

## MENGGUNAKAN SENSOR IMU SEBAGAI PENSTABIL ROBOT EXOSKELETON ASSISTIVE WALKER DENGAN METODE PID

**Daniel Sutopo Pamungkas**

Jurusan Teknik Elektro, Program Studi Teknik Mekatronika  
Politeknik Negeri Batam  
Email: [daniel@polibatam.ac.id](mailto:daniel@polibatam.ac.id)

**Ryan Satria Wijaya**

Jurusan Teknik Elektro, Program Studi Teknik Robotika  
Politeknik Negeri Batam  
Email: [ryan@polibatam.ac.id](mailto:ryan@polibatam.ac.id)

**Ardana Syah Wicaksono**

Jurusan Teknik Elektro, Program Studi Teknik Robotika  
Politeknik Negeri Batam  
Email: [ardanasw007@gmail.com](mailto:ardanasw007@gmail.com)

### ABSTRAK

Teknologi dan robotika sudah tidak asing lagi bagi manusia khususnya di dunia industry zaman sekarang yang sudah semakin canggih. Semakin canggihnya teknologi dan robotika, maka semakin banyak pula ide ide yang bisa direalisasikan. Contohnya di jurnal ini kami membuat robot *Exoskeleton Lower Limb*. Kegunaan robot ini adalah untuk membantu penyandang disabilitas, pasien yang terkena cedera lower limb agar bisa bergerak secara normal. Tapi pada kasus ini pasien yang kami tujukan adalah untuk anak kecil usia 10 -15 tahun. Dimana nantinya robot ini bisa menopang berat tubuh si pengguna dan membantu si pengguna bisa berjalan seperti orang normal pada umumnya. Dalam permasalahan tersebut robot *Exoskeleton Assistive Walker* membutuhkan metode atau sistem yang dapat membuat robot tetap pada jalur dan mengontrol pergerakan motor. Maka metode yang digunakan disini adalah metode PID. Pengendali PID merupakan hasil dari gabungan tiga unsur yaitu *proportional*, *integral* dan *derivative* yang dapat digunakan secara bersamaan ataupun sendiri-sendiri tergantung dari respon yang diinginkan pada suatu system. Tujuan dari penelitian adalah untuk memahami dan menemukan grafik terbaik dalam mencari setpoint dari beberapa percobaan yang dilakukan terhadap sensor Imu. Grafik terbaik dari penelitian ini dihasilkan dari nilai  $K_p = 2.2$ ,  $K_i = 0.6$  dan  $K_d = 0.7$  dengan waktu dari gangguan menuju stabil yaitu 0.3 detik.

**Kata kunci:** *PID, Exoskeleton Assistive Walker, Kendali*

### ABSTRACT

*Technology and robotics are already familiar to humans, especially in today's industrial world which is increasingly sophisticated. The more sophisticated technology and robotics, the more ideas can be realized. For example, in this journal, we made a Lower Limb Exoskeleton robot. The use of this robot is to help people with disabilities, and patients with lower limb injuries, so they can move normally. But in this case, the patient we are referring to is a small child aged 10-15 years. Later, this robot can support the user's body weight and help the user walk like a normal person in general. In this problem, the Assistive Walker Exoskeleton Robot requires a method or system that can keep the robot on track and control the movement of the motor. Then the method used here is the PID method. PID controller is the result of a combination of three elements, namely proportional, integral and derivative which can be used simultaneously or individually depending on the desired*

*response in a system. This research aims to understand and find the best graph in finding the setpoint from several experiments conducted on the IMU sensor. The best graphs from this study resulted from  $K_p = 2.2$ ,  $K_i = 0.6$  and  $K_d = 0.7$  with a time from disturbance to stability is 0.3 seconds.*

**Keywords:** *PID, Exoskeleton Assistive Walker, Control*

## 1. PENDAHULUAN

Teknologi dan robotika sudah tidak asing lagi bagi manusia khususnya di dunia industry zaman sekarang yang sudah semakin canggih [1]. Semakin canggihnya teknologi dan robotika, maka semakin banyak pula ide ode yang bisa direalisasikan. Contohnya di jurnal ini kami membuat robot Exoskeleton Lower Limb. Kegunaan robot ini adalah untuk membantu penyandang disabilitas, pasien yang terkena cedera lower limb agar bisa bergerak secara normal [2]. Tapi pada kasus ini pasien yang kami tujuan adalah untuk anak kecil usia 10 -15 tahun. Dimana nantinya robot ini bisa menopang berat tubuh si pengguna dan membantu si pengguna bisa berjalan seperti orang normal pada umumnya [3].

Permasalahan utama yang dihadapi disini adalah bagaimana robot assistive walker ini dapat bergerak secara otomatis ke tempat tujuan meskipun ada distraksi dari si pengguna. Pada robot ini nantinya terdapat sensor Imu GY-25, Arduino Mega ATMEGA 2560, Vexta motor BLDC, driver vexta, dan battery lippo 6 cell. Sensor imu GY-25 ini mempunyai akurasi pengukuran mencapai  $1^\circ$  dan merupakan sensor yang mampu mendeteksi perubahan suatu objek [4].

Dalam permasalahan tersebut Robot *Exoskeleton Assitive Walker* membutuhkan metode atau sistem yang dapat membuat robot tetap pada jalur dan mengontrol pergerakan motor. Maka metode yang digunakan disini adalah metode PID. Pengendali PID merupakan hasil dari gabungan tiga unsur yaitu *proportional*, *integral* dan *derivative* yang dapat digunakan secara bersamaan ataupun sendiri-sendiri tergantung dari respon yang diinginkan pada suatu system [5], [6].

Metode PID *controller* banyak sekali diperlukan untuk keperluan system tertentu seperti mengontrol kecepatan motor DC, mengontrol suhu, mengontrol sudut dan lain lain [7]. Setiap kelebihan dan kekurangan dari setiap masing masing unsur P, I dan D dapat saling melengkapi dengan menggabungkan ketiganya secara parallel menjadi PID *Controller*. Unsur controller P, I dan D secara menyeluruh bertujuan untuk mempercepat reaksi suatu system, yaitu menghilangkan offset dan menghasilkan perubahan awal yang besar [8].

Pada bab selanjutnya akan dibahas mengenai apa saja yang ada pada robot ini, kemudian pada bab 3 dilakukan pengujian untuk membandingkan beberapa nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang nantinya pada kesimpulan akan disimpulkan hasil dari pebandingan nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang didapatkan untuk menstabilkan robot *Exoskeleton Assitive Walker*.

## 2. METODOLOGI PENELITIAN

### 2.1. Instrumen Penelitian

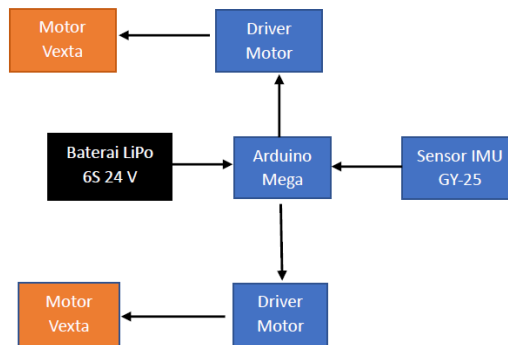
Pada penelitian perancangan sistem kendali ini digunakan sebuah robot yaitu Exoskeleton Assistive Walker dimana robot ini digunakan untuk rehabilitasi kepada pasien yang cedera di bagian Lower Limb. Robot ini dipergunakan hanya untuk rehabilitasi.

Pada gambar 1 bisa dilihat bahwa ini adalah robot Exoskeleton Assitive Walker bahan utama yang di pakai adalah logam holo dan dibantu oleh 4 penggerak yaitu 4 roda Motor Vexta BLDC dengan output 100 watt dan speed 2500 r/min dan terdapat 2 driver motor serta 1 battery untuk menggerakkan Motor tersebut, Baterai yang digunakan adalah baterai LiPo 6 sel 24 volt.



**Gambar 1. Exoskeleton Assitive Walker**

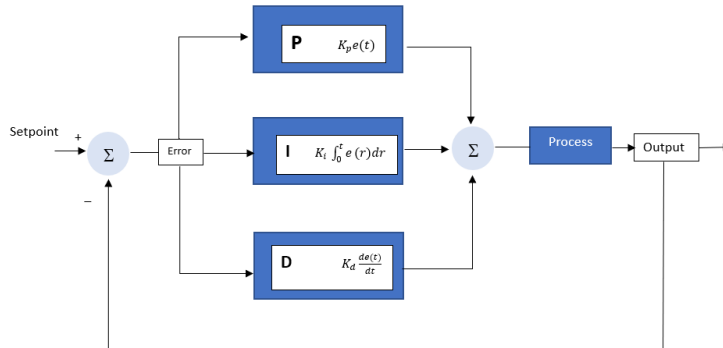
Robot *Exoskeleton Assitive Walker* ini dikontrol oleh 2 motor yang terdapat di 2 roda belakang yang kemudian ada 2 driver motor untuk setiap motornya dimana kedua driver ini terhubung pada baterai LiPo 6s 24 yang nantinya akan di kontrol oleh Arduino sebagai mikrokontrollernya. Tidak lupa ada sensor IMU GY-25 yang terhubung ke arduino seperti yang terlihat pada Gambar 2 sensor ini nantinya akan berperan sebagai input sekaligus menjadi setpoint untuk menentukan sudut.



**Gambar 2. Blok Diagram Sistem Exoskeleton Assitive Walker**

## 2.2. PID Controller

PID Controller merupakan hasil dari penggabungan tiga unsur yaitu *proportional*, *integral* dan *derivative* yang dapat digunakan secara bersamaan ataupun masing-masing sesuai kebutuhan.

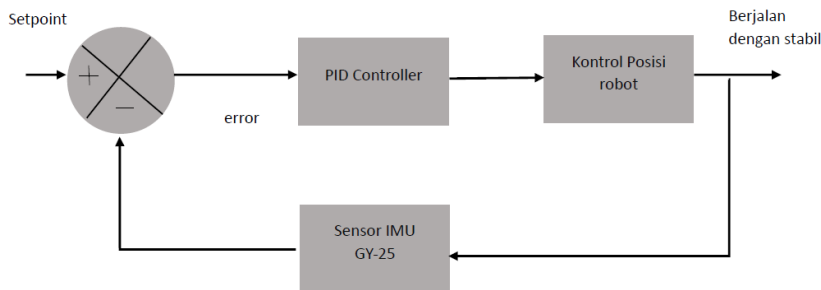


**Gambar 3. Blok Diagram Kontrol PID**

Pada Gambar 3 menunjukkan blok diagram kontrol PID yang umum digunakan di bidang industri[5]. Sistem kendali PID di penelitian ini berguna untuk mencapai nilai setpoint yang diinginkan. Sehingga didapatkan persamaan dari PID controller sebagai berikut:

$$u(t) = Kp e(t) + Ki \int_0^t e(t) dt + Kd \frac{de(t)}{dt} \quad (1)$$

Persamaan (1) menjelaskan bahwasannya nilai keluaran  $u(t)$ , merupakan jumlah gain dari gain ( $Kp$ ), gain ( $Ki$ ) dan gain ( $Kd$ ) yang masing – masingnya dipengaruhi error ( $e$ ) dalam selang waktu tertentu. Gambar 4 ini menjelaskan bahwa sistem kontrol menggunakan sistem kontrol *closed loop* dimana PID bertindak seperti penggerak motor dan mengontrol kecepatan motor ketika mendapat gangguan.[9], [10]. Yang di control dari system ini adalah variable output pada saat menuju titik tujuan yaitu lurus konstan 10 meter. Setpoint disini adalah setpoint dari sensor, jadi ketika motor jalan sampai tujuan dan pada rentang itu terjadi gangguan berupa dorongan terhadap robot dan setpoint berubah maka disinilah muncul sinyal error dan selanjutnya PID bekerja mengontrol motor untuk mengembalikan robot pada setpoint awal.



**Gambar 4.. Blok Diagram Sistem Kendali PID**

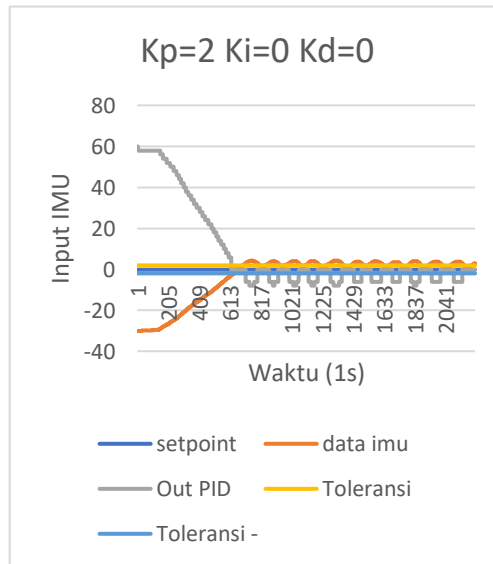
### 3. HASIL DAN PEMBAHASAN

#### 3.1. Membandingkan nilai $Kp$ , $Ki$ , $Kd$

Motor Vexta BLDC yang digunakan pada penelitian ini di atur supaya memiliki kecepatan konstan yang rendah yaitu pada 15 rpm agar robot dapat berjalan dengan lambat., dan perlu diingat bahwa penelitian ini bertujuan untuk mempermudah pemahaman konsep PID beserta pengaruhnya pada sistem dengan mengaplikasikannya langsung secara *Real Time*. Sehingga data yang digunakan disini tidak harus berupa numerik melainkan lebih kearah grafik perubahan parameter PID.

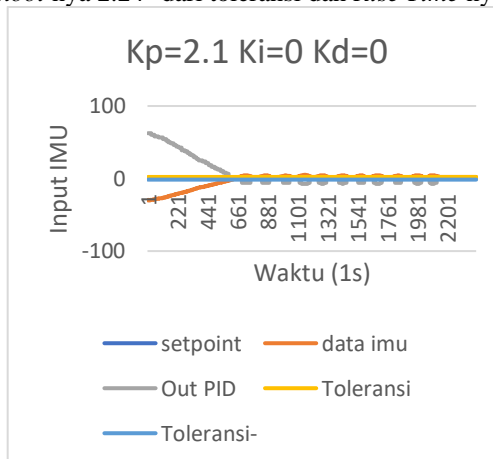
Ada 6 data yang diambil dan dari keenam gambar yang nantinya terdapat nilai  $K_p$ ,  $K_i$  dan  $K_d$  yang berbeda beda dengan kecepatan konstan motor jika lurus itu konstan 15 rpm dan dalam setiap pengujiannya setpointnya sama yaitu  $0^\circ$ . Pada pengujian yang dilakukan ada nilai toleransi sebesar  $2^\circ$ . Jadi apabila *Overshoot* lebih 1 sampai 2 derajat itu masih dianggap baik.

Pada Gambar 5a sampai dengan 5f merupakan Pengujian dari pencarian nilai  $K_p$  yang baik. Pemberian nilai  $K_p$  mengakibatkan rising timenya yang lama dan *Overshoot*nya yang tinggi. Jadi disini kita mencari nilai  $K_p$  yang baik dengan cara membandingkan dari ke enam data ini mana yang *Rise Time* nya lebih cepat dan *Overshoot* nya tidak terlalu tinggi.



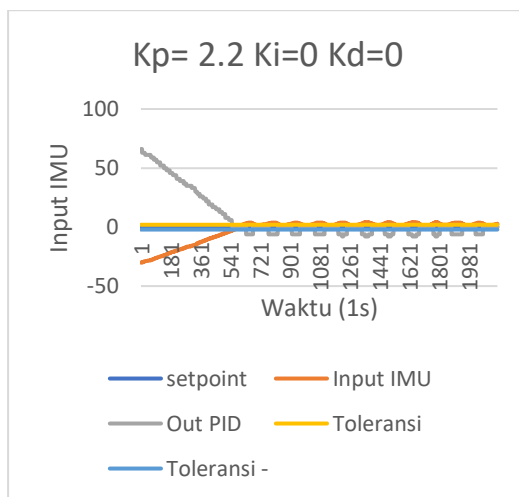
**Gambar 5a. Grafik data  $K_p$**

Pada gambar 5a *Overshoot* nya  $2.24^\circ$  dari toleransi dan *Rise Time* nya 5.8 detik.



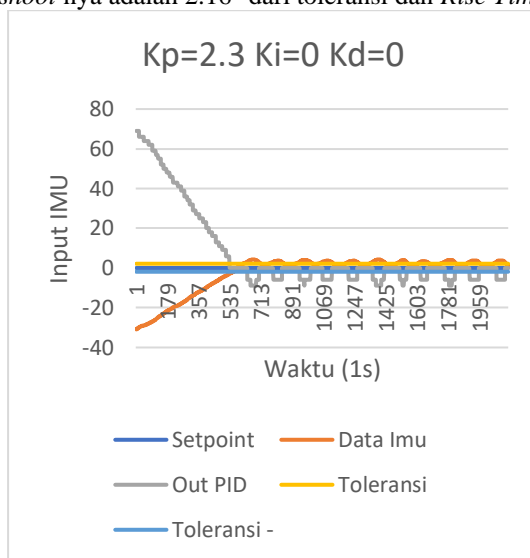
**Gambar 5b. Grafik data  $K_p$**

Pada gambar 5b. *Overshoot* nya lebih  $2.58^\circ$  dari toleransi dan *Rise Time* nya di angka 5.8 detik.



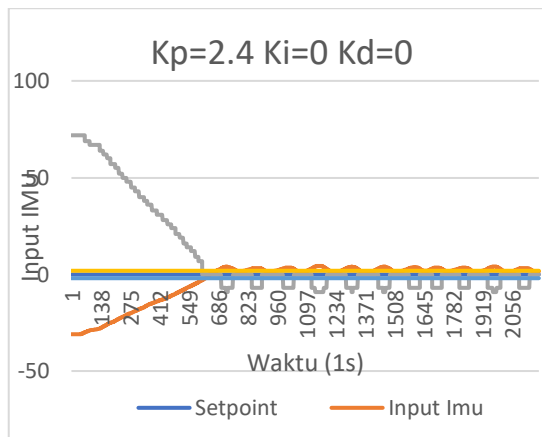
Gambar 5c. Grafik data Kp

Pada gambar 5c *Overshoot* nya adalah  $2.16^\circ$  dari toleransi dan *Rise Time* nya 4.9 detik.



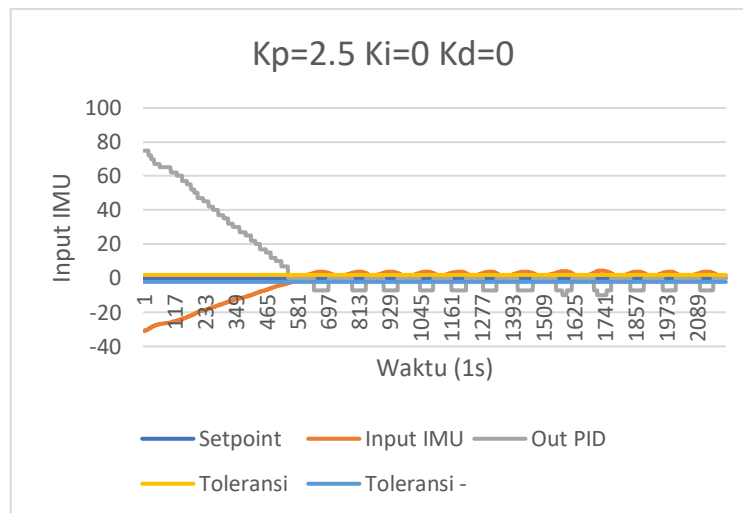
Gambar 5d. Grafik data Kp

Pada gambar 5d *Overshoot* nya lebih  $2.31^\circ$  dari toleransi dan *Rise Time* nya 5 detik.



Gambar 5e. Grafik data Kp

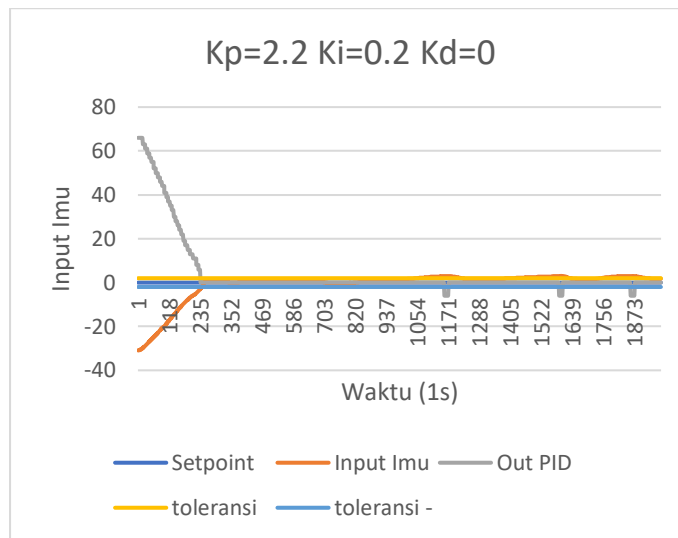
Pada gambar 5e *Overshoot* yang didapat yaitu lebih 2.26° dari nilai toleransi, sedangkan *Rise Time* nya adalah 5 detik.



Gambar 5f. Grafik data Kp

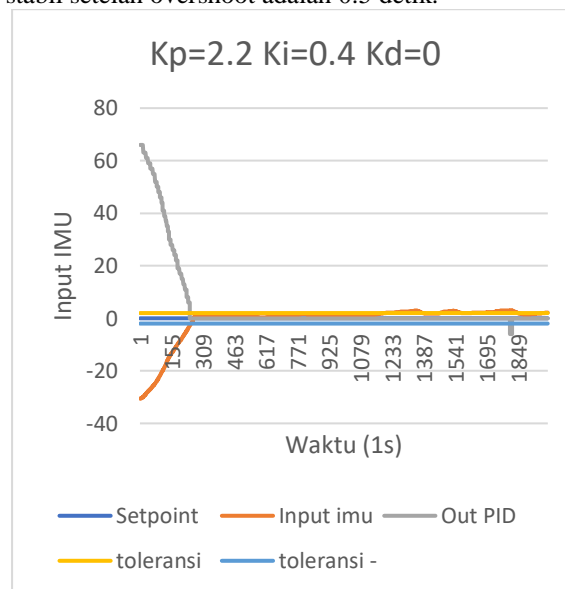
Pada gambar 5f *Overshoot* yang didapat mencapai 2.51°, lebih dari nilai toleransi dan nilai *Rise Time* nya yaitu 5.1 detik. Jadi dari pengujian ini dapat dilihat bahwa nilai Kp 2.2 memiliki *Rise Time* yang lebih singkat sehingga bisa lebih cepat menuju stabil dibandingkan nilai Kp yang lain bahkan nilai *overshoot* nya juga lebih rendah dan baik dibandingkan nilai Kp yang lain karena robot bisa lebih stabil berjalan meskipun belum ditambah nilai Ki dan Kd. Sehingga dapat ditentukan nilai Kp yang cukup baik adalah 2.2 karena *Overshoot* nya lebih rendah dan *rising time* yang cukup baik.

Langkah selanjutnya yaitu mencari nilai Ki yang nantinya akan digabung dengan Kp bisa dilihat ada enam data juga untuk mencari nilai Ki yang baik, tapi disini hanya akan ditampilkan 3 data saja. Menggabungkan nilai Kp dan Ki akan membuat robot tidak berosilasi secara terus menerus dan membuat robot lebih stabil. Dan disini untuk mencari nilai PI yang baik dilihat dari *Overshoot* nya *rise time* dan berapa waktu yang dibutuhkan robot dari *Overshoot* menuju stabil. Langkah pengambilan data ini dilakukan sama seperti pengambilan nilai Kp. Mengapa demikian? Karena metode yang digunakan adalah *Trial and Error*.



**Gambar 6a. Grafik data Ki**

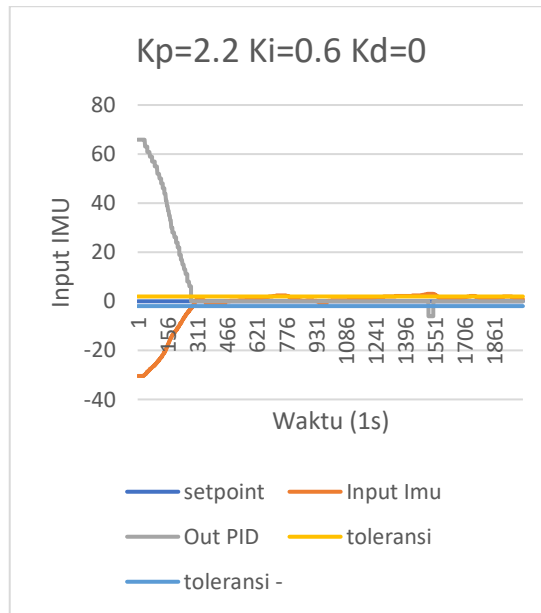
Pada Gambar 6a mengalami *Overshoot* sebesar  $1.6^\circ$ , lalu *Rise Time* nya 2.2 detik, dan waktu yang dibutuhkan untuk stabil setelah overshoot adalah 0.5 detik.



**Gambar 6b. Grafik data Ki**

Pada Gambar 6b terjadi *Overshoot* sebesar  $1.05^\circ$ , *Rise Time* nya adalah 2.2 detik, serta waktu yang dibutuhkan untuk mencapai stabil setelah *Overshoot* yaitu 0.5 detik.

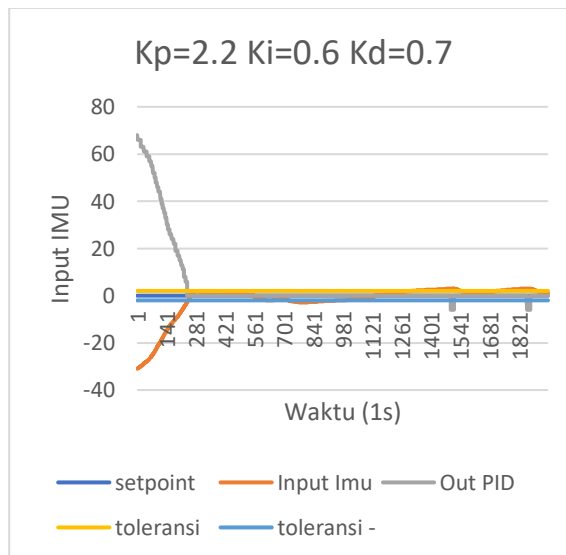




Gambar 6c. Grafik data Ki

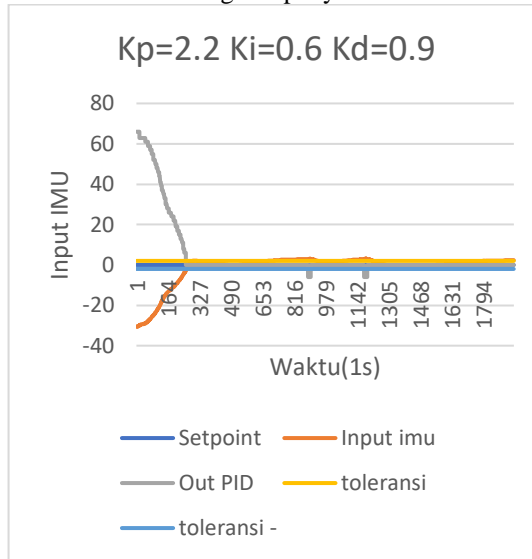
Lalu Pada Gambar 6c terjadi Overshoot sebesar  $1.11^\circ$  dengan Rise Time.2 detik dan waktu yang dibutuhkan untuk stabil setelah overshoot adalah 0.4 detik yang mana dari ketiga data yang ditampilkan ditentukan nilai Ki yang terbaik adalah 0.6.

Langkah terakhir adalah mencari nilai Kd yang berfungsi untuk meningkat respon system pada saat digabungkan juga dengan nilai Kp dan Ki supaya didapatkan waktu tercepat untuk mencapai stabil setelah overshoot. Disini dilakukan 6 kali pengujian yang cara pengujiannya sama persis seperti mencari nilai Kp dan Ki dan 6 data yang sudah tercatat tapi disini hanya 3 data saja yang akan ditampilkan.



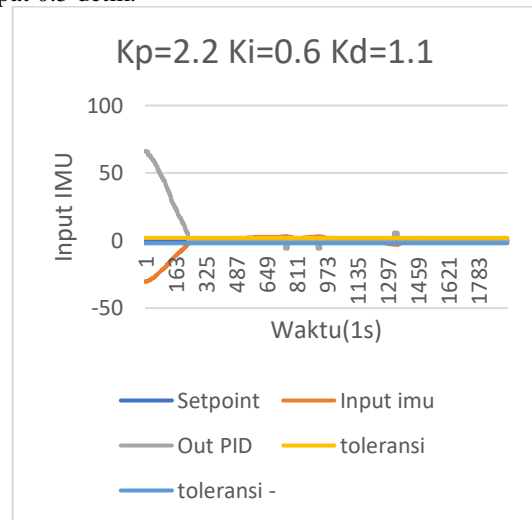
Gambar 7a. Grafik data Kd

Pada Gambar 7a. *Overshoot* yang didapat sebesar  $1.04^\circ$ , rise time 2.1 detik dan waktu yang dibutuhkan untuk stabil setelah *Overshoot* sangat cepat yaitu 0.3 detik.



Gambar 7b. Grafik data Kd

Pada Gambar 7b. *Overshoot* nya mencapai  $1.03^\circ$  dengan *Rise Time* 2.5 detik dan waktu untuk mencapai stabil cukup cepat 0.5 detik.

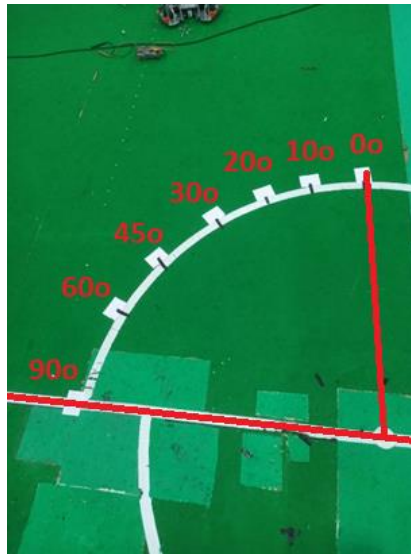


Gambar 7c. Grafik data Kd

Lalu yang terakhir pada Gambar 7c. *Overshoot* nya mencapai  $1.04^\circ$  dan ada *Undershoot* sebesar  $-1.17^\circ$ , lalu *Rise Time* nya adalah 2.3 detik dengan waktu untuk mencapai stabil yang cukup cepat 0.5 detik. Jadi dengan ini ditentukan bahwa nilai Kd yang terbaik untuk membuat robot stabil adalah 0.7. Dengan membandingkan beberapa nilai Kp, Ki dan Kd di tentukanlah nilai Kp = 2.2, Ki = 0.6, dan Kd = 0.7 sebagai nilai PID yang bisa membuat robot Exoskeleton Assitive Walker ini berjalan dengan lebih stabil.

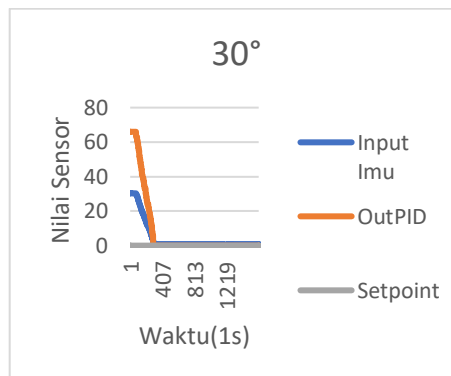
### 3.2. Mencari Nilai Setpoint

Pada Gambar 8 ini kita bisa menempatkan robot pada tengah tengah garis lalu setpoint kita atur  $0^\circ$  dan arahkan robot ke sudut yg sudah tertulis di kertas , sudut yang tertulis di kertas adalah sudut  $0^\circ$ ,  $10^\circ$ ,  $20^\circ$ ,  $30^\circ$ ,  $45^\circ$ ,  $60^\circ$  dan  $90^\circ$  di pengujian kali ini imu yang ada pada robot di arahkan pada sudut  $0^\circ$  lalu misalkan robot digeser ke arah  $30^\circ$ , robot akan kembali ke setpoint  $0^\circ$  dalam pengujian ini hanya mengambil sample sudut  $30^\circ$ ,  $60^\circ$ ,  $90^\circ$ .



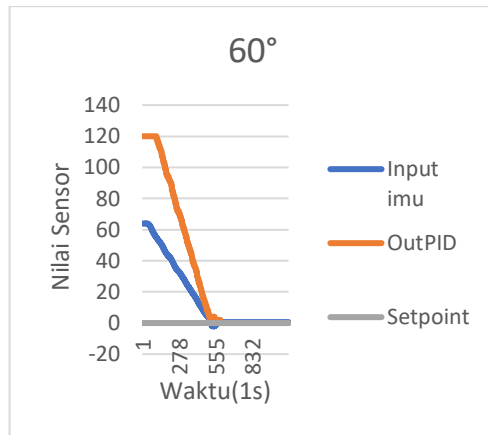
Gambar 8. Acuan Sudut

Gambar 9a adalah grafik yang ditampilkan disaat setpoint  $0^\circ$  , robot di gerakkan manual menuju  $30^\circ$  dan kembali ke  $0^\circ$  dengan rentang waktu untuk kembali ke setpoint hanya 2 detik.



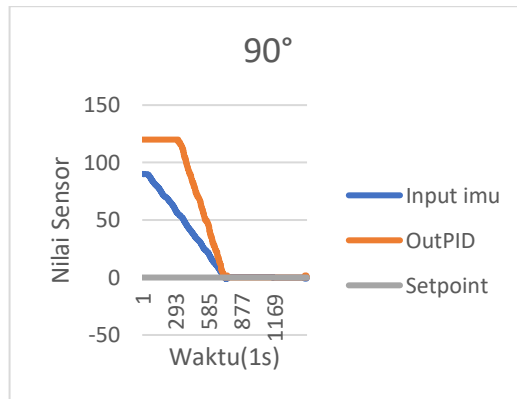
Gambar 9a. Grafik data dari sudut  $30^\circ$  ke  $0^\circ$

Gambar 9b adalah grafik yang ditampilkan disaat setpoint  $0^\circ$  robot di gerakkan manual menuju  $60^\circ$  dan kembali ke  $0^\circ$ . Rentang waktu yang dibutuhkan untuk mencapai setpoint 5 detik.



Gambar 9b. Grafik data dari sudut 60° ke 0°

Gambar 9c adalah grafik yang ditampilkan saat setpoint 0° digerakkan manual menuju 90° dan kembali ke 0°. Rentang waktu yang dibutuhkan untuk mencapai setpoint adalah 6,5 detik. Semua Kp, Ki dan Kd pada pengujian setpoint ini adalah Kp =2.2, Ki =0.6 dan Kd =0.7.



Gambar 9c. Grafik data dari sudut 90° ke 0°

#### 4. KESIMPULAN

Sistem kendali untuk mencapai setpoint telah dirancang menggunakan PID. Digunakan juga metode trial and error untuk menentukan nilai nilai gain yang cukup stabil pada sistem kendali PID dengan hasil Kp= 2.2, Ki = 0.6, Kd = 0.7. Nilai-nilai gain tersebut berhasil mengontrol motor untuk mencapai setpoint yang diinginkan dimana didapatkan nilai *Overshoot* yang kecil, *Rise Time* yang cukup cepat dan waktu yang dibutuhkan untuk mencapai setpoint yang lebih cepat dibanding percobaan percobaan yang lain. Lalu pada pengujian kedua dimana Setpoint telah di set ke 0° lalu diarahkan ke sudut yang berbeda beda dengan nilai gain dari Kp, Ki dan Kd yang sama hasilnya hampir akurat dengan waktu tempuh yang berbeda - beda untuk menuju setpoint.

#### DAFTAR PUSTAKA

[1] "RANCANG\_BANGUN\_ROBOT\_PEMADAM\_API\_BERODA\_DENGAN\_NAV".

- [2] T. Wang *dkk.*, “A Review on the Rehabilitation Exoskeletons for the Lower Limbs of the Elderly and the Disabled,” *Electronics (Switzerland)*, vol. 11, no. 3. MDPI, Feb 01, 2022. doi: 10.3390/electronics11030388.
- [3] A. Singla, G. S. Virk, B. Singh Rupal, dan G. Singh Virk, “Lower Limb Exoskeletons: A Brief Review 76 PUBLICATIONS 435 CITATIONS SEE PROFILE Lower Limb Exoskeletons: A Brief Review,” 2016. [Daring]. Available: <https://www.researchgate.net/publication/283532445>
- [4] Abu Hatim Kurniawan dan Muhammad Rivai, “Sistem Stabilisasi Nampun Menggunakan IMU Sensor dan Arduino Nano,” *JURNAL TEKNIK ITS*, vol. Vol.7, 2018.
- [5] Danu Wisnu, Arif Wahjudi, dan Hendro Nurhadi, “Perancangan Sistem Kontrol PID Untuk Pengendali Sumbu AzimuthTurretPada Turret-gunKaliber 20mm,” *JURNAL TEKNIK ITS* , vol. Vol. 5, 2016.
- [6] Mohammad Irfa’anul Ma’arif, Fauzi Imaduddin Adhim, dan Fivitria Istiqomah, “Implementasi Metode Pid untuk Mengontrol Posisi Motor Servo pada Sistem Sortir Berat Adonan,” *JURNAL TEKNIK ITS* , vol. Vol.10, 2021.
- [7] H. Wicaksono dan J. Pramudijanto, “Kontrol PID Untuk Pengaturan Kecepatan Motor DC Dengan Metode Tuning Direct Synthesis,” 2004. [Daring]. Available: <http://puslit.petra.ac.id/journals/electrical/>
- [8] R. Arindya, “PENALAAAN KENDALI PID UNTUK PENGENDALI PROSES,” Seminar Nasional Cendekiawan, 2015.
- [9] Indro Zaini, Handry Khoswanto, dan Felix Pasila, “Prototipe Balancing Robot Dengan Metode Kendali PID,” *Jurnal Teknik Elektro*, vol. Vol. 10, 2017.
- [10] W. Purbowaskito dan C.-H. Hsu, “Sistem Kendali PID untuk Pengendalian Kecepatan Motor Penggerak Unmanned Ground Vehicle untuk Aplikasi Industri Pertanian,” *JURNAL INFOTEL*, vol. 9, no. 4, hlm. 376, Nov 2017, doi: 10.20895/infotel.v9i4.253.