

## **BAB II**

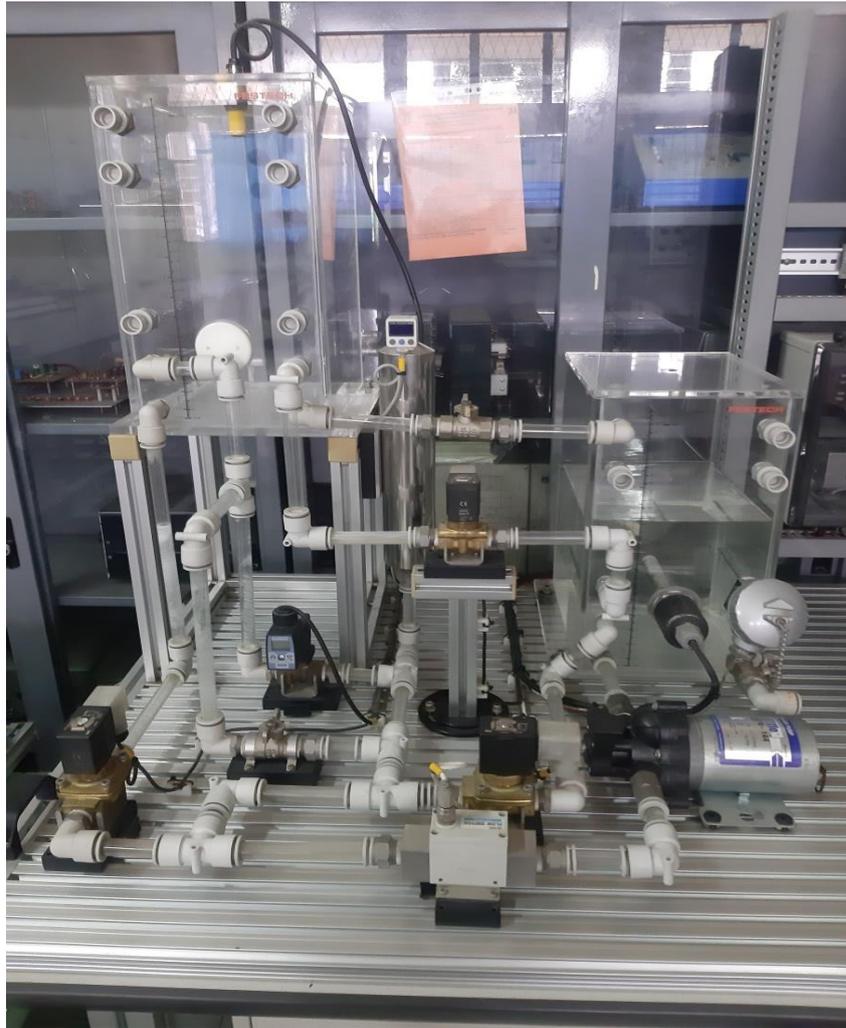
### **TINJAUAN PUSTAKA**

#### **2.1 Process Control System (PCS)**

Sistem kontrol proses adalah sistem kontrol otomatis yang mengawasi pabrik dan peralatan terkait yang menggunakan sistem komputer dan mengontrol aliran proses untuk mempertahankan operasi terus menerus, terdiri dari sistem komputer, perangkat kontrol proses dan sistem antarmuka proses. Perangkat kontrol proses yaitu mengukur variabel proses dan melakukan kontrol melalui sinyal komputer, yaitu actuator, sensor, dll. Sedangkan proses sistem antarmuka yaitu sistem unit fungsional yang menghubungkan perangkat kontrol proses dan sistem komputer [13].

Sistem kontrol adalah proses pengaturan ataupun pengendalian terhadap satu atau beberapa besaran (*variabel, parameter*) sehingga berada pada suatu *range* tertentu. Di dalam dunia industri, dituntut suatu proses kerja yang aman dan berefisiensi tinggi untuk menghasilkan produk dengan kualitas dan kuantitas yang baik serta dengan waktu yang telah ditentukan. Otomatisasi sangat membantu dalam hal kelancaran operasional, keamanan (investasi, lingkungan), ekonomi (biaya produksi), mutu produk, dll.

Ada banyak proses yang harus dilakukan untuk menghasilkan suatu produk sesuai standar, sehingga terdapat parameter yang harus dikontrol atau di kendalikan antara lain tekanan (*pressure*), aliran (*flow*), suhu (*temperature*) dan ketinggian (*level*). Gabungan kerja dari berbagai alat-alat kontrol dalam proses produksi dinamakan sistem pengontrolan proses (*process control system*). Sedangkan semua peralatan yang membentuk sistem pengontrolan disebut pengontrolan instrumentasi proses (*process control instrumentation*). Dalam istilah ilmu kendali, kedua hal tersebut berhubungan erat, namun keduanya sangat berbeda hakikatnya. Pembahasan disiplin ilmu *Process Control Instrumentation* lebih kepada pemahaman tentang kerja alat instrumentasi, sedangkan disiplin ilmu *Process Control System* mengenai sistem kerja suatu proses produksi. Berikut merupakan foto alat Integrated Process Control FPCS 4ALL yang ada di Lab Teknik Elektro Politeknik Negeri Sriwijaya.



*Gambar 2. 1 Alat Integrated Process Control (FPCS 4ALL)  
(Sumber : Dokumen Penulis, 2019)*

## **2.2 Sistem Kontrol Otomatis**

Kontrol secara otomatis artinya tidak memerlukan atau memanfaatkan manusia sebagai operator untuk mengendalikan besaran suatu proses. Pada prinsipnya, kerja pengontrolan otomatis sama dengan kontrol secara manual, hanya saja pada kontrol otomatis peran operator digantikan oleh suatu alat otomatis yang disebut kontroler. Kontroler menggantikan tugas operator untuk mengatur agar sistem bekerja sesuai dengan yang dikehendaki. Peran dari operator disini hanya bertugas untuk memberikan set value.

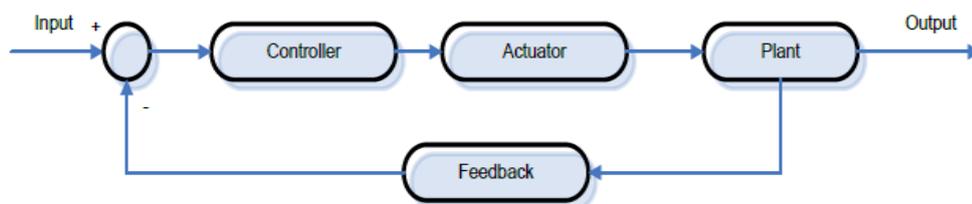
Suatu sistem kontrol otomatis dalam suatu proses kerja berfungsi mengendalikan proses tanpa adanya campur tangan manusia (otomatis). Satu

diantara dua jenis kontrol otomatis adalah sistem kontrol tertutup atau yang biasa dikenal dengan sistem kontrol *closed loop*.

### 2.2.1 Sistem Kontrol Tertutup (*Closed Loop*)

Sistem kontrol umpan balik (*feedback*) sering disebut sebagai sistem kontrol *loop* tertutup. Dalam prakteknya, istilah kontrol umpan balik dan kontrol *loop* tertutup digunakan secara bergantian. Dalam sistem kontrol *loop* tertutup sinyal *error* yang merupakan perbedaan antara sinyal input dan sinyal umpan balik, diumpankan ke pengontrol sehingga mengurangi kesalahan (*error*) dan membawa keluaran dari sistem ke nilai yang diinginkan. Istilah kontrol *loop* tertutup selalu mengimplikasikan penggunaan tindakan kontrol umpan balik untuk mengurangi kesalahan (*error*) sistem.

Keuntungan dari sistem kontrol *closed loop* adalah adanya pemanfaatan nilai *feedback* yang dapat membuat respon sistem kurang peka terhadap gangguan eksternal ataupun internal pada parameter sistem dan mudah untuk mendapatkan *plant* dengan teliti. Kerugian dari sistem kontrol *closed loop* adalah tidak dapat mengambil aksi perbaikan terhadap suatu gangguan sebelum gangguan tersebut mempengaruhi nilai prosesnya.



**Gambar 2.2** Diagram Blok Sistem Kontrol Tertutup

Sumber : <http://repository.usu.ac.id/bitstream/123456789/18172/3/Chapter%20II.pdf>  
(Diakses Pada 27 Februari 2019)

Berdasarkan Gambar 2.2 diagram sistem kontrol tertutup mempunyai sebuah input yang berupa set point sebagai masukan ke *controler* sebelum masuk ke *process* kemudian diolah menjadi nilai output. Nilai output akan diumpan balik kembali menuju *summing point* sebagai acuan apakah nilai output sesuai dengan *set point* yang telah ditentukan.

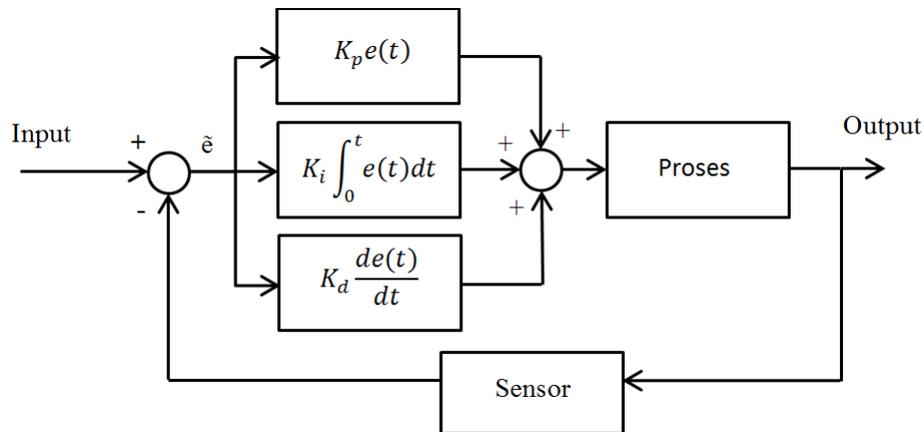
### 2.3 Mode Kontrol Otomatis

Pada sistem kontrol otomatis, sinyal keluaran diharapkan dapat memperkecil *error* yang dengan respon cepat, tepat dan stabil sehingga sistem kontrol tersebut membutuhkan pengaturan yang disebut mode kontrol.

#### 2.3.1 Aksi Pengendali Proporsional + Derivatif + Integral

PID (Proportional–Integral–Derivative controller) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Pengontrol PID adalah pengontrol konvensional yang banyak dipakai dalam dunia industri. Pengontrol PID akan memberikan aksi kepada motor pompa berdasarkan besar error yang diperoleh. motor akan menjadi aktuatur yang mengatur aliran fluida dalam proses industri yang terjadi Level air yang diinginkan disebut dengan Set Point. Error adalah perbedaan dari *Set Point* dengan level air aktual.

PID Blok Diagram dapat dilihat pada gambar dibawah :



**Gambar 2.3** Diagram Blok Kontroler PID

Sumber: <https://www.elektroindonesia.com/elektro/tutor12.html>  
(Diakses Pada 1 Maret 2019)

$$mv(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \dots\dots\dots(2.1)$$

Transfer function aksi pengendali PID:

$$\frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \dots\dots\dots(2.2)$$

$$\frac{p(s)}{q(s)} = \frac{K_D s^3 + K_P s + K_I}{s^3 + (5K_D) s^2 + (8 + K_P) s + K_I} \dots\dots\dots (2.3)$$

Dengan :

$$K_i = K_p \times \frac{1}{T_i} \dots\dots\dots (2.4)$$

$$K_d = K_p \times T_d \dots\dots\dots (2.5)$$

Keterangan :

$mv(t)$  = Output dari pengontrol PID atau Manipulated Variable

$K_p$  = Konstanta Proporsional

$T_i$  = Time Integral

$T_d$  = Time Derivatif

$e(t)$  = Error (selisih antara set point dengan level aktual)

Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proportional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang kita inginkan terhadap suatu plant. Karena kontroler PID hanya mengandalkan variabel proses terukur, bukan pengetahuan mengenai prosesnya, maka dapat secara luas digunakan. Dengan penyesuaian (tuning) ketiga parameter model, kontroler PID dapat memenuhi kebutuhan proses.

### 2.3.1.1 Kontrol Proporsional ( P )

Kontrol P jika  $G(s) = k_p$ , dengan  $k$  adalah konstanta. Jika  $u = G(s) \times e$  maka  $u = K_p \times e$  dengan  $K_p$  adalah Konstanta Proporsional.  $K_p$  berlaku sebagai *Gain* (penguat) saja tanpa memberikan efek dinamik kepada kinerja kontroler. Penggunaan kontrol P memiliki berbagai keterbatasan karena sifat kontrol yang tidak dinamik ini. Walaupun demikian dalam aplikasi-aplikasi dasar yang sederhana kontrol P ini cukup mampu untuk memperbaiki respon transien khususnya rise time dan settling time. Pengontrol proporsional memiliki keluaran yang sebanding/proporsional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya). Kontroler Proporsional memberi pengaruh langsung (sebanding) pada error. Semakin besar *error*, semakin besar sinyal kendali yang dihasilkan kontroler.

Ciri-ciri pengontrol proporsional :

1. Jika nilai  $K_p$  kecil, pengontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat
2. Jika nilai  $K_p$  dinaikkan, respon/tanggapan sistem akan semakin cepat mencapai keadaan mantapnya (mengurangi *rise time*).
3. Namun jika nilai  $K_p$  diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil/respon sistem akan berosilasi.

Pengaruh pada sistem :

1. Menambah atau mengurangi kestabilan.
2. Dapat memperbaiki respon transien khususnya : *rise time*, *settling time*
3. Mengurangi (bukan menghilangkan) *Error steady state*

### 2.3.1.2 Kontrol Integratif ( I )

Pengontrol Integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol (*Error Steady State* = 0 ). Jika sebuah pengontrol tidak memiliki unsur integrator, pengontrol proporsional tidak mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol. Keluaran pengontrol ini merupakan hasil penjumlahan yang terus menerus dari perubahan masukannya. Jika sinyal kesalahan tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan. Sinyal keluaran pengontrol integral merupakan luas bidang yang dibentuk oleh kurva kesalahan.

Ciri-ciri pengontrol integral :

1. Keluaran pengontrol integral membutuhkan selang waktu tertentu, sehingga pengontrol integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, maka keluaran pengontrol akan bertahan pada nilai yang sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai  $K_i$ .
4. Konstanta integral  $K_i$  yang berharga besar akan mempercepat hilangnya offset. Tetapi semakin besar nilai konstanta  $K_i$  akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengontrol.

Pengaruh pada sistem :

1. Menghilangkan *Error Steady State*



2. Respon lebih lambat (dibandingkan dengan P)
3. Dapat Menambah Ketidakstabilan (karena menambah orde pada sistem)

### 2.3.1.3 Kontrol Derivatif ( D )

Keluaran pengontrol diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan pengontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Ketika masukannya tidak mengalami perubahan, keluaran pengontrol juga tidak mengalami perubahan, sedangkan apabila sinyal masukan berubah mendadak dan menaik (berbentuk fungsi step), keluaran menghasilkan sinyal berbentuk impuls. Jika sinyal masukan berubah naik secara perlahan (fungsi ramp), keluarannya justru merupakan fungsi step yang besar magnitudenya sangat dipengaruhi oleh kecepatan naik dari fungsi ramp dan faktor konstanta  $K_d$ . Besarnya sinyal kontrol sebanding dengan perubahan *error* ( $e$ ) Semakin cepat error berubah, semakin besar aksi kontrol yang ditimbulkan. Lebih jelasnya maka lihat gambar berikut.

Ciri-ciri pengontrol derivatif :

1. Pengontrol tidak dapat menghasilkan keluaran jika tidak ada perubahan pada masukannya (berupa perubahan sinyal kesalahan)
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengontrol tergantung pada nilai  $K_d$  dan laju perubahan sinyal kesalahan.
3. Pengontrol diferensial mempunyai suatu karakter untuk mendahului, sehingga pengontrol ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi pengontrol diferensial dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif dan cenderung meningkatkan stabilitas sistem.
4. Dengan meningkatkan nilai  $K_d$ , dapat meningkatkan stabilitas sistem dan mengurangi overshoot.

Pengaruh pada sistem :

1. Memberikan efek redaman pada sistem yang berosilasi sehingga bisa memperbesar pemberian nilai  $K_p$ .
2. Memperbaiki respon transien sebab memberikan aksi saat ada perubahan error.
3. Derivatif hanya berubah saat ada perubahan *error*, sehingga saat ada *error* statis Derivatif tidak beraksi. Sehingga Derivatif tidak boleh digunakan sendiri.



Berdasarkan karakteristik pengontrol ini, pengontrol diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja pengontrol diferensial hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu pengontrol diferensial tidak pernah digunakan tanpa ada kontroler lainnya. Respon kontroler dapat dijelaskan dengan bagaimana responnya terhadap kesalahan, besarnya overshoot dari setpoint, dan derajat osilasi sistem. penggunaan algoritma PID tidak menjamin kontrol optimum sistem atau bahkan kestabilannya.

Pengendali proporsional  $K_p$  akan memberikan efek mengurangi waktu naik tetapi tidak menghapus kesalahan keadaan tunak. Pengendali integral  $K_i$  akan memberikan efek kesalahan keadaan tunak tetapi berakibat memburuknya tanggapan transient. Pengendali derivatif  $K_d$  akan memberikan efek meningkatnya stabilitas sistem, mengurangi lewatan maksimum dan menaikkan tanggapan fungsi transfer. Efek dari setiap pengendali dalam sistem lingkaran tertutup dapat dilihat pada tabel berikut ini :

**Tabel 2.1.** Efek dari  $K_p$ ,  $K_i$ ,  $K_d$

<b>Parameter</b>	<b>Rise Time</b>	<b>Overshoot</b>	<b>Settling Time</b>	<b>S- S Error</b>
$K_p$	Berkurang	Bertambah	Perubahan Kecil	Berkurang
$K_i$	Berkurang	Bertambah	Bertambah	Menghilangkan
$K_d$	Perubahan Kecil	Berkurang	Berkurang	Perubahan Kecil

Dari tabel diatas dapat diketahui bahwa pengendali proporsional akan mengurangi waktu naik, meningkatkan persentase lewatan maksimum dan mengurangi keadaan tunak. Sedangkan pengendali Proporsional derivatif mereduksi lewatan maksimum dan waktu turun. Selain itu, pengendali proporsional integral menurun pada waktu naik, meningkatkan lewatan maksimum dan waktu turun dan akan menghilangkan kesalahan keadaan.

Salah satu permasalahan terbesar dalam desain controller PID yaitu masalah tuning untuk menentukan nilai  $K_i$ ,  $K_p$ , dan  $K_d$  yang pas. Metode-metode tuning dilakukan berdasarkan model matematika plant / sistem. Jika model tidak diketahui, maka dilakukan eksperimen terhadap sistem. Bisa juga pakai system try and error.

Setiap kekurangan dan kelebihan dari masing-masing pengontrol P, I, D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi pengontrol proporsional plus integral plus diferensial (pengontrol PID). Elemen-elemen pengontrol P, I dan D masing-masing secara keseluruhan bertujuan :

- Mempercepat reaksi sebuah sistem mencapai set pointnya
- Menghilangkan offset
- Menghasilkan perubahan awal yang besar dan mengurangi overshoot.

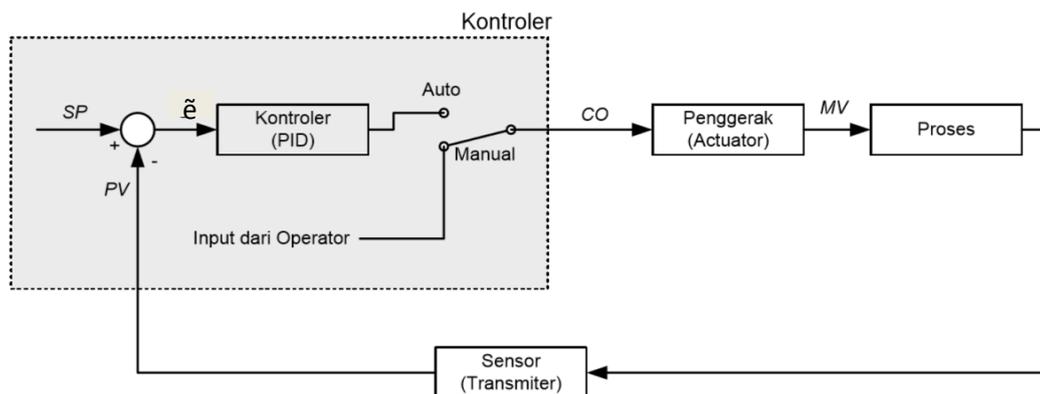
#### 2.4 Model Self Regulating Process

Model self regulating process adalah model yang paling umum dijumpai di industri proses. Model ini hampir muncul pada setiap kasus pengontrolan praktis, yaitu seperti pengontrolan temperature pada sistem heat exchanger, pengontrolan level fluida pada sistem tanki penampung dan sebagainya.

Model self regulating process pada dasarnya dapat didekati oleh sebuah model matematis FOPDT (*First Order Plus Dead Time*) yang hanya dicirikan oleh tiga buah parameter berikut:

1. Process transport delay (delay atau keterlambatan transportasi proses) - L
2. Process time constant (konstanta waktu proses) - T
3. Process static gain (gain statis proses) - K

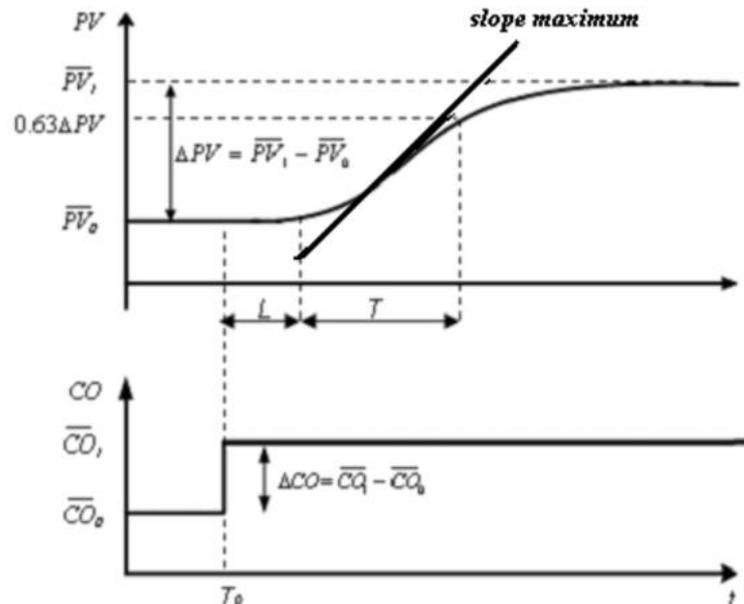
Ketiga parameter yang menggambarkan dinamika proses tersebut, secara praktis dapat diperoleh atau diidentifikasi melalui eksperimen sederhana *Bump test* atau test sinyal tangga pada mode kontrol manual.



**Gambar 2.4** Eksperimen Bump Test pada Mode Kontrol Manual

Sumber: *Kontrol PID untuk Proses Industri* (2015, p. 7)

Secara teknis, eksperimen *bump test* dilakukan dengan cara memberi perubahan tangga (*step*) sinyal output kontroler oleh operator pada saat proses telah mengalami keadaan *steady* (menetap) disekitar titik kerja nominalnya. Respon variabel output (PV) kemudian direkam dan dianalisis dengan menggunakan perangkat lunak tertentu atau dapat juga dianalisis secara manual oleh operator yang bertanggung jawab terhadap proses tersebut.



**Gambar 2.5** Respon Tangga pada Eksperimen Bump Test untuk Model FOPDT  
Sumber: Kontrol PID untuk Proses Industri (2015, p. 8)

Dengan mengacu grafik respon tangga pada Gambar 2.4, parameter-parameter proses FOPDT dapat dicari/dihitung sebagai berikut:

- **Process Transport Delay (keterlambatan transportasi proses) - L** : Waktu keterlambatan transportasi atau waktu ketidakpastian yang terjadi pada proses dihitung sejak terjadi perubahan tangga pada CO (Control Output) sampai variabel proses (PV) yang dikontrol mulai menanggapi perubahan input CO tersebut.
- **Process Time Constant (Konstanta waktu proses) - T** : waktu yang diperlukan sehingga nilai PV mencapai kurang lebih 63% dari keadaan steady akhirnya. Perhitungan nilai konstanta waktu dimulai sejak PV mulai menanggapi perubahan CO (setelah waktu tunda berlalu). Seperti halnya

keterlambatan transportasi, konstanta waktu proses dapat berkisar dari satuan detik sampai menit. Selain dengan cara mengamati respon dari grafik, konstanta waktu proses dapat juga dihitung berdasarkan gradien atau *slope* maksimum yang terjadi pada saat transien. Dengan menggunakan teknik ini, konstanta waktu dapat dicari berdasarkan perumusan dibawah :

$$T = \frac{\Delta PV}{slope\ max} \dots\dots\dots (2.7)$$

- **Process static gain (gain statis proses) – K** : Gain statis proses adalah perbandingan perubahan PV terhadap perubahan CO dalam keadaan steady-nya. Nilai gain proses ini secara langsung menunjukkan kesensitifan dari proses, semakin besar gain statis maka proses semakin sensitif.

$$K = \frac{\Delta PV}{\Delta CO} \dots\dots\dots (2.8)$$

Perubahan kecil pada CO akan menghasilkan deviasi PV yang relatif besar (lihat relasi pada persamaan). Seperti halnya parameter konstanta waktu, gain proses besarnya sangat ditentukan oleh dimensi plant, daya penggerak serta beban yang terlibat pada proses tersebut.

Berdasarkan ketiga parameter diatas, persamaan diferensial yang menggambarkan perilaku proses diatas secara umum dapat dipresentasikan kedalam bentuk *transfer function* seperti yang direlasikan oleh perasaman berikut ini:

$$H(s) = \frac{K}{Ts+1} e^{-sL} \dots\dots\dots (2.9)$$

## 2.5 Tuning Kontroler

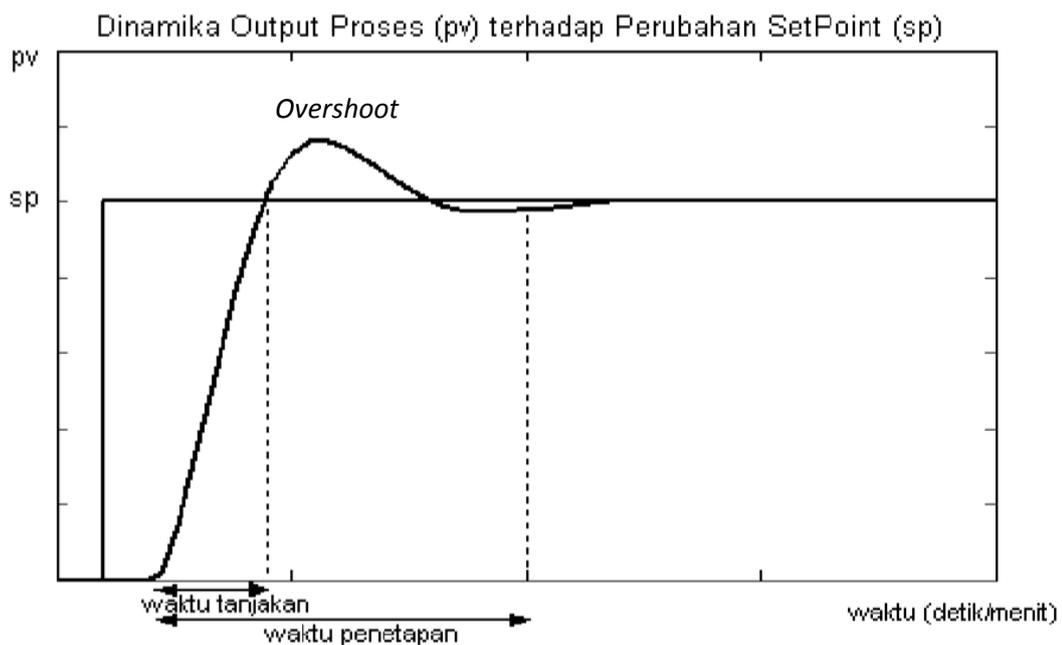
Terkait dengan penggunaan modul PID komersil dalam sebuah kontrol proses industri. Algoritma atau struktur dari PID yang tertanam pada modul tersebut merupakan satu hal yang penting untuk dipahami oleh pengguna. Hal ini dikarenakan algoritma PID dimungkinkan berbeda antar produk satu vendor dengan produk vendor lainnya.

Ditinjau dari aspek tuning parameter kontrolnya, pengetahuan struktur PID tersebut juga sangat penting, karena struktur PID yang berbeda akan memiliki

metode *tuning* yang *relative* berbeda pula. Nilai parameter PID yang optimal pada dasarnya dapat dicari secara mudah berdasarkan metode-metode *tuning* yang telah dikembangkan sebelumnya oleh sejumlah peneliti dan praktisi bidang kontrol proses, baik secara empiris maupun berdasarkan metode-metode optimasi tertentu.

**Tabel 2.2** Pengaruh *Tuning* parameter PID terhadap unjuk kerja proses

	<b>Waktu Tanjakan</b>	<i>Overshoot</i>	<b>Waktu Penetapan</b>	<b>Kestabilan</b>
<b>Pembesaran Kp</b>	Berkurang	Bertambah	Sedikit bertambah	Menurun
<b>Pembesaran Ki (Pengecilan Ti)</b>	Berkurang	Bertambah	Bertambah	Menurun
<b>Pembesaran Kd</b>	Sedikit berkurang	Berkurang	Berkurang	Meningkat



**Gambar 2.6** Respon Proses sebagai Akibat Perubahan Set Point  
Sumber: *Kontrol PID untuk Proses Industri* (2015, p. 32)

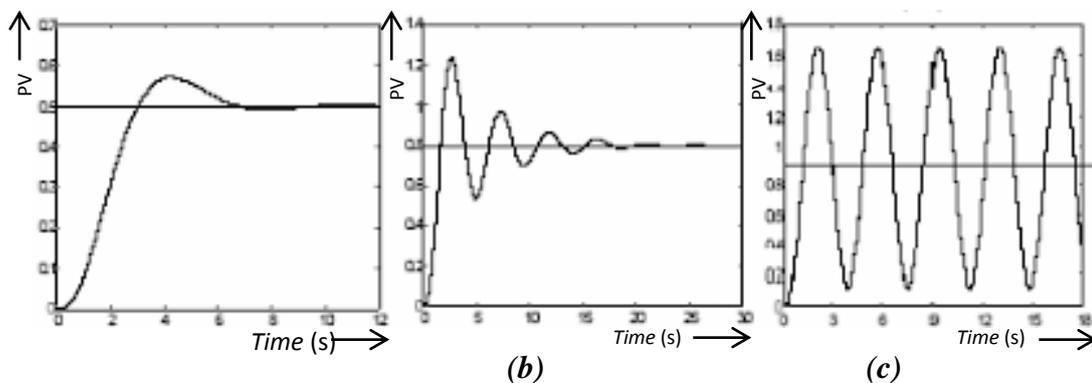
Berikut ini adalah penjelasan mengenai parameter-parameter unjuk kerja pada Tabel 2.2 dan Gambar 2.6.

- **Waktu tahanan:** Waktu yang diperlukan respon (deviasi output variable) untuk naik dari 0 sampai 100% harga akhirnya.
- **Overshoot:** Lonjakan maksimum yang dialami oleh respon proses.
- **Waktu penetapan:** Waktu yang diperlukan respon untuk mencapai dan menetap disekitar 95%-98% dari harga akhirnya.

## 2.6 Tuning Kontroler PID Metode Osilasi Nichols & Ziegler

Pada metode osilasi Nichols & Ziegler, tuning dilakukan dalam loop tertutup dimana masukan referensi yang digunakan adalah fungsi tangga (*step*). Kontroler pada metode ini hanya pengendali proporsional. Pengendali proporsional ini nilainya dinaikkan dari 0 hingga nilai kritis  $K_p$ , sehingga diperoleh keluaran yang terus-menerus berosilasi dengan nilai amplitudo (tinggi gelombang) yang sama. Nilai kritis  $K_p$  tersebut disebut sebagai *ultimate gain*.

Respon keluaran yang dihasilkan pada kondisi penguatan proporsional ditunjukkan pada Gambar 2.7 Sistem akan berosilasi terus-menerus dengan stabil pada saat nilai  $K_p = K_u$ .



**Gambar 2.7** (a) nilai  $K_p = 1$ , (b) nilai  $1 < K_p < K_u$ , (c) Nilai  $K_p = K_u$   
Sumber: Auto Tuning PID Berbasis Metode Osilasi Ziegler-Nichols Menggunakan Mikrokontroler AT89S52 pada Pengendalian Suhu (n.d., p. 2)

Nilai *ultimate periode* ( $T_u$ ) diperoleh setelah keluaran sistem mencapai kondisi yang terus-menerus berosilasi dengan amplitudo yang sama. Nilai perioda dasar ( $T_u$ ) dan penguatan dasar ( $K_u$ ), digunakan untuk menentukan konstanta-konstanta pengendali sesuai dengan rumusan Nichols-Ziegler pada Tabel 2.3.

**Tabel 2.3** Parameter Kontrol Metode Osilasi Nichols-Ziegler

Control Type	Kp	Ki	Kd
P	Ku/2	-	-
PI	Ku/2.2	1.2Kp/Tu	-
<i>Classic PID</i>	0.60Ku	2Kp/Tu	KpTu/8
<i>Pessen Integral Rule</i>	0.7Ku	2.5Kp/Tu	0.15KpTu
<i>Some Overshoot</i>	0.33Ku	2Kp/Tu	KpTu/3
<i>No Overshoot</i>	0.2Ku	2Kp/Tu	KpTu/3

## 2.7 Tuning Kontroler *Proportional Integral Derivative* David W. St. Clair

Tuning PID David. St. Clair merupakan pengembangan dari metode tuning Nichols-Ziegler. Aturan tuning PID untuk metode David W. St. Clair ini telah ditentukan sebagai berikut:

### *Aggressive Type*

$$K_p = \frac{1}{RL} \dots\dots\dots(2.19)$$

$$T_i = 5L \dots\dots\dots(2.11)$$

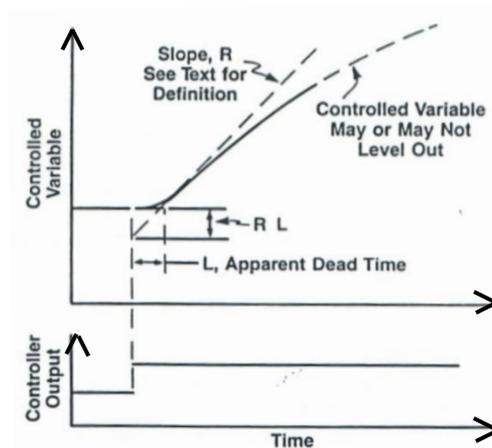
$$T_d = \frac{L}{2} \dots\dots\dots(2.12)$$

### *Conservative Type*

$$K_p = \frac{1}{2RL} \dots\dots\dots(2.13)$$

$$T_i = 5L \dots\dots\dots(2.14)$$

$$T_d = \frac{L}{2} \dots\dots\dots(2.15)$$



**Gambar 2.8** Step Response Parameter Hasil R dan L

Sumber: *Controller Tuning and Control Loop Performance* (1993, p. 21)

## 2.8 Switching Power Supply

*Switching power supply* atau yang lebih dikenal dengan *switched mode power supply* (SMPS), adalah catu daya elektronika yang terdiri dari sebuah regulasi switching yang disediakan sesuai kebutuhan pada tegangan keluaran. Sebuah SMPS adalah daya pengubah yang meneruskan daya dari sebuah sumber untuk beban ideal tanpa rugi-rugi. Fungsi dari pengubah adalah untuk menyediakan tegangan keluaran pada level yang berbeda dibandingkan tegangan masukan.[14]



**Gambar 2.9** Switching Power Supply

Sumber: *Dokumen Penulis, 2019*

Sebuah regulator linier mempertahankan tegangan keluaran yang dikehendaki dengan menghilangkan kelebihan daya pada rugi-rugi tahanan



(misalnya, dalam sebuah resistor atau daerah kolektor-emitor dari transistor dalam modus aktif). Sebuah regulator linier mengatur keluaran baik tegangan atau arus dengan menghilangkan kelebihan daya listrik dalam bentuk panas sebaliknya, mode yang diaktifkan catu daya untuk mengatur keluaran baik tegangan ataupun arus dengan beralih unsur-unsur switching yang ideal. Seperti induktor dan kapasitor yang masuk dan keluar dari konfigurasi listrik yang berbeda *switching* ideal (misalnya, transistor dioperasikan di luar modus aktif).

Jika tidak memiliki tahanan ketika “tertutup” dan tidak membawa arus ketika “terbuka”. Sehingga secara teoritis konverter dapat beroperasi dengan efisiensi 100% (yaitu, semua input daya diberikan beban, dimana tidak ada daya yang terhubung. Tegangan regulasi dihasilkan dengan cara men-switching transistor seri ‘on’ atau ‘off’. Dengan demikian *duty cycle*-nya menentukan tegangan DC rata-rata. *Duty cycle* dapat diatur melalui *feedback* negatif. Feedback ini dihasilkan dari suatu komparator tegangan yang membandingkan tegangan DC rata-rata dengan tegangan referensi.

*Regulator switching* pada dasarnya mempunyai frekuensi yang konstan untuk men-switching transistor seri. Besarnya frekuensi switching tersebut harus lebih besar dari 20KHz agar frekuensi switching tersebut tidak dapat didengar oleh manusia. Frekuensi switching yang terlalu tinggi menyebabkan operasi *switching* transistor tidak efisien dan juga dibutuhkan inti ferrit yang besar atau yang mempunyai permeabilitas tinggi.

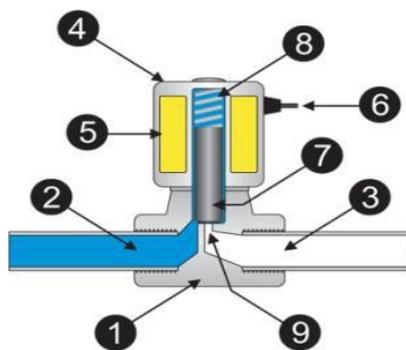
## 2.9 Solenoida Valve

*Solenoid Valve* atau katup listrik merupakan elemen *control* yang paling sering digunakan dalam suatu aliran fluida. Tugas mereka adalah *untuk shut off, release*, mengalirkan atau mencampurkan fluida. *Solenoid Valve* ditemukan di banyak area aplikasi dunia industri seperti *Oil & Gas, water, steam*, petrokimia, pengolahan limbah, dan sebagainya. *Solenoid Valve* bekerja secara *electromechanically* dimana mereka mempunyai kumparan (*coil*) sebagai penggerakannya. Ketika kumparan tersebut mendapatkan *supply* tegangan (AC atau DC) maka kumparan tersebut akan berubah menjadi medan magnet sehingga menggerakkan piston (*plunger*) yang berada di dalamnya.



**Gambar 2.10** Solenoid valve 12V DC

Sumber: Dokumen Penulis, 2019



**Gambar 2.11** Struktur Fungsi Solenoid Valve

Sumber: [www.solenoid-valve-info.com](http://www.solenoid-valve-info.com)

(diakses 9 Maret 2019)

Keterangan gambar :

1. Valve Body
2. Terminal Masukan
3. Terminal Keluaran
4. Koil Solenoida
5. Kumpanan gulungan
6. Kabel suply tegangan
7. Plurger
8. Spring / Per
9. Lubang

*Solenoida valve* adalah katup yang digerakan oleh energi listrik, mempunyai kumpanan sebagai penggeraknya yang berfungsi untuk menggerakkan piston yang dapat digerakan oleh arus AC maupun DC, *solenoida valve* atau katup *solenoida* mempunyai tiga lubang, lubang keluaran, lubang masukan dan lubang *exhaust*, lubang masukan, berfungsi sebagai terminal/tempat cairan masuk atau *supply*, lalu lubang keluaran, berfungsi sebagai terminal atau tempat cairan keluar yang dihubungkan ke beban, sedangkan lubang *exhaust*, berfungsi sebagai saluran untuk mengeluarkan cairan yang terjebak saat piston bergerak atau pindah posisi ketika *solenoida valve* bekerja

Prinsip kerja dari *solenoida valve* yaitu katup listrik yang mempunyai koil sebagai penggerakanya dimana ketika koil mendapat *supply* tegangan maka koil tersebut akan berubah menjadi medan magnet sehingga menggerakkan piston pada bagian dalamnya ketika piston berpindah posisi maka lubang keluaran dari *solenoida valve* mempunyai tegangan mulai kerja di 12 VDC.

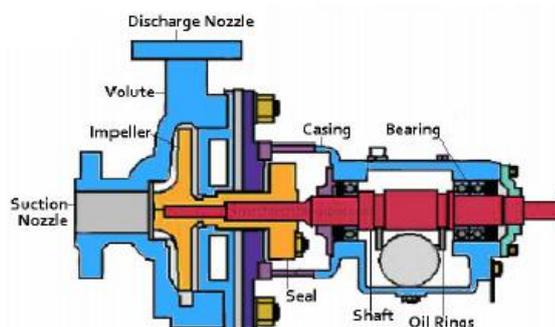
## 2.10 Pompa Air

Pompa air adalah suatu alat atau mesin yang digunakan untuk memindahkan cairan (fluida) dari suatu tempat ke tempat yang lain melalui suatu media perpipaan (saluran) dengan cara menambahkan energi pada cairan yang dipindahkan dan berlangsung secara terus menerus. Berikut adalah tampilan fisik dan bagian-bagian pompa air yang disajikan dalam Gambar 2.12 dan Gambar 2.13



**Gambar 2.12** Bentuk Fisik Pompa Air

Sumber: Dokumen Penulis, 2019



**Gambar 2.13** Bagian-Bagian Pompa Air

Sumber: <http://4mechtech.blogspot.com/2014/07/components-of-centrifugal-pump.html>

(Diakses pada 30 Januari 2019)

Pompa beroperasi dengan prinsip membuat perbedaan tekanan antara bagian masuk (*Suction*) dengan bagian keluar (*Discharge*). Perbedaan tekanan

dihasilkan dari mekanisme misalkan putaran roda impeler yang membuat keadaan sisi masuk nyaris vakum. Perbedaan tekanan inilah yang mengisap cairan sehingga dapat berpindah dari suatu reservoir ke tempat lain. [15]

### 2.11 *Flowmeter (Water Flow Sensor)*

Flowmeter adalah alat untuk mengukur jumlah atau laju aliran air dari suatu fluida yang mengalir dalam pipa atau sambungan terbuka. Alat ini terdiri dari primary device, yang disebut sebagai alat utama dan secondary device (alat bantu sekunder). Flowmeter umumnya terdiri dari dua bagian, yaitu alat utama dan alat bantu sekunder. Alat utama menghasilkan suatu signal yang merespon terhadap aliran karena laju aliran tersebut telah terganggu. Alat utamanya merupakan sebuah orifis yang mengganggu laju aliran, yaitu menyebabkan terjadinya penurunan tekanan. Alat bantu sekunder menerima sinyal dari alat utama lalu menampilkan, merekam, dan/atau mentransmisikannya sebagai hasil dari laju aliran.



**Gambar 2.14** *Digital Flow Switch*  
(Sumber: Dokumen Penulis, 2019)

#### 2.7.1 **Manfaat Penggunaan Flow Switch**

Perangkat flow switch telah digunakan di berbagai macam aplikasi industri. Berikut beberapa manfaat dari flow switch:

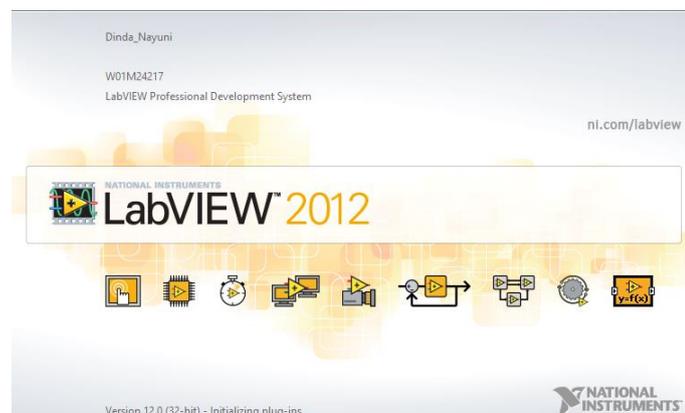
1. **Memantau laju aliran;** dari suhu yang dideteksi oleh RTD, jauh lebih mudah untuk menentukan laju aliran fluida. Ini karena jumlah panas yang didisipasi sebanding dengan laju aliran.
2. **Pemicu alarm;** dengan asumsi bahwa laju aliran fluida diatur untuk tidak melewati level yang ditentukan, setiap kenaikan yang melewati titik setel akan dideteksi oleh flow switch yang kemudian akan mengaktifkan alarm

untuk tindakan yang perlu diambil. Ini juga dapat melindungi perangkat lain seperti pompa agar tidak kelebihan beban.

3. ***Kemudahan dalam kontrol proses industri;*** sebagian besar flow switch yang memiliki fitur teknologi dispersi termal dapat dengan mudah diintegrasikan dengan sistem kontrol seperti PLC (programmable logic controller). Ini karena suhu dari RTD diubah menjadi sinyal listrik oleh pengkondisi sinyal. Ingat, lebih mudah untuk berurusan dengan sinyal listrik ketika datang ke proses kontrol apa pun.
4. ***Tersedia untuk berbagai aplikasi;*** switch ini, dapat digunakan untuk aplikasi yang sangat banyak termasuk garis cair, kolam renang dan air laut. Selama cairan itu tidak bereaksi dengan bahan dari apa flow switch itu dibuat. Oleh karena itu, pertimbangan material sangatlah penting saat memilih flow switch ini. Itu juga akan menentukan rentang hidupnya.

## 2.12 Pengenalan *LabVIEW*

LabVIEW adalah sebuah *software* pemrograman yang diproduksi oleh National *instruments* dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau *Visual basic* , LabVIEW juga mempunyai fungsi dan peranan yang sama, perbedaannya bahwa labVIEW menggunakan bahasa pemrograman berbasis grafis atau blok diagram sementara bahasa pemrograman lainnya menggunakan basis text.

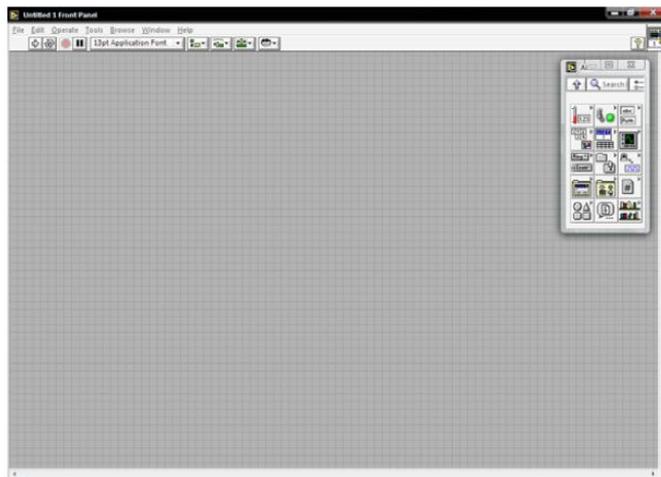


**Gambar 2.15** Software LabVIEW 2012  
(Sumber : Dokumen Penulis, 2019)

Program labVIEW dikenal dengan sebutan Vi atau *Virtual instruments* karena penampilannya dan operasinya dapat meniru sebuah *instrument* [16]. Pada labVIEW, *user* pertama-tama membuat *user interface* atau *front panel* dengan menggunakan *control* dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push buttons*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu *user* menyusun blok diagram yang berisi kode-kode VIs untuk mengontrol *front panel*. *Software* LabVIEW terdiri dari empat komponen utama, yaitu :

### 2.8.1 Front Panel

*Front Panel* adalah bagian *window* yang berlatar belakang abu-abu serta mengandung *control* dan indikator. *front panel* digunakan untuk membangun sebuah VI, menjalankan program dan *mendebug* program. Tampilan dari *front panel* dapat dilihat pada Gambar 2.16

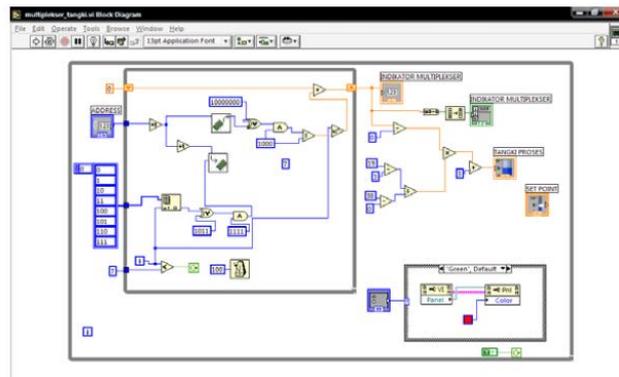


**Gambar 2.16** Front Panel

Sumber : <https://elib.unikom.ac.id/files/disk1/398/jbptunikompp-gdl-setiawanar-19859-9-babii.pdf> (diakses 11 Maret 2019)

### 2.8.2 Blok diagram dari Vi

Blok diagram adalah bagian *window* yang berlatar belakang putih berisi *source code* yang dibuat dan berfungsi sebagai instruksi untuk *front panel*. Tampilan dari blok diagram dapat dilihat pada Gambar 2.17



**Gambar 2.17** Blok Diagram

Sumber : <https://elib.unikom.ac.id/files/disk1/398/jbptunikompp-gdl-setiawanar-19859-9-babii.pdf> (diakses 11 Maret 2019)

### 2.8.3 Control dan Functions Pallette

*Control* dan *Functions Pallette* digunakan untuk membangun sebuah *Vi*. *Control Pallette* merupakan tempat beberapa *control* dan indikator pada *front panel*, *control pallette* hanya tersedia di *front panel*, untuk menampilkan *control pallette* dapat dilakukan dengan mengklik *windows >> show control pallette* atau klik kanan pada *front panel*. Contoh *control pallette* ditunjukkan pada Gambar 2.18



**Gambar 2.18** Control Pallette

Sumber : <https://elib.unikom.ac.id/files/disk1/398/jbptunikompp-gdl-setiawanar-19859-9-babii.pdf> (diakses 11 Maret 2019)

### 2.8.4 Functions Pallette

*Functions Pallette* di gunakan untuk membangun sebuah blok diagram, *functions pallette* hanya tersedia pada blok diagram, untuk menampilkannya dapat dilakukan dengan mengklik *windows >> show control pallette* atau klik kanan

pada lembar kerja blok diagram. Contoh dari *functions pallette* ditunjukkan pada Gambar 2.19



**Gambar 2.19** Functions pallette

Sumber : <https://elib.unikom.ac.id/files/disk1/398/jbptunikompp-gdl-setiawanar-19859-9-babii.pdf> (diakses 11 Maret 2019)

### 2.8.5 Istilah-istilah Penting Dalam LabVIEW

Istilah-istilah penting yang harus diketahui dalam pemrograman LabVIEW adalah sebagai berikut.

1. G : dari kata “*Graphical*” merupakan sebutan untuk bahasa pemrograman LabVIEW yang menggunakan kode berbentuk grafik
2. VI : dari kata “*Virtual Instruments*” merupakan sebutan untuk program yang dibuat dengan LabVIEW.
3. Sub VI : sebagian besar kode program di LabVIEW adalah subVI. SubVI ini sama seperti subrutin dalam bahasa pemrograman teks, yaitu sebuah VI didalam VI. SubVI ini berbentuk iko, atau kotak kecil dengan gambar yang unik di dalamnya, dengan kaki input berada di sebelah kiri dan kaki output berada disebelah kanan.
4. Front Panel: adalah tampilan program. Objek-objek pada jendela ini akan terlihat oleh pengguna saat program dijalankan. Objek-objek pada Front Panel ini, akan secara otomatis memiliki representasi ikon di blok diagram, khususnya untuk objek-objek yang membawa data, baik data yang masuk dari pengguna ke program maupun data yang keluar dari program ke pengguna.
5. Blok Diagram: adalah tempat pembuatan program. jendela ini tidak akan terlihat oleh pengguna saat program dijalankan. Pembuatan program di sini



dilakukan dengan cara menempatkan beberapa node dan meghubungkan satu sama lain.

6. Node: adalah semua objek di jendela Blok Diagram yang memiliki input/output dan melakukan operasi tertentu ketika dijalankan termasuk didalam subVI, terminal, struktur, dan fungsi.
7. Terminal: adalah ikon-ikon di blok diagram yang mewakili objek objek di Front Panel, dimana objek-objek tersebut membawa data baik data yang masuk dari pengguna ke program, maupun data yang keluar dari program pengguna. Contoh terminal adalah Control dan Indikator.
8. Control: adalah semua objek di Front Panel yang memasukkan data dari pengguna ke program. Disebut juga Terminal Input. Contoh Control adalah knob, tombol, saklar dan alat input lainnya.
9. Indicator: adalah semua objek di Front Panel yang mengeluarkan atau menampilkan data dan program ke pengguna. Disebut juga Terminal Output. Contoh Indikator adalah grafik dan LED
10. Struktur: adalah semua bentuk alur pemrograman, seperti perulangan, percabangan, urutan, dan lain-lain. contoh struktur ini adalah For Loop, While Loop, Sekuensial, Case.
11. Fungsi: adalah semua kode-kode dasar yang telah disediakan untuk membuat subVI. Contoh fungsi adalah seperti add, subtract.
12. Wire: atau kawat, digunakan untuk menghubungkan ikon-ikon, sekaligus untuk menunjukkan aliran data dan tipe data.
13. Pemrograman Dataflow: yaitu konsep pemrograman yang akan mengeksekusi node ketika semua inputnya telah tersedia. Ketika node ini telah selesai dieksekusi, maka data akan diteruskan dari output node tersebut ke node berikutnya.

### **2.13 Ultrasonic Proximity Sensor (Sensor Jarak Ultrasonik)**

Sensor Jarak Ultrasonik atau Ultrasonic Proximity Sensor adalah sensor jarak yang menggunakan prinsip operasi yang mirip dengan radar atau sonar yaitu dengan menghasilkan gelombang frekuensi tinggi untuk menganalisis gema yang diterima setelah terpantul dari objek yang mendekatinya. Sensor Proximity Ultrasonik ini akan menghitung waktu antara pengiriman sinyal dengan

penerimaan sinyal untuk menentukan jarak objek yang bersangkutan. sering digunakan untuk mendeteksi keberadaan objek dan mengukur jarak objek di proses otomasi pabrik.



**Gambar 2.20** Ultrasonik Proximity Sensor  
Sumber : Dokumen Penulis. 2019

#### 2.14 State of The Art

Dalam membuat dan merancang penelitian kali ini, penulis melakukan suatu literatur yaitu mencari referensi dengan membandingkan penelitian – penelitian yang sudah ada sebelumnya. Referensi yang dibandingkan berupa karya tulis seperti jurnal ilmiah yang memiliki teori dasar dan data – data yang terpercaya mengenai penelitian yang akan dikembangkan oleh penulis. Adapun beberapa perbandingan penelitian yang telah dilakukan mengenai permasalahan kendali level air terdapat dalam Tabel 2.4. di bawah ini.

**Tabel 2.4.** Tabel *State of The Art*

Parameter	Metode	Aplikasi	Kekurangan	Kelebihan	Referensi
Simulasi kontrol level air	Kontrol er P	Simulasi Visual Basic	waktu naik (rise time) sangat lambat	Menggunakan GUI Virtual	[17]
Penerapan Kontrol PID	Ziegler Nichols	Sensor Ultrasonik	Keluaran lebih cepat stabil dengan menggunakan kontroler PI	Kondisi Osilasi yang stabil konstan	[18]



Pengendalian Level Cairan	FOPDT bump test	Sensor Potensio meter	Menggunakan TCP/IP untuk komunikasi pengendalian level air. sehingga kesulitan saat mengendalikan banyak plant	Tinggi pembacaan sensor dengan tinggi yang terukur proporsional	[19]
Kontrol Level air Steam Drum	PID Genetic Algorit hm	Sensor level transmitt er	Perlu dilakukan penghitungan menggunakan persamaan	Mampu meredam gangguan pada sistem	[6]
Monitoring Level air Pada PLTA	Kontrol er PID	Sensor ultrasoni k	Terdapat noise yang sangat besar saat pengiriman data	Mampu memberikan peningkatan efektifitas kinerja pembangkit sebesar 0.03%	[2]