikrokontroler

Mikrokontroler populer yang pertama dibuat oleh Intel pada tahun 1976, yaitu mikrokontroler 8-bit Intel 8748. Mikrokontroler tersebut adalah bagian dari keluarga mikrokontroler MCS-48. Sebelumnya, Texas *instruments* telah memasarkan mikrokontroler 4-bit pertama yaitu TMS 1000 pada tahun 1974. TMS 1000 yang mulai dibuat sejak 1971 adalah mikrokomputer dalam sebuah *chip*, lengkap dengan RAM dan ROM.

Pengendali mikro (*microcontroller*) adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah *chip*. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O.

Berbeda dengan CPU serba-guna, mikrokontroler tidak selalu memerlukan memori eksternal, sehingga mikrokontroler dapat dibuat lebih murah dalam kemasan yang lebih kecil dengan jumlah *pin* yang lebih sedikit.

Sebuah *chip* mikrokontroler umumnya memiliki fitur:

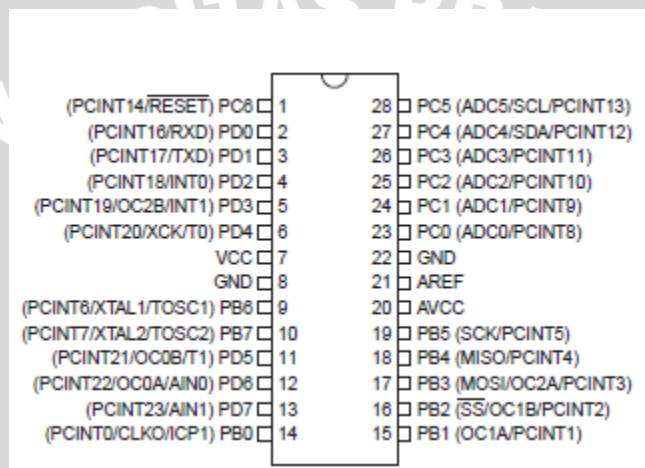
- a) *central processing unit* - mulai dari prosesor 4-bit yang sederhana hingga prosesor kinerja tinggi 64-bit.
- b) input/output antarmuka jaringan seperti *port serial* (UART)
- c) antarmuka komunikasi serial lain seperti I<sup>2</sup>C, *Serial Peripheral Interface* and *Controller Area Network* untuk sambungan sistem
- d) periferal seperti *timer* dan *watchdog*
- e) RAM untuk penyimpanan data
- f) ROM, EPROM, EEPROM atau *Flash memory* untuk menyimpan program komputer
- g) pembangkit *clock* - biasanya berupa resonator rangkaian RC
- h) pengubah analog-ke-digital

### 2.6.1 Mikrokontroler ATmega328

Atmel ATmega328 adalah mikrokontroler CMOS 8-bit berdaya rendah berbasis AVR yang arsitektur RISCnya telah ditingkatkan. Hampir semua instruksi dieksekusi dalam satu siklus *clock*, mempunyai *throughput* mendekati 1 MIPS per

MHz membuat desainer sistem untuk mengoptimasi konsumsi daya versus kecepatan proses.

Atmel ATmega328 memiliki beberapa fitur antara lain 8K *bytes In-system Programmable Flash with Read-While-Write*, 1K *bytes EEPROM*, 2K *bytes SRAM*, 23 jalur I/O untuk tujuan umum, 32 *working registers* untuk tujuan umum, tiga *timer/counter* yang fleksibel dengan *compare mode*, internal dan *external interrupt*, sebuah *serial programmable USART*, sebuah *byte-oriented 2-wire Serial Interface*, sebuah port SPI serial, sebuah *6-channel 10-bit ADC*, sebuah *Watchdog Timer* yang *programmable* dengan internal osilator. Seperti dalam Gambar 2.15.

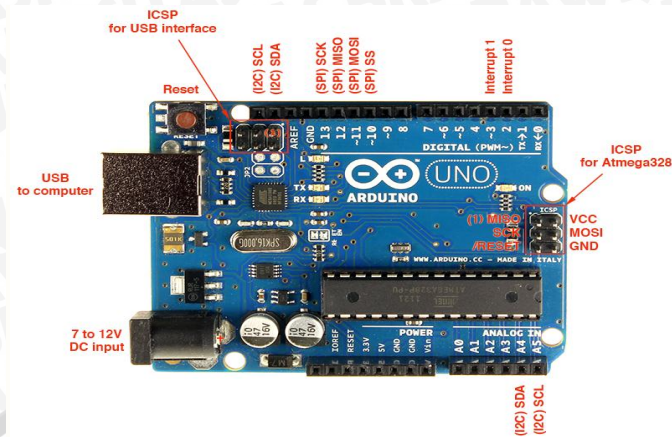


Gambar 2.15 Konfigurasi *Pin* ATmega328

Sumber: [www.atmel.com](http://www.atmel.com)

## 2.7 Arduino Uno

Arduino Uno adalah sebuah board mikrokontroler berbasis ATmega328. Arduino Uno mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino Uno memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.



Gambar 2.16 : Skema Arduino Uno  
Sumber : electroschematics.com

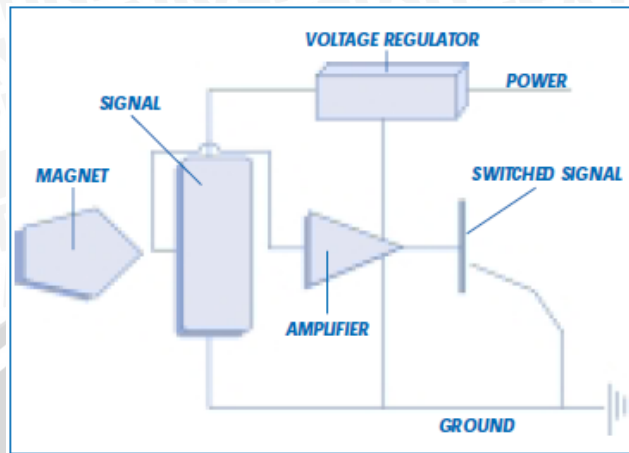
## 2.8 Sensor Hall effect

*Sensor hall effect* yang digunakan oleh banyak distributor sistem pengapian untuk memicu sirkuit primer (*fire the coil*) dan memantau mesin rpm .

*Sensor hall effect* terkadang dikenal dengan "switch" daripada sensor tegangan karena sinyal yang dihasilkan adalah digital (on-off). Tidak seperti sensor magnetik yang menghasilkan arus bolak-balik (AC) sinyal tegangan yang bervariasi dengan kecepatan, *Sensor hall effect* menghasilkan sinyal tegangan konstan yang bisa berubah tiba-tiba dari tegangan maksimum hampir nol dan kembali lagi terlepas dari kecepatan mesin. Hal ini menghasilkan gelombang sinyal keluaran kotak yang mudah digunakan oleh komputer untuk tujuan pewaktuan .

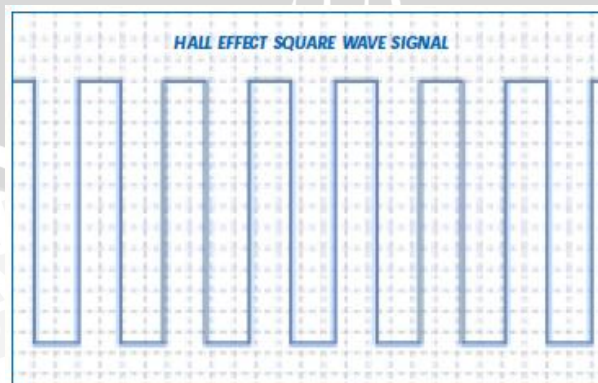
Tipikal *Sensor hall effect* memiliki tiga kabel atau terminal yaitu *ground*, suplai atau tegangan referensi, dan sinyal *output*. Untuk menghasilkan sinyal *output*, *Sensor hall effect* harus dipasang dengan tegangan referensi dari sumber (yang mungkin 5 sampai 12 volt tergantung pada aplikasi). Tegangan suplai diperlukan untuk menciptakan efek beralih yang berlangsung di dalam sensor. Prinsip kerja pada *Sensor hall effect* bermula pada tahun 1879 ketika Edwin H. Hall, seorang ilmuwan Amerika, menemukan sebuah fenomena listrik baru. Ketika saya menerapkan arus listrik untuk sepotong logam dimasukkan antara dua magnet, menciptakan tegangan sekunder dalam logam di sudut kanan dengan tegangan yang diberikan. Penemuan ini tidak memiliki banyak penggunaan praktis pada saat itu, tetapi dimasa depan ternyata insinyur akan perlu untuk membuat perangkat *switching* yang mampu menghasilkan sinyal tegangan kotak yang efisien. *Hall Effect* diadaptasikan sehingga

tegangan akan berubah ketika sebuah chip silikon ditempatkan di sudut kanan medan magnet.



**Gambar 2.17 : Diagram Sensor hall effect**  
Sumber : WELLS Manufacturing Corp, 1999

Ketika pisau logam melewati medan magnet celah udara dan diantara chip silikon, blok medan magnet dan menyebabkan tegangan output chip turun menjadi nol secara tiba-tiba. Dengan sirkuit tambahan, sensor dapat dibuat untuk melakukan hal yang berlawanan yaitu untuk menghasilkan sinyal tegangan ketika pisau Memasuki medan magnet. Sirkuit tambahan mengatur tegangan suplai ke chip dan menguatkan tegangan output. Dalam sistem pengapian otomotif, pisau rana (*shutter blade*) yang dipasang pada poros distributor, rotor, crankshaft pulley, atau camshaft sehingga sensor dapat menghasilkan memicu dan/ atau sinyal posisi sebagaimana crankshaft berputar. Dalam beberapa aplikasi, lekukan dalam katrol, gigi-gigi atau bahkan tombol magnet berputar bertujuan yang sama seperti pisau rana untuk mengganggu jendela magnet sensor dan saklar.



**Gambar 2.17 : Gelombang Sinyal Kotak Hall Effect**  
Sumber : WELLS Manufacturing Corp, 1999