

PEMANFAATAN METODE *CANNY EDGE DETECTION* UNTUK PENGENALAN KARAKTER TANDA NOMOR KENDARAAN BERMOTOR (TNKB)

Dina Ayuni^{1*}, Andre Rabiula², Haerul Pathoni³

^{1,2,3}Universitas Jambi; Jl. Jambi-Muara Bulian KM. 15, Mendalo Indah, Kecamatan Jambi Luar Kota, Kabupaten Muaro Jambi, Jambi; Telp. 0741-583122

Keywords:

Deteksi TNKB;
Canny Edge Detection;
YoloV8;
KerasOCR

Correspondent Email:

dinaayuni0808@gmail.com

Abstrak. Tanda Nomor Kendaraan Bermotor (TNKB) merupakan identitas penting kendaraan dalam sistem parkir otomatis dan keamanan. Namun, sistem parkir di Indonesia masih banyak menggunakan verifikasi manual yang kurang efisien dan rawan kesalahan. Penelitian ini mengembangkan sistem otomatis untuk mendeteksi dan mengenali karakter TNKB menggunakan kombinasi YOLOv8, *Canny Edge Detection*, dan Keras-OCR berbasis bahasa pemrograman Python dengan metode ADDIE. Hasil pengujian menunjukkan sistem bekerja optimal pada jarak 40–50 cm dengan akurasi hingga 100%, namun menurun menjadi 70–75% pada jarak di atas 60 cm. Pengujian realtime memperlihatkan performa terbaik saat kendaraan berhenti selama 10–30 detik, sedangkan gerakan cepat menyebabkan penurunan akurasi akibat motion blur. Pada kondisi pencahayaan terang dan normal, akurasi mencapai sekitar 85%, sedangkan pada kondisi gelap turun menjadi 75%. Berdasarkan *confusion matrix*, sistem memperoleh akurasi 83,33% dengan recall 100%. Sistem ini efektif pada jarak dekat dan pencahayaan cukup, namun masih terbatas pada kondisi cahaya rendah dan jarak jauh.



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. *Vehicle License Plate (TNKB) serves as an essential vehicle identity in automated parking systems and security applications. However, parking systems in Indonesia still largely rely on manual verification, which is inefficient and prone to errors. This study develops an automatic system for detecting and recognizing license plate characters using a combination of YOLOv8, Canny Edge Detection, and Keras-OCR, implemented in Python and developed using the ADDIE method. Experimental results show that the system performs optimally at a distance of 40–50 cm with an accuracy of up to 100%, but accuracy decreases to 70–75% at distances above 60 cm. Real-time testing indicates that the system achieves the best performance when vehicles remain stationary for the 10–30 seconds, while rapid movement causes accuracy degradation due to motion blur. Under bright and normal lighting conditions, the system achieves approximately 85% accuracy, whereas accuracy drops to 75% in low-light conditions. Based on the confusion matrix, the system attains an overall accuracy of 83.33% with a recall of 100%. Overall, the system is effective at short distances and adequate lighting, but limitations remain under low-light conditions and longer detection distances.*

1. PENDAHULUAN

Tanda Nomor Kendaraan Bermotor (TNKB) memainkan fungsi vital di berbagai bidang penerapan, termasuk mekanisme parkir otomatis, pengendalian arus lalu lintas, serta

sejumlah aplikasi lainnya. Khusus dalam skema parkir, TNKB bertindak sebagai penanda identitas, yang memastikan bahwa hanya kendaraan yang telah tercatat yang diperbolehkan masuk atau keluar dari zona

parkir, sehingga membantu menekan potensi tindak kriminal. Meskipun begitu, di Indonesia, prosedur keamanan parkir masih sangat mengandalkan pemeriksaan manual oleh staf, baik ketika kendaraan tiba maupun saat meninggalkan lokasi. Pendekatan ini terbukti tidak optimal, khususnya dalam situasi di mana jumlah kendaraan tinggi dan tenaga manusia yang tersedia sangat minim.[1]

Pengenalan Tanda Nomor Kendaraan Bermotor (TNKB) merupakan elemen krusial dalam pembangunan sistem transportasi cerdas. Selanjutnya, ide dan mekanisme pengenalan ini akan memberikan dampak positif yang besar di berbagai bidang aplikasi, seperti pengawasan kendaraan ketika terjadi insiden kriminal.[2]

Salah satu tahap penting dalam sistem pengenalan TNKB otomatis adalah proses pengenalan pola pada plat nomor kendaraan. Teknologi pengenalan TNKB merupakan bagian dari pengembangan perangkat lunak komputer yang memungkinkan sistem membaca dan mengidentifikasi Tanda Nomor Kendaraan Bermotor secara otomatis dari citra atau gambar digital.[3]

Pentingnya identifikasi dan pengenalan Tanda Nomor Kendaraan Bermotor (TNKB) diperhatikan mengingat meningkatnya pelanggaran lalu lintas.[4] dan kasus pencurian kendaraan bermotor.[5]

Pengolahan citra digital adalah metode pemrosesan citra statis maupun dinamis yang bertujuan untuk meningkatkan kualitas visual agar dapat dianalisis secara optimal oleh manusia maupun sistem komputer. Teknik ini menawarkan berbagai keunggulan, antara lain efisiensi waktu, tingkat ketelitian yang tinggi, peningkatan keamanan dalam proses evaluasi, serta kemampuan menampilkan representasi objek secara jelas dan akurat.[6]

Salah satu solusi untuk menggantikan proses pencatatan nomor kendaraan secara manual adalah penerapan sistem pengenalan plat nomor kendaraan secara otomatis berbasis pengolahan citra digital. Beragam metode telah dikembangkan untuk meningkatkan ketepatan pengenalan karakter pada plat nomor. Meskipun demikian, proses tersebut masih menghadapi berbagai kendala, seperti kemiripan bentuk antar karakter dan adanya gangguan noise saat akuisisi citra, yang dapat menurunkan tingkat akurasi pengenalan.[7]

Penelitian sebelumnya telah membahas mengenai sistem deteksi kendaraan bermotor. Implementasi Canny Filter Optical Character Recognition (OCR) untuk Identifikasi Tanda Nomor Kendaraan Bermotor (TNKB) [8]. Berdasarkan permasalahan tersebut, penelitian ini mengusulkan integrasi *Canny Edge Detection*, YOLOv8, dan Keras-OCR untuk meningkatkan performa sistem pengenalan TNKB.

2. TINJAUAN PUSTAKA

2.1. *Canny Edge Detection*

Canny Edge Detection merupakan metode pendeteksian tepi yang menerapkan serangkaian tahapan algoritma untuk mengekstraksi dan mengidentifikasi berbagai jenis tepi pada suatu citra. Metode ini pertama kali dikembangkan oleh John F. Canny pada tahun 1986 dan hingga kini banyak digunakan dalam bidang pengolahan citra digital.[9]

Algoritma Canny menggunakan teknik ambang batas ganda (*double thresholding*) setelah proses *non-maximum suppression* untuk menentukan keberadaan titik tepi. Piksel dengan nilai gradien yang melebihi ambang batas atas diklasifikasikan sebagai tepi yang kuat, sedangkan piksel dengan nilai gradien di bawah ambang batas bawah dianggap bukan tepi. Sementara itu, piksel dengan nilai gradien di antara kedua ambang tersebut diperlakukan sebagai kandidat tepi. Kandidat tepi yang memiliki keterhubungan dengan tepi kuat akan dipertahankan sebagai tepi akhir. Pendekatan ini dirancang untuk meminimalkan pengaruh noise sehingga menghasilkan deteksi tepi yang lebih akurat.[10]

Deteksi tepi Canny mampu menghasilkan tepi yang lebih tajam dan lebih sempurna jika dibandingkan dengan operator deteksi tepi lainnya seperti *Sobel*, *Prewitt*, dan *Robert*. Metode ini menggunakan *Kernel Gaussian Derivatif* untuk menghilangkan *noise* dari gambar sehingga menghasilkan deteksi tepi yang halus. Terdapat dua ambang yang digunakan dalam metode *Canny*, yaitu ambang rendah dan ambang tinggi. Metode *Canny* menggunakan algoritma dengan beberapa tahap untuk mendeteksi tepi pada gambar. Algoritma ini memiliki tingkat kesalahan yang rendah, mampu memetakan titik tepi (jarak antara piksel tepi) dengan sangat dekat dengan

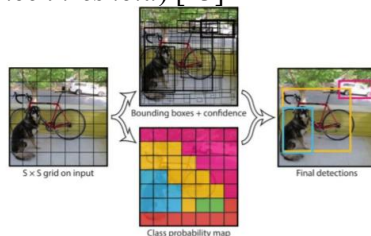
tepi sebenarnya, dan hanya memberikan satu respons untuk setiap tepi yang terdeteksi.[11]

2.2. Optical Character Recognition (OCR)

OCR (*Optical Character Recognition*) merupakan teknik dalam ranah pengolahan citra dan visi komputer yang digunakan untuk mengonversi representasi visual berupa huruf dan angka menjadi bentuk karakter digital yang dapat dikenali oleh sistem. Teknologi ini berperan dalam meningkatkan kemampuan, efisiensi, serta tingkat kecerdasan sistem komputer dalam melakukan proses identifikasi dan pengolahan informasi berbasis teks.[12]

2.3. You Only Look Once (YOLO)

YOLO (*You Only Look Once*) merupakan salah satu model berbasis Convolutional Neural Network (CNN) yang dirancang untuk melakukan deteksi objek secara cepat dan efisien. Metode ini bekerja dengan melakukan penyesuaian ukuran citra masukan, kemudian memprosesnya melalui satu jaringan konvolusi terpadu, serta menentukan hasil deteksi objek berdasarkan nilai ambang kepercayaan (*confidence threshold*).[13]



Gambar 1. Model Algoritma YOLO

2.4. Bahasa Pemrograman Python

Python sering digunakan sebagai bahasa pemrograman dalam peran scripting. Python kerap disebut sebagai object-oriented scripting language karena mengintegrasikan dukungan untuk pemrograman berorientasi objek dengan orientasi utama pada tugas scripting. Jika harus dirangkum, Python dapat digambarkan sebagai bahasa pemrograman umum yang menggabungkan paradigma prosedural, fungsional, dan berorientasi objek.[14]

2.5. Confussion Matrix

Confusion matrix atau matriks konfusi merupakan sebuah tabel yang digunakan untuk mencatat hasil proses klasifikasi, sehingga dapat menggambarkan kemampuan suatu sistem dalam memprediksi kelas atau kategori

data. Matriks ini umum dimanfaatkan sebagai alat evaluasi kinerja model klasifikasi dengan menyajikan perbandingan antara hasil prediksi dan kondisi sebenarnya. Melalui matriks konfusi, dapat diketahui sejauh mana sistem mampu mengklasifikasikan data secara tepat ke dalam kelas yang sesuai. Oleh karena itu, matriks konfusi memberikan informasi yang komprehensif mengenai tingkat akurasi, presisi, dan efektivitas sistem klasifikasi dalam melakukan prediksi.[15]

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (1)$$

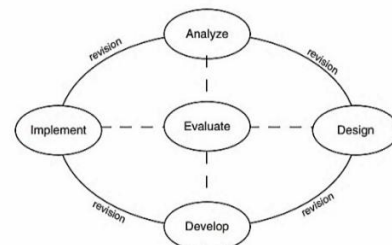
$$Precision = \frac{TP}{TP+FP} \quad (2)$$

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

$$F1-Score = \frac{2TP}{2TP+FN+FP} \quad (4)$$

3. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian ADDIE, yaitu *Analysis, Design, Development, Implementation, dan Evaluation*. Metode ADDIE dipilih karena setiap tahapannya melibatkan revisi untuk memastikan pembuatan sistem yang diminimalisir kemungkinan kegagalan. Penelitian ini dilakukan bertujuan untuk meningkatkan keamanan suatu kendaraan serta menerapkan sistem parkir cerdas.



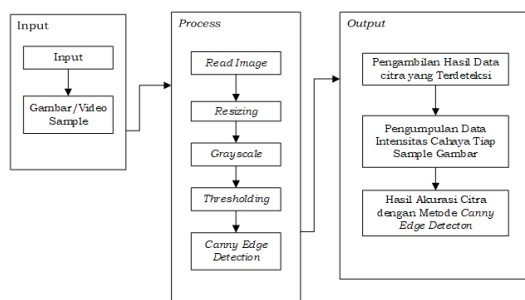
Gambar 2. Metode Penelitian ADDIE

3.1. Analisis

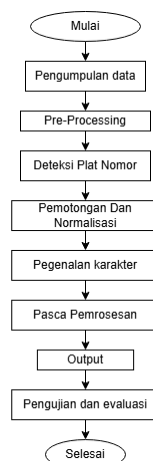
Tahap analisis dilakukan untuk mengkaji permasalahan serta menentukan kebutuhan sistem yang akan dikembangkan. Permasalahan utama yang dihadapi adalah proses identifikasi TNKB di area parkir yang masih dilakukan secara manual sehingga kurang efektif dan efisien. Adapun kebutuhan sistem mencakup perangkat keras berupa laptop dan kamera webcam, serta perangkat lunak berbasis Python yang didukung oleh metode YOLOv8, *Canny Edge Detection*, dan Keras-OCR untuk proses deteksi dan pengenalan karakter.

3.2. Desain (Design)

Pada tahap ini dirancang alur sistem deteksi TNKB yang mencakup akuisisi citra, pra-pemrosesan, deteksi area plat nomor, pengenalan karakter, dan evaluasi hasil. Desain sistem divisualisasikan dalam diagram blok dan *flowchart* untuk menggambarkan alur proses secara keseluruhan. Kamera ditempatkan secara statis dengan jarak tertentu untuk memperoleh citra TNKB yang optimal.



Gambar 3. Diagram Blok Sistem



Gambar 4. Flowchart Pengolahan Citra

3.3. Pengembangan (Development)

Tahap pengembangan dilakukan dengan mengimplementasikan metode *Canny Edge Detection* untuk mengekstraksi tepi karakter TNKB, YOLOv8 untuk mendeteksi lokasi plat nomor, serta Keras-OCR untuk pengenalan karakter. Sistem dikembangkan menggunakan bahasa pemrograman *Python* dan diuji pada berbagai jarak serta kondisi pencahayaan.

3.4. Implementasi (Implementation)

Implementasi dilakukan dengan merealisasikan desain sistem menggunakan webcam yang terhubung ke laptop. Sistem

dijalankan secara realtime untuk mendeteksi dan mengenali karakter TNKB sesuai skenario pengujian yang telah dirancang.

3.5. Pengujian

3.5.1. Pengujian Webcam

Pengujian dilakukan untuk mengetahui kualitas citra yang dihasilkan webcam pada berbagai jarak pengambilan gambar.

3.5.2. Pengujian Jarak

Pengujian jarak bertujuan menentukan rentang optimal kamera dalam mendeteksi dan mengenali karakter TNKB secara akurat.

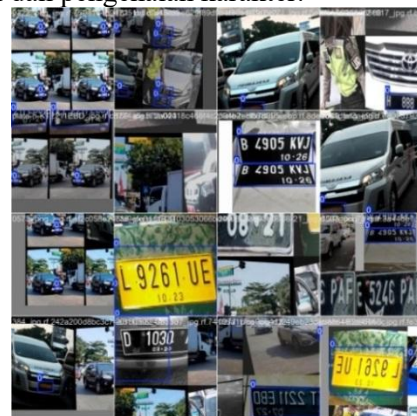
3.5.3. Pengujian Realtime

Pengujian *realtime* dilakukan untuk mengevaluasi kinerja sistem dalam kondisi mendekati penggunaan nyata di lapangan, termasuk variasi jarak dan pencahayaan.

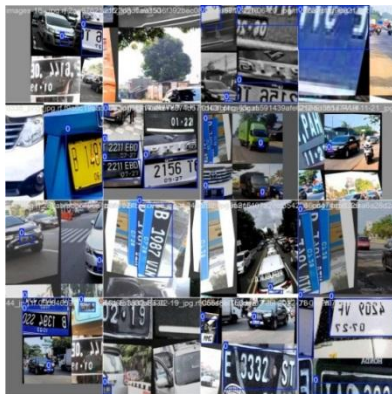
4. HASIL DAN PEMBAHASAN

4.1. Data dan Pra-Pemrosesan Citra

Data yang digunakan pada tahap pelatihan dan pengujian berupa citra kendaraan bermotor dengan variasi jenis kendaraan, jarak perekaman, posisi plat, serta kondisi pencahayaan yang berbeda. Keanekaragaman data ini bertujuan untuk meningkatkan kemampuan generalisasi model YOLOv8 dalam mendeteksi area Tanda Nomor Kendaraan Bermotor (TNKB). Sebagian citra juga diproses menggunakan metode *Canny Edge Detection* untuk menegaskan batas tepi karakter sehingga mempermudah proses anotasi dan pengenalan karakter.



Gambar 5. Data Latih Plat



Gambar 6. Data Latih Plat

Tahapan pra-pemrosesan meliputi penyesuaian ukuran citra (*resizing*), konversi citra ke skala abu-abu (*grayscale*), serta penerapan *Canny Edge Detection* untuk menonjolkan kontur karakter TNKB. Proses ini dilakukan guna meminimalkan noise dan meningkatkan kejelasan struktur karakter sebelum citra diproses lebih lanjut pada tahap *Optical Character Recognition (OCR)*.

4.2. Implementasi Sistem Deteksi TNKB

Sistem pendeteksian TNKB dikembangkan dengan memanfaatkan YOLOv8 untuk mengidentifikasi posisi plat nomor, metode *Canny Edge Detection* untuk memperjelas kontur karakter, serta Keras-OCR sebagai modul pengenalan karakter. Sistem ini dirancang untuk bekerja secara *realtime* dengan memanfaatkan input citra dari *webcam*.

Setelah YOLOv8 berhasil mendeteksi area plat nomor, bagian citra tersebut diekstraksi sebagai *Region of Interest (ROI)* dan selanjutnya diproses oleh OCR. Hasil pengenalan karakter kemudian diseleksi berdasarkan ukuran dan posisi karakter guna meminimalkan kesalahan akibat *noise*. Selain itu, sistem juga melakukan pengelompokan kondisi pencahayaan menjadi terang, normal, dan gelap berdasarkan nilai rata-rata intensitas piksel untuk menghindari pemrosesan citra dengan kualitas yang tidak memadai.

4.3. Pengujian Jarak Deteksi TNKB

Pengujian sistem dilakukan menggunakan 10 sampel TNKB dengan variasi jarak pengambilan citra antara 40 hingga 85 cm.

Tabel 1. Hasil Pengujian Jarak TNKB

No	Gambar TNKB	Jarak (cm)	Karakter Yang Terbaca	Akurasi
1.		50cm	BH 5021 ZJ	100%
2.		40 cm	BH 6242 LL	85%
3.		40 cm	BH 4675 OP	100%
4.		45cm	BH 3097 ZM	100%

No	Gambar TNKB	Jarak (cm)	Karakter Yang Terbaca	Akurasi
5.		50cm	BH 1707 IZ	90%
6.		80cm	3023 GD	70%
7.		40cm	BH 6079 BI	100%
8.		40cm	BH 3351 NU	100%
9.		85cm	BH 2780	70%
10.		70cm	BH 2954	75%

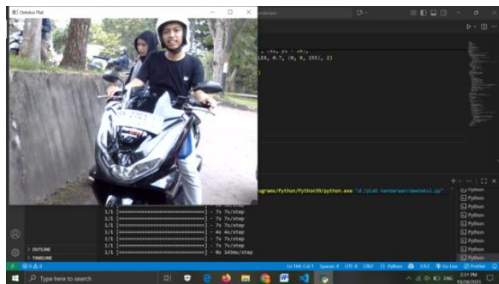
Hasil evaluasi menunjukkan bahwa tingkat ketepatan tertinggi, yaitu mencapai 100%, diperoleh pada rentang jarak 40–50 cm karena karakter plat masih terlihat jelas dan memiliki ketajaman yang baik. Namun, ketika jarak pengambilan melebihi 60 cm, tingkat akurasi mengalami penurunan hingga berada pada kisaran 70–75%, yang disebabkan oleh berkurangnya detail visual karakter serta meningkatnya gangguan *noise* pada citra.

Hasil ini sejalan dengan prinsip pengolahan citra digital, di mana resolusi objek dan rasio

ukuran karakter terhadap sensor kamera sangat memengaruhi keberhasilan OCR.

4.4. Pengujian *Realtime* dan Pengaruh Pencahayaan

Pengujian *realtime* dilakukan dengan webcam yang dipasang pada tripod dengan jarak tetap 80 cm. Pengujian secara *realtime* dilakukan untuk mengevaluasi kemampuan sistem dalam mendeteksi dan mengenali Tanda Nomor Kendaraan Bermotor (TNKB) secara langsung melalui kamera (webcam) tanpa melalui proses penyimpanan atau pemrosesan citra terlebih dahulu.



Gambar 7. Motor Memasuki Parkiran

Sistem bekerja optimal ketika kendaraan berhenti selama 10–30 detik, karena kamera memiliki waktu untuk menyesuaikan fokus dan eksposur.

Tabel 2. Hasil Pengujian TNKB Secara *Realtime*

No	Gambar TNKB	Kondisi Cahaya	Tingkat Kecerahan Video (lux)	Jarak (cm)	Plat Terbaca	Plat Sebenarnya	Akurasi	Keterangan
1.		Terang (pagi-siang)	12280	80cm	BH 2761 EC	BH 2701 EC	85%	Angka "0" terbaca sebagai angka "6"
2.		Normal (sore)	652	80cm	BH 1953 YT	BH 4953 YT	85%	Angka "4" terbaca sebagai angka "1"
3.		Gelap (malam)	23	80cm	BH 2953	BH 4953 YT	75%	Angka "4" terbaca sebagai angka "2" dan Huruf YT tidak terbaca

Pada kondisi pencahayaan terang maupun normal, sistem mampu menghasilkan tingkat akurasi sekitar 85%. Sebaliknya, pada kondisi pencahayaan rendah, tingkat akurasi mengalami penurunan hingga 75%. Penurunan performa tersebut terjadi karena *metode Canny Edge Detection* mengalami kesulitan dalam mengekstraksi kontur karakter secara optimal pada citra yang memiliki tingkat kontras rendah.

4.5. Evaluasi Kinerja Menggunakan *Confusion Matrix*

Evaluasi dilakukan pada 24 karakter TNKB dari tiga kondisi pencahayaan. Untuk analisis *confusion matrix* ini, fokus pada pengenalan karakter individu (huruf/angka) dari ketiga hasil di atas. Total karakter pada setiap plat sebenarnya rata-rata terdiri dari 8 karakter (contoh: "BH2701EC").

Tabel 3. Hasil Pengujian Data

Kondisi	True Positive	True Negative	False Positive	False Negative	Total
Terang	7	0	1	0	8
Normal	7	0	1	0	8
Gelap	6	0	2	0	8
Total	20	0	4	0	24

Sistem berhasil mendapatkan nilai *True Positive* (TP) sebanyak 20, yang merepresentasikan bahwa terdapat 20 karakter TNKB yang berhasil dideteksi dengan benar sesuai kelasnya. Nilai *False Positive* (FP) sebesar 4 menunjukkan bahwa terdapat 4 karakter yang terdeteksi salah, misalnya akibat pantulan cahaya atau kemiripan bentuk huruf dan angka. Sementara itu, *False Negative* (FN) bernilai 0, artinya tidak ada karakter yang gagal terdeteksi oleh sistem.

Tabel 4. Perhitungan *Confusion Matrix*

Kondisi	Akurasi	Presisi	Recall	F-1 Score
Terang	87.5 %	87.5 %	100 %	93.33 %
Normal	87.5 %	87.5 %	100 %	93.33 %
Gelap	75 %	75 %	100 %	85.71 %
Rata-Rata	83.33 %	83.33 %	100 %	90.91 %

Berdasarkan hasil evaluasi menggunakan *confusion matrix*, sistem memperoleh tingkat akurasi sebesar 83,33%, yang menunjukkan proporsi karakter TNKB yang berhasil dikenali dengan benar terhadap keseluruhan proses deteksi. Nilai presisi sebesar 83,33% menandakan bahwa sebagian besar karakter yang terdeteksi sesuai dengan karakter sebenarnya. Sementara itu, nilai *recall* mencapai 100%, yang berarti seluruh karakter TNKB berhasil terdeteksi tanpa adanya karakter yang terlewat. Nilai *F1-Score* sebesar 90,91% mencerminkan keseimbangan yang baik antara ketepatan dan kelengkapan deteksi, sehingga dapat disimpulkan bahwa sistem menunjukkan performa yang stabil dan cukup andal dalam mengenali karakter TNKB pada berbagai kondisi pencahayaan.

5. KESIMPULAN

- a. Sistem deteksi TNKB berbasis YOLOv8, *Canny Edge Detection*, dan OCR mampu mengenali karakter plat nomor secara *realtime* dengan akurasi rata-rata 83,33%. Kinerja optimal diperoleh pada kondisi pencahayaan terang dan normal dengan akurasi 87,5%, sedangkan pada kondisi gelap akurasi menurun menjadi 75% karena rendahnya kontras citra.
- b. Hasil pengujian menunjukkan bahwa jarak dan pencahayaan berpengaruh signifikan terhadap akurasi pembacaan karakter. Performa terbaik dicapai saat kendaraan berada pada jarak 30–60 cm dan berhenti selama 10–30 detik, sehingga sistem dapat membaca TNKB secara optimal dan stabil.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini. Atas segala bantuan dan kerjasamanya semoga menjadi amal Saudara yang diberkahi rahmat yang melimpah dari Allah SWT.

DAFTAR PUSTAKA

- [1] Utama, S. W., & Kusumawardhani, A. (2017). Aplikasi Pendeteksi Plat Nomor Negara Indonesia Menggunakan OpenCV dan Tesseract OCR pada Android Studio". *Jurnal Utama Aplikasi*, 1–6.
- [2] Anwariyah, K., Sains, F., & Teknologi, D. (2020). Deteksi Objek Nomor Kendaraan Pada Citra Kendaraan Bermotor (Detection of Vehicle License Objects in Motorized Vehicle Images). *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 1(4), 311–317.
- [3] Galahartlambang, Y., Khotiah, T., Zahrudin Fanani, & Afifatul Aprilia Yani Solekhah. (2023). Deteksi Plat Nomor Kendaraan Otomatis Dengan Convolutional Neural Network Dan Ocr Pada Tempat Parkir Itb Ahmad Dahlan Lamongan. *Jurnal Manajemen Informatika Dan Sistem Informasi*, 6(2), 114–122.
- [4] Adi Saputra, R., Reskal, & Mimi Wahyuni, F. (2022). Segmentasi Pada Plat Kendaraan Dinas Menggunakan deteksi Tepi (Reskal) |328. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 6(1), 328–339.
- [5] Belajar Memahami Warna Tanda Nomor Kendaraan Bermotor (TNKB). (2022). Johorejo.
- [6] Effendy, V. A., & Maspiyanti, F. (2021). Perbandingan Algoritma Canny Edge Detection Dan Prewitt Pada Deteksi Stadium Diabetik Retinopati. *Jurnal Ilmiah Informatika*, 9(02), 87–94.
- [7] Iman, S. G., & Syamsuddin, E. Y. (2020). Perancangan Algoritma Optimasi Pada Pengenalan Karakter Plat Nomor Kendaraan Menggunakan Pengolahan Citra. *Komputika : Jurnal Sistem Komputer*, 9(2), 155–164.
- [8] Nehru, Hais, Y. R., & Devina I, A. C. (2024). Implementasi Canny Filter Optical CharacterRecognition(OCR)untuk Identifikasi Tanda Nomor Kendaraan Bermotor (TNKB). *Jurnal EngineeringI*.
- [9] Sulistiyanti, S. R., Setyawan, F. A., & Komarudin, M. (2016). PENGOLAHAN CITRA; DASAR DAN CONTOH PENERAPANNYA (Warsito (Ed.)). Teknosain.
- [10] Saluky, & Yoni Marine. (2023). Penerapan Algoritma Deteksi Tepi Canny Menggunakan Python Dan Opencv. *Smart Techno (Smart Technology, Informatics and Technopreneurship)*, 5(1), 1–7.
- [11] Sundani, D., Widiyanto, S., Karyanti, Y., & Wardani, D. T. (2019). Identification of image edge using quantum canny edge detection algorithm. *Journal of ICT Research and Applications*, 13(2), 133–144.
- [12] Hanif, A. R., Nasrullah, E., & Setyawan, F. (2023). DETEKSI KARAKTER PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN METODE OPTICAL CHARACTER RECOGNITION (OCR). *Jurnal Informatika Dan Teknik Elektro Terapan*, 11(1). <https://doi.org/10.23960/jitet.v11i1.2897>
- [13] Redmon, J., S. Divvala. R. Girshick. A. Farhadi. 2016. You only look once: Unified, real-time object detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 779-788.
- [14] M. Lutz, *Learning Python*, 5th ed. Sebastopol, CA, USA: O'Reilly Media, 2013.
- [15] Santra, A.K., C. J. Christy. 2012. Genetic Algorithm and Confusion Matrix for Document Clustering. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. 3(7):322-328.