BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital telah membawa perubahan signifikan dalam berbagai bidang kehidupan, termasuk pendidikan dan kesehatan. Digitalisasi memungkinkan aktivitas pembelajaran dan latihan yang sebelumnya bersifat konvensional menjadi lebih interaktif, terukur, dan adaptif terhadap pengguna. Pendekatan digital berpotensi meningkatkan keterlibatan pengguna (*engagement*) serta kepatuhan dalam melakukan latihan secara rutin, karena adanya umpan balik langsung dan visualisasi performa yang mendukung pemantauan progres dan motivasi pengguna (Nugraha et al., 2025). Dengan demikian, digitalisasi tidak hanya berperan sebagai sarana penyampaian materi, tetapi juga sebagai alat pendukung untuk meningkatkan efektivitas dan keberlanjutan suatu aktivitas latihan.

Salah satu bentuk latihan yang relevan untuk dikembangkan dalam pendekatan digital adalah senam otak tangan. Senam otak tangan merupakan latihan koordinasi motorik halus yang dirancang untuk menstimulasi aktivitas otak melalui gerakan jari dan tangan. Latihan ini banyak digunakan dalam bidang pendidikan, terapi kognitif, serta pengembangan konsentrasi karena dapat membantu meningkatkan fokus, koordinasi, dan ketajaman fungsi kognitif. Latihan koordinasi tangan yang dilakukan secara rutin juga berpotensi meningkatkan daya ingat serta menurunkan risiko penurunan fungsi kognitif pada lansia (Faleri et al., 2024). Oleh karena itu, senam otak tangan memiliki relevansi tinggi sebagai aktivitas stimulasi kognitif yang sederhana, murah, dan dapat dilakukan oleh berbagai kelompok usia.

Namun, efektivitas senam otak tangan sangat bergantung pada ketepatan gerakan yang dilakukan. Dalam praktiknya, evaluasi ketepatan gerakan masih dilakukan secara manual melalui observasi instruktur atau panduan visual. Pendekatan ini menimbulkan dua permasalahan utama, yaitu ketidakobjektifan penilaian serta keterbatasan instruktur dalam memberikan koreksi secara detail pada setiap individu. Kondisi tersebut semakin terlihat pada sesi latihan kelompok, seperti kegiatan pembelajaran di sekolah maupun terapi/fisioterapi di ruangan dengan banyak peserta, di mana instruktur sulit memantau seluruh gerakan secara teliti dan konsisten. Akibatnya, kesalahan gerakan dapat terjadi berulang tanpa disadari, sehingga manfaat latihan tidak tercapai secara optimal.

Di sisi lain, meningkatnya ketergantungan masyarakat terhadap perangkat digital turut memengaruhi pola aktivitas fisik dan kemampuan fokus, khususnya pada anak-anak dan remaja. Aktivitas pasif yang berulang dapat memicu tingginya distraksi serta kelelahan kognitif. Survei menunjukkan bahwa penggunaan perangkat digital secara berlebihan berkorelasi dengan penurunan konsentrasi, meningkatnya distraksi, serta melemahnya kemampuan regulasi diri pada siswa (Susanti et al., 2025). Kondisi ini menegaskan pentingnya menghadirkan bentuk latihan yang tidak

hanya menstimulasi otak, tetapi juga mampu menarik keterlibatan pengguna agar latihan dilakukan secara konsisten.

Berdasarkan permasalahan tersebut, terdapat urgensi untuk menghadirkan sistem evaluasi gerakan senam otak tangan yang bersifat otomatis dan objektif. Tanpa adanya umpan balik yang konsisten, kesalahan gerakan berpotensi menjadi kebiasaan dan menurunkan efektivitas latihan. Sistem evaluasi otomatis diharapkan mampu mendukung latihan mandiri (*self-training*), memungkinkan pengguna berlatih kapan saja tanpa ketergantungan penuh pada instruktur. Selain itu, keberadaan umpan balik real-time dapat meningkatkan efektivitas latihan karena pengguna memperoleh koreksi langsung saat gerakan dilakukan. Sistem semacam ini juga memungkinkan penerapan senam otak tangan dalam skala besar, misalnya di lingkungan sekolah, komunitas, maupun program latihan lansia, tanpa keterbatasan pengawasan instruktur.

Meskipun demikian, hingga saat ini belum banyak sistem yang mampu mendeteksi dan mengevaluasi gerakan senam otak tangan secara otomatis. Sebagian besar latihan masih mengandalkan instruksi visual atau pelatihan langsung tanpa dukungan teknologi yang mampu menganalisis pose tangan secara presisi. Tantangan utama dalam pengembangan sistem tersebut terletak pada kompleksitas pola jari, variasi bentuk tangan antar individu, perbedaan pencahayaan, serta kondisi latar belakang yang beragam. Oleh karena itu, diperlukan metode pendeteksian pose tangan yang tidak hanya akurat, tetapi juga cepat, stabil, dan adaptif terhadap kondisi nyata.

Perkembangan teknologi *computer vision* dan *deep learning* membuka peluang besar untuk mengatasi permasalahan tersebut. Salah satu pendekatan yang banyak digunakan dalam analisis citra adalah *Convolutional Neural Network* (CNN), yang mampu mengekstraksi fitur spasial secara bertingkat. Model modern berbasis CNN seperti YOLOv11-Pose merupakan model deteksi objek yang dilengkapi dengan kemampuan mengidentifikasi titik kunci (*keypoint*), sehingga memungkinkan analisis pose tangan secara real-time dengan performa yang cepat dan presisi. Karakteristik ini menjadikan YOLOv11-Pose potensial untuk diimplementasikan dalam sistem evaluasi gerakan senam otak tangan.

Namun, pelatihan model pose detection membutuhkan dataset dengan label keypoints yang presisi. Proses pelabelan manual terhadap 21 keypoints tangan pada ribuan citra membutuhkan waktu yang panjang dan berpotensi menimbulkan ketidakkonsistenan anotasi. Oleh karena itu, diperlukan pendekatan yang lebih efisien dalam pembentukan dataset keypoint. MediaPipe Hands merupakan pustaka yang mampu mengekstraksi 21 landmark tangan secara cepat dan relatif stabil, sehingga dapat dimanfaatkan sebagai pseudo-labeling untuk membangun anotasi keypoints secara otomatis. Integrasi MediaPipe pada tahap pra-pemrosesan memungkinkan proses pelabelan menjadi lebih efisien serta membantu menjaga konsistensi data, sehingga model YOLOv11-Pose dapat dilatih dengan kualitas dataset yang lebih baik.

Beberapa penelitian sebelumnya menunjukkan bahwa kombinasi ekstraksi keypoint dan deep learning mampu memberikan performa tinggi pada pengenalan

gestur tangan. Penerapan CNN pada sistem pengenalan gestur tangan dilaporkan mencapai akurasi 98,14% pada dataset MU dengan waktu deteksi sekitar 0,5219 detik per gestur (Sahoo et al., 2022). Namun, sebagian besar penelitian tersebut berfokus pada gestur umum atau bahasa isyarat, sementara penerapan khusus untuk gerakan senam otak tangan masih sangat terbatas. Selain itu, kajian yang mengintegrasikan keypoints MediaPipe secara langsung ke dalam pipeline pelatihan YOLO-Pose untuk menghasilkan deteksi gerakan yang stabil pada lingkungan nyata juga masih relatif sedikit.

Berdasarkan celah tersebut, penelitian ini difokuskan pada perancangan dan pembangunan model deteksi gerakan senam otak tangan menggunakan arsitektur YOLOv11-Pose dengan dukungan ekstraksi keypoint dari MediaPipe Hands. Model yang dikembangkan diarahkan untuk mengenali enam jenis gerakan, yaitu *thumb_side*, *default*, *little-finger*, *peace*, *pistol*, dan *back-slap*. Sistem ini diharapkan mampu mendeteksi pose tangan secara presisi serta memberikan stabilitas deteksi yang tinggi meskipun terdapat variasi pencahayaan dan latar belakang. Hasil penelitian ini diharapkan dapat menjadi fondasi bagi pengembangan sistem evaluasi latihan yang lebih objektif, terukur, serta mendukung digitalisasi latihan kognitif dan terapi ringan berbasis AI.

1.2 Rumusan Masalah

Dengan mengacu pada latar belakang di atas, maka rumusan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut :

1. Bagaimana tahapan *pre-processing* data mulai dari mengumpulkan, labelling, partisi dan pembersihan data untuk membangun model deteksi gerakan untuk senam otak tangan?
2. Bagaimana membangun dan meningkatkan kinerja model deteksi gerakan tangan senam otak menggunakan YOLOv11-Pose dengan metode *enrichment* data dan *keypoint* labeling menggunakan *MediaPipe*?
3. Bagaimana kinerja model klasifikasi pose tangan menggunakan metrik *Precision, Recall, Mean Average Precision (mAP50–95),* dan F1-*Score* pada keluaran *bounding box (B*) dan *keypoint/pose (P),* serta bagaimana kemampuan deteksi model pada skenario *real-time* yang dievaluasi berdasarkan stabilitas prediksi dan *confidence score* sebagai indikator keandalan deteksi?

1.3 Batasan Masalah

Untuk menjaga fokus penelitian dan menghindari perluasan masalah, maka ditetapkan beberapa batasan sebagai berikut:

1. Model deteksi gerakan tangan yang digunakan adalah YOLOv11-Pose berbasis CNN dengan dukungan MediaPipe Hands untuk ekstraksi dan normalisasi 21 *keypoint* tangan.
2. Kelas gerakan tangan yang dikenali dibatasi pada enam gerakan: *thumb_side, default, little-finger, peace, pistol, dan back-slap*.

3. Dataset berupa citra tangan hasil ekstraksi frame video dan hasil *enrichment* data
4. Model yang dikembangkan hanya mencakup pengenalan dan klasifikasi gerakan tangan, tanpa analisis fisiologis atau pengukuran aktivitas otak.
5. Evaluasi model hanya mencakup metrik *Precision*, *Recall*, *F1-Score*, dan mAP50–95, tanpa menganalisis sudut atau pergerakan dinamis antar *keypoint*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, tujuan penelitian sebagai berikut:

1. Menghasilkan dataset yang siap digunakan untuk membangun model terbaik untuk klasifikasi 6 kelas gerakan senam otak tangan.
2. Membangun model deteksi gerakan tangan berbasis YOLOv11-Pose yang memiliki kinerja terbaik menggunakan metode *enrichment* data serta penambahan label keypoint otomatis dengan MediaPipe.
3. Mengevaluasi kinerja model klasifikasi pose tangan menggunakan metrik metrik *Precision*, *Recall*, *Mean Average Precision* (mAP50–95), dan F1-*Score* baik pada keluaran *bounding box* (B) dan *keypoint/pose* (P) serta mengevaluasi kemampuan deteksi model pada skenario real-time berdasarkan kestabilan prediksi dan *confidence score* sebagai indikator keandalan deteksi.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memiliki manfaat sebagai berikut:

1. Penelitian ini memperkaya kajian di bidang computer vision, khususnya pada pengembangan model deteksi gerakan tangan berbasis YOLOv11-Pose dan ekstraksi keypoint MediaPipe.
2. Menyediakan model yang dapat dimanfaatkan sebagai sistem deteksi gerakan senam otak tangan untuk keperluan latihan kognitif, pelatihan motorik halus, atau media pembelajaran gerakan tangan secara otomatis.
3. Memberikan contoh implementasi pipeline lengkap mulai dari pembuatan dataset, augmentasi semi-otomatis dengan MediaPipe, hingga pelatihan model pose estimation pada enam kelas gerakan tangan.

1.6 Landasan Teori

1.6.1 Senam Otak Tangan

Senam otak tangan merupakan bentuk latihan motorik halus yang bertujuan untuk menstimulasi fungsi kognitif melalui koordinasi gerakan jari dan tangan. Konsep dasar senam otak (*Brain Gym*) pertama kali diperkenalkan oleh Paul E. Dennison dan Gail E. Dennison yang menekankan pentingnya gerakan tubuh terkoordinasi untuk meningkatkan integrasi otak, fokus, dan kemampuan belajar (Dennison, P. E., & Dennison, 1989). Penelitian-penelitian terkini juga mendukung bahwa variasi gerakan jari dan tangan berkaitan erat dengan aktivasi area korteks motorik dan

prefrontal yang berperan dalam perhatian, kontrol eksekutif, dan perencanaan (Liu et al., 2025). Latihan motorik halus tangan dilaporkan mampu meningkatkan koordinasi visuomotor, konsentrasi, serta ketahanan kognitif pada berbagai kelompok usia, termasuk anak-anak dan lansia (Kong et al., 2025).

Berdasarkan landasan tersebut, enam gerakan senam otak tangan yang digunakan dalam penelitian ini dipilih karena merepresentasikan variasi konfigurasi jari, orientasi tangan, serta tingkat kompleksitas motorik yang berbeda. Setiap gerakan melibatkan kombinasi ekstensi, fleksi, dan isolasi jari tertentu yang menuntut koordinasi motorik halus dan kontrol kognitif yang bervariasi, sehingga relevan untuk melatih sinkronisasi antara otak kiri dan kanan sebagaimana ditekankan dalam konsep *Brain Gym* (Dennison, P. E., & Dennison, 1989). Selain bermanfaat secara kognitif, keenam gerakan tersebut memiliki karakteristik visual yang saling membedakan, sehingga sesuai untuk evaluasi sistem deteksi pose tangan berbasis *computer vision*. Perbedaan konfigurasi jari dan orientasi tangan memungkinkan pengujian akurasi, stabilitas, dan robustness model dalam mengenali pose tangan pada berbagai kondisi visual, menjadikan gerakan tersebut relevan baik sebagai latihan kognitif maupun sebagai objek penelitian sistem evaluasi gerakan otomatis.

Gambar 1 Senam Otak Tangan
Sumber : Hypnocare Indonesia. (2021). *SENAM OTAK GERAKAN DASAR (new version)* [Video]. YouTube. https://www.youtube.com/watch?v=A1HUh8FMCpE

1.6.2 Computer Vision

Computer vision merupakan bidang ilmu dalam *artificial intelligence* yang berfokus pada bagaimana komputer dapat memperoleh, mengolah, dan memahami informasi visual dari citra atau video, layaknya kemampuan persepsi manusia. Tujuan utamanya adalah memungkinkan mesin untuk mengenali objek, menganalisis adegan, serta mengambil keputusan berdasarkan data visual yang ditangkap oleh kamera atau sensor optik (Szeliski, 2021)**.**

Proses dalam *computer vision* melibatkan beberapa tahapan utama, yaitu akuisisi citra, pra-pemrosesan, ekstraksi fitur, dan interpretasi visual. Pada tahap akuisisi, sistem memperoleh data mentah dari kamera atau perangkat sensor. Data tersebut kemudian melalui proses prapemrosesan, seperti konversi warna, normalisasi, dan penghilangan derau, agar kualitas citra menjadi optimal. Tahap berikutnya adalah ekstraksi fitur, di mana sistem mencari pola penting dari citra yang

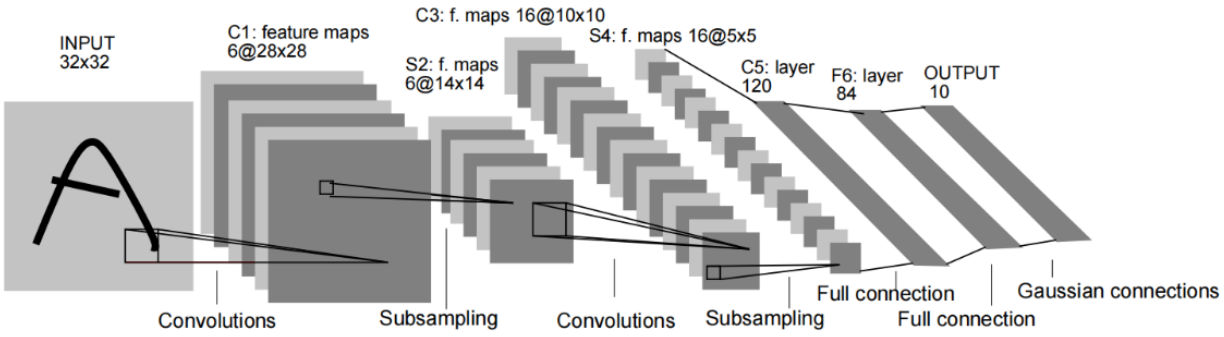
mewakili karakteristik unik objek. Terakhir, sistem melakukan interpretasi untuk mengklasifikasikan dan mengenali objek berdasarkan fitur-fitur tersebut.

1.6.3 *Deep Learning*

Deep learning merupakan subbidang dari *machine learning* yang memanfaatkan jaringan saraf tiruan berlapis banyak (*deep neural networks*) untuk mempelajari representasi data secara otomatis. Tidak seperti pendekatan tradisional yang mengandalkan rekayasa fitur manual, *deep learning* memungkinkan sistem mempelajari pola penting langsung dari data mentah melalui proses bertahap di setiap lapisan jaringan (Goodfellow et al., 2016). Kemampuan ini membuat *deep learning* menjadi salah satu teknik pembelajaran modern yang paling efektif untuk data berukuran besar, kompleks, dan memiliki struktur non-linier.

Setiap lapisan dalam jaringan *deep learning* berfungsi mengekstraksi informasi dengan tingkat abstraksi yang semakin tinggi, mulai dari pola dasar hingga struktur konseptual yang lebih kompleks. Mekanisme hierarkis ini memungkinkan model memahami hubungan spasial, temporal, maupun semantik secara lebih mendalam, sehingga meningkatkan ketelitian dalam proses pengenalan dan pengambilan keputusan (Liu et al., 2022). Proses pembelajaran berlangsung melalui optimisasi bobot secara iteratif menggunakan algoritma propagasi balik (*backpropagation*), yang memungkinkan model menyesuaikan diri terhadap variasi data. Dengan karakteristik tersebut, *deep learning* menjadi fondasi utama dalam banyak aplikasi modern, termasuk analisis citra, pemrosesan bahasa alami, dan pengenalan pola berbasis sensor.

1.6.4 Convolutional Neural Network (CNN)



Gambar 2 Arsitektur CNN (LeCun et al., 1998)

Convolutional Neural Network (CNN) merupakan arsitektur *deep learning* yang dirancang untuk mengenali pola visual dengan memanfaatkan struktur spasial pada citra. CNN pertama kali dipopulerkan melalui LeNet-5 oleh LeCun et al., pada tahun 1998 dan hingga kini menjadi fondasi utama berbagai model *computer vision* modern. Berbeda dari jaringan saraf *feed-forward* konvensional (*multilayer perceptron, MLP*) yang membutuhkan rekayasa fitur manual, CNN mampu mengekstraksi fitur secara otomatis melalui proses konvolusi berlapis.

Secara umum CNN terdiri atas empat komponen utama, yaitu *convolution layer* yang mengekstraksi pola lokal, *pooling layer* untuk menurunkan dimensi dan meningkatkan ketahanan terhadap pergeseran posisi, fungsi aktivasi (umumnya ReLU) untuk memperkenalkan non-linearitas, serta *fully connected layer* yang berfungsi melakukan klasifikasi berdasarkan fitur yang telah diekstraksi.

Operasi konvolusi didefinisikan sebagai:

$$S(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \cdot K(m,n) \quad (1)$$

Keterangan:

I = citra *input* (*input* image)
K = kernel atau filter berukuran $M \times N$
$S(i,j)$ = nilai aktivasi hasil *convolution* di lokasi (i,j)
m,n = indeks kernel

Lapisan pooling dapat berbentuk max-pooling:

$$P(i,j) = \max_{(m,n)\in R_{ij}} F(m,n) \quad (2)$$

Keterangan:

F = *feature map* hasil *convolution*
R_{ij} = area *pooling* di sekitar (i,j)
$P(i,j)$ = nilai *pooling*

dan fungsi aktivasi ReLU diberikan sebagai:

$$f(x) = \max(0, x) \quad (3)$$

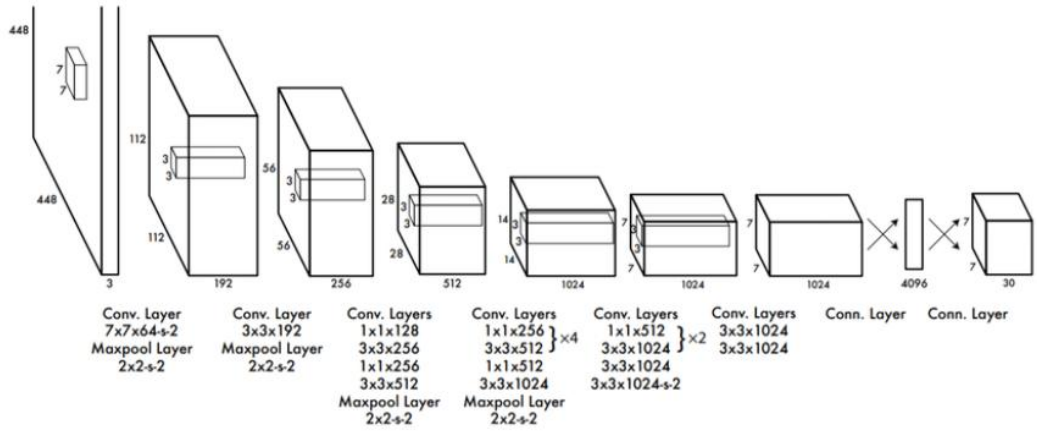
Keterangan:

x = nilai *input* neuron

CNN menjadi salah satu arsitektur paling dominan dalam pengolahan citra karena kemampuannya mengekstraksi pola visual secara otomatis melalui operasi konvolusi berlapis. Melalui mekanisme ini, CNN mampu mempelajari representasi spasial mulai dari tepi dan tekstur hingga pola bentuk yang lebih kompleks tanpa memerlukan rekayasa fitur manual. CNN juga memiliki ketahanan terhadap variasi latar belakang, perubahan pencahayaan, serta pergeseran posisi objek, sehingga menghasilkan performa yang lebih stabil dibanding metode konvensional berbasis fitur buatan manusia (*hand-crafted)* (Noreen et al., 2021). Kemampuan tersebut menjadikan CNN fondasi utama berbagai model *computer vision* modern dan relevan

untuk berbagai *task* seperti klasifikasi citra, deteksi objek, dan pengenalan pose tangan.

1.6.5 You Only Look Once (YOLO)



Gambar 3 Arsitektur Algoritma YOLO (Redmon et al., 2016)

You Only Look Once (YOLO) adalah keluarga algoritma deteksi objek berbasis CNN yang melakukan prediksi *bounding box*, skor kepercayaan, dan kelas objek secara *end-to-end*. Pendekatan ini diperkenalkan oleh Redmon et al., pada tahun 2016 sebagai alternatif yang jauh lebih cepat dibandingkan metode dua-tahap seperti R-CNN. Pada YOLO, seluruh proses dilakukan dalam satu *forward pass*, sehingga deteksi dapat dijalankan pada kecepatan *real-time*.

Secara struktural, YOLO terdiri atas *backbone* untuk mengekstraksi fitur visual, *neck* yang menggabungkan fitur multiresolusi, dan *detection head* yang memprediksi lokasi serta kelas objek. Proses deteksi boks memanfaatkan regresi langsung terhadap koordinat (x, y, w, h) dari setiap kotak prediksi. Umumnya koordinat dihitung menggunakan formulasi:

$$\hat{b} = (\hat{x}, \hat{y}, \hat{w}, \hat{h}) \tag{4}$$

Keterangan:

$\hat{x}, \hat{y}$ = koordinat pusat *bounding box* prediksi
$\hat{w}, \hat{h}$ = lebar dan tinggi prediksi *bounding box*

dengan skor keyakinan objek dievaluasi melalui:

$$\text{conf} = P(\text{object}) \cdot \text{IoU}(\hat{b}, b_{\text{gt}}) \tag{5}$$

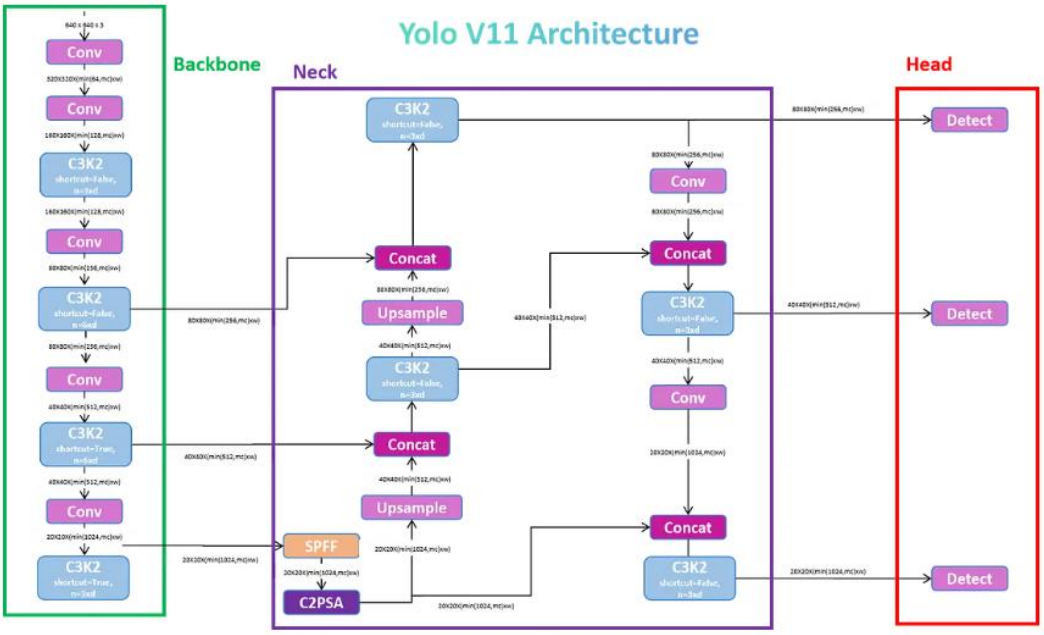
Keterangan:

$P(\text{object})$= probabilitas keberadaan objek

IoU = *Intersection over Union* antara prediksi dan *ground truth*
$\hat{b}$ = *bounding box* prediksi
b_{gt} = *bounding box ground truth*

Evolusi YOLO selama satu dekade terakhir menunjukkan peningkatan yang konsisten dalam hal kecepatan deteksi, akurasi prediksi, dan efisiensi komputasi. Setiap versi baru menghadirkan perbaikan pada struktur *backbone*, integrasi fitur multiskala, dan mekanisme optimasi prediksi, sehingga menghasilkan detektor objek yang semakin stabil, andal, dan efektif untuk berbagai kondisi visual (Sapkota et al., 2025). Perkembangan tersebut menjadikan YOLO sebagai salah satu pendekatan paling dominan dalam deteksi objek *real-time*, terutama pada aplikasi yang membutuhkan respons cepat dan presisi tinggi seperti pengenalan gestur dan analisis pose.

1.6.5.1 You Only Look Once versi 11 (YOLOv11)



Gambar 4 Arsitektur YOLOv11 (Rao, 2024)

YOLOv11 merupakan pengembangan terbaru yang dirilis oleh Ultralytics pada tahun 2024 dengan peningkatan pada stabilitas prediksi, efisiensi parameter, dan ketelitian koordinat terutama pada objek kecil. Arsitektur ini menggunakan *backbone* beresolusi tinggi serta mekanisme *hierarchical multiscale features* yang diperbarui, memungkinkan ekstraksi fitur yang lebih kaya pada berbagai skala. Selain itu, YOLOv11 sepenuhnya mengadopsi pendekatan *anchor-free* sehingga prediksi *bounding box* dilakukan secara langsung tanpa bergantung pada *anchor preset*,

menghasilkan proses deteksi yang lebih fleksibel dan adaptif terhadap variasi ukuran objek.

Komputasi yang lebih ringan pada YOLOv11 membuat model ini mampu mencapai throughput tinggi sekaligus mempertahankan akurasi deteksi pada kondisi real-time (Ultralytics, 2024). Efisiensi tersebut memungkinkan YOLOv11 berjalan stabil pada GPU kelas menengah, sehingga cocok digunakan pada skenario aplikasi yang membutuhkan respons cepat dan konsistensi prediksi, termasuk pelacakan pose tangan dan pengenalan gestur pada sistem interaktif berbasis *computer vision*. Secara keseluruhan, peningkatan efisiensi arsitektural, ketelitian deteksi, dan kemampuan generalisasi yang lebih luas menjadikan YOLOv11 solusi yang tangguh untuk menangani berbagai tantangan pengenalan visual modern, baik dalam konteks penelitian maupun implementasi praktis (Khanam & Hussain, 2024).

Selama proses pelatihan, YOLOv11 mengoptimalkan beberapa komponen loss, yaitu:

1. *Box Loss* (regresi *bounding box*)
   Menggunakan kombinasi CIoU/DIoU untuk mengukur kualitas prediksi bentuk dan posisi *bounding box*:

$$L_{\text{box}} = 1 - \text{IoU}(\hat{b}, b_{\text{gt}}) \tag{6}$$

   Keterangan:

   $\hat{b}$ = *bounding box* prediksi
   b_{gt} = *bounding box ground truth*
   IoU = *Intersection over Union*, rasio area irisan terhadap area gabungan kotak prediksi dan ground truth.

2. *Objectness Loss* (*kobj_loss*)
   Mengukur keyakinan model terhadap keberadaan objek:

$$L_{\text{obj}} = \text{BCE}(p_{\text{obj}}, \hat{p}_{\text{obj}}) \tag{7}$$

   Keterangan:

   p_{obj} = label 1 (ada objek) atau 0 (tidak ada objek)
   $\hat{p}_{\text{obj}}$ = probabilitas objek yang diprediksi model
   BCE = *Binary Cross Entropy*, fungsi yang mengukur jarak antara probabilitas prediksi dan label sebenarnya

3. *Classification Loss* (cls_loss)
   Menggunakan *Binary Cross Entropy* (BCE) untuk memprediksi kelas:

$$L_{\text{cls}} = \text{BCE}(p_{\text{cls}}, \hat{p}_{\text{cls}}) \tag{8}$$

   Keterangan:

p_{cls} = label kelas sebenarnya (*one-hot encoding*)
$\hat{p}_{\text{cls}}$ = probabilitas kelas yang diprediksi model

4. *Distribution Focal Loss* (*DFL*)
   Digunakan untuk memprediksi distribusi koordinat *bounding box* secara lebih halus:

$$L_{\text{dfl}} = \sum w_i \cdot \text{CE}(q_i, \hat{q}_i) \quad (9)$$

Keterangan:

q_i = distribusi target koordinat (*ground truth* yang dikuantisasi)
$\hat{q}_i$ = distribusi prediksi koordinat
w_i = bobot per-bin distribus
CE = *Cross Entropy*

1.6.5.2 YOLOv11-Pose (Integrasi YOLOv11 dan Estimasi Pose)

YOLO-Pose merupakan perluasan dari arsitektur YOLO yang tidak hanya melakukan deteksi objek, tetapi juga memprediksi *keypoints* dalam satu arsitektur terpadu. Pada generasi terbaru, YOLOv11 dikembangkan dengan dukungan eksplisit terhadap beragam tugas *computer vision* termasuk *object detection, instance segmentation*, serta *pose estimation* (Khanam & Hussain, 2024). Secara khusus, varian YOLOv11-Pose dirancang untuk mendeteksi *keypoints* pada citra atau video untuk melacak gerakan dan konfigurasi pose. *Object detection* dan *pose/posture estimation* dijalankan secara paralel, sehingga satu jaringan mempelajari lokasi tangan dan struktur jari secara bersamaan. Hal ini lebih efisien dibanding *pipeline* dua-tahap yang memerlukan dua model terpisah.

YOLOv11-Pose memprediksi 21 *keypoint* tangan dalam format (x_i, y_i, v_i) untuk setiap *keypoint*. Pendekatan terintegrasi ini sejalan dengan definisi *pose estimation* sebagai proses pendeteksian *keypoints* untuk melacak pose atau gerakan. Integrasi deteksi objek dan estimasi pose dalam satu arsitektur memberikan efisiensi komputasi yang lebih tinggi, sementara optimasi YOLOv11 untuk kinerja *real-time* membuatnya stabil digunakan dalam berbagai skenario aplikasi yang menuntut prediksi cepat dan presisi tinggi. Estimasi koordinat dilakukan melalui regresi langsung terhadap nilai ter-normalisasi. Ketelitian keypoint dievaluasi dengan Mean Squared Error (MSE):

$$L_{\text{kpt}} = \frac{1}{N}\sum_{i=1}^{N} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \quad (10)$$

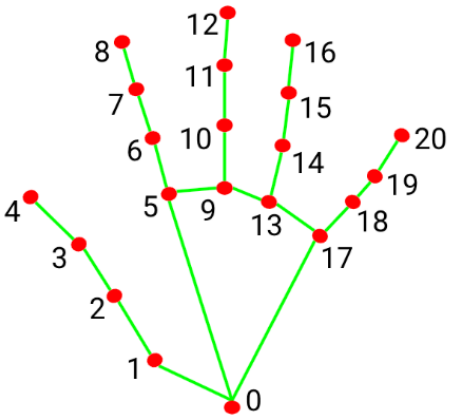
Keterangan:

N = jumlah keypoint
(x_i, y_i) = koordinat ground truth keypoint ke-i

$(\hat{x}_i, \hat{y}_i)$ = prediksi koordinat keypoint ke-i

yang sensitif terhadap kesalahan kecil, terutama pada ujung jari yang memiliki geometri halus.

1.6.6 MediaPipe Hands



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Gambar 5 Struktur 21 *Landmark* pada MediaPipe Hands (MediaPipe, 2025)

MediaPipe Hands adalah kerangka kerja estimasi pose tangan yang dikembangkan oleh Google untuk mendeteksi dan melacak 21 titik kunci (*hand landmarks*) secara *real-time*. Sistem ini dirancang menggunakan pendekatan dua tahap, yaitu *palm detection* untuk menemukan lokasi telapak tangan dan *hand landmark regression* untuk memprediksi struktur tangan secara detail (Lugaresi et al., 2019). Pemisahan dua tahap ini membuat proses lebih stabil karena bentuk telapak tangan relatif konsisten dan menjadi titik awal yang kuat untuk mengestimasi konfigurasi jari.

Pada tahap regresi, MediaPipe memprediksi koordinat landmark dalam format ter-normalisasi (x, y, z) beserta nilai visibilitas yang menunjukkan tingkat kepercayaan model. Representasi berbasis 3D ini memberikan keuntungan penting dalam memahami orientasi tangan pada sudut pandang yang bervariasi, kondisi pencahayaan kompleks, maupun oklusi sebagian. (Brown, 2024) menegaskan bahwa penggunaan prediksi 3D dan representasi 2.5D pada model landmark modern membantu meningkatkan ketahanan sistem terhadap perubahan orientasi dan deformasi tangan, karena struktur ruang memungkinkan model mempertahankan konsistensi bentuk meskipun bagian tangan mengalami rotasi atau tertutup sebagian.

Secara mendasar, estimasi landmark dilakukan melalui regresi koordinat, yang lazimnya menggunakan pendekatan Mean Squared Error (MSE) untuk meminimalkan deviasi posisi landmark terhadap anotasi acuan. Meskipun MediaPipe tidak merilis loss function secara detail dalam publikasi resminya, mekanisme regresi yang digunakan mengikuti prinsip umum jaringan prediksi pose modern.

1.6.7 *Epoch, Batch Size*, dan *Learning Rate*

Dalam pelatihan model deep learning, pemilihan hiperparameter seperti *epoch, batch size,* dan *learning rate* berperan penting dalam menentukan dinamika pembelajaran, stabilitas gradien, dan kemampuan generalisasi jaringan saraf. Ketiga komponen ini mengatur bagaimana model memproses data dan memperbarui bobot selama proses training, sehingga sangat mempengaruhi hasil akhir model.

1.6.7.1 *Epoch*

Epoch merupakan salah satu hiperparameter utama dalam pembelajaran *deep learning* yang menentukan berapa kali seluruh dataset pelatihan diproses secara penuh oleh model. *Epoch* mengacu pada satu siklus penuh ketika seluruh dataset pelatihan diproses oleh model melalui *forward pass* dan backward pass. Semakin banyak epoch, semakin sering model memperbarui bobot berdasarkan keseluruhan data yang tersedia. Namun, jumlah *epoch* yang terlalu besar dapat menyebabkan *overfitting*, sementara *epoch* yang terlalu sedikit menyebabkan *underfitting*. Penelitian menunjukkan bahwa proses pelatihan pada model *computer vision* umumnya membutuhkan pengulangan yang cukup banyak agar pola visual dapat dipelajari secara stabil, di mana performa model cenderung meningkat seiring penambahan *epoch* hingga mencapai titik konvergensi tertentu (Jonathan & Hermanto, 2024). Pemilihan jumlah *epoch* yang tepat karenanya harus mempertimbangkan kompleksitas model, ukuran dataset, dan tingkat variasi fitur visual.

1.6.7.2 *Batch Size*

Batch size adalah jumlah sampel yang diproses sekaligus dalam satu iterasi pelatihan sebelum bobot jaringan diperbarui. Pemilihan *batch size* sangat memengaruhi dinamika pembelajaran, stabilitas gradien, dan kemampuan generalisasi model. *Batch size* besar menghasilkan estimasi gradien yang lebih stabil dan dapat mempercepat konvergensi, tetapi sering disertai kecenderungan penurunan akurasi dan peningkatan jumlah prediksi keliru, meskipun besarnya perubahan relatif kecil (Muralidhar et al., 2024). Jika *batch size* berada pada nilai yang terlalu kecil, proses pembaruan bobot cenderung menjadi kurang stabil karena setiap iterasi hanya merefleksikan sebagian kecil sampel. Kondisi ini membuat proses pelatihan berjalan lebih lambat dan menghasilkan gradien yang berfluktuasi, sebab informasi yang dipelajari pada tiap langkah belum mewakili karakteristik data secara menyeluruh (Akmal Hariz et al., 2022). Oleh karena itu, pemilihan *batch size* harus mempertimbangkan keseimbangan antara efisiensi komputasi dan kualitas generalisasi.

1.6.7.3 *Learning Rate*

Learning rate adalah parameter yang mengatur besarnya langkah pembaruan bobot pada setiap iterasi. *Learning rate* yang terlalu besar membuat pembaruan bobot tidak stabil, sedangkan nilai yang terlalu kecil memperlambat proses konvergensi. Pendekatan modern biasanya menggunakan *learning rate scheduling*, seperti *cosine*

annealing dan teknik *warmup*, untuk menyesuaikan nilai *learning rate* secara dinamis selama pelatihan. Metode ini terbukti meningkatkan stabilitas pembelajaran pada model *neural network* berskala besar (Dosovitskiy et al., 2021). Nilai learning rate umumnya dipilih berdasarkan ukuran *batch*, kompleksitas model, dan karakteristik dataset agar proses pembelajaran berlangsung efektif.

1.6.8 Enrichment Data

Enrichment data merupakan proses memperluas dan memperkaya variasi data pelatihan untuk meningkatkan kemampuan model dalam mengenali pola secara lebih umum dan tidak terbatas pada kondisi tertentu. Pada bidang *computer vision*, *enrichment* data berfungsi memperkenalkan berbagai variasi visual seperti perubahan pose, rotasi, pencahayaan, maupun latar belakang, sehingga jaringan saraf mampu mempelajari representasi fitur yang lebih stabil dan tidak mudah terjebak pada pola yang spesifik terhadap dataset awal. Perluasan data mampu meningkatkan *generalization performance* sekaligus mengurangi risiko *overfitting*, terutama ketika jumlah data asli terbatas (Wang et al., 2025). *Enrichment data* bekerja dengan mempertahankan makna semantik sebuah objek sambil mengubah karakteristik visualnya. Pendekatan ini memungkinkan model *deep learning* memperoleh representasi yang lebih komprehensif mengenai bentuk dan konfigurasi suatu objek. Keberagaman sampel *training* dapat membuat model lebih robust terhadap kondisi visual yang berubah-ubah (Wang et al., 2025)**.** Dengan demikian, *enrichment data* merupakan strategi fundamental dalam pelatihan model *vision* modern karena membantu membangun model yang lebih robust, adaptif, dan mampu beroperasi secara akurat pada berbagai kondisi nyata.

1.7 Penelitian Terdahulu

Dalam penyusunan penelitian ini, penulis menggunakan acuan dari berbagai hasil penelitian terdahulu sebagai referensi. Berikut ini penelitian terdahulu yang memiliki keterkaitan dengan topik penelitian ini, antara lain:

Dalam penelitian berjudul “Real-Time Indonesian Hand Sign Language Gesture Detection and Text Translation Using YOLOv11” oleh (Meylie & Waworuntu, 2025) mengkaji pemanfaatan YOLOv11 untuk mendeteksi gerakan tangan BISINDO secara real-time sebagai upaya menjembatani hambatan komunikasi komunitas Tuli di Indonesia. Peneliti menggabungkan beberapa dataset publik BISINDO, melakukan kurasi dan re-annotasi melalui Roboflow, lalu melatih varian YOLOv11n dengan resolusi 640×640 piksel dengan Hasil evaluasi model terbaik mencapai presisi 97,5%, recall 97,1%, dan Presisi Rata-rata (mAP) 92,5% pada ambang batas IoU dari 0,5 hingga 0,95, sehingga sistem mampu mengenali beberapa kelas gestur BISINDO secara andal dalam video langsung.

Penelitian dengan judul “Utilizing the YOLOv8 Model for Accurate Hand Recognition with Complex Background” oleh (Kristianto et al., 2025) membahas permasalahan deteksi tangan pada citra dan video dengan latar belakang kompleks menggunakan arsitektur YOLOv8 berbasis Convolutional Neural Network (CNN). Penelitian ini bertujuan untuk meningkatkan akurasi dan keandalan deteksi tangan

pada berbagai kondisi pencahayaan, occlusion, motion blur, serta variasi ukuran dan posisi tangan melalui pemanfaatan varian YOLOv8n dan YOLOv8s yang dilatih selama 50 dan 100 epoch pada Oxford Hand Dataset dan EgoHand Dataset. Hasil penelitian menunjukkan bahwa model YOLOv8n dengan 100 epoch memberikan performa terbaik, dengan nilai mAP sebesar 86,7% pada Oxford Hand Dataset dan 98,9% pada EgoHand Dataset, serta precision dan recall yang tinggi pada keempat kelas tangan (*myleft, myright, yourleft, yourright*). Peneliti menyimpulkan bahwa YOLOv8n efektif digunakan untuk deteksi tangan pada lingkungan dengan latar belakang rumit secara real-time, dan mampu melampaui performa beberapa metode terdahulu pada kedua dataset benchmark tersebut.

Penelitian (Rimbawa et al., 2025) berjudul *"*Artificial intelligence-based hand gesture recognition for sign language interpretation*"* memanfaatkan MediaPipe Hands sebagai fitur utama melalui ekstraksi 21 *landmark* tangan. Landmark ini kemudian diubah menjadi vektor fitur dan diklasifikasikan menggunakan jaringan saraf tiruan untuk mengenali beberapa gestur secara real-time. Penelitian ini menyoroti peningkatan keandalan sistem pada latar belakang yang beragam tanpa memerlukan anotasi bounding box manual, sehingga mengurangi beban pelabelan secara signifikan. Hasil eksperimen menghasilkan akurasi klasifikasi rata-rata sebesar 88,03%, bersama dengan presisi, recall, dan skor F1 yang melebihi 90%. Sistem yang dibangun berpotensi meningkatkan layanan publik dan alat bantu komunikasi bagi penyandang disabilitas pendengaran, sehingga memungkinkan interaksi yang lebih mudah diakses dalam kehidupan sehari-hari.

Penelitian *"Smart Sign Language Recognition System using MediaPipe"* oleh (Rathod et al., 2023) mengembangkan sistem pengenalan bahasa isyarat real-time menggunakan *MediaPipe Hand Landmark Model* dan jaringan saraf hibrida. MediaPipe digunakan untuk mendeteksi *keypoints* tangan, sementara model deep learning kustom dan model Google pra-latih menangani klasifikasi gestur. Sistem mencakup tahap akuisisi video, pra-pemrosesan, ekstraksi ROI, normalisasi fitur, dan pemetaan gestur ke teks. Hasil uji menunjukkan sistem mencapai akurasi 95%, cukup andal untuk membantu pengguna tunarungu dalam komunikasi sehari-hari. Penelitian ini menegaskan potensi pendekatan berbasis visi komputer yang ringan dan real-time, serta membuka peluang pengembangan lanjutan pada gestur dinamis, multimodal, dan implementasi pada perangkat mobile.

Penelitian berjudul *"MPH-YOLO method for isolating the controller hand amidst multiple hands in digital interaction environments"* oleh Himamunanto et al., 2025

mengusulkan metode untuk memilih satu tangan pengendali di tengah beberapa tangan dalam satu frame dengan menggabungkan MediaPipe Hands (lokalisasi tangan via 21 landmark) dan YOLOv8 (klasifikasi tangan yang memakai marker di pergelangan). Metode ini dilatih dan diuji pada ribuan video gestur satu–tangan dan multi–tangan, lalu diuji pada 800 video uji dengan berbagai kondisi. Sistem berhasil mengisolasi tangan pengendali dengan akurasi 97,5% dan kecepatan sekitar 25–26 FPS, sehingga dinilai cukup andal untuk aplikasi kontrol gestur real-time, dengan ruang pengembangan ke depan untuk deteksi tangan utama tanpa marker.

Penelitian oleh Mealdi et al., pada tahun 2025 mengembangkan sistem deteksi gerakan fitness dengan menggabungkan YOLOv11 untuk deteksi tubuh dan MediaPipe Pose untuk ekstraksi 33 keypoint, lalu memvalidasi kualitas gerakan—khususnya *squat*—berdasarkan analisis sudut lutut. Dataset berisi 13 jenis latihan yang dikelompokkan menjadi dua kategori utama: gerakan otot kaki dan otot punggung. Pada pengujian mendalam untuk gerakan squat menggunakan 30 video (15 benar, 15 salah), sistem mencapai akurasi 73,3%, dengan precision, recall, specificity, dan F1-score yang identik pada nilai 73,3%. Analisis kesalahan menunjukkan performa sangat baik untuk mendeteksi squat terlalu dangkal (<60°), namun lebih lemah untuk kesalahan posisi lutut ke depan. Penelitian menyimpulkan bahwa kombinasi YOLOv11 dan MediaPipe mampu menjadi alat bantu evaluasi gerakan fitness secara mandiri, meskipun peningkatan diperlukan untuk mendeteksi variasi kesalahan yang lebih halus.

Penelitian *"YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss"* oleh Maji, Nagori, et al., pada tahun 2022 memperkenalkan metode estimasi pose multi-orang berbasis YOLO yang tidak menggunakan heatmap dan langsung memprediksi bounding box beserta 2D pose (17 keypoint) setiap orang dalam satu *forward pass*. Model dilatih end-to-end dengan Object Keypoint Similarity (OKS) loss, sehingga optimasi selaras dengan metrik evaluasi pose dan tidak memerlukan tahap grouping keypoint yang rumit seperti pada metode bottom-up. Pada dataset COCO, YOLO-Pose mencapai AP50 (90,2% pada *val* dan 90,3% pada *test-dev*) melampaui metode bottom-up terkini dengan kebutuhan komputasi yang lebih rendah, tanpa bantuan *test-time augmentation* seperti *flip test* dan *multi-scale testing*.

Penelitian dengan judul "Analyzing Yoga Pose Recognition: A Comparison of MediaPipe and YOLO Keypoint Detection with Ensemble Techniques" oleh Debalaxmi et al., pada tahun 2024 menganalisis pengenalan pose yoga berbasis *keypoint* dengan membandingkan dua metode ekstraksi rangka tubuh, yaitu YOLOv8 (32 keypoint) dan MediaPipe (33 keypoint), yang kemudian dipadukan dengan berbagai algoritma machine learning dan ensemble pada dataset 1.248 citra yang mencakup 20 pose yoga, dengan penyeimbangan kelas menggunakan SMOTE. Keypoint yang dihasilkan kemudian diklasifikasikan dengan berbagai model ML dan ensemble (SVM, KNN, MLP, Random Forest, XGBoost, LightGBM, dll.). Hasil terpenting: kombinasi MediaPipe + LightGBM memberikan performa terbaik dengan akurasi 96,52% dan metrik precision/recall/F1 di atas 0,94, secara konsisten mengungguli YOLOv8. Studi ini menyimpulkan bahwa kombinasi ekstraksi rangka MediaPipe dan ensemble LightGBM sangat efektif untuk klasifikasi multi-pose yoga, termasuk pose yang mirip secara visual, dan merekomendasikan pengembangan lanjutan dengan arsitektur deep learning (misalnya CNN/GNN), penambahan variasi pose, integrasi inferensi real-time, dan implementasi sebagai aplikasi mobile interaktif untuk koreksi pose yoga secara mandiri.

Untuk memberikan gambaran yang lebih jelas, berikut disajikan tabel ringkasan penelitian terdahulu yang relevan dengan topik penelitian ini, yang dapat dilihat pada Tabel 1.

Tabel 1 Ringkasan Penelitian Terdahulu

No	Penulis (Tahun)	Judul	Identifikasi Masalah	Teknik/Metode	Hasil
1.	(Meylie & Waworuntu, 2025)	*Real-Time Indonesian Hand Sign Language Gesture Detection and Text Translation Using YOLOv11*	Kebutuhan deteksi BISINDO real-time untuk membantu komunikasi komunitas Tuli di Indonesia.	YOLOv11n, transfer learning, data BISINDO multi-sumber	Model terbaik mencapai presisi 97,5%, recall 97,1%, dan mAP 92,5% (IoU 0,5–0,95), menunjukkan bahwa sistem dapat mengenali beberapa gestur BISINDO secara andal pada video real-time.
2.	(Kristianto et al., 2025)	*Utilizing the YOLOv8 Model for Accurate Hand Recognition with Complex Background*	Deteksi tangan pada citra dan video dengan latar belakang kompleks	YOLOv8 berbasis CNN	YOLOv8n mencapai mAP 86,7% (Oxford Hand) dan 98,9% (EgoHand), dengan precision–recall tinggi, sehingga efektif untuk deteksi tangan real-time pada latar kompleks dan unggul dibanding metode sebelumnya

No	Penulis (Tahun)	Judul	Identifikasi Masalah	Teknik/Metode	Hasil
3.	(Rimbawa et al., 2025)	*Artificial intelligence-based hand gesture recognition for sign language interpretation*	Ketiadaan sistem pengenalan BISINDO yang akurat dan real-time akibat keterbatasan dataset, serta minimnya riset yang memanfaatkan MediaPipe untuk kondisi dunia nyata	MediaPipe Hand ekstraksi 21 *landmark* tangan	Eksperimen mencapai akurasi rata-rata 88,03% dengan presisi, recall, dan F1 di atas 90%, menunjukkan sistem ini berpotensi mendukung layanan publik dan membantu komunikasi penyandang disabilitas pendengaran.
4.	(Rathod et al., 2023)	*Smart Sign Language Recognition System using MediaPipe*	Kurangnya sistem pengenalan bahasa isyarat yang akurat, real-time, dan fleksibel.	*MediaPipe Hand Landmark Model* dan jaringan saraf hibrida	Sistem mencapai akurasi 95%, cukup andal untuk membantu pengguna tunarungu dalam komunikasi sehari-hari.
5.	(Himamunanto et al., 2025)	*MPH-YOLO method for isolating the controller hand amidst multiple hands in digital interaction environments*	YOLO secara *default* akan mendeteksi semua tangan tanpa bisa memilih mana tangan yang menjadi pengendali yang menyebabkan ambiguitas ketika lebih dari satu tangan muncul dalam satu frame.	MediaPipe Hands dan YOLOv8	Sistem berhasil mengisolasi tangan pengendali dengan akurasi 97,5% dan kecepatan sekitar 25–26 FPS, sehingga dinilai cukup andal untuk aplikasi kontrol gestur real-time

No	Penulis (Tahun)	Judul	Identifikasi Masalah	Teknik/Metode	Hasil
6.	(Mealdi et al., 2025)	Deteksi Gerakan Fitness Menggunakan Pose Estimation dan YOLOv11	Klasifikasi gerakan fitness secara otomatis	YOLOv11, Pose Estimation	Sistem mampu mengidentifikasi dua kelompok gerakan utama dalam aktivitas fitness yaitu latihan otot kaki dan latihan otot punggung dengan akurasi klasifikasi rata-rata 73,3%.
7.	(Maji et al., 2022)	*YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss*	Belum tersedia metode pose multi-orang yang efisien tanpa heatmap dan post-processing, serta belum mampu dilatih end-to-end dan berjalan konstan seperti deteksi satu tahap pada YOLO.	YOLO-Pose dan regresi langsung keypoint menggunakan OKS Loss pada arsitektur YOLOv5.	YOLO-Pose mencapai AP50 (90,2% pada *val* dan 90,3% pada *test-dev*) melampaui metode bottom-up terkini dengan kebutuhan komputasi yang lebih rendah, tanpa bantuan *test-time augmentation* seperti *flip test* dan *multi-scale testing*.
8.	(Debalaxmi et al., 2024)	*Analyzing Yoga Pose Recognition: A Comparison of MediaPipe and YOLO Keypoint Detection with Ensemble Techniques*	Bagaimana membangun sistem pengenalan pose yoga yang akurat, bisa membedakan pose yang mirip, tidak tergantung perangkat mahal, dan tahan terhadap	Ekstraksi keypoint dengan MediaPipe atau YOLOv8, penyeimbangan data menggunakan SMOTE, dan klasifikasi melalui	Kombinasi MediaPipe + LightGBM memberikan performa terbaik dengan akurasi 96,52% dan metrik precision/recall/F1 di atas 0,94, secara konsisten

No	Penulis (Tahun)	Judul	Identifikasi Masalah	Teknik/Metode	Hasil
			ketidakseimbangan data	beragam model machine learning serta ensemble	mengungguli YOLOv8

BAB II
METODE PENELITIAN

2.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimental, karena berfokus pada pengujian kinerja model deteksi citra berbasis CNN dan YOLOv11-Pose. Data yang digunakan berupa kumpulan citra hasil dokumentasi primer yang diolah dan dilabeli secara terstruktur untuk keperluan pelatihan model. Penelitian diawali dengan studi literatur untuk merumuskan landasan teoritis, menentukan arsitektur model yang digunakan, serta merancang alur eksperimen secara sistematis.

Proses eksperimen dilakukan dengan melatih model pada dataset citra tangan beranotasi untuk enam kelas gerakan tangan, kemudian mengukur akurasi, *precision, recall,* dan *mean Average Precision* (mAP50–95) sebagai indikator performa. Pendekatan kuantitatif dipilih karena seluruh keluaran penelitian dinyatakan dalam bentuk nilai numerik dan metrik evaluasi yang dapat diinterpretasikan secara objektif untuk menilai efektivitas model dalam mendeteksi dan mengenali gerakan tangan.

2.2 Waktu dan Tempat Penelitian

Penelitian dilaksanakan secara lokal di lingkungan rumah dan kampus peneliti dengan memanfaatkan laptop pribadi sebagai sarana utama pelatihan dan pengujian model. Seluruh rangkaian kegiatan, mulai dari studi literatur, hingga analisis hasil, dilakukan pada periode Agustus hingga November 2025. Adapun untuk detail waktu penelitian dapat dilihat pada Tabel 2.

Tabel 2 Waktu Penelitian

Kegiatan	Agustus				September				Oktober				November			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi Literatur																
Pre-processing Data																
Processing Data (Pembuatan Model)																
Pengujian Trained Model																
Analisis Hasil dan Penulisan Kesimpulan																

2.3 Instrumen Penelitian

Dalam penelitian ini terdapat beberapa instrumen yang membantu proses penelitian. Instrumen tersebut terbagi menjadi dua yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*).

2.3.1 Perangkat keras (*hardware*)

Perangkat keras atau *hardware* beserta dengan spesifikasi yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.

Tabel 3 Perangkat Keras (*Hardware*)

Jenis	Spesifikasi
Sistem Operasi	Windows 10 (64-bit)
Prosesor	Intel ® Core™ i3-1005G1 CPU @ 1.20 GHz, 4 cores
Memori (RAM)	8 GB DDR4
GPU	NVIDIA GeForce RTX 3050

2.3.2 Perangkat Lunak (*Software*)

Terdapat beberapa perangkat lunak atau *software* yang digunakan dalam penelitian ini yang bertujuan untuk mempermudah proses kerja. Perangkat lunak yang digunakan dapat dilihat pada Tabel 4.

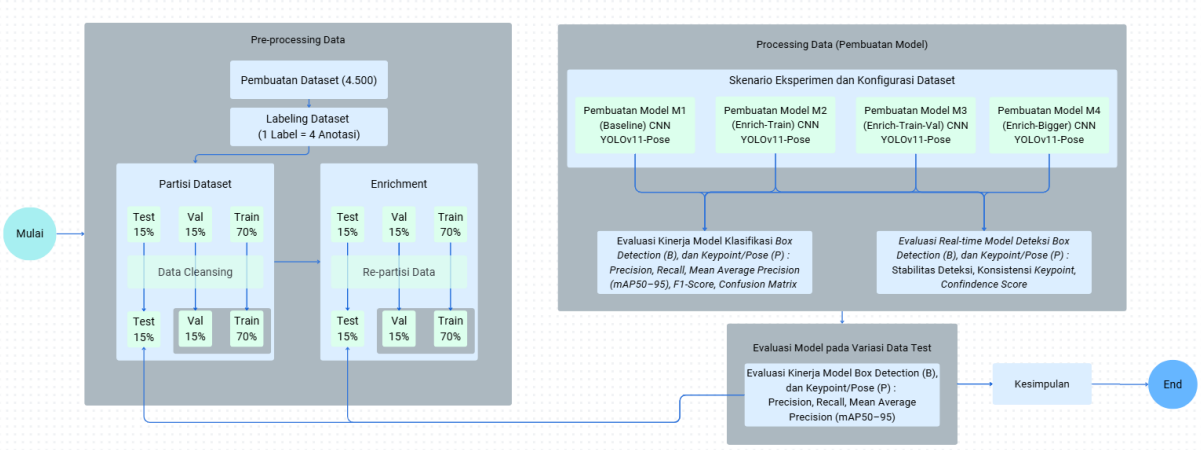
Tabel 4 Perangkat Lunak *(Software)*

Perangkat Lunak	Fungsi Utama
Python	Bahasa pemrograman utama untuk seluruh *pipeline*, mulai dari *pre-processing*, pelabelan, training, hingga evaluasi model.
NumPy	*Library* untuk komputasi numerik dan manipulasi *array* multidimensi.
Pandas	*Library* untuk pengolahan dataset, termasuk manipulasi CSV/Excel dan persiapan data *training*.
OpenCV	*Library* untuk pemrosesan citra dan video, digunakan dalam *pre-processing,* ekstraksi *frame,* dan visualisasi *keypoint.*
MediaPipe Hands	*Framework* untuk deteksi dan ekstraksi 21 *keypoint* tangan secara otomatis dari citra/video
Roboflow	*Platform* untuk ubah video jadi dataset, anotasi citra, export dataset ke format YOLOv11-Pose, serta pengaturan *train/validation/test split*
PyTorch / Ultralytics YOLOv11-Pose	*Framework deep learning* untuk membangun, melatih, dan menguji model deteksi gerakan tangan berbasis CNN dan YOLOv11-Pose.
COCO / pycocotools	Format anotasi citra standar, pembacaan dan validasi *bounding box/keypoint,* interoperabilitas dengan YOLOv11-Pose.

Perangkat Lunak	Fungsi Utama
JSON	Format penyimpanan anotasi, *metadata, dan keypoint* dari Roboflow/MediaPipe
Protocol Buffers (protobuf)	Digunakan oleh MediaPipe untuk serialisasi data *keypoint,* memastikan komunikasi dan penyimpanan efisien.
Matplotlib & Seaborn	*Library* visualisasi untuk menampilkan grafik *loss*, akurasi, mAP, dan *keypoint overlay*.
Scikit-learn	*Library* untuk pembagian dataset, normalisasi, dan perhitungan metrik evaluasi seperti *Precision, Recall, F1-score*.
Jupyter Notebook	Lingkungan interaktif untuk eksplorasi data, dokumentasi eksperimen, dan visualisasi hasil pelatihan model.
Visual Studio Code	Editor kode yang digunakan untuk pengembangan program, integrasi *library*, dan implementasi model penelitian.
YAML	Digunakan untuk konfigurasi dataset (data.yaml) agar YOLOv11-Pose dapat membaca dataset dengan benar.

2.4 Alur Penelitian

Untuk memperjelas rangkaian proses yang dilakukan dalam penelitian ini, berikut ditampilkan alur penelitian yang menggambarkan tahapan mulai dari pengumpulan data hingga evaluasi performa model, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6 Diagram Alur Penelitian

2.4.1 Pre-processing Data

Tahapan *pre-processing* dilakukan untuk memastikan bahwa dataset memenuhi standar kualitas sebelum digunakan dalam pelatihan. Pada penelitian ini, *pre-processing* mencakup pengumpulan dan pelabelan data, pembersihan data (*cleansing)*, serta partisi data. Tujuan akhirnya adalah menghasilkan dataset gerakan tangan yang bersih, terstruktur, dan kompatibel dengan format YOLOv11-Pose.

2.4.1.1 Pengumpulan Data (Pembuatan Dataset)

Data dikumpulkan dengan merekam video pendek gerakan tangan manusia untuk enam kelas pose: *back-slap, default, little-finger, peace, pistol* dan *thumb_side.* Perekaman dilakukan di lingkungan terkontrol dengan karakteristik sebagai berikut:

1. Latar belakang sederhana cenderung polos.
2. Pencahayaan bervariasi dari terang hingga agak redup, namun tetap cukup untuk menampilkan kontur tangan dengan jelas.
3. Posisi kamera dijaga sejajar dengan tangan untuk meminimalkan distorsi sudut.
4. Tangan kanan dan kiri digunakan secara bergantian sehingga model dapat mempelajari variasi orientasi dan *flip* horizontal secara alami.

Video yang diperoleh kemudian diekstraksi menjadi frame citra. Seluruh citra diunggah ke Roboflow dan dikelompokkan menurut kelas gerakan*,*

2.4.1.2 Labeling Data Dan Anotasi

Dataset hasil ekstraksi video selanjutnya disimpan dan dikelola menggunakan platform Roboflow untuk mempermudah proses pelabelan dan manajemen anotasi. Pada tahap ini, setiap citra diberi anotasi berupa bounding box pada area tangan serta label kelas sesuai dengan gerakan tangan yang diperagakan. Proses pelabelan dilakukan secara manual untuk memastikan ketepatan lokasi objek dan konsistensi antar kelas pose.

Dataset hasil ekstraksi video selanjutnya disimpan dan dikelola menggunakan platform Roboflow untuk mempermudah proses pelabelan dan manajemen anotasi. Pada tahap ini, setiap citra diberi anotasi berupa bounding box pada area tangan serta label kelas sesuai dengan gerakan tangan yang diperagakan. Proses pelabelan dilakukan secara manual untuk memastikan ketepatan lokasi objek dan konsistensi antar kelas pose.

Format anotasi yang digunakan adalah COCO (*_annotations.coco.json*), yang memuat informasi identitas gambar, identitas anotasi, koordinat *bounding box*, dan label kelas. Meskipun pelabelan awal hanya mencakup *bounding box* dan kelas pose, dataset ini dirancang untuk dilatih menggunakan model YOLOv11-Pose yang membutuhkan informasi *keypoint*. Oleh karena itu, anotasi COCO tersebut selanjutnya dikonversi ke format YOLO-Pose dengan menambahkan informasi 21 titik *keypoint* tangan sesuai dengan struktur input model.

2.4.1.3 Partisi dan Cleansing Data

Dataset awal kemudian dibagi menjadi tiga subset, yaitu *train, validation*, dan *test*, dengan target proporsi 70% untuk data *train*, 15% untuk *validation*, dan 15% untuk data *test*. Proses pembagian dilakukan di Roboflow dengan memperhatikan distribusi kelas agar tetap seimbang pada setiap subset. Subset *train* digunakan untuk melatih model, subset *validation* berperan dalam memantau performa model serta melakukan penyesuaian *hyperparameter*, sedangkan subset *test* disimpan sebagai data murni untuk evaluasi akhir, sehingga pengukuran kinerja model dilakukan pada data yang benar-benar baru dan tidak pernah digunakan selama pelatihan.

Setelah pembagian, dilakukan tahap *cleansing* untuk menyingkirkan citra yang tidak memenuhi standar kualitas. Tahap ini berlangsung dalam dua Langkah, yaitu:

1. Langkah pertama dilakukan secara manual, di mana citra diperiksa melalui antarmuka Roboflow untuk menghapus *frame* yang kabur, terlalu gelap, menampilkan gerakan ambigu, atau memiliki label yang jelas tidak sesuai.
2. Langkah kedua berlangsung secara otomatis, dengan *pycocotools* digunakan untuk memverifikasi kelengkapan anotasi, OpenCV memastikan setiap file citra dapat dibuka tanpa *error*, dan MediaPipe Hands mendeteksi 21 *keypoint* tangan. *Frame* yang tidak terbaca, memiliki anotasi tidak lengkap, atau *keypoint* tidak stabil dikeluarkan dari dataset.

2.4.1.4 Data Enrichment dan Keypoint Labeling dengan MediaPipe

Proses data *enrichment* dilakukan untuk memperluas keragaman dataset yang pada tahap awal masih bersifat relatif homogen dan berisiko mendorong model menghafal pola (*memorization*). Perekaman citra dilakukan secara otomatis menggunakan kamera dengan pemilihan kelas pose melalui input numerik dan menghasilkan sekitar empat frame per detik. MediaPipe Hands dimanfaatkan sebagai *auto-labeler* untuk mengekstraksi 21 titik *keypoint* tangan beserta nilai *visibility*, sekaligus menghasilkan *bounding* box ter-normalisasi dalam rentang [0,1]. Anotasi yang dihasilkan diperlakukan sebagai *pseudo-label* dan digunakan dalam pendekatan pelabelan semi-otomatis yang sesuai dengan kebutuhan input YOLOv11-Pose.

Variasi data yang diperoleh mencakup perbedaan sudut kamera, jarak objek, kondisi pencahayaan, latar belakang, serta dinamika pergerakan jari. Karena YOLOv11-Pose mendukung deteksi *multi-hand*, satu citra dapat memuat lebih dari satu tangan dan dihitung sebagai beberapa *instance*, sehingga jumlah *instance* per kelas dapat melebihi jumlah file citra. Setelah proses perekaman, data tetap melalui pengecekan kualitas dasar untuk memastikan file citra dapat dibaca dengan baik dan anotasi tidak korup. Seluruh variasi visual yang valid dipertahankan agar model mampu mempelajari representasi pose tangan yang lebih robust dan mendekati kondisi penggunaan nyata. Dataset hasil *enrichment* ini selanjutnya digunakan pada proses fine-tuning YOLOv11-Pose untuk meningkatkan stabilitas dan ketahanan model pada tahap pengujian.

2.4.1.5 Re-partisi Data

Setelah proses enrichment dilakukan, dataset dalam penelitian ini disusun dalam dua versi utama, yaitu dataset awal dan dataset hasil enrichment. Kedua dataset tersebut dipartisi menggunakan skema yang sama, yakni 70% data *train*, 15% data *validation*, dan 15% data *test*. Penggunaan skema partisi yang konsisten bertujuan untuk menjaga kesetaraan antar skenario eksperimen, sehingga perbandingan kinerja model tidak dipengaruhi oleh perbedaan proporsi data, melainkan oleh perbedaan karakteristik distribusi dataset yang digunakan.

Pada dataset hasil enrichment, subset train dan validation digunakan dalam proses pelatihan dan pemantauan kinerja model, sedangkan subset test dipisahkan sepenuhnya dan tidak dilibatkan dalam proses pelatihan maupun validasi. Selain dataset test awal, penelitian ini juga menyiapkan dataset test hasil enrichment yang digunakan secara terpisah sebagai skenario evaluasi tambahan. Proses pembagian data dilakukan secara acak (shuffled) dengan tetap menjaga keseimbangan jumlah sampel antar kelas pada setiap subset. Dengan strategi ini, evaluasi model dapat dilakukan secara terkontrol pada data dengan tingkat homogenitas dan kompleksitas visual yang berbeda, tanpa menimbulkan bias distribusi kelas.

2.4.2 Processing Data (Pembuatan Model)

Bagian ini menjelaskan pembentukan model deteksi gerakan senam otak tangan menggunakan YOLOv11-Pose, mulai dari model awal, skema validasi, *enrichment* data, hingga pembentukan model akhir dan pemilihan *checkpoint* terbaik. Bagian ini juga menekankan perbedaan antara evaluasi kuantitatif berbasis dataset statis dan performa kualitatif pada pengujian real-time.

2.4.2.1 Skenario Eksperimen dan Konfigurasi Dataset

Pada penelitian ini, seluruh model dikembangkan menggunakan arsitektur YOLOv11-Pose dengan implementasi *framework PyTorch* melalui modul pelatihan Ultralytics. Model diinisialisasi menggunakan bobot *pretrained* yolo11n-pose.pt untuk mempercepat konvergensi dan meningkatkan stabilitas pembelajaran pada tugas deteksi gerakan tangan dan estimasi pose 21 *keypoint*. Pelatihan utama dijalankan pada lingkungan GPU dengan konfigurasi ukuran citra 640×640 piksel, *batch size* 16, dan 100 *epoch*. Konfigurasi berbasis CPU hanya digunakan untuk pengujian teknis awal dan tidak digunakan sebagai hasil evaluasi utama.

Proses pelatihan menggunakan optimizer AdamW dengan *learning rate scheduler* bawaan Ultralytics. Fungsi kerugian yang dioptimalkan meliputi *box loss*, *classification loss*, dan *keypoint regression loss*. Selama pelatihan, Ultralytics secara otomatis menerapkan augmentasi data online standar, seperti transformasi geometrik dan fotometrik, tanpa pengaturan hyperparameter augmentasi secara eksplisit. Untuk meningkatkan efisiensi komputasi, pelatihan pada GPU dijalankan dengan *Automatic Mixed Precision* (AMP).

Evaluasi kinerja model dilakukan melalui beberapa skenario eksperimen terkontrol yang dirancang untuk menganalisis pengaruh variasi komposisi dataset terhadap performa model. Seluruh skenario menggunakan arsitektur dan konfigurasi

pelatihan yang identik, sehingga perbedaan hasil yang diperoleh dapat dikaitkan langsung dengan perbedaan komposisi dataset. Rancangan eksperimen mencakup penggunaan dataset awal, dataset hasil enrichment, serta kombinasi keduanya pada data pelatihan dan validasi. Rincian skenario eksperimen dan konfigurasi dataset disajikan pada Tabel 5.

Tabel 5 Rancangan Skenario Eksperimen dan Konfigurasi Dataset (Epoch 100)

Kode Eksperiment	Data Latih (Train)	Data Validasi (Val)	Jumlah Data (Train + Val)	Augmentasi	Tujuan Eksperimen
M1	Data awal	Data awal	2750 + 575	Default YOLO	Baseline model pada dataset homogen.
M2	Data awal + data *enrich (baseline)*	Data awal	2750 + 575	Default YOLO	Uji generalisasi model terhadap data awal
M3	Data awal enrich + data *enrich (baseline)*	Data awal enrich + data *enrich (baseline)*	2750 + 575	Default YOLO	Uji stabilitas model pada distribusi data campuran
M4	Data *enrichment*	Data *enrichment*	5.500 + 1.150	Default YOLO	Eksplorasi awal pada dataset dengan kompleksitas tinggi

2.4.2.2 Mengecek Fitting Model

Tahap pengecekan *fitting* model dilakukan untuk memastikan bahwa proses pembelajaran YOLOv11-Pose berlangsung secara stabil dan konvergen sebelum model digunakan pada tahap pengujian lanjutan. Pengecekan ini bertujuan untuk mengidentifikasi potensi *underfitting* maupun *overfitting*, serta memastikan bahwa konfigurasi pelatihan dan komposisi dataset menghasilkan perilaku pembelajaran yang wajar. Evaluasi *fitting* dilakukan melalui dua pendekatan. Pertama, melalui evaluasi klasifikasi berbasis metrik. Pada evaluasi ini, digunakan empat metrik utama sebagai parameter evaluasi, yaitu *precision, recall, F1-score*, dan *mean Average Precision* (mAP50–95).

1. *Precision* mengukur tingkat ketepatan model dalam mengidentifikasi objek yang benar dari seluruh prediksi positif yang dihasilkan. Metrik ini menyoroti kemampuan model dalam meminimalkan kesalahan deteksi berupa *false positive*, yaitu kondisi ketika sistem mendeteksi objek yang sebenarnya tidak ada. Nilai precision dihitung menggunakan rumus:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \tag{11}$$

Keterangan:

TP = *True Positive* (prediksi benar)
FP = *False Positive* (prediksi ada objek padahal tidak)

2. *Recall* mengukur sejauh mana model mampu mengenali seluruh objek yang benar dari keseluruhan data uji. Nilai recall berfokus pada pengurangan jumlah *false negative*, yaitu objek yang seharusnya terdeteksi namun tidak teridentifikasi oleh model. Nilai recall dihitung dengan persamaan:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \tag{12}$$

Keterangan:

FN = *False Negative* (objek tidak terdeteksi)

3. *F1-Score* merupakan rata-rata harmonik antara precision dan recall, yang memberikan keseimbangan antara kedua metrik tersebut. F1-Score berguna ketika data antar kelas tidak seimbang (*imbalanced class*) atau ketika diperlukan kompromi antara ketepatan dan kelengkapan deteksi. Nilai F1-Score dihitung dengan rumus:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

4. *Mean Average Precision* (mAP50–95) sebagai indikator utama untuk menilai performa keseluruhan model deteksi. Nilai mAP menghitung rata-rata area di bawah kurva Precision–Recall (PR Curve) pada berbagai ambang *Intersection over Union (IoU)* mulai dari 0,50 hingga 0,95. Semakin tinggi nilai mAP, semakin baik kemampuan model dalam mendeteksi dan mengklasifikasikan objek secara akurat pada berbagai tingkat ketelitian lokasi. Nilai mAP diperoleh melalui persamaan:

$$\text{mAP}_{50-95} = \frac{1}{10} \sum_{t=0.50}^{0.95} \text{mAP}_t \tag{14}$$

Keterangan:

t = threshold IoU (0.50, 0.55, …, 0.95)

mAP_t= nilai AP pada threshold t

Kedua, evaluasi dilakukan pada aspek deteksi melalui penggunaan *confidence score* yang dihasilkan oleh model. *Confidence score* merepresentasikan tingkat keyakinan model terhadap keberadaan serta klasifikasi objek pada setiap hasil prediksi. Pada tahap ini, nilai *confidence score* digunakan sebagai ambang (*threshold*) untuk memfilter hasil deteksi, sehingga hanya prediksi dengan tingkat keyakinan tertentu yang dipertahankan. Pengaturan ambang ini memungkinkan analisis sensitivitas model dalam mendeteksi objek tangan serta kestabilan prediksi *bounding box* dan *keypoint* pada berbagai tingkat kepercayaan.

Evaluasi berbasis *confidence score* digunakan sebagai indikator numerik tambahan untuk memastikan bahwa model tidak menghasilkan prediksi berkeyakinan rendah secara berlebihan, yang dapat mengindikasikan ketidakstabilan pembelajaran atau ketidaksesuaian distribusi data. Selain sebagai indikator numerik tambahan, penggunaan *confidence score* juga memberikan gambaran awal mengenai kestabilan prediksi model pada kondisi penggunaan yang menyerupai skenario nyata, di mana sistem diharapkan mampu menghasilkan deteksi dengan tingkat keyakinan yang konsisten.

2.4.3 Ukuran Kinerja Model pada Variasi Data *Test*

Evaluasi model dilakukan untuk mengukur kemampuan generalisasi dan stabilitas kinerja YOLOv11-Pose terhadap data yang tidak pernah dilibatkan dalam proses pelatihan maupun validasi. Pengujian dilakukan menggunakan dua variasi dataset uji, yaitu dataset test awal dan dataset test hasil *enrichment*.

Dataset *test* awal memiliki karakteristik visual yang relatif homogen dan serupa dengan data pelatihan awal, sedangkan dataset test hasil *enrichment* memiliki tingkat variasi yang lebih tinggi, mencakup perbedaan latar belakang, pencahayaan, sudut pandang, serta kompleksitas pose tangan. Penggunaan dua variasi data uji ini bertujuan untuk menganalisis perilaku model ketika dihadapkan pada perubahan distribusi data (*distribution shift*) yang umum terjadi pada kondisi penggunaan nyata. Setiap model dievaluasi menggunakan konfigurasi evaluasi yang sama pada kedua dataset uji, sehingga perbedaan performa yang dihasilkan dapat dikaitkan langsung dengan karakteristik data uji yang digunakan (tambahkan dataset uji)