
Labsheet Praktik PLC

Belajar PLC dengan mudah dengan Trainer PLC
Omron

Ardy Seto Priambodo, Aris Nasuha, Yusril Ali Riza
Pratama

Daftar Isi

1	Tutorial	1
1.1	Referensi gambar dan tabel	1
1.2	Memasukkan gambar	1
1.3	Memasukkan tabel	2
1.4	Membuat note, tip, warning, caution, dan important	2
1.5	Rumus matematika	2
1.6	Kode program	2
1.7	Teks, link, dan daftar	3
1.8	Contoh hasil perplexity ai	3
1.9	Cara Kerja PID Control	3
1.10	Komponen Utama PID	4
1.11	Cara Kerja Secara Umum	4
1.12	Ilustrasi Sederhana	4
1.13	Penyesuaian (Tuning) PID	5
1.14	Aplikasi	5
1.15	Kesimpulan	5
2	Pengenalan Sistem Kendali	7
2.1	Dasar Sistem Kendali	7
3	Persiapan Simulasi Berbasis Python	9
3.1	Instalasi dan Pengelolaan Lingkungan Python	9
3.1.1	Instalasi Anaconda	9
3.1.2	Pengelolaan Virtual Environment	9

Draft by ardyseto

1 Tutorial

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1.1 Referensi gambar dan tabel

Gambar 1.1 merupakan contoh gambar. Tabel 1.1 adalah contoh tabel.

1.2 Memasukkan gambar



Gambar 1.1: Contoh ini Caption Gambar

1.3 Memasukkan tabel

Tabel 1.1: Contoh ini Caption Tabel

Komponen	Deskripsi Singkat
Sensor	Mengukur proses
Kontrol	Mengolah error dan menghasilkan aksi kendali
Aktuator	Menggerakkan plant sesuai sinyal kendali

1.4 Membuat note, tip, warning, caution, dan important

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquet libero quis lectus elementum fermentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquet libero quis lectus elementum fermentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquet libero quis lectus elementum fermentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquet libero quis lectus elementum fermentum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquet libero quis lectus elementum fermentum.

1.5 Rumus matematika

Penulisan rumus inline $e(t) = SP - PV$. Penulisan rumus dalam baris baru:

$$e(t) = SP - PV$$

1.6 Kode program

Penulisan kode program dalam baris `print("Hello, World!")`. Penulisan kode program dalam blok:

```
1 def prime_numbers(n):
2     """Mengembalikan daftar bilangan prima hingga n."""
3     primes = []
4     for num in range(2, n + 1):
5         is_prime = True
6         for i in range(2, int(num**0.5) + 1):
7             if num % i == 0:
8                 is_prime = False
9                 break
10        if is_prime:
11            primes.append(num)
12    return primes
13 print(prime_numbers(20))
```

1.7 Teks, link, dan daftar

Penulisan bold **bold** dan italic *italic*. Pembuatan link Google. penulisan list dengan tanda –:

- Item pertama
- Item kedua
- Item ketiga

Penulisan list dengan angka:

1. Item pertama
2. Item kedua
3. Item ketiga

1.8 Contoh hasil perplexity ai

1.9 Cara Kerja PID Control

Pengertian Dasar Kontrol PID (Proportional-Integral-Derivative) adalah algoritma kontrol umpan balik yang sangat umum digunakan di industri untuk menjaga suatu variabel proses (seperti suhu, tekanan, kecepatan) agar tetap pada nilai yang diinginkan (setpoint). Kontroler PID bekerja dengan membandingkan nilai aktual (feedback) dengan nilai setpoint, kemudian menghitung selisihnya (error), dan menghasilkan sinyal kontrol berdasarkan tiga komponen utama: Proporsional (P), Integral (I), dan Derivatif (D)[1][3][2].

1.10 Komponen Utama PID

1. Proporsional (P) - Komponen proporsional menghasilkan output yang sebanding dengan besarnya error saat ini. - Semakin besar error, semakin besar pula respon kontrol yang diberikan. - Fungsi utamanya adalah mempercepat respon sistem terhadap perubahan setpoint, tetapi tidak selalu menghilangkan error sepenuhnya[1][2][3][7].

2. Integral (I) - Komponen integral mengakumulasi error dari waktu ke waktu. - Berfungsi untuk mengatasi error yang tersisa (offset) yang tidak bisa dihilangkan oleh komponen proporsional saja. - Integral akan terus menambah output hingga error benar-benar nol, sehingga sistem menjadi lebih akurat[1][2][3][7].

3. Derivatif (D) - Komponen derivatif merespons terhadap laju perubahan error. - Membantu mengantisipasi perubahan cepat dan meredam osilasi atau overshoot, sehingga sistem menjadi lebih stabil. - Derivatif memberikan aksi koreksi sebelum error menjadi terlalu besar[1][2][3][7].

1.11 Cara Kerja Secara Umum

- Kontroler PID secara terus-menerus membaca nilai proses aktual dari sensor.
- Menghitung error sebagai selisih antara setpoint dan nilai aktual.
- Setiap komponen (P, I, D) memberikan kontribusi pada sinyal kontrol berdasarkan error dan karakteristiknya masing-masing.
- Sinyal kontrol ini kemudian digunakan untuk mengatur aktuator (misal, membuka katup, mengatur kecepatan motor, dll) agar variabel proses kembali ke setpoint[1][2][3][6].

Rumus PID:

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

- $u(t)$: Output kontroler
- $e(t)$: Error (setpoint - nilai aktual)
- K_p : Konstanta proporsional
- K_i : Konstanta integral
- K_d : Konstanta derivatif[3]

1.12 Ilustrasi Sederhana

Misal pada sistem pemanas ruangan: - Setpoint suhu: 25°C - Sensor membaca suhu aktual: 23°C (error = 2°C) - Komponen P akan langsung menaikkan daya pemanas sesuai besarnya error. - Komponen I akan terus menambah daya jika error bertahan lama, hingga suhu benar-benar mencapai 25°C. - Komponen D akan mengurangi daya pemanas jika suhu naik terlalu cepat, untuk mencegah suhu melewati 25°C (overshoot)[1][2][3].

1.13 Penyesuaian (Tuning) PID

Agar kontrol PID bekerja optimal, nilai parameter

$$K_p$$

,

$$K_i$$

, dan

$$K_d$$

harus disesuaikan (tuning) sesuai karakteristik sistem yang dikendalikan (plant). Penyetelan ini penting agar respon sistem cepat, akurat, dan stabil[3][5].

1.14 Aplikasi

PID banyak digunakan di berbagai bidang, seperti: - Pengendalian suhu - Otomatisasi industri - Kontrol kecepatan motor - Proses kimia dan ilmiah[1][2][6]

1.15 Kesimpulan

PID control bekerja dengan mengoreksi error antara nilai aktual dan setpoint secara terus-menerus melalui tiga aksi utama: proporsional untuk respon cepat, integral untuk menghilangkan error sisa, dan derivatif untuk menstabilkan sistem dari perubahan cepat. Kombinasi ketiganya memungkinkan pengendalian yang presisi dan stabil pada berbagai sistem otomasi industri[1][2][3][7].

Draft by ardyseto

2 Pengenalan Sistem Kendali

2.1 Dasar Sistem Kendali

Sistem kendali adalah cabang ilmu teknik yang mempelajari bagaimana mengatur perilaku suatu sistem agar dapat beroperasi sesuai dengan tujuan yang diinginkan. Dalam kehidupan sehari-hari, sistem kendali banyak ditemukan pada perangkat elektronik, mesin industri, kendaraan, hingga sistem otomasi rumah tangga.

Pada dasarnya, sistem kendali bertujuan untuk menjaga variabel keluaran (output) suatu proses tetap pada nilai yang diharapkan meskipun terjadi gangguan atau perubahan pada lingkungan sekitarnya. Untuk mencapai tujuan ini, sistem kendali memanfaatkan prinsip umpan balik (feedback), di mana hasil keluaran diukur dan dibandingkan dengan nilai referensi, kemudian digunakan untuk memperbaiki aksi sistem.

Komponen utama dalam sistem kendali meliputi:

- Sensor: Mengukur variabel proses, seperti suhu, tekanan, kecepatan, atau posisi.
- Kontroller: Mengolah data dari sensor dan menentukan aksi yang diperlukan.
- Aktuator: Melaksanakan aksi berdasarkan perintah dari kontroller, misalnya menggerakkan motor atau membuka katup.
- Plant: Objek atau proses yang dikendalikan, seperti mesin, kendaraan, atau sistem pemanas.

Sistem kendali dapat diklasifikasikan menjadi dua jenis utama:

- Sistem Kendali Terbuka (Open-Loop): Tidak menggunakan umpan balik. Kontroller memberikan aksi berdasarkan input tanpa memperhatikan hasil keluaran. Contoh: mesin cuci yang beroperasi dengan waktu tetap tanpa memperhatikan kebersihan cucian.
- Sistem Kendali Tertutup (Closed-Loop): Menggunakan umpan balik. Kontroller memperbaiki aksi berdasarkan perbandingan antara keluaran aktual dan referensi. Contoh: AC otomatis yang mengatur suhu ruangan berdasarkan sensor suhu.

Salah satu metode pengendalian tertutup yang paling banyak digunakan adalah kendali PID (Proportional-Integral-Derivative). Kendali PID bekerja dengan mengombinasikan tiga aksi koreksi, yaitu proporsional (P), integral (I), dan derivatif (D), untuk menghasilkan respons sistem yang stabil dan sesuai harapan. PID digunakan luas dalam pengaturan suhu, kecepatan motor, posisi lengan robot, hingga sistem otomasi industri.

Draft by ardyseto

3 Persiapan Simulasi Berbasis Python

Sebelum memulai simulasi dan implementasi sistem kendali, sangat penting untuk menyiapkan lingkungan pemrograman Python yang stabil, terstruktur, dan mudah digunakan. Pada bab ini, Anda akan belajar cara menginstal Python menggunakan Anaconda (conda), mengelola virtual environment, menginstal library utama, serta penggunaan dasar library seperti NumPy, Matplotlib, Jupyter Notebook, dan python-control. Penjelasan disertai contoh kode yang sering digunakan dalam simulasi dan analisis sistem kendali.

3.1 Instalasi dan Pengelolaan Lingkungan Python

3.1.1 Instalasi Anaconda

1. Unduh Anaconda dari <https://www.anaconda.com/products/distribution> sesuai sistem operasi Anda.
2. Jalankan installer dan ikuti instruksi pemasangan hingga selesai.
3. Buka terminal (Anaconda Prompt di Windows) dan ketik:

```
1 conda --version
```

Perintah di atas digunakan untuk memastikan conda sudah terinstal dengan benar. Jika muncul versi conda, maka instalasi berhasil.

3.1.2 Pengelolaan Virtual Environment

- Membuat environment baru bernama `kendali` dengan Python 3.10:

```
1 conda create -n kendali python=3.10
```

Perintah ini membuat environment terpisah dengan nama `kendali` dan Python versi 3.10.

- Mengaktifkan environment:

```
1 conda activate kendali
```

Setelah diaktifkan, semua instalasi library akan masuk ke environment ini.

- Menonaktifkan environment:

```
1 conda deactivate
```

Environment yang aktif akan dinonaktifkan dan kembali ke base environment.

- Melihat daftar environment:

```
1 conda env list
```

Menampilkan seluruh environment yang tersedia di sistem.

- Menghapus environment:

```
1 conda remove -n kendali --all
```

Menghapus environment kendali beserta seluruh isinya.