

Pengembangan Sistem Kontrol Palang Otomatis dan Informasi Slot Parkir Real-Time Menggunakan LCD dan ESP8266

1st Khalishah

Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia

khaliishah@student.telkomuniversity.ac.id

2nd Agus Virgono

Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia

avirgono@telkomuniversity.ac.id

line 1: 3rd Randy Erfa Saputra

Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia

resaputra@telkomuniversity.ac.id

Abstrak — Penelitian ini merancang sistem monitoring parkir otomatis yang mengintegrasikan sensor *infrared* TSOP1738, sensor ultrasonik HC-SR04, Arduino Uno, dan modul ESP8266 untuk mengirimkan data status parkir ke server MySQL secara *real-time*. Informasi parkir ditampilkan melalui LCD dan data pergerakan kendaraan disimpan dalam *database*. Pengujian menunjukkan bahwa waktu pembukaan palang parkir rata-rata 344,27 ms, dan waktu kirim data dari ESP8266 hingga tersimpan di *database* tercatat 3,60 ms. Sistem menunjukkan stabilitas tinggi dalam komunikasi data dan keakuratan dalam menampilkan status parkir secara *real-time*, baik dalam kondisi parkir penuh maupun saat slot tersedia.

Kata kunci— Sistem parkir otomatis, sensor *infrared*, ESP8266, LCD, MySQL, *real-time*

I. PENDAHULUAN

Kendala utama dalam sistem parkir tradisional adalah keterlambatan informasi mengenai ketersediaan slot serta waktu yang terbuang saat mencari tempat parkir. Teknologi sensor dan komunikasi nirkabel membuka peluang pengembangan sistem monitoring parkir berbasis *infrared*[1]. Sistem ini memanfaatkan pemancar *infrared* sebagai identifikasi kendaraan dan sensor penerima TSOP1738 untuk memicu pembukaan palang otomatis menggunakan motor servo. Selain itu, status slot parkir ditampilkan melalui LCD yang dikendalikan oleh Arduino Uno, memberikan informasi langsung kepada pengguna mengenai kondisi parkir[2]. Sistem ini bekerja secara otomatis, sehingga kebutuhan akan tenaga kerja tidak diperlukan yang berarti biaya operasional untuk tenaga kerja juga dapat dihemat[3] [4].

Agar informasi parkir dapat dipantau secara luas dan tersimpan dengan rapi, sistem ini dilengkapi dengan modul ESP8266 yang berfungsi mengirimkan data status parkir secara *real-time* ke *database* MySQL. Dengan begitu, sistem tidak hanya menampilkan informasi secara lokal melalui LCD, tetapi juga memungkinkan pengelola dan pengguna untuk mengakses data. Penelitian ini bertujuan untuk merancang sistem monitoring parkir *real-time* yang terintegrasi antara sensor, Arduino Uno, motor servo, LCD, dan ESP8266, sehingga menghasilkan sistem yang efisien,

informatif, dan mendukung pengelolaan parkir berbasis data[5].

II. KAJIAN TEORI

A. Identifikasi Menggunakan *Infrared*

Sistem ini menggunakan kombinasi LED *infrared* sebagai pemancar dan TSOP1738 sebagai penerima. Pemancar dikendalikan oleh kendaraan dan mengirimkan sinyal yang akan diterima oleh sensor TSOP1738. Teknologi ini memungkinkan pengenalan kendaraan secara otomatis melalui kode unik yang dipancarkan oleh sensor *infrared*, sehingga meningkatkan keamanan dan efisiensi sistem parkir[6], yang kemudian memicu pembukaan palang parkir.

B. Motor Servo SG90

Motor servo SG90 digunakan sebagai penggerak palang parkir otomatis. Motor ini dapat diarahkan pada sudut tertentu menggunakan sinyal dari mikrokontroler. Motor servo biasanya beroperasi pada sudut tertentu dan tidak bekerja secara kontinu, tetapi dapat dimodifikasi untuk memungkinkan operasi kontinu[7]. Dalam sistem parkir otomatis, motor servo hanya akan aktif setelah kendaraan dikenali dan slot tersedia. Keunggulan utama dari motor ini adalah presisi pergerakan dan konsumsi energi yang rendah.

C. LCD 16x2

LCD berfungsi sebagai media informasi visual bagi pengguna. Dalam sistem parkir, LCD menampilkan status slot seperti “TERSEDIA” atau “PENUH”. Penggunaan modul LCD I2C memungkinkan koneksi dua kabel sederhana ke Arduino dan efisien dalam menampilkan data sensor secara *real-time*[8]. Hal ini membantu pengguna dalam mengambil keputusan secara cepat saat memasuki area parkir.

D. Modul ESP8266

ESP8266 adalah modul WiFi berbiaya rendah yang memungkinkan perangkat Arduino untuk terhubung ke jaringan internet. Dalam sistem ini, ESP8266 digunakan untuk mengirimkan data status parkir secara *real-time* ke *database* MySQL melalui protokol HTTP GET/POST. Modul ini mendukung protokol TCP/IP, hemat daya, dan

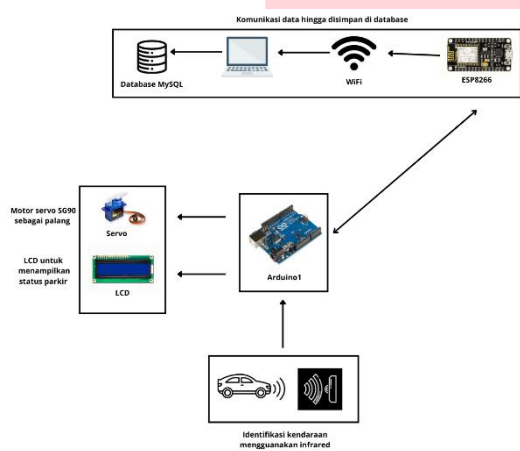
cocok untuk integrasi dengan sistem monitoring berbasis IoT karena ukurannya yang ringkas dan kemudahan integrasi[9].

E. Arduino Uno

Arduino Uno berperan sebagai pusat kendali seluruh sistem monitoring parkir. Mikrokontroler ini mengolah data dari sensor *infrared*, mengaktifkan motor servo, dan mengirimkan informasi ke LCD serta modul komunikasi ESP8266.

III. METODE

Penelitian ini dimulai dari perancangan sistem, perakitan komponen, dan pengujian. Sistem dirancang menggunakan Arduino Uno, ESP8266, motor servo SG90, dan LCD 16x2. Data diperoleh dari pengujian waktu pembukaan palang parkir dan waktu kirim data dari ESP8266 hingga tersimpan di *database*.



GAMBAR 1

Rancangan Sistem Kontrol Palang dan Monitoring Parkir

Gambar 1 menggambarkan rancangan sistem kontrol palang otomatis dan monitoring status slot parkir secara *real-time*. Kendaraan dikenali oleh sistem melalui sensor infrared, dan Arduino Uno mengatur pembukaan palang menggunakan motor servo. Status slot parkir ditampilkan melalui LCD, serta dikirimkan ke server *database* MySQL secara *real-time* menggunakan modul WiFi ESP8266.



GAMBAR 2

Implementasi servo SG90 dan tampilan LCD

Gambar 2 menampilkan area pintu masuk parkir yang dilengkapi dengan palang otomatis dan layar LCD. Layar tersebut berfungsi untuk menampilkan informasi kepada pengguna, seperti status sistem atau ucapan selamat datang. Palang akan terbuka secara otomatis jika sistem mendeteksi

kendaraan yang valid berdasarkan sinyal *infrared* yang diterima

ID	status
1	masuk
2	keluar
3	masuk
4	keluar
5	masuk
6	keluar
7	masuk
8	keluar
9	masuk
10	keluar

GAMBAR 3

Database kode unik (ID) terdaftar

Gambar 3 menunjukkan tabel pengguna berisi data utama yang mencatat setiap kendaraan yang terdeteksi oleh sistem, dengan dua atribut penting yaitu kode dan status. Atribut kode merupakan ID unik untuk setiap pengguna atau kendaraan, sedangkan status menunjukkan apakah kendaraan dalam kondisi masuk atau keluar dari area parkir.

status	terakhir_update
masuk	2025-07-11 10:41:30
keluar	2025-07-11 10:41:30
masuk	2025-07-11 10:41:30
keluar	2025-07-11 10:41:30
masuk	2025-07-11 10:41:30
keluar	2025-07-11 10:41:30
masuk	2025-07-11 10:41:30
keluar	2025-07-11 10:41:30
masuk	2025-07-11 10:41:30
keluar	2025-07-11 10:41:30

GAMBAR 4

Database status slot parkir

Gambar 4 menunjukkan tabel status slot parkir yang menyajikan kondisi *real-time* setiap slot parkir melalui dua atribut, yaitu status dan terakhir *update*. Kolom status menandai apakah slot sedang terisi atau kosong, sedangkan kolom terakhir *update* mencatat tanggal dan waktu pembaruan terakhir secara otomatis setiap kali sensor mendeteksi perubahan.

kode	waktu_masuk	waktu_keluar	durasi
1	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
2	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
3	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
4	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
5	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
6	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
7	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
8	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
9	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11
10	2025-07-11 10:42:20	2025-07-11 10:42:40	00:00:11

GAMBAR 5

Database log parkir

Gambar 5 menunjukkan tabel log parkir merekam setiap siklus parkir secara lengkap melalui empat atribut, yaitu kode, waktu *masuk*, waktu *keluar*, dan durasi. Kolom kode bertindak sebagai referensi ke identitas kendaraan pada tabel pengguna, sementara waktu *masuk* dan waktu *keluar* mencatat tanggal dan waktu saat kendaraan memasuki dan meninggalkan area parkir. Dari waktu *masuk* dan waktu *keluar* ini, sistem secara otomatis menghitung durasi dari detik, menit hingga jam

Untuk memastikan sistem berjalan sesuai dengan tujuan yang telah dirancang, dilakukan beberapa serangkaian pengujian. Pengujian ini bertujuan untuk mengevaluasi kinerja, keakuratan, serta keandalan sistem parkir otomatis. Adapun pengujian yang dilakukan meliputi:

A. Pengujian Waktu Respon Pembukaan Palang

Pengujian ini bertujuan untuk mengukur seberapa cepat palang terbuka setelah sistem menerima dan memvalidasi kode kendaraan.

B. Pengujian Tampilan Informasi Pada LCD

LCD diuji untuk menampilkan informasi sistem seperti status parkir, nomor slot, dan notifikasi "Parkir Penuh".

C. Pengujian Latensi Waktu Kirim Data Dari ESP8266 Hingga Tersimpan di Database

Pada pengujian ini, sistem diuji dengan menjalankan alur penuh mulai dari kendaraan mengirimkan kode masuk atau keluar, hingga data tersebut tersimpan di dalam database melalui NodeMCU (ESP8266). Waktu kirim dan waktu simpan dicatat, lalu selisihnya dihitung sebagai latensi komunikasi sistem. Pengujian dilakukan berulang kali untuk memastikan bahwa sistem dapat memproses dan menyimpan data parkir secara stabil dan responsif, baik untuk proses masuk maupun keluar kendaraan.

IV. HASIL DAN PEMBAHASAN

Evaluasi sistem dilakukan untuk mengukur seberapa efektif kontrol otomatis palang parkir dan pengelolaan informasi slot parkir dalam skenario nyata. Pengujian difokuskan pada waktu respon sistem terhadap sinyal masuk, ketepatan penyajian informasi melalui LCD, serta latensi komunikasi data antara NodeMCU dan database. Hasil pengujian ini memberikan gambaran menyeluruh terhadap performa dan kestabilan sistem dalam mendukung manajemen parkir otomatis. Adapun pengujian yang dilakukan meliputi:

A. Hasil Pengujian Waktu Respon Pembukaan Palang

Waktu respon dalam suatu sistem mencakup keseluruhan durasi mulai dari saat sensor mendeteksi keberadaan objek, dilanjutkan dengan proses pengolahan data oleh mikrokontroler, hingga pelaksanaan aksi oleh aktuator seperti servo[10].

TABEL 1
Latensi waktu kirim dan terima kode

No.	Waktu Perintah Buka Palang	Waktu Palang Terbuka	Latensi (ms)
1.	23:01:42.987	23:01:43.300	313 ms
2.	23:01:48.912	23:01:49.284	372 ms
3.	23:01:54.877	23:01:55.204	327 ms
4.	23:02:00.779	23:02:01.117	338 ms
5.	23:02:06.872	23:02:07.210	338 ms
6.	23:02:12.849	23:02:13.199	350 ms
7.	23:02:18.912	23:02:19.253	341 ms
8.	23:02:24.850	23:02:25.208	358 ms
9.	23:02:30.770	23:02:31.112	342 ms
10.	23:02:36.831	23:02:37.198	367 ms
11.	23:02:42.774	23:02:43.114	340 ms
12.	23:02:48.911	23:02:49.251	340 ms

13.	23:02:55.289	23:02:55.628	339 ms
14.	23:03:01.268	23:03:01.637	369 ms
15.	23:03:07.296	23:03:07.626	330 ms
Maksimum			372 ms
Minimum			313 ms
Rata-rata			344,27 ms
Standar Deviasi			16,38 ms

Berdasarkan Tabel 1, pengujian data yang dikumpulkan mencakup waktu saat perintah buka dikirim, waktu aktual palang mulai terbuka, dan selisih waktu dalam satuan milidetik (ms) sebagai indikator waktu respon sistem. Dari total 15 kali pengujian, waktu respon sistem menunjukkan variasi antara 313 ms hingga 372 ms, dengan rata-rata waktu respon sebesar 344,27 ms. Nilai waktu respon tercepat tercatat sebesar 313 ms, sedangkan waktu respon terlama mencapai 372 ms. Untuk mengukur kestabilan sistem, dihitung pula nilai standar deviasi sebesar 16,38 ms, yang menunjukkan bahwa fluktuasi waktu respon sistem tergolong rendah dan masih dalam rentang toleransi untuk sistem kontrol berbasis mikrokontroler.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem palang otomatis bekerja secara responsif dan konsisten, dengan latensi yang relatif kecil dan stabil. Hal ini mengindikasikan bahwa komunikasi antara perangkat lunak pengontrol dan aktuator palang berjalan efektif.

B. Hasil Pengujian Tampilan Informasi Pada LCD

Pengujian ini dilakukan untuk memastikan bahwa LCD 16x2 pada sistem mampu menampilkan informasi dengan benar sesuai kondisi aktual yang terjadi.

TABEL 2
Pengujian tampilan LCD

Nomor Slot Sudah Terisi	Hasil yang Diharapkan	Hasil Aktual	Status
0	Slot Tersedia: SLOT 1	Slot Tersedia: SLOT 1	Lulus
1	Slot Tersedia: SLOT 2	Slot Tersedia: SLOT 2	Lulus
1, 2	Slot Tersedia: SLOT 3	Slot Tersedia: SLOT 3	Lulus
1, 2, 3	Slot Tersedia: SLOT 4	Slot Tersedia: SLOT 4	Lulus
1, 2, 3, 4	Slot Tersedia: SLOT 5	Slot Tersedia: SLOT 5	Lulus
1, 2, 3, 4, 5	Slot Tersedia: SLOT 6	Slot Tersedia: SLOT 6	Lulus
1, 2, 3, 4, 5, 6	PARKIR PENUH	PARKIR PENUH	Lulus

Berdasarkan Tabel 2, terdapat pengujian sistem yang diminta untuk menampilkan slot parkir yang tersedia berdasarkan slot yang telah terisi, dimulai dari kosong hingga penuh. Pada kondisi awal saat belum ada slot yang terisi, sistem menampilkan informasi "Slot Tersedia: SLOT 1".

Setelah SLOT 1 terisi, sistem mengalihkan informasi ke “Slot Tersedia: SLOT 2”, dan begitu seterusnya. Ketika slot 1 hingga 5 telah terisi, sistem menampilkan “Slot Tersedia: SLOT 6”. Terakhir, saat semua enam slot telah terisi, sistem secara otomatis menampilkan pesan “PARKIR PENUH”.

Seluruh hasil tampilan yang diberikan oleh sistem menunjukkan kesesuaian dengan kondisi aktual slot yang telah terisi. Dengan demikian, sistem dapat disimpulkan telah bekerja secara akurat dan responsif, menampilkan slot kosong secara *real-time* dan menghentikan akses saat kapasitas penuh tercapai.

C. Hasil Latensi Waktu Kirim Data Dari ESP8266 Hingga Tersimpan di Database

Pengujian dilakukan dengan mengirimkan kode masuk dan keluar secara berulang melalui Arduino ke NodeMCU (ESP8266), yang kemudian meneruskannya ke server menggunakan HTTP. Waktu pengiriman dicatat oleh NodeMCU menggunakan fungsi `millis()`, sedangkan waktu penyimpanan akhir dicatat di sisi server menggunakan timestamp saat data masuk ke *database*. Selisih antara kedua waktu tersebut dihitung sebagai latensi komunikasi sistem.

TABEL 3
Latensi waktu kirim dan terima kode

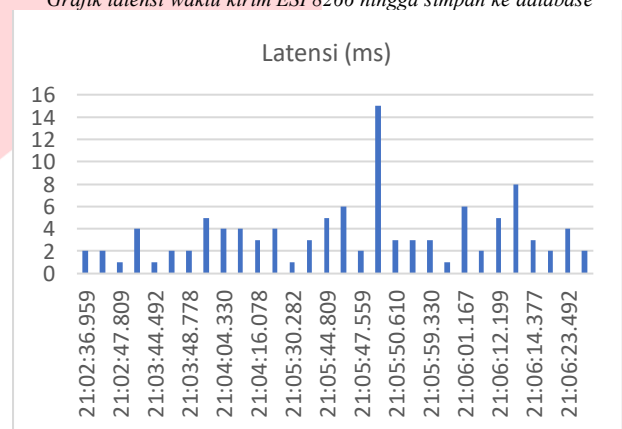
No.	Waktu Kirim (ESP8266)	Waktu Simpan (MySQL)	Latensi (ms)
1.	21:02:36.959	21:02:36.961	2 ms
2.	21:02:40.116	21:02:40.118	2 ms
3.	21:02:47.809	21:02:47.810	1 ms
4.	21:03:44.146	21:03:44.150	4 ms
5.	21:03:44.492	21:03:44.493	1 ms
6.	21:03:45.046	21:03:45.048	2 ms
7.	21:03:48.778	21:03:48.780	2 ms
8.	21:04:04.102	21:04:04.107	5 ms
9.	21:04:04.330	21:04:04.334	4 ms
10.	21:04:05.502	21:04:05.506	4 ms
11.	21:04:16.078	21:04:16.081	3 ms
12.	21:05:29.524	21:05:29.528	4 ms
13.	21:05:30.282	21:05:30.283	1 ms
14.	21:05:38.130	21:05:38.133	3 ms
15.	21:05:44.809	21:05:44.814	5 ms
16.	21:05:45.352	21:05:45.358	6 ms
17.	21:05:47.559	21:05:47.561	2 ms
18.	21:05:45.885	21:05:45.900	15 ms
19.	21:05:50.610	21:05:50.613	3 ms
20.	21:05:58.906	21:05:58.909	3 ms
21.	21:05:59.330	21:05:59.333	3 ms
22.	21:05:59.854	21:05:59.855	1 ms
23.	21:06:01.167	21:06:01.173	6 ms
24.	21:06:05.372	21:06:05.374	2 ms
25.	21:06:12.199	21:06:12.204	5 ms
26.	21:06:13.276	21:06:13.284	8 ms
27.	21:06:14.377	21:06:14.380	3 ms
28.	21:06:18.036	21:06:18.038	2 ms
29.	21:06:23.492	21:06:23.496	4 ms
30.	21:06:25.093	21:06:25.095	2 ms
Maksimum			15 ms
Minimum			1 ms
Rata-rata			3,60 ms
Standar Deviasi			2,69 ms

Berdasarkan Tabel 3, hasil pengujian menunjukkan bahwa latensi paling cepat adalah 1 ms, sedangkan latensi paling lambat mencapai 15 ms. Nilai rata-rata latensi adalah 3,60 ms dengan standar deviasi sebesar 2,69 ms, yang berarti variasi antar percobaan relatif kecil.

Sebagian besar data dikirim dan tersimpan hanya dalam waktu 1 hingga 6 milidetik, menunjukkan bahwa proses komunikasi antara ESP8266 dan *database* berjalan sangat cepat dan efisien. Bahkan pada percobaan dengan latensi tertinggi, yaitu 15 ms, sistem masih tergolong responsif dan layak untuk aplikasi *real-time* seperti sistem parkir otomatis. Secara keseluruhan, pengujian membuktikan bahwa ESP8266 mampu mengirim data ke MySQL dengan waktu tunda yang sangat kecil dan stabil, sehingga dapat diandalkan untuk sistem monitoring dan pencatatan otomatis.

GAMBAR 6

Grafik latensi waktu kirim ESP8266 hingga simpan ke database



Gambar 6 menunjukkan grafik latensi atau jeda waktu dalam milidetik yang terjadi sejak ESP8266 mengirimkan data hingga proses penyimpanan ke *database* selesai. Grafik ini menggambarkan performa jaringan dan kecepatan respons server dalam menangani permintaan.

V. KESIMPULAN

Berdasarkan hasil pengujian terhadap sistem parkir otomatis yang dikembangkan, dapat disimpulkan bahwa sistem telah bekerja secara responsif, akurat, dan stabil. Pertama, pengujian terhadap waktu respons pembukaan palang menunjukkan bahwa sistem memiliki rata-rata latensi sebesar 344,27 ms dengan standar deviasi 16,38 ms, yang mengindikasikan kestabilan respons aktuator terhadap perintah buka dari mikrokontroler. Kedua, tampilan informasi slot parkir pada LCD 16x2 menunjukkan kesesuaian penuh dengan kondisi aktual, di mana sistem mampu memperbarui informasi slot yang tersedia secara *real-time* dan secara otomatis menampilkan status "PARKIR PENUH" ketika kapasitas maksimum tercapai. Ketiga, hasil pengujian latensi komunikasi antara ESP8266 dan *database* MySQL mencatat rata-rata waktu tunda sebesar 3,60 ms, yang tergolong sangat cepat dan efisien untuk aplikasi monitoring *real-time*. Secara keseluruhan, sistem terbukti dapat mengintegrasikan deteksi kendaraan, kontrol palang otomatis, tampilan informasi, dan pencatatan data ke *database* secara *real-time* dengan performa yang andal dan layak diterapkan pada sistem parkir cerdas.

REFERENSI

- [1] Andi Kurniawan and Rina Puspita Sari, "Pengembangan Sistem Parkir Pintar Berbasis IoT Menggunakan ESP32, Sensor Ultrasonik HC-SR04, dan OLED Display," *Journal of IoT Applications and Engineering Systems*, vol. 5, 2020.
- [2] G. R. Koten *et al.*, "Penerapan internet of things pada smart parking system untuk kebutuhan pengembangan smart city," *Jurnal Teknik Industri dan Manajemen Rekayasa*, vol. 1, no. 1, pp. 49–59, Jun. 2023, doi: 10.24002/jtimr.v1i1.7204.
- [3] C. Biyik *et al.*, "Smart parking systems: Reviewing the literature, architecture and ways forward," *Smart Cities*, vol. 4, no. 2, pp. 623–642, Jun. 2021, doi: 10.3390/smartcities4020032.
- [4] Engel Manuel Punuh, "Rancang Bangun Sensor Parkir Kendaraan Roda Empat Berbasis Mikrokontroler Arduino Uno," *ResearchGate*, 2020.
- [5] E. Sunandar *et al.*, "PROTOTYPE MONITORING AREA PARKIR MOBIL BERBASIS ARDUINO UNO UNTUK MENDETEKSI KETERSEDIAAN SLOT PARKIR SECARA OTOMATIS," 2017.
- [6] A. C. Zhang and Y. H. Lo, "Non-Invasive Blood Flow Speed Measurement Using Optics," *Sensors*, vol. 22, no. 3, Feb. 2022, doi: 10.3390/s22030897.
- [7] Rayhan Al Hayubi, Salsabila Aulia, Dafairro Abbil Gunawan, Syarif Hidayatullah, and Didik Aribowo, "Implementasi Sistem Penggerak Servo SG 90 Berbasis Arduino Uno dengan Kontrol Sudut Dinamis," *Mars : Jurnal Teknik Mesin, Industri, Elektro Dan Ilmu Komputer*, vol. 2, no. 6, pp. 130–140, Dec. 2024, doi: 10.61132/mars.v2i6.535.
- [8] D. Nataliana, I. Syamsu, and G. Giantara, "Sistem Monitoring Parkir Mobil menggunakan Sensor Infrared berbasis RASPBERRY PI," 2014.
- [9] L. Pitriyanti, Y. Saragih, U. Latifa, and F. Teknik, "IMPLEMENTASI MODUL INFRARED PADA RANCANG BANGUN SMART DETECTION FOR QUEUE OTOMATIC BERBASIS IOT," *Jurnal POLEKTRO: Jurnal Power Elektronik*, vol. 11, no. 2, 2022.
- [10] N. S. Amir and A. A. Shafie, "Time response for sensor sensed to actuator response for mobile robotic system," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Nov. 2017. doi: 10.1088/1757-899X/260/1/012030.