

Penerapan Algoritma *You Only Look Once* (YOLO) Versi 8 dan *Convolutional Neural Network* (CNN) Dan *Image Processing* pada Aplikasi *Automatic Number Plate Recognition* (ANPR)

Herfandi Herfandi^{1*}, M. Alif Aldiansyah², M. Julkarnain³, Agung Susilo Yuda Irawan⁴

^{1,2,3}Program Studi Informatika, Universitas Teknologi Sumbawa

⁴Program Studi Informatika, Universitas Singaperbangsa Karawang

Email: *herfandi@uts

Abstrak. Penelitian ini mengembangkan sistem *Automatic Number Plate Recognition* (ANPR) untuk meningkatkan keamanan dan efisiensi manajemen kendaraan di Universitas Teknologi Sumbawa (UTS). Pendekatan ini menggunakan YOLOv8 untuk deteksi plat kendaraan, CNN untuk pengenalan karakter, dan image processing untuk segmentasi serta mendukung proses deteksi dan rekognisi. Data dikumpulkan dari *Kaggle* dan diolah melalui *Roboflow* untuk anotasi kemudian dilatih. Hasil menunjukkan akurasi segmentasi 97.3% dan prediksi karakter 94.6%, lebih baik dibandingkan penelitian sebelumnya dengan akurasi 93.3%. Pengujian menggunakan 20 sampel gambar menunjukkan rata-rata *confidence* deteksi 94.8% untuk hasil deteksi YOLOv8. Model ANPR ini berhasil meningkatkan akurasi dan efisiensi identifikasi kendaraan, mengatasi tantangan pencahayaan dan kualitas gambar. Studi ini menyarankan peningkatan *dataset*, teknik pemrosesan gambar, dan pengembangan antarmuka aplikasi yang ramah pengguna untuk implementasi lebih lanjut.

Kata kunci: *anpr; yolov8; cnn; image processing*

1 Pendahuluan

Di era perkembangan teknologi informasi dan komunikasi yang pesat saat ini, kebutuhan akan sistem keamanan dan pemantauan yang efisien menjadi sangat penting. *Automatic Number Plate Recognition* (ANPR) adalah teknologi yang memainkan peran krusial dalam sistem keamanan transportasi [1]. Teknologi ini memungkinkan komputer untuk secara otomatis membaca dan menginterpretasi nomor plat kendaraan dari gambar yang diambil oleh kamera [2]. Penerapan ANPR sangat luas, mencakup sistem keamanan publik [3], manajemen lalu lintas perkotaan [4], dan pengidentifikasian kendaraan secara cepat dan akurat [5]. Teknologi ini juga membantu dalam penegakan hukum dan keamanan dengan memfasilitasi pelacakan dan pencatatan kendaraan secara otomatis. Selain itu, ANPR digunakan untuk mengoptimalkan aliran lalu lintas dan mengurangi kemacetan, serta mendukung sistem pembayaran tol tanpa henti, yang semakin meningkatkan efisiensi dalam pengelolaan transportasi. Dengan kemampuan

untuk beroperasi di berbagai kondisi pencahayaan dan kecepatan, ANPR menjadi alat yang sangat berharga dalam era digital ini.

Kampus Universitas Teknologi Sumbawa (UTS), yang terletak di Kabupaten Sumbawa, Provinsi Nusa Tenggara Barat (NTB) mengalami pertumbuhan jumlah mahasiswa, staf pengajar, dan pengunjung dari luar setiap harinya. Kampus ini memiliki area yang luas dan terbuka, mendukung berbagai kegiatan akademik dan non-akademik yang berlangsung secara dinamis. Namun, tingginya volume kendaraan yang masuk dan keluar setiap hari tanpa sistem identifikasi nomor kendaraan yang efektif, berpotensi menyebabkan risiko keamanan yang signifikan serta kendala dalam manajemen keluar masuknya kendaraan dalam lingkungan kampus. Saat ini, UTS belum memiliki sistem identifikasi nomor kendaraan masuk yang terintegrasi, sehingga pengelolaan akses kendaraan masih dilakukan secara manual. Hal ini tidak efisien, terutama pada jam sibuk, dan meningkatkan risiko kecelakaan serta kejadian atau peristiwa yang tidak diinginkan. Penerapan sistem yang mampu mengidentifikasi dan merekam data kendaraan secara otomatis akan sangat membantu dalam mengoptimalkan kontrol keamanan dan manajemen keluar masuknya kendaraan dalam lingkungan kampus.

Penelitian terdahulu yang dilakukan oleh peneliti lain terkait dengan penelitian saat ini adalah sebagai berikut, Penelitian oleh Setiawan S., dkk (2023) bertujuan mengembangkan sistem pengenalan nomor plat kendaraan yang cepat dan akurat menggunakan deteksi tepi, segmentasi karakter, OCR, dan template matching, dengan tingkat keberhasilan sekitar 80% [6]. Penelitian ini terbatas pada pengenalan karakter saja. Penelitian saat ini menggunakan YOLOv8 untuk deteksi plat, diikuti oleh segmentasi dan pengenalan karakter untuk hasil yang lebih komprehensif. Penelitian oleh Aprilino A., dkk (2022) mengembangkan sistem deteksi plat nomor otomatis menggunakan YOLO dan *Tesseract* OCR, mencapai akurasi deteksi 100% dan pengenalan teks 92.32% [7]. Keterbatasannya termasuk ketergantungan pada pencahayaan dan kualitas gambar. Penelitian saat ini menggunakan teknik segmentasi dan CNN dengan *dataset* yang disesuaikan dengan kondisi plat Indonesia. Penelitian oleh Zakiyamani M., dkk (2022) bertujuan mengotomatisasi pencatatan nomor kendaraan menggunakan CNN melalui *OpenCV*, mencapai akurasi 96% [8]. Keterbatasannya termasuk ketergantungan pada pencahayaan dan kualitas gambar tinggi. Penelitian saat ini menggunakan YOLOv8 untuk deteksi plat sebelum pengenalan karakter. Penelitian oleh Hanif A.R., dkk (2023) mengembangkan sistem pengenalan plat nomor menggunakan OCR dengan *Tesseract*-OCR, mencapai keberhasilan 93.3% pada jarak 150 cm [9]. Keterbatasannya adalah sensitivitas terhadap pencahayaan dan kualitas citra. Penelitian saat ini menggunakan teknik segmentasi dan CNN dengan deteksi awal oleh YOLOv8. Penelitian oleh Saputra R.A., dkk (2022) mengevaluasi efektivitas

metode deteksi tepi dalam segmentasi plat nomor, dengan metode *Canny* paling unggul [10]. Keterbatasannya termasuk sampel terbatas dan kondisi pencahayaan spesifik. Penelitian saat ini menggunakan teknik segmentasi dan CNN dengan *dataset* lokal, dengan pendeteksian awal oleh YOLOv8.

Tujuan dari penelitian ini adalah untuk mengembangkan sistem ANPR yang akurat dan efisien di lingkungan kampus UTS, dengan kontribusi sebagai berikut: (1) Pengembangan proses *Image Processing*, khususnya teknik segmentasi dinamis untuk mengatasi kontur tidak teratur dan kondisi pencahayaan berubah-ubah. Segmentasi ini akan memungkinkan pemisahan karakter dari latar belakang dengan lebih jelas, meningkatkan akurasi pemrosesan gambar. (2) Penggunaan algoritma *Convolutional Neural Network* (CNN) untuk meningkatkan keandalan pengenalan karakter pada plat nomor kendaraan. CNN akan dikombinasikan dengan teknik segmentasi untuk meningkatkan efisiensi dan mengurangi tingkat kesalahan dalam kondisi kompleks. Jaringan ini akan dilatih menggunakan *dataset* yang sesuai dengan berbagai jenis plat dan lingkungan. (3) Integrasi YOLOv8, teknik segmentasi responsif, dan CNN diharapkan meningkatkan akurasi dan kecepatan pengenalan plat nomor kendaraan, mempercepat proses identifikasi, serta meningkatkan keamanan dan efisiensi pengelolaan keluar-masuk kendaraan.

2 Metodologi Penelitian

Penelitian ini dilaksanakan melalui serangkaian tahapan sistematis. Metode yang diterapkan yaitu teknik *Image Processing* [11], YOLOv8 [12], dan CNN [13] guna mengimplementasikan sistem *Automatic Number Plate Recognition* (ANPR) [14]. Proses dimulai dengan studi literatur untuk mengumpulkan informasi teoritis yang relevan.

Tahap pengumpulan data melibatkan wawancara, observasi, dan pencarian *dataset*. Wawancara dilakukan dengan petugas keamanan dan pengelola parkir di kampus UTS untuk mengidentifikasi kebutuhan sistem pemantauan kendaraan yang efisien dan tantangan dalam proses manajemen kendaraan. Observasi dilakukan di area gerbang masuk kampus selama tiga hari kerja untuk mencatat volume kendaraan, jenis plat nomor, serta kondisi pencahayaan lingkungan secara aktual.

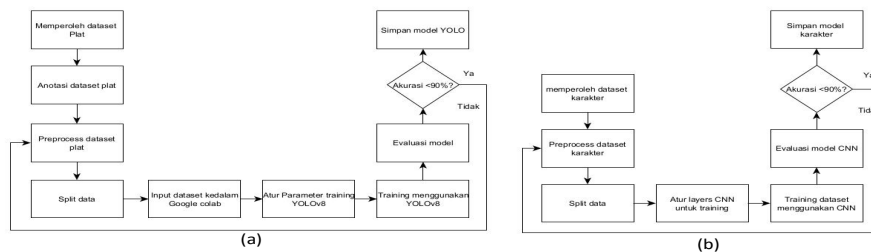
Dataset kemudian dibagi menjadi dua bagian utama: (1) sebanyak 326 gambar plat kendaraan roda dua yang diperoleh dari Kaggle dan dianotasi secara manual menggunakan *Roboflow*, dan (2) sebanyak 37.234 gambar karakter alfanumerik (A–Z, 0–9) dalam resolusi 40×40 piksel, yang juga diambil dari Kaggle dan dikelompokkan ke dalam 36 folder label untuk pelatihan model CNN.

YOLOv8 dilatih dengan data plat untuk mendeteksi objek plat nomor, sementara CNN dilatih dengan data karakter untuk pengenalan pola karakter. Kedua model tersebut disimpan dan digunakan dalam perancangan program. Setelah proses

perancangan aplikasi selesai, sistem diuji dan dievaluasi untuk menilai efektivitas dan performanya dalam kondisi nyata.

2.1 Alur Proses YOLOv8 dan CNN

Dalam penelitian ini, pengembangan dan evaluasi model YOLOv8 dan *Convolutional Neural Network* (CNN) merupakan langkah krusial untuk meningkatkan sistem ANPR. Berikut merupakan alur proses YOLOv8 dan CNN yang dapat dilihat pada Gambar 1 (a) dan (b).

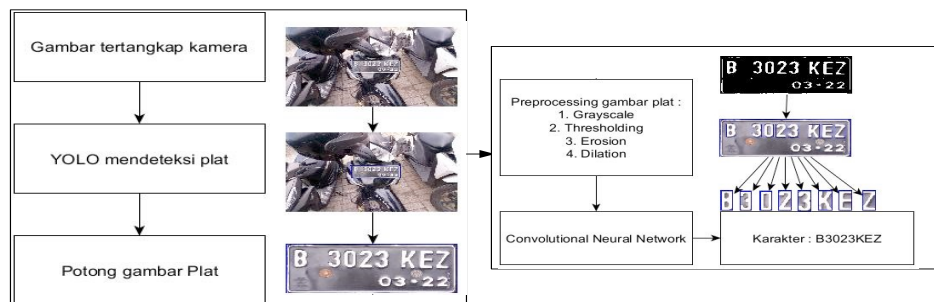


Gambar 1. (a) Alur Proses YOLOv8 (b) Alur Proses CNN

Alur Proses YOLOv8 dimulai dengan memperoleh *dataset* plat kendaraan, diikuti anotasi menggunakan *Roboflow* [15]. Setelah anotasi, dilakukan *preprocessing* untuk menyiapkan data, termasuk normalisasi dan pengaturan skala. Data dibagi menjadi set pelatihan dan pengujian, kemudian diimpor ke *Google Colab* untuk pelatihan model menggunakan sumber daya komputasi tinggi. Evaluasi dilakukan untuk menilai akurasi deteksi. Jika akurasi kurang dari 90%, model dievaluasi dan dilatih ulang sampai mencapai tingkat akurasi yang diharapkan. Setelah mencapai target akurasi, model YOLOv8 disimpan dan diintegrasikan ke dalam sistem ANPR untuk diuji dalam skenario nyata. Sedangkan Alur Proses CNN dimulai dengan memperoleh *dataset* karakter untuk mengidentifikasi setiap karakter pada plat kendaraan. Setelah *dataset* diperoleh, dilakukan *preprocessing* seperti normalisasi dan penyesuaian ukuran. *Dataset* dibagi menjadi data latih dan data uji. Proses *Training* dilakukan dengan mengatur parameter dan lapisan pada CNN, termasuk pengaturan *hyperparameter*. Setelah model dilatih, evaluasi dilakukan untuk mengukur akurasi pengenalan karakter. Jika akurasi kurang dari 90%, model dievaluasi ulang dan dilatih kembali hingga mencapai akurasi yang diharapkan. Setelah mencapai akurasi yang memuaskan, model disimpan dan diimplementasikan dalam sistem ANPR untuk pengujian dan penerapan di lapangan.

2.2 Perancangan Program (*Image Processing*)

Perancangan program (*Image Processing*) atau alur proses aplikasi dalam penelitian ini mencakup pengembangan sistem *Image Processing* terintegrasi yang menggunakan teknologi YOLOv8 dan CNN untuk deteksi dan pengenalan plat nomor kendaraan secara efisien dan akurat. Tahapannya diilustrasikan pada Gambar 2.



Gambar 2. Alur Proses Aplikasi

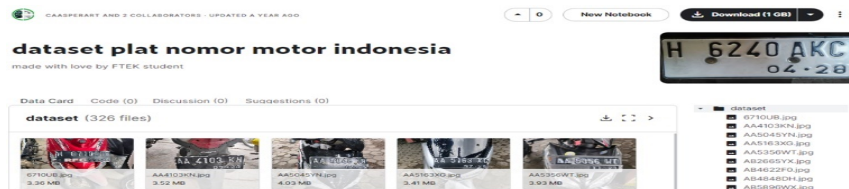
Alur kerja aplikasi dimulai saat gambar kendaraan tertangkap oleh kamera. Gambar ini diproses oleh model YOLOv8 yang terlatih untuk mendeteksi lokasi plat kendaraan dalam gambar. Setelah plat berhasil dideteksi, gambar plat tersebut dipotong dari gambar utama untuk diproses lebih lanjut. Proses *preprocessing* dilakukan pada gambar plat yang telah dipotong, meliputi konversi ke grayscale, penerapan thresholding untuk memperjelas kontras antara karakter dan latar belakang, serta erosi dan dilatasi untuk mengurangi noise dan mempertajam karakter pada plat. Gambar yang telah diproses ini kemudian diserahkan ke model CNN yang dioptimalkan untuk pengenalan karakter. CNN mengenali setiap karakter pada plat, menghasilkan *string* teks yang mewakili nomor plat tersebut. Setiap karakter yang diidentifikasi oleh CNN diurai dan dikompilasi menjadi data teks yang komprehensif.

3 Hasil dan Pembahasan

3.1 Pengumpulan *Dataset* Plat untuk YOLOv8

Pada tahap ini, penulis mengumpulkan dan mempersiapkan *dataset* untuk melatih model YOLOv8 dalam mendeteksi plat nomor kendaraan. Penulis mencari *dataset* yang relevan dari berbagai sumber publik, terutama *Kaggle* [16], dan memilih *dataset* “*Dataset Plat Nomor Motor Indonesia*” karena sesuai dengan tujuan penelitian, adapun *dataset* yang digunakan bisa dilihat di tautan: <https://www.Kaggle.com/datasets/caasperart/haarcascadePlatenumber>. *Dataset* ini mencakup 326 gambar plat nomor motor Indonesia dengan format dan font

spesifik, serta warna dasar hitam dengan tulisan putih atau sebaliknya, penting untuk meningkatkan akurasi dan keandalan model. Format JPG dipilih untuk keseimbangan kualitas gambar dan ukuran *file*, mempermudah penyimpanan dan pemrosesan data. Setelah mengunduh *dataset* berukuran 1.01 GB. Tampilan *Kaggle* sumber *dataset* plat dapat dilihat pada Gambar 3 dibawah ini.



Gambar 3. Sumber *Dataset* Plat

3.2 Proses Anotasi

Setelah mengumpulkan *dataset*, langkah berikutnya adalah melakukan anotasi untuk memastikan model YOLOv8 dapat mendeteksi plat nomor dengan akurasi tinggi. Hal ini perlu dilakukan karena *dataset* belum terlabel. Penulis menggunakan *Roboflow*, sebuah platform yang membantu menyederhanakan pembuatan model penglihatan komputer dengan menyediakan alat untuk mengorganisir, menganotasi, dan mengoptimalkan *dataset* gambar. Anotasi menggunakan *Bounding Box* Tool untuk gambar dengan posisi ideal dan Polygon Tool untuk gambar dengan sudut tidak ideal. Setelah anotasi selesai, gambar yang telah dianotasi ditambahkan ke *dataset* dan dibuat versi baru dengan pengaturan *preprocessing* yang sesuai seperti *Auto Orient* dan *Resize Fit within 640x640*. *Dataset* dibagi menjadi 70% untuk pelatihan, 20% untuk validasi, dan 10% untuk pengujian. Adapun annotation *Roboflow* dapat di akses melalui tautan sebagai berikut: Universe: <https://universe.Roboflow.com/myplat/indonesian-Plate-model/dataset/4> dan Project: <https://app.Roboflow.com/myplat/indonesian-Plate-model/4>.

3.3 Training dan Model YOLOv8

Setelah proses anotasi *dataset* selesai, langkah selanjutnya adalah melakukan pelatihan model YOLOv8 menggunakan *dataset* yang telah dianotasi. Pelatihan ini dilakukan di *Google Colab* yang menyediakan lingkungan dengan dukungan GPU, memungkinkan proses berjalan lebih cepat dan efisien. Jenis runtime GPU T4 digunakan untuk menangani beban komputasi berat. Kode *API key* dari *Roboflow* digunakan untuk mengimpor *dataset* yang telah dianotasi ke *Google Colab*. Selain itu, *ultralytics* digunakan untuk menjalankan YOLOv8. Setelah semua persiapan selesai, penulis menjalankan perintah pelatihan untuk model YOLOv8 selama 100 *Epochs*, menggunakan parameter seperti ukuran gambar

dan ukuran *Batch* yang telah disesuaikan. Adapun perintah *Training* nya dapat di lihat pada Gambar 4.

```
!yolo task=detect mode=train epochs=100 data=/content/Plate_YOLOv8_model-4/data.yaml model=yolov8n.pt imgsz=640 batch=8
```

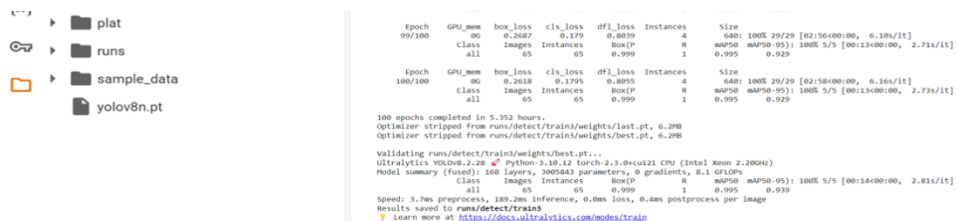
Gambar 4. Perintah *Training* YOLOv8

Adapun keterangan dari *hyperparameter* yang digunakan pada YOLOv8 dapat di lihat pada tabel 1 dibawah ini.

Tabel 1. *Hyperparameter* YOLOv8

No	Hyperparameter	Keterangan
1	<i>Task=detect</i>	Tugas yang akan dilakukan YOLO
2	<i>Mode=train</i>	Mode yang digunakan YOLO
3	<i>Epoch=100</i>	Iterasi latihan model YOLO
4	<i>Data=data.yaml</i>	Lokasi <i>file</i> konfigurasi path data untuk <i>Training</i>
5	<i>Model=yolov8n.pt</i>	Model awal yang digunakan untuk <i>transfer learning</i>
6	<i>Imgz=640</i>	Ukuran input gambar yang sudah disesuaikan di <i>Roboflow</i> sebelumnya
7	<i>Batch=8</i>	Sampel data yang akan dilatih sekaligus
8	<i>Patience=128</i>	Batas <i>Epoch</i> untuk berhenti <i>Training</i> jika tidak ada perubahan

Selama proses pelatihan, penulis memantau metrik seperti *Loss* dan *mAP* (*mean Average Precision*) untuk mengevaluasi kinerja model. Setelah pelatihan selesai, hasilnya disimpan di *folder* "runs", yang berisi model yang telah dilatih, *log* pelatihan, dan visualisasi kinerja model. Model divalidasi dengan *mean Average Precision* (*mAP*) mencapai 0.995 pada *IoU threshold* 0.50 dan 0.939 pada *threshold* 0.50-0.95, untuk processing timenya 100 *Epoch* dengan mencapai 5,352 jam, *Batch* berisi 65 gambar dan *instance*, dengan kecepatan proses rata-rata *preprocessing* 3.7ms, *inference* 189.2ms. adapun proses *Training* model YOLOv8 bisa di lihat pda Gambar 5 dibawah ini.



```

Epoch      GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
99/100      90        0.2687    0.179     0.8039     4          640: 100% 29/29 [02:56:00:00, 6.10s/it]
Class      Images  Instances  Box(P      mAP50  mAP50-95)
all         65        65         0.999     1      0.995    0.929

Epoch      GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
100/100     90        0.2618    0.1795    0.8055     4          640: 100% 29/29 [02:58:00:00, 6.16s/it]
Class      Images  Instances  Box(P      mAP50  mAP50-95)
all         65        65         0.999     1      0.995    0.929

100 epochs completed in 5.352 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.29B
Optimizer stripped from runs/detect/train/weights/best.pt, 6.29B

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.2.28 a Python 3.10.12 torch 2.3.0+cu121 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 168 layers, 3005843 parameters, 0 gradients, 8.1 GiB
Class      Images  Instances  Box(P      mAP50  mAP50-95)
all         65        65         0.999     1      0.995    0.929
Speed: 3.7ms preprocess, 189.2ms inference, 0.4ms loss, 0.4ms postprocess per image
Results saved to runs/detect/train
Learn more at https://docs.ultralytics.com/modes/train

```

Gambar 5. Proses *Training* Model YOLOv8 sampai selesai

3.4 Hasil dari YOLOv8

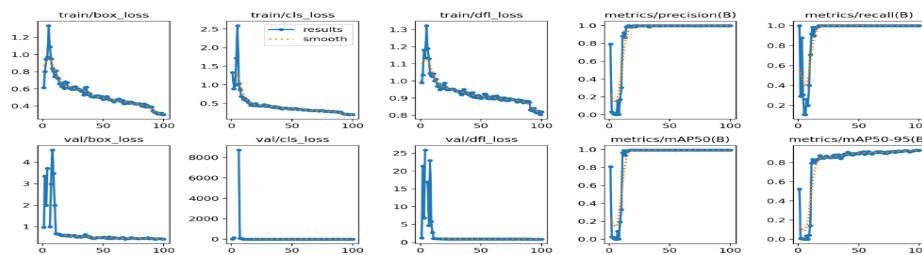
Pada tahap ini, hasil dari model YOLOv8 diuji untuk memastikan kinerjanya dalam mendeteksi plat nomor kendaraan. Pengujian ini mencakup beberapa langkah penting. Pertama, penulis memeriksa ringkasan model di *Google Colab* untuk memahami struktur dan parameter model yang telah dilatih. Model YOLOv8 yang digunakan terdiri dari 225 lapisan dengan 3,011,043 parameter dan memerlukan sekitar 8.2 GFLOPs untuk inferensi, menunjukkan kemampuan komputasi yang tinggi. Ringkasan Model YOLOv8 dapat dilihat pada Gambar 6 dibawah ini.

```
[4] from ultralytics import YOLO
model = YOLO('/content/runs/detect/train/weights/best.pt')
model.info()

Model summary: 225 layers, 3011043 parameters, 0 gradients, 8.2 GFLOPs
(225, 3011043, 0, 8.1941504)
```

Gambar 6. Ringkasan Model YOLOv8

berbagai metrik dan *Loss* selama proses pelatihan dan validasi model YOLOv8 menunjukkan hasil yang sangat baik. *Train/Box Loss* dan *Train/Cls Loss* konsisten menurun hingga stabil pada nilai rendah, menunjukkan perbaikan dalam prediksi *bounding Box* dan klasifikasi. *Train/Dfl Loss* juga menurun dan stabil. *Metrics/Precision* dan *Metrics/Recall* meningkat cepat dan mencapai nilai tinggi mendekati 1.0. *Val/Box Loss*, *Val/Cls Loss*, dan *Val/Dfl Loss* turun signifikan dan stabil selama validasi. *Metrics/mAP50* dan *Metrics/mAP50-95* menunjukkan *mean Average Precision* yang cepat meningkat dan stabil pada nilai tinggi. Secara keseluruhan, hasil ini menunjukkan bahwa model YOLOv8 dilatih dengan baik dan memberikan performa yang sangat baik pada *dataset* validasi, dengan penurunan *Loss* yang signifikan dan metrik evaluasi yang tinggi. Adapun hasil keseluruhan *train* YOLOv8 dapat dilihat pada Gambar 7 dibawah ini.



Gambar 7. Keseluruhan Hasil *Training* YOLOv8

3.5 Pengumpulan *Dataset* Plat untuk CNN

Pada tahap ini, penulis mengumpulkan dan mempersiapkan *dataset* untuk melatih model CNN dalam merekognisi nomor kendaraan. *Dataset* terdiri dari karakter alfanumerik 0-9 dan A-Z dalam format hitam putih dan sebaliknya, setiap gambar

berukuran 40x40 piksel untuk memastikan pemrosesan cepat dan akurasi tinggi. *Dataset* ini diperoleh dari *Kaggle*, dapat di lihat pada tautan <https://www.Kaggle.com/datasets/caasperart/haarcascadePlatenumber>, *Dataset* berisi 37.234 gambar yang dikelompokkan ke dalam 36 label atau folder.

3.6 Training dan Model CNN

Pada tahap ini, penulis menggunakan TensorFlow Keras untuk melatih model *Convolutional Neural Network* (CNN) yang akan digunakan untuk mengenali karakter alfanumerik pada plat nomor kendaraan. *Dataset* gambar karakter alfanumerik diunggah, dan gambar-gambar di-normalisasi ke ukuran 40x40 piksel untuk menjaga konsistensi input. *Dataset* terdiri dari 37.234 gambar yang dikelompokkan ke dalam 36 label. *Dataset* dibagi menjadi 80% untuk pelatihan dan 20% untuk validasi. Model CNN dibuat menggunakan arsitektur Sequential dengan beberapa lapisan *Convolutional*, *MaxPooling*, dan *Dense*. Model dikompilasi dengan *optimizer Adam* dan menggunakan *SparseCategoricalCrossentropy* sebagai *Loss function*. Adapun arsitektur dan *optimizer* model CNN dapat dilihat pada Gambar 8 dibawah ini.

```
num_classes = len(class_names)

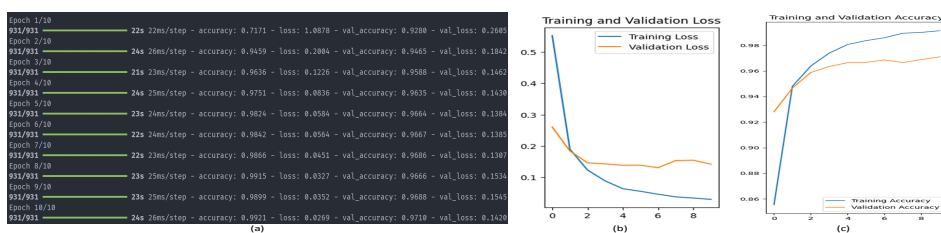
model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

epochs=10
history = model.fit(train_ds, validation_data=val_ds, epochs=epochs)
```

Gambar 8. Arsitektur dan *Optimizer* Model CNN

Proses pelatihan dilakukan selama 10 *Epoch* dengan *Batch* size 32, menunjukkan peningkatan akurasi dan penurunan *Loss* dari *Epoch* ke *Epoch* yang bisa di lihat pada Gambar 9 (a) dibawah ini.



Gambar 9. Hasil *Training* CNN

Grafik Gambar 9 (b) dan (c) menunjukkan perkembangan akurasi selama proses pelatihan dan validasi model. Sumbu X menggambarkan jumlah *Epoch*,

sedangkan sumbu Y menunjukkan akurasi dari 0.86 hingga 0.98. Akurasi pelatihan (garis biru) meningkat dari sekitar 0.88 hingga lebih dari 0.98 dalam 4 *Epoch*, menunjukkan model cepat belajar. Akurasi validasi (garis oranye) naik tajam hingga mendekati 0.94 dalam periode yang sama. Grafik ini menunjukkan model berkinerja baik dalam mempelajari data.

3.7 Pengembangan Program ANPR

Dalam pengembangan program pengenalan plat nomor kendaraan, langkah pertama adalah mengimpor modul-modul penting seperti *OpenCV* untuk pemrosesan gambar, *Ultralytics* untuk algoritma YOLO, *Matplotlib* untuk visualisasi, *TensorFlow* untuk *machine learning*, serta *NumPy* dan *Math* untuk operasi matematika. Data dan model disiapkan dengan mendefinisikan label klasifikasi dan mengimpor data model YOLOv8 dan CNN serta gambar yang akan digunakan. Program memprediksi plat kendaraan menggunakan YOLOv8 dengan mengecilkan gambar dan mencari bagian yang dilabeli sebagai plat. Setelah mendapatkan koordinat *bounding box*, visualisasi *bounding Box* dilakukan pada gambar untuk verifikasi keakuratan deteksi. Hasil dapat dilihat pada Gambar 10 dibawah ini.



Gambar 10. Hasil *bounding Box* YoloV8

Setelah itu gambar dipotong berdasarkan koordinat *bounding box*, dikonversi ke grayscale untuk mengurangi kompleksitas, dan kemudian menjadi biner menggunakan metode *Otsu*. Proses morfologi seperti *erosion* dan *dilation* diterapkan untuk memperhalus dan memperjelas gambar biner. *Contours* karakter pada plat nomor ditemukan menggunakan *OpenCV*, diurutkan, dan diekstraksi berdasarkan kriteria ukuran dan proporsi yang sesuai. Gambar yang dipotong sesuai nilai-nilai *Contour* diubah ukurannya menjadi 40x40 piksel dan dikonversi menjadi array untuk prediksi karakter menggunakan model CNN. Hasil prediksi kemudian divisualisasikan untuk verifikasi, menunjukkan bahwa model deteksi plat menggunakan YOLOv8 memiliki tingkat keyakinan 95%, dan

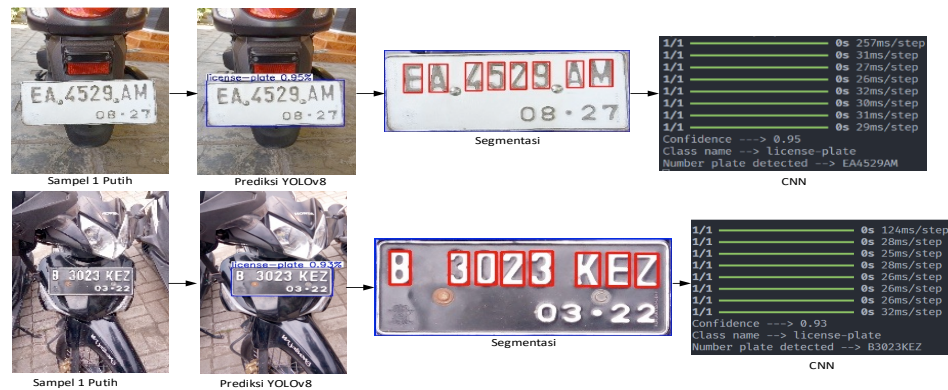
model CNN dapat memprediksi serta membedakan karakter dengan akurasi tinggi seperti yang terlihat pada Gambar 11 dibawah ini.



Gambar 11. Visualisasi Hasil

4 Pengujian dan Evaluasi Aplikasi ANPR

Pengujian menggunakan 10 *dataset* gambar plat hitam dan 10 *dataset* putih melibatkan penilaian dari setiap langkah dalam sistem ANPR, mulai dari berapa persen *confidence* yang didapatkan dari YOLOv8, seberapa banyak segmentasi yang benar berdasarkan dari *Image Processing* yang di terapkan dan dan seberapa banya karakter yang benar berdasarkan algoritma CNN yang di terapkan. Berikut contoh 1 dari 10 sample plat putih dan plat hitam yang dilakukan pengujian yang bisa di lihat pada Gambar 12.



Gambar 12. Sample pengujian plat putih dan hitam

Pada pengujian 10 sample plat putih, segmentasi dan prediksi karakter menggunakan CNN menunjukkan *precision*, *recall*, *F1-score*, dan *accuracy* yang tinggi yang bisa di lihat pada Tabel 2 dibawah ini.

Tabel 2. Rangkuman Rata-Rata Pengujian 10 Sample Plat Hitam dan Putih

Plate	Method	Precision	Recall	F1-score	Accuracy
Plat Putih	Segmentasi	0.9875	0.9857	0.9855	0.9732
Plat Putih	Prediksi	0.9607	0.9857	0.9712	0.9464
Plat Hitam	Segmentasi	0.9764	0.9428	0.9524	0.9192
Plat Hitam	Prediksi	0.975	0.9192	0.939	0.8942

Tabel hasil analisis di atas menunjukkan bahwa performa segmentasi dan prediksi pada plat putih lebih baik dibandingkan plat hitam, yang dipengaruhi oleh kualitas gambar dan kondisi plat. Rata-rata *confidence* deteksi plat menggunakan algoritma YOLOv8 adalah 0.948% untuk plat putih dan 0.94% untuk plat hitam. Hasil ini menegaskan bahwa model pengenalan plat nomor lebih akurat pada plat putih dibandingkan plat hitam. Hasil bisa di lihat pada Tabel 3 dibawah ini

Tabel 3. Rata-Rata *Confidence* Deteksi Pengujian 10 Sample Plat Hitam dan Putih

Sampel	Confidence Plate hitam	Confidence Plate putih
1	0.93%	0.95%
2	0.92%	0.96%
3	0.95%	0.94%
4	0.93%	0.94%

5	0.96%	0.95%
6	0.95%	0.94%
7	0.94%	0.94%
8	0.95%	0.95%
9	0.94%	0.96%
10	0.94%	0.95%
average	0.94%	0.95%

5 Kesimpulan

Penelitian ini berhasil mengembangkan sistem *Automatic Number Plate Recognition* (ANPR) yang terintegrasi menggunakan YOLOv8 untuk deteksi plat, image processing untuk segmentasi karakter, dan CNN untuk pengenalan karakter alfanumerik. Sistem yang dibangun menunjukkan performa tinggi dengan akurasi segmentasi mencapai 97,3% dan akurasi prediksi karakter sebesar 94,6%, melebihi pencapaian penelitian sebelumnya yang hanya mencapai 93,3%. Rata-rata *confidence score* dari YOLOv8 dalam mendeteksi plat kendaraan mencapai 94,8%, membuktikan bahwa pendekatan ini andal dan efisien dalam konteks operasional kampus.

Pengujian lebih lanjut menunjukkan bahwa plat nomor berwarna putih memberikan hasil akurasi yang lebih baik dibandingkan dengan plat hitam. Faktor-faktor seperti pencahayaan, bentuk dan posisi plat, serta sudut pengambilan gambar turut memengaruhi kualitas segmentasi maupun pengenalan karakter. Pada kondisi tertentu, YOLOv8 tetap mampu mendeteksi plat dengan baik, namun model CNN mengalami kesulitan dalam mengenali karakter, terutama ketika jarak pengambilan gambar terlalu jauh atau terdapat noise visual, yang menunjukkan batasan performa sistem dalam skenario ekstrem.

Untuk pengembangan lebih lanjut, disarankan agar penelitian ini memperluas dan memvariasikan *dataset*, khususnya dalam kondisi pencahayaan dan jenis plat yang lebih kompleks. Selain itu, eksplorasi teknik segmentasi lanjutan seperti *adaptive thresholding* atau *edge-preserving filters* dapat meningkatkan akurasi pemrosesan citra. Arsitektur CNN juga perlu dioptimalkan agar lebih tahan terhadap gangguan pencahayaan rendah dan noise, serta antarmuka aplikasi ANPR sebaiknya dikembangkan agar lebih ramah pengguna dan mudah diintegrasikan ke sistem keamanan kampus secara real-time.

6 Referensi

- [1] J. Tang, L. Wan, J. Schooling, P. Zhao, J. Chen, and S. Wei, "Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases," *Cities*, vol. 129, p. 103833, Oct. 2022, doi: 10.1016/j.cities.2022.103833.
- [2] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," *IEEE Access*, vol. 9, pp. 11203–11225, 2021, doi: 10.1109/ACCESS.2020.3047929.
- [3] F. F. Petiwala, V. K. Shukla, V. P. Mishra, and S. Saini, "Smart Parking System through Automation in License Plate Recognition," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, Sep. 2021, pp. 1–7. doi: 10.1109/ICRITO51393.2021.9596554.
- [4] Y. El Hansali, S. Farrag, A. Yasar, and D. Zavantis, "Artificial Intelligence-based Smart Traffic Enforcement and Management System in urban areas," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4227717.
- [5] M. G. Khan, Salma, M. Saeed, A. Zulfiqar, Y. Y. Ghadi, and M. Adnan, "A Novel Deep Learning Based ANPR Pipeline for Vehicle Access Control," *IEEE Access*, vol. 10, pp. 64172–64184, 2022, doi: 10.1109/ACCESS.2022.3183101.
- [6] S. Setiawan, D. N. Sulistyowati, and N. Machmud, "IMPLEMENTATION OF IMAGE PROCESSING IN THE RECOGNITION OF OFFICIAL VEHICLE LICENSE PLATES," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, no. 1, pp. 23–29, Aug. 2023, doi: 10.33480/jitk.v9i1.4181.
- [7] I. H. Al amin and A. Aprilino, "IMPLEMENTASI ALGORITMA YOLO DAN TESSERACT OCR PADA SISTEM DETEKSI PLAT NOMOR OTOMATIS," *Jurnal Teknoinfo*, vol. 16, no. 1, p. 54, Jan. 2022, doi: 10.33365/jti.v16i1.1522.
- [8] M. Zakiyamani, T. I. Cahyani, D. Riana, and S. Hardianti, "Deteksi Dan Pengenalan Plat Karakter Nomor Kendaraan Menggunakan OpenCV Dan Deep Learning Berbasis Python," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 5, no. 1, pp. 56–64, May 2022, doi: 10.31539/intecom.v5i1.3403.
- [9] A. R. Hanif, E. Nasrullah, and F. X. A. Setyawan, "DETEKSI KARAKTER PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN METODE OPTICAL CHARACTER RECOGNITION (OCR)," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 1, Jan. 2023, doi: 10.23960/jitet.v11i1.2897.
- [10] S. Pada Plat Kendaraan Dinas dengan Metode Deteksi Tepi Canny, R. Rizal Adi Saputra, and F. Mimi Wahyuni, "Segmentasi Pada Plat Kendaraan Dinas Menggunakan deteksi Tepi (Reskal) |328," 2022.
- [11] A. Distanto and C. Distanto, *Handbook of Image Processing and Computer Vision*. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-42374-2.
- [12] Z. Huangfu and S. Li, "Lightweight You Only Look Once v8: An Upgraded You Only Look Once v8 Algorithm for Small Object Identification in Unmanned Aerial Vehicle Images," *Applied Sciences*, vol. 13, no. 22, p. 12369, Nov. 2023, doi: 10.3390/app132212369.

- [13] H. Habibi Aghdam and E. Jahani Heravi, *Guide to Convolutional Neural Networks*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-57550-6.
- [14] R. K. Prajapati, Y. Bhardwaj, R. K. Jain, and Dr. Kamal Kant Hiran, “A Review Paper on Automatic Number Plate Recognition using Machine Learning : An In-Depth Analysis of Machine Learning Techniques in Automatic Number Plate Recognition: Opportunities and Limitations,” in *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, IEEE, Apr. 2023, pp. 527–532. doi: 10.1109/CICTN57981.2023.10141318.
- [15] D. Deepa, A. Sivasangari, R. Roonwal, and R. Nayan, “Pothole Detection using Roboflow Convolutional Neural Networks,” in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, May 2023, pp. 560–564. doi: 10.1109/ICICCS56967.2023.10142700.
- [16] H. Das, P. Kumar Sahu, and K. Khanikar, “AI-based license plate detection of on-road vehicle,” in *2023 International Conference on Microwave, Optical, and Communication Engineering (ICMOCE)*, IEEE, May 2023, pp. 1–4. doi: 10.1109/ICMOCE57812.2023.10167115.