

BAB III

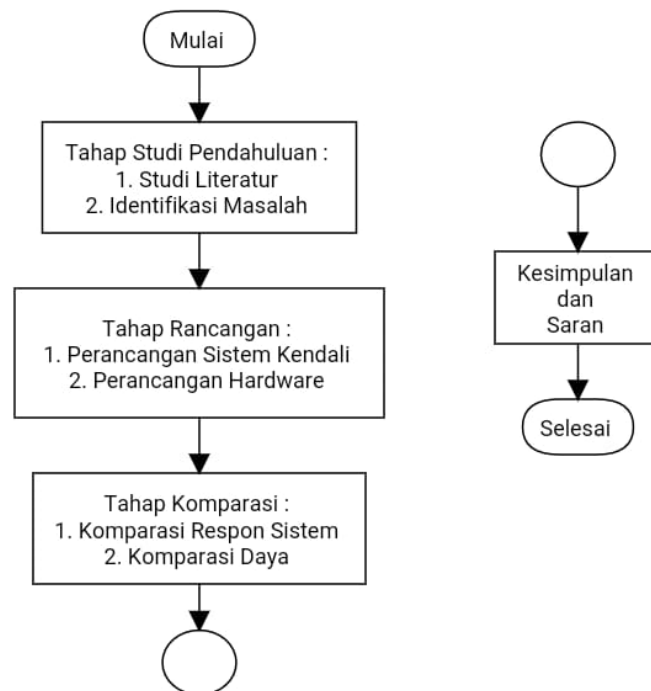
METODE PENELITIAN

3.1. Jenis Penelitian

Jenis penelitian ini adalah *library research*, yaitu penelitian yang objeknya dicari dengan berbagai informasi pustaka [19]. Penelitian ini menggunakan metode eksperimental dalam pembuatan kedua sistem untuk mencari respon sistem yang stabil kemudian melakukan analisa terhadap nilai keluaran pada kedua sistem menggunakan metode komparatif dengan pendekatan kuantitatif.

3.2. Tahapan Penelitian

Tahapan penelitian adalah serangkaian langkah-langkah yang akan dilakukan dalam penelitian dan direpresentasikan dalam bentuk *flowchart*. Adapun tahapan pada penelitian ini dapat dilihat pada gambar 3.1 sebagai berikut:



Gambar 3.1 Tahap penelitian

Keterangan :

- Tahap Pendahuluan adalah langkah awal yang dilakukan dalam sebuah penelitian. pada tahap ini dilakukan studi literatur dari berbagai sumber yang membahas penelitian terkait sehingga didapatkan identifikasi masalah pada penelitian yang akan diteliti.
- Tahap Rancangan merupakan tahap yang melakukan perancangan pada sistem berdasarkan identifikasi masalah. Pada penelitian ini, dirancang perangkat *solar tracker* dan sistem kendali PI dan PID.
- Tahap Komparasi merupakan proses untuk membandingkan data-data yang telah didapatkan. Pada penelitian ini melakukan perbandingan 2 perangkat *solar tracker* yang sudah diterapkan sistem kendali PI dan PID berdasarkan respon sistem dan daya yang dihasilkan panel surya.
- Tahap Kesimpulan dan Saran merupakan tahap terakhir yang dilakukan untuk menentukan sistem kendali manakah yang lebih baik antara kendali PI dan PID ketika diterapkan pada *solar tracker*.

3.3. Alat dan Bahan

1. Perangkat Keras (*hardware*)

Perangkat keras (*hardware*) yang digunakan pada penelitian ini dapat dilihat pada tabel 3.1 sebagai berikut :

Tabel 3.1 Perangkat Keras

No	Nama Perangkat Keras	Jumlah
1	Mikrokontroler Arduino Uno	2
2	Sensor LDR	4
3	Servo SG90	2

No	Nama Perangkat Keras	Jumlah
4	Panel Surya 10wp	2
5	<i>Solar Charger Controller</i>	2
5	Akumulator 12V7Ah	2

2. Perangkat Lunak (*software*)

Perangkat lunak (*software*) yang digunakan pada penelitian ini dapat dilihat pada tabel 3.2 sebagai berikut :

Tabel 3.2 Perangkat Lunak

No	Nama Perangkat Lunak	Keterangan
1	Arduino IDE	Arduino IDE (<i>Integrated Development Environment</i>) digunakan untuk memprogram mikrokontroler Arduino UNO
2	Matlab	Aplikasi Matlab digunakan untuk melihat respon sistem pada kendali PI dan PID

3.4. Rencana Perancangan Sistem Kendali

Sistem kendali yang dirancang yaitu sistem kendali PI dan PID pada *solar tracker* yang diharapkan mampu menjaga kestabilan motor servo dalam menggerakkan panel surya berdasarkan intensitas cahaya matahari. Kendali PI dan PID akan memberikan nilai output kepada motor servo berdasarkan error yang telah dikalkulasi ditambah dengan base PWM yang telah ditentukan.

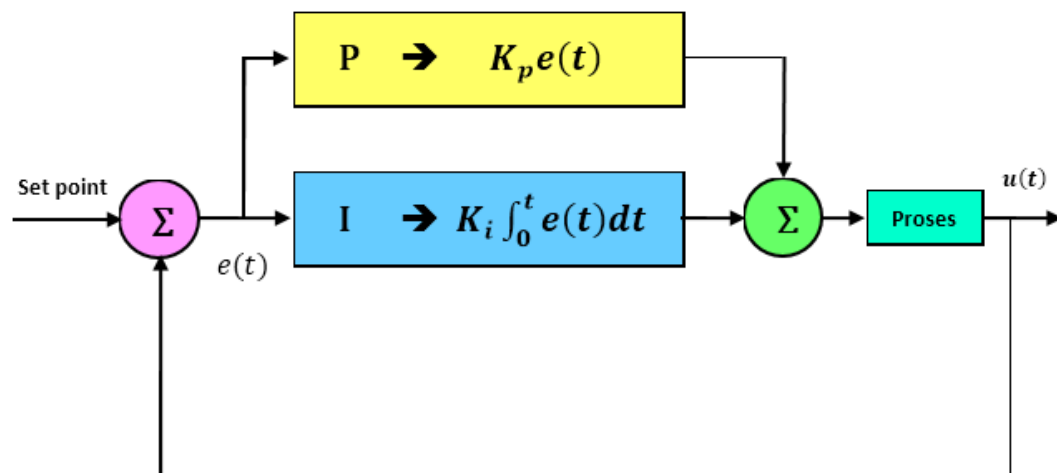
3.4.1. Kendali PI

Sistem Kendali PI memiliki persamaan matematis sebagai berikut.

$$PI = K_p e(t) + K_i \left[\int_0^t e(t) dt \right]$$

Nilai-nilai K_p dan K_i harus ditentukan melalui proses tuning untuk mencapai kinerja yang diinginkan sesuai dengan sistem yang dikendalikan. Proses tuning ini melibatkan pengaturan nilai-nilai K_p dan K_i hingga mencapai respon yang cepat dan stabil serta mengurangi *steady-state error*.

Adapun blok diagram kendali PI dapat dilihat pada gambar 3.2 sebagai berikut :



Gambar 3.2 Blok Diagram Kendali PI

Keerangan :

- *Setpoint* : *Setpoint* merupakan nilai referensi yang diinginkan untuk sistem, dan variabel proses yang merupakan nilai aktual dari sistem yang dikendalikan.
- *Error* : Komponen *error* adalah selisih antara setpoint dan nilai aktual dari sistem. *Error* ini menjadi dasar bagi tindakan kontrol dalam sistem kendali PID.

- Proporsional (P) : Komponen proporsional mengambil *error* sebagai masukan dan menghasilkan *output* yang berbanding lurus dengan besarnya *error*. Ini berarti semakin besar *error*, semakin besar pengaruh komponen proporsional terhadap *output* kendali.
- Integral (I) : Komponen integral mengintegrasikan *error* seiring waktu dan menghasilkan *output* berdasarkan akumulasi *error*. Ini membantu mengatasi *error* yang bersifat kumulatif dan mencegah *drift* jangka panjang.
- *Output* : *Output* dari komponen proporsional, integral, dan derivatif dijumlahkan untuk menghasilkan *output* kendali berupa nilai PWM (*Pulse Width Modulation*).
- Proses : *Output* kendali diarahkan ke sistem yang dikendalikan, dalam hal ini adalah motor servo. Sistem ini akan merespons tindakan kontrol dan menghasilkan variabel proses yang akan kembali menjadi input pada sistem kendali.

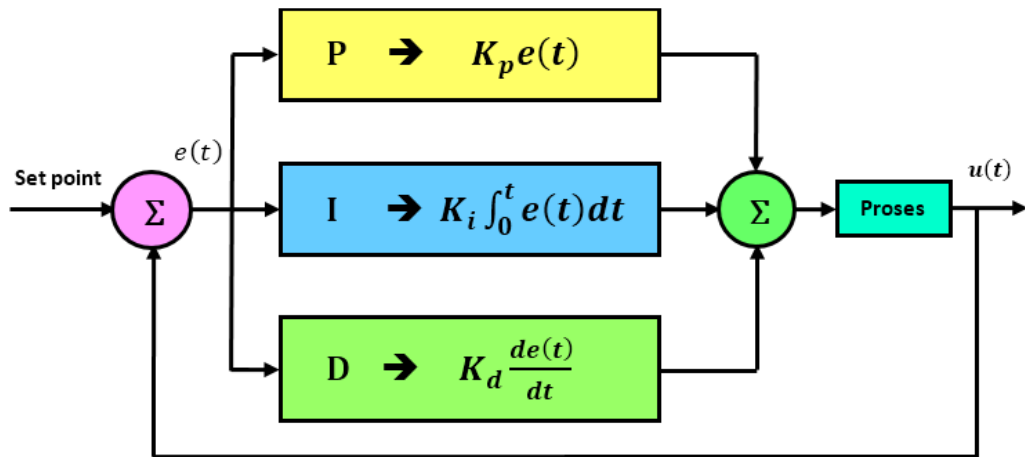
3.4.2. Kendali PID

Sistem Kendali PI memiliki persamaan matematis sebagai berikut.

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

Nilai-nilai K_p , K_i dan K_d harus ditentukan melalui proses tuning untuk mencapai kinerja yang diinginkan sesuai dengan sistem yang dikendalikan. Proses tuning ini melibatkan pengaturan nilai-nilai K_p , K_i dan K_d hingga mencapai respon yang cepat dan stabil serta mengurangi *steady-state error*.

Adapun blok diagram kendali PI dapat dilihat pada gambar 3.3 sebagai berikut :



Gambar 3.3 Blok Diagram PID

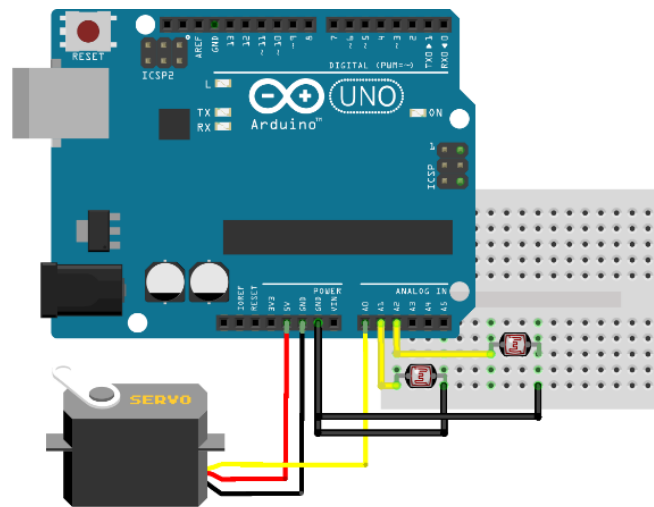
Keerangan :

- *Setpoint* : *Setpoint* merupakan nilai referensi yang diinginkan untuk sistem, dan variabel proses yang merupakan nilai aktual dari sistem yang dikendalikan.
- *Error* : Komponen *error* adalah selisih antara setpoint dan nilai aktual dari sistem. *Error* ini menjadi dasar bagi tindakan kontrol dalam sistem kendali PID.
- Proporsional (P) : Komponen proporsional mengambil *error* sebagai masukan dan menghasilkan *output* yang berbanding lurus dengan besarnya *error*. Ini berarti semakin besar *error*, semakin besar pengaruh komponen proporsional terhadap *output* kendali.
- Integral (I) : Komponen integral mengintegrasikan *error* seiring waktu dan menghasilkan *output* berdasarkan akumulasi *error*. Ini membantu mengatasi *error* yang bersifat kumulatif dan mencegah *drift* jangka panjang.

- Derivatif (D) : Komponen derivatif menghitung laju perubahan *error* dan menghasilkan *output* berdasarkan perubahan cepat *error*. Ini membantu dalam merespons perubahan cepat dalam sistem dan mengurangi dampak *overshoot*.
- *Output* : *Output* dari komponen proporsional, integral, dan derivatif dijumlahkan untuk menghasilkan *output* kendali berupa nilai PWM (*Pulse Width Modulation*).
- Proses : *Output* kendali diarahkan ke sistem yang dikendalikan, dalam hal ini adalah motor servo. Sistem ini akan merespons tindakan kontrol dan menghasilkan variabel proses yang akan kembali menjadi input pada sistem kendali.

3.5. Rencana Perancangan Perangkat Keras

Perancangan perangkat keras merupakan hasil penggabungan komponen-komponen sistem yang saling terhubung. Sistem ini berinteraksi dengan mikrokontroler Arduino Uno yang bertindak sebagai penerima masukan, pelaksana proses, dan pengendali aktuator. Sudut *single-axis solar tracker* yang digunakan dalam konteks ini mengadopsi pengaturan sudut horizontal. Adapun rangkaian sistem *single-axis solar tracker* dapat dilihat pada Gambar 3.4 sebagai berikut.



Gambar 3.4 Rangkaian Sistem

Rangkaian sistem pada gambar 3.4, perangkat sensor dan aktuator saling terhubung melalui arduino, dimana terdapat dua sensor LDR yang terhubung pada pin *signal* analog 1 dan analog 2, dan aktuator berupa motor servo yang terhubung pada pin sinyal analog 0 dengan pin tegangan input pada 5V.

3.6. Rencana Komparasi

Data pada penelitian ini akan diuji dengan metode komparatif, dimana setiap data yang didapatkan akan dibandingkan untuk dianalisis manakah dari ke dua metode kendali yang lebih efektif ketika diterapkan pada *single axis solar tacker*. Uji komparasi dibagi menjadi 2, yaitu komparasi respon sistem dan komparasi daya.

3.6.1. Komparasi Respon Sistem.

Respon sistem adalah hasil atau reaksi yang diberikan oleh sistem terhadap perubahan input atau stimulus yang diterimanya. Komparasi ini dilakukan untuk melakukan uji terhadap respon sistem kendali PI dan PID berdasarkan grafik dengan parameter uji sebagai berikut.

1. *Overshoot* adalah pergeseran sudut *tracker* yang terlalu besar ketika mencapai posisi *set point*.
2. *Rise Time* adalah waktu yang dibutuhkan untuk mencapai nilai *set point* yang diinginkan dari posisi awal, semakin kecil *rise time* maka semakin cepat sistem mencapai posisi *set point*.
3. *Settling Time* adalah waktu yang dibutuhkan untuk mencapai posisi stabil setelah mencapai nilai *set point*.
4. *Settling Min* adalah waktu yang dibutuhkan oleh sistem kendali untuk mencapai keadaan setimbang atau mencapai ketetapan, setelah mengalami respon yang tidak stabil atau terjadi *overshoot*.
5. *Settling Max* adalah waktu maksimum yang diperbolehkan bagi sistem kendali untuk mencapai keadaan setimbang atau mencapai ketetapan setelah terjadi perubahan input atau gangguan.
6. *Overshoot* adalah kondisi di mana output atau respon sistem melebihi nilai setpoint atau nilai target yang diinginkan saat mencapai keadaan setimbang.
7. *Undershoot* adalah kondisi di mana output atau respon sistem kendali berada di bawah nilai setpoint atau nilai target yang diinginkan saat mencapai keadaan setimbang.
8. *Peak* adalah nilai maksimum atau puncak yang dicapai oleh variabel yang dikendalikan dalam sistem kendali.

9. *PeakTime* adalah waktu yang diperlukan oleh sistem kendali untuk mencapai nilai puncak atau nilai maksimum dari respons variabel setelah terjadi perubahan input atau stimulus
10. *Respon Time* adalah waktu yang dibutuhkan sistem untuk merespon perubahan pada *input* (dalam hal ini, perubahan pada intensitas cahaya). [7].

Parameter-parameter tersebut ditentukan menggunakan *software* matlab berdasarkan nilai *output* sistem kendali yang telah didapatkan.

3.6.2. Komparasi Daya

Komparasi ini dilakukan untuk melakukan uji terhadap daya *input* dan output yang dihasilkan *solar cell* secara *open circuit* dan *close circuit* dengan parameter uji sebagai berikut.

1. *Voltage (Volt)* atau tegangan adalah perbedaan potensial listrik antara dua titik dalam sebuah rangkaian listrik. Tegangan diukur dalam satuan *volt* (V) dan menunjukkan seberapa besar gaya dorong listrik yang mengalir di antara dua titik tersebut.
2. *Current (Ampere)* adalah satuan ukuran untuk mengukur besarnya arus listrik dalam sebuah rangkaian. Dalam sebuah rangkaian listrik, arus listrik diukur dalam satuan ampere (A) dan menunjukkan jumlah muatan listrik yang mengalir melalui rangkaian tersebut dalam suatu waktu tertentu.
3. *Power (Watt)* adalah satuan ukuran untuk daya listrik atau laju penggunaan energi dalam sebuah rangkaian listrik (Balisranislam *et al.*, 2021).