

BAB II

DASAR TEORI

Bab ini akan membahas tinjauan pustaka dari penelitian sebelumnya yang berkaitan dengan eksperimen yang akan dilakukan, serta dasar teori yang digunakan untuk merancang dan membuat sistem kontrol tekanan guna mengoptimalkan distribusi air dengan metode PID berbasis PLC. Teori yang mendasari pembuatan alat ini meliputi komponen seperti PLC, Pengolahan Air, Sistem Distribusi Air, Kontrol PID, VFD, dan *Pressure Transmitter*.

2.1 Tinjauan Pustaka

Dalam penelitian mengenai "Rancang Bangun *Prototype* Sistem Distribusi Air Berbasis PLC" dilakukan studi literatur dari beberapa referensi yang relevan untuk membantu dalam penyelesaian dari permasalahan yang ada. Berdasarkan penelitian yang telah diperoleh dari referensi terdahulu yaitu penelitian yang berjudul "Sistem Kontrol Pompa Air Menggunakan Kontroler PID Berbasis Raspberry Pi" yang telah disusun oleh Kevin Rosada (2017) untuk menjaga tekanan dalam pipa agar tetap stabil dan mencegah *overpressure* yang dapat menyebabkan kebocoran pada pipa dan kerusakan pada pompa. Sistem ini menggunakan tekanan air pada pipa sebagai parameter pengatur kecepatan putaran pompa air, dimana debit air keluaran pompa air dikontrol agar tetap stabil pada ± 8 Lpm dengan rasio perubahan debit kurang dari 2 Lpm ketika *set point* tekanan air sistem diatur pada 11 kPa. Hasil pengujian menunjukkan bahwa sistem mampu mengontrol tekanan agar tetap stabil meskipun terjadi perubahan jumlah kran yang terbuka sehingga pengontrolan tekanan menyebabkan perubahan pada debit air. Pengaturan Kp, Ki, dan Kd dalam sistem tersebut telah menghasilkan respon sistem yang stabil dengan nilai Kp sebesar 0.000024, nilai Ki sebesar 0.000000001 dan nilai Kd sebesar 0.0009 dengan *rise time* lebih kecil dari 1 s dan *settling time* lebih kecil dari 4.2 s dengan rasio kesalahan sebesar 5%. [3].

Penelitian lain yang relevan dilakukan oleh Mohammad Wirandi, M.M. Lanny W. Pandjaitan, dan Lukas (2024) yang berjudul "Desain Sistem Pemantauan Level Tangki Air Menggunakan PLC dan HMI secara *Real-time* dalam Skala

Laboratorium dan Industri". Penelitian ini mengembangkan sistem pemantauan level air dalam tangki menggunakan PLC Omron CP2E dan HMI Haiwell, dengan sensor level transmitter sebagai input utama. Sistem dirancang untuk mendeteksi secara otomatis kondisi *overflow* dan *underfill*, serta menyajikan informasi melalui antarmuka pengguna yang informatif dan mudah dipahami. Dalam implementasinya, ilustrasi pengujian menunjukkan bahwa saat sensor membaca input *set point* ketinggian air sebesar 0,125 meter (yang diasumsikan setara dengan arus input 12 mA dan skala pembacaan $3000 \pm$), motor pompa 1 akan berhenti mengisi air. Setelah berhenti sejenak, air akan dialirkan kembali menggunakan pompa 2 ke tangki 1, dengan asumsi tangki 2 disisakan hingga ketinggian 0,0625 meter (setara dengan 8 mA atau $1500 \pm$ skala pembacaan). Keunggulan dari penelitian ini terletak pada keandalan penggunaan PLC dalam lingkungan industri serta kemampuannya berintegrasi dengan berbagai sensor dan aktuator, yang memastikan pemantauan berjalan secara *real-time* dengan ketahanan sistem yang baik [4].

Penelitian selanjutnya yang berjudul “Perancangan *Variable Frequency Drive* (VFD) Satu Phasa dengan Menggunakan Metode Sinusoida *Pulse Width Modulation* Berbasis Mikrokontroler” yang disusun oleh Indrawan Nugrahanto, Galih Putra Riatma, dan Anindya Dwi Risdhayanti (2022). Penelitian ini merancang *Variable Frequency Drive* (VFD) satu fasa berbasis mikrokontroler ATmega8535 dengan metode Sinusoida *Pulse Width Modulation* (SPWM), yang digunakan untuk mengatur kecepatan motor AC kapasitor melalui pengaturan frekuensi output inverter dari 8 Hz hingga 80 Hz. Hasil penelitian ini menunjukkan bahwa rangkaian ini terbukti mampu menghasilkan gelombang sinusoidal mendekati ideal dan mengatur kecepatan motor secara presisi dengan error mendekati 0% karena menggunakan sistem database. Perubahan nilai frekuensi keluaran inverter diikuti dengan perubahan nilai tegangan, semakin rendah frekuensi, tegangan keluaran akan menurun, semakin tinggi nilai frekuensi, tegangan keluaran juga akan naik, V/Hz [5].

Berdasarkan studi literatur tersebut, rangkuman dari kedua penelitian tersebut disajikan dalam **Tabel 2.1** berikut untuk mempermudah perbandingan dan pemahaman terhadap penelitian ini.

Tabel 2.1 Perbandingan Penelitian yang Relevan

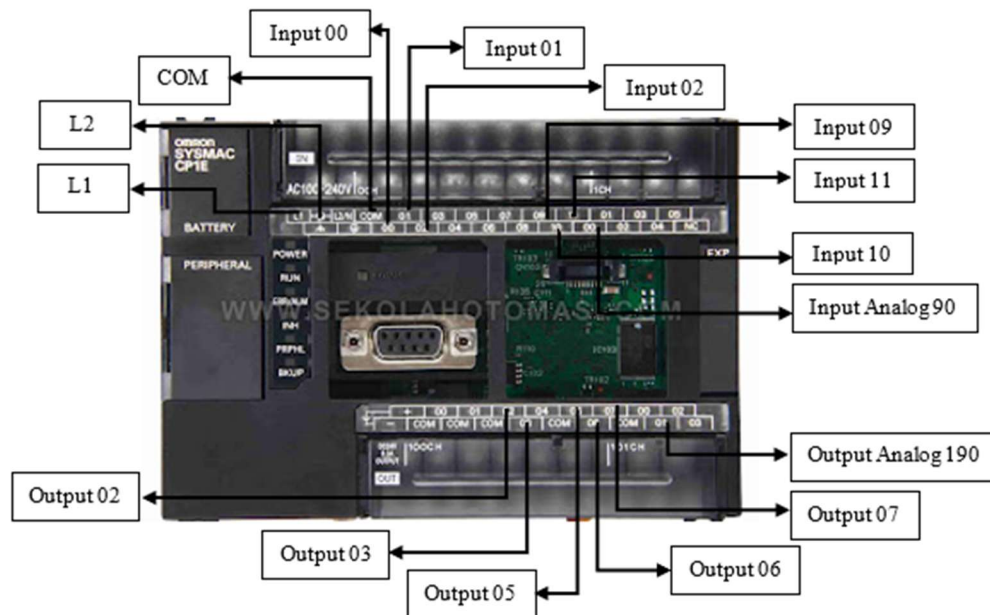
No	Peneliti & Tahun	Fokus Penelitian	Keterangan (Persamaan & Perbedaan)
1.	Rosada (2017) <i>Sistem Kontrol Pompa Air Menggunakan Kontroler PID Berbasis Raspberry Pi</i>	Menjaga kestabilan tekanan dalam pipa untuk mencegah <i>overpressure</i> menggunakan algoritma PID.	Persamaan: Sama-sama menerapkan metode kontrol PID untuk menstabilkan tekanan air. Perbedaan: Referensi menggunakan Raspberry Pi (skala terbatas), sedangkan penelitian ini menggunakan PLC yang lebih handal untuk skala industri (<i>miniplant</i>).
2.	Wirandi, dkk. (2024) <i>Desain Sistem Pemantauan Level Tangki Air Menggunakan PLC dan HMI</i>	Sistem monitoring ketinggian air (<i>level</i>) dan deteksi kondisi abnormal (<i>overflow/underfill</i>) secara <i>real-time</i> .	Persamaan: Sama-sama menggunakan PLC sebagai kontroler utama dan HMI sebagai penampil data. Perbedaan: Referensi berfokus pada pemantauan Level Air , sedangkan penelitian ini berfokus pada pengendalian Tekanan Air .
3.	Nugrahanto, dkk. (2022) <i>Perancangan VFD Satu Fasa dengan Metode Sinusoida PWM</i>	Pengaturan kecepatan motor AC 1 fasa secara presisi menggunakan VFD berbasis mikrokontroler.	Persamaan: Sama-sama memanfaatkan VFD sebagai aktuator utama untuk mengatur kecepatan motor. Perbedaan: Referensi menggunakan Mikrokontroler sebagai otak sistem, sedangkan penelitian ini mengintegrasikan VFD dengan PLC Omron CP1E .

Gambar 2.1 merupakan salah satu jenis *Programmable Logic Controller* (PLC) keluaran Omron yang dirancang untuk aplikasi kontrol otomatisasi industri skala kecil hingga menengah. Di situs resmi Omron Industrial Automation, PLC CP1E dijelaskan sebagai kontroler logika terprogram kompak yang dirancang untuk kebutuhan otomasi sederhana hingga menengah. CP1E menawarkan fleksibilitas tinggi, integrasi yang mudah dengan berbagai sensor dan aktuator, serta mendukung berbagai fungsi seperti pengendalian kecepatan motor, pemrosesan sinyal analog dan digital, serta komunikasi serial. PLC ini cocok digunakan untuk aplikasi industri kecil hingga menengah yang memerlukan solusi kontrol yang stabil, dapat diandalkan, dan mendukung peningkatan kinerja sistem [6]. Konfigurasi model angka pada PLC Omron CP1E-NA20DR-A dijelaskan pada **Tabel 2.2**.

Tabel 2.2 Konfigurasi Model Angka pada PLC Omron CP1E-NA20DR-A [8]

CP1E	Jenis PLC
NA	Tipe Built-in Analog
20	Kapasitas Input/Output (12 Input, 8 Output)
D	Tegangan Input DC
R	Tipe Output berupa Relay
A	<i>AC Power Supply</i>

PLC Omron CP1E-NA20DR-A adalah sebuah pengendali logika terprogram yang dirancang dengan tegangan input digital sebesar 24 VDC dan sumber daya utama sebesar 100–240 VAC. Modul ini mendukung output tipe relay dengan kapasitas maksimum 250 VAC dan arus hingga 2 A untuk beban resistif, sehingga ideal untuk mengontrol aktuator dalam sistem distribusi air. Perangkat ini juga dilengkapi dengan port komunikasi RS-232C, yang memungkinkan koneksi dengan perangkat eksternal seperti HMI, komputer, atau modul tambahan lainnya, sehingga memungkinkan integrasi sistem yang lebih fleksibel. Selain itu, CP1E-NA20DR-A dilengkapi dengan fitur kontrol PID internal, yang dapat digunakan untuk mengatur tekanan air secara real-time berdasarkan perubahan beban dan kondisi proses, guna menjaga kestabilan tekanan.[9]. Rangkaian koneksi dari PLC Omron CP1E-NA20DR-A ini ditunjukkan pada **Gambar 2.2**.



Gambar 2.2 Bagian PLC CP1E Omron [7]

2.2.1 Proses PID Auto-Tuning dalam PLC Omron

Dalam implementasi sistem kendali berbasis PLC Omron (Tipe CP1E/CP1L), instruksi yang digunakan untuk menjalankan algoritma PID adalah PIDAT. Salah satu fitur utama dari instruksi ini adalah kemampuan Auto-Tuning, yaitu proses pencarian parameter K_p (*Proportional*), T_i (*Integral*), dan T_d (*Derivative*) secara otomatis oleh sistem.

A. Metode *Limit Cycle* dan Ziegler-Nichols

Berdasarkan literatur teknis resmi dari Omron dan studi literatur terkait, metode autotuning yang bekerja secara internal di dalam PLC menggunakan pendekatan *Limit Cycle Method* (atau dikenal sebagai *Relay Feedback Method*) untuk akuisisi data karakteristik sistem, yang kemudian dikalkulasi menggunakan kaidah Ziegler-Nichols. Mengacu pada *Omron Technical Explanation for Temperature Controllers*, metode autotuning dijelaskan sebagai berikut:

"Outputs a manipulated variable at 100% and 0% alternately, and the PID constants are calculated from the hunting cycle and amplitude values that occur in the controlled object. Autotuning typically refers to the limit cycle method".

Secara teknis, proses ini bekerja dengan langkah-langkah berikut:

1. *Limit Cycle* (Pengambilan Data): Saat autotuning diaktifkan, PLC akan memberikan sinyal keluaran (Manipulated Variable) secara ekstrem, yaitu 100% (Maksimum) dan 0% (Minimum) secara bergantian (*On-Off*). Hal ini sengaja dilakukan untuk menciptakan osilasi kecil (*hunting*) pada sistem
2. Pengukuran Karakteristik: Dari respons osilasi tersebut, PLC akan mengukur dua variabel kunci, yaitu Amplitudo (A) dan Periode Osilasi (Tu atau Hunting Cycle).
3. Kalkulasi Ziegler-Nichols: Data amplitudo dan periode yang didapat dari metode *Limit Cycle* tersebut kemudian dimasukkan ke dalam rumus penentuan parameter PID berbasis metode Ziegler-Nichols tipe respon frekuensi (*Closed Loop*).

Hal ini sejalan dengan penelitian yang dilakukan pada Plant Filtrasi menggunakan PLC Omron CP1E dengan instruksi PIDAT, di mana peneliti secara eksplisit menyatakan bahwa metode kontrol yang dihasilkan dari proses tuning tersebut adalah berbasis Ziegler-Nichols 1. Dengan demikian, dapat disimpulkan bahwa meskipun pengambilan data dilakukan secara otomatis melalui mekanisme osilasi (*Limit Cycle*), dasar perhitungan matematis untuk penentuan konstanta P, I, dan D pada PLC ini tetap mengacu pada metode standar Ziegler-Nichols.

B. Implementasi Ladder Diagram

Dalam perancangan perangkat lunak (*ladder diagram*), proses kontrol PID dan *autotuning* melibatkan beberapa tahapan data *processing* seperti yang ditunjukkan pada perancangan sistem ini.

a. Aktivasi Auto-Tuning

Pada PLC Omron, aktivasi autotuning dilakukan dengan memanipulasi Control Word dari instruksi PID. Berdasarkan diagram ladder yang dirancang, autotuning diaktifkan dengan memindahkan nilai heksadesimal #8000 ke dalam *word* parameter PID.

- MOV #8000 D209: Instruksi ini mengaktifkan bit ke-15 pada alamat D209. Dalam struktur memori PID Omron, bit ini berfungsi sebagai *trigger* untuk memulai proses *autotuning*.

- MOV #0000 D209: Instruksi ini digunakan untuk menonaktifkan *autotuning* atau mengembalikan ke mode operasi PID normal.

b. Scaling Data (Konversi Nilai)

PLC memproses data dalam format digital (biner/heksa) sementara sensor dan aktuator bekerja pada besaran fisik (Bar/Hz), maka diperlukan proses *scaling* (skala).

1. **Input Scaling (Sensor ke PLC):** Nilai dari sensor tekanan (*Pressure Transmitter*) yang masuk melalui *Analog Input* perlu dikonversi agar sesuai dengan *Set Value* (SV) yang diinginkan. Dalam ladder diagram, proses ini menggunakan instruksi aritmatika *Floating Point* (*F, /F) untuk presisi yang lebih tinggi sebelum dikonversi ke integer.
 - Contoh pada ladder: FLT (Convert to Float) *F (Kali) dan /F (Bagi). Rumus ini mengonversi data mentah analog menjadi satuan teknik (engineering unit) yang dapat dibaca sebagai *Process Variable* (PV).
2. **Output Scaling (PLC ke Inverter/VFD):** Hasil perhitungan PID (OUT PIDAT) berupa nilai digital (0-100% atau resolusi data) dikonversi kembali untuk dikirim ke *Analog Output*.
 - Data digital hasil PID dikonversi agar sesuai dengan resolusi *Analog Output* modul PLC (misalnya 0-6000) yang kemudian diterjemahkan oleh VFD menjadi frekuensi 0-50Hz untuk mengatur kecepatan pompa

c. Scaling Rumus dalam Ladder

Dalam pemrograman PLC, instruksi PID tidak dapat memproses satuan fisik seperti "Bar" secara langsung. Oleh karena itu, diperlukan proses Scaling untuk menyamakan resolusi antara nilai pembacaan sensor (*Process Variable*), nilai target (*Set Value*), dan sinyal keluaran (*Output*). Berdasarkan perancangan, sistem ini menggunakan **Resolusi Data 0 – 6000**. Berikut adalah penjelasan matematis untuk ketiga rumus yang digunakan:

1. Rumus Scaling *Set Value* (Target Tekanan)

Rumus ini digunakan untuk mengubah input target tekanan (dalam satuan Bar) yang dimasukkan pengguna menjadi nilai digital *integer* yang dipahami PLC ditunjukkan pada **Persamaan 2.1**.

$$SV_{data} = \frac{Set\ Value\ Pressure \times 6000}{HSC} \quad (2.1)$$

Keterangan:

- *Set Value Pressure*: Nilai target tekanan yang diinginkan (contoh: 0.5 Bar).
- 6000 : Resolusi maksimal data PLC yang ditetapkan.
- HSC (*High Scale Count*): Batas ukur maksimum dari sensor tekanan yang digunakan (misalnya: 1 Bar atau 10 Bar).

Fungsi: Memastikan bahwa jika pengguna meminta tekanan 50% dari kapasitas sensor, PLC akan menerima nilai 3000 (50% dari 6000).

2. Rumus Scaling *Point Value (Feedback Sensor)*

Rumus ini digunakan untuk menormalisasi sinyal masukan dari sensor tekanan agar memiliki skala yang sama dengan *Set Value* ditunjukkan pada **Persamaan 2.2**.

$$PV_{data} = \frac{F_{Out}PIDAT \times 6000}{HSC} \quad (2.2)$$

Keterangan:

- $F_{Out}PIDAT$: Nilai aktual pembacaan sensor dalam format *Floating Point*.
- 6000: Resolusi sistem.
- HSC: Kapasitas maksimum sensor.

Fungsi: Mengubah tegangan/arus dari sensor menjadi nilai 0-6000. Hal ini krusial agar algoritma PID dapat membandingkan nilai aktual dengan nilai target secara "apple-to-apple" (skala yang sebanding).

3. Rumus Scaling Output PIDAT ke PWM (Sinyal Kontrol)

Rumus ini digunakan untuk menyalurkan hasil perhitungan PID (*Manipulated Variable*) menuju ke Output Analog (VFD) ditunjukkan pada **Persamaan 2.3**.

$$Output_{data} = \frac{F_{outPIDAT} \times 6000}{6000} \quad (2.3)$$

Keterangan:

- $F_{outPIDAT}$: Hasil perhitungan koreksi dari algoritma PID (0-6000).
- Pembagian 6000/6000: Secara matematis bernilai 1.

Fungsi: Rumus ini menunjukkan Pemetaan Linear 1:1. Artinya, nilai keluaran yang dihitung oleh PID (rentang 0-6000) langsung dikirimkan apa adanya ke modul output analog untuk mengontrol frekuensi Inverter (VFD). Jika PID mengeluarkan nilai 6000 (100%), maka VFD akan diperintahkan berputar pada frekuensi maksimum (50 Hz).

2.3 Water Treatment

Pengolahan air atau *water treatment* adalah serangkaian langkah yang dilakukan untuk meningkatkan kualitas air sehingga memenuhi standar tertentu sesuai dengan kebutuhan penggunaannya, baik untuk keperluan industri, domestik, maupun aplikasi lainnya. Proses ini sangat penting dalam industri karena air yang digunakan dalam proses produksi harus memiliki kualitas yang terjaga, terutama dalam hal tekanan, volume, dan kebersihan.[10]. Dalam proses *water treatment* terdiri dari beberapa proses yaitu:

a. *Clarifier Plant*

Sumber air baku, seperti air sungai, pertama-tama dipompa menuju clarifier. Pada tahap ini, air dicampurkan dengan bahan kimia koagulan dan flokulan untuk membantu pengendapan partikel padat. Proses ini dibantu oleh agitator untuk memastikan distribusi bahan kimia yang merata.

b. *Sedimentation*

Setelah proses koagulasi, flok-flok partikel besar akan mengendap di dasar clarifier dan lumpur yang dihasilkan dikeluarkan melalui sistem blowdown.

c. *Filtration (Sand Filter)*

Air yang telah melalui proses sedimentasi dialirkan ke sand filter untuk menghilangkan partikel halus yang masih tersisa. Filtrasi ini menggunakan media pasir dan kerikil, serta dilengkapi dengan sistem backwash otomatis.

d. *Filter Water Storage*

Air bersih hasil filtrasi ditampung dalam tangki penyimpanan (*filter water storage tank*) untuk menjaga kontinuitas suplai ke proses-proses berikutnya.

e. *Distribution System*

Dari *filter water storage*, air didistribusikan ke berbagai unit pemakaian seperti demineralisasi plant dan cooling tower. Pada tahap ini, pengaturan tekanan distribusi menjadi sangat penting untuk memastikan aliran air yang stabil dan aman.

Pada penjelasan diatas, sistem *water treatment* yang dijelaskan merujuk pada proses yang diterapkan di salah satu perusahaan yaitu Pabrik Utilitas PUSRI 1B Palembang. Namun, penelitian ini difokuskan hanya pada sistem distribusi air dari *filter water storage*, khususnya pada pengendalian tekanan distribusi menggunakan metode kontrol PID berbasis PLC. Dengan demikian, tahapan *water treatment* secara keseluruhan bertujuan untuk memberikan gambaran sistem yang melatarbelakangi fokus penelitian.

2.4 Hukum Bernoulli

Hukum Bernoulli adalah prinsip fundamental dalam dinamika fluida yang menyatakan bahwa dalam aliran fluida yang stabil, terdapat hubungan antara tekanan statis, kecepatan, dan energi potensial gravitasi di setiap titik aliran. Konsep ini diperkenalkan oleh ilmuwan asal Belanda, Daniel Bernoulli, pada abad ke-18. Hukum Bernoulli menjelaskan bahwa dalam aliran fluida ideal (tanpa viskositas dan tidak dapat dimampatkan), jumlah energi tetap konstan sepanjang jalur alirannya. Prinsip ini juga menyatakan bahwa dalam aliran fluida yang stabil, tekanan fluida akan menurun seiring dengan peningkatan kecepatan aliran. Akibatnya, tekanan udara di sekitarnya menjadi lebih rendah, dan udara bergerak dari area bertekanan tinggi ke area bertekanan rendah.[11]. Secara matematis, hukum Bernoulli dapat dinyatakan secara umum dalam **Persamaan 2.2** dan diterapkan penelitian ini dengan kondisi pada 2 titik pada **Persamaan 2.3**:

$$P + \frac{1}{2}\rho v^2 + \rho gh = \text{konstan} \quad (2.2)$$

$$P_1 + \frac{1}{2}\rho v_1^2 + \rho gh_1 = P_2 + \frac{1}{2}\rho v_2^2 + \rho gh_2 \quad (2.3)$$

Jika ketinggian pipa sama maka **Persamaan 2.3** menjadi **Persamaan 2.4**:

$$P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho v_2^2 \quad (2.4)$$

Sehingga menjadi **Persamaan 2.5**:

$$P_2 = P_1 - \frac{1}{2}\rho v_1^2 \left(\left(\frac{A_1}{A_2} \right)^2 - 1 \right) \quad (2.5)$$

Keterangan:

P = Tekanan fluida

P₁ = Tekanan di titik 1

P₂ = Tekanan di titik 2

ρ = Massa jenis fluida

v = Kecepatan aliran fluida

v₁ = Kecepatan di titik 1

v₂ = Kecepatan di titik 2

g = Percepatan gravitasi

h = Tinggi fluida

Hukum ini berlaku untuk aliran fluida ideal (tanpa gesekan) dan aliran tunak (*steady flow*). Berdasarkan persamaan Bernoulli di semua kondisi akan konstan sehingga jika satu komponen meningkat, misalnya kecepatan, maka komponen yang lain harus berubah, hal ini bertujuan untuk mempertahankan jumlah tersebut untuk tetap konstan[11].

2.5 Sistem Kontrol

Sistem kontrol adalah kumpulan komponen yang dirancang untuk mengatur, mengarahkan, atau memanipulasi perilaku suatu sistem agar menghasilkan keluaran yang sesuai dengan tujuan yang diinginkan. Sistem kontrol banyak digunakan dalam bidang teknik dan industri untuk memastikan proses berjalan

dengan aman, efisien, dan stabil. Komponen utama dalam sistem kontrol umumnya terdiri dari: masukan (*input*), kontroler, aktuator, proses (*plant*), dan keluaran (*output*). Tujuan utama dari sistem kontrol adalah mengatur variabel proses seperti tekanan, suhu, aliran, atau level, sehingga nilainya tetap berada pada set point meskipun terjadi gangguan dari luar atau perubahan beban proses. Dalam implementasinya, sistem kontrol dapat dibedakan menjadi dua jenis utama, yaitu sistem kontrol loop terbuka (*open-loop*) dan loop tertutup (*closed-loop*). Perbedaan mendasar antara keduanya terletak pada keberadaan mekanisme umpan balik (*feedback*).[12].

2.5.1 Sistem Kontrol *Open Loop*

Sistem kontrol open loop, atau sistem kontrol terbuka, adalah sistem di mana keluaran tidak mempengaruhi tindakan kontrol. Dalam kontrol open loop, keluaran sistem tidak dipertimbangkan dalam menentukan sinyal kontrol. Keluaran dari sistem kontrol ini tidak dapat digunakan sebagai umpan balik untuk masukan. Pada sistem kontrol open loop, keluaran tidak dapat dibandingkan dengan masukan acuan. Setiap masukan acuan terkait dengan operasi tertentu. Sistem ini hanya dapat diterapkan jika hubungan antara masukan dan keluaran sudah diketahui dan tidak ada gangguan internal maupun eksternal.[13]. **Gambar 2.3** menunjukkan hubungan masukan-keluaran untuk sistem kontrol loop terbuka.

Open Loop System

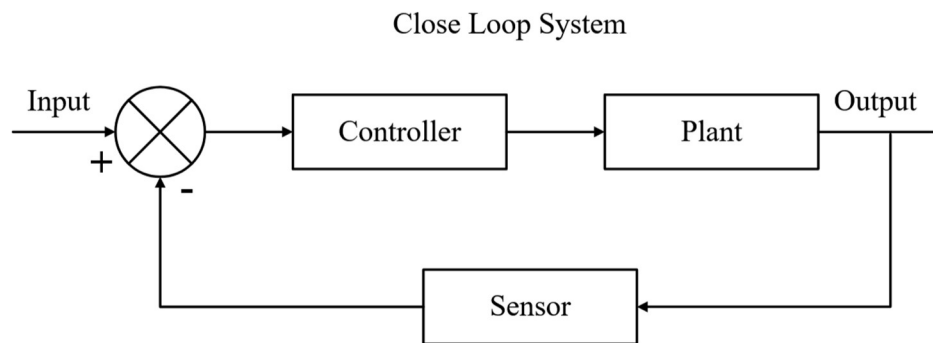


Gambar 2.3 Sistem *Open Loop*

2.5.2 Sistem Kontrol *Close Loop*

Sistem kontrol tertutup, atau close loop, adalah sistem di mana keluaran memiliki pengaruh langsung terhadap tindakan pengendalian, atau keluaran tersebut akan diumpan balik ke masukan. Sistem ini menghasilkan nilai error yang berasal dari selisih antara nilai referensi dan nilai umpan balik. Nilai error ini

digunakan sebagai masukan pada bagian yang dikendalikan. Feedback tersebut menjadi acuan bagi sistem untuk mengurangi nilai error dengan sinyal kontrol tertentu, sehingga keluaran sistem dapat mencapai nilai yang diinginkan[13]. **Gambar 2.4** menunjukkan hubungan masukan-keluaran untuk sistem kontrol loop tertutup.



Gambar 2.4 Sistem *Close Loop*

2.6 Fungsi Transfer

Fungsi transfer adalah representasi matematis yang menggambarkan hubungan antara input dan output dari suatu sistem linear yang tidak berubah seiring waktu dalam domain frekuensi (Laplace). Fungsi ini umumnya dinyatakan sebagai perbandingan antara transformasi Laplace dari output dan input, dengan asumsi bahwa kondisi awal sistem adalah nol.[14].

Secara umum, fungsi transfer dinyatakan dalam **Persamaan 2.6**:

$$G(s) = \frac{Y(s)}{U(s)} \quad (2.6)$$

Keterangan:

- $G(s)$ adalah fungsi transfer sistem,
- $Y(s)$ adalah transformasi Laplace dari keluaran,
- $U(s)$ adalah transformasi Laplace dari masukan,
- s adalah variabel kompleks dalam domain Laplace.

Fungsi transfer banyak digunakan dalam analisis dan perancangan sistem kontrol karena mempermudah pengamatan karakteristik dinamik sistem, seperti

waktu naik, waktu tunak, overshoot, dan kestabilan. Fungsi transfer sangat krusial dalam desain dan penyetelan kontrol PID karena memungkinkan analisis stabilitas dan karakteristik sistem sebelum diterapkan di lapangan. Fungsi transfer dibagi menjadi 2 yaitu [14]:

1. Fungsi Transfer Orde 1

Fungsi transfer orde 1 adalah model matematis yang menggambarkan perilaku sistem dinamis yang turunan tertingginya adalah turunan pertama. Fungsi transfer orde 1 sering digunakan untuk menggambarkan sistem dengan tanggapan eksponensial, seperti sistem peredam massa atau sistem pemanas massa. Contohnya sistem peredam massa, sistem pemanas massa, dan lain-lain [15]. Bentuk umum ditunjukkan dalam **Persamaan 2.7**:

$$G(s) = \frac{K}{(\tau s + 1)} \quad (2.7)$$

Dimana:

$G(s)$ adalah fungsi transfer.

K adalah perolehan statis.

τ adalah konstanta waktu (waktu yang dibutuhkan untuk mencapai 63.2% dari nilai akhir setelah masukan step).

Perilaku: Sistem orde 1 menanggapi input dengan cara eksponensial, dengan konstanta waktu τ menentukan kecepatan tanggapan sistem.

2. Fungsi Transfer Orde 2

Fungsi transfer orde 2 menggambarkan sistem dengan turunan tertinggi turunan kedua. Fungsi transfer orde 2, di sisi lain, sering digunakan untuk model sistem mekanis seperti pegas-massa-peredam, yang dapat menunjukkan respons yang osilasi. Contohnya adalah sistem pegas-massa-peredam, sistem rangkaian listrik seri RC [16]. Bentuk umum ditunjukkan dalam **Persamaan 2.8**:

$$G(s) = \frac{\omega n^2}{(s^2 + 2\zeta\omega n s + \omega n^2)} \quad (2.8)$$

Dimana:

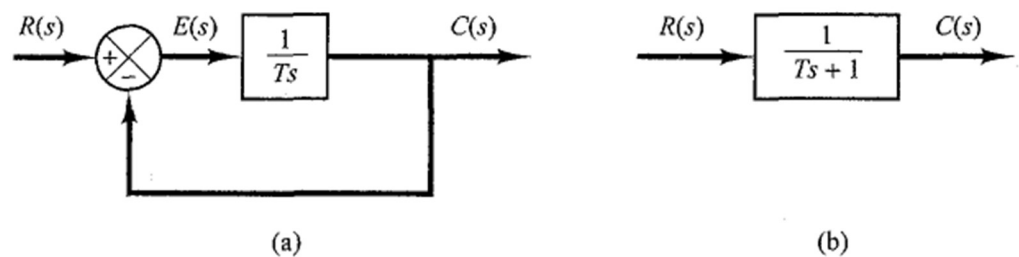
ωn adalah frekuensi alami (frekuensi osilasi tanpa peredam).

ζ adalah faktor peredam (memengaruhi osilasi sistem).

Perilaku: Sistem orde 2 dapat menanggapi input dengan berbagai cara tergantung pada nilai faktor peredam ζ [16]:

- $\zeta < 1$ (peredam kurang): sistem akan osilasi (undamped).
- $\zeta = 1$ (peredam kritis): sistem akan kembali ke nilai akhir dengan cepat tanpa osilasi.
- $\zeta > 1$ (peredam over): sistem akan kembali ke nilai akhir tanpa osilasi, tetapi lebih lambat daripada peredam kritis.

2.6.1 Sistem Orde 1 Unit Step



Gambar 2.5 Sistem Orde Satu [17] (a) diagram blok (b) diagram blok yang disederhanakan

Sistem orde satu ditunjukkan pada **Gambar 2.5** (a) yang juga merupakan blok diagram dari sistem orde satu. Secara fisik, sistem ini dapat mewakili sirkuit RC, sistem termal, atau semacamnya. Diagram blok yang disederhanakan ditunjukkan pada **Gambar 2.5** (b). Hubungan masukan-keluaran diberikan oleh **Persamaan 2.9**[12].

$$\frac{C(s)}{R(s)} = \frac{1}{Ts + 1} \quad (2.9)$$

Dalam unit respon step dari sistem orde satu transformasi Laplace dari fungsi langkah satuan adalah $1/s$, dengan mensubstitusikan $R(s) = 1/s$ ke dalam **Persamaan 2.9**, diperoleh **Persamaan 2.10**[12]:

$$C(s) = \frac{1}{Ts + 1} \cdot \frac{1}{s} \quad (2.10)$$

Memperluas $C(s)$ menjadi pecahan parsial menghasilkan **Persamaan 2.11**[12]:

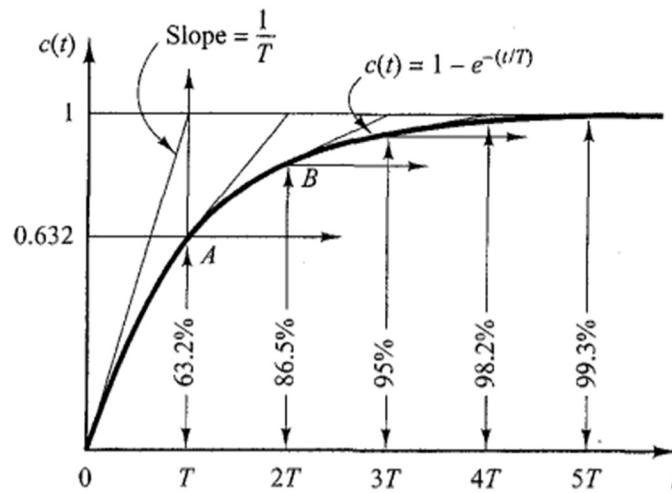
$$C(s) = \frac{1}{s} - \frac{T}{Ts + 1} = \frac{1}{s} - \frac{1}{s + \left(\frac{1}{T}\right)} \quad (2.11)$$

Mengambil transformasi Laplace invers dari **Persamaan 2.11**, kita memperoleh **Persamaan 2.12**[12]:

$$c(t) = 1 - e^{-\frac{t}{T}}, \text{ for } t \geq 0 \quad (2.12)$$

Persamaan 2.12 menyatakan bahwa awalnya output $c(t)$ adalah nol dan akhirnya menjadi satu. Salah satu karakteristik penting dari kurva respons eksponensial $c(t)$ adalah bahwa pada $t = T$ nilai $c(t)$ adalah 0,632, atau respons $c(t)$ telah mencapai 63,2% dari total perubahannya. Hal ini dapat dengan mudah dilihat dengan mensubstitusi $t = T$ pada $c(t)$. Maka dimasukkan dalam **Persamaan 2.13**[12]:

$$c(T) = 1 - e^{-1} = 0.632 \quad (2.13)$$



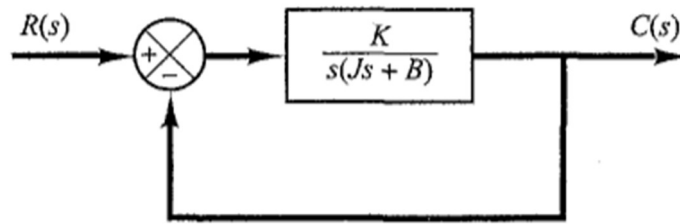
Gambar 2.6 Sistem *Close Loop* [12]

Perhatikan **Gambar 2.6** bahwa semakin kecil konstanta waktu T , semakin cepat respons sistem. Karakteristik penting lainnya dari kurva respons eksponensial adalah kemiringan garis singgung pada $t = 0$ adalah $1/T$, sesuai **Persamaan 2.14**[12]:

$$\left. \frac{dc}{dt} \right|_{t=0} = \left. \frac{dc}{dt} e^{-\frac{t}{T}} \right|_{t=0} = \frac{1}{T} \quad (2.14)$$

Output akan mencapai nilai akhir pada $t = T$ jika mempertahankan kecepatan respons awalnya. Dari **Persamaan 2.14** kita melihat bahwa kemiringan kurva respons $c(t)$ menurun secara monoton dari $1/T$ pada $t = 0$ menjadi nol pada $t = \infty$. Kurva respons eksponensial $c(t)$ yang diberikan oleh **Persamaan 2.13** ditunjukkan pada **Gambar 2.6**. Dalam satu konstanta waktu, kurva respons eksponensial telah berubah dari 0 menjadi 63,2% dari nilai akhir. Dalam dua konstanta waktu, respons mencapai 86,5% dari nilai akhir. Pada $t = 3T, 4T$, dan $5T$, respons mencapai 95%, 98,2%, dan 99,3%, masing-masing, dari nilai akhir. Jadi, untuk $t \geq 4T$, respons tetap dalam 2% dari nilai akhir. Seperti yang terlihat dari **Persamaan 2.13**, kondisi stabil dicapai secara matematis hanya setelah waktu yang tak terbatas. Namun, dalam praktiknya, estimasi waktu respons yang wajar adalah lamanya waktu yang dibutuhkan kurva respons untuk mencapai dan tetap berada dalam garis 2% dari nilai akhir, atau empat konstanta waktu [12].

2.6.2 Sistem Orde 2 Unit Step

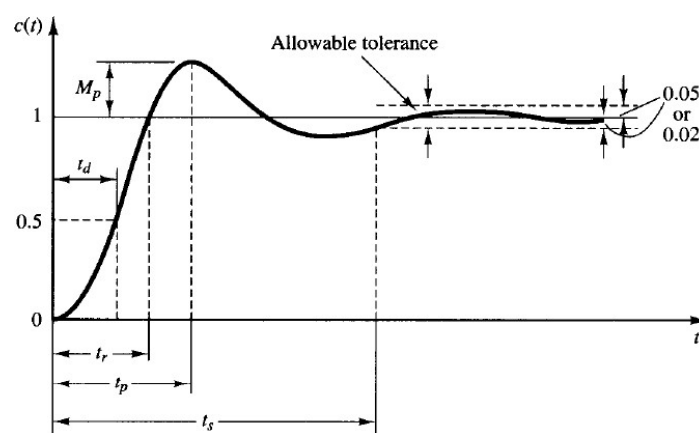


Gambar 2.7 Diagram blok orde dua yang disederhanakan [12]

Sistem di mana fungsi transfer loop tertutup memiliki dua kutub disebut sistem orde kedua. (Beberapa sistem orde kedua mungkin melibatkan satu atau dua nol). Fungsi transfer loop tertutup dari sistem yang ditunjukkan pada **Gambar 2.7** ditunjukkan dalam **Persamaan 2.15**[12]:

$$\frac{C(s)}{R(s)} = \frac{K}{Js^2 + Bs + K} \quad (2.15)$$

Karakter performansi sistem kontrol sering dinyatakan dalam bentuk respon transien terhadap masukan tangga satuan (step) karena mudah dibangkitkan dan cukup radikal (bila respon terhadap masukan step diketahui, maka secara matematis dapat dihitung respon terhadap setiap masukan). Respon transien suatu sistem terhadap masukan step bergantung pada syarat awal. Untuk memudahkan perbandingan respon transien berbagai sistem, hal yang biasa dilakukan adalah menggunakan syarat awal standar bahwa sistem mula-mula dalam keadaan diam sehingga keluaran dan semua turunan waktunya pada awal respon sama dengan nol, selanjutnya karakteristik respon secara mudah dapat dibandingkan. Untuk menganalisis suatu respon transien sistem orde 2 terhadap masukan step ditunjukkan melalui **Gambar 2.8**[12]:



Gambar 2.8 Kurva respon tangga satuan yang menunjukkan t_d , t_r , t_p , M_p dan t_s [12]

Respon transien sistem kontrol praktis sering menunjukkan osilasi teredam sebelum mencapai keadaan tunak. Dalam menentukan karakteristik respon transien sistem kontrol terhadap masukan step biasanya dicari parameter yang ditunjukkan pada **Gambar 2.8**, yaitu [12]:

1. **Waktu tunda (t_d)**, yaitu waktu yang diperlukan respon untuk mencapai setengah harga akhir yang pertama kali.
2. **Waktu naik (t_r)**, yaitu waktu yang diperlukan respon untuk naik dari 10% sampai 90%, 5% sampai 95%, atau 0 sampai 100% dari harga akhir. Untuk sistem orde kedua redaman kurang, biasanya digunakan waktu naik 0% - 100%, untuk sistem redaman lebih, biasanya digunakan waktu naik 10% - 90%.

3. **Waktu puncak (t_p)**, yaitu waktu yang diperlukan respon untuk mencapai puncak untuk pertama kali.
4. **Persen maksimum (M_p)**, yaitu nilai presentase dari nilai puncak maksimum kurva respon terhadap nilai keadaan tunak respon. Jika nilai steady-state akhir dari respons berbeda dari satu, maka biasanya digunakan persentase overshoot maksimum. Nilai ini didefinisikan oleh **Persamaan 2.16**.

$$\text{Max percent overshoot} = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\% \quad (2.16)$$

5. **Waktu penetapan (t_s)**, yaitu waktu yang diperlukan kurva respon untuk mencapai dan menetap dalam daerah di sekitar harga akhir, yang ukurannya ditentukan dengan presentase mutlak dari harga akhir, biasanya 2% atau 5%.

2.7 Kontrol PID (*Proportional-Integral-Derivative*)

Sistem kendali *Proportional Integral Derivative* (PID) adalah sebuah pengendali yang digunakan untuk menentukan ketepatan suatu sistem dengan adanya umpan balik dalam sistem tersebut. Sistem pengendali PID menggabungkan tiga metode pengaturan, yaitu pengendali P (*Proportional*), pengendali D (*Derivative*), dan pengendali I (*Integral*). Setiap pengendali memiliki parameter spesifik yang perlu disesuaikan agar dapat berfungsi dengan optimal, yang dikenal sebagai konstanta.[18]. Setiap jenis pengendali memiliki kelebihan dan kekurangan masing-masing seperti pada **Tabel 2.3** di bawah ini:

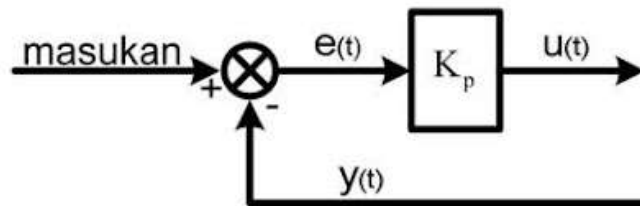
Tabel 2.3 Respon kendali PID terhadap perubahan konstanta[18]

Ruang Lingkup Tertutup	Waktu Naik (t_r)	Nilai Puncak (<i>overshoot</i>)	<i>Settling Time</i>	Galat Keadaan Tunak
K_p	Menurunkan	Meningkatkan	Perubahan Kecil	Menurunkan
K_i	Menurunkan	Meningkatkan	Meningkatkan	Mengeliminasi
K_d	Perubahan Kecil	Menurunkan	Menurunkan	Perubahan Kecil

1. Metode P (*Proportional*)

Pengontrol proporsional menghasilkan sinyal kontrol yang berbanding lurus dengan selisih antara nilai *setpoint* dan nilai aktual dari sistem. Secara sederhana, output dari pengontrol proporsional adalah hasil perkalian antara

konstanta proporsional dan nilai kesalahan (*error*) dari sistem. Karakteristik kontrol proporsional adalah merespons kesalahan saat ini dengan menghasilkan keluaran yang sebanding dengan besarnya kesalahan, tetapi dapat menyebabkan *overshoot* atau getaran jika diterapkan terlalu besar.[19]. Diagram blok pengontrol *proporsional* ditunjukkan pada **Gambar 2.9**.



Gambar 2.9 Diagram Blok Kontrol Proporsional [20]

Dari diagram blok diatas dapat diketahui persamaan kontrol Proportional ditunjukkan pada **Persamaan 2.17**:

$$u(t) = K_p e(t) \quad (2.17)$$

Dimana:

$u(t)$ = sinyal keluaran sistem kendali

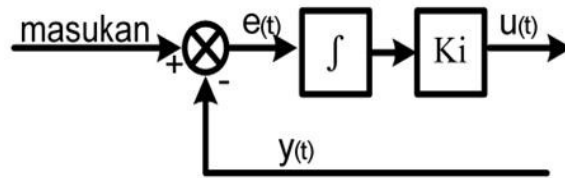
K_p = konstanta penguatan proporsional

$e(t)$ = sinyal error

2. Metode I (*Integral*)

Pengontrol integral berfungsi untuk menghasilkan respons sistem yang memiliki kesalahan stabil pada nilai nol. Bahkan jika sebuah plant tidak memiliki elemen integrator ($1/s$), pengontrol integral dapat memperbaiki respons sistem sehingga kesalahan dalam kondisi stabil menjadi nol. Pengontrol ini memiliki karakteristik yang serupa dengan operasi integral, di mana keluarannya dipengaruhi secara proporsional oleh perubahan sinyal kesalahan. Output dari pengontrol integral adalah akumulasi berkelanjutan dari perubahan inputnya. Jika tidak ada perubahan dalam sinyal kesalahan, output akan tetap pada kondisi sebelumnya. Karakteristik kontrol integral adalah membantu mengatasi kesalahan statis dan memberikan stabilitas jangka panjang untuk sistem. Namun, jika disetel terlalu tinggi, dapat menyebabkan overshoot atau

membuat sistem lambat dalam merespons perubahan.[19]. Diagram blok pengontrol integral ditunjukkan pada **Gambar 2.10**.



Gambar 2.10 Diagram Blok Kontrol Integral [20]

Dari diagram blok diatas dapat diketahui persamaan kontrol integral ditunjukkan pada **Persamaan 2.18**:

$$u(t) = K_i \int_0^t (t) dt \quad (2.18)$$

Dimana:

$u(t)$ = sinyal keluaran sistem kendali

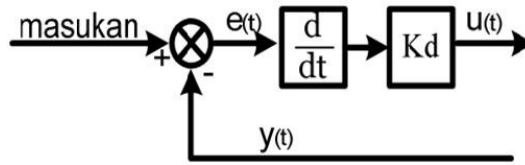
K_i = konstanta penguatan integral

$e(t)$ = sinyal error

dt = sinyal error

3. Metode D (*Derivative*)

Keluaran dari pengontrol diferensial memiliki karakteristik yang serupa dengan operasi diferensial. Ketika ada perubahan tiba-tiba pada masukan pengontrol, hal ini akan menyebabkan perubahan yang besar dan cepat pada keluaran. Pengontrol diferensial sering disebut sebagai pengendali laju karena keluarannya sebanding dengan laju perubahan sinyal kesalahan. Komponen diferensial menghitung laju perubahan kesalahan dan menghasilkan sinyal kontrol yang berlawanan dengan arah perubahan tersebut. Karakteristik kontrol derivatif membantu mengurangi *overshoot* dan menstabilkan respons sistem. Namun, jika diatur terlalu besar, dapat menyebabkan respons yang tidak stabil.[19]. Diagram blok pengontrol derivatif ditunjukkan pada **Gambar 2.11**.



Gambar 2.11 Diagram Blok Kontrol Derivatif [20]

Dari diagram blok diatas dapat diketahui persamaan kontrol derivatif ditunjukkan pada **Persamaan 2.19**:

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.19)$$

Dimana:

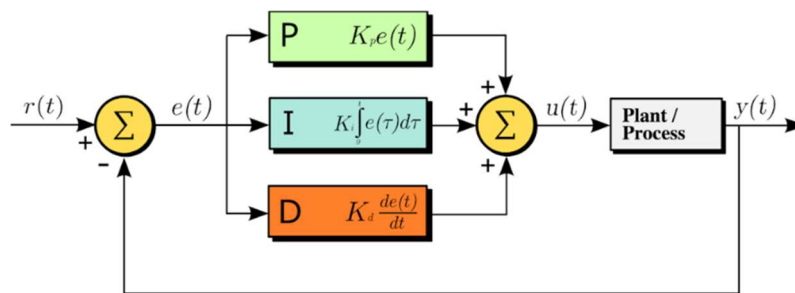
$u(t)$ = sinyal keluaran sistem kendali

K_d = konstanta penguatan derivatif

$e(t)$ = sinyal error

dt = sinyal error

Parameter-parameter tersebut tidak bersifat independen, sehingga pada saat salah satu nilai konstanta diubah maka kemungkinan sistem bereaksi tidak sesuai dengan yang diinginkan. Kendali PID dapat direpresentasikan dalam diagram blok seperti **Gambar 2.12** di bawah ini[21]:



Gambar 2.12 Diagram Blok Kendali PID [21]

Dari **Gambar 2.12** diperoleh persamaan kendali PID secara keseluruhan **Persamaan 2.20** berikut:

$$PID = K_p \cdot e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.20)$$

Keterangan:

$u(t)$ = Variabel yang diatur

$e(t)$ = Nilai kesalahan

K_p = Konstanta *proporsional*

K_i = Konstanta *integral*

K_d = Konstanta *derivative*

Dalam sistem ini, variabel yang dikendalikan adalah $u(t)$ dan $e(t)$, yang merupakan perbedaan antara nilai referensi dan nilai keluaran sistem. Parameter K_p , K_i , dan K_d adalah konstanta penguatan untuk aksi proporsional, integral, dan derivatif dalam pengendalian PID. Nilai K_i dapat dinyatakan sebagai K_p/T_i , dan K_d sebagai $K_p T_d$. [21]. Dengan demikian, persamaan kendali PID dapat ditulis pada **Persamaan 2.21** berikut:

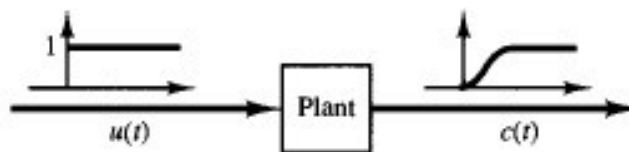
$$PID = K_p \cdot e(t) + \frac{K_p}{T_i} \int e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.21)$$

Keterangan:

Penggunaan T_i dan T_d menyatakan waktu integrasi (*integration time*) dan waktu turunan (*derivative time*).

2.7.1 Metode Ziegler-Nichols

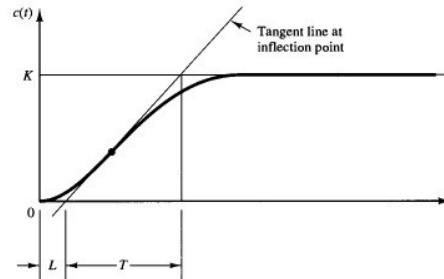
Metode Ziegler-Nichols adalah salah satu cara untuk menyetel parameter kontroler PID (Proportional, Integral, Derivative) yang dikembangkan oleh John G. Ziegler dan Nathaniel B. Nichols pada tahun 1942. Tujuan dari metode ini adalah untuk menentukan nilai konstanta kontrol PID yang paling optimal sehingga sistem dapat beroperasi dengan stabil dan responsif terhadap gangguan atau perubahan *setpoint* [22].



Gambar 2.13 Tanggapan Unit-Step [22]

Nilai PID yang diperoleh dari pengujian dengan memasukkan unit-step menunjukkan bahwa respons plant terhadap masukan tersebut menghasilkan kurva berbentuk S seperti yang terlihat pada **Gambar 2.13**. Kurva berbentuk S ini memiliki dua karakteristik utama, yaitu waktu tunda (L) dan konstanta waktu (T). Kedua parameter ini didapatkan dengan menggambar garis singgung pada titik

infleksi kurva S dan menemukan titik perpotongan garis singgung tersebut dengan sumbu waktu dan garis $c(t) = K$ seperti yang ditunjukkan pada **Gambar 2.14**[22].



Gambar 2.14 Kurva S [22]

Dari kurva tersebut dapat dilakukan pendekatan fungsi transfer dalam *first order* pada **Persamaan 2.22** berikut [22]:

$$\frac{C(s)}{U(s)} = \frac{Ke^{-Ls}}{Ts + 1} \quad (2.22)$$

Formula PID yang telah disebutkan sebelumnya, kemudian dijabarkan pada **Persamaan 2.23** sampai dengan **Persamaan 2.25** berikut[22]:

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2.23)$$

$$G_c(s) = 1.2 \frac{T}{L} \left(1 + \frac{1}{2Ls} + 0.5Ls \right) \quad (2.24)$$

$$G_c(s) = 0.6T \frac{\left(s + \frac{1}{L} \right)^2}{s} \quad (2.25)$$

Dengan menggunakan formula PID di atas dan nilai parameter L dan T, maka dapat diperoleh nilai K_i , T_i , dan T_d . Secara lebih ringkasnya dapat dilihat pada **Tabel 2.4**.

Tabel 2.4 Penentuan Parameter PID [22]

Tipe Pengontrol	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$\frac{1.2 T}{K L}$	2L	0.5L

2.8 Variable Frequency Drive (VFD)

Variable Frequency Drive (VFD) adalah alat elektronik yang berfungsi untuk mengontrol kecepatan motor listrik dengan mengubah frekuensi dan tegangan yang disuplai ke motor, terutama motor induksi arus bolak-balik (AC). VFD juga dikenal sebagai inverter, AC drive, atau *adjustable speed drive*. Prinsip kerja VFD melibatkan tiga tahap utama: penyearah, filter DC, dan inverter. Pada tahap pertama, arus AC dari sumber listrik diubah menjadi arus DC melalui rangkaian penyearah. Setelah itu, arus DC tersebut disaring dan distabilkan oleh komponen filter, sebelum akhirnya diubah kembali menjadi arus AC oleh inverter dengan frekuensi dan tegangan yang dapat disesuaikan sesuai kebutuhan motor.[5]. Perubahan frekuensi inilah yang menentukan kecepatan putar motor berdasarkan rumus yang ditunjukkan pada **Persamaan 2.26**.

$$RPM = \frac{120 \times f}{P} \quad (2.26)$$

Di mana:

f adalah frekuensi (Hz)

P adalah jumlah kutub motor.

Berdasarkan persamaan di atas, terlihat hubungan linear di mana kecepatan putar motor (RPM) berbanding lurus dengan frekuensi (f). Hal ini mendasari prinsip kerja sistem kendali pada penelitian ini, di mana VFD memanipulasi besaran frekuensi sebagai variabel kontrol utama untuk mengatur kecepatan pompa secara presisi guna menjaga kestabilan tekanan air. Spesifikasi VFD yang digunakan dalam penelitian ini dijelaskan pada **Tabel 2.5** berikut.

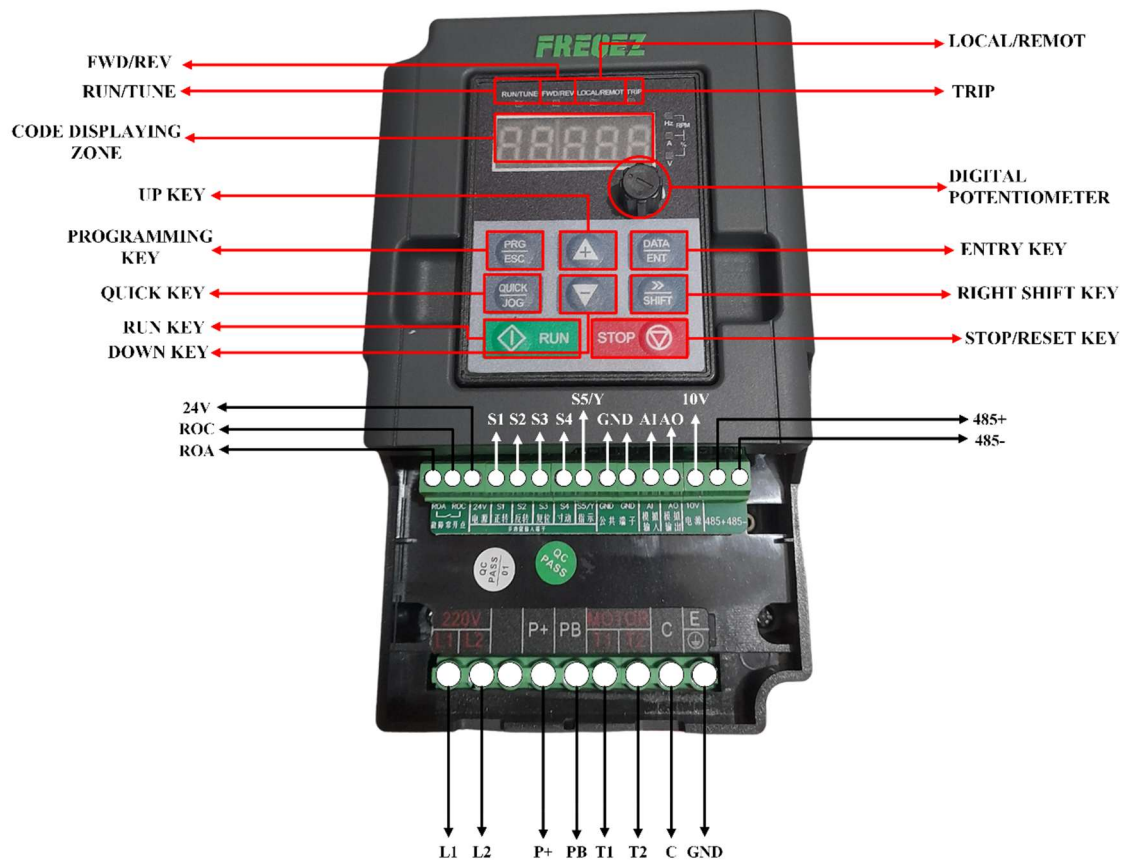
Tabel 2.5 Spesifikasi VFD *Single Phase Frequez* [23]

Parameter	Spesifikasi	Keterangan
Model Series	280 Series	AC Drive
Kapasitas Daya	0.75 kW (1 HP)	Sesuai beban motor
Tegangan Input	1-Phase 220 V $\pm 15\%$	Listrik PLN Standar
Frekuensi Input	47 – 63 Hz	-
Tegangan Output	0 – Input Voltage	220V AC
Arus Output Nominal	5.0 A	Kapasitas Arus
Rentang Frekuensi Output	0.00 – 400.00 Hz	Adjustable
Mode Kontrol	V/F Control (SVPWM)	Cocok untuk Pompa/Fan
Analog Input (AI)	0–10V DC / 4–20mA	Kompatibel dengan PLC CP1E
Kapasitas Overload	150% Arus Nominal	Selama 60 detik
Metode Pendinginan	Air Cooling	Kipas internal

a. Konfigurasi Terminal Daya (*Main Circuit*)

Berdasarkan fisik alat yang ditunjukkan pada **Gambar 2.15**, VFD ini memiliki konfigurasi terminal daya sebagai berikut [23]:

1. **Input Daya (L1, L2):** Terminal ini berfungsi menerima suplai tegangan jala-jala PLN satu fasa 220V.
2. **Output Motor (T1, T2, C):** Terminal ini adalah fitur khusus dari VFD tipe ini.
 - **T1 & T2:** Terhubung ke kumparan utama (*Main Winding*) motor pompa.
 - **C:** Terhubung ke kumparan bantu (*Auxiliary/Start Winding*) atau sisi kapasitor motor. (*Konfigurasi ini memungkinkan VFD mengatur torsi dan kecepatan motor satu fasa dengan lebih halus dibandingkan metode pemotongan tegangan biasa*).



Gambar 2.15 Konfigurasi Terminal Daya (Wiring) VFD

Detail konfigurasi terminal daya pada VFD disajikan pada **Tabel 2.6** yang menjabarkan fungsi dari terminal tegangan tinggi, meliputi terminal input (L1, L2) untuk sumber PLN dan terminal output khusus (T1, T2, C) yang terhubung langsung ke motor pompa satu fasa [23].

Tabel 2.6 Konfigurasi Terminal Sirkuit Utama (*Main Circuit*) [23]

Label Terminal	Nama Terminal	Fungsi / Deskripsi
L1, L2	<i>AC Power Input</i>	Terminal masukan sumber tegangan PLN 1 Fasa (220 VAC). L1 untuk Fasa (Line) dan L2 untuk Netral.
P+, PB	<i>Braking Resistor</i>	Terminal penghubung untuk resistor pengereman (<i>braking resistor</i>) eksternal. (Opsional, digunakan jika beban motor memiliki inersia tinggi).
T1, T2	<i>Motor Output</i>	Terminal keluaran tegangan variabel ke motor induksi satu fasa. T1 dan T2 terhubung ke gulungan utama (<i>run/main winding</i>) dan gulungan bantu (<i>start</i>) sesuai konfigurasi motor.
C	<i>Capacitor Connection</i>	Terminal khusus untuk sambungan kapasitor motor (jika menggunakan konfigurasi kapasitor dipisah/ <i>removed capacitor</i>).
GND / E	<i>Grounding</i>	Terminal pembumian (<i>earth ground</i>) untuk proteksi keselamatan dan mengurangi interferensi elektromagnetik.

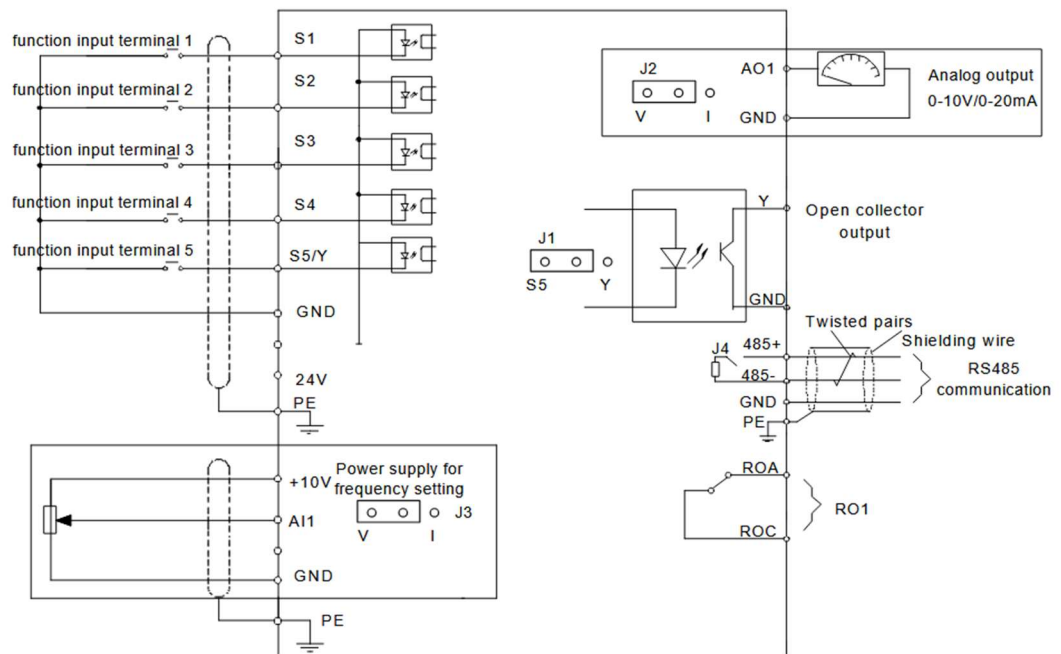
Deskripsi fungsi dari masing-masing tombol dan indikator pada *keypad* yang ditunjukkan **Gambar 2.15** dijabarkan pada **Tabel 2.7** berikut.

Tabel 2.7 Deskripsi Fungsi Tombol dan Indikator *Keypad* VFD [23]

Simbol / Nama	Fungsi / Keterangan
PRG / ESC	Tombol Program/Keluar. Digunakan untuk masuk ke menu parameter atau membatalkan perubahan dan kembali ke menu sebelumnya.
DATA / ENT	Tombol Enter. Digunakan untuk masuk ke level menu yang lebih dalam dan menyimpan (<i>save</i>) nilai parameter yang telah diubah.
UP	Tombol Naik. Berfungsi untuk menaikkan nilai parameter atau nomor fungsi.
DOWN	Tombol Turun. Berfungsi untuk menurunkan nilai parameter atau nomor fungsi.
SHIFT	Tombol Geser. Digunakan untuk memindahkan kursor digit aktif saat mengubah nilai parameter, atau mengganti parameter monitoring pada layar saat beroperasi.
RUN	Tombol Jalan. Berfungsi untuk memberikan perintah <i>Start</i> (menjalankan motor) saat mode kontrol keypad aktif.
STOP / RST	Tombol Stop/Reset. Berfungsi untuk menghentikan motor atau mereset alarm saat VFD mengalami <i>fault</i> (gangguan).
QUICK / JOG	Tombol Multifungsi. Dapat diatur fungsinya melalui parameter (misalnya untuk mode <i>Jogging</i> atau pintasan menu).
Potensiometer	Kenop putar untuk mengatur frekuensi referensi secara manual dan <i>real-time</i> .
Indikator LED	Lampu status yang menunjukkan kondisi VFD (RUN/TUNE untuk status jalan, FWD/REV untuk arah putaran, TRIP untuk indikasi error).

b. Konfigurasi Terminal Kontrol (*Control Circuit*)

Terminal kontrol berfungsi sebagai antarmuka sinyal bertegangan rendah untuk menghubungkan VFD dengan kontroler eksternal (PLC) dan sensor. Diagram skematik internal sirkuit kontrol dapat dilihat pada **Gambar 2.16** [23].



Gambar 2.16 Rangkaian Skematik VFD AT2 [23]

Selanjutnya, **Tabel 2.8** menunjukkan konfigurasi terminal sirkuit kontrol. Terminal ini berfungsi sebagai antarmuka sinyal bertegangan rendah yang menghubungkan VFD dengan sistem kendali eksternal (PLC), mencakup input digital untuk perintah Start/Stop dan input analog untuk pengaturan frekuensi referensi [23].

Tabel 2.8 Fungsi Terminal Kontrol [23]

Label Terminal	Nama Terminal	Fungsi / Deskripsi
ROA, ROC	<i>Relay Output</i>	Output relay multifungsi. Kontak akan menutup (NO) saat kondisi tertentu terpenuhi (misal: VFD <i>Running</i> atau <i>Fault</i>). Kapasitas kontak: 3A/250VAC.
24V	<i>Power Output</i>	Keluaran tegangan DC 24V (maks. 100mA) untuk suplai daya sensor eksternal atau saklar digital.
S1, S2, S3, S4	<i>Digital Input</i>	Terminal input digital multifungsi (dapat diprogram). Digunakan untuk perintah Start/Stop (<i>Forward/Reverse</i>) atau <i>Multi-step speed</i> .
S5 / Y	<i>Digital In/Out</i>	Terminal multifungsi yang dapat diatur sebagai <i>Digital Input</i> ke-5 atau sebagai <i>High Speed Pulse Output</i> .
GND	<i>Signal Ground</i>	Titik nol (ground) untuk sinyal analog dan digital (referensi 0V untuk terminal 24V dan 10V).
AI	<i>Analog Input</i>	Terminal input analog untuk pengaturan frekuensi referensi. Mendukung sinyal tegangan (0-10V) atau arus (0-20mA) yang dipilih melalui jumper J3.
AO	<i>Analog Output</i>	Terminal output analog (0-10V atau 0-20mA). Digunakan untuk mengirim sinyal status (seperti frekuensi berjalan) ke meteran eksternal atau PLC.
10V	<i>Power Reference</i>	Keluaran tegangan referensi stabil +10V. Digunakan sebagai sumber tegangan untuk potensiometer eksternal.
485+, 485-	<i>Communication</i>	Terminal komunikasi serial RS485 (Modbus RTU) untuk pengendalian dan pemantauan jarak jauh via PC atau PLC.

Setelah memahami fungsi tombol pada keypad dan memastikan pengkabelan terpasang dengan benar, langkah selanjutnya adalah melakukan konfigurasi parameter internal. Parameter ini berfungsi sebagai instruksi logika bagi VFD untuk mengendalikan motor, seperti penentuan sumber perintah

start/stop, batas frekuensi kerja, hingga karakteristik akselerasi. Parameter pengaturan VFD dapat dilihat pada **Tabel 2.9** [23].

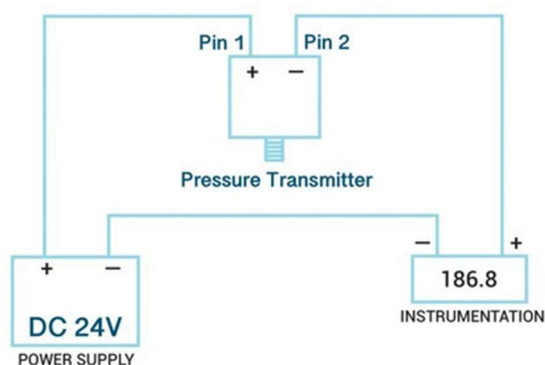
Tabel 2.9 Parameter Pengaturan VFD [23]

<i>Function Code</i>	<i>Parameter specification</i>	<i>Parameter Range</i>	<i>Default</i>
P00.00	<i>Speed control mode</i>	2: SVPWM control	2
P00.01	<i>Channel of running commands</i>	0: Keypad	0
P00.03	<i>Max. output frequency</i>	50 Hz	50 Hz
P00.04	<i>Upper limit of the running frequency</i>	50 Hz	50 Hz
P00.05	<i>Lower limit of the running frequency</i>	0 Hz	0 Hz
P00.06	<i>A frequency command selection</i>	0: Keypad 1: Analog AI 2: Analog AI2 setting	0
P00.11	<i>ACC time 1</i>	<i>ACC time means the time needed if the VFD speeds up from 0 Hz to the Max</i>	<i>Depend on model</i>
P00.12	<i>DEC time 1</i>	<i>DEC time means the time needed if the VFD speeds down from the Max. Output frequency to 0Hz (P00.03)</i>	<i>Depend on model</i>
P02.01	<i>Asynchronous motor rated power</i>	0.1–3000.0kW	<i>Depend on model</i>
P02.02	<i>Asynchronous motor rated frequency</i>	0.01 Hz–50 Hz	50 Hz

2.9 Pressure Transmitter

Pressure transmitter adalah alat sensor yang berfungsi untuk mengukur tekanan fluida, baik gas maupun cairan, dan mengonversinya menjadi sinyal listrik yang dapat diinterpretasikan oleh sistem kontrol atau perangkat pemantauan. Salah satu jenis output yang sering digunakan adalah sinyal arus 4–20 mA dalam konfigurasi 2-wire, di mana arus ini menggambarkan rentang tekanan yang diukur.

Konfigurasi *2-wire* ini memungkinkan perangkat untuk mendapatkan daya sekaligus mengirimkan sinyal melalui dua kabel yang sama, sehingga mempermudah instalasi dan mengurangi biaya kabel. Wiring dari *pressure transmitter* dapat dilihat pada **Gambar 2.17** [24].



Gambar 2.17 Wiring Pressure Transmitter [24]

Pressure transmitter dengan output 4–20 mA banyak digunakan dalam berbagai aplikasi industri, seperti sistem hidrolik, pneumatik, pengolahan air, dan proses manufaktur, karena kemampuannya dalam memberikan sinyal yang stabil dan tahan terhadap gangguan elektromagnetik. Selain itu, sinyal arus ini tidak terpengaruh oleh resistansi kabel, memungkinkan transmisi data yang akurat bahkan pada jarak yang jauh. Spesifikasi dari *pressure transmitter* yang digunakan dijelaskan pada **Tabel 2.10**.

Tabel 2.10 Spesifikasi *Pressure Transmitter* [25]

Parameter	Spesifikasi
Rentang Tekanan	0–5 bar (0–500 kPa / 72,519 psi)
Output Sinyal	4–20 mA (2-wire)
Catu Daya	24 VDC
Koneksi Proses	G1/2 (BSPP)
Material Housing	Stainless Steel
Akurasi	±0.5% FS
Suhu Operasi	-40 hingga 85°C

Sinyal keluaran dari *pressure transmitter* berupa arus listrik analog dengan rentang 4–20 mA. Agar data tersebut dapat diproses oleh *Programmable Logic Controller* (PLC) sebagai satuan tekanan (Bar), diperlukan proses konversi data

atau scaling. Dalam sistem kendali digital, sinyal analog 4–20 mA pertama-tama dikonversi oleh modul *Analog-to-Digital Converter* (ADC) pada PLC menjadi nilai digital mentah (*raw value*), misalnya 0–6000 (tergantung resolusi PLC). Selanjutnya, nilai digital tersebut dikonversi kembali menjadi satuan teknik (*engineering unit*) menggunakan persamaan linear. Dalam penelitian ini, karena rentang ukur *pressure transmitter* dimulai dari 0 Bar, maka persamaan linear dapat disederhanakan menggunakan metode perbandingan rasio. Pada pemrograman *Ladder Diagram*, rumus konversi yang digunakan terdapat pada **Persamaan 2.27**.

$$Pressure = \frac{Data\ Input}{Resolusi\ PLC} \times Max\ Pressure\ (HSC) \quad (2.27)$$

Keterangan:

Pressure : Nilai Tekanan Aktual (disimpan di D8).

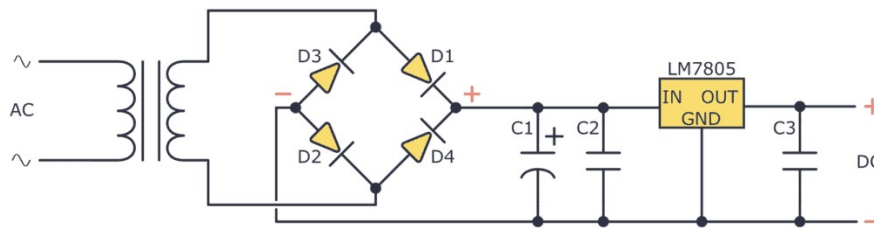
Data Input : Nilai pembacaan sensor pada alamat 90 (Range 0-6000).

Resolusi PLC : Nilai konstanta 6000 (Resolusi 12-bit/Special I/O).

Max Pressure : Kapasitas maksimum sensor yang digunakan.

2.10 Power Supply

Power Supply (catu daya) adalah elemen krusial dalam sistem elektronik, berperan dalam menyediakan energi listrik yang dibutuhkan oleh perangkat. Biasanya, power supply mengubah arus listrik bolak-balik (AC) dari sumber utama menjadi arus searah (DC) yang stabil. Stabilitas *output* dari *power supply* sangat penting dalam sistem kontrol PID, karena fluktuasi tegangan dapat menyebabkan ketidakakuratan dalam pengukuran dan pengendalian proses. Selain itu, *power supply* juga harus mampu menyediakan arus yang cukup untuk mengoperasikan aktuator seperti motor, memastikan bahwa perangkat tersebut berfungsi pada tegangan dan arus yang sesuai. Dalam konteks ini, penggunaan *power supply* DC 24V dengan kapasitas arus 5A menjadi pilihan yang tepat untuk mendukung kinerja sistem secara efisien dan stabil.[26]. Rangkaian skematik *Power Supply* dapat dilihat pada **Gambar 2.18**.



Gambar 2.18 Rangkaian Skematik Power Supply [27]

2.11 Pompa Air AC

Pompa air adalah alat mekanis yang berfungsi untuk memindahkan air dari satu lokasi ke lokasi lain dengan cara mentransfer energi dari motor ke cairan. Dalam sistem rumah tangga serta skala kecil hingga menengah, pompa air satu fasa sering digunakan karena praktis, efisien, dan memadai untuk kebutuhan tekanan dan aliran air yang sedang. Pompa air satu fasa digerakkan oleh motor listrik satu fasa yang beroperasi dengan arus bolak-balik (AC) 220V. Motor ini dilengkapi dengan kapasitor (*start* atau *run capacitor*) untuk membantu menciptakan medan magnet berputar, karena motor satu fasa secara alami tidak memiliki torsi awal. Kapasitor ini memungkinkan motor untuk berputar dan menjaga arah rotasinya [5].

2.11.1 Konstruksi dan Prinsip Kerja Mekanik

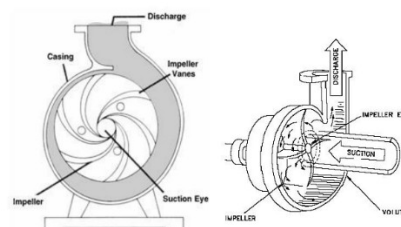
Pompa air satu fasa umumnya termasuk dalam kategori pompa sentrifugal, di mana cara kerjanya mengandalkan impeller yang berputar untuk memberikan energi kinetik pada air. Energi ini kemudian diubah menjadi energi tekanan saat air keluar dari impeller dan masuk ke volute atau casing pompa. Pompa sentrifugal cocok untuk distribusi air bersih karena mampu menghasilkan aliran yang stabil dan tekanan yang cukup untuk mengalir jaringan pipa rumah tangga atau industri kecil.[5]. Motor yang digunakan dalam pompa air AC bisa berupa motor sinkron atau motor induksi. Motor sinkron berputar dengan kecepatan yang sesuai dengan frekuensi listrik yang disuplai, sedangkan motor induksi berputar sedikit lebih lambat dari kecepatan sinkron, tergantung pada beban yang diterima. Motor AC bekerja dengan memanfaatkan medan magnet yang dihasilkan dari arus listrik yang mengalir melalui lilitan kumparan. Medan magnet yang berputar ini menciptakan gaya putar atau torsi, yang digunakan untuk memutar rotor motor. Torsi ini kemudian digunakan untuk menggerakkan impeller pada pompa, sehingga fluida

dapat dialirkan dengan peningkatan kecepatan dan tekanan.[28]. Spesifikasi pompa Shimizu PS-128 bit yang digunakan dijelaskan pada **Tabel 2.11** sebagai berikut.

Tabel 2.11 Spesifikasi Pompa Air Shimizu PS-128 bit

Parameter	Spesifikasi
Merek / Model	Shimizu / PS-128 BIT
Jenis Motor	Induksi 1-Fasa (<i>Single Phase</i>)
Tegangan Kerja	220 VAC
Frekuensi	50 Hz
Arus Nominal (I_n)	1.3 Ampere
Daya Output Motor (P_2)	125 Watt
Daya Input (P_1)	≈ 300 Watt
Kecepatan Putar (RPM)	2900 rpm
Kapasitas Kapasitor	$8\mu F/450V$
Daya Hisap Maksimum	9 m
Daya Dorong Maksimum	24 m
Total Head (H_{max})	33 m
Debit Air Maksimum (Q_{max})	34 liter/menit

Pompa sentrifugal merupakan jenis pompa dinamis yang paling sering digunakan dalam sistem pemompaan fluida, terutama untuk air bersih dan cairan dengan viskositas rendah. Cara kerja pompa ini didasarkan pada prinsip sentrifugal, di mana energi kinetik yang dihasilkan oleh impeller yang berputar diubah menjadi energi tekanan untuk mengalirkan fluida. Dalam pompa sentrifugal, fluida masuk ke bagian tengah impeller (mata impeller) dan didorong keluar melalui saluran impeller oleh gaya sentrifugal yang dihasilkan dari putaran impeller. Proses ini meningkatkan kecepatan fluida. Selanjutnya, kecepatan tersebut diubah menjadi tekanan saat fluida melewati volute atau diffuser casing, yang dirancang untuk memperlambat aliran dan meningkatkan tekanannya[29]. Bagian dari konstruksi pompa sentrifugal dapat dilihat pada **Gambar 2.19**.



Gambar 2.19 Bagian dari konstruksi pompa sentrifugal [30]

Bagian-bagian utama pompa sentrifugal diantaranya sebagai berikut [31]:

1. *Impeller*, merupakan komponen utama yang berputar dan bertugas memberikan energi kinetik kepada fluida.
2. *Casing (Volute atau Spiral Casing)*, untuk mengumpulkan fluida yang telah dipercepat oleh impeller dan mengubah energi kinetik menjadi tekanan.
3. *Suction Pipe* (Pipa Hisap), saluran masuk fluida ke pompa, mengarah ke pusat impeller (*eye*). Biasanya dipasang *check valve* atau *foot valve* untuk mencegah aliran balik saat pompa berhenti.
4. *Discharge Pipe* (Pipa Keluar), saluran keluarnya fluida dari casing setelah mengalami peningkatan tekanan. Terhubung ke sistem distribusi atau tangki penampung.
5. *Shaft* (Poros), menghubungkan impeller dengan motor penggerak. Shaft mentransmisikan torsi dari motor ke impeller dan biasanya dilengkapi dengan seal dan bearing untuk menghindari kebocoran dan mengurangi gesekan.
6. *Bearing* (Bantalan), menopang poros dan menjaga poros tetap stabil selama berputar.
7. *Mechanical Seal* atau *Gland Packing*, bertugas mencegah kebocoran fluida dari ruang bertekanan tinggi keluar melalui celah poros.
8. *Coupling*, alat penghubung antara poros motor dan poros pompa. Memungkinkan transmisi daya dengan fleksibilitas terhadap sedikit ketidaksejajaran antar poros.

2.11.2 Karakteristik Daya Listrik Motor Induksi

Motor pompa air AC merupakan beban listrik yang bersifat induktif karena menggunakan kumparan (lilitan kawat) untuk membangkitkan medan magnet. Karakteristik beban induktif ini menyebabkan terjadinya pergeseran fasa antara tegangan dan arus, sehingga daya listrik yang dikonsumsi terbagi menjadi tiga komponen utama yang digambarkan dalam Segitiga Daya (*Power Triangle*), yaitu:

- a. Daya Semu (*Apparent Power, S*). Daya semu adalah kapasitas daya total yang disuplai oleh sumber listrik PLN ke beban. Nilai ini merupakan hasil perkalian

murni antara tegangan dan arus terukur. Rumus daya semu ditunjukkan pada **Persamaan 2.28**.

$$S = V \times I \quad (2.28)$$

Keterangan:

S : Daya Semu (*Apparent Power*) dalam satuan Volt-Ampere (VA).

V : Tegangan sumber (*Voltage*) dalam satuan Volt (V).

I : Arus listrik (*Current*) dalam satuan Ampere (A).

- b. Daya Aktif (*Active Power*, P). Daya aktif atau daya nyata adalah daya listrik yang benar-benar dikonversi oleh motor menjadi energi mekanik (putaran poros) dan kerugian panas. Daya inilah yang merepresentasikan beban kerja sesungguhnya dari pompa. Rumus daya aktif ditunjukkan pada **Persamaan 2.29** sampai **Persamaan 2.30**.

$$P = S \times \cos \phi \quad (2.29)$$

$$P = V \times I \times \cos \phi \quad (2.30)$$

Keterangan:

P : Daya Aktif (*Active Power*) dalam satuan Watt (W).

V : Tegangan sumber (*Voltage*) dalam satuan Volt (V).

I : Arus listrik (*Current*) dalam satuan Ampere (A).

$\cos \phi$: Faktor Daya (*Power Factor*), nilai efisiensi tanpa satuan.

- c. Daya Reaktif (*Reactive Power*, Q). Daya reaktif adalah daya yang tidak menghasilkan kerja nyata, namun diperlukan untuk membangkitkan medan magnet pada kumparan stator motor agar rotor dapat berputar. Rumus daya reaktif ditunjukkan pada **Persamaan 2.31**.

$$Q = S \times \sin \phi \quad (2.31)$$

Keterangan:

Q : Daya Reaktif (*Reactive Power*) dalam satuan Volt-Ampere Reactive (VAR).

V : Tegangan sumber (*Voltage*) dalam satuan Volt (V).

I : Arus listrik (*Current*) dalam satuan Ampere (A).

$\sin \phi$: Faktor Reaktif (*Reactive Factor*).

Ketiga jenis daya tersebut memiliki hubungan matematis yang dikenal sebagai Segitiga Daya. Rasio perbandingan antara Daya Aktif (P) dan Daya Semu (S) disebut sebagai **Faktor Daya ($\cos \phi$)**. Faktor daya bukanlah jenis energi, melainkan nilai efisiensi yang menunjukkan seberapa efektif daya semu diubah menjadi kerja nyata. Rumus hubungan faktor daya ditunjukkan pada **Persamaan 2.32**.

$$\cos \phi = \frac{P}{S} \quad (2.32)$$

Keterangan:

$\cos \phi$: Faktor Daya (*Power Factor*), bernilai antara 0.0 hingga 1.0.

P : Daya Aktif (Watt).

S : Daya Semu (VA).

2.12 Buzzer

Buzzer adalah perangkat output elektronik yang berfungsi untuk menghasilkan suara sebagai tanda peringatan atau notifikasi. Dalam sistem otomatisasi yang menggunakan *Programmable Logic Controller* (PLC), *buzzer* sering dipakai sebagai alarm suara untuk memberi tahu operator tentang kondisi tertentu, seperti kesalahan, batas parameter yang terlampaui, atau selesainya suatu proses. *Buzzer* dapat dihubungkan langsung ke *output* digital PLC, terutama jika jenisnya adalah *buzzer* aktif, yaitu *buzzer* yang dapat beroperasi hanya dengan suplai tegangan tanpa memerlukan sinyal frekuensi eksternal. *Buzzer* aktif biasanya diberi tegangan DC (5V, 12V, atau 24V tergantung spesifikasinya), dan PLC cukup memberikan sinyal ON/OFF melalui output digitalnya. Jika konsumsi arus *buzzer* lebih tinggi dari kemampuan output PLC, atau jika *buzzer* memerlukan tegangan AC, maka *buzzer* dapat dikendalikan menggunakan *relay output*. Pemrograman *buzzer* pada ladder diagram PLC umumnya diatur melalui sinyal logika tertentu, misalnya *buzzer* aktif saat suhu melebihi ambang batas atau ketika sensor mendeteksi anomali dalam proses. Penggunaan *buzzer* meningkatkan responsivitas

sistem, karena operator dapat segera menyadari kondisi kritis tanpa harus terus-menerus memantau tampilan visual atau HMI. [32].

2.13 CX Programmer

CX-Programmer adalah perangkat lunak yang dibuat oleh Omron untuk memprogram dan mengatur berbagai jenis PLC (Programmable Logic Controller) yang diproduksi oleh Omron. Perangkat lunak ini termasuk dalam paket CX-One, yang menawarkan solusi terpadu untuk otomasi industri, mencakup pemrograman PLC, desain antarmuka manusia-mesin (HMI), dan simulasi sistem.[33]. Dengan menggunakan CX-Programmer, pengguna dapat menyusun program kontrol memakai berbagai bahasa pemrograman yang sesuai dengan standar IEC 61131-3, seperti ladder diagram dan *structured text*. Perangkat lunak ini juga menawarkan fitur-fitur seperti debugging, pemantauan secara real-time, dan simulasi, yang memungkinkan pengujian serta pemeliharaan program tanpa harus terhubung langsung ke perangkat keras PLC. [34]. Salah satu keunggulan CX-Programmer adalah kemampuannya untuk berintegrasi dengan perangkat lunak lain dalam paket CX-One, seperti CX-Designer untuk desain HMI dan CX-Simulator untuk simulasi sistem. Ini memudahkan proses pengembangan sistem otomasi secara keseluruhan, mulai dari tahap perancangan hingga implementasi dan pemeliharaan.[33].

2.14 Haiwell Cloud Scada

Haiwell Cloud SCADA adalah platform perangkat lunak *Supervisory Control and Data Acquisition* (SCADA) berbasis *cloud computing* yang dikembangkan untuk memfasilitasi pemantauan dan pengendalian sistem industri secara jarak jauh. Berbeda dengan sistem SCADA konvensional yang beroperasi secara lokal (*on-premise*), Haiwell Cloud SCADA menawarkan integrasi antara perangkat keras industri dengan jaringan internet, memungkinkan visualisasi data secara *real-time* melalui berbagai terminal seperti PC, tablet, maupun telepon pintar [35].

Secara arsitektur, sistem ini terdiri dari tiga komponen fundamental yang saling terintegrasi:

1. Haiwell Cloud SCADA Designer: Lingkungan pengembangan (*development environment*) yang digunakan perancang untuk membangun antarmuka grafis

(*Human Machine Interface*), mengonfigurasi *tag* data, dan menyusun logika skrip.

2. Haiwell Cloud Engine: Mesin inti yang berfungsi sebagai server manajemen data, menangani komunikasi antara perangkat lapangan (*field devices*) dan antarmuka pengguna.
3. Haiwell Cloud App/Web: Terminal akses bagi pengguna akhir untuk memantau status sistem dan menerima notifikasi alarm di mana saja selama terhubung dengan internet [35].

Keunggulan utama Haiwell Cloud SCADA dalam implementasi sistem otomatisasi modern adalah fleksibilitas konektivitasnya. Perangkat lunak ini mendukung berbagai protokol komunikasi industri standar, termasuk Modbus TCP/IP, Modbus RTU, serta protokol *proprietary* dari berbagai produsen PLC terkemuka seperti Omron, Siemens, dan Mitsubishi. Hal ini memungkinkan integrasi langsung dengan PLC tanpa memerlukan perangkat keras *gateway* tambahan yang kompleks[36].

Selain itu, Haiwell Cloud SCADA telah mengadopsi protokol MQTT (*Message Queuing Telemetry Transport*). MQTT merupakan protokol komunikasi ringan yang didesain khusus untuk kondisi *bandwidth* rendah dan latensi tinggi, sehingga sangat efisien untuk pengiriman data telemetri dalam ekosistem *Internet of Things* (IoT). Penggunaan protokol ini menjamin kecepatan transmisi data dari sensor ke *cloud* tetap stabil meskipun pada jaringan yang terbatas [37].

2.15 MATLAB

MATLAB (*Matrix Laboratory*) adalah perangkat lunak dan bahasa pemrograman tingkat tinggi yang dikembangkan oleh MathWorks Inc. Dirancang khusus untuk komputasi numerik, analisis data, dan visualisasi, MATLAB memanfaatkan struktur data berbasis matriks, menjadikannya alat yang efisien dalam menyelesaikan masalah teknik dan ilmiah. Awalnya dikembangkan sebagai antarmuka untuk koleksi rutin numerik dari proyek LINPACK dan EISPACK, MATLAB telah berkembang menjadi platform yang mendukung berbagai aplikasi, termasuk pemodelan, simulasi, dan pengembangan algoritma. Dalam lingkungan

MATLAB, pengguna dapat mengintegrasikan komputasi, visualisasi, dan pemrograman dalam satu antarmuka yang intuitif. Hal ini memungkinkan ekspresi masalah dan solusi dalam notasi matematika yang familiar, memudahkan proses analisis dan interpretasi data. Fitur-fitur seperti *toolbox* khusus untuk pengolahan sinyal, sistem kontrol, logika fuzzy, dan pengolahan citra digital memperluas kemampuan MATLAB dalam berbagai bidang teknik dan sains. MATLAB juga mendukung integrasi dengan bahasa pemrograman lain seperti C, Java, dan Python, serta dapat dijalankan pada berbagai sistem operasi, termasuk Windows, Linux, dan macOS. Kemampuan ini menjadikan MATLAB sebagai alat yang fleksibel dan powerful dalam pengembangan aplikasi teknik dan ilmiah.[38].

2.16 SolidWork

SolidWorks adalah perangkat lunak Computer-Aided Design (CAD) berbasis parametrik yang dikembangkan oleh Dassault Systèmes. Perangkat lunak ini banyak digunakan di bidang teknik dan manufaktur untuk merancang dan memodelkan objek dalam bentuk tiga dimensi (3D) maupun dua dimensi (2D). SolidWorks memungkinkan pengguna untuk membuat desain produk, melakukan analisis struktural, dan menyiapkan data yang diperlukan untuk proses manufaktur. Salah satu keunggulan SolidWorks adalah kemampuannya dalam pemodelan parametrik, yang memungkinkan perubahan desain dilakukan dengan mudah dan efisien. Selain itu, SolidWorks menyediakan fitur simulasi yang memungkinkan pengguna untuk menguji kinerja desain secara virtual sebelum diproduksi, sehingga dapat mengidentifikasi dan memperbaiki potensi masalah sejak dini. Dalam konteks pendidikan, SolidWorks telah diterapkan sebagai alat bantu pembelajaran untuk meningkatkan kompetensi siswa dalam menggambar teknik dan memahami konsep desain produk. Penggunaan modul pembelajaran SolidWorks di sekolah menengah kejuruan (SMK) telah terbukti efektif dalam meningkatkan kemampuan siswa dalam membuat model 3D berbasis CAD.[39].

2.17 Microsoft Visio

Microsoft Visio adalah aplikasi pembuat diagram dan grafik vektor yang dikembangkan sebagai bagian dari keluarga Microsoft Office, yang dirancang khusus untuk membantu pengguna memvisualisasikan informasi teknis yang kompleks menjadi representasi grafis yang terstandarisasi. Berbeda dengan perangkat lunak desain grafis konvensional, Visio berfokus pada ketepatan teknis dengan menyediakan berbagai pustaka template dan stencil (cetakan simbol) baku yang mempermudah insinyur dalam mendokumentasikan sistem [40].

Dalam perancangan sistem kendali, perangkat lunak ini memiliki peran vital pada dua tahapan desain. Pertama, untuk perancangan logika, Visio menyediakan simbol standar untuk pembuatan Diagram Blok arsitektur sistem dan Flowchart (Diagram Alir) algoritma, yang berfungsi memetakan alur logika pemrograman sebelum diimplementasikan ke dalam PLC[41]. Kedua, untuk perancangan elektrik, fitur Electrical Engineering pada Visio digunakan untuk menggambar Wiring Diagram (rangkaiian pengkabelan). Fitur konektor cerdas (*smart connector*) memungkinkan jalur kabel tetap terhubung pada komponen seperti relay dan motor meskipun posisi komponen diubah, sehingga menjamin akurasi desain panel kontrol [42].