

Sistem Navigasi Berbasis Modul Kompas & GPS Pada Prototipe *Autonomous Car*
Dengan Pengendali *Steering* Metode PI

**Sistem Navigasi Berbasis Modul Kompas & GPS Pada Prototipe *Autonomous Car*
Dengan Pengendali *Steering* Metode PI**

Naufal Muzakki

Program Studi S1 Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia

e-mail : naufal.21077@mhs.unesa.ac.id

I Gusti Putu Asto Buditjahjanto, Bambang Suprianto, Sayyidul Aulia Alamsyah

Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya, Ketintang 60231, Indonesia

e-mail : asto@unesa.ac.id, bambangsuprianto@unesa.ac.id, sayyidulalamsyah@unesa.ac.id

Abstrak

Perkembangan kendaraan otonom telah menjadi salah satu inovasi paling signifikan dalam industri transportasi, mendorong penelitian yang luas karena potensinya untuk meningkatkan keselamatan penumpang, mobilitas penyandang disabilitas, dan mengurangi emisi gas rumah kaca. Permasalahan utama yang diangkat adalah bagaimana menyederhanakan sistem navigasi berbasis sensor dengan tetap mempertahankan akurasi arah (*heading*) dan posisi kendaraan saat mengikuti rute *waypoint*. Penelitian ini mengembangkan dan menguji sistem navigasi pada prototipe *autonomous car* dengan kendali *steering* berbasis *Proportional-Integral* (PI). Sistem ini memanfaatkan sensor GPS Ublox SAM-M8Q dan kompas HMC5883L yang terintegrasi dengan IMU BNO055 untuk menentukan posisi dan arah kendaraan secara *real-time*. Jarak antar koordinat dihitung menggunakan *Haversine Formula*, sedangkan *Bearing Formula* digunakan untuk menentukan sudut *heading* sebagai *setpoint* kendali. Pengujian dilakukan selama tiga hari dengan rute berbentuk persegi dan empat *waypoint* berbeda, disesuaikan dengan kondisi lapangan dan *obstacle* yang ada. Hasil menunjukkan sistem navigasi mampu mencapai *waypoint* dengan rata-rata *error* posisi sebesar 1.58 meter, *error* terkecil 0.11 meter, dan terbesar 5.10 meter. Selain itu, Integrasi sensor HMC5883L dan BNO055 menghasilkan *error heading* rata-rata 1.57°, sedangkan pengujian GPS Ublox SAM-M8Q menunjukkan *drift* posisi rata-rata 0.4 meter dalam durasi 1 menit. Temuan ini menunjukkan bahwa kendali PI efektif dalam menjaga arah kendaraan secara stabil di lingkungan nyata.

Kata Kunci: *Autonomous Car*, Kontrol PI, *Haversine Formula*, Akurasi *Heading*, Kendali *Steering*, Modul Kompas & GPS

Abstract

The development of autonomous vehicles has become one of the most significant innovations in the transportation industry, driving extensive research due to its potential to improve passenger safety, mobility of people with disabilities, and reduce greenhouse gas emissions. The main issue raised is how to simplify the sensor-based navigation system while maintaining the accuracy of the vehicle's heading and position while following the waypoint route. This research develops and tests a navigation system on an autonomous car prototype with *Proportional-Integral* (PI) based steering control. The system utilizes Ublox SAM-M8Q GPS sensor and HMC5883L compass integrated with BNO055 IMU to determine the position and direction of the vehicle in *real-time*. The distance between coordinates is calculated using the *Haversine Formula*, while the *Bearing Formula* is used to determine the heading angle as the control *setpoint*. Testing was conducted for three days with a square-shaped route and four different waypoints, adjusted to field conditions and obstacles. The results show that the navigation system is able to reach the waypoint with an average position error of 1.58 meters, the smallest error is 0.11 meters, and the largest is 5.10 meters. In addition, the integration of HMC5883L and BNO055 sensors resulted in an average heading error of 1.57°, while the Ublox SAM-M8Q GPS test showed an average position drift of 0.4 meters within a duration of 1 minute. These findings show that PI control is effective in reducing the drift.

Keywords: *Autonomous Car*, PI Control, *Haversine Formula*, Heading Accuracy, Steering Control. Compass & GPS Module.

PENDAHULUAN

Kendaraan otonom merupakan sistem transportasi yang mampu bergerak tanpa campur tangan manusia,

dengan mengandalkan sensor untuk mengumpulkan informasi lingkungan dan menghasilkan keputusan dalam pengendalian pergerakan (Yadav et al., 2020). Inovasi ini menjadi fokus utama berbagai riset karena potensinya

dalam meningkatkan keselamatan penumpang, membuka akses mobilitas bagi penyandang disabilitas, dan mengurangi emisi gas rumah kaca akibat efisiensi rute dan pengurangan intervensi manusia dalam mengemudi (Olayode et al., 2023), khususnya dalam konteks implementasi di negara berkembang seperti Indonesia. Oleh karena itu, pengembangan sistem kendaraan otonom menjadi urgensi dalam mendorong transportasi masa depan yang aman, inklusif, dan ramah lingkungan.

Untuk mendukung sistem pengendalian pada kendaraan otonom, dibutuhkan sistem navigasi yang mampu memberikan informasi posisi dan arah secara akurat. GPS merupakan sistem navigasi luar ruangan yang umum digunakan karena keunggulannya dalam akurasi dan performa waktu nyata (Ketsayom et al., 2023). Namun, GPS tidak dapat memberikan informasi sudut arah (*heading*), sehingga perlu dikombinasikan dengan sensor kompas. Integrasi GPS dan kompas memberikan informasi koordinat dan orientasi yang saling melengkapi dalam meningkatkan presisi navigasi (Ušinskis et al., 2024).

Salah satu pendekatan umum dalam sistem navigasi kendaraan otonom adalah penggunaan metode *waypoint*, yaitu serangkaian titik koordinat yang menjadi referensi jalur yang harus diikuti kendaraan (Christensen et al., 2021). Tantangan dalam navigasi *waypoint* adalah menjaga *heading* kendaraan tetap stabil terhadap berbagai gangguan, seperti interferensi medan magnet dan *noise* sensor, karena *heading* yang tidak stabil dapat menyebabkan deviasi jalur berupa *oversteer* atau *understeer*.

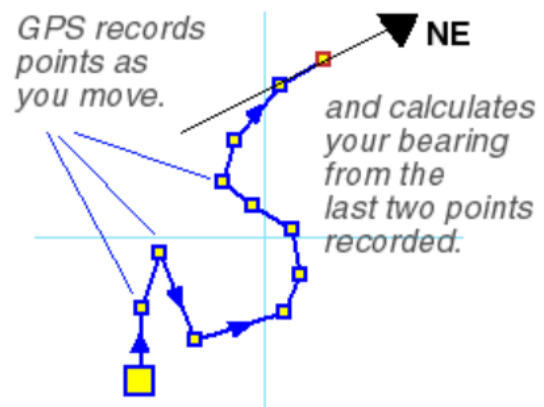
Untuk menjaga stabilitas arah kendaraan, sistem kontrol *feedback* sangat diperlukan. Meskipun kontrol PID banyak digunakan dalam aplikasi industri karena kestabilan dan responsivitasnya (Pendleton et al., 2017), tuning parameter PID (K_p , K_i , K_d) seringkali kompleks dan membutuhkan penyesuaian yang presisi. Oleh karena itu, penelitian ini memfokuskan pada penggunaan kontrol PI (*Proportional-Integral*) dengan mengabaikan komponen *Derivative* ($K_d = 0$) untuk menyederhanakan proses tuning, namun tetap menjaga kinerja sistem dalam mempertahankan *heading* (Song et al., 2023).

TINJUAN PUSTAKA

Sistem Navigasi Waypoint

Sistem navigasi *waypoint* merupakan metode yang umum digunakan dalam kendaraan otonom untuk memandu pergerakan kendaraan dari satu titik koordinat GPS ke titik berikutnya hingga mencapai tujuan akhir. Setiap titik dianggap sebagai *waypoint* dalam sistem koordinat *latitude* dan *longitude*. Dengan bantuan GPS untuk menentukan posisi dan kompas magnetik untuk arah, sistem ini memungkinkan kendaraan melakukan koreksi arah dan perhitungan

jarak tempuh secara akurat (Ashar Ashari, 2020).



Gambar 1. Sistem Navigasi Waypoint

Dalam implementasinya, GPS mencatat pergerakan sebagai serangkaian titik koordinat, di mana dua titik terakhir digunakan untuk menghitung bearing atau sudut arah gerak. *Bearing* ini penting untuk menentukan orientasi kendaraan menuju *waypoint* berikutnya. Sistem juga dilengkapi dengan mekanisme koreksi arah (*bearing correction*) dan perhitungan jarak (odometer), sehingga kendaraan dapat mengenali posisi dan arah secara dinamis berdasarkan sistem koordinat Bumi, serta bergerak secara teratur dan presisi mengikuti rute yang telah ditentukan (Ahmad Sulkhan Taufik, 2013).

Haversine Formula

Haversine Formula merupakan salah satu persamaan penting dalam navigasi, terutama dalam perhitungan jarak antara dua titik pada permukaan bumi berdasarkan koordinat *longitude* dan *latitude*. Dalam metode navigasi *waypoint*, rumus ini digunakan untuk menghitung jarak terpendek antara dua titik yang dinyatakan dalam koordinat geografis (Saputra Kurnia et al., 2019). Hal ini menjadi dasar penting dalam pergerakan kendaraan otonom, di mana perhitungan jarak antar *waypoint* sangat penting untuk menentukan jalur yang optimal.

Rumus *Haversine* pertama-tama mengubah nilai lintang dan bujur dari angka integer menjadi radian, lalu angka-angka ini dihitung menggunakan algoritma *Haversine*. Rumus ini memungkinkan perhitungan jarak yang lebih akurat dengan mempertimbangkan kelengkungan bumi yang sangat penting terutama untuk navigasi jarak jauh. Implementasi *Haversine* di dalam kendaraan otonom yang menggunakan sistem GPS memungkinkan perhitungan jarak antar *waypoint* menjadi lebih presisi yang pada gilirannya meningkatkan akurasi dan efisiensi dalam mencapai tujuan akhir. Rumus Haversine dapat dijelaskan pada persamaan (1).

$$\begin{aligned}
 a &= \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\Delta long}{2}\right) \\
 c &= 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \\
 d &= R \cdot c
 \end{aligned} \tag{1}$$

Sistem Navigasi Berbasis Modul Kompas & GPS Pada Prototipe *Autonomous Car* Dengan Pengendali *Steering* Metode PI

Keterangan :

R = jari - jari bumi sebesar 6371(km).

Δlat = $(lat_2 - lat_1)$

$\Delta long$ = $(long_2 - long_1)$

c = kalkulasi perpotongan sumbu

d = jarak (km)

1° = 0.0174532925 radian

Bearing Formula

Pada tahap ini, proses bertujuan untuk menghitung sudut bearing yang menunjukkan arah tujuan pergerakan prototipe *autonomous car*. Informasi ini diperoleh dari koordinat posisi mobil dan tujuan yang ingin dicapai. Dalam konteks ini, panjang koordinat diwakili oleh *longitude* (L), sedangkan sudut lintang diwakili oleh *latitude* (θ). Sudut bearing dilambangkan dengan simbol (β). Titik awal atau koordinat mobil ditandai sebagai titik (a), sedangkan titik tujuan sebagai titik (b). Untuk menghitung bearing dari titik A ke titik B, digunakan rumus berikut:

$$\beta = a \tan 2(Y, X) \quad (2)$$

Untuk menghitung nilai Y dan X, perlu dilakukan penghitungan terlebih dahulu. Nilai X dapat dihitung dengan menggunakan rumus:

$$X = \cos \theta a \cdot \sin \theta a - \sin \theta a \cdot \cos \theta b \cdot \cos \Delta L \quad (3)$$

Setelah mendapatkan nilai X, rumus untuk menghitung nilai Y adalah sebagai berikut:

$$Y = \cos \theta b \cdot \cos \Delta L \quad (4)$$

Namun, nilai bearing yang diperoleh masih dalam satuan radian. Untuk mengubahnya menjadi derajat, perlu dilakukan konversi dengan mengalikan nilai tersebut dengan $\left(\frac{180}{\pi}\right)$. Selanjutnya, untuk menghitung jarak antara dua titik, rumus *Haversine* digunakan. Pertama, koordinat yang diberikan dalam derajat perlu diubah menjadi radian dengan cara mengalikan dengan $\left(\frac{\pi}{180}\right)$. Rumus *Haversine* dituliskan sebagai berikut, di mana ϕ melambangkan *latitude* dan λ melambangkan *longitude*, sedangkan angka 1 dan 2 menunjukkan titik koordinat pertama dan kedua:

$$hav(\theta) = hav(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) hav(\lambda_1 - \lambda_2) \quad (5)$$

Untuk mendapatkan jarak antar koordinat, penting untuk mengetahui radius Bumi. Radius Bumi yang digunakan dalam perhitungan ini diambil dari data NASA, yaitu 6371 km. Jarak dalam kilometer kemudian dihitung dengan mengalikan radius Bumi

dengan nilai $hav(\theta)$. Rumus *Haversine* itu sendiri memerlukan fungsi *haversine* untuk menghitung jarak antara dua titik koordinat yang dirumuskan sebagai berikut:

$$hav(\theta) = \sin^2\left(\frac{\theta}{2}\right) \quad (6)$$

Metode Kontrol PI

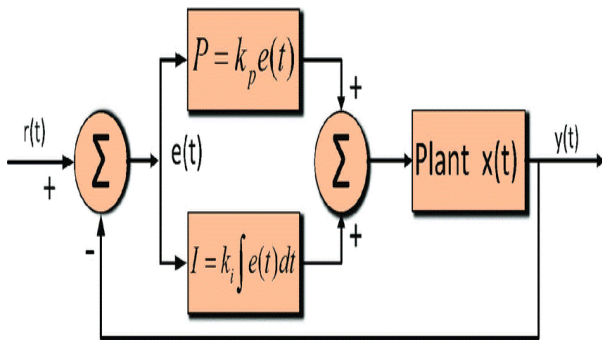
Dalam sistem *autonomous car*, sistem kendali diperlukan untuk menjaga arah kendaraan tetap menuju jalur yang telah ditentukan melalui titik-titik *waypoint*. Pengaturan sudut kemudi dilakukan berdasarkan selisih antara arah yang diinginkan (*setpoint*) dan arah aktual kendaraan, yang dikenal sebagai *error*. Salah satu metode yang digunakan untuk mengatasi *error* ini adalah kontrol PI (*Proportional-Integral*), yang merupakan bentuk penyederhanaan dari kendali PID dengan menghilangkan komponen *derivative* (D). Dalam implementasinya, sinyal kendali yang dihasilkan oleh pengendali PI dikirimkan ke motor driver untuk mengatur sudut kemudi secara dinamis berdasarkan nilai *error* tersebut.

Pengendali PI bekerja dengan dua komponen utama, yaitu *proporsional* dan *integral*. Komponen *proporsional* (K_p) berfungsi untuk memberikan respons langsung terhadap besarnya *error* saat ini, sehingga kendaraan dapat segera mengoreksi arah jika terjadi penyimpangan. Sementara itu, komponen *integral* ($K_i = \frac{1}{\tau_i}$) mengakumulasi *error* sepanjang waktu, yang memungkinkan sistem untuk mengatasi kesalahan *steady-state* secara efektif. Secara matematis, bentuk umum kendali PI dapat ditulis sebagai:

$$u(t) = K_p \left[e(t) + \frac{1}{\tau_i} \int e(t) dt \right] \quad (7)$$

dengan $u(t)$ sebagai sinyal kendali, $e(t)$ sebagai *error* sudut arah, dan τ_i sebagai konstanta waktu *integrator* (Katsuhiko Ogata, 2017).

Dalam pengembangannya, kendali PI banyak digunakan dalam berbagai sistem kendali karena kesederhanaan desain dan kestabilan responnya terhadap gangguan maupun perubahan parameter sistem. Terdapat berbagai metode tuning parameter PI seperti *Ziegler-Nichols*, *Tyres-Luyben*, dan *CHR* yang digunakan untuk mendapatkan nilai optimal dari K_p dan K_i dalam aplikasi spesifik. Dengan kombinasi yang tepat, pengendali PI mampu memberikan performa yang cukup stabil dan akurat untuk sistem kemudi kendaraan otonom yang memerlukan kecepatan respon serta kestabilan arah dalam mengikuti jalur *waypoint* (Laoprom & Tunyasirut, 2020).



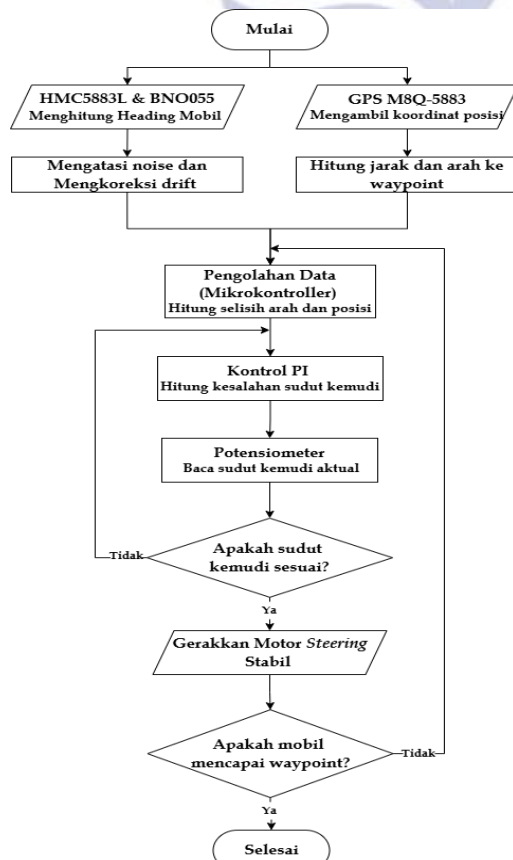
Gambar 2. Closed loop PI control system

METODE PENELITIAN

Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif yang berfokus pada pengumpulan dan analisis data numerik secara objektif untuk mengevaluasi kinerja kendali PI dalam sistem kemudi kendaraan otonom. Dalam implementasinya, pengendali PI dioptimalkan melalui proses tuning parameter yang dilakukan dengan empat metode, yaitu Tyreus-Luyben, Ziegler-Nichols, PID Tuner MATLAB, dan metode *Trial & Error* secara langsung. Keempat metode ini digunakan untuk menentukan nilai optimal dari parameter K_p dan K_i agar sistem kendali menghasilkan respons yang stabil, cepat, dan minim *overshoot* dalam mengikuti jalur *waypoint*.

Kerangka Konseptual



Gambar 3. Blok Kerangka Konseptual

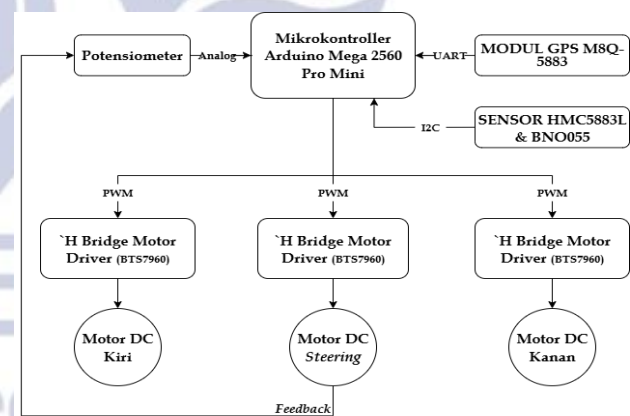


Gambar 4. Sistem Elektronika Prototipe Autonomous Car

Pada gambar 4 menunjukkan flowchart penelitian untuk model konseptual sistem navigasi pada *autonomous car* dirancang dengan mengintegrasikan mengintegrasikan sensor BNO055 dan HMC5883L untuk menentukan heading, serta GPS Ublox M8Q-5883 untuk posisi. Kombinasi sensor ini membantu sistem memperoleh arah dan lokasi kendaraan secara *real-time*, mengurangi kesalahan akibat *noise* dan *drift*.

Pengolahan data dilakukan oleh Arduino Mega 2560 Pro Mini, yang menghitung error arah dan mengendalikan motor steering melalui kontrol PI. Potensiometer digunakan sebagai umpan balik sudut kemudi. Motor DC sebagai aktuator dikendalikan oleh motor driver, dengan suplai daya dari baterai 12V dan DC to DC converter.

Perancangan Hardware



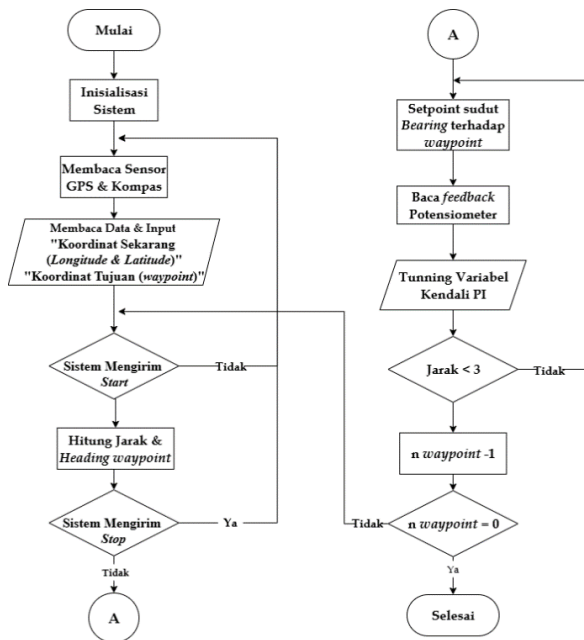
Gambar 5. Blok Sistem Hardware

Pada gambar 5 menggambarkan perancangan *hardware* sistem prototipe mengintegrasikan mikrokontroler Arduino Mega 2560 Pro Mini sebagai pusat kendali, yang memproses data dari sensor GPS M8Q-5883, BNO055, dan HMC5883L melalui antarmuka UART dan I2C. Tiga motor driver BTS7960 digunakan untuk mengendalikan dua motor DC roda belakang dan satu motor steering menggunakan sinyal PWM. Sistem steering dilengkapi potensiometer sebagai *feedback* sudut kemudi aktual, yang kemudian diolah dengan algoritma PI pada mikrokontroler untuk menghasilkan kontrol kemudi yang presisi dan responsif.

Sistem Navigasi Berbasis Modul Kompas & GPS Pada Prototipe *Autonomous Car* Dengan Pengendali *Steering* Metode PI

Perancangan Software

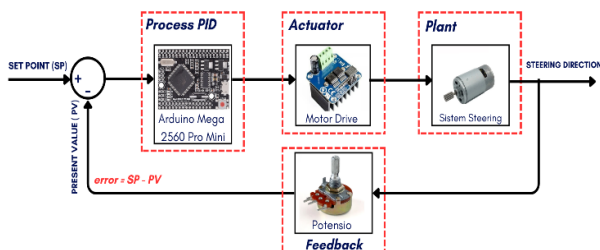
Perancangan *software* dimulai dengan inisialisasi sensor GPS dan kompas untuk membaca posisi dan arah kendaraan. Berdasarkan perhitungan arah dan jarak ke *waypoint*, sistem mengatur sudut kemudi menggunakan metode kendali PI. Selama bergerak, sistem memantau jarak ke *waypoint* dan akan mengurangi jumlah *waypoint* ketika jarak kurang dari 3 meter. Proses ini berlangsung iteratif hingga semua *waypoint* tercapai dan sistem berhenti secara otomatis.



Gambar 6. Blok Sistem Software

Sistem Kendali Kemudi

Dalam penelitian ini, sistem kendali arah (*heading*) kendaraan otonom dirancang menggunakan algoritma kontrol PI digital yang dijalankan oleh mikrokontroler Arduino Mega 2560 Pro Mini. Nilai *setpoint* diperoleh dari hasil perhitungan *bearing* menuju *waypoint*, yang dihitung menggunakan kombinasi data koordinat dari sensor GPS (Ublox M8Q-5883) dan arah *heading* dari kompas digital (BNO055 & HMC5883L). Nilai ini dibandingkan dengan *present value*, yaitu sudut kemudi aktual yang dibaca dari potensiometer, sehingga menghasilkan nilai *error*. Bersama dengan *last error* mikrokontroler menghitung koreksi kendali PI.



Gambar 7. Blok Sistem Kendali Kemudi

Sinyal kendali dari PI kemudian dikonversi menjadi sinyal PWM dan dikirim ke motor driver BTS7960. Motor driver ini mengatur tegangan input sebesar 0–12 V yang diberikan ke *plant*, yaitu motor DC penggerak kemudi. Tegangan ini secara langsung memengaruhi kecepatan dan arah putaran motor dalam memanuver roda depan. Dengan sistem loop tertutup ini, *feedback* dari potensiometer akan terus diperbarui sehingga sistem dapat menyesuaikan sudut kemudi secara presisi untuk menjaga kendaraan tetap pada jalur menuju *waypoint* secara stabil dan responsif.

HASIL DAN PEMBAHASAN

Hasil Penelitian

1. Pengujian Integrasi Sensor Kompas dan IMU

Tabel 1. Hasil Pengujian Integrasi Sensor Kompas dan IMU

| Iterasi | Heading (°) (Sebenarnya) | Heading (°) (Pengukuran) | Error (°) |
|---------|-----------------------------|-----------------------------|-----------|
| 1 | 0°/360° | 358° | 2° |
| 2 | 45° | 46° | 1° |
| 3 | 90° | 90° | 0° |
| 4 | 135° | 134° | 1° |
| 5 | 180° | 179° | 1° |
| 6 | 225° | 222° | 3° |
| 7 | 270° | 266° | 4° |
| 8 | 315° | 312° | 3° |

Pengujian integrasi sensor HMC5883L dan BNO055 bertujuan mengevaluasi keakuratan data *heading* yang digunakan pada sistem navigasi kendaraan otonom. Dari Tabel 1, diperoleh bahwa rata-rata *error* antara nilai *heading* sebenarnya dan hasil pengukuran sensor adalah sebesar 1,57°, dengan *error* maksimum sebesar 4° dan minimum 0°. Meskipun terdapat fluktuasi kecil, kombinasi sensor kompas dan IMU ini terbukti mampu menghasilkan data *heading* yang cukup stabil dan akurat, serta berada dalam batas toleransi yang dapat diterima untuk aplikasi navigasi.

2. Pengujian GPS Ublox SAM-M8Q-5883

Tabel 2. Hasil Pengujian Modul GPS SAM-M8Q-5883

| Waktu | Latitude | Longitude | Jarak (m) |
|--------------------|-----------|------------|-----------|
| 28/2/2025 15:52:37 | -7.317537 | 112.726119 | 0 |
| 28/2/2025 15:52:42 | -7.317535 | 112.726119 | 0.09 |
| 28/2/2025 15:52:48 | -7.317531 | 112.726119 | 0.57 |
| 28/2/2025 15:52:54 | -7.317532 | 112.726119 | 0.47 |
| 28/2/2025 15:53:00 | -7.317532 | 112.726119 | 0.47 |
| 28/2/2025 15:53:06 | -7.317536 | 112.726119 | 0 |
| 28/2/2025 15:53:12 | -7.317536 | 112.726119 | 0 |
| 28/2/2025 15:53:18 | -7.317538 | 112.726119 | 0.28 |
| 28/2/2025 15:53:24 | -7.317542 | 112.726119 | 0.66 |
| 28/2/2025 15:53:30 | -7.317542 | 112.726119 | 0.66 |
| 28/2/2025 15:53:37 | -7.317552 | 112.726119 | 1.71 |

Pengujian modul GPS Ublox SAM-M8Q-5883 menunjukkan bahwa terdapat pergeseran (*drift*) posisi meskipun kendaraan dalam keadaan diam. Rata-rata *drift* mencapai sekitar 0,4 meter, dengan nilai maksimum sebesar 1.8 meter yang masih dalam batas toleransi untuk

sistem navigasi non-presisi tinggi. *Longitude* terpantau tetap, sedangkan variasi lebih besar terjadi pada *latitude* karena sensitivitas lokasi pengujian.

Fluktuasi nilai drift terlihat meningkat seiring waktu, terutama setelah detik ke-50, akibat akumulasi noise dan perubahan sinyal satelit. Hal ini menunjukkan bahwa penggunaan sensor GPS ini ideal untuk durasi pendek atau dikombinasikan dengan sensor korektif tambahan seperti kompas atau IMU untuk meningkatkan stabilitas posisi.

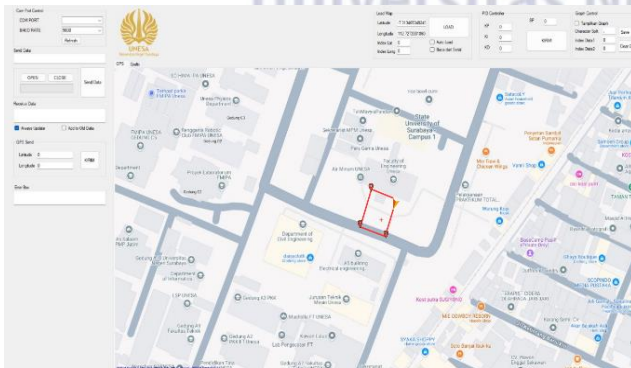
3. Pengujian Sistem Navigasi Waypoint

Pengujian sistem navigasi dilakukan untuk mengevaluasi performa prototipe mobil otonom dalam mengikuti rute *square* dengan empat *waypoint* melewati *obstacle* pembatas jalan. Koordinat *waypoint* diambil dari Google Maps, kemudian diproses menggunakan *Haversine Formula* untuk menghitung jarak dan *Bearing Formula* untuk menentukan arah heading. Pengujian dilakukan sebanyak sembilan kali selama tiga hari dengan pencatatan koordinat GPS secara *real-time* saat kendaraan bergerak, guna menilai akurasi navigasi serta kestabilan sistem kendali PI berbasis integrasi sensor GPS dan kompas. Titik-titik *waypoint* tersebut disajikan seperti pada Gambar 8.



Gambar 8. Titik Koordinat Waypoint Rute di FT Unesa

Serta visualisasi pada *Graphical User Interface* (GUI) yang telah dibuat sebagai sarana pengambilan data *heading* dan koordinat ketika prototipe *autonomous car* berjalan dapat dilihat pada Gambar 9.



Gambar 9. Tampilan GUI Rute Prototipe

Pengujian sistem navigasi dilakukan untuk menilai kemampuan mobil otonom dalam mencapai empat *waypoint* (WP 1, WP 2, WP 3, dan WP 4) yang telah

ditentukan melalui GUI, dengan penyesuaian berbeda setiap harinya berdasarkan kondisi lapangan dan akurasi sinyal GPS guna menghindari obstacle. Data uji yang dianalisis berasal dari percobaan yang berhasil, yakni saat mobil menyelesaikan lintasan tanpa menabrak halangan, menuju *waypoint* meskipun terdapat sedikit penyimpangan lintasan.

Perhitungan error ini dilakukan dengan menggunakan formula *Haversine* persamaan (1). Sebagai contoh, pada iterasi ke-7 untuk *waypoint* 1 (WP 1), mobil berhenti di koordinat (*Lat*: -7.3159903; *Long*: 112.7269583), sedangkan titik WP1 berada pada (*Lat*: -7.3160361, *Long*: 112.7269583). Karena nilai *longitude* sama, maka perubahan hanya terjadi pada *latitude*. Dengan menggunakan *Formula Haversine* di atas:

$$\begin{aligned}\Delta\phi &= -7.3160361 - (-7.3159903) \\ &= -0.0000458^\circ \\ &= -7.995 \times 10^{-7} \text{ radian}\end{aligned}$$

$$a = \sin^2\left(\frac{-7.995 \times 10^{-7}}{2}\right) \approx 1.598 \times 10^{-13} \quad (8)$$

$$c = 2 \cdot \text{atan} 2(\sqrt{a}, \sqrt{1-a}) \approx 2.526 \times 10^{-6}$$

$$\begin{aligned}d &= R \cdot c \\ &= 6,371,000 \cdot 2.526 \times 10^{-6} \approx 5.10 \text{ meter}\end{aligned}$$

Tabel 3. Data Hasil Pengujian

| Iterasi ke- | Kesalahan (meter) | | | | Rata-rata Error (m) |
|-----------------------|-------------------|------|------|------|---------------------|
| | WP 1 | WP 2 | WP 3 | WP 4 | |
| 1 | 0.56 | 0.44 | 1.05 | 2.22 | 1.06 |
| 2 | 0.11 | 0.88 | 1.00 | 1.11 | 0.77 |
| 3 | 1.27 | 2.02 | 1.47 | 2.68 | 1.86 |
| 4 | 1.77 | 0.85 | 1.76 | 4.19 | 2.14 |
| 5 | 1.97 | 0.76 | 0.83 | 0.59 | 1.03 |
| 6 | 1.34 | 0.89 | 1.38 | 3.61 | 1.80 |
| 7 | 5.10 | 1.59 | 2.21 | 3.67 | 3.14 |
| 8 | 2.65 | 1.68 | 1.44 | 0.96 | 1.68 |
| 9 | 0.14 | 0.84 | 1.48 | 0.52 | 0.74 |
| Rata-rata Total Error | | | | | 1.58 |

Penilaian akurasi navigasi dilakukan menggunakan metode *Mean Absolute Error* (MAE), yang menghitung rata-rata selisih absolut antara posisi aktual kendaraan dan titik *waypoint*. Metode ini memberikan gambaran umum terhadap deviasi posisi dalam setiap pengujian. Rumus MAE yang digunakan dalam konteks pengujian ini adalah:

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (9)$$

Keterangan :

e_i = rata-rata error pada iterasi ke- i

n = jumlah total iterasi

MAE = rata-rata total error dari semua iterasi

Sistem Navigasi Berbasis Modul Kompas & GPS Pada Prototipe *Autonomous Car* Dengan Pengendali *Steering* Metode PI

Berdasarkan data hasil pengujian yang tercantum dalam Tabel 3, nilai rata-rata *error* dari pada iterasi ke-9 dapat dihitung untuk memperoleh sampel gambaran akurasi sistem pada satu percobaan prototipe *autonomous car* dalam melewati rute *square*. Berikut adalah contoh perhitungan MAE :

$$MAE = \left(\frac{0.14 + 0.84 + 1.48 + 0.52}{4} \right) \quad (8)$$

$MAE = 0.74$ meter

Dari hasil pengujian sistem navigasi berdasarkan kesalahan posisi terhadap empat *waypoint* dalam sembilan iterasi. Rata-rata *error* keseluruhan adalah 1.58 meter, dengan kesalahan terkecil 0.11 meter dan terbesar 5.10 meter. Iterasi ke-9 mencatat akurasi terbaik (0.74 m), sedangkan iterasi ke-7 mengalami *error* tertinggi (3.14 m), yang disebabkan oleh keterbatasan akurasi dan delay pembacaan dari GPS Ublox SAM-M8Q-5883. Meski terdapat fluktuasi posisi, sistem tetap dapat menjalankan fungsi navigasi dengan baik pada skala prototipe.

Pembahasan

Pada tahap pembahasan ini, analisis difokuskan pada hasil pengujian sistem navigasi mobil otonom berdasarkan data pergerakan kendaraan menuju titik-titik *waypoint* yang telah ditentukan.



Gambar 10. Hasil Pengujian Sistem Navigasi

Selama pengujian, koordinat GPS direkam secara *real-time* dan dikirimkan ke GUI melalui komunikasi ESP-NOW, lalu disimpan dalam format teks untuk diolah lebih lanjut menggunakan Ms.Excel. Visualisasi lintasan dilakukan menggunakan Google Earth Pro, seperti terlihat pada Gambar 10, yang menunjukkan salah satu hasil percobaan pergerakan mobil di area Fakultas Teknik UNESA. Meskipun terdapat penyimpangan jalur dari rute ideal akibat keterbatasan

akurasi sensor GPS, secara umum mobil mampu mencapai keempat *waypoint* dengan cukup baik.

Keberhasilan sistem dalam mengikuti lintasan juga didukung oleh pemilihan parameter kontrol PI yang optimal, yaitu $K_p = 30$ dan $K_i = 0.005$, yang diperoleh melalui metode *trial & error* langsung di lapangan. Pendekatan ini dinilai paling sesuai karena mempertimbangkan faktor nyata seperti gesekan, respons motor, serta ketidaksempurnaan mekanik seperti pemasangan potensiometer yang kurang presisi.

Sebelumnya, metode klasik seperti *Tyresus-Luyben*, *Ziegler-Nichols*, dan *PID Tuner MATLAB* digunakan sebagai referensi awal untuk menentukan nilai K_p melalui pendekatan osilasi, di mana K_u (*Ultimate Gain*) adalah gain maksimum yang menyebabkan sistem mulai berosilasi terus-menerus, dan P_u (*Ultimate Period*) adalah periode osilasi dari puncak ke puncak. Meskipun berguna sebagai acuan awal, nilai terbaik tetap didapat dari pengujian langsung yang menunjukkan kestabilan dan keakuratan yang lebih tinggi dalam kondisi operasional nyata.

PENUTUP Simpulan

Sistem navigasi berbasis GPS dan kompas berhasil diimplementasikan pada prototipe kendaraan otonom dengan kendali *steering* PI yang terbukti efektif dalam mengikuti jalur *waypoint*. Dengan integrasi sensor HMC5883L, BNO055, dan GPS SAM-M8Q, sistem mencatat rata-rata *error* heading 1.57° dan drift GPS 0.4 meter. Dari sembilan percobaan selama tiga hari, diperoleh rata-rata *error* posisi 1.58 meter. Nilai parameter PI terbaik ($K_p = 30$, $K_i = 0.005$) diperoleh melalui metode *trial & error*, yang lebih adaptif terhadap kondisi riil seperti gesekan dan ketidaksempurnaan mekanik, dibanding metode klasik seperti *Tyresus-Luyben* dan *Ziegler-Nichols* yang hanya digunakan sebagai acuan awal dengan pendekatan osilasi (K_u dan P_u). Hasil ini menunjukkan bahwa sistem mampu bekerja secara stabil, akurat, dan layak untuk dikembangkan lebih lanjut.

Saran

Untuk peningkatan akurasi dan kemampuan adaptif sistem, disarankan penggunaan GPS berpresisi tinggi (RTK/DGPS), penambahan sensor *vision*, LiDAR, ultrasonik, serta odometri sebagai pelengkap saat sinyal GPS lemah. Pengembangan algoritma penghindaran rintangan seperti VFH, DWA, atau RRT, serta integrasi logika fuzzy, A*, dan *Extended Kalman Filter* (EKF) akan meningkatkan kecerdasan dan keandalan navigasi. Meskipun PI cukup andal untuk sistem sederhana, metode PID tetap direkomendasikan sebagai pengendali utama untuk skenario dinamis karena kemampuannya merespons perubahan dengan lebih cepat dan stabil.

DAFTAR PUSTAKA

- Ahmad Sulkhan Taufik. (2013). Sistem Navigasi Waypoint pada Autonomous Mobile Robot. *Journal Article // Jurnal Mahasiswa Teknik Elektro Universitas Brawijaya*. <https://download.garuda.kemdikbud.go.id/article.php?article=187359&val=6454&title=Sistem%20Navigasi%20Waypoint%20pada%20Autonomous%20Mobile%20Robot#:~:text=Sistem%20navigasi%20waypoint%20dirancang%20agar,yang%20telah%20ditentukan%20oleh%20operator>.
- Ashar Ashari, E. S. D. S. (2020). Sistem Navigasi Waypoint Pada Robot Beroda Berdasarkan Global Positioning System Dan Filter Kalman. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol.4 No.7(e-ISSN: 2548-964X), 2075–2082. <http://j-ptiik.ub.ac.id>
- Christensen, H., Paz, D., Zhang, H., Meyer, D., Xiang, H., Han, Y., Liu, Y., Liang, A., Zhong, Z., & Tang, S. (2021). Autonomous vehicles for micro-mobility. *Autonomous Intelligent Systems*, 1(1). <https://doi.org/10.1007/s43684-021-00010-2>
- Katsuhiko Ogata. (2017). *Modern control engineering. In Modern Control Engineering*.
- Ketsayom, S., Maneetham, D., Aung, M. M., & Srichaipanya, W. (2023). Enhancing Outdoor Warehouse Operations: Design and Development of GPS and Compass-Guided Automatic Vehicles. *2023 11th International Conference on Cyber and IT Service Management, CITSM 2023*. <https://doi.org/10.1109/CITSM60085.2023.10455340>
- Laoprom, I., & Tunyasirirut, S. (2020). Design of PI Controller for Voltage Controller of Four-Phase Interleaved Boost Converter Using Particle Swarm Optimization. *Journal of Control Science and Engineering*, 2020. <https://doi.org/10.1155/2020/9515160>
- Olayode, I. O., Du, B., Severino, A., Campisi, T., & Alex, F. J. (2023). Systematic literature review on the applications, impacts, and public perceptions of autonomous vehicles in road transportation system. In *Journal of Traffic and Transportation Engineering (English Edition)* (Vol. 10, Issue 6, pp. 1037–1060). KeAi Communications Co. <https://doi.org/10.1016/j.jtte.2023.07.006>
- Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., Rus, D., & Ang, M. H. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1). <https://doi.org/10.3390/machines5010006>
- Saputra Kurnia, Nazaruddin, Yunardi Dalila Husna, & Andriyani Renny. (2019). *Proceedings, IEEE CyberneticsCom 2019: 2019 IEEE International Conference on Cybernetics and Computational Intelligence: "Towards a Smart and Human-Centered Cyber World": August 22-24, 2019 - Banda Aceh, Indonesia*. IEEE.
- Song, X., Gao, H., Ding, T., Gu, Y., Liu, J., & Tian, K. (2023). A Review of the Motion Planning and Control Methods for Automated Vehicles. In *Sensors* (Vol. 23, Issue 13). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/s23136140>
- Ušinskis, V., Makulavičius, M., Petkevičius, S., Dziedzickis, A., & Bučinskas, V. (2024). Towards Autonomous Driving: Technologies and Data for Vehicles-to-Everything Communication. In *Sensors* (Vol. 24, Issue 11). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/s24113411>
- Yadav, A., Gaur, A., Jain, M., Chaturvedi, D. K., & Sharma, R. (2020). NSES 2018 Development Navigation, Guidance & Control Program for GPS based Autonomous Ground Vehicle (AGV) using Soft Computing Techniques-review under responsibility of National Conference on "Smart Materials: Energy and Environment for Smart Cities (NSES-2018)". In *Materials Today: Proceedings* (Vol. 29). www.sciencedirect.comwww.materialstoday.com/proceedings