Buletin Ilmiah Sarjana Teknik Elektro
Vol. 1, No. 3, Desemner 2019, pp. 131~143
ISSN: 2685-9572, DOI:10.12928/biste.v1i3.1126

Implementasi PID Navigasi Pelacakan Titik Api dengan Sensor *Flame Array* pada Robot *Hexapod* KRPAI

Muhammad Dhia Dzulfiqar[1], Nuryono Satya Widodo[2]
[1] Mahasiswa Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Indonesia
[2] Dosen Program Studi Teknik Elektro, Universitas Ahmad Dahlan, Indonesia

INFORMASI ARTIKEL

Riwayat Artikel:

Dikirimkan 01 Oktober 2019,
Direvisi 13 Oktober 2019,
Diterima 16 Oktober 2019.

Kata Kunci:

Hexapod,
Pengendali PID,
flame array,
Sharp GP,
Light Following.

Penulis Korespondensi:

Muhammad Dhia Dzulfiqar,
Program Studi Teknik Elektro,
Universitas Ahmad Dahlan,
Kampus 4 UAD,
Jln. Ring Road Selatan,
Tamanan, Banguntapan,
D.I. Yogyakarta, Indonesia.
Surel:
dhiadzulfiqar@gmail.com

ABSTRAK

Robot *Hexapod* Pemadam Api merupakan robot berkaki 6 yang bertugas menyusuri ruangan labirin untuk menemukan dan memadamkan api dalam waktu yang singkat. Dalam menjalankan tugasnya dibutuhkan 2 buah sensor untuk menjalankan tugas ini yaitu sensor *flame array* dan sensor *Sharp GP*. Sensor *flame array* terdiri dari 16 buah sensor *flame* yang digunakan untuk mengetahui nilai *error* berdasarkan letak keberadaan api yang mampu dijangkau dalam rentang horisontal 180 derajat. Sementara sensor Sharp GP berfungsi untuk mengetahui jarak antara robot dengan lilin sebagai sumber api. Kedua sensor ini berperan sebagai pelengkap dalam bernavigasi *light following* agar robot mampu menemukan, menghampiri, dan memposisikan bagian tengah muka robot berhadapan dengan api tanpa menabraknya. Dalam bernavigasi robot dikontrol oleh Pengendali PID (Proportional-Integral-Derivative). Hasil pengujian menunjukan bahwa pengendali proportional berperan mempercepat robot mengarah ke api. Pengendali derivative berperan dalam meredam terjadinya osilasi yang disebabkan kontrol proportional dalam mengejar titik api. Sementara kontrol integral berperan dalam merevisi kekeliruan robot pada set point. Hasil pengujian lain menunjukan keberhasilan dari implementasi pada robot *hexapod* untuk memadamkan api membutuhkan waktu rata-rata 5,5 detik. Sementara nilai parameter PID terbaik adalah Kp=35, Ki=20, dan Kd=20.

Fire Extinguisher Hexapod Robot is a 6-legged robot whose job is to navigate the labyrinth room to find and extinguish the fire in a short time. In carrying out their duties, two sensors are needed to carry out this task, namely the flame array sensor and the Sharp GP sensor. Flame array sensor consists of 16 flame sensors that are used to determine the error value based on the location of the existence of a fire that can be reached within the horizontal range of 180 degrees. While the Sharp GP sensor functions to determine the distance between the robot and the candle as a source of the fire. Both of these sensors act as a complement in navigating the light following so that the robot is able to find, approach, and position the centre of the face of the robot facing the fire without crashing into it. In navigating the robot is controlled by PID (Proportional-Integral-Derivative) Controller. The test results show that the proportional controller has the role of accelerating the robot to fire. The derivative controller plays a role in reducing the occurrence of oscillations caused by proportional control in the pursuit of hotspots. While integral control plays a role in revising the errors of the robot at the set point. Other test results show the success of the implementation of the hexapod robot to extinguish the fire takes an average of 5.5 seconds. While the best PID parameter values are Kp = 35, Ki = 20, and Kd = 20.

Sitasi Dokumen ini:

M. D. Dzulfiqar and N. S. Widodo, "Implementasi PID Navigasi Pelacakan Titik Api dengan Sensor Flame Array pada Robot Hexapod untuk Kontes Robot Pemadam Api Indonesia," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 1, no. 3, pp. 131--143, 2019. DOI: 10.12928/biste.v1i3.1126

1. PENDAHULUAN

Kontes Robot Pemadam Api Indonesia (KRPAI) merupakan ajang rancang bangun dan rekayasa dibidang robotika yang diadakan rutin setiap tahun yang menggunakan robot berkaki bertipe hexapod. Ajang tersebut merupakan lomba skala regional dan nasional yang diselenggarakan oleh Kementrian Riset, Teknologi, dan Pendidikan Tinggi (Kemenristekdikti) [1]. Kontes robot ini memiliki antusiasme yang tinggi dari berbagai mahasiswa dari beragam banyak universitas yang ada di Indonesia. Robot *Hexapod* yang digunakan dalam Kontes ini merupakan Robot Pemadam Api berkaki 6 yang bertugas menyusuri ruangan labirin untuk menemukan dan memadamkan api dalam waktu yang singkat [2] [3].

Masalah yang dihadapi oleh robot hexapod KRPAI adalah robot sebelumnya hanya mengandalkan 1 buah sensor *Thermopile array* (TPA) yang mengharuskan robot melakukan pelacakan berputar untuk menghampiri api [4] [5]. Meskipun tingkat keberhasilanya tinggi, tetapi kinerja yang dilakukan dinilai lamban karena robot dalam memadamkan api harus berputar sampai sensor TPA mendeteksi api [6]. Jika sensor mendeteksi api robot baru akan menghampiri api hingga robot cukup dekat dengan lilin. Sementara jika saat menghampiri api lalu sensor TPA tiba-tiba tidak mendeteksi api, maka robot akan berputar lagi hingga sensor TPA mendeteksi keberadaan api. Dengan kata lain robot akan menghampiri api jika sensor TPA mendeteksi adanya api dan robot akan berputar mecari keberadaan api jika sensor TPA tidak mendeteksi adanya api. Selain itu pada penelitian sebelumnya telah berhasil mengimplementasikan algoritma *wall following* yang dikontrol oleh kontrol *Proportional-Integral-Derivative* (PID) [7]. Selain pengendali PID, terdapat penelitian lain yang menggunakan metode lain seperti metode pengendali fuzzy [8]. Pengendali fuzzy memiliki kelemahan yaitu memerlukan memori lebih banyak dibandingkan dengan pengendali PID oleh karena penelitian ini akan menggunakan Pengendali PID.

Berdasarkan uraian latar belakang tersebut, penelitian ini mengusulkan tentang penggantian sensor TPA dengan sensor *flame array* dalam tujuannya untuk mengetahui keberadaan api, menghampiri api sembari menyejajarkan muka robot bagian tengah simetris dengan api, dan berhenti didepan lilin sebagai sumber api tanpa menabraknya. Sensor ini bekerja dengan memberikan nilai *error* yang nantinya akan diproses oleh kontrol PID dalam algoritma *light following*. Dengan mengimplementasikan Algoritma *light following* yang dikontrol PID, diharapkan robot tidak perlu melakukan pelacakan berputar lagi. Ketika sensor mendeteksi keberadaan api, saat itu juga robot bermanuver untuk menghampiri api sembari mengarahkan muka robot agar simetris dengan letak sumber api. Lalu ketika sudah dekat dengan sumber api robot akan berhenti dan siap memadamkannya tanpa menabrak lilin sebagai sumber api. Sensor yang digunakan untuk mengiplementasikan algoritma ini adalah sensor *flame array* 16 bit dan sensor Sharp GP. Sensor *flame array* berfungsi untuk mendeteksi keberadaan api dengan rentang jangkauan horisontal 180^o dan sensor Sharp GP berfungsi untuk mengetahui jarak antar lilin dengan robot. Sensor Sharp GP berperan penting untuk menghindari robot menabrak lilin sebagai sumber api.

2. METODE PENELITIAN

Dalam sistem robot terdapat rancangan perangkat keras, rancangan elektronik, dan perancangan perangkat lunak**.** Setelah itu dilanjutkan dengan penjelasan pengujian sistem yang dilakukan.

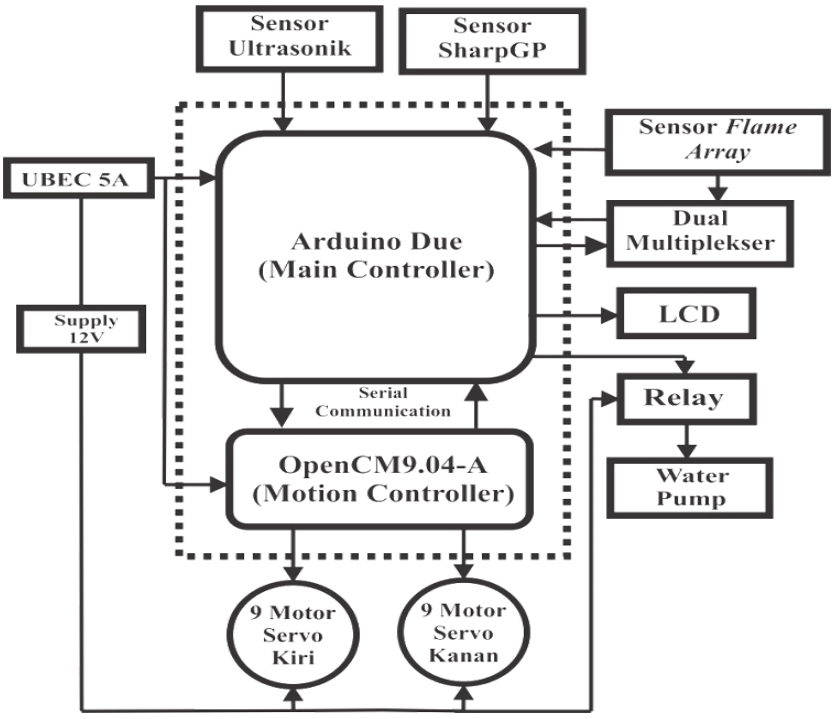
2.1. Desain Sistem

Diagram blok sistem robot ditunjukkan pada Gambar 1 dengan keterangan sebagai berikut. Pada penelitian ini robot menggunakan 2 mikrokontroler yaitu Arduino Due dan Open CM 9.04A. Arduino Due berperan sabagai pemroses yang bertugas untuk mengakuisi nilai-nilai sensor yang digunakan dan memperoses nilai *error* pada kontrol PID yang menentukan keluaran pulsa langkah untuk kaki sektor kiri dan sektor kanan. Keluaran PID tadi lalu dikirim ke Open CM 9.04 sebagai *slave* via komunikasi serial. OpenCM 9.04 ini berperan dalam mengatur *gait* pada kaki-kaki robot.

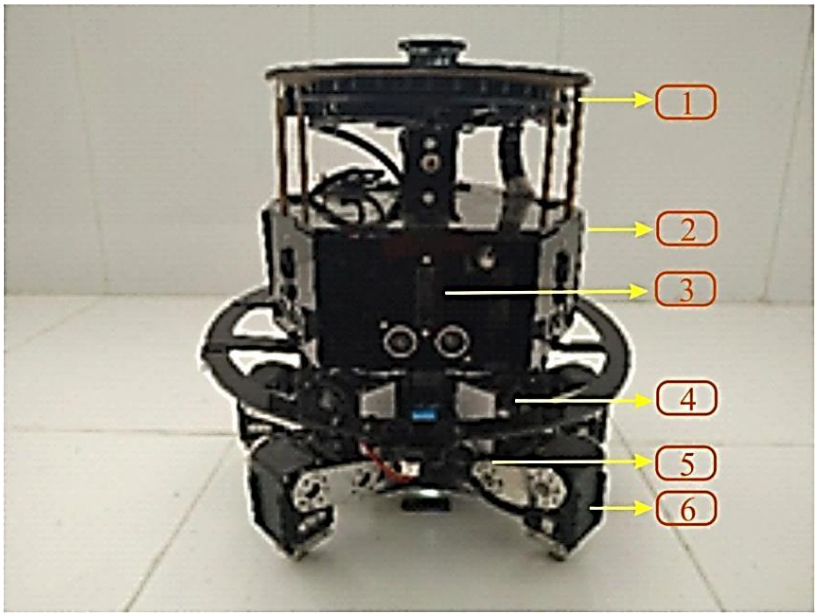
Sementara, perangkat yang lain adalah sensor ultrasonic dan sensor sharp yang berfungsi untuk mendeteksi dinding pada labirin. Sensor Flame array yang berfungsi untuk mendeteksi titik api (lilin), LCD yang berfungsi untuk menampilkan data sensor, Relai yang berfungsi sebagai driver water pump, dan water pump yang berfungsi untuk menyemprotkan air untuk memadamkan api.

2.2. Desain Alat

Robot dituntut memiliki rancangan perangkat keras yang sesuai dengan kebutuhanya dalam menjalankan misi. Selain itu komponen penting pada robot diletakan pada posisi yang seharusnya agar kinerja robot dapat berjalan sebagaimana mestinya. Komponen penting robot beserta tata letaknya dapat dilihat Gambar 2. Bagian penyusun robot pada Gambar 1 terdiri dari komponen-komponen yang dipaparkan pada Tabel 1.



Gambar 1. Diagram blok sistem robot



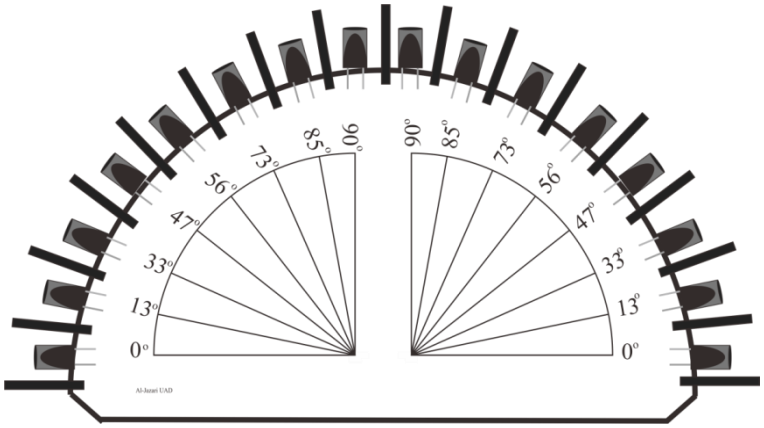
Gambar 1. Hasil perancangan perangkat keras

Tabel 1. Komponen penyusun rancangan robot

No	Nama Komponen
1	Sensor *Flame Array*
2	Bagian Kepala Robot
3	Sensor Sharp GP
4	Bagian Badan Robot
5	Bagian Kaki Robot
6	Servo Dynamixel AX-18 A

Pada Gambar 1 Robot yang dirancang memiliki dimensi 25 cm x 25 cm x 27 cm (panjang x lebar x tinggi). Selain itu pada bagian robot diberikan sensor Sharp GP yang letaknya tidak boleh lebih tinggi dari lilin. Sensor

Sharp GP berperan untuk mengetahui jarak antara lilin dengan robot. Terdapat juga sensor *flame array* berada pada ketinggian yang sesuai dengan tinggi lilin sebagai sumber api. Penampakan sensor *flame array* dapat dilihat Gambar 3.



Gambar 2. Sensor *flame array* 16 bit

Sensor *flame array* adalah susunan dari 16 buah sensor *flame* menjadi satu kesatuan yang dapat mengetahui api. Letak keberadaan api ini nantinya akan menghasilkan kondisi biner 16-bit yang bisa dijadikan refrensi dalam menemukan nilai *error* untuk diproses oleh kontrol PID.

2.3. Pengendali PID

Algoritma navigasi pelacakan titik api dengan kontrol PID berperan dalam menentukan pergerakan robot untuk mecapai tujuannya. PID ini memberikan nilai *output* untuk menentukan seberapa besar kaki melangkah, sedangkan inputnya didapat dari nilai *error* yang berkaitan dengan kondisi sensor *flame array*. Proses pengolahan PID dilakukan secara aritmatika yang prosesnya berlangsung dalam mikrokontroler Arduino Due. Persamaan PID [9][10] yang terdapat dalam Arduino Due adalah sebagai berikut,

$$PID = ((Kp * error) + \left(\frac{integral\_error * Ki}{Ti}\right) + \big((error - last\_error) * Kd\big), \quad (1)$$

Dengan nilai error didapat dari selisih antara nilai *set point* dengan nilai *present value* dengan persamaan sebagai berikut,

$$Error = Set\_Point - Present\_Value, \quad (2)$$

Sementara nilai last error adalah nilai error yang sebelumnya, nilai integral error adalah nilai total penjumlahan seluruh error dari error yang pertama sampai error yang terakhir. Variabel Kp adalah nilai penguatan pengendali proporsional, Ki adalah nilai penguatan pengendali integral dan Kd adalah nilai penguatan pengendali derivatif.

2.4. Algoritma

Dalam menjalankan misinya robot harus memiliki kecerdasan berdasarkan instruksi yang merupakan hasil *compile* dari kegiatan program yang dilakukan. Instuksi pada robot inilah yang disebut perangkat lunak. Perangkat lunak pada robot disimpan pada mikrokontroler Arduino Due yang terdiri dari program untuk algoritma *light following* keseluruhan dan program PID sedangkan pada mikrokontroler OpenCM 9.04A disimpan program pola gerakan kaki robot. Alur proses PID robot dapat dilihat Gambar 3.

Diagram alir yang ditunjukkan pada Gambar 3 adalah diagram alir sistem kontrol PID yang dirancang pada robot. Proses perhitungan PID dimulai dari proses akumulasi sensor, kemudian akan dilakukan pengecekan apakah *error* terakhir sama dengan nilai *error* saat itu, jika nilai *error* sama maka akan dianggap *error* = 0, namun jika tidak sama maka nilai *error* bisa berupa bilangan negatif atau positif dan akan dilakukan pengecekan kembali untuk memastikan nilai *error* yang akan diakumulasi.

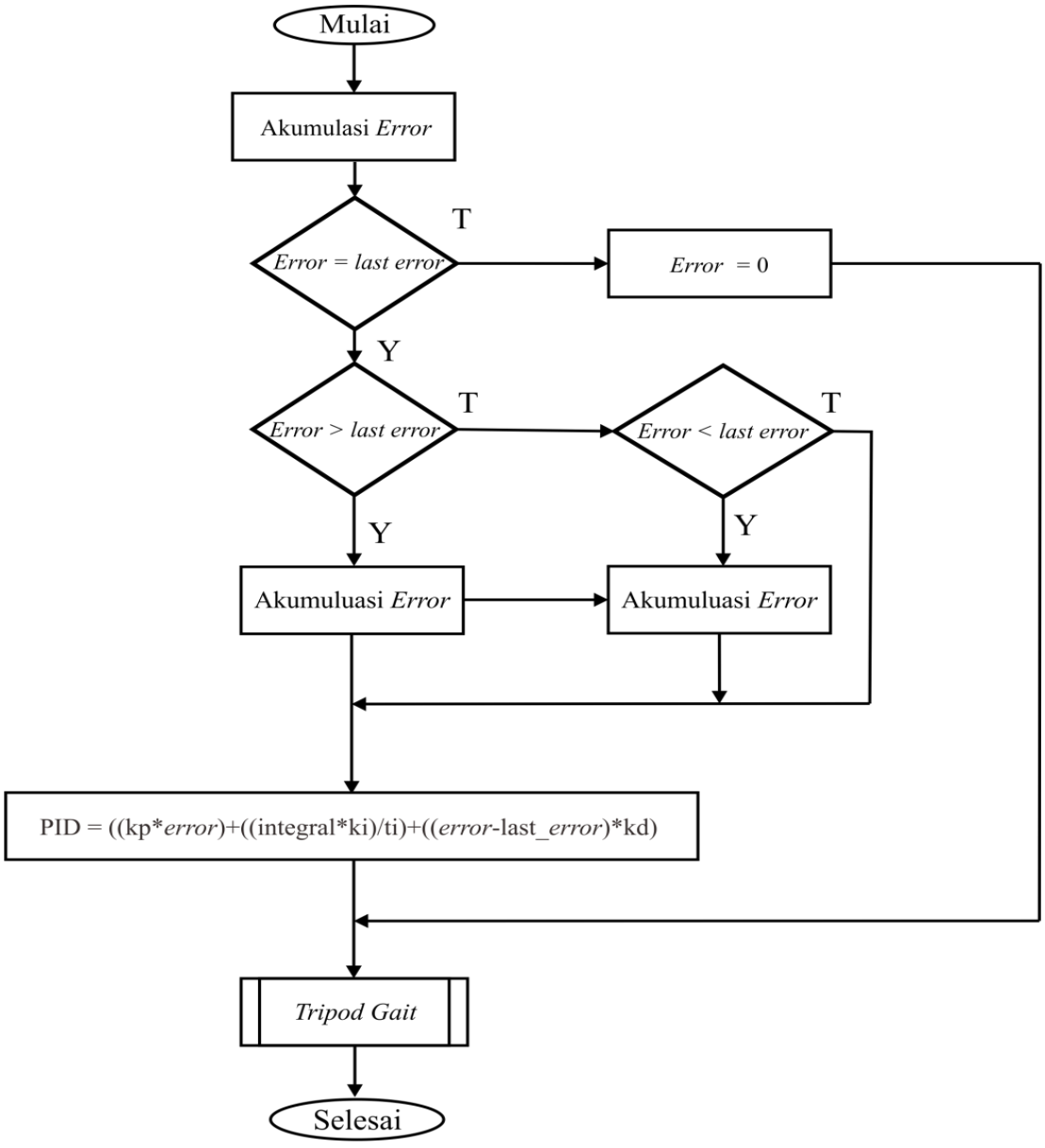
Dalam menuju nilai *error*=0 sesuai yang diharapkan, Kontrol PID mejadikan nilai *error* sebelumnya sebagai refrensi untuk perhitungan selanjutnya sehingga nilai *error* yang didapat mendekati bahkan bernilai 0. Robot selalu berusaha bergerak searah dengan *set point* yang telah ditentukan.

Output PID adalah hasil penjumlahan dari kontrol P, Kontrol I, dan Kontrol D. Kontrol *proportional* berperan untuk kondisi terkini yang bertugas untuk sistem mencapai *set point* yang diinginkan dalam waktu singkat, sedangkan kontrol *derivative* berperan untuk kondisi mendatang yang bertugas untuk meredam

terjadinya *overshoot* yang disebabkan oleh kontrol P, dan kontrol integral berperan untuk kondisi lampau yang bertugas menaikan atau menurunkan *output* jika sistem sudah stabil namun pada *set point* yang tidak tepat.

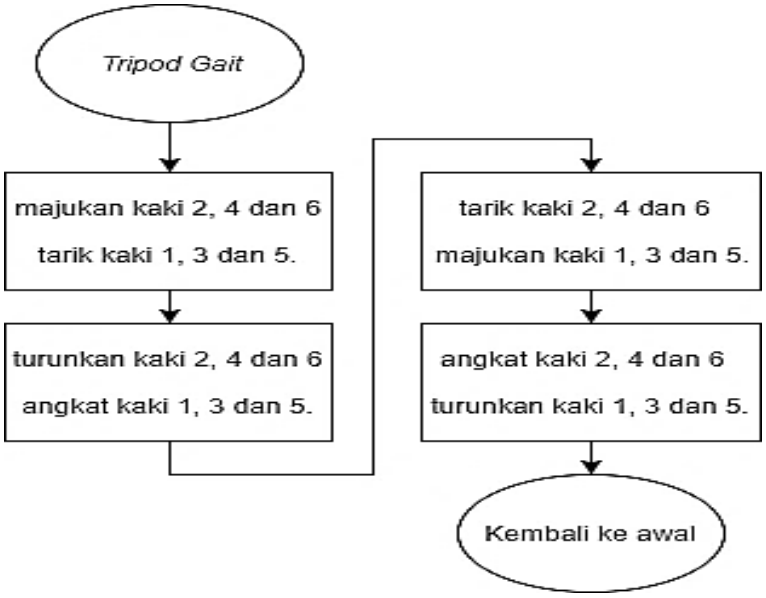
Output PID ini pada akhirya akan digunakan oleh kaki robot dalam menentukan besaran langkah robot harus bergerak. Semakin besar langkah kiri maka robot akan bergerak condong ke arah kanan, sebaliknya semakin besar langkah kanannya maka robot akan bergerak codong ke arah kiri.

Pergerakan kaki yang digunakan dalam robot *hexapod* ini adalah metode *tripod gait. Tripod gait* merupakan pola gerakan robot dengan menjadikan 3 kaki sebagai penopang sedangkan 3 kaki lainnya mengangkat untuk melakukan perpindahan. Proses ini dilakukan secara bergantian setiap 3 kaki. Diagram alir *tripod gait* dapat dilihat Gambar 4.

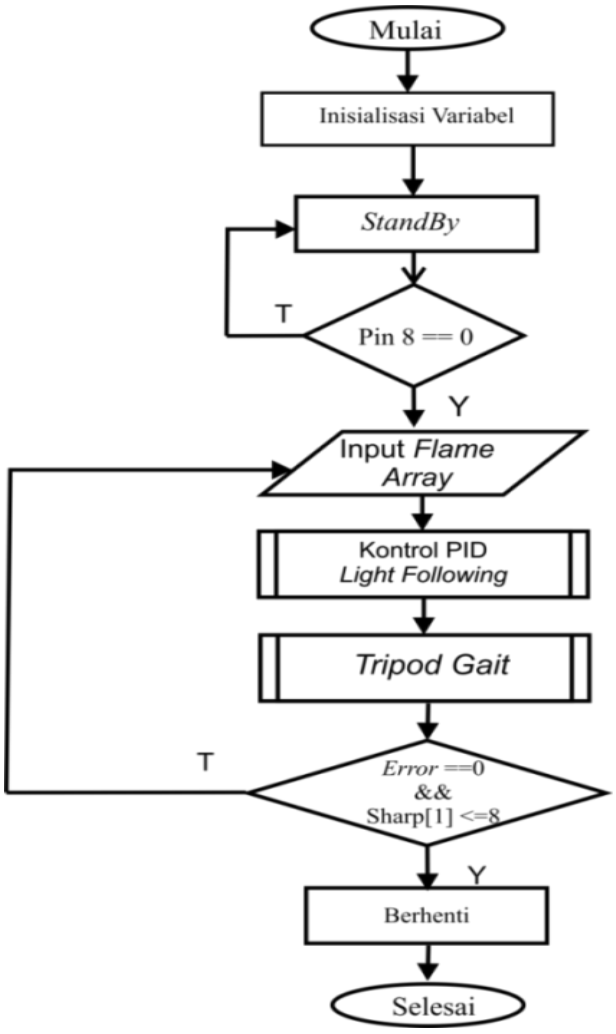


Gambar 3. Diagram alir sistem kontrol PID

Dalam melaksanakan instruksi *tripod gait* dibutuhkan *trim* pada setiap kaki-kakinya agar gerakan robot lebih pasti. *Trim* adalah kegiatan menyejajarkan kaki robot guna memastikan seluruh kaki menyentuh permukaan dengan tekanan yang tepat. Setelah mengetahui alur perhitungan sistem kontrol PID selanjutnya mengetahui sistem alur algoritma robot dalam mengejar titik api. Diagram alirnya dapat dilihat pada Gambar 5.



Gambar 4. Diagram alir *tripod gait*



Gambar 5. Diagram alir sistem navigasi *light following*

Pada Gambar 5 dijelaskan bahwa sistem *software* yang dibagun pada robot, dimulai dari inisialisasi variabel, kemudian masuk ke posisi *default* robot yaitu adalah posisi *home* robot saat catu daya dinyalakan, pada tahap ini robot bisa diatur nilai Kp, Ki, dan Kdnya dengan menekan tombol yang tersedia pada robot. Kemudian dilakukan penekanan tombol sebagai intruksi robot memulai berkerja sesuai program yang tertanam khususnya program kendali PID yang mengontrol pergerakan kaki robot sehingga robot segera mendekati titik api, sehingga ketika tombol ditekan robot akan masuk ke intruksi menerima nilai sensor lalu memprosesnya pada program PID yang telah dibuat dengan nilai yang telah diolah dijadikan *output* pengendali langkah kaki kanan dan kiri sehingga robot akan bergerak mengikuti dan mendekati api. Hingga pada akhirnya robot berhenti tepat didepan api dengan bagian muka robot berada sejajar dengan titik api.

2.5. Pengujian Sistem

Pengujian dilakukan dengan menggunakan kamera sebagai media untuk merekam pergerakan robot mengejar titik api dari atas. Selanjutnya hasil rekaman video akan diproses oleh citra atau biasa disebut *tracking*. *Tracking* dilakukan dengan tujuan mendapatkan *plotting* atau jejak pergerakan robot dalam menghampiri titik api.

3. HASIL DAN PEMBAHASAN

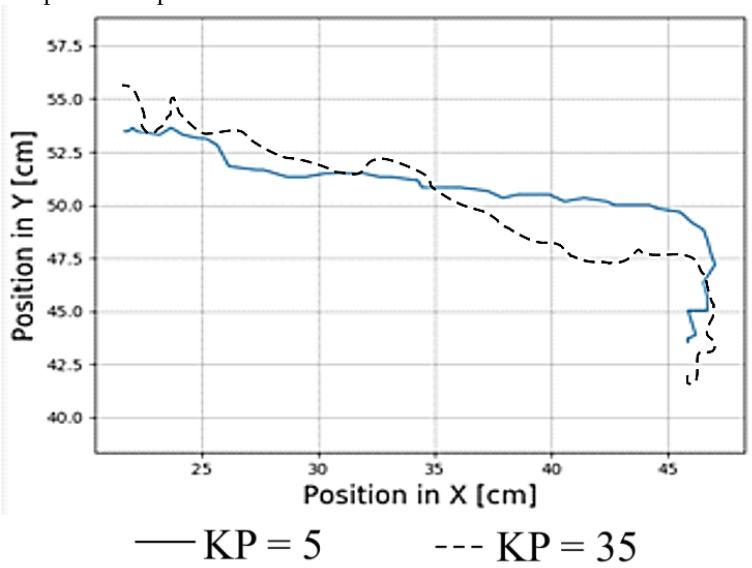
Pengujian ini dilakukan dengan mengamati *plot* yang dihasilkan dari pergerakan robot mendekati api hingga berhasil. Keberhasilan ditandai dengan kondisi dimana bagian depan tengah robot sejajar dan dekat dengan titik api tanpa menabrak lilin sebagai sumber api.

Setpoint yang digunakan adalah 0 dan akan mempengaruhi nilai *error* berdasarkan operasi pengurangan oleh *present value* dari sensor *flame array* yang bisa bernilai positif maupun negatif. Nilai-nilai ini terdapat dalam perangkat lunak yang jika dijelaskan akan rumit dipahami, sehingga pada hasil pengujian *setpoint* diimajinasikan sebagai titik api yang diwakili oleh objek biru bulat yang harus dihampiri robot. Hal ini bisa dilakukan karena *setpoint* hakikatnya adalah nilai yang diinginkan dicapai oleh sistem tapi dalam *plot* ini *setpoint* adalah objek yang ingin dicapai robot.

Percobaan dilakukan dengan nilai Kp, Ki, dan Kd yang bervariasi. Pada pengujian Kp dilakukan pengujian dengan nilai konstanta Kp bernilai 5 dan 35. Sedangkan pengujian Kd menggunakan nilai konstanta 5 dan 35 juga hanya saja dengan Kp 20. Pengujian Ki dilakukan dengan nilai Ki 5 dan 35 dengan permberian nilai Kp dan Kd 20.

3.1. Pengujian Kontrol Proporsional (P)

Pengujian ini dilakukan dengan memberi nilai Kp=5 dan Kp=35 sementara nilai Ki=0 dan Kd=0. Percobaan ini menguji keakuratan robot hanya dengan menggunakan kontrol P. Hasil pengujian robot dengan nilai KP=5 dan Kp=35 dapat dilihat pada Gambar 5.



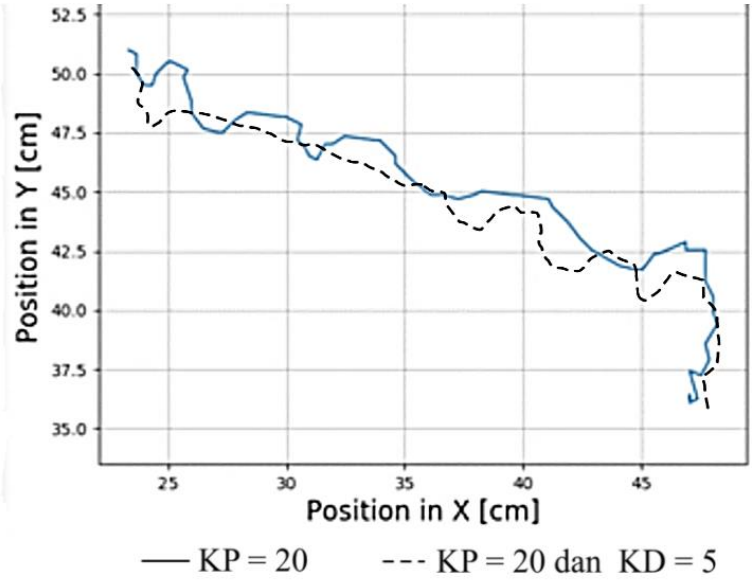
Gambar 6. Perbandingan hasil ploting Kp

Pada Gambar 6 dapat dipahami bahwa kontrol *propotional* berperan sebagai dorongan dari responsifnya kontrol untuk mencapai nilai *set point*, namun kontrol ini memiliki kekurangan yaitu tidak dapat meredam *over shot*. Semakin kecil KP yang diberikan maka akan dibutuhkan waktu yang lama untuk mencapai *set point* namun *over shot* yang ditimbulkan relatif sedikit. Sedangkan semakin besar nilai KP maka waktu yang

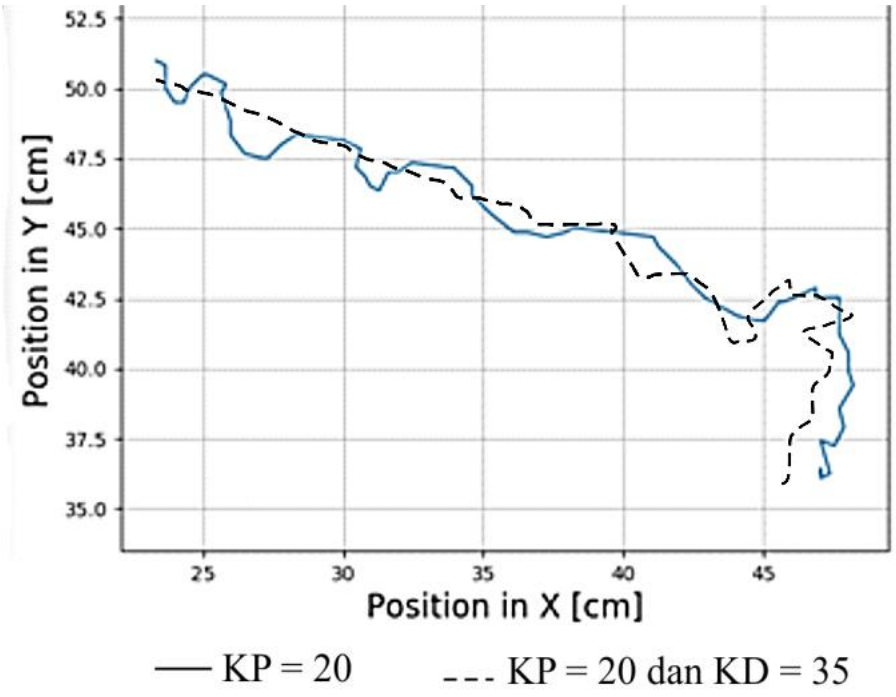
dibutuhkan untuk mencapai *set point* lebih cepat, namun *over shot* yang dihasilkan besar sehingga menyebabkan osilasi yang panjang. Sebab itulah pemberian nilai KP bernilai 5 menunjukan *plot* yang lebih lembut dibanding pemberian KP dengan nilai 35.

3.2. Pengujian Kontrol Proporsional-Derivatif (PD)

Pengujian ini dilakukan dengan nilai Kp = 20 sedangkan Kd yang diberikan bervariasi yaitu dengan nilai 5 dan 35. Pengaruh pemberian nilai Kd maka dapat dilihat Gambar 7 dan Gambar 8.



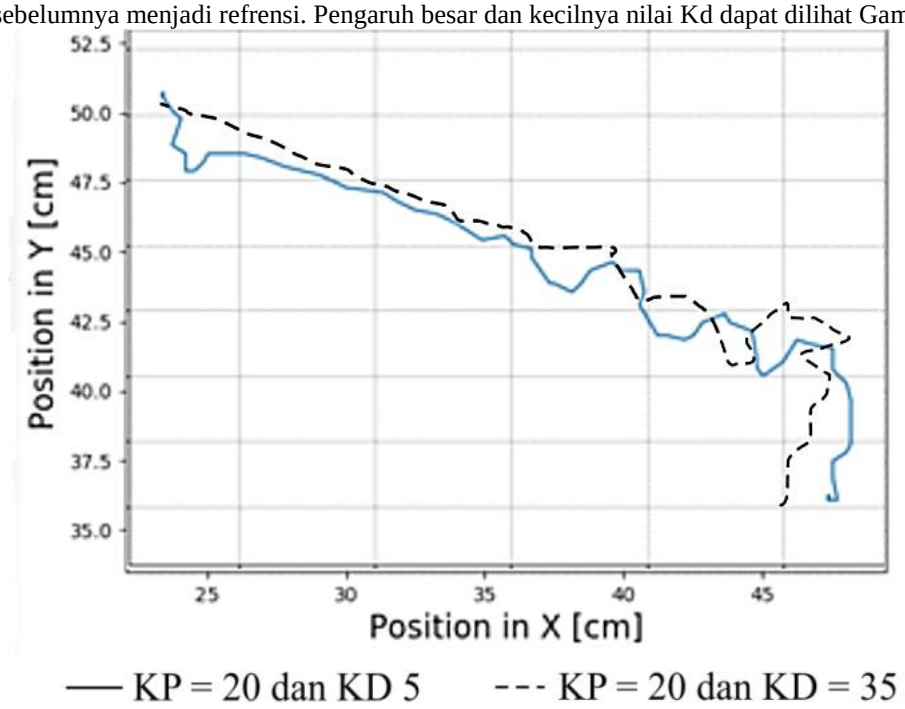
Gambar 7. Perbandingan ploting KD bernilai 5



Gambar 8. Perbandingan ploting KD bernilai 35

Pada Gambar 7 dan Gambar 8 dapat dipahami bahwasannya nilai Kd yang diberikan menyebabkan pergerakan lebih cepat untuk stabil sehingga osilasi yang terjadi lebih minimal bahkan seperti tidak ada osilasi sama sekali. Hal ini terjadi karena Kd berperan untuk menghambat kecepatan Kp untuk sampai *set point* yang

semakin dekat set point maka peranannya makin besar untuk menghambatnya. Sedangkan setelah terjadi *over shot* maka Kd berperan membantu untuk kembali ke *set point* sesegera mungkin dengan menjadikan nilai keluaran sebelumnya menjadi refrensi. Pengaruh besar dan kecilnya nilai Kd dapat dilihat Gambar 8.

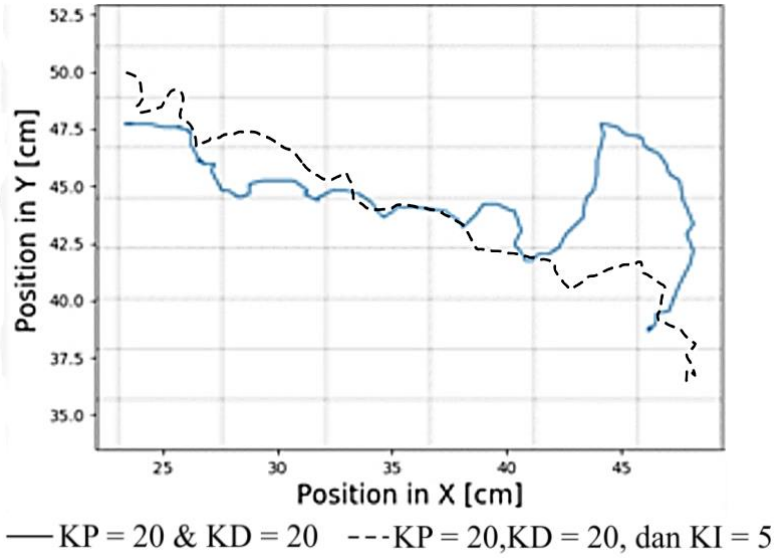


Gambar 9. Perbandingan Plot KD = 5 dan KD = 35

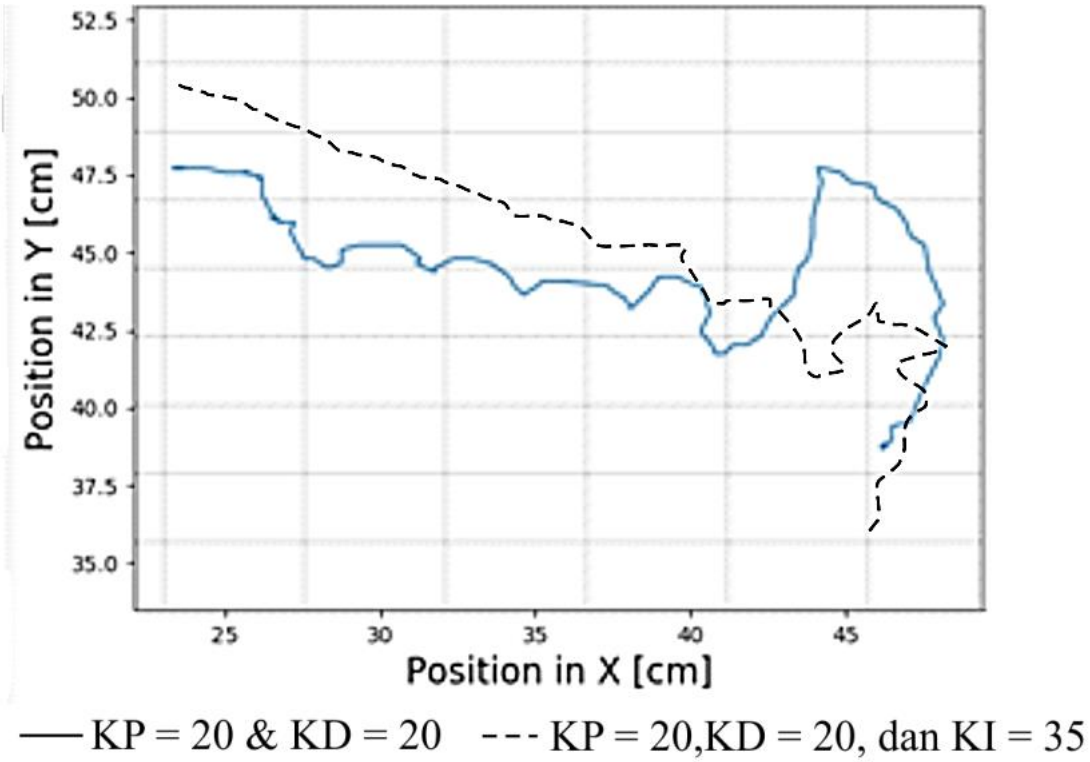
Pada Gambar 9. dapat dipahami bahwa kontrol *derivative* yang diberikan mampu mempercepat sistem bergerak dengan stabil lebih cepat karena menghindari terjadinya osilasi sedini mungkin. Pemberian nilai Kd harus tepat yaitu nilai yang diberikan mampu untuk mengakhiri osilasi secepat mungkin tanpa menghambat nilai keluaran mencapai nilai *set point*.

3.3. Pengujian Kontrol Proporsional-Integral-Derivatif (PID)

Pengujian ini dilakukan dengan nilai Kp = 20 dan Kd = 20 sedangkan Ki yang diberikan bervariasi yaitu dengan nilai 5 dan 35. Pengaruh pemberian nilai Ki maka dapat dilihat Gambar 10 dan Gambar 11

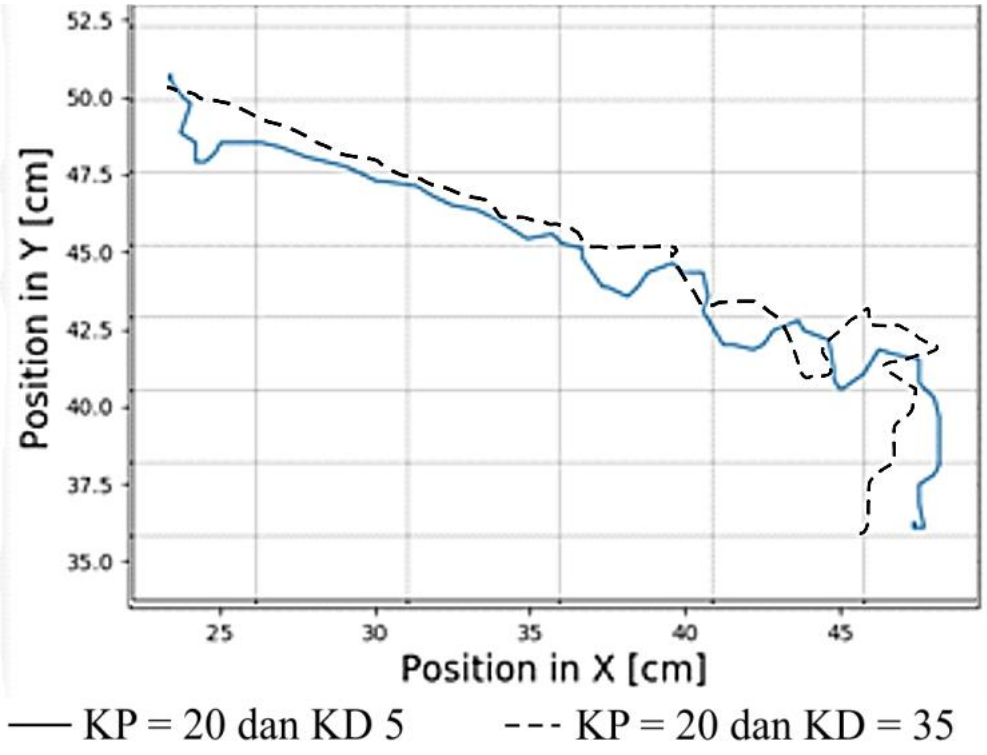


Gambar 10. Perbandingan ploting KI bernilai 5



Gambar 11. Perbandingan ploting KI bernilai 35

Pada Gambar 10 dan Gambar 11 dapat dipahami bahwasannya nilai Ki yang diberikan menyebabkan nilai keluaran mengalami penaikan yang dengan kata lain seperti mengoreksi nilai *set point*nya. Ketika sistem sudah berkeja hingga stabil namun belum juga mencapai *set point*, di saat itulah terjadi kesalahan karena sistem kontrol mengira sudah bekerja stabil pada *set point* yang ditentukan. Dalam pengoreksianya maka diberikan nilai Ki. Pengaruh besar dan kecilnya nilai Ki dapat dilihat Gambar 12.

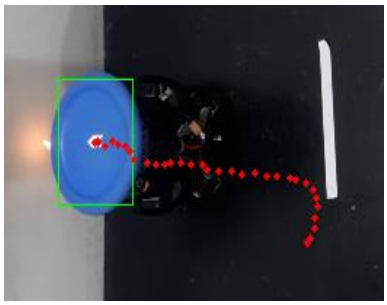
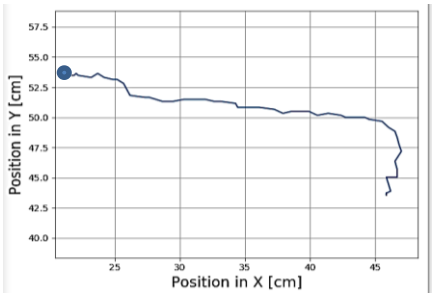
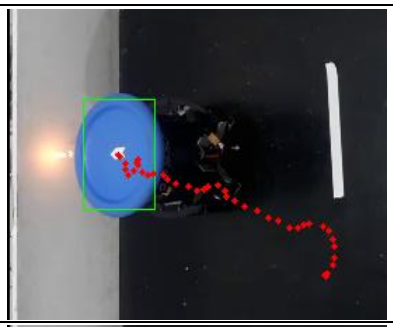
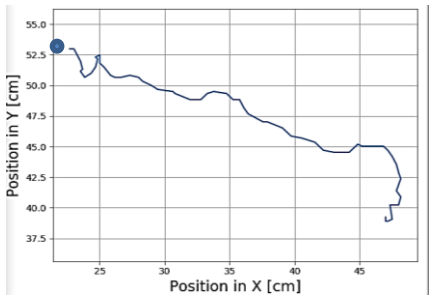
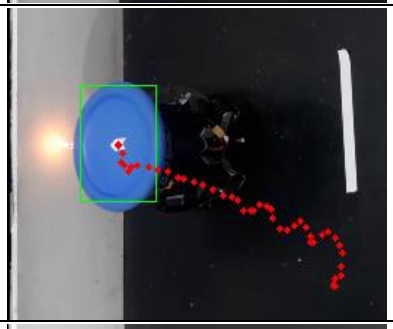
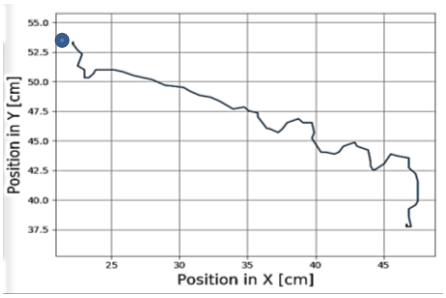
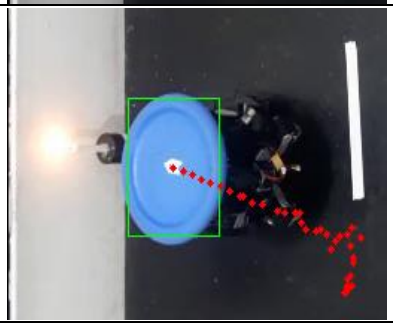
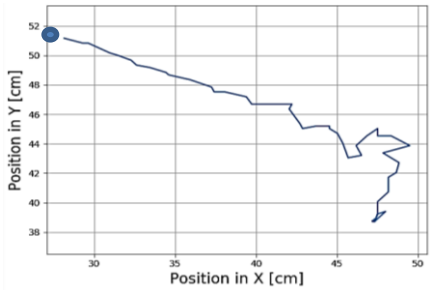


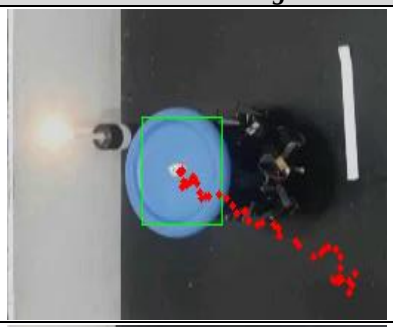
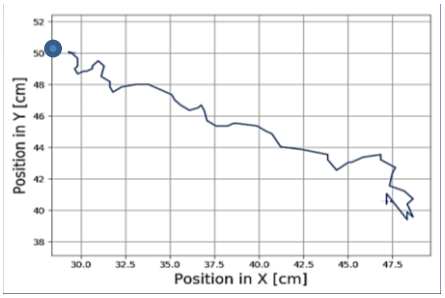
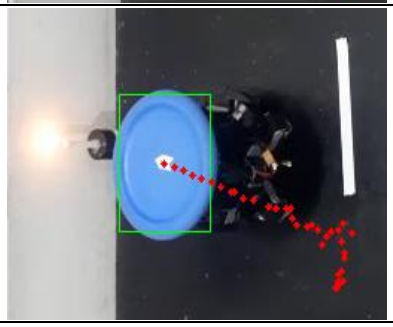
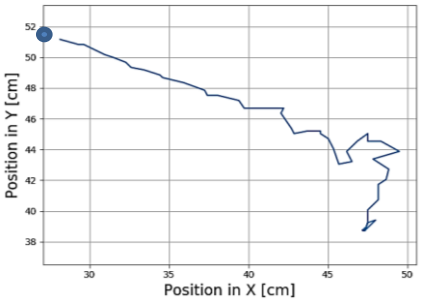
Gambar 12. Perbandingan Plot KD = 5 dan KD = 35

Pada Gambar 12 dapat dipahami bahwa kontrol integral yang diberikan mampu mengoreksi nilai keluarannya mengalami peningkatan untuk lebih mendekati atau melebihi nilai *set point* untuk keperluan pengoreksian.

Berdasarkan 6 pengujian yang telah dilakukan, telah menjelaskan pergerakan robot dalam bentuk *plot* beserta perbandingan dengan nilai konstanta Kp, Ki, dan Kd yang divariasikan untuk mengetahui pengaruh dari masing-masing kontrol. Selanjutnya akan dipaparkan waktu yang dibutuhkan robot dalam menghampiri api sampai berhasil beserta data keberhasilan robot. Berikut adalah hasil keseluruhan pengujian dengan *tracking* visi dan *plot* yang dihasilkan tiap pengujian dapat dilihat Tabel 2.

Tabel 2. Hasil Pengujian Keseluruhan

Konfigurasi	Grafis *Tracking*	Plot
KP=5 KI=0 KD=0 Derajat=-90		
KP=35 KI=0 KD=0 Derajat=-90		
KP=20 KI=0 KD=5 Derajat=-90		
KP=20 KI=0 KD=35 Derajat=-90		

Konfigurasi	Grafis *Tracking*	Plot
KP=20 KI=5 KD=20 Derajat=-90		
KP=20 KI=35 KD=20 Derajat=-90		

Berikut adalah hasil pengujian yang memaparkan waktu dan keberhasilan robot pada Tabel 3

Tabel 3. Waktu dan keberhasilan kontrol PID

No	Konfigurasi	Waktu (s)	Pengujian
1	Kp=5 Ki=0 Kd = 0 Derajat=-90	5	Berhasil
2	Kp=35 Ki=0 Kd = 0 Derajat=-90	5	Berhasil
3	Kp=20 Ki=0 Kd = 5 Derajat=-90	7	Berhasil
4	Kp=20 Ki=0 Kd = 35 Derajat=-90	5	Berhasil
5	Kp=20 Ki=5 Kd = 20 Derajat=-90	6	Berhasil
6	Kp=20 Ki=35 Kd = 20 Derajat=-90	5	Berhasil
Rata-rata waktu (s)			5,5
Persentase Keberhasilan			100%

4. KESIMPULAN

Penerapan PID *light following* berbasis sensor *flame array* mampu memberikan robot kemampuan untuk mengetahui titik api, menghampiri api hingga tepat di depan robot sejajar dengan sumber api atau lilin, dan berhasil berhenti sehingga tidak menabrak sumber api. Hasil penelitian ini menunjukan bahwa kontrol *proportional* berperan agar robot lebih cepat mengarah ke api. Kontrol *derivative* berperan dalam meredam terjadinya osilasi yang di sebabkan kontrol proportional dalam mengejar titik api. Sedangkan kontrol integral berperan dalam merevisi kekeliruan robot pada *set point*. Hasil pengujian pada penelitian ini memperoleh keberhasilan dengan persentase 100% dan rata-rata waktu yang dibutuhkan adalah 5,5 detik. Dengan nilai parameter PID terbaik adalah Kp=35, Ki=20, dan Kd=20.

UCAPAN TERIMA KASIH

Ucapan terimakasih disampaikan kepada *reviewer* dan *editor* yang telah membantu jurnal penelitian dalam bentuk tulisan ini untuk dipublikasikan sebagai refrensi keilmuan. Ucapan terimakasih juga disampaikan kepada pihak yang telah memberi dukungan materil maupun moral sehingga penelitian ini berhasil diselesaikan terutama kepada bapak Nuryono S.W. S.T., M.Eng. selaku dosen pembimbing pada penitian ini. Terimakasih

juga kepada Universitas Ahmad Dahlan yang telah mendukung kegiatan berkarya dari penulis. Selain itu terimakasih kepada Tim Robot UAD khususnya divisi KRPAI Berkaki.

REFERENSI

[1] N. I. Juang, W. Djuriatno, and M. Rif'an, "Desain dan Implementasi GRID-Based Map sebagai Sistem Pengenalan Posisi pada Kontes Robot Pemadam Api Indonesia (KRPAI) Divisi Beroda," *Journal Mahasiswa TEUB*, vol. 1, no. 5, pp. 1–6, 2013. Online

[2] S. Komputer and A. Siswanto, "Analisis Algoritma Untuk Mengidentifikasi Ruangan Pada Map Kontes Robot Pemadam Api Indonesia Menggunakan Logika Fuzzy," *Jurnal Processor* vol. 14, no. 1, pp. 1–13, 2019. DOI: 10.33998/processor.2019.14.1.557

[3] P. Gunoto, "Pencarian Rute Terpendek Arena Kontes Robot Pemada Api Indonesia (KRPAI) Menggunakan Aalgoritma Hill Climbing," *Jurnal Dimensi*, vol. 2, no. 1, pp. 1–8, 2013. DOI: 10.33373/dms.v2i1.109

[4] M. Ramdani, Sahrudin, S. Atisina, E. Ramdan, O. Heriyani, and H. Ramza, "Pengembangan Robot Pemadam Api Berkaki Uroita-18," *Jurnal Kajian Teknik Elektro*, vol. 4, no. 1, pp. 83–94, 2019. Online

[5] Alfith, "Perancangan Robot Cerdas Pemadam Api Dengan Sensor Thermal Array Tpa 81 Berbasis Microcontroller Arduino Mega 2560," *Jurnal Teknik Elektro ITP*, vol. 5, no. 2, pp. 95–102, 2016. Online

[6] D. Santoso, D. Susilo, and J. Wasesa, "Pengembangan Robot Berkaki Enam yang dapat Mengidentifikasi Ruang pada Map Kontes Robot Pemadam Api Indonesia menggunakan Algoritma Pengenalan Karakter Ruang," *Techné : Jurnal Ilmiah Elektroteknika*, vol. 16, no. 01, pp. 11–23, 2017. DOI: 10.31358/techne.v16i01.155

[7] S. B. Suharto, Purwanto, and G. D. Nusantoro, "Sistem Navigasi Wall Following Robot KRPAI Divisi Berkaki Menggunakan Kontroler PID," *Jurnal Mahasiswa TEUB*, vol. 1, no. 1, pp. 1–6, 2014. Online

[8] Darwison and R. Wahyudi, "Kontrol Kecepatan Robot Hexapod Pemadam Api menggunakan Metoda Logika Fuzzy," *Jurnal Nasional Teknik Elektro*, vol. 4, no. 2, pp. 227–234, 2015. DOI: 10.25077/jnte.v4n2.170.2015

[9] A. Maarif, S. Iskandar, and I. Iswanto, "New Design of Line Maze Solving Robot with Speed Controller and Short Path Finder Algorithm," *International Review of Automatic Control (IREACO)*, vol. 12, no. 3, p. 154, 2019. DOI: 10.15866/ireaco.v12i3.16501

[10] A. Maarif. "Perancangan Line Maze Solving Robot Dengan Algoritma Short Path Finder," *Skripsi Universitas Islam Indonesia*, 2014. Online

BIOGRAFI PENULIS

Muhammad Dhia Dzulfiqar adalah Mahasiswa Program Studi Teknik Elektro, Fakultas Teknologi Industri, Univesitas Ahmad Dahlan, Yogyakarta yang telah menyelesaikan pendidikan sarjana pada Program Studi tersebut.

Nuryono Satya Widodo adalah dosen Program Studi Teknik Elektro, Fakultas Teknologi Industri, Univesitas Ahmad Dahlan, Yogyakarta.