

Deteksi Plat Nomor Kendaraan Angkutan Bus Menggunakan YOLOv11

Benni Agung Nugroho¹, Toga Aldila Cinderatama², Abidatul Izzah³, Ellya Nurfarida⁴

Program Studi Manajemen Informatika, PSDKU Polinema Kediri, Jl. Lingkar Maskumambang No.1, Kediri 64119, Indonesia^{1,2,3,4}

benni.agung@polinema.ac.id¹, toga.aldila@polinema.ac.id², abidatul.izzah@polinema.ac.id³, ellya.nurfarida@polinema.ac.id⁴

Abstrak – Penelitian ini mengatasi permasalahan pendataan plat nomor bus secara konvensional di terminal yang rentan terhadap kesalahan manusia, pelaporan yang lambat, dan kurangnya transparansi. Tujuan penelitian ini adalah merancang dan mengimplementasikan sistem deteksi dan pengenalan plat nomor angkutan bus secara otomatis menggunakan kombinasi teknologi Kecerdasan Buatan (AI), khususnya model *deep learning* YOLOv11, dan *Internet of Things* (IoT). Penelitian ini menggunakan Raspberry Pi 5 sebagai perangkat *edge* yang terhubung dengan *webcam* untuk menangkap video beresolusi 1280x720, yang kemudian melakukan deteksi plat nomor secara *real-time*. Model YOLOv11 dilatih menggunakan 682 *frame* gambar, menunjukkan kinerja yang sangat baik pada dataset validasi dengan nilai *mean Average Precision* (mAP50) sebesar 0.98674. Meskipun keterbatasan Raspberry Pi 5 tanpa dedicated *Neural Processing Unit* (NPU) membatasi kecepatan pemrosesan *real-time* menjadi 10–18 FPS pada resolusi 640x480, kecepatan ini dinilai mencukupi untuk deteksi bus di lingkungan terminal karena pergerakan kendaraan yang tidak cepat. Teks plat nomor yang berhasil dikenali dan timestamp kemudian dikirimkan ke database Firebase melalui arsitektur IoT. Sistem yang dihasilkan diharapkan dapat membantu dalam pendataan kendaraan bus di terminal dengan lebih cepat, efisien, dan akuntabel.

Kata Kunci: YOLOv11, Deteksi Plat Nomor, Raspberry Pi 5, IoT, Terminal Bus

I. PENDAHULUAN

Terminal bus merupakan titik strategis dalam sistem transportasi darat yang melayani mobilitas masyarakat secara luas. Di tengah arus kendaraan yang keluar masuk setiap hari, pendataan plat nomor bus menjadi elemen krusial dalam menjaga keteraturan dan keselamatan operasional. Pendataan plat nomor bus merupakan bagian integral dari sistem pengelolaan transportasi publik yang aman, tertib, dan transparan. Tujuan dan manfaat dari pendataan plat nomor angkutan bus diantaranya adalah : 1) pengawasan operasional dan keamanan dimana data plat nomor mendukung pelaksanaan inspeksi keselamatan dan pelacakan bila terjadi pelanggaran atau insiden, dan penjadwalan keberangkatan[1], 2) integrasi data kendaraan dengan sistem manifest penumpang, e-ticketing dan database perizinan operasional kendaraan[2]. Pendataan bus di terminal bus saat ini masih banyak yang menggunakan cara konvensional, yaitu mencatat pada lembaran

kertas dimana kekurangannya diantaranya adalah : 1) rentan terhadap kesalahan manusia dalam pencatatan, 2) tidak realtime dan lambat dalam pelaporan, dan 3) tidak mendukung transparansi dan akuntabilitas sehingga sulit membuktikan keabsahan data bila terjadi sengketa atau audit. Permasalahan tersebut dapat diatasi dengan melibatkan teknologi yang saat ini sedang booming dan berkembang pesat yaitu teknologi kecerdasan buatan dan internet-of-things (IoT).

Kecerdasan buatan memiliki berbagai macam cabang ilmu dimana salah satunya adalah computer vision yang dapat membantu dalam pengenalan obyek didalam gambar, menggali informasi dan mengambil keputusan dan machine learning dimana sistem dapat belajar sepanjang waktu bila diberikan data baru. *Deep learning* merupakan salah satu metode didalam machine learning yang dapat memberikan hasil yang memuaskan di bidang computer vision, terutama dalam hal deteksi dan pengenalan obyek dengan menyediakan neural network yang canggih[3].

YOLO (*You Only Look Once*) merupakan model *deep learning* berbasis Convolutional Neural Network (CNN) dengan kemampuan analisis gambar secara real time. YOLO memberikan hasil pengenalan obyek yang sangat baik ketika diuji pada dataset COCO dan dataset VOC[4]. CNN menggunakan konvolusi yang mengacu pada struktur jaringan syaraf yang memproses data melalui operasi konvolusi sebelum diteruskan ke lapisan jaringan syaraf selanjutnya [5]. Menurut Ultralytics, YOLO v11 memberikan berbagai peningkatan yang cukup signifikan dibanding pendahulunya yang diantaranya adalah pemrosesan yang lebih cepat dengan tetap mempertahankan keseimbangan antara akurasi dan performa, akurasi yang lebih tinggi dengan lebih sedikit parameter sehingga komputasinya lebih efisien tanpa mengorbankan akurasi[6].

IoT (*Internet of Things*) adalah jaringan dari perangkat fisik yang didalamnya terdapat sensor, *software*, dan konektivitas jaringan yang memungkinkan antar perangkat untuk mengumpulkan dan saling berbagi data. YOLO adalah algoritma *object detection* yang sangat cepat dan efisien, mampu mendeteksi obyek dalam sekali proses (*single pass*) tanpa perlu pemrosesan berulang sehingga memungkinkan YOLO untuk dijalankan pada perangkat IoT yang dilengkapi kamera sebagai sensor untuk dapat mendeteksi obyek. Kombinasi antara YOLO dan IoT sangat ideal untuk aplikasi seperti terminal bus, gerbang tol, sistem parkir otomatis, dan pengawasan lalu lintas.

Berdasarkan dari uraian yang disampaikan, maka penelitian ini bertujuan untuk merancang suatu perangkat (*things*) yang dapat mendeteksi dan mengenali plat nomor angkutan bus yang berada di terminal secara otomatis menggunakan YOLO v11 dan IoT sehingga diharapkan dapat membantu dalam pendataan kendaraan angkutan bus di terminal dengan lebih cepat dan efisien

II. TINJAUAN PUSTAKA

Kecerdasan Buatan (AI) merupakan bidang ilmu komputer yang bertujuan untuk menciptakan mesin yang dapat meniru fungsi kognitif manusia, seperti belajar, pemecahan masalah, dan pengambilan keputusan[7]. Dalam konteks pengenalan objek visual, *Computer Vision* (CV) yang merupakan cabang

dari AI, memainkan peran krusial. CV memungkinkan mesin untuk mendapatkan, memproses, menganalisis, dan memahami gambar digital dari dunia nyata[8].

Kemajuan dalam *deep learning*, yang merupakan turunan dari AI, telah mengubah paradigma CV. Model jaringan saraf konvolusional (*Convolutional Neural Networks/CNN*) telah menjadi inti dari algoritma ALPR (*Automatic License Plate Recognition*) modern.

Salah satu keluarga algoritma deteksi objek yang paling dominan adalah YOLO (*You Only Look Once*). Model YOLO merevolusi deteksi objek karena kemampuannya untuk melakukan deteksi dan klasifikasi secara bersamaan dalam satu lintasan jaringan (*one-stage detector*), yang menghasilkan kecepatan yang jauh lebih tinggi dibandingkan detektor dua-tahap (*two-stage detector*) seperti R-CNN[9]. Kecepatan ini sangat penting untuk aplikasi *real-time* seperti ALPR di jalan raya atau sistem parkir. varian seperti YOLOv8 dan model yang lebih baru (seperti yang diusulkan pada YOLOv11) telah meningkatkan akurasi (diukur dengan *mean Average Precision* - mAP) sambil tetap mempertahankan kecepatan pemrosesan yang tinggi[10]. Dalam ALPR, model YOLO digunakan secara eksklusif untuk langkah pertama, yaitu Deteksi Plat Nomor, memastikan bahwa plat nomor dilokalisasi dengan cepat dan akurat terlepas dari sudut pandang dan kondisi pencahayaan[11].

Sistem ALPR modern semakin mudah diimplementasikan dengan menggunakan perangkat komputasi *edge* yang merupakan bagian dari *ekosistem Internet of Things* (IoT)[12]. IoT merujuk pada jaringan perangkat fisik yang tertanam dengan sensor, perangkat lunak, dan teknologi lain untuk tujuan menghubungkan dan bertukar data dengan perangkat dan sistem lain melalui internet[13]. Dengan IoT, deteksi plat nomor menggunakan model YOLO dapat dilakukan langsung pada perangkat *edge*[14]. Hal ini secara signifikan mengurangi latensi dan *bandwidth* yang dibutuhkan, memungkinkan respons *real-time*[15]. Setelah plat nomor dideteksi, maka dapat dilanjutkan dengan pengenalan nomor plat kendaraan dimana nomor plat kendaraan yang berupa teks dapat dikirimkan ke *cloud server* atau sistem

manajemen data melalui koneksi IoT (Wi-Fi, 4G, dll.)[16].

Dengan mengintegrasikan model CV (YOLO) dengan arsitektur IoT, sistem ALPR menjadi lebih skalabel, efisien, dan responsif, memungkinkan implementasi di lingkungan yang luas mulai dari manajemen lalu lintas pintar hingga keamanan perumahan[17].

III. ANALISA DAN PERANCANGAN SISTEM

A. Metode Pengambilan Data

1. Pengumpulan Data

Proses pengumpulan data menggunakan metode observasi, yaitu dengan memakai sistem IoT yang dikembangkan, merekam kendaraan bus yang keluar-masuk terminal dalam bentuk video berformat MP4 dan beresolusi 1280x720 pixel. Berdasarkan dari video tersebut, setiap *frame* yang terdapat gambar bus didalamnya diekstrak yang kemudian nantinya dijadikan dataset dalam proses *training* dalam pembuatan model YOLOv11. Gambar 1 menunjukkan contoh gambar yang berasal dari video yang didalamnya terdapat obyek berupa kendaraan berupa bus. Proses ekstraksi gambar dari video yang didalamnya terdapat kendaraan bus memanfaatkan model *pre-trained* YOLOv11 yang dapat mengenali 80 jenis obyek berbeda. Proses ekstraksi ini hanya akan mengekstrak *frame-frame* gambar yang didalamnya terdapat gambar bus



Gambar 1 *Frame* hasil ekstraksi gambar pada video

2. Dataset Penelitian

Berdasarkan dari kumpulan *frame-frame* gambar yang berhasil dikumpulkan dari hasil proses ekstraksi *frame* gambar bus di video, setiap gambar bus yang terdapat didalam *frame* gambar kemudian di-*crop* sesuai ukuran dimensi gambar bus seperti yang ditunjukkan

oleh Gambar 2 sehingga didalam dataset hanya terdapat obyek berupa bus dan plat nomor bus tersebut



Gambar 2 Cropping obyek kendaraan bus

Setelah gambar kandidat dataset tersebut terkumpul maka dilakukanlah proses pelabelan *dataset* yaitu plat nomor kendaraan secara manual menggunakan *tool* anotasi yang disediakan oleh Roboflow dengan hasil berupa *bounding box* untuk setiap objek target [18]. Posisi dari *bounding box* yang dihasilkan dari Roboflow nantinya akan digunakan oleh YOLOv11 untuk melakukan proses *training* pengenalan plat nomor kendaraan pada bus. Hasil dari *bounding box* ditunjukkan oleh Gambar 3.

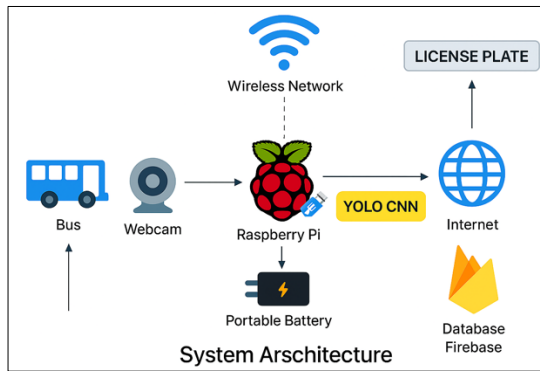


Gambar 3 *Bounding box* pada plat nomor kendaraan bus

B. Metode Pengembangan Sistem

1. Perancangan Arsitektur IoT

Pendataan kendaraan bus di terminal memerlukan perangkat IoT yang dapat menangkap gambar moda transportasi bus yang keluar-masuk terminal melalui kamera, mengenali kendaraan berupa bus yang terdapat didalamnya, membaca plat nomor kendaraan dan mengirimkannya ke database server. Perangkat IoT yang dikembangkan memiliki arsitektur sistem seperti ditunjukkan oleh Gambar 4.



Gambar 4 Arsitektur sistem IoT pembacaan plat nomor moda transportasi bus

Arsitektur sistem IoT yang ditunjukkan pada Gambar 4 dapat dijelaskan sebagai berikut :

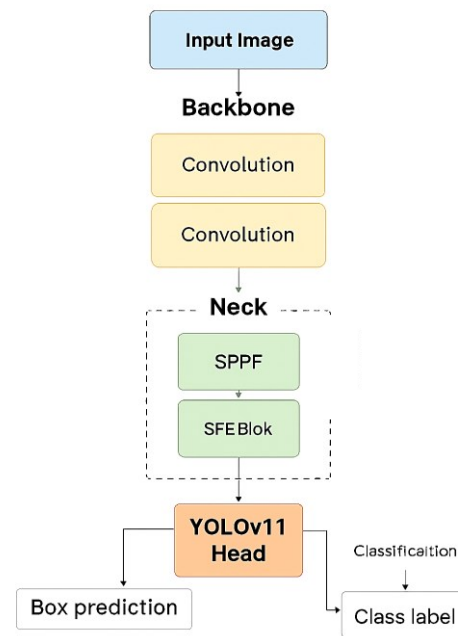
- 1) **Bus**, objek yang akan dikenali sistem.
- 2) **Webcam**, terhubung ke Raspberry Pi, digunakan untuk menangkap gambar kendaraan berupa bus.
- 3) **Raspberry Pi**, menjalankan *software* AI berbasis algoritma YOLO dan CNN untuk mengenali bus dan membaca plat nomor.
- 4) **Dongle WiFi**, tertancap di Raspberry Pi, menghubungkannya ke jaringan wireless.
- 5) **Wireless network ke internet**, jalur konektivitas menuju database Firebase
- 6) **Baterai portable**, menyediakan tenaga untuk Raspberry Pi di lapangan

2. Model YOLOv11

YOLOv11 (*You Only Look Once version 11*) adalah iterasi terbaru dalam keluarga model *object detection real-time* yang populer dalam bidang *computer vision*. Model ini dikembangkan untuk memberikan keseimbangan kinerja yang lebih baik antara kecepatan (*inferensi real-time*) dan akurasi (*mean Average Precision - mAP*) dibandingkan versi sebelumnya, seperti YOLOv8 dan YOLOv10.

a) Arsitektur YOLOv11

Arsitektur YOLOv11 ditunjukkan oleh Gambar 5, YOLOv11 memperkenalkan komponen baru seperti C3k2 block, SPPF (*Spatial Pyramid Pooling – Fast*), dan C2PSA (*Parallel Spatial Attention*) yang meningkatkan efisiensi ekstraksi fitur dan akurasi deteksi objek.



Gambar 5 Arsitektur YOLOv11

Berikut penjelasan dari Arsitektur YOLOv11

- 1) **Backbone**, lapisan convolutional yang mengekstrak fitur dari gambar input. YOLOv11 menggunakan C3k2 block untuk efisiensi komputasi dan ekstraksi fitur yang lebih dalam [19].
- 2) **Neck**, menggabungkan fitur dari berbagai skala menggunakan :
- 3) **SPPF (*Spatial Pyramid Pooling – Fast*)**: menangkap konteks spasial dari berbagai resolusi.
- 4) **Max Pooling dan SFE-Block (*Spatial Feature Enhancement*)**: memperkuat fitur spasial untuk deteksi yang lebih akurat [20].
- 5) **YOLOv11 Head**, menghasilkan dua output:
 - **Box Prediction** melalui YOLO Layer, memberikan koordinat objek (x, y, w, h).
 - **Classification** memberikan label kelas dari objek yang terdeteksi.

b) Keunggulan YOLOv11

YOLOv11 adalah evolusi dari YOLO versi sebelumnya yang menekankan pada efisiensi, kecepatan, dan adaptabilitas, menjadikannya sangat relevan untuk aplikasi *real-time* seperti sistem transportasi cerdas, pengawasan, dan

perangkat mobile. Berikut adalah keunggulan YOLOv11

- 1) Cepat dan ringan, cocok untuk perangkat *edge* dan *real-time* inference.
- 2) Modular dan fleksibel, dapat disesuaikan untuk berbagai jenis dataset.
- 3) Akurasi tinggi, terutama dalam mendeteksi objek kecil dan kompleks.

IV. IMPLEMENTASI DAN PEMBAHASAN

Hasil dan pembahasan penelitian yang dilakukan disajikan dalam beberapa hal pembahasan yaitu pembahasan perangkat keras, pembahasan perangkat lunak,

a) Perangkat keras

Sistem deteksi plat nomor kendaraan pada bus secara otomatis ini memakai perangkat IoT dengan spesifikasi seperti ditunjukkan oleh Tabel 1.

TABEL 1
SPESIFIKASI PERANGKAT KERAS IOT

Perangkat Keras	Spesifikasi
Raspberry Pi 5	CPU: Broadcom BCM2712 (Quad-core 64-bit Arm Cortex-A76 @ 2.4GHz) RAM : 8 GB Storage : Micro-SD 32 GB
Webcam	Resolusi : 2K QHD (2560 x 1440 pixel), auto focus
Power supply	27 Watt

Raspberry Pi 5 tidak memiliki *dedicated* NPU (*Neural Processing Unit*) sehingga performanya akan sangat dibatasi oleh kemampuan CPU dalam proses deteksi plat nomor kendaraan

Kemudian, untuk prototipe perangkat keras IoT yang digunakan didalam penelitian ditunjukkan oleh Gambar 6.



Gambar 6 Perangkat IoT untuk pendeteksi plat nomor kendaraan bus

b) Perangkat Lunak

Perangkat lunak berupa prototipe aplikasi pendeteksi plat nomor kendaraan dikembangkan berbasis *desktop* yang berjalan pada sistem operasi Windows. Aplikasi akan mengambil gambar yang dikirimkan oleh Raspberry Pi 5 ke Firebase Storage dan kemudian menampilkan gambar yang diperoleh ke aplikasi seperti ditunjukkan oleh Gambar 7.



Gambar 7 Aplikasi desktop pendata plat nomor kendaraan

Penjelasan mengenai Gambar 7 disampaikan pada Tabel 2 sebagai berikut

TABEL 2
KETERANGAN KOMPONEN PADA ANTARMUKA APLIKASI PENDATAAN PLAT NOMOR KENDARAAN BUS

No	Komponen	Keterangan
1	Camera View	Komponen ini berguna untuk menampilkan video yang berhasil ditangkap menggunakan web cam ataupun CCTV untuk monitoring bus keluar-masuk terminal.
2	Bus View	Komponen ini berguna untuk menampilkan gambar bus yang berhasil dideteksi dari video yang ditampilkan di Camera View.
3	Number Plate View	Komponen ini berguna untuk menampilkan gambar plat nomor yang berhasil dideteksi dari gambar bus yang ditampilkan di Bus View. Tujuan pengambilan gambar plat nomor adalah untuk mempermudah dalam proses membaca/ mengekstrak plat nomor kendaraan bus. Tujuan lainnya adalah untuk mencocokkan plat nomor asli dengan plat nomor nomor dari pembacaan OCR (Optical Character Recognition)

4	Number Plate Text View	Komponen ini berguna untuk menampilkan plat nomor kendaraan / TNKB hasil dari proses pengenalan karakter atau OCR pada gambar plat nomor. (belum berfungsi)
5	Number Plate Edit Text	Komponen ini berguna untuk merubah / mengkoreksi plat nomor kendaraan yang salah dikenali. (belum berfungsi)
6	Timestamp Text View	Komponen ini berguna untuk menampilkan dan mencatat waktu saat bus terdeteksi di terminal
7	Tombol Simpan	Komponen ini berguna untuk menyimpan data kendaraan bus yang masuk terminal. Data yang disimpan antara lain : gambar plat nomor, plat nomor kendaraan yang benar, serta tanggal dan waktu kendaraan memasuki terminal
8	Tombol Start/Stop Kamera	Komponen ini digunakan untuk menghidupkan ataupun mematikan kamera monitoring kendaraan bus yang keluar-masuk

V. PENGUJIAN

1. Pengujian Perangkat Keras

Berdasarkan hasil pengujian, Raspberry Pi 5 tidak bisa melakukan deteksi plat nomor kendaraan secara *realtime* (kecepatan deteksi mencapai 30 fps) dikarenakan Raspberry Pi 5 tidak memiliki *dedicated* NPU (*Neural Processing Unit*) sehingga performanya akan sangat dibatasi oleh kemampuan CPU. Selain itu, kompleksitas model YOLOv11 yang lebih tinggi, tetapi lebih akurat dibanding model sebelumnya, akan membebani tugas CPU dalam melakukan komputasi deteksi obyek. Hasil pengujian kecepatan CPU dalam mendeteksi plat nomor kendaraan pada berbagai resolusi gambar ditunjukkan oleh Tabel 3.

TABEL 3

HASIL UJI COBA PENDETEKSIAN PLAT NOMOR KENDARAAN PADA BERBAGAI RESOLUSI GAMBAR PADA RASPBERRY PI 5

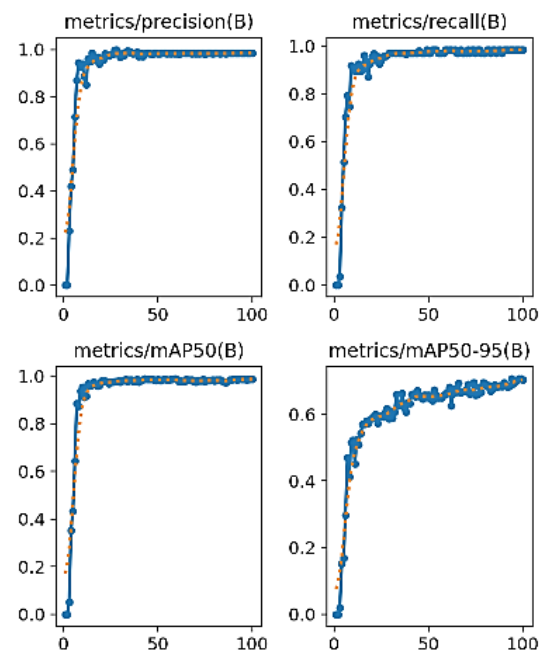
Resolusi Gambar	Kecepatan CPU dalam mendeteksi plat nomor kendaraan (dalam FPS)
1920 x 1080	1 – 5
1280 x 720	3 – 8
640 x 480	10 - 18

Meskipun Raspberry Pi 5 belum dapat bekerja secara *realtime* akan tetapi kecepatan dalam mendeteksi plat nomor kendaraan bus

sudah mencukupi untuk keperluan deteksi plat nomor kendaraan bus di area terminal dikarenakan kendaraan bus di terminal tidak berjalan dengan cepat sehingga cukup waktu mendeteksi plat nomor kendaraan bus yang keluar masuk terminal.

2. Pengujian YOLOv11

Pengujian dilakukan dengan menggunakan dataset yang sudah dibentuk, yaitu terdiri dari 682 gambar yang dianotasi menggunakan format YOLOv11 dengan tahapan preprocessing meliputi *auto-orientation*, *resize/stretch* gambar ke ukuran 640x640, dan *auto-contrast* menggunakan *contrast stretching* dengan proses training dilakukan sebanyak 100 epoch dengan hasil ditunjukkan oleh Gambar 8 dengan penjelasan ditunjukkan oleh Tabel 4.



Gambar 8 Grafik hasil training YOLOv11

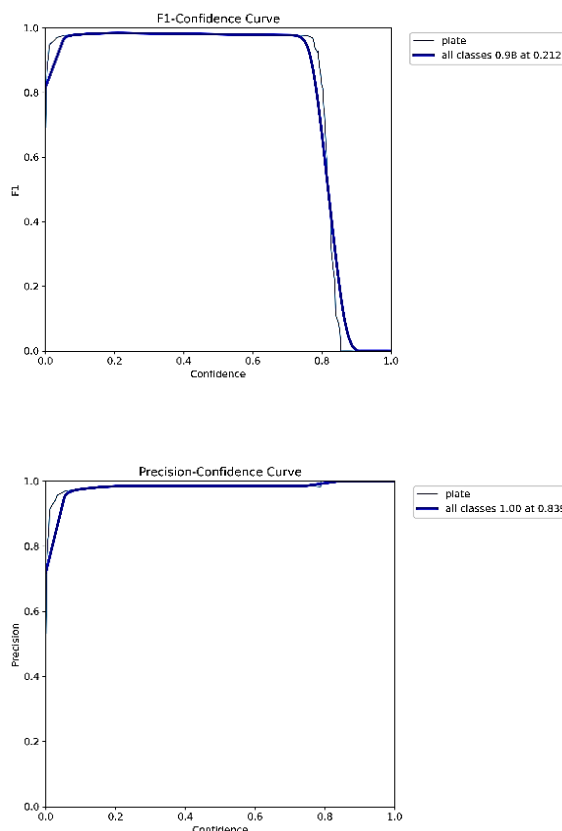
TABEL 4
METRIK VALIDASI

Metrik	Nilai pada Epoch=100
<i>Precision</i> (B)	0,98507
<i>Recall</i> (B)	0,98498
mAP50 (B)	0,98674
mAP50-95 (B)	0,70355

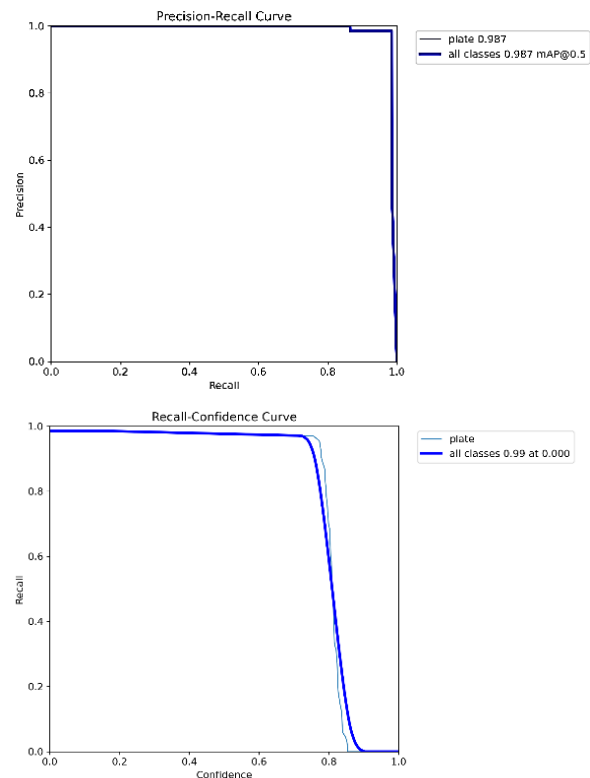
Metrik ini dihitung pada *dataset* validasi yang tidak pernah dilihat oleh model selama pelatihan *epoch* tersebut. Metrik ini adalah indikator utama kemampuan model untuk

menggeneralisasi data baru dimana nilai *precision*, *recall*, mAP50 mencapai nilai diatas 98%, sementara untuk mAP50-95 yang merupakan nilai rata-rata presisi untuk semua kelas, dihitung rata-rata di berbagai ambang batas IoU (*intersection over union*) dari 50% hingga 95% dengan langkah 5% merupakan metrik yang lebih ketat yang mengukur deteksi yang sangat akurat (lokalisasi dan klasifikasi) dengan nilai mencapai 70%

Kemudian untuk kurva *confidence* ditampilkan pada Gambar 9a dan Gambar 9b dengan penjelasan sebagai berikut:



Gambar 9a Kurva F1-Confidence dan Kurva Precision-Confidence



Gambar 9b Kurva Precision Recall dan Kurva Recall-Confidence

1) Kurva F1-Confidence

Model ini sangat akurat, mencapai skor F1 yang sangat tinggi (0.98). Kemudian, thresholdnya optimal, karena untuk mendapatkan hasil terbaik saat menjalankan model di dunia nyata maka harus mengatur ambang batas kepercayaan model menjadi 0.212 atau sekitar 0.2

2) Kurva Precision-Confidence

Pada nilai 0.839 menunjukkan bahwa Model ini menunjukkan kinerja yang sangat baik, dengan hampir tidak ada *false positives* pada tingkat kepercayaan yang tinggi.

3) Kurva Precision-Recall

Kurva yang hampir persegi dengan nilai 0.987 pada mAP50 menunjukkan bahwa model dapat mencapai *recall* yang sangat tinggi (menemukan hampir semua plat) sambil mempertahankan *precision* yang sangat tinggi (hampir semua yang dideteksi adalah plat yang benar).

4) Kurva Recall-Confidence

Kurva menunjukkan bahwa dari *confidence* 0.0 hingga sekitar 0.75, *Recall* model berada pada nilai maksimum, yaitu 1.0 (atau 0.99). *Recall* maksimum (0.99) dicapai ketika ambang batas *confidence* diatur pada 0.000 yang artinya, apabila *confidence* diatur pada nilai dibawah 0,75 maka akan sangat efektif dalam menemukan hampir semua plat nomor kendaraan. Akan tetapi bila diatas 0.75 maka akan banyak melewati banyak deteksi karena skor *confidence* rendah

VI. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem deteksi plat nomor kendaraan angkutan bus secara otomatis berbasis IoT menggunakan perangkat Raspberry Pi 5 dan algoritma YOLOv11.

1. **Kinerja Deteksi Model (Akurasi):** Model YOLOv11 yang dilatih untuk deteksi plat nomor menunjukkan kinerja yang sangat baik dan teruji. Metrik validasi mencapai nilai yang sangat tinggi, dengan mAP50 sebesar 0.98674 dan Presisi serta *Recall* di atas 98%³. Kurva F1-Confidence mengkonfirmasi bahwa ambang batas kepercayaan optimal model berada pada 0.212 dengan skor F1 maksimum 0.98, menunjukkan keseimbangan yang sangat baik antara Presisi dan *Recall*.
2. **Performa Perangkat Keras (Edge Computing):** Implementasi pada Raspberry Pi 5 menunjukkan batasan karena tidak adanya *dedicated* NPU. Pada resolusi optimal 640x480 piksel, kecepatan deteksi berkisar antara 10 hingga 18 FPS. Meskipun belum mencapai 30 FPS (deteksi *real-time*), kecepatan ini dinilai sudah mencukupi dan fungsional untuk keperluan pendataan kendaraan bus di area terminal yang memiliki pergerakan lambat.
3. **Arsitektur Sistem:** Kombinasi YOLOv11 dan arsitektur IoT terbukti efektif dalam merancang suatu perangkat (*things*) yang dapat mendeteksi plat nomor, serta mengirimkan data ke *database* server (Firebase) melalui koneksi nirkabel. Sistem ini dapat menggantikan metode konvensional yang rentan terhadap kesalahan manusia dan tidak *real-time*.

Secara keseluruhan, penelitian ini menunjukkan bahwa YOLOv11 pada Raspberry Pi 5 adalah solusi yang layak dan efisien untuk otomatisasi pendataan plat nomor bus di terminal.

REFERENSI

- [1] R. Nursin, "Kemenhub minta keberangkatan bus dikelola dengan pendataan akurat," *Medcom.id*, 2024. <https://www.medcom.id/ekonomi/bisnis/n92PgEb-kemenhub-minta-keberangkatan-bus-dikelola-dengan-pendataan-akurat>
- [2] R. Fitriana, "Ditjen Hubdat Tekankan Pentingnya Bus Masuk Terminal," *Majalah Bandara*, 2025. <https://www.majalahbandara.com/ditjen-hubdat-tekanan-pentingnya-bus-masuk-terminal/>
- [3] M. Bensouilah, M. N. Zennir, and M. Taffar, "An ALPR System-based Deep Networks for the Detection and Recognition," in *International Conference on Pattern Recognition Applications and Methods*, 2021, pp. 204–211. doi: 10.5220/0010229202040211.
- [4] D. Laroca, R., Zanolensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., & Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *IET Intell. Transp. Syst.*, vol. 15, no. 4, pp. 483–503, 2021.
- [5] E. W. Prisma, R., Pamungkas, "PEMANFAATAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI CITRA AWAN MENGGUNAKAN RASPBERRY PI," 2024.
- [6] Ultralytics, "Ultralytics YOLO11," *Ultralytics YOLO Docs*, 2025. <https://docs.ultralytics.com/models/yolo11/> (accessed Nov. 25, 2025).
- [7] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach, Global Edition 4th," *Foundations*, vol. 19, 2021.
- [8] "Computer vision: algorithms and applications," *Choice Rev. Online*, vol. 48, no. 09, 2011, doi: 10.5860/choice.48-5140.
- [9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv Prepr. arXiv1804.02767*, 2018.
- [10] A. Amin, R. Mumtaz, M. J. Bashir, and S. M. H. Zaidi, "Next-Generation License Plate Detection and Recognition System

- Using YOLOv8,” in *2023 IEEE 20th International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT, HONET 2023*, 2023. doi: 10.1109/HONET59747.2023.10374756.
- [11] et al Li, “An End-to-End License Plate Detection and Recognition Approach using Enhanced YOLO and CNN,” *Sensors*, vol. 22, no. 12, p. 4467, 2022.
- [12] A. Al-Fuqaha, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.
- [13] R. Z. Alhamri and D. Elfarozi, “Sistem Kendali Ketinggian dan Debit Air Berbasis Mobile pada Inkubator Telur Ikan Nila Memanfaatkan Internet of Things,” *J. Inform. dan Multimed.*, vol. 15, no. 2, pp. 34–42, 2023, doi: 10.33795/jtim.v15i2.4970.
- [14] M. M. A. and A. Et, “Edge Computing for Real-Time Computer Vision Applications: A Review,” *J. Netw. Comput. Appl.*, vol. 200, p. 103328, 2022.
- [15] S. S. Wani and P. B. Rane, “A low-latency embedded system for vehicle license plate recognition using optimized deep learning models on Raspberry Pi,” *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 3, pp. 3175–3188, 2023.
- [16] J. Li and A. Et, “An IoT-based intelligent transportation system using deep learning for vehicle detection and traffic flow analysis,” *Futur. Gener. Comput. Syst.*, vol. 134, pp. 268–281, 2022.
- [17] R. V. Kulkarni and D. C. Kulkarni, “Design and implementation of a smart parking system using IoT and machine learning,” *Microprocess. Microsyst.*, vol. 93, p. 104597, 2022.
- [18] Roboflow, “Platform for Computer Vision,” <https://roboflow.com>, 2025. <https://roboflow.com>
- [19] M. H. Rahima Khanam, “YOLOv11: An overview of the key architectural enhancements,” *arXiv Prepr. arXiv2410.17725*, 2024, [Online]. Available: [arxiv:2410.17725](https://arxiv.org/abs/2410.17725)
- [20] V. K. S. Sivadi Balakrishna, Shivani Yadao, “Enhancing YOLOv11 for Real-Time Object Detection: Advanced Architectures and Edge-Optimized Training Pipeline,” in *Proceedings of the Ninth International Conference on Intelligent Computing in Engineering*, ACSIS Vol. 42, 2024, pp. 115–122.