

SISTEM PARKIR OTOMATIS BERBASIS IOT BERDASARKAN DETEKSI PLAT NOMOR KENDARAAN MENGGUNAKAN YOLOV8 DAN RESNET50

Widi Pangestu Danar Wijanarko
Program Studi Teknologi Informasi
Universitas Telkom
Surabaya, Indonesia
danarwijanarko@student.telkomunivers
ity.ac.id

Muhammad Adib Kamali
Program Studi Teknologi Informasi
Universitas Telkom
Surabaya, Indonesia
adibmkamali@telkomuniversity.ac.id

Farah Zakiyah Rahmanti
Program Studi Teknologi Informasi
Universitas Telkom
Surabaya, Indonesia
line 4: Kota, Negara
farahzakiyah@telkomuniversity.ac.id

Abstrak — Identifikasi kendaraan manual untuk sistem parkir konvensional di Universitas Telkom Surabaya mengakibatkan masalah keamanan karena petugas tidak selalu berada di tempat parkir setiap saat. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem parkir otomatis yang mampu mendeteksi plat nomor kendaraan serta membuka palang pintu secara otomatis berdasarkan verifikasi identitas mahasiswa menggunakan Kartu Tanda Mahasiswa (KTM) berbasis RFID. Sistem ini menggabungkan model YOLOv8 untuk mendeteksi dan melakukan deteksi karakter pada plat nomor kendaraan, model ResNet-50 untuk melakukan klasifikasi karakter, sistem IoT untuk verifikasi identitas menggunakan RFID, serta sistem monitoring berbasis web. Hasil pengujian menunjukkan bahwa model YOLOv8 mencapai *precision* sebesar 92% dan *mAP50* sebesar 88% pada deteksi plat nomor, sedangkan model deteksi karakter mencapai *precision* sebesar 85% dan *mAP50* sebesar 85%. Model ResNet50 untuk klasifikasi karakter mencapai akurasi rata-rata sebesar 98% dengan rata-rata nilai F1-Score pada setiap kelas diatas 95%. Sistem juga berhasil dijalankan pada perangkat *edge computing* menggunakan *Raspberry Pi 5*, dengan waktu pemrosesan keseluruhan sekitar 3 - 4 detik. Dengan adanya sistem ini dapat meningkatkan efisiensi, keamanan, dan kenyamanan dalam pengelolaan parkir kampus secara otomatis.

Kata kunci— Sistem Parkir Otomatis, YOLOv8, ResNet50, IoT, edge Computing

I. PENDAHULUAN

Universitas Telkom Surabaya masih menggunakan metode parkir konvensional. Dalam metode ini, mahasiswa harus menyerahkan STNK saat keluar. Sistem ini dinilai kurang efisien seiring dengan meningkatnya jumlah kendaraan bermotor di Indonesia [1]. Oleh karena itu, kebutuhan akan sistem parkir yang lebih efisien dan otomatis semakin penting. Hal ini dapat mengakibatkan berkurangnya tingkat keamanan karena petugas tidak selalu berada

ditempat parkir setiap saat, sehingga dapat menimbulkan ketidaknyamanan bagi para pengguna.

Untuk mengatasi permasalahan ini, telah dilakukan beberapa penelitian untuk mengembangkan sistem parkir otomatis berbasis deteksi pelat nomor kendaraan. Metode yang sering dipakai adalah *Convolutional Neural Network* (CNN) dan *Optical Character Recognition* (OCR) [1], [2]. Sistem ini mampu mendeteksi dan mengenali nomor plat kendaraan secara otomatis ketika masuk dan keluar area parkir, sehingga dapat mempercepat waktu pemeriksaan dan meningkatkan efisiensi [1], [3]. Namun, pada penelitian sebelumnya masih menghadapi sejumlah batasan yaitu deteksi plat nomor dilakukan dengan *input* berupa gambar [1], [4]. Selain itu, beberapa penelitian hanya berfokus pada pengenalan nomor plat tanpa adanya integrasi dengan sistem parkir yang lebih komprehensif [5], [6].

Oleh karena itu, tujuan dari penelitian ini adalah untuk mengembangkan sistem parkir otomatis yang memanfaatkan teknologi pengenalan pelat nomor kendaraan bermotor berbasis YOLO (*You Only Look Once*) dan ResNet50 (*Residual Network 50 Layers*). Metode YOLO telah terbukti efektif dalam mendeteksi objek dengan tingkat akurasi yang tinggi [6]. Di sisi lain, ResNet50 mampu mengenali karakter pada plat nomor dengan baik [7]. Dengan menggabungkan kedua metode ini serta teknologi RFID (*Radio Frequency Identification*) yang akan digunakan untuk verifikasi pengguna, diharapkan dapat memberikan solusi keamanan, kenyamanan, dan efisiensi untuk masalah parkir di Universitas Telkom Surabaya.

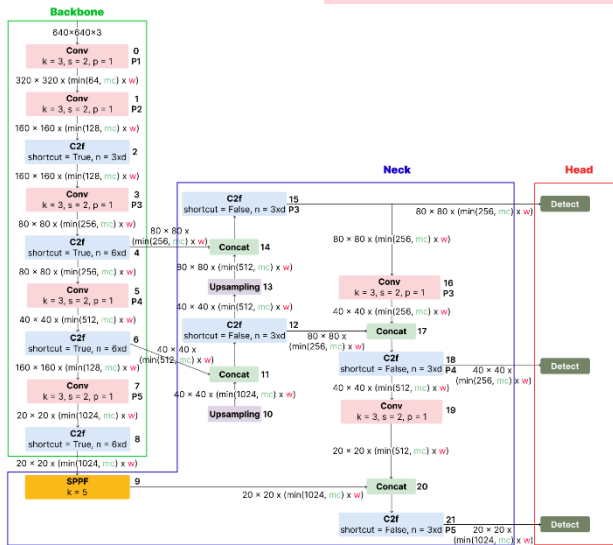
II. KAJIAN TEORI

A. You Only Look Once (YOLO)

You Only Look Once (YOLO) merupakan metode deteksi objek yang merancang proses deteksi sebagai masalah *regresi*. Berbeda dengan metode sebelumnya yang memanfaatkan pengklasifikasi, YOLO menggunakan sebuah

jaringan saraf *Convolutional Neural Network (CNN)* yang mampu memprediksi *bounding boxes* dan *class probabilities* secara langsung dari gambar penuh dalam satu kali evaluasi [8].

YOLOv8 adalah versi yang dirilis oleh Ultralytics pada 10 Januari 2023 [9]. menggunakan model *anchor-free* dengan kepala (*head*) terpisah untuk memproses tugas-tugas *objectness*, *classification*, dan *regression* secara independen, memungkinkan setiap cabang untuk fokus pada tugasnya dan meningkatkan akurasi model secara keseluruhan [10]. Arsitektur ini mengintegrasikan beberapa *convolutional layers* yang berfungsi untuk mengekstraksi fitur dari gambar input. Proses ini melibatkan beberapa tahapan, seperti *backbone*, *neck*, dan *head* [9]. Pada gambar 2 merupakan arsitektur dari YOLOv8.



Gambar 1 Arsitektur YOLOv8

B. Convolutional Neural Network (CNN)

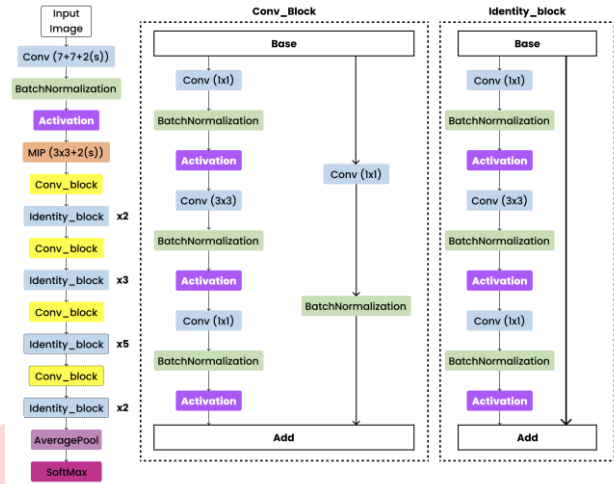
Convolutional Neural Network (CNN) adalah arsitektur jaringan saraf khusus untuk pengolahan gambar yang meniru cara kerja penglihatan manusia. Prosesnya diawali dengan Convolutional Layers yang berfungsi mengekstraksi fitur-fitur lokal dari gambar, seperti tepi dan tekstur, menggunakan filter khusus. Selanjutnya, Pooling Layers bertugas mengurangi dimensi data dari fitur tersebut untuk membuat komputasi lebih efisien. Terakhir, Fully Connected Layers menerima semua informasi fitur yang telah diproses, lalu menggabungkannya untuk melakukan klasifikasi akhir dan mengidentifikasi objek di dalam gambar [11].

C. Residual Network (ResNet)

Residual Network (ResNet) merupakan salah satu model spesifik dari CNN yang menerapkan *skip connections* atau *shortcut*. ResNet dirancang agar performa dari model ini tidak menurun walaupun arsitekturnya semakin dalam [12]. ResNet memiliki kinerja yang lebih baik dari pada model lain dan dapat meningkatkan akurasi dengan meningkatkan kedalaman arsitektur [13].

ResNet-50 adalah salah satu variasi dengan 50 *layers*. Jaringan ini memiliki 48-layer *convolutional* bersama dengan 1 *MaxPool* layer dan 1 *AveragePool* hal ini dapat dilihat pada gambar 3. ResNet didasari pada pembelajaran yang dalam,

yang memungkinkan pemecahan masalah mengenai *vanishing gradient* [14].



Gambar 2 Arsitektur ResNet-50

D. Confusion Matrix

Confusion Matrix digunakan untuk mengevaluasi hasil dari klasifikasi dengan nilai *accuracy*, *precision*, *recall* dan *F1-Score* dengan rumus masing-masing *matrix* berikut [15]:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (4)$$

Keterangan:

TP (True Positive) adalah nilai dimana model berhasil memprediksi hasil nilai positif dengan benar.
TN (True Negative) adalah Nilai di mana model berhasil memprediksi hasil nilai negatif dengan benar.
FP (False Positive) adalah Nilai di mana model salah memprediksi hasil nilai negatif sebagai positif.
FN (False Negative) adalah Nilai di mana model salah memprediksi hasil nilai positif sebagai negatif.
F1-Score adalah Menghitung perbandingan antara nilai *precision* dan nilai *recall* [16].

E. Precision, Recall, mAP

Precision, Recall, dan *mean Average Precision (mAP)* merupakan metrik evaluasi utama yang digunakan untuk menilai performa model dalam melakukan deteksi objek. *Precision* mengukur tingkat ketepatan model dalam memprediksi objek positif dengan benar, yaitu rasio antara jumlah prediksi benar (*True Positive*) terhadap total prediksi positif (*True Positive + False Positive*) pada persamaan (2). Semakin tinggi nilai *precision*, maka semakin sedikit kesalahan dalam prediksi positif yang dilakukan oleh model [17].

Sementara itu, *recall* mengukur kemampuan model dalam menemukan seluruh objek positif yang ada, yaitu rasio antara jumlah prediksi benar terhadap total objek positif yang sebenarnya (*True Positive + False Negative*) pada persamaan (3). *Recall* yang tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar objek yang relevan [17]. Untuk memperoleh gambaran performa model secara menyeluruh, digunakan metrik *mean Average Precision* (mAP) yang merupakan rata-rata dari nilai *Average Precision* (AP) pada berbagai tingkat ambang batas (*threshold*) *Intersection over Union* (IoU) [17] dengan rumus pada persamaan (5).

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k \quad (5)$$

F. Internet of Things (IoT)

Internet of Things (IoT) merupakan jaringan perangkat fisik yang saling terhubung melalui internet, yang memungkinkan pertukaran data dan interaksi secara *real-time* [18]. Pada saat ini IoT telah berkembang dengan pesat, mencakup berbagai aplikasi mulai dari rumah pintar, tempat parkir, dan kesehatan [1], [19], [20], [21].

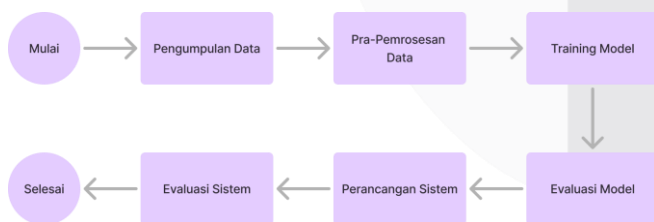
G. Blackbox Testing

Black box testing merupakan salah satu metode pengujian perangkat lunak yang berfokus pada fungsi eksternal sistem tanpa mempertimbangkan struktur internal, logika, atau kode program yang mendasarinya. Pengujian dilakukan dengan memberikan berbagai macam input, kemudian mengevaluasi hasil keluaran sistem apakah sesuai dengan yang diharapkan atau tidak [22].

III. METODE

A. Alur Penelitian

Pada penelitian ini terdapat beberapa proses yang akan dilakukan dengan pengumpulan data, pra-pemrosesan data, training model, evaluasi model, perancangan sistem, dan evaluasi sistem. Alur proses dapat dilihat dari Gambar 3 di bawah ini.



Gambar 3 Alur Penelitian

B. Pengumpulan Data

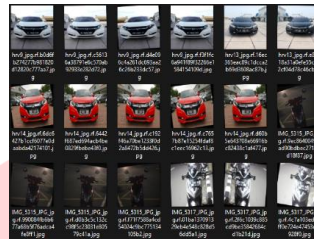
Penelitian ini menggunakan dua jenis data yaitu primer dan sekunder.

1. Data Primer

Berupa rekaman video langsung dari area parkir Universitas Telkom Surabaya. Data ini digunakan untuk menguji coba sistem dalam kondisi nyata dengan berbagai variasi pencahayaan dan sudut kamera.

2. Data Sekunder

Berasal dari data yang diunduh dari internet dari platform roboflow dengan 4.573 citra plat nomor kendaraan di Indonesia untuk deteksi plat nomor, kemudian 3.841 citra plat nomor yang setiap karakternya telah diberi label bounding box untuk dataset deteksi karakter dan untuk Dataset klasifikasi karakter didapatkan dari kaggle yang terdiri dari 36 kelas (huruf A-Z dan angka 0-9) dengan variasi jenis *font* yang beragam. Untuk tiga tipe *dataset* nya dapat dilihat pada gambar 4 (A) sampai gambar 4 (C).



Gambar 4 Dataset Deteksi Plat Nomor YOLOv8 (A)



Gambar 4 Dataset Klasifikasi Karakter ResNet50 (B)



Gambar 4 Dataset Deteksi Karakter YOLOv8 (B)

C. Pra-pemrosesan Data

Sebelum memasuki tahap *training* model, *dataset* dilakukan pra-pemrosesan data terlebih dahulu, di mana dalam tahap ini *dataset* akan dilakukan normalisasi, augmentasi, dan pembersihan data untuk memastikan kualitas *dataset* yang optimal. Dalam tahap pra-pemrosesan data dibagi menjadi tiga bagian yaitu sebagai berikut:

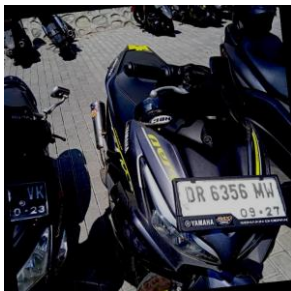
1. Dataset Deteksi Plat Nomor



Gambar 5 Dataset deteksi plat nomor dengan bounding box

Pada gambar 5 terlihat dataset pertama digunakan untuk melatih model deteksi plat nomor kendaraan, model yang bertugas mencari yang muncul dari video kendaraan secara

utih. Dataset diunduh dari platform *Roboflow* dengan total 4573 citra plat nomor kendaraan indonesia, yang terbagi menjadi 80% data latih, 10% data validasi, dan 10% data uji dengan efek *stretch* pada ukuran 640x640 pixel. Masing-masing citra telah dilabeli dengan *bounding box* yang mengelilingi area plat nomor.



Gambar 6 Citra Penyesuaian exposure dan rotasi (A)



Gambar 6 Citra penyesuaian kecerahan dan bounding box shear (B)

Gambar 6 (A) dan (B) menunjukkan proses augmentasi seperti rotasi antara -15° dan $+15^\circ$, penyesuaian kecerahan antara -45% sampai $+45\%$, penyesuaian *exposure* antara -25% sampai $+25\%$, dan transformasi shear pada bounding box secara acak antara -15° sampai $+15^\circ$ horizontal dan -10 sampai $+10$ vertikal untuk meningkatkan keberagaman data serta mengurangi *overfitting*.

2. Dataset Deteksi Karakter Plat Nomor



Gambar 7 Dataset deteksi karakter plat nomor dengan bounding box

Dataset yang kedua digunakan untuk melatih model deteksi karakter, model yang bertugas mencari dan memisahkan tiap huruf atau angka yang muncul pada plat nomor. Dataset diunduh dari platform *Roboflow* dengan total 3449 citra plat nomor, yang terbagi menjadi 80% data latih, 10% data validasi, dan 10% data uji. Dataset ini terdiri atas sekumpulan gambar plat nomor yang sudah diberi label *bounding box* di sekeliling masing-masing karakter yang dapat dilihat pada Gambar 8.

Kemudian pada Gambar 9 (A) dan (B) menunjukkan proses augmentasi seperti rotasi citra antara -15° sampai $+15^\circ$, penyesuaian kecerahan antara -25% sampai $+25\%$, penyesuaian *exposure* antara -15% sampai $+15\%$ dan transformasi shear secara acak antara -15° sampai $+15^\circ$ horizontal dan -15° sampai $+15^\circ$ vertikal.



Gambar 8 Citra Penyesuaian kecerahan dan rotasi (A)



Gambar 8 Citra penyesuaian exposure dan transformasi shear (B)

3. Dataset Klasifikasi Karakter

Dataset ketiga dipakai untuk melatih model klasifikasi karakter, model yang bertugas mengenali setiap huruf atau angka setelah proses deteksi karakter memisahkannya. Data ini diambil dari *Kaggle* dan mencakup 36 kelas (A-Z dan 0-9), dengan masing-masing kelas rata-rata berisi sekitar 1500 citra kecuali pada kelas "O" yang hanya memiliki sekitar 700 citra. Seluruh citra kemudian dibagi menjadi 80% untuk data latih, 10% untuk data validasi, dan 10% untuk data uji.

Dengan pembagian data ini memastikan bahwa model yang dihasilkan memiliki performa yang baik dalam melakukan tugasnya.

D. Training Model

Pada tahap ini data yang sudah dilakukan pra-pemrosesan akan dilatih dengan beberapa konfigurasi dan dilakukan *tuning hyperparameter* yang menghasilkan nilai seperti pada Tabel 3.3 yang akan digunakan pada pelatihan model. Pada pelatihan model akan menghasilkan nilai seperti *accuracy*, *precision*, *recall*, *mAP50*, dan *F1-Score*, yang kemudian akan dibandingkan mana yang terbaik antara beberapa model yang telah dilatih.

Tabel 1 Hyperparameter Tuning

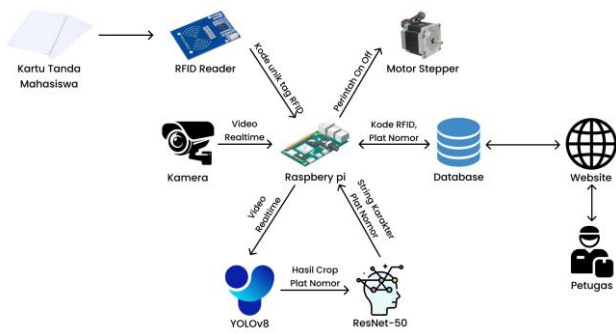
Parameter	Deskripsi
Epochs	Jumlah total pelatihan iterasi untuk melewati seluruh dataset baik secara forward maupun backward.
Batch Size	Jumlah sampel data yang harus dikerjakan sebelum memperbarui parameter model.
Image Size	Ukuran gambar yang digunakan untuk pelatihan model dalam piksel.
lr0	Learning rate awal, memengaruhi stabilitas pelatihan awal.
lrf	Faktor penurunan learning rate hingga akhir epoch.
Optimizer	Algoritma dalam kecerdasan buatan untuk menyesuaikan nilai parameter.

E. Evaluasi Model

Tahap ini adalah pengujian untuk melihat seberapa baik model YOLOv8 dan ResNet-50 dalam mendeteksi dan membaca plat nomor. Kami mengukurnya menggunakan beberapa metrik, seperti *Precision* dan *Recall*. Proses ini dilakukan pada data uji yang baru, bukan data yang dipakai untuk melatih model. Model dengan skor terbaik dari hasil

pengujian ini akan dipilih untuk digunakan pada sistem yang sebenarnya.

F. Perancangan Sistem



Gambar 9 Diagram Sistem

Seperti pada gambar 4 sistem parkir otomatis berbasis *Internet of Things (IoT)* yang menggunakan Kartu Tanda Mahasiswa (KTM) sebagai akses. Mahasiswa mendaftarkan KTM mereka terlebih dahulu kepada petugas. Saat masuk, mahasiswa menempelkan KTM pada *RFID reader*, kemudian *Raspberry Pi* memvalidasi data melalui *database*. Jika valid, kamera akan mengambil gambar plat nomor secara *real-time* yang dideteksi oleh *YOLOv8* dan dikenali karakternya oleh *ResNet-50*. Hasil plat nomor disimpan ke *database* dan *Raspberry Pi* mengaktifkan *motor stepper* untuk membuka palang. Proses keluar serupa, namun hasil pengenalan plat nomor akan dicocokkan dengan data saat masuk di *database*; jika cocok, palang akan terbuka. Seluruh aktivitas, termasuk data mahasiswa, plat nomor, dan waktu, dapat dipantau oleh petugas melalui sebuah *website monitoring*.

G. Alat dan Bahan Penelitian

Dalam penelitian ini, alat dan bahan yang terlibat meliputi perangkat keras yang tertera pada Tabel 2 (A) serta perangkat lunak yang tertera pada Tabel 2 (B), keduanya saling mendukung dalam proses pengembangan sistem deteksi plat nomor kendaraan sekaligus pengendalian otomatis pintu parkir.

Tabel 2 Perangkat Keras (A)

No	Perangkat Keras	Spesifikasi
1	Laptop	Intel Core i5 12500H / 16GB / 1TB
2	Kamera	Resolusi 1920x1080 / 30 FPS
3	Raspberry Pi 5	ARM-Cortex / 8GB / 32GB
4	RFID	13.56MHz RFID module
5	Motor Stepper	Torsi >34.3mN.m

Tabel 2 Perangkat Lunak (B)

No	Perangkat Lunak	Spesifikasi
1	Windows OS	Versi 11

2	Raspbian OS	Versi 12
3	PHP	Versi 8.2
4	Python	Versi 3.11.2
5	OpenCV	Versi 4.11
6	Ultralytics	Versi 8.3.123
7	Tensorflow	Versi 2.19
8	GPIOZero	Versi 2.0.1

IV. HASIL DAN PEMBAHASAN

A. Pelatihan Model

Pada tahap ini, dilakukan serangkaian percobaan pelatihan yang terpisah untuk setiap sub-tugas, yaitu deteksi plat nomor, deteksi karakter, dan klasifikasi karakter. Dalam proses pelatihan model *YOLOv8* untuk deteksi plat nomor sekaligus deteksi karakter plat nomor, serta model *ResNet50* untuk klasifikasi karakter, dilakukan serangkaian eksperimen terhadap sejumlah parameter pelatihan.

1. Deteksi Plat Nomor menggunakan YOLOv8

Deteksi plat nomor kendaraan dilakukan menggunakan arsitektur *YOLOv8* dari *Ultralytics*. *Dataset* yang digunakan terdiri atas citra kendaraan yang telah dianotasi dengan *bounding box* untuk menandai area plat nomor. Agar model mencapai kinerja maksimal, beberapa percobaan pelatihan dijalankan dengan beragam konfigurasi parameter pelatihan yang tertera pada Tabel 3.

Tabel 3 Konfigurasi Parameter Deteksi YOLOv8

Percobaan Ke	Epochs	Batch	Image Size	lr0	lrf	Optimizer
1	100	16	640x640	0.01	0.1	Adam
2	150	16	640x640	0.005	0.1	Adam
3	100	32	640x640	0.01	0.2	SGD
4	150	32	640x640	0.005	0.2	SGD
5	200	16	640x640	0.01	0.1	Adam
6	300	16	640x640	0.03	0.2	Adam
7	200	32	640x640	0.005	0.1	SGD
8	300	32	640x640	0.03	0.2	SGD
9	350	16	640x640	0.01	0.1	Adam
10	400	16	640x640	0.005	0.1	Adam

Dalam tahap ini, dilakukan sepuluh skenario pelatihan model deteksi plat nomor *YOLOv8* dengan resolusi citra 640x640 piksel. Seluruh percobaan ini dirancang untuk mengevaluasi dampak dari berbagai kombinasi hyperparameter, termasuk jumlah *epochs*, *batch size* (16 dan 32), *learning rate initial* (lr0), *learning rate final* (lrf), dan

jenis *optimizer* (Adam atau SGD). Skenario pelatihan bervariasi mulai dari pengujian dengan *batch size* kecil dan besar, penambahan *epoch* untuk mengamati performa pada pelatihan yang lebih lama, hingga penggunaan *learning rate* yang lebih tinggi, dan diakhiri dengan satu percobaan pelatihan intensif selama 400 *epoch*.

2. Deteksi Karakter menggunakan YOLOv8

Pada tahap deteksi karakter plat nomor kendaraan, model yang digunakan masih berbasis YOLOv8, namun dengan penyesuaian pada dataset dan anotasi, yaitu setiap karakter individu pada plat nomor kini dilengkapi dengan bounding box agar sistem bisa mengenali tiap karakter sebagai objek terpisah. Proses pelatihan model itu sendiri berjalan melalui beberapa skenario konfigurasi yang dirangkum dalam Tabel 4.

Tabel 4 Konfigurasi Parameter Deteksi Karakter YOLOv8

Percobaan Ke	Epochs	Batch	Image Size	lr0	lrf	Optimizer
1	100	16	333x333	0.01	0.1	Adam
2	150	32	333x333	0.005	0.2	Adam
3	200	24	333x333	0.02	0.1	SGD
4	300	16	333x333	0.01	0.3	Adam
5	250	32	333x333	0.003	0.1	SGD
6	150	16	333x333	0.01	0.05	SGD
7	200	16	333x333	0.005	0.25	Adam
8	350	32	333x333	0.007	0.15	Adam
9	100	64	333x333	0.02	0.2	SGD
10	400	24	333x333	0.005	0.1	Adam

Pada Tabel 4 menunjukkan sepuluh percobaan pelatihan model deteksi karakter plat nomor dengan kombinasi parameter yang bervariasi untuk mengevaluasi pengaruh parameter terhadap performa model. Seluruh percobaan menggunakan ukuran citra 333x333 piksel yang dipilih karena dimensi karakter pada plat nomor kendaraan relatif kecil. Parameter *epochs* divariasikan dari 100 hingga 400 untuk mengamati pengaruh jumlah iterasi pelatihan terhadap model. Nilai *batch* yang digunakan juga beragam, mulai dari 16, 24, 32, hingga 64, untuk menyeimbangkan antara kestabilan pembelajaran dan efisiensi memori selama pelatihan.

Selanjutnya, nilai *lr0* atau *initial learning rate* disesuaikan dalam rentang 0.003 hingga 0.02, sedangkan *lrf* atau *learning rate final* ditetapkan antara 0.05 hingga 0.25, untuk mengamati seberapa besar pengaruh kecepatan pembelajaran terhadap akurasi dan kestabilan model. Dua jenis algoritma optimasi digunakan, yaitu *Adam* dan *SGD*, guna membandingkan efektivitas terhadap model. Variasi yang dilakukan pada parameter-parameter ini bertujuan untuk memperoleh konfigurasi paling optimal dalam mendeteksi karakter plat nomor secara akurat dan efisien.

3. Klasifikasi Karakter menggunakan ResNet-50

Tabel 5 Konfigurasi Parameter Klasifikasi ResNet-50

Percobaan ke	Batch	Epoch	lr0	Image Size	Buffer	Optimizer
1	64	22	0.01	40x40	1000	Adam
2	16	50	0.1	120x120	1000	Adam
3	16	50	0.005	120x120	1000	SGD
4	32	14	0.02	120x120	1000	Adam

Berdasarkan Tabel 5 terdapat empat percobaan dengan masing-masing memiliki konfigurasi yang berbeda. Pada percobaan pertama, model dilatih dengan *batch size* sebesar 64, jumlah *epoch* sebanyak 22 (mengalami *early stopping*), dan resolusi input gambar sebesar 40×40 piksel. *Optimizer* yang digunakan adalah *Adam*, dan parameter *buffer* ditetapkan sebesar 1000 untuk mengatur jumlah data pada *prefetching* agar efisien. Percobaan kedua menggunakan *batch size* lebih kecil yaitu 16, namun dengan resolusi gambar yang lebih tinggi yaitu 120×120 piksel, jumlah *epoch* sebanyak 50, dan tetap menggunakan *optimizer Adam*.

Pada percobaan ketiga, seluruh konfigurasi pelatihan dijaga sama seperti percobaan kedua, namun *optimizer* diganti ke *SGD* untuk melihat dampak jenis algoritma optimasi terhadap akurasi model. Percobaan keempat menggunakan *batch size* sebesar 32, resolusi gambar tetap 120×120 piksel, dan pelatihan dihentikan lebih awal (*early stopping*) pada epoch ke-14. *Optimizer* yang digunakan kembali ke *Adam*.

B. Analisis Hasil Pelatihan Model

1. Analisis Model Deteksi Plat Nomor menggunakan YOLOv8

Tabel 6 Hasil metrik pelatihan model deteksi plat nomor YOLOv8

Percobaan ke	Precision	Recall	mAP50
1	85%	77%	81%
2	83%	82%	85%
3	97%	55%	81%
4	89%	82%	88%
5	84%	77%	83%
6	83%	75%	79%
7	89%	85%	90%
8	92%	40%	76%
9	82%	81%	84%
10	87%	79%	86%

Berdasarkan analisis hasil pelatihan model deteksi plat nomor, percobaan ketujuh menunjukkan performa terbaik dan paling seimbang. Meskipun percobaan ketiga mencapai *precision* tertinggi sebesar 97%, nilai *recall*-nya sangat rendah (55%), yang berarti model tersebut melewatkan banyak deteksi plat nomor. Sebaliknya, percobaan ketujuh tidak hanya memiliki *recall* tertinggi (85%) tetapi juga *precision* yang tinggi (89%), sehingga menghasilkan nilai mAP50 tertinggi sebesar 90%. Percobaan lain seperti yang kedelapan juga menunjukkan ketidakseimbangan, dengan *precision* tinggi (92%) namun *recall* yang sangat rendah (40%), yang membuktikan bahwa percobaan ketujuh adalah konfigurasi yang paling optimal.

2. Analisis Model Deteksi Karakter menggunakan YOLOv8

Tabel 7 Hasil metrik pelatihan model deteksi karakter YOLOv8

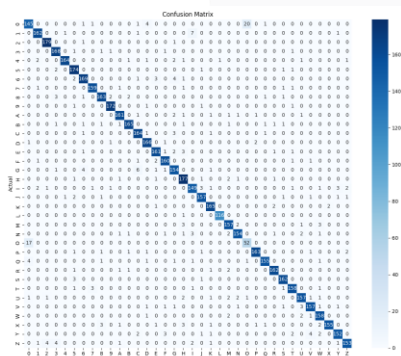
Percobaan ke	Precision	Recall	mAP50
1	80%	86%	84%
2	75%	89%	85%
3	82%	82%	85%
4	79%	89%	85%
5	80%	87%	81%
6	80%	88%	85%
7	81%	88%	86%
8	80%	87%	86%
9	81%	88%	85%
10	81%	89%	86%

Berdasarkan hasil analisis pada 10 percobaan pelatihan model deteksi karakter, performa yang dihasilkan cukup stabil dan baik secara keseluruhan, dengan rentang *precision* antara 75% hingga 82%, *recall* antara 82% hingga 89%, dan mAP50 antara 81% hingga 86%. Meskipun beberapa percobaan seperti kedua, keempat, dan kesepuluh mencapai *recall* tertinggi sebesar 89%, menandakan kemampuan mendeteksi hampir seluruh karakter, nilai *precision* pada percobaan kedua tergolong rendah yaitu 75%. Pada akhirnya, nilai mAP50 tertinggi sebesar 86% dicapai oleh percobaan ketujuh, kedelapan, dan kesepuluh, yang menunjukkan adanya kombinasi paling seimbang antara *precision* dan *recall* untuk performa deteksi karakter yang optimal.

3. Analisis Model Deteksi Karakter menggunakan YOLOv8

Berdasarkan hasil analisis, model ResNet-50 dievaluasi untuk klasifikasi karakter plat nomor melalui tiga percobaan dengan hasil yang bervariasi.

Percobaan pertama mencapai akurasi 94%, namun menunjukkan kelemahan dalam membedakan karakter yang visualnya mirip, seperti 'O' dan '0'. Untuk hasil bisa dilihat pada gambar 10 dan 11

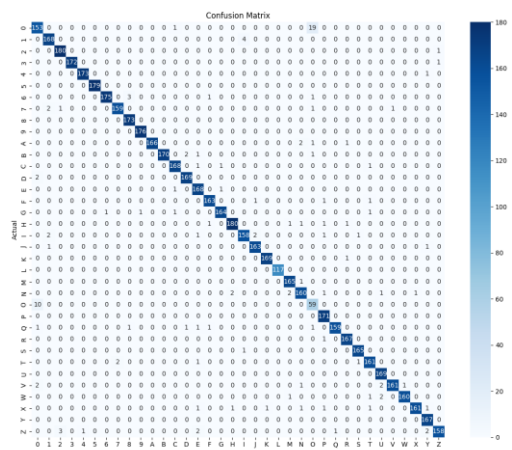


Gambar 10 Confusion Matrik ResNet-50 percobaan 1

label	precision	recall	f1-score	support
0	0.86	0.84	0.85	173.0
1	0.96	0.94	0.95	172.0
2	0.97	0.99	0.98	181.0
3	0.94	0.97	0.95	173.0
4	0.99	0.94	0.96	174.0
5	0.93	0.97	0.95	179.0
6	0.94	0.94	0.94	180.0
7	0.97	0.97	0.97	164.0
8	0.95	0.94	0.95	173.0
9	0.97	0.98	0.97	176.0
A	0.96	0.95	0.95	170.0
B	0.96	0.95	0.96	174.0
C	0.93	0.96	0.94	171.0
D	0.94	0.97	0.95	171.0
E	0.95	0.95	0.95	170.0
F	0.95	0.96	0.96	166.0
G	0.91	0.92	0.91	168.0
H	0.91	0.96	0.93	185.0
I	0.91	0.9	0.91	165.0
J	0.95	0.95	0.95	165.0
K	0.95	0.97	0.96	170.0
L	0.99	0.99	0.99	117.0
M	0.93	0.95	0.94	166.0
N	0.96	0.92	0.94	267.0
O	0.7	0.73	0.73	69.0
P	0.98	0.95	0.96	171.0
Q	0.97	0.94	0.95	165.0
R	0.96	0.96	0.96	168.0
S	0.96	0.98	0.97	166.0
T	0.95	0.96	0.95	165.0
U	0.94	0.93	0.93	169.0
V	0.95	0.94	0.94	167.0
W	0.92	0.96	0.94	164.0
X	0.99	0.92	0.95	168.0
Y	0.96	0.91	0.93	167.0
Z	0.96	0.92	0.94	167.0
accuracy			0.94	17928.94
macro avg	0.94	0.94	0.94	5976.0
weighted avg	0.95	0.94	0.94	5976.0
macro avg	0.94	0.94	0.94	5976.0
weighted avg	0.95	0.94	0.94	5976.0

Gambar 11 Classification report ResNet-50 percobaan 1

Percobaan kedua menunjukkan peningkatan performa yang signifikan, dengan akurasi rata-rata mencapai 98% dan sebagian besar karakter memiliki nilai *F1-score* di atas 95%. Meskipun demikian, karakter 'O' tetap menjadi yang paling sulit untuk diklasifikasikan dengan benar.

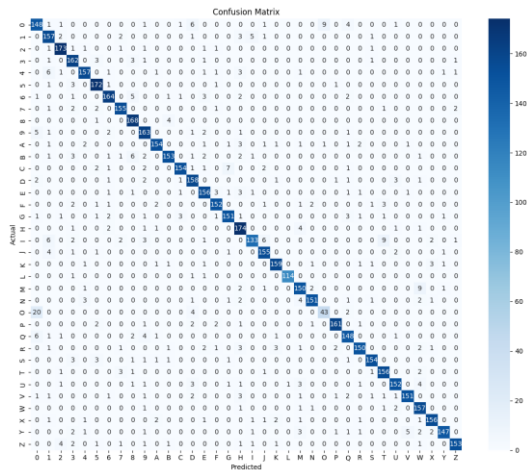


Gambar 12 Confusion Matrik ResNet-50 percobaan 2

label	precision	recall	f1-score	support
0	0.91	0.93	0.92	173.0
1	0.97	0.98	0.97	172.0
2	0.98	0.99	0.99	181.0
3	1.0	0.99	1.0	173.0
4	0.99	0.99	0.99	174.0
5	1.0	1.0	1.0	179.0
6	0.99	0.97	0.98	180.0
7	0.99	0.97	0.98	164.0
8	0.98	1.0	0.99	173.0
9	0.99	1.0	1.0	176.0
A	1.0	0.98	0.99	170.0
B	1.0	0.99	0.99	174.0
C	0.98	0.98	0.98	171.0
D	0.98	0.99	0.99	171.0
E	0.95	0.99	0.97	170.0
F	0.98	0.98	0.98	166.0
G	0.99	0.98	0.98	168.0
H	0.98	0.97	0.98	185.0
I	0.97	0.96	0.96	165.0
J	0.98	0.99	0.98	165.0
K	0.99	0.99	0.99	170.0
L	1.0	1.0	1.0	117.0
M	0.98	0.99	0.99	166.0
N	0.96	0.96	0.96	167.0
O	0.71	0.86	0.78	69.0
P	0.97	1.0	0.98	171.0
Q	0.99	0.96	0.98	165.0
R	0.98	0.99	0.99	168.0
S	0.99	0.99	0.99	166.0
T	0.97	0.98	0.97	165.0
U	0.97	1.0	0.99	169.0
V	0.99	0.96	0.98	167.0
W	0.99	0.98	0.98	164.0
X	0.99	0.96	0.98	168.0
Y	0.97	1.0	0.99	167.0
Z	0.99	0.95	0.97	167.0
accuracy			0.98	17928.98
macro avg	0.97	0.98	0.98	5976.0
weighted avg	0.98	0.98	0.98	5976.0
macro avg	0.97	0.98	0.98	5976.0
weighted avg	0.98	0.98	0.98	5976.0

Gambar 13 Classification report ResNet-50 percobaan 2

Percobaan ketiga mengalami penurunan performa dengan akurasi keseluruhan menjadi 91%, di mana kesalahan klasifikasi pada karakter seperti 'O' dan 'M' semakin terlihat.



Gambar 14 Confusion Matrix ResNet-50 percobaan 3

label	precision	recall	f1-score	support
0	0.8	0.86	0.83	173.0
1	0.85	0.91	0.88	172.0
2	0.94	0.96	0.95	181.0
3	0.87	0.94	0.9	173.0
4	0.94	0.9	0.92	174.0
5	0.91	0.96	0.93	179.0
6	0.92	0.91	0.92	180.0
7	0.93	0.95	0.94	164.0
8	0.89	0.97	0.93	173.0
9	0.87	0.93	0.9	176.0
A	0.94	0.91	0.92	170.0
B	0.94	0.88	0.91	174.0
C	0.86	0.9	0.93	171.0
D	0.87	0.92	0.9	171.0
E	0.9	0.92	0.91	170.0
F	0.93	0.92	0.92	166.0
G	0.92	0.9	0.91	168.0
H	0.84	0.94	0.89	185.0
I	0.93	0.81	0.86	165.0
J	0.91	0.94	0.93	165.0
K	0.95	0.94	0.94	170.0
L	0.99	0.97	0.98	117.0
M	0.88	0.9	0.89	166.0
N	0.86	0.9	0.93	167.0
O	0.81	0.62	0.7	69.0
P	0.96	0.94	0.95	171.0
Q	0.89	0.9	0.89	165.0
R	0.84	0.89	0.86	168.0
S	0.95	0.93	0.94	166.0
T	0.91	0.95	0.93	165.0
U	0.93	0.9	0.92	169.0
V	0.96	0.9	0.93	167.0
W	0.86	0.96	0.9	164.0
X	0.95	0.93	0.94	168.0
Y	0.97	0.88	0.92	167.0
Z	0.97	0.92	0.94	167.0
accuracy	0.91	0.91	0.91	17928.91
macro avg	0.92	0.91	0.91	5976.0
weighted avg	0.91	0.91	0.91	5976.0
macro avg	0.91	0.91	0.91	5976.0
weighted avg	0.92	0.91	0.91	5976.0

Gambar 15 Classification report ResNet-50 percobaan 3

Dari ketiga percobaan tersebut, model pada percobaan kedua terbukti menjadi yang terbaik dengan akurasi 98%, yang mengindikasikan bahwa konfigurasi pada percobaan tersebut paling optimal untuk tugas klasifikasi karakter plat nomor.

C. Hasil Kinerja Sistem

1. Hasil Kinerja IoT

Sistem IoT yang dikembangkan dalam penelitian ini memanfaatkan Raspberry Pi sebagai pusat kendali untuk mengelola proses input dari perangkat RFID reader, kamera, serta kontrol motor stepper. RFID digunakan sebagai media verifikasi identitas mahasiswa dengan memanfaatkan KTM yang sudah dilengkapi dengan chip RFID.

2. Hasil Kinerja Pembacaan RFID

Tabel 8 Hasil Pengujian Pembacaan RFID Tag

No	Jenis Kartu	ID Kartu	Jarak Tap Kartu	Status
1	RFID Card Mifare	D5FBCC72	1 – 2 CM	Terbaca
2	RFID Card Mifare	E374D05	1 – 2 CM	Terbaca
3	RFID Card Mifare	D09A46A4	1 – 2 CM	Terbaca
4	Kartu Tanda Mahasiswa	CCF53E2B	1 – 2 CM	Terbaca

No	Jenis Kartu	ID Kartu	Jarak Tap Kartu	Status
5	Kartu Tanda Mahasiswa	7C603D2B	1 – 2 CM	Terbaca

Pengujian sistem pembacaan RFID, seperti yang ditunjukkan pada Tabel 8, mencapai tingkat keberhasilan 100%. Sistem mampu membaca berbagai jenis kartu, termasuk *RFID Card Mifare* dan Kartu Tanda Mahasiswa, secara konsisten pada jarak optimal 1-2 cm dengan waktu respons rata-rata di bawah 300 ms. Selain itu, sistem memberikan respons audio yang jelas satu *beep* untuk RFID yang tidak terdaftar dan dua *beep* untuk RFID valid yang datanya berhasil dikirim ke server API.

3. Koneksi dan Integrasi dengan API

Tabel 9 Hasil Pengujian API Backend

No	Jenis API	Data	Response Time	Status
1	Registrasi Tag RFID	Kode RFID	125 ms	Sukses
2	Verifikasi Tag RFID	Kode RFID	119 ms	Sukses
3	Aktivitas Pengguna	ID Mahasiswa, Plat Nomor	157 ms	Sukses

Raspberry Pi terhubung dengan sistem backend melalui protokol HTTP menggunakan API. Pengujian dilakukan dengan beberapa permintaan data seperti pada Tabel 5.12. Seluruh komunikasi berlangsung secara *real-time* dengan latensi rata-rata di bawah 200 ms. Sistem mampu menyimpan data aktivitas kendaraan keluar-masuk secara otomatis ke database dan menampilkannya pada dashboard website untuk petugas parkir.

4. Hasil Kinerja Model Deteksi Plat Nomor (YOLOv8)

Pengujian model deteksi plat nomor dilakukan dengan berbagai variasi kondisi, seperti pencahayaan yang rendah, sudut miring, dan posisi plat nomor yang tidak sejajar dengan kamera. Pengujian menggunakan model YOLOv8 terbaik berdasarkan nilai precision, recall, mAP50 dari hasil pelatihan

No	Citra	Plat Nomor Aktual	Inference Time
1		L5053CAB	261 ms
2		AG6681YBE	243 ms
3		W6225DZ	253 ms

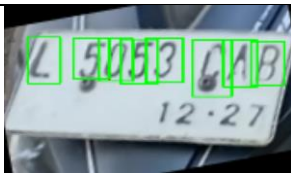
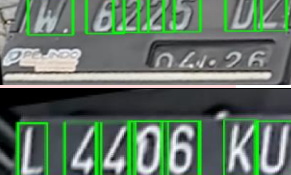
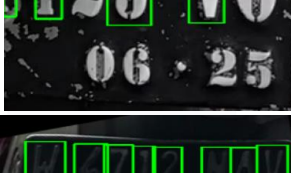
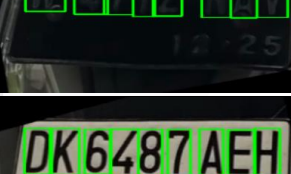
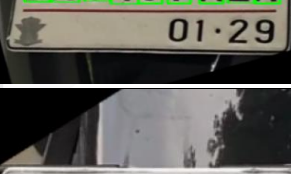
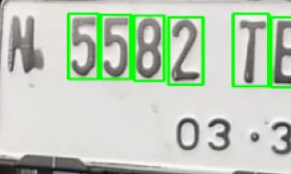
No	Citra	Plat Nomor Aktual	Inference Time
4		L4406KU	254 ms
5		L6124AAU	249 ms
6		S4425VO	266 ms
7		W4712NAV	244 ms
8		DK6487AEH	253 ms
9		N5582TEZ	248 ms

Berdasarkan pengujian, model deteksi plat nomor YOLOv8 menunjukkan kinerja yang stabil dalam mengenali plat nomor di berbagai kondisi. Model ini berhasil mendeteksi plat nomor dengan presisi tinggi, bahkan saat plat berada pada posisi miring atau tidak sejajar dengan kamera. Meskipun demikian, ditemukan beberapa kelemahan, seperti pada sampel plat nomor yang rusak di mana deteksi menjadi tidak optimal dan *bounding box* yang dihasilkan tidak lengkap. Selain itu, pada kondisi pencahayaan gelap, terjadi kesalahan deteksi (*False Positive*) di mana model juga menandai bagian bodi motor sebagai plat nomor.

5. Hasil Kinerja Model Deteksi Karakter (YOLOv8)

Pengujian model deteksi karakter dilakukan menggunakan data citra hasil deteksi plat nomor dengan menambahkan *preprocessing* berupa rotasi gambar berdasarkan garis tepi plat nomor secara horizontal. Berikut adalah beberapa hasil deteksi karakter plat nomor pada Tabel

Tabel 10 Hasil pengujian segmentasi karakter plat nomor

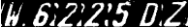
No	Citra	Plat Nomor Aktual	Inference Time
1		L5053CAB	209 ms
2		AG6681YBE	191 ms
3		W6225DZ	212 ms
4		L4406KU	210 ms
5		L6124AAU	233 ms
6		S4425VO	200 ms
7		W4712NAV	223 ms
8		DK6487AEH	259 ms
9		N5582TEZ	211 ms

Berdasarkan pengujian deteksi karakter menggunakan model YOLOv8, hasilnya menunjukkan performa yang stabil pada berbagai bentuk plat nomor. Model berhasil mendeteksi seluruh karakter dengan baik pada sampel citra pertama hingga kelima, serta pada sampel ketujuh dan kedelapan. Namun, tantangan muncul pada plat nomor dengan gaya

huruf yang terlalu dimodifikasi, seperti pada sampel keenam di mana karakter '2' dan 'O' tidak terdeteksi. Selain itu, *noise* pada karakter juga menjadi masalah, seperti pada sampel kesembilan di mana karakter 'N' gagal terdeteksi.

6. Hasil Klasifikasi Karakter (ResNet-50)

Tabel 11 Hasil pengujian pengenalan karakter plat nomor

No	Citra Karakter	Aktual	Prediksi	Inferenc e Time tiap Karakte r
1		L5053CAB	L5053CAB	201 ms
2		AG6681YB E	AG6681YB E	197 ms
3		W6225DZ	W6225DZ	215 ms
4		L4406KU	L4406KU	219 ms
5		L6124AAU	L6T24AAU	220 ms
6		S4425VO	LM5V	208 ms
7		W41712NA V	M4712NAV	214 ms
8		DK6487AE H	DK6487AE H	195 ms
9		N5582TEZ	5582TEZ	219 ms

Berdasarkan hasil pengujian klasifikasi karakter menggunakan ResNet-50, model mampu mengklasifikasikan sebagian besar karakter dengan benar, terutama ketika kualitas citra baik dan karakter terlihat jelas. Hal ini dibuktikan dengan empat dari sembilan sampel (pengujian kedua, ketiga, keempat, dan kedelapan) yang berhasil diklasifikasikan dengan akurasi 100%. Namun, terjadi beberapa kesalahan pada sampel lainnya yang disebabkan oleh kemiripan bentuk visual antar karakter, seperti 'I' yang salah dikenali sebagai 'T' dan 'W' sebagai 'M'. Selain itu, pada sampel keenam, beberapa karakter bahkan tidak terdeteksi oleh model YOLOv8 sebelumnya, dan terjadi kesalahan klasifikasi angka '4' sebagai 'M'. Kesalahan-kesalahan ini menunjukkan bahwa performa model dapat dipengaruhi oleh plat nomor yang dimodifikasi dan kualitas citra yang buruk.

7. Hasil Kinerja Website Monitoring Keluar Masuk Tempat Parkir

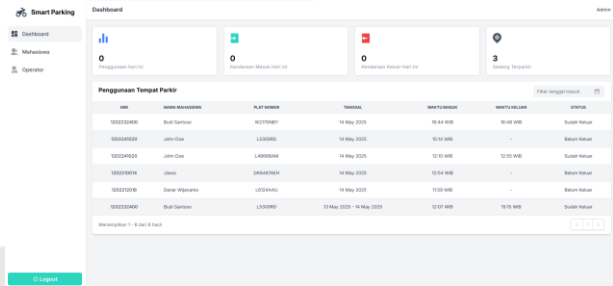
Website monitoring yang dikembangkan menampilkan informasi riwayat keluar-masuk kendaraan berdasarkan hasil pembacaan RFID dan pengenalan plat nomor, serta terdapat operator dan mahasiswa yang terdaftar di dalam sistem ini. Pengujian website menggunakan metode *black box testing* dengan fitur yahong diuji seperti pada Tabel

Tabel 12 Pengujian Fitur Website

No	Fitur	Input	Expected Output	Hasil
1	View Log	Klik menu log	Data mahasiswa & plat nomor tampil	Passed
2	View Log (tanpa refresh)	Kendaraan masuk, tidak refresh halaman	Log tidak bertambah otomatis	Passed

3	View Log (dengan refresh)	Refresh halaman	Log terbaru tampil	Passed
4	Register Mahasiswa	Nama: Lala, NIM: 1202290012, RFID: DF6BGDF	Data tersimpan & tampil	Passed
5	Register Mahasiswa	Nama kosong, NIM & RFID terisi	Validasi: "Name is required"	Passed
6	Register Mahasiswa	NIM kosong	Validasi: "NIM is required"	Passed
7	Register Mahasiswa	RFID kosong	Validasi: "RFID code is required"	Passed
8	Tambah Admin	Nama: Admin1, Username: admin1, Pass: 123456	Data admin tersimpan	Passed
9	Tambah Admin	Nama kosong	Validasi: "Name is required"	Passed
10	Login Admin	Username: admin1, Password: 123456	Berhasil login ke dashboard	Passed
11	Login Admin	Password salah	Pesan error: "Incorrect username or password"	Passed

Hasil menunjukkan bahwa sistem *website monitoring* telah memenuhi kebutuhan operasional dasar dalam mengelola data mahasiswa dan mencatat aktivitas kendaraan, serta kemampuan penanganan eror yang efektif dan performa yang stabil di seluruh fungsi utamanya. Untuk tampilan dari dashboard websitenya dapat dilihat pada gambar 16.



Gambar 16 Tampilan Website Dashboard

V. KESIMPULAN

Sistem parkir otomatis ini telah berhasil dirancang dengan model YOLOv8 untuk deteksi plat nomor dan karakter, yang masing-masing mencapai nilai mAP50 sebesar 88% dan 85%. Untuk klasifikasi karakter, model ResNet50 mampu mencapai akurasi rata-rata 98%, meskipun masih terdapat kekeliruan pada karakter dengan bentuk yang mirip seperti '0' dan 'O'. Keseluruhan sistem yang diimplementasikan pada Raspberry Pi ini mampu bekerja secara efisien dan stabil, dengan total waktu pemrosesan dari pembacaan RFID hingga pengiriman data ke dasbor dan pengendalian motor stepper memakan waktu sekitar 3 hingga 4 detik.

REFERENSI

[1] Y. Galahartlambang, T. Khotiah, Z. Fanani, A. Aprilia Yani Solekhah, P. Studi Teknologi Informasi, and I. K. Teknologi dan Bisnis Ahmad Dahlan Lamongan Jl Ahmad Dahlan No, "DETEKSI PLAT

NOMOR KENDARAAN OTOMATIS DENGAN CONVOLUTIONAL NEURAL NETWORK DAN OCR PADA TEMPAT PARKIR ITB AHMAD DAHLAN LAMONGAN,” 2023, doi: 10.36595/misi.v5i2.

- [2] A. W. Hamdani and A. Prapanca, “Sistem Deteksi Plat Kendaraan pada Parkiran Rumah Pribadi dengan Metode Background Subtraction dan Optical Character Recognition,” *Journal of Informatics and Computer Science*, vol. 03, 2022.
- [3] D. Fernandes, H. Sunardi, and Zulkifli, “Implementasi Deteksi Plat Nomor Kendaraan Bermotor Roda Dua Berbasis OpenCV Untuk Keamanan Parkir,” 2023.
- [4] L. Satya, M. R. D. Septian, M. W. Sarjono, M. Cahyanti, and E. R. Swedia, “SISTEM PENDETEKSI PLAT NOMOR POLISI KENDARAAN DENGAN ARSITEKTUR YOLOV8,” *Sebatik*, vol. 27, no. 2, pp. 753–761, Dec. 2023, doi: 10.46984/sebatik.v27i2.2374.
- [5] H. Oktavia, M. Elisabeth Taena, S. Raudhatul Syifa, and P. Rosyani, “BIN: Bulletin Of Informatics Implementasi Sistem Pendeteksian Plat Nomor Otomatis Menggunakan Metode Pengolahan Citra dengan OpenCV dan EasyOCR,” 2024. [Online]. Available: <https://ojs.jurnalmahasiswa.com/ojs/index.php/bin>
- [6] G. Ramadhan, R. D. A. Khairiyah, S. Natania, and A. Harris, “Vehicle Police Number Detection Using YOLOv8,” *Media Journal of General Computer Science*, vol. 1, no. 2, pp. 62–70, Jun. 2024, doi: 10.62205/mjgcs.v1i2.25.
- [7] F. M. Adhari, T. F. Abidin, and R. Ferdhiana, “License Plate Character Recognition using Convolutional Neural Network,” *Journal of Information Systems Engineering and Business Intelligence*, vol. 8, no. 1, pp. 51–60, Apr. 2022, doi: 10.20473/jisebi.8.1.51-60.
- [8] D. Nafis Alfarizi, R. Agung Pangestu, D. Aditya, M. Adi Setiawan, and P. Rosyani, “Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis,” 2023. [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [9] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, “UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios,” *Sensors*, vol. 23, no. 16, Aug. 2023, doi: 10.3390/s23167190.
- [10] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS,” Dec. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/make5040083.
- [11] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, “A Survey of CNN-Based Network Intrusion Detection,” Aug. 01, 2022, *MDPI*. doi: 10.3390/app12168162.
- [12] D. Sarwinda, R. H. Paradisa, A. Bustamama, and P. Anggia, “Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer,” *ScienceDirect*, 2021.
- [13] J. Liang, “Image classification based on RESNET,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Oct. 2020. doi: 10.1088/1742-6596/1634/1/012110.
- [14] B. Mandal, A. Okeukwu, and Y. Theis, “Masked Face Recognition using ResNet-50,” Apr. 2021, [Online]. Available: <http://arxiv.org/abs/2104.08997>
- [15] N. Basuni and Amril Mutoi Siregar, “Comparison of the Accuracy of Drug User Classification Models Using Machine Learning Methods,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1348–1353, Dec. 2023, doi: 10.29207/resti.v7i6.5401.
- [16] M. Yoga Wibowo, H. Hikmayanti, A. Fitri Nur Masruriyah, E. Novalia, and N. Heryana, “Mask Use Detection in Public Places Using the Convolutional Neural Network Algorithm,” 2023.
- [17] U. Ali, M. A. Ismail, R. A. Ariyaluran Habeeb, and S. R. Ali Shah, “Performance Evaluation of YOLO Models in Plant Disease Detection,” *Journal of Informatics and Web Engineering*, vol. 3, no. 2, pp. 199–211, Jun. 2024, doi: 10.33093/jiwe.2024.3.2.15.
- [18] N. Vincent and A. B. Primawan, *ID: 16 Sistem Informasi Parkir Pintar berbasis Web dan IoT Web and IoT-based Smart Parking Information System*. 2020.
- [19] M. Wijayanti, “PROTOTYPE SMART HOME DENGAN NODEMCU ESP8266 BERBASIS IOT,” *JUIT*, vol. 1, no. 2, 2022.
- [20] M. M. Islam, A. Rahaman, and M. R. Islam, “Development of Smart Healthcare Monitoring System in IoT Environment,” *SN Comput Sci*, vol. 1, no. 3, May 2020, doi: 10.1007/s42979-020-00195-y.
- [21] S. Gunawan Zain, “Sistem Object Recognition Plat Nomor Kendaraan Untuk Sistem Parkir Bandara,” 2022. [Online]. Available: <https://ojs.unm.ac.id/JESSI/index>
- [22] N. Made, D. Febriyanti, A. A. Kompiani, O. Sudana, and N. Piarsa, “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen,” 2021.