

Implementasi Sistem Penjaga Jarak Kendaraan Menggunakan ESP32-CAM Dengan Aplikasi Flutter

1st John Piter Natanael Siboro
School of Electrical Engineering
Universitas Telkom
Bandung, Indonesia

aidipalaidi@student.telkomuniversity.ac.id

2nd Agus Virgono
School of Electrical Engineering
Universitas Telkom
Bandung, Indonesia

anggunmekat@telkomuniversity.ac.id

3rd Randy Efra Saputra
School of Electrical Engineering
Universitas Telkom
Bandung, Indonesia

faicanhasfcb@telkomuniversity.ac.id

Abstrak— Keselamatan berkendara sangat bergantung pada kemampuan pengemudi menjaga jarak aman antar kendaraan. Data menunjukkan bahwa sebagian besar kecelakaan lalu lintas terjadi akibat pengemudi yang tidak mampu menjaga jarak aman, khususnya pada kondisi lalu lintas padat maupun kecepatan tinggi. Berdasarkan permasalahan tersebut, penelitian ini merancang sistem penjaga jarak aman kendaraan berbasis modul ESP32-CAM. Sistem ini bertujuan untuk memberikan informasi estimasi jarak secara real-time dengan biaya yang relatif murah dan implementasi yang sederhana, sehingga dapat mendukung upaya peningkatan keselamatan berkendara. Metode yang digunakan adalah pemrosesan citra digital berbasis kamera monokular. ESP32-CAM digunakan untuk menangkap gambar objek di depan kendaraan, kemudian menghitung jarak berdasarkan lebar objek dalam piksel dengan pendekatan proporsionalitas focal length. Data hasil estimasi dikirimkan melalui koneksi Bluetooth Low Energy (BLE) ke aplikasi mobile berbasis Flutter. Aplikasi ini menampilkan informasi jarak secara langsung kepada pengguna melalui teks maupun notifikasi, sehingga pengemudi dapat segera mengetahui kondisi jarak aman. Hasil pengujian menunjukkan bahwa sistem mampu melakukan estimasi jarak dengan tingkat akurasi yang cukup baik serta delay rendah, sehingga layak digunakan sebagai pendukung keselamatan berkendara. Penelitian ini diharapkan dapat menjadi kontribusi dalam pengembangan sistem keselamatan kendaraan modern yang lebih terjangkau.

Kata kunci— ESP32-CAM, Estimasi Jarak, Bluetooth Low Energy (BLE)

I. PENDAHULUAN

Saat berkendara di jalan raya lalu lintas, di mana kendaraan sering melaju dengan kecepatan sedang sampai tinggi, sangat penting untuk menjaga jarak aman dengan kendaraan di depannya. Banyak kecelakaan lalu lintas yang terjadi karena pengemudi tidak menjaga jarak aman dengan kendaraan di depannya akibat kelalaian atau kurangnya informasi mengenai jarak yang benar [1]. Selain itu, mengemudi dengan kecepatan tinggi dapat meningkatkan risiko tidak menjaga jarak aman kendaraan, yang dapat menyebabkan kecelakaan fatal dan membahayakan pengemudi dan kendaraan pengemudi lainnya [2]. Menurut data Badan Pusat Statistik, kecelakaan lalu lintas di Indonesia pada periode tahun 2020-2022 meningkat setiap tahun menyentuh angka 139.258 kejadian pada tahun 2022 [3]. Kasus kecelakaan tersebut mencapai 50 ribu kejadian yang merupakan penyebab terbesarnya adalah karena gagal menjaga jarak aman kendaraan [4]. Hal ini terutama terjadi di jalan raya atau jalan antar kota dimana kendaraan melaju dengan kecepatan tinggi, dimana pengereman mendadak atau perubahan kecepatan yang tinggi dapat berakibat fatal. Kemudian, kondisi lingkungan jalan seperti jalan berlubang, jalan berliku, dan kondisi cuaca dapat terjadinya kecelakaan

[5]. Menurut Wakil Ketua Umum Bidang Keselamatan Transportasi Indonesia (MTI) memperkirakan bahwa angka kecelakaan lalu lintas di Indonesia akan mencapai 116 ribu pada tahun 2023. Banyak dari kasus ini disebabkan oleh kelalaian pengemudi, rasa kantuk, dan kelelahan pengemudi [6]. Untuk itu diperlukan suatu alat yang dapat membantu pengemudi dalam berkendara aman dengan mengetahui informasi jarak antar kendaraan agar pengemudi dapat membuat keputusan yang tepat saat berkendara dengan jarak aman kendaraan secara akurat.

II. KAJIAN TEORI

A. Sensor Pengukur Jarak

ESP32-CAM merupakan modul kamera digital yang digunakan untuk menangkap citra objek di depan kendaraan secara *real-time*. Modul ini memanfaatkan pendekatan *computer vision* berbasis kamera monokular untuk mendeteksi objek dan menghitung estimasi jarak berdasarkan lebar objek dalam satuan piksel. Metode ini memungkinkan estimasi jarak dilakukan dengan efisien tanpa memerlukan sensor tambahan.

B. Sistem Embedded dan antarmuka pemrograman

ESP32-CAM *Downloader* adalah Modul *downloader* digunakan untuk melakukan pemrograman awal pada ESP32-CAM melalui koneksi USB. Perangkat ini memungkinkan transfer kode Arduino dari komputer ke ESP32-CAM secara langsung, dan menjadi bagian penting dalam proses pengembangan dan debugging sistem. Sistem memperoleh asupan daya dari aki kendaraan yang dikonversi melalui inverter menjadi tegangan USB 5V. Hal ini memungkinkan perangkat bekerja secara mandiri tanpa harus bergantung pada sistem kelistrikan utama kendaraan.

C. Sistem Komunikasi

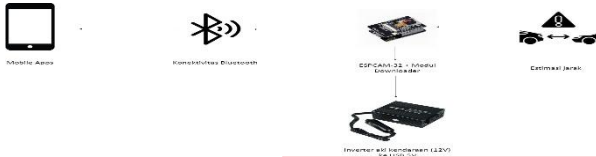
Aplikasi *Mobile* Berbasis Flutter untuk menampilkan hasil estimasi jarak kepada pengguna, dikembangkan sebuah aplikasi mobile menggunakan *framework* Flutter. Aplikasi ini menerima data jarak dari ESP32-CAM melalui koneksi Bluetooth *Low Energy* (BLE) dan menampilkannya secara *real-time* dalam bentuk teks atau grafik. Penggunaan Flutter memungkinkan pengembangan aplikasi yang responsif dan kompatibel dengan berbagai perangkat

III. METODE

Dalam pengembangan prototipe “Sistem Penjaga Jarak Aman Kendaraan”, pemilihan alternatif solusi tepat untuk mendeteksi jarak minimal kendaraan aman dengan akurasi tinggi, khususnya teknologi yang mampu mendeteksi jarak minimal pada kendaraan. Di satu sisi, kebutuhan teknologi

yang andal untuk mendukung keselamatan berkendara semakin meningkat seiring dengan kemajuan teknologi otomotif [7]. Namun, keterbatasan sistem dan biaya sering menjadi faktor yang membatasi penerapan teknologi tersebut. Oleh karena itu, sangat penting untuk mempertimbangkan alternatif solusi yang sesuai dengan kebutuhan dan sumber daya yang tersedia. Beberapa opsi seperti sensor LiDAR, Kamera, dan laser rangefinder dapat menjadi opsi solusi yang layak yang dapat dipertimbangkan.

A. Gambaran Rancangan Sistem



GAMBAR 1 Rancangan Sistem

Pada Gambar 3.1 Dalam perancangan sistem pendeteksi jarak aman kendaraan, terdapat tiga komponen utama yang digunakan, yaitu ESP32-CAM, modul *downloader* ESP32-CAM, dan inverter listrik dari aki ke USB sebagai sumber daya. Ketiga komponen ini saling terintegrasi untuk membentuk sistem pemantauan jarak kendaraan. ESP32-CAM berfungsi sebagai kamera utama yang dipasang di bagian depan kendaraan untuk menangkap gambar secara langsung. Gambar yang diperoleh kemudian diproses menggunakan metode perhitungan berbasis kamera monokular, yang memungkinkan sistem melakukan estimasi jarak kendaraan di depan berdasarkan ukuran objek dalam satuan piksel. Modul *downloader* ESP32-CAM berfungsi sebagai jembatan komunikasi antara ESP32-CAM dan sumber daya, sekaligus sebagai koneksi USB untuk mengalirkan daya ke modul utama. Selain itu, kabel USB digunakan untuk menghubungkan ESP32-CAM ke sumber daya dari inverter.

B. Gambaran Alur keseluruhan Sistem



GAMBAR 2 Alur Keseluruhan Sistem

Gambar 3.2 Alat sistem pendeteksi jarak aman kendaraan bekerja dengan menggunakan ESP32-CAM dan Arduino Uno. Ketika proses dimulai, sistem akan melakukan inisiasi dan kalibrasi kamera (ESP32-CAM) untuk memastikan bahwa dapat berfungsi secara optimal dan memberikan data yang sangat akurat oleh sensor. Setelah kalibrasi, kamera mulai menangkap gambar secara *real-time*. Kemudian

sistem akan membaca lebar objek kendaraan pada gambar dalam satuan piksel, dan akan melakukan perhitungan jarak dengan rumus monokular kamera. Setelah itu, sistem akan membandingkan hasil perhitungan jarak dengan *threshold* jarak aman apakah hasil data jarak tersebut termasuk jarak aman atau tidak, jika hasil data jaraknya kurang dari 30 meter maka termasuk kategori peringatan, sedangkan jika lebih dari 30 meter maka termasuk kategori aman. Hasil perbandingan kategori jarak aman atau peringatan kemudian akan dikirim ke aplikasi *mobile* melalui bluetooth. Data jarak kendaraan siap ditampilkan ke *mobile* pengguna.

C. Desain Perangkat Keras

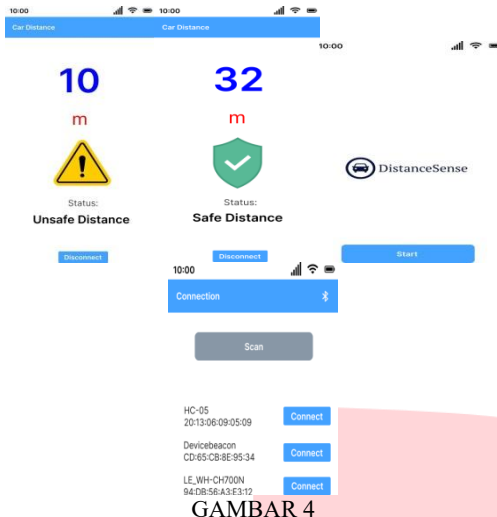
Dalam perancangan sistem pendeteksi jarak aman kendaraan, terdapat tiga komponen utama yaitu ESP32-CAM, aki kendaraan (12V) dan Inverter usb (5V). Pada tiga komponen ini memiliki fungsi yang saling terintegrasi untuk merancang sistem pemantauan jarak kendaraan secara real time. ESP32-CAM berfungsi sebagai kamera utama yang ditempatkan di bagian depan kendaraan untuk menangkap gambar secara langsung. Gambar tersebut kemudian diproses dengan metode perhitungan berbasis kamera monokular, yang memungkinkan sistem mendeteksi kendaraan hingga jarak sekitar 30 meter atau lebih. Aki kendaraan (12V) berfungsi sebagai sumber daya utama sistem, yang kemudian dikonversi menjadi tegangan 5V melalui inverter USB untuk memberikan daya kepada ESP32-CAM. Kemudian, melalui Bluetooth bawaan ESP32-CAM, data jarak yang dihitung oleh ESP32-CAM dikirimkan secara langsung ke aplikasi *mobile* pengemudi.



GAMBAR 3 Desain Sistem

Ketiga komponen tersebut dipasang dalam kotak khusus pada bagian depan mobil yang dirancang pada gambar 3.3. Rancangan ini berfungsi untuk mengintegrasikan semua komponen sistem dalam satu unit yang kompak, sehingga instalasi menjadi lebih mudah dan komponen tetap stabil ketika dipasang di bagian depan kendaraan.

D. Desain Perangkat Lunak



GAMBAR 4 Desain Perangkat Lunak

Pada Gambar 3.5 Desain mockup adalah tampilan atau gambaran sebagai interface untuk pengguna (UI) dari aplikasi atau produk yang belum dikembangkan, serta diterapkan secara keseluruhan. mockup desain mempertunjukkan bagaimana visual aplikasi ini akan di desain dan bagaimana fungsinya aplikasi tersebut bekerja serta interaksi yang akan ada di aplikasi tersebut. Pada aplikasi mobile sistem pendeteksi jarak aman kendaraan, mockup desain berfungsi untuk menampilkan elemen-elemen antarmuka pengguna termasuk yang berfungsi untuk menampilkan halaman utama mengenai informasi jarak aman serta menampilkan fitur lainnya seperti pemberitahuan atau peringatan status jarak kendaraan. Berikut merupakan tampilan pada aplikasi mobile sistem pendeteksi jarak aman kendaraan

IV. HASIL DAN PEMBAHASAN

Aplikasi sistem penjaga jarak aman kendaraan ini dirancang dengan mengintegrasikan komponen ESP32-CAM beserta kamera bawaan sebagai sensor utama untuk mendeteksi jarak kendaraan di depan, serta menggunakan Flutter sebagai platform pengembang aplikasinya. Flutter adalah SDK yang dibuat oleh Google dan berlisensi open source yang memungkinkan pengembangan aplikasi mobile berkinerja tinggi untuk platform iOS atau Android[8]. Flutter juga mendukung fitur Hot Reload dan Hot Restart. Hot Reload merupakan fitur untuk perubahan kode langsung terlihat tanpa perlu menghidupkan kembali aplikasi, sedangkan Hot Restart adalah fitur yang memungkinkan aplikasi memulai ulang tanpa kehilangan state awal yang dibutuhkan selama pengujian. Kamera pada ESPCAM-32 mengukur jarak kendaraan dan dibaca oleh ESPCAM-32. Kemudian data tersebut dikirim ke aplikasi melalui Bluetooth dari bawaan pada ESPCAM-32. Setelah data tersebut diterima oleh aplikasi Flutter dari ESPCAM-32, maka hasil data tersebut dapat ditampilkan di aplikasi secara real-time, sehingga memungkinkan memantau jarak secara akurat dan responsif. Aplikasi juga mengelola koneksi Bluetooth dengan perangkat ESP32-CAM, memastikan komunikasi yang stabil dan lancar selama pemantauan. Untuk meningkatkan keselamatan berkendara, sistem ini

memberikan peringatan dini dan membantu pengemudi menjaga jarak aman.

A. Pengujian Jarak

TABEL 1 Pengujian Jarak

| No. | Jarak Asli (cm) | Lebar piksel (P) | Perhitungan Estimasi Jarak | Jarak Estimasi (cm) |
|-----|-----------------|------------------|----------------------------|---------------------|
| 1 | 10 | 54 | $(91.67 \times 6) / 54$ | 10.19 |
| 2 | 20 | 27 | $(91.67 \times 6) / 27$ | 20.37 |
| 3 | 30 | 18 | $(91.67 \times 6) / 18$ | 30.56 |
| 4 | 40 | 13.5 | $(91.67 \times 6) / 13.5$ | 40.74 |

TABEL 2 Pengujian Jarak

| | | | | |
|----|-----|------|---------------------------|--------|
| 5 | 50 | 10.8 | $(91.67 \times 6) / 10.8$ | 50.94 |
| 6 | 60 | 9 | $(91.67 \times 6) / 9$ | 61.11 |
| 7 | 70 | 7.71 | $(91.67 \times 6) / 7.71$ | 71.39 |
| 8 | 80 | 6.75 | $(91.67 \times 6) / 6.75$ | 81.52 |
| 9 | 90 | 6 | $(91.67 \times 6) / 6$ | 91.67 |
| 10 | 100 | 5.4 | $(91.67 \times 6) / 5.4$ | 101.89 |

Pada Tabel 4.1 Pengujian ini dilakukan untuk mengevaluasi kemampuan sistem dalam mengukur jarak objek menggunakan kamera ESP32-CAM. Kamera mendeteksi lebar objek dalam gambar (dalam satuan piksel), kemudian jarak objek dihitung menggunakan rumus kalibrasi berdasarkan panjang fokus kamera.

B. Pengujian Error Data Dan Delay

TABEL 3 Pengujian Error Data Dan Delay

| No | Data Kirim (m) | Jarak FastAPI(m) | Selisih (m) | Delay (ms) |
|----|----------------|------------------|-------------|------------|
| 1 | 20 | 20 | 0 | 30 |
| 2 | 20 | 22 | 2 | 22 |
| 3 | 23 | 23 | 0 | 78 |
| 4 | 28 | 29 | 1 | 47 |
| 5 | 30 | 31 | 1 | 36 |
| 6 | 35 | 34 | 1 | 64 |
| 7 | 37 | 39 | 2 | 104 |
| 8 | 40 | 42 | 2 | 121 |
| 9 | 43 | 42 | 1 | 71 |
| 10 | 41 | 43 | 2 | 52 |

Pada Tabel 4.2 Pengujian ini dilakukan untuk mengukur persentase error data penerimaan data jarak dan waktu delay saat mengirim data jarak dari ESP32-CAM ke perangkat mobile melalui Bluetooth. Pengujian ini dengan memanfaatkan framework FastAPI untuk membantu dalam tracking log data jarak yg diterima mobile dan perhitungan delay, kemudian membandingkan hasil data dari ESP32CAM dengan hasil pada FastAPI untuk mencari error. Jarak FastAPI merupakan data yang diterima oleh mobile, sedangkan data kirim adalah data yang kirim dari ESP32CAM. Pada Tabel tersebut merupakan sample 10 data mewakili simpulan data dari 50 data , sehingga untuk data

lengkapnya 50 data dapat dilihat pada Lampiran. Sample data di atas menunjukkan variasi *delay* dengan rentang antara 22-121 ms, serta perbedaan hasil data yang dikirim dan data yang diterima pada *mobile* dengan menselisihkan data kirim dengan data yang diterima untuk mendapatkan *error* data. Sehingga pada hasil akhir saat melakukan perhitungan diperoleh berikut:

- Jumlah data: 50
- Delay minimal: 22 ms
- Delay maksimal: 121 ms
- Rata-rata error (m): 1.14 m
- Rata-rata error keseluruhan (%): 3.81%
- Standar Deviasi: 0.69 m

C. Pengujian Blackbox Testing

Pengujian *Blackbox* dilakukan dengan membuat berbagai skenario pengujian yang mencakup semua aspek aplikasi. Teknik ini mengevaluasi respons aplikasi dengan memberikan berbagai jenis input dan memastikan bahwa aplikasi memberikan hasil yang benar serta menangani input yang tidak valid dengan tepat. *Blackbox* testing membantu memastikan bahwa aplikasi memenuhi kebutuhan pengguna dan berfungsi dengan baik dalam berbagai kondisi.

TABEL 3
Pengujian Blackbox Testing

| No | Fitur | Pengujian | Hasil yang diharapkan | Hasil |
|----|---------------------------|--|---|----------|
| 1 | Tampilan Awal | Halaman awal ditampilkan | Menampilkan logo dan tombol "Start" | Berhasil |
| 2 | Navigasi ke Halaman Utama | Menekan tombol "Start" | Menuju halaman utama untuk koneksi bluetooth | Berhasil |
| 3 | Scan Bluetooth BLE | Menekan tombol "Scan" | Menampilkan daftar perangkat BLE di sekitar | Berhasil |
| 4 | Pilih Perangkat ESP32-CAM | Menekan tombol "Connect" pada nama ESP-CAM32 | Berpindah ke halaman Deteksi jika sesuai UUID pada ESPCAM32 | Berhasil |

TABEL 4
Pengujian Blackbox Testing

| | | | | |
|---|-------------------------------|---|--|----------|
| 5 | Terima Data BLE dan ESP32-CAM | ESP32-CAM mengirim data via karakteristik BLE | Memastikan bahwa sinkron dan terhubung dengan ESP32-CAM saat pengiriman data | Berhasil |
| 6 | Peringatan Jarak Aman | Ketika Jarak ≥ 30 meter | Menampilkan status visual dan audio "Safe Distance" dengan ikon centang | Berhasil |
| 7 | Peringatan Jarak Tidak Aman | Ketika Jarak < 30 meter | Menampilkan status visual dan audio "Unsafe Distance" dengan ikon tanda seru | Berhasil |
| 8 | Tombol Disconnect | Tekan tombol "Disconnect" | Kembali ke halaman utama | Berhasil |
| 9 | Respons <i>Real-time</i> | Mengamati apakah informasi jarak langsung berubah di aplikasi saat objek bergerak mendekati atau menjauh dari kamera. | Informasi jarak diperbarui secara langsung | Berhasil |

Pada Gambar 4.1 Secara keseluruhan, metode ini bertujuan untuk menemukan kesalahan fungsional, kekurangan antarmuka pengguna, dan inkonsistensi dalam performa

aplikasi yang dapat mempengaruhi pengalaman pengguna. Pendekatan ini memungkinkan tim pengembangan untuk memastikan bahwa aplikasi siap untuk digunakan.

V. KESIMPULAN

Hasil pengujian dari Sistem Pemantauan Jarak Aman Kendaraan, yang berfokus pada validasi fungsional dan evaluasi kinerja sistem. Pengujian dilakukan dengan menggunakan metode black-box testing untuk memastikan bahwa sistem memenuhi spesifikasi yang diharapkan dan kebutuhan pengguna. Sistem menunjukkan kinerja yang konsisten dalam mendeteksi dan menampilkan jarak aman, dengan pembaruan *real-time* yang terlihat melalui aplikasi mobile. Selain itu, hasil estimasi jarak dari sistem dibandingkan dengan pengukuran aktual, yang mengonfirmasi keandalan dan akurasi sistem

REFERENSI

[1] Nagina, "Jangan Sepelekan Pentingnya Jarak Aman Berkendara," *asukacartv.com*, 2024. <https://www.asukacartv.com/jarak-aman-berkendara/> (accessed Aug. 20, 2025).

[2] W. K. W. Staff, "Dangers of Speeding: Why Is it Illegal, and What Are the Legal Consequences," *https://www.wkw.com/auto-accidents/blog/dangers-of-speeding-why-is-it-illegal-and-what-ar%20e-the-legal-consequences/*, 2024. <https://www.wkw.com/auto-accidents/blog/dangers-of-speeding-why-is-it-illegal-and-what-ar-e-the-legal-consequences/>

[3] bps.go.id, "Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi, 2022," *bps.go.id*, 2022. <https://www.bps.go.id/id>

[4] toyota, "Jaga Jarak Aman Sering Dilupakan, Begini Rumus 3 Detik Sebagai Patokan Jarak Mobil yang Aman di Jalan," <https://www.toyota.astra.co.id/>. <https://www.toyota.astra.co.id/corporate-information/news-promo/read/jaga-jarak-aman-sering-dilupakan-begini-rumus-3-detik-sebagai-patokan-jarak-mobil-yang-aman-di-jalan>

[5] M. N. Nofi Susanti, Chairunnisa T D E Angkat, Dhea Afriza Pohan, "ANALISIS FAKTOR – FAKTOR YANG BERHUBUNGAN DENGAN RESIKO KECELAKAAN LALU LINTAS," *Vol. 5, Nomor 2, 2024*, 2024.

[6] J. W. and R. Rajamani, "Adaptive cruise control system design and its impact on highway traffic flow." <https://ieeexplore.ieee.org/document/1024501>

[7] and J. C.-C. J. Guerrero-Ibáñez, S. Zeadally, "Sensor technologies for intelligent transportation systems," *MFPI*. <https://www.mdpi.com/1424-8220/18/4/1212>

[8] S. R. muslim muslim, Renny Puspita Sari, "IMPLEMENTASI FRAMEWORK FLUTTER PADA SISTEM INFORMASI PERPUSTAKAAN MASJID," 2022, [Online]. Available: <https://jurnal.untan.ac.id/>