

Pemanfaatan Model *YOLOv8* Untuk Mendeteksi Plat Nomor Kendaraan Mobil Pada Gerbang Masuk Universitas XYZ

Esi Putri Silmina¹⁾, Restu Agil Yuli Arjun²⁾

¹⁾²⁾ Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Aisyiyah Yogyakarta

Jl. Siliwangi (Ring Road Barat) No. 63, Sleman, Daerah Istimewa Yogyakarta

¹⁾ esiputrisilmina@unisayogya.ac.id

²⁾ restuagil.ya@gmail.com

Abstrak

Kendaraan bermotor merupakan salah satu transportasi yang sering digunakan oleh masyarakat Indonesia. Dalam penggunaan kendaraan bermotor ini, diperlukan adanya plat nomor sebagai Tanda Kendaraan Bermotor (TNKB) yang memuat kode wilayah, nomor registrasi, masa berlaku, dan memenuhi spesifikasi yang diatur. Sistem deteksi otomatis plat nomor kendaraan perlu disadari menjadi hal yang dibutuhkan guna memudahkan pencatatan, pengawasan, dan identifikasi kendaraan. Tujuan penelitian ini untuk mendeteksi plat nomor kendaraan yang masuk ke dalam lingkungan kampus Universitas XYZ. Metode penelitian yang digunakan yaitu pengumpulan data, anotasi data, pembagian data, *preprocessing* data, pelatihan model, implementasi model, dan pengujian model pada sistem. Tujuan penelitian ini untuk melakukan pendeteksian plat nomor kendaraan jenis mobil yang masuk ke lingkungan kampus Universitas XYZ menggunakan model *YOLOv8*. Hasil penelitian menunjukkan algoritma *YOLOv8* dengan *PaddleOCR* memberikan nilai performansi yang sangat baik dengan nilai hasil pelatihan model mendapatkan *mAP50* sebesar 98.9%, *precision* 98,8%, *recall* sebesar 96.5%, dan akurasi sistem sebesar 90% dalam skala likert.

Kata kunci: deteksi objek, gerbang masuk, mobil, plat nomor, *YOLOv8*

Abstract

*Motorized vehicles are one of the most common forms of transportation used by the people of Indonesia. In the use of these motorized vehicles, it is necessary to have a license plate as a Motor Vehicle Sign (TNKB) which contains the region code, registration number, validity period, and meets the regulated specifications. Automatic detection system of vehicle number plates needs to be realized to be needed to facilitate recording, monitoring, and identification of vehicles. The purpose of this research is to detect vehicle license plates that enter the XYZ University campus environment. The research methods used are data collection, data annotation, data sharing, data preprocessing, model training, model implementation, and model testing on the system. The purpose of this research is to detect car license plates that enter the XYZ University campus environment using the *YOLOv8* model. The results showed that the *YOLOv8* algorithm with *PaddleOCR* provided excellent performance values with the value of the model training results getting *mAP50* of 98.9%, *precision* of 98.8%, *recall* of 96.5%, and system accuracy of 90% on a Likert scale.*

Keywords: object detection, entrance gate, number plate, cars, *YOLOv8*

1. PENDAHULUAN

Masyarakat memerlukan alat mobilisasi yang nyaman dan sesuai kebutuhan untuk memenuhi kebutuhan hidup. Transportasi darat seperti penggunaan kendaraan bermotor menjadi bagian dari kehidupan masyarakat Indonesia. Penggunaan kendaraan bermotor berkaitan dengan Tanda

Nomor Kendaraan Bermotor (TNKB) sesuai dengan Undang Undang yang mewajibkan nomor registrasi, kode wilayah, masa berlaku, serta memenuhi spesifikasi yang sudah ditetapkan [1].

Penggunaan plat nomor kendaraan di Indonesia, terdapat formasi khusus dengan menggunakan tiga nomor seri. Pada kode huruf awal menunjukkan tempat dimana nomor tersebut didaftarkan. Sedangkan kode huruf belakang menunjukkan tempat terdaftarnya kendaraan. Nomor polisi kendaraan didistribusikan untuk jenis kendaraan-kendaraan tertentu. Kendaraan penumpang memiliki klasifikasi nomor seri khusus, yang dibagi ke dalam tiga jenis kendaraan yakni kendaraan beban, motor, dan bus. Nomor polisi dikeluarkan selaras dengan urutan registrasi yang terdiri dari satu hingga empat digit angka setelah kode area registrasi. Nomor urut akan kembali ke awal dengan pemberian huruf seri A-Z sebagai tanda pengenal di belakang angka registrasi jika nomor urut registrasi habis. Penetapan kode wilayah registrasi disesuaikan dengan Surat Keputusan Kapolri nomor 4 tahun 2006 [2].

Peningkatan populasi di Indonesia menjadi suatu isu yang serius terutama berkaitan dengan kesiapan prasarana umum yang dapat digunakan. Salah satu sarana tersebut ialah lahan parkir terutama di lingkungan kampus. Peningkatan populasi juga meningkatkan kepemilikan kendaraan. Hal ini dapat menyebabkan kondisi parkir menjadi tidak tertib yang akan memudahkan tindak kejahatan seperti pencurian. Selain keamanan, hal ini juga akan menyulitkan pendataan kuantitas kendaraan yang masuk lingkungan kampus yang dapat dijadikan data agar pihak pengelola dapat mengetahui kebutuhan lahan parkir [3].

Penggunaan teknik *Deep Learning* dalam pendeteksian objek serta pengenalan karakter menunjukkan progresivitas yang baik. *YOLO (You Only Look Once)* merupakan model yang paling diminati karena menjadi algoritma pendeteksian objek yang dinilai akurat dan cepat [4].

YOLOv8 memiliki spesifikasi dan keunggulan dalam mendeteksi objek. *YOLOv8* sendiri memiliki arsitektur yang paling efisien saat ini dikarenakan menggabungkan arsitektur dari *YOLOv5* dan *YOLOv7* sehingga memiliki waktu komputasi yang lebih cepat [5].

Penelitian sebelumnya yang telah dilakukan untuk memanfaatkan *YOLOv8* sebagai model untuk mendeteksi plat nomor polisi kendaraan [4], klasifikasi warna plat nomor polisi kendaraan [5], serta menghitung jumlah kendaraan [6], *YOLOv8* terbukti memiliki akurasi yang cukup baik.

Tujuan penelitian ini yaitu melakukan pendeteksian plat nomor kendaraan jenis mobil yang masuk ke lingkungan kampus Universitas XYZ dengan memanfaatkan model *YOLOv8* yang dinilai memiliki akurasi yang cukup baik. Hal ini diharapkan akan memudahkan proses identifikasi kendaraan jenis mobil di lingkungan kampus.

2. TINJAUAN PUSTAKA

Penulis mengambil beberapa teori yang tepat berkaitan dengan penelitian ini dari berbagai sumber. Hal ini dimaksudkan agar memudahkan proses penulisan sehingga menghasilkan informasi pemanfaatan model sistem yang bermanfaat. Berikut penulis paparkan beberapa teori yang dijadikan landasan oleh penulis dalam penelitian ini.

2.1 *Machine Learning*

Keputusan cerdas yang dibuat oleh komputer berdasarkan pembelajaran data yang telah ada sebelumnya dihasilkan oleh *Machine learning* yang merupakan bagian dari *Artificial Intelligence (AI)*. Mesin akan mengenali pola, menetapkan keputusan, dan terus memproses serta mempelajari data yang ada sehingga akan meningkatkan performa pekerjaannya [7]. *Machine learning* berfokus pada pengembangan sistem yang dapat mengambil keputusan sendiri tanpa diprogram berulang kali oleh manusia [8].

2.2 *Deep Learning*

Kemampuan dalam menganalisa serta menarik kesimpulan pada komputer disebut sebagai *Deep Learning*. Cara kerja yang dilakukan oleh metode *Deep Learning* serupa dengan jaringan saraf tiruan, namun metode *Deep Learning* tidak hanya mengandalkan 1 *layer*, dimana pembelajaran komputer dapat semakin lama ketika *layer* yang digunakan semakin tinggi. Data

yang tersedia akan dipelajari menggunakan algoritma yang ada, yakni dengan menggunakan *Long-Short Term Memory*. Terdapat tiga pilihan yang dapat dilakukan dalam *Deep Learning*, yakni secara terarah, tidak terarah, dan semi terarah. Algoritma pada pembelajaran secara terarah berdasarkan data yang tersedia. Algoritma pada pembelajaran secara tidak terarah tidak berdasarkan data atau tidak ada data satupun. Sedangkan pada semi terarah, data tidak mumpuni walaupun tersedia [9].

2.3 YOLO (You Only Look Once)

YOLO (You Only Look Once) merujuk pada suatu pengembangan terbaru untuk pendeteksian objek *real-time* dengan penggunaan *Convolutional Neural Network (CNN)* [6]. Cara kerja YOLO dalam memprediksi objek ialah dengan melihat satu kali gambar pada data yang tersedia karena YOLO memanfaatkan *convolutional network*, sehingga YOLO akan menilai *bounding box* dan *probabilitas class* secara terus menerus pada box [1]. YOLO membagi gambar masukan ke dalam sebuah *grid* dengan ukuran $S \times S$. *Input Size* pada suatu arsitektur akan memengaruhi dari ukuran *Grid Cell*. Prediksi objek akan dilakukan oleh setiap sel disertai *bounding box* dan *confidence* yang menjadi nilai probabilitas keberadaan suatu objek pada *bounding box* tersebut. *Class Probability Map* akan terbentuk setelah *bounding box* dipetakan berdasarkan nilai *confidence* yang dihasilkan dan kemudian YOLO akan memprediksi objek yang terdapat dalam *bounding box* tersebut beserta kemungkinannya [10].

YOLOv8 merupakan keluaran terbaru dari *ultralytics* yang memiliki kinerja lebih cepat dan akurat dibandingkan versi yang lain sehingga *YOLOv8* ini cocok digunakan sebagai algoritma deteksi objek [11]. *YOLOv8* dibekali arsitektur jaringan saraf yang memanfaatkan *Feature Pyramid Network (FPN)* dan *Path Aggregation Network (PAN)* serta alat pelabelan baru yang akan menyederhanakan proses anotasi. Alat pelabelan ini memiliki fitur-fitur yang bermanfaat seperti pelabelan otomatis, pintasan pelabelan, dan tombol pintas yang dapat disesuaikan. Kombinasi dari fitur-fitur ini akan memudahkan proses anotasi gambar untuk melatih model. *FPN* bekerja dengan mengurangi resolusi spasial gambar masukan secara bertahap sambil meningkatkan jumlah saluran fitur. Hal ini menciptakan peta fitur yang mampu mendeteksi objek di berbagai tempat skala dan resolusi. Di sisi lain, arsitektur *PAN* menggabungkan fitur-fitur dari berbagai tingkat jaringan melalui *skip connections*. Melalui hal tersebut jaringan dapat menangkap fitur dengan lebih baik pada berbagai skala dan resolusi, yang bermanfaat dalam menangkap objek secara akurat dalam ukuran dan bentuk yang berbeda [12].

2.4 Paddle OCR

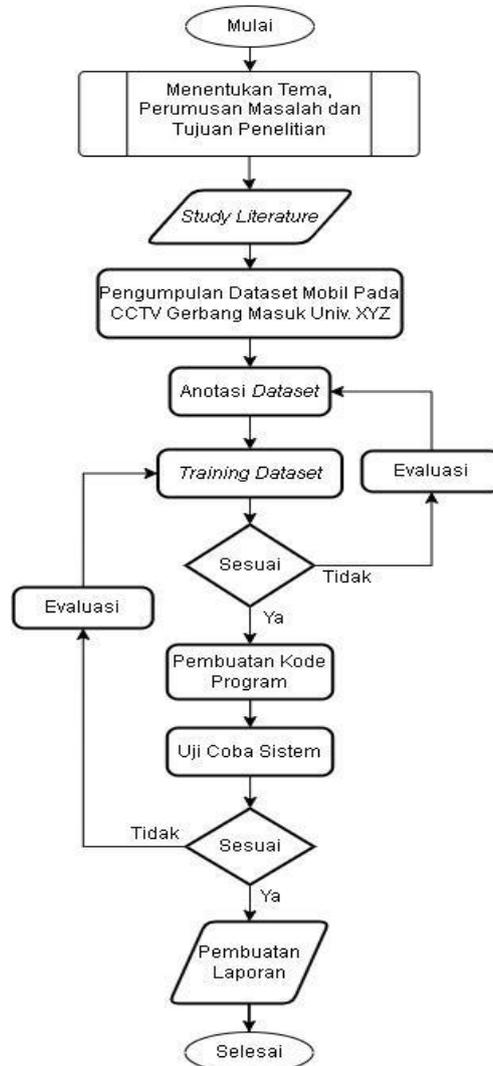
OCR (Optical Character Recognition) mengacu pada pengenalan otomatis konten teks dalam gambar, yang juga merupakan salah satu cabang penting dari *computer vision*. *OCR* tradisional mengadopsi teknologi pengenalan pola yang memiliki banyak kelemahan, seperti terlalu banyak *processing link*, proses yang panjang, fleksibilitas pengenalan yang buruk, dan pemeliharaan yang sulit. Teknologi *OCR* berdasarkan *Deep Learning* banyak diaplikasikan dalam berbagai bidang. Saat ini, proses pengenalan *OCR* terdiri dari 3 tahapan, 1) *preprocessing*, 2) deteksi teks, dan 3) pengenalan karakter, *preprocessing* biasanya untuk memperbaiki masalah pencitraan gambar, termasuk pengurangan *noise* gambar, koreksi kemiringan, penghapusan *blur*, dan lain sebagainya. Saat ini, ekstraksi fitur peningkatan model jaringan saraf berbasis *CNN* banyak digunakan untuk menyelesaikan masalah di atas. Pada deteksi teks, algoritma pengenalan target berdasarkan jaringan saraf konvolusional diterapkan pada deteksi teks. Sedangkan pengenalan karakter berfungsi dalam mengidentifikasi konten teks dalam gambar.

Paddle OCR merupakan seperangkat perpustakaan alat *OCR* yang kaya, canggih dan praktis berdasarkan teknologi *Deep Learning* yang bersifat *open source*. *Paddle OCR* menyediakan berbagai algoritma pelatihan deteksi teks dan pengenalan karakter yang dapat berjalan di *Windows*, *Linux*, *MacOS*, dan sistem lainnya.

3. METODE PENELITIAN

3.1 Diagram Alir Penelitian

Metode penelitian yang dilakukan melalui serangkaian tahapan, tahapan tersebut digambarkan pada gambar 1 dibawah ini:



Gambar 1. Diagram alir metode penelitian

Penelitian ini dimulai dengan menentukan tema, merumuskan masalah, dan menetapkan tujuan. Tahap berikutnya adalah melakukan studi literatur dengan meneliti dan membaca laporan serta jurnal penelitian terkait. *Dataset* dikumpulkan melalui rekaman CCTV Universitas XYZ. Setelah terkumpul, dilakukan anotasi dan pelatihan menggunakan *platform Roboflow* dan *Tesla T4 GPU* pada *Google Colaboratory*. Jika hasilnya tidak memuaskan, anotasi ulang akan dilakukan. Selanjutnya membuat kode program menggunakan bahasa pemrograman *Python*. Kemudian, dilakukan uji coba sistem, jika hasilnya tidak memuaskan, pelatihan dataset akan diulang. Langkah terakhir adalah menyusun laporan.

3.2 Tahap Penelitian

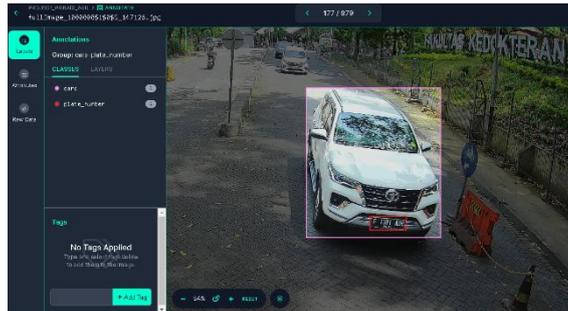
Fokus penelitian ini adalah menerapkan *YOLOv8* dan *Paddle OCR* untuk mendeteksi mobil dan plat nomor kendaraan. Hasil deteksi akan menampilkan gambar dengan label “cars” dan “plate_number”. Hasil deteksi dari label “plate_number” akan diekstraksi menjadi sebuah teks. Tahapan penelitian ini dibagi menjadi beberapa bagian diantaranya pengumpulan data, anotasi data, pembagian data, data *pre-processing*, pelatihan model, dan implementasi model.

3.3.1 Pengumpulan Data

Data penelitian dikumpulkan melalui rekaman CCTV gerbang masuk di Universitas XYZ yang didapat dari proyek magang. Data yang didapat berupa gambar dengan dua kelas yaitu “plate_number” dan “cars”.

3.3.2 Anotasi Data

Dataset dilakukan anotasi menggunakan platform Roboflow dengan dua kelas yaitu, “plate number” dan “cars”. Proses anotasi ini merupakan langkah krusial dalam mempersiapkan dataset untuk selanjutnya masuk kepada tahap pelatihan data sehingga didapatkan akurasi model deteksi yang akurat.

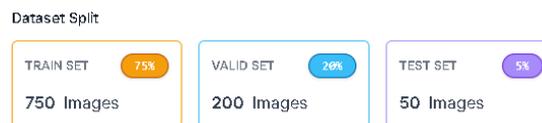


Gambar 2. Anotasi gambar

Pada Gambar 2, anotasi dilakukan untuk dua jenis kelas yaitu, “cars” yang diberi label bounding box berwarna merah muda dan “plate_number” yang diberi label bounding box berwarna merah.

3.3.3 Pembagian Data

Pembagian data dilakukan secara *random* untuk mencegah terjadinya bias pada setiap bagian. Selain itu, hal ini dilakukan guna memastikan bahwa masing-masing bagian merepresentasikan variasi dan karakteristik dataset secara menyeluruh.



Gambar 3. Pembagian data

Dalam tahap pembagian data, dataset dibagi menjadi tiga bagian yang berbeda yaitu data latih, data valid, dan data uji. Data latih adalah bagian terbesar, dengan persentase sebesar 75% dari total data dan terdiri dari 750 images. Pada proses pelatihan, model akan belajar mengenali pola yang ada dalam data latih.

Kemudian, data valid menjadi bagian kedua dengan persentase sebesar 20% dari total data dan terdiri dari 200 images. Dengan menggunakan data valid, performa model selama pelatihan akan diukur sehingga dapat diidentifikasi apakah model tersebut mengalami *Overfitting* (keadaan saat suatu model mempelajari data pelatihan dengan terlalu baik sehingga membuatnya berkinerja buruk pada data baru) atau *Underfitting* (keadaan saat suatu model *machine learning* tidak bisa mempelajari relasi antara variabel dalam data dan memperkirakan atau mengklasifikasikan data point baru).

Selanjutnya, data uji menjadi bagian paling kecil dengan proporsi 5% dari total data, yang terdiri dari 50 images. Sesuai namanya, bagian ini berfungsi untuk menguji performa model secara mandiri atau terpisah. Data uji menguji kemampuan model dalam mengenali images yang belum pernah dilihat sebelumnya.

3.3.4 Data Pre-processing

Preprocessing data adalah tahap penting dalam mempersiapkan *dataset* sebelum dilatih pada pengenalan objek. Dalam penelitian ini, *preprocessing* data menggunakan dua teknik yaitu, *Auto-Orient* dan *Resize*.

Teknik *Auto-Orient* berfungsi untuk menyesuaikan orientasi gambar kendaraan agar seragam pada *dataset*. Hal ini penting sebab variasi orientasi gambar kendaraan akan menyulitkan pelatihan model. Selanjutnya, melakukan proses *Resize* untuk mengubah seluruh ukuran gambar menjadi 640 x 640 *piksel*, proses ini memastikan ukuran gambar dalam *dataset* menjadi konsisten. Dengan ukuran yang seragam, model dapat belajar pola objek dengan lebih efisien dari gambar-gambar tersebut.

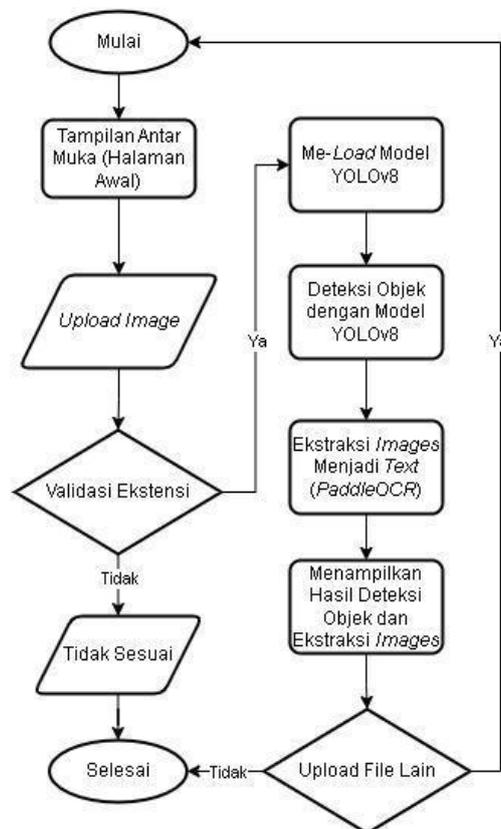
3.3.5 Pelatihan Model

Tahapan awal dalam pelatihan model adalah mempersiapkan *dataset* yang telah disusun sebelumnya di *platform Roboflow*. Setelah *dataset* tersusun, selanjutnya adalah model dilatih menggunakan *Transfer Learning* dengan memanfaatkan model *YOLOv8n*. *Transfer Learning* dilakukan untuk menggunakan pengetahuan yang telah dipakai oleh model *YOLOv8n* pada proyek-proyek sebelumnya.

Proses pelatihan dilakukan dalam 100 *epoch*, model disesuaikan dengan data pelatihan, kemudian diuji pada data valid untuk mengetahui dan mengevaluasi kinerja pada model. Proses pelatihan berjalan secara iteratif, model terus diperbarui dan disesuaikan dengan *dataset* pelatihan. Setelah proses pelatihan selesai, model yang telah dilatih akan memiliki pemahaman yang lebih bagus tentang kemampuannya untuk mendeteksi dan mengenali objek.

3.3.6 Implementasi Model

Sistem akan dibuat menggunakan bahasa pemrograman *Python* dengan *library Flask* dan *Pytorch*. Alur sistem dapat dilihat pada gambar 4 dibawah ini:



Gambar 4. Flowchart sistem

Alur dimulai ketika *user* mengunjungi halaman utama yang menampilkan halaman “*index.html*” untuk meng-*upload* file *images*. File yang diizinkan berupa file dengan ekstensi *.jpg*, *.png*, dan *.jpeg*. Jika file yang di *upload* tidak sesuai ekstensi, maka program akan *error* dan berhenti. Jika file yang di *upload* sesuai, maka program akan memuat model *YOLOv8* yang sudah dilatih sebelumnya dengan nama “*best.pt*”. Kemudian, akan didapatkan hasil deteksi gambar berupa “*cars*” dan “*plate_number*”. Hasil deteksi “*plate_number*” akan di *crop* menyesuaikan dengan hasil deteksi. Hasil dari *cropping* tersebut akan diproses oleh *Paddle OCR* untuk di ekstraksi menjadi teks.

Selanjutnya, program akan mengarahkan menuju tampilan “*hasildeteksi.html*”. Halaman ini memberikan informasi berupa gambar yang di *upload*, hasil deteksi “*cars*” dan “*plate_number*”, hasil *cropping* gambar, dan hasil ekstraksi gambar menjadi teks pada hasil *cropping*.

3.3.7 Pengujian Sistem

Pengujian dilakukan untuk mengetahui tingkat akurasi sistem dalam mengekstrak gambar menjadi teks. Pengujian dilakukan menggunakan Persamaan (1) dan skala likert pada Tabel 1.

$$\text{Persentase Hasil Akurasi Pengujian} = \frac{x}{n} \times 100\% \quad (1)$$

Keterangan:

n = jumlah data uji

x = jumlah data uji yang benar

Tabel 1. Skala likert

Persentase	Predikat
0% - 55%	Buruk
56% - 70%	Cukup
71% - 80%	Baik
81% - 100%	Sangat Baik

4. PEMBAHASAN

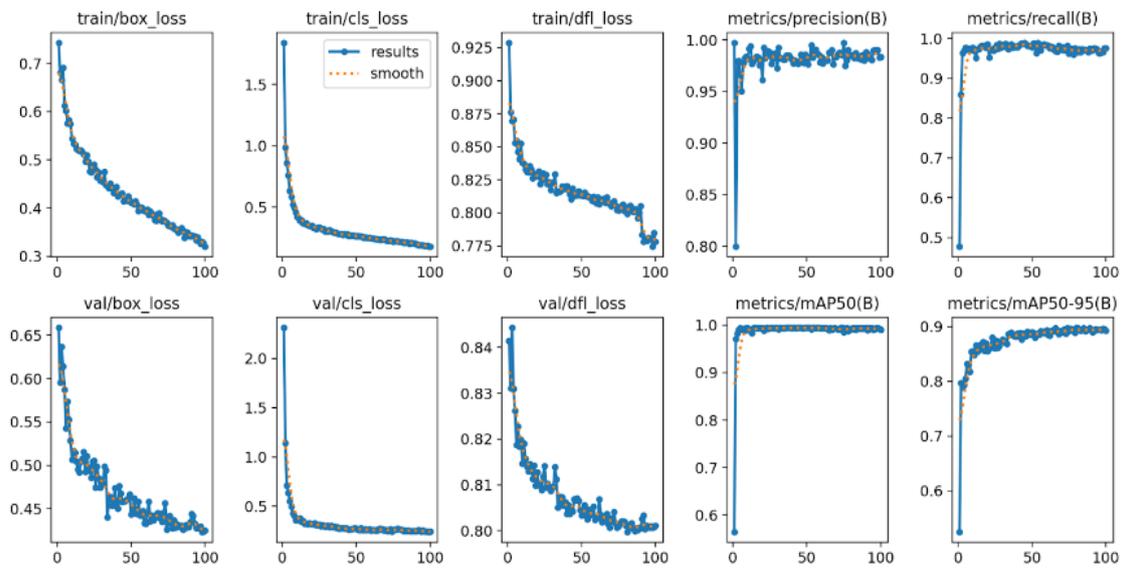
Bagian ini membahas mengenai hasil *training dataset* dan hasil uji coba sistem beserta persentase nilai akurasi dari sistem yang telah dibuat.

4.3 Hasil Training Dataset

Dataset dilakukan *training* menggunakan *Tesla T4 GPU* pada *Google Colaboratory* untuk membuat model *YOLOv8* yang akan digunakan, model tersebut hanya akan mendeteksi mobil dan plat nomor polisi saja. *Training dataset* dilakukan selama 38 menit untuk 1000 *dataset* yang akan digunakan yang di bagi menjadi tiga sub *dataset* yaitu data latih sebanyak 750 data gambar, data valid sebanyak 200 data gambar, dan data uji sebanyak 50 data gambar.

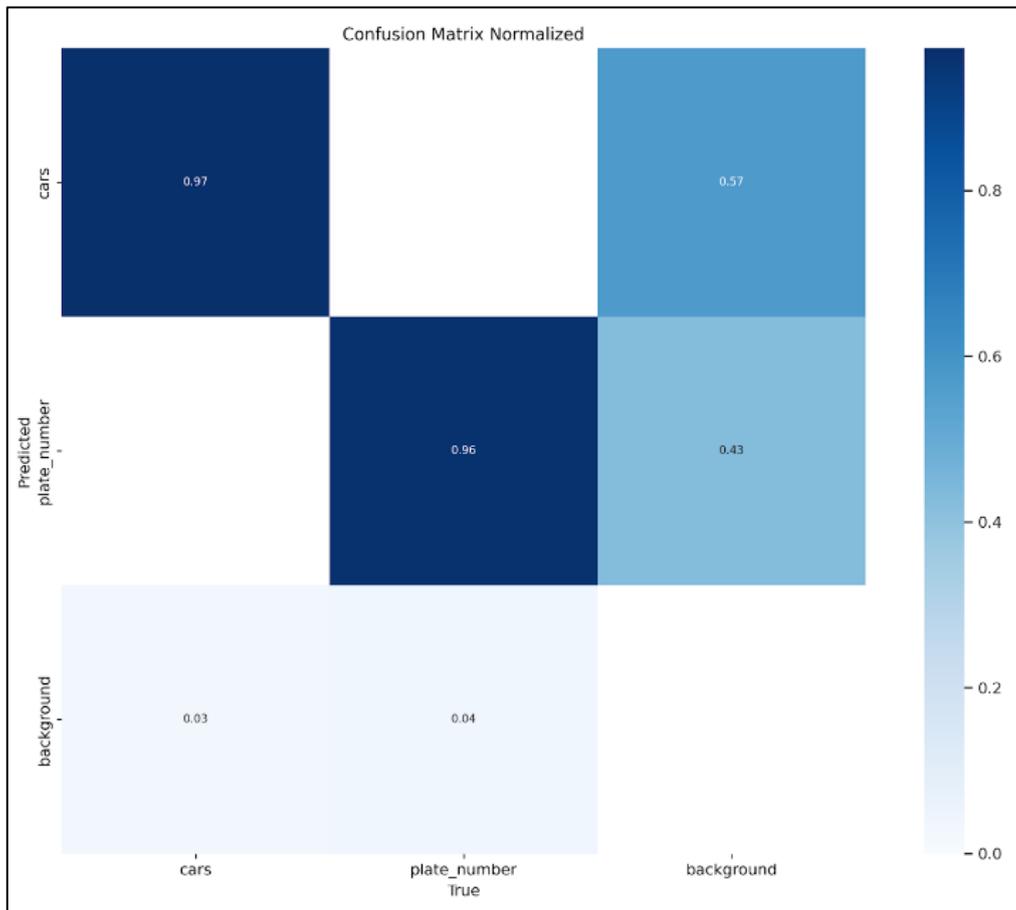
Training dataset juga dilakukan menggunakan *platform roboflow*, hal ini dilakukan guna mengetahui nilai dari *mAP*, *recall*, dan *precision* yang didapat. Untuk *mAP50* nilai yang dihasilkan sebesar 98,9%, kemudian *precision* sebesar 98,8% dan *recall* sebesar 96,5%.

Saat proses *training*, model mengalami perkembangan seiring dengan metrik yang dievaluasi dalam 100 *epoch*. Gambar 5 menunjukkan hasil dari pelatihan model *dataset*.



Gambar 5. Hasil pelatihan model *dataset*

Selama pelatihan model *dataset*, nilai *loss* pada model terus menurun. “*box loss*” digunakan untuk mengetahui seberapa besar perbedaan antara prediksi model dan lokasi objek pada gambar yang sebenarnya, dengan nilai akhir mencapai 0,31. “*cls_loss*” digunakan untuk mengetahui kesalahan model saat mendeteksi objek dengan nilai akhir mencapai 0,17.



Gambar 6. Multiclass confusion matrix

Pada Gambar 6, *Multiclass Confusion Matrix* mendapatkan hasil yang baik. Kelas *plate_number* mendapatkan *True Positive (TP)* sebesar 0,96 dan *False Negative* sebesar 0,43. Kelas *cars* mendapatkan *True Positive (TP)* hingga 0,97 dan *False Negative* sebesar 0,57. Nilai *False Negative* yang tinggi dapat berdampak pada model sehingga salah mendeteksi kelasnya.

4.4 Hasil Uji Coba Sistem

Pengujian sistem dilakukan dengan 20 gambar mobil yang berbeda berdasarkan gambar hasil rekaman CCTV dari gerbang Masuk Universitas XYZ. Secara garis besar, sistem mampu mendeteksi dan mengekstrak gambar plat nomor kendaraan menjadi teks. Namun, ada dua gambar yang tidak akurat dalam ekstraksi gambar menjadi teks, salah satunya terdapat pada gambar 8.



Gambar 7. Hasil deteksi berhasil



Gambar 8. Hasil deteksi tidak berhasil

Pada Gambar 8, sistem mengalami kesulitan dalam mengekstrak dua huruf belakang pada plat nomor, nomor polisi kendaraan yang tertera adalah AB XX28 XW, namun sistem mendeteksi sebagai AB XX28. Hal ini mungkin terjadi sebab kualitas gambar yang kurang baik serta terdapat kerancuan huruf seperti pada Gambar 8. Selanjutnya, diperlukan perhitungan untuk mencari nilai akurasi pada sistem dengan Persamaan (1) dan skala likert pada Tabel 1.

$$\text{Persentase Hasil Akurasi Pengujian} = \frac{18}{20} \times 100\% = 90\%$$

Berdasarkan perhitungan tersebut, nilai akurasi pada sistem deteksi plat nomor di Universitas XYZ mendapatkan skor sebesar 90%. Hal ini mendapat predikat sangat baik sesuai pada skala likert pada Tabel 1.

5. KESIMPULAN

Berdasarkan hasil pelatihan model yang dilakukan untuk deteksi kendaraan mobil pada gerbang masuk Universitas XYZ, Model *YOLOv8* terbukti berhasil mendeteksi mobil dan plat nomor kendaraan dengan baik. Hal ini dapat dilihat pada hasil *training* model yang mendapatkan nilai *mAP* sebesar 98.9%, *precision* 98,8%, *Recall* sebesar 96.5% dan hasil pengujian yang dihitung dengan nilai akurasi sistem dalam mendeteksi dan mengekstrak gambar menjadi teks menggunakan *PaddleOCR* mendapatkan nilai sebesar 90% atau predikat sangat baik dalam skala

likert. Hasil penelitian ini diharapkan akan memudahkan proses identifikasi kendaraan jenis mobil di lingkungan kampus Universitas XYZ.

DAFTAR PUSTAKA

- [1] E. Tirtana, K. Gunadi, and I. Sugiarto, "Penerapan Metode YOLO dan Tesseract-OCR untuk Pendataan Plat Nomor Kendaraan Bermotor Umum di Indonesia Menggunakan Raspberry Pi," *J. Infra*, vol. 9, no. 2, pp. 241–247, 2021.
- [2] Y. O. L. Rema, "Deteksi Plat Nomor Kendaraan Bermotor dengan Segmentasi Gambar," *J. Saintek Lahan Kering*, vol. 2, no. 1, pp. 20–23, 2019.
- [3] S. Syaifullah, H. Radiles, and T. K. Ahsyar, "Manajemen Kebutuhan Ruang Parkir Kampus (Studi Kasus: Fakultas Sains dan Teknologi UIN Suska Riau)," *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 18, no. 1, pp. 1–10, 2020, doi: 10.24014/sitekin.v18i1.10607.
- [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, 2023, doi: 10.46984/sebatik.v27i2.2374.
- [5] Siswanti, F. Utaminingrum, and R. R. M. Putri, "Purwarupa Sistem Klasifikasi Warna Plat Kendaraan Berbasis YOLOv8 Pada Gerbang Masuk UB," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, 2024.
- [6] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan," *Komputa J. Ilm. Komput. Dan Inform.*, vol. 12, no. 2, pp. 91–99, 2023, doi: 10.34010/komputa.v12i2.10654.
- [7] A. Fathurohman, "Machine Learning untuk Pendidikan: Mengapa dan Bagaimana," *J. JITEK*, vol. 1, no. 3, pp. 57–62, 2021.
- [8] R. Diana, H. Warni, and T. Sutabri, "Penggunaan Teknologi Machine Learning untuk Pelayanan Monitoring Kegiatan Belajar Mengajar pada SMK Bina Sriwijaya Palembang," *JUTEKIN (Jurnal Tek. Inform.*, vol. 11, no. 1, 2023, doi: 10.51530/jutekin.v11i1.709.
- [9] R. A. Tilasefana and R. E. Putra, "Penerapan Metode Deep Learning Menggunakan Algoritma CNN Dengan Arsitektur VGG NET Untuk Pengenalan Cuaca," *J. Informatics Comput. Sci.*, vol. 5, no. 1, pp. 48–57, 2023, doi: 10.26740/jinacs.v5n01.p48-57.
- [10] A. Aprilino and I. H. Al Amin, "Implementasi Algoritma Yolo dan tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis," *J. Teknoinfo*, vol. 16, no. 1, pp. 54–59, 2022.
- [11] M. I. Maulana and R. Noviana, "Training Custom Model Deteksi Uang Menggunakan YOLOv8," *J. Ilm. Komputasi*, vol. 22, no. 4, pp. 505–514, 2024, doi: 10.32409/jikstik.22.4.3526.
- [12] D. Reis, J. Hong, J. Kupec, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," 2024.

Biodata Penulis

Esi Putri Silmina, Dosen tetap pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Aisyiyah Yogyakarta.

Restu Agil Yuli Arjun, Mahasiswa pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Aisyiyah Yogyakarta.