

PROGRAM STUDI
TEKNIK ELEKTRO
UNIVERSITAS PGRI SEMARANG



JURNAL
JELEKTRO DAN
TEKNOLOGI
INFORMASI

APRIL 2026



VOLUME 5
NO. 1
2026

Editorial Team

Editor-in-Chief

[Dr.Adhi Kusmantoro, ST, MT](#) Universitas PGRI Semarang, **Semarang**

Editorial Board

[Dr.Achmad Buchori, M.Pd](#) Universitas PGRI Semarang, **Semarang**

[Dr.Andi Kurniawan Nugroho, ST, M.Eng](#) Universitas Semarang, **Semarang**

[Imam Abdul Rozag, S.Pd, MT](#) Universitas Muria Kudus, **Kudus**

[Mega Novita, S.Si, M.Si, M.Sc, Ph.D](#) Universitas PGRI Semarang, **Semarang**

[Dr.Aryo Andri Nugroho, S.Si, M.Pd](#) Universitas PGRI Semarang, **Semarang**

Associate Editor

[Margono, ST, M.Eng](#) Universitas PGRI Semarang, **Semarang**

[Arif Wibisono, S.Pd, M.Kom](#) Universitas PGRI Semarang, **Semarang**

[Theodora Indriati Wardani, S.Kom, M.Kom](#) Universitas PGRI Semarang, **Semarang**

[Irna Farikhah, S.Pd, M.Sc, Ph.D](#) Universitas PGRI Semarang, **Semarang**

[Bambang Hadi Kunaryo, ST, MT](#) Universitas PGRI Semarang, **Semarang**

[Alfia Putri Wulandari, ST, MT](#) Universitas Diponegoro, **Semarang**

[Rita Hariningrum, S.T., M.Kom.](#) Universitas IVET, **Semarang**

Reviewer

[Dr.Roy Rudolf Huizen, ST, MT](#) STIKOM, **Bali**

[Dr.Ari Endang Jayati, ST, MT](#) Universitas Semarang, **Semarang**

[Dr.Adri Gabriel Sooi, ST, MT](#) Universitas Katolik Widya Mandira, **Kupang**

[Dr.Eng.Imam Wahyudi Farid, ST, MT](#) Institut Teknologi Sepuluh Nopember, **Surabaya**

[Dr.Dandy Pramana Hostiadi, S.Kom, MT](#) STIKOM, **Bali**

[Dr.Sigit Ristanto, ST, M.Sc](#) Universitas PGRI Semarang, **Semarang**

Layout Editor

[Afeef Kurnia Rahmawan, S.T., M.T.](#) Universitas PGRI Semarang, **Semarang**

Administrator / IT Support

[Ganjar Winasis, S.T., M.T.](#) Universitas PGRI Semarang, **Semarang**

Daftar Isi

<u>RANCANG BANGUN SISTEM KEAMANAN PARKIR OTOMATIS TERINTEGRASI MENGGUNAKAN ARDUINO MEGA</u>	1-5
Kahfi Sukma Widi, Affandi Faisal Kurniawan, Muhammad Amiruddin	
<u>Implementasi Dan Optimasi Sistem Parkir Terintegrasi Menggunakan Yolo Dan Cnn Pada Lingkungan Edge Computing Berbasis Raspberry Pi</u>	6-11
Muhammad Caesar Abhista Raya Bima Saputra, Imaduddin Harjanto, Ganjar Winasis	
<u>SISTEM DETEKSI PLAT NOMOR KENDARAAN OTOMATIS DENGAN ALGORITMA YOLOV8 DAN OCR</u>	12-16
Lisa Indriyani, Sigit Ristanto, Imadudin Harjanto	
<u>Prediksi Beban Harian Menggunakan Metode Multi Layer Perceptron</u>	17-20
hidayah dwi putra baraka, Anggraeny Sekar Kinasih, Syafiq Varel Maulana	
<u>Penerapan Jaringan Saraf Tiruan Model McCulloch-Pitts pada Sistem Pengendalian Lampu Taman Otomatis</u>	21-24
Ruli Gunawan, Adiel Kevin Kharisty S.	

RANCANG BANGUN SISTEM KEAMANAN PARKIR OTOMATIS TERINTEGRASI MENGGUNAKAN ARDUINO MEGA

Kahfi Sukma Widi^{1*}, Affandi Faisal Kurniawan², Muhammad Amiruddin³

^{1,2,3} Program Studi Teknik Elektro Fakultas Teknik dan Informatika Universitas PGRI Semarang

Jl. Sidodadi Timur No.24-Dr.Cipto, Semarang, 50232

*Email: kahfisukmawidi@gmail.com

Abstrak—Sistem parkir konvensional yang masih mengandalkan pencatatan manual oleh petugas sering menimbulkan kesalahan input, keterlambatan pencatatan, dan ketidakefisienan dalam proses masuk–keluar kendaraan. Oleh karena itu, penelitian ini bertujuan merancang dan membangun sistem keamanan parkir otomatis terintegrasi. Sistem menggunakan Arduino Mega sebagai pengendali gerbang servo, program Python sebagai pemroses data sekaligus pengirim string plat masuk, plat keluar, serta durasi parkir. Metode penelitian mencakup perancangan perangkat keras, komunikasi data serial, serta pengujian performa yang meliputi kecepatan sistem menerima data, akurasi pembacaan plat, respons pergerakan servo, serta perbandingan pencatatan otomatis dengan pencatatan manual manusia. Hasil pengujian menunjukkan bahwa total waktu untuk proses satu kendaraan adalah ± 196.477 ms, serta maksimal jarak pembacaan barcode adalah 20 cm. Dibandingkan dengan pencatatan manual, sistem ini terbukti lebih cepat, lebih konsisten, dan mengurangi potensi kesalahan manusia. Dengan demikian, sistem yang dikembangkan berhasil memenuhi tujuan penelitian dan dapat menjadi solusi otomatis yang efisien dan reliabel untuk parkir modern.

Kata kunci: Sistem Parkir, Arduino Mega, Mikrokontroler

I. PENDAHULUAN

Kota Semarang sebagai kota metropolitan mengalami peningkatan jumlah penduduk dan aktivitas ekonomi yang berdampak pada pertumbuhan kendaraan bermotor setiap tahun. Berdasarkan data Dinas Perhubungan Kota Semarang Tahun 2022, jumlah kendaraan terdaftar mencapai sekitar 1,9 juta unit, sehingga kebutuhan lahan parkir di berbagai fasilitas publik dan swasta terus meningkat. [2]

Namun, sistem parkir di banyak lokasi seperti pusat perbelanjaan, rumah sakit, dan kampus masih bersifat konvensional dengan pencatatan manual [1]. Kondisi ini menimbulkan berbagai permasalahan, antara lain kesalahan pendataan, hilangnya tiket parkir, serta rendahnya tingkat keamanan kendaraan.

Meskipun Pemerintah Kota Semarang telah menerapkan sistem e-parking di sejumlah titik, penerapannya masih terbatas dan belum sepenuhnya mendukung pengawasan parkir dalam ruangan secara otomatis dan real-time [3]. Oleh karena itu, diperlukan sistem keamanan parkir otomatis yang terintegrasi untuk meningkatkan efisiensi, keamanan, dan kenyamanan pengguna kendaraan serta mendukung konsep Smart City di Kota Semarang.

II. STUDI PUSTAKA

2.1 ARDUINO MEGA 2560

Arduino Mega 2560 merupakan papan mikrokontroler berbasis ATmega2560 yang dikembangkan untuk mendukung proyek sistem tertanam dengan kebutuhan

input dan output yang besar. Arduino Mega memiliki 54 pin digital, 16 pin analog, serta beberapa port komunikasi seperti UART, SPI, dan I2C, sehingga mampu mengintegrasikan berbagai sensor dan aktuator secara bersamaan.

Arduino Mega banyak digunakan pada sistem otomasi dan monitoring karena kemampuannya dalam mengolah data secara real-time serta kemudahan pemrograman melalui Arduino IDE. Dengan kapasitas memori yang lebih besar dibandingkan Arduino Uno, Arduino Mega sangat sesuai digunakan pada sistem yang membutuhkan pemrosesan data kompleks dan pengendalian banyak perangkat, seperti sistem parkir otomatis dan sistem keamanan terintegrasi [1], [4].

2.2 PRINTER THERMAL

Printer thermal merupakan perangkat pencetak yang bekerja menggunakan panas untuk menghasilkan teks atau gambar pada kertas khusus tanpa memerlukan tinta. Teknologi ini banyak digunakan pada sistem transaksi dan pencatatan otomatis karena memiliki kecepatan cetak yang tinggi, biaya perawatan rendah, serta ukuran perangkat yang relatif kecil.

Dalam sistem otomatisasi, printer thermal sering diintegrasikan dengan mikrokontroler melalui komunikasi serial (TTL) untuk mencetak data secara langsung, seperti nomor identitas, waktu, atau informasi transaksi. Keandalan dan respon cepat printer thermal menjadikannya cocok digunakan pada sistem parkir otomatis sebagai media pencetak tiket [6], [7].

2.3 BARCODE SCANNER GM66

Barcode Scanner GM66 merupakan perangkat pemindai kode batang berbasis kamera (image scanner)

yang mampu membaca berbagai jenis barcode satu. Perangkat ini bekerja dengan menangkap citra barcode, kemudian mengolahnya menjadi data digital berupa string yang dapat diproses oleh sistem.

GM66 umumnya menggunakan antarmuka komunikasi serial (TTL atau USB) sehingga mudah diintegrasikan dengan mikrokontroler maupun komputer. Dengan kecepatan pembacaan yang tinggi dan tingkat akurasi yang baik, barcode scanner GM66 banyak digunakan dalam sistem identifikasi otomatis, termasuk sistem parkir otomatis untuk membaca kode kendaraan secara cepat dan efisien. [5], [11]

2.4 MOTOR SERVO

Motor servo merupakan aktuator elektromekanis yang digunakan untuk mengontrol posisi sudut secara presisi. Motor ini bekerja berdasarkan sinyal Pulse Width Modulation (PWM), di mana lebar pulsa menentukan sudut putaran poros servo. Umumnya, motor servo memiliki rentang sudut antara 0° hingga 180° .

Motor servo banyak digunakan dalam sistem otomatis karena kemampuannya dalam mengatur gerakan secara akurat dan stabil. Dalam sistem parkir otomatis, motor servo dimanfaatkan sebagai penggerak palang parkir untuk membuka dan menutup akses kendaraan secara otomatis berdasarkan perintah dari mikrokontroler.

2.5 LCD 16X2

LCD 16×2 merupakan modul tampilan berbasis liquid crystal display yang mampu menampilkan 16 karakter pada 2 baris. Modul ini banyak digunakan dalam sistem berbasis mikrokontroler sebagai media penampil informasi karena konsumsi daya yang rendah dan kemudahan dalam pemrograman. LCD 16×2 dapat dihubungkan menggunakan antarmuka paralel maupun I2C untuk menghemat jumlah pin.

Dalam sistem parkir otomatis, LCD 16×2 digunakan untuk menampilkan informasi seperti status kendaraan masuk atau keluar serta pesan sistem secara real-time kepada pengguna.

2.6 EXCEL DATA STREAMER

Excel Data Streamer merupakan fitur pada Microsoft Excel yang digunakan untuk menerima, menampilkan, dan menyimpan data secara real-time dari perangkat eksternal melalui komunikasi serial. Fitur ini memudahkan proses pencatatan data tanpa memerlukan pemrograman database yang kompleks.

Dalam sistem parkir otomatis, Excel Data Streamer digunakan untuk merekam data kendaraan masuk dan keluar secara otomatis, sehingga memudahkan proses monitoring, penyimpanan, dan analisis data parkir secara terstruktur. [8], [10]

III. METODE/DESAIN

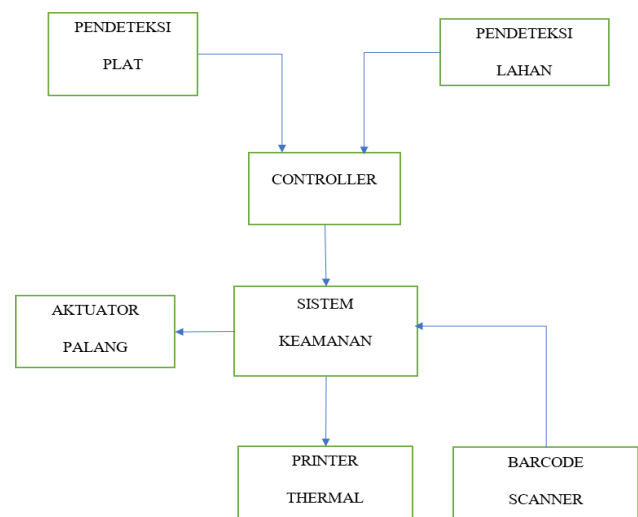
3.1 Pendekatan Penelitian

Pendekatan penelitian yang digunakan adalah pendekatan rekayasa (engineering approach) dengan

metode eksperimen. Pendekatan itu bertujuan untuk merancang, mengimplementasikan dan menguji system parkir otomatis berbasis Arduino Mega 2560 dan Program Python melalui komunikasi serial.

Tahapan penelitian meliputi analisis kebutuhan sistem, perancangan sistem, implementasi perangkat keras dan perangkat lunak, serta pengujian dan evaluasi kinerja sistem. Pengujian dilakukan untuk memastikan keakuratan pembacaan barcode, respons motor servo, serta kestabilan komunikasi data. Pendekatan ini diharapkan mampu menghasilkan sistem parkir otomatis yang bekerja secara optimal, real-time, dan meningkatkan efisiensi serta keamanan parkir kendaraan.

3.2 DIAGRAM SISTEM KESELURUHAN



Gambar 1. Diagram Sistem

Keterangan dari gambar diatas sebagai berikut: Alur Kerja Masuk:

1. Motor masuk, dideteksi platnya, kirim data plat kecontroller.
2. Pendeteksi lahan, menginformasikan ketersediaan slot.
3. Jika slot tersedia, cetak data plat dan waktu masuk ke barcode, palang terbuka.

Alur Kerja Keluar

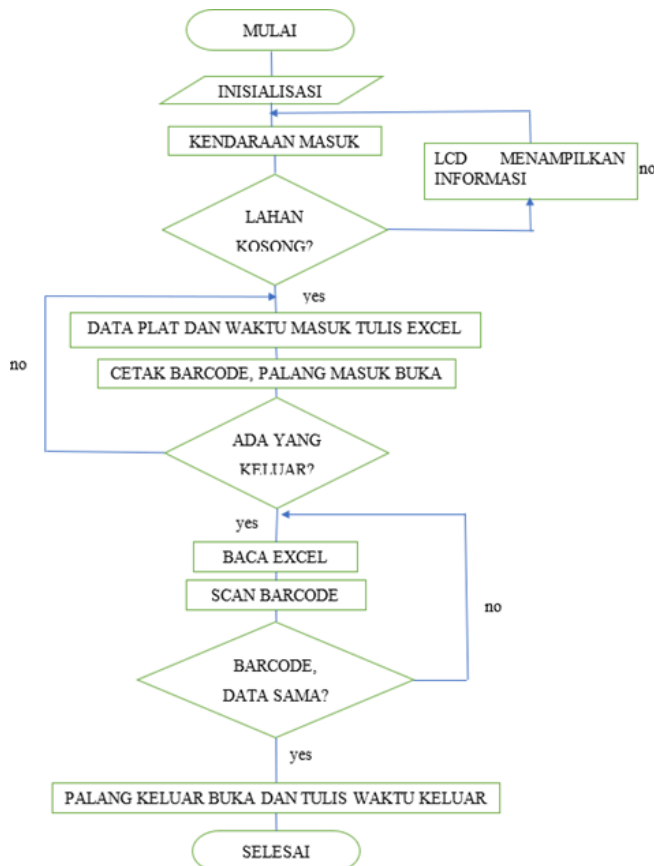
1. Motor keluar, deteksi plat, kirim data ke controller.
2. Pendeteksi lahan, mengupdate ketersediaan slot.
3. Scan barcode, jika hasil pembacaan barcode sama dengan plat nomor plat yang keluar, buka palang.

3.3 FLOWCHART SISTEM

Flowchart di atas menggambarkan alur kerja sistem parkir otomatis berbasis Arduino dan aplikasi pendukung. Proses diawali dengan inisialisasi sistem, yaitu menyiapkan seluruh perangkat keras dan perangkat lunak agar siap digunakan.

Selanjutnya, sistem mendeteksi kendaraan yang masuk ke area parkir. Sistem kemudian melakukan pengecekan ketersediaan lahan parkir. Apabila lahan parkir

tidak tersedia, kendaraan tidak dapat masuk dan sistem kembali menunggu kendaraan berikutnya. Jika lahan parkir tersedia, sistem akan mencatat data plat kendaraan dan waktu masuk ke dalam file Excel sebagai basis data.



Gambar 2. Flowchart Sistem

Setelah data berhasil disimpan, sistem akan mencetak barcode sebagai tiket parkir dan membuka palang masuk untuk mengizinkan kendaraan memasuki area parkir. Sistem kemudian memantau apakah terdapat kendaraan yang akan keluar. Jika tidak ada kendaraan keluar, sistem kembali ke proses pemantauan. Apabila terdapat kendaraan keluar, sistem akan membaca data pada file Excel, kemudian melakukan pemindaian barcode yang dibawa oleh pengguna.

Hasil pemindaian barcode akan dibandingkan dengan data yang tersimpan. Jika data barcode tidak sesuai, palang keluar tidak akan dibuka dan sistem kembali menunggu pemindaian ulang. Jika data barcode sesuai, sistem akan membuka palang keluar dan secara otomatis mencatat waktu keluar kendaraan ke dalam file Excel. Proses berakhir setelah data waktu keluar berhasil disimpan, dan sistem kembali siap untuk melayani kendaraan berikutnya.

IV. HASIL DAN PEMBAHASAN

4.1 PENGUJIAN JARAK PEMBACAAN BARCODE.

Menentukan jarak efektif pembacaan barcode oleh scanner GM66 dengan mengukur persentase keberhasilan

pembacaan pada beberapa jarak berbeda. Hasil digunakan untuk menetapkan jarak operasi optimal di lapangan.

Hasil pengujian dilakukan dengan 3 percobaan, bisa dilihat pada table berikut :

Tabel. 1 Pengujian jarak barcodeGM66

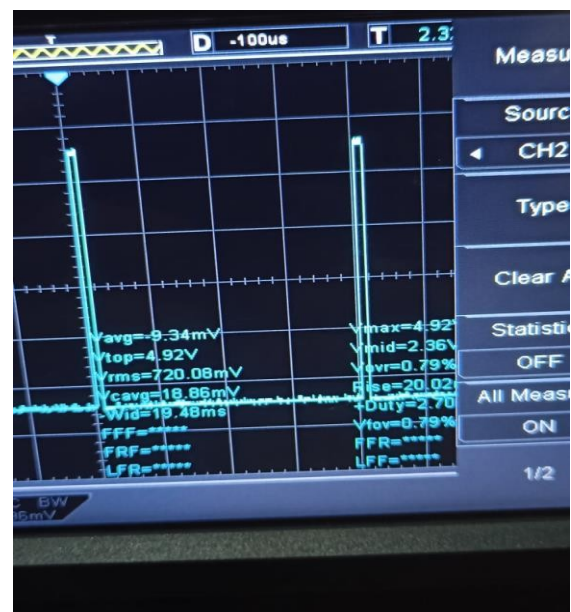
Jarak	Percobaan	Terbaca	Tidak Terbaca	Keberhasilan
5cm	3	3	0	100%
10cm	3	3	0	100%
15cm	3	3	0	100%
20cm	3	3	0	100%
25cm	3	2	1	66,66%
30cm	3	0	3	0%

Kesimpulan Analisis:

1. Jarak optimal pembacaan barcode ada pada 5–20 cm dengan tingkat keberhasilan 100%.
2. Pada jarak 25 cm, tingkat keberhasilan turun drastis menjadi 66,67%, artinya sensor mulai kehilangan akurasi karena jarak terlalu jauh atau cahaya kurang optimal.
3. Pada jarak 30 cm, scanner tidak mampu membaca sama sekali (0%).
4. Maka, jarak operasional yang ideal untuk sistemmu di lapangan adalah ≤ 20 cm agar akurasi terjamin.
5. Rentang 5–20 cm menunjukkan kinerja scanner sangat stabil, baik untuk barcode kecil maupun besar.

4.2 PENGUJIAN PALANG SERVO

1. Posisi 0 derajat



Gambar 3. Hasil uji 0 derajat

Dari gambar pertama terlihat parameter:

1. V_{max} : 4.92 V $\rightarrow$ cocok, karena sinyal PWM servo adalah 5V.
2. Periode (Wid) : 19.48 ms $\rightarrow$ sama dengan 50 Hz, standar servo.
3. Duty Cycle : 2.70%

4. Lebar pulsa (Rise → Fall) : sekitar 0.52 ms – 0.60 ms

Duty Cycle = 2.70% → dihitung:

$$\text{Duty cycle} = \frac{\text{Pulse width}}{\text{Period}} \times 100\%$$

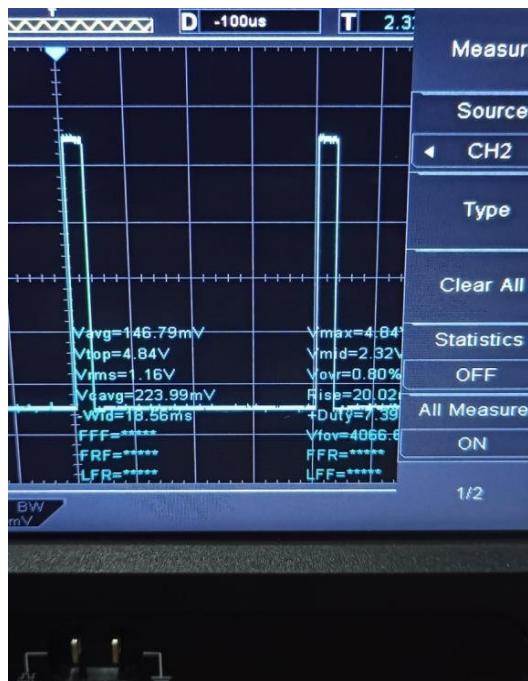
Dengan periode 19.48 ms:

$$\text{Pulse width} = 0.027 \times 19.48 \approx 0.52 \text{ ms}$$

Artinya width sinyal ≈ 0.5 ms

Lebar pulsa (pulse width) yang terbaca adalah sekitar 0,52 ms dengan duty cycle sebesar 2,7%. Nilai ini sesuai dengan karakteristik beberapa jenis motor servo yang menggunakan rentang lebar pulsa 0,5 ms hingga 2,5 ms untuk pergerakan 0°–180°. Tegangan maksimum sinyal tercatat sebesar 4,92 V, sesuai dengan level logika output Arduino.

2. Posisi 90 Derajat



Gambar 4. Hasil uji 90 derajat

Dari gambar pertama terlihat parameter :

1. Vmax : 4.84 V → cocok, karena sinyal PWM servo adalah 5V.
2. Periode (Wid) : 18.36 ms → sama dengan 50 Hz, standar servo.
3. Duty Cycle : 7.39%
4. Lebar pulsa (Rise → Fall) : sekitar 1.35 ms – 1.37 ms

Duty Cycle = 7.39% → dihitung menggunakan rumus:

$$\text{Duty cycle} = \frac{\text{Pulse width}}{\text{Period}} \times 100\%$$

Dengan periode 18.36 ms, maka:

$$\text{Pulse width} = 0.0739 \times 18.36$$

$$\text{Pulse width} \approx 1.35 \text{ ms}$$

Artinya lebar pulsa (width) sinyal PWM untuk posisi 90° adalah sekitar 1.3–1.4 ms, yang sesuai dengan standar servo (posisi tengah).

Pengujian sinyal PWM untuk posisi 90° dilakukan menggunakan osiloskop. Dari hasil pengukuran diperoleh periode sinyal sebesar 18.36 ms, duty cycle 7.39%, dan tegangan puncak sekitar 4.84 V. Dengan perhitungan digit-per-digit, lebar pulsa (pulse width) = $0.0739 \times 18.36 = 1.3568 \text{ ms}$ ($\approx 1.357 \text{ ms}$). Nilai ini konsisten dengan karakteristik sinyal servo pada posisi tengah, sehingga dapat disimpulkan bahwa sinyal PWM yang dihasilkan oleh Arduino saat perintah s.write(90) sudah sesuai dan stabil untuk menggerakkan servo ke posisi 90°.

4.3 PENGUJIAN TOTAL WAKTU PROSES SATU KENDARAAN

1. Kecepatan Pencacatan Data Plat Masuk

Tabel 2. Kecepatan pencacatan data plat masuk

Data Plat	Waktu Masuk	Kecepatan pencacatan
Z5566VWX	12.27	83,24 ms
A2323HIJ	12.28	95,76 ms
E3344STU	12.28	84,50 ms
D5678JKL	12.29	93,11 ms
H9101MNO	12.30	104,00 ms
F7788YZA	12.30	80,48 ms
B1234ABC	12.32	98,19 ms
N6060UVW	14.39	141,78 ms
W4545NOP	14.42	69,42 ms
L1122PQR	14,43	77,15 ms
Rata-rata kecepatan		92,763 ms

2. Kecepatan Pencacatan Data Plat Keluar

Tabel 3. Kecepatan pencacatan data plat keluar

Data plat	Waktu Keluar	Kecepatan
Z5566VWX	14.29	115,06 ms
A2323HIJ	14.30	119,88 ms
E3344STU	14.31	102,93 ms
D5678JKL	14.31	105,30 ms
H9101MNO	14.32	82,85 ms
F7788YZA	14.32	101,74 ms
B1234ABC	14.33	93,62 ms
N6060UVW	14.48	78,61 ms
W4545NOP	14.48	74,44 ms
L1122PQR	14.45	162,71 ms
Rata-rata kecepatan		103,714 ms

Berdasarkan perhitungan tersebut, dapat disimpulkan bahwa sistem parkir otomatis mampu memproses satu kendaraan secara keseluruhan dalam waktu rata-rata $\pm 196.477 \text{ ms}$. Hasil ini menunjukkan bahwa sistem bekerja dengan cepat, responsif, dan jauh lebih efisien dibandingkan pencacatan manual oleh manusia, sehingga layak diterapkan sebagai solusi parkir otomatis yang real-time dan andal.

V. KESIMPULAN

Setelah melakukan perancangan, pembuatan, pengujian dan analisa terhadap alat ini, maka dapat diambil beberapa kesimpulan:

1. Berdasarkan hasil pengujian, total waktu proses sistem untuk satu kendaraan, mulai dari pendeteksian plat masuk hingga pencatatan plat keluar, adalah ± 196.477 ms. Hal ini menunjukkan bahwa sistem bekerja sangat cepat dan responsif dalam memproses satu kendaraan secara keseluruhan.
2. Berdasarkan hasil pengujian, sistem mampu membaca barcode dengan baik pada jarak maksimal 20 cm, sehingga dapat disimpulkan bahwa jarak tersebut merupakan batas optimal keberhasilan pembacaan barcode pada alat yang digunakan.

REFERENSI

- [1] A. Andriansyah, "Perancangan Sistem Parkir Otomatis Menggunakan RFID Berbasis Arduino Uno," *Jurnal Teknologi dan Sistem Informasi*, vol. 7, no. 2, pp. 115–122, 2021.
- [2] Dinas Perhubungan Kota Semarang, *Laporan Kinerja Dinas Perhubungan Kota Semarang Tahun 2022*. Semarang, Indonesia: Pemerintah Kota Semarang, 2022.
- [3] Dinas Perhubungan Kota Semarang, *Dokumen Evaluasi Penerapan E-Parking Kota Semarang Tahun 2023*. Semarang, Indonesia: Pemerintah Kota Semarang, 2023.
- [4] R. Hidayat and N. Lestari, "Sistem Parkir Otomatis Berbasis Arduino Menggunakan Sensor Ultrasonik untuk Deteksi Slot Parkir," *Jurnal Elektro dan Otomasi*, vol. 5, no. 1, pp. 33–40, 2023.
- [5] S. Putri and T. Rahman, "Sistem Parkir Barcode Berbasis ESP32 Terintegrasi MySQL Online," *Jurnal Informatika dan Rekayasa Sistem*, vol. 9, no. 3, pp. 221–230, 2022.
- [6] M. Rohman, "Implementasi Printer Thermal TTL pada Sistem Kasir Otomatis," *Jurnal Teknologi Informasi*, vol. 4, no. 1, pp. 44–51, 2021.
- [7] D. Saputra, "Analisis Kebutuhan Daya pada Printer Thermal TTL Berbasis Mikrokontroler," *Jurnal Teknik Elektro Terapan*, vol. 6, no. 2, pp. 90–98, 2022.
- [8] B. Sutanto, "Pengembangan Sistem Parkir Berbasis Python dengan GUI untuk Pengelolaan Data Kendaraan," in *Prosiding Seminar Nasional Teknologi dan Rekayasa*, vol. 2, no. 1, 2023, pp. 77–85.
- [9] Universitas PGRI Semarang, *Laporan Data Kendaraan Mahasiswa dan Dosen 2023*. Semarang, Indonesia: UPGRIS, 2023.
- [10] F. Wijaya, A. Santoso, and I. Pratama, "Sistem Parkir IoT Menggunakan NodeMCU dan MQTT untuk Monitoring Slot Parkir," *Jurnal Internet of Things dan Sistem Cerdas*, vol. 2, no. 1, pp. 55–63, 2024.
- [11] A. R. Baha'uddin, N. Kholis, N. Yannuansa, and J. Widyo Leksono, "Rancang Bangun Automasi Sortir Barang Berdasarkan Barcode Menggunakan Sensor GM66 untuk Tujuan Kota," *Elconika: Jurnal Teknik Elektro*, vol. 2, no. 1, pp. 23–33, 2023, doi: 10.33752/elconika.v2i1.4654.

Implementasi Dan Optimasi Sistem Parkir Terintegrasi Menggunakan Yolo Dan Cnn Pada Lingkungan *Edge Computing* Berbasis Raspberry Pi

Muhammad Caesar Abhista Raya Bima Saputra^{1*}, Imadudin Harjanto², Ganjar Winasis³

^{1,2,3}Universitas PGRI Semarang

Jl.Sidodadi Timur No.24-Dr Cipto, Semarang, Jawa Tengah 50232, Indonesia

* Email: muhammadcaesar68@gmail.com

Abstrak— Pertumbuhan jumlah kendaraan di lingkungan universitas seperti Universitas PGRI Semarang (UPGRIS) memerlukan sistem parkir otomatis yang efisien. Implementasi *deep learning* pada perangkat *edge* seperti Raspberry Pi menghadapi kendala latensi dan keterbatasan sumber daya. Penelitian ini bertujuan mengoptimalkan model YOLOv8n untuk deteksi plat nomor dan MobileNetV2 untuk deteksi lahan parkir melalui dua skema optimasi: konversi TensorFlow Lite (*float32*) dan gabungan *structured pruning* 20% dengan *Post-Training Quantization* (INT8). Hasil penelitian menunjukkan bahwa optimasi gabungan (*Pruning*+INT8) berhasil memangkas ukuran model YOLOv8n dari 11,71 MB menjadi 3,19 MB dan MobileNetV2 dari 10,84 MB menjadi 2,74 MB. Pada Raspberry Pi 4B, model YOLOv8n INT8 mencapai kecepatan stabil 2,72 FPS dengan efisiensi RAM sebesar 27,4%. Sementara pada MobileNetV2, optimasi mempercepat waktu inferensi hingga empat kali lipat dan menekan konsumsi RAM sebesar 35,6% (dari 940 MB ke 605 MB). Meskipun terjadi kompromi akurasi sebesar 2%-8% pada kondisi cahaya redup, model tetap mempertahankan performa deteksi yang reliabel. Penelitian ini menyimpulkan bahwa skema gabungan *pruning* dan kuantisasi INT8 merupakan solusi paling efisien untuk menyeimbangkan kecepatan inferensi dan keterbatasan memori pada perangkat *edge* guna mendukung sistem parkir cerdas secara *real-time*.

Kata kunci: Raspberry Pi; *Edge Computing*; YOLOv8; MobileNetV2; *Pruning*; *Quantization*; TensorFlow Lite.

I. PENDAHULUAN

Pertumbuhan jumlah kendaraan yang pesat di lingkungan universitas, seperti di Universitas PGRI Semarang (UPGRIS), menuntut adanya sistem tata kelola parkir yang lebih efisien, terotomatisasi, dan hemat sumber daya. Sistem parkir konvensional yang mengandalkan tenaga manusia memiliki berbagai kelemahan, mulai dari waktu tunggu yang lama hingga kerentanan terhadap kesalahan pencatatan data. Integrasi teknologi pengolahan citra berbasis kecerdasan buatan (Artificial Intelligence) dan perangkat Internet of Things (IoT) menawarkan solusi sistem parkir modern yang bekerja secara *real-time*. Komponen utama dalam sistem ini mencakup algoritma YOLO untuk identifikasi plat nomor dan Convolutional Neural Network (CNN) untuk mendeteksi ketersediaan slot parkir secara otomatis.

Implementasi model *deep learning* yang kompleks pada perangkat dengan sumber daya terbatas (*edge computing*) seperti Raspberry Pi menghadapi tantangan besar terkait tuntutan komputasi yang tinggi. Model baseline berisiko mengalami latensi tinggi dan konsumsi daya yang tidak efisien, sehingga menghambat tercapainya kinerja deteksi objek secara *real-time*. Oleh karena itu, penelitian ini berfokus pada optimasi kinerja model YOLO dan CNN pada Raspberry Pi 4B melalui penerapan metode *pruning* untuk menghapus bobot yang tidak signifikan, *quantization* untuk mereduksi ukuran model, serta konversi ke format TensorFlow Lite.

Tujuan utama dari penelitian ini adalah mengaplikasikan model YOLO dan CNN pada perangkat Raspberry Pi serta

membandingkan performa antara model baseline dengan model yang telah teroptimasi. Evaluasi dilakukan dengan membandingkan parameter waktu inferensi, akurasi, dan ukuran model terhadap perangkat komputer konvensional sebagai pembanding kinerja. Penelitian ini diharapkan dapat memberikan kontribusi literatur terkait metode optimasi *deep learning* pada perangkat *edge* serta menunjukkan efektivitas sistem deteksi lahan parkir dan plat nomor dalam kondisi keterbatasan perangkat keras.

II. STUDI PUSTAKA

Penelitian mengenai sistem parkir cerdas berbasis visi komputer telah banyak dikembangkan untuk meningkatkan efisiensi manajemen kendaraan. Purwalaksana et al. [1] dan Pradhan et al. [2] merancang sistem parkir berbasis Raspberry Pi dan IoT yang terintegrasi dengan Optical Character Recognition (OCR). Meskipun sistem tersebut terbukti fungsional, keduanya belum menekankan pada aspek optimasi model *deep learning* untuk mengatasi keterbatasan komputasi pada perangkat keras. Dari sisi algoritma, Satya et al. [3] membuktikan bahwa YOLOv8 memiliki akurasi tinggi (mAP 99,3%) untuk deteksi plat nomor, sementara Suhartono et al. [4] memanfaatkan CNN untuk deteksi lahan parkir. Namun, implementasi model-model kompleks ini pada perangkat *edge* seringkali terkendala oleh latensi tinggi jika tidak dilakukan penyesuaian arsitektur.

Untuk mengatasi kendala sumber daya pada perangkat *edge*, teknik optimasi model menjadi solusi krusial. Ameen et al. [5] dan Mohandas et al. [6] menunjukkan bahwa

penerapan pruning dan kuantisasi pada Raspberry Pi mampu mereduksi ukuran model secara signifikan dan mempercepat waktu inferensi tanpa penurunan akurasi yang drastis. Tank et al.[7] juga menegaskan bahwa penggunaan arsitektur model yang ringan sangat menentukan keberhasilan diagnosis berbasis AI pada perangkat embedded. Berdasarkan kajian tersebut, penelitian ini mengintegrasikan metode optimasi pada model YOLO dan CNN untuk menjembatani celah antara kompleksitas model dengan keterbatasan perangkat keras.

Metode optimasi yang diterapkan dalam penelitian ini meliputi Structured Pruning, Kuantisasi, dan Konversi Model. (1) Pruning adalah teknik mereduksi parameter model dengan menghapus filter atau neuron yang memiliki kontribusi minim terhadap hasil prediksi. Penelitian ini menerapkan structured pruning untuk menghapus channel konvolusi secara sistematis, sehingga struktur jaringan menjadi lebih ringkas [8]. (2) Quantization (Kuantisasi) merupakan proses pemetaan nilai kontinu ke dalam nilai diskrit. Dalam konteks ini, digunakan Post-Training Quantization (PTQ) untuk mengubah representasi bobot dan aktivasi dari presisi tinggi (float32) menjadi presisi rendah (int8). Teknik ini bertujuan mereduksi ukuran model hingga empat kali lipat dan mempercepat inferensi aritmatika pada CP[9]. (3) Konversi Model dilakukan menggunakan kerangka kerja TensorFlow Lite (TFLite). Format ini dirancang khusus untuk lingkungan edge computing, memungkinkan model hasil optimasi dijalankan melalui interpreter ringan tanpa memerlukan akselerasi GPU eksternal [10].

III. METODE/DESAIN

A. Pendekatan Penelitian

Penelitian ini menerapkan pendekatan kuantitatif yang mengombinasikan metode eksperimental dan komparatif untuk mengevaluasi efektivitas optimasi model. Metode eksperimental dilakukan dengan memberikan perlakuan khusus pada model melalui teknik pemangkasan parameter dan penurunan presisi data. Pendekatan komparatif digunakan untuk membedakan performa model secara objektif pada dua platform komputasi yang kontras, yakni laptop sebagai representasi komputasi awan/lokal berperforma tinggi, dan Raspberry Pi 4B sebagai representasi perangkat edge dengan sumber daya terbatas. Parameter numerik yang diukur meliputi waktu inferensi, penggunaan memori, dan akurasi, guna menarik kesimpulan yang valid secara statistik.

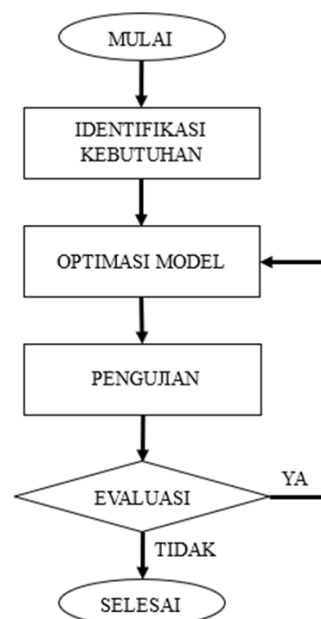
B. Alat dan Bahan

Perangkat keras dan lunak yang digunakan dalam penelitian ini meliputi:

- Dataset PKLot dan LPR, sebagai sumber data pelatihan dan optimasi model.
- Model *Baseline* MobileNetV2 dan YOLOv8n, sebagai model awal atau acuan utama model sebelum optimasi.
- Laptop HP 245 G8, sebagai perangkat utama proses optimasi dan pengembangan kode program
- Raspberry Pi 4B, sebagai perangkat utama target pengujian model baseline dan model teroptimasi
- Webcam Jete W9 dan adaptor daya USB Type-C 5V 3A, sebagai perangkat pendukung
- Visual Studio Code, Google Colab, Python, sebagai perangkat lunak pendukung proses optimasi dan pengembangan kode program

- RealVNC, sebagai perangkat pengoperasian Raspberry Pi secara remote.

C. Tahapan Optimasi Model



Gambar 1 Diagram alir tahapan penelitian

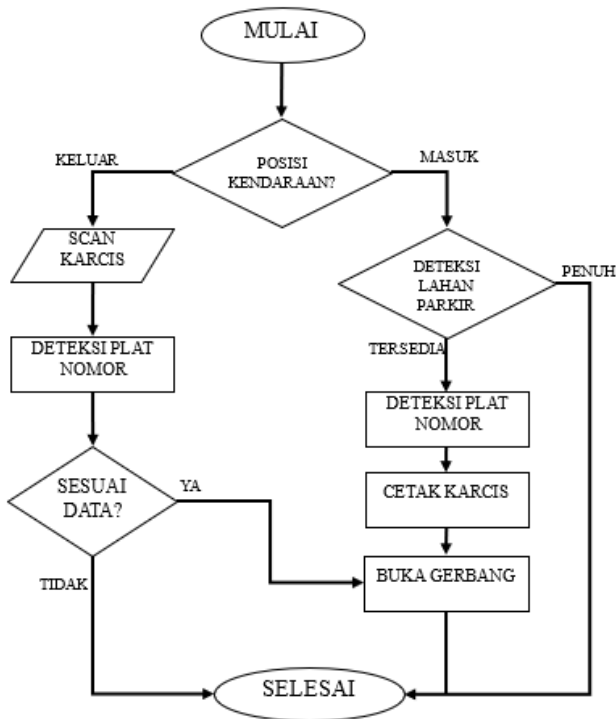
Tahapan penelitian dimulai dengan fase identifikasi kebutuhan yang mencakup pengumpulan informasi, analisis kebutuhan sistem, serta persiapan data dan model baseline. Peneliti melakukan studi literatur secara mendalam guna memperoleh landasan teoretis terkait metode deteksi objek, optimasi deep learning, serta implementasi teknis pada perangkat Raspberry Pi. Pada fase ini, dataset publik PKLot dan LPR Roboflow dipilih dan disiapkan bersama dengan data tambahan berupa foto uji lapangan manual. Model YOLOv8n ditetapkan sebagai baseline untuk tugas deteksi plat nomor kendaraan, sedangkan MobileNetV2 CNN dialokasikan untuk tugas klasifikasi ketersediaan lahan parkir.

Selanjutnya, proses optimasi model dilaksanakan menggunakan platform Google Colab untuk memanfaatkan akselerasi GPU cloud dalam mentransformasikan model baseline menjadi model yang lebih efisien. Optimasi dijalankan melalui dua skema pendekatan utama: Skema I melibatkan konversi langsung ke format TensorFlow Lite (FP32) untuk menyederhanakan struktur graf model agar kompatibel dengan perangkat edge; sedangkan Skema II menerapkan optimasi gabungan yang lebih kompleks. Skema II diawali dengan structured pruning untuk memangkas 20% parameter redundan pada struktur jaringan, yang kemudian diikuti dengan Post-Training Full Integer Quantization (INT8) untuk mengubah representasi bobot dari presisi floating point menjadi integer 8-bit. Langkah ini bertujuan untuk mereduksi ukuran model secara drastis serta menurunkan latensi inferensi pada CPU Raspberry Pi.

Setelah model teroptimasi berhasil dibuat, dilakukan tahap pengujian dan evaluasi komparatif pada perangkat laptop dan Raspberry Pi 4B. Pengujian dilakukan secara

konsisten menggunakan dataset uji yang sama untuk mengukur parameter performa yang meliputi waktu inferensi (latency), ukuran file model, tingkat akurasi deteksi, serta penggunaan sumber daya CPU dan RAM. Seluruh data hasil pengujian kemudian dievaluasi secara kritis untuk menganalisis rasio peningkatan efisiensi terhadap stabilitas akurasi. Jika performa model telah memenuhi kriteria kinerja real-time dan menunjukkan keunggulan dibandingkan model baseline, maka penelitian dinyatakan selesai dan model final direkomendasikan untuk diintegrasikan ke dalam sistem manajemen parkir kampus yang otonom.

D. Alur Kerja Sistem



Gambar 2. Flowchart system

Alur kerja sistem parkir otomatis yang dirancang dalam penelitian ini secara komprehensif diilustrasikan melalui Gambar 2. Proses operasional dimulai dengan identifikasi posisi kendaraan melalui sensor input untuk menentukan apakah kendaraan berada di pintu masuk atau pintu keluar. Pada skenario kendaraan masuk, sistem terlebih dahulu mengecek model klasifikasi lahan parkir (MobileNetV2) untuk memverifikasi ketersediaan slot. Apabila lahan parkir terdeteksi penuh, sistem secara otomatis menghentikan proses deteksi. Namun, jika lahan tersedia, sistem melanjutkan ke tahap deteksi plat nomor menggunakan algoritma YOLOv8n untuk mencatat identitas kendaraan. Data hasil deteksi tersebut kemudian diintegrasikan ke dalam karcis berbasis *barcode* yang dicetak secara otomatis bersamaan dengan pembukaan gerbang masuk selama sepuluh detik.

Sebaliknya, pada alur kendaraan keluar, sistem memulai proses dengan memindai *barcode* pada karcis untuk memanggil data kendaraan dari basis data. Secara bersamaan, kamera pada pintu keluar menangkap citra plat nomor untuk dilakukan verifikasi ulang menggunakan model deteksi objek. Sistem akan melakukan pencocokan data antara nomor plat yang terdeteksi saat keluar dengan data yang tersimpan saat masuk. Jika data dinyatakan sesuai

(*match*), gerbang keluar akan terbuka selama sepuluh detik untuk memberikan akses jalan. Namun, apabila ditemukan ketidaksesuaian data, sistem akan memberikan peringatan dan kembali ke tahap awal pemindaian untuk memastikan keamanan kendaraan.

Definisikan singkatan dan akronim pada saat pertama sekali digunakan pada teks, walaupun singkatan dan akronim tersebut telah didefinisikan di abstrak.

Singkatan seperti IEEE, SI, MKS, CGS, sc, dc, dan rms tidak perlu didefinisikan kembali. Jangan gunakan singkatan pada judul, kecuali tidak ada pilihan lain.

E. Persamaan

Nomori *persamaan* secara berurutan dengan nomor persamaan dalam tanda kurung rata dengan margin kanan, seperti pada (1). Supaya representasi persamaan lebih kompak, gunakan fungsi exp function atau eksponen yang sesuai. Untuk kuantitas dan variable, cetak miring (*italic*) symbol Roman, bukan symbol Greek. Gunakan en dash (–) dan bukan hyphen untuk tanda minus. Gunakan tanda kurung untuk memperjelas bagian penyebut pada bilangan pecahan. Pisahkan persamaan dengan koma jika persamaan tersebut merupakan bagian dari kalimat seperti contoh berikut

$$\frac{d[F_1]}{d\omega_2} = SAm_2 \cos \omega, \quad \frac{d[F_1]}{d\omega_3} = SAm_2 \cos \omega \quad (1)$$

Simbol pada persamaan seharusnya didefinisikan terlebih dahulu sebelum persamaannya muncul atau segera sesudahnya. Gunakan “(1),” bukan “Pers. (1)” atau “persamaan (1),” kecuali pada permulaan kalimat: “Persamaan (1) adalah ...”

IV. HASIL DAN PEMBAHASAN

A. Implementasi dan optimasi model

Pada penelitian ini dilakukan optimasi model deep learning yang digunakan pada sistem parkir otomatis, yaitu model YOLOv8n untuk deteksi plat nomor kendaraan dan MobileNetV2 untuk deteksi ketersediaan lahan parkir. Optimasi diperlukan untuk menyesuaikan keterbatasan sumber daya perangkat edge seperti Raspberry Pi, sehingga model yang diimplementasikan mampu mencapai keseimbangan antara efisiensi komputasi, kecepatan inferensi, dan akurasi.

Strategi optimasi dilakukan melalui dua skenario utama, yaitu konversi model ke format TensorFlow Lite (FP32) dan optimasi lanjutan menggunakan kombinasi pruning dan quantization INT8. Seluruh proses optimasi dilakukan pada lingkungan Google Colab, sedangkan Raspberry Pi digunakan secara khusus untuk tahap pengujian performa nyata guna memastikan kesiapan model dalam implementasi edge computing.

B. Hasil pengujian model MobileNetV2

Berdasarkan data pada Tabel 1 dan Tabel 2, dilakukan analisis komparatif performa berdasarkan variasi kondisi pencahayaan. Pada kondisi siang hari dengan cahaya optimal, laptop menunjukkan akurasi sempurna sebesar

100% dengan waktu inferensi tercepat pada varian TFLite FP32 (5,226 ms). Sementara itu, Raspberry Pi mampu mempertahankan akurasi stabil di angka 97%, meskipun waktu inferensinya jauh lebih lambat dengan varian INT8 sebagai yang tercepat (18,652 ms). Hal ini mengonfirmasi adanya keterbatasan *hardware* pada perangkat *edge* dalam memproses beban komputasi visi komputer.

Tabel 1. Hasil pengujian model MobileNetV2 di Laptop

Performa di Laptop						
Kondisi Cahaya	Varian model	Waktu Inferensi (ms)	Konsumsi RAM (mb)	Konsumsi CPU (%)	Size (mb)	AKURASI
SIANG	Baseline	42,370	404,050	2,631	10,84%	100%
	TFLite FP32	5,226	349,652	2,442	9,08	100%
	TFLite INT8	7,754	351,421	5,978	2,743	100%
SORE	Baseline	31,987	405,027	1,816	10,847	93%
	TFLite FP32	6,362	350,276	1,328	9,08	92%
	TFLite INT8	7,734	352,056	8,504	2,743	92%
MALAM	Baseline	31,660	404,315	1,762	10,847	94%
	TFLite FP32	5,649	348,915	1,701	9,08	94%
	TFLite INT8	8,032	350,385	6,058	2,743	92%

Tabel 2. Hasil pengujian model mobileNetV2 di Raspberry Pi

Performa di Raspberry Pi						
Kondisi Cahaya	Varian model	Waktu Inferensi (ms)	Konsumsi RAM (mb)	Konsumsi CPU (%)	Size (mb)	AKURASI
SIANG	Baseline	68,633	946,101	70,543	10,847	97%
	TFLite FP32	36,114	626,375	71,165	9,08	97%
	TFLite INT8	18,652	606,601	71,230	2,743	97%
SORE	Baseline	65,384	943,494	73,379	10,847	91%
	TFLite FP32	36,632	626,591	72,952	9,08	91%
	TFLite INT8	17,692	606,883	74,047	2,743	89%
MALAM	Baseline	67,856	891,699	71,049	10,847	94%
	TFLite FP32	36,011	634,122	70,612	9,08	95%
	TFLite INT8	18,062	605,394	72,522	2,743	92%

Memasuki kondisi sore hari dengan cahaya redup, terjadi penurunan akurasi pada kedua perangkat. Pada Raspberry Pi, penurunan paling signifikan terlihat pada varian INT8 (89%), yang menunjukkan bahwa kondisi cahaya rendah menjadi tantangan teknis terbesar bagi model yang telah terkuantisasi. Namun, pada kondisi malam hari dengan bantuan cahaya lampu, model menunjukkan pemulihan performa yang cukup baik dengan akurasi pada Raspberry Pi berkisar antara 92% hingga 95%.

Dari sisi penggunaan sumber daya, pengujian menunjukkan perbedaan beban kerja yang kontras. Penggunaan CPU pada laptop sangat rendah (1,3% - 8,5%), mengindikasikan bahwa MobileNetV2 bukan merupakan tugas yang berat bagi prosesor laptop. Sebaliknya, Raspberry Pi bekerja secara intensif dengan utilisasi CPU mencapai 74,1%. Efisiensi penggunaan RAM menjadi poin krusial pada perangkat *edge*, di mana model INT8 berhasil menekan penggunaan memori secara signifikan menjadi 605 MB dibandingkan varian *baseline* yang mencapai 940 MB.

Secara keseluruhan, model TFLite FP32 terbukti sebagai pilihan paling efektif untuk penggunaan pada laptop karena menghasilkan akurasi maksimal tanpa kendala latensi. Untuk implementasi pada Raspberry Pi, model TFLite INT8 menjadi solusi paling ideal karena mampu mempercepat waktu proses hingga empat kali lipat serta memangkas ukuran model dan penggunaan RAM secara signifikan, meskipun terdapat sedikit kompromi pada margin akurasi di kondisi cahaya tertentu.

C. Hasil pengujian model YOLOv8n

Tabel 3. Hasil pengujian model YOLOv8n di Laptop

LAPTOP						
Model	Avg Latency (ms)	Avg FPS	Avg CPU (%)	Avg RAM (MB)	Akurasi (%)	Size (mb)
Baseline .pt	354,4909063	2,820946834	60,93%	536,697	91%	5,69
Fp32 .tflite	287,4247678	3,479171289	94,93%	764,936	91%	11,712
INT8 .tflite	28088,4035	0,035601881	93,76%	183,826	83%	3,193

Tabel 4. Hasil pengujian model YOLOv8n di Raspberry Pi

LAPTOP						
Model	Avg Latency (ms)	Avg FPS	Avg CPU (%)	Avg RAM (MB)	Akurasi (%)	Size (mb)
Baseline .pt	354,4909063	2,820946834	60,93%	536,697	91%	5,69
Fp32 .tflite	287,4247678	3,479171289	94,93%	764,936	91%	11,712
INT8 .tflite	28088,4035	0,035601881	93,76%	183,826	83%	3,193

Berdasarkan data pada Tabel 3 dan 4, dilakukan analisis mendalam mengenai pengaruh optimasi terhadap kinerja sistem pada dua platform yang berbeda. Dari aspek kecepatan (*latency* dan FPS), laptop menunjukkan performa yang cenderung stabil pada model *baseline*, namun model INT8 .tflite justru mengalami penurunan kecepatan yang sangat drastis hingga 0,035 FPS. Hal ini mengindikasikan bahwa arsitektur prosesor laptop tertentu belum teroptimasi secara *native* untuk instruksi kuantisasi tersebut. Sebaliknya, pada Raspberry Pi, model INT8 menunjukkan hasil yang impresif dengan kecepatan 2,72 FPS, yang hampir identik dengan model *baseline*. Optimasi kuantisasi terbukti berhasil memulihkan kecepatan inferensi yang sempat melambat signifikan pada varian FP32 di perangkat *edge*.

Ditinjau dari efisiensi sumber daya, penggunaan CPU pada Raspberry Pi tergolong sangat tinggi, mencapai puncaknya pada model FP32 (334,53%) yang menunjukkan beban kerja prosesor yang berat. Namun, dari sisi penggunaan RAM, model INT8 berhasil memberikan efisiensi yang signifikan dengan menekan konsumsi memori kembali ke angka 795,76 MB pada Raspberry Pi, serta menjadi varian paling hemat pada laptop dengan penggunaan hanya 183,82 MB.

Terkait akurasi dan dimensi model, proses optimasi berhasil memangkas ukuran file secara drastis dari 11,71 MB menjadi 3,19 MB. Walaupun pada laptop model INT8 mengalami sedikit penurunan akurasi menjadi 83%, pada pengujian di Raspberry Pi, seluruh varian model termasuk

INT8 berhasil mempertahankan akurasi 100% pada sampel uji yang digunakan. Secara keseluruhan, model INT8 .tflite terbukti sebagai varian paling optimal untuk diimplementasikan pada Raspberry Pi karena mampu memberikan keseimbangan terbaik antara kecepatan, efisiensi memori, dan penghematan ruang penyimpanan tanpa mengorbankan akurasi deteksi.

D. Pembahasan

Berdasarkan seluruh rangkaian pengujian yang telah dilakukan, terdapat beberapa temuan krusial mengenai implementasi deep learning pada lingkungan edge computing. Pertama, hasil optimasi menunjukkan adanya trade-off yang berbeda antara arsitektur model YOLOv8n dan MobileNetV2. Pada YOLOv8n, penggunaan teknik quantization INT8 berhasil mereduksi ukuran model hingga 46% tanpa mengorbankan akurasi pada perangkat Raspberry Pi, yang secara konsisten tetap berada di angka 100%. Sebaliknya, pada MobileNetV2, meskipun reduksi ukuran model jauh lebih drastis (74,7%), terdapat margin penurunan akurasi sebesar 1% hingga 5% tergantung pada kondisi pencahayaan. Hal ini mengindikasikan bahwa model klasifikasi (CNN) cenderung lebih sensitif terhadap kehilangan presisi data dibandingkan model deteksi objek (anchor-based) saat melewati proses kuantisasi.

Kedua, fenomena perlambatan drastis model INT8 pada perangkat laptop (x86) dibandingkan akselerasi signifikan pada Raspberry Pi (ARM) menegaskan bahwa optimasi TensorFlow Lite sangat bergantung pada dukungan instruksi hardware yang spesifik. Prosesor ARM pada Raspberry Pi 4B terbukti lebih efisien dalam mengeksekusi aritmatika integer, yang memungkinkan model INT8 memulihkan kecepatan inferensi yang sempat melambat pada varian FP32. Temuan ini sangat krusial bagi pengembangan sistem embedded di masa depan, di mana pemilihan format model harus diselaraskan dengan arsitektur komputasi perangkat target, bukan sekadar mengejar ukuran file yang kecil.

Terakhir, dari aspek keberlanjutan operasional, efisiensi penggunaan RAM yang mencapai 35,6% pada MobileNetV2 memberikan ruang bagi sistem untuk menjalankan tugas tambahan seperti manajemen basis data atau komunikasi jaringan secara simultan tanpa risiko system crash akibat out-of-memory. Meskipun kondisi cahaya redup masih menjadi tantangan bagi akurasi model terkuantisasi, secara keseluruhan kombinasi strategi pruning dan quantization telah memenuhi kriteria kinerja yang diharapkan untuk sistem parkir otomatis yang responsif, hemat energi, dan memiliki throughput yang stabil pada perangkat dengan sumber daya terbatas.

V. KESIMPULAN

Penelitian ini berhasil menunjukkan bahwa optimasi model deep learning melalui teknik gabungan pruning dan kuantisasi INT8 sangat efektif untuk implementasi pada perangkat edge. Berdasarkan analisis komparatif, proses optimasi ini mampu mereduksi ukuran model secara drastis, hingga 46% pada YOLOv8n dan 75% pada MobileNetV2, yang menjadikannya solusi ideal bagi keterbatasan ruang penyimpanan. Meskipun terdapat trade-off berupa penurunan akurasi pada rentang 2%–8%—terutama pada kondisi cahaya redup—sistem tetap mempertahankan nilai recall yang tinggi sehingga fungsionalitas sistem parkir otomatis tetap terjaga. Kecepatan inferensi ditemukan sangat bergantung pada kecocokan format model dengan

arsitektur perangkat keras, di mana format INT8 memberikan performa paling stabil di Raspberry Pi (2,72 FPS) dibandingkan format FP32. Selain itu, penggunaan model teroptimasi berhasil menekan konsumsi RAM hingga 35,6%, sebuah aspek krusial untuk menjaga stabilitas sistem agar tidak mengalami crash saat menjalankan proses paralel seperti kontroler Arduino dan unit pencetakan karcis. Namun, tantangan utama masih ditemukan pada akurasi EasyOCR yang menurun hingga 70% pada kondisi real-time akibat motion blur dan penurunan detail bobot pasca-kuantisasi.

Sebagai langkah pengembangan di masa depan, penelitian dapat diarahkan pada eksplorasi arsitektur YOLO versi terbaru yang lebih ringan serta penerapan teknik image preprocessing tambahan, seperti perspective correction dan thresholding, guna meningkatkan akurasi pengenalan karakter pada engine OCR. Metode optimasi lain seperti Knowledge Distillation dan Weight Clustering juga berpotensi untuk dieksplorasi lebih lanjut guna menyeimbangkan presisi dan kecepatan. Terakhir, pengembangan sistem dapat ditingkatkan menuju integrasi penuh pada prototipe sistem parkir yang memiliki mekanisme keamanan berlapis berbasis mikrokontroler untuk menciptakan ekosistem parkir otomatis yang lebih komprehensif dan tangguh.

REFERENSI

- [1] A. Z. Purwalaksana, D. Siburian, I. Sianturi, and S. Sianturi, "Camera-Based Parking System Management Using Raspberry Pi," *Piston: Journal of Technical Engineering*, vol. 5, no. 1, pp. 22–34, 2021, doi: 10.32493/pjte.v5i1.14721.
- [2] G. Pradhan, M. R. Prusty, V. S. Negi, and S. Chinara, "Advanced IoT-integrated parking systems with automated license plate recognition and payment management," *Sci. Rep.*, vol. 15, no. 1, p. 2388, 2025, doi: 10.1038/s41598-025-86441-w.
- [3] B. Satya, D. Manongga, Hendry, and A. Aminuddin, "Optimized YOLOv8 for Automatic License Plate Recognition on Resource Constrained Devices," *Engineering, Technology and Applied Science Research*, vol. 15, no. 2, pp. 21976–21981, Apr. 2025, doi: 10.48084/etasr.9983.
- [4] Suhartono, Satria Gunawan Zain, and A. F. Sadaruddin, "Implementation of Convolutional Neural Network Algorithm for Detecting Empty Parking Area Based on Raspberry Pi," *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, vol. 8, no. 1, pp. 154–167, Jun. 2024, doi: 10.37339/e-komtek.v8i1.1220.
- [5] S. Ameen, K. Siriwardana, and T. Theodoridis, "Optimizing Deep Learning Models For Raspberry Pi," Apr. 2023. Accessed: Sep. 19, 2025. [Online]. Available: <http://dx.doi.org/10.48550/arXiv.2304.13039>
- [6] R. Mohandas, M. Bhattacharya, M. Penica, K. Van Camp, and M. J. Hayes, "TensorFlow Enabled Deep Learning Model Optimization for enhanced Realtime Person Detection using Raspberry Pi operating at the Edge," 2020.
- [7] M. B. Tank, M. Patel, and K. Deshmukh, "Evaluating Deep Learning Model Performance on Raspberry Pi 4

- for COVID-19 Diagnosis,” 2024. [Online]. Available: <https://www.jisem-journal.com/>
- [8] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, “Pruning Filters for Efficient ConvNets,” Mar. 2017, [Online]. Available: <http://arxiv.org/abs/1608.08710>
- [9] B. Jacob *et al.*, “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” Dec. 2017, [Online]. Available: <http://arxiv.org/abs/1712.05877>
- [10] Google, “Convert Models to TensorFlow Lite,” Google AI Documentation. Accessed: Nov. 10, 2025. [Online]. Available: <https://ai.google.dev/edge/litert/models/convert>

SISTEM DETEKSI PLAT NOMOR KENDARAAN OTOMATIS DENGAN ALGORITMA YOLOV8 DAN OCR

Lisa Indriyani^{1*}, Sigit Ristanto², Imadudin Harjanto³

¹²³Universitas PGRI Semarang

Jl.Sidodadi Timur No.24-Dr Cipto, Semarang, Jawa Tengah 50232, Indonesia

*Email: lisaindriyani1309@gmail.com

Abstrak— Peningkatan jumlah kendaraan di lingkungan kampus menyebabkan permasalahan dalam pengelolaan parkir, terutama pada sistem manual yang kurang efisien dan tidak terintegrasi. Penelitian ini bertujuan untuk mengembangkan sistem smart parking berbasis Artificial Intelligence dengan memanfaatkan algoritma YOLOv8 untuk deteksi plat nomor kendaraan dan EasyOCR untuk pengenalan karakter. Dataset yang digunakan terdiri dari 80 citra yang dibagi menjadi data latih, validasi, dan pengujian. Model YOLOv8 dilatih selama 50 epoch dengan ukuran citra 640 piksel dan batch size 16, menghasilkan nilai presisi 0,973, recall 0,952, dan mAP@0.5 sebesar 0,992. Pada tahap OCR, akurasi pengenalan karakter mencapai 96,36% tanpa preprocessing dan meningkat menjadi 98,3% setelah dilakukan preprocessing berupa cropping, grayscale, CLAHE, thresholding, dan operasi morfologi. Pengujian sistem secara real-time menggunakan webcam menunjukkan bahwa sistem mampu mencatat data kendaraan secara otomatis dengan akurasi tinggi dan konsistensi yang baik. Hasil penelitian ini menunjukkan bahwa integrasi YOLOv8 dan EasyOCR efektif dalam membangun sistem parkir cerdas yang cepat, akurat, dan efisien.

Kata kunci: Smart Parking; YOLOv8; EasyOCR; OCR; Artificial Intelligence

I. PENDAHULUAN

Peningkatan jumlah kendaraan di lingkungan kampus menimbulkan berbagai permasalahan dalam sistem parkir, seperti antrian panjang, kesulitan menemukan lahan parkir, serta rendahnya tingkat keamanan kendaraan. Sistem parkir konvensional yang masih menggunakan metode manual belum mampu memberikan pencatatan data yang akurat dan real-time.

Untuk mengatasi permasalahan tersebut, diperlukan sistem parkir cerdas yang mampu bekerja secara otomatis dan terintegrasi. Salah satu solusi yang dapat diterapkan adalah pemanfaatan teknologi Artificial Intelligence, khususnya pada bidang computer vision.

Penelitian ini mengusulkan sistem deteksi plat nomor kendaraan otomatis menggunakan algoritma YOLOv8 untuk mendeteksi objek plat nomor dan EasyOCR untuk mengenali karakter. YOLOv8 dipilih karena memiliki akurasi tinggi dan kemampuan deteksi real-time, sedangkan EasyOCR dipilih karena kemampuannya dalam mengenali karakter pada berbagai kondisi citra.

Tujuan dari penelitian ini adalah untuk mengimplementasikan YOLOv8 untuk deteksi plat nomor kendaraan, meningkatkan akurasi OCR melalui preprocessing citra, serta untuk mengembangkan sistem smart parking yang efisien dan akurat.

II. STUDI PUSTAKA

Penerapan Optical Character Recognition (OCR) untuk pengenalan plat nomor kendaraan telah diteliti sejak beberapa tahun terakhir. Aher dan Kapale [10] mengembangkan sistem pengenalan plat nomor kendaraan berbasis OCR sebagai salah satu fondasi awal penerapan teknologi ini untuk keperluan identifikasi kendaraan secara otomatis.

Penelitian oleh Ditta et al. [1] mengembangkan sistem smart parking berbasis Internet of Things (IoT) yang memanfaatkan Optical Character Recognition (OCR) untuk mengenali plat nomor kendaraan secara otomatis. Sistem ini menggunakan ESP32 yang terintegrasi dengan sensor dan database berbasis web untuk memantau kondisi parkir secara real-time. Hasil penelitian menunjukkan bahwa integrasi antara perangkat keras dan pengolahan citra mampu meningkatkan efisiensi sistem parkir.

Selanjutnya, penelitian oleh Aguswandi et al. [2] menggunakan algoritma YOLOv11 untuk mendeteksi slot parkir kosong. Model yang digunakan menunjukkan performa sangat tinggi dengan nilai precision, recall, dan mAP mencapai 99% serta mampu berjalan secara real-time. Hal ini menunjukkan bahwa algoritma YOLO sangat efektif dalam aplikasi deteksi objek pada sistem parkir.

Penelitian oleh Meirza dan Puteri [3] mengimplementasikan YOLOv5 untuk deteksi plat nomor dan Tesseract OCR untuk pengenalan karakter. Hasil pengujian menunjukkan tingkat akurasi sebesar 70%, yang mengindikasikan bahwa kombinasi metode tersebut masih memiliki keterbatasan terutama dalam pengenalan karakter pada kondisi citra yang kompleks.

Anthony [4] mengembangkan sistem pengenalan plat nomor menggunakan YOLOv8 dan EasyOCR berbasis web. Hasil penelitian menunjukkan akurasi deteksi sebesar 99% dan akurasi pengenalan karakter sebesar 81%. Penelitian ini menunjukkan bahwa kombinasi YOLOv8 dan EasyOCR cukup efektif, namun masih dapat ditingkatkan melalui optimasi preprocessing citra.

Penelitian oleh Nugroho et al. [5] menggunakan YOLOv5 untuk deteksi plat nomor kendaraan secara real-time dengan akurasi mencapai 84% dan presisi 92,9%. Sistem ini menunjukkan bahwa YOLO memiliki performa

yang baik dalam mendeteksi objek, namun masih memerlukan peningkatan pada tahap pengenalan karakter.

Angelica et al. [6] mengembangkan sistem palang parkir otomatis berbasis YOLOv4 dan OCR yang terintegrasi dengan IoT. Sistem ini mampu mencapai akurasi deteksi hingga 98,75%, menunjukkan bahwa pendekatan berbasis deep learning sangat efektif dalam sistem parkir otomatis. Sejalan dengan hal tersebut, Rizkatama et al. [9] mengembangkan sistem deteksi kendaraan untuk memantau ketersediaan lahan parkir menggunakan YOLOv4, yang menunjukkan bahwa varian YOLO generasi sebelumnya juga cukup andal untuk aplikasi pemantauan slot parkir secara real-time.

Khotimah et al. [7] menggunakan YOLOv3 untuk deteksi plat nomor kendaraan dengan hasil precision sebesar 100% dan recall sebesar 82%. Penelitian ini menunjukkan bahwa YOLO versi sebelumnya juga memiliki performa yang baik, meskipun masih terdapat keterbatasan dalam kondisi tertentu seperti jarak dan pencahayaan.

Selain itu, penelitian oleh Awan Aprilino [8] mengkombinasikan YOLOv3 dan Tesseract OCR untuk sistem pengenalan plat nomor berbasis web dengan akurasi OCR sebesar 92,32%. Hasil ini menunjukkan bahwa OCR konvensional masih memiliki keterbatasan dalam kondisi citra yang kompleks.

Berdasarkan berbagai penelitian terdahulu, dapat disimpulkan bahwa algoritma YOLO memiliki keunggulan dalam mendeteksi objek secara real-time dengan akurasi tinggi, sedangkan metode OCR digunakan untuk mengenali karakter pada plat nomor. Namun, sebagian besar penelitian masih memiliki keterbatasan pada tahap pengenalan karakter dan pengujian di kondisi nyata.

Oleh karena itu, penelitian ini mengusulkan sistem deteksi plat nomor kendaraan otomatis menggunakan YOLOv8 dan EasyOCR yang dilengkapi dengan tahapan preprocessing citra untuk meningkatkan akurasi pengenalan karakter. Sistem ini diharapkan mampu memberikan performa yang lebih baik serta dapat diimplementasikan secara langsung pada sistem smart parking di lingkungan nyata.

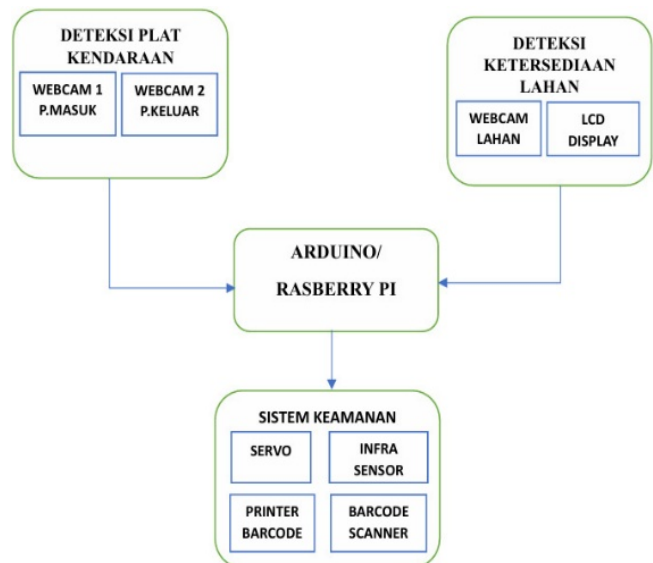
III. METODE/DESAIN

A. Desain Sistem

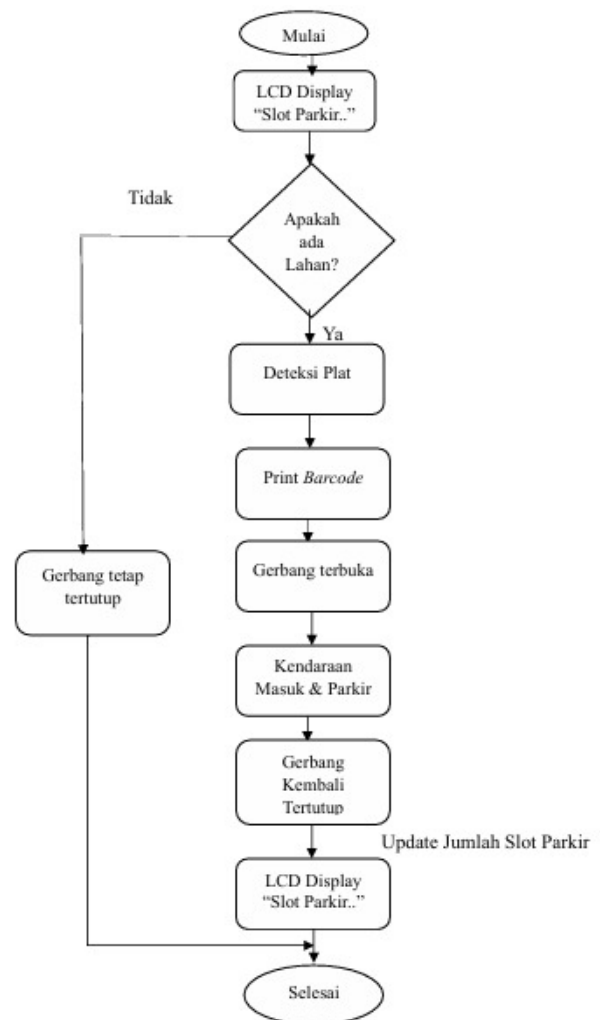
Sistem terintegrasi yang direncanakan mencakup tiga komponen utama, yaitu deteksi plat nomor, deteksi ketersediaan lahan, dan sistem keamanan.

Adapun komponen yang digunakan dalam system terintegrasi dijelaskan pada Gambar 1. Alur sistem dimulai dari pengecekan ketersediaan lahan parkir. Jika tersedia, sistem mendeteksi plat nomor serta mencatat waktu masuk kemudian palang masuk terbuka otomatis. Pengguna akan menerima barcode digital sebagai verifikasi keluar. Saat kendaraan keluar, barcode discan dan diverifikasi dengan data plat nomor, lalu sistem mencatat waktu keluar. Semua data disimpan dalam format Excel untuk keperluan monitoring dan keamanan. Sistem juga memperbarui informasi ketersediaan lahan parkir dan menampilkannya secara real-time melalui LCD display yang terpasang di pintu masuk, sehingga pengguna dapat mengetahui jumlah

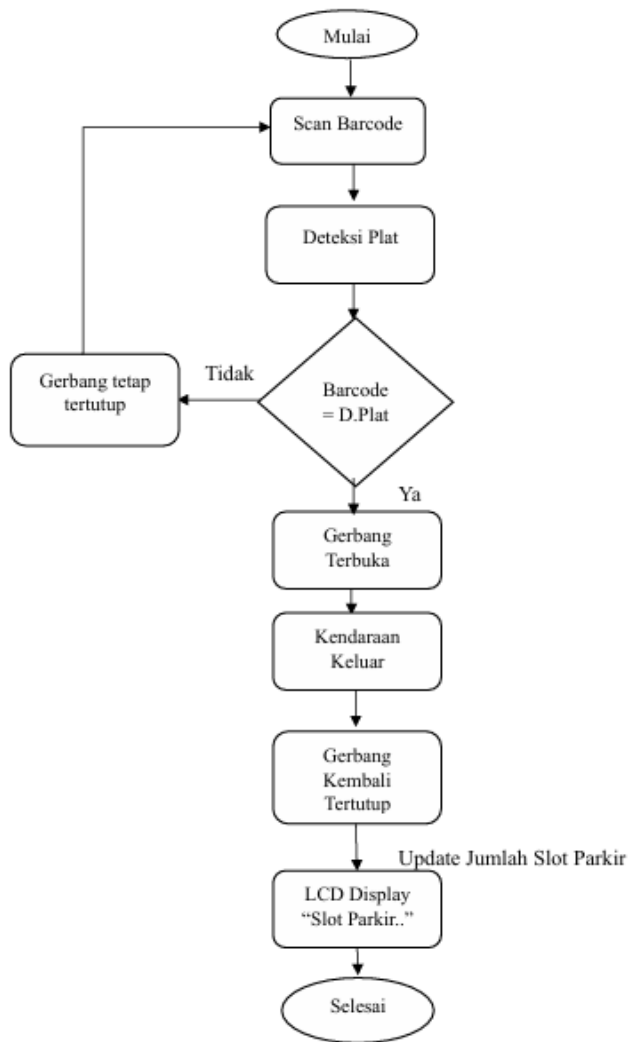
slot kosong sebelum memasuki area parkir. Alur kerja system secara keseluruhan ditunjukkan pada Gambar 2 dan Gambar 3.



Gambar 1. Komponen sistem terintegrasi



Gambar 2. Alur sistem skema keluar



Gambar 3. Alur sistem skema keluar

B. Data dan Peralatan

Dataset yang digunakan dalam penelitian ini terdiri dari 80 citra plat nomor kendaraan yang dibagi menjadi, 50 data latih, 20 data validasi, 11 data uji.

Perangkat keras utama yang digunakan terdiri dari 2 Webcam untuk menangkap citra plat nomor kendaraan, yang terpasang pada gerbang masuk dan gerbang keluar. Serta Laptop/PC sebagai pusat pemrosesan data dan pelatihan model.



Gambar 4. Perangkat keras utama

Spesifikasi laptop: Intel(R) Core (TM) i3-1005G1 CPU @ 1.20GHz, RAM 4 GB, sistem operasi 64-bit, prosesor x64. Spesifikasi webcam JETE-W9: resolusi FHD 1920×1080 piksel, sensor 1/6.9" HD Color CMOS, fixed focus 1080P/30FPS atau 720P/30FPS, sudut pandang 75°.

Sedangkan perangkat lunak yang digunakan mencakup Roboflow, VS Code, Google Colab untuk pelatihan YOLOv8 dan OCR. Serta aplikasi Spreadsheet/Excel untuk pengolahan file Excel.

C. Pelatihan Model YOLOv8

Tahap pelatihan model YOLO (You Only Look Once) dilakukan untuk membangun sistem deteksi plat nomor secara otomatis yang cepat dan akurat. Proses pelatihan dilakukan dengan parameter dengan rincian, Epoch (50), Image size (640 piksel), Batch size (16). Model dilatih menggunakan dataset yang telah diberi label (bounding box) sehingga mampu mengenali lokasi plat nomor pada citra kendaraan secara akurat. Model yang telah dilatih kemudian dapat digunakan untuk mendeteksi posisi plat nomor pada citra atau video, yang selanjutnya akan diproses pada tahap OCR untuk mengenali karakter di dalamnya.

D. Preprocessing Citra

Pada tahap preprocessing dataset untuk YOLO, data plat nomor yang diperoleh dari platform Roboflow telah melalui proses augmentasi seperti zoom in/out, rotasi, flip horizontal/vertical, random cropping, dan perspective transform. Augmentasi ini bertujuan untuk memperkaya variasi data latih sehingga model YOLOv8 dapat beradaptasi dengan berbagai sudut pandang, jarak pengambilan gambar, orientasi objek, serta kondisi lingkungan yang berbeda. Dengan demikian, model menjadi lebih robust dan mampu mendeteksi plat nomor secara konsisten di berbagai situasi nyata.

Sebelum dilakukan proses OCR, citra hasil deteksi plat nomor diproses melalui beberapa tahapan preprocessing untuk meningkatkan kualitas citra, dengan langkah langkah sebagai berikut, cropping untuk mengambil area plat nomor, konversi grayscale untuk menyederhanakan warna, CLAHE untuk meningkatkan kontras citra, gaussian blur untuk mengurangi noise, thresholding untuk menghasilkan citra biner, operasi morfologi (opening dan closing) untuk memperjelas bentuk karakter. Beberapa tahapan ini bertujuan untuk meningkatkan akurasi pengenalan karakter pada proses OCR.

E. Pengenalan Karakter (OCR)

Tahap Optical Character Recognition (OCR) merupakan proses penting dalam sistem smart parking yang bertujuan untuk mengenali karakter huruf dan angka pada plat nomor kendaraan dari gambar hasil tangkapan kamera. Dalam penelitian ini, OCR dilakukan menggunakan pustaka EasyOCR.

Pengenalan karakter dilakukan menggunakan EasyOCR yang mampu membaca teks dari citra secara akurat. EasyOCR dipilih karena memiliki kemampuan dalam mengenali karakter pada berbagai kondisi pencahayaan dan tingkat noise yang berbeda. Hasil dari proses OCR berupa teks plat nomor kendaraan yang kemudian digunakan sebagai data identifikasi kendaraan.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pelatihan Model YOLOv8

Model YOLOv8 yang digunakan untuk mendeteksi area plat nomor kendaraan dilatih menggunakan dataset sebanyak 80 citra yang terdiri dari data latih, validasi, dan pengujian. Proses pelatihan dilakukan selama 50 epoch dengan ukuran citra 640 piksel dan batch size 16. Hasil evaluasi model menunjukkan performa yang sangat baik dengan nilai precision sebesar 0,973, recall sebesar 0,952, dan mAP@0.5 sebesar 0,992.

Nilai tersebut menunjukkan bahwa model mampu mendeteksi objek plat nomor dengan tingkat akurasi yang tinggi dan kesalahan deteksi yang rendah. Tingginya nilai mAP@0.5 mengindikasikan bahwa bounding box yang dihasilkan model memiliki kesesuaian yang sangat baik terhadap objek sebenarnya.

B. Hasil Deteksi Plat Nomor

Pengujian dilakukan pada 40 sampel citra plat nomor kendaraan. Hasil deteksi menunjukkan nilai confidence rata-rata sebesar 0,799, yang menandakan bahwa model memiliki tingkat kepercayaan yang cukup tinggi dalam mendeteksi area plat nomor. Model YOLOv8 mampu mendeteksi plat nomor dalam berbagai kondisi, termasuk variasi jarak dan pencahayaan. Namun, pada kondisi tertentu seperti pencahayaan rendah atau posisi plat yang miring, tingkat confidence cenderung menurun.

C. Hasil Pengenalan Karakter (OCR)

Pengujian OCR dilakukan dalam dua skenario, yaitu tanpa preprocessing dan dengan preprocessing citra. Skenario pertama tanpa preprocessing, EasyOCR menghasilkan akurasi sebesar 96,36%. Skenario kedua dengan preprocessing, setelah dilakukan preprocessing berupa cropping, grayscale, CLAHE, thresholding, dan operasi morfologi, akurasi meningkat menjadi 98,3%. Hasil ini menunjukkan bahwa preprocessing citra memberikan peningkatan signifikan terhadap akurasi pengenalan karakter. Tahapan preprocessing mampu memperjelas bentuk karakter serta mengurangi noise yang dapat mengganggu proses OCR.

D. Pengujian Sistem Secara Keseluruhan

Pengujian sistem dilakukan secara real-time menggunakan dua webcam yang dipasang pada gerbang masuk dan keluar. Sistem diuji menggunakan 20 data plat nomor kendaraan. Hasil pengujian menunjukkan bahwa sistem mampu untuk mendeteksi plat nomor kendaraan secara otomatis, mampu untuk membaca karakter dengan akurasi tinggi, mampu untuk mencatat data kendaraan (nomor plat dan waktu), mampu untuk menyimpan data ke dalam file Excel. Sistem juga menunjukkan konsistensi dalam pencatatan data tanpa adanya kehilangan informasi selama proses pengujian.

E. Pembahasan

Berdasarkan hasil pengujian, kombinasi algoritma YOLOv8 dan EasyOCR terbukti efektif dalam membangun sistem deteksi plat nomor kendaraan otomatis. YOLOv8 menunjukkan keunggulan dalam mendeteksi objek secara real-time dengan akurasi tinggi, sedangkan EasyOCR mampu mengenali karakter dengan baik, terutama setelah dilakukan preprocessing citra.

Dibandingkan dengan penelitian sebelumnya yang menggunakan YOLOv5 atau YOLOv3, penggunaan YOLOv8 pada penelitian ini memberikan peningkatan performa terutama pada nilai akurasi dan stabilitas deteksi. Selain itu, penggunaan EasyOCR juga menunjukkan hasil yang lebih baik dibandingkan metode OCR konvensional.

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, sistem deteksi plat nomor kendaraan otomatis berbasis algoritma YOLOv8 dan EasyOCR berhasil dikembangkan dan diimplementasikan dengan baik dalam aplikasi smart parking. Model YOLOv8 mampu mendeteksi area plat nomor kendaraan secara akurat dengan nilai precision sebesar 0,973, recall 0,952, dan mAP@0.5 sebesar 0,992, sehingga efektif digunakan untuk deteksi objek secara real-time.

Pada tahap pengenalan karakter, penggunaan EasyOCR menunjukkan performa yang sangat baik, dengan tingkat akurasi sebesar 96,36% tanpa preprocessing dan meningkat menjadi 98,3% setelah diterapkan tahapan preprocessing citra. Hal ini membuktikan bahwa preprocessing memiliki peran penting dalam meningkatkan kualitas citra dan akurasi OCR.

Pengujian sistem secara keseluruhan menunjukkan bahwa sistem mampu bekerja secara otomatis dalam mendeteksi, membaca, dan mencatat data kendaraan menggunakan webcam pada gerbang masuk dan keluar, serta menyimpan data ke dalam format digital dengan konsistensi yang baik. Dengan demikian, sistem yang dikembangkan dapat menjadi solusi yang efektif untuk meningkatkan efisiensi, akurasi, dan keamanan dalam pengelolaan parkir di lingkungan kampus.

UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih kepada pihak Universitas PGRI Semarang, khususnya Program Studi Teknik Elektro, yang telah memberikan dukungan dan fasilitas dalam pelaksanaan penelitian ini. Ucapan terima kasih juga disampaikan kepada dosen pembimbing yang telah memberikan arahan, masukan, dan bimbingan selama proses penelitian hingga penyusunan naskah. Selain itu, saya juga mengapresiasi semua pihak yang telah membantu dalam proses pengambilan data dan pengujian sistem sehingga penelitian ini dapat diselesaikan dengan baik.

REFERENSI

- [1] A. Ditta et al., "Smart Parking System Berbasis IoT dengan Optical Character Recognition untuk Identifikasi Kendaraan," *Jurnal Teknologi Informasi*, 2025.
- [2] R. Aguswandi et al., "Deteksi Slot Parkir Kosong Menggunakan Algoritma YOLOv11 Berbasis Deep Learning," *Jurnal Informatika dan Sistem Cerdas*, 2025.
- [3] M. Meirza and D. Puteri, "Implementasi YOLOv5 dan Tesseract OCR untuk Pengenalan Plat Nomor Kendaraan," *Jurnal Teknologi Komputer*, 2024.
- [4] A. Anthony, "Sistem Pengenalan Plat Nomor Kendaraan Menggunakan YOLOv8 dan EasyOCR Berbasis Web," *Jurnal Artificial Intelligence*, 2024.
- [5] B. P. Nugroho, Y. Prihati, and S. T. Galih, "Implementasi Algoritma YOLOv5 dalam Rancangan Aplikasi Pendeteksi Plat Nomor Kendaraan," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 7, no. 3, pp. 851–859, Jun. 2024, doi: 10.31539/intecom.v7i3.10376.
- [6] S. Angelica et al., "Sistem Palang Parkir Otomatis Berbasis AI dan IoT Menggunakan YOLOv4 dan OCR," *Jurnal Sistem Cerdas*, 2023.
- [7] Khotimah, "Pendeteksi Plat Nomor Kendaraan Bermotor Berbasis Algoritma YOLO (You Only Look Once) Menggunakan Kamera CCTV," *SCIENCE ELECTRO*, vol. 1, no. 1, 2023.
- [8] A. Aprilino, "Sistem Deteksi Plat Nomor Kendaraan Menggunakan YOLOv3 dan Tesseract OCR Berbasis Web," *Jurnal Informatika*, 2022.
- [9] N. Rizkatama et al., "Sistem Deteksi Kendaraan untuk Monitoring Ketersediaan Parkir Menggunakan YOLOv4," *Jurnal Sistem Informasi*, 2021.
- [10] S. R. Aher and N. D. Kapale, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition," *IRJET*, vol. 4, no. 6, Jun. 2017.

PREDIKSI BEBAN LISTRIK HARIAN MENGUNAKAN METODE MULTILAYER PERCEPTRON (MLP)

Hidayah Dwi Putra Baraka^{1*}, Anggraeny Sekar Kinasih², Syafiq Varel Maulana³

¹²³Program Studi Teknik Elektro Fakultas Teknik dan Informatika Universitas PGRI Semarang

*Email : Hidayahdwiputrabaraka10@gmail.com

Abstrak— Prediksi beban listrik merupakan salah satu langkah penting dalam mendukung pengelolaan sistem tenaga listrik yang andal dan efisien. Informasi mengenai kondisi prediksi listrik pada periode berikutnya dapat digunakan untuk membantu proses pemantauan, evaluasi kinerja sistem, serta pengambilan putusan dalam pengoprasian peralatan listrik. Penelitian ini bertujuan untuk memprediksi kelistrikan menggunakan metode Multi Layer Perceptron (MLP), berdasarkan data pengukuran kelistrikan Tingkat tinggi dari sebuah rumah indekos yang terdiri atas Real Power (Daya aktif), Apparent Power (Daya semu), dan Power Factor (Faktor daya). Model MLP kemudian dilatih menggunakan Algoritma Backpropagation agar mampu mempelajari hubungan nonlinear antarparameter kelistrikan. Selanjutnya performa model dievaluasi menggunakan Mean Absolute Percentage Error (MAPE) untuk mengukur Tingkat akurasi hasil prediksi. Hasil penelitian diharapkan menunjukkan bahwa metode MLP mampu menghasilkan prediksi parameter kelistrikan dengan Tingkat kesalahan rendah sehingga digunakan sebagai salah satu alternatif dalam mendukung proses pemantauan dan pengelolaan sistem tenaga listrik secara lebih efektif.

Kata kunci: Prediksi Beban Listrik; Multi Layer Perceptron; Artificial Neural Network; Peramalan Beban; Sistem Tenaga Listrik

I. PENDAHULUAN

Perkembangan sistem tenaga Listrik menuntut pengelolaan energi yang lebih efisien dan andal. Salah satunya Upaya yang dapat dilakukan Adalah memprediksi parameter kelistrikan untuk membantu proses pemantauan, pengendali beban dan pengambilan Keputusan operasional. Prediksi akurat dapat meningkatkan efisiensi penggunaan energi serta menjaga kestabilan sistem tenaga listrik [2].

Parameter yang umum digunakan dalam analisis sistem tenaga meliputi preal *Real Power*, *Apparent Power*, dan *Power Factor*. Ketiga parameter tersebut sa;ing berhubungan memlalui persamaan $PF = P/S$, sehingga perubahan daya aktif dan daya semu akan memengaruhi nilai factor daya. Dalam penelitian ini, *Power Factor* dihitung berdasarkan data aktual *Real Power* dan *Apparen Power*.

Metode Multi Layer Perceptron (MLP) dipilih karena mampu memodelkan hubungan nonlinear antarparameter dan menghasilkan prediksi yang baik pada berbagai penelitian beban listrik [2], [3]. Oleh karena itu, penelitian ini menerapkan MLP untuk memprediksi *Real Power* menggunakan data aktual yang diproses. Akurasi model evaluasi menggunakan *Mean Absolute Percentage Error* (MAPE).

II. STUDI PUSTAKA

A. Real Power, Apparent Power, dan Power Factor

Pada sistem tenaga listrik AC, *Real Power* (P) merupakan daya yang digunakan untuk melakukan kerja, sedangkan *Apparent Power* (S) adalah total daya yang disalurkan ke beban. Hubungan keduanya menghasilkan *Power Factor* (PF) yang menunjukkan efisiensi

penggunaan daya listrik dan dihitung menggunakan persamaan:

$$PS = \frac{F}{S}$$

Semakin tinggi nilai *Power Factor*, semakin efisien pemanfaatan energi listrik.

B. Multi Layer Perceptron (MLP)

Multi Layer Perceptron (MLP) adalah salah satu metode *Artificial Neural Network* yang terdiri atas lapisan input, hidden layer, dan output layer. MLP mampu mempelajari hubungan nonlinier antarvariabel melalui proses pelatihan menggunakan *algoritma Backpropagation*, sehingga banyak digunakan pada penelitian prediksi beban Listrik [2], [3], [6].

C. Normalisasi Data

Normalisasi dilakukan untuk menyamakan rentang nilai setiap parameter agar proses pelatihan jaringan lebih stabil. Penelitian ini menggunakan metode *Min-Max Normalization* dengan persamaan:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

D. Evaluasi Model

Kinerja model dievaluasi menggunakan *Mean Absolute Percentage Error* (MAPE) karena mampu menunjukkan tingkat kesalahan prediksi dalam bentuk persentase. Semakin kecil nilai MAPE, semakin baik performa model dalam memprediksi data aktual [4], [5].

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right|$$

III. METODE/DESAIN

Penelitian ini menggunakan pendekatan kuantitatif untuk mengembangkan model prediksi *Real Power* dengan metode *Multi Layer Perceptron* (MLP). Data yang digunakan merupakan rekaman data kelistrikan Tingkat tinggi dari sebuah rumah indekos yang dihuni oleh 7 orang selama 7 hari [1]. Pengukuran aktual berupa *Real Power* (kW) dan *Apparent Power* (kVA). Nilai *Power Factor* (PF) diperoleh melalui perhitungan menggunakan persamaan $PF=P/S$. seluruh tahapan pengolahan data, pelatihan model, dan proses prediksi dilakukan menggunakan *MATLAB R2018a*.

A. Data Penelitian

Data penelitian diperoleh dari hasil pengukuran parameter kelistrikan yang terdiri atas *Real Power* dan *Apparent Power* selama tujuh hari pengamatan. Selanjutnya, nilai *Power Factor* dihitung berdasarkan perbandingan antara *Real Power* dan *Apparent Power* sehingga diperoleh tiga parameter yang digunakan sebagai masukan model MLP.

Tabel 1. Data rekaman harian

Hari	Real Power (kW)	Apparent Power (kVA)
1	162,2554	180,7906
2	155,0452	173,4676
3	155,7388	174,2482
4	152,4227	170,7976
5	165,3632	184,1306
6	154,8861	173,4588
7	157,7194	176,5723

B. Perhitungan Power Factor

Nilai *Power Factor* dihitung berdasarkan perbandingan antara *Real Power* dan *Apparent Power* dengan persamaan berikut.

$$PF = \frac{P}{S}$$

Sebagai contoh pada data hari pertama diperoleh:

$$PF = \frac{162.2554}{180.7906} = 0.8975$$

Perhitungan tersebut dilakukan pada keseluruhan data sehingga diperoleh nilai *Power Factor* yang selanjutnya digunakan sebagai salah satu variabel masukan.

C. Normalisasi Data

Sebelum proses pelatihan, seluruh data dinormalisasikan menggunakan metode *Min-Max* agar berada pada rentang 0-1. Sebagai contoh normalisasi *Real Power* hari pertama:

$$\begin{aligned} x &= 162.2554 \\ x_{\min} &= 152.4227 \\ x_{\max} &= 165.3632 \end{aligned}$$

$$x_{\text{norm}} = \frac{162.2554 - 152.4227}{165.3632 - 152.4227}$$

$$x_{\text{norm}} = 0,7599$$

D. Pembentukan Data Input dan Target

Model MLP menggunakan tiga variable masukan, yaitu: *Real Power*, *Apparent Power*, dan *Power Factor*. Dengan skema tersebut diperoleh enam pasang data yang digunakan dalam proses pelatihan jaringan.

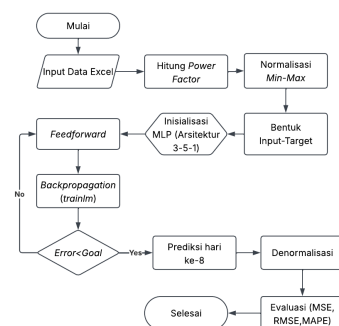
E. Arsitektur Jaringan MLP

Jaringan MLP yang digunakan terdiri atas tiga lapisan yaitu *input layer*, *hidden layer* dan *output layer*. *Input layer* memiliki tiga neuron yang mewakili variable *Real Power*, *Apparent Power*, dan *Power Factor*, sedangkan *hidden layer* terdiri atas lima neuron dengan fungsi aktivasi *tansig*. *Output layer* memiliki satu neuron dengan fungsi aktivasi *purelin* yang menghasilkan prediksi *Real Power*.

F. Proses Pelatihan dan Prediksi

Tahapan penelitian diawali dengan pembacaan data dari *Microsoft Excel*, kemudian dilakukan perhitungan *Power Factor*, normalisasi data menggunakan metode *Min-Max*, serta pembentukan pasangan input dan target. Selanjutnya dibangun model MLP untuk proses pelatihan menggunakan *algoritma trainlm*. Setelah model selesai dilatih, jaringan digunakan untuk memprediksi nilai *Real Power* pada hari berikutnya. Hasil prediksi kemudian dikembalikan ke skala asli melalui proses denormalisasi dan evaluasi menggunakan nilai *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE).

G. Flowchart



Gambar 1. Flowchart metode penelitian MLP.

Tahapan penelitian secara keseluruhan seperti Gambar 1. *Flowchart* menunjukkan urutan proses mulai dari pengambilan data, perhitungan *Power Factor*, normalisasi, pembentukan data masukan dan target, pelatihan jaringan MLP, proses prediksi, hingga evaluasi kinerja model menggunakan parameter MSE, RMSE, dan MAPE.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengolahan Data

Tahapan awal penelitian dilakukan dengan mengolah data hasil pengukuran yang terdiri atas *Real Power* dan

Apparent Power selama tujuh hari. Selanjutnya nilai *Power Factor* dihitung menggunakan persamaan $PF=P/S$. Ketiga variabel tersebut digunakan sebagai data masukan pada proses pelatihan jaringan saraf tiruan.

Sebelum proses pelatihan dilakukan, seluruh data dinormalisasi menggunakan metode *Min-Max* agar berada pada rentang 0–1. Normalisasi bertujuan mengurangi perbedaan skala antarvariabel sehingga proses pembelajaran jaringan menjadi lebih stabil dan konvergen.

Tabel 2. Tabel *Real Power*, *Apparent Power*, dan *Power Factor*

Hari	Real Power (kW)	Apparent Power (kVA)	Power Factor
1	162,2554	180,7906	0,897476971
2	155,0452	173,4676	0,893799188
3	155,7388	174,2482	0,89377566
4	152,4227	170,7976	0,892417107
5	165,3632	184,1306	0,898075605
6	154,8861	173,4588	0,892927312
7	157,7194	176,5723	0,89322844

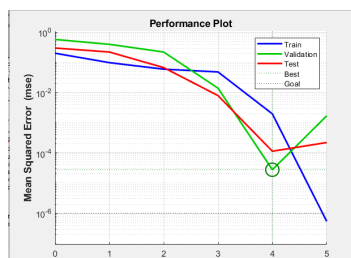
B. Hasil Pelatihan Jaringan Multi Layer Perceptron

Model prediksi dibangun menggunakan arsitektur 3-5-1, yang terdiri atas tiga neuron pada lapisan input, lima neuron pada *hidden layer*, dan satu neuron pada lapisan *output*. Fungsi aktivasi yang digunakan Adalah tansig pada *hidden layer* dan *purelin* pada *output layer*, sedangkan proses pelatihan menggunakan *algoritma Levenberg*-

C. Hasil Training Jaringan

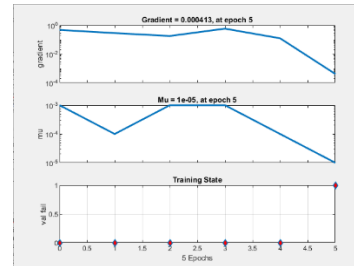
Selama proses pengujian, MATLAB menghasilkan beberapa grafik yang digunakan untuk mengevaluasi proses pembelajaran jaringan.

Performance plot menunjukkan perubahan nilai *Mean Square Error* (MSE) terhadap jumlah epoch selama proses pelatihan. Grafik ini digunakan untuk mengetahui apakah jaringan telah menjapai kondisi konvergen.



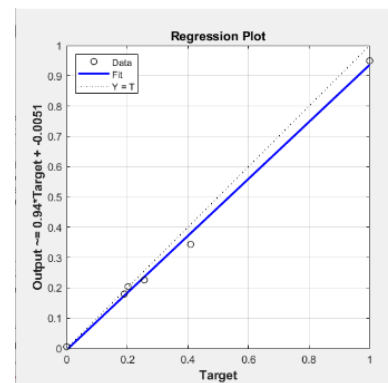
Gambar 2. *Performance plot*.

Setelah pengujian pada Gambar 2, nilai MSE mengalami penurunan seiring bertambahnya jumlah epoch hingga mencapai kondisi terbaik (*best performance*). Hal tersebut menunjukkan bahwa proses pelatihan berhasil meminimalisir kesalahan prediksi dan jaringan mampu mempelajari hubungan antara variabel masukan.



Gambar 3. *Training state*.

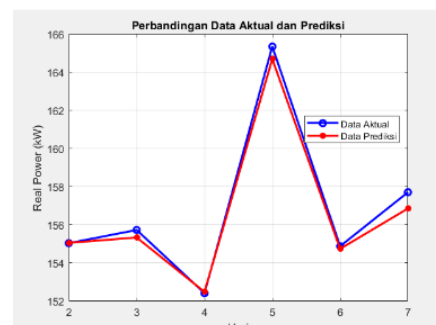
Berdasarkan Gambar 3, menunjukkan kondisi pelatihan jaringan selama proses optimasi. Penurunan nilai gradien mengindikasikan bahwa bobot jaringan semakin mendekati kondisi optimum sehingga proses pelatihan dapat dihentikan sesuai kriteria yang telah ditentukan.



Gambar 4. *Regresi plot*

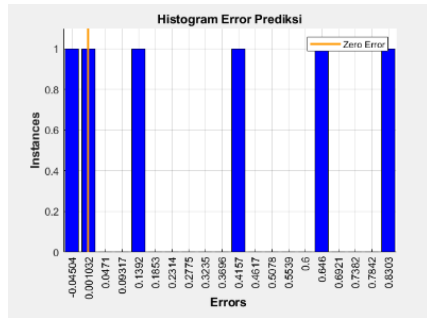
Nilai koefisien regresi (R) yang mendekati 1 menunjukkan bahwa hasil prediksi memiliki tingkat kesesuaian yang baik terhadap data aktual. Semakin tinggi nilai R, semakin baik kemampuan jaringan dalam memodelkan pola data.

D. Hasil Prediksi Beban Listrik



Gambar 5. Perbandingan data aktual dan prediksi.

Memperlihatkan bahwa pola hasil prediksi mengikuti kecenderungan perubahan data aktual. Meskipun terdapat selisih pada beberapa titik pengamatan, secara umum model mengikuti perubahan beban Listrik harian sehingga dapat digunakan sebagai model prediksi jangka pendek.



Gambar 6. Error histogram.

Berdasarkan Gambar 6, Sebagian besar nilai error berada di sekitar nol. Hal tersebut menunjukkan bahwa model memiliki Tingkat kesalahan yang relative kecil dan hasil prediksi masih berapa dalam batas yang dapat diterima.

E. Evaluasi Model

Parameter	Nilai
MSE	0.223119
RME	0.472355
MAPE	0.2226 %

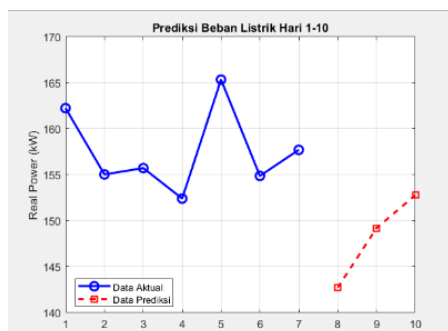
Tabel 3. Hasil evaluasi model.

Nilai MSE menunjukkan rata-rata kuadrat kesalahan antara hasil prediksi dan data aktual. Nilai RMSE menggambarkan besar penyimpangan prediksi terhadap data sebenarnya dalam satuan yang sama dengan variabel target, sedangkan MAPE menunjukkan persentase kesalahan rata-rata model. Semakin kecil nilai ketiga parameter tersebut, semakin baik kemampuan model dalam melakukan prediksi.

F. Prediksi Real Power Hari ke-8 hingga ke-10

Tabel 4. Hasil prediksi hari ke-8 hingga ke-10

Hari	Real Power (kW)	Apparent Power (kVA)	Power Factor
8	142.71	159.54	0.89453
9	149.21	166.8	0.89453
10	152.8	170.81	0.89453



Gambar 7. Grafik prediksi hari ke-8 hingga hari ke-10

Dari Tabel 4, dan Gambar 7 menunjukkan hasil prediksi beban listrik setelah periode pengamatan berakhir. Nilai prediksi diperoleh dari model MLP yang telah dilatih menggunakan data histori. Hasil tersebut memberikan Gambaran mengenai kecenderungan perubahan beban listrik pada periode berikutnya dan dapat dimanfaatkan sebagai informasi pendukung dalam perencanaan operasi sistem tenaga listrik.

V. KESIMPULAN

Penelitian ini berhasil menerapkan metode *Multi Layer Perceptron* (MLP) untuk memprediksi beban listrik menggunakan rekaman data kelistrikan dari sebuah rumah indekos dengan variabel *Real Power*, *Apparent Power*, dan *Power Factor* sebagai data masukan. Tahapan penelitian meliputi perhitungan *Power Factor*, normalisasi data menggunakan metode *Min-Max*, pembentukan data input dan target, pelatihan jaringan MLP, serta evaluasi menggunakan MSE, RMSE, dan MAPE.

Berdasarkan hasil simulasi MATLAB, model MLP mampu mengikuti pola perubahan data historis dan menghasilkan prediksi *Real Power* pada periode berikutnya. Meskipun demikian, jumlah data yang terbatas menyebabkan kemampuan generalisasi model masih belum optimal. Oleh karena itu penelitian selanjutnya disarankan menggunakan data historis lebih banyak serta melakukan optimasi parameter jaringan agar akurasi prediksi dapat ditingkatkan.

REFERENSI

- [1] N. D. Sidiq, "Analisa Listrik Kos," Kaggle, 2026. [Online]. Available: <https://www.kaggle.com/datasets/nauvaldhonandsidiq/analisa-listrik-kos>
- [2] G. Dudek, "Multilayer Perceptron for Short-Term Load Forecasting: From Global to Local Approach," *Neural Computing and Applications*, vol. 32, no. 10, pp. 3695–3707, 2020, doi: 10.1007/s00521-019-04130-y.
- [3] F. Baskoro, F. Alamsyah, B. Suprianto, and W. Ariwibowo, "Peramalan Beban Listrik Harian Menggunakan Artificial Neural Network," *Jurnal Teknik Elektro*, vol. 10, no. 1, pp. 203–209, 2021, doi: 10.26740/jte.v10n1.p203-209.
- [4] P. V. Matrenin, V. Z. Manusov, A. I. Khalyasmaa, D. V. Antonenkov, S. A. Eroshenko, and D. N. Butusov, "Improving Accuracy and Generalization Performance of Small-Size Recurrent Neural Networks Applied to Short-Term Load Forecasting," *Mathematics*, vol. 8, no. 12, p. 2169, 2020, doi: 10.3390/math8122169.
- [5] Z. Hamad and I. Abdulrahman, "Deep Learning-Based Load Forecasting Considering Data Reshaping Using MATLAB/Simulink," *International Journal of Energy and Environmental Engineering*, vol. 13, no. 2, pp. 853–869, 2022, doi: 10.1007/s40095-022-00480-x.
- [6] Y. Chen and D. Zhang, "Theory-Guided Deep Learning for Electrical Load Forecasting (TgDLF) via Ensemble Long Short-Term Memory," *Advances in Applied Energy*, vol. 1, p. 100004, 2021, doi: 10.1016/j.adapen.2020.100004.

PENERAPAN JARINGAN SARAF TIRUAN MODEL MCCULLOCH-PITTS PADA SISTEM PENGENDALIAN LAMPU TAMAN OTOMATIS

Adiel Kevin Kharisty S.^{1*}, Ruli Gunawan²

¹²Program Studi Teknik Elektro Fakultas Teknik dan Informatika Universitas PGRI Semarang

*Email : satriawanadiel@gmail.com

Abstrak— Penggunaan energi listrik pada fasilitas umum seperti lampu taman sering terjadi pemakaian tidak efektif akibat kelalaian manusia dalam mematikan sakelar atau ketergantungan pada sensor tunggal yang kurang akurat. Penelitian ini bertujuan untuk mengimplementasikan Jaringan Saraf Tiruan (JST) model McCulloch-Pitts sebagai sistem pengambil keputusan biner pada sistem pengendalian lampu taman otomatis. Sistem ini menggunakan kombinasi input dari sensor intensitas cahaya (LDR) dan sensor gerak (PIR), serta dilengkapi dengan tombol manual override sebagai fungsi inhibisi. Model McCulloch-Pitts dikonfigurasi menggunakan fungsi aktivasi threshold keras untuk menentukan status output berupa lampu taman (menyala atau mati). Hasil pengujian menunjukkan bahwa model McCulloch-Pitts mampu mengklasifikasikan kondisi lingkungan dengan akurasi 100% berdasarkan nilai ambang batas (threshold) yang ditentukan, memberikan solusi komputasi yang ringan, hemat energi, dan responsif untuk sistem tertanam (embedded system).

Kata kunci: Jaringan Saraf Tiruan; McCulloch-Pitts; Lampu Taman Otomatis; LDR; PIR.

I. PENDAHULUAN

Efisiensi penggunaan energi listrik menjadi salah satu fokus utama dalam pengembangan infrastruktur cerdas modern. Penerangan luar ruangan, seperti lampu taman, sering kali menyala tanpa efisiensi akibat sistem yang masih manual atau hanya mengandalkan sensor cahaya tunggal, yang rentan memicu lampu tetap menyala saat mendung tebal padahal tidak ada aktivitas di sekitar taman.

Untuk meningkatkan kecerdasan sistem kendali, penerapan Jaringan Saraf Tiruan (JST) menjadi salah satu solusi yang efektif. Model McCulloch-Pitts adalah salah satu bentuk JST paling awal dan sederhana, namun sangat kuat dalam memproses logika biner. Model ini bekerja menyerupai cara kerja neuron biologis, di mana ia menerima beberapa input berbobot dan menghasilkan output biner (0 atau 1) berdasarkan nilai ambang batas (threshold).

Penelitian ini memanfaatkan kesederhanaan komputasi McCulloch-Pitts agar dapat ditanamkan pada mikrokontroler dengan spesifikasi rendah, guna mengendalikan lampu taman secara real-time, akurat, dan hemat daya.

II. STUDI PUSTAKA

A. Jaringan Syaraf Tiruan

Jaringan Saraf Tiruan (artificial neural network) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasinya diadopsi dari pengetahuan sel saraf biologi di dalam otak [1]. Jaringan saraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba mensimulasikan proses pembelajaran pada otak

manusia tersebut. Istilah buatan di sini digunakan karena jaringan saraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran [2].

B. McCulloch-Pitts

Model Neuron McCulloch-Pitts merupakan model jaringan saraf tiruan pertama yang diperkenalkan oleh Warren McCulloch dan Walter Pitts pada tahun 1943. Model ini meniru cara kerja neuron biologis dengan menggunakan konsep logika biner, di mana keluaran neuron hanya memiliki dua keadaan, yaitu 1 (aktif) atau 0 (tidak aktif). Neuron menerima beberapa masukan (input), kemudian setiap masukan dikalikan dengan bobot (weight). Hasil perkalian tersebut dijumlahkan dan dibandingkan dengan nilai ambang (threshold). Apabila jumlah masukan lebih besar atau sama dengan nilai ambang, neuron akan menghasilkan keluaran 1, sedangkan jika lebih kecil dari nilai ambang maka keluarannya adalah 0 [3].

1) Persamaan Matematis

Jumlah masukan dihitung menggunakan persamaan:

$$\text{Net} = \sum x_i.w_i, i = 1 \dots n \quad (1)$$

dengan Net = total masukan (input), x_i = nilai input ke- i , w_i = bobot input ke- i , dan n = jumlah input.

Kemudian fungsi aktivasi threshold dinyatakan sebagai:

$$y = 1, \text{ jika } \text{Net} \geq \theta; y = 0, \text{ jika } \text{Net} < \theta \quad (2)$$

dengan y = keluaran neuron dan θ = nilai ambang (threshold).

2) Karakteristik Model McCulloch-Pitts

Karakteristik umum model McCulloch-Pitts meliputi beberapa hal sebagai berikut [4]:

1. Menggunakan input dan output biner (0 atau 1).
2. Memiliki bobot (weight) pada setiap input.
3. Menggunakan fungsi ambang (threshold).
4. Tidak memiliki proses pembelajaran (learning); bobot ditentukan sebelumnya.
5. Cocok digunakan untuk merepresentasikan gerbang logika seperti AND, OR, dan NOT.

C. Sensor Cahaya LDR

Sensor Light Dependent Resistor (LDR) adalah sensor yang bekerja berdasarkan perubahan nilai hambatan (resistansi) akibat perubahan intensitas cahaya. Semakin tinggi intensitas cahaya yang diterima, maka nilai resistansi LDR semakin kecil. Sebaliknya, pada kondisi gelap nilai resistansinya meningkat sehingga sensor dapat digunakan untuk mendeteksi kondisi terang atau gelap [5].

LDR dapat dipasang pada aneka rangkaian elektronika untuk memutuskan dan menyambungkan aliran listrik berdasarkan cahaya. Semakin banyak cahaya yang mengenai LDR, maka nilai resistansinya akan menurun. Semakin sedikit cahaya yang mengenai LDR, maka nilai resistansinya akan meningkat.

D. Sensor Gerak PIR

Sensor Passive Infrared (PIR) merupakan sensor yang mendeteksi radiasi inframerah yang dipancarkan oleh tubuh manusia atau objek hidup [6]. Sensor ini bersifat pasif karena tidak memancarkan sinyal inframerah sendiri, melainkan hanya menerima perubahan radiasi inframerah dari lingkungan. PIR banyak digunakan pada sistem keamanan, lampu otomatis, dan aplikasi smart home [7].

Sensor PIR bekerja dengan cara menangkap pancaran infra red, kemudian pancaran infra red yang tertangkap akan masuk melalui lensa Fresnel dan mengenai sensor pyroelektrik. Sinar infra red mengandung energi panas yang dapat membuat sensor pyroelektrik menghasilkan arus listrik. Arus listrik inilah yang akan menimbulkan tegangan dan dibaca secara analog oleh sensor. Kemudian komparator akan membandingkan sinyal yang sudah diterima dengan tegangan referensi tertentu berupa keluaran sinyal 1-bit.

Sensor PIR hanya akan mengeluarkan logika 0 dan 1. Logika 0 saat sensor tidak mendeteksi adanya perubahan pancaran infra red dan 1 saat sensor mendeteksi infra red. Sensor PIR hanya dapat mendeteksi pancaran infra red dengan panjang gelombang 8–14 mikrometer. Manusia memiliki suhu badan yang dapat menghasilkan pancaran infra red dengan panjang gelombang antara 9–10 mikrometer. Panjang gelombang tersebut dapat terdeteksi oleh sensor PIR sehingga membuat sensor ini sangat efektif digunakan sebagai human detector. Sensor PIR hanya akan mendeteksi jika objek bergerak atau secara teknis saat terjadi adanya perubahan pancaran infra red.

E. Google Colab

Google Colab (Google Colaboratory) adalah layanan berbasis cloud yang dikembangkan oleh Google untuk menjalankan notebook Jupyter secara daring. Google Colab mendukung bahasa pemrograman Python, menyediakan akses ke CPU, GPU, dan TPU tanpa perlu instalasi perangkat lunak di komputer pengguna, serta terintegrasi dengan Google Drive sehingga memudahkan penyimpanan, berbagi, dan kolaborasi dalam pengembangan program maupun analisis data. Karena bersifat gratis dan mudah diakses melalui peramban web, Google Colab banyak dimanfaatkan dalam pembelajaran pemrograman, machine learning, deep learning, dan penelitian berbasis kecerdasan buatan [8].

III. METODE/DESAIN

A. Metode Penelitian

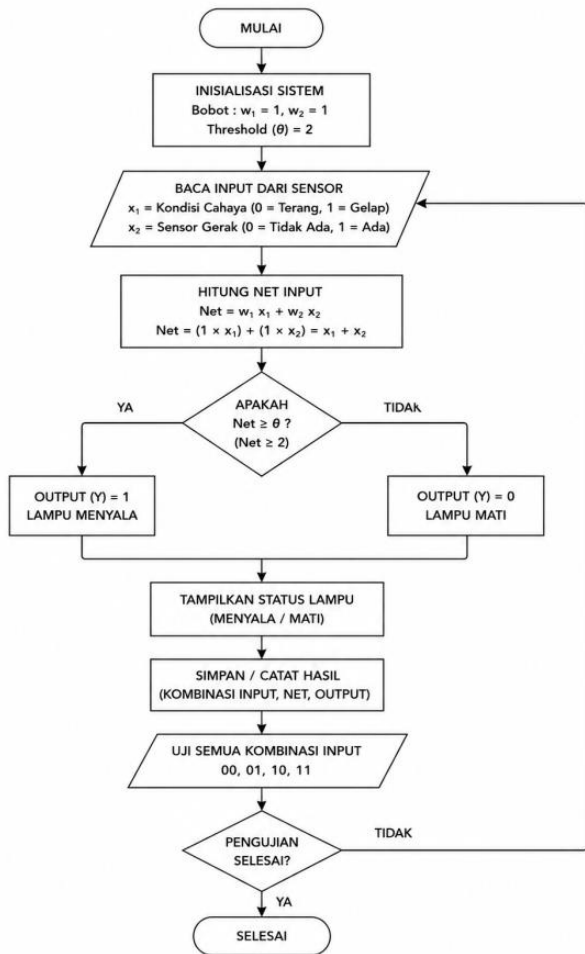
Penelitian ini menggunakan metode eksperimen (experimental research) dengan menerapkan Model Neuron McCulloch-Pitts sebagai dasar pengambilan keputusan untuk mengendalikan lampu taman otomatis. Sistem dirancang menggunakan dua variabel masukan, yaitu kondisi cahaya lingkungan dan keberadaan gerakan di area taman.

Tahapan penelitian dimulai dengan studi literatur mengenai jaringan saraf tiruan, khususnya model neuron McCulloch-Pitts. Selanjutnya dilakukan identifikasi kebutuhan sistem berupa penentuan variabel input dan output. Variabel input terdiri dari sensor cahaya (LDR) yang menghasilkan nilai biner x_1 (0 = terang, 1 = gelap) dan sensor gerak (PIR) yang menghasilkan nilai biner x_2 (0 = tidak ada gerakan, 1 = ada gerakan). Sedangkan output berupa kondisi lampu, yaitu 0 = lampu mati dan 1 = lampu menyala.

Pada tahap implementasi digunakan bobot masing-masing input sebesar $w_1 = 1$ dan $w_2 = 1$, serta nilai ambang (threshold) sebesar $\theta = 2$. Nilai masukan dihitung menggunakan persamaan (1), kemudian hasil perhitungan dibandingkan dengan nilai ambang menggunakan fungsi aktivasi pada persamaan (2). Hasil keluaran kemudian diuji pada seluruh kombinasi kondisi lingkungan untuk memastikan sistem bekerja sesuai fungsi logika AND, yaitu lampu hanya menyala ketika kondisi lingkungan gelap dan terdapat aktivitas di area taman.

B. Tahapan Penelitian

1. Studi literatur mengenai Jaringan Saraf Tiruan dan Model McCulloch-Pitts.
2. Analisis kebutuhan sistem.
3. Menentukan variabel input dan output.
4. Menentukan bobot dan nilai threshold.
5. Mengimplementasikan persamaan McCulloch-Pitts.
6. Melakukan simulasi seluruh kombinasi input.
7. Menganalisis hasil keluaran.
8. Menarik kesimpulan.



Gambar 1. Langkah penelitian.

C. Arsitektur Sistem

Sistem ini menggunakan dua parameter input utama yang dikonversi menjadi logika biner sebelum diproses oleh neuron McCulloch-Pitts:

- Input 1 (x_1): Sensor Intensitas Cahaya (LDR). Bernilai 1 jika kondisi lingkungan gelap/malam, bernilai 0 jika terang/siang.
- Input 2 (x_2): Sensor Gerak (PIR). Bernilai 1 jika terdeteksi adanya manusia/aktivitas di taman, bernilai 0 jika sebaliknya.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Berdasarkan Teori

Berdasarkan bobot dan nilai ambang yang telah ditentukan, nilai net input untuk seluruh kombinasi kondisi lingkungan (tanpa adanya intervensi dari tombol override) dapat dilihat pada Tabel 4.1.

B. Pengujian Menggunakan Google Colab

Melalui simulasi data pada Gambar 2, sistem McCulloch-Pitts berhasil mengimplementasikan fungsi logika AND dengan sempurna. Lampu hanya akan menyala jika kondisi benar-benar malam hari dan terdapat aktivitas di area taman. Hal ini secara signifikan memangkas konsumsi listrik saat malam hari ketika taman sedang tidak digunakan.

Tabel 1. Tabel Pengujian

Kondisi Lingkungan	x1	x2	Net	Y	Status Lampu
Siang-Sunyi	0	0	0	0	Mati
Siang-Aktivitas	0	1	1	0	Mati
Malam-Sunyi	1	0	1	0	Mati
Malam-Aktivitas	1	1	2	1	Menyala

SIMULASI KENDALI LAMPU TAMAN DENGAN NEURON MCCULLOCH-PITTS

Kondisi Lingkungan	x1	x2	Net	Output Y	Status Lampu
Siang - Sunyi	0	0	0	0	Lampu Mati
Siang - Ada Aktivitas	0	1	1	0	Lampu Mati
Malam - Sunyi	1	0	1	0	Lampu Mati
Malam - Ada Aktivitas	1	1	2	1	LAMPU MENYALA

Gambar 2. Hasil Google Colab

Keunggulan utama dari model ini adalah efisiensi memorinya. Karena tidak memerlukan proses training (pembelajaran) yang rumit seperti algoritma Backpropagation, algoritma ini dapat dieksekusi oleh mikrokontroler murah dalam hitungan mikrodetik, menjadikannya sangat andal untuk embedded system.

V. KESIMPULAN

Berdasarkan hasil perancangan simulasi melalui Google Colab, dapat disimpulkan beberapa poin utama sebagai berikut.

1. Keberhasilan Model: Model JST McCulloch-Pitts berhasil diimplementasikan sebagai algoritma pengambil keputusan pada sistem pengendalian lampu taman otomatis dengan tingkat akurasi 100%.
2. Efisiensi Energi: Melalui kombinasi logika gerbang AND, lampu hanya menyala pada kondisi malam hari yang memiliki aktivitas, mengoptimalkan penghematan energi listrik.
3. Fungsi Inhibisi Sempurna: Fitur inhibitory input bekerja dengan validitas penuh; tombol manual override mampu mematikan sistem lampu secara mutlak demi keselamatan operasional saat perawatan, meskipun kondisi lingkungan gelap dan terdapat gerakan.

REFERENSI

- [1] N. D. Sidiq, "Analisa Listrik Kos," Kaggle, 2026. [Online]. Available: <https://www.kaggle.com/datasets/nauvaldhonandsidiq/analisa-listrik-kos>
- [2] F. Dudek, "Multilayer Perceptron for Short-Term Load Forecasting: From Global to Local Approach," Neural Computing and Applications, vol. 32, no. 10, pp. 3695–3707, 2020, doi: 10.1007/s00521-019-04130-y.
- [3] F. Baskoro, F. Alamsyah, B. Suprianto, and W. Ariwibowo, "Peramalan Beban Listrik Harian Menggunakan Artificial Neural Network," Jurnal Teknik Elektro, vol. 10, no. 1, pp. 203–209, 2021, doi: 10.26740/jte.v10n1.p203-209.
- [4] P. V. Matrenin, V. Z. Manusov, A. I. Khalyasmaa, D. V. Antonenkov, S. A. Eroshenko, and D. N. Butusov, "Improving Accuracy and Generalization Performance of Small-Size Recurrent Neural Networks Applied to Short-Term Load

- Forecasting," *Mathematics*, vol. 8, no. 12, p. 2169, 2020, doi: 10.3390/math8122169.
- [5] Z. Hamad and I. Abdulrahman, "Deep Learning-Based Load Forecasting Considering Data Reshaping Using MATLAB/Simulink," *International Journal of Energy and Environmental Engineering*, vol. 13, no. 2, pp. 853–869, 2022, doi: 10.1007/s40095-022-00480-x.
- [6] Y. Chen and D. Zhang, "Theory-Guided Deep Learning for Electrical Load Forecasting (TgDLF) via Ensemble Long Short-Term Memory," *Advances in Applied Energy*, vol. 1, p. 100004, 2021, doi: 10.1016/j.adapen.2020.100004.