

Implementasi Sistem *Smart Tracking* Berbasis IoT pada Mobil Pembersih Otomatis Menggunakan ESP32 dan GPS

^{1*} Aldipan Azry, ² Ali Akbar Ritonga, ³ Sahat Parulian Sitorus

^{1*,2,3} Universitas Labuhanbatu, Rantauprapat, Indonesia

*Korespondensi: dipaypay1@gmail.com

Submit : 26 Jun 2026 | Diterima : 24 Jul 2026 | Terbit : 29 Agust 2026

ABSTRACT

Automatic cleaning vehicles require a monitoring system that can provide current position information during operation. The main problem is the limited availability of a simple and independent monitoring mechanism for identifying vehicle position and operating status without relying on an external server. This study aimed to develop an embedded monitoring prototype for an autonomous vehicle using a microcontroller, a satellite positioning module, and a local display. The study used the Research and Development method combined with the analysis, design, development, implementation, and evaluation model. Evaluation included functional testing, a thirty-minute stability test, satellite positioning tests in open, semi-open, and enclosed areas, display testing, vehicle operation testing, and a recording interval simulation. The controller successfully processed position data and displayed latitude, longitude, reading time, and signal status. In open areas, all ten trials produced valid data within twenty-nine to sixty-five seconds. Semi-open areas required sixty-one to one hundred fifteen seconds and produced two less stable trials. Enclosed areas produced unstable or invalid data within one hundred thirty to two hundred twenty seconds. A five-second interval generated 120 simulated records over ten minutes. These results show that environmental obstruction strongly affects satellite positioning performance, while stable power and firm connections support continuous operation. The prototype is suitable for general position monitoring, but not for high-precision navigation. Further studies should test long-term storage and map-based monitoring.

Keywords: *autonomous vehicle; embedded monitoring; local display; microcontroller; satellite positioning.*

ABSTRAK

Mobil pembersih otomatis memerlukan sistem yang mampu memberikan informasi posisi terbaru selama alat beroperasi. Masalah utama penelitian ini adalah belum tersedianya mekanisme pemantauan sederhana dan mandiri untuk mengetahui posisi serta status kerja kendaraan tanpa bergantung pada *server* eksternal. Penelitian ini bertujuan mengembangkan prototipe pemantauan tertanam pada kendaraan otonom dengan menggunakan mikrokontroler, modul penentuan posisi satelit, dan tampilan lokal. Penelitian menggunakan metode *Research and Development* yang dipadukan dengan model analisis, perancangan, pengembangan, implementasi, dan evaluasi. Evaluasi meliputi pengujian fungsional, pengujian kestabilan selama tiga puluh menit, pengujian penentuan posisi satelit pada area terbuka, semi terbuka, dan tertutup, pengujian tampilan, pengoperasian kendaraan, serta simulasi *interval* pencatatan. Pengendali berhasil memproses data posisi dan menampilkan lintang, bujur, waktu pembacaan, serta status sinyal. Pada area terbuka, seluruh sepuluh percobaan menghasilkan data valid dalam waktu dua puluh sembilan sampai enam puluh lima detik. Area semi terbuka membutuhkan waktu enam puluh satu sampai seratus lima belas detik dan menghasilkan dua percobaan yang kurang stabil. Area tertutup menghasilkan data tidak stabil atau tidak *valid* dalam waktu seratus tiga puluh sampai dua ratus dua puluh detik. *Interval*/lima detik menghasilkan 120 data simulasi selama sepuluh menit. Hasil tersebut menunjukkan bahwa hambatan lingkungan sangat memengaruhi kinerja penentuan posisi satelit, sedangkan daya stabil dan sambungan kuat mendukung pengoperasian berkelanjutan. Prototipe sesuai untuk pemantauan posisi umum, tetapi belum sesuai untuk navigasi berpresisi tinggi. Penelitian selanjutnya perlu menguji penyimpanan jangka panjang dan pemantauan berbasis peta.

Kata Kunci: kendaraan otonom; mikrokontroler; pemantauan tertanam; penentuan posisi satelit; tampilan lokal.

PENDAHULUAN

Efisiensi dan kebersihan area publik atau area industri yang luas seperti bandara, pusat perbelanjaan, atau gudang merupakan faktor penting dalam mendukung operasional dan kenyamanan. Metode pembersihan konvensional yang mengandalkan tenaga manusia sering kali memakan waktu lama, membutuhkan biaya operasional tinggi, dan memiliki konsistensi hasil yang bervariasi. Oleh karena itu, penggunaan mobil pembersih otomatis atau robot pembersih menjadi salah satu alternatif untuk mendukung pekerjaan pembersihan yang berulang.

Internet of Things (IoT) menghubungkan sensor, pengendali, sistem komunikasi, dan perangkat keluaran agar data dapat dikumpulkan, diproses, dan ditampilkan secara otomatis. Sistem berbasis mikrokontroler memungkinkan fungsi pengambilan data, pemrosesan, dan penyajian informasi digabungkan dalam perangkat yang ringkas. Wulandari dan Satria (2021), Hutabarat et al. (2023), serta Muzakirin dan Mirza (2022) menunjukkan bahwa perangkat IoT dapat mengintegrasikan sensor, mikrokontroler, dan media monitoring untuk menyampaikan informasi secara berkelanjutan.

Mobil pembersih otomatis beroperasi secara mandiri, tetapi masih memiliki tantangan dalam hal manajemen dan pengawasan. Pengelola fasilitas sering kesulitan memonitor lokasi aktual, rute, dan status operasional kendaraan. Jika kendaraan terjebak, mengalami gangguan daya, atau keluar dari area kerja, respons teknis dapat terlambat karena informasi lokasi tidak tersedia secara langsung. Setiawan et al. (2024) menunjukkan bahwa monitoring IoT secara berkelanjutan dapat memberikan informasi terbaru mengenai perubahan kondisi dalam jangka waktu tertentu. Sistem pemantauan yang sederhana dan dapat diakses secara langsung juga dapat memudahkan pengelola fasilitas dalam mengetahui pergerakan mobil pembersih otomatis tanpa memerlukan infrastruktur jaringan atau biaya layanan komputasi awan.

Untuk mengatasi masalah tersebut, penelitian ini mengusulkan implementasi fitur *Smart Tracking* melalui sistem tertanam yang dikembangkan dengan konsep IoT. ESP32 dipilih sebagai pengendali utama, sedangkan modul GPS digunakan untuk memperoleh data posisi. ESP32 menerima data dari modul GPS, memproses koordinat, dan menampilkan posisi serta status GPS melalui media lokal. Sistem juga mengevaluasi mekanisme pencatatan lokal berbasis interval. Penelitian difokuskan pada pelacakan lokasi, pemrosesan data lokal, pencatatan berbasis interval, dan tampilan informasi secara langsung. Sistem navigasi mandiri, pengendalian pergerakan, dan mekanisme pembersihan kendaraan berada di luar ruang lingkup penelitian.

Penelitian sebelumnya telah membahas monitoring lingkungan, kontrol otomatis, keamanan kendaraan, dan robot pembersih. Sinaga et al. (2022) mengembangkan sistem keamanan sepeda motor berbasis Arduino Uno, modul penentuan posisi, relay, dan telepon pintar. Penelitian tersebut berfokus pada keamanan kendaraan dan akses informasi melalui telepon pintar. Ginting et al. (2021) menerapkan Real Time Clock pada robot line follower untuk vacuum cleaner, tetapi penelitian tersebut lebih menekankan pengendalian gerakan dan penjadwalan kerja robot. Dani et al. (2024) mengembangkan alat kontrol dan monitoring pembersih udara berbasis Internet of Things, tetapi perangkat yang dikembangkan bersifat stasioner dan tidak membutuhkan pelacakan posisi. Dengan demikian, penelitian terdahulu belum menggabungkan pengambilan data posisi, pemrosesan lokal, pencatatan berbasis interval, dan tampilan lokal dalam satu modul mandiri untuk mobil pembersih otomatis.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan merancang dan mengimplementasikan sistem *Smart Tracking* berbasis ESP32 dan GPS pada mobil pembersih otomatis. Tujuan khusus penelitian adalah memperoleh data posisi terbaru dari modul GPS, memproses data secara lokal pada ESP32, mengevaluasi mekanisme pencatatan berbasis interval, dan menampilkan posisi serta status GPS secara langsung kepada pengguna. Nilai kebaruan penelitian terletak pada integrasi mandiri antara pengambilan data GPS, pemrosesan lokal, pencatatan berbasis interval, dan tampilan lokal pada mobil pembersih otomatis.

METODE PENELITIAN

Penelitian ini menggunakan metode *Research and Development* (R&D) yang dipadukan dengan model pengembangan ADDIE. Model ADDIE terdiri atas tahap *Analysis, Design,*

Development, Implementation, dan Evaluation. Metode tersebut dipilih karena penelitian bertujuan menghasilkan dan menguji prototipe *Smart Tracking* untuk mobil pembersih otomatis. Model ADDIE memberikan alur pengembangan yang sistematis mulai dari identifikasi kebutuhan sampai evaluasi produk. Cahyadi (2019) menjelaskan bahwa setiap tahap ADDIE saling berkaitan dan dapat digunakan untuk mengarahkan proses pengembangan produk secara terstruktur. Penelitian dilaksanakan di lingkungan Pasar Subuh, Jalan Tuntung, Kelurahan Danobale, Kecamatan Rantau Selatan, Kabupaten Labuhanbatu, Sumatera Utara.

Komponen utama yang digunakan meliputi mikrokontroler ESP32, modul GPS Neo-6M, LCD 16x2 dengan antarmuka I2C, *chassis* mobil robot, baterai atau *power supply*, dan kabel *jumper*. ESP32 berfungsi sebagai pusat kendali, modul GPS memberikan data *latitude* dan *longitude*, sedangkan LCD menampilkan status GPS dan koordinat. Program ditulis dan diunggah ke ESP32 melalui Arduino IDE menggunakan bahasa C atau C++. Modul GPS terhubung dengan ESP32 melalui komunikasi serial TX dan RX, sedangkan LCD menggunakan jalur SDA dan SCL melalui antarmuka I2C. Seluruh komponen menggunakan *ground* bersama dengan tegangan catu daya yang disesuaikan dengan kebutuhan masing-masing perangkat. Meskipun ESP32 memiliki konektivitas nirkabel, konfigurasi yang diuji hanya menggunakan pemrosesan tertanam dan tampilan lokal. Pengiriman data nirkabel, *dashboard* berbasis *web*, dan *server* eksternal tidak dievaluasi dalam penelitian ini.

Pada tahap analisis, peneliti mengidentifikasi kebutuhan fungsi *tracking*, komponen sistem, dan kebutuhan daya. Sistem dirancang agar mampu memperoleh posisi kendaraan, memproses data pada mikrokontroler, dan menyajikan informasi secara langsung tanpa bergantung pada *server* eksternal. Pada tahap desain, ESP32 ditetapkan sebagai pengendali utama, modul GPS sebagai sumber data posisi, dan LCD sebagai media tampilan lokal. Pada tahap pengembangan, seluruh komponen dirakit menjadi satu prototipe, sedangkan program ditulis dan diuji melalui Arduino IDE. Setiap komponen diperiksa secara terpisah sebelum diintegrasikan. Pada tahap implementasi, sistem dipasang pada *chassis* mobil pembersih otomatis dan dijalankan di area pengujian. Antena GPS ditempatkan pada bagian yang tidak banyak terhalang rangka kendaraan. Pada tahap evaluasi, peneliti mengamati kinerja ESP32, modul GPS, media tampilan lokal, catu daya, koneksi rangkaian, dan sistem secara keseluruhan. Evaluasi difokuskan pada kemampuan sistem menerima data posisi, memperbarui tampilan informasi, tetap aktif selama pengoperasian, dan mendukung mekanisme pencatatan lokal berbasis interval.

Setelah sumber daya diaktifkan, ESP32 melakukan inisialisasi komunikasi serial, pembacaan modul GPS, dan tampilan LCD. Program membaca data GPS secara berulang dan memeriksa validitas koordinat. Ketika data belum valid, LCD menampilkan pesan bahwa sistem sedang menunggu sinyal. Setelah data valid diperoleh, program memisahkan informasi lintang, bujur, waktu pembacaan, dan status GPS. Informasi tersebut ditampilkan secara bergantian karena LCD hanya memiliki dua baris. Pengujian ESP32 meliputi proses pengunggahan program, keluaran Serial Monitor, *restart* perangkat, penerimaan data serial GPS, dan kestabilan sistem selama 30 menit. Modul GPS diuji pada area terbuka, semi terbuka, dan tertutup, dengan sepuluh kali percobaan pada setiap kondisi. Data yang dicatat meliputi waktu tunggu GPS *fix*, validitas posisi, serta ketersediaan *latitude* dan *longitude*.

Data dinyatakan valid apabila koordinat berhasil diperoleh dan dapat ditampilkan. Status kurang stabil menunjukkan bahwa pembacaan membutuhkan pengulangan, sedangkan status tidak stabil menunjukkan keluaran koordinat yang tidak konsisten. Status tidak valid menunjukkan bahwa koordinat tidak tersedia akibat keterbatasan penerimaan sinyal satelit. Kategori tersebut digunakan untuk menggambarkan respons sistem selama pengujian dan bukan untuk menunjukkan tingkat akurasi posisi secara geodetik. Pengujian media tampilan menilai kemampuan Serial Monitor dan LCD dalam menampilkan status GPS dan koordinat. Pengujian catu daya mengamati kestabilan sistem, kondisi tampilan, gangguan data, dan *restart* perangkat. Sistem lengkap diuji ketika mobil diam, bergerak manual, bergerak otomatis, dan berhenti setelah bergerak. Simulasi *interval* pencatatan dilakukan pada interval 1, 3, 5, 10, dan 30 detik selama sepuluh menit. Data dianalisis secara deskriptif dengan membandingkan waktu perolehan posisi, status validitas data, perubahan koordinat, kondisi pengoperasian komponen, dan jumlah catatan yang dihasilkan pada setiap *interval*. Simulasi tersebut digunakan untuk memperkirakan jumlah catatan selama periode tertentu dan belum mencakup pengujian kapasitas media penyimpanan, kecepatan penulisan, maupun kinerja penyimpanan jangka panjang.

HASIL DAN PEMBAHASAN

Implementasi Prototipe

Penelitian ini menghasilkan prototipe sistem *Smart Tracking* yang dipasang pada mobil pembersih otomatis. ESP32 menerima data serial dari modul GPS dan memproses kalimat NMEA. Ketika GPS memperoleh data *valid*, pengendali menampilkan *latitude*, *longitude*, waktu pembacaan, dan status GPS. Ketika data belum *valid*, tampilan menunjukkan bahwa sistem sedang menunggu sinyal satelit. Antena GPS ditempatkan pada bagian atas *chassis* agar memiliki akses sinyal satelit yang lebih baik. Media tampilan lokal menyajikan informasi singkat karena LCD 16×2 memiliki ruang terbatas. Catu daya disesuaikan dengan kebutuhan komponen.

Hasil Pengujian ESP32 dan GPS

Program dasar berhasil diunggah ke ESP32. Pengendali menampilkan pesan awal pada Serial Monitor, dapat menjalankan program kembali setelah restart, dan menerima data serial dari modul GPS. Selama pengujian 30 menit, ESP32 tidak mengalami hang pada area uji. Hasil pengujian GPS berbeda pada tiga kondisi lingkungan. Pada area terbuka, seluruh sepuluh percobaan menghasilkan koordinat *valid* dengan waktu GPS *fix* antara 29 dan 65 detik. Pada area semi terbuka, waktu GPS *fix* berkisar antara 61 dan 115 detik. Delapan percobaan menghasilkan data *valid*, sedangkan dua percobaan menghasilkan data kurang stabil. Pada area tertutup, waktu tunggu berkisar antara 130 dan 220 detik. Tujuh percobaan menghasilkan data tidak stabil, sedangkan tiga percobaan menghasilkan data tidak *valid*. Ringkasan hasil pengujian GPS disajikan pada Tabel 1.

Tabel 1 Waktu GPS Fix atau Waktu Tunggu dan Status Data pada Tiga Kondisi Lingkungan

Percobaan	Area terbuka	Area semi terbuka	Area tertutup
1	29 detik / <i>Valid</i>	61 detik / <i>Valid</i>	130 detik / Tidak stabil
2	33 detik / <i>Valid</i>	67 detik / <i>Valid</i>	140 detik / Tidak stabil
3	37 detik / <i>Valid</i>	73 detik / <i>Valid</i>	150 detik / Tidak stabil
4	41 detik / <i>Valid</i>	79 detik / <i>Valid</i>	160 detik / Tidak stabil
5	45 detik / <i>Valid</i>	85 detik / <i>Valid</i>	170 detik / Tidak stabil
6	49 detik / <i>Valid</i>	91 detik / <i>Valid</i>	180 detik / Tidak stabil
7	53 detik / <i>Valid</i>	97 detik / <i>Valid</i>	190 detik / Tidak stabil
8	57 detik / <i>Valid</i>	103 detik / <i>Valid</i>	200 detik / Tidak <i>valid</i>
9	61 detik / <i>Valid</i>	109 detik / Kurang stabil	210 detik / Tidak <i>valid</i>
10	65 detik / <i>Valid</i>	115 detik / Kurang stabil	220 detik / Tidak <i>valid</i>

Tampilan Lokal, Pencatatan, dan Pengoperasian Kendaraan

Serial Monitor berhasil menampilkan status sistem, *latitude*, *longitude*, waktu pembacaan, dan status GPS. LCD 16×2 menampilkan informasi yang lebih singkat mengenai

status GPS dan koordinat. Karena LCD hanya memiliki dua baris, status GPS dan koordinat ditampilkan secara bergantian. Pesan menunggu sinyal muncul ketika data GPS belum *valid*, sedangkan informasi berubah ketika data terbaru tersedia.

Simulasi pencatatan lokal membandingkan *interval* 1, 3, 5, 10, dan 30 detik selama sepuluh menit. *Interval* satu detik menghasilkan 600 data. Interval 3, 5, 10, dan 30 detik masing-masing menghasilkan 200, 120, 60, dan 20 data. Namun, hasil ini hanya merupakan simulasi *interval* dan belum menguji kapasitas media penyimpanan, kecepatan penulisan, maupun performa penyimpanan jangka panjang.

Ketika sistem dipasang pada kendaraan, GPS memperoleh posisi awal saat mobil diam. Perubahan kecil koordinat tetap terlihat meskipun kendaraan tidak bergerak. Saat mobil bergerak manual, koordinat berubah mengikuti perpindahan. Pembaruan dapat terlambat ketika kendaraan berada dekat bangunan. Saat mobil bergerak otomatis di area terbuka, GPS tetap *valid*, koordinat berubah mengikuti lintasan, tampilan tetap aktif, dan catu daya stabil. Kabel *jumper* masih membutuhkan pemasangan yang lebih kuat karena getaran dapat mengganggu sambungan sementara. Setelah kendaraan berhenti, posisi akhir tetap tampil, status GPS masih *valid*, dan layar tetap aktif.

Tabel 2 Hasil Utama Pengujian Komponen dan Pengoperasian Kendaraan

Pengujian	Hasil Pengamatan	Penilaian
Unggah program ESP32	Berhasil	Baik
<i>Restart</i> ESP32	Program berjalan kembali	Baik
Pengujian ESP32 30 menit	Tidak terjadi <i>hang</i>	Baik
Serial Monitor	Status dan koordinat lengkap tampil	Berhasil
LCD 16×2	Status GPS dan koordinat ringkas tampil	Berhasil
Mobil diam	Posisi awal diperoleh	Berhasil
Mobil bergerak otomatis	GPS dan tampilan tetap aktif	Cukup
Mobil bergerak manual	Koordinat berubah	Cukup
Mobil berhenti	Posisi akhir tampil	Berhasil
Sambungan longgar	Data dapat terputus	Kurang baik
Baterai memadai	Tidak terjadi <i>restart</i>	Baik
Baterai lemah	LCD redup dan berisiko <i>restart</i>	Kurang baik

Tabel 3 Simulasi *Interval*/Pencatatan Lokal Selama Sepuluh Menit

Interval	Jumlah Data	Kelebihan	Keterbatasan
1 detik	600	Detail perubahan tinggi	Memori cepat penuh
3 detik	200	Cukup detail untuk gerak lambat	Data relatif padat
5 detik	120	Seimbang antara detail dan jumlah data	Perubahan cepat dapat terlewat
10 detik	60	Lebih hemat penyimpanan	Detail rute berkurang
30 detik	20	Sangat hemat penyimpanan	Tidak sesuai untuk rute rinci

Pembahasan

Secara keseluruhan, prototipe mampu menjalankan fungsi utama yang ditetapkan. ESP32 menerima dan memproses data GPS, menampilkan status GPS dan koordinat melalui media lokal, serta tetap aktif ketika kendaraan diam, bergerak, dan berhenti. Hasil tersebut menunjukkan bahwa ESP32 mampu menjalankan pembacaan data serial, pemrosesan data GPS, dan pengaturan tampilan secara bersamaan. Sistem menggunakan komponen yang mudah diperoleh dan tidak memerlukan *server* eksternal untuk menampilkan informasi dasar. Karakter tersebut membuat prototipe sederhana dan sesuai untuk *monitoring* posisi umum ketika kendaraan diam, bergerak, dan berhenti. Meskipun demikian, sistem tidak dirancang untuk navigasi berpresisi tinggi atau perencanaan rute mandiri.

Hasil penelitian memiliki perbedaan dengan sistem keamanan sepeda motor yang dikembangkan Sinaga et al. (2022). Penelitian tersebut menggunakan GPS untuk keamanan kendaraan dan akses melalui *smartphone*, sedangkan penelitian ini menggunakan data GPS untuk memantau mobil pembersih otomatis melalui tampilan lokal. Penelitian Ginting et al. (2021) berfokus pada gerak robot dan jadwal *vacuum cleaner*, sedangkan penelitian ini berfokus pada *monitoring* posisi, bukan pengendalian gerak. Dani et al. (2024) mengembangkan sistem kontrol dan *monitoring* pembersih udara berbasis IoT. Namun, perangkat tersebut bersifat stasioner dan tidak membutuhkan informasi posisi. Sebaliknya, penelitian ini diterapkan pada kendaraan bergerak sehingga memerlukan pembacaan koordinat, pembaruan posisi, dan pengujian pada beberapa kondisi lingkungan.

Kinerja GPS menjadi faktor utama yang menentukan keberhasilan tracking. Hasil pengujian menunjukkan bahwa fungsi tracking sangat dipengaruhi oleh kondisi penerimaan sinyal satelit. Area terbuka menghasilkan data paling konsisten karena antena memiliki akses yang lebih baik ke langit dan tidak banyak terhalang. Area semi terbuka masih memungkinkan GPS memperoleh posisi, tetapi membutuhkan waktu lebih lama. Pada area tertutup, bangunan dan atap menghalangi penerimaan sinyal sehingga data posisi menjadi tidak stabil atau tidak *valid*. Penurunan kualitas data tersebut tidak menunjukkan kegagalan ESP32, tetapi memperlihatkan keterbatasan teknologi GPS yang bergantung pada sinyal satelit. Perubahan kecil koordinat yang terjadi ketika kendaraan diam menunjukkan variasi pembacaan pada GPS standar dan tidak selalu menandakan bahwa kendaraan mengalami perpindahan. Oleh karena itu, data koordinat prototipe lebih tepat digunakan untuk monitoring posisi umum daripada penentuan posisi berpresisi tinggi.

Stabilitas catu daya dan kekuatan sambungan juga memengaruhi kinerja sistem. ESP32, modul GPS, dan LCD membutuhkan tegangan yang stabil agar dapat beroperasi secara berkelanjutan. Baterai yang lemah dapat menyebabkan tampilan redup dan meningkatkan risiko *restart*, sedangkan kabel *jumper* yang longgar dapat menyebabkan gangguan atau terputusnya data. Kabel *jumper* juga mudah terganggu oleh getaran kendaraan. Oleh karena itu, penerapan di lapangan memerlukan papan rangkaian tercetak, konektor yang lebih kuat, dan *box* pelindung.

Tampilan lokal memberi informasi langsung kepada pengguna tanpa memerlukan komputer atau *server* eksternal. Perbedaan antara pesan menunggu sinyal dan tampilan koordinat *valid* membantu pengguna membedakan data posisi terbaru dari data yang belum tersedia. Namun, LCD 16×2 hanya mampu menampilkan informasi singkat. Pengembangan berikutnya dapat menggunakan OLED, layar TFT, *dashboard* lokal berbasis *web*, atau peta digital. Berdasarkan hasil simulasi, *interval* lima detik dapat menjadi pilihan yang seimbang untuk prototipe yang bergerak lambat karena masih memberikan detail perubahan posisi tanpa menghasilkan jumlah catatan sebanyak *interval* satu detik. Namun, hasil ini belum membuktikan kinerja penyimpanan aktual karena pengujian belum mencakup kapasitas media, kecepatan penulisan, dan penggunaan jangka panjang.

Keterbatasan penelitian meliputi ketergantungan pada penerimaan sinyal GPS, keterbatasan tampilan lokal, simulasi pencatatan yang belum menggunakan media penyimpanan aktual, dan koneksi kabel yang rentan terhadap getaran. Pengujian kestabilan ESP32 hanya dilakukan selama 30 menit, sedangkan pengujian GPS dilakukan sebanyak sepuluh kali pada setiap kondisi lingkungan. Hasil penelitian menggambarkan kemampuan fungsional prototipe pada skenario pengujian yang dilakukan, tetapi belum menunjukkan ketahanan jangka panjang atau tingkat ketelitian posisi. Prototipe juga belum ditujukan untuk navigasi berpresisi tinggi, perencanaan rute mandiri, atau sistem manajemen armada lengkap. Keterbatasan tersebut menjadi dasar untuk pengembangan lebih lanjut.

KESIMPULAN

Sistem *Smart Tracking* berbasis ESP32 dan modul GPS berhasil dirancang sebagai prototipe pelacakan lokasi pada mobil pembersih otomatis. ESP32 mampu menerima data serial GPS, memproses kalimat NMEA menjadi *latitude*, *longitude*, waktu pembacaan, dan status GPS, serta menampilkan informasi melalui Serial Monitor dan LCD tanpa menggunakan *server* eksternal. ESP32 tetap beroperasi selama pengujian komponen dan pengoperasian kendaraan, termasuk pada pengujian kestabilan selama 30 menit. Kinerja GPS terbaik diperoleh pada area terbuka, dengan sepuluh data *valid* dalam waktu 29 sampai 65 detik. Area semi terbuka meningkatkan waktu tunggu dan menurunkan kestabilan data, sedangkan area tertutup menghasilkan data tidak stabil atau tidak *valid*. Simulasi pencatatan menunjukkan bahwa *interval* lima detik menghasilkan 120 data selama sepuluh menit dan dapat menjadi pilihan awal yang seimbang bagi kendaraan yang bergerak lambat. Kontribusi penelitian ini adalah menghasilkan integrasi mandiri antara pembacaan koordinat GPS, pemrosesan data pada ESP32, simulasi pencatatan berbasis *interval*, dan penyajian informasi melalui tampilan lokal pada mobil pembersih otomatis. Prototipe dapat digunakan untuk *monitoring* posisi umum pada area dengan akses sinyal satelit yang memadai, tetapi belum sesuai untuk navigasi berpresisi tinggi, perencanaan rute, atau pengoperasian jangka panjang tanpa penyempurnaan perangkat keras. Pengembangan selanjutnya perlu menggunakan sambungan yang lebih permanen, *box* pelindung, media penyimpanan aktual, sensor baterai, *dashboard* lokal, dan tampilan peta digital, serta melakukan pengujian kestabilan dan ketelitian posisi dalam jangka waktu yang lebih panjang.

REFERENSI

- Cahyadi, R. A. H. (2019). Pengembangan Bahan Ajar Berbasis ADDIE Model. *Halaqa: Islamic Education Journal*, 3(1), 35–42. <https://doi.org/10.21070/halaqa.v3i1.2124>
- Dani, A. W., Sirait, F., & Saputra, I. S. (2024). Rancang Bangun Alat Kontrol dan Monitoring Pembersih Udara Menggunakan Metode Ionisasi Berbasis IoT. *Jurnal Ilmu Teknik dan Komputer*, 8(2), 98–102. <https://doi.org/10.22441/jitkom.v8i2.004>
- Effendi, N., Ramadhani, W., & Farida, F. (2022). Perancangan Sistem Penyiraman Tanaman Otomatis Menggunakan Sensor Kelembapan Tanah Berbasis IoT. *Jurnal CoSciTech (Computer Science and Information Technology)*, 3(2), 91–98. <https://doi.org/10.37859/coscitech.v3i2.3923>
- Exaudi, K., Setiawan, N. S., Prasetyo, A. P. P., Passarella, R., Ubaya, H., Sari, P., ... Mediswati, R. Y. T. (2023). Purwarupa Sistem Pemantauan Polusi Udara di Ruang Tertutup Menggunakan Platform ThingSpeak. *JTT (Jurnal Teknologi Terapan)*, 9(2), 101. <https://doi.org/10.31884/jtt.v9i2.412>
- Ginting, R. C., Ishak, & Yakub. (2021). Implementasi Real Time Clock (RTC) pada Robot Line Follower untuk Vacuum Cleaner Berbasis Arduino. *Jurnal Teknisi*, 1(1), 8–12. <https://doi.org/10.54314/teknisi.v1i1.483>
- Hasanuddin, M., & Herdianto, H. (2023). Sistem Monitoring dan Deteksi Dini Pencemaran Udara Berbasis Internet of Things (IoT). *Journal of Computer System and Informatics (JoSYC)*, 4(4), 976–984. <https://doi.org/10.47065/josyc.v4i4.4034>
- Hutabarat, B. F., Peslinof, M., Afrianto, M. F., & Fendriani, Y. (2023). Sistem Basis Data Pemantauan Parameter Air Berbasis Internet of Things (IoT) dengan Platform ThingSpeak. *Journal Online of Physics*, 8(2), 42–50. <https://doi.org/10.22437/jop.v8i2.24365>
- Maulana, R. F., Ramadhan, M. A., Maharani, W., & Maulana, M. I. (2023). Rancang Bangun Sistem Monitoring Suhu dan Kelembapan Berbasis IoT: Studi Kasus Ruang Server IT Telkom Surabaya. *Indonesian Journal of Multidisciplinary on Social and Technology*, 1(3), 224–231. <https://doi.org/10.31004/ijmst.v1i3.169>
- Melvi, Ulvan, A., Sidiq, M. R., & Batubara, M. A. M. (2023). Rancang Bangun Sistem Monitoring Ketinggian Muka Air Laut Menggunakan Arduino Pro Mini dan NodeMCU ESP8266. *Jurnal Teknologi Riset Terapan*, 1(1), 27–38. <https://doi.org/10.35912/jatra.v1i1.1794>
- Muzakirin, M., & Mirza, A. H. (2022). Implementasi Monitoring dan Notifikasi Kualitas Udara Menggunakan Arduino Berbasis IoT. *Journal of Computer and Information Systems Ampera*, 3(2), 99–110. <https://doi.org/10.51519/journalcisa.v3i2.176>
- Oktaviani, A. M., Yulkifli, & Nofriandi, A. (2023). Rancang Bangun Sistem Kontrol dan Monitoring Suhu Udara pada Smart Farming Stroberi Berbasis IoT. *Jurnal Pendidikan Tambusai*, 7(3), 25526–25535. <https://doi.org/10.31004/jptam.v7i3.10683>

- Salam, F., & Alexander, O. (2023). Perancangan Monitoring Suhu Berbasis Internet of Things dengan NodeMCU ESP8266, DHT11, dan ThingSpeak. *Jurnal Ilmiah Informatika*, 11(1), 22–26. <https://doi.org/10.33884/jif.v11i01.6546>
- Saputra, A., Khumaini, H., & Azkiya, A. (2022). Perancangan Alat Monitoring Arus pada Circuit Breaker dengan Sensor ACS712 Menggunakan Tampilan LCD. *INFORMATIKA*, 14(2), 86–91. <https://doi.org/10.36723/juri.v14i2.505>
- Saputra, R. D. (2022). Prototype Sistem Pengaturan Kecepatan Kipas DC Otomatis Menggunakan Sensor PIR, Sensor Ultrasonik, Sensor DHT11 Berbasis Mikrokontroler Arduino Uno dan NodeMCU. *Electrician*, 16(1), 45–55. <https://doi.org/10.23960/elc.v16n1.2208>
- Satria, B. (2022). IoT Monitoring Suhu dan Kelembaban Udara dengan NodeMCU ESP8266. *Sudo Jurnal Teknik Informatika*, 1(3), 136–144. <https://doi.org/10.56211/sudo.v1i3.95>
- Setiawan, A., Sulistiyanto, S., Riyanto, C. A., & Wiguna, G. A. (2024). Desain Sistem Pemantauan Kualitas Udara Jangka Panjang Berbasis Internet of Things. *Jurnal Teras Fisika*, 7(1), 24–29. <https://doi.org/10.20884/1.jtf.2024.7.1.11859>
- Sinaga, G. E. L., Gunawan, I., Irawan, & Poningsih. (2022). Rancang Bangun Sistem Keamanan Sepeda Motor Berbasis Arduino Uno Menggunakan GPS dan Relay melalui Smartphone. *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, 1(1), 1–7. <https://doi.org/10.55123/storage.v1i1.154>
- Sjafrina, F., Chandra, Y. I., & Arnesia, P. D. (2023). Rancang Bangun Purwarupa Alat Monitoring Kelembaban dan Suhu Tanaman Bunga Mawar Menggunakan Model Big Bang Berbasis IoT. *KESATRIA: Jurnal Penerapan Sistem Informasi*, 4(1), 157–169. <https://doi.org/10.30645/kesatria.v4i1.126>
- Wicaksana, D. A., & Hirzan, A. M. (2024). Pemantauan Kebocoran Gas dan Panas Udara dengan Metode Fuzzy Berbasis IoT. *Jurnal Teknologi Informasi dan Komunikasi*, 15(1), 125–132. <https://doi.org/10.51903/jtikp.v15i1.802>
- Wulandari, S., & Satria, B. (2021). Rancang Bangun Alat Pendeteksi Warna Menggunakan Arduino Uno Berbasis IoT (Internet of Things). *Paradigma - Jurnal Komputer dan Informatika*, 23(1). <https://doi.org/10.31294/p.v23i1.9861>