

***IMAGE SEGMENTATION* TUBUH MANUSIA MENGGUNAKAN  
*METODE INSTANCE SEGMENTATION***

**SKRIPSI**

**AMRU KHAIR AL HAKIM  
201401002**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

***IMAGE SEGMENTATION TUBUH MANUSIA MENGGUNAKAN  
METODE INSTANCE SEGMENTATION***

**SKRIPSI**

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Ilmu Komputer**

**AMRU KHAIR AL HAKIM**

**201401002**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN**

**PERSETUJUAN**

Judul : *IMAGE SEGMENTATION* TUBUH  
MANUSIA MENGGUNAKAN METODE  
*INSTANCE SEGMENTATION*  
Kategori : SKRIPSI  
Nama : AMRU KHAIR AL HAKIM  
Nomor Induk Mahasiswa : 201401002  
Program Studi : SARJANA (S1) ILMU KOMPUTER  
Fakultas : ILMU KOMPUTER DAN TEKNOLOGI  
INFORMASI UNIVERSITAS SUMATERA  
UTARA

Medan, 12 Juni 2024

Komisi Pembimbing:

Pembimbing II



Dr. Ir. Elviawaty Muisa Zamzami S.T., M.T., MM. IPU  
NIP. 197007162005012002

Pembimbing I



Anandhini Medianty Nababan S.Kom., M.T.  
NIP. 199304132021022001

Diketahui/disetujui oleh

Program Studi S1 Ilmu Komputer

Ketua,  
  
Dr. Amalia ST., M.T  
NIP. 197812212014042001

**PERNYATAAN*****IMAGE SEGMENTATION TUBUH MANUSIA MENGGUNAKAN  
METODE INSTANCE SEGMENTATION*****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 10 Juni 2024



Amru Khair Al Hakim

201401002

## PENGHARGAAN

Tiada kata yang pantas terucap selain kata syukur kepada Allah Swt, berkat limpahan dan rahmat-Nya penulis mampu menyelesaikan skripsi ini sebagai syarat gelar Sarjana Komputer, pada Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Shalawat serta salam tak lupa penulis haturkan kepada junjungan nan mulia Nabi Muhammad SAW yang mana berkat rahmat Beliau kita mampu merasakan dunia yang penuh dengan ilmu pengetahuan ini.

Penulis ingin mengungkapkan penghargaan dan terima kasih yang besar kepada:

1. Orang tua penulis yang teristimewa, Abah dan Emak yang telah memberikan dukungan, semangat dan doa kepada penulis.
2. Keluarga terdekat penulis, Umi dan Bunda yang telah memberikan dukungan dan nasehatnya kepada penulis dalam masa pendidikan serta adik kandung penulis, Imam Syahid Al Hakim.
3. Bapak Prof. Dr. Muryanto Amin, S.Sos, M.Si. selaku Rektor Universitas Sumatera Utara.
4. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
5. Ibu Dr. Amalia, ST. MT., selaku Kepala Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
6. Ibu Sri Melvani Hardi S.Kom., M.Kom., sebagai Sekretaris Program Studi S1 Ilmu Komputer Universitas Sumatera Utara
7. Ibu Hayatunnufus S.Kom, M.Cs, selaku Dosen Pembimbing Akademik, yang telah memberikan arahan, nasihat, dan dukungan kepada penulis sebagai mahasiswa dalam menempuh berbagai tahapan akademik.
8. Ibu Anandhini Medianty Nababan S.KOM., M.T, selaku dosen pembimbing I yang telah memberikan bimbingan, kritik, saran, serta dukungan semangat kepada penulis dalam menyelesaikan skripsi ini.

9. Ibu Dr. Ir. Elviawaty Muisa Zamzami S.T., M.T., MM., IPU, selaku dosen pembimbing II yang telah memberikan bimbingan, arahan, dukungan serta masukan kepada penulis.
10. Bapak Pauzi Ibrahim Nainggolan S.Komp., M.Sc., selaku Dosen Penguji yang telah memberikan saran dan masukan yang membangun dalam menyempurnakan penulisan skripsi ini.
11. Seluruh Dosen Program Studi S-1 Ilmu Komputer yang telah meluangkan waktu dan energi untuk mengajar dan membimbing, sehingga penulis dapat mencapai tahap penyusunan skripsi ini.
12. Teman seperjuangan terbaik, Sabila Tamimi Putri. Terimakasih telah berkontribusi dalam penelitian, telah mendukung, mendengarkan keluh kesah dan memberikan semangat untuk penulis dalam pendidikan.
13. Orang terdekat, sahabat, serta teman penulis yaitu atas nama Akbar, Putra, Jaya, Ja'far, Habib, Raihan, Uga, Kevin, Dinda, Sonia, Siti Jubaidah, Regina, dan teman-teman lainnya yang belum dapat disebutkan oleh penulis.
14. Teman-teman mahasiswa/i program studi S-1 Ilmu Komputer Stambuk 2020, khususnya KOM – A .
15. Dan semua pihak yang terlibat, baik secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu.

Medan, 10 Juni 2024



Amru Khair Al Hakim

201401002

## ***IMAGE SEGMENTATION TUBUH MANUSIA MENGGUNAKAN METODE INSTANCE SEGMENTATION***

### **ABSTRAK**

*Image Segmentation* adalah langkah awal dalam pemrosesan citra digital untuk memisahkan objek penting dari latar belakang, mempermudah analisis, dan pengenalan lebih lanjut. Dalam citra tubuh manusia, ini krusial untuk mengenali dan menganalisis bagian tubuh seperti tangan, badan, atau kepala, yang memiliki keunikan untuk deteksi dan pengenalan. Penelitian ini bertujuan untuk membangun sebuah sistem berbasis web untuk menguraikan bagian tubuh manusia kedalam segmen-segmen yang jelas, seperti kepala, badan, lengan kanan dan lengan kiri. Penelitian ini melibatkan tahapan studi pustaka untuk memahami konsep *instance segmentation* dan YOLOv8, pengumpulan dataset, pelabelan, *training*, evaluasi dan implementasi web. Dari hasil pengujian yang dilakukan untuk mengevaluasi performa terhadap model didapatkan nilai mAP tertinggi yaitu sebesar 0.979 dengan jumlah 200 *epoch* pada *training* dataset, 0.914 nilai *precision*, 0.995 nilai *recall* dan 0.95 F1-score. Kesimpulan dari penelitian ini, menunjukkan bahwa *instance segmentation* dengan model YOLOv8 dapat memberikan hasil segmentasi pada bagian tubuh manusia.

**Kata kunci:** *Image Segmentation, Instance Segmentation, You Only Look Once Version 8*

## **HUMAN BODY IMAGE SEGMENTATION USING INSTANCE SEGMENTATION METHOD**

### **ABSTRACT**

Image Segmentation is the initial step in digital image processing that separates important objects from the background, facilitating further analysis and recognition. In human body imagery, it is crucial for identifying and analyzing body parts such as hands, body, or head, which have unique characteristics for detection and recognition. This research aims to develop a web-based system to delineate human body parts into clear segments, such as head, body, right arm, and left arm. The study involves stages of literature review to understand the concepts of instance segmentation and YOLOv8, dataset collection, labeling, training, evaluation, and web implementation. The testing results for evaluating the model's performance showed the highest mAP value of 0.979 with 200 epochs on the training dataset, 0.914 precision, 0.995 recall, and 0.95 F1-score. The conclusion of this study indicates that instance segmentation with the YOLOv8 model can provide effective segmentation of human body parts.

**Keyword:** Image Segmentation, Instance Segmentation, You Only Look Once  
Version 8



## DAFTAR ISI

PERSETUJUAN .....	iii
PERNYATAAN.....	iv
PENGHARGAAN .....	v
ABSTRAK .....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xii
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	3
1.3    Batasan Masalah.....	3
1.4    Tujuan Penelitian.....	4
1.5    Manfaat Penelitian.....	4
1.6    Metodologi Penelitian .....	4
1.7    Penelitian Relevan .....	5
1.8    Sistematika Penulisan.....	7
BAB II LANDASAN TEORI .....	9
2.1    Tubuh Manusia.....	9
2.2 <i>Image Segmentation</i> .....	9
2.3 <i>Instance Segmentation</i> .....	10
2.4 <i>You Only Look Once (YOLO)</i> .....	11
BAB III ANALISIS DAN PERANCANGAN .....	14
3.1    Analisis Sistem .....	14
3.1.1    Analisis Masalah .....	14
3.1.2    Analisis Kebutuhan .....	14
3.1.3    Arsitektur Sistem.....	15
3.1.4    Arsitektur Umum .....	16
3.2    Pemodelan Sistem .....	17
3.2.1 <i>Use Case Diagram</i> .....	17
3.2.2 <i>Activity Diagram</i> .....	18

3.2.3	<i>Sequence Diagram</i> .....	19
3.3	<i>Flowchart</i> .....	20
3.3.1	<i>Flowchart Sistem</i> .....	21
3.3.2	<i>Flowchart YOLO</i> .....	22
3.4	<i>Perancangan Interface</i> .....	22
3.4.1	Halaman Beranda .....	23
3.4.2	Halaman Hasil .....	24
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....		25
4.1	Implementasi Tahap Persiapan Dataset.....	26
4.1.1	Pengumpulan Dataset.....	26
4.1.2	Pelabelan Data.....	27
4.1.3	<i>Preprocessing</i> , Augmentasi, dan <i>Split</i> Dataset .....	28
4.2	Training Dataset .....	28
4.2.1	Mengunduh Library Ultralytics.....	29
4.2.2	Import YOLO.....	29
4.2.3	Inisialisasi Model YOLO untuk <i>Instance Segmentation</i> .....	29
4.2.4	Proses Mengakses Dataset .....	30
4.2.5	Proses Mendefinisikan Jumlah <i>Class</i> pada Dataset .....	31
4.2.6	Proses <i>Training</i> Dataset .....	31
4.3	Tuning Hyperparameter dan Pengujian Model .....	32
4.4	Implementasi Model kedalam Web.....	38
4.5	Pengujian Sistem .....	40
4.6	Hasil Pengujian Sistem.....	42
BAB 5 KESIMPULAN DAN SARAN .....		47
5.1	Kesimpulan.....	47
5.2	Saran .....	47
DAFTAR PUSTAKA .....		49

## DAFTAR TABEL

<b>Tabel 4.1</b> Training Epoch 100, Learning Rate 0.01 .....	32
<b>Tabel 4.2</b> Summary Training Epoch 100, Learning Rate 0.01.....	32
<b>Tabel 4.3</b> Training Epoch 150, Learning Rate 0.01 .....	33
<b>Tabel 4.4</b> Summary Training Epoch 150, Learning Rate 0.01.....	33
<b>Tabel 4.5</b> Training Epoch 200, Learning Rate 0.01 .....	34
<b>Tabel 4.6</b> Summary Training Epoch 200, Learning Rate 0.01.....	34
<b>Tabel 4.7</b> Training Epoch 250, Learning Rate 0.01 .....	35
<b>Tabel 4.8</b> Summary Training Epoch 250, Learning Rate 0.01.....	35
<b>Tabel 4.9</b> Hasil Pengujian Sistem.....	42

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Semantik Segmentation dan <i>Instance Segmentation</i> .....	10
<b>Gambar 2.2</b> Instance Segmentation .....	11
<b>Gambar 2.3</b> Arsitektur YOLO .....	12
<b>Gambar 2.4</b> Rumus IoU .....	12
<b>Gambar 3.1</b> Arsitektur Sistem .....	16
<b>Gambar 3.2</b> Arsitektur Umum .....	16
<b>Gambar 3.3</b> Use Case Diagram .....	18
<b>Gambar 3.4</b> Activity Diagram .....	19
<b>Gambar 3.5</b> Sequence Diagram .....	20
<b>Gambar 3.6</b> Flowchart Sistem .....	21
<b>Gambar 3.7</b> Flowchart YOLO .....	22
<b>Gambar 3.8</b> Desain Halaman Beranda .....	23
<b>Gambar 3.9</b> Desain Halaman Hasil .....	24
<b>Gambar 4.1</b> Implementasi Sistem.....	25
<b>Gambar 4.2</b> Proses Persiapan Dataset .....	26
<b>Gambar 4.3</b> Contoh Dataset.....	26
<b>Gambar 4.4</b> Proses Pelabelan Dataset .....	27
<b>Gambar 4.5</b> Pembagian Dataset.....	28
<b>Gambar 4.6</b> Kode untuk Mengunduh Library Ultralytics .....	29
<b>Gambar 4.7</b> Kode untuk Import YOLO .....	29
<b>Gambar 4.8</b> Kode untuk menginisialisasi Model YOLO .....	29
<b>Gambar 4.9</b> Kode untuk Mengakses Dataset .....	30
<b>Gambar 4.10</b> File data.yaml .....	30
<b>Gambar 4.11</b> Kode untuk Mendefinisikan Jumlah Class pada Data set.....	31
<b>Gambar 4.12</b> Kode untuk Training Dataset.....	31
<b>Gambar 4.13</b> Grafik Performa Model.....	36
<b>Gambar 4.14</b> Pengujian Model pada Google Colab (1) .....	37
<b>Gambar 4.15</b> Pengujian Model pada Google Colab (2) .....	37
<b>Gambar 4.16</b> Pengujian Model pada Google Colab (3) .....	38
<b>Gambar 4.17</b> Tampilan Utama (Beranda) .....	39

<b>Gambar 4.18</b> Tampilan Hasil.....	39
<b>Gambar 4.19</b> Hasil Uji Sistem Data No. 1 .....	40
<b>Gambar 4.20</b> Hasil Uji Sistem Data No. 11 .....	41
<b>Gambar 4.21</b> Hasil Uji Sistem Data No. 22 .....	42

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Inovasi teknologi kesehatan yang memanfaatkan kecerdasan buatan semakin berkembang (Komalasari, 2022; Oktaviani, 2023). Perhatian terhadap perawatan kesehatan berbasis data cerdas telah meningkat pesat dalam beberapa tahun terakhir (Komalasari, 2022; Rundo et al., 2020). Hampir seluruh aspek kehidupan manusia berkaitan erat dengan komputer, didukung oleh pertumbuhan yang cepat dalam teknologi digital, yang memungkinkan komputer melakukan komputasi yang semakin canggih. Salah satu contoh nyata dari perkembangan ini adalah pengembangan *Artificial Intelligence* (AI). Salah satu perkembangan dalam *Artificial Intelligence* terlihat dalam bidang *computer vision*.

*Computer vision* adalah bidang pengetahuan yang bertujuan untuk meningkatkan kemampuan komputer dalam menggambarkan informasi dari gambar digital yang diamati. Di dalam bidang *computer vision*, terdapat beberapa sub-bidang seperti *image classification*, *image segmentation*, dan *object detection* (Arifianto, 2022). *Image Segmentation* merupakan salah satu langkah awal dalam pemrosesan citra digital yang bertujuan untuk memisahkan objek-objek penting dari latar belakangnya, sehingga mempermudah analisis dan pengenalan lebih lanjut terhadap setiap bagian dalam gambar (Helnawan et al., 2023). Dalam konteks *image segmentation* bagian tubuh manusia, langkah ini menjadi krusial untuk memungkinkan pengenalan dan analisis yang lebih mendalam terhadap berbagai bagian tubuh, seperti tangan, badan, atau kepala.

Bagian tubuh manusia berlimpah keunikan yang mana masing-masing bagian tersebut memiliki ciri sendiri. Keunikan pada bagian tubuh tersebut dapat digunakan dalam bidang ilmu komputer sebagai sarana untuk melakukan deteksi dan pengenalan terhadap potongan tubuh manusia (Heryana et al., 2020). Segmentasi tubuh manusia memiliki tantangan tersendiri, karena kompleksitas bentuk dan tubuh yang bervariasi. Oleh karena itu, diperlukan metode yang mampu melakukan segmentasi secara akurat dan efisien. Dalam *Image segmentation* terbagi menjadi dua, yaitu *semantic segmentation* dan *instance segmentation*

(Helnawan et al., 2023). *Instance segmentation* adalah cabang dari bidang *computer vision* yang mengkombinasikan deteksi objek dan segmentasi semantik. Dengan kata lain, *instance segmentation* melibatkan proses pemberian label, prediksi lokasi, dan pembuatan masker segmentasi piksel-piksel untuk setiap objek yang terdapat dalam citra visual (Kristal & Harintaka, 2022; Viana et al., 2020).

*Instance segmentation* merupakan metode segmentasi gambar yang tidak hanya mengidentifikasi batas-batas objek tetapi juga membedakan antara instansi (*instances*) dari objek yang sama. Hal ini sangat penting dalam analisis tubuh manusia, karena memungkinkan identifikasi dan analisis setiap bagian tubuh individu secara lebih detail. *Instance segmentation* berguna untuk mengukur ukuran objek yang terdeteksi, memotongnya dari latar belakangnya, dan lebih akurat mendeteksi objek. Dalam penelitian ini penguraian bagian tubuh manusia bermaksud untuk memecah citra manusia menjadi segmen-segmen yang lebih rinci, seperti kepala, lengan dan badan. Penguraian tubuh manusia (*human parsing*) merupakan tugas pemahaman visual yang mendasar, yang memerlukan *image segmentation* manusia ke dalam bagian-bagian tubuh yang jelas. Penelitian terkait *human parsing* sering dimanfaatkan dalam industri *fashion* untuk analisis gaya (He et al., 2021).

Salah satu teknologi terbaru yang digunakan dalam segmentasi tubuh manusia adalah *You Only Look Once* (YOLO). YOLO adalah algoritma yang intensif dalam komputasi agar mampu memberikan akurasi yang tinggi (Abdul Hadi et al., 2021). YOLO menetapkan arsitektur tunggal yang terpadu untuk membagi *image* menjadi kotak pembatas dan menghitung probabilitas kelas untuk setiap kotak, jika dibandingkan dengan pendekatan identifikasi objek yang ada sebelumnya, seperti R-CNN. Sebagai hasilnya, YOLO mampu dieksekusi dengan jauh lebih cepat dan dengan presisi yang lebih tinggi. Algoritma ini juga dapat dengan tepat memprediksi *image* atau ilustrasi (Mohanapriya et al., 2023). Metode YOLOv8 akan diimplementasikan dalam penelitian ini untuk membagi citra manusia menjadi bagian-bagian yang lebih rinci. YOLOv8 menyediakan kemampuan deteksi objek dengan tingkat kecepatan, akurasi, dan efisiensi yang lebih tinggi daripada versi sebelumnya (Drantantiyas et al., 2023; Motwani & S, 2023; Wang et al., 2023).

Algoritma YOLOv8 pada penelitian ini bertujuan untuk mendeteksi objek karena kecepatan dan akurasi yang tinggi dan kemudian menggunakan metode *instance segmentation* untuk memisahkan objek individual menjadi beberapa segmen. Dengan terlebih dahulu mendeteksi objek menggunakan algoritma deteksi objek dan kemudian menyempurnakan segmentasi menggunakan klasifikasi tingkat piksel, metode ini diharapkan dapat mencapai segmentasi yang akurat dan detail. Hal ini memanfaatkan kekuatan kedua tugas tersebut, meningkatkan kinerja keseluruhan dan menghasilkan masker objek berkualitas tinggi. Dari penjelasan di atas, penelitian ini akan menghasilkan sebuah sistem berbasis web untuk menguraikan bagian tubuh manusia ke dalam segmen-segmen yang jelas, seperti kepala, badan, lengan kanan dan lengan kiri.

## 1.2 Rumusan Masalah

*Image Segmentation* digunakan untuk memisahkan objek penting dari latar belakang untuk mempermudah analisis dan pengenalan citra. *Image segmentation* tubuh manusia penting untuk mengenali dan menganalisis bagian seperti badan, lengan, dan kepala. Namun, segmentasi tubuh manusia menghadapi berbagai tantangan seperti orientasi, kondisi pencahayaan, dan latar belakang yang kompleks. Untuk mengatasi hal ini, dibangun sebuah sistem segmentasi bagian tubuh manusia berbasis web yang akurat, efisien, dan detail dengan menggabungkan algoritma deteksi objek YOLOv8 dan metode *instance segmentation*. Penelitian ini diharapkan dapat menghadapi tantangan dan dapat mensegmentasikan bagian tubuh manusia dengan akurat.

## 1.3 Batasan Masalah

Pembatasan pada masalah dalam penelitian ini dirinci sebagai berikut:

1. Bahasa pemrograman menggunakan Python.
2. Batasan usia dalam penelitian ini adalah wanita usia subur (15-49 tahun)
3. Gambar menampilkan pose tangan kiri terangkat, bebas dari gangguan pakaian
4. Gambar yang dipergunakan dalam penelitian ini harus memiliki pencahayaan yang cukup terang dan kualitas yang baik.



5. Dataset diperoleh dari foto mahasiswi Fakultas Kesehatan Masyarakat Universitas Sumatera Utara yang bersedia berpartisipasi dalam penelitian ini.
6. Jumlah kelas atau bagian tubuh manusia yang akan diidentifikasi oleh model adalah bagian kepala, badan, lengan kanan dan lengan kiri. Dengan membatasi jumlah kelas diharapkan dapat membantu meningkatkan akurasi dan interpretabilitas model.
7. *Output image* hasil segmentasi.

#### **1.4 Tujuan Penelitian**

Penelitian ini bertujuan untuk mengkaji bagaimana metode *instance segmentation* dapat diterapkan untuk segmentasi tubuh manusia dalam gambar digital, mengingat tantangan-tantangan seperti orientasi, kondisi pencahayaan, dan latar belakang yang kompleks. Penelitian ini bertujuan untuk membangun sebuah sistem segmentasi tubuh manusia berbasis web yang menggabungkan YOLOv8 dan *instance segmentation*. Sistem ini diharapkan dapat membantu berbagai aplikasi yang membutuhkan segmentasi tubuh manusia seperti pengenalan bagian tubuh atau pemantauan medis.

#### **1.5 Manfaat Penelitian**

Berikut manfaat yang dapat diperoleh dari penelitian ini:

1. Sistem segmentasi tubuh manusia yang dihasilkan diharapkan dapat memberikan kontribusi dalam berbagai aplikasi yang membutuhkan segmentasi tubuh manusia seperti pengenalan bagian tubuh atau pemantauan medis.
2. Dengan memberikan solusi terhadap tantangan dalam segmentasi tubuh manusia, diharapkan penelitian ini dapat membuka potensi penggunaan teknologi yang lebih luas di berbagai bidang.
3. Temuan dan pengalaman dari penelitian ini dapat menjadi pijakan bagi penelitian selanjutnya.

#### **1.6 Metodologi Penelitian**

Berikut adalah tahapan dari metode penelitian yang akan dilakukan, yaitu:

1. Studi Pustaka

Studi pustaka digunakan untuk mengumpulkan informasi dan dasar teori melalui literatur yang relevan baik jurnal, buku, dan sumber data dari internet yang mendukung pembuatan sistem ini.

2. Pengumpulan Data Set

Mengumpulkan data *image* berformat png dan jpg. Data yang digunakan berupa foto mahasiswi Fakultas Kesehatan Masyarakat Universitas Sumatera Utara.

3. Pelabelan Data

Berikan label pada dataset untuk menunjukkan objek atau bagian tubuh yang akan disegmentasi.

4. *Training*

Setelah dilakukan pelabelan, data set akan dilatih untuk mendapatkan model segmentasi yang diinginkan.

5. Evaluasi

Pada tahap evaluasi model, performa YOLOv8 dalam *instance segmentation* akan dinilai menggunakan metrik seperti akurasi, recall, dan presisi. Langkah ini bertujuan untuk memastikan bahwa model dapat menghasilkan identifikasi yang akurat dan efisien.

6. Implementasi Web

Model yang telah dievaluasi kemudian diimplementasikan kedalam sebuah aplikasi berbasis web.

7. Penyusunan Laporan

Menyusun laporan akhir penelitian yang mencakup permasalahan, metode, hasil, dan analisis dalam bentuk skripsi.

## 1.7 Penelitian Relevan

Berikut adalah penelitian-penelitian terdahulu yang relevan, yaitu:

1. Penelitian oleh (Hassanudin et al., 2024) yang berjudul “*Fine-Grained Analysis of Coral Instance Segmentation using YOLOv8 Models*” Penelitian ini menunjukkan keefektifan model YOLOv8 untuk segmentasi terumbu karang secara real-time. Hasilnya menunjukkan bahwa YOLOv8 berhasil mendeteksi dan mensegmentasi berbagai spesies karang dengan tingkat

akurasi yang tinggi (96,7% presisi, 95,9% *recall*, 98,2% mAP@50) menggunakan model YOLOv8m-Pp dengan pra-pemrosesan Histogram Equalization. YOLOv8 secara efektif mampu mensegmentasi berbagai spesies karang.

2. Penelitian oleh (Sampurno et al., 2024) dengan judul "*Intrarow Uncut Weed Detection Using You-Only-Look-Once Instance Segmentation for Orchard Plantations*" bertujuan untuk mengembangkan modul visi menggunakan kumpulan data yang dilatih khusus pada algoritma *instance segmentation* YOLO untuk mendukung robot penyiang otonom dalam mengenali gulma yang belum ditebang dan hambatan. Model yang digunakan adalah YOLOv5n-seg, YOLOv5s-seg, YOLOv8n-seg, dan YOLOv8s-seg. Pada hasilnya, model YOLOv8n-seg terbukti memiliki akurasi segmentasi yang lebih baik dibandingkan YOLOv5n-seg, sekaligus memiliki kecepatan inferensi tercepat. Ini menjadikannya pilihan yang efektif untuk modul visi dalam aplikasi robot penyiang otonom. Dengan demikian, algoritma *instance segmentation* YOLO terbukti efektif untuk aplikasi robot penyiang otonom dalam lingkungan kebun buah-buahan.
3. Penelitian oleh (Lee et al., 2022) berjudul "*Deep Learning-Based Human Body Posture Recognition and Tracking for Unmanned Aerial Vehicles*". Para peneliti menggunakan model YOLOv3 untuk mendeteksi postur manusia, dengan menggunakan data dari OpenPose. *Framework* YOLOv3 diterapkan untuk deteksi tubuh manusia, menghasilkan kotak pembatas yang menunjukkan posisi tubuh. Penelitian ini menunjukkan hasil yang mengesankan, dengan akurasi pengenalan postur mencapai 95,2% sambil tetap mempertahankan kinerja *real-time*. Hasil ini menegaskan bahwa algoritma YOLOv3 efektif dalam tugas pengawasan dan pengenalan postur manusia. Selain itu, sistem yang dibangun berdasarkan pendekatan ini terbukti efisien dan mampu memberikan kinerja *real-time* yang tinggi. Kesimpulan dari penelitian ini mengindikasikan potensi besar penggunaan drone dengan pembelajaran mendalam untuk aplikasi pengawasan, pencarian, dan penyelamatan.

4. Penelitian oleh (Hurtik et al., 2020) berjudul “*Poly-YOLO: Higher Speed, More Precise Detection And Instance Segmentation For YOLOv3*” menghasilkan sebuah temuan baru dengan mengembangkan YOLOv3 menjadi Poly-YOLO. Dibandingkan dengan YOLOv3, Poly-YOLO hanya memiliki 60% parameter, namun mampu meningkatkan akurasi mAP hingga 40% secara relatif. Untuk tugas *instance segmentation*, tim peneliti merancang ekstensi yang memungkinkan deteksi poligon pembatas dengan jumlah simpul yang dinamis untuk setiap objek yang terdeteksi. Pendekatan ini memungkinkan deteksi poligon pembatas yang dapat belajar bentuk yang tidak bergantung pada ukuran, sehingga menyederhanakan proses deteksi. Poly-YOLO dapat dijalankan secara *real-time* pada kartu grafis kelas menengah.

## 1.8 Sistematika Penulisan

Struktur skripsi ini terdiri dari lima bab yang mencakup:

### **BAB I PENDAHULUAN**

Bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, penelitian relevan dan sistematika penulisan skripsi.

### **BAB II LANDASAN TEORI**

Landasan teori yang berisi tentang pembahasan pengertian Tubuh Manusia, *Image Segmentation*, *Instance Segmentation*, *You Only Look Once* (YOLO).

### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini mengulas tentang kebutuhan dalam sistem. yang sedang dirancang, serta langkah-langkah merancang sebuah sistem dengan menggunakan *instance segmentation* YOLOv8.

### **BAB IV IMPLEMENTASI DAN PENGUJIAN**

Bab ini memuat hasil implementasi dari analisis dan perancangan yang telah dijalankan, serta hasil pengujian. Bagian ini juga menunjukkan hasil uji coba dan analisis yang diperoleh dari studi yang dilakukan.

## **BAB V            KESIMPULAN DAN SARAN**

Bab ini memaparkan ringkasan hasil pengujian dari penelitian yang dilaksanakan dan saran untuk mengembangkan sistem.

## **BAB II**

### **LANDASAN TEORI**

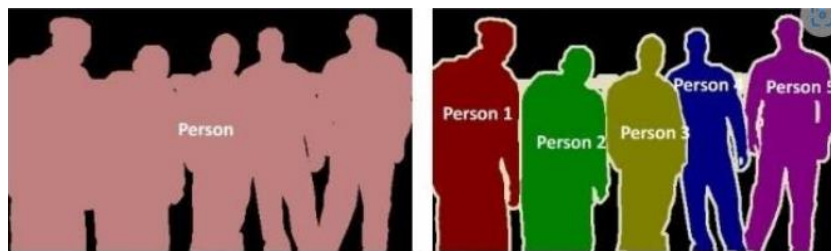
#### **2.1 Tubuh Manusia**

Tubuh manusia terbagi menjadi dua bagian utama yaitu batang tubuh (*truncus*) dan anggota tubuh atas dan bawah. Batang tubuh terdiri dari kepala, leher, dan torso. Torso sendiri terdiri dari dada, perut, dan pinggang. Anggota tubuh atas terhubung dengan batang tubuh melalui gelang bahu, yang terdiri dari *clavicula* (tulang selangka) dan *scapula* (tulang belikat), yang bergerak bersama batang tubuh. Gelang panggul terdiri dari dua tulang panggul dan *sacrum* (tulang belakang bawah), menghubungkan anggota tubuh bawah dengan batang tubuh (Sudibjo et al., 2011). Tubuh manusia merupakan keseluruhan struktur fisik organisme manusia. Terdiri atas kepala, leher, batang badan, dua lengan, dan dua kaki. Kepala adalah bagian atas tubuh yang berisi otak dan fitur-fitur seperti mata, telinga, hidung, dan mulut. Leher menghubungkan kepala dengan batang badan. Batang badan mencakup dada dan perut, tempat berbagai organ vital seperti jantung, paru-paru, dan lambung berada. Dua lengan terletak di sisi tubuh dan berfungsi untuk melakukan berbagai gerakan dan tugas, sementara dua kaki memberikan dukungan dan memungkinkan pergerakan. Struktur tubuh manusia sangatlah kompleks dan beragam, dengan setiap bagian memiliki peran yang penting dalam menjaga kesehatan dan keseimbangan tubuh secara keseluruhan.

#### **2.2 *Image Segmentation***

*Image segmentation* adalah suatu langkah untuk membagi sebuah objek dari latar belakang, sehingga objek tersebut dapat diproses untuk keperluan lain (Mawarni et al., 2023). Segmentasi gambar adalah teknik untuk membagi atau memisahkan citra menjadi beberapa daerah (*region*) berdasarkan kemiripan atribut yang dimilikinya. Proses ini penting dalam pengolahan citra digital karena memungkinkan analisis yang lebih mendalam terhadap struktur dan informasi yang terkandung dalam citra. Segmentasi membantu dalam identifikasi objek, ekstraksi fitur, dan analisis konten visual secara lebih terperinci. Ini bukanlah proses sekali jalan, melainkan merupakan langkah penting dalam rangkaian proses pengolahan citra digital (Pratama et al., 2022).

Dalam konteks citra digital daerah hasil segmentasi merupakan kelompok piksel yang bertetangga atau berhubungan (Panjaitan, 2020). *Image segmentation* akan melewati tahap dimana gambar digital dibagi atau dipartisi menjadi beberapa segmen atau bagian yang lebih kecil. Sasaran utama dari proses segmentasi gambar adalah untuk mengenali dan memisahkan objek atau area tertentu dalam gambar tersebut. Secara sederhana, segmen atau bagian yang dihasilkan dari proses ini mencakup objek atau wilayah dengan ciri visual yang seragam atau serupa. Terdapat dua jenis utama dari segmentasi citra yaitu *instance segmentation* dan *semantic segmentation* (Arnita et al., 2022).



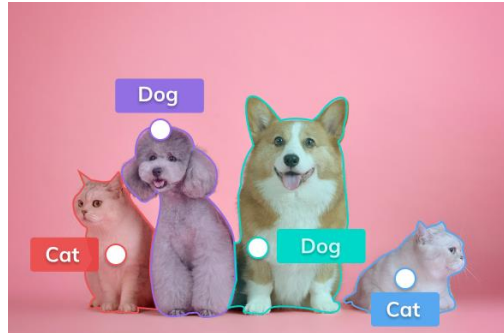
**Gambar 2.1** Semantik *Segmentation* dan *Instance Segmentation*

### 2.3 *Instance Segmentation*

Segmentasi ini terkait dengan identifikasi yang tepat dari semua objek yang terdapat dalam satu gambar. Oleh karena itu, menggabungkan deteksi objek, penentuan lokasi objek, dan klasifikasi objek menjadi elemen kunci dalam segmentasi instan. Dengan kata lain, pendekatan segmentasi ini difokuskan pada tujuan untuk membedakan dengan jelas antara setiap objek yang dikategorikan sebagai *instance* yang serupa. Tujuan dari segmentasi *instance* adalah untuk mendeteksi dan mensegmentasi setiap *instance* objek dalam suatu gambar (Cai et al., 2020). Penggunaan metode *Instance Segmentation* diharapkan dapat meningkatkan akurasi dan efisiensi, serta mengurangi kesalahan yang mungkin terjadi akibat kompleksitas bentuk tubuh manusia.

Segmentasi *instance* telah menjadi fokus penelitian yang sangat signifikan dalam bidang visi mesin, terutama dalam konteks aplikasi yang kompleks seperti robotika, kendaraan otonom, dan pengawasan. Beberapa kerangka kerja segmentasi *instance* telah diusulkan, dan kebanyakan dari mereka memanfaatkan pembelajaran mendalam untuk meningkatkan akurasi segmentasi dengan pesat. Metode

segmentasi instan dapat dikelompokkan menjadi tiga kategori utama, yaitu metode multi-tahap, metode satu tahap, dan metode dengan pengawasan semi/lemah (Chuang et al., 2023).



**Gambar 2.2** *Instance Segmentation*

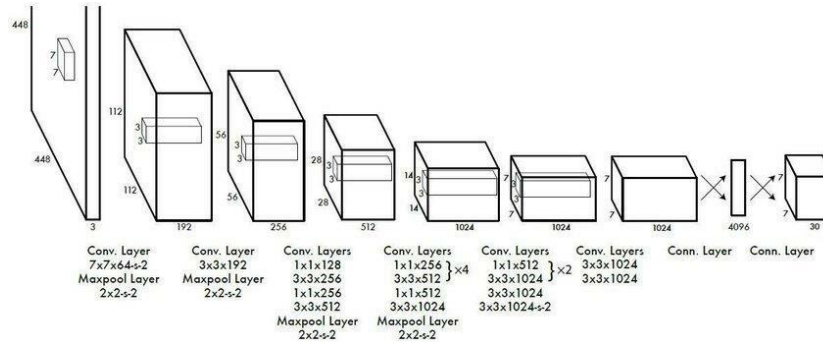
#### 2.4 *You Only Look Once (YOLO)*

Joseph Redmon pertama kali menguraikan model asli YOLO dalam makalah tahun 2015 berjudul "*You Only Look Once: Unified, Real-Time Object Detection*". Meskipun memiliki kecepatan operasi yang tinggi, berkisar dari 45 fps hingga 155 fps dengan optimalisasi kecepatan, Algoritma ini menghadirkan tingkat akurasi prediksi yang lebih rendah (Kosudiwandi, 2020). Sejak diperkenalkan pertama kali, YOLO telah mengalami beberapa iterasi, dengan perbaikan dalam akurasi, kecepatan, dan fitur. Versi-versi populer termasuk YOLOv1, YOLOv2 (juga dikenal sebagai YOLO9000), YOLOv3, YOLOv4, YOLOv5, YOLOv6, YOLOv7 dan varian terbaru seperti YOLOv8, serta implementasi yang dibuat oleh komunitas.

Secara keseluruhan, YOLO (*You Only Look Once*) bekerja dengan membagi gambar menjadi bagian-bagian kecil yang disebut sel. Setiap sel pada grid menghasilkan prediksi beberapa kotak pembatas (*bounding box*) beserta tingkat keyakinan tertentu untuk setiap kotak tersebut. Skor kepercayaan tertentu ini mengindikasikan seberapa yakin model bahwa terdapat objek dalam sel tersebut, sekaligus memberikan penilaian atas keakuratan kotak pembatas yang dibuat. Selain itu, setiap sel juga memprediksi kelas objek yang mungkin ada di dalamnya. Namun, satu sel hanya dapat memprediksi satu kelas, yang merupakan salah satu keterbatasan algoritma YOLO. *Output* prediksi dari model ini berbentuk tensor yang menggambarkan seluruh grid pada gambar, yang mencakup prediksi kelas,



kotak pembatas, dan skor kepercayaan untuk masing-masing sel (Vinh & Byeon, 2023).



**Gambar 2.3** Arsitektur YOLO

YOLO menggunakan skor kepercayaan (*confidence score*) untuk setiap kotak pembatas (*bounding box*). Skor kepercayaan ini menunjukkan probabilitas bahwa kotak tersebut mengandung objek, dikali dengan nilai IoU (*Intersection over Union*) antara kotak prediksi dan *ground truth* yang didapatkan selama proses pelatihan. IoU digunakan sebagai metrik evaluasi untuk mengukur seberapa baik deteksi objek pada dataset tertentu.

$$\begin{aligned}
 \text{Intersection over Unit (IoU)} &= \frac{\text{Area of overlap}}{\text{Area of union}} \\
 &= \frac{|A \cap B|}{|A \cup B|}
 \end{aligned}$$

**Gambar 2.4** Rumus IoU

Skor keyakinan kelas (*class confidence score*), yang merupakan hasil dari probabilitas kondisional kelas dan skor kepercayaan kotak pembatas, adalah faktor kunci dalam prediksi akhir. Skor ini mencerminkan tingkat kepercayaan terhadap kelas tertentu untuk setiap kotak pembatas, menunjukkan kemungkinan adanya

kelas spesifik dalam kotak tersebut, serta seberapa baik kotak prediksi cocok dengan objek yang sebenarnya.

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis Sistem**

Analisis sistem adalah tahapan pada penelitian yang berfokus pada merinci komponen-komponen yang diperlukan agar suatu sistem berfungsi. Analisis sistem dibagi menjadi dua kategori, yaitu analisis masalah dan analisis kebutuhan. Analisis masalah digunakan untuk mengidentifikasi sumber serta dampak dari suatu masalah, sedangkan analisis kebutuhan berperan dalam menentukan data dan proses yang diperlukan dalam mendesain suatu rangkaian.

##### **3.1.1 Analisis Masalah**

Dalam penelitian, analisis masalah digunakan untuk mengidentifikasi sumber utama permasalahan, kemudian dilakukan pemeriksaan mendalam pada isu yang perlu diselesaikan, sehingga dapat mengembangkan sistem yang efisien. Dalam konteks ini, isu pengenalan dan analisis tubuh melalui citra dengan cara segmentasi menjadi fokus utama, memungkinkan penguraian bagian tubuh manusia untuk deteksi dan pengenalan yang lebih akurat. Analisis masalah ini membantu mengidentifikasi kendala utama dalam pengenalan dan analisis tubuh manusia melalui citra, di mana salah satu pendekatan yang diusulkan adalah dengan memanfaatkan *Instance Segmentation* menggunakan YOLO (*You Only Look Once*) dengan tujuan setiap segmen tubuh manusia dapat diidentifikasi secara individual, memungkinkan pemisahan antar bagian seperti kepala, badan, lengan kanan, dan lengan kiri. Dengan demikian, pendekatan ini dapat berguna dalam berbagai aplikasi, mulai dari analisis citra medis hingga pengukuran antropometri, karena memberikan detail dan konteks yang lebih baik tentang struktur tubuh manusia.

##### **3.1.2 Analisis Kebutuhan**

Sasaran utama analisis kebutuhan adalah memastikan bahwa sistem dapat memenuhi harapan pengguna. Untuk itu, analisis kebutuhan digunakan untuk mengenali data dan proses yang diperlukan dalam perancangan sistem. Analisis ini mencakup persyaratan fungsional dan non-fungsional yang harus dipenuhi dalam desain sistem agar sistem yang sedang dibangun dapat mencapai tujuannya.

### 3.1.2.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional melibatkan penjelasan mengenai prosedur yang harus dijalankan oleh sistem untuk memenuhi persyaratan. Persyaratan fungsional yang diperlukan dalam sistem ini mencakup:

1. Sistem dapat memisahkan objek tubuh manusia.
2. Mendeteksi dan menyensori kepala, badan, lengan atas dan lengan kiri.
3. Menghasilkan hasil segmentasi yang dapat digunakan untuk perhitungan lebih lanjut.

### 3.1.2.2 Kebutuhan Non-Fungsional

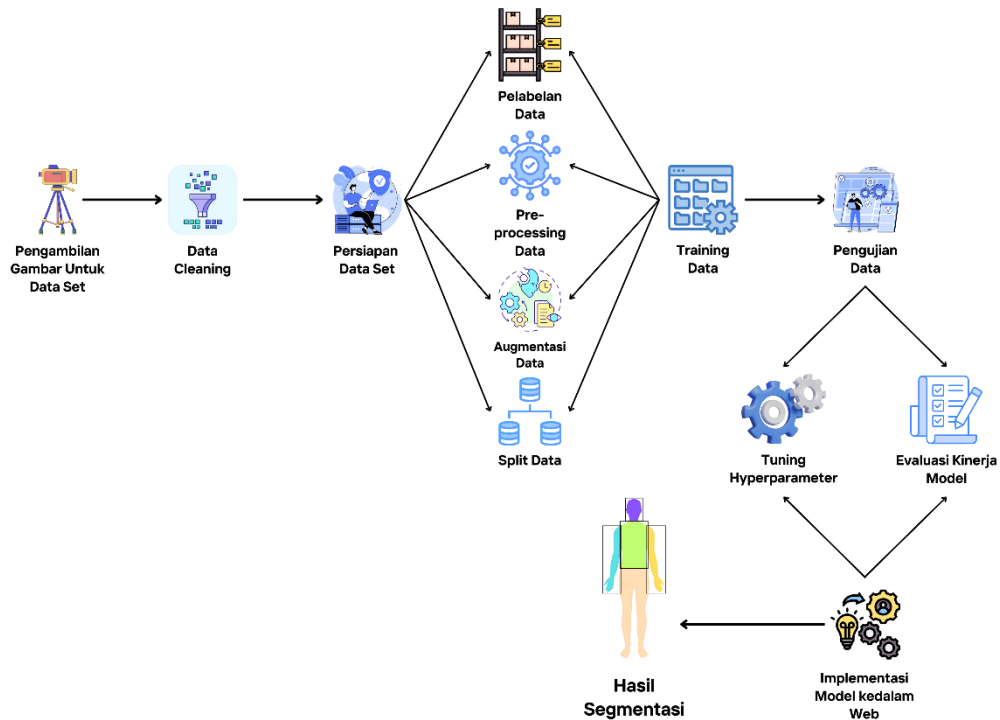
Kebutuhan non-fungsional mengacu pada fitur, karakteristik, atau batasan yang terkait dengan fungsi atau pelayanan yang disediakan oleh sistem. Berikut adalah persyaratan fungsional yang diperlukan untuk sistem ini:

1. Antarmuka sistem ini dirancang agar mudah dipahami, sehingga dapat digunakan dengan mudah.
2. Untuk mengoperasikan sistem, perangkat harus terhubung ke internet.
3. Batasan bagian tubuh manusia yang dapat terdeteksi adalah kepala, badan, lengan kanan dan lengan kiri.
4. Sistem mampu memproses *image segmentation* dengan cepat

### 3.1.3 Arsitektur Sistem

Arsitektur umum sistem yaitu melakukan pengambilan gambar manusia yang kemudian digunakan sebagai data set. Setelah terkumpul, gambar tersebut dilakukan pemilahan (*Data Cleaning*) mana gambar yang dapat digunakan dan mana gambar yang tidak dapat digunakan. Sebelum data set dilatih dilakukan terlebih dahulu persiapan terhadap data set tersebut, yakni dengan melakukan pelabelan manual terhadap data set yang dilakukan pada *platform Roboflow*. Selain pelabelan manual *pre-processing* juga dilakukan terhadap data set, pada tahap ini peneliti melakukan *resize* terhadap gambar (data set). Pada tahap ini juga peneliti melakukan augmentasi dan *split* terhadap data set. Setelah dilakukan persiapan dengan langkah-langkah tersebut, data set kemudian dilatih pada Google Colab. Setelah dilakukan *training* terhadap data set, kemudian dilakukan pengujian terhadap hasil dari *training* tersebut dengan melakukan proses *tuning hyperparameter* dan evaluasi kinerja model. Setelah didapatkan hasil terbaik model

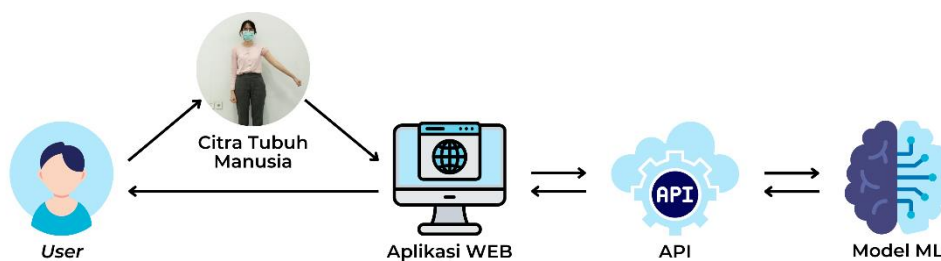
melalui *training* dan pengujian data set, kemudian model dapat langsung diimplementasikan kedalam web. Adapun hasil keluaran sistem berupa hasil gambar segmentasi.



**Gambar 3.1** Arsitektur Sistem

### 3.1.4 Arsitektur Umum

Arsitektur umum adalah struktur suatu sistem dirancang dan bagaimana komponen-komponen di dalamnya saling berinteraksi untuk mencapai tujuan. Berikut adalah desain arsitektur umum dari sistem ini:



**Gambar 3.2** Arsitektur Umum

Berikut adalah penjelasan tentang alur proses dalam arsitektur umum yang terlihat pada gambar di atas:

1. Pengguna memulai proses dengan mengunjungi website

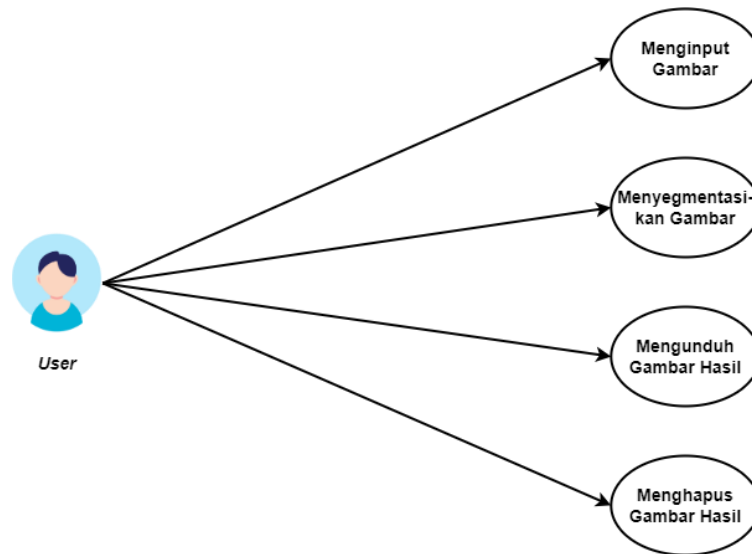
2. Pengguna mengunggah foto yang akan dilakukan segmentasi
3. Foto yang diunggah oleh pengguna akan dikirim ke API untuk diproses lebih lanjut
4. API menggunakan model segmentasi untuk melakukan segmentasi pada foto yang diunggah oleh pengguna berdasarkan objek dan kelas-kelas yang telah ditentukan sebelumnya
5. Hasil gambar yang telah disegmentasi dikembalikan ke API
6. API mengirim foto hasil segmentasi kembali
7. Aplikasi berbasis web menampilkan gambar hasil segmentasi kepada pengguna

### **3.2 Pemodelan Sistem**

Pemodelan sistem melibatkan penjelasan tahapan interaksi antara pengguna dengan aplikasi yang dibangun untuk memastikan sistem beroperasi secara optimal. Biasanya, pemodelan sistem digambarkan menggunakan UML (*Unified Modeling Language*), bahasa pemodelan yang umum digunakan untuk menggambarkan hubungan antar komponen dalam sistem agar dapat berinteraksi melalui pengguna. Dalam penelitian ini, kerangka UML yang dimanfaatkan terdiri dari *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

#### **3.2.1 Use Case Diagram**

*Use case diagram* adalah cara pemodelan yang digunakan untuk memaparkan hubungan antara pengguna dan sistem yang telah dibuat.

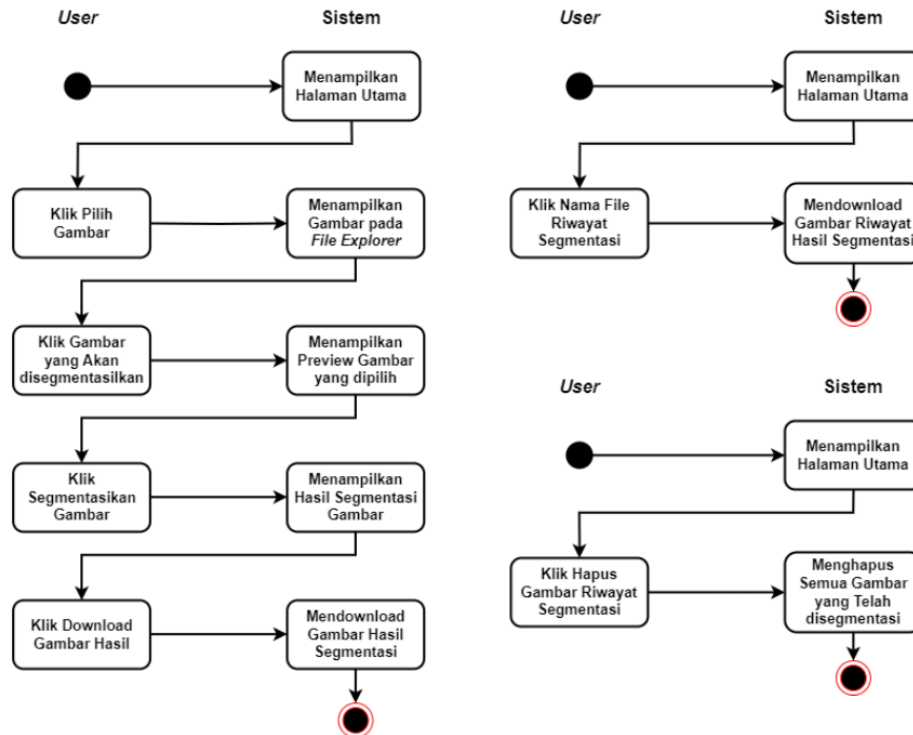


**Gambar 3.3** *Use Case Diagram*

Berdasarkan gambar 3.2 mengilustrasikan interaksi antara pengguna dengan sistem. Dimulai dari *user* membuka web dan akan disambut dengan halaman utama. Pada halaman utama, pengguna akan diberikan dua pilihan, diantaranya adalah "pilih gambar". Setelah memilih "pilih gambar" pengguna bisa mengimpor gambar dari penyimpanan. Setelah diimpor, sistem akan mensegmentasikan gambar tersebut dan pengguna akan masuk ke halaman hasil segmentasi. Selanjutnya pengguna bisa mendownload hasil segmentasi dan kembali ke halaman utama.

### 3.2.2 *Activity Diagram*

*Activity Diagram* merupakan alat visual yang dipakai untuk mendeskripsikan alur kerja dan aktivitas dalam proses atau sistem. Ini mencatat berbagai langkah dan kegiatan yang terjadi dari awal hingga akhir suatu proses atau aktivitas. Diagram ini memperlihatkan urutan langkah-langkah, keputusan, garis waktu, dan aliran kontrol dalam proses tersebut. Dengan demikian, *activity diagram* membantu dalam memahami secara visual bagaimana suatu sistem atau proses berjalan dan bagaimana berbagai elemen terkait berinteraksi satu sama lain.



**Gambar 3.4** Activity Diagram

Pada gambar 3.3 dimulai dari *user* membuka halaman utama, lalu sistem akan diberi beberapa pilihan yaitu "hapus gambar riwayat segmentasi", "nama file riwayat segmentasi" dan "pilih gambar". Jika *user* memilih "hapus gambar riwayat segmentasi" maka sistem akan menghapus semua gambar yang telah disegmentasi dan menampilkan halaman utama kembali. Lalu jika *user* memilih "nama file riwayat segmentasi", *user* akan dibawa ke halaman *download* gambar riwayat hasil segmentasi dan menampilkan halaman utama kembali.

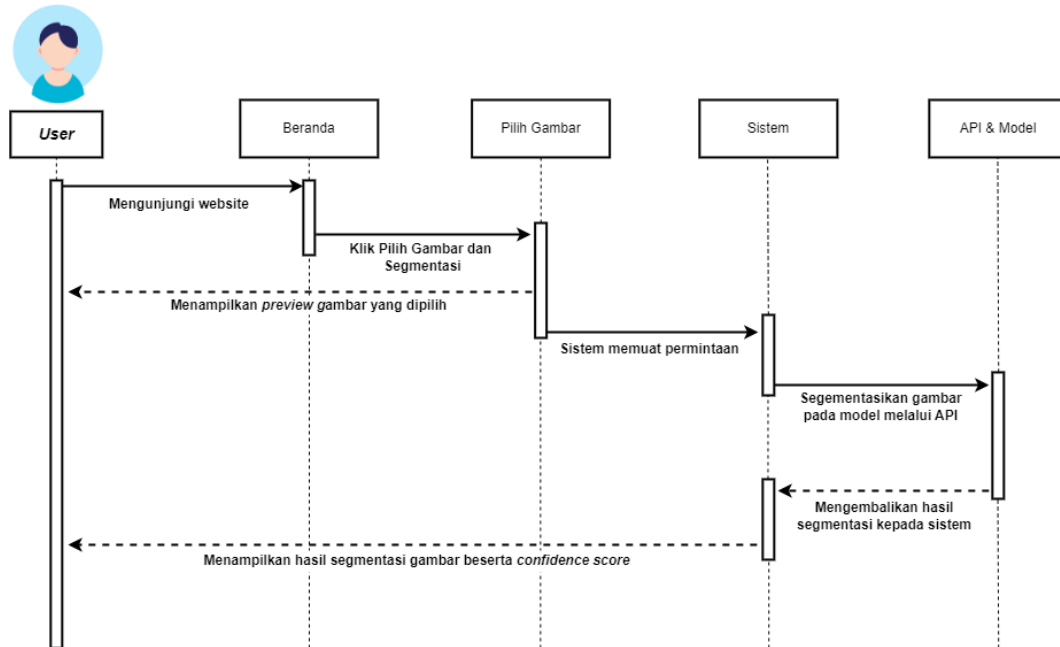
Sedangkan jika *user* memilih "pilih gambar" maka *user* akan diarahkan ke halaman menampilkan gambar pada *file explorer*. Lalu *user* akan memilih gambar yang akan disegmentasikan. Kemudian sistem menampilkan *preview* gambar yang dipilih. Jika *user* mengklik segmentasi gambar maka sistem akan menunjukkan hasil segmentasi gambar. Lalu jika *user* mengklik *download* hasil gambar pada halaman hasil, maka sistem akan mendownload gambar hasil segmentasi dan kembali ke halaman utama.

### 3.2.3 Sequence Diagram

Diagram urutan adalah jenis diagram interaksi dalam pembentukan sistem dengan memvisualisasikan bagaimana objek dalam sistem berinteraksi dalam urutan waktu



tertentu. Diagram ini berguna untuk memahami bagaimana berbagai objek atau komponen sistem berkomunikasi dan berinteraksi antara satu dengan yang lain dalam suatu skenario tertentu.



**Gambar 3.5** *Sequence Diagram*

Pada *sequence diagram* mengilustrasikan beberapa komponen yang saling berinteraksi. Gambar 3.4 menjabarkan proses dari awal *user* mengunjungi website kemudian memilih gambar yang akan disegmentasikan sampai dengan sistem menampilkan gambar hasil segmentasi beserta *confidence score*-nya.

### 3.3 *Flowchart*

*Flowchart* adalah diagram Yang menggunakan simbol-simbol grafis untuk menjelaskan tahapan dalam suatu proses atau alur kerja. Simbol-simbol tersebut dihubungkan dengan panah untuk menunjukkan urutan dan arah aliran. *Flowchart* membantu memvisualisasikan proses secara sederhana, memudahkan pemahaman, dan digunakan untuk menganalisis atau mendesain sistem.

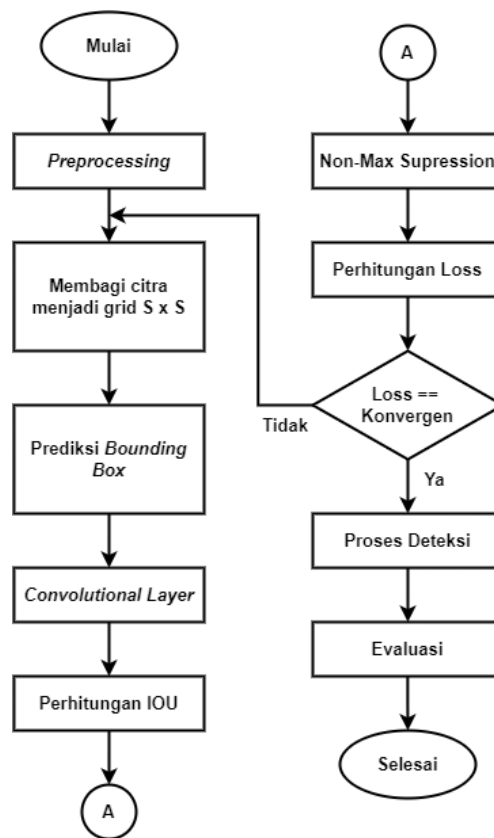
### 3.3.1 Flowchart Sistem



**Gambar 3.6** Flowchart Sistem

*Flowchart* sistem menggambarkan urutan kerja sistem. Tahapan pertama yaitu menginisiasi model deteksi dan server. Kemudian pengguna menginput gambar yang akan disegmentasi. Selanjutnya sistem akan mengirim gambar ke API segmentasi gambar. Lalu setelah gambar disegmentasikan dengan model, sistem akan menampilkan gambar hasil segmentasi.

### 3.3.2 Flowchart YOLO



**Gambar 3.7** Flowchart YOLO

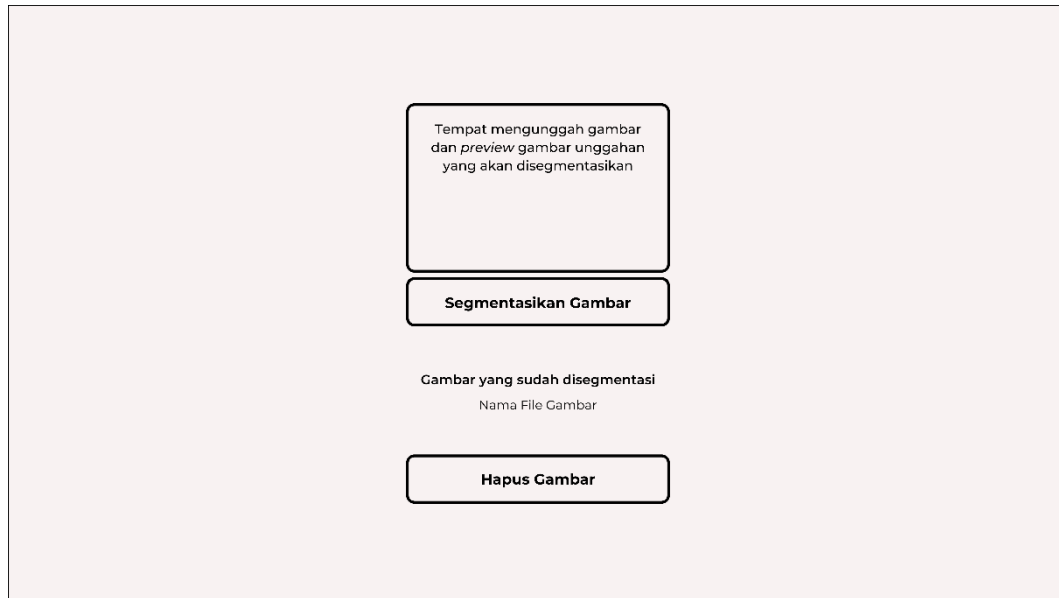
Flowchart menggunakan metode YOLO (*You Only Look Once*) dimulai dengan tahap *preprocessing*. Citra dibagi menjadi grid  $s \times s$  untuk memudahkan deteksi objek di setiap grid. Dilakukan prediksi *bounding box* dan konvolusi untuk mengekstraksi fitur citra. IoU dihitung untuk mengevaluasi akurasi deteksi, diikuti dengan seleksi non-maximal untuk mempertahankan *bounding box* yang paling relevan. Proses berlanjut dengan menghitung nilai loss dan jika tidak konvergen, proses kembali ke tahap awal. Deteksi objek menggunakan hasil prediksi *bounding box* yang disesuaikan, dan hasil deteksi dievaluasi untuk mengukur kinerja sistem sebelum proses selesai.

### 3.4 Perancangan Interface

Perancangan *interface* adalah proses merancang tampilan dan interaksi pengguna dalam sebuah aplikasi, situs web, atau perangkat lunak. Ini melibatkan pemikiran tentang bagaimana pengguna akan berinteraksi dengan sistem, termasuk tata letak

elemen-elemen antarmuka, pengaturan warna dan desain, serta navigasi *interface*. Interface harus dirancang sesuai kebutuhan dan preferensi pengguna untuk mencapai tujuan yang diinginkan.

### 3.4.1 Halaman Beranda

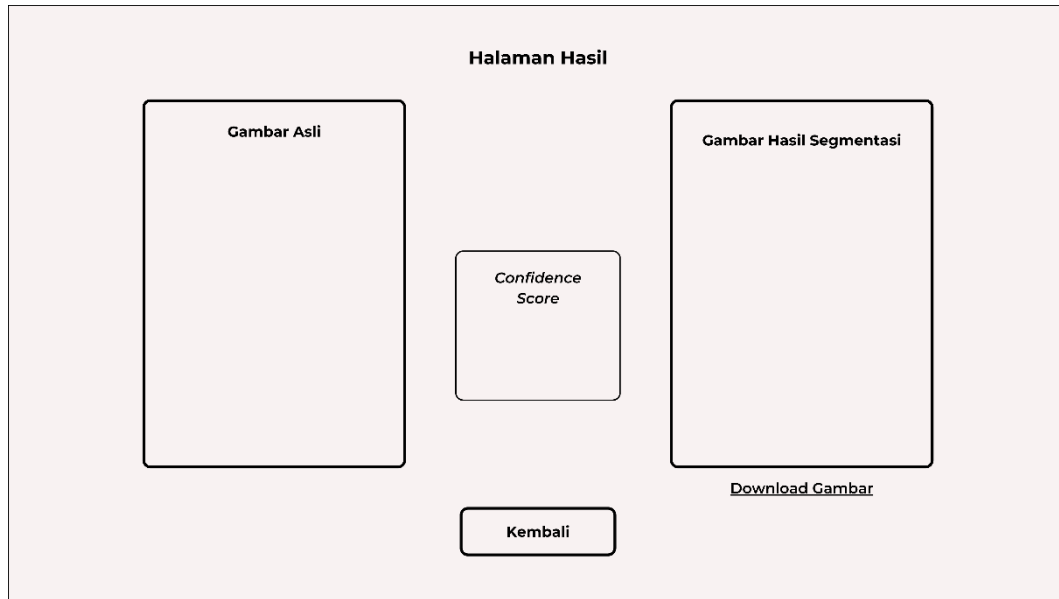


**Gambar 3.8** Desain Halaman Beranda

Gambar diatas merupakan desain halaman beranda sederhana yang mencakup semua menu utama. Pada tampilannya terdapat menu “pilih gambar”, “segmentasikan gambar” dan “hapus gambar”. Pada bagian menu “pilih gambar”, pengguna akan diminta untuk memilih gambar di *file explorer* yang akan disegmentasi. Kemudian user akan memilih menu “segmentasikan gambar” untuk menampilkan hasil gambar yang telah disegmentasikan.

Pengguna masih bisa melihat gambar hasil segmentasi sebelumnya dengan mengklik gambar yang sudah disegmentasi. Sedangkan untuk menghapus hasil gambar yang sudah disegmentasi, pengguna bisa mengklik menu “hapus gambar”.

### 3.4.2 Halaman Hasil

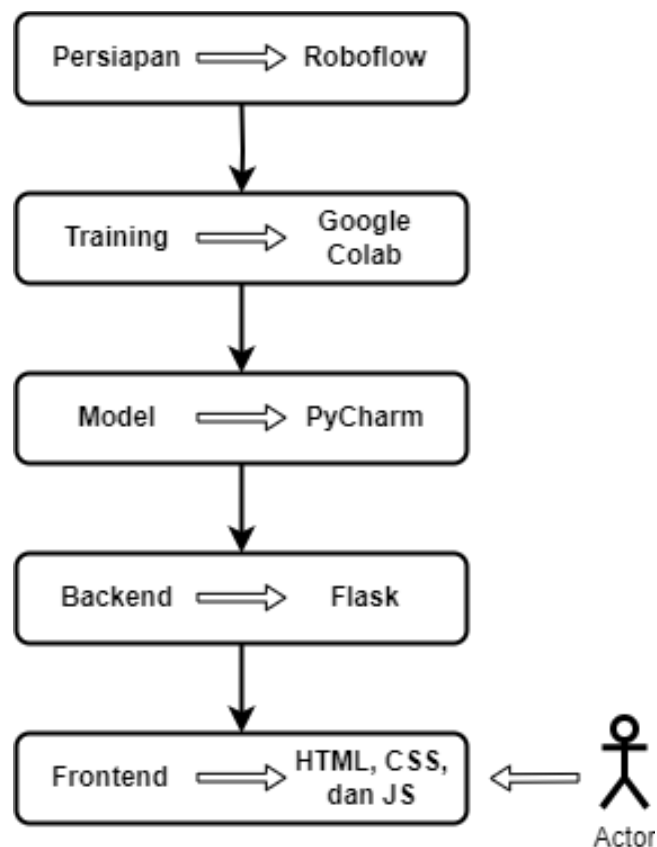


**Gambar 3.9** Desain Halaman Hasil

Pada gambar 3.8 menampilkan desain halaman hasil ketika gambar telah diinput dan menghasilkan gambar yang telah disegmentasi. Pada halaman ini juga ditampilkan *confidence score* dari hasil deteksi. Serta pada halaman ini juga pengguna dapat menyimpan gambar hasil segmentasi dengan menekan tombol *Download Gambar*.

## BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

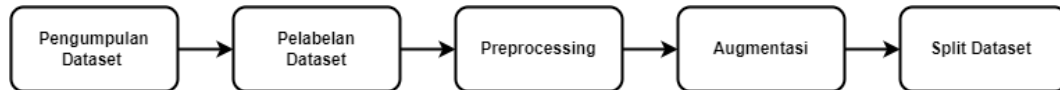
Proses implementasi dimulai dengan menyiapkan data set menggunakan platform *Roboflow*, di mana data dikumpulkan, diproses, dan diatur agar siap untuk proses pelatihan. Kemudian, model deteksi hasil proses pada *Roboflow* dilatih menggunakan *Google Colab*, di mana algoritma pembelajaran mesin diterapkan dan model diperbaiki sesuai kebutuhan. Setelah model terlatih, kita mengelola dan mengintegrasikannya dengan *PyCharm* untuk pengembangan lebih lanjut. *Backend* untuk autentikasi dibangun untuk keamanan data dan manajemen pengguna menggunakan *Flask* untuk analisis data. Terakhir, tampilan pengguna dibuat dengan HTML, CSS, dan JavaScript, untuk memastikan antarmuka yang responsif dan integrasi yang baik dengan *backend*.



**Gambar 4.1** Implementasi Sistem

#### 4.1 Implementasi Tahap Persiapan Dataset

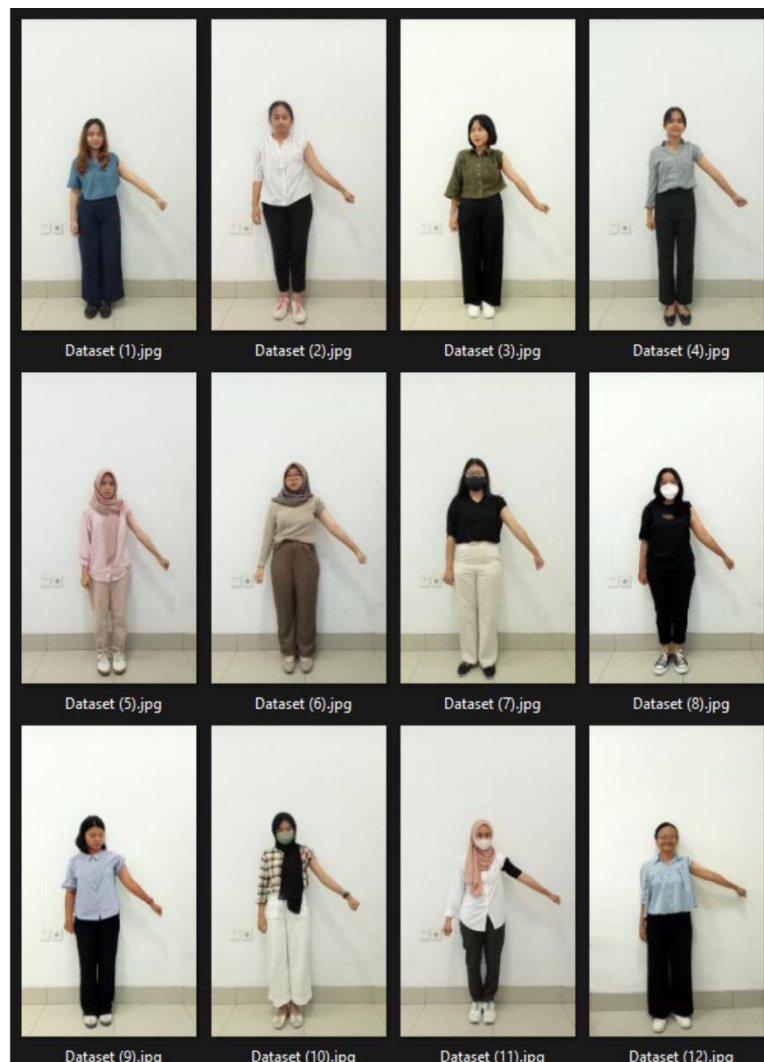
Dataset yang digunakan didalam penelitian ini diambil secara langsung pada mahasiswi Fakultas Kesehatan Masyarakat (FKM) USU. Proses persiapan dapat diilustrasikan kedalam gambar sebagaimana berikut:



**Gambar 4.2** Proses Persiapan Dataset

##### 4.1.1 Pengumpulan Dataset

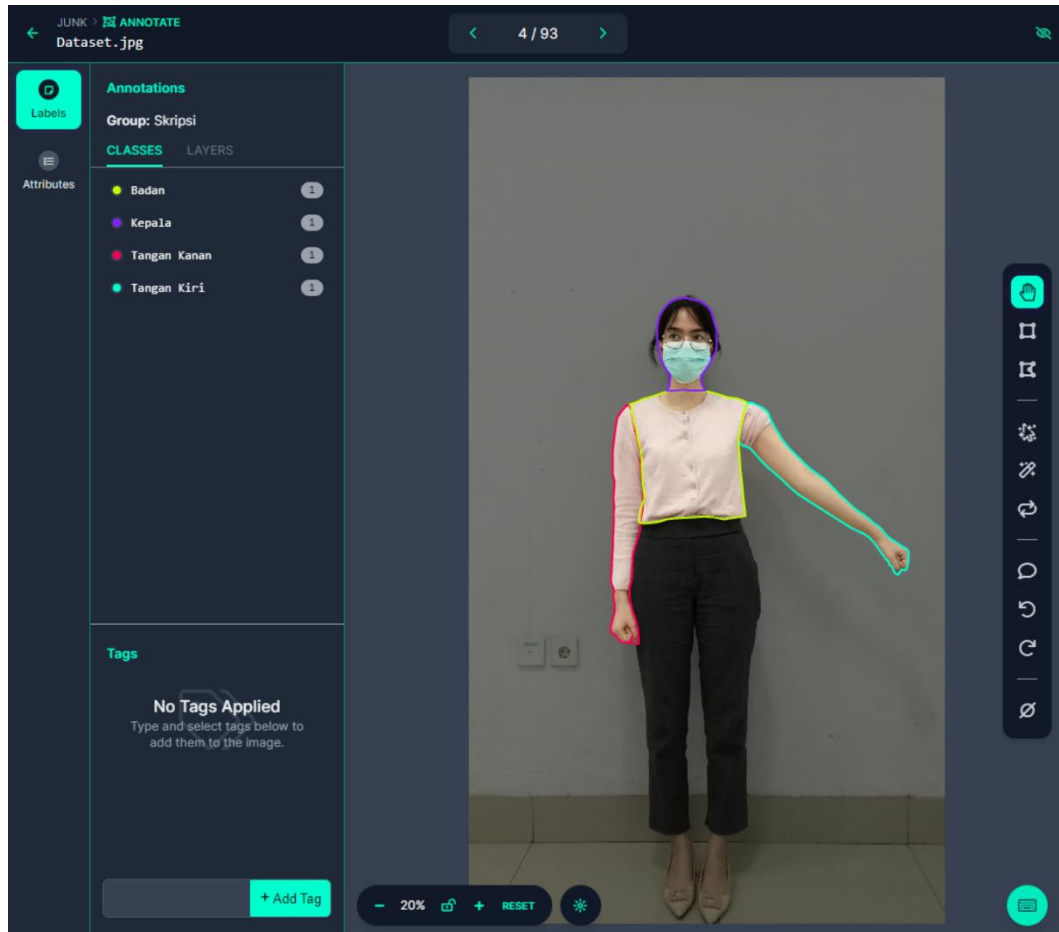
Data set dikumpulkan secara langsung di Fakultas Kesehatan Masyarakat, Universitas Sumatera Utara. Gambar diambil saat responden berdiri tegak, dengan tangan kiri tidak terhalangi oleh pakaian dan diangkat sedikit, sementara tangan kanan diarahkan lurus kebawah. Adapun contoh dataset sebagai berikut:



**Gambar 4.3** Contoh Dataset

### 4.1.2 Pelabelan Data

Pelabelan adalah proses menandai atau menetapkan area pembatas dan kotak pembatas (*bounding box*) untuk objek yang ingin dideteksi dalam gambar. Pelabelan ini penting karena YOLO memerlukan data latihan yang sudah ditandai dengan benar untuk melakukan pelatihan model deteksi objek.



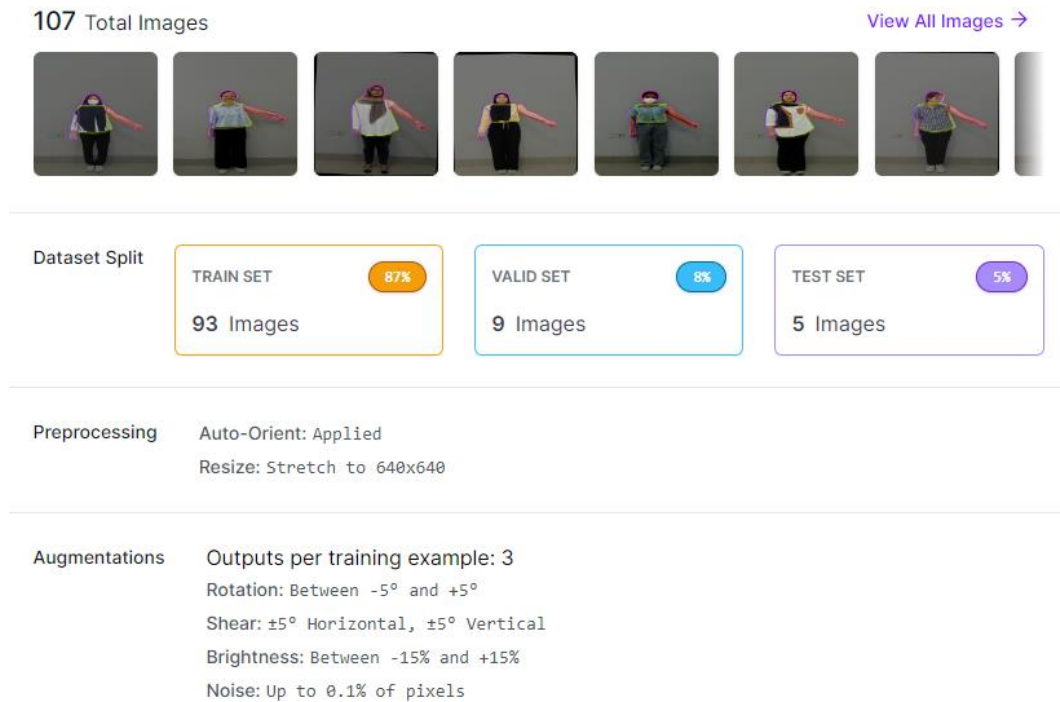
**Gambar 4.4** Proses Pelabelan Dataset

Proses pelabelan dataset dalam penelitian ini menggunakan menu *Polygon Tool* yang disediakan oleh platform Roboflow. Langkah ini memungkinkan penandai untuk secara akurat menetapkan area pembatas di sekitar objek yang ingin dideteksi dalam gambar. Dengan menggunakan menu ini, objek dapat ditandai dengan tepat, dan kemudian label yang sesuai dapat diterapkan untuk memberikan informasi yang jelas tentang jenis objek yang terdeteksi. Proses ini memastikan bahwa data latihan yang dihasilkan memiliki kualitas yang tinggi, yang nantinya akan meningkatkan kinerja model deteksi objek yang akan dilatih. Dengan



demikian, proses pelabelan ini memainkan peran kunci dalam memastikan akurasi dan keandalan model deteksi objek yang dihasilkan.

#### 4.1.3 *Preprocessing, Augmentasi, dan Split Data set*



**Gambar 4.5** Pembagian Dataset

Pada penelitian ini dilakukan *preprocessing* yang bertujuan untuk menyiapkan data agar sesuai dengan format atau persyaratan yang diperlukan oleh model. Pada penelitian ini *preprocessing* yang dilakukan adalah melakukan resize gambar menjadi 640x640. Augmentasi dilakukan dengan tujuan untuk meningkatkan keragaman-keragaman data latihan, sehingga model dapat belajar dari beragam situasi atau kondisi. Augmentasi yang dilakukan pada dataset penelitian ini meliputi *rotation*, *shear*, *brightness*, *darken* dan *noise*. Setelah proses diatas selesai didapatkan 107 total gambar kemudian gambar tersebut dibagi menjadi tiga bagian set yaitu *Train set* 93 gambar, *Valid Set* 9 gambar dan *Test set* sebanyak 5 gambar, pembagian ini dipilih dengan mempertimbangkan jumlah dataset yang tidak terlalu banyak.

#### 4.2 *Training Data set*

Setelah dilakukan proses persiapan data set, kemudian dilakukan *training* dataset pada Google Colab.

#### 4.2.1 Mengunduh Library Ultralytics

```
!pip install ultralytics
```

**Gambar 4.6** Kode untuk Mengunduh *Library Ultralytics*

Gambar 4.6 diatas merupakan proses mengunduh *Library Ultralytics* sehingga dapat mengakses fungsionalitasnya dan menggunakan alat dan model yang disediakan.

#### 4.2.2 Import YOLO

```
from ultralytics import YOLO
from matplotlib import pyplot as plt
from PIL import Image
```

**Gambar 4.7** Kode untuk *Import YOLO*

Gambar 4.7 diatas merupakan proses mengimpor YOLO dari *Library Ultralytics* yang sebelumnya telah dideklarasikan. Pada baris ini juga terdapat pemanggilan pyplot. Pyplot adalah submodul dari Matplotlib yang memungkinkan pembuatan visualisasi data dengan Python secara sederhana. Ini menyediakan fungsi untuk membuat plot, mengatur penampilan plot, mengelola sumbu dan label, menambahkan anotasi, legenda, serta menyimpan plot. Pyplot membantu membuat grafik seperti *line* plot, *scatter* plot, bar plot, histogram, dan lainnya dengan mudah dan cepat.

#### 4.2.3 Inisialisasi Model YOLO untuk *Instance Segmentation*

```
model = YOLO('yolov8n-seg.yaml')
model = YOLO('yolov8n-seg.pt')
```

**Gambar 4.8** Kode untuk menginisialisasi Model YOLO

Gambar 4.8 diatas merupakan proses inisialisasi model YOLO untuk *Instance Segmentation*. Yang pertama, `model = YOLO('yolov8n-seg.yaml')` menginisialisasi model YOLO menggunakan konfigurasi yang didefinisikan dalam file '*yolov8n-seg.yaml*'. Konfigurasi ini berisi detail mengenai arsitektur model, *hyperparameters*, dan pengaturan lain yang diperlukan untuk melakukan *Instance*

*Segmentation*. Kemudian, `model = YOLO('yolov8n-seg.pt')` menggantikan model sebelumnya dengan model yang sudah dilatih sebelumnya dan disimpan dalam file `'yolov8n-seg.pt'`. Model yang telah dilatih ini siap digunakan untuk memprediksi objek dalam gambar baru tanpa perlu melalui proses pelatihan lagi.

Yang pertama ada `'yolov8n-seg.yaml'` merupakan sebuah dokumen konfigurasi yang diperlukan untuk menjalankan proses *Instance Segmentation* dengan menggunakan YOLOv8n. *Instance Segmentation* adalah teknik dalam pengolahan citra yang bertujuan untuk mengenali dan memisahkan objek-objek individual dalam sebuah gambar, serta memberikan label kelas pada setiap objek dan menggambar kotak pembatas untuk masing-masing objek tersebut. Baris ini umumnya berisi informasi mengenai spesifikasi model, *hyperparameters*, dan pengaturan lain yang diperlukan untuk melatih model atau menjalankan proses *Instance Segmentation*.

#### 4.2.4 Proses Mengakses Data set

```
import yaml
with open("/content/drive/MyDrive/InstanceFinal/data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

**Gambar 4.9** Kode untuk Mengakses Dataset

Gambar 4.9 diatas merupakan proses pengaksesan data set dalam bentuk file YAML yang sebelumnya sudah diupload pada Google Drive yang kemudian dapat digunakan untuk melatih model atau melakukan berbagai tugas pengolahan data lainnya. Melalui file `data.yaml` ini akan diarahkan ke folder tempat penyimpanan dataset.

```
data.yaml X
1 train: /content/drive/MyDrive/InstanceFinal/train/images
2 val: /content/drive/MyDrive/InstanceFinal/valid/images
3 test: ../test/images
4
5 nc: 4
6 names: ['Badan', 'Kepala', 'Tangan Kanan', 'Tangan Kiri']
```

**Gambar 4.10** File `data.yaml`

#### 4.2.5 Proses Mendefinisikan Jumlah *Class* pada Data set

```
import yaml
with open("/content/drive/MyDrive/InstanceFinal/data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])
```

**Gambar 4.11** Kode untuk Mendefinisikan Jumlah *Class* pada Data set

Gambar 4.11 diatas bertujuan untuk membaca file YAML yang berisi informasi tentang jumlah kelas (*num\_classes*) yang ada dalam data set. Yang pertama mengimpor modul 'yaml', yang memungkinkan untuk membaca dan menulis data YAML dalam Python. Kemudian file YAML yang ada pada Google Drive dibuka dan mengkonversinya menjadi struktur data Python. Kemudian mengakses nilai yang terkandung dalam file YAML setelah dibaca dan mengubahnya menjadi *string* dan disimpan dalam variabel 'num\_classes'.

#### 4.2.6 Proses *Training* Data set

```
results = model.train(data='/content/drive/MyDrive/InstanceFinal/data.yaml',
                      project=project,
                      name=name,
                      epochs=200,
                      patience=0,
                      batch=4,
                      imgsz=640)
```

**Gambar 4.12** Kode untuk *Training* Dataset

Gambar 4.12 diatas merupakan proses pelatihan model deteksi objek menggunakan arsitektur YOLOv8n dengan menggunakan *library Ultralytics*. Dalam proses ini, sebuah objek model YOLOv8n dibuat dengan menyediakan *path* menuju file model yang telah dilatih sebelumnya. Selanjutnya, metode 'train()' dipanggil pada objek model tersebut dengan menyediakan *path* menuju file YAML yang berisi konfigurasi dataset, parameter-parameter pelatihan seperti jumlah *epochs*, *patience*, ukuran *batch*, dan ukuran gambar. Proses pelatihan dilakukan menggunakan dataset yang telah ditentukan dan hasilnya disimpan dalam variabel 'results'. Kode ini memastikan bahwa model dilatih sesuai dengan konfigurasi yang ditentukan, dan hasil pelatihan dapat digunakan untuk evaluasi lebih lanjut atau untuk penerapan praktis dalam tugas deteksi objek. Hasil dari pelatihan ini juga untuk mendapatkan file model yang nantinya akan digunakan pada implementasi website.

### 4.3 *Tuning Hyperparameter dan Pengujian Model*

Pada penelitian ini, penyetelan *hyperparameter* dilakukan dengan kombinasi berbagai nilai jumlah *epochs*, yakni 100, 150, 200, dan 250 *epochs* dengan *learning rate* 0.01. Kombinasi-kombinasi ini dieksplorasi secara cermat untuk menemukan model terbaik dan optimal. Hasil *tuning hyperparameter* dapat diperhatikan dalam tabel dibawah ini:

**Tabel 4.1** Training Epoch 100, Learning Rate 0.01

Epochs	Box			Mask		
	Precision	Recall	mAP	Precision	Recall	mAP
1/100	0.91373	0.99423	0.97455	0.91373	0.99423	0.97455
2/100	0.93656	0.9531	0.98106	0.93656	0.9531	0.98106
3/100	0.93906	0.95	0.97917	0.93906	0.95	0.97917
4/100	0.91238	0.98564	0.99045	0.91238	0.98564	0.99045
5/100	0.92969	0.95532	0.96619	0.92969	0.95532	0.96619
6/100	0.92356	0.95	0.95429	0.92356	0.95	0.95429
7/100	0.92349	0.95	0.95307	0.92349	0.95	0.95307
8/100	0.92822	0.95	0.97598	0.92822	0.95	0.97598
9/100	0.94454	0.95	0.97959	0.94454	0.95	0.97959
10/100	0.94922	0.95	0.97479	0.94922	0.95	0.97479
...	...	...	...	...	...	...
91/100	0.91288	0.99636	0.9542	0.91288	0.99636	0.9542
92/100	0.91374	0.99505	0.9542	0.91374	0.99505	0.9542
93/100	0.91451	0.99298	0.9542	0.91451	0.99298	0.9542
94/100	0.91496	0.99388	0.97227	0.91496	0.99388	0.97227
95/100	0.91539	0.99364	0.97682	0.91539	0.99364	0.97682
96/100	0.91512	0.99524	0.98364	0.91512	0.99524	0.98364
97/100	0.91449	0.99619	0.98591	0.91449	0.99619	0.98591
98/100	0.91327	0.99655	0.97455	0.91327	0.99655	0.97455
99/100	0.91365	0.99667	0.97227	0.91365	0.99667	0.97227
100/100	0.91323	0.99644	0.96773	0.91323	0.99644	0.96773

**Tabel 4.2** Summary Training Epoch 100, Learning Rate 0.01

Class	Box			Mask		
	Precision	Recall	mAP	Precision	Recall	mAP
All	0.913	0.996	0.972	0.913	0.996	0.972
Badan	0.919	1	0.995	0.919	1	0.995
Kepala	0.918	1	0.995	0.918	1	0.995

<b>Tangan Kanan</b>	0.909	0.994	0.931	0.909	0.94	0.931
<b>Kiri</b>	0.908	0.99	0.968	0.908	0.99	0.968

**Tabel 4.3** Training Epoch 150, Learning Rate 0.01

<b>Epochs</b>	<b>Box</b>			<b>Mask</b>		
	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>1/150</b>	0.94409	0.95311	0.96764	0.94409	0.95311	0.96764
<b>2/150</b>	0.94244	0.95174	0.96604	0.94244	0.95174	0.96604
<b>3/150</b>	0.94218	0.95508	0.97902	0.94218	0.95508	0.97902
<b>4/150</b>	0.9313	0.98722	0.99045	0.9313	0.98722	0.99045
<b>5/150</b>	0.9704	0.975	0.97795	0.9704	0.975	0.97795
<b>6/150</b>	0.94222	0.95156	0.9456	0.94222	0.95156	0.9456
<b>7/150</b>	0.93508	0.95	0.94132	0.93508	0.95	0.94132
<b>8/150</b>	0.94183	0.95362	0.96197	0.94183	0.95362	0.96197
<b>9/150</b>	0.93851	0.95	0.9601	0.93851	0.95	0.9601
<b>10/150</b>	0.94251	0.95	0.96119	0.94251	0.95	0.96119
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>141/150</b>	0.94375	0.95138	0.95718	0.94375	0.95138	0.95718
<b>142/150</b>	0.94377	0.95136	0.95718	0.94377	0.95136	0.95718
<b>143/150</b>	0.94362	0.95137	0.95945	0.94362	0.95137	0.95945
<b>144/150</b>	0.94367	0.95136	0.95695	0.94367	0.95136	0.95695
<b>145/150</b>	0.94282	0.95	0.94869	0.94282	0.95	0.94869
<b>146/150</b>	0.94371	0.95136	0.95195	0.94371	0.95136	0.95195
<b>147/150</b>	0.94287	0.95	0.95119	0.94287	0.95	0.95119
<b>148/150</b>	0.94337	0.95	0.95869	0.94337	0.95	0.95869
<b>149/150</b>	0.94361	0.95	0.95619	0.94361	0.95	0.95619
<b>150/150</b>	0.94372	0.95	0.96119	0.94372	0.95	0.96119

**Tabel 4.4** Summary Training Epoch 150, Learning Rate 0.01

<b>Class</b>	<b>Box</b>			<b>Mask</b>		
	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>All</b>	0.946	0.952	0.977	0.946	0.952	0.977
<b>Badan</b>	0.993	1	0.995	0.993	1	0.995
<b>Kepala</b>	0.992	1	0.995	0.992	1	0.995
<b>Tangan Kanan</b>	0.897	0.9	0.957	0.897	0.9	0.957
<b>Kiri</b>	0.901	0.907	0.959	0.901	0.907	0.959

**Tabel 4.5** Training Epoch 200, Learning Rate 0.01

Epochs	Box			Mask		
	Precision	Recall	mAP	Precision	Recall	mAP
<b>1/200</b>	0.94313	0.95222	0.96947	0.94313	0.95222	0.96947
<b>2/200</b>	0.941	0.95507	0.96924	0.941	0.95507	0.96924
<b>3/200</b>	0.94465	0.95255	0.95731	0.94465	0.95255	0.95731
<b>4/200</b>	0.94414	0.95145	0.94766	0.94414	0.95145	0.94766
<b>5/200</b>	0.9389	0.95	0.93081	0.9389	0.95	0.93081
<b>6/200</b>	0.94042	0.95171	0.95397	0.94042	0.95171	0.95397
<b>7/200</b>	0.9433	0.95	0.9425	0.9433	0.95	0.9425
<b>8/200</b>	0.94219	0.95	0.93777	0.94219	0.95	0.93777
<b>9/200</b>	0.94325	0.95	0.96417	0.94325	0.95	0.96417
<b>10/200</b>	0.94246	0.95	0.97077	0.94246	0.95	0.97077
...	...	...	...	...	...	...
<b>191/200</b>	0.94561	0.95218	0.98152	0.94561	0.95218	0.98152
<b>192/200</b>	0.91144	0.99697	0.97455	0.91144	0.99697	0.97455
<b>193/200</b>	0.94543	0.9523	0.96877	0.94543	0.9523	0.96877
<b>194/200</b>	0.94542	0.95239	0.97174	0.94542	0.95239	0.97174
<b>195/200</b>	0.94543	0.95243	0.96854	0.94543	0.95243	0.96854
<b>196/200</b>	0.94347	0.95	0.96554	0.94347	0.95	0.96554
<b>197/200</b>	0.94541	0.95253	0.96991	0.94541	0.95253	0.96991
<b>198/200</b>	0.94542	0.95254	0.97241	0.94542	0.95254	0.97241
<b>199/200</b>	0.94741	0.95255	0.97174	0.94741	0.95255	0.97174
<b>200/200</b>	0.91153	0.99678	0.97455	0.91153	0.99678	0.97455

**Tabel 4.6** Summary Training Epoch 200, Learning Rate 0.01

Class	Box			Mask		
	Precision	Recall	mAP	Precision	Recall	mAP
<b>All</b>	0.914	0.995	0.979	0.914	0.995	0.979
<b>Badan</b>	0.92	1	0.995	0.92	1	0.995
<b>Kepala</b>	0.921	1	0.995	0.921	1	0.995
<b>Tangan Kanan</b>	0.908	0.986	0.977	0.908	0.986	0.977
<b>Kiri</b>	0.909	0.994	0.95	0.909	0.994	0.95

**Tabel 4.7** Training Epoch 250, Learning Rate 0.01

Epochs	Box			Mask		
	Precision	Recall	mAP	Precision	Recall	mAP
<b>1/250</b>	0.91378	0.99542	0.97909	0.91378	0.99542	0.97909
<b>2/250</b>	0.94373	0.95303	0.96316	0.94373	0.95303	0.96316
<b>3/250</b>	0.94388	0.95287	0.96907	0.94388	0.95287	0.96907
<b>4/250</b>	0.93584	0.95	0.9491	0.93584	0.95	0.9491
<b>5/250</b>	0.94286	0.95	0.9424	0.94286	0.95	0.9424
<b>6/250</b>	0.9444	0.95	0.92333	0.9444	0.95	0.92333
<b>7/250</b>	0.94336	0.95144	0.95432	0.94336	0.95144	0.95432
<b>8/250</b>	0.91267	0.99466	0.98364	0.91267	0.99466	0.98364
<b>9/250</b>	0.91278	0.99398	0.97909	0.91278	0.99398	0.97909
<b>10/250</b>	0.94587	0.95087	0.95851	0.94587	0.95087	0.95851
...	...	...	...	...	...	...
<b>241/250</b>	0.94601	0.95	0.95208	0.94601	0.95	0.95208
<b>242/250</b>	0.94489	0.95	0.95188	0.94489	0.95	0.95188
<b>243/250</b>	0.94475	0.95	0.95188	0.94475	0.95	0.95188
<b>244/250</b>	0.94465	0.95	0.95002	0.94465	0.95	0.95002
<b>245/250</b>	0.94453	0.95	0.94673	0.94453	0.95	0.94673
<b>246/250</b>	0.94448	0.95	0.95173	0.94448	0.95	0.95173
<b>247/250</b>	0.94445	0.95	0.95173	0.94445	0.95	0.95173
<b>248/250</b>	0.94438	0.95	0.95252	0.94438	0.95	0.95252
<b>249/250</b>	0.94437	0.95	0.95094	0.94437	0.95	0.95094
<b>250/250</b>	0.94445	0.95	0.95094	0.94445	0.95	0.95094

**Tabel 4.8** Summary Training Epoch 250, Learning Rate 0.01

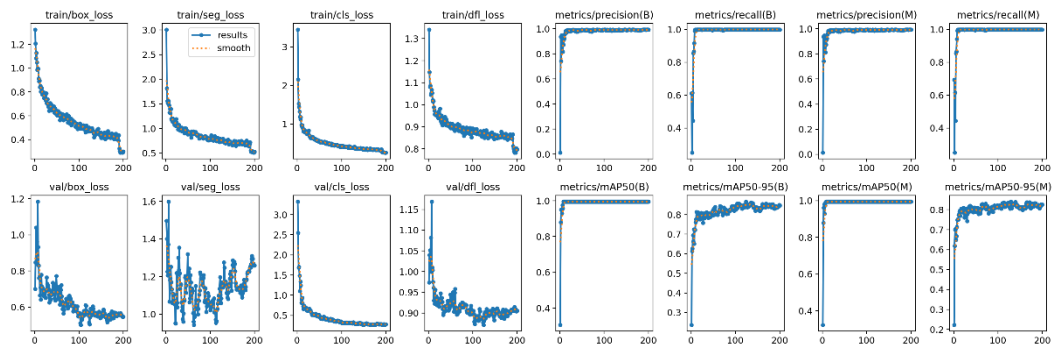
Class	Box			Mask		
	Precision	Recall	mAP	Precision	Recall	mAP
<b>All</b>	0.931	0.974	0.973	0.931	0.974	0.973
<b>Badan</b>	0.916	1	0.995	0.916	1	0.995
<b>Kepala</b>	1	1	0.995	1	1	0.995
<b>Tangan Kanan</b>	0.9	0.9	0.934	0.9	0.9	0.934
<b>Kiri</b>	0.909	0.994	0.968	0.909	0.994	0.968

Setelah melakukan *tuning hyperparameter* termasuk mengatur epoch dan juga *learning rate*, ditemukan kombinasi optimal pada **epoch 200 dengan learning rate 0.01**. Kombinasi ini menghasilkan nilai mAP tertinggi di antara kombinasi lainnya, yaitu sebesar 0.979 untuk semua kelas seperti yang terlihat pada tabel 4.8



Oleh karena itu, model yang dihasilkan dari pelatihan tersebut dipilih untuk menjadi model sistem.

Berdasarkan proses pelatihan model pada epoch 200 dan *learning rate* 0.01 disajikan hasil evaluasi dari model yang berhasil dibangun. Grafik yang dihasilkan memberikan gambaran visual tentang performa model secara menyeluruh. Grafik tersebut mencakup beberapa metrik penting seperti *box loss*, *class loss*, dan *dfl loss*, serta metrik evaluasi kinerja seperti *precision* dan *recall* pada gambar 4.13 .



**Gambar 4.13** Grafik Performa Model

Secara umum, grafik menunjukkan bahwa model memiliki performa yang memuaskan dengan metrik seperti *box\_loss*, *class\_loss*, *dfl\_loss*, *precision*, *recall*, dan mAP menunjukkan kinerja yang baik. Ketika kita melakukan evaluasi hasil untuk setiap kelas secara spesifik, terlihat adanya variasi yang tidak jauh berbeda dengan kelas lainnya. Dapat kita lihat nilai hasil yang relatif mirip pada nilai mAP. *Mean Average Precision* atau mAP adalah metrik penting yang memberikan gambaran tentang seberapa baik model dapat mengenali objek dalam semua kelas.

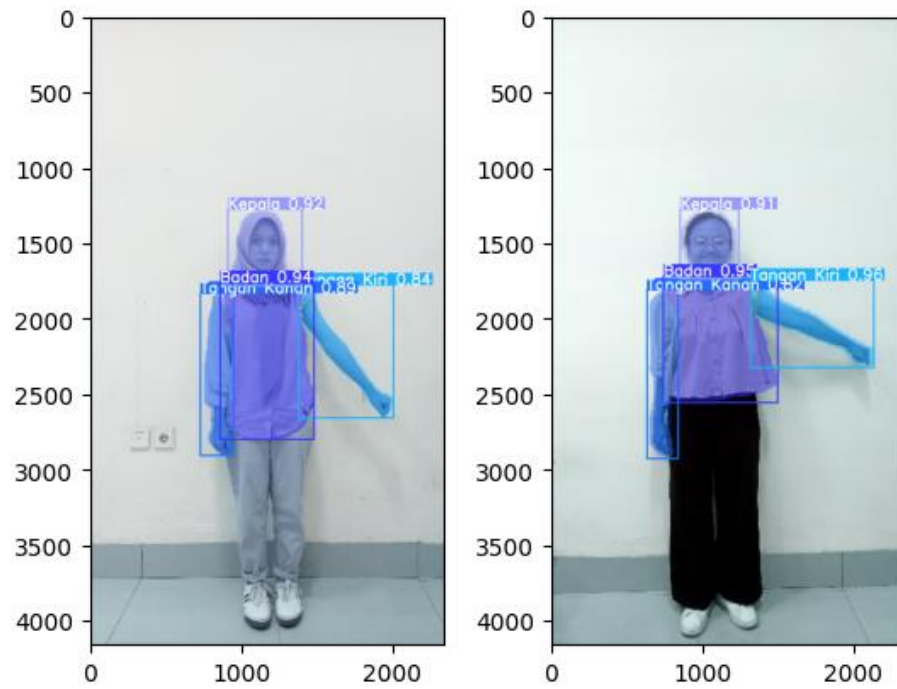
Setelah melakukan *tuning hyperparameter* dan didapatkan nilai mAP tertinggi, diambil nilai *precision* dan *recall* dari hasil *training* model tersebut, langkah selanjutnya adalah mengevaluasi performa model menggunakan metrik evaluasi tambahan seperti *F1 Score*. *F1 Score* memberikan gambaran yang lebih komprehensif tentang kinerja model klasifikasi. Adapun rumus perhitungan *F1 Score* adalah sebagai berikut:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

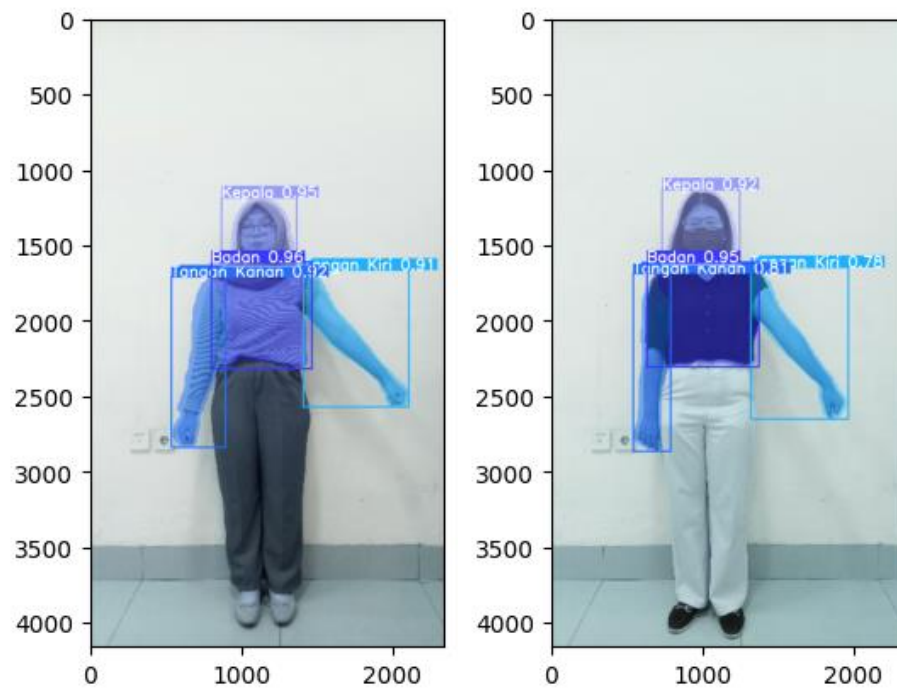
$$F1 = 2 \times \frac{0.914 \times 0.995}{0.914 + 0.995}$$

$$F1 = 2 \times \frac{0.9094}{1.909}$$

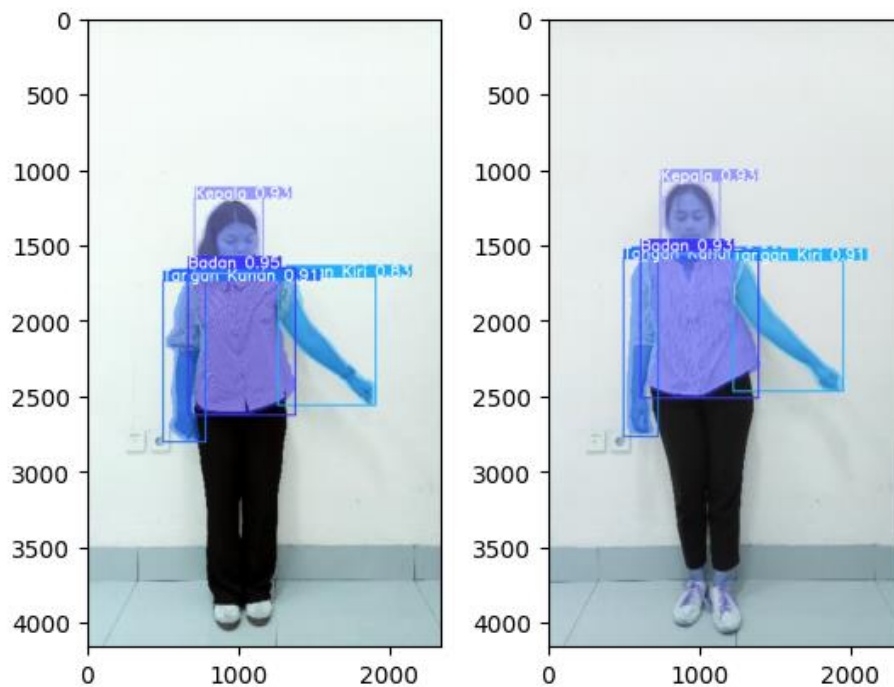
$$F1 = 0.95$$



**Gambar 4.14** Pengujian Model pada Google Colab (1)



**Gambar 4.15** Pengujian Model pada Google Colab (2)



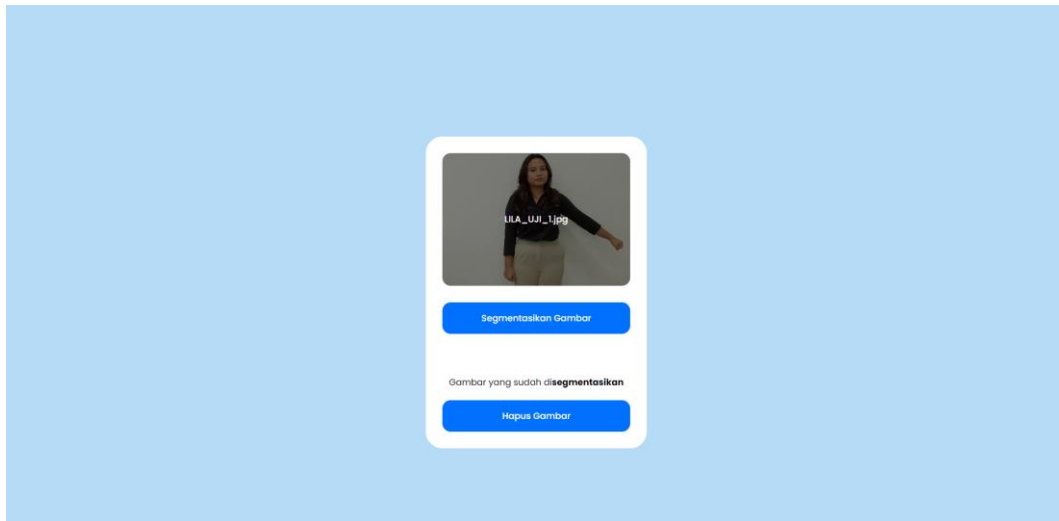
**Gambar 4.16** Pengujian Model pada Google Colab (3)

Hasil pengujian model menunjukkan bahwa kemampuan model dalam melakukan segmentasi sangat baik. Hasil pengujian ini juga menunjukkan bahwa model mampu dengan akurat mengidentifikasi dan memisahkan objek-objek dalam gambar dengan tepat. Dari hasil ini memungkinkan model untuk memahami batas-batas objek dengan lebih tepat, sehingga memungkinkan untuk pengenalan objek yang lebih akurat. Proses segmentasi merupakan langkah penting dalam pengolahan citra komputer yang memungkinkan model untuk memahami konteks spasial objek dalam suatu gambar. Kemampuan segmentasi yang baik menandakan bahwa model dapat menghasilkan hasil yang lebih presisi dan relevan dalam mengenali area dari gambar yang berkaitan dengan objek yang ditentukan.

#### **4.4 Implementasi Model kedalam Web**

Dalam penelitian ini, aplikasi dikembangkan menggunakan metode deteksi objek YOLOv8. Aplikasi ini dirancang dengan dua bagian utama, yakni bagian *frontend* dan *backend*. Bagian *backend* dikembangkan dengan *framework* Flask sementara bagian *frontend* dengan HTML, CSS, dan JavaScript. Kemudian aplikasi dijalankan pada server lokal.

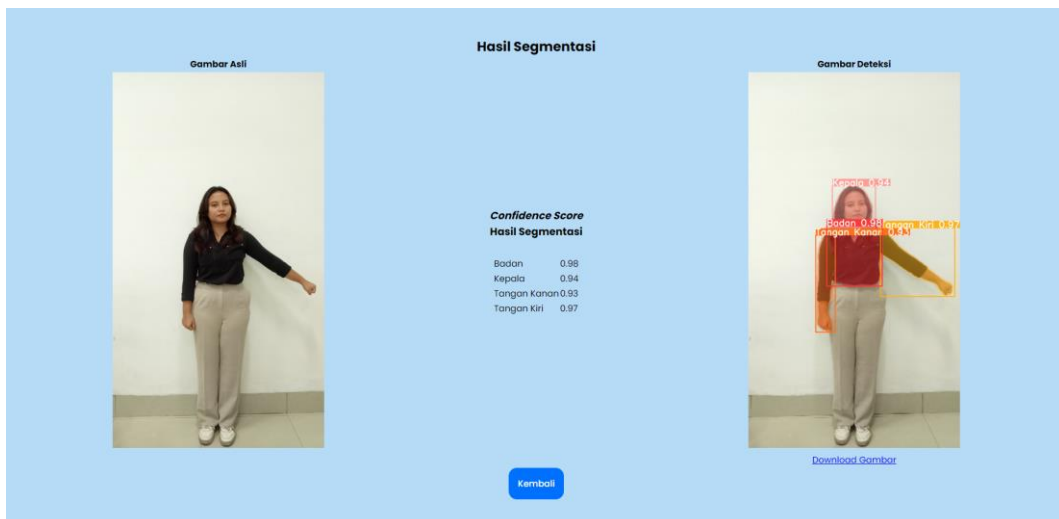
#### 4.4.1 Halaman Utama



**Gambar 4.17** Tampilan Utama (Beranda)

Gambar 4.16 merupakan tampilan utama website, dimana pada halaman ini memungkinkan untuk pengguna melakukan segmentasi gambar dengan memilih gambar mana yang akan dilakukan segmentasi. Selain itu juga memungkinkan pengguna untuk mengunduh ulang gambar yang sudah disegmentasikan ataupun menghapus gambar yang sudah disegmentasikan.

#### 4.4.2 Halaman Hasil



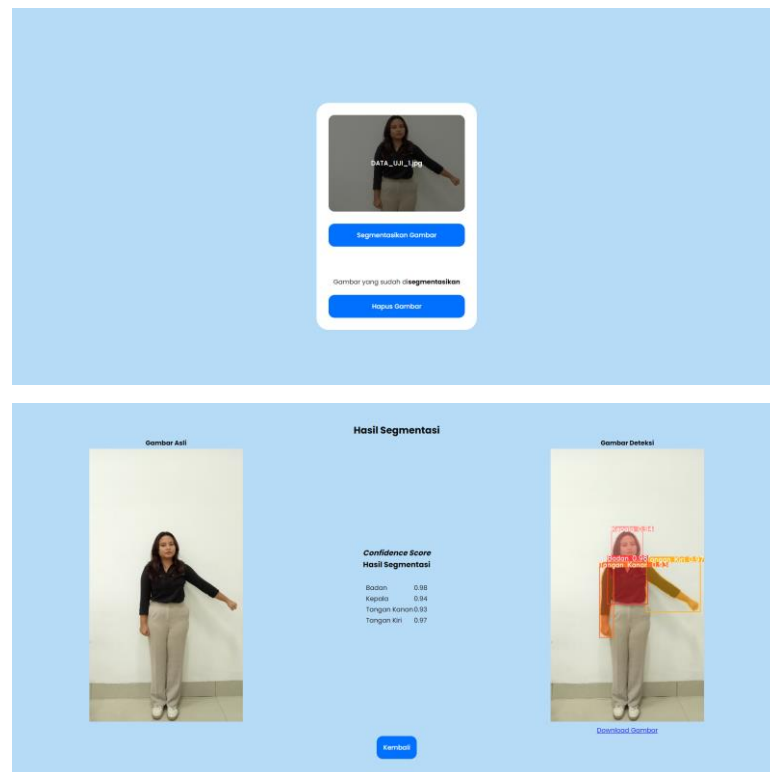
**Gambar 4.18** Tampilan Hasil

Gambar 4.17 merupakan tampilan hasil gambar yang telah dilakukan segmentasi. Pada bagian kiri halaman ini terdapat gambar asli yang diunggah oleh pengguna, dan pada bagian kanan nya terdapat gambar yang telah dilakukan

segmentasi atau gambar hasil segmentasi beserta tombol untuk mendownload gambar hasil segmentasi tersebut. Pada bagian tengah nya terdapat nilai hasil prediksi. Dan pada bagian bawahnya terdapat tombol untuk kembali ke halaman utama.

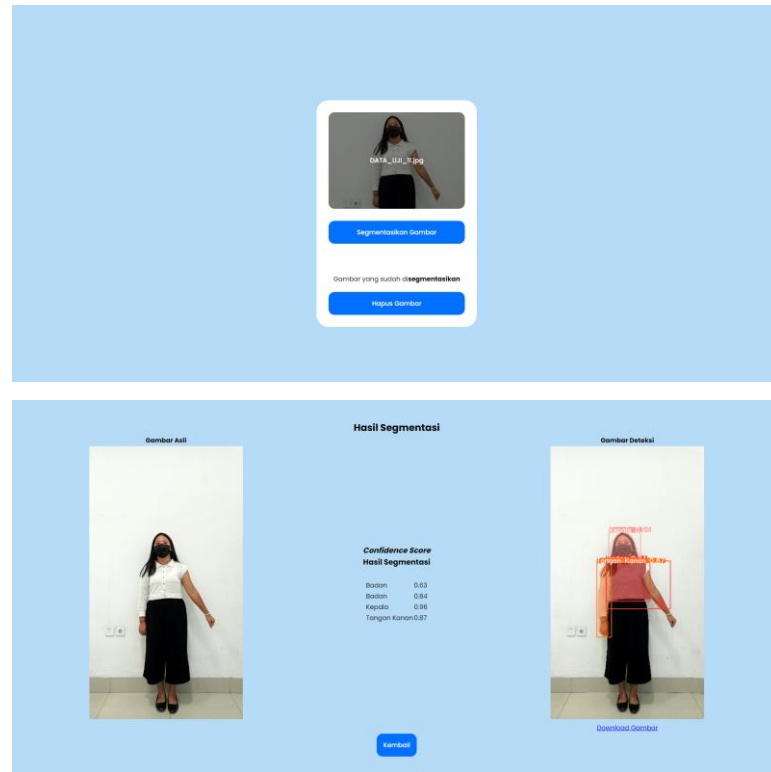
#### 4.5 Pengujian Sistem

Pengujian sistem dimaksudkan guna memeriksa keandalan sistem yang sudah dikembangkan dalam mengenali bagian tubuh manusia secara akurat. Selama tahap pengujian, kami akan menggunakan serangkaian 30 gambar yang menampilkan berbagai bagian tubuh manusia. Tujuan dari penggunaan gambar-gambar ini adalah untuk memverifikasi kemampuan sistem dalam mengidentifikasi dengan tepat bagian tubuh yang akan disegmentasi.



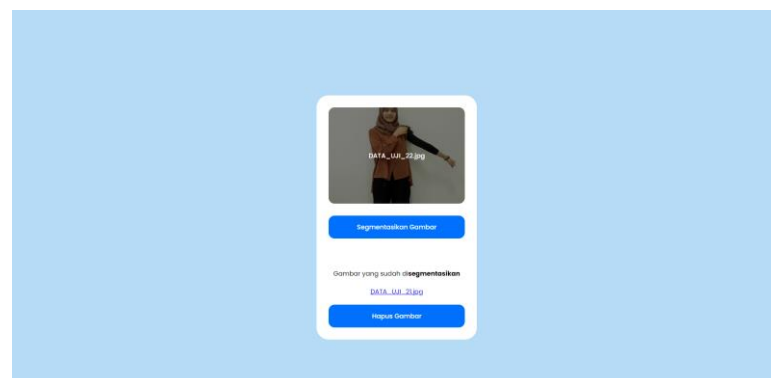
**Gambar 4.19** Hasil Uji Sistem Data No. 1

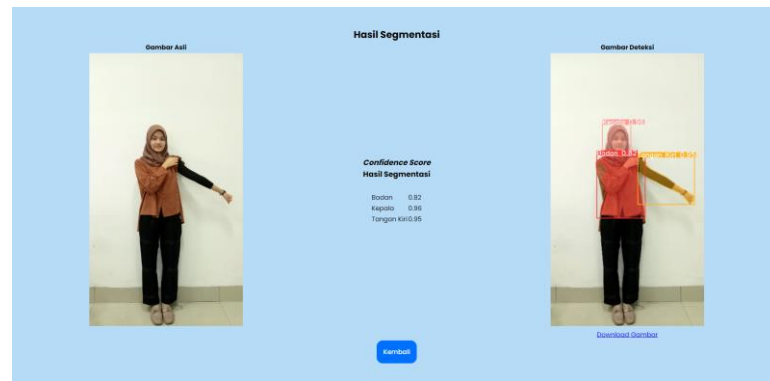
Pada Gambar 4.19, model dapat melakukan segmentasi pada *image* dengan sangat baik, terlihat dengan adanya *confidence score* yang tinggi.



**Gambar 4.20** Hasil Uji Sistem Data No. 11

Pada Gambar 4.20, model mengalami kesulitan dalam melakukan segmentasi terhadap tangan kiri. Hal ini disebabkan oleh posisi tangan kiri yang kurang diangkat, kurangnya batas antara tangan kiri dan badan membuatnya sulit untuk model membedakan antara tangan dan badan. Akibatnya, model mengidentifikasi tangan kiri sebagai bagian dari badan dengan nilai kepercayaan yang rendah.





**Gambar 4.21** Hasil Uji Sistem Data No. 22

Pada Gambar 4.21, model mengalami kesulitan dalam melakukan segmentasi terhadap tangan kanan. Hal ini disebabkan oleh posisi tangan kanan yang menempel pada bagian area badan, membuatnya sulit untuk model membedakan antara tangan dan badan. Akibatnya, model mengidentifikasi tangan kanan sebagai bagian dari badan.

#### 4.6 Hasil Pengujian Sistem

Berikut adalah hasil yang diperoleh dari melakukan segmentasi pada semua gambar yang telah diuji:

**Tabel 4.9** Hasil Pengujian Sistem

No	Nama	Kelas Sebenarnya	Hasil Uji	Confidence Score
1	DATA_UJI_1	Badan	Badan	0.98
		Kepala	Kepala	0.94
		Tangan Kanan	Tangan Kanan	0.93
		Tangan Kiri	Tangan Kiri	0.97
2	DATA_UJI_2	Badan	Badan	0.99
		Kepala	Kepala	0.93
		Tangan Kanan	Tangan Kanan	0.92
		Tangan Kiri	Tangan Kiri	0.95
3	DATA_UJI_3	Badan	Badan	0.89
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.93
		Tangan Kiri	Tangan Kiri	0.98
4	DATA_UJI_4	Badan	Badan	0.85
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.93
		Tangan Kiri	Tangan Kiri	0.99

5	DATA_UJI_5	Badan	Badan	0.98
		Kepala	Kepala	0.93
		Tangan Kanan	Tangan Kanan	0.9
		Tangan Kiri	Tangan Kiri	0.98
6	DATA_UJI_6	Badan	Badan	0.96
		Kepala	Kepala	0.93
		Tangan Kanan	Tangan Kanan	0.97
		Tangan Kiri	Tangan Kiri	0.99
7	DATA_UJI_7	Badan	Badan	0.9
		Kepala	Kepala	0.94
		Tangan Kanan	Tangan Kanan	0.87
		Tangan Kiri	Tangan Kiri	0.93
8	DATA_UJI_8	Badan	Badan	0.98
		Kepala	Kepala	0.97
		Tangan Kanan	Tangan Kanan	0.92
		Tangan Kiri	Tangan Kiri	0.96
9	DATA_UJI_9	Badan	Badan	0.98
		Kepala	Kepala	0.97
		Tangan Kanan	Tangan Kanan	0.99
		Tangan Kiri	Tangan Kiri	0.89
10	DATA_UJI_10	Badan	Badan	0.84
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.83
		Tangan Kiri	Tangan Kiri	0.86
11	DATA_UJI_11	Badan	Badan	0.63
		Kepala	Kepala	0.96
		Tangan Kanan	Tangan Kanan	0.87
		Tangan Kiri	Tidak Terdeteksi	0
12	DATA_UJI_12	Badan	Badan	0.97
		Kepala	Kepala	0.94
		Tangan Kanan	Tangan Kanan	0.96
		Tangan Kiri	Tangan Kiri	0.93
13	DATA_UJI_13	Badan	Badan	0.95
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.92
		Tangan Kiri	Tangan Kiri	0.9
14	DATA_UJI_14	Badan	Badan	0.99
		Kepala	Kepala	0.94
		Tangan Kanan	Tangan Kanan	0.86
		Tangan Kiri	Tangan Kiri	0.92
15	DATA_UJI_15	Badan	Badan	0.88
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.95



		Tangan Kiri	Tangan Kiri	0.97
16	DATA_UJI_16	Badan	Badan	0.93
		Kepala	Kepala	0.97
		Tangan Kanan	Tangan Kanan	0.69
		Tangan Kiri	Tangan Kiri	0.91
17	DATA_UJI_17	Badan	Badan	0.87
		Kepala	Kepala	0.93
		Tangan Kanan	Tangan Kanan	0.9
		Tangan Kiri	Tangan Kiri	0.89
18	DATA_UJI_18	Badan	Badan	0.97
		Kepala	Kepala	0.98
		Tangan Kanan	Tangan Kanan	0.96
		Tangan Kiri	Tangan Kiri	0.88
19	DATA_UJI_19	Badan	Badan	0.97
		Kepala	Kepala	0.93
		Tangan Kanan	Tangan Kanan	0.91
		Tangan Kiri	Tangan Kiri	0.59
20	DATA_UJI_20	Badan	Badan	0.98
		Kepala	Kepala	0.92
		Tangan Kanan	Tangan Kanan	0.92
		Tangan Kiri	Tangan Kiri	0.94
21	DATA_UJI_21	Badan	Badan	0.85
		Kepala	Kepala	0.96
		Tangan Kanan	Tangan Kanan	0.9
		Tangan Kiri	Tangan Kiri	0.84
22	DATA_UJI_22	Badan	Badan	0.82
		Kepala	Kepala	0.96
		Tangan Kanan	Tidak Terdeteksi	0
		Tangan Kiri	Tangan Kiri	0.95
23	DATA_UJI_23	Badan	Badan	0.98
		Kepala	Kepala	0.98
		Tangan Kanan	Tangan Kanan	0.92
		Tangan Kiri	Tangan Kiri	0.77
24	DATA_UJI_24	Badan	Badan	0.99
		Kepala	Kepala	0.96
		Tangan Kanan	Tangan Kanan	0.9
		Tangan Kiri	Tangan Kiri	0.87
25	DATA_UJI_25	Badan	Badan	0.87
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.89
		Tangan Kiri	Tangan Kiri	0.96
26	DATA_UJI_26	Badan	Badan	0.81
		Kepala	Kepala	0.96

		Tangan Kanan	Tangan Kanan	0.5
		Tangan Kiri	Tangan Kiri	0.99
27	DATA_UJI_27	Badan	Badan	0.96
		Kepala	Kepala	0.92
		Tangan Kanan	Tangan Kanan	0.93
		Tangan Kiri	Tangan Kiri	0.97
28	DATA_UJI_28	Badan	Badan	0.93
		Kepala	Kepala	0.91
		Tangan Kanan	Tangan Kanan	0.89
		Tangan Kiri	Tangan Kiri	0.96
29	DATA_UJI_29	Badan	Badan	0.96
		Kepala	Kepala	0.95
		Tangan Kanan	Tangan Kanan	0.87
		Tangan Kiri	Tangan Kiri	0.93
30	DATA_UJI_30	Badan	Badan	0.94
		Kepala	Kepala	0.96
		Tangan Kanan	Tangan Kanan	0.95
		Tangan Kiri	Tangan Kiri	0.96

Berdasarkan hasil pengujian di atas, dapat dilihat bahwa model sangat baik dalam mengenali objek pada gambar. Hasil ini didukung juga dengan hasil *confidence score* yang sangat baik dari model. *Confidence score* adalah ukuran yang menunjukkan tingkat keyakinan atau kepastian dalam hasil yang diberikan oleh suatu sistem atau model prediksi. Skor ini biasanya dinyatakan dalam bentuk angka antara 0 dan 1, atau dalam bentuk persentase antara 0% dan 100%. Semakin tinggi skor kepercayaan, semakin yakin sistem atau model terhadap hasil atau prediksi yang dibuat.

Pada hasil pengujian dengan data uji ini, kita juga dapat melihat beberapa kekurangan atau tantangan yang dihadapi oleh model. Salah satu tantangan utama adalah dalam mengenali bagian tubuh yang tumpang tindih dengan bagian tubuh lainnya. Hal ini terlihat jelas pada data nomor 11 dan 22. Ketika bagian-bagian tubuh saling tumpang tindih, model mengalami kesulitan untuk mengenali dan mengidentifikasi setiap bagian secara akurat.

Kesulitan ini mungkin disebabkan oleh kompleksitas visual yang meningkat ketika objek saling tumpang tindih, sehingga fitur-fitur penting menjadi kurang jelas atau terdistorsi. Dalam kasus ini, meskipun *confidence score* secara

keseluruhan baik, skor untuk objek-objek yang tumpang tindih mungkin lebih rendah atau bahkan tidak terprediksi, menunjukkan bahwa model kurang yakin dalam prediksinya.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berikut merupakan kesimpulan penelitian menggunakan *instance segmentation* dengan YOLOv8 dalam penguraian bagian tubuh manusia.

1. *Instance segmentation* dengan YOLOv8 dapat digunakan dalam menguraikan dan mensegmentasi bagian tubuh manusia.
2. Dilakukan pengujian model dan didapatkan nilai mAP tertinggi yaitu sebesar 0.979 dengan jumlah 200 *epoch* pada *training* data set, 0.914 nilai *precision*, 0.995 nilai *recall* dan 0.95 F1-score. Hal ini memungkinkan sistem mampu mensegmentasi gambar dengan tingkat 95%.
3. Dari hasil pengujian model tersebut, bagian tubuh dapat disegmentasikan dengan baik. Hal ini menunjukkan bahwa *instance segmentation* dengan YOLOv8 dapat digunakan dalam segmentasi bagian tubuh manusia
4. Model akan mengalami kesulitan dalam mengenali bagian tubuh yang tumpang tindih dengan bagian tubuh lainnya, seperti yang terlihat pada uji sistem dengan data uji no 11 dan 22.

#### **5.2 Saran**

Berikut adalah beberapa usulan yang dapat dijadikan saran untuk dipertimbangkan dalam penelitian selanjutnya:

1. Melakukan pengumpulan data yang lebih luas dan representatif dari berbagai posisi bagian tubuh manusia untuk meningkatkan keberagaman dataset yang digunakan dalam pelatihan model.
2. Melakukan pengembangan lebih lanjut untuk terus meningkatkan kinerja model agar dapat memberikan hasil yang lebih akurat dan berguna dalam mensegmentasi bagian tubuh manusia.
3. Melakukan perbandingan *instance segmentation* model YOLO dengan model algoritma lainnya untuk perbandingan hasil segmentasi bagian tubuh manusia yang lebih baik dan akurat

4. Diharapkan kedepannya dapat menjadi dasar untuk mengembangkan aplikasi yang membutuhkan segmentasi tubuh manusia seperti pengenalan bagian tubuh atau pemantauan medis.

## DAFTAR PUSTAKA

- Abdul Hadi, M., Ferdian, R., & Arief, L. (2021). Klasifikasi Tingkat Ancaman Kriminalitas Bersenjata Menggunakan Metode You Only Look Once (YOLO). *CHIPSET*, 2(01), 33–40. <https://doi.org/10.25077/chipset.2.01.33-40.2021>
- Pratama, E. F. A., Khairil, K., & Jumadi, J. (2022). Implementasi Metode K-Means Clustering Pada Segmentasi Citra Digital. *Jurnal Media Infotama*, 18(2), 291–301. <https://doi.org/10.37676/jmi.v18i2.2899>
- Arifianto, J. (2022). *Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Model Deep Learning Dengan Tensorflow*. Universitas Islam Indonesia.
- Arnita, Marpaung, F., Aulia, F., Suryani, N., & Nabila, R. C. (2022). *Computer Vision dan Pengolahan Citra Digital* (A. B. Ahmad, Ed.; 1st ed.). Pustaka Akasara.
- Cai, W., Xiong, Z., Sun, X., Rosin, P. L., Jin, L., & Peng, X. (2020). Panoptic Segmentation-Based Attention for Image Captioning. *Applied Sciences*, 10(1), 391. <https://doi.org/10.3390/app10010391>
- Chuang, Y., Zhang, S., & Zhao, X. (2023). Deep learning-based panoptic segmentation: Recent advances and perspectives. *IET Image Processing*, 17(10), 2807–2828. <https://doi.org/10.1049/ipr2.12853>
- Drantantiyas, N. D. G., Yulita, W., Ridwan, N. T., Ramadhani, U. A., Kesuma, R. I., Rakhman, A. Z., Bagaskara, R., Miranto, A., & Mufidah, Z. (2023). Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8. *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika Dan Komputer)*, 5(2), 63–68. <https://doi.org/10.26905/jasiek.v5i2.11605>
- Hassanudin, W. M., Utomo, V. G., & Apriyanto, R. (2024). Fine-Grained Analysis of Coral Instance Segmentation using YOLOv8 Models. *Sinkron*, 8(2), 1047–1055. <https://doi.org/10.33395/sinkron.v8i2.13583>
- He, H., Zhuang, B., Zhang, J., Cai, J., & Tao, D. (2021). *End-to-end One-shot Human Parsing*. <http://arxiv.org/abs/2105.01241>
- Helnawan, A., Attamimi, M., & Irfansyah, A. N. (2023). Sistem Segmentasi Jalan dan Objek untuk Kendaraan Otonom Menggunakan Kamera RGB-D. *Jurnal Teknik ITS*, 12(1), A53–A62.
- Heryana, N., Mayasari, R., & Baihaqi, K. A. (2020). Penerapan Haar Cascade Classification Model untuk Deteksi Wajah, Hidung, Mulut, dan Mata Menggunakan Algoritma Viola-Jones. *Techno Xplore: Jurnal Ilmu Komputer Dan Teknologi Informasi*, 5(1), 21–25.

- Hurtik, P., Molek, V., Hula, J., Vajgl, M., Vlasanek, P., & Nejezchleba, T. (2020). *Poly-YOLO: higher speed, more precise detection and instance segmentation for YOLOv3*.
- Komalasari, R. (2022). Pemanfaatan Kecerdasan Buatan (Artificial Intelligence) Dalam Telemedicine: Dari Perspektif Profesional Kesehatan. *J. Ked. Mulawarman*, 9(2), 72–81.
- Kosudiwandi, N. R. P. (2020). *Implementasi Algoritma You Only Look Once (Yolo) Untuk Mendeteksi Pelanggaran Rambu Dilarang Parkir*. Universitas Komputer Indonesia.
- Kristal, A., & Harintaka. (2022). Analisis Keandalan Ekstraksi Garis Tepi Bangunan dari Data Foto Udara Menggunakan Pendekatan Deep Learning Berbasis Mask R-CNN. *Journl of Geodesy and Geomatics*, 17(2), 273–285.
- Lee, M.-F. R., Chen, Y.-C., & Tsai, C.-Y. (2022). Deep Learning-Based Human Body Posture Recognition and Tracking for Unmanned Aerial Vehicles. *Processes*, 10(11), 2295. <https://doi.org/10.3390/pr10112295>
- Mawarni, D. I., Indarto, Deendarlianto, & Yuana, K. A. (2023). Metode Digital Image Prosessing untuk Menentukan Distribusi Ukuran Diameter Gelembung Udara Pada Microgelembung Generator. *Journal of Information System Management (JOISM)*, 4(3), 132–136.
- Mohanapriya, S., Mohana Saranya, S., Kumaravel, T., & Sumithra, P. (2023). Image Detection and Segmentation using YOLO v5 for surveillance. *Applied and Computational Engineering*, 8(1), 142–147. <https://doi.org/10.54254/2755-2721/8/20230109>
- Motwani, N. P., & S, S. (2023). Human Activities Detection using DeepLearning Technique- YOLOv8. *ITM Web of Conferences*, 56, 03003. <https://doi.org/10.1051/itmconf/20235603003>
- Oktaviani, N. H. (2023). *Pengembangan Alat Deteksi Kek Pada Ibu Hamil Dan Rekomendasinya Memanfaatkan Kecerdasan Buatan* [Poltekkes Kemenkes Semarang]. [https://repository.poltekkes-smg.ac.id/?p=show\\_detail&id=36439](https://repository.poltekkes-smg.ac.id/?p=show_detail&id=36439)
- Panjaitan, N. A. (2020). Segmentasi Citra Digital Dengan Menggunakan Algoritma Khutlang. *JOSH: Jurnal Penelitian Sistem Informasi*, 1(4).
- Rundo, L., Pirrone, R., Vitabile, S., Sala, E., & Gambino, O. (2020). Recent advances of HCI in decision-making tasks for optimized clinical workflows and precision medicine. *Journal of Biomedical Informatics*, 108, 103479. <https://doi.org/10.1016/j.jbi.2020.103479>
- Sampurno, R. M., Liu, Z., Abeyrathna, R. M. R. D., & Ahamed, T. (2024). Intrarow Uncut Weed Detection Using You-Only-Look-Once Instance

- Segmentation for Orchard Plantations. *Sensors*, 24(3), 893. <https://doi.org/10.3390/s24030893>
- Sudibjo, P., Noerhadi, M., Sunardi, J., Sukamti, E. R., Purnomo, E., Prihartanti, H., Prasetyo, Y., & Wicaksono, D. (2011). *Diktat Anatomi Manusia* (P. Sudibjo, M. Noerhadi, & J. Sunardi, Eds.). Laboratorium Anatomi FIK UNY.
- Viana, F. X., Araujo, G. M., Pinto, M. F., Colares, J., & haddad, D. B. (2020). Aerial Image Instance Segmentation Through Synthetic Data Using Deep Learning. *Learning and Nonlinear Models*, 18(1), 35–46. <https://doi.org/10.21528/lnlm-vol18-no1-art3>
- Vinh, T. Q., & Byeon, H. (2023). Enhancing Alzheimer's Disease Diagnosis: The Efficacy of the YOLO Algorithm Model. *International Journal of Advanced Computer Science and Applications*, 14(11). <https://doi.org/10.14569/IJACSA.2023.0141182>
- Wang, G., Chen, Y., An, P., Hong, H., Hu, J., & Huang, T. (2023). UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors*, 23(16), 7190. <https://doi.org/10.3390/s23167190>