

## BAB II

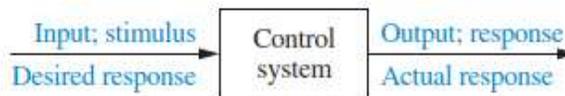
### TINJAUAN PUSTAKA

#### 2.1. Landasan Teori

##### 2.1.1. Sistem Kendali

Sistem kendali terdiri dari sub sistem dan proses (*plant*) yang disusun untuk tujuan memperoleh *output* dan kinerja yang diinginkan dengan *input*

yang telah ditentukan (Nise, 2015).



Gambar 2. 1 Deskripsi sederhana sistem kendali

Sumber : (Nise, 2015)

##### 2.1.1.1. Sistem Kendali *Openloop*

Sistem kendali *openloop* (*loop* terbuka) *output* sistem tidak berpengaruh pada aksi kendali disebut sistem kendali *loop* terbuka. Dengan kata lain, dalam sistem kendali *loop* terbuka, *output* tidak diukur atau diumpan balik untuk dibandingkan dengan *input*. Dalam sistem kendali *loop* terbuka, *output* tidak dibandingkan dengan *input* referensi. Dengan demikian, untuk setiap *input* referensi ada kondisi operasi yang tetap; sebagai hasilnya, keakuratan sistem tergantung pada kalibrasi. Dengan adanya gangguan, sistem kendali *loop* terbuka tidak akan melakukan tugas yang diinginkan. Kendali *loop* terbuka dapat digunakan, dalam

praktiknya, hanya jika hubungan antara *input* dan *output* diketahui dan jika tidak ada gangguan internal maupun eksternal (Ogata, 2010).

Sistem kendali *loop* terbuka memiliki keuntungan karena relatif sederhana dan relatif terjangkau dengan keandalan yang baik. Namun demikian, sistem ini sering kali tidak akurat karena tidak ada koreksi untuk kesalahan pada *output* yang mungkin diakibatkan oleh gangguan luar (Bolton, 2021).

Sistem kendali *loop* terbuka ini memiliki tiga unsur dasar yaitu kendali (*control*), koreksi (*correction*), dan proses (*process*) di mana suatu variabel dikendalikan (Bolton, 2021).

1. Unsur kendali (*Control elemen*)

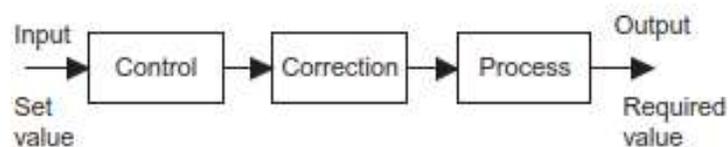
Ini menentukan tindakan yang akan diambil sebagai hasil dari *input* sinyal nilai yang diperlukan ke sistem.

2. Unsur koreksi (*Correction elemen*)

Ini memiliki masukan dari pengontrol dan memberikan keluaran dari beberapa tindakan yang dirancang untuk mengubah variabel yang dikendalikan.

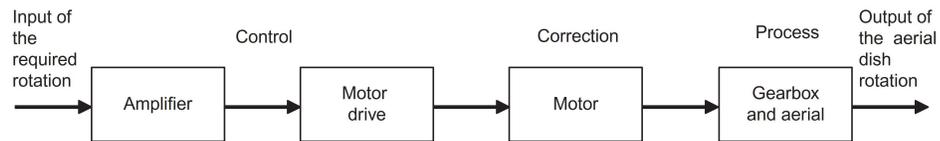
3. Proses (*Process*)

Ini adalah proses dimana sebuah variabel dikendalikan.



Gambar 2. 2 Unsur dasar sebuah sitem kendali *open-loop*

Sumber : (Bolton, 2021)



Gambar 2. 3 Deskripsi sederhana sistem kendali

Sumber : (Nise, 2015)

#### 2.1.1.2. Sistem Kendali *Closetloop*

Sistem kendali *closetloop* (*loop* tertutup) sistem ini memiliki hubungan yang diatur antara *output* dan *input* referensi dengan membandingkannya dan menggunakan perbedaannya sebagai alat kendali disebut sistem kendali umpan balik (Ogata, 2010). Sistem kendali umpan balik (*feedback*) sering disebut sebagai sistem kendali *loop* tertutup (*close loop*). Dalam praktiknya, istilah kendali umpan balik (*feedback*) dan *loop* tertutup digunakan secara bergantian. Dalam sistem kendali *loop* tertutup, kesalahan aktuasi sinyal, yang merupakan perbedaan antara sinyal *input* dan sinyal umpan balik (yang mungkin sinyal *output* itu sendiri atau fungsi dari sinyal *output* dan turunannya atau integral), diumpankan ke pengontrol untuk mengurangi kesalahan dan menghasilkan *output* dari sistem ke nilai yang diinginkan, istilah kendali *loop* tertutup selalu menyiratkan penggunaan tindakan kendali umpan balik untuk mengurangi kesalahan sistem (*error*) (Ogata, 2010).

Nilai kesalahan atau nilai *error* pada suatu sistem dapat didefinisikan sebagai berikut:

$$e = SV - PV \quad (2. 1)$$

Dimana,

- $e$  : Nilai kesalahan (*error*)
- $SV$  : Nilai set atau nilai keadaan yang ingin dicapai (*Set Value*)
- $PV$  : Nilai proses atau nilai hasil dari suatu proses (*Prosses Value*)

Unsur-unsur penyusun sistem kendali *loop* tertutup (*close-loop control systems*) (Bolton, 2021, p. 89) :

1. Unsur perbandingan (*comparison elemen*)

Unsur ini membandingkan nilai yang diperlukan dari variabel yang dikendalikan dengan nilai terukur dari apa yang dicapai dan menghasilkan sinyal kesalahan. Jadi, jika *outputnya* adalah nilai yang diperlukan maka tidak ada kesalahan sehingga tidak ada sinyal yang diumpankan untuk memulai kendali. Hanya jika ada perbedaan antara nilai yang diperlukan dan nilai aktual variabel, maka akan ada sinyal kesalahan dan tindakan kendali jalankan.

2. Unsur implementasi hukum kendali (*control law implementation element*)

Unsur hukum kendali menentukan tindakan apa yang harus diambil ketika sinyal kesalahan diterima. Hukum kendali yang digunakan oleh elemen mungkin hanya untuk memasok sinyal yang menyala atau mati ketika ada kesalahan, seperti pada termostat ruangan, atau mungkin sinyal yang sebanding dengan ukuran kesalahan sehingga jika kesalahan kecil, sinyal kendali kecil dihasilkan dan jika kesalahan besar, kendali proporsional besar sinyal diproduksi. Hukum kendali lainnya termasuk mode integral di mana sinyal kendali terus meningkat selama ada kesalahan

dan mode turunan di mana sinyal kendali sebanding dengan laju di mana kesalahan berubah.

Istilah unit kendali atau pengontrol sering digunakan untuk kombinasi elemen pembanding, yaitu detektor kesalahan, dan elemen implementasi hukum kendali. Contoh dari elemen semacam itu adalah diferensial penguat yang memiliki dua *input*, satu nilai yang ditetapkan dan satu sinyal umpan balik, dan perbedaan antara keduanya diperkuat untuk memberikan sinyal kesalahan. Ketika tidak ada perbedaan, tidak ada sinyal kesalahan yang dihasilkan.

### 3. Unsur koreksi (*correction element*)

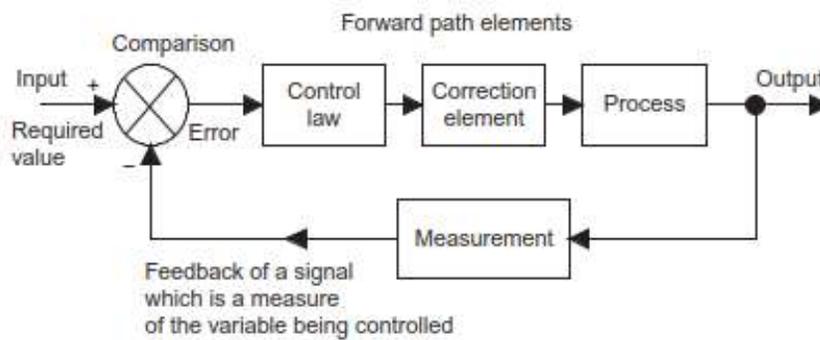
Unsur koreksi atau, seperti yang sering disebut, elemen kendali akhir, menghasilkan perubahan dalam proses yang bertujuan untuk mengoreksi atau mengubah kondisi yang dikendalikan. Istilah aktuator digunakan untuk unsur unit koreksi yang menyediakan daya untuk melakukan aksi kendali. Contoh elemen koreksi adalah katup kendali arah yang digunakan untuk mengalihkan arah aliran fluida dan dengan demikian mengontrol gerakan aktuator seperti gerakan piston dalam silinder. Contoh lainnya adalah motor listrik di mana sinyal digunakan untuk mengontrol kecepatan putaran poros motor.

### 4. Proses (*process*)

Proses adalah sebuah sistem yang di dalamnya terdapat sebuah variabel yang dikendalikan, misalnya sebuah ruangan di dalam rumah dengan variabel suhunya yang dikendalikan.

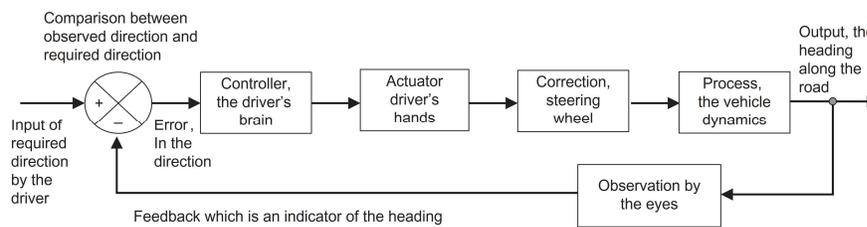
### 5. Unsur pengukuran (*measurement element*)

Unsur pengukuran menghasilkan sinyal yang terkait dengan kondisi variabel dari proses yang sedang dikendalikan. Sebagai contoh, ini mungkin sensor suhu dengan pemrosesan sinyal yang sesuai.



Gambar 2. 4 Unsur dasar sebuah sistem kendali *close-loop*

Sumber : (Bolton, 2021)

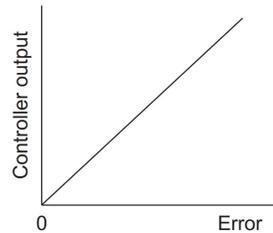


Gambar 2. 5 Contoh sebuah sistem kendali *close-loop*

Sumber : (Bolton, 2021)

#### 2.1.1.3. Kendali Proporsional (P)

Dengan kendali proporsional, ukuran *output* pengendalian sebanding dengan ukuran kesalahan, yaitu *input* pengendalian (Bolton, 2021). Dapat di gambarkan pada Gambar 2. 6.



Gambar 2. 6 Grafik *ouput* kendali proporsional terhadap *error*

Sumber : (Bolton, 2021)

Dengan demikian dapat kita diketahui bahwa keluaran pengontrol berbanding lurus dengan masukan pengontrol. Secara matematis kendali proporsional dapat ditulis dalam persamaan:

$$u(t) = K_p \cdot e(t) \quad (2. 2)$$

Dimana,

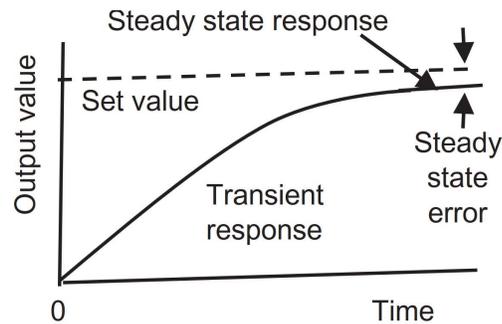
$u(t)$  : *output* kendali

$K_p$  : Konstanta proporsional

$e(t)$  : *Error*

Dimana  $K_p$  adalah konstanta yang disebut penguatan (*gain*). Ini berarti elemen koreksi dari sistem kontrol akan memiliki *input* sinyal yang sebanding dengan ukuran koreksi yang diperlukan. Jadi, misalnya, jika kita memiliki pemanas yang akan dikontrol untuk menghasilkan suhu dengan nilai *set point* (SV) tertentu, maka saat suhu mendekati *set point* (SV), daya yang disuplai ke pemanas akan berkurang dan memperlambat pemanasan sehingga tidak akan mengakibatkan suhu melebihi *set point* (SV) dan akhirnya beresilasi di sekitar titik *set point* hingga mencapai keadaan tunak (*steady state*).

kendali proporsional digunakan untuk mempercepat respons sistem menuju ke keadaan *set point* (SV), tetapi akan menghasilkan kesalahan pada keadaan tunak (*steady-state error*) atau bisa dikatakan *offset* dari *set point*.

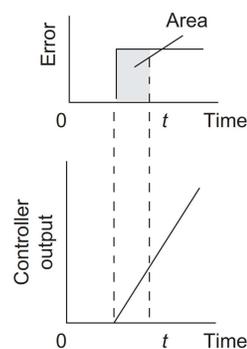


Gambar 2. 7 Grafik *output* kendali proporsional

Sumber : (Bolton, 2021)

#### 2.1.1.4. Kendali Integral (I)

Kendali integral merupakan pengendalian dimana hasil sinyal kendali (*output*) berbanding lurus dengan integral kesalahan terhadap waktu (Bolton, 2021, p. 112). Grafik kendali integral dapat digambarkan pada Gambar 2. 8.



Gambar 2. 8 Grafik *output* kendali Integral kesalahan terhadap waktu

Sumber : (Bolton, 2021)

Pada Gambar 2. 8 menggambarkan grafik integral kesalahan terhadap waktu. Nilai integral pada suatu waktu ( $t$ ) adalah luas daerah pada grafik antara  $t = 0$  dan  $t$ . Secara matematis kendali integral dapat ditulis dalam persamaan:

$$u(t) = K_I \int_0^t e(t) . dt \quad (2. 3)$$

Dimana,

$u(t)$  : *output* kendali

$K_I$  : Konstanta integral

$e(t)$  : *Error*

$dt$  : integral terhadap waktu

Kendali integral berfungsi untuk mengukur dan menghilangkan kesalahan pada keadaan tunak (*steady-state error*) yang terjadi pada respons kendali proporsional yang telah dalam keadaan tunak (*steady-state*). Kendali integral (I) biasanya tidak digunakan sendiri tetapi umumnya dalam hubungannya dengan kendali proporsional (P) (Bolton, 2021, p. 112) . Ketika aksi integral ditambahkan ke sistem kontrol proporsional, *output* pengontrol dapat dinyatakan dalam persamaan:

$$u(t) = K_p . e(t) + K_i \int_0^t e(t) . dt \quad (2. 4)$$

Dimana,

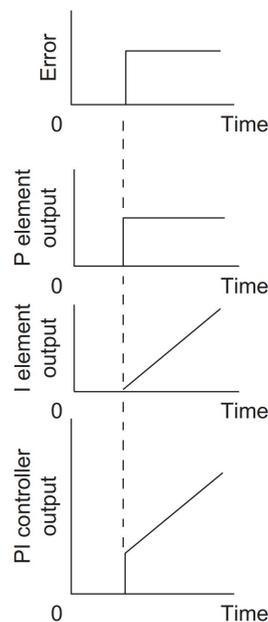
$u(t)$  : *output* kendali

$K_I$  : Konstanta integral

$K_p$  : Konstanta proporsional

$e(t)$  : *Error*

$dt$  : integral terhadap waktu



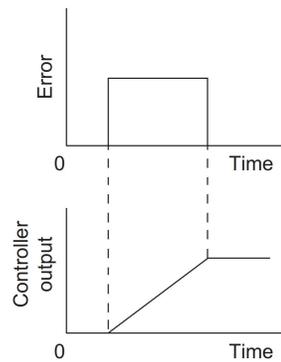
Gambar 2. 9 Grafik *output* kendali PI

Sumber : (Bolton, 2021)

Gambar 2. 9 menunjukkan bagaimana sistem dengan kendali PI bereaksi ketika ada perubahan mendadak pada kesalahan konstan. Kesalahan tersebut menghasilkan *output* kendali proporsional yang tetap konstan karena kesalahan tidak berubah. Kemudian ditambahkan ke dalam *output* pengontrol yang terus meningkat karena aksi integral.

Kombinasi mode integral dengan mode proporsional memiliki satu keuntungan besar dibandingkan mode proporsional saja yaitu *steady-state error* dapat dihilangkan. Hal ini karena bagian integral dari kontrol dapat memberikan

*output* pengontrol bahkan ketika kesalahan adalah nol. *Output* pengontrol adalah jumlah dari area sampai ke waktu  $t = 0$  dan dengan demikian bahkan ketika kesalahan telah menjadi nol, kendali akan memberikan *output* karena kesalahan sebelumnya dan dapat digunakan untuk mempertahankan kondisi tersebut. Gambar 2. 10 mengilustrasikan hal ini.

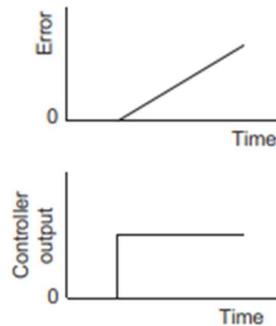


Gambar 2. 10 Grafik *output* kendali PI ketika *error* menjadi nol

Sumber : (Bolton, 2021)

#### 2.1.1.5. Kendali Derivatif (D)

Dengan kendali derivatif, perubahan dalam *output* kendali dari nilai *set point* sebanding dengan laju perubahan dengan waktu sinyal kesalahan, dapat dikatakan *output* kendali berbanding lurus dengan laju perubahan kesalahan (Bolton, 2021). Dapat digambarkan pada.



Gambar 2. 11 Grafik *ouput* kendali derivatif terhadap *error*

Sumber : (Bolton, 2021)

*Output* yang dihasilkan merupakan turunan atau diferensial dari kesalahan (*error*) terhadap waktu. Secara matematis dapat ditulis dalam persamaan:

$$u(t) = K_d \frac{de(t)}{dt} \quad (2. 5)$$

Dimana,

$u(t)$  : *output* kendali

$K_d$  : Konstanta diferensial atau turunan

$\frac{de(t)}{dt}$  : turunan kesalahan (*error*) terhadap waktu (t)

$K_d$  adalah konstanta proporsional dan biasanya disebut sebagai turunan waktu karena memiliki satuan waktu. Karena tingkat perubahan kesalahan dengan waktu adalah konstan, kendali derivatif memberikan sinyal keluaran kendali yang konstan (Bolton, 2021).

Kendali derivatif berfungsi untuk mengurangi respons yang terlalu berlebih yang dapat mengakibatkan *overshoot* pada proses pengendalian karena nilai  $K_p$  yang terlalu besar pada pengendali *propotional*.

Kendali derivatif memberikan respons terhadap sinyal kesalahan yang berubah, tetapi tidak merespons sinyal kesalahan yang konstan, karena dengan kesalahan yang konstan, laju perubahan kesalahan dengan waktu adalah nol. Karena itu, kontrol derivatif (D) dikombinasikan dengan kontrol proporsional (P). Secara matematis dapat ditulis dalam persamaan:

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt} \quad (2.6)$$

Dimana,

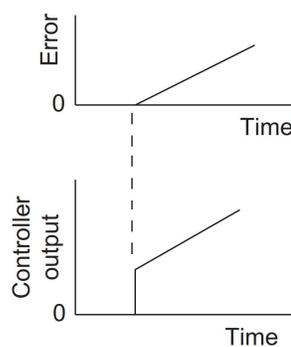
$u(t)$  : *output* kendali

$K_I$  : Konstanta integral

$K_p$  : Konstanta proporsional

$e(t)$  : *Error*

$dt$  : integral terhadap waktu



Gambar 2. 12 Grafik *output* kendali PD

Sumber : (Bolton, 2021)

Gambar 2.12 menunjukkan bagaimana respons kendali proporsional dengan kendali derivatif, keluaran pengontrol dapat bervariasi ketika ada kesalahan Click or tap here to enter text. yang terus berubah. Perubahan awal yang cepat pada kendali karena aksi derivatif yang diikuti oleh perubahan bertahap karena aksi proporsional. Dengan demikian, bentuk kontrol ini dapat menangani perubahan proses yang cepat dengan lebih baik daripada hanya kontrol proporsional saja. Namun, seperti sama halnya dengan kendali proporsional, diperlukan kesalahan kondisi tunak untuk mengatasi perubahan kondisi *input* yang konstan atau perubahan nilai yang ditetapkan (Bolton, 2021).

#### **2.1.1.6. Kendali PID (*Propotional, Integral, Derivative*)**

Sistem kendali PID telah banyak digunakan untuk kendali suhu pada proses industri karena hasilnya yang tepat dengan menghitung dan menghasilkan operasi korektif yang dapat mengatur proses dengan tepat, pengontrol PID digunakan untuk koreksi kesalahan antara variabel proses yang diukur dengan titik set yang ditentukan.

Sistem kendali PID merupakan sistem kendali *close loop* yang menggunakan pengukuran dari hasil proses kemudian dijadikan masukan untuk menjadi nilai koreksi pada sistem. Sistem kendali PID diperlukan untuk sebuah proses kendali yang pada prosesnya menginginkan nilai *output* yang konstan ketika ada nilai kesalahan (*error*).

Sistem kendali PID merupakan gabungan dari ketiga mode kendali (*propotional, integral, dan derivative*) memungkinkan pengontrol menjadi sangat baik yang tidak memiliki kesalahan kondisi tunak (*steady state*) dan mengurangi

kecenderungan untuk berosilasi. Pengendali seperti itu dikenal sebagai pengendali tiga mode atau kendali PID. Kendali PID dapat dianggap sebagai kendali proporsional yang memiliki kendali integral yang ditambahkan untuk menghilangkan kesalahan *offset* dan kendali derivatif untuk mengurangi jeda waktu (Bolton, 2021).

Secara matematis kendali Proporsional, Integral, dan Derivatif (PID) dapat ditulis dalam persamaan:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) \cdot d\tau + K_d \frac{de(t)}{dt} \quad (2.7)$$

atau,

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) \cdot d\tau + T_d \frac{de(t)}{dt} \right] \quad (2.8)$$

Dimana,

$u(t)$  : *output* kendali

$K_p$  : Konstanta / gain proporsional

$K_I$  : Konstanta integral ( $K_p/T_i$ )

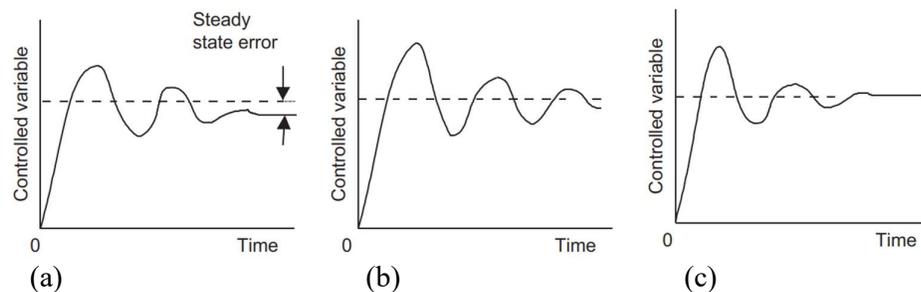
$K_d$  : Konstanta integral ( $K_p \cdot T_d$ )

$T_i$  : *Time integral*

$T_d$  : *Time derivative*

$e(t)$  : *Error*

Gambar 2. 13 menjelaskan respons sistem yang dapat terjadi dengan mode kendali yang berbeda ketika diberikan *input* step, yaitu perubahan mendadak ke *setvalue* konstan yang berbeda atau mungkin gangguan konstan yang tiba-tiba. Kendali proporsional (P) memberikan respons yang cepat dengan osilasi yang akan hilang dengan menghasilkan kondisi *steady-state error*. Kendali proporsional integral (PI) tidak memiliki *steady-state error* tetapi cenderung menunjukkan lebih banyak osilasi sebelum stabil. Kendali proporsional, integral, derivatif (PID) juga tidak memiliki *steady-state error*, karena adanya elemen integral, dan cenderung menunjukkan lebih sedikit osilasi daripada kendali PI dengan ditambahkan kendali derivatif akan mengurangi osilasi (Bolton, 2021).



Gambar 2. 13 Respons Sistem Kendali (a) P, (b) PI, (c) PID

Sumber : (Bolton, 2021)

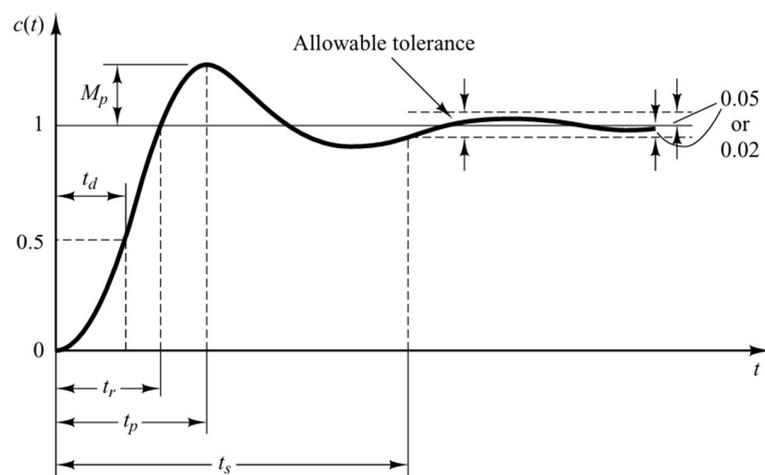
Respons dari setiap kendali proporsional, integral dan derivatif dapat disimpulkan pada Tabel 2. 1

Tabel 2. 1 Respons Kendali PID

Respons Kendali	Rise Time	Overshoot	Settling Time	Steady State Error
Proporsional	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan/ Mengurangi
Integral	Menurunkan	Meningkatkan	Meningkatkan	Menghilangkan
Derivatif	Perubahan Kecil	Menurunkan	Mengurangi	Perubahan Kecil

### 2.1.1.7. Spesifikasi Respons Waktu (Transien)

Sistem kendali sering kali ditentukan dalam hal respons transien terhadap respons suatu sistem ketika diberikan *input* berupa sinyal *unit-step*, karena mudah dibuat dan cukup signifikan (Jika respons terhadap *input step* diketahui, maka secara matematis dimungkinkan untuk menghitung respons terhadap *input* apa pun). Respons transien dari suatu sistem terhadap *input unit-step* tergantung pada kondisi awal dengan tujuan untuk memudahkan dalam membandingkan respons transien dari berbagai sistem, dalam praktiknya menggunakan kondisi awal standar bahwa sistem dalam keadaan diam pada awalnya dengan *output* dan semua turunan waktu nol. Kemudian karakteristik respons dari banyak sistem dapat dengan mudah dibandingkan. Respons transien dari sistem kontrol praktis sering kali menunjukkan osilasi teredam sebelum mencapai kondisi tunak (*steady state*). (Ogata, 2010)



Gambar 2. 14 Spesifikasi Respons Transien

Sumber : (Ogata, 2010)

Dari Gambar 2. 14 dalam menentukan karakteristik *respons transien* dari sistem kendali terhadap *input unit-step*, ditentukan sebagai berikut:

1. *Time constan (t)*

*Time constan* adalah ukuran waktu yang menyatakan kecepatan respons, diukur mulai  $t = 0$  s/d respons mencapai 63,2% dari respons *steady state*.

2. *Rise time (tr)*

*Rise time* adalah ukuran waktu yang menyatakan keberadaan suatu respons, yang diukur mulai respons 5% s/d 95% , 10%/s/d 90% atau 0% s/d 100%

3. *Settling time (ts)*

*Settling time* adalah ukuran waktu yang menyatakan respons telah masuk  $\pm 0,5%$ ,  $\pm 2%$ , atau  $\pm 5%$  dari respons *steady state*

4. *Delay time (td)*

*Delay time* adalah ukuran waktu yang menyatakan faktor keterlambatan respons *output* terhadap *input*, diukur mulai  $t = 0$  s/d respons mencapai 50% dari respons *steady state*.

5. *Maksimum Overshoot (Mp)* atau *overshoot*

*Maximum overshoot* atau *overshoot* adalah nilai puncak dari respons dalam bentuk persen.

6. *Time peak (tp)*

*Time peak* adalah ukuran waktu yang diukur mulai dari  $t = 0$  sampai mencapai nilai puncak maksimum.

Spesifikasi respons *steady state*, adalah spesifikasi respons sistem yang diamati mulai saat respons masuk dalam keadaan *steady state* sampai waktu tak terbatas

(dalam praktik waktu pengamatan dilakukan saat  $T_s \leq t \leq 5T_s$ ). Tolak ukur yang digunakan untuk mengukur kualitas *respons steady state* ini antara lain; *error steady state*.

### **2.1.2. Electric Muffle Furnace**

Pemanas listrik adalah salah satu peralatan yang banyak digunakan pada kehidupan sehari-hari, prinsip kerjanya adalah dengan menggunakan suatu elemen pemanas yang dialiri oleh arus listrik. Kemudian, energi listrik berubah menjadi energi panas yang terjadi pada elemen pemanas. Panas yang timbul disebabkan karena material elemen pemanas terbuat dari logam yang memiliki resistansi yang tinggi. Prinsip kerja tungku pemanas listrik adalah memanfaatkan prinsip induksi untuk memanaskan logam hingga titik leburnya. Pemanasan tersebut diterapkan pada pemanasan induksi medium konduktif (biasanya logam) dan tidak menutup kemungkinan pada material logam selain baja (Purwanto and Nasa, 2021).

Ada sejumlah metode yang digunakan untuk menghantarkan energi listrik ke material, yaitu pemanasan melalui resistansi, busur api (*arc*), induksi, dielektrik, dan inframerah (Mullinger and Jenkins, 2008).

#### **2.1.2.1. Kawat pemanas**

*Electric muffle furnace* pada penelitian ini proses pemanasannya terjadi akibat dari aliran listrik yang mengalir melalui bahan konduktor menimbulkan timbulnya panas pada bahan karena hambatan listriknya. Menurut George Simon Ohm pada Tahun 1826 mengemukakan bahwa kuat arus ( $I$ ) yang mengalir pada kawat penghantar sebanding dengan beda potensial ( $V$ ) antara ujung-ujung kawat penghantar dan berbanding terbalik dengan hambatan ( $R$ ) suatu penghantar.

Hukum Ohm secara matematis dapat dinyatakan dalam persamaan sebagai berikut:

$$V = I \cdot R \quad (2.9)$$

$$R = \frac{V}{I} \quad (2.10)$$

$$I = \frac{V}{R} \quad (2.11)$$

Dimana,

$V$  : Beda potensial listrik/ tegangan listrik dalam satuan *Volt (V)*

$R$  : Hambatan listrik dalam satuan *Ampere (A)*

$I$  : Hambatan listrik dalam satuan *Ohm ( $\Omega$ )*

Pada setiap material memiliki hambatan jenis berbeda-beda Tabel 2. 2, pada pemanas material yang digunakan untuk melepas suatu energi adalah Ni-Cr. Hambatan suatu konduktor bergantung pada ukuran dan bahan pembuatnya (Duncan, 2000).

Tabel 2. 2 Resistansi jenis material pada 20°C

Sumber : (Duncan, 2000)

Jenis Material	Nilai Resistansi ( $\Omega/meter$ )	Penggunaan
<b>A. KONDUKTOR</b>		
Perak	$1.6 \times 10^{-8}$	Kontak pada saklar kecil
Tembaga	$1,7 \times 10^{-8}$	Penghubung kabel /
Alumunium	$2,7 \times 10^{-8}$	Kabel listrik
Tungsten	$5,5 \times 10^{-8}$	Filamen lampu

Manganin (Cu-Mn-Ni)	$44 \times 10^{-8}$	Resistor standar presisi tinggi
<i>Constantan/ Eureka</i> (Cu-Ni)	$49 \times 10^{-8}$	Kotak resistansi, resistor variabel
Nikrom (Ni-Cr)	$110 \times 10^{-8}$	Elemen pemanas
Carbon	$3000 \times 10^{-8}$	Resistor radio
B. SEMIKONDUKTOR		
Germanium	0,6	Transistor
Silikon	2300	Transistor
C. INSULATOR		
Kaca	$10^{10} - 10^{14}$	
Polistiren	$10^{15}$	

Persamaan hambatan terhadap ukuran konduktor secara matematis dapat dinyatakan sebagai :

$$R = \rho \frac{l}{A} \quad (2.12)$$

Dimana,

$R$  : Hambatan yang dimiliki bahan dalam satuan Ohm ( $\Omega$ )

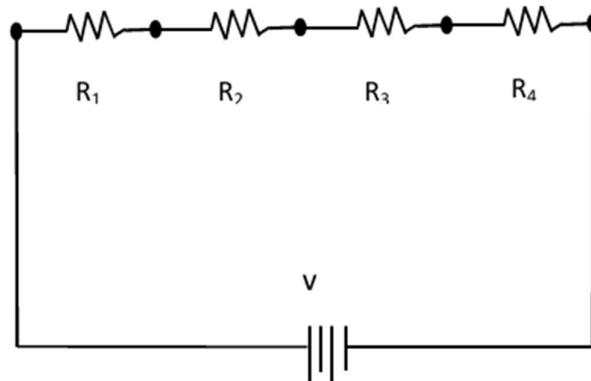
$\rho$  : Hambatan jenis ( $\Omega \text{ mm}^2/\text{m}$ )

$l$  : Panjang penghantar dalam satuan Meter (m)

$A$  : Luas penampang penghantar dalam satuan Milimeter persegi ( $\text{mm}^2$ )

Pengaruh diameter kawat berpengaruh terhadap daya yang digunakan. Semakin besar luas penampang (diameter kawat), maka resistansinya semakin kecil. Dalam Hukum Ohm, nilai arus dan resistansi berbanding terbalik, jika nilai resistansinya kecil, maka nilai arusnya besar. Sedangkan daya yang dikonsumsi berbanding lurus dengan arus yang dihasilkan (Vingtsabta, Syakur dan Warsito, 2019).

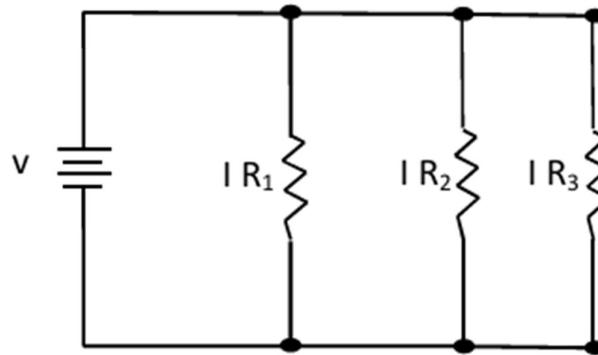
Karakteristik dari suatu arus dan tegangan pada suatu rangkaian memiliki ciri dan sifat yang khas, tergantung pada jenis rangkaianannya. Pada rangkaian seri Gambar 2. 15 terjadi pembagian tegangan sedangkan pada rangkaian paralel sebaliknya terjadi pembagian arus .Karakteristik dari rangkaian seri adalah tegangan dari sumber akan dibagi dengan jumlah tahanan yang dipasang pada cabang masing-masing. Sedangkan arus yang mengalir pada masing-masing beban adalah sama. Karakteristik dari rangkaian paralel Gambar 2. 16 adalah arus yang mengalir pada masing-masing cabang tergantung nilai tahanan yang dipasang pada cabang. Sedangkan tegangan pada masing-masing beban listrik sama dengan tegangan sumber. (Rosman *et al.*, 2020).



Gambar 2. 15 Gambar Rangkaian Seri

Untuk mengetahui nilai tahanan total dari rangkaian seri menggunakan persamaan :

$$\frac{1}{R_p} = R_1 + R_2 + R_3 + R_4 \quad (2. 13)$$



Gambar 2. 16 Gambar Rangkaian Paralel

Untuk mengetahui nilai tahanan total dari rangkaian paralel menggunakan persamaan :

$$\frac{1}{R_p} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4} \quad (2. 14)$$

Hambatan jenis dan konduktivitas suatu bahan merupakan sifat khas yang tidak dipengaruhi oleh ukuran dan bentuk bahan, tetapi dipengaruhi oleh perubahan suhu. Pada batas perubahan suhu tertentu, maka hambatan jenis suatu bahan memenuhi persamaan :

$$\rho_T = \rho_0 (1 + \alpha(\Delta T)) \quad (2. 15)$$

Dimana,

$\rho_T$  : Hambatan jenis pada suhu akhir ( $\Omega m$ )

$\rho_0$  : Hambatan jenis pada suhu awal ( $\Omega m$ )

$\alpha$  : Koefisien suhu hambatan jenis ( $^{\circ}C^{-1}$ )

$\Delta T$  : Perubahan suhu ( $^{\circ}C$ )

Bila sebatang logam dialiri arus listrik (akibat perbedaan potensial) pembawa muatan bertumbukan dengan atom logam mendapat energi sehingga menjadi panas dan atom-atom akan bergerak semakin kuat. Daya hilang yang diubah menjadi getaran atom dalam logam, dengan kata lain hilang sebagai kalor. Ini dapat dipahami bahwa muatan  $dq$  yang bergerak akan mendapat tambahan energi sebesar  $dU = (dq) V$ . Karena arus dan kecepatan tetap, maka energi yang hilang persatuan waktu (daya), adalah:

$$P = \frac{dU}{dt} = V \frac{dq}{dt} \quad (2.16)$$

$$P = I.V \quad (2.17)$$

$$P = \frac{W}{t} \quad (2.18)$$

$$P = \frac{V^2}{R} \quad (2.19)$$

Karena beda potensial,

$$V = I.R \quad (2.20)$$

Maka,

$$P = I^2.R \quad (2.21)$$

Dimana,

$P$  : Daya (Watt)

$V$  : Beda potensial (Volt)

$I$  : Kuat arus (Ampere)

$R$  : Hambatan (Ohm)

Pada persamaan diatas dikenal sebagai hukum Joule yang menyatakan daya yang hilang (daya disipasi) pada konduktor dengan hambatan ( $R$ ) dan dialiri arus ( $I$ ). Sedangkan besaran disipasi (kalor Joule) dalam waktu ( $t$ ) adalah:

$$W = Q = P . dt \quad (2. 22)$$

$$W = Q = I^2 . R . dt \quad (2. 23)$$

$$W = Q = \frac{V^2}{R} \quad (2. 24)$$

$$W = Q = V . I . dt \quad (2. 25)$$

Dimana,

$Q/W$  : Jumlah kalor (Joule)

$I$  : Kuat arus listrik (Ampere)

$R$  : Hambatan (Ohm)

$V$  : Beda potensial (Volt)

$dt$  : Waktu (detik)

James Prescott Joule pada tahun 1840 mengatakan bahwa panas yang dihasilkan oleh arus listrik yang mengalir melalui konduktor sama dengan hasil kali kuadrat dari arus yang dilalui, resistansi konduktor dan waktu yang dilalui oleh arus tersebut.

Menurut hukum kekekalan energi, energi listrik dapat berubah menjadi energi kalor. Begitu pula sebaliknya, energi kalor dapat berubah menjadi energi listrik. Energi kalor yang dibutuhkan oleh suatu benda atau suatu zat bergantung pada tiga hal. Tiga hal tersebut ialah massa ( $m$ ) zat atau benda tersebut, jenis benda atau zat tersebut, serta perubahan suhu ( $\Delta T$ ) yang terjadi pada benda tersebut. Hubungan tersebut dapat ditulis secara matematis sebagai persamaan:

$$W = Q \quad (2. 26)$$

$$Q = m. c. \Delta T \quad (2. 27)$$

Dimana,

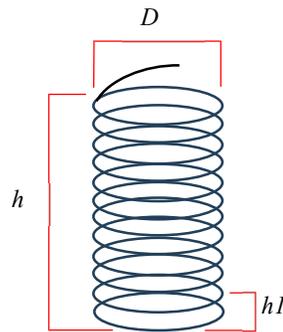
$Q$  : Jumlah kalor (Joule)

$m$  : Massa benda (m)

$c$  : Kalor jenis zat (J/kg°C)

$\Delta T$  : Perubahan suhu (°C)

Penempatan kawat pemanas berpengaruh terhadap efektifitas dalam melepaskan kalor. Berbagai macam bentuk penempatan kawat pemanas pada *furnace* salah satunya adalah memasang kawat dalam bentuk spiral Gambar 2. 17 pada dinding. Penggunaan dalam bentuk spiral untuk menekan ruang yang dihabiskan kawat, apabila peletakan dalam bentuk panjang maka akan menghabiskan ruang yang banyak. Untuk menghitung kebutuhan panjang kawat pada *furnace* maka dapat dihitung menggunakan persamaan (2. 28).



Gambar 2. 17 Bentuk spiral kawat

$$L = \sqrt{\left(\pi \frac{h}{h1} D\right)^2 + h^2} \quad (2. 28)$$

Dimana,

$L$  : Panjang kawat

$\pi$  : 3,14

$h$  : Panjang lilitan kawat

$h1$  : Jarak antar kawat

$D$  : Diameter lilitan

#### 2.1.2.2. Isolasi Pemanas

Seperti yang telah diketahui setiap bahan atau material memiliki karakteristik yang berbeda-beda, seperti sifat fisis, mekanis, dan kimiawi. Salah satu sifat fisis adalah kemampuan penghantar panas. Ada tiga prinsip dalam perpindahan panas, yaitu konduksi, konveksi, dan radiasi (Muin, Lazim dan Veranika, 2023). Nilai konduktivitas termal yang tinggi menunjukkan bahwa bahan tersebut merupakan konduktor panas yang baik, dan nilai yang rendah menunjukkan bahwa bahan

tersebut merupakan konduktor panas yang buruk atau bisa di sebut sebagai isolator (Cengel and Ghajar, 2014).

Salah satu material untuk menahan panas yang dihasilkan oleh *furnance* adalah batu bata api bahan penyusunnya merupakan campuran material refraktori.

Bahan penyusun pada produk refraktori, khususnya bata tahan api di antaranya

1. Lempung

Mengandung bahan *kaolinite*, *silica*, dan *illite* tergantung dari jenis material yang digunakan, selain itu banyak mengandung mineral besi.

2. Alumina

Dengan Rumus kimia  $AL_2O_3$ , banyak digunakan karena memiliki sifat penghambat panas yang baik hingga temperatur  $1580^{\circ}C$  tahan terhadap asam dan juga korosi, serta memiliki titik lebur tinggi hingga  $2072^{\circ}C$

3. Silika

Sumber alami silika dapat ditemukan pada kuarsa dan material lainnya. Sifat yang dimiliki oleh silika seperti tahan terhadap korosi dan asam, merupakan insulator yang baik, dan memiliki nilai penyusutan yang rendah (Pranaka, 2022).

Karakteristik yang dimiliki batu bata tahan api tersebut dapat dilihat pada

Tabel 2. 3 dan Tabel 2. 4

Tabel 2. 3 Spesifikasi bata *fireclay*

Sumber : (Duncan, 2000)

Spesifikasi	Tipe		
	SK-30	SK-32	SK-34
Desnistas ( $g/cm^3$ )	1,95	2,10	2,20
Porositas (%)	20-25	20-24	20-23

CCS (kg/cm <sup>2</sup> )	>250	>250	>300
Refractoriness (°C)	1300	1350	1400
Komposisi Kimia (%) :			
Al <sub>2</sub> O <sub>3</sub>	20-30	32-35	40-45
SiO <sub>2</sub>	65-75	60-65	45-55

Tabel 2. 4 Dasar Klasifikasi Bata Tahan Api

Sumber : (Duncan, 2000)

Bata Tahan Api	Al <sub>2</sub> O <sub>3</sub> (%)	SiO <sub>2</sub> (%)
<b>1. Fireclay Brick</b>		
<i>Low duty</i>	24-26	60-70
<i>Silica duty</i>	18-30	65-80
<i>Medium duty</i>	34-38	60-70
<i>High duty</i>	36-40	50-80
<i>Super duty</i>	40-44	49-53
<i>High-fired duty</i>	40-44	49-53
<b>2. High Alumina Brick</b>		
HA-50	50	
HA-60	60	
HA-70	70	
HA-80	80	
HA-85	85	
HA-90	90	
HA-99	99	

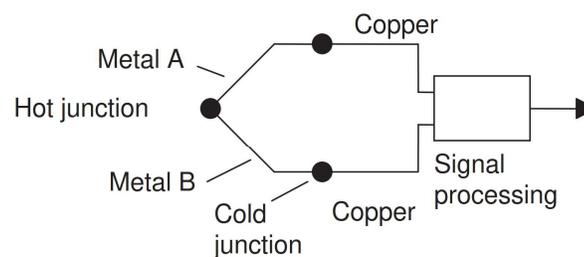
### 2.1.2.3. Temperatur Sensor

Istilah sensor digunakan untuk elemen yang menghasilkan sinyal yang berkaitan dengan kuantitas yang sedang diukur (Bolton, 2015a). Contohnya, pada suatu elemen suhu resistansi listrik, kuantitas yang diukur adalah suhu dan sensor mengubah *input* suhu menjadi perubahan resistansi.

Istilah transduser sering digunakan sebagai pengganti istilah sensor. transduser didefinisikan sebagai elemen yang ketika mengalami perubahan fisik mengalami perubahan terkait. dengan demikian sensor adalah transduser. namun, sistem pengukuran dapat menggunakan transduser, selain sensor, di bagian lain dari sistem untuk mengubah sinyal dalam satu bentuk ke bentuk lain (Bolton, 2015a).

Sebuah sensor atau transduser dikatakan analog jika memberikan *output* yang analog dan berubah secara kontinu dan biasanya memiliki keluaran yang ukurannya sebanding dengan ukuran variabel yang diukur. Istilah digital pada sensor digunakan jika sensor memberikan *output* yang bersifat digital, yaitu urutan sinyal *on/off* yang pada dasarnya menguraikan angka tertentu yang nilainya terkait dengan ukuran variabel yang diukur (Bolton, 2015a).

Termokopel (*thermocouple*) Gambar 2. 18 merupakan sensor suhu yang paling sering atau kebanyakan digunakan pada mesin *boiler*, mesin press, *furnace*, dan lain sebagainya. Termokopel dapat mengukur temperatur dalam jangkauan suhu yang cukup luas dengan batas kesalahan pengukuran kurang dari 1° C. Termokopel terdiri dari dua jenis kawat logam konduktor yang digabung pada ujungnya sebagai ujung pengukuran.



Gambar 2. 18 Termokopel

Sumber : (Bolton, 2015b)

Konduktor ini kemudian akan mengalami gradiasi suhu dan dari perbedaan suhu antara ujung termokopel/ ujung pengukuran dengan ujung kedua kawat logam konduktor yang terpisah akan menghasilkan tegangan listrik Hal ini disebut sebagai efek termoelektrik. Perbedaan ini umumnya berkisar antara 1 hingga  $70\mu\text{V}$  setiap perbedaan  $1^\circ\text{C}$  untuk kisaran yang dihasilkan dari kombinasi logam modern. Jadi sangat penting untuk diingat bahwa termokopel hanya mengukur perbedaan temperatur di antara 2 titik (Rahmat, 2015).

Tabel 2. 5 menjelaskan jenis, karakteristik dan bahan dasar material dari sensor termokopel.

Tabel 2. 5 Jenis termokopel dan karakteristiknya

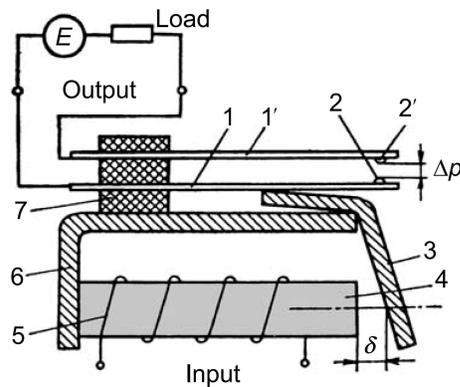
Sumber : (Bolton, 2015a)

Tipe	Material	Rentang ( $^\circ\text{C}$ )	( $\mu\text{V}/^\circ\text{C}$ )
B	<i>Platinum 30% Rhodium/platinum 6% rhodium</i>	0 s.d 1800	3
E	<i>Chromel/ constantan</i>	-200 s.d 1000	63
J	<i>Iron/ constantan</i>	-200 s.d 900	53
K	<i>Chromel alumel</i>	-200 s.d 1300	41
N	<i>Nirosil/ nisol</i>	-200 s.d 1300	28
R	<i>Paltinum/ Platinum 13% rhodium</i>	0 s.d 1400	6
S	<i>Paltinum/ Platinum 10% rhodium</i>	0 s.d 1400	6
T	<i>Copper/ constantan</i>	-200 s.d 400	43

#### 2.1.2.4. *Solid State Relay*

*Relay* adalah sakelar yang dioperasikan secara elektrik di mana perubahan arus dalam satu sirkuit listrik akan menghidupkan atau mematikan arus di sirkuit lainnya (Bolton, 2015a). Contohnya pada *electromagnetic relay*, Relay tersebut memutus dan mematikan arus pada *output* relay dengan cara mengatur medan elektro

magnetik yang terjadi pada lilitan yang dialiri arus. Sehingga, medan elektro magnetik tersebut menggerakkan dua buah kontak sehingga saling bersentuhan dan arus pun mengalir pada sisi *output* beban.

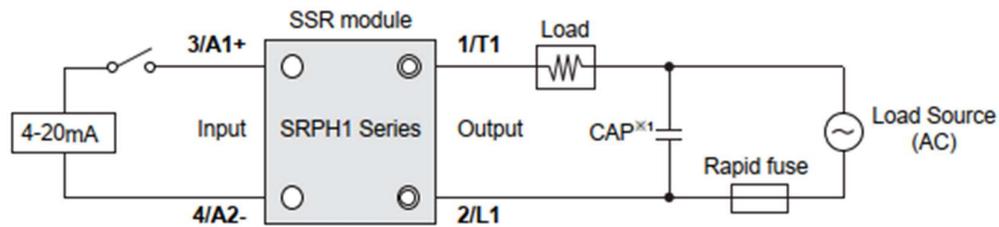


Gambar 2. 19 Kontruksi *electromagnetic relay* (1) pegas, (2) kontak, (3) *armature*, (4) inti, (5) lilitan, (6) inti magnet, (7) isolator

Sumber : (Gurevich, 2006)

Penggunaan *solid state relay* pada *furnace* berfungsi untuk mengendalikan arus listrik pada kawat pemanas Gambar 2. 20. Adapun komponen elektronik *solid-state* untuk sakelar sirkuit ini adalah komponen elektronik aktif berbahan semi konduktor seperti dioda, thyristor, triak, transistor dan MOSFET.

Komponen semi konduktor tersebut banyak digunakan pada sistem elektronika daya yang mengatur parameter-parameter listrik seperti arus, tegangan dan daya. Komponen semi konduktor pada sistem elektronika daya memiliki tiga fungsi utama yaitu pengsaklaran (*switching*), pengubahan (*converting*) dan pengontrolan (*controlling*). Fungsi tersebut tergantung dari karakteristik komponen aktif yang digunakan.



Gambar 2. 20 Rangkaian Penggunaan SSR

Sumber : (*Datasheet* SSR Autonics SRPH1 Series)

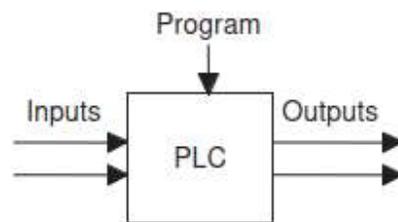
### 2.1.3. Programmable Logic Control (PLC)

Pengontrol logika yang dapat diprogram (PLC) adalah bentuk khusus pengontrol berbasis mikroprosesor yang menggunakan memori yang dapat diprogram untuk menyimpan instruksi dan untuk mengimplementasikan fungsi-fungsi seperti logika, pengurutan, pengaturan waktu, penghitungan, dan aritmatika untuk mengontrol mesin dan proses (Bolton, 2015b).

Alat ini dirancang untuk dioperasikan oleh para teknisi yang mungkin memiliki pengetahuan terbatas tentang komputer dan bahasa komputasi. Mereka tidak dirancang untuk mengatur atau mengubah program hanya oleh seorang pemrogram komputer. Dengan demikian, pembuat PLC telah menyiapkan programnya sehingga program kendali dapat dimasukkan dengan menggunakan bentuk bahasa yang sederhana, dengan bentuk bahasa yang cukup intuitif (Bolton, 2015b).

Istilah logika digunakan karena pemrograman terutama berkaitan dengan penerapan logika dan operasi *switching*; misalnya, jika A atau B terjadi, maka menyalakan C; jika A dan B terjadi, maka menyalakan D. Perangkat *input* (yaitu, sensor seperti sakelar) dan perangkat *output* (motor, katup, dan lainnya) pada sistem

yang dikendalikan tersebut terhubung ke PLC. Operator kemudian memasukkan urutan instruksi, program, ke dalam memori PLC. Pengontrol kemudian memonitor *input* dan *output* sesuai dengan yang ini program dan menjalankan aturan yang telah diprogram.(Bolton, 2015b)

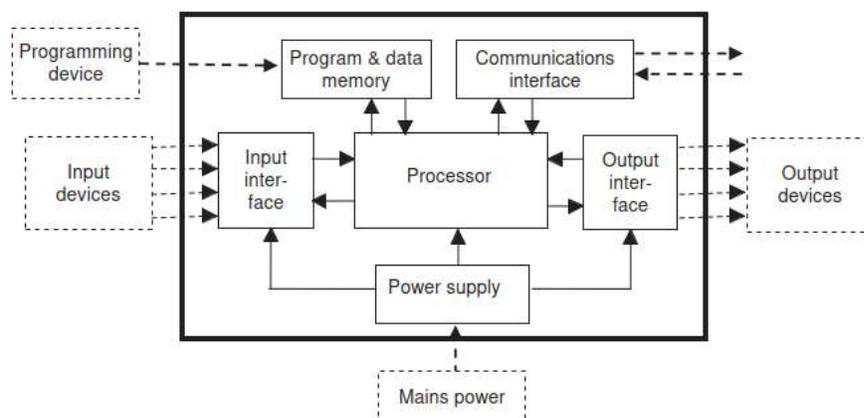


Gambar 2. 21 Sistem sebuah *programmable logic control*

Sumber : (Bolton, 2015b)

### 2.1.3.1. *Hardware (Perangkat Keras)*

Biasanya sebuah PLC tersusun dari beberapa komponen yaitu *prosesor unit, memory, power supply unit, input/output interface, communication interface,* dan *programmer device*. Pada umumnya PLC memiliki 6, 8, 12, atau 24 *input interface* dan 4, 8, atau 16 *output interface* dan memori yang dapat menyimpan data sekitar 300 hingga 1000 instruksi (Bolton, 2015b).



Gambar 2. 22 Susunan dasar sebuah sistem PLC

Sumber : (Bolton, 2015b)

### 2.1.3.2. *Processor Unit*

*Processor unit* atau *central processor unit* (CPU) adalah unit yang berisi sebuah mikroprosesor. Unit ini merubah sinyal *input* dan melaksanakan tindakan kontrol sesuai dengan program yang tersimpan pada memori dan mengkomunikasikan keputusan sebagai tindakan berupa sinyal ke *output*.

### 2.1.3.3. *Memori*

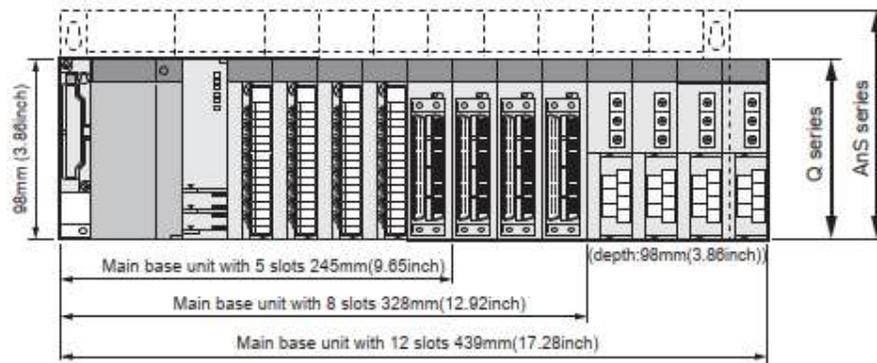
Memori unit dibutuhkan untuk menyimpan instruksi-instruksi atau program dan menyediakan lokasi penyimpanan untuk hasil-hasil perhitungan yang akan di proses oleh prosesor.

Kapasitas penyimpanan unit memori ditentukan oleh jumlah kata biner yang dapat disimpan. Dengan demikian, jika ukuran memori adalah 256 *words*, memori dapat menyimpan  $256 \times 8 = 2048$  bit jika menggunakan 8-bit *words* dan  $256 \times 16 = 4096$  bit jika menggunakan 16-bit *words*. Istilah *byte* digunakan untuk kata dengan panjang 8-bit. Ukuran memori sering ditentukan dalam hal jumlah lokasi penyimpanan yang tersedia, dengan 1K mewakili angka  $2^{10}$ , yaitu 1024 byte. Dengan demikian memori 4 Kbyte dapat menyimpan 4096 byte, memori 50 Kbyte menyimpan 51.200 byte.

### 2.1.3.4. *Power Supply*

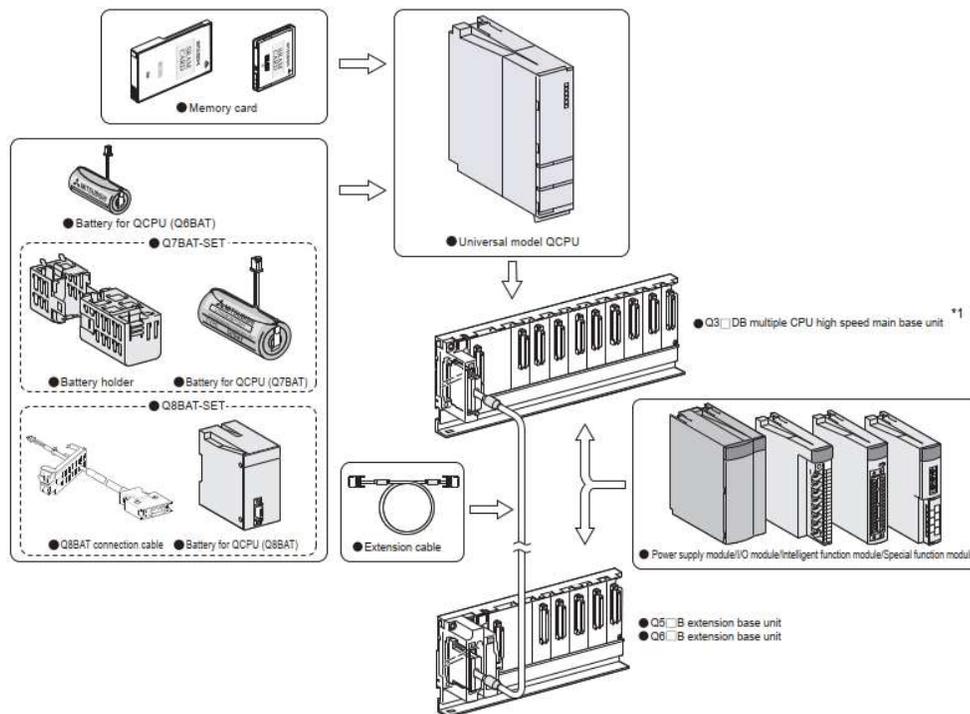
Power supply merupakan unit untuk merubah tegangan *input* AC menjadi DC sesuai kebutuhan tegangan pada PLC. Pada umumnya *power supply* untuk kebutuhan PLC terdapat dua tipe internal dan eksternal. Penggunaan *power supply* tergantung pada tipe PLC, pada tipe PLC modular Gambar 2. 24 menggunakan *power supply* eksternal dikarenakan bagian unit PLC terpisah. Sedangkan Untuk

tipe PLC *compact* Gambar 2. 25 menggunakan *power supplay* internal yang bagian unitnya satu kesatuan dengan PLC dan unit lainnya.



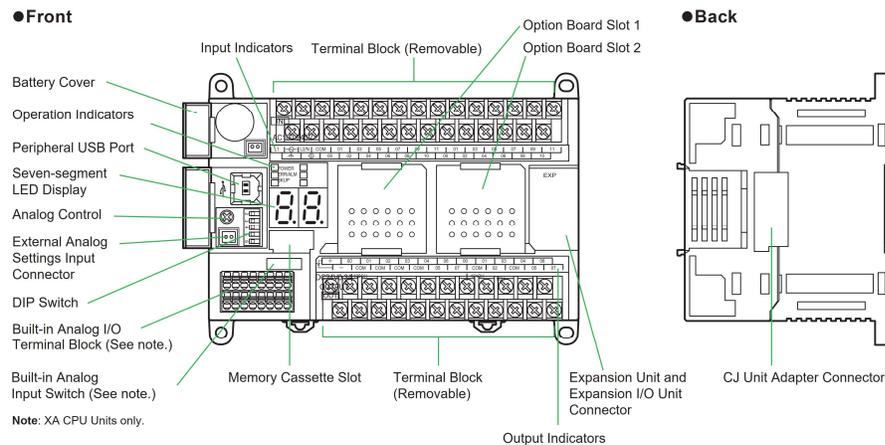
Gambar 2. 23 Antarmuka PLC Mitsubishi

Sumber : (Training manual book Mitsubishi)



Gambar 2. 24 Modul PLC Mitsubishi Q-series

Sumber : (Training manual book Mitsubishi)



Gambar 2. 25 Deskripsi CP1H

Sumber : (Nise, 2015, p. 7)

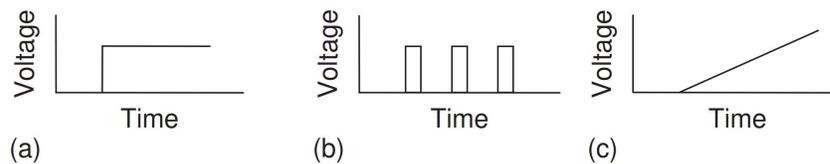
### 2.1.3.5. *Input / Output (I/O) PLC*

Bagian *input* dan *output* adalah tempat prosesor menerima informasi dari perangkat eksternal dan mengkomunikasikan informasi ke perangkat eksternal (Bolton, 2015b, p. 8). Adapun perangkat *input* yang dimaksud meliputi perangkat digital dan analog seperti sakelar mekanis untuk deteksi posisi, sensor jarak (*proximity*), sensor fotolistrik (*photoelectric*), pembuat encode (*encoder*), sensor suhu dan sensor tekanan, potensiometer, transformator diferensial variabel linier, pengukur regangan, termistor, termotransistor, dan termokopel. Perangkat *output* yang dipertimbangkan termasuk relay, kontaktor, katup solenoid, dan motor.

Perangkat *input* dan *output* dapat diklasifikasikan berdasarkan pemberian sinyal yang bersifat diskrit, digital, atau analog Gambar 2. 26. Perangkat yang memberikan sinyal diskrit atau digital adalah perangkat yang sinyalnya mati atau hidup. Jadi, sakelar berfungsi untuk memberikan sinyal diskrit, baik tanpa tegangan maupun bertegangan. Perangkat digital pada dasarnya dapat dianggap sebagai

perangkat diskrit yang memberikan deretan sinyal *on/off* secara terus menerus. (Bolton, 2015b, p. 8)

Perangkat analog memberikan sinyal berupa besaran yang proporsional dengan besaran variabel yang termonitor. Sebagai contoh, sensor suhu dapat memberikan tegangan sebanding dengan suhu yang sensor tersebut memberikan nilai resistansi yang berbeda-beda tergantung keadaan suhu sehingga menghasilkan tegangan yang berbeda (Bolton, 2015b, p. 8).

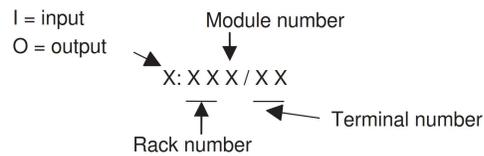


Gambar 2. 26 Bentuk Sinyal ; (a) diskrit (b) digital (c) analog

Sumber : (Bolton, 2015b)

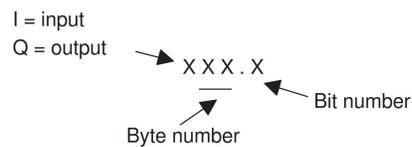
PLC harus dapat mengidentifikasi setiap *input* dan *output* tertentu. Ini dilakukan dengan mengalokasikan alamat (*address*) untuk setiap *input* dan *output*. Alamat *input* atau *output* (*addressing I/O*) berbentuk berupa angka, diawali dengan huruf untuk menunjukkan apakah itu *input* atau *output*. Addressing bertujuan untuk membedakan dan mengidentifikasi *input/output*, dikarenakan PLC memiliki jumlah I/O lebih dari satu.

Pada PLC Mitsubishi memiliki *input* dengan alamat X400, X401, X402, dan seterusnya dan *output* dengan alamat Y430, Y431, Y432, dan seterusnya, X menunjukkan *input* dan Y *output*. Contoh *address I/O* PLC dapat dilihat pada Gambar 2. 27 dan Gambar 2. 28 serta untuk implementasi pada program *ladder diagram* dapat dilihat pada Gambar 2. 39



Gambar 2. 27 Alamat I/O PLC pada Allen-Bradley

Sumber : (Bolton, 2015b)



Gambar 2. 28 Alamat I/O pada PLC Siemens

Sumber : (Bolton, 2015b)

#### 2.1.4. *Human-Machine Interface (HMI)*

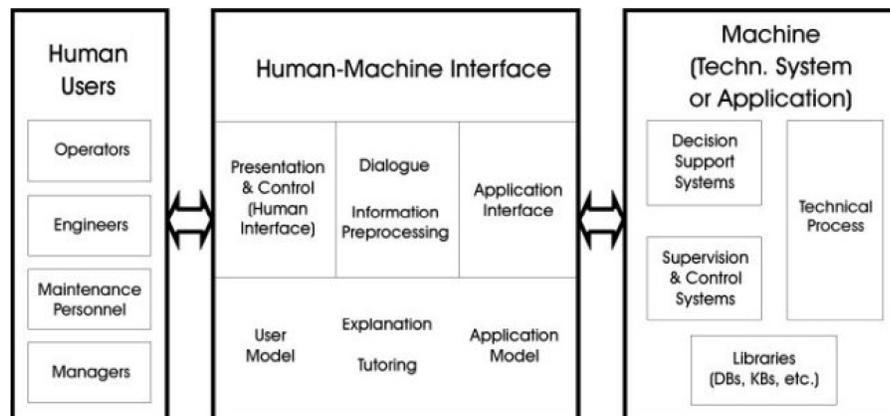
*Human Machine Interface* atau HMI merupakan sebuah sarana penghubung dan media komunikasi antara mesin dengan manusia. Sebagai media penghubung, tentu saja sistem HMI memiliki kemampuan untuk mengumpulkan dan mengolah data yang didapat dari mesin yang sedang dikendalikan menjadi sebuah informasi yang mudah dimengerti oleh manusia (Priswanto *et al.*, 2018). Selain menampilkan data, HMI juga dapat menggambarkan proses yang sedang berlangsung pada mesin yang dikontrol (Priswanto *et al.*, 2018). Hubungan antara HMI, pengguna (*user*), dan mesin, dapat dilihat pada Gambar 2. 29.



Gambar 2. 29 Hubungan pengguna, HMI dan mesin

Sumber : (Nise, 2015, p. 7)

Pada Gambar 2. 29 menjelaskan bahwa terjadi hubungan dua arah antara Pengguna (manusia) dengan HMI dan HMI dengan mesin. Hubungan dua arah terjadi ketika ketiga unsur tersebut saling bertukar data dan informasi.



Gambar 2. 30 Hubungan utaman HMI, user (manusia), dan mesin

Sumber : (Nise, 2015, p. 7)

Seperti yang terlihat pada Gambar 2. 30 HMI dibagi menjadi komponen untuk *Presentation & Control* (sebagai Human Interface), *Dialogue* dan *Information Preprocessing*, dan *Application Interface* sebagai komponen fungsi dasar dari sistem HMI, sedangkan *User Model*, *Explanation* dan *Tutoring*, dan *Application Model* adalah komponen fungsi lanjutan dari sistem HMI (Johannsen, 2019).

### 2.1.5. Pemograman PLC

Perangkat pemrograman dapat berupa perangkat genggam, konsol desktop, atau komputer. Hanya ketika program telah dirancang pada perangkat pemrograman dan siap, apakah itu ditransfer ke unit memori PLC. Perangkat pemograman terhubung ke PLC dengan Ethernet, RS-232, RS-485 atau RS-422

dengan menggunakan kabel. Berikut penjelasan mengenai macam perangkat pemrograman:

1. Perangkat pemrograman genggam (*handheld programming devices*)

Perangkat pemrograman genggam biasanya berisi memori yang cukup untuk memungkinkan unit untuk mempertahankan program saat dibawa dari satu tempat ke tempat lain.

2. Konsol desktop (*desktop Consoles*)

cenderung memiliki unit tampilan visual dengan *keyboard* dan layar penuh tampilan.

3. Komputer pribadi (*personal computers*)

Komputer pribadi banyak digunakan untuk memprogram PLC. Keuntungan utama menggunakan komputer adalah bahwa program dapat disimpan di *harddisk* atau CD dan salinannya dengan mudah dibuat.

Untuk dapat melakukan pemrograman pada PLC pada umumnya produsen PLC menyediakan *software* atau aplikasinya. Berikut merek dan tipe PLC beserta *software* pemrograman yang banyak digunakan di Indonesia:

1. Allen Bradley

Software : RS Logix

Type Logix-5 Family : PLC-5

Type Logix-500 Family : SLC-500 & Micrologix

Logix-5000 Family : ControlLogix, CompactLogix & FlexLogix

2. Siemens

Software : Simatic

Type Micro PLC : S7-200 & S7-1200

Type Modular PLC : S5-115U, S7-300 & S7-400

3. Omron

Software : CX-Programmer

Type Micro PLC : CPM1A, CP1E & CP1L

Type Basic PLC : CJ1M & CQM1H

Type Modular : CJ1H/CJ1G & CS1H/CS1G

4. Schneider

Software : PL7, ProWORX 32, Zelio Soft , So Machine dan Eco Structure

Type Micro PLC : Modicon M340

Type Machine Control PLC : Modicon Premium

Type Process Control PLC : Modicon Quantum

Type Programmable Controller : Twido

Type Smart Relay : Zelio Smart Relay

5. Mitsubishi

Software : MELSOFT series

Type Compact PLC : MELSEC FX3UC, MELSEC FX3G, MELSEC

FX1N & MELSEC FX1S

Type Modular PLC : Q-Series Q00UJCPU

Type Process Control : Q12PHCPU

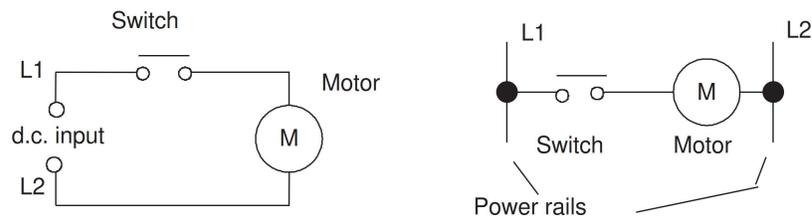
### 2.1.5.1. Bahasa Pemrograman

Program yang digunakan untuk pemrograman PLC dapat ditulis dalam berbagai bentuk. Bahasa pemrograman *ladder* dikembangkan untuk memudahkan para teknisi yang tidak memiliki pengetahuan yang baik tentang pemrograman dalam menulis program untuk PLC. Sebagian besar produsen PLC mengadopsi metode penulisan program ini namun, masing-masing cenderung mengembangkan versinya sendiri sehingga standar internasional mengadopsi pemrograman *ladder* menjadi metode yang digunakan untuk pemrograman PLC (Bolton, 2015b).

Menurut standar IEC 61131 (*International Electrotechnical Commission standart*) yang membahas tentang *Programmable Coltroller*, pada IEC 61131-3 menjelaskan tentang sebuah standar bahasa pemrograman yang harus ada pada PLC yaitu *ladder diagram* (LAD), *sequential function charts* (SFC), *function block diagram* (FBD), *structured text* (ST), dan *instruction list* (IL).

### 2.1.5.2. Ladder Diagram (LAD)

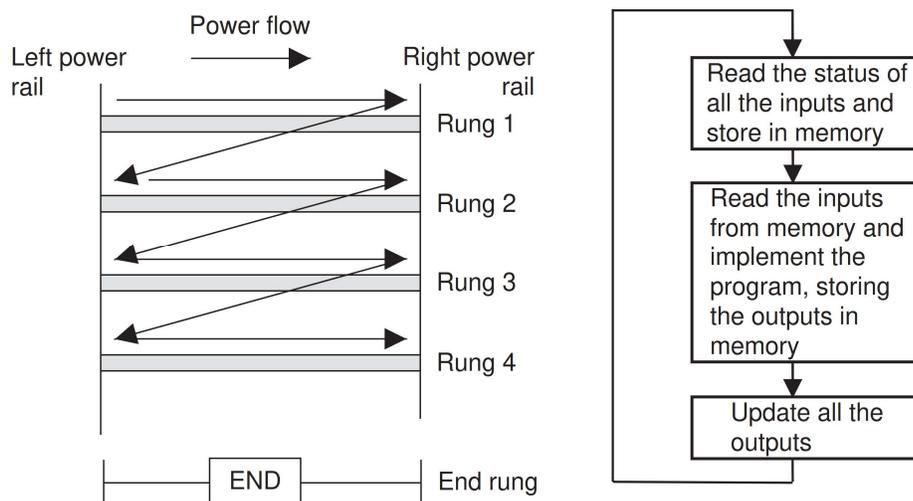
Menulis program diagram tangga (*ladder diagram*) setara dengan menggambar sirkuit *switching*. Diagram tangga terdiri dari dua garis vertikal yang mewakili rel daya. Sirkuit dihubungkan sebagai garis horizontal, yaitu anak tangga, di antara dua garis vertikal ini. Garis vertikal diagram mewakili rel daya di mana sirkuit terhubung. Aliran daya diambil dari vertikal sebelah kiri melintasi anak tangga. Program kemudian memulai kembali siklus dari awal. *Input* dan *output* semuanya diidentifikasi oleh masing-masing alamat (*address*). *Ladder diagram* dapat ditulis untuk menjalankan sistem logika AND, OR, NOT, NAND, NOR, dan XOR dengan menggunakan *input*. (Bolton, 2015b).



Gambar 2. 31 Rangkaian saklar listrik

Sumber : (Nise, 2015, p. 7)

Setiap anak tangga pada tangga pada *ladder diagram* mendefinisikan satu operasi dalam proses kontrol. Diagram tangga dibaca dari kiri ke kanan dan dari atas ke bawah. Gambar 2. 32 menunjukkan gerakan pemindaian yang digunakan oleh PLC. Anak tangga teratas dibaca dari kiri ke kanan. Kemudian anak tangga kedua dibaca dari kiri ke kanan dan seterusnya. Ketika PLC dalam mode jalankan, ia melewati seluruh program tangga sampai akhir, anak tangga akhir dari program tersebut dilambangkan dengan jelas, dan kemudian segera dilanjutkan dari awal.

Gambar 2. 32 Alur pembacaan program *ladder diagram*

Sumber : (Bolton, 2015b)

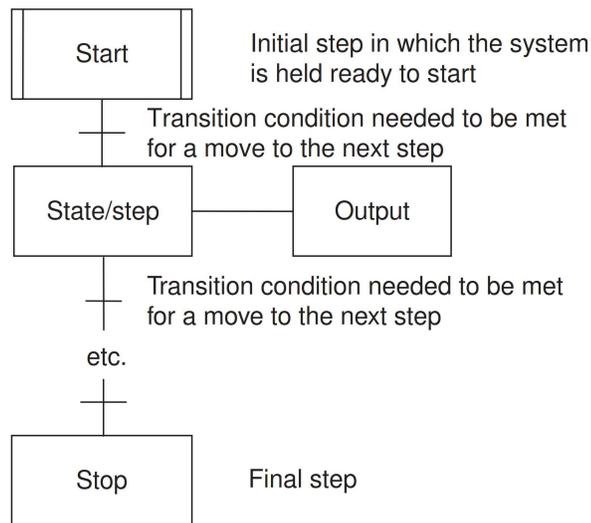
Prosedur melalui semua anak tangga program ini disebut siklus. Anak tangga akhir dapat ditunjukkan dengan blok dengan kata END atau RET, untuk kembali, karena program segera kembali ke awal. Waktu pemindaian tergantung pada jumlah program yang dijalankan, memakan waktu sekitar 1 ms per 1000 *byte* program dan biasanya berkisar antara 10 ms hingga 50 ms.

Setiap anak tangga harus dimulai dengan sebuah *input* atau beberapa *input* dan harus diakhiri dengan setidaknya satu *output*. Istilah *input* digunakan untuk tindakan kontrol, seperti menutup kontak sakelar. Istilah *output* digunakan untuk perangkat yang terhubung ke *output* PLC, seperti motor.

### **2.1.5.3. *Sequential Function Chart* (SFC)**

Istilah *Sequential Function Chart* (SFC) digunakan untuk representasi visual dari operasi sistem untuk menunjukkan urutan kejadian yang terlibat dalam operasi yang sedang dijelaskan (Bolton, 2015b). Bagan SFC memiliki elemen-elemen berikut:

- a. Operasi digambarkan dengan sejumlah keadaan atau langkah terpisah yang terhubung secara berurutan yang diwakili oleh kotak persegi panjang, masing-masing mewakili keadaan tertentu dari sistem yang sedang dikontrol dan di mana ada beberapa tindakan yang dilakukan. Langkah awal dalam sebuah program diwakili dengan garis ganda, berbeda dengan langkah lainnya Gambar 2. 33 menunjukkan sebuah langkah awal dan langkah-langkah selanjutnya.



Gambar 2. 33 Keadaan, langkah dan transisi SFT

Sumber : (Bolton, 2015b)

- b. Setiap garis penghubung antar *step* memiliki garis horizontal yang mewakili kondisi transisi yang harus direalisasikan sebelum sistem dapat berpindah dari satu *step* ke *step* berikutnya. Dua langkah tidak akan pernah bisa terhubung secara langsung; mereka harus selalu dipisahkan oleh sebuah transisi.

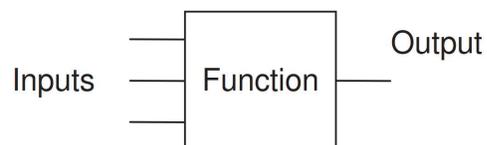
Dua transisi tidak pernah bisa secara langsung mengikuti satu sama lain; mereka harus selalu dipisahkan oleh sebuah langkah.

- c. Program memeriksa kondisi transisi sehingga ketika direalisasikan, *step* berikutnya yang mengikuti transisi tersebut akan berpindah.
- d. Proses ini berlanjut dari satu keadaan ke keadaan berikutnya sampai siklus mesin selesai.
- e. Keluaran/tindakan pada setiap *step*/ langkah diwakili oleh kotak-kotak yang terhubung secara horizontal dan terjadi ketika *step* tersebut telah

direalisasikan. Dengan demikian, kita mungkin memiliki item seperti tunggu satu menit atau buka katup satu menit atau tutup katup satu menit.

#### 2.1.5.4. *Function Block Diagram (FBD)*

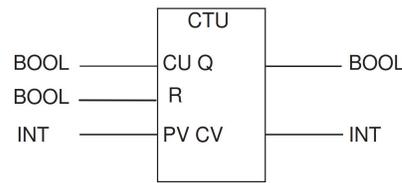
Istilah diagram blok fungsi (FBD) digunakan untuk program PLC yang dijelaskan dalam bentuk blok grafis. Ini digambarkan sebagai bahasa grafis untuk menggambarkan aliran sinyal dan data melalui blok, yang merupakan elemen perangkat lunak yang dapat digunakan kembali. Blok fungsi adalah unit instruksi program yang, ketika dieksekusi, menghasilkan satu atau lebih nilai *output*.



Gambar 2. 34 Bentuk *function blok*

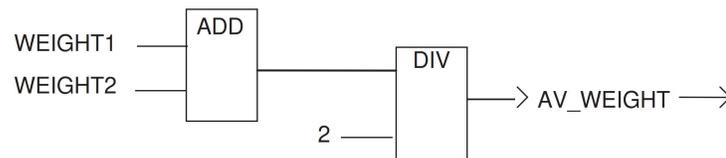
Sumber : (Bolton, 2015b)

Dengan demikian sebuah blok direpresentasikan dengan nama fungsi yang ditulis di dalam kotak. Standar. Blok fungsi digambarkan sebagai blok persegi panjang dengan *input* masuk dari kiri dan *output* muncul dari kanan. Nama jenis blok fungsi ditunjukkan dalam blok, seperti ADD dan DIV pada Gambar 2. 36, dengan nama blok fungsi dalam sistem yang ditunjukkan di atasnya, misalnya *Timer1*. Nama-nama *input* blok fungsi ditampilkan di dalam blok pada titik *input* dan *output* yang sesuai seperti Gambar 2. 35 yaitu titik CU Q, R dan PV CV. Blok fungsi dapat memiliki fungsi standar, seperti gerbang logika, penghitung, atau *timer*, atau memiliki fungsi yang ditentukan oleh pengguna.



Gambar 2. 35 Up-counter CTU

Sumber : (Bolton, 2015b)



Gambar 2. 36 Contoh blok yang ditentukan pengguna

Sumber : (Bolton, 2015b)

**2.1.5.5. Structured Text (ST)**

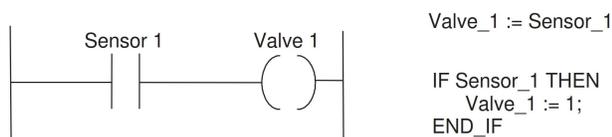
*Structured Text* adalah bahasa pemrograman yang sangat mirip dengan bahasa pemrograman Pascal. Program ditulis sebagai serangkaian pernyataan yang dipisahkan oleh titik koma. Pernyataan-pernyataan tersebut merupakan *statement* dan *subroutine* yang telah ditentukan untuk mengubah variabel, baik nilai yang telah ditentukan, nilai yang tersimpan secara internal, maupun *input* dan *output*.

Tabel 2. 6 Operasi *Structured Text*

Sumber : (Duncan, 2000)

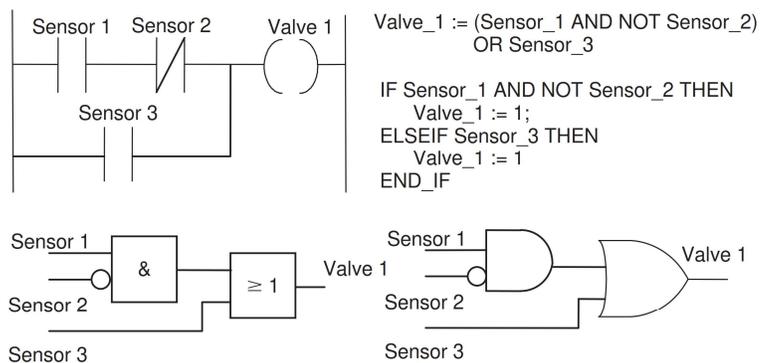
Operasi/ Intruksi	Deskripsi	Skala prioritas
(...)	keterangan dalam kurung (tanda kurung)	Tinggi
Function(...)	Daftar parameter dari suatu fungsi	
**	Menaikkan pangkat ke suatu tingkatan	
-, NOT	Negasi, Boolean NOT	
*, /, MOD	Operasi perkalian, pembagian, modulus	
+, -	Penambahan, pengurangan	

<, >, <=, >=,	Kurang dari, lebih besar dari, kurang dari atau sama dengan, lebih besar dari atau sama dengan	
=, ≠	Kesetaraan, ketidaksetaraan	
AND, &	Boolean AND	
XOR	Boolean XOR	
OR	Boolean OR	Rendah



Gambar 2. 37 Sebuah program *ladder* dalam bentuk *structured text*.

Sumber : (Bolton, 2015b)



Gambar 2. 38 Sebuah program *ladder* dalam bentuk *structured text*, FBD dan gerbang logika

Sumber : (Bolton, 2015b)

**2.1.5.6. Instruction List (IL)**

Metode pemrograman ini dapat dianggap sebagai memasukkan program tangga menggunakan teks dalam daftar instruksi (IL). Daftar instruksi memberikan program sebagai serangkaian instruksi, dengan setiap instruksi pada baris baru.

Setiap instruksi terdiri dari sebuah operator diikuti oleh satu atau lebih masukan (*operand*), yaitu subyek dari operator (Bolton, 2015b).

Contohnya, LD adalah setara dengan memulai anak tangga dengan kontak terbuka untuk *input* A. Instruksi lain adalah OUT Q untuk mengindikasikan bahwa harus ada *output* ke Q.

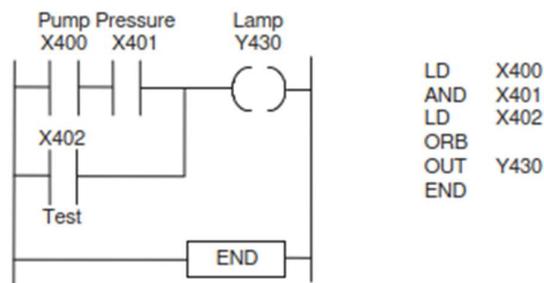
Kode mnemonik Tabel 2. 7 digunakan untuk sebuah intruksi, setiap kode sesuai dengan instruksi / tangga elemen. Kode yang digunakan berbeda dari satu produsen PLC ke produsen lainnya, meskipun standar di bawah IEC 61131-3 telah diusulkan dan diadopsi secara luas.

Tabel 2. 7 Instruksi kode *mnemonics*

Sumber : (Duncan, 2000)

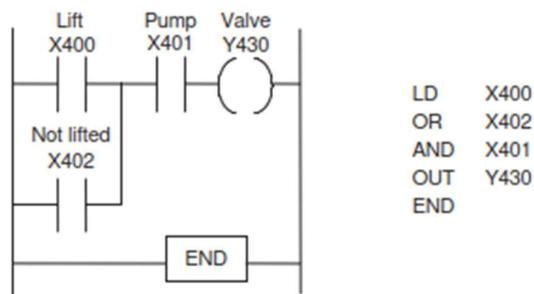
IEC 61131-3	Mitsubishi	OMRON	Siemens	Operasi	Ladder Diagram
LD	LD	LD	A	Memuat operasi ke dalam register hasil	Memulai anak tangga dengan kontak terbuka
LDN	LDI	LD NOT	AN	Memuat operasi negatif ke dalam register hasil.	Memulai anak tangga dengan anak tangga tertutup
AND	AND	AND	A	Boolean AND	Elemen seri dengan kontak terbuka
ANDN	ANI	AND NOT	AN	Boolean AND dengan operan negatif	Elemen seri dengan kontak tertutup
OR	OR	OR	O	Boolean OR	Elemen paralel dengan kontak terbuka
ORN	ORI	OR NOT	ON	Boolean OR dengan operan negatif	Elemen paralel dengan kontak tertutup
ST	OUT	OUT	-	Menyimpan register hasil ke dalam operasi	Keluaran

Tabel 2. 7 menunjukkan beberapa kode yang digunakan oleh produsen dan standar yang diusulkan untuk instruksi yang digunakan. Daftar instruksi adalah bahasa tekstual tingkat rendah yang mudah diimplementasikan dan digunakan oleh sejumlah produsen PLC, terutama untuk PLC kecil dan menengah. Ini sangat cocok untuk program kecil dan mudah. Beberapa produsen PLC tidak mendukung IL tetapi hanya menggunakan bahasa tingkat yang lebih tinggi dari *structured text*.



Gambar 2. 39 Sinyal perintah lampu

Sumber : (Duncan, 2000)



Gambar 2. 40 Program pengoperasian katup

Sumber : (Duncan, 2000)

Pada Gambar 2. 39 menunjukkan sebuah program *Ladder Diagram* (LD) dan *Instruction List* (IL) pada suatu sistem kendali tekanan yang sinyalnya dari sebuah sensor *pressure* dan pompa mengendalikan sebuah lampu.

## 2.2. Penelitian Terkait dan Kebaruan Penelitian

Berdasarkan konsep dan tema penelitian, banyak penelitian yang telah mengkaji tentang kendali suhu menggunakan kendali PID (*Proporsional, Integral, Derivative*) namun masing-masing penelitian tentu memiliki karakteristik tersendiri terkait tema tersebut, baik penggunaan mikrokontroler sebagai pengimplementasian PID, objek yang dikendalikan, spesifikasi alat dan hambatan yang dilalui peneliti terdahulu.

Tabel 2. 8 Beberapa penelitian terkait kendali PID

Judul	Penulis	Tahun	Isi Penelitian
<b>Sistem Kontrol Suhu Electric Muffle Furnace Menggunakan Sensor Thermocouple Type-K Berbasis Mikrokontroler Arduino Uno</b>	Farhan Arista Fhadillah	2023	Penelitian ini membahas tentang pembatasan sistem kendali suhu pada electric muffle furnace menggunakan mikrokontroler Arduino UNO menggunakan kendali PID untuk mengetahui kestabilan suhu menggunakan sistem tersebut terhadap penyalaan tungku pembakaran ( <i>furnace</i> ). Drive untuk mengatur arus , tegangan dan daya pada kawat pemanas menggunakan rangkaian dimmer SCR. Metode tuning yang digunakan adalah metode tuning <i>trial and error</i> untuk mengetahui nilai

			<p>Kp (<i>Konstanta Propotional</i>), Ki (<i>Konstanta Integral</i>) dan Kd (<i>Konstanta Derivative</i>) pada parameter PID. Sehingga pada penelitian tersebut didapatkan nilai terbaik yaitu <math>K_p = 1,25</math>, <math>K_i = 0,67</math>, <math>K_d = 0,75</math> didapatkan kestabilan suhu dengan deviasi suhu tidak lebih dari 2%.</p>
<p><b>Perancangan Kendali Kecepatan Motor Induksi 3 Fasa dengan <i>Variable Frequency Drives</i> Menggunakan Kendali PID Berbasis PLC</b></p>	<p>Rizky Budi Wibowo</p>	<p>2022</p>	<p>Penelitian ini membahas tentang pembuatan sistem pengendalian kecepatan motor induksi 3 fasa menggunakan VSD yang terintegrasi dengan PLC melalui komunikasi <i>modbus serial</i> dengan menggunakan kendali PID yang bertujuan untuk menstabilkan kecepatan putaran motor jika terkena gangguan atau pembebanan. Tuning parameter <math>K_p</math>, <math>K_i</math>, <math>K_d</math> menggunakan metode <i>trial error</i>.</p>
<p><b>Sistem Kendali Suhu <i>Furnace</i> Berbasis PID</b></p>	<p>Insan Kamil Ahmad P</p>	<p>2010</p>	<p>Penelitian ini membahas pembuatan sistem pengendalian suhu pada <i>furnace</i> menggunakan kendali PID dengan menggunakan perangkat mikrokontroler</p>

			<p>ATMega8535 sebagai pengendali prosesnya. Mikrokontroler tersebut nantinya akan mengatur keluaran terhadap rangkaian <i>driver</i> tegangan AC yang mengatur tegangan pada kawat pemanas. Rangkaian <i>driver</i> tegangan AC tersebut menggunakan komponen aktif TRIAC dan OptoTRIAC. Metode tuning yang digunakan untuk mencari nilai <math>K_p</math>, <math>K_i</math>, dan <math>K_d</math> menggunakan metode trial and error dan komparasi menggunakan metode Nichole Ziegler sehingga mendapatkan nilai <math>K_p=10.728</math>, <math>K_i=0.042364</math>, <math>K_d=2364</math> yang memiliki step respons yang paling baik. Untuk <i>setpoint</i> suhu <math>600^{\circ}\text{C}</math> memiliki <i>overshoot</i> yang cukup kecil yaitu 0.3333% dengan suhu tertinggi <math>602^{\circ}\text{C}</math> dan waktu untuk mencapai suhu tertinggi 2005 <i>second</i> (<math>\pm 33.4</math> menit).</p>
--	--	--	--

<p><b>Desain Kontroler PID Menggunakan PLC CP1E-NA Untuk Mengatur Fluida Pada Plant Filtrasi Menggunakan Modul Ultrafiltration</b></p>	<p>Hafidz Aly Hidayat, Aris Triwiyatno, Budi Setiyono</p>	<p>2015</p>	<p>Penelitian ini membahas tentang desain pengendalian tekanan tekanan fluida menggunakan pengendalian PID. Pada pengendalian ini peneliti menggunakan PLC Omron CP1E-NA dengan memanfaatkan intruksi PIDAT (191). Kendali PID ini akan mengendalikan proses kontrol motor untuk memompa tekanan fluida <i>driver</i> yang digunakan adalah <i>inverter</i> untuk mengatur kecepatan ada motor. Metode tuning PID menggunakan metode Ziegler Nichols 1 sehingga mendapatkan nilai parameter PID terbaik yaitu <math>K_p=1</math>, <math>T_i = 6,7</math> dan <math>T_d = 0</math> dimana dari hasil pengujian pada set <i>value</i> 400mbar berhasil stabil saat detik ke-15, pada set <i>value</i> 600mbar berhasil stabil saat detik ke-22, dan pada set <i>value</i> 800mbar berhasil stabil saat detik ke-8.</p>
--	---	-------------	--

<p><b>Desain Optimasi PID Controller Pada Heating Furnace Temperature Menggunakan Metode Particle Swarm Optimization (PSO)</b></p>	<p>Muhammad Agil Haikal, Dandy Tulus Herlambang, Machrus Ali, Muhsalin</p>	<p>2021</p>	<p>Pada penelitian ini membahas tentang desain model temperatur pada heating <i>furnace</i> menggunakan kontrol PID yang dituning dengan Artificial Inteligent ( AI ) dengan metode Particle Swarm Optimization (PSO) Algorithm untuk mengontrol temperatur <i>heating furnace</i> agar lebih stabil dan efisiensi. Penelitian ini merubah <i>furnace</i> temperatur sistem kedalam bentuk persamaan transfer <i>function</i> dan diimplementasikan dalam program MATLAB Simulink. Pada program tersebut membandingkan tiga metode tuning yaitu PID konvensional, PID <i>autotuning</i> dan PID-PSO sehingga mendapatkan nilai PID-Konv (konvensional) dengan nilai <math>K_p = 1</math>, <math>K_i = 1</math>, dan <math>K_d = 0</math>, dengan hasil <i>overshot</i> = 1.2872, <i>undershot</i> = 0.0161, dan <i>settling time</i> pada 38.4422 detik. Metode PID-Auto (tuning matlab 2013a) dengan <math>K_p = 4.2875</math>, <math>K_i = 0.2875</math>, <math>K_d = 1.1256</math>, dengan hasil <i>overshot</i></p>
--	--	-------------	---

			<p>= 1.0716, <i>undershot</i> = 0.0277, dan <i>settling time</i> pada 36.5342 detik. Sedangkan pada metode PID-PSO dengan nilai konstanta <math>K_p = 7.0177</math>, <math>K_i = 0.3486</math>, <math>K_d = 0.4323</math> menghasilkan <i>overshot</i> = 0.0721, <i>undershot</i> 0.0081, dan <i>settling time</i> pada 30.4283 detik.</p>
--	--	--	--