

Pengembangan Antarmuka dan Sistem Monitoring pada Robot Delivery

Penulis 1, Windy Nadilla Sukmaputri Andriani 2, Anugerah Wibisana

Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Kepulauan Riau, Indonesia

Article Info

Article history:

Received month dd, yyyy

Revised month dd, yyyy

Accepted month dd, yyyy

Keywords:

Robot Delivery

Sistem Monitoring

Kontrol Jarak Jauh

ESP32-CAM

ABSTRACT

Pengembangan robot delivery terus dilakukan karena tingginya permintaan pengiriman barang dan makanan, bertujuan meringankan pekerjaan manusia dengan biaya operasional lebih rendah. Sistem monitoring yang efektif diperlukan untuk memantau kinerja robot, termasuk navigasi dan kontrol jarak jauh yang mudah dioperasikan pengguna. Robot ini dilengkapi dengan sistem kendali jarak jauh berbasis Wi-Fi yang memungkinkan pengendalian jarak jauh menggunakan UI. Robot ini juga memiliki modul kamera untuk mendeteksi hambatan dan kondisi sekitar. Pengujian menunjukkan jarak kontrol maksimal tanpa halangan adalah 25 meter, sementara dengan halangan tembok beton berkurang menjadi 12 meter. Pada pengujian kamera, didapat latensi dan FPS yang bervariasi sesuai dengan resolusi kamera yang digunakan.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Anugerah Wibisana

Department of Electrical Engineering, Politeknik Negeri Batam

Jl. Ahmad Yani, Kel. Teluk Tering, Kec. Batam Kota, Kota Batam, Kepulauan Riau 29461, Indonesia Email:

mail.wibisana@gmail.com

1. INTRODUCTION

Robotika merupakan cabang ilmu di bidang elektronika, mekanik dan perangkat lunak. Dewasa ini, pengembangan robot terus dilakukan[1]. Salah satunya adalah pembuatan robot delivery. Pengembangan robot delivery terus dilakukan karena banyaknya permintaan akan pengiriman barang atau makanan yang bertujuan meringankan pekerjaan manusia dengan biaya operasional yang lebih rendah.

Namun, untuk pengembangannya, diperlukan sistem monitoring yang efektif. Sistem monitoring tersebut harus mampu memantau berbagai aspek kinerja robot, termasuk navigasi, kondisi lingkungan sekitar serta pengiriman data sensor dari perangkat keras robot ke antarmuka user yang lebih mudah untuk dioperasikan oleh user[2]. Sistem monitoring banyak dikembangkan untuk membantu dalam melakukan pemantauan dan mendapatkan sebuah informasi dari sebuah tempat dengan kondisi tertentu. Sistem monitoring banyak dikembangkan untuk kebutuhan pemantauan. Salah satu contoh implementasi sistem monitoring adalah penelitian yang dilakukan untuk mengetahui keadaan cuaca di suatu tempat, data dari modul sensor suhu dan kelembapan (DHT11), tekanan barometrik (BMP180) dan sensor *Light Dependent Resistor* (LDR) dikirim secara nirkabel melalui jaringan internet dan disimpan di *cloud* serta dapat diakses secara *real-time*[3].

Oleh karena itu, penelitian ini yang bertujuan untuk mengembangkan antarmuka sistem monitoring yang komprehensif untuk Robot delivery. Antarmuka ini akan memungkinkan *user* untuk memantau pergerakan robot secara *real-time* melalui kamera *stream* dari ESP32-CAM. Dalam penelitian ini penulis menggunakan ESP32-CAM sebagai salah satu mikrokontroler yang memiliki fitur berupa *bluetooth*, WI-FI, kamera[4]. Dengan bantuan kamera yang dimiliki ESP32-CAM, sistem monitoring ini dapat menghasilkan output berupa video *stream* yang bisa diakses oleh *user*[5].

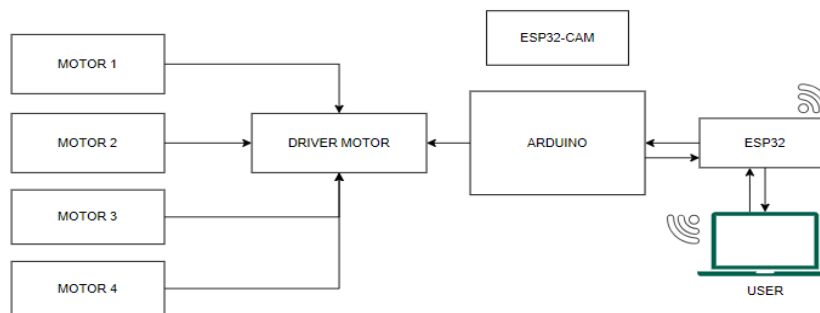
Antarmuka ini juga akan dilengkapi dengan kontroler sederhana yang memungkinkan user untuk mengontrol Robot Delivery dari jarak jauh. Sistem kontrol merupakan sebuah cara dimana pengguna dapat mengontrol, mengorganisir dan memanfaatkan sesuatu kontroler yang sudah dibuat agar robot dapat bekerja sesuai dengan keinginan *user*[6]. Pengiriman data berupa perintah untuk kontroler robot dapat dikirim menggunakan protokol MQTT. Protokol MQTT ini cukup mudah untuk diimplementasikan, protokol ini mampu mengatur beberapa *client* dengan hanya satu *server*. Sebuah penelitian yang berjudul "Analisa Perbandingan Protokol MQTT dengan HTTP pada IOT Platform Patriot" menyatakan bahwa MQTT memiliki keunggulan dalam konteks aplikasi IoT, khususnya dalam efisiensi transmisi data dan keterbatasan *bandwith* yang diperlukan[7]. Pola transmisi data yang digunakan adalah *publish-subscribe* yang membutuhkan *broker* pesan untuk mendistribusikan pesan ke *client*[8]. ESP32 adalah *module* WI-FI yang akan digunakan oleh penulis pada penelitian ini dengan menggunakan area lokal. Tujuan dari pemantauan lokal adalah untuk memastikan proses *request* dan *response* memiliki *delay* yang kecil dengan konsumsi daya rendah. Untuk itu penggunaan ESP32 sangat dibutuhkan sebagai *module* untuk pertukaran data lokal[9].

2. METHOD

Tahapan ini menjelaskan proses arsitektur perangkat keras, perangkat lunak untuk mencapai tujuan penelitian.

2.1. Perangkat Keras

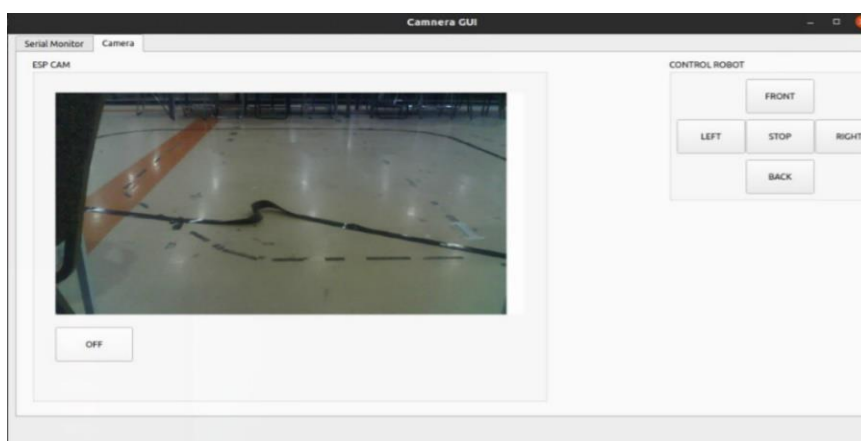
Robot yang dikembangkan merupakan robot yang dirancang agar dapat dikendalikan dari jarak jauh melalui jaringan WI-FI dengan antarmuka yang terdapat pada laptop *user* yang sudah terintegrasi dengan robot sebagai *server*. Robot delivery ini juga dilengkapi dengan ESP32-CAM sebagai webcam bagi *user* untuk monitoring robot. Hasil stream dari kamera akan ditampilkan secara *real-time* pada *Graphical User Interface* (GUI) yang sudah dirancang. Laptop *user* dapat digunakan untuk mengontrol dan memonitoring pergerakan robot dari GUI dengan cara memanggil perintah melalui laptop *user*. Blok diagram pengendalian jarak jauh delivery robot dapat dilihat pada Gambar 1.



Gambar 1. Diagram blok sistem

2.2. Perangkat Lunak

Antarmuka pengguna merupakan sebuah tempat dimana cara program dan pengguna dapat berinteraksi. Dengan adanya antarmuka pengguna dalam pembuatan sistem, pengguna diharapkan dapat dengan mudah memahami cara penggunaan sistem yang telah dibuat[10].

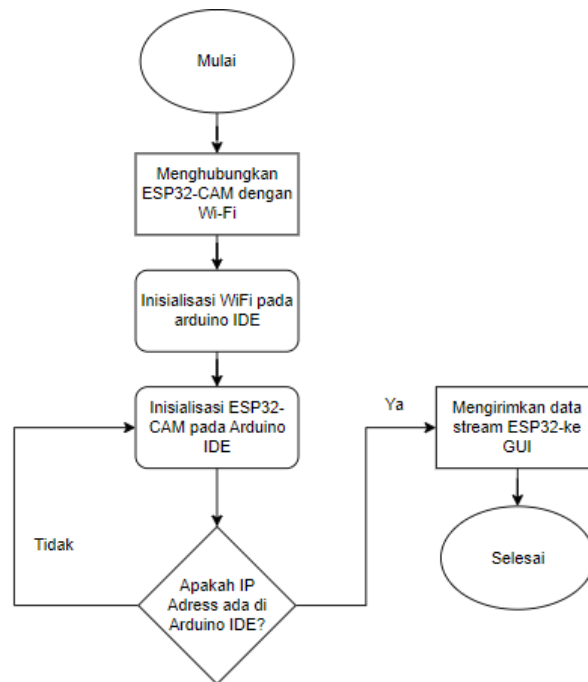


Gambar 2. Tampilan *Graphical User Interface*

Pada penelitian ini menggunakan antarmuka yang terdiri dari sebuah *frame* yang menampilkan *stream* dari ESP32-CAM untuk mendeteksi lingkungan sekitar robot. Selain itu, terdapat sebuah *widget* dengan beberapa *button* untuk mengontrol pergerakan robot seperti maju, mundur, kanan, kiri dan tombol stop yang memerintahkan robot untuk berhenti. Tampilan GUI untuk mengendalikan robot dapat dilihat seperti Gambar 2.

2.2.1. ESP32-CAM

ESP32-CAM merupakan modul kamera yang berperan penting untuk melakukan monitoring pada lingkungan sekitar robot ketika robot dijalankan. Modul kamera ini terintegrasi dengan Wi-Fi dan dapat mengirimkan video *stream* ke dalam GUI yang sudah dirancang[11].



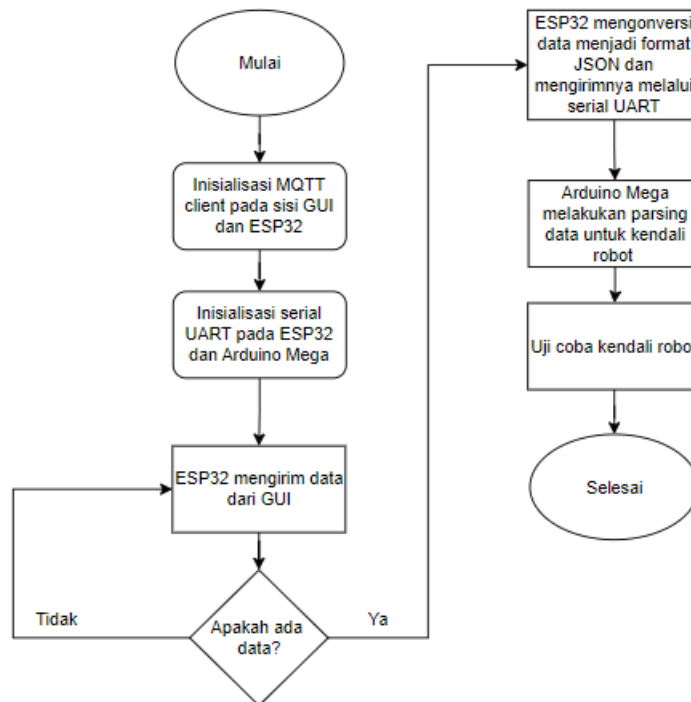
Gambar 3. Flowchart ESP32-CAM

Berdasarkan flowchart pada Gambar 3, proses dimulai dengan menghubungkan module ESP32-CAM dengan jaringan Wi-Fi yang tersedia, selanjutnya dilakukan inisialisasi Wi-Fi pada Arduino untuk komunikasi dengan *module* kamera. Kemudian dilakukan inisialisasi modul kamera dengan Arduino-IDE untuk mendapatkan IP Address. Setelah konfigurasi tersebut selesai, data *stream* dari *module* kamera ESP32-CAM bisa dikirim ke antarmuka dan ditampilkan secara *real-time* menggunakan antarmuka yang telah dirancang.

2.2.2. Sistem Kendali pada Robot

User Interface yang menjadi antarmuka untuk mengendalikan robot dari jarak jauh menggunakan program Python untuk menerima, memproses dan mengirimkan perintah ke perangkat robot. Pengiriman data antara program Python ke robot menggunakan protokol MQTT.

Message Queue Telemetry Transport (MQTT) merupakan protokol untuk komunikasi yang bergantung pada koneksi jaringan. Metode komunikasi yang digunakan pada protokol ini merupakan metode *publish/subscribe*[12]. Data yang dikirim melalui protokol ini, nantinya akan dikirim ke *broker* dan berisi topik yang akan diteruskan ke *subscriber* sesuai permintaan pengguna.

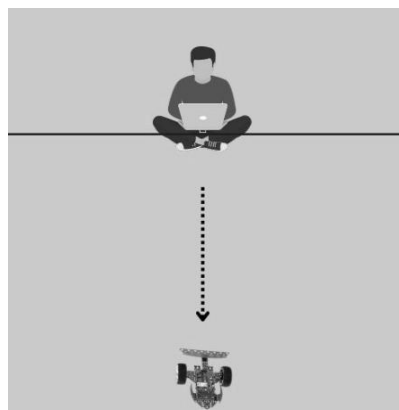


Gambar 4. Flowchart Sistem Kendali Robot

Berdasarkan flowchart sistem pada Gambar 5, proses di atas mencakup inisialisasi komunikasi antara GUI dan ESP32 melalui MQTT, serta inisialisasi komunikasi serial antara ESP32 dan Arduino Mega. Selanjutnya, ESP32 mengirimkan data yang diterima dari GUI dan melakukan konversi data tersebut menjadi format JSON sebelum mengirimkannya ke Arduino Mega melalui koneksi serial UART menggunakan protokol MQTT[13]. Di pihak Arduino Mega, data JSON diterima melalui serial UART dan diproses melalui langkah *parsing* untuk mempersiapkan data tersebut agar dapat digunakan dalam kendali robot menggunakan GUI. Dengan demikian, proses ini mencakup langkah-langkah dalam inisialisasi komunikasi antara komponen-komponen sistem, pengiriman dan konversi data, serta pemrosesan data yang diterima. Hal ini memberikan gambaran mengenai jalannya data dari GUI hingga ke pemrosesan dan penggunaan data tersebut di Arduino Mega.

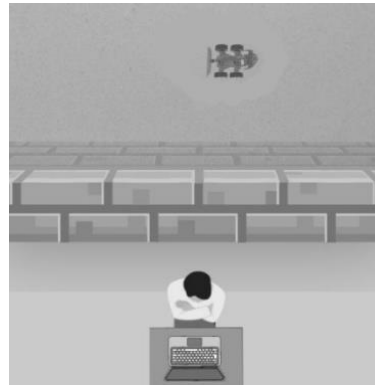
2.3. Skenario Pengambilan Data

Pengujian akan dilakukan menggunakan dua pendekatan yang berbeda, yaitu pengujian tanpa halangan di lingkungan ruang terbuka bebas hambatan, serta pengujian dengan keberadaan halangan di lingkungan yang terbatas dan dibatasi oleh tembok beton. Ilustrasi pengujian ditampilkan pada gambar di bawah ini.



Gambar 5. Ilustrasi Pengujian Tanpa Halangan

Pada Gambar 5 di atas menampilkan ilustrasi pengujian kendali robot jarak jauh tanpa halangan yang terbuka. Robot dikendalikan untuk menjauhi base station, dan selama uji coba ini, robot terus bergerak hingga mencapai jarak maksimal, dimana pada titik ini robot tidak lagi dapat dikendalikan.



Gambar 6. Ilustrasi Pengujian dengan Halangan

Sementara pada pengujian kedua, seperti tampak pada Gambar 6 dilakukan eksperimen untuk menguji kendali robot dalam kondisi dengan adanya halangan. Pada bagian ini, uji coba dilakukan dengan base station server berada di luar ruangan, sementara robot berada di dalam ruangan yang dibatasi oleh tembok beton. Dalam pengujian ini, robot dikendalikan dari luar ruangan menggunakan stream kamera dari modul esp32-cam yang terhubung ke GUI yang sama dengan GUI yang digunakan untuk kendali robot secara jarak jauh.

3. RESULTS AND DISCUSSION

Pada bagian ini akan membahas hasil penelitian. Yang dimana penelitian ini dimulai dari membahas hasil pengujian pertama yaitu pengujian jarak maksimal pengendalian robot dengan halangan maupun tanpa halangan, lalu pengujian respon antarmuka grafis (GUI) terhadap robot, dan kemampuan *streaming* video dari ESP32-CAM.

3.1. Pengujian Kendali Robot

Pada bagian ini, uji coba dilakukan dengan dua cara yaitu tanpa halangan dan dengan halangan berupa tembok beton. Pengujian ini dilakukan agar penulis mengetahui kemampuan interaksi antara antarmuka dengan robot di berbagai kondisi, yaitu tanpa halangan dan dengan halangan.

3.1.1. Pengujian Tanpa Halangan

Uji coba pengendalian robot terhadap kualitas Wi-Fi dilakukan pada ruangan terbuka tanpa halangan agar penulis dapat mengambil data jarak maksimal kendali robot Delivery tanpa halangan sebagai gangguan penyebaran sinyal. Hasil uji coba pengendalian Delivery robot tanpa halangan disajikan pada Tabel 1.

Tabel 1. Pengujian Tanpa Halangan

No	Jarak (Meter)	Latensi (ms)	Status	Keterangan
1	5	30	Terhubung	Respons baik
2	10	30	Terhubung	Respons baik
3	15	30	Terhubung	Respons baik
4	20	32	Terhubung	Respons baik
5	22	35	Terhubung	Respons baik
6	25	45	Terhubung	Jarak Maksimal

Pada Tabel 1. Terdapat hasil dari uji coba pengendalian Delivery robot dengan modul Esp32 pada mikronkontroler. Pengujian yang dilakukan pada jarak 5 hingga 22 meter, memiliki latensi sebesar 30 ms yang berarti komunikasi antara GUI dan robot berjalan dengan cepat. Respons dari robot baik dan perintah yang diberikan hampir dieksekusi saat itu juga. Pada jarak 25 meter, latensi mengalami peningkatan menjadi 45 ms. Meski begitu, koneksi tetap stabil tetapi jarak tersebut merupakan jarak maksimal pengendalian robot tanpa penundaan yang signifikan. Pada pengujian ini, rata-rata latensi yang didapat adalah 33,67 ms. Rata-rata tersebut dihitung berdasarkan latensi setiap jarak yang diuji, dari 5 hingga 25 meter.

3.1.2. Pengujian dengan Halangan

Uji coba pengendalian robot terhadap kualitas Wi-Fi kembali dilakukan pada ruangan berbeda, robot berada di dalam ruangan sedangkan laptop berada di luar ruangan yang dibatasi dengan dinding beton sebagai halangan agar

penulis dapat mengambil data jarak maksimal kendali robot dengan dinding beton sebagai halangannya. Hasil uji coba pengendalian robot disajikan pada Tabel 2.

Tabel 2. Pengujian dengan Halangan

No	Jarak	Latency (ms)	Status	Keterangan
1	2	45	Terhubung	Respons baik
2	4	45	Terhubung	Respons baik
3	6	50	Terhubung	Respons baik
4	8	55	Terhubung	Respons baik
5	10	60	Terhubung	Respons baik
6	12	66	Terhubung	Respons baik

Pada Tabel 2. Pengujian menunjukkan bahwa robot dapat dikendalikan dengan baik hingga jarak 12 meter dari ruangan yang berbeda dengan dinding beton sebagai halangannya. Latensi cukup rendah dan koneksi tetap stabil hingga jarak 12 meter. Dari tabel pengujian di atas, dapat disimpulkan bahwa jarak maksimal untuk pengendalian robot dengan halangan berupa beton hanya sejauh 12 meter dari base station. Dengan latensi rata-rata 53.5 ms, interaksi antar *base station* dengan robot dianggap cukup responsif karena koneksi yang tetap stabil.

3.2. Pengujian Respon Robot Terhadap *User Interface*

Pada bagian ini, dilakukan uji coba terhadap antarmuka yang telah dikembangkan untuk mengontrol robot dan memantau lingkungan sekitar robot. Uji coba ini bertujuan untuk mengevaluasi fungsi dari antarmuka pengguna yang terintegrasi dengan robot yang digunakan. Hasil uji terlihat pada Tabel 3. sebagai berikut.

Tabel 3. Pengujian Antarmuka

No	Fungsi	Deskripsi	Lama Respon Tombol (ms)	Hasil Pengujian
1	Tombol Maju	Tombol untuk menggerakkan robot maju	10	Berhasil
2	Tombol Mundur	Tombol untuk menggerakkan robot mundur	10	Berhasil
3	Tombol Kanan	Tombol untuk menggerakkan robot ke kanan	10	Berhasil
4	Tombol Kiri	Tombol untuk menggerakkan robot ke kiri	10	Berhasil
5	Tombol Stop	Tombol untuk menghentikan semua gerakan robot	10	Berhasil
6	Frame Kamera	Frame untuk menampilkan stream video dari ESP32-CAM	30	Berhasil

Berdasarkan Tabel 3. di atas, tersaji hasil pengujian respons tombol GUI untuk semua fungsi kontrol baik maju, mundur, kanan, kiri dan stop. Lama respon tombol untuk setiap fungsi adalah 10 ms. Hal ini menunjukkan bahwa setiap tombol ditekan, respon yang didapat relatif singkat. Berbeda dengan tombol ON untuk menampilkan stream video dari ESP32-CAM yang memiliki lama respon sebesar 30 ms. Hal ini menunjukkan bahwa fungsi ini memiliki respon yang sedikit lebih lama dibandingkan fungsi tombol lainnya. Kecepatan respon ini dapat dipengaruhi

oleh kecepatan proses komunikasi antarmuka, kecepatan pemrosesan instruksi oleh sistem kendali dan kecepatan respon robot itu sendiri.

3.3. Pengujian Kemampuan Video Stream ESP32-CAM

Pada bagian ini dilakukan uji coba video *stream* menggunakan modul ESP32-CAM yang diletakkan pada robot guna mengetahui kemampuan stream kamera ESP32-CAM dengan resolusi maksimal pada GUI. Pengujian dilakukan sebanyak 15 kali dengan 3 resolusi kamera yang berbeda dengan hasil seperti Tabel 4. di bawah ini.

Tabel 4. Pengujian ESP32-CAM

Jarak (Meter)	Resolusi	Latency (ms)	FPS
1	HD 1280 x 720	230	4.17
2	HD 1280 x 720	238	4.13
3	HD 1280 x 720	400	3.85
4	HD 1280 x 720	600	3.28
5	HD 1280 x 720	630	2.68
1	VGA 640 x 128	70	12.51
2	VGA 640 x 128	76	12.51
3	VGA 640 x 128	108	11.54
4	VGA 640 x 128	136	11.11
5	VGA 640 x 128	148	9.62
1	QVGA 320 x 240	30	17.24
2	QVGA 320 x 240	30	17.3
3	QVGA 320 x 240	32	17.32
4	QVGA 320 x 240	38	20.21
5	QVGA 320 x 240	45	23.21

Pada tabel pengujian di atas, setiap resolusi terjadi peningkatan latensi dan penurunan FPS sesuai dengan peningkatan jarak pengujian. Resolusi HD menghasilkan latensi yang lebih tinggi dan FPS yang lebih rendah terutama jika pengujian mencapai jarak yang lebih jauh. Pada resolusi QVGA, latensi yang dihasilkan agak lebih rendah dan FPS mengalami kenaikan dan memiliki performa yang lebih konsisten di setiap jarak. Resolusi VGA merupakan resolusi dengan performa lebih baik dari HD tetapi tidak sebaik QVGA karena memiliki lebih banyak data yang harus diproses *per-frame* dibandingkan dengan QVGA.

Secara umum, semakin tinggi resolusi kamera, semakin besar latensi yang terjadi dan semakin rendah FPS-nya karena kamera harus menangani lebih banyak data dalam setiap frame. Sebaliknya, semakin rendah resolusi, semakin kecil latensinya dan semakin tinggi FPS-nya karena jumlah data yang lebih sedikit yang diproses dan ditransmisikan oleh kamera.

4. CONCLUSION

Penelitian ini dilakukan dengan tujuan mengembangkan antarmuka sistem monitoring yang komprehensif untuk Robot Delivery. Penelitian ini berhasil menunjukkan bahwa user dapat mengendalikan robot dari antarmuka yang telah dirancang. Uji coba pengendalian robot dilakukan dengan dua kondisi yaitu dengan halangan dan tanpa halangan. Berdasarkan uji coba penelitian, jarak maksimal pengendalian tanpa halangan mencapai 25 meter dengan latensi paling besar yaitu 45 ms di jarak maksimal, sementara pengujian dengan halangan berupa dinding beton mencapai 12 meter latensi yang lebih lama yaitu 66 ms di jarak maksimal karena adanya halangan fisik berupa dinding yang mempengaruhi kinerja sistem dalam memberikan respons terhadap perintah pengendalian. Pada pengujian kamera dilakukan uji coba terhadap 3 resolusi kamera, dari tiga resolusi yang diuji, didapatkan bahwa resolusi QVGA memberikan latensi terendah dengan fps yang cukup bagus, yaitu 30 ms. Hal ini menunjukkan bahwa pemilihan resolusi kamera dapat memengaruhi respons sistem dalam memproses informasi visual.

ACKNOWLEDGEMENTS

Penulis mengucapkan terima kasih kepada dosen pembimbing, Bapak Anugerah Wibisana S.Tr.T., M.Tr.T. yang telah membimbing dan memberi arahan selama penelitian berlangsung. Penulis juga mengucapkan terima kasih kepada rekan-rekan yang turut membantu dalam menyelesaikan penelitian ini serta kepada Bareleng Robotics and Artificial Intelligence Lab (BRAIL) yang telah menyediakan tempat dan fasilitas dalam melakukan penelitian.

REFERENCES

- [1] A. Isrofi, S. N. Utama, and O. V. Putra, "RANCANG BANGUN ROBOT PEMOTONG RUMPUT OTOMATIS MENGGUNAKAN WIRELESS KONTROLER MODUL ESP32-CAM BERBASIS INTERNET of THINGS (IoT)," *Jurnal Teknoinfo*, vol. 15, no. 1, p. 45, Jan. 2021, doi: 10.33365/jti.v15i1.675.
- [2] F. R. Hadiputra and T. K. Priyambodo, "Purwarupa Pengendalian Jarak Jauh Pada Mobile Robot Berbasis Web Melalui Jaringan Wireless TCP/IP," *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, vol. 6, no. 1, p. 105, Apr. 2016, doi: 10.22146/ijeis.15247.
- [3] Y. Ardiyanto and M. Yusvin Mustar, "RANCANG BANGUN GRAPHICAL USER INTERFACE SEBAGAI SISTEM MONITORING NIRKABEL PENDETEKSI HUJAN, SUHU DAN KELEMBABAN," *Jurnal Edukasi Elektro*, vol. 4, no. 1, 2020, [Online]. Available: <http://journal.uny.ac.id/index.php/jee/>
- [4] I. Agustine Cahyaningtyas and A. Stefanie, "IMPLEMENTASI ESP32 CAM DAN KODULAR BERBASIS ANDROID UNTUK MONITORING SMART GARDEN," 2023.
- [5] E. Fadly, S. A. Wibowo, and A. P. Sasmito, "SISTEM KEAMANAN PINTU KAMAR KOS MENGGUNAKAN FACE RECOGNITION DENGAN TELEGRAM SEBAGAI MEDIA MONITORING DAN CONTROLLING," 2021.
- [6] H. A. Rochman, R. Primananda, and H. Nurwasito, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," 2017. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] N. P. Windryani, N. Bogi, and R. Mayasari, "ANALISA PERBANDINGAN PROTOKOL MQTT DENGAN HTTP PADA IOT PLATFORM PATRIOT COMPARISON ANALYSIS BETWEEN MQTT AND HTTP PROTOCOL IN PATRIOT IOT PLATFORM."
- [8] G. Yudha Saputra, A. Denhas Afrizal, F. Khusnu Reza Mahfud, F. Angga Pribadi, and F. Jati Pamungkas, "PENERAPAN PROTOKOL MQTT PADA TEKNOLOGI WAN (STUDI KASUS SISTEM PARKIR UNIVERISTAS BRAWIJAYA)," 2017.
- [9] M. Rizal, M. S. Hadis, R. Angriawan, and A. Arifin, "EVALUASI KINERJA BLUETOOTH PADA MODUL ESP32 DI LINGKUNGAN LINE OF SIGHT." [Online]. Available: <https://ojs.unm.ac.id/JESSI/index>
- [10] W. S. Ningsih, H. Muslimah Az-Zahra, and T. Afrianto, "Perancangan Antarmuka Pengguna Sistem Informasi Monitoring dan Evaluasi Prakerin berbasis Website menggunakan Metode Human Centered Design (Studi Kasus: SMKN 2 Sragen)," 2021. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [11] M. Kelvin Difa and J. Endri, "Implementasi Sistem Pengenalan Wajah Sebagai Automatic Door Lock Menggunakan Modul ESP32 CAM," *PATRIA ARTHA Technological Journal* •, vol. 5, 2021.
- [12] A. Kurnianto, J. Dedy Irawan, and F. X. Ariwibisono, "PENERAPAN IOT (INTERNET OF THINGS) UNTUK CONTROLLING LAMPU MENGGUNAKAN PROTOKOL MQTT BERBASIS WEB," 2022. [Online]. Available: <https://www.embedded.com/>
- [13] Z. B. Abilovani, W. Yahya, and F. A. Bakhtiar, "Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT," 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>

BIOGRAPHIES OF AUTHORS
