

## PEMODELAN SOLAR TRACKER DENGAN PENGENDALI PI, PD, DAN PID MENGUNAKAN MATLAB-SIMULINK DENGAN BEBAN BATERAI

Putri Maharani<sup>1</sup>, Ahmad Antares A<sup>2</sup>, Baso Mukhlis<sup>3</sup>, Yusnaini Arifin<sup>4</sup>, Nurhani Amin<sup>5</sup>, Martdiansyah<sup>6</sup>

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Tadulako<sup>1,2,3,4,5,6</sup>  
email : putrimrani620@gmail.com<sup>1</sup>

### ABSTRACT

*A solar tracker is a photovoltaic (PV) system support tool aimed at optimizing solar energy absorption by directing solar cells to follow the movement of the sun. In this research, a solar tracker for PV with battery load model was created using MATLAB-Simulink with each controller, namely Proportional Integral (PI), Proportional Derivative (PD) and Proportional Integral Derivative (PID) and the simulation results were compared. The simulation results show that the PID controller is the controller with the most optimal results because all set point values entered in the output remain the same and the overshoot value is also low, the PD controller is the controller that produces high errors and also overshoot reaching 120 V, and the PI controller is the controller that the results are closest to a PID controller where the resulting overshoot value is only around 20%.*

Keywords: solar tracker, photovoltaic, controller, battery, MATLAB-Simulink.

### INTISARI

*Solar Tracker merupakan alat pendukung sistem fotovoltaik (PV) yang bertujuan untuk mengoptimalkan penyerapan energi matahari dengan mengarahkan sel surya mengikuti pergerakan matahari. Pada penelitian ini dibuat solar tracking untuk PV dengan model beban baterai menggunakan MATLAB-Simulink dengan masing-masing pengontrol yaitu Proportional Integral (PI), Proportional Derivative (PD) dan Proportional Integral Derivative (PID) dan hasil simulasinya dibandingkan. Hasil simulasi menunjukkan bahwa pengontrol PID merupakan pengontrol dengan hasil paling optimal karena semua nilai set point yang dimasukkan pada output tetap sama dan nilai overshootnya juga rendah, pengontrol PD merupakan pengontrol yang menghasilkan error yang tinggi dan juga mencapai overshoot 120 V, dan pengontrol PI merupakan pengontrol yang hasilnya paling mendekati dengan pengontrol PID dimana nilai overshoot yang dihasilkan hanya berkisar 20%.*

Kata kunci: Solar Tracker, photovoltaic, Pengendali, Baterai, MATLAB-Simulink

## I. PENDAHULUAN

Sistem *photovoltaic (PV)* telah berperan sebagai pengganti energi konvensional dan dalam menjalankan fungsinya sistem ini dibantu beberapa perangkat lain seperti sel surya dan sistem penyimpanan energi listrik. Pemasangan *PV* yang statis atau diam menyebabkan penyerapan sinar matahari menjadi tidak optimal. Untuk mengatasi hal tersebut maka dibuatlah suatu alat yang dinamakan *solar tracker*. *Solar tracker* merupakan sebuah alat penunjang sistem *photovoltaic (PV)* yang ditujukan untuk optimasi penyerapan energi matahari dengan mengarahkan *solar cell* mengikuti pergerakan matahari. *Solar tracker* dibuat untuk menggerakkan *solar cell*

secara otomatis pada sudut  $0^\circ - 180^\circ$  dan sebaliknya (Sihite, 2021).

Pada penelitian ini dibuat pemodelan solar tracker dengan masing-masing pengendali, yaitu *Proportional Integral (PI)*, *Proportional Derivative (PD)* dan *Proportional Integral Derivative (PID)* dan hasil simulasinya dibandingkan. *Proportional Integral (PI)* adalah sistem gabungan antara kontrol *proportional* dan *integral*, dimana pengendali ini digunakan agar reaksi dari sebuah sistem lebih cepat dan menghilangkan *offset*. Pengendali *Proportional Derivative (PD)* adalah sistem gabungan antara kontrol *proportional* dan *derivative*, dimana pengendali dapat memperkecil eror atau

meredam *overshot*. Sedangkan pengendali *Proportional Integral Derivative (PID)* adalah sistem gabungan antara kontrol *proportional*, *integral*, dan *derivative* dimana pengendali ini dapat menghasilkan *output* dengan *rise time* yang tinggi dan *error* yang kecil. Untuk memenuhi sistem yang diinginkan maka parameter P, I, dan D harus ditetapkan secara optimal.

Pada penelitian sistem PV dengan *solar tracker* ini digunakan baterai sebagai bebannya. Sistem fisik baterai sebagai beban utama dibuat ke dalam bentuk model matematis untuk kemudian disimulasikan, sehingga dapat diketahui respon tiap bagian pengendali dan diketahui pula perbaikan respon yang perlu dilakukan. Untuk membuat simulasi dari model matematis tersebut digunakan perangkat MATLAB-Simulink. MATLAB adalah bahasa tingkat tinggi dan lingkungan yang interaktif untuk perhitungan numerik, *visualisasi*, dan pemrograman. Dengan menggunakan MATLAB, kita dapat menganalisis data, mengembangkan algoritma, dan membuat model dan aplikasi.

Dari uraian di atas, penelitian ini mengangkat judul “Pemodelan Solar Tracker dengan Pengendali PI, PD, dan PID menggunakan MATLAB-Simulink dengan Beban Baterai”. Tujuan penelitian ini adalah merancang dan membuat desain atau pemodelan solar tracker dengan pengendali PI, PD, dan PID dengan beban baterai menggunakan MATLAB-Simulink dan menentukan hasil simulasi paling optimal dari pengendali-pengendali tersebut dengan cara mengubah-ubah nilai parameter P, I, dan D berdasarkan sudut slope.

## II. LANDASAN TEORI

### A. Solar Tracker

*Solar tracker* adalah perangkat yang berfungsi untuk mengatur panel surya agar mengikuti matahari saat bergerak melintasi angkasa. Ketika *solar tracker* digabungkan dengan panel surya maka panel surya akan mengikuti jalur matahari dan menghasilkan lebih

banyak energi. *Solar tracker* biasanya dipasangkan dengan sistem tenaga surya yang diinstal di tanah atau di atap rumah (Hakim dan Sukma, 2022)..

### B. Cara kerja solar tracker

Cara kerja *solar tracker* adalah mendeteksi perubahan pergerakan cahaya matahari dan mengubah sudut panel surya untuk diarahkan ke titik matahari berada. Perangkat ini mengubah sudutnya sepanjang hari untuk mengikuti jalur matahari sehingga memaksimalkan penangkapan energi. *Solar tracker* membantu meminimalkan sudut yang dibuat oleh sinar matahari dengan garis tegak lurus ke permukaan antara cahaya yang masuk dan panel surya .

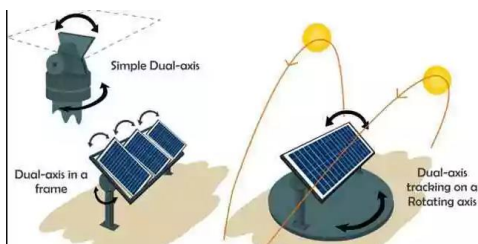
### C. Jenis Solar Tracker

Berdasarkan cara pelacakan matahari bergerak, ada tiga jenis sistem solar tracker :

1. Solar tracker manual. Pelacak manual membutuhkan seseorang yang secara fisik bertugas menyesuaikan panel surya pada waktu yang berbeda sepanjang hari untuk mengikuti matahari. Cara semacam ini tidak praktis karena membutuhkan operator untuk terus memantau matahari dan mengubah posisi sistem panel surya.
2. Solar tracker pasif. Pelacak pasif berisi cairan dengan titik didih rendah yang akan menguap saat terkena radiasi matahari. Ketika cairan menguap, sistem kemiringan menjadi tidak seimbang. Ketidakseimbangan ini menyebabkan panel surya miring ke arah sinar matahari.
3. Solar tracker aktif. Pelacak aktif menggunakan dinamo atau silinder hidrolik untuk mengubah posisi. Motor pada pelacak aktif akan menggerakkan

panel surya sehingga menghadap matahari. Ada beberapa bentuk solar tracker aktif, diantaranya *solar tracker single-axis* atau pelacak sumbu satu arah dan *solar tracker dual-axis* atau pelacak sumbu dua arah.

Gambar 1 adalah pelacak sumbu satu arah yang mengikuti posisi matahari saat bergerak dari timur ke barat. Pelacak ini biasanya digunakan dalam proyek PLTS skala besar. Pelacak sumbu tunggal dapat meningkatkan produksi antara 25% hingga 35%.



Gambar 2. Solar tracker dual-axis (www.solarreviews.com)

#### D. Pengendali

Sistem pengendalian dirancang untuk melakukan dan menyelesaikan tugas tertentu. Syarat utama sistem pengendalian adalah harus stabil. Di samping kestabilan mutlak, sistem harus memiliki kestabilan secara relatif, yakni tolak ukur kualitas kestabilan sistem dengan menganalisis sampai sejauh mana batas-batas kestabilan sistem tersebut jika dikenai gangguan (Ogata, 1997). Selain itu analisis juga dilakukan untuk mengetahui bagaimana kecepatan sistem dalam merespon masukan, dan bagaimana peredaman terhadap adanya lonjakan (*overshoot*).

Suatu sistem dikatakan stabil jika diberi gangguan maka sistem tersebut akan kembali ke keadaan *steady state* di mana *output* berada dalam keadaan tetap seperti tidak ada gangguan.

Sistem dikatakan tidak stabil jika *outputnya* berosilasi terus menerus ketika dikenai suatu gangguan. Karena suatu sistem pengendalian biasanya melibatkan penyimpanan energi maka *output* sistem ketika diberi suatu masukan, tidak dapat mengikuti masukan secara serentak, tapi menunjukkan respon transien berupa suatu osilasi teredam sebelum mencapai *steady state*.

#### 1. Pengendali Proporsional

Pengendali proporsional memiliki keluaran yang sebanding dengan sinyal kesalahan. Untuk kontroler dengan aksi kontrol proporsional, hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah:

$$m(t) = K_p \cdot e(t)$$

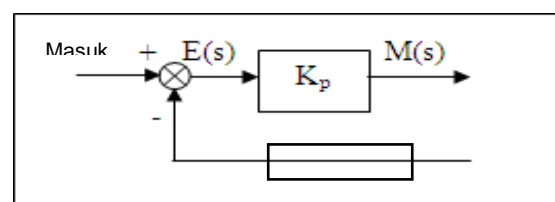
atau, dalam besaran transformasi Laplace,

$$\frac{M(s)}{E(s)} = K_p$$

Dimana:

- $m(t)$  = penggerak yang berasal dari pengendali pada waktu
- $e(t)$  = Selisih antara set point dan variabel proses yang diukur pada waktu
- $K_p$  = Konstanta (penguat) pengendali proposional

Diagram blok kontroler proporsional ditunjukkan dalam Gambar 3.



Gambar 3. Diagram blok kontroler *proporsional* (Hakim, 2018)

## 2. Pengendali Integral

Pengendali integral berfungsi mengurangi kesalahan keadaan mantap pada kontroler proporsional sebelumnya. Pada kontroler dengan aksi integral, harga keluaran kontroler  $m(t)$  diubah dengan laju yang sebanding dengan sinyal kesalahan penggerak  $e(t)$ .

$$\frac{dm(t)}{dt} = K_i \cdot e(t)$$

Dimana:

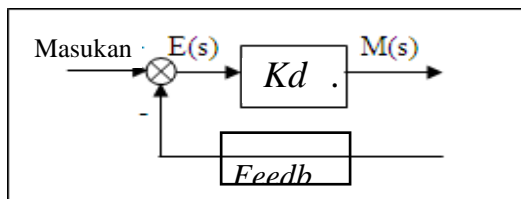
$dm(t)$  = penggerak yang berasal dari pengendali pada waktu

$de(t)$  = Selisih antara setpoint dan variabel proses yang diukur pada waktu

$K_i$  = Konstanta (penguat) pengendali integral

$e(t)$  = Kesalahan pengontrol

Gambar 4 menunjukkan diagram blok kontroler integral.



Gambar 4. Diagram blok kontroler *integral* (Hakim, 2018)

## Pengendali Diferensial

Pengendali ini digunakan untuk memperbaiki atau mempercepat respons transien sebuah sistem kontrol dengan cara memperbesar phase lead terhadap penguatan kontrol dan mengurangi phase lag penguatan tersebut (Ogata, 1997). Kontroler diferensial tidak dapat mengeluarkan output bila tidak ada perubahan masukan dan tidak dapat digunakan untuk proses yang mengandung noise. Hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan penggerak  $e(t)$  adalah

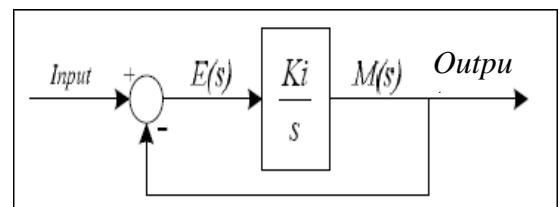
$$\frac{M(s)}{E(s)} = K_d \cdot s$$

Dimana:

$m(t)$  = penggerak yang berasal dari pengendali pada waktu

$e(t)$  = Selisih antara setpoint dan variabel proses yang diukur pada waktu

$K_d$  = Konstanta (penguat) pengendali di



Gambar 5. Diagram blok kontroler *diferensial* (Hakim, 2018)

### 5. Pengendali Proporsional Integral Diferensial (PID)

Gabungan aksi kontrol proporsional, integral, dan diferensial mempunyai keunggulan dapat saling menutupi kekurangan dan menampilkan kelebihan dari masing-masing kontroler. Elemen-elemen P, I, dan D secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan offset dan menghasilkan perubahan awal yang besar. Persamaan kontroler PID ini dapat dinyatakan sebagai berikut (Ogata, 1997):

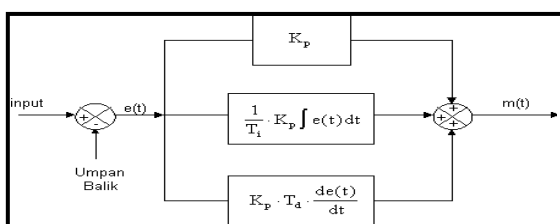
$$m(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \int e(t) dt + K_p \cdot T_d \frac{de(t)}{dt}$$

Dalam transformasi Laplace dinyatakan sebagai berikut:

$$\frac{M(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right)$$

Dimana:

- $m(t)$  = penggerak yang berasal dari pengendali pada waktu
- $e(t)$  = Selisih antara setpoint dan variabel proses yang diukur pada waktu
- $K_p$  = Konstanta (penguat) pengendali Proporsional



Gambar 6. Diagram blok kontroler PID (Hakim, 2018)

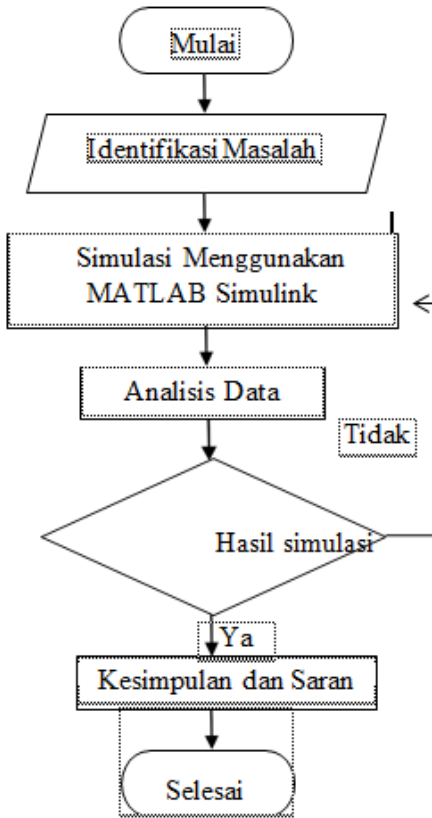
### III. METODE PENELITIAN

#### A. Cara Penelitian

Metode penelitian yang digunakan meliputi langkah-langkah sebagai berikut:

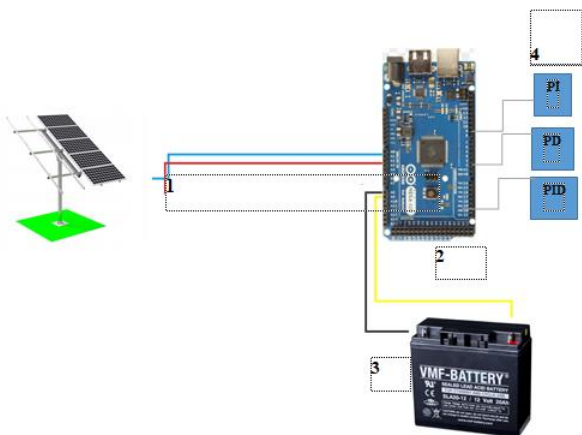
1. Menentukan pemodelan yang akan diteliti dan komponen-komponen yang menyusun pemodelan tersebut
2. Setelah diketahui komponen-komponen penyusun pemodelan beserta parameternya, maka sistem dapat dimodelkan menjadi sebuah persamaan matematis yang sesuai dalam hal ini termasuk dengan pengendali PI, PD, dan PID.
3. Kemudian setelah pemodelan selesai, dilakukan simulasi untuk mendapatkan respon dari tiap pengendali yang digunakan.
4. Selanjutnya respon dari setiap pengendali itu dianalisa dan kemudian disimpulkan yang mana yang mendapatkan hasil yang paling baik.

Adapun flowchart dari penelitian ini adalah sebagai berikut:



Gambar 7. Flowchart penelitian

Skema fisik dari rangkaian yang hendak disimulasikan ditunjukkan pada Gambar 8.



Gambar 8. Skema fisik yang hendak di simulasikan

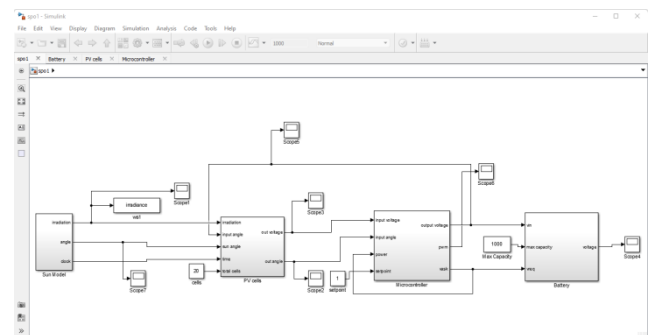
Keterangan Gambar :

- 1.PV & Solar Tracker
- 2.Mikrokontroler
- 3.Baterai
- 4.Pengendali ( PI, PD, PID)

#### IV. HASIL PENELITIAN DAN PEMBAHASAN

##### A. Simulasi Menggunakan Software MATLAB-SIMULINK

Tampilan pemodelan keseluruhan pada aplikasi MATLAB menunjukkan keseluruhan dari pemodelan yang di dalamnya terdapat sun model, PV cell, micro kontroler dan juga baterai.

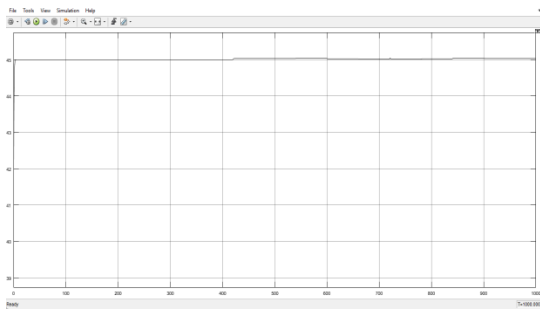


Gambar 9. Tampilan pemodelan keseluruhan

##### B. Hasil Simulasi Dengan Menggunakan Pengendali PI

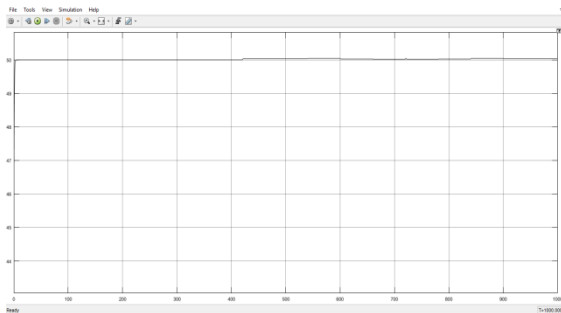
Hasil simulasi menggunakan pengendali PI grafiknya diperlihatkan seperti Gambar 10, 11, dan 12. Pada pengendali PI ini nilai yang dimasukkan pada pengendali *proportional* (P) 1 dan nilai *integral* (I) 0,5. Simulasikan dilakukan mulai dari jam 8.00 pagi dengan sudut 45°, 50°, dan 60°.

- 1) Sudut 45° dengan pengendali PI



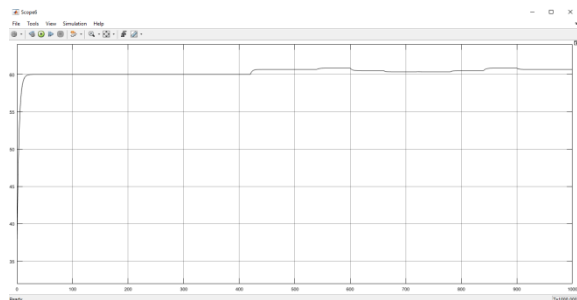
Gambar 10. Tegangan keluaran mikrokontroler terhadap waktu pengendali

2) Sudut 50° dengan pengendali PI



Gambar 11. Tegangan keluaran mikrokontroler terhadap waktu pengendali PI sudut 50°

3) Sudut 60° dengan pengendali PI



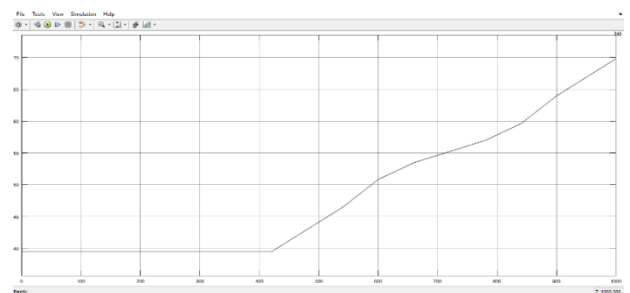
Gambar 12. Tegangan keluaran dari mikrokontroler terhadap waktu pengendali PI sudut 60°

Gambar 10 terlihat kenaikan awal keluaran tegangan berada di nilai 44 *time seat* 0 dengan nilai puncak 45 pada *time seat* 2 sesuai dengan *set poin* yang di masukan diawal pada rangkaian kemudian pada *time seat* 410 terjadi *overshoot* 10% sehingga hanya menimbulkan *offset* yang

sedikit namun dalam grafik ini terlihat bahwa nilai *overshoot* naik turun sampai akhir kemudian *overshoot* tertinggi ke dua berada pad *time seat* 710 dengan nilai sekitar 20%. Gambar 11 terlihat kenaikan awal keluaran tegangan berada di nilai 48 *time seat* 0 dengan nilai puncak 50 pada *time seat* 1 sesuai dengan *set poin* yang di masukan diawal pada rangkaian kemudian pada *time seat* 410 dan 810 terjadi *overshoot* 10% dan terjadi juga di *time seat* 710 dengan *overshoot* sebesar 20% dengan waktu yang singkat sehingga hanya menimbulkan *offset* yang sedikit namun dalam grafik ini terlihat bahwa nilai *overshoot* terjadi naik turun sampai akhir walaupun dengan *overshot* yang tidak terlalu tinggi .

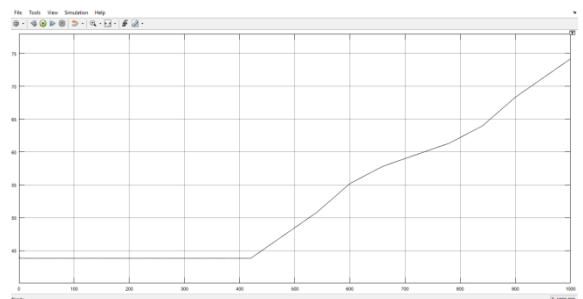
C. Hasil Simulasi Dengan Menggunakan Pengendali PD

1). Sudut 45° dengan pengendali PD



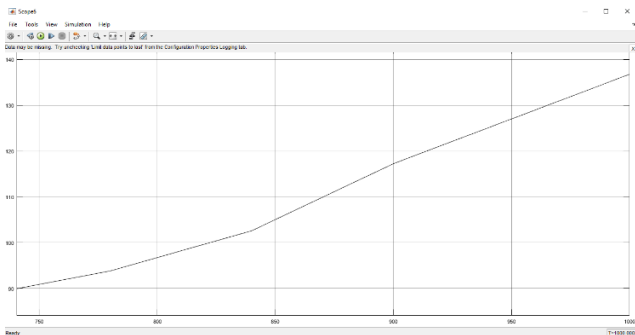
Gambar 13. Tegangan keluaran mikrokontroler terhadap waktu pengendali PD sudut 45°

2). Sudut 50° dengan pengendali PD



Gambar 14. Tegangan keluaran mikrokontroler terhadap waktu pengendali PD sudut 50°

3). Sudut 60° dengan pengendali PD



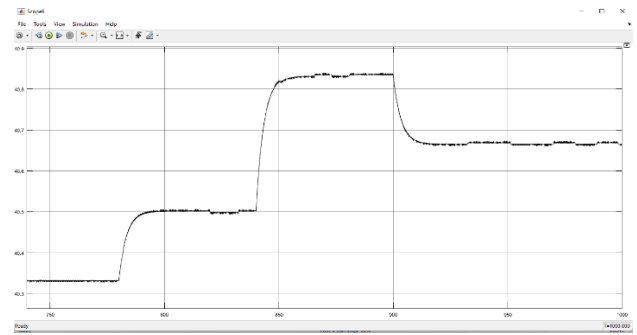
Gambar 15. Tegangan keluaran mikrokontroler terhadap waktu pengendali PD sudut  $60^\circ$

Gambar 13 keluaran tegangan terlihat naik pada time seat 410 dengan tinggi 39 yang kemudian terus naik sampai dengan time seat 1000 dengan nilai puncak 69. Pada grafik ini terlihat bahwa steady state error yang ada lumayan jauh dari set point yang telah ditetapkan namun tidak sebesar pada nilai uji coba di sudut  $40^\circ$ . Gambar 14 keluaran tegangan terlihat naik pada time seat 410 dengan tinggi 44 yang kemudian terus naik sampai dengan time seat 1000 dengan nilai puncak 74. Pada grafik ini terlihat bahwa steady state error syang ada lumayan tinggi dari set point yang telah ditetapkan. Gambar 15 keluaran tegangan terlihat naik pada time seat 748 dengan tinggi 90 yang kemudian terus naik sampai dengan time seat 1000 dengan tinggi 138. Pada pengendali ini terlihat bahwa steady state error sangat jauh dari set point yang telah ditetapkan.

#### D. Hasil Simulasi Dengan Menggunakan Pengendali PID

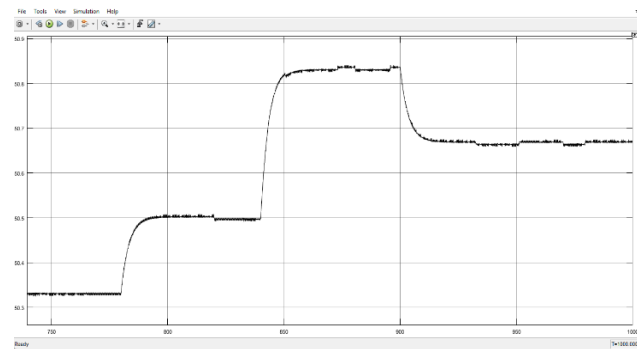
Hasil simulasi menggunakan pengendali PID ditunjukkan pada Gambar 16, 17, dan 18.

##### 1). Sudut $45^\circ$ dengan pengendali PID



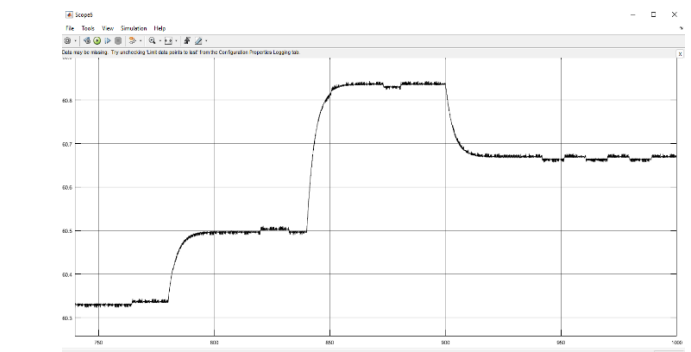
Gambar 16. Tegangan keluaran mikrokontroler terhadap waktu pengendali PID sudut  $45^\circ$

##### 2). Sudut $50^\circ$ dengan pengendali PID



Gambar 17. Tegangan keluaran mikrokontroler terhadap waktu pengendali PID sudut  $50^\circ$

##### 3). Sudut $60^\circ$ dengan pengendali PID



Gambar 18. Tegangan keluaran mikrokontroler terhadap waktu pengendali PID sudut  $60^\circ$



Gambar 16 keluaran tegangan terlihat naik pada *time seat* 780 dengan nilai puncak 45 sesuai dengan *set poin* yang di masukan diawal kemudian pada *time seat* 840 terjadi *overshot* dengan nilai yang kecil dimana nilai tingginya menjadi 60,8 sedikit lebih rendah dari pada pengendali PI sehingga hanya menimbulkan *offset* yang sedikit dan terlihat turun di angka 45,6. Gambar 17 keluaran tegangan terlihat naik pada *time seat* 780 dengan nilai puncak 50 sesuai dengan *set poin* yang di masukan diawal kemudian pada *time seat* 840 terjadi *overshot* dengan nilai yang kecil dimana nilai tingginya menjadi 50,8 sedikit lebih rendah dari pada pengendali PI sehingga hanya menimbulkan *offset* yang sedikit dan terlihat turun di angka 50,6. Gambar 18 keluaran tegangan terlihat naik pada *time seat* 700 dengan nilai puncak 60 sesuai dengan *set poin* yang di masukan diawal kemudian pada *time seat* 840 terjadi *overshot* dengan nilai yang kecil dimana nilai tingginya menjadi 60,8 sedikit lebih rendah dari pada pengendali PI sehingga hanya menimbulkan *offset* yang sedikit dan terlihat turun di angka 60,6.

#### E. Analisis Perbandingan Hasil Penelitian Yang Diperoleh

Pada penelitian ini digunakan 3 perbandingan pengendali yaitu, PI, PD, dan PID. Dari hasil penelitian didapatkan masing-masing 3 grafik dari setiap pengendali dengan menggunakan sudut  $45^\circ$ ,  $50^\circ$ , dan  $60^\circ$  diperoleh:

- a) pengendali PI dapat menghasilkan tegangan 70 volt dalam waktu 6-7 menit.
- b) pengendali PD menghasilkan tegangan sebesar 140 volt dalam waktu 6-7 menit.
- c) pengendali PID menghasilkan tegangan sebesar 40 volt dalam waktu 2 menit.

Sehingga dapat disimpulkan bahwa dari 3 perbandingan pengendali tersebut, pengendali PID adalah pengendali yang cocok digunakan pada penelitian ini dikarenakan tegangan yang dihasilkan konstan sesuai dengan sudut yang dimasukkan serta waktu yang dibutuhkan tergolong cepat dibandingkan 2 pengendali lainnya karena hanya membutuhkan waktu selama 2 menit.

## V. KESIMPULAN

### A. Kesimpulan

Dari ketiga pengendali yang digunakan

1. Pengendali PID adalah pengendali yang hasilnya paling optimal karena semua nilai *set point* yang dimasukkan keluarannya tetap sama dan nilai *overshoot*nya juga rendah.
2. Pengendali PD adalah pengendali yang menghasilkan error dan juga *overshot* yang tinggi mencapai angka 120 V.
3. Pengendali PI adalah pengendali yang hasilnya paling mendekati dengan pengendali PID dimana nilai *overshoot* yang dihasilkan hanya sekitar 20%.

## DAFTAR PUSTAKA

- [1] A. Hakim, *Perancangan dan Simulasi Penerapan Pengendali PID Pada Sistem Kendali Tegangan Keluaran Generator Pesawat N291 di PT. Dirgantara Indonesia*, Universitas Negeri Malang, 2018.
- [2] K. Ogata, *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga., 1997.
- [3] T.D. Hakim dan Muhammad Sukma, "Rancang Bangun Dual-Axis Solar Tracker Menggunakan Mikrokontroler Arduiono Mega 2560", *Jurnal Teknik Elektro Universitas Krisnadwipayana*, Vol 10 No. 2 ,2022.
- [4] J. Sihite, *Studi Pengaruh Sudut Kemiringan Panel Surya Terhadap Intensitas Cahaya Panel Surya* , Universitas HKBP Nommensen P.Siantar , 2021.