

IMPLEMENTASI APLIKASI MOBILE PADA SISTEM PENJAGA JARAK AMAN KENDARAAN

1st Raisya Athaya Kamilah
Faculty of Electrical Engineering
Universitas Telkom
Bandung, Indonesia
raisyaakk@student.telkomuniversity.ac.id

2nd Ir Agus Virgono, M.T.
Faculty of Electrical Engineering
Universitas Telkom
Bandung, Indonesia
avirgono@telkomuniversity.ac.id

3rd Randy Erfa Saputra, S.T., M.T.
Faculty of Electrical Engineering
Universitas Telkom
Bandung, Indonesia
resaputra@telkomuniversity.ac.id

Abstrak — Dengan konektivitas *Bluetooth Low Energy* (BLE) dengan ESPCAM32, aplikasi mobile ini dapat menerima, menampilkan, dan memberikan peringatan jarak aman secara real-time. Antarmuka pengguna menampilkan informasi jarak dalam bentuk numerik dan memberikan peringatan audio-visual ketika jarak terdeteksi kurang dari ambang batas. Aplikasi dihubungkan ke ESPCAM32 pada kendaraan untuk melakukan pengujian. Setelah itu, mereka memeriksa penerimaan data, keakuratan tampilan, dan respons peringatan. Hasil pengujian menunjukkan bahwa aplikasi mobile mampu menampilkan data jarak dengan tingkat keterlambatan rendah dengan akurasi informasi data dari ESPCAM32 ke *mobile* 96,19% dan persentase *error* data 3,81%. Aplikasi ini secara konsisten memberikan peringatan dan dapat digunakan dalam berbagai kondisi operasional.

Kata kunci — Aplikasi mobile, ESPCAM32, Bluetooth Low Energy, Jarak aman kendaraan.

I. PENDAHULUAN

Jarak aman antara kendaraan adalah faktor penting yang sering diabaikan saat berkendara. Pengemudi yang tidak menjaga jarak yang cukup dengan kendaraan di depannya saat pengereman mendadak sering menjadi penyebab kecelakaan lalu lintas [1]. Aplikasi *mobile* yang dapat digunakan sebagai bagian dari sistem penjaga jarak aman kendaraan akan menjadi fokus pada penelitian ini. Sistem ini dapat menampilkan informasi jarak yang dideteksi secara *real-time* melalui koneksi *Bluetooth Low Energy* (BLE) dari perangkat sensor yang terpasang pada kendaraan.

Aplikasi *mobile* dimaksudkan untuk menampilkan jarak dalam satuan meter, memberikan peringatan ketika jarak kurang dari ambang batas, dan meminimalkan keterlambatan atau keterlambatan pengiriman data sehingga pengemudi dapat segera merespons. Pengujian dilakukan dengan mengukur akurasi dan konsistensi data aplikasi dibandingkan dengan jarak nyata di lapangan. Untuk memberikan saran pengembangan tambahan, juga dilakukan analisis faktor-faktor yang mempengaruhi kinerja aplikasi, seperti kestabilan koneksi BLE dan keterbatasan perangkat keras.

Jurnal ini diharapkan aplikasi mobile dapat membantu pengemudi menjaga jarak aman saat berkendara dan mengurangi risiko kecelakaan di jalan raya.

II. KAJIAN PUSTAKA

A. ESPCAM32

ESPCAM32 adalah modul mikrokontroler berbasis ESP32 yang memiliki kamera OV2640 dan mendukung konektivitas Wi-Fi dan *Bluetooth Low Energy* (BLE). ESPCAM32 pada sistem penjaga jarak aman kendaraan berfungsi untuk mengolah dan mengirimkan data jarak yang diukur oleh sensor ke aplikasi *mobile*. Informasi jarak yang dikirim ditampilkan pada antarmuka aplikasi sebagai informasi jarak kendaraan di depan dan memicu peringatan jika jarak yang diukur kurang dari ambang batas.[2].

B. Aplikasi Mobile

Aplikasi *mobile* adalah perangkat lunak yang dibuat untuk digunakan pada perangkat bergerak seperti smartphone atau tablet dengan antarmuka dan fitur yang dioptimalkan untuk layar sentuh sambil mempertimbangkan sumber daya yang tersedia untuk perangkat tersebut [3]. Aplikasi mobile dalam penelitian ini berfungsi sebagai penerima data jarak dari ESPCAM32 dengan koneksi BLE. Aplikasi ini juga menampilkan informasi jarak dan peringatan secara visual-audio.

C. Bluetooth Low Energy (BLE)

Deep Learning adalah subbidang pembelajaran mesin yang melibatkan penggunaan jaringan saraf berlapis-lapis yang dikenal sebagai jaringan saraf dalam untuk mempelajari pola dan membuat prediksi dari kumpulan data besar seperti gambar, video atau suara [4].

D. Flutter

Flutter sebagai platform pengembang aplikasi pada sistem penjaga jarak aman kendaraan. Flutter adalah SDK yang dibuat oleh Google dan berlisensi open source yang memungkinkan pengembangan aplikasi *mobile* berkinerja tinggi untuk platform iOS atau Android [5]. 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 [6].

III. METODE

Metode untuk aplikasi *mobile* dirancang dan dipasang untuk menerima data jarak dari perangkat pengukur jarak yang terpasang pada kendaraan melalui koneksi BLE. Pengujian dilakukan dengan menjalankan aplikasi pada perangkat *mobile*:

1. Pengiriman Data

Memastikan bahwa aplikasi *mobile* dapat dengan stabil mengirim dan menerima data jarak.

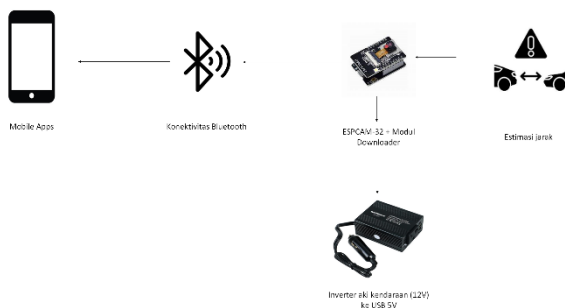
2. Tampilan Informasi

Mempertahankan tampilan yang jelas dan mudah dibaca dari jarak yang diterima pada antarmuka pengguna.

Peringatan Jarak Tidak Aman

Memverifikasi bahwa sistem memberikan peringatan atau notifikasi visual atau auditori saat jarak terdeteksi di bawah batas.

A. Gambaran Rancangan Sistem

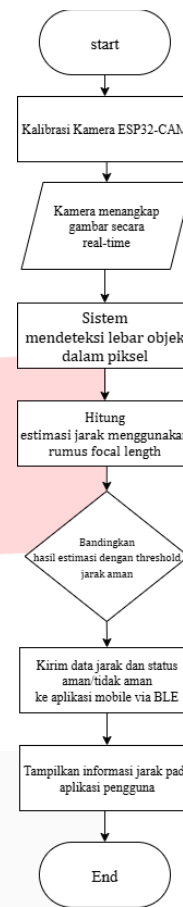


GAMBAR 1
(A)

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 koneksi USB untuk mengalirkan daya ke modul utama. Selain itu, kabel USB digunakan untuk menghubungkan ESP32-CAM ke sumber daya dari *inverter*. Sumber daya sistem diperoleh dari *inverter* yang mengubah tegangan DC dari aki kendaraan menjadi *output* USB 5V, sehingga memungkinkan seluruh perangkat bekerja secara mandiri tanpa bergantung pada sistem kelistrikan utama kendaraan. Semua komponen sistem ini dipasang dalam

kotak khusus yang dirancang dan ditempatkan di bagian depan kendaraan.

B. Flowchart



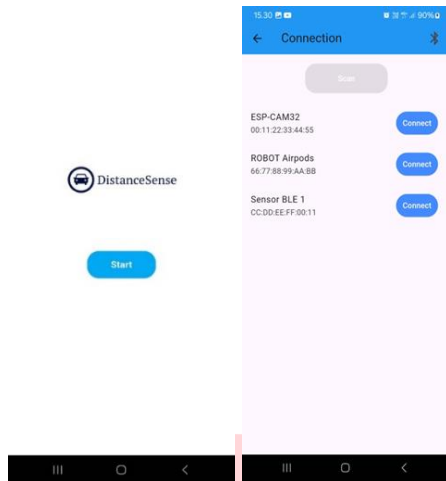
GAMBAR 2
(B)

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.

IV. HASIL DAN PEMBAHASAN

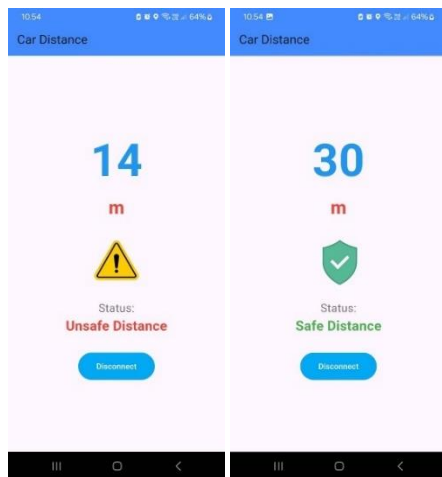
Implementasi pada penelitian yang dilakukan pada sistem ini merupakan perancangan sistem yang telah dibuat sesuai dengan fungsional dan kebutuhan sistem. Bagian ini menjelaskan Implementasi aplikasi *mobile* pada sistem penjaga jarak aman kendaraan, *blackbox testing*, dan pengujian *error* data dan *delay* pada *mobile*.

A. Implementasi



GAMBAR 3
(C)

Pada saat membuka aplikasi sistem penjaga jarak aman kendaraan pada perangkat pengguna, maka akan muncul tampilan awal aplikasi, kemudian pengguna dapat melakukan *scanning device* agar dapat menemukan device ESPCAM32



GAMBAR 4
(D)

Setelah pengguna menghubungkan ESPCAM32 ke mobile, maka aplikasi akan memasuki ke halaman notifikasi peringatan jarak. Jika sistem mendeteksi jarak dibawah 30 meter dari data ESPCAM saat mengirim data ke mobile, maka aplikasi akan memberikan notifikasi peringatan jarak tidak aman melalui tampilan visual-audio pada mobile, sedangkan jika mendeteksi lebih dari 30 meter maka memberikan notifikasi peringatan jarak aman.

B. 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 1
(A)

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
5	Terima Data BLE dari 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 $\geq$ 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 Real-time	Mengamati apakah informasi jarak langsung berubah di aplikasi saat objek bergerak mendekat atau menjauh dari kamera.	Informasi jarak diperbarui secara langsung	Berhasil

Tabel pengujian di atas menunjukkan bahwa seluruh 9 skenario pengujian *Blackbox* telah berhasil dilakukan sesuai dengan perencanaan fitur yang dijelaskan pada bab sebelumnya.

C. Pengujian *error* data dan *delay*

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.

TABEL 2
(B)

No	Data Kirim (m)	Jarak FastAPI(m)	Selisih (m)	Delay (ms)
1	20	20	0	30
2	20	20	0	41
3	20	22	2	22
4	22	23	1	53
5	23	23	0	78
6	23	21	2	53
7	21	22	1	75
8	22	24	2	51
9	24	25	1	49
10	24	24	0	77
11	25	25	0	73
12	25	23	2	49
13	23	26	3	73
14	26	27	1	47
15	27	27	0	71
16	27	26	1	46

17	26	28	2	72
18	28	29	1	47
19	29	28	1	69
20	28	27	1	37
21	27	29	2	67
22	29	30	1	43
23	30	30	0	60
24	30	31	1	36
25	31	30	1	67
26	30	32	2	40
27	32	33	1	116
28	33	32	1	38
29	32	34	2	63
30	34	35	1	37
31	35	34	1	64
32	34	36	2	37
33	36	37	1	60
34	37	36	1	35
35	36	35	1	60
36	35	37	2	34
37	37	38	1	58
38	38	39	1	30
39	39	38	1	58
40	38	37	1	74
41	37	39	2	104
42	39	40	1	30

43	40	40	0	56
44	40	41	1	80
45	41	40	1	95
46	40	42	2	121
47	42	43	1	46
48	43	42	1	71
49	42	41	1	100
50	41	43	2	52

Pada Tabel tersebut merupakan data saat melakukan pengujian data error dan delay saat ESPCAM32 mengirim data ke *mobile*. Pada 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.

Untuk memperoleh hasil persentase *error* maka harus mencari error dengan selisih per data masing-masing pada data yang dari ESPCAM dengan data yang diterima oleh FastAPI berikut dengan rumus:

$$\text{Persentase Error} = \left(\frac{|\text{Data Kirim ESP} - \text{Data terima FastAPI}|}{\text{ESP}} \right) \times 100\%$$

Kemudian untuk memperoleh standar deviasi, perlu mencari rata-rata error dahulu dan setelah itu dapat melakukan substitusi dengan rumus standar deviasi berikut:

$$\text{Rata - rata error} = \frac{\sum |\text{ESP} - \text{FASTAPI}|}{N}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - u)^2}$$

Keterangan:

x_i = *error* per data

μ = rata-rata *error*

n = jumlah data

Sehingga pada saat melakukan perhitungan memperoleh rata-rata error(m) yaitu 1.14 m, dan melakukan substitusi pada rumus standar deviasi mendapatkan hasil standar deviasinya adalah 0.69 m.

Berdasarkan hasil akhir dari total jumlah data 50 memperoleh hasil *delay* minimal 22 ms serta *delay* maksimal 121 ms, kemudian memperoleh hasil persentase *error* secara keseluruhan adalah 3.81% sehingga akurasi informasi data akurasi informasi data dari ESPCAM32 ke *mobile* adalah 96,19%. Terjadinya data error dapat disebabkan karena data

dari ESPCAM-32 berubah sebelum terkirim, masalah kalibrasi pada ESPCAM-32, data yang hilang atau tertukar saat pengiriman data, *overlap data*, dan *noise* dalam citra pada ESPCAM-32. Terdapat beberapa faktor yang dapat menyebabkan *delay* pengiriman data melalui Bluetooth ke aplikasi di handphone, seperti kualitas sinyal Bluetooth, pengolahan data pada perangkat, proses *parsing*, beban sistem, serta interferensi elektromagnetik, sehingga besarnya nilai *delay* tidak secara langsung menunjukkan tingkat kesalahan data namun, kesalahan pengukuran dapat disebabkan oleh gangguan sensor, kondisi lingkungan, atau ketidakakuratan estimasi jarak, sedangkan *delay* umumnya dipengaruhi oleh proses transmisi komunikasi data.

V. KESIMPULAN

Berdasarkan hasil analisis dan pengujian pada implementasi aplikasi *mobile* pada sistem penjaga jarak aman kendaraan dapat disimpulkan bahwa aplikasi *mobile* yang dikembangkan mampu menampilkan informasi jarak kendaraan di depan secara jelas dan mudah dipahami oleh pengguna melalui koneksi BLE ke perangkat pengukur jarak yang terpasang pada kendaraan. Berdasarkan hasil *error* data, aplikasi memperoleh persentase *error* data 3.81% dan akurasi informasi data yang diterima pada perangkat *mobile* 96.19%, serta hasil pada *delay* menunjukkan bahwa performa aplikasi dapat dengan berjalan stabil.

REFERENSI

- [1] Nigina, "Jangan Sepelekan Pentingnya Jarak Aman Berkendara!" [Online]. Available: <https://www.asukacartv.com/jarak-aman-berkendara/>
- [2] M5Stack, "What is ESP32 CAM Used For?" [Online]. Available: https://shop.m5stack.com/blogs/news/what-is-esp32-cam-used-for?srsId=AfmBOoq9en_WLapyvnlIPtTeGE48JfK3lSe70-n0jKM_4LO3wJwE0Qqx
- [3] D. M. Irsan, T. Jl, H. Hadari, and N. Pontianak, "RANCANG BANGUN APLIKASI MOBILE NOTIFIKASI BERBASIS ANDROID UNTUK MENDUKUNG KINERJA DI INSTANSI PEMERINTAHAN."
- [4] AVSystem, "BLE | Bluetooth Low Energy – Why Everyone Is Using It?" [Online]. Available: <https://avsystem.com/blog/linkyfi/bluetooth-low-energy-ble>
- [5] P. R. Setiawan, R. A. Ramadhan, D. A. Labellapansa, P. Koresponden, : Panji, and R. Setiawan, "Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan Pelatihan Pemrograman Flutter."
- [6] R. Puspita Sari, S. Rahmayuda, J. Sistem Informasi, F. Mipa, U. Tanjungpura Jalan ProfDrH Hadari Nawawi, and P. Telp, "Coding : Jurnal Komputer dan Aplikasi IMPLEMENTASI FRAMEWORK FLUTTER PADA SISTEM INFORMASI PERPUSTAKAAN MASJID (Studi Kasus: Masjid di Kota Pontianak)."