Implementasi *Fuzzy Logic* dengan *Library OpenCV* pada Robot *E-puck* sebagai *Line Follower* dalam Simulasi Jalan Raya

Alika Dwi Rahmadania[1]; Farhan Prasetio[2]; Hengki Wildan Noviana[3]; Ardy Seto Priambodo, S.T., M.Eng.[4]

Prodi D4-Teknik Elektronika, Fakultas Vokasi, Universitas Negeri Yogyakarta, Jl. Mandung, Serut, Pengasih, Kec. Wates, Kabupaten Kulon Progo, Daerah Istimewa Yogyakarta 55651, Indonesia

*E-mail: ardyseto@uny.ac.id

ABSTRACT

Robot line following is one of the developing types of robots in the field of technology, where a simple robot system is able to follow a line. It consists of a series of electronic components to achieve optimal response and speed. A line following robot is equipped with sensors that affect the wheel rotation, and the speed control motor is influenced by the speed limits and friction between the robot's wheels and the ground. The robot line following system comprises both software and hardware components. However, when the robot needs to follow a specific path, such as in a highway simulation, an automatic control system is required to ensure accurate and stable movement. In this research, we developed a fuzzy logic controller using the OpenCV library to improve the motion capabilities of the E-Puck robot as a line follower. Fuzzy logic technology proves to handle environmental variations and movement speed better than other methods. Through experimental testing, we obtained significant improvements in the line following movement by implementing fuzzy logic with the OpenCV library. The results showed an accuracy and stability of over 90% in robot movement. This research contributes significantly to researchers and operators in the field of automatic control system development, particularly in the robotics domain, especially in highway simulation. Furthermore, fuzzy logic technology can be further developed for application in other automatic control systems or even real-time applications. The programming language used in this research is Python, with motor access utilizing a fuzzy logic

controller. Through this research, we concluded that implementing a fuzzy logic controller using the OpenCV library is a suitable choice for enhancing the performance of line following robots like the E-Puck, enabling them to move accurately and steadily along a specific path.
Keywords: *line following robot, fuzzy logic, OpenCV library, E-Puck, highway simulation.*

ABSTRAK

Robot *line following* merupakan salah satu jenis robot yang sedang berkembang di dunia teknologi melalui sistem robot sederhana dapat mengikuti garis yang terdiri dari serangkaian komponen elektronik untuk mencapai respons dan kecepatan yang ideal. Robot pengikut linier dilengkapi dengan sensor yang mempengaruhi roda yang memutar motor pengaturan kecepatan dipengaruhi oleh batas kecepatan dan gesekan antara roda robot dengan tanah. Robot *line following* terdiri dari perangkat lunak dan perangkat keras. Namun, ketika robot ini harus mengikuti jalur tertentu, seperti simulasi jalan raya, diperlukan sistem kontrol otomatis untuk bergerak secara akurat dan stabil. Pada penelitian ini kami mengembangkan metode kontroler logika fuzzy dengan menggunakan *library OpenCV* untuk meningkatkan kemajuan untuk gerak robot *E-puck* sebagai pengikut garis. Teknologi logika fuzzy dapat mengelola variasi lingkungan dan kecepatan gerakan lebih baik daripada metode lainnya. Pada pengujian penelitian yang kami lakukan, kami mendapatkan hasil bahwa pengenalan teknologi logika fuzzy dengan *library OpenCV* berhasil meningkatkan kemajuan pergerakan robot line following secara signifikan. Ini terlihat lebih dari 90% akurasinya dan stabilitasnya dengan melihat gerakan robot. Penelitian ini memberikan kontribusi penting bagi para peneliti dan operator dalam pengembangan sistem kontrol otomatis dan penerapannya pada sistem robotik khususnya dalam simulasi jalan raya. Selain itu, teknologi logika fuzzy juga dapat dikembangkan lebih lanjut untuk diterapkan pada sistem kontrol otomatis lainnya atau bahkan aplikasi real-time. Bahasa program yang kami gunakan yaitu bahasa Phyton dengan akses motor yang menggunakan fuzzy logic controller Melalui penelitian ini, hasil yang kami dapat adalah metode kontroler logika fuzzy menggunakan *library OpenCV* merupakan pilihan yang tepat untuk meningkatkan performa gerakan robot mengikuti garis seperti *E-Puck*, memungkinkan mereka untuk bergerak di sepanjang jalur tertentu secara tepat dan stabil.
Kata Kunci: robot *line follower*, logika fuzzy, *library OpenCV, E-puck*, simulasi jalan raya.

1. Pendahuluan

Teknologi robot telah memainkan peran penting dalam sejarah perkembangan dunia. Robot dapat dijelaskan sebagai alat mekanik yang mampu melakukan tugas manusia dengan menggunakan sistem kecerdasan buatan yang diimplementasikan dalam bentuk perangkat keras dan perangkat lunak. Dalam perkembangannya, robot banyak digunakan di industri untuk meningkatkan efisiensi, efektifitas dan kecepatan dalam berbagai proses produksi. Awalnya, pekerjaan pengembangan robot dilakukan oleh manusia, yang tujuannya untuk mendukung pekerjaan dengan presisi tinggi, berisiko tinggi, dan intensif. Salah satu kemajuan teknologi yang paling luas dalam pengembangan robot adalah robot *line follower*. Robot *line follower* adalah jenis robot yang menggunakan metode deteksi garis dengan mengukur intensitas cahaya yang dipantulkan dari suatu permukaan yang melewatinya. Penggunaan robot pengikut garis menjadi semakin umum di berbagai industri di seluruh dunia karena kemampuannya untuk melakukan tugas secara efisien. [1]

Penerapan robot *line follower* dalam simulasi lingkungan jalan raya memberikan keuntungan dalam melakukan pengujian yang lebih terstruktur dan pengembangan algoritma kontrol yang lebih presisi sebelum diterapkan dalam dunia nyata. Dengan perkembangan dalam bidang kecerdasan buatan dan *Fuzzy Logic* kita dapat menggunakan strategi yang lebih canggih, metode yang populer adalah menggunakan *Fuzzy Logic* untuk memeriksa dan mengenali pola pada gambar yang diambil oleh kamera robot. Dalam konteks ini, kami mengimplementasikan *neural network* menggunakan *OpenCV* di robot *E-Puck* untuk melakukan *line following* di simulasi jalan raya.

Tujuan dari penelitian ini adalah mengimplementasikan algoritma *Fuzzy Logic* menggunakan library *OpenCV* dan *Tensorflow* pada robot *E-puck* sebagai *line follower* pada simulasi jalan raya. Dibangun sebuah model yang dapat mengolah data citra dari kamera robot *e-puck* dan menggunakan algoritma *Fuzzy Logic* untuk mengolah data tersebut sehingga robot dapat mengikuti jalur yang

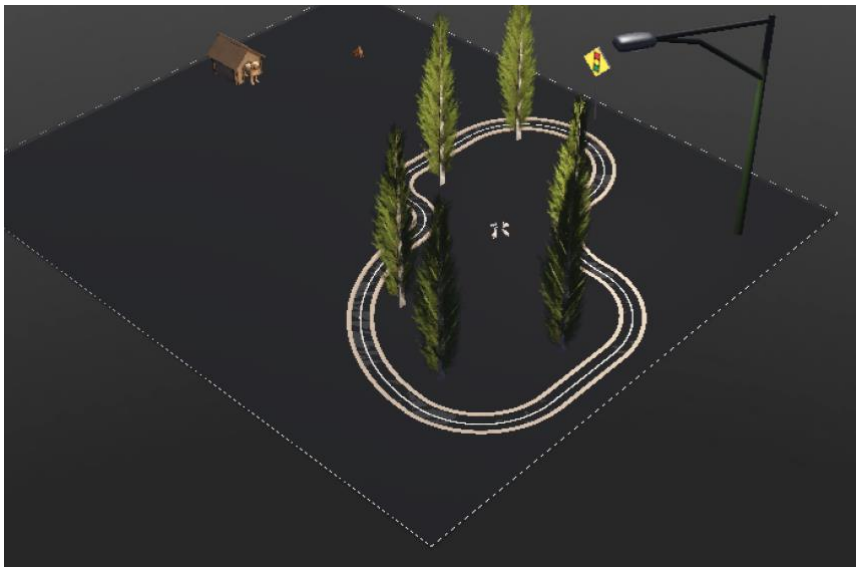
telah ditentukan.Proses implementasi menggunakan bahasa pemrograman Python dan *library OpenCV* dan *Tensorflow* untuk membuat model logika yang tidak pasti. Selain itu, beberapa percobaan dilakukan untuk mengetahui kinerja model yang dibangun. Hasil penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan kendaraan otonom terutama sejalan dengan penggunaan sensor *vision* seperti kamera dan penggunaan teknologi pembelajaran mesin melalui *Fuzzy Logic*. [5]

2. KAJIAN PUSTAKA

2.1 Webots

Webots merupakan sebuah aplikasi desktop yang sumber terbuka dan dapat digunakan di berbagai platform yang berfungsi untuk mensimulasikan robot. Aplikasi ini menyediakan lingkungan pengembangan yang lengkap untuk memodelkan, memprogram, dan melakukan simulasi terhadap robot. Webots dirancang untuk digunakan secara profesional dan telah banyak digunakan di industri, pendidikan, dan penelitian. Cyberbotics Ltd. secara konsisten menjaga dan mengembangkan Webots sebagai produk utama mereka sejak tahun 1998. [2]

Webots menyediakan koleksi besar model robot, sensor, aktuator, dan objek yang dapat diubah sesuai kebutuhan. Selain itu, pengguna juga dapat membuat model baru dari awal atau mengimpor model dari perangkat lunak CAD 3D. Saat merancang model robot, pengguna menentukan properti grafis dan fisik dari objek tersebut. Properti grafis meliputi bentuk, dimensi, posisi, orientasi, warna, dan tekstur objek. Sedangkan properti fisik melibatkan massa, faktor gesekan, serta konstanta pegas dan redaman. Webots juga memiliki kemampuan sederhana dalam simulasi dinamika fluida.Webots menggunakan versi modifikasi dari ODE (Open Dynamics Engine) untuk mendeteksi tumbukan dan mensimulasikan dinamika benda yang kaku. Pustaka ODE memungkinkan simulasi yang akurat terhadap properti fisik objek, seperti kecepatan, inersia, dan gesekan. Webots menyertakan kumpulan sensor dan aktuator yang sering digunakan dalam eksperimen robotik, seperti lidar, radar, sensor proximity, sensor cahaya, sensor sentuhan, GPS, akselerometer, kamera, pemancar dan penerima, motor servo (rotasi & linear), sensor posisi dan gaya, LED, gripper, giroskop, kompas, IMU, dan lain-lain. Program pengendali robot dapat ditulis di luar Webots menggunakan bahasa pemrograman C, C++, Python, ROS, Java, dan MATLAB dengan menggunakan antarmuka pemrograman aplikasi (API) yang sederhana. [3]



Gambar 1 Tampilan awal project dari webots

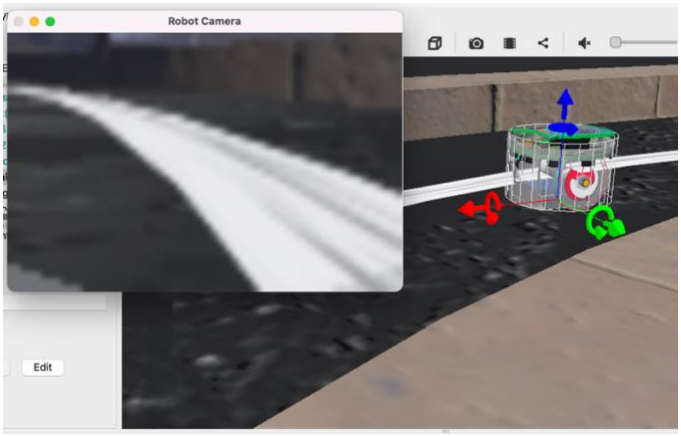
2.2 *E-puck*

E-puck adalah robot seluler kecil yang awalnya dikembangkan di EPFL untuk tujuan pendidikan oleh pembuat robot Khepera yang sukses. Perangkat keras dan perangkat lunak *E-puck* benar-benar *open source*, yang berarti bahwa semua komponen elektronik tersedia secara mendetail dan menawarkan peluang tak terbatas untuk pengembangan lebih lanjut. Model ini mendukung motor diferensial (encoder juga disimulasikan sebagai sensor posisi), sensor inframerah untuk pengukuran jarak dan cahaya, akselerometer, girometer, kamera, 8 LED di sekeliling, LED di bodi dan di depan, koneksi Bluetooth dan ekstensi sensor lantai. Namun, perangkat elektronik lainnya tidak disimulasikan dalam model ini. Untuk informasi terbaru tentang robot ini, kunjungi situs web resmi *E-puck*. [4]

Gambar 2 Tampilan pada robot *E-puck*

2.3 *OpenCv*

OpenCV atau *Open Source Computer Vision Library* adalah *library open source* untuk pemrosesan gambar dan video. *Library* ini ditulis dengan bahasa pemrograman C++ dan dapat digunakan di beberapa platform seperti Windows, Linux dan MacOS. *OpenCV* juga mendukung beberapa bahasa pemrograman lain seperti Python, Java dan MATLAB. *OpenCV* dikembangkan oleh Intel pada tahun 1999 sebagai bagian dari proyek penelitian untuk meningkatkan kemampuan komputer dalam mengenali gerakan manusia. Sejak saat itu, *OpenCV* telah berkembang menjadi pustaka komprehensif dengan lebih dari 2.500 fungsi, yang mencakup banyak bidang pemrosesan gambar, seperti segmentasi gambar, deteksi wajah dan objek, kalibrasi kamera, dan pembelajaran mesin. Penggunaan umum *OpenCV* termasuk deteksi gerakan, pengenalan objek, *augmented reality (AR)*, robotika, dan kendaraan otonom.*OpenCV* sangat populer di kalangan peneliti dan pengguna visi komputer karena melakukan operasi pemrosesan gambar dengan andal, cepat, dan mudah digunakan. Selain itu, sifat *open source* memungkinkan pengguna untuk mengakses *source code* sehingga mudah untuk melakukan perubahan sesuai dengan kebutuhan masing-masing. [6]



Gambar 3 Tampilan kamera *opencv* pada webots

2.4 *Fuzzy Logic*

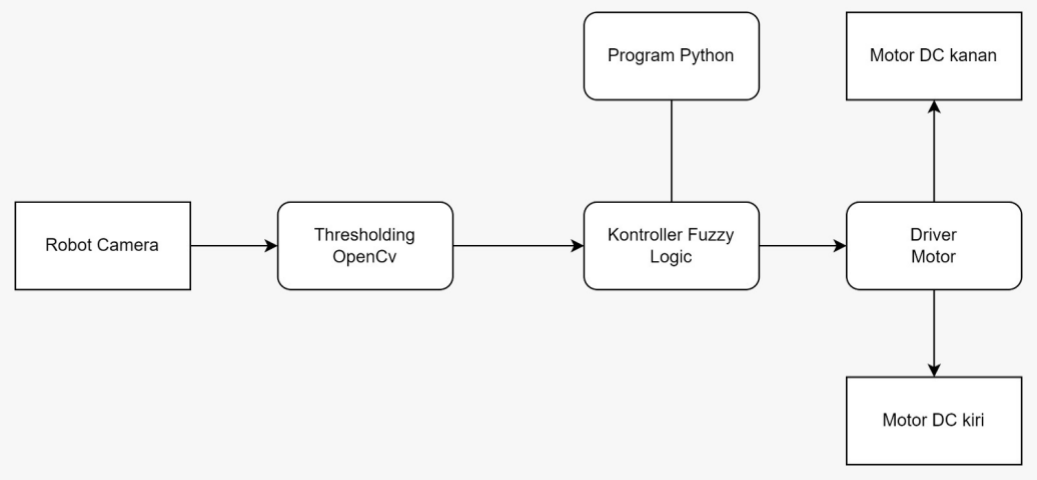
Tentang Fuzzy Logic, menurut Jannah dan Arifin (2018), fuzzy logic adalah suatu metode yang dapat digunakan untuk mengatasi ketidakpastian pada informasi dan data yang tidak lengkap. Teknologi ini memungkinkan sebuah sistem untuk menangani masalah-masalah non-linear dengan lebih baik daripada teknologi kontroler lainnya. [7] Dalam aplikasinya pada robotika, fuzzy logic sering digunakan sebagai sistem kendali otomatis untuk membantu robot navigasi di tengah lingkungan yang tidak terstruktur seperti misalnya dalam simulasi jalan raya (Nugraha et al., 2020). [8] Selain itu, beberapa penelitian juga mencoba menggabungkan fuzzy logic dengan teknologi lain seperti neural network atau genetic algorithm untuk meningkatkan performa pergerakan robot secara signifikan.Menurut Sari et al. (2019) prinsip kerja dari fuzzy logic adalah dengan menggunakan aturan-aturan logika kabur seperti "jika A maka B" atau "jika A dan B maka C". Aturan tersebut kemudian diaplikasikan pada variabel input sehingga menghasilkan output sesuai kondisi saat itu. [9]

Penggunaan teknologi Fuzzy Logic dalam pengembangan sistem kendali otomatis telah banyak dilakukan dan terbukti efektif dalam menangani masalah kompleks serta memberikan hasil akurat. Dalam aplikasinya pada Robot Line Following, teknologi Fuzzy Logic mampu meningkatkan kemampuan navigasi serta kontrol gerakan yang dapat membantu Robot bekerja dengan lebih akurat dan stabil.

3. Metode Penelitian

3.1 Perancangan Sistem

Pada *project* ini kami merancang robot berbasis *line follower* dengan Gerakan mengikuti jalur yang telah di sediakan. Robot *e-puck* ini bekerja dengan berada di tengah-tengah jalan raya, lebih tepatnya berjalan dengan mengikuti garis marka jalan. Dalam proses pendeteksian, garis bantu akan ditambahkan untuk memastikan gerakan robot yang stabil. Untuk mengontrol pergerakan robotnya, digunakan *Fuzzy Logic Controller* dengan program bahasa Python dan motor driver serta motor DC sebagai penggerak jalannya. Dalam operasinya nanti, robot diprogram untuk bergerak mengikuti garis yang telah disediakan menggunakan akses motor melalui sistem *fuzzy.* Diagram blok perancangan sistem dapat digunakan untuk memvisualisasikan bagaimana operasi robot tersebut berjalan.



Gambar 4 Diagram blok perancangan sistem

Keterangan:

1. *Start* Program: Titik awal program.
2. *Thresholding*: Pada blok ini, gambar pra-pemrosesan akan diubah menjadi citra biner, di mana hanya dua nilai piksel yang diperbolehkan

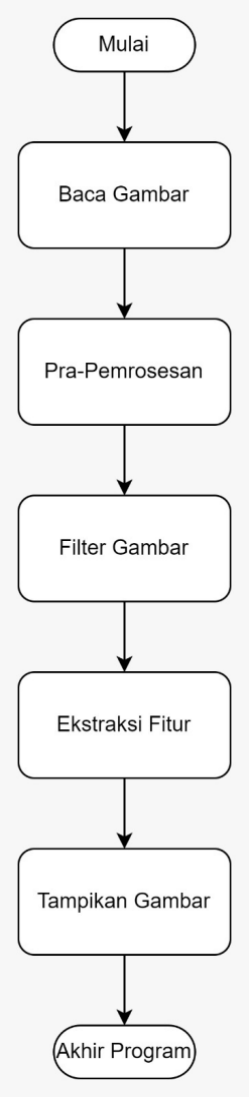
(hitam dan putih). Ini membantu dalam memisahkan objek garis dari latar belakang.

3. *Driver* Motor : kecepatan motor yang dikendalikan oleh hasil proses *controll fuzzy*
4. *Motor Driver*: Blok ini mewakili motor penggerak robot. Sinyal kendali dari *Fuzzy Controller* akan dikirim ke motor untuk mengatur kecepatan dan arah gerakan robot agar mengikuti garis dengan tepat.

Dengan itu robot akan bergerak mengikuti garis putih atau garis marka jalan karena sudah di beri program dan njuga dengan adanya *control fuzzy logic*.

3.2 Perancangan Pembacaan *OpenCv*

OpenCV Reading Design merupakan bagian dari pengembangan sistem robot berbasis *vision* dimana kamera digunakan sebagai sensor utama dalam proses pengumpulan data dan kontroler robot diprogram dengan algoritma *computer vision* yang dibangun menggunakan *library OpenCV*. Proses pembacaan dilakukan dengan kamera atau video dengan kamera yang terpasang pada robot. Kemudian gambar tersebut diolah oleh *software OpenCV* untuk mendapatkan informasi penting seperti pendeteksian objek tertentu, *motion tracking* dan gambar lainnya. Diagram blok perancangan pembacaan *OpenCv* dapat digunakan untuk memvisualisasikan bagaimana operasi robot tersebut berjalan dengan *OpenCv*.



Gambar 5 Digram blok pembacaan *OpenCv*

Keterangan:

1. Mulai Program: Titik awal program.
2. Baca Gambar: Membaca gambar dari sumber yang ditentukan (misalnya file gambar).
3. Pra-Pemrosesan: Pra-pemrosesan gambar dilakukan di sini, seperti mengubah ke skala abu-abu, menyesuaikan kontras, atau menghilangkan *noise* yang tidak diinginkan.

4. Filter Gambar: Filter dapat diterapkan pada gambar untuk menghilangkan noise atau meningkatkan fitur tertentu, seperti filter median, filter Gaussian, atau filter lainnya.
5. Ekstraksi Fitur: Ekstraksi fitur dilakukan untuk mengidentifikasi atau menyoroti fitur penting dalam gambar. Ini dapat mencakup deteksi tepi, deteksi sudut, pengenalan pola, atau algoritma lain yang digunakan untuk mengekstraksi informasi penting dari gambar.
6. Tampilkan Gambar: Gambar hasil dengan fitur yang ditemukan ditampilkan untuk visualisasi atau analisis lebih lanjut.
7. Akhir Program: Titik akhir program.

Ada beberapa hal yang perlu diperhatikan dalam mendesain layar *OpenCV* ini, antara lain konfigurasi kamera dan pengaturan parameter pada *software OpenCV* untuk memberikan hasil dan kebutuhan aplikasi yang akurat. Hal ini sangat penting karena kesalahan dalam pembacaan dapat mempengaruhi kinerja robot secara keseluruhan.

4. Hasil dan Pembahasan

4.1 Hasil pengujian

Gambar 6 Hasil pengujian menggunakan rule 1

*Rule*1 = ctrl.*Rule*(jarak['negatif'], (kecepatan_kiri['maju_cepat'], kecepatan_kanan['berhenti']))Aturan ini berlaku ketika jarak memiliki nilai negatif. Fungsi keanggotaan untuk jarak negatif telah ditentukan sebelumnya. Aturan ini

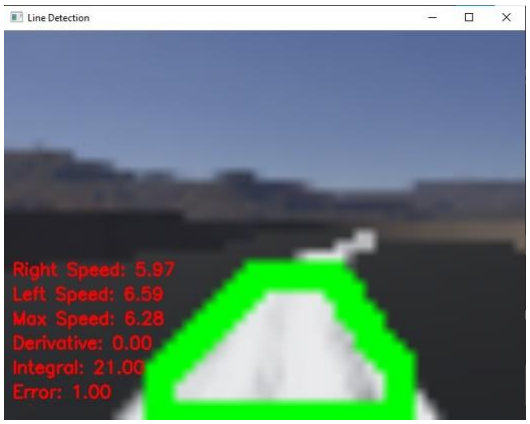
menghasilkan keluaran fuzzy dimana kecepatan roda kiri fast_forward dan kecepatan roda kanan berhenti.

Gambar 7 Hasil pengujian menggunakan rule 2

Rule2 = ctrl.Rule(jarak['nol'], (kecepatan_kiri['maju_lambat'], kecepatan_kanan['maju_lambat'])) Aturan ini berlaku ketika nilai jarak adalah nol. Fungsi anggota jarak nol juga sudah ditentukan sebelumnya. Aturan ini menghasilkan keluaran fuzzy untuk kecepatan roda kiri dan kanan sama dengan *"slow_forward"*.

Gambar 7 Hasil pengujian menggunakan rule 3

Rule3 = ctrl.Rule(jarak['positif'], (kecepatan_kiri['berhenti'], kecepatan_kanan['maju_cepat'])) Aturan ini berlaku ketika jarak memiliki nilai positif. Fungsi keanggotaan untuk jarak positif juga diberikan. Aturan ini menghasilkan keluaran fuzzy dengan kecepatan roda kiri sebagai *"Stop"* dan kecepatan roda kanan sebagai *"fast_forward"*.



Gambar 6 Hasil pengujian

Setelah dilakukan pengujian teknologi fuzzy logic pada robot e-puck, didapatkan hasil bahwa robot bekerja lebih lancar dan terdapat perbedaan yang jelas antara motor kiri dan kanan saat berputar. Nilai error yang dihasilkan juga lebih rendah dari sebelumnya, hal ini terlihat dari nilai pada layar OpenCV, dimana nilai error hanya 1.00 yang artinya hampir tidak ada error atau bahkan smooth yang optimal tercapai. Oleh karena itu, pengenalan teknologi logika fuzzy pada Robot E-Puck sebagai line follower dalam simulasi lintasan dapat secara efektif dan akurat meningkatkan kinerja robot dan mengurangi nilai kesalahan, sehingga pergerakan robot lebih stabil dan akurat dalam menjalankan tugasnya.

4.2 Pembahasan

Gambar 7 Hasil simulasi saat robot berada di sebelah kiri dan kanan

Pada sistem kendali robot menggunakan sensor garis, nilai *error* atau kesalahan gerak dihitung berdasarkan jarak antara posisi robot dengan posisi garis. Jika robot meninggalkan garis, nilai kesalahannya negatif. Sebaliknya, jika robot mendekati atau berada di sebelah kanan garis, maka nilai *error*nya positif.Untuk menghindari kesalahan pergerakan dan memastikan kestabilan gerak saat mengikuti rute yang telah ditentukan, *engine controller* biasanya menggunakan batas kecepatan maksimum dan minimum untuk setiap sisi sepeda motor. Hal ini dilakukan agar gerakan tetap seimbang dan terkontrol dengan baik serta tidak melenceng dari jalurnya. Kesimpulannya adalah pada sistem kendali robot berbasis vision dengan menggunakan sensor linier, kita dapat mengetahui dimana robot berada pada jalur yang ada dan mengatur kecepatan sepeda motor sesuai dengan kondisi saat ini untuk mencapai pergerakan yang stabil dan akurat sehingga dapat mengikuti jalur.

Tabel 1 Hasil simulasi

Error	Integral	Derivative	Max Speed	Left Speed	Right Speed
-9.00	-52.00	2.00	6.28	4.88	7.68
-9.00	-61.00	0.00	6.28	4.77	7.79
-7.00	-68.00	2.00	6.28	4.92	7.64
-6.00	-74.00	1.00	6.28	4.95	7.61
-5.00	-79.00	1.00	6.28	5.00	7.56

Tabel hasil menunjukkan hasil yang menunjukkan variabel relevan. *Error* adalah selisih antara posisi tengah garis yang terdeteksi dan posisi tengah layar (Lebar/2). Nilai error yang lebih besar menunjukkan posisi garis yang lebih jauh dari pusat

layar, sedangkan nilai kesalahan yang lebih kecil menunjukkan posisi garis yang lebih dekat dengan pusat layar. Komponen integral adalah jumlah dari semua kesalahan dalam setiap iterasi simulasi. Integral digunakan untuk mengatasi masalah kesalahan stabilitas pada kontrol PID. Jika ada perbedaan konstan antara posisi garis dan pusat layar, nilai integral bertambah secara kumulatif. Derivatif adalah selisih antara kesalahan iterasi saat ini dan kesalahan iterasi sebelumnya. Derivatif digunakan untuk memperkirakan perubahan kesalahan dari waktu ke waktu. Nilai turunan yang lebih tinggi berarti perubahan kesalahan yang cepat, sedangkan nilai turunan yang lebih rendah berarti perubahan kesalahan yang lambat. Kecepatan Maksimum adalah kecepatan maksimum yang dapat dicapai motor robot dan digunakan dalam simulasi sebagai batas atas kecepatan motor. Kecepatan Kiri dan Kecepatan Kanan adalah kecepatan yang ditetapkan untuk motor roda kiri dan kanan. Nilai-nilai ini dihitung menggunakan rumus kontrol PID, dengan mempertimbangkan komponen kesalahan, integral dan turunan. Kecepatan motor ini mempengaruhi pergerakan robot. Menganalisis nilai variabel tersebut memberikan pemahaman tentang cara kerja kontrol PID dalam simulasi robot dan bagaimana perubahan nilai variabel tersebut memengaruhi gerak robot.

5. Kesimpulan

Tugas *project* ini melibatkan penggunaan robot *e-puc*k untuk mengikuti garis secara otomatis. Tujuannya adalah untuk mengembangkan keterampilan pemrograman, pemahaman sistem kendali robotik, dan kemampuan pemecahan masalah. Dalam project ini, diharapkan dapat memahami tentang penggunaan sensor untuk mendeteksi garis dan algoritma kontrol yang memungkinkan robot mengikuti garis dengan akurasi. Tantangan yang mungkin dihadapi termasuk kalibrasi sensor, optimalisasi algoritma kontrol, dan mengatasi situasi garis yang putus atau berbelok tajam. Melalui proyek ini, penulis dapat meningkatkan pemahaman tentang robotika, sistem kendali, dan pemrograman. Keahlian yang

diperoleh juga dapat diterapkan dalam bidang robotika dan otomasi lainnya, terutama dalam pengembangan sistem navigasi otonom.

DAFTAR PUSTAKA

[1] A. Lorenza, B. Hidayat, L. Sidka Alia, L. Aufa Nurachanta, I. Halmahera, M.S.Irsan Syahputra. (2022, Jun). *"Simulation Understanding Line Follower Robot C Program with Webots Simulasi Pemahaman Robot Pengikut Garis Program C dengan Webots*". [On-line]. Vol. 2 Iss, hlm. 07-18. Tersedia di: https://media.neliti.com/media/publications/410723-simulation-understanding-line-follower-r-3db1dd0b.pdf [8 Jun. 2023].

[2] Webmaster. "Cyberbotics." Internet: https://cyberbotics.com/ [8 Jun. 2023]

[3] J. Wales. "Webots ." Internet: https://en.wikipedia.org/wiki/Webots 25 Des. 2022 [8 Jun. 2023]

[4] Webmaster. "Webots User Guide R2023 GCTronic' e-puck." Internet: https://cyberbotics.com/doc/guide/epuck [8 Jun. 2023]

[5] A.Taufiq Rizaldi, N.A.P.Widiastuti and P.D.Sari (2019). *"Implementation of Neural Network OpenCV on Robot E-Puck as Line Follower in Highway Simulation"*. [On-line]. Volume 13 - Special Issue on Emerging Trends in e-Learning and Innovative Learning Technologies (ICELT 2019), hlm 11-21.

[6] Admin. "OpenCV." Internet: https://opencv.org/ [15 Jun.2023]

[7] Jannah, M & Arifin,A.Z.(2018). *"Fuzzy Logic pada Sistem Kendali Otomatis"*. [On-line] Jurnal Teknik Elektro, 13(2), hlm. 98-103. [16 Jun. 2023]

[8] Nugraha, I.T., Rasyidin, M.E.A., & Wihandika R.S. (2020). "*Implementasi Fuzzy Logic untuk Mengontrol Gerakan Robot Line Follower Beroda Tiga dengan Sensor Infrared*". Jurnal Ilmiah Pendidikan dan Teknologi Informasi (JPTI), 7(1), hlm. 25-30. [16 Jun. 2023]

[9] Sari, R., Sudiro, T.H,. & Kurniawan Y.K.(2019). *"Implementasi Metode Fuzzy Logic untuk Deteksi Posisi Mobil Berdasarkan Citra Hasil Kamera CCTV di Jalan Raya Menggunakan Raspberry Pi"*. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 6(2), hlm.117-125. [16 Jun. 2023]