

**MONITORING GERAKAN DUDUK DENGAN  
MENGUNAKAN METODE ALGORITMA  
YOLO PADA KARYAWAN**

**SKRIPSI**



**DINO ODE KAMBISA  
105841104219**

**PROGRAM STUDI INFORMATIKA FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MAKASSAR**

**2025**



## PENGESAHAN

Skripsi atas nama **Dimo Ode Kambisa** dengan nomor induk Mahasiswa **105841104219**, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0005/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 20 Desember 2025.

Panitia Ujian :

1. Pengawas Umum

Makassar,

2 Rajab, 1447 H

22 Desember 2025 M

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST.,MT.,IPU

b. Dekan Fakultas Teknik Universitas Muhammadiyah Makassar

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng

2. Penguji

a. Ketua : Prof. Dr. Ir. Zahir Zainuddin, M.Sc.

b. Sekretaris : Prof. Dr. Ir. Hafsa Nirwana, M.T.

3. Anggota : 1. Rizki Yusliana Bakti, S.T., M.T.

2. Desi Anggreani, S.Kom., M.T.

3. Muhyiddin A M Hayat, S.Kom, M.T.

Mengetahui :

Pembimbing I

Pembimbing II

Fahrim Irhamna Rachman S.Kom., M.T.Titin Wahyuni, S.Pd, M.T



## HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **Monitoring Gerakan Duduk Dengan Menggunakan Metode Algoritma Yolo Pada Karyawan**

Nama : Dino Ode Kambisa

Stambuk : 105 84 11042 19

Makassar, 2 Rajab 2025

Telah Diperiksa dan Disetujui  
Oleh Dosen Pembimbing:

Pembimbing I

Fahrim Irhamna Rachman S.KOM., M.T

Pembimbing II

Titin Wahyuni, S.Pd, M.T

Mengetahui,

Ketua Prodi Informatika



Rizki Yusliana Bakti, S.T., M.T.

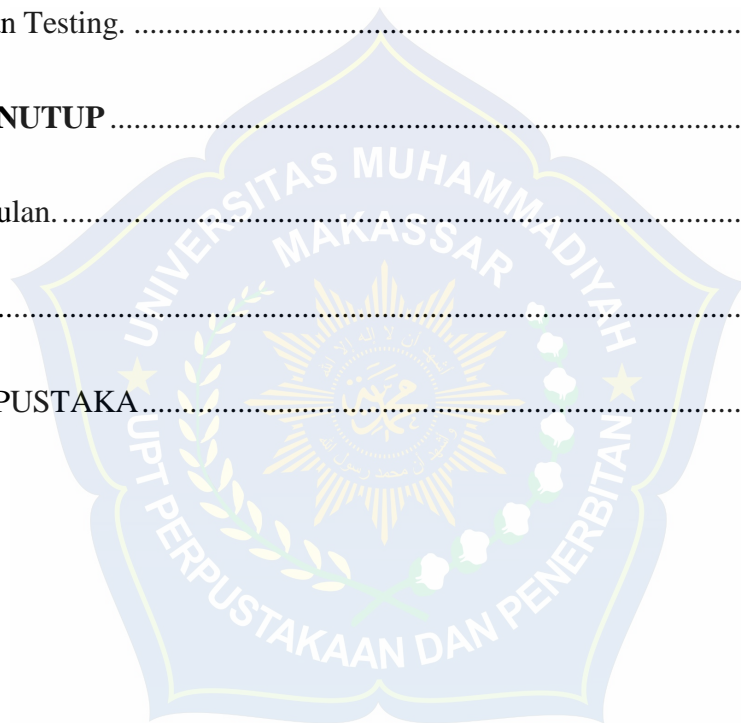
NBM : 1307 284



## DAFTAR ISI

<b>HALAMAN JUDUL.....</b>	<b>i</b>
<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>vi</b>
<b>DAFTAR TABEL.....</b>	<b>viii</b>
<b>DAFTAR LAMPIRAN.....</b>	<b>ix</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
A. Latar Belakang.....	1
B. Rumusan Masalah.....	2
C. Tujuan Penelitian.....	2
D. Manfaat Penelitian.....	2
E. RuangLingkupPenelitian.....	3
F. Sistematika Penulisan.....	3
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
A. Landasan Teori.....	5
B. Penelitian Terkait.....	9
C. Kerangkapikir.....	12
<b>BAB III METODE PENELITIAN.....</b>	<b>13</b>
A. Tempat dan Waktu Penelitian.....	13
B. Bahan dan Alat Penelitian.....	13
C. Perancangan Sistem.....	13

D. Cara Kerja YOLOv8.....	18
E. Teknik Pengujian Sistem.....	20
F. Teknik Analisis Data.....	23
<b>BAB VI HASIL DAN PEMBAHASAN.....</b>	<b>26</b>
A. Pembuatan Model.....	26
B. Training Googlecolab.....	30
C. Pengujian Testing.....	50
<b>BAB V PENUTUP.....</b>	<b>52</b>
A. Kesimpulan.....	52
B. Saran.....	53
DAFTAR PUSTAKA.....	54



## DAFTAR GAMBAR

Gambar 1	Arsitektur Mobilenetssd.....	7
Gambar 2	kerangkapikir.....	12
Gambar 3	Flowchartdiagram.....	16
Gambar 4	Cara kerja YOLOv8.....	18
Gambar 5	Duduk yang baik dalam bekerja.....	26
Gambar 6	Duduk yang tidak baik dalam bekerja.....	27
Gambar 7	Upload Dataset.....	27
Gambar 8	Class Dataset.....	28
Gambar 9	Pelabelan Dataset.....	28
Gambar 10	Preprocessing Gambar.....	29
Gambar 11	Proses Training 1.....	30
Gambar 12	Grafik Hasil Training 1.....	31
Gambar 13	Hasil Validasi Gambar Pelatihan 1.....	34
Gambar 14	Hasil akurasi training 1.....	34
Gambar 15	Proses Training 2.....	35
Gambar 16	Grafik Hasil Training 2.....	36
Gambar 17	Validasi Pelatihan 2.....	39
Gambar 18	Hasil Akurasi Pelatihan 2.....	39
Gambar 19	Proses Training 3.....	40
Gambar 20	Grafik Pelatihan 3.....	41
Gambar 21	Hasil Validasi Pelatihan 3.....	44
Gambar 22	Hasil Akurasi Pelatihan 3.....	44

Gambar 23 Proses Training ke 4 .....46

Gambar 24 Grafik Training 4 .....47

Gambar 25 Hasil Validasi Training ke 4. ....49

Gambar 26 Hasil Akurasi Training ke 4.....49



# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Pada era modern ini, sebagian besar karyawan di berbagai sektor bekerja dengan duduk dalam jangka waktu yang lama, terutama di lingkungan kantor. Aktivitas ini berpotensi menyebabkan sejumlah masalah kesehatan, seperti nyeri punggung, gangguan postur tubuh, dan risiko penyakit jantung. Oleh karena itu, gerakan duduk sehat mulai diperkenalkan untuk mengedukasi masyarakat mengenai pentingnya postur tubuh yang baik dan penggunaan posisi duduk yang benar selama bekerja. Lee, (J. H., & Chang, T. Y. (2021)

Gerakan duduk sehat bertujuan untuk mengurangi dampak negatif dari kebiasaan duduk yang tidak sehat, dengan memperkenalkan cara duduk yang benar, sesekali berdiri, dan melakukan peregangan atau olahraga ringan. Sebaliknya, duduk tidak sehat biasanya terjadi ketika karyawan duduk dengan postur yang buruk, misalnya membungkuk, terlalu lama duduk tanpa bergerak, atau tanpa mendukung punggung dengan baik. Kebiasaan ini dapat menyebabkan ketegangan otot, gangguan sirkulasi darah, dan bahkan memengaruhi produktivitas kerja. (Occupational Safety and Health Administration (OSHA). (2020).

Kebiasaan duduk yang buruk tidak hanya berdampak pada kesehatan fisik, tetapi juga dapat memengaruhi kinerja karyawan. Riset menunjukkan bahwa postur tubuh yang tidak benar dapat mengakibatkan penurunan energi, kelelahan, dan

bahkan stres. Oleh karena itu, pemantauan kebiasaan duduk di tempat kerja menjadi penting untuk memastikan karyawan bekerja dengan efisien dan sehat.

Pengenalan gerakan duduk sehat melalui pelatihan atau program kesehatan di tempat kerja telah menjadi bagian dari inisiatif banyak perusahaan untuk menjaga kesehatan karyawan mereka. Namun, pengawasan terhadap kebiasaan duduk karyawan dalam keseharian masih merupakan tantangan, mengingat sulitnya melakukan pemantauan secara manual dan terus-menerus. (Pratt, A. D., & Green, R. P. (2021). Untuk mendukung program gerakan duduk sehat, teknologi dapat berperan penting dalam memantau postur tubuh karyawan secara otomatis. Salah satu teknologi yang sangat berpotensi adalah YOLO (You Only Look Once), yaitu algoritma deteksi objek yang dapat mengenali dan melacak objek dalam gambar atau video secara real-time dengan kecepatan tinggi. ( Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016).

1. Algoritma YOLO dapat digunakan untuk mendeteksi postur tubuh karyawan dalam lingkungan kerja, mendeteksi apakah posisi duduk mereka sehat atau tidak. Dengan menggunakan kamera yang dipasang di sekitar tempat kerja, algoritma YOLO akan menganalisis posisi tubuh dan mengidentifikasi apakah karyawan duduk dengan postur yang sehat atau tidak sehat. Algoritma ini sangat efisien dalam mendeteksi perbedaan kecil dalam postur tubuh, memberikan solusi otomatis dan real-time bagi manajer untuk menilai dan memberikan umpan balik kepada karyawan. Oleh dengan latar belakang di atas penulis ingin menganjag mengangkat judul Monitoring Gerakan duduk Dengan menggunakan algoritma Yolo Pada Karyawan

## **B. Rumusan Masalah**

1. Bagaimana memonitoring gerakan duduk sehat menggunakan metode algoritma YOLOv8 secara real-time

## **C. Tujuan Masalah**

1. Mengetahui bagaimana memonitoring gerakan duduk sehat dengan menggunakan metode YOLOv8 secara real-time

## **D. Manfaat penelitian**

Adapun tujuan dari penelitian ini berdasarkan rumusan masalah di atas adalah sebagai berikut.

Penelitian ini akan fokus pada penerapan sistem monitoring di lingkungan kerja.

Penelitian akan membahas implementasi metode YOLOv8 dalam mendeteksi dan menganalisis gerakan duduk, serta tantangan dan solusi dalam penerapannya.

Penelitian ini akan fokus pada objek gerakan duduk karyawan RSUD Sayang Rakyat dan mendeteksi objek berbagai bentuk gerakan duduk

## **E. Ruang Lingkup**

Fokus masalah yang dibatasi pada penelitian ini, akan dibatasi dengan memonitoring Gerakan duduk sehat pada karyawan, secara real-time dengan menggunakan Algoritma YOLOv8

## **F. Sistematika Penulisan**

Secara garis besar penulisan laporan tugas akhir ini terbagi menjadi beberapa bagian bab yang tersusun sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini menerangkan secara singkat tentang latar belakang masalah, tujuan penelitian, manfaat penelitian dari masalah yang diangkat, serta sistematika penulisan.

### **BAB II TINJAUAN PUSTAKA**

Bab ini berisi tentang teori yang berhubungan Judul yang diangkat dalam pembuatan skripsi.

### **BAB III METODE PENELITIAN**

Bab ini membahas tentang metode penelitian dan alat yang digunakan dalam pembuatan sistem dalam skripsi serta Langkah-langkah dalam penelitian.

### **BAB IV HASIL DAN PEMBAHASAN**

Memberikan gambaran umum tentang apa yang akan dibahas hasil penelitian

### **BAB V PENUTUP**

Bagian ini menyajikan rangkuman singkat dari hasil penelitian dan berisi saran yang didasarkan pada hasil penelitian

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Landasan Teori**

##### **1. Gerakan Duduk**

Gerakan duduk adalah aktivitas yang dilakukan oleh manusia dengan posisi tubuh yang terletak di atas permukaan, biasanya dengan bokong sebagai tumpuan. Aktivitas ini merupakan bagian penting dari kehidupan sehari-hari, terutama dalam konteks pekerjaan dan interaksi sosial. Posisi duduk yang benar yakni duduk dengan posisi tegap sudut 90 derajat yang mengharuskan seseorang untuk selalu menjaga bagian-bagian vital pada tubuh untuk selalu berada pada postur yang tepat (Supriyanto et al. 2023).

##### **2. Duduk sehat dan tidak sehat**

Posisi duduk yang tidak ergonomis, seperti duduk membungkuk dalam waktu lama, dapat meningkatkan risiko nyeri punggung bawah. duduk dalam posisi statis yang buruk menyebabkan otot bekerja terlalu keras tanpa cukup waktu untuk pemulihan, sehingga aliran darah ke otot terganggu, sedangkan posisi duduk yang ergonomis seperti duduk dengan lurus tidak membungkuk bisa mendapatkan kenyamanan dalam bekerja (Amin et al. 2023). Yang tepat. Pemahaman isi gambar ini mencakup ekstraksi informasi seperti objek, deskripsi teks, dan model tiga dimensi (Astuti, 2021).

### 3. YOLOv8

adalah model SOTA (tercanggih) yang mutakhir yang dibangun berdasarkan keberhasilan versi YOLO sebelumnya dan memperkenalkan fitur dan penyempurnaan baru untuk lebih meningkatkan kinerja dan fleksibilitas. Fitur dan penyempurnaan ini meliputi jaringan backbone baru, kepala deteksi bebas jangkar baru, dan fungsi kerugian baru YOLO, yang merupakan singkatan dari "You Only Look Once," adalah algoritma deteksi objek yang revolusioner dalam bidang penglihatan komputer. mengubah cara deteksi objek dilakukan dengan pendekatan end-to-end yang memungkinkan deteksi objek dilakukan dalam satu langkah tunggal

Algoritma ini membagi citra input menjadi grid berukuran  $S \times S \times S$ , jika pusat suatu objek jatuh ke dalam sel grid tertentu, maka sel tersebut bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid kemudian memprediksi sejumlah kotak batas (bounding boxes) dan skor kepercayaan untuk setiap kotak tersebut. Kotak batas didefinisikan dengan lima parameter: posisi tengah kotak ( $x, y$ ), lebar ( $w$ ), tinggi ( $h$ ), dan skor kepercayaan yang mencerminkan seberapa yakin model bahwa kotak tersebut mengandung objek serta seberapa akurat prediksinya. Salah satu keunggulan utama dari YOLO adalah kecepatannya. Dengan kemampuan untuk memproses gambar hingga 45 frame per detik (FPS) pada versi awalnya dan bahkan lebih cepat pada versi terbaru seperti YOLOv4 dan YOLOv7, algoritma ini sangat cocok untuk aplikasi real-time seperti pengawasan video dan mobil otonom. Selain itu, meskipun cepat, YOLO tetap

mempertahankan akurasi yang tinggi, sering kali mengungguli metode lain dalam hal generalisasi dari gambar alami ke domain lain.

Lapisan pooling mengurangi dimensi fitur untuk mengurangi beban komputasi pooling juga membantu meningkatkan ketahanan model terhadap pergeseran dan distorsi kecil dalam gambar kesederhanaan dengan mengaburkan prediksi bounding box dan kelas objek dalam satu langkah, YOLO menyederhanakan proses deteksi dibandingkan dengan metode tradisional

Teknologi deteksi objek, khususnya yang berbasis kecerdasan buatan (AI) dan pembelajaran mendalam (deep learning), telah mengalami perkembangan pesat. Salah satu metode yang populer dalam deteksi objek adalah algoritma YOLO (You Only Look Once). Metode ini memungkinkan sistem untuk mendeteksi dan mengenali objek dalam citra atau video secara real-time dengan akurasi yang tinggi. Penerapan YOLO dalam mendeteksi kehadiran peternak di dalam peternakan ayam dapat memberikan solusi efektif untuk meningkatkan pengawasan secara otomatis dan terus-menerus tanpa memerlukan intervensi peternak yang berlebihan. Dengan menggunakan YOLO v8, yang dikenal karena kecepatan dan akurasinya dalam deteksi objek, proses ini dapat mendeteksi berapa objek terdeteksi yaitu Peternak pada peternakan ayam. Teknologi ini memungkinkan identifikasi dan pelacakan objek, seperti keberadaan Peternak, selama berada di peternakan. Penggunaan metode YOLO v8 dalam mendeteksi keberadaan objek pada peternakan ayam bertujuan untuk mengetahui efisiensi dan akurasi keberadaan Peternak pada peternakan ayam. (Ikbal et al. 2024)

Salah satu kecerdasan peternak adalah kemampuan dalam mengenali suatu objek yang ada disekitarnya misalkan objek peternak. Peternak dapat mengenali sebuah objek, dengan menggunakan mata sebagai indra pengelihatn untuk menangkap sebuah citra objek yang kemudian akan direkam dan disimpan dalam memori otak. Perkembangan teknologi saat ini menunjang sebuah mesin untuk dapat belajar seperti peternak dalam mengenali sebuah objek. Mesin membutuhkan kecerdasan buatan untuk dapat mengenali dan mengklasifikasikan sebuah objek citra digital. Deep learning merupakan bagian dalam ilmu AI (artificial intelligence) yang merupakan penerapan transformasi abstraksi nonlinier dan lanjutan kedalam basis data . Salah satu algoritma yang sering digunakan dalam bidang deteksi objek adalah YOLO (*You Only Look One*) yang masuk pada katgori Algoritma CNN yang cara kerjanya meniru cara kerja otak peternak. Algoritma YOLO berdasarkan ujicoba dapat beroperasi pada kecepatan 45 fps (frame per second) dengan menggunakan kartu grafis Titan X Dengan penelitian tersebut algoritma YOLO sangat baik digunakan pada proyek yang memiliki sifat realtime seperti pada deteksi objek pada image ataupun video. Seiring banyaknya penelitian pada bidang computer vision maka algoritma deteksi objek, YOLO berkembang dengan cepat dan yang terbaru YOLO v8. Perbedaan pada setiap versi YOLO terletak dari peningkatan akurasi dan kecepatan pendeteksian objek. (Setiyadi, Utami, |891, et al. 2023)



## **Blok Utama**

Langkah pertama untuk memahami arsitektur YOLO adalah memahami bahwa ada 3 blok penting dalam algoritma dan semuanya akan terjadi di blok-blok ini, yaitu: *Tulang Punggung*, *Leher*, dan *Kepala*. Fungsi masing-masing blok dijelaskan di bawah ini.

### **Tulang punggung:**

Fungsi: Tulang punggung, yang juga dikenal sebagai ekstraktor fitur, bertanggung jawab untuk mengekstraksi fitur-fitur yang bermakna dari input.

Aktivitas:

1. Menangkap pola-pola sederhana di lapisan-lapisan awal, seperti tepi dan tekstur.
2. Dapat memiliki beberapa skala representasi saat Anda melakukannya, menangkap fitur-fitur dari berbagai tingkat abstraksi.  
- Akan menyediakan representasi input yang kaya dan hierarkis

### **Leher:**

Fungsi: Leher bertindak sebagai jembatan antara tulang punggung dan kepala, melakukan operasi fusi fitur dan mengintegrasikan informasi kontekstual. Pada dasarnya, Leher menyusun piramida fitur dengan menggabungkan peta fitur yang diperoleh oleh Tulang Punggung, dengan kata lain, leher mengumpulkan peta fitur dariberbagai tahap tulang punggung.

Aktivitas:

1. Melakukan penggabungan atau fusi fitur dengan skala yang berbeda untuk memastikan bahwa jaringan dapat mendeteksi objek dengan ukuran yang berbeda.
2. Mengintegrasikan informasi kontekstual untuk meningkatkan akurasi deteksi dengan mempertimbangkan konteks yang lebih luas dari pemandangan.
3. Mengurangi resolusi spasial dan dimensionalitas sumber daya untuk memfasilitasi komputasi, fakta yang meningkatkan kecepatan tetapi juga dapat mengurangi kualitas model.

#### **Kepala:**

Fungsi: Kepala adalah bagian akhir dari jaringan dan bertanggung jawab untuk menghasilkan keluaran jaringan, seperti kotak pembatas dan skor keyakinan untuk deteksi objek

Aktivitas: Menghasilkan kotak pembatas yang terkait dengan kemungkinan objek dalam gambar.

1. Menetapkan skor keyakinan untuk setiap kotak pembatas untuk menunjukkan seberapa besar kemungkinan suatu objek ada.
2. Mengurutkan objek dalam kotak pembatas menurut kategorinya.

#### **4. *Deep Learning***

*Deep Learning* adalah cabang kecerdasan buatan canggih yang terinspirasi dari struktur otak manusia, menggunakan jaringan saraf tiruan berlapis-lapis untuk mencapai performa luar biasa dalam tugas-tugas seperti deteksi objek, pengenalan suara, dan terjemahan bahasa.

Bayangkan sebuah komputer yang mampu mempelajari pola kompleks dari gambar, memahami percakapan manusia, dan bahkan menerjemahkan bahasa dengan tingkat akurasi yang tinggi. Itulah yang ditawarkan oleh Deep Learning, sebuah terobosan dalam kecerdasan buatan (AI) yang meniru cara kerja otak manusia. *Deep Learning* memanfaatkan jaringan saraf tiruan (*artificial neural network*) yang tersusun atas berlapis-lapis neuron, seperti neuron pada otak manusia. Neuron-neuron ini saling terhubung dan berkomunikasi satu sama lain, memungkinkan jaringan untuk belajar dan beradaptasi dengan informasi baru (Raup et al., 2022).

PyTorch adalah sebuah *framework open-source* yang digunakan untuk pengembangan aplikasi machine learning dan deep learning. Dikembangkan oleh tim riset kecerdasan buatan di *Facebook*, *PyTorch* menyediakan alat yang fleksibel dan efisien untuk membangun dan melatih model-model neural network. Salah satu fitur utama dari *PyTorch* adalah kemampuannya untuk melakukan komputasi tensor yang mirip dengan *NumPy*, tetapi dengan kemampuan untuk memanfaatkan GPU, sehingga mempercepat proses pelatihan model.

## 5. *Roboflow*

*Roboflow* adalah *framework* pengembang *computer vision* untuk pengumpulan data yang lebih baik ke prapemrosesan, dan teknik pelatihan model. *Roboflow* memiliki kumpulan data publik yang tersedia bagi pengguna

dan juga memiliki akses bagi pengguna untuk mengunggah data khusus mereka sendiri. (Dio Riza Pratama et al. 2022).

## B. Penelitian terkait

1. **Pendeteksian objek menggunakan openCV dan metode YOLOv4-tINY untuk membantu tunanetra (Randy muh Yusup et al., 2024).** Pada penelitian yang di lakukan oleh Randy muh yusuf, aldof faris anugrah, dinda desmonda muslimah, sri mentari widya ningum permana, shindi yuliyani penelitian tersebut bertujuan untuk membangun sebuah sitem deteksi objek yang handal dan efisien menggunakan OpenCV dan YOLOv4-Tiny penelitian mengevaluasi kinerja sistem pda tolak ukur standar visi kmputer dan berbagai penerapannya.

Hasil dari penelitian tersebut menunjukkan bahwa YOLOv4-Tniy menawarkan keseimbangan antara akurasi dan kesepataan meskipu akurasinya ( $mAP$  20-30%) lebih rendah dibandingkan dendan metode YOLOv4 penuh dan beberapa model lain, YOLOv4-Tiny tetap mampu mendeteksi berbagai objek (orang, kendaraan, hewan, dll.) dengan baik. Kemampuan deteksi objek kecilnya mungkin sedikit terbatas karena arsitekturnya yang diperkecil. Namun, kelebihan utama YOLOv4-Tiny adalah kecepatan inferensinya yang jauh lebih cepat dibandingkan dengan YOLOv4 penuh. Hal ini membuatnya ideal untuk aplikasi real-time pada perangkat dengan sumber daya terbatas, seperti sistem tertanam atau perangkat edge. Kecepatan inferensi yang tinggi memungkinkan YOLOv4Tiny untuk memproses streaming video atau umpan kamera secara *realtime*. Meskipun akurasinya tidak setinggi model lain, YOLOv4-Tiny tetap

menjadi pilihan yang menarik untuk aplikasi *real-time* yang membutuhkan kecepatan tinggi.

## **2. Pengembangan Aplikasi Pendeteksi Objek Untuk Tunanetra Menggunakan Operator Tepi Canny Dengan Library OpenCV Berbasis Android (Pradana & Sula, 2023).**

Pada penelitian yang dilakukan Teguh Pradana, Eko Hajianto Sula penelitian tersebut bertujuan untuk mengembangkan aplikasi pendeteksi objek berbasis Android yang dapat digunakan oleh penyandang tunanetra dan anak usia dini. Aplikasi ini bertujuan untuk membantu pengguna dalam mengenali objek-objek di sekitar mereka dengan cara yang mudah dan interaktif.

Hasil dari penelitian tersebut Penelitian ini menghasilkan aplikasi pendeteksi objek berbasis Android untuk membantu penyandang tunanetra dan anak usia dini. Halaman utama aplikasi menampilkan logo, nama, versi aplikasi, nama pembuat, dan tombol untuk dua mode pendeteksian: deteksi tepi objek dan deteksi objek terdekat. Mode deteksi tepi menggunakan kamera smartphone untuk menampilkan tepi objek dengan latar belakang hitam dan objek berwarna putih. Pengguna dapat menyimpan gambar hasil deteksi untuk pembelajaran anak usia dini. Mode deteksi objek terdekat menampilkan hasil deteksi dengan label objek dan persentase akurasi, serta mengonversi label objek menjadi suara untuk membantu tunanetra. Penelitian ini menggunakan Operator Tepi Canny dan OpenCV untuk deteksi tepi, serta TensorFlow Lite untuk deteksi objek. Saran

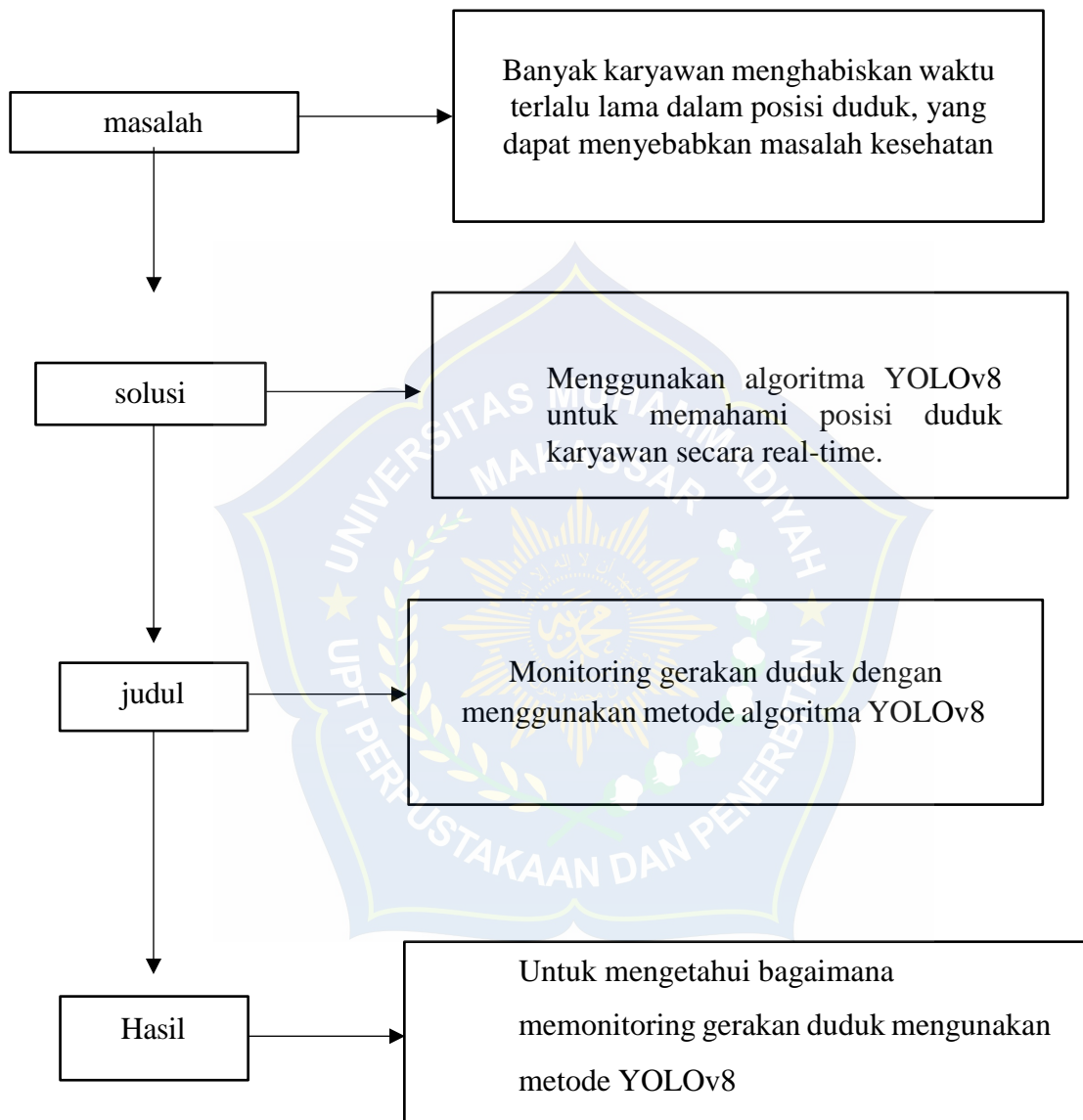
pengembangan meliputi penambahan data model untuk lebih banyak objek dan penyimpanan otomatis gambar hasil deteksi. Penelitian ini berhasil menciptakan aplikasi yang bermanfaat dan dapat ditingkatkan dengan fitur tambahan dan data yang lebih luas.

### **3. Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo (Aini et al., 2021).**

Pada penelitian yang di lakukan Qurotul Aini, Ninda Lutfiani, Hendra Kusumah, Muhammad Suzaki Zahran penelitian tersebut bertujuan

Hasil dari penelitian tersebut perkembangan algoritma pengenalan objek YOLO, mulai dari versi awal hingga versi terbaru. YOLO diciptakan untuk menjawab kebutuhan akan model pengenalan objek yang cepat dan akurat. Versi awal YOLO, seperti YOLO v1, memiliki kecepatan tinggi namun akurasi masih rendah. Seiring perkembangannya, YOLO terus ditingkatkan dengan berbagai versi baru, seperti YOLOv2, YOLO9000, YOLOv3, dan YOLO v4. Setiap versi memiliki fokus pengembangan yang berbeda, seperti peningkatan recall dan lokalisasi, pengoptimasian komputasi paralel, dan lain sebagainya. Versi terbaru YOLO, yaitu YOLO v5, masih tergolong baru dan belum memiliki banyak informasi yang tersedia. YOLOv5 menggunakan *framework PyTorch* dan hadir dalam berbagai versi dengan spesifikasi dan kegunaannya masing-masing. Secara keseluruhan, YOLO merupakan algoritma pengenalan objek yang terus berkembang dengan berbagai kelebihan dan kekurangan di setiap versinya.

YOLO v5, sebagai versi terbaru, masih perlu diteliti lebih lanjut untuk mengetahui potensinya secara lebih mendalam.



## **BAB III**

### **METODE PENELITIAN**

#### **A. Tempat dan Waktu Penelitian**

Tempat penelitian merupakan suatu tempat atau objek yang akan di lakukan suatu penelitian. Penentuan lokasi penelitian merupakan langkah penting dalam proses penelitian karena memudahkan peneliti untuk melakukan penelitian. Lokasi RSUD Sayang Rakyat, Jl. Pahlawan No.10000, Bulurokeng, Kec. Biringkanaya, Kota Makassar, Sulawesi Selatan 90243 pengambilan data pada penelitian ini di lakukan di kecamatan Biringkanaya, Kota Makassar. Waktu kegiatan penelitian di mulai pada bulan oktober 2024 sampai semua proses pengumpulan data selesai

#### **B. Alat dan Bahan**

Adapun alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut:

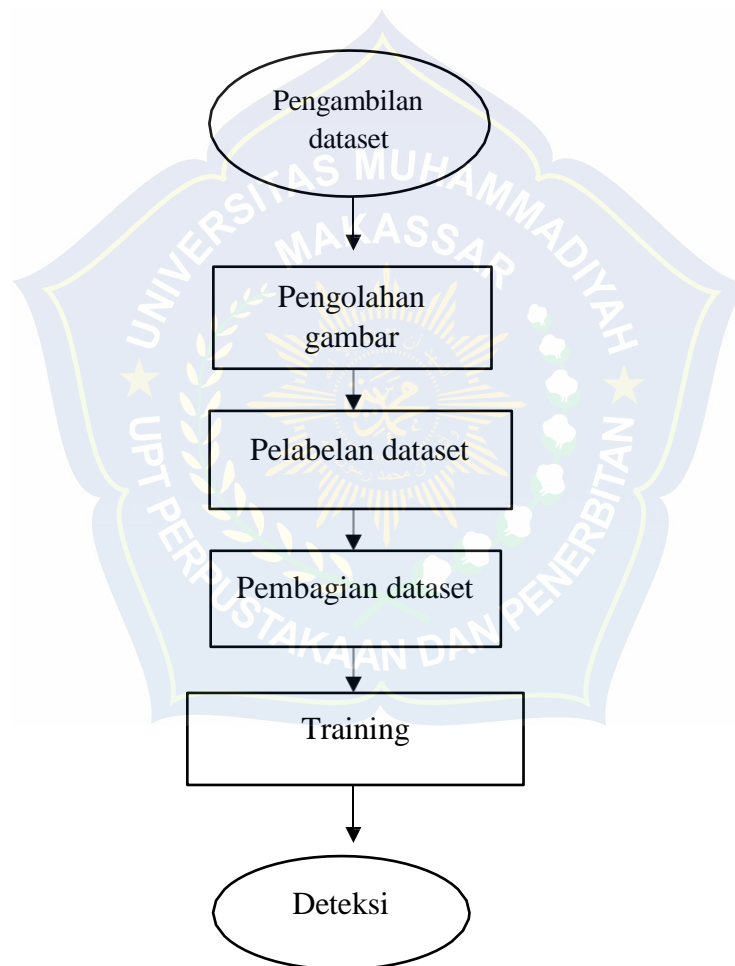
1. Kebutuhan *Hardware* (Perangkat Keras)
  - a. Laptop Acer aspire A314-32
  - b. Oppo A54 RAM 4 GB CPH 2239
2. Kebutuhan software (perangkat lunak)
  - a. VS Code
  - b. Google Cola

c. Python

d. Roboflow (Untuk anotasi gambar)

### C. Perancangan Sistem

Membangun sistem deteksi objek secara real-time yang dapat mengenali dan memberi kotak pembatas (bounding box) serta label pada objek-objek yang ada di dalam gambar atau video.



#### 1. Pengambilan Dataset

Data diambil dalam bentuk gambar dengan menggunakan kamera pada handphone untuk mengambil gambar dari objek-objek yang telah di

tentukan dan juga berupa gambar objek-objek yang tersedia pada internet guna mengumpulkan data. Nantinya dataset ini akan digunakan sebagai contoh untuk melatih sistem pendeteksi objek yang sedang dikerjakan

## 2. **Pengolahan Gambar**

Pada tahap *preprocessing* data, akan dilakukan proses *resize* untuk mengubah ukuran gambar-gambar yang ada. Proses ini penting untuk memastikan bahwa semua gambar memiliki ukuran yang seragam dan sesuai dengan persyaratan input dari model yang akan dilatih. Selain itu, proses *resize* juga membantu dalam mengoptimalkan penggunaan memori dan sumber daya komputasi selama pelatihan model. Dengan demikian, langkah ini merupakan bagian krusial dalam mempersiapkan data gambar sebelum digunakan untuk pelatihan model machine learning.

## 3. **Pelabelan Dataset**

Proses ini akan melabeli data yang telah diproses menggunakan Roboflow. Tujuan Roboflow adalah untuk memberikan identifikasi atau label pada setiap gambar objek dalam sistem deteksi objek yang dibuat.

## 4. **Pembagian Dataset**

Setelah dataset diberi label, dataset akan dibagi menjadi tiga bagian: data training, data validasi, dan data testing. Model akan dilatih menggunakan data yang ada dalam bagian training. Bagian data validasi digunakan untuk mengevaluasi kinerja model dengan mengidentifikasi

dan menguji data, sehingga memungkinkan penilaian akurasi model terhadap data baru yang belum pernah dilihat sebelumnya.

## 5. Training

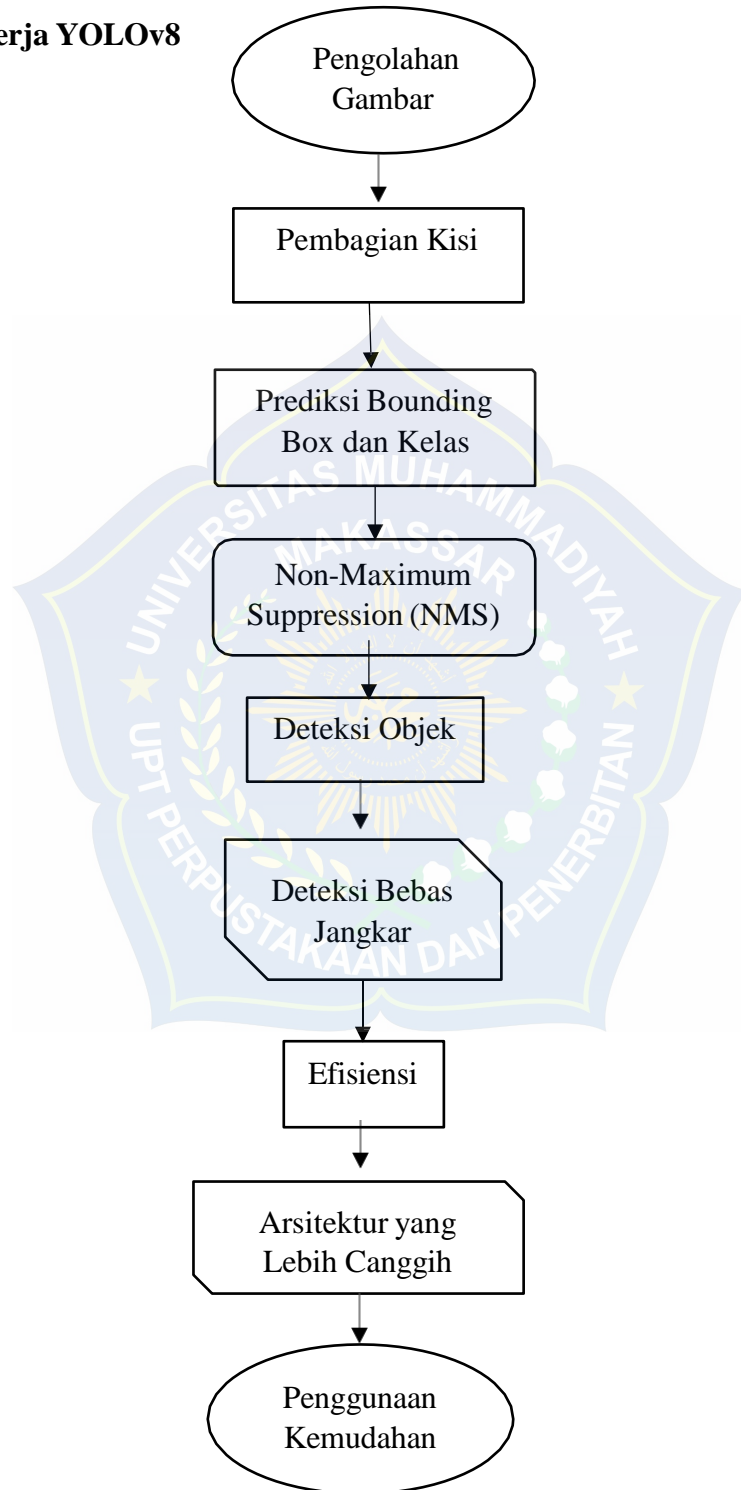
Pada tahap ini, data yang telah dibagi menjadi data training dan data testing akan dipanggil dan diproses. Data dalam bentuk API yang tersimpan dalam LabelImg akan diakses melalui aplikasi Gogle Colab. Setelah data tersedia, pengujian model dilakukan dengan menggunakan algoritma YOLOv8. Akurasi akan digunakan sebagai parameter utama untuk mengevaluasi kinerja model. Jika hasil pengujian menunjukkan bahwa akurasi model masih rendah, maka proses pelatihan ulang akan dilakukan untuk meningkatkan akurasi. Pelatihan ulang ini bertujuan untuk mengoptimalkan model agar dapat menghasilkan prediksi yang lebih baik lagi.

## 6. Deteksi

Proses pendeteksian dimulai dengan memasukkan gambar yang telah terkumpul ke dalam sistem. Setelah gambar dimasukkan, sistem menginisiasi proses analisis menggunakan YOLOv8 untuk mengenali objek-objek. Selama proses pengujian, algoritma YOLOv8 digunakan untuk menandai objek-objek yang ditemukan dalam gambar dengan *bouding box* sebagai label. Ini memungkinkan model untuk mengklasifikasikan objek dengan tepat dan memisahkan mereka dari latar belakang. Output dari proses ini mencakup nilai akurasi, yang merupakan

ukuran seberapa baik model dapat mengidentifikasi objek, serta ROI (*Region of Interest*) untuk menentukan area deteksi yang relevan.

#### D. Cara Kerja YOLOv8



## **1. Pengambilan dataset**

Data di ambil menggunakan kamera smartphone dari beberapa subjek dan objek.

## **2. Pengolahan gambar**

Dalam pengolahan data di perlukan preprocessing data yang di mana Preprocessing merupakan tahap krusial sebelum memasukkan citra ke dalam model atau algoritma, yang bertujuan untuk menyiapkan dan meningkatkan kualitas atau kesesuaian citra sesuai dengan kebutuhan.

## **3. Pemberian label**

Dalam langkah ini, dataset telah di proses dengan pengolahan gambar yang di beri label menggunakan platform Roboflow. Roboflow bertujuan untuk menambahkan kotak pembatas pada setiap objek yang terdapat dalam gambar.

## **4. Pembagian dataset**

Dataset yang telah di beri label akan di bagi menjadi tiga bagian yaitu data training, data validasi, dan data testing. Data training di gunakan untuk melatih model deteksi, sementara data validasi di gunakan untuk mengevaluasi performa model selama proses pelatihan. Data testing kemudian di gunakan untuk menguji performa model dan mengukur akurasi dengan menggunakan data yang belum pernah di lihat sebelumnya

## 5. Training

Pada langkah ini, data yang telah di bagi menjadi data latih dan data uji akan di akses. Data yang tersedia dalam bentuk API di Roboflow akan di ambil melalui lingkungan pengembangan seperti VSCode

## 6. Deteksi

YOLO adalah algoritma deteksi objek satu tahap (*one-stage object detector*) yang memproses gambar sekali saja untuk mendeteksi semua objek di dalamnya, berbeda dengan metode seperti R-CNN yang membutuhkan beberapa tahap.

### 1. Pengolahan Gambar:

YOLOv8 menerima gambar sebagai input dan memprosesnya menggunakan CNN. CNN ini terdiri dari beberapa lapisan konvolusi yang digunakan untuk mengekstrak fitur dari gambar.

### 2. Pembagian Kisi:

Gambar kemudian dibagi menjadi kisi-kisi sel yang berukuran sama. Setiap sel kisi bertanggung jawab untuk memprediksi objek yang berada di dalam sel tersebut.

### 3. Prediksi Bounding Box dan Kelas:

Untuk setiap sel kisi, YOLOv8 memprediksi sejumlah kotak pembatas dan probabilitas kelas untuk setiap kotak pembatas.

#### **4. Non-Maximum Suppression (NMS):**

Setelah memprediksi kotak pembatas, YOLOv8 menggunakan NMS untuk menyingkirkan kotak pembatas yang tumpang tindih dan memilih kotak pembatas yang paling mungkin untuk setiap objek.

#### **5. Deteksi Objek:**

Hasil akhir adalah daftar objek yang dideteksi, lengkap dengan lokasi kotak pembatas dan kelas objeknya.

Peningkatan YOLOv8 dibandingkan versi sebelumnya:

##### **Deteksi Bebas Jangkar:**

YOLOv8 beralih ke deteksi bebas jangkar, yang berarti ia tidak menggunakan kotak jangkar yang telah ditentukan sebelumnya. Hal ini meningkatkan generalisasi model dan mempercepat pembelajaran.

##### **Efisiensi:**

YOLOv8 dirancang untuk lebih efisien, baik dalam segi waktu inferensi maupun penggunaan memori.

##### **Arsitektur yang Lebih Canggih:**

YOLOv8 memperkenalkan arsitektur yang lebih canggih, termasuk Attention mechanism dan Context aggregation.

### **Kemudahan Penggunaan:**

YOLOv8 dilengkapi dengan CLI (Command Line Interface) dan paket Python yang membuat pelatihan dan penggunaan model lebih intuitif, seperti yang dijelaskan di

### **E. Teknik Pengujian Sistem YOLOv8**

Sistem pengujian empiris dalam penelitian algoritma pendeteksi objek sekitar berbasis deep learning untuk mendeteksi gerakan duduk secara real-time dengan menggunakan metode YOLOv8 merupakan proses yang krusial untuk memastikan keefektifan dan keandalan sistem yang dikembangkan. Pengujian ini melibatkan serangkaian langkah dan metode yang dirancang untuk mengevaluasi berbagai aspek kinerja sistem, seperti akurasi, presisi, dan recall dalam mendeteksi objek

Berikut adalah beberapa langkah umum yang biasanya dilakukan dalam pengujian empiris:

#### **1. Pengumpulan Dataset:**

Tahap awal pengembangan sistem melibatkan pengumpulan dataset yang berisi berbagai objek sehari-hari yang relevan bagi penyandang tunanetra. Dataset ini kemudian perlu dibagi menjadi beberapa subset, yaitu subset pelatihan, subset validasi, dan subset pengujian. Setiap gambar dalam dataset harus diberi label yang menunjukkan posisi dan nama objek yang ada di dalamnya.

#### **2. Pembuatan Model**

Model deteksi objek deep learning, seperti YOLOv8, merupakan salah satu alat yang ampuh untuk mendeteksi objek dalam gambar dan video. Namun, model ini tidak selalu dapat langsung digunakan dalam penelitian, karena perlu disesuaikan dengan kebutuhan spesifik penelitian.

### 3. **Pelatihan Model**

Model deteksi objek dilatih dengan menggunakan subset data dari dataset yang telah disiapkan. Proses ini melibatkan beberapa iterasi, di mana model secara bertahap belajar mengenali pola dan fitur yang terkait dengan objek-objek dalam gambar. S

### 4. **Validasi Model**

Setelah model deep learning selesai dilatih, tahapan selanjutnya adalah mengujinya dengan menggunakan data validasi yang terpisah dari dataset utama. Hal ini bertujuan untuk memastikan bahwa model tidak hanya mampu mempelajari data pelatihan dengan baik, tetapi juga dapat memberikan hasil yang akurat pada data baru yang belum pernah dilihat sebelumnya. Proses validasi ini dapat dilakukan dengan mengukur berbagai metrik, seperti akurasi deteksi objek, presisi, recall, dan F1-score.

### 5. **Evaluasi Kinerja**

Setelah melalui proses validasi, langkah terakhir dalam pengembangan model adalah pengujian kinerja pada subset data baru yang terpisah dari dataset pelatihan. Tujuannya adalah untuk mengetahui kemampuan model dalam mengenali objek-objek baru yang belum pernah

dilihat sebelumnya. Evaluasi kinerja ini dapat mencakup berbagai metrik, seperti akurasi deteksi objek, waktu inferensi, dan metrik relevan lainnya. Hasil evaluasi ini kemudian digunakan untuk menilai performa model secara keseluruhan.

## **F. Teknik Analisis Data**

Jika performa model belum sesuai dengan target, proses pengujian ini dapat diulang dengan melakukan penyesuaian pada model atau parameternya. Penyesuaian ini dapat berupa memodifikasi arsitektur model, mengubah algoritma pelatihan, atau menambahkan data pelatihan baru. Proses pengujian empiris yang sistematis dan terukur ini sangat penting untuk memastikan bahwa model deteksi objek yang dikembangkan memiliki performa yang optimal dan dapat diandalkan dalam aplikasi dunia nyata.

Analisis data adalah langkah-langkah yang melibatkan pengumpulan, pengelompokan, dan seleksi data secara terstruktur dari berbagai sumber seperti wawancara, catatan lapangan, dan dokumentasi. Proses ini mencakup pengorganisasian data ke dalam kategori yang relevan, merangkumnya menjadi unit-unit yang dapat dipahami, melakukan sintesis informasi, mengidentifikasi pola yang muncul, menentukan data yang krusial untuk dianalisis lebih lanjut, dan membuat kesimpulan yang mudah dimengerti. Terdapat tiga teknik utama yang digunakan dalam analisis data:

### **1. Reduksi Data (*Data Reduction*)**

Mereduksi data adalah proses mengurangi jumlah data yang relevan atau signifikan tanpa kehilangan informasi yang penting. Tujuan dari reduksi data adalah untuk membuat dataset yang lebih kecil tetapi masih mempertahankan representasi yang akurat dari data asli. Reduksi data membantu dalam mempercepat analisis data, mengurangi beban komputasi, dan meningkatkan efisiensi penyimpanan dan pengambilan data. Ini menjadi penting terutama ketika bekerja dengan dataset yang besar atau kompleks.

## 2. **Penyajian Data (*Data Display*)**

Penyajian data adalah suatu proses yang terstruktur dalam mengatur informasi dengan sistematis untuk memfasilitasi analisis mendalam serta mengambil kesimpulan yang tepat. Dalam proses ini, informasi disusun secara terstruktur dan didesain dengan cermat untuk memudahkan pemahaman dan interpretasi. Penyajian data tidak hanya sekadar menampilkan informasi, tetapi juga mempertimbangkan bagaimana cara terbaik untuk menyampaikan pesan secara efektif kepada pemirsa atau pengguna. Dengan penyajian data yang baik, informasi dapat disampaikan dengan jelas dan mudah dipahami, memungkinkan untuk pengambilan keputusan yang lebih baik dan tindakan yang tepat.

## 3. **Penarikan Kesimpulan (*Concluding Drawing Verification*)**

Penarikan kesimpulan yang diambil merupakan hasil awal dari analisis dan dapat mengalami transformasi seiring dengan penemuan bukti baru

dalam tahap pengumpulan data yang berikutnya. Proses ini mencakup upaya untuk menemukan atau mengurai makna, pola, dan penjelasan yang tersembunyi dalam dataset.

Penarikan kesimpulan menandai tahap penutup dalam analisis data, di mana informasi yang dikumpulkan dianalisis secara komprehensif untuk mendapatkan pemahaman yang lebih dalam.



## BAB IV

### BAB IV HASIL DAN PEMBAHASAN

#### A. Pembuatan Model

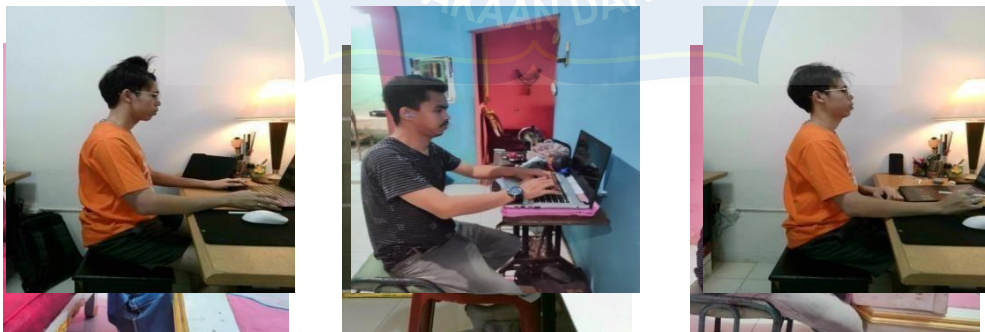
##### 1. Pengambilan Dataset

Dataset gambar di ambil menggunakan kamera HP oppo a54 ram 4GB dan terdiri dari 200 gambar postur duduk. Dataset ini mencakup berbagai

#### A. Pembuatan Model

*Table 1. Jumlah Dataset*

Tempat pengambilan data	Objek data yang di ambil	Jumlah Data Gambar
RSUD Sayang Rakyat	Duduk yang sehat dalam kesehatan	100
	Duduk tidak tidak sehat dalam kesehatan	100



*Gambar 5. Duduk yang sehat dalam kesehatan*

Di tengah rutinitas sehari-hari, terutama bagi mereka yang bekerja atau belajar dalam waktu lama, aktivitas duduk menjadi hal yang tak terhindarkan. Namun, tidak semua orang menyadari pentingnya menjaga postur duduk yang sehat.

Padahal, duduk dengan posisi yang benar sangat berpengaruh terhadap kesehatan tubuh, terutama tulang belakang.

Duduk sehat berarti menjaga postur tubuh tetap tegak, bahu rileks, dan punggung disangga dengan baik oleh sandaran kursi. Posisi kaki juga sebaiknya menapak rata di lantai, dengan lutut membentuk sudut 90 derajat. Hindari duduk menyilangkan kaki terlalu lama, membungkuk, atau menyender terlalu jauh ke belakang, karena hal ini bisa menimbulkan nyeri otot dan gangguan tulang belakang dalam jangka panjang.

Selain postur, durasi duduk juga perlu diperhatikan. Disarankan untuk berdiri atau melakukan peregangan setiap 30 hingga 60 menit sekali guna melancarkan peredaran darah dan mencegah kekakuan otot. Menggunakan kursi yang ergonomis dan mengatur tinggi meja kerja juga bisa membantu menciptakan kebiasaan duduk sehat.

Dengan membiasakan diri duduk secara sehat, kita tidak hanya menjaga kenyamanan saat *beraktivitas*, tetapi juga mencegah berbagai gangguan kesehatan



yang bisa muncul di kemudian hari. Duduk sehat, tubuh pun tetap bugar dan produktivitas meningkat.

*gambar 6. Duduk tidak sehat dalam kesehatan*

## 2. Duduk tidak Sehat

Di era modern ini, *aktivitas* duduk dalam jangka waktu lama telah menjadi



bagian dari gaya hidup banyak orang, terutama mereka yang bekerja di depan komputer. Namun, tanpa disadari, kebiasaan duduk dalam posisi yang salah dan terlalu lama dapat membawa dampak buruk bagi kesehatan tubuh.

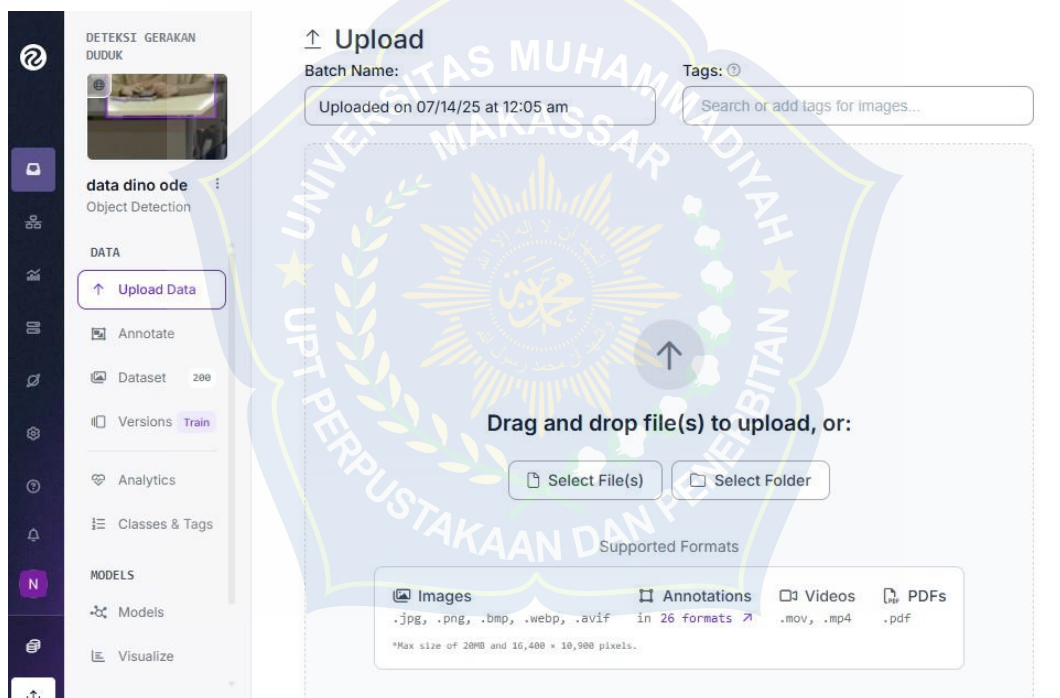
Duduk dengan posisi membungkuk, tanpa sandaran yang tepat, atau terlalu sering menyilangkan kaki bisa menyebabkan berbagai masalah kesehatan, seperti nyeri punggung, leher kaku, dan gangguan pada tulang belakang. Lebih parah lagi, duduk terlalu lama tanpa diselingi *aktivitas* fisik juga meningkatkan risiko penyakit jantung, obesitas, dan bahkan diabetes tipe

Sayangnya, banyak orang mengabaikan pentingnya postur duduk yang benar. Mereka terbiasa duduk berjam-jam tanpa jeda, karena tuntutan pekerjaan atau kebiasaan menatap layar gadget. Padahal, solusi sederhana seperti memperbaiki posisi duduk, menggunakan kursi ergonomis, serta rutin berdiri dan meregangkan tubuh setiap satu jam sekali bisa membuat perbedaan besar bagi kesehatan.

Oleh karena itu, mari kita mulai memperhatikan cara kita duduk setiap hari. Duduklah dengan tegak, pastikan kaki menapak rata di lantai, dan sesekali bergerak agar tubuh tetap aktif. Dengan langkah kecil ini, kita bisa menjaga kesehatan tubuh jangka panjang dan terhindar dari dampak buruk duduk yang tidak sehat.

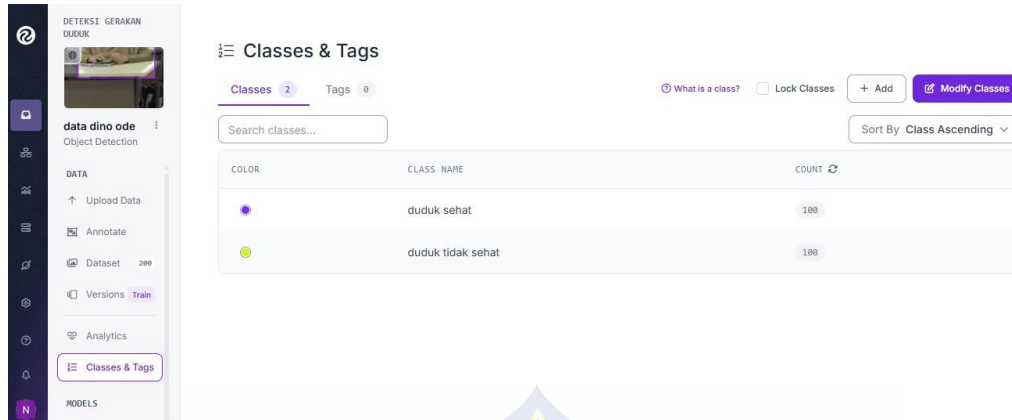
#### a. Upload dataset

Gambar yang telah di kumpulkan sebelumnya akan di unggah ke *Roboflow* sebelum proses pelabelan di lakukan



Gambar 7. Upload Dataset

## b. Pembuatan Class Dataset

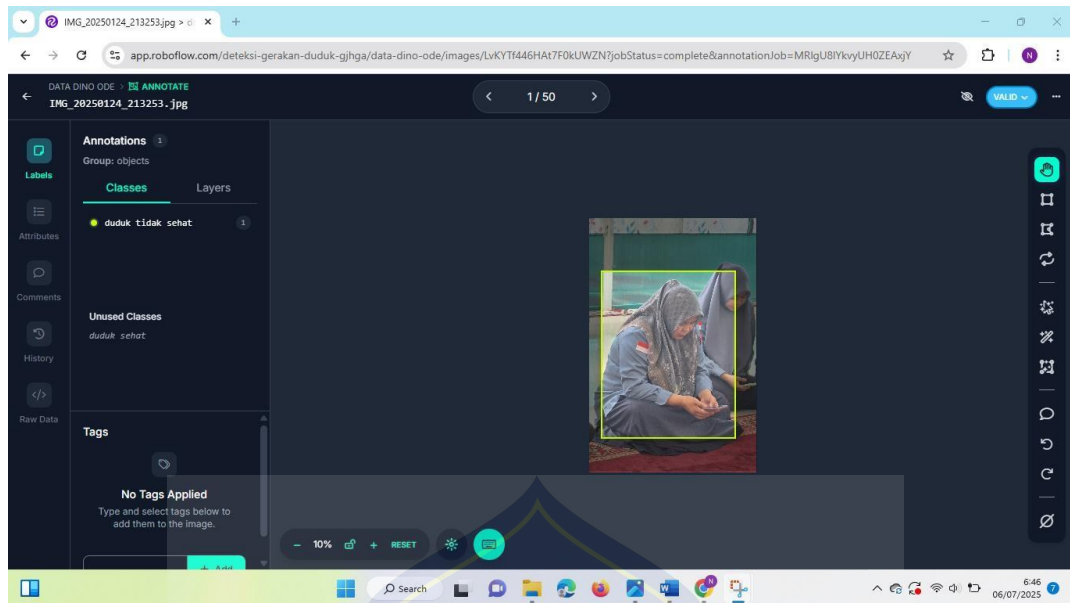


Gambar 8 Class Dataset

Tujuan pembuatan kelas ini adalah untuk mempermudah penentuan dataset gambar yang telah diberi *bonding box* sehingga otomatis akan muncul dua kelas yang telah dibuat dengan demikian dataset akan masuk ke dalam kategori yang sesuai.

## c. Proses pelabelan pembagian kategori gambar

Proses pelabelan gambar menggunakan alat *Roboflow* dilakukan dengan menambahkan *bounding box* atau frame pada objek dalam gambar. Setelah *bounding box* di tambahkan, kelas dengan kategori duduk bekerja dan duduk tidak bekerja



Gambar 9. Pelabelan Dataset

#### a. Pengolahan Gambar

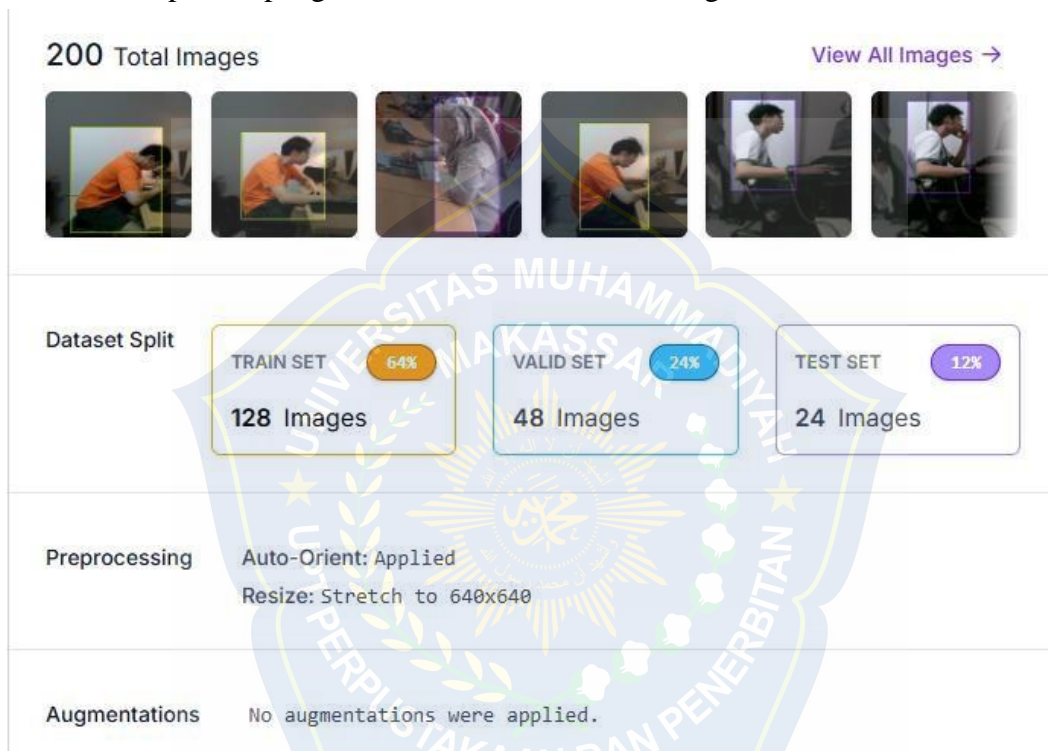
Manajemen citra dengan *Roboflow* menawarkan beragam penyesuaian, termasuk modifikasi warna, bentuk, dan ukuran, serta penerapan teknik *Preprocessing dan Augmentation*. Ukuran citra di augmentasi 640 x 640 piksel. Tujuan dari langkah ini adalah untuk meningkatkan *efektivitas* kinerja GPU saat melakukan pengujian model, yang berpengaruh pada percepatan keseluruhan proses. Pendekatan yang diterapkan dalam fase ini meliputi pengubah ukuran, dengan ukuran gambar yang dituju adalah 640 x 640 piksel. Langkah ini

dirancang untuk memperlancar kinerja GPU ketika melakukan pengujian model, sehingga meningkatkan efektivitas secara keseluruhan

## 1. Training

Gambar 10. Split dataset setelah augmentasi

Setelah proses pengolahan data disesuaikan dengan kebutuhan sistem, data



tersebut kemudian diekspor untuk dimanfaatkan dalam tahap pelatihan model pada platform *Google Colab*. Hasil ekspor berupa dataset ini turut menyertakan API yang diperlukan sebagai antarmuka integrasi, sehingga dapat digunakan secara optimal dalam melatih model YOLOv8 di lingkungan *Google Colab*.

### a. Ekspor Dataset *Reoboflow*



API tersebut bertindak sebagai media penghubung antara dataset dan lingkungan pengembangan, memungkinkan proses pemanggilan data berlangsung secara otomatis dalam tahap pelatihan model YOLOv8 yang diimplementasikan di platform *Google Colab*.

a. Traing *Google Colab*

Proses pelatihan model dilakukan dengan memanfaatkan API yang diperoleh melalui platform *Roboflow*. API tersebut diakses sebagai sarana untuk menjalankan pelatihan model secara langsung di lingkungan *Google Colab*.

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="UQpXn7kipsvsT2bYwNfh")
project = rf.workspace("deteksi-gerakan-duduk-gjhga").project("data-dino-ode")
version = project.version(5)
dataset = version.download("yolov8")
```

Gambar 13 Traing *Google Colab*

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt
data=/content/datasets/datafix-8/data.yaml epochs=50 imgsz=640
plots=True
```

Gambar 14 API dari *Roboflow*

API dari *Roboflow* berhasil, API tersebut digunakan sebagai jalan untuk mengintegrasikan proses pelatihan model

Pelaksanaan program dimulai dengan mengarahkan sistem ke direktori utama pengguna (*HOME*). Setelah itu, model deteksi objek dijalankan menggunakan perintah YOLO, yang dikonfigurasi untuk melakukan tugas pendeteksian objek dalam mode pelatihan. Model awal yang digunakan adalah “yolov8s.pt”, sebuah model pralatih. Dataset yang dimanfaatkan, yaitu “datafix-8”, telah dipersiapkan

lengkap dengan anotasi untuk keperluan pelatihan. Proses pelatihan dijalankan sebanyak 50 epoch dengan resolusi gambar sebesar 640 piksel. Di samping itu, pengaturan `plots = True` diaktifkan untuk membuat grafik yang menampilkan perkembangan hasil selama proses pelatihan.

```
82 epochs completed in 0.066 hours.
Optimizer stripped from data-dino-ode/YOLOv8n-train/weights/last.pt, 6.3MB
Optimizer stripped from data-dino-ode/YOLOv8n-train/weights/best.pt, 6.3MB

Validating data-dino-ode/YOLOv8n-train/weights/best.pt...
Ultralytics 8.3.165 Python-3.11.13 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,006,038 parameters, 0 gradients, 8.1 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	48	48	0.764	0.802	0.89	0.497
duduk sehat	21	21	0.87	0.714	0.871	0.493
duduk tidak sehat	27	27	0.659	0.889	0.909	0.501

```
Speed: 0.2ms preprocess, 2.0ms inference, 0.0ms loss, 2.0ms postprocess per image
Results saved to data-dino-ode/YOLOv8n-train
ultralytics.utils.metrics.DetMetrics object with attributes:

ap_class_index: array([0, 1])
box: ultralytics.utils.metrics.Metric object
confusion_matrix: <ultralytics.utils.metrics.ConfusionMatrix object at 0x7f0d9cf7b350>
curves: ['Precision-Recall(B)', 'F1-Confidence(B)', 'Precision-Confidence(B)', 'Recall-Confidence(B)']
curves_results: [[array([
0.011011, 0.012012, 0.013013, 0.014014, 0.015015, 0.016016, 0.017017, 0.018018, 0.019019, 0.02002, 0.021021,
0.022022, 0.023023,
0.024024, 0.025025, 0.026026, 0.027027, 0.028028, 0.029029, 0.03003, 0.031031, 0.032032, 0.033033, 0.034034,
0.035035, 0.036036, 0.037037, 0.038038, 0.039039, 0.04004, 0.041041, 0.042042, 0.043043, 0.044044, 0.045045, 0.046046,
0.047047,
0.048048, 0.049049, 0.05005, 0.051051, 0.052052, 0.053053, 0.054054, 0.055055, 0.056056, 0.057057, 0.058058,
0.059059, 0.06006, 0.061061, 0.062062, 0.063063, 0.064064, 0.065065, 0.066066, 0.067067, 0.068068, 0.069069, 0.07007,
0.071071,
0.072072, 0.073073, 0.074074, 0.075075, 0.076076, 0.077077, 0.078078, 0.079079, 0.08008, 0.081081, 0.082082,
0.083083, 0.084084, 0.085085, 0.086086, 0.087087, 0.088088, 0.089089, 0.09009, 0.091091, 0.092092, 0.093093, 0.094094,
0.095095,

```

Gambar 15. Hasil Training

Gambar di atas menunjukkan hasil dari proses pelatihan model YOLOv8 yang telah diselesaikan dalam 82 epoch. Berikut adalah beberapa detail penting dari hasil pelatihan tersebut:

1. Detail Model:

- a. Model YOLOv8 yang digunakan berbasis pada *PyTorch* versi 2.6.0 dengan dukungan CUDA untuk GPU Tesla T4, dengan total 3.006.038 parameter dan 8.1 GFLOPs.
- b. Model ini dilatih untuk mendeteksi dua kelas, yaitu duduk sehat dan duduk tidak sehat.

## 2. Status Pelatihan:

Proses pelatihan berlangsung selama 0,066 jam (sekitar 4 menit) , dua versi bobot model disimpan, yaitu last.pt dan best.pt, yang digunakan untuk evaluasi lebih lanjut.

## 3. Hasil Validasi:

Kelas: Model ini melibatkan dua kelas yang disebutkan di atas.

## 4. Metrik Kinerja:

a. Box: Nilai mAP50 (mean average precision pada IoU 50%) adalah 0.802 untuk semua kelas, yang menunjukkan kinerja yang cukup baik dalam mendeteksi objek.

b. Precision dan Recall: Nilai precision (ketepatan) adalah 0.764 dan recall (*sensitivitas*) adalah 0.802, yang menunjukkan model mampu mendeteksi objek dengan baik tanpa terlalu banyak kesalahan false positive atau false negative.

c. mAP50-95: Nilai mAP50-95 (nilai rata-rata presisi pada berbagai ambang batas IoU) adalah 0.497, yang mengindikasikan model memiliki kinerja yang baik meskipun ada ruang untuk perbaikan dalam deteksi objek pada level IoU yang lebih ketat.

5. Kecepatan: Proses preprocessing membutuhkan waktu 0.2ms, *inference* (proses prediksi) 2.0ms, dan postprocessing 2.0ms per gambar. Waktu ini cukup efisien untuk penggunaan dalam aplikasi nyata.

## 6. Metrik Lain:

Hasil pelatihan disimpan di folder `data-dino-ode/YOLOv8n-train` untuk evaluasi lebih lanjut. Model menunjukkan hasil yang baik dalam hal kecepatan dan akurasi, meskipun ada potensi untuk meningkatkan kinerja, terutama pada metrik `mAP50-95`.

Secara keseluruhan, model ini menunjukkan hasil yang baik dalam mendeteksi posisi duduk sehat dan tidak sehat dengan akurasi yang memadai,

```
model.train(  
  data=os.path.join(dataset.location, "data.yaml"),  
  epochs=100,  
  imgsz=640,  
  batch=16,  
  patience=20,  
  project='data-dino-ode',  
  name='YOLOv8n-train',  
  pretrained=True  
)
```

Gambar 16 terminal yang berisi hasil eksekusi kode *Python*

Gambar di atas menunjukkan tampilan dari terminal yang berisi hasil eksekusi kode Python yang digunakan untuk melatih model deteksi objek YOLOv8. Berikut adalah penjelasan rinci dari gambar tersebut:

1. Kode Python: Bagian pertama dari kode (`model.train()`) menunjukkan bahwa model sedang dilatih menggunakan dataset yang telah ditentukan dalam file `data.yaml`. File ini kemungkinan berisi informasi tentang dataset, seperti jalur gambar dan label kelas.

Beberapa parameter dilibatkan dalam pelatihan: `epochs=100`: Menunjukkan jumlah epoch (iterasi pelatihan) yang akan dilakukan, yaitu 100 `imgsz=640`: Menentukan ukuran gambar yang digunakan selama pelatihan, yaitu

640x640 piksel. `batch=16`: Menentukan ukuran batch yang digunakan selama pelatihan, yaitu 16 gambar per `batch`. `patience=20`: Parameter ini mengontrol jumlah epoch tanpa perbaikan dalam kinerja yang dapat diterima sebelum pelatihan dihentikan lebih awal. `pretrained=True`: Menunjukkan bahwa model yang dilatih dimulai dengan bobot yang sudah dilatih sebelumnya

## 2. Eksekusi Terminal:

Terminal menunjukkan penggunaan versi tertentu dari *Ultralytics*, yang merupakan library untuk pelatihan model YOLOv8, dengan versi Python 3.11.3 dan PyTorch 2.6.0 yang berjalan pada GPU Tesla T4. Ada juga peringatan atau notifikasi tentang file font Arial yang sedang diunduh untuk digunakan dalam visualisasi pelatihan.

## 3. Model Summary:

Bagian berikutnya menampilkan rincian dari model yang sedang dilatih, yaitu susunan layer atau blok yang digunakan dalam model YOLOv8. Setiap modul dilengkapi dengan informasi seperti jumlah parameter dan jenis layer. Misalnya, pada layer pertama, ada `ultralytics.nn.modules.conv.Conv` dengan parameter 464 dan ukuran kernel [3, 16, 3, 2], yang menunjukkan lapisan konvolusi pertama dari model.

## 4. Ringkasan Penggunaan GPU.

Beberapa informasi tambahan mengenai penggunaan GPU juga ditampilkan, seperti `cache` yang digunakan dan download font Arial untuk keperluan visualisasi model.

Secara keseluruhan, gambar ini memperlihatkan tahap awal dari pelatihan model YOLOv8, yang mengonfigurasi berbagai parameter pelatihan dan memuat model menggunakan file konfigurasi untuk mendeteksi objek dalam gambar atau video.



Gambar 17 proses deteksi *real time* camera duduk sehat

### **G. Proses Uji Coba Deteksi**

Kebiasaan duduk yang tidak tepat dalam jangka waktu lama dapat menimbulkan berbagai risiko kesehatan, seperti nyeri punggung, skoliosis, dan gangguan postur tubuh. Terutama di era digital saat ini, banyak orang menghabiskan waktu berjam-jam duduk di depan komputer tanpa memperhatikan posisi tubuh mereka. Oleh

karena itu, dibutuhkan sebuah sistem yang mampu mendeteksi gerakan dan postur duduk secara real-time untuk memastikan posisi duduk yang sehat tetap terjaga.

**Tujuan:**

Mengembangkan sistem berbasis visi komputer yang dapat memantau dan mendeteksi gerakan duduk sehat secara real-time menggunakan model deteksi objek YOLOv8, sehingga dapat memberikan notifikasi atau rekomendasi perbaikan postur apabila terdeteksi postur duduk yang tidak sehat



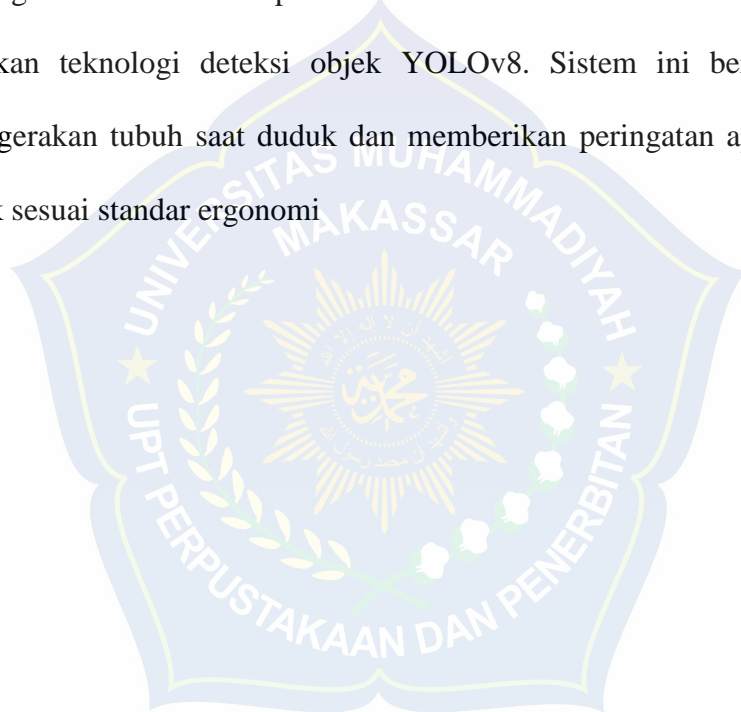
Gambar 18 proses deteksi *real time* camera duduk tidak sehat

Postur duduk yang buruk atau tidak ergonomis merupakan salah satu penyebab utama gangguan muskuloskeletal seperti nyeri leher, sakit punggung, dan gangguan tulang belakang.

Di era kerja dan belajar digital saat ini, banyak individu menghabiskan waktu berjam-jam duduk di depan komputer tanpa menyadari posisi duduknya. Oleh karena itu, dibutuhkan sistem yang mampu mendeteksi secara otomatis dan real-time apakah seseorang sedang duduk dalam posisi yang tidak sehat, agar tindakan koreksi dapat segera dilakukan.

**Tujuan:**

Mengembangkan sistem deteksi postur duduk tidak sehat secara real-time dengan memanfaatkan teknologi deteksi objek YOLOv8. Sistem ini bertugas untuk memantau gerakan tubuh saat duduk dan memberikan peringatan apabila postur duduk tidak sesuai standar ergonomi



## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Monitoring gerakan menggunakan metode algoritma yolov8 dengan mendeteksi duduk sehat dan duduk tidak sehat nilai mAP50 (*mean average precision* pada IoU 50% ) adalah 0,802 untuk semua kelas yang menunjukkan kinerja yang cukup baik dalam mendeteksi objek dengan baik tanpa terlalu banyak kesalahan false positive atau false negative. Model ini menunjukkan hasil yang baik dalam mendeteksi posisi duduk sehat dan tidak sehat dengan akurasi yang baik.

#### **1. Saran**

Berdasarkan hasil yang diperoleh dari studi ini, ada beberapa rekomendasi yang bisa dipertimbangkan sebagai pedoman atau dasar untuk pengembangan penelitian di masa yang akan datang:

1. Mengimplementasikan algoritma terbaru atau menggabungkan beberapa algoritma lain untuk meningkatkan efektivitas dalam deteksi dan klasifikasi.
2. Menciptakan metode deteksi dan klasifikasi yang lebih efisien dengan mengeksplorasi teknik-teknik yang dapat meningkatkan akurasi, terutama dalam kondisi duduk yang kurang ideal.

## DAFTAR PUSTAKA

Aini, Q., Lutfiani, N., Kusumah, H., & Zahran, M. S. (2021). Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo. *CESS (Journal of Computer Engineering, System and Science)*, 6(2), 192. <https://doi.org/10.24114/cess.v6i2.25840>

Astuti, F. A. (2021). *Pemanfaatan Teknologi Artificial Intelligence untuk Penguatan Kesehatan dan Pemulihan Ekonomi Nasional*. 04(01), 25–34.

Eriana, E. S. (2023). *ARTIFICIAL INTELLIGENCE (AI)* (1st ed.). EUREKA MEDIA AKSARA, DESEMBER 2023 ANGGOTA IKAPI JAWA TENGAH NO. 225/JTE/2021.

<https://repository.penerbiteureka.com/publications/567027/artificialintelligence-ai>

Fauzi, Y., Andiono, E., & Khamali, M. (2020). Aplikasi Object Detection and Tracking Untuk Penyandang Tunanetra dengan Internet of Things (IoT) (Menggunakan Bahasa Pemrograman Phyton). *Universitas Budiluhur, Jakarta 1 Jln. Raya Cilegon Serang KM.08 Kramatwatu, 12260*.

Indaryanto, F., Nugroho, A., & Suni, A. F. (2021). Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19. *Edu*

*Komputika Journal*, 8(1), 31–38.

<https://doi.org/10.15294/edukomputika.v8i1.47837>

Kadiva Dwilia Rosadi Putri. (2023). *Difabel Netra dan Dunia Visual*.

<https://sie.telkomuniversity.ac.id/difabel-netra-dan-dunia-visual/>

Liunanda, C. N., Rostianingsih, S., & Purbowo, A. N. (2020). Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android. *Jurnal Infra*, Vol 8, No., 1–7.

Manu, G. A., Masan, P. L., Bangsa, U. C., Nusa, P., Timur, T., & Learning, A.

(2020). *APLIKASI TEXT TO SPEECH UNTUK MENINGKATKAN*

*PEMBELAJARAN BAHASA INGGRIS BAGI SISWA DISABILITAS Gerlan.*

03(02), 17–26.

Nur, M., Muhlashin, I., & Stefanie, A. (2023). *KLASIFIKASI PENYAKIT MATA BERDASARKAN CITRA FUNDUS MENGGUNAKAN YOLO V8*. 7(2), 1363–

1368.

Pendidikan, T., Pendidikan, F. I., & Malang, U. N. (2020). *PENERAPAN BAHAN AJAR AUDIO UNTUK ANAK TUNANETRA TINGKAT SMP DI INDONESIA*

*Agnes Praptaningrum seperangkat tunanetra dan noncetak . Bahan ajar cetak dapat*. 5, 1–19.

Pradana, T., & Sula, E. H. (2023). Pengembangan Aplikasi Pendeteksi Objek Untuk Tunanetra Menggunakan Operator Tepi Canny Dengan Library Opencv

Berbasis Android. *Spirit*, 15(1), 23–27.

<https://doi.org/10.53567/spirit.v15i1.285>

Prisky Ratna Aningtiyas, A. S. dan S. W. (2020). *Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan*

*Memfaatkan SSD MobileNet V2 Sebagai Model Pra-Terlatih. 19*(September), 421–430.

Randy Moh Yusup, Aldof Faris Anugrah, Muslimah, D. D., Permana, S. M. W. N., & Shindi Yuliani. (2024). PENDETEKSIAN OBJEK MENGGUNAKAN

OPENCV DAN METODE YOLOv4-TINY UNTUK MEMBANTU

TUNANETRA. *Journal of Computer Science and Information Technology, 1*(2), 59–68. <https://doi.org/10.59407/jcsit.v1i2.532>

Raup, A., Ridwan, W., Khoeriyah, Y., & Zaqiah, Q. Y. (2022). *Deep Learning dan Penerapannya dalam Pembelajaran. 5*(September), 3258–3267.

Rosaly, R. (2019). *Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan.*

Sarmah, A. J., Bhagawati, K., Duwarah, K., & Purkayastha, S. D. (2023). *Object detection and conversion of text to speech for visually impaired. September.*

Tamang, S., Sen, B., Pradhan, A., Sharma, K., & Singh, V. K. (2023).

*INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING*

*Enhancing COVID-19 Safety : Exploring YOLOv8 Object Detection for Accurate Face Mask Classification.*

Yudha, Y. F., Aditya, A. A. Y., Rasyid, R. A. Y., Indra, N. I. A., & Melati, M. W.

W. (2024). Perancangan Sistem Deteksi Objek Pada Robot Transporter Menggunakan Metode Darknet YOLOv8. *Electrician : Jurnal Rekayasa Dan Teknologi Elektro, 18*(2), 161–170. <https://doi.org/10.23960/elc.v18n2.2595>

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## SURAT KETERANGAN BEBAS PLAGIAT

Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,  
mengatakan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Dino Ode Kambisa

Nim : 105841104219

Program Studi : Teknik Informatika

Nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	10%	10 %
2	Bab 2	15%	25 %
3	Bab 3	3%	10 %
4	Bab 4	7%	10 %
5	Bab 5	0%	5 %

Penyakit telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan  
Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan  
sebagaimana mestinya.

Makassar, 14 Agustus 2025

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,



Nursulita, S.Hum., M.I.P

NBM. 964.591