



Evaluasi Komparatif Kinerja YOLOv11 dan ResNet-50 untuk Deteksi Gestur Tangan pada Kendali *Differential Drive Mobile Robot*

Diana Mar'atus Sholihah^{1*}, Muhammad Latif², Muhammad Fuad³, Vivi Tri Widyaningrum⁴,
Hairil Budiarto⁵

^{1,2,3,4,5} Program Studi Teknik Mekatronika, Universitas Trunodjoyo Madura, Indonesia

E-Mail *: 220491100037@student.trunojoyo.ac.id

ABSTRACT

Computer vision-based hand gesture recognition provides a more intuitive interaction method for controlling a Differential Drive Mobile Robot (DDMR). This study compares two different deep learning approaches: YOLOv11 as an object detection model and ResNet-50 as an image classification model. The dataset consists of 500 images divided into five gesture classes: forward, backward, turn right, turn left, and stop. Both models were trained using transfer learning and evaluated through internal testing and direct camera-based testing on the robot control system. The internal evaluation showed that YOLOv11 achieved an mAP50 of 0.995 and an mAP50-95 of 0.784, while ResNet-50 obtained precision, recall, and F1-score values of 1.000 for all classes. However, the direct testing results revealed a substantial performance difference. YOLOv11 produced confidence values ranging from 0.80 to 0.90, with an average of 0.844, whereas ResNet-50 produced confidence values ranging from 0.32 to 0.77, with an average of 0.442. The discrepancy between internal evaluation and direct testing indicates a domain gap between the dataset images and the operational camera images. Under the testing conditions used in this study, YOLOv11 produced higher and more consistent confidence values, making it more suitable for hand gesture recognition in DDMR control. The main contribution of this study is the comparative evaluation of object detection and image classification approaches not only on an internal dataset but also through direct implementation in a camera-based robot control system.

Keywords: YOLOv11, ResNet-50, gestur tangan, Differential Drive Mobile Robot, deep learning, confidence.

ABSTRAK

Pengenalan gestur tangan berbasis visi komputer menawarkan metode interaksi yang lebih intuitif untuk mengendalikan *Differential Drive Mobile Robot* (DDMR). Penelitian ini bertujuan membandingkan dua pendekatan *deep learning* yang berbeda, yaitu YOLOv11 sebagai model deteksi objek dan ResNet-50 sebagai model klasifikasi citra. Dataset terdiri atas 500 citra yang terbagi ke dalam lima kelas gestur, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Kedua model dilatih menggunakan pendekatan *transfer learning* dan dievaluasi melalui pengujian internal serta pengujian langsung menggunakan kamera pada sistem kendali robot. Hasil evaluasi internal menunjukkan bahwa YOLOv11 memperoleh nilai mAP50 sebesar 0,995 dan mAP50-95 sebesar 0,784, sedangkan ResNet-50 menghasilkan nilai *precision*, *recall*, dan *F1-score* sebesar 1,000 pada seluruh kelas. Meskipun demikian, hasil pengujian langsung menunjukkan perbedaan yang signifikan. YOLOv11 menghasilkan nilai *confidence* pada rentang 0,80–0,90 dengan rata-rata 0,844, sedangkan ResNet-50 memperoleh nilai *confidence* pada rentang 0,32–0,77 dengan rata-rata 0,442. Perbedaan antara evaluasi internal dan pengujian langsung menunjukkan adanya *domain gap* antara citra dataset dan citra pada kondisi operasional. Berdasarkan skenario pengujian penelitian ini, YOLOv11 menghasilkan nilai *confidence* yang lebih tinggi dan lebih konsisten sehingga lebih sesuai digunakan untuk pengenalan gestur tangan pada kendali DDMR. Kontribusi penelitian ini terletak pada evaluasi komparatif antara pendekatan deteksi objek dan klasifikasi citra, tidak hanya pada dataset internal, tetapi juga pada penerapan langsung untuk kendali robot berbasis kamera.

Kata Kunci : YOLOv11, ResNet-50, *hand gesture*, *Differential Drive Mobile Robot*, *deep learning*, *confidence*.

Naskah diterima 18 Mei 2026; Revisi 09 Juni 2026; Diterima 03 Agustus 2026. Tanggal Publikasi 17 September 2026

Jurnal teknika berada pada lisensi *Creative Commons Attribution-ShareAlike 4.0 International License*

DOI: [10.30736/jt.v18i2.1654](https://doi.org/10.30736/jt.v18i2.1654), Hal 165-176



1. PENDAHULUAN

Perkembangan teknologi robotika dan kecerdasan buatan (*Artificial Intelligence/AI*) telah mendorong terciptanya berbagai sistem interaksi manusia dan robot yang lebih intuitif dan efisien (Ali Akbar Ritonga & Elysa Rohayani Hasibuan, 2025; Bambang Karyadi, 2023; Imam Zaenuddin & Ade Bani Riyan, 2024; Wajdi & Hajiri, 2024). Salah satu pendekatan yang berkembang dalam bidang *Human-Robot Interaction* (HRI) adalah penggunaan pengenalan gestur tangan sebagai media komunikasi antara manusia dan robot (Ahmad Rifa'i, 2025; Al Rivan & Setiawan, 2022; Amelia Dewi Agustiani dkk., 2024; Chalik dkk., 2021; Nur Budiman dkk., 2022). Pendekatan ini memungkinkan pengguna memberikan perintah kepada robot melalui gerakan tangan tanpa menggunakan perangkat tambahan seperti *joystick* atau *remote control*. Dengan memanfaatkan kamera dan algoritma pengolahan citra berbasis *deep learning*, sistem dapat mengenali gestur tangan secara otomatis dan menerjemahkannya menjadi perintah kendali bagi robot.

Dalam beberapa tahun terakhir pengenalan gestur tangan berbasis visi komputer semakin banyak diterapkan pada berbagai bidang, seperti kendali robot, *augmented reality*, dan sistem interaksi manusia-mesin. Perkembangan *deep learning*, khususnya arsitektur *Convolutional Neural Network* (CNN) telah memungkinkan sistem komputer mengenali pola visual dengan performa yang lebih baik (Bima Dinda Nurwibowo & Ahmad Saikhu, 2025). Telah memungkinkan sistem komputer untuk mengenali pola visual secara lebih akurat. Model deteksi objek seperti *You Only Look Once* (YOLO) mampu mengenali kelas dan lokasi objek dalam satu proses deteksi sehingga sesuai untuk aplikasi yang membutuhkan pemrosesan cepat (Awan Aprilino, 2022; Dede Haris Saputra & Bahtiar Imran, 2023; Inayatul Arifah dkk., 2022; Restu Gilang Wijanarko & Afu Ichsan Pradana, 2024). Di sisi lain, model klasifikasi citra seperti *Residual Network* (ResNet) mampu mengekstraksi fitur visual melalui jaringan yang dalam dan mekanisme *residual connection* untuk mengurangi permasalahan *vanishing gradient* (Berliani dkk., 2023).

Penerapan model *deep learning* pada pengenalan gestur tangan memiliki potensi untuk meningkatkan kinerja sistem kendali robot berbasis visi komputer. Sistem dapat mengenali gestur tangan dari citra kamera dan menerjemahkannya menjadi perintah gerak robot. Dalam konteks robot *mobile*, salah satu *platform* yang banyak digunakan dalam penelitian adalah *Differential Drive Mobile Robot* (DDMR). Jenis robot ini menggunakan dua roda penggerak utama dan menghasilkan pola gerak berdasarkan perbedaan kecepatan roda kiri dan roda kanan (Nobby Bagus Muliawan & Indra Adji Sulistijono, 2023). Melalui integrasi deteksi gestur tangan, pengguna dapat mengontrol gerakan robot seperti maju, mundur, belok kanan, belok kiri, dan berhenti dengan cara yang lebih alami serta mudah

dipahami.

Beberapa penelitian sebelumnya telah menggunakan model YOLO untuk mendeteksi objek secara *real-time* karena kemampuannya dalam melakukan klasifikasi dan lokalisasi objek secara bersamaan (Inayatul Arifah dkk., 2022; Syakir & Syani, 2024). Penelitian lain menggunakan arsitektur CNN atau ResNet untuk melakukan klasifikasi citra berdasarkan pola visual yang terdapat pada citra masukan Al Rivan & Setiawan, 2022; Berliani dkk., 2023). Namun, penelitian-penelitian tersebut umumnya mengevaluasi satu pendekatan model secara terpisah. Perbandingan antara model deteksi objek dan model klasifikasi citra dalam konteks pengenalan gestur tangan untuk kendali DDMR masih relatif terbatas. Selain itu, performa model sering kali hanya dievaluasi menggunakan dataset internal tanpa dibandingkan dengan hasil pengujian langsung menggunakan kamera pada kondisi operasional.

Berdasarkan permasalahan tersebut, penelitian ini membandingkan dua pendekatan *deep learning*, yaitu YOLOv11 dan ResNet-50, untuk mendeteksi gestur tangan pada sistem kendali robot *mobile*. Model YOLOv11 digunakan sebagai metode deteksi objek yang mampu mengenali lokasi dan kelas gestur secara simultan dalam satu proses (Therino Elean dkk., 2025). Sementara itu, ResNet-50 digunakan untuk mengklasifikasikan gestur berdasarkan fitur visual yang diekstraksi dari keseluruhan citra masukan. Pemilihan kedua model tersebut bertujuan untuk membandingkan kemampuan pendekatan lokalisasi objek dengan pendekatan klasifikasi citra dalam menghadapi perubahan posisi tangan, jarak kamera, pencahayaan, dan kondisi latar belakang.

Kontribusi utama penelitian ini terletak pada evaluasi komparatif YOLOv11 dan ResNet-50 yang tidak hanya dilakukan berdasarkan performa pada dataset internal, tetapi juga berdasarkan nilai *confidence* pada pengujian langsung menggunakan kamera dan penerapannya sebagai *input* kendali *Differential Drive Mobile Robot*. Melalui pendekatan tersebut, penelitian ini menganalisis kesenjangan antara performa model pada lingkungan pelatihan dan performa model pada kondisi operasional.

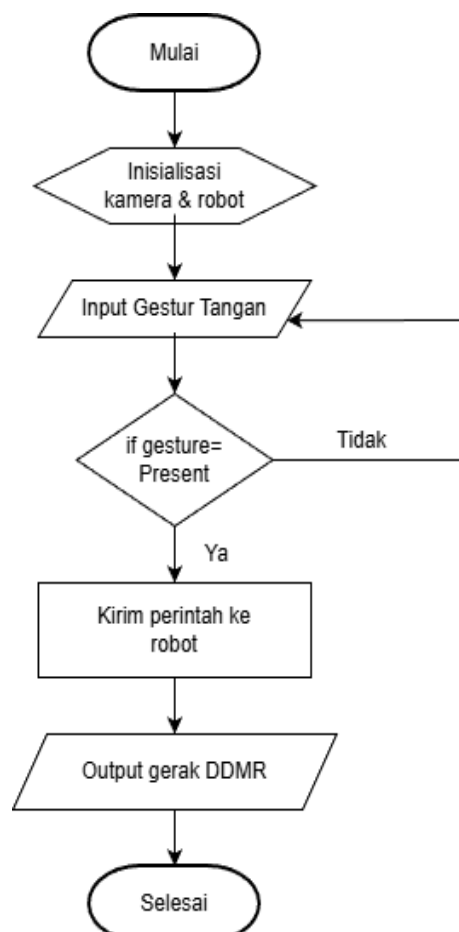
Penelitian ini bertujuan untuk menganalisis dan membandingkan performa YOLOv11 dan ResNet-50 dalam mengenali gestur tangan sebagai sarana kendali DDMR. YOLOv11 dievaluasi menggunakan *precision*, *recall*, mAP50, mAP50-95, dan waktu inferensi, sedangkan ResNet-50 dievaluasi menggunakan *accuracy*, *precision*, *recall*, F1-score, *confusion matrix*, dan waktu inferensi. Pengujian langsung kedua model dianalisis menggunakan nilai *confidence* yang dihasilkan pada setiap kelas gestur. Hasil penelitian diharapkan dapat memberikan dasar dalam menentukan model yang lebih sesuai untuk pengenalan gestur tangan pada sistem kendali robot berbasis kamera.

2. METODE

Penelitian ini melakukan perbandingan performa dua model berbasis *Convolutional Neural Network* (CNN), yaitu YOLOv11 dan ResNet-50, dalam mengenali gestur tangan sebagai media pengendalian *Differential Drive Mobile Robot* (DDMR). Proses penelitian mencakup proses pengumpulan dataset, pelatihan model, penerapan model pada sistem, serta evaluasi kinerja kedua model secara komparatif, serta evaluasi perbandingan kinerja kedua model menggunakan beberapa metrik pengujian seperti *accuracy*, *precision*, *recall*, *mAP*, dan waktu inferensi.

2.1 Flowchart Sistem

Sistem kerja dimulai dari proses penangkapan citra oleh kamera yang digunakan sebagai *input* sistem. Citra yang diperoleh kemudian diproses oleh model YOLOv11 dan ResNet-50 untuk mendeteksi serta mengenali gestur tangan pengguna. Hasil deteksi tersebut selanjutnya diterjemahkan menjadi perintah kendali untuk mengontrol pergerakan *Differential Drive Mobile Robot* (DDMR). Alur kerja sistem digambarkan dalam bentuk *flowchart* sebagaimana ditunjukkan pada Gambar 1, alur penelitian diawali dengan proses akusisi citra, proses deteksi gestur tangan, hingga penerjemahan hasil deteksi menjadi perintah gerak robot.



Gambar 1. Flowchart Sistem

2.2 Dataset Gestur Tangan

Dataset yang digunakan dalam penelitian ini terdiri dari lima kelas gestur tangan yang merepresentasikan perintah gerakan robot (Roziqin dkk., 2021). Gestur tangan yang digunakan terdiri dari perintah maju, mundur, belok kanan, belok kiri, dan berhenti. Setiap gestur berfungsi sebagai instruksi kendali untuk menentukan arah serta pola pergerakan *Differential Drive Mobile Robot* (DDMR). Dataset berjumlah 500 citra, dengan masing-masing kelas terdiri dari 100 citra. Data diambil menggunakan kamera dengan variasi posisi tangan, sudut pengambilan, jarak kamera, dan kondisi latar belakang.

Model YOLOv11, citra diberi anotasi *bounding box* pada area tangan, sedangkan pada ResNet-50 citra digunakan sebagai data klasifikasi berdasarkan label kelas. Dataset dibagi secara *stratified* dengan rasio 70:15:15, yaitu 350 citra data latih, 75 citra data validasi, dan 75 citra data uji. Pembagian ini dilakukan agar proporsi setiap kelas tetap seimbang pada proses pelatihan dan evaluasi. Bentuk gestur tangan serta jenis pergerakan robot yang dihasilkan ditunjukkan pada Tabel 1.

Tabel 1. Datasheet Gestur Tangan

No	Bentuk Gestur	Jumlah	Gerak
1		100	Maju
2		100	Mundur
3		100	Belok Kanan
4		100	Belok Kiri
5		100	Berhenti

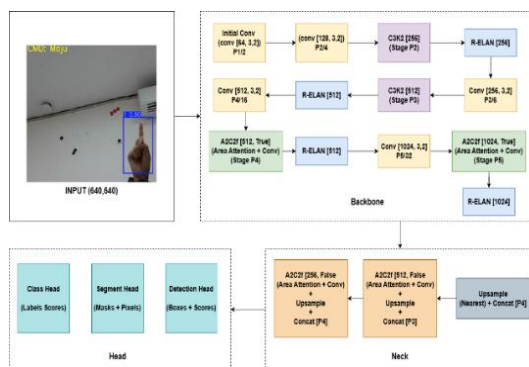
2.3 Arsitektur Model YOLOv11

Model YOLOv11 digunakan dalam penelitian ini sebagai model deteksi objek untuk mengenali gestur tangan berdasarkan kelas dan posisi objek melalui *bounding box*. Arsitektur YOLOv11 yang digunakan ditunjukkan pada Gambar 2. Model ini dipilih karena memiliki keunggulan dalam efisiensi komputasi, kecepatan inferensi, dan kemampuan deteksi objek secara *real-time*.

Proses pelatihan dilakukan menggunakan metode *transfer learning* dengan memanfaatkan

bobot awal (*pre-trained weights*) yang kemudian disesuaikan dengan dataset gestur tangan pada penelitian ini. Dataset terdiri dari lima kelas gestur, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Sebelum pelatihan, citra diberi anotasi *bouding box* pada area tangan dan disesuaikan ke ukuran input 640×640 piksel.

Parameter pelatihan YOLOv11 meliputi ukuran citra 640×640 piksel, jumlah *epoch* 100, *batch size* 16, dan *learning rate* 0,001. Ukuran *input* 640×640 piksel dipilih untuk menjaga keseimbangan antara detail objek dan efisiensi komputasi, sedangkan *batch size* 16 dan *learning rate* 0,001 digunakan agar proses pelatihan berjalan stabil. Evaluasi model YOLOv11 dilakukan menggunakan *precision*, *recall*, mAP50, mAP50-95, *confidence*, dan waktu inferensi.



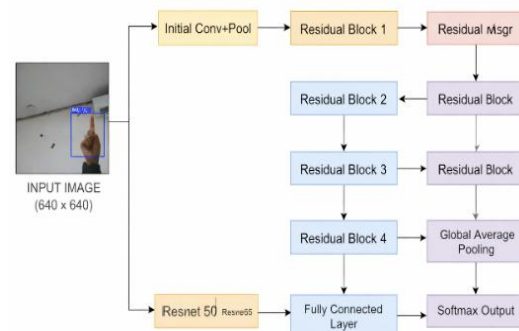
Gambar 2. Arsitektur YOLO

2.4 Arsitektur Model ResNet-50

Model ResNet-50 digunakan dalam penelitian ini sebagai model klasifikasi citra untuk mengenali gestur tangan berdasarkan fitur visual yang diekstraksi dari citra *input* (Behar & Shrivastava, 2022; Islam dkk., 2022; Leong & Zhao, 2025). ResNet atau *Residual Network* adalah salah satu arsitektur *Convolutional Neural Network* (CNN) yang menerapkan *residual learning* melalui mekanisme *skip connection* untuk mengurangi masalah *vanishing gradient* pada jaringan berlapis dalam. Arsitektur ResNet-50 tersusun atas sejumlah lapisan konvolusi yang dibentuk dalam blok residual, sehingga memungkinkan proses ekstraksi fitur yang lebih optimal dan stabil selama pelatihan. Pada penelitian ini, model ResNet-50 dilatih menggunakan pendekatan *transfer learning* dengan memanfaatkan *pre-trained weights* sebagai bobot awal yang kemudian disesuaikan dengan dataset gestur tangan.

Citra *input* pada ResNet-50 diubah ke ukuran 224×224 piksel sesuai dengan standar *input* model. Lapisan klasifikasi akhir dimodifikasi menjadi lima kelas, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Model ini dievaluasi menggunakan *accuracy*, *precision*, *recall*, *F1-score*, *confusion matrix*, *confidence*, dan waktu inferensi. Penggunaan ResNet-50 bertujuan untuk membandingkan pendekatan klasifikasi citra dengan

pendekatan deteksi objek pada YOLOv11. Adapun arsitektur ResNet-50 yang digunakan pada penelitian ini disajikan pada Gambar 3.



Gambar 3. Arsitektur ResNet-50

2.5 Pelatihan Model

Pelatihan model dilakukan dengan menggunakan dataset gestur tangan yang telah disiapkan pada tahap sebelumnya. Pada penelitian ini digunakan dua model berbasis *Convolutional Neural Network* (CNN) yaitu YOLOv11 dan ResNet-50. Kedua model dilatih menggunakan pendekatan *transfer learning* dengan memanfaatkan *pre-trained weights* sebagai bobot awal, kemudian disesuaikan kembali dengan dataset penelitian. Dataset yang digunakan mencakup lima kelas gestur tangan, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Dataset dibagi menjadi data latih, data validasi, dan data uji dengan rasio 70:15:15. Pembagian ini dilakukan secara *stratified* agar jumlah data pada setiap kelas tetap seimbang.

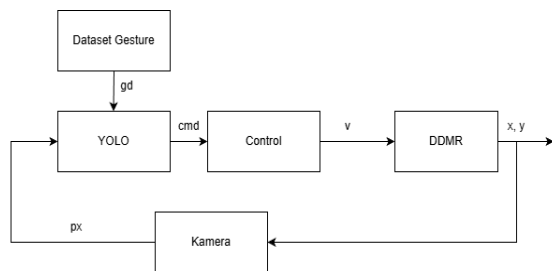
Konfigurasi pelatihan YOLOv11 menggunakan ukuran citra 640×640 piksel, *epoch* 100, *batch size* 16, dan *learning rate* 0,001. Sementara itu, ResNet-50 menggunakan ukuran citra 224×224 piksel dengan jumlah kelas keluaran sebanyak lima kelas. Pemisahan ukuran *input* dilakukan karena YOLOv11 digunakan sebagai model deteksi objek, sedangkan ResNet-50 digunakan sebagai model klasifikasi citra. Proses pelatihan dilakukan hingga model mampu mengenali pola gestur tangan sebelum diterapkan pada tahap pengujian dan evaluasi performa.

2.6 Kendali Differential Drive Mobile Robot

Model sistem kendali pergerakan *Differential Drive Mobile Robot* (DDMR) ditunjukkan pada Gambar 4. Sistem ini memanfaatkan kamera sebagai sensor visi yang berfungsi untuk memperoleh citra gestur tangan sebagai masukan utama sistem. Citra yang diperoleh kemudian diproses oleh model YOLOv11 dan ResNet-50 untuk mendeteksi serta mengenali gestur tangan pengguna. Hasil deteksi tersebut selanjutnya diterjemahkan menjadi perintah kendali untuk mengontrol pergerakan DDMR. Setiap gestur tangan dipetakan menjadi perintah gerak robot, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Gestur maju digunakan untuk menggerakkan kedua roda ke

depan, gestur mundur untuk menggerakkan kedua roda ke belakang, gestur belok kanan dan belok kiri untuk mengatur perbedaan gerak roda kanan dan kiri, sedangkan gestur berhenti digunakan untuk menghentikan kedua motor.

Apabila gestur tidak dikenali atau nilai *confidence* berada di bawah ambang batas yang ditentukan, sistem tidak mengirimkan perintah gerak baru ke robot. Kamera akan kembali melakukan proses akuisisi citra hingga gestur berhasil terdeteksi dengan nilai *confidence* yang memadai. Mekanisme ini digunakan untuk mengurangi risiko kesalahan gerak akibat prediksi yang tidak stabil. Diagram blok sistem kendali DDMR pada penelitian ini ditunjukkan pada Gambar 4, yang menggambarkan alur sistem mulai dari dataset gestur, proses deteksi menggunakan model CNN, hingga perintah kendali yang diteruskan ke robot.



Gambar 4. Diagram blok kontrol DDMR.

DDMR memiliki dua roda penggerak aktif yang terletak pada bagian belakang robot, yaitu roda kanan dan roda kiri (Latif, 2020). Model kinematika pergerakan DDMR ditentukan berdasarkan perbedaan kecepatan roda kiri (V_L) dan roda kanan (V_R). Jarak antara kedua roda dinyatakan sebagai $2d$. Apabila kedua roda bergerak dengan kecepatan yang sama ($V_L = V_R$), maka robot akan bergerak lurus. Namun jika kecepatan kedua roda berbeda ($V_L \neq V_R$), maka robot akan berbelok mengelilingi titik *Instantaneous Center of Curvature* (ICC) dengan kecepatan sudut ω . Model kinematika DDMR ditunjukkan pada Gambar 5. Secara matematis hubungan antara parameter tersebut dapat dinyatakan pada rumus 1.

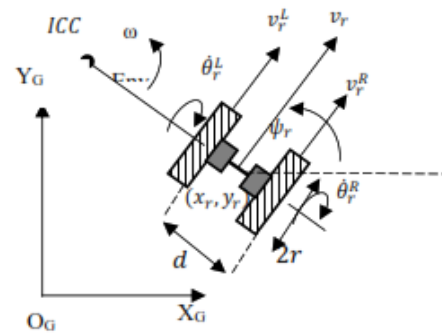
$$V = \frac{V_L + V_R}{2} \quad (1)$$

Keterangan:

V = Kecepatan linear pada pusat robot

V_L = Kecepatan roda bagian kiri

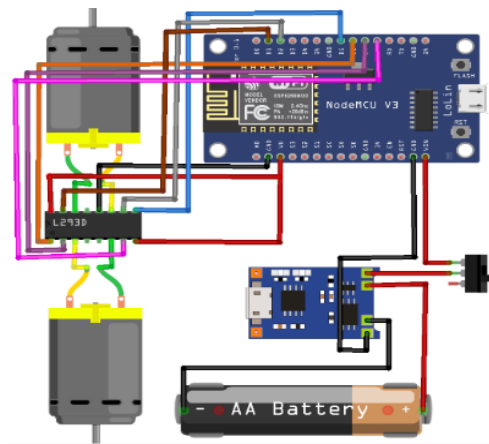
V_R = Kecepatan roda bagian kanan



Gambar 5. Differential Drive Mobile Robot (Fuad dkk., 2023)

2.7 Rangkaian Elektronika

Rangkaian elektronika DDMR terdiri dari modul ESP8266 sebagai mikrokontroler (Azhar dkk., 2024; Kemala dkk., 2022). mikrokontroler berguna sebagai pengendali motor driver L293d untuk mengatur arah dan kecepatan DDMR. Proses pengenalan gestur dilakukan oleh komputer yang akan dikirim ke mikrokontroler dalam bentuk perintah. Komunikasi antara komputer dan mikrokontroler dilakukan melalui protokol *WiFi*, sehingga proses deteksi dapat dijalankan secara paralel. Sumber daya energi berasal dari baterai Lion 4.2 V dengan kapasitas 1500 mAh yang mampu memberikan suplai energi stabil selama pengujian. Rangkaian elektronika ditunjukkan pada Gambar 6.



Gambar 6. Rangkaian elektronika

2.8 Parameter Evaluasi Model

Evaluasi performa model dilakukan untuk mengetahui serta membandingkan kemampuan YOLOv11 dan ResNet-50 dalam mengenali gestur tangan pada sistem kendali *Differential Drive Mobile Robot* (DDMR). Sejumlah parameter evaluasi digunakan sebagai acuan untuk menilai tingkat akurasi dan ketepatan model dalam mengenali gestur tangan. Parameter evaluasi disesuaikan dengan karakteristik model. YOLOv11 dievaluasi sebagai model deteksi objek, sedangkan ResNet-50 dievaluasi sebagai model klasifikasi citra. Oleh karena itu, mAP hanya digunakan untuk mengevaluasi YOLOv11, sedangkan ResNet-50

dievaluasi menggunakan *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*. Selain metrik evaluasi pelatihan, pengujian langsung menggunakan kamera juga dianalisis berdasarkan rata-rata *confidence* dan waktu inferensi. Rata-rata *confidence* digunakan untuk menunjukkan tingkat keyakinan model saat mengenali gestur secara langsung. Parameter evaluasi yang digunakan dalam penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. Parameter Perbandingan Model

No	Parameter	Model	Deskripsi
1	<i>Accuracy</i>	ResNet-50	Tingkat ketepatan model dalam mengklasifikasikan gestur tangan.
2	<i>Precision</i>	YOLO v11 dan ResNet-50	Ketepatan model dalam memprediksi gestur yang benar.
3	<i>Recall</i>	YOLO v11 dan ResNet-50	Kemampuan model menemukan gestur yang sesuai dengan kelas sebenarnya.
4	<i>F1-score</i>	ResNet-50	Nilai keseimbangan antara <i>precision</i> dan <i>recall</i> .
5	<i>mAP50</i> dan <i>mAP50-95</i>	YOLO v11	Rata-rata presisi deteksi objek berdasarkan nilai <i>Intersection over Union</i> (IoU).
6	<i>Mean confidence</i>	YOLO v11 dan ResNet-50	Rata-rata tingkat keyakinan model saat pengujian langsung menggunakan kamera.
7	Waktu inferensi	YOLO v11 dan ResNet-50	Waktu yang dibutuhkan model untuk menghasilkan prediksi pada satu citra.

3. Hasil Dan Pembahasan

Penelitian ini membandingkan performa model YOLO dan ResNet-50 dalam proses pengenalan objek berbasis citra. Evaluasi dilakukan menggunakan beberapa metrik kinerja, meliputi akurasi, *precision*, *recall*, *F1-score*, serta *mean Average Precision* (mAP) yang digunakan khusus untuk menilai performa model deteksi.

3.1. Hasil Training Model YOLO

Tahap ini mencakup pelatihan model YOLOv11 untuk mendeteksi gestur tangan secara *real-time* menggunakan pendekatan *transfer learning* dengan

pre-trained weights. Dataset gestur tangan yang dikembangkan secara internal dikurasi, diberi anotasi *bounding box*, kemudian dikelompokkan menjadi data latih, data validasi, dan data uji. Tahap pelatihan dijalankan pada mesin berbasis GPU dengan teknik *augmentasi*, seperti *flip*, penyesuaian *brightness/contrast*, dan *light blur* untuk meningkatkan ketahanan model terhadap variasi pencahayaan dan sudut kamera.

Dataset terdiri dari lima kelas gestur, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Setiap kelas terdiri dari 100 citra, sehingga total dataset berjumlah 500 citra. Pembagian dataset dilakukan secara *stratified* dengan rasio 70:15:15, yaitu 350 citra sebagai data latih, 75 citra sebagai data validasi, dan 75 citra sebagai data uji. Pembagian ini digunakan agar proporsi setiap kelas tetap seimbang dan data latih memiliki jumlah yang lebih memadai untuk proses pembelajaran model. Jumlah dan pembagian dataset ditunjukkan pada Tabel 3.

Tabel 3. Jumlah Data Model.

No	Nama Kelas	Jumlah	Train	Val	Test
1	Maju	100	70	15	15
2	Mundur	100	70	15	15
3	Belok kanan	100	70	15	15
4	Belok kiri	100	70	15	15
5	Berhenti	100	70	15	15
Total	5 kelas	500	350	75	75

Pada tahap *training*, hasil pelatihan menunjukkan bahwa model mampu memperoleh performa tinggi pada setiap kelas gestur, sebagaimana ditunjukkan pada Tabel 4.

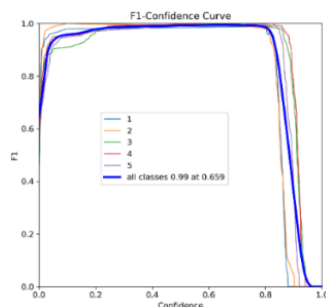
Tabel 4. Performa Hasil Model YOLO

<i>Gesture</i>	<i>Class</i>	<i>Precision</i>	<i>Recall</i>	<i>mAP</i> ₅₀	<i>mAP</i> ₅₀₋₉₅
All	-	0,992	0,996	0,995	0,784
Maju	1	0,975	1	0,994	0,751
Mundur	2	1	0,983	0,995	0,779
Belok Kanan	3	0,999	1	0,995	0,790
Belok Kiri	4	1	0,997	0,995	0,776
Berhenti	5	0,989	1	0,995	0,826

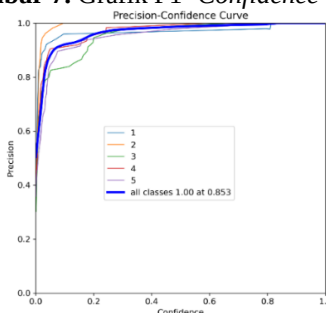
Proses validasi model YOLOv11 menghasilkan performa yang baik. Secara umum, nilai mAP50 mencapai 0,995 dan mAP50-95 sebesar 0,784. Nilai mAP50 yang tinggi menunjukkan bahwa model mampu mendeteksi gestur tangan dengan baik pada ambang IoU 0,50. Namun, nilai mAP50-95 yang lebih rendah menunjukkan bahwa model masih memiliki keterbatasan dalam menghasilkan posisi *bounding box* yang sangat presisi pada ambang IoU yang lebih ketat.

Hasil uji per kelas juga menunjukkan performa yang konsisten, dengan nilai *precision* dan *recall* mendekati 1,000. Hal ini menunjukkan bahwa YOLOv11 mampu mengenali sebagian besar gestur tangan dengan kesalahan prediksi yang rendah. Namun, performa validasi yang tinggi tetap perlu dibandingkan dengan hasil pengujian langsung menggunakan kamera untuk memastikan kestabilan model pada kondisi nyata.

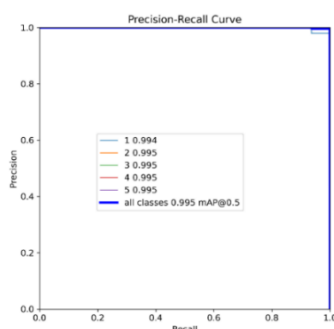
Jika dianalisis lebih lanjut, perbedaan antara mAP50 dan mAP50-95 mengindikasikan bahwa model sudah baik dalam mengenali kelas gestur, tetapi masih terdapat potensi deviasi pada penempatan *bounding box*. Kondisi ini dapat dipengaruhi oleh variasi posisi tangan, sudut kamera, pencahayaan, dan latar belakang. Selain aspek akurasi, waktu inferensi model juga diperhatikan karena sistem ditujukan untuk kendali robot secara *real-time*. Pada hasil pengujian, rata-rata waktu inferensi YOLOv11 adalah sekitar 1,58 ms per citra. Grafik evaluasi model YOLOv11 ditunjukkan pada Gambar 7, Gambar 8, Gambar 9, dan Gambar 10.



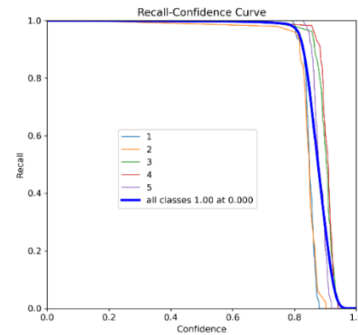
Gambar 7. Grafik F1- Confidence Curve



Gambar 8. Grafik Precision-Confidence Curve

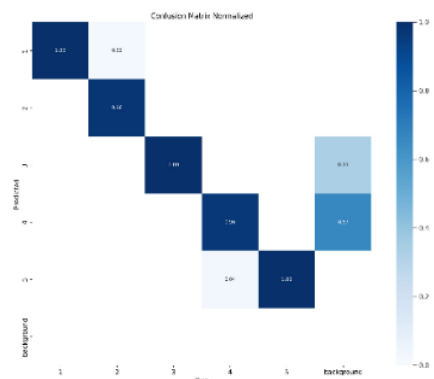


Gambar 9. Grafik Precision-Recall Curve



Gambar 10. Grafik Recall-Confidence Curve

Berdasarkan Gambar 11, model mampu mengenali sebagian besar kelas dengan baik, ditunjukkan oleh nilai diagonal utama yang mendekati 1,00. Kelas 1, 3, dan 5 memperoleh nilai prediksi benar sebesar 1,00, sedangkan kelas 2 dan 4 masing-masing memperoleh nilai 0,98 dan 0,96. Namun, masih terdapat sedikit kesalahan prediksi, terutama pada kelas *background* yang terklasifikasi sebagai kelas 3 dan 4. Hal ini menunjukkan bahwa peningkatan variasi dataset dan kualitas anotasi masih diperlukan untuk memperkuat kemampuan generalisasi model.



Gambar 11. Confusion Matrix YOLOv11

3.2. Hasil Pengujian YOLO

Tahap pengujian dilakukan dengan menjalankan model YOLOv11 pada komputer yang terhubung dengan kamera. Model diuji secara langsung untuk mengenali berbagai gestur tangan di depan kamera pada kondisi pencahayaan normal. Pengujian ini bertujuan untuk mengetahui kestabilan model dalam mengenali gestur secara *real-time* berdasarkan nilai *confidence* yang dihasilkan. Hasil pengujian gestur dan kontrol robot ditunjukkan pada Tabel 5.

Tabel 5. Hasil Pengujian Model

No	Gestur	Robot	Nilai Akurasi
1			0.90

No	Gestur	Robot	Nilai Akurasi
2			0.82
3			0.85
4			0.80
5			0.85

Pada Tabel 5 disajikan hasil deteksi gestur tangan menggunakan model YOLOv11 dan gerak robot yang dihasilkan. Pengujian dilakukan terhadap lima jenis gestur, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti. Nilai pada tabel menunjukkan *mean confidence*, yaitu tingkat keyakinan model saat mendeteksi gestur secara langsung menggunakan kamera. Nilai ini tidak dihitung sebagai mAP karena mAP merupakan metrik evaluasi deteksi objek yang dihitung berdasarkan *Average Precision* dan ambang *Intersection over Union* (IoU).

Rata-rata *confidence* pengujian langsung dihitung menggunakan Persamaan (2).

$$\text{Mean Conf} = \frac{1}{n} \sum_{i=1}^n C_i \quad (2)$$

Keterangan:

C_i = Nilai *confidence* pada setiap kelas gestur

n = Jumlah kelas gestur yang diuji

Berikut Perhitungannya:

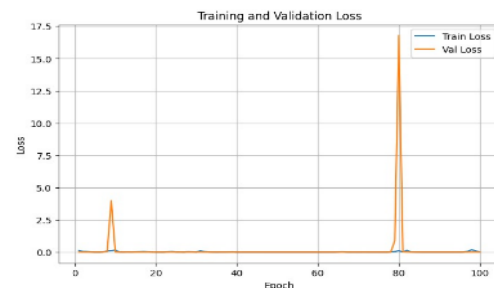
$$\text{Mean Conf} = \frac{0,90 + 0,82 + 0,85 + 0,80 + 0,85}{5} = \frac{4,22}{5} = 0,844 = 84,4\%$$

3.3. Hasil Training Model ResNet-50

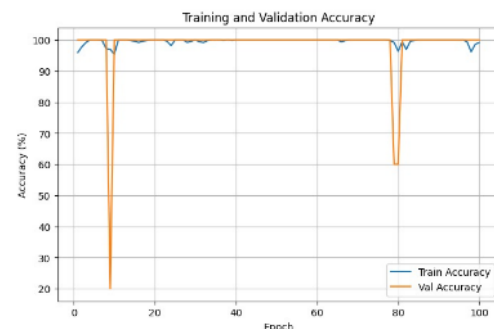
Pelatihan model dengan arsitektur ResNet-50 dilakukan untuk mengelompokkan citra gestur tangan ke dalam beberapa kelas yang telah ditentukan. Model dilatih menggunakan pendekatan *transfer learning*. Citra terlebih dahulu melalui tahap prapemrosesan berupa perubahan ukuran menjadi 224×224 piksel, normalisasi, dan *augmentasi* data. Tahapan tersebut dilakukan untuk menyesuaikan dimensi citra dengan masukan ResNet-50 dan meningkatkan kemampuan model dalam mempelajari variasi visual gestur tangan. Selama proses pelatihan, performa model dipantau menggunakan nilai akurasi dan fungsi *loss* pada data

training dan *validation*. Berdasarkan kurva pada Gambar 12, Gambar 13 dan Gambar 14, nilai akurasi *training* dan *validation* meningkat dengan cepat hingga mendekati 100% setelah beberapa *epoch* awal, sedangkan nilai *loss* menurun hingga mendekati nol. Pola tersebut menunjukkan bahwa model mampu mempelajari karakteristik visual pada dataset internal dengan cepat.

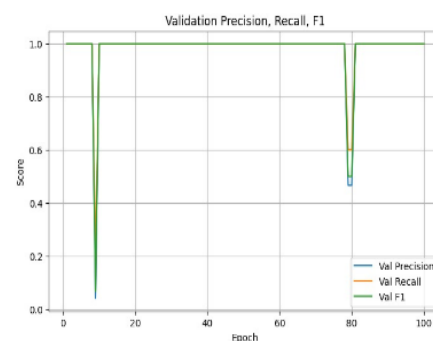
Meskipun demikian, kurva *validation loss* menunjukkan lonjakan pada sekitar *epoch* ke-10 dan ke-80. Pada *epoch* yang sama, *validation accuracy*, *precision*, *recall*, dan F1-score mengalami penurunan sementara sebelum kembali meningkat. Kondisi ini menunjukkan adanya ketidakstabilan prediksi pada sejumlah data validasi selama proses pelatihan. Lonjakan tersebut dapat dipengaruhi oleh ukuran dataset yang terbatas, karakteristik batch validasi, variasi hasil *augmentasi*, atau sensitivitas model terhadap citra tertentu. Oleh karena itu, nilai akhir yang tinggi tidak langsung menunjukkan bahwa model memiliki generalisasi sempurna terhadap kondisi di luar dataset.



Gambar 12. *Validation Loss* Model ResNet-50



Gambar 13. Kurva *Training* dan *Validation Accuracy* Model ResNet-50



Gambar 14. Kurva *Validation Precision*, *Recall*, dan F1-Score Model ResNet-50

Hasil evaluasi internal model ResNet-50 ditunjukkan pada Tabel 6. Berdasarkan *output* evaluasi yang diperoleh, nilai *precision*, *recall*, dan *F1-score* mencapai 1,000 pada seluruh kelas gestur.

Tabel 6. Performa Evaluasi Internal Model ResNet-50

Class	Image	Instances	Precision	Recall	F1-Score
1	100	100	1.000	1.000	1.000
2	100	100	1.000	1.000	1.000
3	100	100	1.000	1.000	1.000
4	100	100	1.000	1.000	1.000
5	100	100	1.000	1.000	1.000
All	500	500	1.000	1.000	1.000

Nilai *precision*, *recall*, dan *F1-score* sebesar 1,000 menunjukkan bahwa seluruh citra pada evaluasi internal dapat dikenali sesuai kelasnya. Hasil tersebut juga terlihat pada *confusion matrix* dalam Gambar 15, yang memperlihatkan seluruh nilai prediksi berada pada diagonal utama dan tidak ditemukan kesalahan klasifikasi pada data yang dievaluasi.

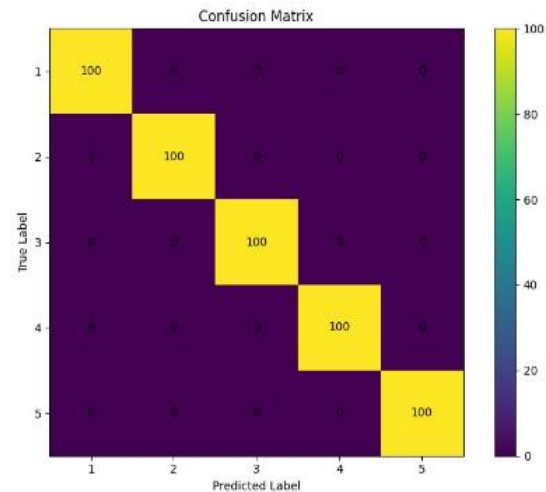
Namun, hasil sempurna tersebut harus diinterpretasikan secara hati-hati. Jumlah dataset yang relatif terbatas dan karakteristik citra yang cenderung homogen dapat menyebabkan model mengenali pola dataset internal dengan sangat mudah. Selain itu, apabila terdapat kemiripan tinggi antara citra pada data *training*, *validation*, dan *testing*, hasil evaluasi dapat menjadi terlalu optimistis. Dengan demikian, nilai 1,000 pada evaluasi internal dapat menjadi indikasi *overfitting*, kemiripan distribusi antarsubset, atau kemungkinan data *leakage* yang perlu diverifikasi melalui pemeriksaan pembagian dataset.

Indikasi tersebut semakin terlihat ketika model diterapkan secara langsung menggunakan kamera untuk mengendalikan *Differential Drive Mobile Robot*. Pada pengujian langsung, ResNet-50 hanya menghasilkan nilai *confidence* pada rentang 0,32–0,77. Perbedaan antara metrik evaluasi internal yang mencapai 1,000 dan *confidence* pengujian langsung yang lebih rendah menunjukkan adanya domain gap antara citra dataset dan citra pada lingkungan operasional.

Domain gap tersebut dapat disebabkan oleh variasi pencahayaan, jarak tangan terhadap kamera, posisi dan sudut tangan, ukuran objek dalam *frame*, kualitas citra, serta kondisi latar belakang. ResNet-50 sebagai model klasifikasi citra memproses keseluruhan citra masukan sehingga perubahan posisi dan skala tangan dapat memengaruhi fitur yang diekstraksi. Oleh karena itu, performa sempurna pada evaluasi internal tidak digunakan untuk menyimpulkan bahwa model memiliki akurasi 100% pada kondisi nyata.

Berdasarkan temuan tersebut, model ResNet-50 memiliki kemampuan yang sangat baik dalam mengenali pola pada dataset internal, tetapi kestabilan generalisasinya pada kondisi pengujian

langsung masih terbatas. Diperlukan dataset yang lebih besar dan beragam, pemisahan data yang lebih ketat, serta pengujian pada berbagai kondisi pencahayaan, jarak, posisi tangan, dan latar belakang untuk memperoleh estimasi performa yang lebih representatif.



Gambar 15. *Confusion Matrix* Evaluasi Internal Model ResNet-50

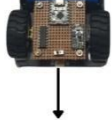
3.4. Hasil Pengujian ResNet

Pengujian langsung model ResNet-50 dilakukan untuk mengevaluasi kemampuan model dalam mengenali lima kelas gestur tangan pada kondisi operasional menggunakan kamera. Citra gestur yang ditangkap secara langsung diproses oleh model, kemudian hasil klasifikasi diterjemahkan menjadi perintah gerak pada *Differential Drive Mobile Robot*.

Pengujian pada bagian ini berbeda dengan evaluasi internal menggunakan data *testing*. Evaluasi internal menggunakan *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*, sedangkan pengujian langsung dianalisis berdasarkan nilai *confidence* yang dihasilkan model untuk setiap gestur. Nilai *confidence* menunjukkan tingkat keyakinan model terhadap kelas gestur yang diprediksi dan tidak dapat dinyatakan sebagai nilai akurasi.

Hasil pengujian langsung ResNet-50 pada lima kelas gestur ditunjukkan pada Tabel 7.

Tabel 7. Pengujian Langsung Model ResNet-50

No	Gestur	Robot	Nilai Akurasi
1			0.39
2			0.32

No	Gestur	Robot	Nilai Akurasi
3			0.38
4			0.35
5			0.77

Berdasarkan Tabel 7, nilai *confidence* ResNet-50 pada pengujian langsung berada pada rentang 0,32–0,77. Nilai *confidence* tertinggi diperoleh pada gestur berhenti sebesar 0,77, sedangkan nilai terendah diperoleh pada gestur mundur sebesar 0,32. Gestur maju, belok kanan, dan belok kiri masing-masing menghasilkan nilai *confidence* sebesar 0,39, 0,38, dan 0,35. Rata-rata *confidence* pengujian langsung ResNet-50 dihitung menggunakan Persamaan (3).

$$\text{Mean Conf} = \frac{1}{n} = \sum_{i=1}^n C_i \quad (3)$$

Keterangan:

C_i = Nilai *confidence* pada setiap gestur

n = Jumlah kelas gestur yang diujin

Perhitungan rata-rata *confidence* ResNet-50 adalah sebagai berikut:

$$\text{Mean Conf} = \frac{0,39 + 0,32 + 0,38 + 0,35 + 0,77}{5}$$

$$\text{Mean Conf} = \frac{2,21}{5} = 0,442$$

Dengan demikian, rata-rata *confidence* ResNet-50 pada pengujian langsung adalah 0,442 atau 44,2%. Nilai tersebut tidak menunjukkan bahwa akurasi model hanya sebesar 44,2%, tetapi menggambarkan rata-rata tingkat keyakinan model ketika mengenali gestur dari citra kamera secara langsung.

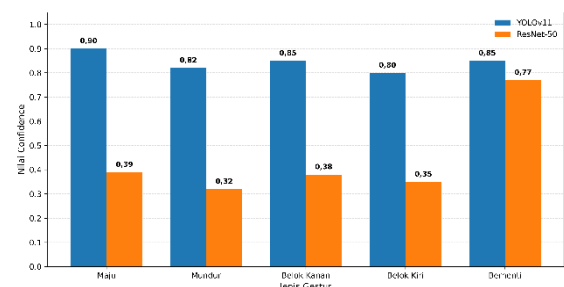
Nilai *confidence* pengujian langsung yang lebih rendah dibandingkan performa evaluasi internal menunjukkan adanya kesenjangan antara karakteristik citra pada dataset dan citra pada lingkungan operasional. Perbedaan pencahayaan, posisi dan sudut tangan, jarak terhadap kamera, ukuran tangan dalam *frame*, serta kondisi latar belakang dapat memengaruhi fitur visual yang diekstraksi oleh ResNet-50.

Sebagai model klasifikasi citra, ResNet-50 memproses keseluruhan citra masukan tanpa secara eksplisit menentukan lokasi tangan melalui *bounding box*. Oleh karena itu, perubahan posisi dan skala objek dalam *frame* dapat menyebabkan tingkat

keyakinan prediksi menurun. Hasil ini menunjukkan bahwa performa tinggi pada evaluasi internal belum sepenuhnya dapat dipertahankan ketika model diterapkan secara langsung pada sistem kendali robot.

3.5. Analisis Hasil

Berdasarkan hasil pengujian, model YOLOv11 dan ResNet-50 memiliki kelebihan dan kekurangan masing-masing. ResNet-50 menunjukkan performa yang sangat baik pada evaluasi internal karena mampu mencapai nilai *precision*, *recall*, dan *F1-score* yang tinggi pada setiap kelas gestur. Namun, ketika dilakukan pengujian secara langsung menggunakan kamera, nilai *confidence* ResNet-50 masih berada di bawah YOLOv11. Hal ini menunjukkan bahwa performa tinggi pada evaluasi internal belum tentu dapat dipertahankan ketika model diterapkan pada kondisi operasional nyata.



Gambar 11. Grafik Perbandingan Nilai Confidence YOLOv11 dan ResNet-50 pada Pengujian Langsung

Berdasarkan perbandingan nilai *confidence* YOLOv11 dan ResNet-50 yang ditunjukkan pada Gambar 11, YOLOv11 menghasilkan nilai *confidence* yang lebih tinggi pada seluruh kelas gestur. Pada gestur maju, YOLOv11 memperoleh nilai 0,90, sedangkan ResNet-50 memperoleh 0,39. Pada gestur mundur, YOLOv11 memperoleh 0,82, sedangkan ResNet-50 memperoleh 0,32. Untuk gestur belok kanan dan belok kiri, YOLOv11 masing-masing memperoleh nilai 0,85 dan 0,80. Sementara itu, ResNet-50 memperoleh nilai 0,38 dan 0,35. Pada gestur berhenti, ResNet-50 menunjukkan hasil yang cukup baik dengan nilai 0,77, tetapi masih berada di bawah YOLOv11 yang memperoleh nilai 0,85.

Perbedaan hasil tersebut menunjukkan bahwa YOLOv11 menghasilkan nilai *confidence* yang lebih tinggi dan lebih konsisten dalam mengenali gestur tangan secara langsung. Hal ini dapat dipengaruhi oleh karakteristik YOLOv11 sebagai model deteksi objek yang mampu mengenali kelas gestur sekaligus menentukan posisi objek melalui *bounding box*. Dengan kemampuan tersebut, YOLOv11 dapat memusatkan proses deteksi pada area tangan meskipun posisi dan ukuran objek berubah di dalam *frame* kamera.

Sementara itu, ResNet-50 lebih berfokus pada klasifikasi citra. Model ini mampu mempelajari pola data *training* dengan baik, tetapi diduga lebih sensitif terhadap perubahan posisi tangan, jarak kamera, pencahayaan, sudut pengambilan citra, dan kondisi

latar belakang. Perbedaan karakteristik antara citra pada dataset dan citra yang diperoleh secara langsung dari kamera menunjukkan adanya *domain gap*. Kondisi ini dapat menyebabkan nilai *confidence* ResNet-50 pada pengujian langsung menjadi lebih rendah dibandingkan YOLOv11.

Secara keseluruhan, YOLOv11 memiliki rata-rata *confidence* sebesar 0,844 atau 84,4%, sedangkan ResNet-50 memperoleh rata-rata *confidence* sebesar 0,442 atau 44,2%. Nilai tersebut merupakan rata-rata *confidence* model pada pengujian langsung dan tidak dapat dinyatakan sebagai nilai *accuracy* atau mAP. Dengan demikian, berdasarkan dataset, konfigurasi model, dan kondisi pengujian yang digunakan dalam penelitian ini, YOLOv11 menunjukkan performa pengujian langsung yang lebih baik karena menghasilkan nilai *confidence* yang lebih tinggi dan lebih konsisten pada seluruh kelas gestur. Oleh karena itu, YOLOv11 lebih sesuai digunakan pada sistem deteksi gestur tangan untuk kendali *Differential Drive Mobile Robot* dalam skenario pengujian penelitian ini.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

1. Penggunaan model *deep learning* pada sistem pengenalan gestur tangan mampu mendukung kendali *Differential Drive Mobile Robot* secara lebih intuitif melalui lima perintah gerak, yaitu maju, mundur, belok kanan, belok kiri, dan berhenti.
2. ResNet-50 menunjukkan performa yang sangat tinggi pada evaluasi internal. Namun, pada pengujian langsung menggunakan kamera, model menghasilkan nilai *confidence* pada rentang 0,32–0,77 dengan rata-rata sebesar 0,442 atau 44,2%.
3. YOLOv11 menghasilkan nilai *confidence* yang lebih tinggi dan lebih konsisten pada pengujian langsung, yaitu pada rentang 0,80–0,90 dengan rata-rata sebesar 0,844 atau 84,4%.
4. Perbedaan antara hasil evaluasi internal dan pengujian langsung menunjukkan adanya *domain gap* antara citra dataset dan citra dari kamera. Kondisi pencahayaan, posisi dan sudut tangan, jarak kamera, ukuran tangan dalam *frame*, serta latar belakang diduga memengaruhi nilai *confidence* model.
5. Berdasarkan dataset, konfigurasi model, dan kondisi pengujian dalam penelitian ini, YOLOv11 lebih sesuai digunakan untuk pengenalan gestur tangan pada kendali *Differential Drive Mobile Robot* karena menghasilkan nilai *confidence* yang lebih

tinggi dan lebih konsisten dibandingkan ResNet-50.

6. Hasil penelitian ini masih terbatas pada dataset berjumlah 500 citra dan kondisi pengujian tertentu. Penelitian selanjutnya perlu menggunakan dataset yang lebih besar dan bervariasi, melibatkan lebih banyak pengguna, serta melakukan pengujian terstruktur terhadap pencahayaan, jarak, *occlusion*, latar belakang, waktu inferensi, latensi komunikasi, dan waktu respon robot.

PUSTAKA

- Agustiani, A. D., Sholahuddin, M. R., Putri, S. M., & Hidayatullah, P. (2024). Penggunaan MediaPipe untuk pengenalan gesture tangan real-time dalam pengendalian presentasi. *Media Jurnal Informatika*, 16(2), 147–153. <https://doi.org/10.35194/mji.v16i2.4788>
- Al Amin, I. H., & Aprilino, A. (2022). Implementasi algoritma YOLO dan Tesseract OCR pada sistem deteksi plat nomor otomatis. *Jurnal Teknoinfo*, 16(1), 54–59. <https://doi.org/10.33365/jti.v16i1.1522>
- Al Rivan, M. E., & Setiawan, A. (2022). Pengenalan gestur angka pada tangan menggunakan arsitektur AlexNet dan LeNet pada metode convolutional neural network. *Komputika: Jurnal Sistem Komputer*, 11(1), 19–28. <https://doi.org/10.34010/komputika.v11i1.5176>
- Arifah, I. I., Fajri, F. N., & Pratamasunu, G. Q. O. (2022). Deteksi tangan otomatis pada video percakapan bahasa isyarat Indonesia menggunakan metode YOLO dan CNN. *Journal of Applied Informatics and Computing*, 6(2), 171–176. <https://doi.org/10.30871/jaic.v6i2.4694>
- Azhar, A. R., Setiawan, D. A., Yasmin, N. A. A., Putri, T. A., & Nama, G. F. (2024). Sistem monitoring kapasitas air dan pengisian otomatis berbasis IoT menggunakan modul ESP8266. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(1), 218–228. <https://doi.org/10.23960/jitet.v12i1.3966>
- Behar, N., & Shrivastava, M. (2022). ResNet50-based effective model for breast cancer classification using histopathology images. *Computer Modeling in Engineering & Sciences*, 130(2), 823–839. <https://doi.org/10.32604/cmescs.2022.017030>
- Berliani, T., Rahardja, E., & Septiana, L. (2023). Perbandingan kemampuan klasifikasi citra X-ray paru-paru menggunakan transfer learning ResNet-50 dan VGG-16. *Journal of Medicine and Health*, 5(2), 123–135. <https://doi.org/10.28932/jmh.v5i2.6116>
- Budiman, S. N., Lestanti, S., Evvandri, S. M., & Putri, R. K. (2022). Pengenalan gestur gerakan jari

- untuk mengontrol volume di komputer menggunakan library OpenCV dan MediaPipe. *Antivirus: Jurnal Ilmiah Teknik Informatika*, 16(2), 223–232. <https://doi.org/10.35457/antivirus.v16i2.2508>
- Chalik, A. M., Qowy, B. A., Hanafi, F., & Nuraminah, A. (2021). Mouse tracking tangan dengan klasifikasi gestur menggunakan OpenCV dan MediaPipe. *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, 1(2), 10–18. <https://doi.org/10.55606/juitik.v1i2.323>
- Elevan, R. T., Hermanto, D., & Widiyanto, E. P. (2025). Deteksi jumlah kendaraan roda empat menggunakan YOLO. *Integrative Perspectives of Social and Science Journal*, 2(2), 1647–1666. <https://ipssj.com/index.php/ojs/article/view/245>
- Fuad, M., Umam, F., Wahyuni, S., Fahriani, N., Nurwahyudi, I., Darwaman, M. I., & Maulana, F. (2023). Towards controlling mobile robot using upper human body gesture based on convolutional neural network. *Journal of Robotics and Control (JRC)*, 4(6), 856–867. <https://doi.org/10.18196/jrc.v4i6.20399>
- Golfantara, M. F. (2024). Penggunaan algoritma YOLOv8 untuk identifikasi rempah-rempah. *Jurnal Informatika dan Teknik Elektro Terapan*, 12(3S1), 3867–3873. <https://doi.org/10.23960/jitet.v12i3S1.5221>
- Islam, W., Jones, M., Faiz, R., Sadeghipour, N., Qiu, Y., & Zheng, B. (2022). Improving performance of breast lesion classification using a ResNet50 model optimized with a novel attention mechanism. *Tomography*, 8(5), 2411–2425. <https://doi.org/10.3390/tomography8050200>
- Karyadi, B. (2023). Pemanfaatan kecerdasan buatan dalam mendukung pembelajaran mandiri. *EDUCATE: Jurnal Teknologi Pendidikan*, 8(2), 253–258. <https://doi.org/10.32832/educate.v8i02.14843>
- Kemala, A. P., Syahputra, M. E., Lucky, H., & Achmad, S. (2022). Pengembangan smart air condition control menggunakan platform Blynk berbasis mikrokontroler ESP8266 dan sensor DHT11. *Engineering, Mathematics and Computer Science Journal*, 4(1), 19–23. <https://doi.org/10.21512/emacsjournal.v4i1.8072>
- Latif, M. (2020). Leader-follower formation tracking of multiple mobile robots with constant leader velocity. *Journal of Physics: Conference Series*, 1569(3), Article 032084. <https://doi.org/10.1088/1742-6596/1569/3/032084>
- Zhao, C. H., & Leong, W. Y. (2025). Optimisation solutions and simple innovative solution research on ResNet50 model. *ASM Science Journal*, 20(1), 1–9. <https://doi.org/10.32802/asmscj.2025.2002>
- Muliawan, N. B., & Sulistijono, I. A. (2023). Sistem lokalisasi mobile-robot pertanian otonom berbasis ultra-wideband (UWB) dan sensor inersia. *The Indonesian Journal of Computer Science*, 12(1), 284–301. <https://doi.org/10.33022/ijcs.v12i1.3135>
- Nurwibowo, B. D., & Saikhu, A. (2025). Pengembangan model klasifikasi kejang epilepsi multiclass pada sinyal EEG menggunakan CNN+Bi-LSTM. *Jurnal Teknika*, 17(1), 21–28. <https://doi.org/10.30736/jt.v17i1.1354>
- Rifa'i, A., & Wardhani, D. F. (2025). Implementasi hand gesture recognition berbasis computer vision untuk kontrol permainan Subway Surfers. *Jurnal Sintaks Logika*, 5(2), 138–146. <https://doi.org/10.31850/jsilog.v5i2.3772>
- Ritonga, A. A., & Hasibuan, E. R. (2025). Eksplorasi masa depan robotika: Integrasi kecerdasan buatan dalam sistem automasi adaptif berbasis lingkungan. *Journal Computer Science and Information Technology (JCoInT)*, 6(1), 387–393. <https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/7831>
- Roziqin, M. F., Joni, K., & Ulum, M. (2021). Rancang bangun robot pengambil bola tenis lapangan dengan kendali hand gesture recognition berbasis Wi-Fi menggunakan metode PID. *Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer*, 1(1), 33–40. <https://doi.org/10.32503/jtecs.v1i1.1428>
- Saputra, D. H., Imran, B., & Juhartini. (2023). Object detection untuk mendeteksi citra buah-buahan menggunakan metode YOLO. *Jurnal Kecerdasan Buatan dan Teknologi Informasi*, 2(2), 70–80. <https://doi.org/10.69916/jkbti.v2i2.18>
- Wajidi, M. F., & Hajiri, M. I. (2024). Tantangan adaptasi kecerdasan buatan dalam layanan perpustakaan perguruan tinggi. *Tik Ilmeu: Jurnal Ilmu Perpustakaan dan Informasi*, 8(2), 217–228. <https://doi.org/10.29240/tik.v8i2.10901>
- Wijanarko, R. G., Pradana, A. I., & Hartanti, D. (2024). Implementasi deteksi drone menggunakan YOLO (You Only Look Once). *Jurnal Fasilkom*, 14(2), 437–442. <https://doi.org/10.37859/jf.v14i2.7374>
- Zaenudin, I., & Riyan, A. B. (2024). Perkembangan kecerdasan buatan (AI) dan dampaknya pada dunia teknologi. *Jurnal Informatika Utama*, 2(2), 128–153. <https://doi.org/10.55903/jitu.v2i2.240>