

SISTEM PENDETEKSI PLAT NOMOR POLISI KENDARAAN DENGAN ARSITEKTUR YOLOV8

Lejar Satya ¹⁾, M Ridwan Dwi Septian ²⁾, Mochammad Wisuda Sarjono ³⁾, Margi Cahyanti ⁴⁾, dan
Ericks Rachmat Swedia ⁵⁾

^{1,5} Informatika, Universitas Gunadarma

^{2,3,4} Sistem Informasi, Universitas Gunadarma

^{1,2,3,4,5} Jalan Margonda Raya No.100, Pondok Cina, Depok

E-mail : lejarsatya@gmail.com¹⁾, ridwandwiseptian@staff.gunadarma.ac.id²⁾, moch_wisuda@staff.gunadarma.ac.id³⁾,
margi@staff.gunadarma.ac.id⁴⁾, ericks_rs@staff.gunadarma.ac.id⁵⁾,

ABSTRAK

Nomor polisi kendaraan bermotor merupakan ciri atau tanda pengenal dari suatu kendaraan yang diberikan oleh kepolisian. Setiap kendaraan bermotor memiliki nomor yang berbeda-beda. bahkan setiap daerah memiliki kode nomor polisi yang berbeda-beda. Nomor polisi setiap kendaraan biasanya dicantumkan pada sebuah plat. Tanda Nomor Kendaraan Bermotor (TNKB) atau sering disebut plat nomor atau nomor polisi (Nopol) adalah salah satu jenis identifikasi yang diberikan kepada setiap kendaraan bermotor. Pengenalan otomatis plat nomor polisi kendaraan memiliki peran yang sangat penting dalam pengaturan lalu lintas, aspek keamanan, serta pengelolaan transportasi. Kemajuan signifikan dalam bidang ini telah tercapai melalui penggunaan teknik *deep learning*, terutama dengan adopsi arsitektur YOLOv8 (*You Only Look Once*) dalam mendeteksi objek. Penelitian ini difokuskan pada penerapan YOLOv8 dalam sistem pendeteksian plat nomor polisi kendaraan secara otomatis, dengan tujuan meningkatkan ketepatan dalam pengenalan karakter plat nomor polisi dengan menggunakan metode *PaddleOCR*. Dengan memanfaatkan YOLOv8, model telah berhasil dikembangkan dengan kemampuan untuk mengidentifikasi berbagai jenis kendaraan dan plat nomor polisi dengan tingkat akurasi yang signifikan. Skor *precision* sebesar 0,871 menggambarkan akurasi model dalam mengidentifikasi objek, sedangkan nilai *recall* sebesar 0,85 menunjukkan kemampuan model dalam mengambil kembali objek yang relevan. Selain itu, penggunaan *PaddleOCR* dalam proses pengenalan karakter pada plat nomor polisi juga memberikan hasil yang positif, bahkan pada gambar yang mengalami distorsi atau memiliki kualitas gambar yang rendah.

Kata Kunci: Deteksi, Kendaraan, Nomor, Paddleocr, Polisi, YOLO

1. PENDAHULUAN

Ketika menuju suatu tempat untuk memenuhi kebutuhan hidup maka seseorang memerlukan transportasi sebagai alat mobilitas. Alat mobilitas yang biasa digunakan adalah kendaraan. Alat mobilitas biasanya dipilih berdasarkan keamanan, kenyamanan, kecepatan dan kebutuhannya yang sesuai. Menurut Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan pada Bab VII pasal 47, terdiri dari kendaraan bermotor dan kendaraan tidak bermotor (Nugroho, 2021). Kendaraan bermotor dikelompokkan berdasarkan jenisnya yaitu sepeda motor, mobil penumpang, mobil bus, mobil barang dan kendaraan khusus. Perkembangan mode transportasi seperti ini memang sangat memudahkan manusia untuk akses perjalanan. Pada masa modern ini, Sistem transportasi merupakan elemen dasar infrastruktur yang berpengaruh pada pola pengembangan perkotaan maupun pedesaan (Rema, 2019).

Plat nomor polisi yang cukup akrab dengan kita adalah Plat Nomor Indonesia. Negara Indonesia

menganut format tiga nomor seri. Bagian depan mewakili tempat nomor tersebut terdaftar. Kode huruf belakang sendiri biasanya mewakili tempat kendaraan tersebut terdaftar. Nomor polisi sendiri dialokasikan untuk jenis kendaraan-kendaraan tertentu. Kendaraan penumpang menggunakan nomor seri 1-2999, 8000-8999. Nomor diantara seri kendaraan penumpang dibagi untuk motor, kendaraan beban, dan bus. Nomor polisi diberikan sesuai dengan urutan pendaftaran kendaraan bermotor. Terdiri dari 1-4 angka, ditempatkan setelah kode wilayah pendaftaran. Apabila nomor urut pendaftaran yang telah dialokasikan habis digunakan, maka nomor urut pendaftaran berikutnya kembali ke nomor awal yang telah dialokasikan dengan diberi tanda pengenal huruf seri A-Z di belakang angka pendaftaran. Kode wilayah pendaftaran kendaraan bermotor ditetapkan berdasarkan Peraturan Kapolri Nomor 4 Tahun 2006 (Rema, 2019).

Pengenalan plat nomor polisi kendaraan secara otomatis merupakan area penelitian yang penting dalam pengendalian lalu lintas, keamanan, dan manajemen



transportasi. Metode deteksi objek dan pengenalan karakter menggunakan teknik *deep learning* telah menunjukkan kemajuan yang signifikan dalam bidang ini. Salah satu algoritma deteksi objek yang populer adalah YOLO (*You Only Look Once*), yang telah menghasilkan tingkat akurasi yang tinggi dengan kecepatan yang baik. Dalam penelitian ini, fokus utama akan diberikan pada implementasi arsitektur YOLOv8 (versi 8 dari YOLO) pada sistem deteksi plat nomor polisi kendaraan otomatis.

Penelitian yang dilakukan oleh Aprilino, Awan, Dkk yang berjudul "Implementasi Algoritma YOLO dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis" telah memberikan kontribusi yang berharga dalam pengembangan sistem deteksi plat nomor menggunakan arsitektur YOLOv8 dan Tesseract OCR. Penelitian ini bertujuan untuk mengoptimalkan proses deteksi plat nomor polisi kendaraan menggunakan YOLOv8 dan meningkatkan pengenalan karakter plat nomor dengan menggunakan *PaddleOCR* (Aprilino & others, 2022).

Meskipun penelitian sebelumnya telah mencapai hasil yang memuaskan, masih terdapat beberapa tantangan yang perlu diatasi dalam sistem deteksi plat nomor otomatis. Beberapa tantangan tersebut meliputi variasi kondisi cahaya yang ekstrem, variasi bentuk dan ukuran plat nomor, serta adanya perbedaan dalam format plat nomor antar negara atau daerah. Dalam penelitian ini, peneliti berupaya mengatasi tantangan tersebut dengan menggunakan arsitektur YOLOv8 yang telah terbukti efektif dalam mendeteksi objek dengan kecepatan tinggi dan akurasi yang tinggi.

Selain itu, peneliti juga memanfaatkan *PaddleOCR* sebagai metode pengenalan karakter pada plat nomor. *PaddleOCR* telah dikembangkan dan diadopsi secara luas dalam bidang pengenalan karakter optik. Metode ini mampu mengenali karakter-karakter pada plat nomor dengan akurasi yang tinggi, termasuk pada karakter yang terdistorsi atau memiliki kualitas rendah. Dengan mengintegrasikan YOLOv8 dan *PaddleOCR* dalam sistem deteksi plat nomor otomatis, diharapkan dapat meningkatkan akurasi dan keandalan sistem dalam mengenali plat nomor kendaraan dengan baik, bahkan dalam kondisi lingkungan yang bervariasi. Melalui penelitian ini, diharapkan dapat diperoleh sistem deteksi plat nomor kendaraan otomatis yang efektif dan efisien.

2. RUANG LINGKUP

Dalam penelitian ini, ruang lingkup batasan masalah dapat diuraikan menjadi beberapa hal sebagai berikut:

1. Penelitian ini menggunakan 2167 gambar yang diambil dari dua dataset yang tersedia secara publik. Dataset pertama adalah dataset plat nomor kendaraan mobil yang disediakan oleh Irvan Teady Sentosa, dan dataset kedua adalah dataset plat nomor kendaraan motor yang disediakan oleh Habib

Robbani. Dengan 4 Kelas yaitu Plat Nomor, Mobil, Motor dan Truk.

2. Penelitian ini akan menggunakan arsitektur YOLOv8 sebagai algoritma utama untuk deteksi plat nomor kendaraan dan *PaddleOCR* sebagai metode pengenalan karakter pada plat nomor.
3. Penelitian ini akan menggunakan Google Colaboratory sebagai lingkungan pengembangan. Google Colaboratory memberikan akses ke sumber daya komputasi yang cukup untuk melatih model dan melakukan eksperimen.
4. Penelitian ini bertujuan menciptakan sistem untuk mendeteksi dan mengenali plat nomor kendaraan dari gambar menggunakan YOLOv8 dan *PaddleOCR*.

3. BAHAN DAN METODE

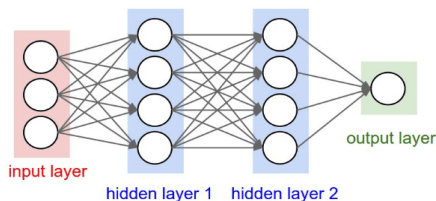
Pada bagian bahan dan metode yang digunakan dalam mendukung penelitian ini adalah sebagai berikut:

3.1 Machine Learning

Machine learning adalah sub bidang dari kecerdasan buatan atau biasa disebut dengan *Artificial Intelligence* (AI) yang melibatkan pengembangan algoritma dan model statistik, yang memungkinkan sistem komputer untuk belajar dan meningkatkan kinerjanya dari pengalaman tanpa harus diprogram secara eksplisit. Ini memungkinkan mesin untuk mengidentifikasi pola, membuat keputusan, dan meningkatkan kinerjanya dalam tugas tertentu dengan terus memproses dan belajar dari data. Proses ini melibatkan melatih model *machine learning* dengan *dataset* besar dan menggunakan pengetahuan tersebut untuk membuat prediksi atau keputusan ketika dihadapkan dengan data baru (Fathurohman, 2021).

3.2 Deep Learning

Deep learning adalah metode khusus dalam *Machine learning* yang menggunakan jaringan syaraf (*neural networks*) dalam lapisan-lapisan berurutan untuk belajar dari data secara iteratif. *Deep learning* sangat berguna ketika mencoba untuk memahami pola dari data yang tidak terstruktur. Jaringan saraf kompleks dalam *deep learning* dirancang untuk meniru cara kerja otak manusia sehingga komputer dapat dilatih untuk menangani abstraksi dan masalah yang kurang terdefinisi dengan baik (Aryanto dkk., 2023).

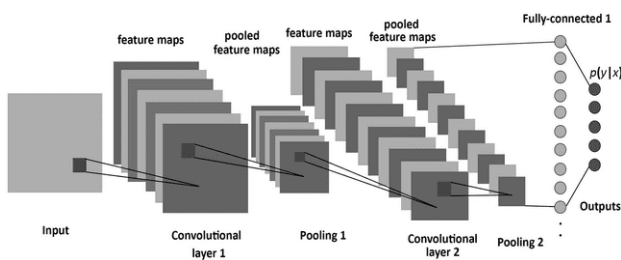


Gambar 1. Arsitektur Neural Network

Jaringan saraf dalam *deep learning* memiliki tiga lapisan utama yaitu Lapisan *input* menerima data masukan, lapisan tersembunyi mengolah dan mengekstraksi fitur-fitur penting, dan lapisan output menghasilkan prediksi atau hasil akhir (Aryanto dkk., 2023).

3.3 Convolutional Neural Network

CNN (*Convolutional Neural Network*) adalah sebuah arsitektur khusus dalam Deep learning yang sangat efektif dalam mengolah data gambar dan tugas-tugas *Computer Vision*. Berbeda dengan *Neural Network* biasa, di mana semua neuron terhubung satu sama lain, CNN menggunakan pendekatan yang lebih efisien dengan menghubungkan setiap neuron hanya pada bagian tertentu dari lapisan sebelumnya (Rasywir dkk., 2020).



Gambar 2. Arsitektur CNN

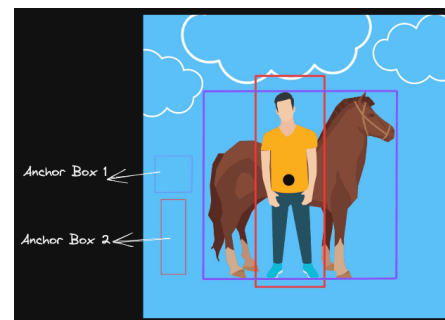
Komponen utama dari CNN adalah lapisan konvolusi, *pooling*, dan *Rectified Linear Unit* (ReLU). Lapisan konvolusi menerapkan filter pada data masukan untuk mengekstraksi fitur-fitur penting. Lapisan pooling melakukan downsampling untuk mengekstraksi fitur-fitur dominan. ReLU adalah fungsi aktivasi *non-linear* yang digunakan untuk memperkenalkan non-linearitas dalam model. Dalam arsitektur CNN, lapisan konvolusi disusun secara berurutan, diikuti oleh lapisan pooling opsional. Arsitektur ini diulang beberapa kali hingga fitur-fitur penting berhasil ditangkap dengan efektif. *Lapisan fully connected* mengikuti lapisan konvolusi, dan fungsi SoftMax digunakan untuk estimasi probabilitas (Christian & Al Idrus, 2023).

3.4 YOLOv8

YOLOv8 adalah model terbaru dari keluarga YOLO yang dikembangkan oleh *Ultralytics*, menciptakan perbedaan signifikan dengan YOLOv5 dalam hal arsitektur dan pengalaman pengembang. Sejak pertama kali diluncurkan pada tahun 2015, model YOLO telah menjadi terkenal karena akurasi tinggi dan ukuran model yang kecil (Yanto dkk., 2023). Dalam perkembangannya, YOLO telah mendapat perhatian dari komunitas visi komputer dan beberapa versi sebelumnya dipelihara dalam bahasa C menggunakan *Darknet*. Pengembangan YOLOv8 dimulai ketika penulisnya, Glenn Jocher, mengikuti YOLOv3 di PyTorch dan

akhirnya meluncurkan YOLOv5 yang menjadi “*State-of-the-Art*” berkat struktur Python yang fleksibel. YOLOv8 merupakan hasil penelitian dan pengembangan selama enam bulan terakhir dan resmi diluncurkan pada Januari 2023 (Wahyuni & Sulaeman, 2022).

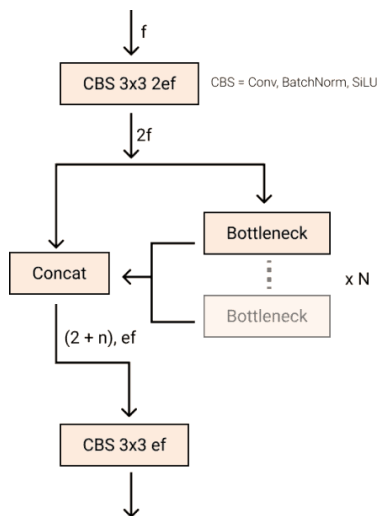
Dalam pendekatan *anchor box free*, model langsung memprediksi pusat objek dari objek yang ada dalam gambar tanpa menggunakan offset dari *anchor box* yang telah ditentukan sebelumnya. Pendekatan ini membantu mengurangi kompleksitas dalam pembuatan prediksi kotak dan dapat meningkatkan kinerja model dengan menghindari masalah yang muncul akibat pemilihan *anchor box* yang tepat untuk setiap objek dalam gambar (Putri, 2023).



Gambar 3. Visualisasi *Anchor boxes free* pada YOLOv8

Model YOLOv8 tidak menggunakan *anchor boxes*, tetapi langsung memprediksi pusat objek. Dengan demikian, model tidak perlu menghitung offset dari *anchor box* untuk mendapatkan posisi objek. Pendekatan ini membantu mengurangi jumlah prediksi kotak (*box predictions*) dan mempercepat proses *Non-Maximum Suppression* (NMS), yaitu tahap pemrosesan lanjutan yang menyaring deteksi kandidat berdasarkan tingkat kepercayaan (*confidence score*) dan menghindari deteksi berlebih (Kim dkk., 2023).

Selain itu, terdapat perbaikan pada struktur konvolusi YOLOv8 yang berdampak positif terhadap performa model. Pada bagian sistem, konvolusi 6x6 YOLOv5 digantikan dengan konvolusi 3x3. Blok pembangun utama juga mengalami perubahan, dengan C2f menggantikan C3 (Zhang, t.t.). Modul ini mengandung komponen-komponen seperti Conv, BatchNorm, dan SiLU (Swish Activation Function). Di C2f, semua *output* dari *Bottleneck* digabungkan, sementara di C3 hanya *output* dari *Bottleneck* terakhir yang digunakan (Niu dkk., 2022).

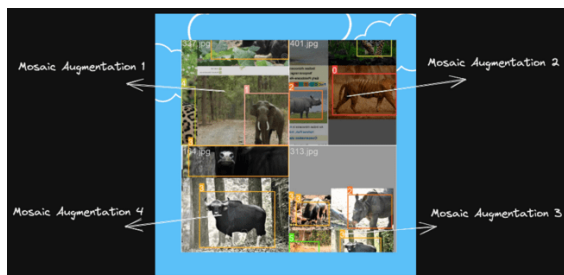


Gambar 4. Convolutional Layer C3 pada YOLOv8

Meskipun *Bottleneck* pada YOLOv8 mirip dengan YOLOv5, namun ukuran kernel konvolusi pertama berubah dari 1x1 menjadi 3x3. Hal ini menunjukkan adanya kembali ke blok *ResNet* yang didefinisikan pada tahun 2015. Di bagian "neck," fitur-fitur langsung digabungkan tanpa memaksa dimensi saluran yang sama, mengurangi jumlah parameter dan ukuran tensor secara keseluruhan (Sama & Sharma, 2023).

Mosaic Data Augmentation adalah teknik augmentasi data dalam pembelajaran mesin yang digunakan dalam pelatihan model deteksi objek, khususnya dalam model YOLOv8. Teknik ini melibatkan penggabungan empat gambar yang berbeda menjadi satu gambar besar, yang kemudian diberikan sebagai input ke model. Gambar-gambar tersebut ditempatkan secara acak dalam ukuran dan posisi yang berbeda, menciptakan efek mozaik pada gambar yang dihasilkan (Pratitis dkk., 2023).

Tujuan dari *Mosaic Data Augmentation* adalah untuk memperkaya dataset pelatihan dengan variasi yang lebih besar dalam posisi, rotasi, dan tumpang tindih objek. Dengan memberikan gambar *mozaik* sebagai *input*, model belajar mengenali objek dalam berbagai posisi dan situasi, termasuk objek yang tumpang tindih sebagian dengan objek lain. Hal ini membantu meningkatkan ketahanan model terhadap variasi dalam data uji yang belum pernah dilihat sebelumnya (Djohari dkk., 2022).



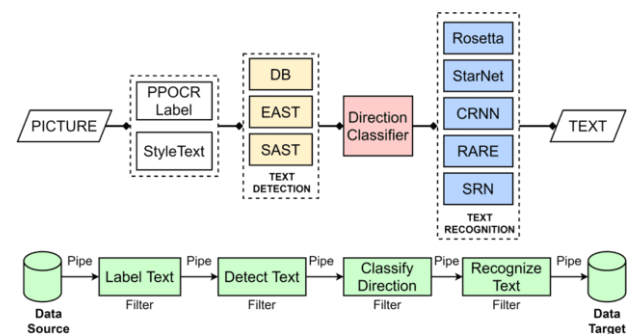
Gambar 5. Mosaic Data Augmentation

Namun, karena augmentasi ini dapat menyebabkan penurunan performa pada tahap pelatihan, dalam YOLOv8, *Mosaic Data Augmentation* dihentikan pada 10 epoch terakhir. Dengan demikian, model tetap mengalami variasi data yang luas selama mayoritas pelatihan, namun pada tahap akhir, model lebih fokus pada data asli untuk menghindari *Overfitting* dan meningkatkan kinerja pada data uji (Ma'ruf & Hardjianto, 2023).

3.5 PaddleOCR

PaddleOCR adalah sebuah sistem *Optical Character Recognition* (OCR) yang sangat efisien dan praktis yang dikembangkan oleh para peneliti dari *Paddle*, perangkat lunak open-source yang dibuat oleh Baidu. Teknologi ini bertujuan untuk mengenali teks dalam gambar secara otomatis. *PaddleOCR* berhasil menciptakan model OCR yang memiliki ukuran sangat ringan tanpa mengorbankan performa. Sistem ini terdiri dari tiga tahap utama, yaitu deteksi teks, klasifikasi arah teks, dan pengenalan teks (Li dkk., 2023).

Dalam deteksi teks, *PaddleOCR* menggunakan metode *Differentiable Binarization* (DB) yang dilengkapi dengan beberapa strategi untuk meningkatkan efektivitas dan efisiensi. Selain itu, mereka juga menggunakan model *MobileNetV3* sebagai backbone dengan dimensi yang lebih kecil untuk mengurangi ukuran model. *PaddleOCR* juga menerapkan strategi pruning dengan *FPGM Pruner* untuk lebih mengompres model deteksi teks. Di tahap klasifikasi arah teks, mereka menggunakan model *MobileNetV3* sebagai *backbone* dan menerapkan teknik *Data Augmentation* serta *PACT Quantization* untuk mengurangi ukuran model tanpa mengorbankan akurasi. Selanjutnya, di tahap pengenalan teks, *PaddleOCR* menggunakan model *CRNN* (*Convolutional Recurrent Neural Network*) sebagai pengenal teks dengan berbagai strategi untuk meningkatkan kinerja dan mengurangi ukuran model. Dengan berbagai pengoptimalan dan strategi tersebut, *PaddleOCR* berhasil menciptakan sistem OCR yang sangat efisien dengan ukuran model yang ringan dan kinerja yang optimal (Liao dkk., 2020).



Gambar 6. Arsitektur PaddleOCR

3.6 Analisis Sistem Secara Umum

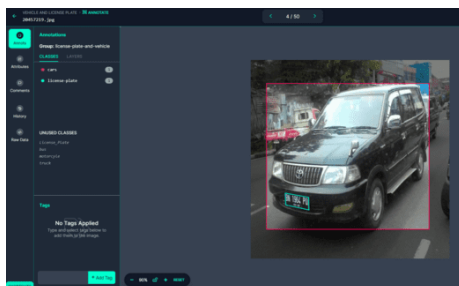
Penelitian ini berfokus kepada implementasi arsitektur YOLOv8 dan *PaddleOCR* untuk mendeteksi plat nomor kendaraan. Hasil dari pendeteksian akan menampilkan gambar dengan label nama kendaraan dan plat nomor. Kemudian dilakukan ekstraksi menjadi teks yang berisikan jenis kendaraan dan plat nomor yang dibagi menjadi beberapa tahapan diantaranya.

3.6.1 Pengumpulan Data

Data untuk penelitian ini dikumpulkan melalui dataset dari *Universe Roboflow* yang diunggah oleh Irvan Tedy Sentosa dan Habib Robbani. Dataset ini berisi dengan total 2,167 gambar yang memiliki 5 label, yaitu plat nomor, mobil, motor, bus, dan truk. Penggabungan data dilakukan menggunakan *website Roboflow*. Dataset yang terkumpul menjadi dasar bagi peneliti untuk melatih dan menguji model pengenalan kendaraan guna mencapai tujuan penelitian yang telah ditetapkan.

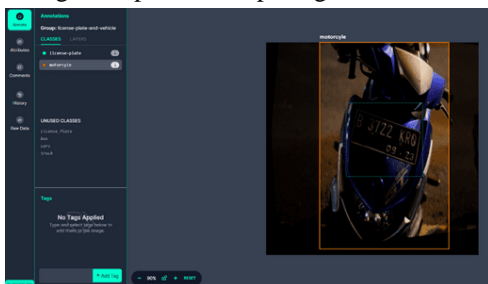
3.6.2 Pelabelan Data

Pelabelan data menggunakan *platform Roboflow*. Pelabelan dilakukan dengan tujuan memberikan label pada setiap gambar kendaraan yang sesuai dengan jenisnya, yaitu “*License_plate*”, “*cars*”, “*motorcycle*”, “*truck*”, dan “*bus*”. Proses pelabelan ini merupakan langkah penting dalam persiapan *dataset* untuk melatih model pengenalan kendaraan dengan akurat dan efisien.



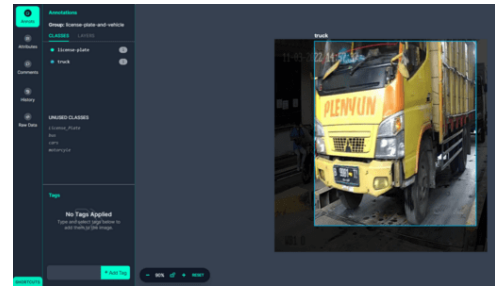
Gambar 7. Pelabelan Mobil dan Plat Nomor

Pelabelan untuk dua jenis objek di dalam dataset, yaitu mobil dan plat nomor. Objek mobil diberi label ‘*cars*’ dengan warna merah, yang ditandai dengan sebuah *bounding box* (kotak pembatas) yang mengelilingi bagian mobil pada gambar. Di sisi lain, objek plat nomor diberi label ‘*License_plate*’ dengan warna hijau dan juga diberi *bounding box* yang mengelilingi area plat nomor pada gambar 7.



Gambar 8. Pelabelan Motor

Objek motor diberi label ‘*motorcycle*’ dengan warna jingga, yang ditandai dengan sebuah *bounding box* (kotak pembatas) yang mengelilingi bagian motor seperti pada gambar 8.

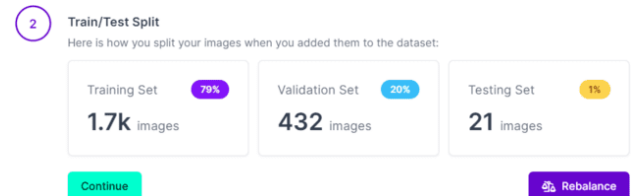


Gambar 9. Pelabelan Truk

Objek truk diberi label ‘*truck*’ dengan warna biru, yang ditandai dengan sebuah *bounding box* (kotak pembatas) yang mengelilingi bagian truk pada gambar.

3.6.3 Data Splitting

Proses data *splitting* ini dilakukan secara acak untuk menghindari bias pada setiap subset. Selain itu, memastikan bahwa data pada setiap subset mewakili variasi dan karakteristik dataset secara keseluruhan.



Gambar 10. Proses Data Splitting

Pada tahap *Data Splitting*, dataset telah dibagi menjadi tiga subset yang berbeda, yaitu *Training Set*, *Validation Set*, dan *Testing Set*. *Training Set* merupakan subset terbesar dengan proporsi sekitar 79% dari total data, yang terdiri dari 1,700 gambar. Subset ini berfungsi sebagai data pelatihan untuk melatih model pengenalan kendaraan. Saat proses pelatihan, model akan belajar mengenali pola dan fitur yang ada pada kendaraan dari data yang ada dalam *Training Set*.

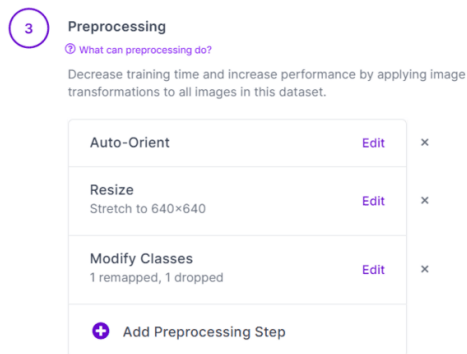
Selanjutnya, *Validation Set* menjadi subset kedua dengan proporsi sekitar 20% dari total data, yang terdiri dari 432 gambar. Subset ini memiliki peran penting dalam validasi model, di mana kinerja dan performa model selama pelatihan akan diukur. Dengan menggunakan *Validation Set*, dapat diidentifikasi apakah model mengalami *Overfitting* atau *Underfitting*.

Sementara itu, *Testing Set* menjadi subset terkecil dengan proporsi sekitar 1% dari total data, yang terdiri dari 21 gambar. *Subset* ini digunakan untuk menguji performa model secara independen setelah melalui proses pelatihan dengan menggunakan *Training Set*. Pengujian pada *Testing Set* memberikan gambaran

tentang kemampuan model dalam mengenali kendaraan yang belum pernah dilihat sebelumnya.

3.6.4 Data Preprocessing

Data *preprocessing* merupakan tahap krusial dalam persiapan *dataset* sebelum melatih model pengenalan kendaraan. Dalam penelitian ini, data *preprocessing* dilakukan menggunakan beberapa teknik, yaitu *Auto-Orient*, *Resize*, dan *Modify Classes*.



Gambar 11. Proses Data Preprocessing

Pada Gambar 11 teknik *Auto-Orient* digunakan untuk mengatur ulang orientasi gambar kendaraan agar seragam dalam dataset. Hal ini penting karena variasi orientasi dapat menyebabkan kesulitan dalam pelatihan model. Selanjutnya, proses *Resize* dilakukan untuk mengubah ukuran semua gambar menjadi 640x640 piksel, memastikan ukuran yang seragam dalam seluruh dataset. Dengan ukuran yang konsisten, model dapat lebih efisien dalam mempelajari pola dan fitur kendaraan dari gambar-gambar tersebut.

Teknik *Modify Classes* diterapkan untuk menghapus kelas bus pada dataset, karena jumlah gambar dan label yang sangat sedikit sehingga kurang representatif dari kelas tersebut, bertujuan meningkatkan fokus dataset pada informasi yang relevan. Data Preprocessing ini menjadi langkah penting dalam meningkatkan kualitas dataset dan kinerja model pengenalan objek.

3.6.5 Training Model

Pada tahap pelatihan model dalam proses pembuatan, langkah pertama adalah mempersiapkan dataset yang sudah dibuat pada aplikasi *Roboflow*. Setelah dataset terbentuk, langkah selanjutnya adalah melatih model menggunakan Transfer Learning dengan menggunakan pretrained model YOLOv8n. Transfer learning memungkinkan kita memanfaatkan pengetahuan yang telah diperoleh oleh model pada tugas-tugas sebelumnya.

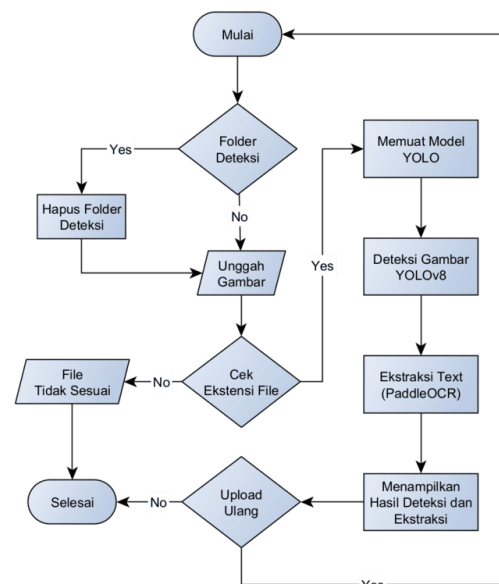
Proses pelatihan dimulai dengan menginisialisasi model YOLOv8 menggunakan bobot yang sudah dilatih sebelumnya. Dataset yang digunakan adalah data *training* dan data *validasi*. *Optimizer* "AdamW" digunakan dalam proses ini, dengan parameter yang telah

ditentukan sebelumnya. *Learning rate* 0.01 mengatur kecepatan penyesuaian bobot model. Momentum dan *weight decay* mengontrol dinamika dan regulasi selama proses optimasi. Proses pelatihan dilakukan dalam sejumlah iterasi yang disebut *epochs*. Setiap *epochs*, model disesuaikan dengan data pelatihan dan kemudian diuji pada data validasi untuk mengukur kinerjanya. Jika kinerja model cenderung stagnan atau menurun pada data validasi, parameter seperti learning rate atau momentum dapat diubah untuk meningkatkan kinerja.

Proses pelatihan berlangsung secara berulang, di mana model terus diperbarui dan disesuaikan dengan dataset pelatihan. Setelah sejumlah telah tercapai, pelatihan berakhir. Pada akhirnya, model yang telah dilatih akan memiliki pemahaman yang lebih baik tentang tugas deteksi objek dan kemampuan untuk mengenali objek-objek mobil, motor, truk dan plat nomor.

3.6.6 Implementasi Model

Untuk implementasi model, model akan di *deploy* ke dalam *environment project* aplikasi *Flask*. Dalam penelitian ini model akan disimpan dalam format *Pytorch*. Berikut adalah alur perancangan sistem yang dibuat.



Gambar 12. Alur Sistem

Untuk implementasi model, model akan di *deploy* ke dalam *environment project* aplikasi *Flask*. Dalam penelitian ini model akan di *save* dalam format *Pytorch*. Berikut adalah alur perancangan sistem yang dibuat pada gambar 12.

Pada saat mengunjungi halaman utama ("/"), *file-file* sebelumnya dan direktori hasil deteksi dibersihkan melalui fungsi "*delete_previous_uploads()*" untuk memastikan kondisi kerja yang bersih. Kemudian menampilkan halaman "*index.html*" untuk mengunggah file gambar.

Setelah berhasil mengunggah file gambar, fungsi "upload()" diaktifkan. File tersebut diperiksa, dan jika memiliki ekstensi yang diizinkan, nama unik dibuat melalui "generate_random_filename()", dan file disimpan dalam direktori "uploads". Kemudian, dilakukan pengecekan apakah file tersebut adalah gambar dengan memanfaatkan OpenCV. Jika file adalah gambar yang didukung, proses selanjutnya adalah deteksi plat nomor. Melalui fungsi "detect_platnomer(filepath)", model YOLO ("platnomor.pt") dimuat dan proses deteksi objek dilakukan pada gambar. Hasil dari tahap ini adalah lokasi plat nomor.

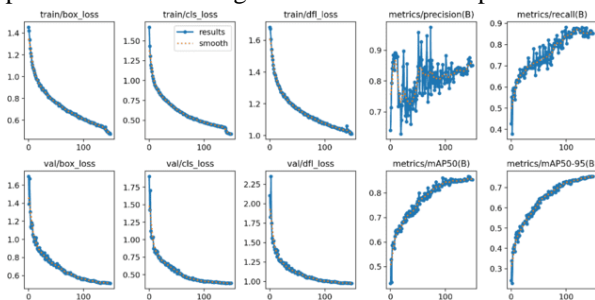
Setelah berhasil mendeteksi plat nomor, dilakukan ekstraksi karakter menggunakan PaddleOCR. Area plat nomor pada gambar diambil dan diubah menjadi format yang sesuai dengan input yang diminta oleh PaddleOCR. Kemudian, gambar yang telah diubah dijalankan melalui PaddleOCR untuk mengenali karakter optik pada plat nomor. Hasil pengenalan karakter diekstraksi dan ditampilkan dalam bentuk teks.

Hasil akhir dari proses ini adalah tampilan halaman "uploaded.html" yang memberikan informasi terkait Gambar yang diupload, hasil deteksi plat nomor, serta teks yang berhasil diekstraksi dari nomor plat tersebut.

4. PEMBAHASAN

Pada bagian ini merupakan pembahasan implementasi dan hasil pengujian keseluruhan pada penelitian.

Selama proses pelatihan, model mengalami perubahan berdasarkan metrik yang diukur dalam 150 epoch. Berikut adalah gambar 13 metrik dari pelatihan.



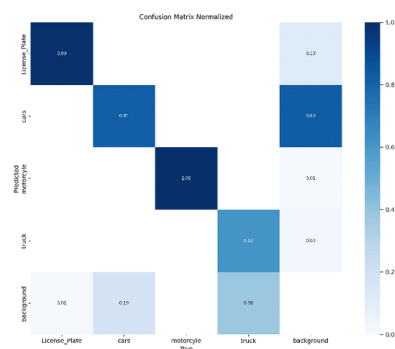
Gambar 13. Hasil Training Model

Pada pelatihan model nilai loss dari model terus berkurang, "box_loss" untuk mengukur seberapa besar perbedaan antara prediksi model dan lokasi objek pada gambar asli dengan nilai akhir 0,47. "cls_loss" digunakan untuk mengukur kesalahan model dalam mendeteksi objek dengan nilai akhir 0,32. Pada tahap validasi berdasarkan pada gambar 4.1, hasil yang didapat tidak jauh berbeda dengan train.

Skor presisi sebesar 0,871 menggambarkan seberapa akurat model dalam mengenali objek, sedangkan nilai recall sebesar 0,85 mengindikasikan sejauh mana model mampu mengambil kembali objek yang relevan. Di sisi lain, nilai mAP50 sebesar 0,868 menggambarkan

performa keseluruhan model dengan menggunakan ambang batas IOU 50%, sedangkan nilai mAP50-95 sebesar 0,753 lebih menekankan pada kemampuan model dalam skenario deteksi yang lebih ketat, dengan rentang IOU 50% hingga 95%.

Pada gambar 14 Multiclass Confusion Matrix mendapatkan hasil yang cukup baik. Kelas motor dan plat kendaraan mendapatkan nilai True Positive (TP) yang tinggi, yaitu 0,99 dan 1. Kelas mobil mendapatkan True Positive (TP) dengan nilai 0,83 tetapi False Negative (FP) memiliki nilai 0,83. Nilai False Negative yang tinggi mengakibatkan model salah mendeteksi kelasnya. Kelas Truk mendapatkan nilai True Positive (TP) 0,62.



Gambar 14. Multiclass Confusion Matrix

Pada gambar 15 melakukan proses pemilihan gambar tahapan selanjutnya mendeteksi objek dan mendapatkan nomor polisi kendaraan tersebut.









Gambar 15. Upload Web Flask

4.1 Hasil Uji Coba Keseluruhan

Berdasarkan data yang tercatat dalam Tabel 1 hasil pengujian, dapat disimpulkan bahwa kinerja model dalam mengenali objek umumnya berjalan dengan baik. Meskipun demikian, terdapat dua kasus pada tabel di bawah dengan nomor 1 dan 2 di mana model tidak berhasil mengenali objek Truk dengan akurat. Meskipun begitu, pengujian pada nomor selanjutnya menunjukkan bahwa model memiliki kemampuan yang memadai dalam mengenali karakter-karakter dengan akurasi yang baik. Berikut hasil tabel uji coba yang dilakukan.



Tabel 1. Hasil Uji Coba

#	Gambar	Objek	Objek	Hasil ORC	Plat
			Asli		Nopol
1.		Plat	Plat Truk	DA 8963CO	DA 8963 CO
2.		Plat	Plat	B2359 BJ	B 2359
3.		Mobil	Mobil	01-27	BJD
4.		Plat	Plat	AE1248FP	AE 1248
5.		Mobil	Mobil	11.24	FP
6.		Plat	Plat	BP1475T1	BP 1475
7.		Mobil	Mobil	TI	TI
8.		Plat	Plat	DK 386	DK 386
9.		Mobil	Mobil	QZ	QZ
10.		Plat	Plat	1074TUA	B 1074
11.		Mobil	Mobil	0020	TUA
12.		Plat	Plat	B	B 1970
13.		Mobil	Mobil	1907SSW 05-21	SSW
14.		Plat	Plat Truk	B9190TC	B 9190 IC
15.		Plat	Plat Motor	BL2271ZE	BL 2271 ZE
16.		Plat	Plat Motor	BL6463Z AH 0721	BL 6463 ZAH

5. KESIMPULAN

YOLOv8 berhasil menghasilkan model yang mampu mendeteksi berbagai jenis kendaraan dan plat nomor dengan tingkat akurasi yang memadai. Skor *precision* sebesar 0,871 dan nilai *recall* sebesar 0,85. Penggunaan *PaddleOCR* sebagai metode pengenalan karakter pada plat nomor juga memberikan hasil yang baik dalam mengenali karakter-karakter pada plat nomor polisi, termasuk pada karakter yang terdistorsi atau memiliki kualitas gambar yang rendah. Pada uji coba menggunakan berbagai gambar, sistem mampu mengenali objek kendaraan dan plat nomor polisi dengan baik pada sebagian besar kasus. Meskipun demikian, masih terdapat beberapa kasus di mana model mengalami kesulitan dalam mengenali objek, terutama pada kelas Truk dan karakter tanggal pada plat nomor polisi. Ini menunjukkan bahwa masih ada ruang untuk peningkatan dalam pengenalan karakter yang lebih akurat.

6. SARAN

Penelitian ini telah mencapai hasil sesuai yang diharapkan oleh peneliti. Terdapat beberapa saran untuk pengembangan lebih lanjut diantaranya dapat mendeteksi dan mengenali karakter plat nomor polisi dalam bentuk *video* atau *streaming video*, meningkatkan model deteksi terutama untuk mobil dan truk, dan membuat format plat nomor Indonesia sehingga model dengan cepat dan tepat mengenali karakter-karakter yang tertera.

7. DAFTAR PUSTAKA

- Aprilino, A., & others. (2022). Implementasi Algoritma Yolo dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis. *Jurnal Teknoinfo*, 16(1), 54–59.
- Aryanto, R., Rosid, M. A., & Busono, S. (2023). Penerapan Deep Learning untuk Pengenalan Tulisan Tangan Bahasa Aksara Lota Ende dengan Menggunakan Metode Convolutional Neural Networks. *Jurnal Informasi dan Teknologi*, 258–264.
- Christian, J., & Al Idrus, S. I. (2023). Introduction to Citrus Fruit Ripens Using the Deep Learning Convolutional Neural Network (CNN) Learning Method. *Asian Journal of Applied Education (AJAE)*, 2(3), 459–470.
- Djohari, N., Setiawan, M. N., & Liauw, F. (2022). Penerapan Teknologi Interaktif Display dalam Perancangan Interior Museum Wayang Jakarta. *Mezanin*, 4(2).
- Fathurohman, A. (2021). Machine Learning Untuk Pendidikan: Mengapa Dan Bagaimana. *Jurnal Informatika Dan Teknologi Komputer (JITEK)*, 1(3), 57–62.
- Kim, J.-H., Kim, N., & Won, C. S. (2023). High-Speed Drone Detection Based On Yolo-V8. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–2.
- Li, Y., Zhang, D., & others. (2023). Research and Application of Health Code Recognition Based on Paddle OCR under the Background of Epidemic Prevention and Control. *Journal of Artificial Intelligence Practice*, 6(1), 9–16.
- Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2020). Real-time scene text detection with differentiable binarization. *Proceedings of the AAAI conference on artificial intelligence*, 34(07), 11474–11481.
- Ma'ruf, A., & Hardjianto, M. (2023). PENERAPAN ALGORITME YOU ONLY LOOK ONCE VERSION 8 UNTUK IDENTIFIKASI ABJAD BAHASA ISYARAT INDONESIA. *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, 2(2), 567–576.
- Niu, H., Liu, J., Yu, Z., Zheng, D., He, P., & Wang, F. (2022). Real-time object tracking system using PTZ camera. *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 471–478.
- Nugroho, Y. (2021). Pelanggaran Lalu Lintas Yang Dilakukan Oleh Anak Ditinjau Dari Undang-Undang Nomor 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Jalan. *FENOMENA*, 19(2), 246–267.
- Pratitis, W. L., Kurniasari, K., & Al Fata, H. (2023). Classification of Spotted Disease on Sugarcane Leaf Image Using Convolutional Neural Network

- Algorithm. *JTECS: Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer*, 3(2), 117–128.
- Putri, V. H. (2023). *Detecting Incoming and Outgoing Passengers on Intelligent Car (iCar Its) Using Computer Vision*. Institut Teknologi Sepuluh Nopember.
- Rasywir, E., Sinaga, R., & Pratama, Y. (2020). Evaluasi pembangunan sistem pakar penyakit tanaman sawit dengan metode deep neural network (DNN). *Jurnal media informatika budidarma*, 4(4), 1206–1215.
- Rema, Y. O. L. (2019). Deteksi Plat Nomor Kendaraan Bermotor dengan Segmentasi Gambar. *Jurnal Saintek Lahan Kering*, 2(1), 20–23.
- Sama, A. K., & Sharma, A. (2023). Simulated uav dataset for object detection. *ITM Web of Conferences*, 54, 2006.
- Wahyuni, S., & Sulaeman, M. (2022). Penerapan Algoritma Deep Learning Untuk Sistem Absensi Kehadiran Deteksi Wajah Di PT Karya Komponen Presisi. *Jurnal Informatika SIMANTIK*, 7(1), 12–21.
- Yanto, Y., Aziz, F., & Irmawati, I. (2023). YOLO-V8 PENINGKATAN ALGORITMA UNTUK DETEKSI PEMAKAIAN MASKER WAJAH. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(3), 1437–1444.
- Zhang, D. (t.t.). Yolodcc: Improved Yolov8 Combined with Dynamic Confidence Compensation For Lightweight Moving Object Detection. Available at SSRN 4571142.