

Rancang Bangun Sistem *Adaptive Cruise Control* (ACC) dengan Metode *Fuzzy Logic* pada *Prototype Robo Car*

Design of Adaptive Cruise Control (ACC) System with Fuzzy Logic Method on Robo Car Prototype

¹Aulia Istiqomah*), ¹Luthfansyah Mohammad dan ¹Muhammad Irham Agil Kurniawan

¹Universitas Diponegoro, 50275, Semarang, Indonesia

*) *corresponding email: auliaistiqomah12@lecturer.undip.ac.id*

Abstrak

Pertumbuhan jumlah kendaraan roda empat di Indonesia yang pesat telah berkontribusi pada peningkatan angka kecelakaan lalu lintas, di mana kesalahan manusia menjadi faktor utama. *Adaptive Cruise Control* (ACC) merupakan teknologi yang telah banyak diterapkan dalam sistem bantuan pengemudi (*Advanced Driver Assistance Systems*/ADAS) untuk meningkatkan keselamatan berkendara. Penelitian ini mengembangkan sistem ACC berbasis *Fuzzy Logic Controller* (FLC) menggunakan ESP8266 sebagai mikrokontroler utama, dengan tujuan mengoptimalkan pengendalian kecepatan kendaraan secara adaptif berdasarkan jarak terhadap objek di depannya. Sistem ini dirancang untuk mengolah data jarak dan kecepatan secara real-time, sehingga dapat menyesuaikan kecepatan kendaraan tanpa perubahan yang tiba-tiba. Model *fuzzy logic* dikembangkan dan divalidasi melalui simulasi MATLAB sebelum diimplementasikan ke dalam perangkat keras. Hasil pengujian menunjukkan bahwa sistem ACC berbasis FLC mampu menjaga jarak aman dan mengatur kecepatan secara dinamis, dengan transisi kecepatan yang lebih halus dibandingkan metode konvensional. Perbandingan antara hasil simulasi MATLAB dan implementasi nyata menunjukkan bahwa sistem dapat beroperasi dengan akurasi tinggi meskipun terdapat perbedaan akibat faktor lingkungan. Dengan keberhasilan ini, penelitian ini membuktikan bahwa FLC dapat menjadi alternatif yang lebih fleksibel dalam sistem ACC dan memiliki potensi untuk diterapkan dalam pengembangan kendaraan listrik dan otonom di masa depan.

Kata Kunci: ESP8266; kontrol kecepatan adaptif; fuzzy logic; motor DC.

Abstract

The rapid growth of four-wheeled vehicles in Indonesia has contributed to an increase in traffic accidents, where human error is a significant factor. *Adaptive Cruise Control* (ACC) is a technology that has been widely implemented in *Advanced Driver Assistance Systems* (ADAS) to improve driving safety. This study develops a *Fuzzy Logic Controller* (FLC)-based ACC system using ESP8266 as the main microcontroller to optimize adaptive vehicle speed control based on the distance to the object in front. This system is designed to process distance and speed data in real-time, allowing it to adjust the vehicle's speed without sudden changes. A fuzzy logic model was developed and validated through MATLAB simulations before being implemented into hardware. Test results indicate that the FLC-based ACC system can maintain a safe distance and dynamically regulate speed, exhibiting smoother speed transitions compared to conventional methods. A comparison between MATLAB simulation results and real-world implementations shows that the system can operate with high accuracy, despite differences due to environmental factors. With this success, this research proves that FLC can be a more flexible alternative in ACC systems and has the potential to be applied in the development of electric and autonomous vehicles in the future.

Keywords: ESP8266; adaptive cruise control; fuzzy logic; DC motor.

Makalah diterima 14 Februari 2025 – makalah direvisi 24 Maret 2025– disetujui 18 April 2025

Karya ini adalah naskah akses terbuka dengan lisensi [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/).



1 Pendahuluan

Kendaraan roda empat telah menjadi alat transportasi yang esensial dalam kehidupan sehari-hari. Bahkan, sebagian orang menganggap mobil sebagai kebutuhan sekunder atau primer, terutama bagi mereka yang sering bepergian jauh [1]. Seiring perkembangan zaman, jenis kendaraan roda empat pun semakin beragam, seperti kendaraan berbahan bakar bensin, diesel, dan kini kendaraan listrik atau *Electric Vehicle* (EV) [2]. Selain itu, munculnya konsep kendaraan otonom (*autonomous car*) membawa perubahan signifikan dalam industri otomotif. Mobil otonom adalah kendaraan yang mampu mengemudi sendiri tanpa intervensi manusia, dengan menggunakan kombinasi sensor, kamera, radar, dan kecerdasan buatan (AI) untuk mengenali

lingkungan sekitarnya dan membuat keputusan berkendara [3]. Mobil otonom diklasifikasikan ke dalam beberapa tingkatan, mulai dari Level 0 (tanpa otomatisasi) hingga Level 5 (otomatisasi penuh), di mana Level 5 tidak memerlukan pengemudi sama sekali [4]. Teknologi ini diharapkan dapat mengurangi kecelakaan lalu lintas yang disebabkan oleh kesalahan manusia, meningkatkan efisiensi transportasi, dan mengurangi kemacetan [5]. Mobil otonom dapat mengurangi kecelakaan hingga 90% jika diimplementasikan secara luas [6].

Peningkatan jumlah kendaraan roda empat di Indonesia terus mengalami perkembangan yang signifikan. Jumlah mobil penumpang pada periode 2012-2022 meningkat hingga mencapai 6,74 juta unit, yang berkontribusi sekitar 65% dari total kendaraan yang ada di Indonesia [7]. Namun, seiring dengan peningkatan jumlah kendaraan, tingkat kecelakaan lalu lintas di Indonesia juga menunjukkan tren yang meningkat. Berdasarkan laporan dari Korlantas Polri, jumlah kecelakaan lalu lintas pada periode Januari hingga September 2022 tercatat sebanyak 94.617 kasus, yang merupakan peningkatan signifikan dibandingkan dengan tahun 2021 yang hanya mencatatkan 70.000 kasus, atau meningkat sebesar 34,65% [8]. Salah satu penyebab utama kecelakaan ini adalah faktor manusia, yang tercatat menyumbang hingga 61% dari total kecelakaan [8]. Angka ini menunjukkan pentingnya upaya untuk mengurangi kesalahan manusia dalam berkendara, terutama di jalan bebas hambatan atau tol. Pada jalan tol, yang merupakan jalan dengan potensi kecelakaan tinggi akibat kelalaian pengemudi, kesalahan manusia sering terjadi akibat menurunnya kewaspadaan pengemudi pada jalan yang lurus dan monoton [9]. Untuk itu, teknologi yang dapat meningkatkan keselamatan dan kenyamanan berkendara sangat diperlukan, salah satunya adalah sistem *Adaptive Cruise Control* (ACC). ACC adalah sistem yang dapat mengatur kecepatan kendaraan berdasarkan jarak kendaraan di depan sesuai dengan pengaturan yang ditentukan dalam program kontrol. Sistem ini bekerja dengan memanfaatkan teknologi sensor seperti radar, lidar, kamera, atau sensor ultrasonik untuk mendeteksi hambatan, kendaraan lain, dan objek di sekitarnya [10]. ACC membantu pengemudi untuk menjaga jarak aman dan mengatur kecepatan kendaraan secara otomatis, yang dapat mengurangi potensi kecelakaan akibat kesalahan manusia [11].

Adaptive Cruise Control (ACC) telah berkembang dengan berbagai metode kontrol. Metode *Proportional-Integral-Derivative* (PID) adalah salah satu pendekatan konvensional yang sering digunakan dalam sistem ACC karena kemampuannya dalam mengendalikan kecepatan berdasarkan parameter yang telah ditentukan. Namun, metode ini memiliki keterbatasan dalam menangani perubahan lingkungan yang dinamis, seperti kondisi lalu lintas yang tidak menentu [12]. *Model Predictive Control* (MPC) juga digunakan dalam ACC, di mana metode ini mengandalkan model matematis untuk memprediksi respons kendaraan dalam beberapa waktu ke depan. Meskipun lebih adaptif dibandingkan PID, metode ini membutuhkan daya komputasi yang tinggi dan tidak selalu optimal dalam kondisi lalu lintas yang kompleks [13]. Selain itu, *Machine Learning-Based* ACC juga mulai dikembangkan dengan menggunakan algoritma pembelajaran mesin untuk meningkatkan adaptivitas dalam pengendalian kendaraan, tetapi masih terbatas dalam implementasi nyata karena memerlukan data pelatihan yang sangat besar serta proses komputasi yang intensif [14].

Salah satu metode yang dapat digunakan untuk meningkatkan responsivitas ACC adalah *fuzzy logic*. *Fuzzy logic* adalah sistem kontrol yang didasarkan pada logika kabur, di mana keputusan diambil berdasarkan derajat keanggotaan tertentu, bukan hanya nilai biner (ya/tidak) [15]. Metode ini sangat cocok untuk sistem adaptif seperti ACC karena mampu menangani ketidakpastian dan variasi data sensor secara lebih efektif dibandingkan metode konvensional seperti PID [16]. *Fuzzy logic* memberikan respons yang lebih halus dan stabil dalam mengatur kecepatan kendaraan, terutama dalam situasi lalu lintas yang dinamis [17]. Penggunaan *cruise control* saat berkendara di jalan tol memberikan manfaat dalam meningkatkan keselamatan dan kenyamanan berkendara. Pengemudi yang menggunakan fitur ini cenderung lebih stabil dalam mengatur kecepatan kendaraan, yang berpotensi mengurangi angka kecelakaan [13].

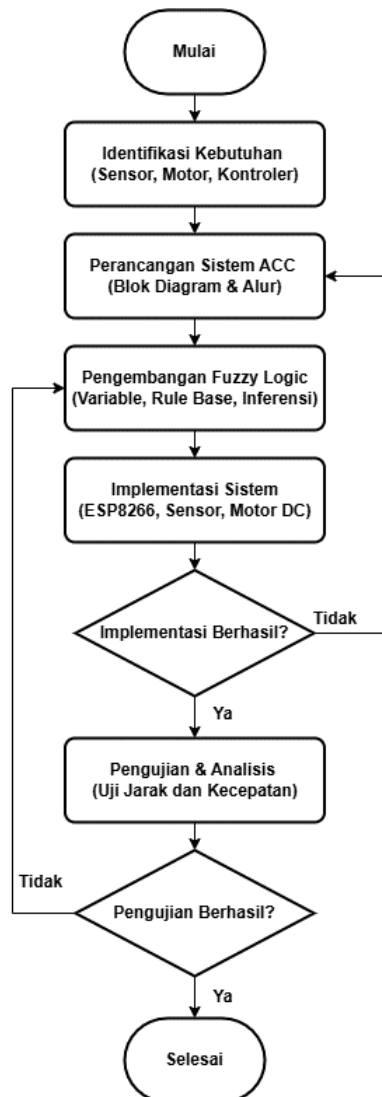
Dalam penelitian ini, sistem ACC berbasis *fuzzy logic* akan diimplementasikan pada prototipe *Robo Car 4WD* yang dikendalikan oleh NodeMCU ESP8266. Sistem ini akan menggunakan sensor ultrasonik sebagai alat deteksi jarak dan sensor LM393 sebagai alat pemantau kecepatan kendaraan. Dengan kombinasi ini, penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem ACC yang lebih adaptif dan efisien, serta dapat menjadi model pembelajaran bagi industri otomotif dan akademisi. Dengan demikian, penelitian ini menawarkan *novelty* berupa penerapan *fuzzy logic* berbasis ESP8266 untuk sistem ACC, yang diharapkan dapat meningkatkan efektivitas kontrol kecepatan kendaraan dengan cara yang lebih adaptif dan responsif terhadap kondisi lingkungan yang berubah-ubah.

2 Metode

Dalam penelitian ini, sebuah prototipe sistem *Adaptive Cruise Control* (ACC) dirancang dengan menggunakan metode *fuzzy logic*, dengan mengintegrasikan komponen-komponen utama seperti sensor ultrasonik, motor driver, dan mikrokontroler ESP8266.

2.1 Perancangan Sistem

Dari perancangan sistem ACC pada prototipe alat, terdapat sebuah *flowchart* atau diagram alir dari konsep kerja sistem ACC yang dipasangkan pada flowchart ini.



Gambar 1. Diagram Alir Perancangan Sistem *Adaptive Cruise Control*

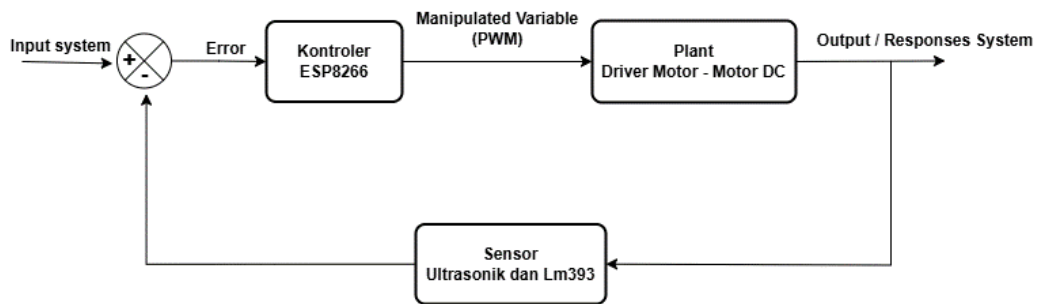
Pada diagram alir di Gambar 1 dijelaskan sistem menggambarkan tahapan utama dalam pengembangan *Adaptive Cruise Control* (ACC) berbasis *fuzzy logic* pada prototipe robocar. Proses diawali dengan identifikasi kebutuhan sistem, termasuk pemilihan sensor, motor, dan mikrokontroler yang sesuai untuk mendukung kontrol kecepatan adaptif. Setelah kebutuhan perangkat keras ditentukan, tahap berikutnya adalah perancangan sistem ACC, yang melibatkan pembuatan diagram blok serta alur kerja sistem untuk memastikan integrasi yang optimal antara perangkat keras dan perangkat lunak.

Langkah selanjutnya adalah pengembangan *Fuzzy Logic Controller* (FLC), di mana variabel *input* dan *output* didefinisikan, aturan *fuzzy* (*Base Rule*) dibentuk, serta metode inferensi dan defuzzifikasi dirancang untuk menghasilkan kontrol kecepatan yang responsif terhadap perubahan kondisi lingkungan. Implementasi sistem dilakukan dengan memprogram ESP8266 untuk mengolah data dari sensor ultrasonik dan sensor kecepatan LM393, lalu menerapkan algoritma *fuzzy* untuk mengatur kecepatan motor DC melalui motor driver.

Tahapan pengujian dan analisis dilakukan dengan mengukur respons sistem terhadap perubahan jarak dan kecepatan, mengevaluasi kestabilan kontrol, serta membandingkan hasil eksperimen dengan simulasi sebelumnya. Uji performa ini bertujuan untuk menentukan efektivitas ACC dalam menjaga jarak aman dan menyesuaikan kecepatan secara adaptif. Tahap terakhir adalah evaluasi dan optimasi parameter, di mana hasil pengujian dianalisis untuk menyempurnakan kinerja FLC sehingga sistem dapat bekerja lebih optimal dalam berbagai skenario pengendalian.

2.2 Perancangan Perangkat Keras

Diagram blok adalah representasi visual yang digunakan untuk menjelaskan konsep secara lebih jelas dan ringkas. Diagram ini bertujuan untuk mempermudah pemahaman hubungan antara *input* dan *output* dalam sebuah sistem.



Gambar 2. Diagram blok perangkat keras

Pada Gambar 2, ditampilkan ilustrasi diagram blok sistem *Adaptive Cruise Control* (ACC) yang dirancang. Sistem ini menggunakan berbagai komponen sensor dan aktuator untuk menjalankan fungsinya. Sensor ultrasonik berperan sebagai *input* utama yang membaca jarak terhadap objek atau kendaraan di depannya. NodeMCU ESP8266 berfungsi sebagai pengendali utama sistem, yang bertugas memproses data dari sensor, menerapkan metode *fuzzy logic*, dan menghasilkan output untuk mengontrol aktuator.

Setelah nilai jarak dibaca oleh sensor ultrasonik, data tersebut diproses oleh ESP8266 menggunakan algoritma *fuzzy logic*. Proses ini menghasilkan sinyal PWM (*Pulse Width Modulation*) 8-bit yang sesuai dengan aturan kontrol yang telah diprogram. Sinyal PWM ini kemudian dikirim ke motor driver L298N sebagai *output*, yang berfungsi mengendalikan motor DC. Kecepatan motor DC selanjutnya dipantau menggunakan sensor LM393 sebagai bagian dari sistem umpan balik untuk memastikan bahwa motor beroperasi sesuai dengan nilai yang telah ditentukan.

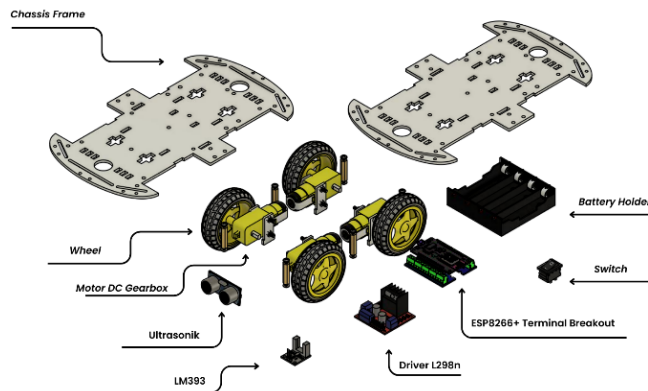
Selain itu, perangkat tambahan seperti aplikasi Blynk terhubung melalui Wi-Fi untuk memungkinkan pengguna memilih mode kontrol, baik manual maupun *fuzzy logic*.

Dalam sistem *Adaptive Cruise Control* (ACC) ini, modul ESP8266 berfungsi sebagai mikrokontroler utama yang mengatur seluruh sistem. Komponen ini menghubungkan *Robo Car* ke platform Blynk untuk kontrol dan pemantauan jarak jauh, serta mampu menjalankan skrip yang kompleks. Fungsi utamanya adalah menerima data dari sensor, mengolah data menggunakan algoritma *fuzzy logic*, dan mengirimkan sinyal PWM 8-bit ke motor driver untuk mengontrol kecepatan kendaraan.

Sensor ultrasonik (HC-SR04) digunakan untuk mengukur jarak antara *Robo Car* dan objek di depannya. Sensor ini memiliki jangkauan pengukuran 2 cm hingga 400 cm dengan akurasi sekitar 3 mm. *Input* jarak dari sensor ini diproses oleh sistem *fuzzy logic* untuk menyesuaikan kecepatan kendaraan secara adaptif berdasarkan jarak terdeteksi. Selanjutnya, empat motor DC digunakan sebagai aktuator untuk menggerakkan roda *Robo Car*. Pengendalian arah dan kecepatan motor dilakukan oleh motor driver L298N, yang menerjemahkan sinyal PWM dari ESP8266 menjadi aksi fisik. Motor driver ini memiliki spesifikasi tegangan operasi 5V–35V dan arus maksimum 2A per *channel*, yang memungkinkannya mendukung kebutuhan operasional sistem.

Selain itu, sensor kecepatan digunakan untuk mengukur putaran per menit (RPM) motor. Data aktual dari sensor ini memberikan masukan penting untuk sistem *fuzzy logic* agar kecepatan motor dapat diatur lebih akurat sesuai dengan kecepatan yang diinginkan. Platform Blynk berfungsi mendukung pengujian dan pemantauan jarak jauh sistem. Melalui platform ini, pengguna dapat memantau status kendaraan, beralih antara mode manual dan *fuzzy*, serta melihat data real-time seperti jarak, kecepatan, dan status sistem lainnya. Fitur *virtual button* dan *virtual slider* pada Blynk memberikan kemudahan bagi pengguna selama fase pengujian, didukung oleh antarmuka yang ramah pengguna serta kompatibilitas dengan berbagai perangkat.

Dengan desain ini, hubungan antara *input*, proses, dan *output* dalam sistem ACC dapat diidentifikasi dengan jelas dan dipahami secara sistematis. Kemudian dibuatlah gambar teknik perangkat keras sistem dan hasil pengembangan implementasinya pada seperti ditunjukkan pada Gambar 3 dan Gambar 4.



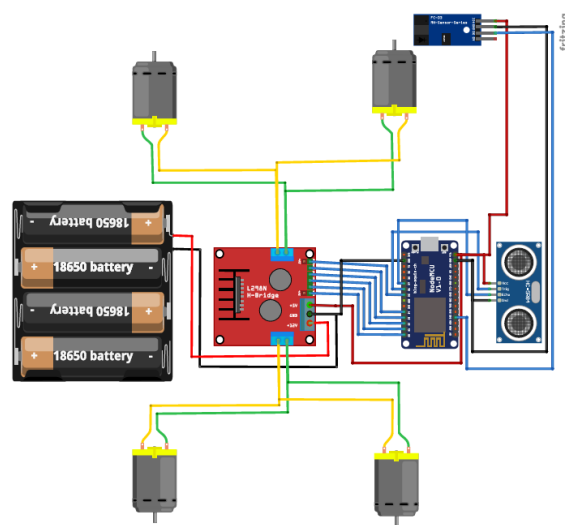
Gambar 3. Gambar 3D dari perangkat keras sistem



Gambar 4. Hasil pengembangan prototipe

Visualisasi gambar 3D dan juga hasil implementasi sistem menunjukkan bahwa komponen-komponen ACC ditempatkan secara strategis pada *chassis Robo Car* untuk memastikan efisiensi dan kemudahan operasional. Modul ESP8266 ditempatkan di pusat sebagai pengendali utama, terhubung ke motor *driver*, sensor ultrasonik, dan sensor kecepatan melalui koneksi kabel. Sensor ultrasonik dipasang di bagian depan kendaraan untuk deteksi jarak yang optimal, sementara sensor kecepatan terhubung ke ESP8266 untuk memberikan data *real-time*. Penataan komponen ini memungkinkan integrasi sistem yang efisien dan mendukung kinerja optimal dari prototipe ACC Robo Car. Selanjutnya juga dibuat rangkaian skematik diagram yang terdiri dari berbagai komponen elektronik yang saling terhubung untuk mendukung fungsionalitasnya secara optimal. Komponen utama adalah mikrokontroler ESP8266, yang berfungsi sebagai pusat kendali seluruh sistem. Mikrokontroler ini bertanggung jawab untuk menerima dan mengolah data dari sensor, serta mengatur keluaran yang sesuai untuk mengendalikan aktuator. Motor DC *gearbox* digunakan sebagai penggerak utama *prototipe*, dengan motor driver L298N yang mengontrol arah dan kecepatan motor berdasarkan sinyal PWM yang diterima dari ESP8266.

Sensor ultrasonik dipasang di bagian depan kendaraan untuk mendeteksi jarak terhadap objek di depannya. Sensor ini memberikan data jarak secara *real-time*, yang kemudian diproses oleh sistem untuk menyesuaikan kecepatan kendaraan secara adaptif. Seluruh sistem diberi daya oleh baterai 18650, yang menyediakan catu daya stabil dan mendukung operasional sistem dalam waktu yang cukup lama. Integrasi dari semua komponen ini dirancang dengan cermat agar dapat bekerja secara selaras, sehingga memungkinkan sistem mencapai tujuan pengembangan dan fungsionalitas yang diinginkan seperti pada gambar 5.



Gambar 5. Skematik rangkaian perangkat keras

Pada sistem ini, ESP8266 berperan sebagai pusat kendali yang mengintegrasikan berbagai komponen elektronik untuk menjalankan fungsionalitas sistem *Adaptive Cruise Control* (ACC) secara optimal. Sensor ultrasonik terhubung ke pin GPIO 12 (Trig) dan GPIO 13 (Echo) pada ESP8266, di mana Trig berfungsi mengirim sinyal ultrasonik, dan Echo menerima pantulan sinyal tersebut untuk menghitung jarak terhadap objek di depan. Motor driver L298N digunakan untuk mengontrol kecepatan dan arah motor DC. Pin GPIO 16 pada ESP8266 terhubung ke ENA untuk mengontrol kecepatan motor A, sedangkan GPIO 5 dan GPIO 4 masing-masing terhubung ke IN1 dan IN2 untuk mengatur arah motor A. Untuk motor B, pin GPIO 0 dan GPIO 2 masing-masing terhubung ke IN3 dan IN4, dengan GPIO 14 terhubung ke ENB untuk kontrol kecepatan motor B.

Sensor kecepatan terhubung ke pin GPIO 10 ESP8266 untuk memberikan data kecepatan motor secara real-time, membantu sistem dalam penyesuaian kecepatan menggunakan algoritma *fuzzy logic*. Selain itu, antarmuka *platform* Blynk mempermudah kontrol dan monitoring sistem. *Virtual Button* (V1) digunakan untuk memilih mode kontrol manual atau *fuzzy logic*, sedangkan *Virtual Slider* (V2) digunakan untuk mengatur kecepatan secara manual. Dalam diagram rangkaian, ESP8266 menerima data jarak dari sensor ultrasonik dan kecepatan motor dari sensor kecepatan, memproses data tersebut dengan algoritma *fuzzy logic*, dan mengirimkan sinyal kontrol PWM ke motor *driver*. Motor *driver* kemudian mengatur kecepatan dan arah motor DC, memungkinkan *Robo Car* beroperasi sesuai dengan kondisi lingkungan yang terdeteksi. Platform Blynk mendukung pengujian dan pemantauan sistem secara real-time, memberikan fleksibilitas dan kemudahan bagi pengguna.

2.3 Perancangan *Fuzzy Logic*

Sistem *Adaptive Cruise Control* (ACC) pada prototipe ini dirancang menggunakan metode *fuzzy logic* yang terdiri dari tiga komponen utama, yaitu Fuzzifikasi, *Base Rule*, dan Defuzzifikasi. Sistem menerima dua variabel input, yaitu jarak (cm) dan kecepatan (RPM), serta satu variabel output berupa nilai PWM motor.

Pada tahap Fuzzifikasi, nilai numerik dari sensor diubah menjadi variabel linguistik berdasarkan fungsi keanggotaan fuzzy. Variabel input pertama, yaitu jarak, diklasifikasikan menjadi empat kategori, yaitu Dekat (0-60 cm), Sedang (50-160 cm), Jauh (120-280 cm), dan Sangat Jauh (200-300 cm). Variabel *input* kedua, yaitu kecepatan (RPM), juga diklasifikasikan menjadi empat kategori, yaitu Sangat Lambat (0-120 RPM), Lambat (100-260 RPM), Cepat (240-400 RPM), dan Sangat Cepat (380-470 RPM). Variabel *output* PWM motor juga dibagi ke dalam empat tingkat, yaitu PWM Off (0), PWM Rendah (0-100), PWM Sedang (55-200), dan PWM Tinggi (155-255).

Tabel 1. Aturan *Fuzzy*

Jarak / Kecepatan	Sangat Lambat	Lambat	Cepat	Sangat Cepat
Dekat	PWM Off (0)	PWM Off (0)	PWM Off (0)	PWM Off (0)
Sedang	PWM Low (1)	PWM Low (1)	PWM Low (1)	PWM Low (1)
Jauh	PWM Mid (2)	PWM Mid (2)	PWM Mid (2)	PWM Mid (2)
Sangat Jauh	PWM High (3)	PWM High (3)	PWM High (3)	PWM High (3)

Tabel 1 menunjukkan tahap *Base Rule*, sistem menggunakan aturan *fuzzy* yang telah ditentukan untuk menentukan *output* PWM berdasarkan kombinasi antara jarak dan kecepatan. Sebanyak 16 aturan *fuzzy* telah dirancang untuk memastikan kendaraan dapat menyesuaikan kecepatan secara otomatis. Jika jarak berada dalam kategori Dekat, motor akan dimatikan (*PWM Off*) untuk menghindari tabrakan, tanpa memandang nilai kecepatan. Jika jaraknya Sedang, maka motor akan beroperasi pada tingkat *PWM Low*, sedangkan jika jaraknya Jauh, motor akan beroperasi dengan *PWM Mid*. Untuk kondisi Sangat Jauh, sistem akan meningkatkan PWM ke *High*, sehingga kendaraan dapat mempertahankan kecepatan optimal.

Pada tahap Defuzzifikasi, *output fuzzy* yang diperoleh dari aturan *fuzzy* dikonversikan kembali menjadi nilai numerik PWM yang digunakan untuk mengontrol kecepatan motor. Metode defuzzifikasi yang digunakan dalam sistem ini adalah *Center of Gravity* (COG) untuk menentukan nilai akhir PWM berdasarkan bobot keanggotaan fuzzy yang dihasilkan dari *Base Rule*. Dengan pendekatan berbasis *fuzzy logic* ini, sistem dapat mengelola transisi kecepatan secara halus, menghindari pengereman mendadak, serta memberikan kenyamanan bagi pengendara. Selain itu, implementasi *fuzzy logic* memungkinkan sistem ACC untuk beradaptasi dengan berbagai kondisi lalu lintas dan menjaga jarak aman secara otomatis.

3 Hasil dan Diskusi

Setelah fabrikasi prototipe *Robo Car* selesai, dilakukan serangkaian pengujian untuk mengevaluasi performa sistem *Adaptive Cruise Control* (ACC) yang dikembangkan menggunakan metode *Fuzzy Logic Controller* (FLC). Fokus utama pengujian adalah untuk memastikan bahwa sistem ACC bekerja sesuai dengan desain yang telah dirancang, serta mengevaluasi efektivitas implementasi *fuzzy logic* dalam mengendalikan kecepatan kendaraan berdasarkan data jarak dan kecepatan.

3.1 Implementasi *Fuzzy Logic Control* (FLC) dari MATLAB ke ESP8266

Fuzzy Logic Controller (FLC) pada sistem ini dikembangkan terlebih dahulu menggunakan MATLAB untuk melakukan analisis dan simulasi sebelum diimplementasikan ke dalam mikrokontroler ESP8266. MATLAB dipilih karena kemampuannya dalam membangun dan menguji sistem kendali berbasis *fuzzy* dengan cepat dan akurat sebelum diterapkan ke perangkat keras. Dalam tahap awal, model *fuzzy logic* dibangun dengan mendefinisikan variabel input berupa jarak dan kecepatan kendaraan, serta variabel output yang mengontrol nilai *Pulse Width Modulation* (PWM) motor. Setiap variabel input dan *output* memiliki fungsi keanggotaan yang dirancang dengan pendekatan berbasis eksperimen dan analisis, sehingga dapat mencerminkan kondisi nyata yang akan diterapkan pada sistem *Adaptive Cruise Control* (ACC).

Setelah model *fuzzy* terbentuk, dilakukan simulasi dan validasi melalui berbagai skenario untuk memastikan bahwa aturan *fuzzy* dapat mengontrol kecepatan motor secara optimal berdasarkan kondisi yang diberikan. Simulasi ini bertujuan untuk menguji bagaimana sistem *fuzzy* merespons perubahan jarak dan kecepatan, serta untuk memverifikasi bahwa pengambilan keputusan yang dilakukan sesuai dengan logika yang telah ditentukan. Jika hasil simulasi menunjukkan performa yang kurang optimal, maka dilakukan penyesuaian pada fungsi keanggotaan dan aturan inferensi hingga sistem bekerja secara efektif.

Setelah tahap simulasi di MATLAB selesai dan hasilnya dianggap valid, langkah berikutnya adalah mengekstraksi parameter *fuzzy* yang terdiri dari fungsi keanggotaan, aturan inferensi, dan tabel keputusan. Data-data ini kemudian diimplementasikan ke dalam mikrokontroler ESP8266 menggunakan pemrograman dalam bahasa C berbasis Arduino IDE. Proses ini melibatkan penerjemahan aturan *fuzzy* ke dalam kode yang dapat dijalankan secara real-time oleh ESP8266 untuk mengontrol kecepatan motor melalui sinyal PWM. Dengan cara ini, sistem ACC yang telah diuji secara teoritis di MATLAB dapat diaplikasikan pada perangkat keras secara langsung, memungkinkan kendaraan untuk beradaptasi terhadap perubahan jarak dan kecepatan secara otomatis sesuai dengan *fuzzy logic* yang telah dirancang.

3.2 Pengujian Sistem ACC pada Prototipe

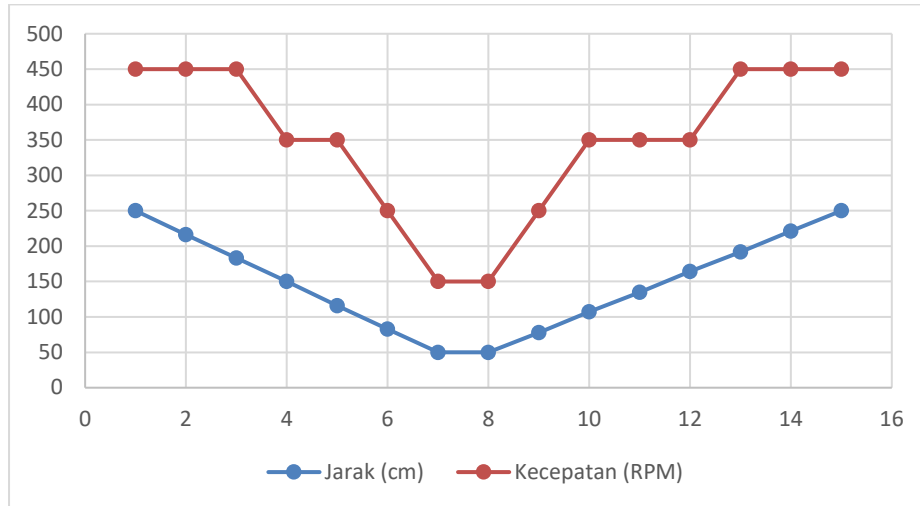
Pengujian difokuskan pada performa sistem ACC dalam berbagai kondisi operasional. Evaluasi dilakukan dengan melihat bagaimana sistem bereaksi terhadap perubahan jarak dan kecepatan kendaraan secara *real-time*.

3.3 Respon ACC terhadap Jarak dan Hambatan dan Evaluasi

Sistem diuji dengan berbagai skenario, di mana kendaraan berjalan pada kecepatan tertentu dan diberikan hambatan pada jarak berbeda. Tujuannya adalah untuk melihat apakah sistem dapat menyesuaikan kecepatan motor sesuai dengan aturan *fuzzy* yang telah diterapkan.

Tabel 2. Implementasi FLC pada Prototipe

Kondisi	Jarak (cm)	Kecepatan (RPM)	PWM 8 bit Prototipe Robo Car	PWM 8 bit MatLab	Kondisi Akhir
Kecepatan lambat dan tiba tiba diberikan halangan secara dadakan	29	207	0	0	Berhasil berhenti sempurna
Kecepatan sedang dan ada obyek dekat	55	228	41	40,7	Robo Car memelan dan motor berjalan lambat
Kecepatan cepat dan ada obyek di jarak dekat	51	372	44	43,5	Robo Car tetap berjalan lambat untuk menghindari tabrakan
Kecepatan sangat kencang dan ada obyek di jarak sedang	166	465	127	127	Robo Car berjalan dengan lancar
Kecepatan kencang dengan jarak sedang	141	366	92	92	Robo Car memelan dengan kecepatan rendah
Kecepatan lambat dan sangat jauh dari halangan (Jarak diatas 250)	1191	129	213	214	Robo Car berhasil berjalan sempurna dan kecepatan dinaikkan menjadi sangat cepat



Gambar 6. Grafik Respon hubungan perubahan jarak dan kecepatan

Tabel 2 dan gambar 6 menunjukkan hasil pengujian menunjukkan bahwa implementasi *fuzzy logic* pada prototipe Robo Car memberikan hasil yang hampir identik dengan simulasi yang dilakukan di MATLAB. Hal ini membuktikan bahwa aturan *fuzzy* yang dirancang dapat diterapkan secara nyata dalam sistem *Adaptive Cruise Control* (ACC) berbasis ESP8266 tanpa adanya penyimpangan signifikan. Keselarasan antara hasil simulasi dan hasil nyata pada prototipe menegaskan bahwa metode *fuzzy logic* yang digunakan efektif dalam mengontrol kecepatan kendaraan berdasarkan *input* jarak dan kecepatan.

Untuk memastikan bahwa sistem yang berjalan pada ESP8266 sesuai dengan model yang telah disimulasikan di MATLAB, dilakukan evaluasi konsistensi dengan membandingkan nilai *Pulse Width Modulation* (PWM) yang dihasilkan oleh kedua platform. Hasil perbandingan menunjukkan bahwa sistem *embedded* pada ESP8266 mampu mereproduksi keputusan *fuzzy* dari MATLAB dengan tingkat akurasi yang tinggi. Hal ini

mengindikasikan bahwa proses translasi aturan *fuzzy* dari MATLAB ke ESP8266 berhasil dilakukan dengan baik, tanpa ada penyimpangan yang signifikan. Keberhasilan ini menunjukkan bahwa model *fuzzy* yang dikembangkan dalam lingkungan simulasi dapat diimplementasikan secara langsung ke dalam perangkat keras, memungkinkan sistem ACC beroperasi dengan presisi tinggi sesuai dengan *fuzzy logic* yang telah dirancang.

Selain validasi terhadap kesesuaian hasil antara MATLAB dan prototipe, evaluasi kinerja sistem ACC dalam mengatur kecepatan juga dilakukan untuk mengamati respons sistem terhadap perubahan kondisi jalan. Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem ACC yang berbasis *fuzzy logic* mampu bekerja secara adaptif dalam berbagai skenario. Ketika hambatan terdeteksi pada jarak yang sangat dekat, sistem ACC secara otomatis menghentikan kendaraan dengan tepat waktu, menghindari kemungkinan tabrakan. Selain itu, kendaraan dapat menyesuaikan kecepatan secara halus sesuai dengan perubahan jarak hambatan, tanpa adanya perubahan kecepatan yang tiba-tiba yang dapat mengganggu kestabilan berkendara. Pada kondisi di mana hambatan berada pada jarak yang jauh atau tidak ada, sistem ACC mampu mempertahankan kecepatan tinggi dengan stabil, memastikan bahwa kendaraan dapat bergerak dengan efisiensi maksimum tanpa gangguan.

Hasil evaluasi ini membuktikan bahwa implementasi *fuzzy logic* pada ESP8266 berhasil menciptakan sistem ACC yang responsif dan adaptif terhadap perubahan lingkungan secara *real-time*. Dengan demikian, metode *fuzzy logic* yang telah diuji dalam simulasi MATLAB terbukti dapat diterapkan secara nyata untuk mengendalikan kecepatan kendaraan listrik secara otomatis berdasarkan kondisi yang terdeteksi oleh sensor. Keberhasilan ini menunjukkan bahwa pendekatan *fuzzy logic* merupakan solusi yang efektif untuk sistem ACC, memberikan kendali yang lebih fleksibel dan optimal dibandingkan metode konvensional seperti PID.

4 Kesimpulan

Penelitian ini telah berhasil mengembangkan dan menguji sistem *Adaptive Cruise Control* (ACC) berbasis *fuzzy logic* pada sebuah prototipe *Robo Car* dengan menggunakan ESP8266 sebagai mikrokontroler utama. Sistem ini dirancang untuk mengatur kecepatan kendaraan secara otomatis berdasarkan data jarak dan kecepatan, menggunakan metode *fuzzy logic* yang memungkinkan sistem beradaptasi terhadap kondisi dinamis di lingkungan nyata.

Dalam penelitian ini, *Fuzzy Logic Controller* (FLC) dikembangkan dan diuji terlebih dahulu menggunakan MATLAB, yang digunakan untuk menyusun aturan *fuzzy*, menguji fungsi keanggotaan, dan melakukan simulasi sebelum diterapkan ke perangkat keras. Setelah simulasi menunjukkan hasil optimal, aturan *fuzzy* dan parameter keanggotaan diterjemahkan ke dalam kode program dan diimplementasikan pada ESP8266 untuk mengontrol kecepatan motor melalui sinyal PWM.

Hasil pengujian menunjukkan bahwa sistem ACC pada prototipe *Robo Car* dapat bekerja secara adaptif dalam berbagai kondisi operasional. Pengujian dilakukan dengan beberapa skenario, seperti kendaraan yang harus berhenti saat mendeteksi hambatan dekat, melambat saat hambatan berada pada jarak menengah, serta mempertahankan kecepatan tinggi saat tidak ada hambatan. Hasil menunjukkan bahwa sistem berhasil menghentikan kendaraan secara tepat waktu untuk mencegah tabrakan, melambat dengan halus ketika mendekati hambatan, dan menjaga kestabilan kecepatan saat kondisi jalan memungkinkan.

Perbandingan antara hasil simulasi MATLAB dan implementasi pada prototipe menunjukkan bahwa terdapat perbedaan antara keduanya yang disebabkan oleh keterbatasan perangkat keras, latensi pemrosesan mikrokontroler, serta faktor lingkungan yang tidak dapat direplikasi dalam simulasi. Namun, perbedaan ini tidak memengaruhi kinerja keseluruhan sistem ACC, karena sistem tetap mampu mengontrol kecepatan kendaraan sesuai dengan aturan *fuzzy* yang telah dirancang.

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa sistem ACC berbasis *fuzzy logic* yang diimplementasikan pada ESP8266 berhasil bekerja secara adaptif dalam berbagai kondisi operasional, memungkinkan kendaraan untuk menyesuaikan kecepatan secara otomatis berdasarkan jarak hambatan yang terdeteksi. Hasil pengujian pada prototipe menunjukkan bahwa sistem dapat menghentikan kendaraan dengan tepat waktu, menyesuaikan kecepatan tanpa perubahan mendadak, dan mempertahankan kecepatan tinggi dengan stabil, sesuai dengan konsep *Adaptive Cruise Control* (ACC). Perbandingan antara hasil simulasi MATLAB dan implementasi pada perangkat keras menunjukkan bahwa meskipun terdapat perbedaan akibat keterbatasan fisik, sistem secara keseluruhan tetap bekerja dengan baik dalam mengontrol kecepatan kendaraan. *Fuzzy logic* terbukti menjadi metode yang efektif untuk sistem ACC, karena mampu menangani ketidakpastian dalam kondisi jalan dan memberikan kontrol yang lebih fleksibel dibandingkan metode konvensional seperti PID.

Dengan demikian, penelitian ini membuktikan bahwa *Adaptive Cruise Control* berbasis *fuzzy logic* dapat diterapkan secara nyata pada kendaraan listrik mini, dengan hasil yang cukup mendekati simulasi MATLAB. Untuk penelitian selanjutnya, disarankan untuk melakukan optimasi parameter *fuzzy*, menggunakan perangkat keras dengan latensi lebih rendah, serta menguji sistem dalam lingkungan yang lebih kompleks untuk meningkatkan keandalan sistem ACC dalam kondisi dunia nyata.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Sekolah Vokasi Universitas Diponegoro, khususnya kepada Unit Penerapan dan Pengembangan Manajemen (UPPM), yang telah memberikan bantuan dana penelitian sehingga penelitian ini dapat terlaksana dengan lancar.

Referensi

- [1] Merdeka, "JK: Di kota besar, mobil sudah menjadi kebutuhan pokok," (28 April 2025). [Online]. Available: <https://www.merdeka.com/uang/jk-di-kota-besar-mobil-sudah-menjadi-kebutuhan-pokok.html> [Accessed: 03 February 2025].
- [2] S. Thrun, "Toward robotic cars," *Commun ACM*, vol. 53, no. 4, pp. 99–106, 2010. <https://doi.org/10.1145/1721654.1721679>
- [3] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," *SAE International*, 400 Commonwealth Drive, Warrendale, PA, United States, Jun. 2018. https://doi.org/10.4271/J3016_201806.
- [4] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, "Autonomous Driving: Technical, Legal and Social Aspects." Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. <https://doi.org/10.1007/978-3-662-48847-8>.
- [5] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transp Res Part A Policy Pract*, vol. 77, pp. 167–181, 2015. <https://doi.org/10.1016/j.tra.2015.04.003>
- [6] A. Ahdiat, "Ini pertumbuhan jumlah mobil di indonesia 10 tahun terakhir," Databoks.katadata.co.id, (15 Maret 2023). [Online]. Available: <https://databoks.katadata.co.id/transportasi-logistik/statistik/f266a30f3f686f1/ini-pertumbuhan-jumlah-mobil-di-indonesia-10-tahun-terakhir>. [Accessed: 03 February 2025].
- [7] Biro Komunikasi dan Informasi, "Tekan angka kecelakaan lalu lintas, kemenhub ajak masyarakat beralih ke transportasi umum dan utamakan keselamatan berkendara," (19 September 2023). [Online]. Available: dephub.go.id
- [8] Astra, "Adaptive Cruise Control adalah : kenali fitur ini pada mobil toyota," (21 September 2022). [Online]. Available: <https://auto2000.co.id/berita-dan-tips/adaptive-cruise-control-adalah>. [Accessed: 03 February 2025].
- [9] Heart Autos Research, "What is Adaptive Cruise Control?," *Car and Driver*, Jul. 2020. [Online]. Available: <https://www.caranddriver.com/research/a32813983/adaptive-cruise-control/>. [Accessed: 03 February 2025].
- [10] J. Levinson et al., "Towards fully autonomous driving : Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, pp. 163–168, 2011. <https://doi.org/10.1109/IVS.2011.5940562>.
- [11] C. Urmson et al., "Autonomous driving in urban environments: boss and the urban challenge," *J Field Robot*, vol. 25, no. 8, pp. 425–466, 2008. https://doi.org/10.1007/978-3-642-03991-1_1.
- [12] H. Winner, S. Witte, W. Uhler, and B. Lichtenberg, "Adaptive Cruise Control system aspects and development trends," *SAE International Journal of Passenger - Electronic and Electrical Systems*, vol. 9, no. 2, pp. 247–256, 2016. <https://doi.org/10.4271/961010>
- [13] A. V. Prabhu, B. Chandra Subash, J. Sai Kumar, D. Ramesh, B. P. Kumar, and A. K. Sai, "Impact of Adaptive Cruise Control system in automobiles," 2023. [Online]. Available: <https://ijrpr.com/uploads/V4ISSUE3/IJRPR10919.pdf>
- [14] L. A. Zadeh, "Fuzzy logic," *Computer (Long Beach Calif)*, vol. 21, no. 4, pp. 83–93, 1988. <https://doi.org/10.1109/2.53>.
- [15] H. P. Nurba, E. Susanto, and A. S. Wibowo, "Desain dan Implementasi Kendali PID Adaptif pada Kecepatan Motor DC," *e-Proceeding of Engineering*, vol. 2, no. 2, pp. 2050–2057, 2015. [Online]. Available : https://openlibrary.telkomuniversity.ac.id/pustaka/files/102655/jurnal_eproc/desain-dan-implementasi-kendali-pid-adaptif-pada-kecepatan-motor-dc.pdf

- [16] K. Alomari, R. Mendoza, S. Sundermann, D. Goehring, and R. Rojas, "Fuzzy logic-based Adaptive Cruise Control for autonomous model car," *Proceedings of the International Conference on Robotics, Computer Vision and Intelligent Systems*, pp. 121–130, 2020. <https://doi.org/10.5220/0010175101210130>
- [17] E. Firlana, H. A. Nurcahyani, and A. D. Dwipayana, "Pengaruh *cruise control* terhadap keselamatan berkendara di jalan tol," *Prosiding Forum Studi Transfortasi Antar Perguruan Tinggi*, vol. 10, no. 2, pp. 699–707, 2023. [Online]. Available : <https://prosiding.jurnal.unej.ac.id/index.php/PFSTPT/article/view/44780>