

**IDENTIFIKASI LUKA RINGAN DENGAN YOLO-CNN PADA CITRA  
DIGITAL UNTUK MANAJEMEN  
PERAWATAN LUKA**

**SKRIPSI**

**JIMMI EDUARD SIMANGUNSONG  
201401106**



**PROGRAM STUDI S1-ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**IDENTIFIKASI LUKA RINGAN DENGAN YOLO-CNN PADA CITRA  
DIGITAL UNTUK MANAJEMEN  
PERAWATAN LUKA**

**SKRIPSI**

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

**JIMMI EDUARD SIMANGUNSONG**

**201401106**



**PROGRAM STUDI S1-ILMU KOMPUTER  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA**

**MEDAN**

**2024**

**PERSETUJUAN**

Judul : IDENTIFIKASI LUKA RINGAN DENGAN  
YOLO-CNN PADA CITRA DIGITAL UNTUK  
MANAJEMEN PERAWATAN LUKA

Kategori : SKRIPSI

Nama : JIMMI EDUARD SIMANGUNSONG

Nomor Induk Mahasiswa : 201401106

Program Studi : SARJANA (S-1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI  
INFORMASI UNIVERSITAS SUMATERA  
UTARA

Tanggal Sidang : Medan, 29 April 2024

Komisi Pembimbing :

Pembimbing 2

Pembimbing 1





Pauzi Ibrahim Nainggolan S.Komp., M.Sc.  
NIP. 198809142020011001



Fuzy Yustika Manik S.Kom., M.Kom  
NIP. 198710152019032010

Diketahui/disetujui oleh  
Program Studi S1 Ilmu Komputer

Ketua

Dr. Amalia S.T., M.T.  
NIP. 197812212014042001

**PERNYATAAN**

**IDENTIFIKASI LUKA RINGAN DENGAN YOLO-CNN PADA CITRA  
DIGITAL UNTUK MANAJEMEN  
PERAWATAN LUKA**

**SKRIPSI**

Saya mengakui bahwa skripsi ini merupakan hasil karya pribadi saya sendiri, kecuali beberapa kutipan dan ringkasan yang telah saya beri sumbernya secara jelas

Medan, 29 April 2024

A handwritten signature in black ink, reading "Jimmi Eduard S". The signature is stylized with a large initial 'J' and a trailing 'S'.

Jimmi Eduard Simangunsong  
201401106

## **PENGHARGAAN**

Dengan penuh rasa syukur, saya mengucapkan terima kasih kepada Tuhan Yesus Kristus atas berkat dan kasih karunia-Nya yang telah memungkinkan saya menyelesaikan penelitian ini. Penulisan skripsi ini merupakan bagian dari perjalanan saya menuju gelar Sarjana Komputer di Program Studi S-1 Ilmu Komputer Universitas Sumatera Utara.

Saya ingin mengungkapkan rasa terima kasih yang mendalam kepada:

1. Bapak Prof. Dr. Muryanto Amin, S.Sos, M.Si., selaku Rektor Universitas Sumatera Utara
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
3. Ibu Dr. Amalia, ST. MT., selaku Kepala Program Studi S-1 Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.
4. Ibu Sri Melvani Hardi S.Kom., M.Kom., sebagai Sekretaris Program Studi S1 Ilmu Komputer Universitas Sumatera Utara
5. Ibu Dian Rachmawati S.Si., M.Kom., selaku Dosen Pembimbing Akademik yang telah memberikan bimbingan, nasihat, dan dorongan kepada kami sebagai mahasiswa dalam melalui tahap-tahap akademik dari masa kuliah hingga penulisan skripsi.
6. Ibu Fuzy Yustika Manik S.Kom., M.Kom., sebagai dosen pembimbing I yang telah memberikan panduan, petunjuk, serta semangat yang membara bagi penulis dalam menuntaskan skripsi ini.
7. Pauzi Ibrahim Nainggolan S.Komp., M.Sc., sebagai dosen pembimbing II yang telah memberikan bimbingan, arahan, dukungan serta masukan kepada penulis.
8. Seluruh Dosen Program Studi S-1 Ilmu Komputer yang dengan senang hati memberikan waktu dan tenaga dalam mengajar dan membimbing sehingga penulis dapat sampai kepada tahap penyusunan skripsi ini.
9. Teristimewa Saya ingin menyampaikan penghargaan yang tulus kepada kedua orang tua saya, Bapak Jondari Sahala Tua Simangunsong dan Ibu Sinta Agustina Simorangkir, yang selalu memberikan dukungan, doa, dan menjadi pendorong utama bagi saya dalam setiap langkah perjalanan ini.

10. Abang kandung Martin Ricardo Simangunsong, adik kandung Burju Orlando Simangunsong dan Marcel Simangunsong yang selalu mendukung serta mendoakan penulis dalam menjalankan aktivitas perkuliahan.
11. Bangkit Academy 2023 yang memberikan tempat belajar machine learning dan cloud computing serta softskill lainnya yang sangat berguna dalam menyusun skripsi penulis.
12. Teman-teman Kost Tentrem, Irfan, Yogi sebagai teman belajar hal baru, tempat berdiskusi bersama.
13. Rekan-rekan seangkatan dalam perkuliahan Kom C 2020 yang telah menemani dan berbagi pengalaman selama empat tahun masa studi.
14. Rekan-rekan Capstone Project Bangkit Academy 2023 batch 2, Yosef, Yordan, Audry, Michael, Ikin yang memberikan masukan berharga dan semangat yang tak tergantikan kepada penulis.
15. Terakhir namun tidak kalah pentingnya, saya ingin mengucapkan terima kasih atas perjuangan yang telah saya lakukan sampai saat ini.

Penulis menyadari bahwa ada kekurangan dalam penelitian ini, dan dengan tulus mengharapkan masukan yang konstruktif untuk meningkatkan kualitas penelitian ini. Penulis ingin mengakhiri dengan ungkapan terima kasih, haranya penelitian ini bermanfaat bagi semua pihak yang membutuhkan nya.

Medan, 29 April 2024



Jimmi Eduard Simangunsong  
201401106

## ABSTRAK

Luka ringan merupakan masalah umum dalam kehidupan sehari-hari yang sering dianggap remeh, namun penanganan yang tepat terhadap luka ringan sangat penting untuk mencegah komplikasi yang lebih serius. Penelitian ini bertujuan untuk mengimplementasikan Metode deteksi objek *You Only Look Once* (YOLO) yang memanfaatkan arsitektur *Convolutional Neural Networks* (CNN) dalam identifikasi luka ringan pada citra digital, serta menyediakan informasi relevan mengenai manajemen perawatan luka seperti penanganan awal dan rekomendasi obat yang sesuai. Penelitian ini melibatkan tahapan studi pustaka untuk memahami konsep dasar *Machine Learning* dan YOLOv8, pengumpulan dataset citra digital yang mencakup berbagai jenis luka ringan, *preprocessing* data untuk menormalkan dan memperluas keragaman dataset, pengembangan model YOLOv8 untuk identifikasi luka ringan, evaluasi kinerja model, pengembangan aplikasi berbasis citra digital untuk identifikasi luka ringan, dan evaluasi keseluruhan aplikasi. Pengujian dilakukan terhadap model aplikasi dengan memanfaatkan metode evaluasi *confusion matrix*, dengan data uji sebanyak 49 gambar luka ringan didapatkan akurasi sebesar 85 %.

**Kata kunci:** *Convolutional Neural Network, You Only Look Once Version 8*, Luka Ringan, Manajemen Perawatan Luka

## **Minor Wounds Identification With YOLO-CNN On Digital Images For Wound Care Management**

### **ABSTRACT**

Minor wounds are a common issue in everyday life often overlooked, yet proper management of minor wounds is crucial to prevent more serious complications. This research aims to implement the You Only Look Once (YOLO) object detection method utilizing Convolutional Neural Networks (CNN) architecture in identifying minor wounds in digital images, while also providing relevant information regarding wound care management such as initial treatment and suitable medication recommendations. This study involves literature review stages to understand the basic concepts of Machine Learning and YOLOv8, collecting a dataset of digital images covering various types of minor wounds, data preprocessing to normalize and enhance dataset diversity, developing the YOLOv8 model for minor wound identification, evaluating model performance, developing a digital image-based application for minor wound identification, and overall application evaluation. Testing was conducted on the application model utilizing the confusion matrix evaluation method, with a test data set consisting of 49 images of minor wounds. An accuracy of 85% was obtained.

**Keywords:** Convolutional Neural Network, You Only Look Once Version 8, Minor Wounds, Wound Care Management



## DAFTAR ISI

<b>SKRIPSI.....</b>	<b>i</b>
<b>SKRIPSI.....</b>	<b>ii</b>
<b>PERSETUJUAN .....</b>	<b>iii</b>
<b>PERNYATAAN.....</b>	<b>iv</b>
<b>PENGHARGAAN.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiv</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Metodologi Penelitian.....	5
1.7 Penelitian Relevan .....	6
1.8 Sistematika Penulisan .....	8
<b>BAB 2 LANDASAN TEORI .....</b>	<b>9</b>
2.1 Luka .....	9
2.2 Luka Ringan.....	9
2.2.1 Luka Tusuk.....	10
2.2.2 Luka Sayat .....	10
2.2.3 Luka Lecet .....	11
2.2.4 Luka Bakar .....	11
2.3 Manajemen Perawatan Luka.....	12
2.4 <i>Convolutional Neural Network (CNN)</i> .....	12
2.4.1 <i>Kernel atau Filter atau Detektor Fitur</i> .....	13
2.4.2 <i>Stride</i> .....	14

2.4.3	<i>Pooling</i>	14
2.4.4	<i>Flatten</i>	15
2.5	<i>YOLO (You Only Look Once)</i>	15
2.6	<i>Pengukuran Evaluasi Kinerja</i>	17
<b>BAB 3 ANALISIS DAN PERANCANGAN</b>		<b>19</b>
3.1	<i>Analisis Sistem</i>	19
3.1.1	<i>Analisis Masalah</i>	19
3.1.2	<i>Analisis Kebutuhan</i>	20
3.2	<i>Arsitektur Umum</i>	21
3.3	<i>Pemodelan Sistem</i>	22
3.3.1	<i>Use Case Diagram</i>	22
3.3.2	<i>Activity Diagram</i>	23
3.3.3	<i>Sequence Diagram</i>	24
3.4	<i>Flowchart (Diagram Alir)</i>	25
3.4.1	<i>Flowchart Sistem</i>	26
3.4.2	<i>Flowchart Deteksi Object YOLO</i>	27
3.5	<i>Perancangan Interface</i>	28
3.5.1	<i>Desain Halaman Login (Masuk dan Daftar)</i>	28
3.5.2	<i>Desain Beranda Aplikasi</i>	29
3.5.3	<i>Desain Artikel Kesehatan</i>	30
3.5.4	<i>Desain Artikel Batasan Luka</i>	31
3.5.5	<i>Desain Halaman Scan</i>	32
3.5.6	<i>Desain Halaman Hasil Deteksi Luka</i>	33
3.5.7	<i>Desain Halaman Penanganan Awal dan Rekomendasi Obat</i>	34
3.5.8	<i>Desain Halaman Chatbot</i>	35
<b>BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM</b>		<b>36</b>
4.1	<i>Implementasi Tahap Persiapan Dataset</i>	37
4.1.1	<i>Pengumpulan Dataset</i>	37
4.1.2	<i>Anotasi Dataset</i>	38
4.1.3	<i>Preprocessing, Augmentasi, Split Dataset</i>	39
4.2	<i>Implementasi Tahap Pelatihan Model</i>	40
4.2.1	<i>Mengakses Dataset</i>	40
4.2.2	<i>Mengunduh Pustaka Ultralytics</i>	40

4.2.3	<i>Import Kelas Yolo</i> .....	41
4.2.4	<i>Membuat Objek Model YOLO</i> .....	41
4.2.5	<i>Tahap Pelatihan Model</i> .....	41
4.2.6	<i>Menyimpan Model</i> .....	42
4.3	<i>Tuning Hyperparameter dan Pengujian Model</i> .....	42
4.4	<i>Analisa Proses Deteksi</i> .....	51
4.4.1	<i>Resize citra</i> .....	51
4.4.2	<i>Arsitektur YOLOv8</i> .....	53
4.4.3	<i>Proses CNN</i> .....	57
4.5	<i>Implementasi Model ke Aplikasi Berbasis Android</i> .....	62
4.5.1	<i>Halaman Login Aplikasi</i> .....	63
4.5.2	<i>Halaman Beranda Aplikasi</i> .....	63
4.5.3	<i>Halaman Artikel Kesehatan</i> .....	64
4.5.4	<i>Halaman Artikel Batasan Luka</i> .....	65
4.5.5	<i>Halaman Scan</i> .....	65
4.5.6	<i>Halaman Hasil Deteksi Luka</i> .....	66
4.5.7	<i>Halaman Penanganan Awal dan Rekomendasi Obat</i> .....	67
4.5.8	<i>Halaman Chatbot</i> .....	68
4.6	<i>Pengujian Sistem</i> .....	68
4.7	<i>Hasil Pengujian Sistem</i> .....	73
<b>BAB 5 KESIMPULAN DAN SARAN</b> .....		<b>84</b>
5.1	<i>Kesimpulan</i> .....	84
5.2	<i>Saran</i> .....	85
<b>DAFTAR PUSTAKA</b> .....		<b>86</b>
<b>LAMPIRAN</b> .....		<b>88</b>

## DAFTAR TABEL

Tabel 4.1 Pengumpulan <i>Dataset</i> .....	37
Tabel 4.2 <i>Training Epoch</i> 100, <i>Learning Rate</i> 0.01 .....	42
Tabel 4.3 <i>Summary Training Epoch</i> 100, <i>Learning Rate</i> 0.01 .....	42
Tabel 4.4 <i>Training Epoch</i> 150, <i>Learning Rate</i> 0.01 .....	43
Tabel 4.5 <i>Summary Training Epoch</i> 150, <i>Learning Rate</i> 0.01 .....	43
Tabel 4.6 <i>Training Epoch</i> 200, <i>Learning Rate</i> 0.01 .....	43
Tabel 4.7 <i>Summary Training Epoch</i> 200, <i>Learning Rate</i> 0.01 .....	43
Tabel 4.8 <i>Training Epoch</i> 100, <i>Learning Rate</i> 0.001 .....	44
Tabel 4.9 <i>Summary Training Epoch</i> 100, <i>Learning Rate</i> 0.001 .....	44
Tabel 4.10 <i>Training Epoch</i> 150, <i>Learning Rate</i> 0.001 .....	44
Tabel 4.11 <i>Summary Training Epoch</i> 150, <i>Learning Rate</i> 0.001 .....	44
Tabel 4.12 <i>Training Epoch</i> 200, <i>Learning Rate</i> 0.001 .....	45
Tabel 4.13 <i>Summary Training Epoch</i> 200, <i>Learning Rate</i> 0.001 .....	45
Tabel 4.14 <i>Training Epoch</i> 250, <i>Learning Rate</i> 0.001 .....	45
Tabel 4.15 <i>Summary Training Epoch</i> 250, <i>Learning Rate</i> 0.001 .....	45
Tabel 4.16 Pengukuran <i>Confusion matrix</i> .....	48
Tabel 4.17 Ilustrasi Komponen Pada <i>Bounding Box</i> .....	53
Tabel 4.18 Arsitektur YOLOv8 Per <i>Layer</i> .....	53
Tabel 4.19 Matriks Citra Pada Koordinat (228, 260) <i>Red</i> .....	57
Tabel 4.20 Matriks Citra Pada Koordinat (228, 260) <i>Green</i> .....	57
Tabel 4.21 Matriks Citra Pada Koordinat (228, 260) <i>Blue</i> .....	58
Tabel 4.22 Filter <i>Konvolusi</i> 3x3.....	58
Tabel 4.23 Matriks Hasil <i>Konvolusi</i> (228, 260) <i>Red</i> .....	58
Tabel 4.24 Matriks Hasil <i>Konvolusi</i> (228, 260) <i>Green</i> .....	58
Tabel 4.25 Matriks hasil <i>konvolusi</i> (228, 260) <i>Blue</i> .....	59
Tabel 4.26 Hasil Dari Proses RELU pada matriks (228, 260) <i>Red</i> .....	59
Tabel 4.27 Hasil Dari Proses RELU pada matriks (228, 260) <i>Green</i> .....	59
Tabel 4.28 Hasil Dari Proses RELU pada matriks (228, 260) <i>Blue</i> .....	59
Tabel 4.29 Hasil <i>Max Pooling</i> citra (228, 260) <i>Red</i> .....	60
Tabel 4.30 Hasil <i>Max Pooling</i> citra (228, 260) <i>Green</i> .....	60
Tabel 4.31 Hasil <i>Max Pooling</i> citra (228, 260) <i>Blue</i> .....	60

Tabel 4.32 Pengujian Sistem.....	69
Tabel 4.33 Hasil Uji Sistem .....	83

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur CNN .....	13
Gambar 2.2 <i>Kernel</i> CNN .....	13
Gambar 2.3 <i>Stride</i> (Tingkat) pada CNN .....	14
Gambar 2.4 <i>Pooling</i> CNN.....	14
Gambar 2.5 <i>Flatten</i> CNN.....	15
Gambar 2.6 Deteksi <i>Object Real Time</i> YOLO .....	15
Gambar 2.7 Arsitektur YOLO .....	16
Gambar 2.8 Konsep YOLO .....	16
Gambar 2.9 Ilustrasi IOU.....	17
Gambar 3.1 Arsitektur Umum .....	21
Gambar 3.2 <i>Use Case Diagram</i> .....	22
Gambar 3.3 <i>Activity Diagram</i> .....	23
Gambar 3.4 <i>Sequence Diagram</i> .....	25
Gambar 3.5 <i>Flowchart</i> Sistem .....	26
Gambar 3.6 <i>Flowchart</i> Deteksi Object YOLO .....	27
Gambar 3.7 Desain Halaman <i>Login</i> .....	28
Gambar 3.8 Desain Beranda Aplikasi.....	29
Gambar 3.9 Desain Halaman Artikel Kesehatan .....	30
Gambar 3.10 Desain Halaman Artikel Batasan Luka .....	31
Gambar 3.11 Desain Halaman <i>Scan</i> Aplikasi.....	32
Gambar 3.12 Desain Halaman Hasil Deteksi Luka .....	33
Gambar 3.13 Desain Halaman Penanganan Awal dan Rekomendasi Obat.....	34
Gambar 3.14 Desain Halaman <i>Chatbot</i> Aplikasi.....	35
Gambar 4.1 Implementasi Sistem .....	36
Gambar 4.2 Proses Persiapan Dataset.....	37
Gambar 4.3 <i>Dataset</i> Luka Sayat .....	37
Gambar 4.4 <i>Dataset</i> Luka Lecet .....	38
Gambar 4.5 <i>Dataset</i> Luka Bakar.....	38
Gambar 4.6 <i>Dataset</i> Luka Tusuk .....	38
Gambar 4.7 Proses <i>Anotasi</i> .....	39
Gambar 4.8 Pembagian <i>Dataset</i> .....	39

Gambar 4.9 Akses ke <i>Dataset</i> .....	40
Gambar 4.10 Unduh Pustaka Utralytics.....	40
Gambar 4.11 <i>Import</i> YOLO.....	41
Gambar 4.12 Membuat Objek Model YOLO .....	41
Gambar 4.13 Pelatihan Model .....	41
Gambar 4.14 Menyimpan Model .....	42
Gambar 4.15 Grafik Performa Model .....	46
Gambar 4.16 Gambar <i>Precision-Recall</i> Model.....	47
Gambar 4.17 Hasil Pengujian Model (1) .....	50
Gambar 4.18 Hasil Pengujian Model (2) .....	50
Gambar 4.19 Hasil Pengujian Model (3) .....	51
Gambar 4.20 Citra Inputan.....	52
Gambar 4.21 Citra <i>Resize</i> 416x416.....	52
Gambar 4.22 Citra Diberi <i>Gridcell</i> 7x7 .....	53
Gambar 4.23 Ilustrasi Pencarian <i>Bounding Box</i> .....	55
Gambar 4.24 Potongan Gambar Koordinat (185,193).....	55
Gambar 4.25 Objek yang Didapatkan Oleh <i>Bounding Box</i> .....	56
Gambar 4.26 Prediksi IOU.....	56
Gambar 4.27 <i>Fully Connected Layer</i> Untuk Mencari Probabilitas .....	61
Gambar 4.28Arsitektur Pada <i>Cloud</i> .....	62
Gambar 4.29 Halaman <i>Login</i> Aplikasi CekLukaID .....	63
Gambar 4.30 Beranda Aplikasi CekLukaID .....	63
Gambar 4.31 Halaman Artikel Kesehatan .....	64
Gambar 4.32 Halaman Batasa Luka .....	65
Gambar 4.33 Halaman <i>Scan</i> Luka .....	65
Gambar 4.34 Halaman Hasil Deteksi.....	66
Gambar 4.35 Halaman Manajemen Perawatan Luka.....	67
Gambar 4.36 Halaman <i>Chatbot</i> Aplikasi.....	68
Gambar 4.37 Hasil Uji Sistem Data Uji No 1 .....	73
Gambar 4.38 Hasil Uji Sistem Data Uji No 2.....	73
Gambar 4.39 Hasil Uji Sistem Data Uji No 3.....	74
Gambar 4.40 Hasil Uji Sistem Data Uji No 4.....	74
Gambar 4.41 Hasil Uji Sistem Data Uji No 5.....	75

Gambar 4.42 Hasil Uji Sistem Data Uji No 6.....	75
Gambar 4.43 Hasil Uji Sistem Data Uji No 7 .....	76
Gambar 4.44 Hasil Uji Data Uji No 8.....	76
Gambar 4.45 Hasil Uji Data Uji No 9.....	77
Gambar 4.46 Hasil Uji Data Uji No 10.....	77
Gambar 4.47 Hasil Uji Data Uji No 11.....	78
Gambar 4.48 Hasil Uji Data Uji No 13.....	79
Gambar 4.49 Hasil Uji Data Uji No 14.....	79
Gambar 4.50 Hasil Uji Data Uji No 15.....	80
Gambar 4.51 Hasil Uji Data Uji No 16.....	80
Gambar 4.52 Hasil Uji Data Uji No 17.....	81
Gambar 4.53 Hasil Uji Data Uji No 18.....	81
Gambar 4.54 Hasil Uji Data Uji No 19.....	82
Gambar 4.55 Hasil Uji Data Uji No 20.....	82



# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Tubuh manusia memiliki organ yang sangat penting yaitu kulit. Kulit berperan sebagai pelindung terhadap lingkungan luar dan menjaga keseimbangan tubuh dengan menyimpan nutrisi serta berperan dalam sistem pertahanan tubuh. Namun ketika kulit mengalami luka akibat tekanan fisik, bahan kimia, atau gesekan, hal ini mengganggu integritas dan fungsi normalnya. Proses penyembuhan luka dimulai segera setelah terjadinya luka dengan tujuan memperbaiki kerusakan pada kulit dan mengembalikan fungsinya. Oleh karena itu, perawatan luka pada kulit memiliki peran penting dalam menjaga kesehatan tubuh secara keseluruhan (Criollo-Mendoza et al., 2023).

Dalam Pasal 229 ayat 3 Undang-Undang Lalu Lintas dan Angkutan Jalan (LLAJ) tahun 2009 dijelaskan bahwa luka yang dianggap sebagai ringan adalah luka yang menyebabkan ketidaknyamanan bagi korban tetapi tidak memerlukan intervensi medis di rumah sakit, atau luka yang tidak masuk dalam klasifikasi luka berat. Pengertian luka ringan ini mengacu pada pasal 351, 352, dan 90 dalam Kitab Undang-Undang Hukum Pidana (KUHP) tahun 2021. Menurut isi pasal 352 KUHP, luka ringan adalah luka yang tidak mengakibatkan gangguan atau hambatan dalam menjalankan pekerjaan, jabatan, atau mencari nafkah. Contoh luka ringan meliputi lecet, sayatan, tusukan, dan luka bakar seringkali terjadi pada banyak individu dalam aktivitas sehari-hari. Meskipun dianggap sebagai masalah kecil, penanganan yang tepat terhadap luka ringan memiliki dampak yang signifikan terhadap kesehatan dan kenyamanan individu. Oleh karena itu, penting untuk memberikan perhatian dan penanganan yang baik terhadap luka ringan guna memastikan pemulihan yang optimal dan mencegah terjadinya komplikasi yang lebih serius.

Menurut Badan Pusat Statistik (BPS) tahun 2022 angka orang yang mengalami luka ringan menempati posisi paling banyak di tahun 2021 yaitu sekitar 117.913 dibandingkan dengan orang yang mengalami luka berat dalam kasus kecelakaan,

Profesor Dr. Yoyos Dias Ismiarto, seorang Guru Besar di Fakultas Kedokteran Universitas Padjadjaran, dalam acara pengukuhan Guru Besar bidang Ilmu Orthopaedi dan Traumatologi, menekankan bahwa luka yang tidak ditangani dengan benar cenderung lebih rentan terinfeksi dan memerlukan waktu penyembuhan yang lebih lama, penanganan yang salah ini merupakan akibat dari ketidakpedulian terhadap luka ringan yang berakar dari kurangnya pemahaman akan potensi risiko yang mungkin terjadi. Sebagian orang mungkin menganggap bahwa luka ringan adalah bagian yang lumrah dalam kehidupan sehari-hari, oleh karena itu tidak memerlukan perawatan khusus. Namun, luka ringan yang tidak ditangani dengan baik bisa berujung pada risiko infeksi, terutama jika area luka terkontaminasi oleh bakteri atau kuman lainnya.

Manajemen perawatan luka sering kali dilakukan dengan cara yang sederhana dan seragam, tetapi sebenarnya harus disesuaikan dengan jenis luka dan masalah yang terkait. Tidak semua luka sama, jadi penting untuk merawatnya secara sesuai. Perawatan luka yang baik sangat penting untuk memastikan luka sembuh dengan cepat dan mencegah masalah tambahan. Tujuannya adalah untuk mempercepat proses penyembuhan, menghindari masalah yang dapat mempengaruhi produktivitas dan biaya yang harus dikeluarkan (Risal Wintoko, 2020) .

Penelitian ini bertujuan untuk menyediakan solusi yang memudahkan dalam mengelola perawatan luka dengan memberikan manajemen perawatan luka yang meliputi tentang langkah-langkah penanganan awal dan rekomendasi obat yang sesuai dengan jenis luka. Dengan demikian, diharapkan dapat menghindari dampak yang merugikan dan memastikan penanganan yang tepat dan efektif.

*Convolutional Neural Network* (CNN) adalah salah satu struktur dalam *Deep Learning* yang terinspirasi oleh sistem visual dalam otak manusia. CNN telah terbukti sebagai algoritma pembelajaran yang sangat efisien dalam memproses informasi gambar, sehingga sering dianggap sebagai model optimal untuk berbagai tugas terkait gambar seperti segmentasi, klasifikasi, *tagging*, deteksi. CNN terdiri dari banyak unit komputasi yang disebut '*neuron*', yang menjalankan operasi sederhana dan berinteraksi satu sama lain untuk menghasilkan keputusan. Analogi dengan otak manusia memperkuat posisi CNN sebagai alat yang sangat kuat dalam menganalisis dan memahami informasi visual (Patel & Patel, 2020).

*You Only Look Once* (YOLO) adalah sebuah pendekatan deteksi objek dalam pengolahan citra yang membagi gambar menjadi *grid* menggunakan satu model

*Convolutional Neural Network* (CNN). Setiap *grid* memperkirakan kotak pembatas (*bounding boxes*) dan skor kepercayaan (*confidence scores*) untuk objek yang ada di dalamnya. Kelas objek dalam kotak pembatas dihitung menggunakan skor kepercayaan yang diprediksi (Syahputra et al., 2024). Arsitektur *Convolutional Neural Network* (CNN) merupakan fondasi dari YOLO, yang memungkinkan pengenalan objek yang cepat dan akurat. YOLO sering dimanfaatkan untuk memprediksi tugas deteksi objek dengan efisiensi yang tinggi (Mesbahi et al., 2023).

Metode deteksi YOLOv8 yang akan diimplementasikan dalam penelitian ini untuk mengidentifikasi luka ringan. YOLOv8 menawarkan deteksi objek yang cepat dan akurat dengan model yang ringkas, serta kemampuan multi-skala untuk mengatasi objek berukuran berbeda. Peningkatan ekstraksi fitur pada YOLOv8 juga meningkatkan akurasi deteksi (Terven et al., 2023). Dengan menggunakan algoritma ini, diharapkan dapat memberikan hasil yang optimal dalam mengidentifikasi dan mengklasifikasikan jenis luka ringan seperti luka lecet, luka sayat, luka tusuk dan luka bakar ringan dalam konteks manajemen perawatan luka.

Dari penelitian ini dihasilkan sebuah aplikasi bernama CekLukaID. Aplikasi ini memudahkan pengguna dengan cara yang sederhana namun efektif, mengunggah gambar luka dan mengolahnya menggunakan model YOLOv8. Begitu luka terdeteksi dalam gambar, aplikasi dengan cepat memberikan informasi yang berguna kepada pengguna, termasuk nama luka yang terdeteksi serta pertanyaan terkait yang bisa dijawab oleh pengguna. Selain itu, pengguna juga bisa melanjutkan untuk melihat langkah-langkah penanganan awal dan rekomendasi obat yang sesuai. Dengan demikian, aplikasi ini tidak hanya memudahkan pengguna dalam mengidentifikasi luka, tetapi juga memberikan solusi praktis untuk perawatan yang sesuai dan efektif.

## **1.2 Rumusan Masalah**

Dalam usaha meningkatkan perawatan luka ringan di masyarakat Indonesia, terdapat tantangan dalam pemahaman dan penanganan yang tepat terhadap berbagai jenis luka ringan, seperti lecet, sayatan, tusukan, dan bakar. Ketidakpahaman yang meluas dapat menyebabkan risiko komplikasi dan perawatan yang tidak optimal. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sebuah aplikasi menggunakan metode deteksi objek *You Only Look Once* (YOLO) yang memanfaatkan arsitektur *Convolutional Neural Networks* (CNN) untuk mengidentifikasi luka dan memberikan

informasi praktis tentang penanganan awal serta rekomendasi obat yang sesuai untuk setiap jenis luka ringan. Dengan demikian, diharapkan kesadaran dan pengetahuan masyarakat tentang perawatan luka ringan bisa ditingkatkan, dan potensi risiko komplikasi dapat diminimalkan

### 1.3 Batasan Masalah

Berikut adalah beberapa pembatasan yang relevan dalam lingkup penelitian ini:

1. Penelitian hanya memfokuskan pada identifikasi luka-luka ringan seperti lecet, sayatan, tusukan, dan bakar yang bisa ditangani di rumah tanpa mengganggu aktivitas sehari-hari.
2. Aplikasi ini hanya digunakan untuk usia 17 tahun keatas.
3. Penelitian ini akan menggunakan metode deteksi objek YOLOv8 untuk melakukan identifikasi luka.
4. Pengembangan program dan model akan menggunakan bahasa pemrograman Python.
5. Input aplikasi ini berupa foto-foto luka ringan yang akan diidentifikasi.
6. Outputnya akan memberikan informasi nama luka ringan, penanganan awal dan rekomendasi obat untuk jenis luka yang telah diidentifikasi

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menerapkan metode deteksi objek *You Only Look Once* (YOLO) yang memanfaatkan arsitektur *Convolutional Neural Networks* (CNN) untuk mengidentifikasi luka ringan, dengan harapan dapat memberikan manajemen perawatan luka yang optimal untuk mempercepat proses penyembuhan.

### 1.5 Manfaat Penelitian

Dibawah ini adalah sejumlah manfaat yang dapat diperoleh dari penelitian ini:

1. Pemahaman Publik: Memberikan kesadaran yang lebih luas kepada masyarakat mengenai penanganan luka ringan melalui *platform* aplikasi *mobile*.
2. Optimalisasi Perawatan: Meningkatkan efektivitas perawatan luka dengan memberikan informasi manajemen perawatan luka yaitu penanganan awal dan rekomendasi obat yang sesuai.
3. Pencegahan Infeksi: Mengurangi risiko infeksi dengan deteksi cepat dan tepat terhadap jenis luka.

4. Efisiensi Medis: Meningkatkan efisiensi waktu dan sumber daya dalam penanganan medis, khususnya dalam situasi darurat.
5. Kontribusi Teknologi Kesehatan: Berpotensi membuka peluang inovasi lebih lanjut dalam pengembangan teknologi kecerdasan buatan dalam konteks perawatan kesehatan.

## 1.6 Metodologi Penelitian

Dalam penelitian ini, metode berikut akan diimplementasikan:

### 1. Studi Pustaka

Langkah pertama penelitian dimulai dengan melakukan studi pustaka untuk mengumpulkan referensi dari berbagai sumber yang dapat dipercaya. Tinjauan pustaka mencakup penelusuran buku-buku, jurnal, artikel ilmiah, *e-book*, makalah, dan sumber informasi *online* yang relevan dengan bidang *Machine Learning* dan YOLOv8. Tujuan dari tahap ini adalah untuk mendalami tentang konsep-konsep dasar dan kerangka konseptual yang mendukung pengembangan model YOLOv8 dalam mengidentifikasi luka ringan.

### 2. Pengumpulan *Dataset*

Langkah berikutnya adalah mengumpulkan *dataset* citra digital yang mencakup berbagai jenis luka ringan seperti luka lecet, luka sayat, luka tusuk, dan luka bakar ringan. *Dataset* ini akan menjadi dasar untuk melatih dan menguji model YOLOv8 agar dapat mengenali dan mengidentifikasi berbagai varian luka ringan.

### 3. *Preprocessing* Data

Pada tahap *preprocessing* data, *dataset* yang telah dikumpulkan akan dinormalisasi dan diubah ukurannya untuk mempersiapkan data pelatihan. Proses augmentasi data akan dilakukan untuk meningkatkan keragaman dan generalisasi model. Informasi mengenai lokasi dan jenis luka ringan akan ditambahkan ke *dataset* melalui *bounding box*, memudahkan proses identifikasi pada tahap pelatihan model.

### 4. Pengembangan Model YOLOv8

Pada tahap ini, model YOLOv8 akan diimplementasikan untuk pengenalan dan identifikasi langsung luka ringan pada citra. Proses pelatihan model YOLOv8

akan dilakukan menggunakan dataset yang telah diproses sebelumnya, sehingga model dapat belajar mengenali berbagai jenis luka ringan dengan akurat.

#### 5. Evaluasi Model

Pada tahap evaluasi model, kinerja model YOLOv8 akan dinilai menggunakan metrik seperti akurasi, recall dan presisi. Evaluasi ini bertujuan untuk memastikan bahwa model mampu memberikan hasil identifikasi luka ringan yang akurat dan efisien.

#### 6. Pengembangan Aplikasi

Tahap pengembangan aplikasi melibatkan pembangunan antarmuka intuitif. Model YOLOv8 akan diintegrasikan ke dalam aplikasi untuk melakukan identifikasi luka ringan secara real-time dan dengan sistem *upload* gambar dan kamera. Aplikasi akan menyediakan fitur tampilan hasil identifikasi, informasi penanganan awal luka, dan rekomendasi obat yang sesuai berdasarkan jenis luka yang teridentifikasi. Hal ini bertujuan untuk memberikan solusi praktis dan informatif kepada pengguna dalam merawat luka ringan.

#### 7. Evaluasi Aplikasi

Evaluasi keseluruhan akan mencakup pengujian antarmuka dan fungsionalitas aplikasi. Pengguna akan diminta untuk mengunggah citra luka ringan, dan kemudian aplikasi akan menampilkan hasil identifikasi yang melibatkan *bounding box*. Keterlibatan pengguna akan dievaluasi untuk memastikan antarmuka intuitif dan mudah digunakan.

#### 8. Dokumentasi dan Penyusunan Laporan

Seluruh proses penelitian dimulai dari awal hingga tahap terakhir akan dicatat sebagai skripsi.

### 1.7 Penelitian Relevan

Berikut adalah penelitian terdahulu yang relevan, yang telah dilaksanakan sebelumnya antara lain:

1. Pada jurnal dengan judul *Automatic wound detection and size estimation using deep learning algorithms* (Carrión et al., 2022) Penelitian ini menggunakan metode *deep learning* untuk mengembangkan sebuah pipeline analisis luka otomatis. Dalam penelitian ini, digunakan algoritma deteksi objek YOLOv3 dan algoritma segmentasi instansi medis *U-Net* yang dikustomisasi untuk mengidentifikasi dan

mengukur ukuran luka secara otomatis dari gambar luka tikus. Metode ini melibatkan langkah-langkah pra-pemrosesan, deteksi lokasi area yang terluka menggunakan YOLOv3, dan estimasi ukuran luka menggunakan algoritma segmentasi *U-Net*. Selain itu, penelitian ini juga mencakup penilaian terhadap kontribusi algoritma pascaproses terhadap kinerja pipeline secara keseluruhan dengan membandingkannya dengan beberapa algoritma segmentasi lainnya. Dengan demikian, penelitian ini bertujuan untuk memberikan solusi yang efisien dan efektif dalam mengukur ukuran luka secara otomatis tanpa banyak campur tangan manusia.

2. Pada jurnal dengan judul *CADFU for Dermatologists: A Novel Chronic Wounds & Ulcers Diagnosis System with DHuNeT (Dual-Phase Hyperactive UNet) and YOLOv8 Algorithm* (Shah et al., 2023) Pada penelitian terdahulu yang telah dilakukan, dikembangkanlah sebuah sistem inovatif bernama CADFU (*Computer-Aided Diagnosis System for Foot Ulcers*) yang dirancang untuk mendeteksi dan memisahkan luka kaki secara efisien. Dengan memanfaatkan algoritma YOLOv8 untuk deteksi luka dan DHuNeT untuk segmentasi, penelitian tersebut berhasil menciptakan solusi yang efektif dan terampil dalam manajemen luka kronis, khususnya luka kaki. Sistem ini telah terbukti memberikan hasil yang mengungguli berbagai arsitektur terkini.
3. Pada jurnal dengan judul *Multiclass wound image classification using an ensemble deep CNN-based classifier* (Rostami et al., 2021) menghasilkan kesimpulan yaitu peneliti mengembangkan suatu sistem komputer yang bertujuan untuk mengidentifikasi dan mengklasifikasikan gambar luka berdasarkan jenisnya. peneliti menciptakan model klasifikasi berbasis *Convolutional Neural Network* (CNN) yang kompleks dan menyeluruh, dengan menggunakan teknik *ensemble* untuk meningkatkan kinerja. Yang menarik, model yang peneliti usulkan merupakan yang pertama kalinya dapat mengklasifikasikan gambar luka ke dalam lebih dari dua jenis. Penelitian ini menggunakan *patch classifiers* dengan arsitektur *AlexNet* yang disesuaikan untuk mengklasifikasikan efisien bagian kecil dari gambar luka. Eksperimen dilakukan untuk memahami pengaruh berbagai jenis luka terhadap akurasi klasifikasi, menguji berbagai kombinasi jenis luka untuk memahami bagaimana model merespons variasi data dalam tugas klasifikasi gambar

secara keseluruhan, peneliti mengadopsi pendekatan *ensemble* dengan menggabungkan klasifikasi patch dan model *AlexNet*. Hasil eksperimen menunjukkan bahwa pendekatan *ensemble* berhasil meningkatkan akurasi klasifikasi gambar luka secara memuaskan.

## **1.8 Sistematika Penulisan**

Skripsi ini terstruktur dalam lima bab yang mencakup:

### **BAB 1: Pendahuluan**

Segala hal yang terkait dengan konteks masalah, seperti kumpulan permasalahan, tujuan penelitian, batasan permasalahan, serta tujuan, fungsi, dan metodologi penelitian, juga diuraikan dalam bab ini.

### **BAB 2: Landasan Teori**

Analisis teoritis yang mencakup luka ringan, *Convolutional Neural Network*, *You Only Look Once*, serta manajemen perawatan luka akan dibahas di sini.

### **BAB 3: Analisis Dan Perancangan**

Pada bagian ini, akan dibahas mengenai pelaksanaan pengamatan yang terkait dengan sistem yang telah diciptakan. Selanjutnya, akan dijelaskan perancangan sistem dengan deteksi objek YOLO yang menggunakan arsitektur CNN.

### **BAB 4: Implementasi Dan Pengujian**

Bab berikut akan membahas pelaksanaan dan pengujian sistem, yang didasarkan pada langkah-langkah analisis dan perancangan sistem yang telah dijelaskan sebelumnya.

### **BAB 5: Kesimpulan Dan Saran**

Bab berikut akan menyoroti temuan-temuan serta memberikan saran untuk arah penelitian mendatang.



## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Luka**

Luka merujuk pada kondisi di mana terjadi gangguan pada kontinuitas jaringan, yang disebabkan oleh kerusakan atau kehilangan substansi jaringan akibat cedera atau intervensi bedah. Luka merupakan bentuk penurunan integritas jaringan epitel, yang bisa timbul akibat ketidaksempurnaan pada kulit, mukosa, atau organ. Berbagai faktor dapat menyebabkan terjadinya luka, baik itu sebagai akibat dari kondisi penyakit atau kejadian yang tidak terduga, baik itu disengaja maupun tidak. Luka yang terjadi disengaja seringkali terkait dengan prosedur medis seperti operasi, sementara yang tidak disengaja seringkali terjadi secara kebetulan. Luka dapat diakibatkan oleh trauma tumpul atau trauma tajam. Trauma tumpul melibatkan kekerasan dari benda tumpul yang dapat menghasilkan luka memar, lecet, atau robek. Sementara itu, trauma tajam melibatkan kontak dengan benda tajam yang dapat menghasilkan luka iris, luka sayat, luka tusuk, atau luka bacok. Luka juga dapat diklasifikasikan berdasarkan tingkat keparahan, mulai dari yang ringan hingga yang berat (Risal Wintoko, 2020).

Terjadinya luka adalah suatu kejadian yang umum dialami oleh manusia, mulai dari anak-anak, remaja, hingga orang tua. Meskipun pada awalnya mungkin hanya terdapat sedikit goresan pada tubuh seseorang, tetapi hal tersebut tidak bisa dianggap enteng. Jika luka tersebut tidak mendapatkan perawatan yang memadai atau dibiarkan tanpa penanganan yang baik, dampaknya bisa sangat fatal bahkan dapat menyebabkan kerusakan parah hingga memerlukan tindakan amputasi (Kemenkes RI, 2022) .

#### **2.2 Luka Ringan**

Luka ringan, sebagaimana dijelaskan dalam penjelasan Pasal 229 ayat [3] Undang-Undang Lalu Lintas dan Angkutan Jalan (UU LLAJ), dapat diartikan sebagai luka yang menyebabkan korban merasa sakit, namun tidak memerlukan perawatan di sakit atau tidak termasuk dalam kategori luka berat. sementara itu, luka berat, yang didefinisikan

dalam penjelasan Pasal 229 ayat (4) UU LLAJ, merujuk pada luka yang menghasilkan kondisi dimana korban mengalami ketidakmampuan untuk pulih sepenuhnya, tidak dapat melanjutkan aktivitas atau pekerjaannya secara normal, kehilangan fungsi salah satu panca indra, mengalami cacat serius atau kehilangan kemampuan bergerak, mengalami gangguan daya pikir selama minimal 4 minggu, mengalami keguguran, atau memerlukan perawatan di fasilitas medis selama periode melebihi 30 hari.

### 2.2.1 *Luka Tusuk*

Luka tusuk merujuk pada jenis cedera yang disebabkan oleh benda tajam, juga dikenal sebagai trauma tajam, dimana terjadi tusukan oleh benda tajam dengan arah yang hampir tegak lurus terhadap kulit. Tidak jarang lebar luka pada permukaan kulit tidak mencerminkan kedalaman sebenarnya dari luka tusuk (Linggom & Sihaloho, 2022). Trauma semacam ini sering terjadi saat seseorang berinteraksi dengan alat-alat rumah tangga seperti mesin jahit, paku, pisau, atau tertusuk duri. Meskipun sebagian besar luka tusuk bersifat ringan dan dapat sembuh dalam beberapa hari, kejadian ini tetap terkait dengan risiko cedera yang serius (Hellosehat, 2021).

### 2.2.2 *Luka Sayat*

Luka sayat merujuk pada jenis luka yang timbul akibat kontak dengan benda atau alat yang memiliki mata tajam, dan seringkali terjadi ketika terdapat tekanan ringan dan goresan pada permukaan tubuh. Karakteristik khas dari luka sayat adalah bentuknya yang bisa berupa celah atau terbuka lebar, dengan kecenderungan untuk memiliki struktur yang asimetris. Dalam kondisi ini, luka dapat terbentuk sebagai suatu alur yang membuka jaringan kulit dan daging (Melati et al., 2022).

Luka sayatan dapat dibagi menjadi dua kategori berdasarkan tingkat keparahannya yaitu luka dangkal dan luka dalam. Luka sayatan yang dangkal hanya memengaruhi permukaan kulit, sedangkan luka sayatan yang dalam dapat menembus lebih dari 1 cm dan memiliki potensi untuk merusak struktur seperti tendon, otot, ligamen, saraf, pembuluh darah, bahkan mencapai tulang. Penanganan luka sayatan yang dangkal biasanya dapat dilakukan sendiri di rumah, namun luka sayatan yang lebih dalam memerlukan perawatan medis yang lebih intensif (Alodokter, 2023).

### 2.2.3 Luka Lecet

Luka lecet pada kulit umumnya disebabkan oleh gesekan atau tindakan garukan. Akibatnya, lapisan luar kulit biasanya terkikis, dan daerah tersebut dapat mengeluarkan cairan keruh atau darah. Selain itu, tanah, pasir, dan materi lainnya seringkali menempel pada area yang terkena luka. Jika luka lecet terkena kotoran, sangat penting untuk segera membersihkannya agar dapat mencegah infeksi dan mempercepat proses penyembuhan (Kemenkes RI, 2022).

Gesekan pada beberapa bagian kulit sering kali menghasilkan luka lecet, terutama di area-area yang berdekatan dengan tulang seperti tangan, lengan bawah, siku, lutut, atau tulang kering. Umumnya, luka lecet tidak menghasilkan perdarahan yang signifikan dan bersifat ringan, sehingga dapat diatasi sendiri di rumah (Hellosehat, 2022).

### 2.2.4 Luka Bakar

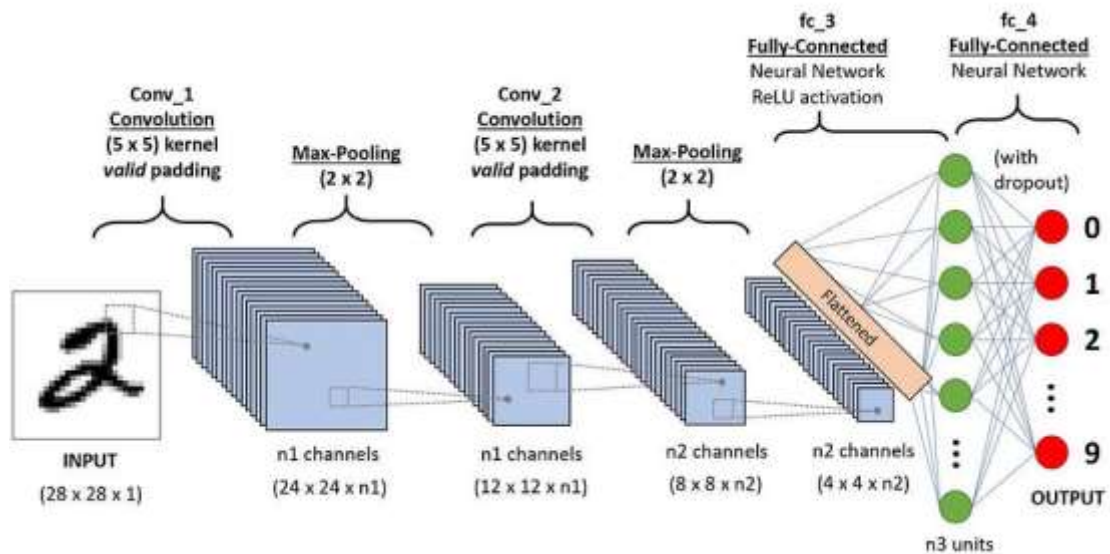
Cedera termal seperti luka bakar merupakan risiko yang dapat dialami oleh berbagai individu yang terpapar oleh sumber panas seperti api atau minyak. Baik itu koki, pekerja las, atau siapa pun yang menggunakan peralatan yang melibatkan api, semuanya dapat mengalami risiko luka bakar. Luka bakar umumnya terbagi menjadi tiga jenis yang dikenal luas yaitu *superficial*, derajat dua, derajat tiga. Luka bakar *superficial* biasanya ditandai dengan kemerahan, sensasi nyeri, bengkak, dan hanya memengaruhi lapisan kulit paling atas. Di sisi lain, luka bakar derajat dua melibatkan kerusakan pada lapisan kulit terluar dan bisa juga mengenai lapisan di bawahnya, seringkali ditandai dengan adanya gelembung berisi cairan di kulit. Sedangkan luka bakar derajat tiga melibatkan kerusakan pada kulit hingga lapisan jaringan yang lebih dalam, bahkan bisa mencapai tulang dan organ dalam. Penyebab luka bakar bisa beragam, mulai dari paparan langsung dengan api hingga cipratan minyak panas, sinar matahari, atau kontak dengan bahan kimia seperti asam akumulator. Menangani luka bakar dengan segera setelah terjadi sangat penting, karena jika tidak diberi perawatan yang tepat dalam waktu yang singkat, dapat menyebabkan kerusakan jaringan kulit yang lebih serius (Kemenkes RI, 2022).

### **2.3 Manajemen Perawatan Luka**

Manajemen perawatan luka melibatkan serangkaian langkah yang bertujuan untuk mencegah kerusakan pada kulit, membran mukosa, dan jaringan akibat trauma, patah tulang, atau prosedur operasi yang mungkin mengganggu keutuhan kulit. Meskipun pendekatan umumnya masih sederhana dan konsisten, penting untuk diingat bahwa setiap luka memerlukan penanganan yang sesuai dengan kondisi dan masalah spesifik yang muncul. Hal ini karena tidak semua luka membutuhkan perawatan yang seragam. Perawatan luka yang tepat sangat vital untuk memastikan proses penyembuhan yang optimal, bukan hanya untuk mencapai kesembuhan, tetapi juga untuk mempercepat proses penyembuhan, mengurangi risiko komplikasi, dan meminimalkan dampaknya terhadap produktivitas dan biaya (Risal Wintoko, 2020).

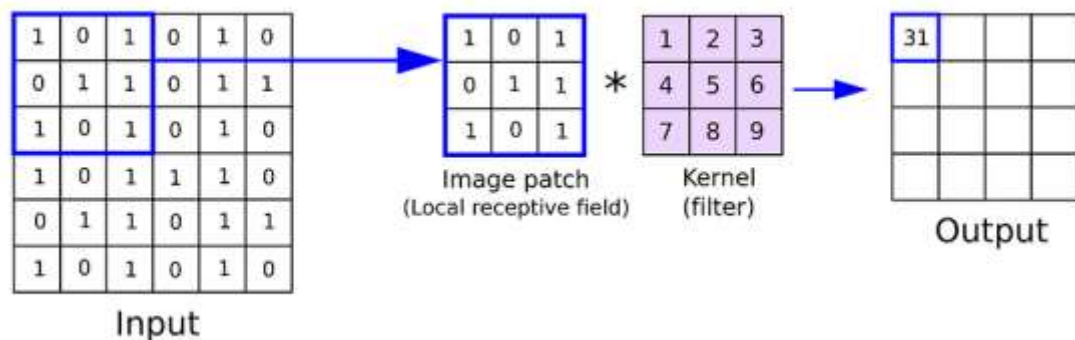
### **2.4 *Convolutional Neural Network (CNN)***

Pada CNN, istilah "*konvolusional*" itu seperti mesin pencari pola. Saat digunakan dalam CNN, lapisan-lapisan ini menemukan pola-pola penting dalam gambar. CNN sendiri terdiri dari beberapa lapisan yang bekerja berurutan. Di antaranya ada lapisan-lapisan yang mencari pola, lapisan-lapisan yang membantu mempersempit gambar agar lebih sederhana, dan lapisan yang menghubungkan pola-pola yang sudah ditemukan dengan apa yang harus dihasilkan. Yang penting di sini, lapisan-lapisan yang mencari pola itu berperan penting dalam mempelajari gambar dan menemukan pola-pola penting di dalamnya. Jadi, secara keseluruhan, CNN itu seperti detektif gambar yang cerdas, yang bisa mengenali berbagai macam gambar dengan baik, seperti wajah atau objek, membuatnya sangat berguna untuk mengenali gambar dan melakukan tugas-tugas lainnya (Ahad et al., 2023).



**Gambar 2.1** Arsitektur CNN

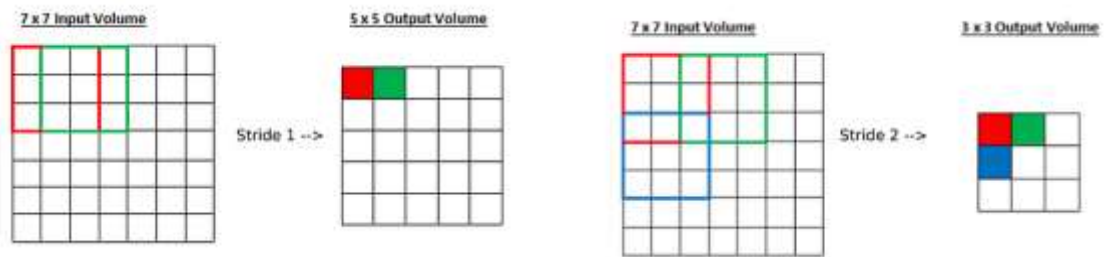
#### 2.4.1 Kernel atau Filter atau Detektor Fitur



**Gambar 2.2** Kernel CNN

Kernel atau Filter atau Detektor Fitur dalam jaringan saraf konvolusi adalah sebuah elemen yang digunakan untuk mengekstrak fitur-fitur dari gambar. Rumus yang digunakan untuk menghitung ukuran keluaran dari proses konvolusi adalah  $[i-k]+1$ , di mana  $i$  merupakan ukuran input dan  $k$  merupakan ukuran kernel.

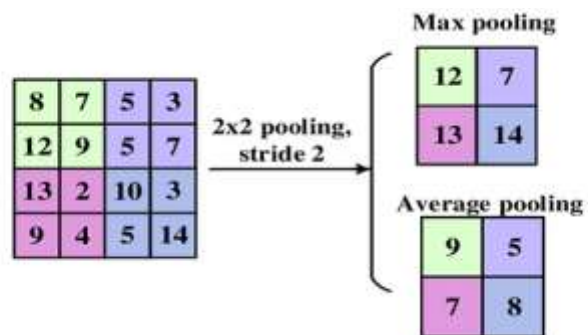
### 2.4.2 Stride



**Gambar 2.3 Stride (Tingkat) pada CNN**

*Stride* adalah parameter dari filter jaringan saraf yang mengubah jumlah pergerakan di atas gambar atau video. Jika kita menggunakan *stride* 1, maka ia akan mengambil satu per satu. Jika kita memberikan stride 2, maka ia akan mengambil nilai dengan melewati 2 piksel berikutnya. Rumusnya adalah  $[i-k/s]+1$ , di mana  $i$  merupakan ukuran input,  $k$  merupakan ukuran kernel, dan  $s$  merupakan *stride*.

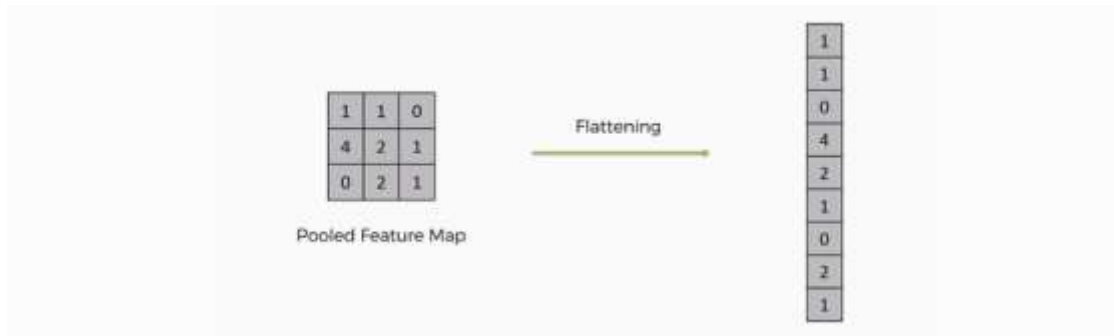
### 2.4.3 Pooling



**Gambar 2.4 Pooling CNN**

*Pooling* dalam jaringan saraf konvolusi merupakan sebuah metode untuk menggeneralisasi fitur yang diekstraksi oleh filter konvolusi, membantu jaringan mengenali fitur tanpa tergantung pada lokasi mereka dalam gambar. Metode ini bertujuan untuk menyederhanakan representasi fitur dengan mempertahankan informasi penting untuk pengenalan gambar.

#### 2.4.4 Flatten

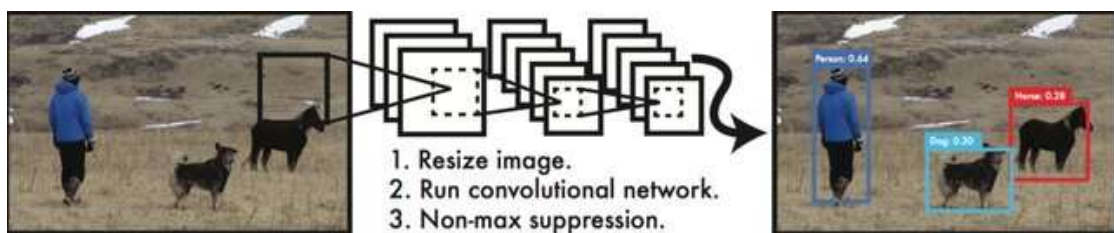


**Gambar 2.5 Flatten CNN**

*Flattening* digunakan untuk mengubah semua matriks 2-Dimensi yang dihasilkan dari peta fitur yang telah dipooling menjadi vektor linear kontinu tunggal yang panjang. Matriks yang telah diratakan disajikan sebagai input ke lapisan terhubung sepenuhnya untuk mengklasifikasikan gambar.

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

YOLO adalah sistem terkini dalam deteksi objek *real-time* yang terkenal karena kecepatannya yang luar biasa. Kinerjanya yang cepat menjadikannya sebagai metode standar dalam deteksi objek di bidang *computer vision*. Algoritma ini diperkenalkan pada tahun 2015 oleh Joseph Redmon. Deteksi YOLO diilustrasikan dalam Gambar 2.6. Pada dasarnya, sistem ini pertama-tama akan mengubah ukuran gambar, kemudian melewati jaringan konvolusi, dan terakhir lapisan *non-max suppression*. Hasilnya adalah gambar yang terdeteksi dan diberi label (Rohaziat et al., 2020).



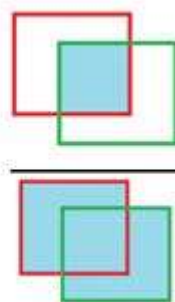
**Gambar 2.6 Deteksi Object Real Time YOLO**

Gambaran jaringan lapisan konvolusi dalam deteksi YOLO ditampilkan dalam Gambar 2.7. Jaringan ini terdiri dari 24 lapisan konvolusi, dua lapisan terhubung penuh, dan lapisan konvolusi 1x1 yang bergantian untuk mengurangi dimensi fitur dari lapisan sebelumnya. Model YOLO secara keseluruhan dilatih bersama dengan fungsi kerugian yang langsung memperhitungkan kinerja deteksi.





YOLO menghasilkan nilai kepercayaan (*confidence score*) pada setiap kotak pembatas sebagai probabilitas bahwa kotak tersebut berisi objek, yang kemudian akan dikalikan dengan IoU (*intersection over union*) antara kotak prediksi dan kotak kebenaran dasar (*ground truth*), yang merupakan nilai referensi dari proses pelatihan. IoU adalah metrik yang mengukur seberapa akurat deteksi objek pada suatu *dataset*. Perhitungan nilai kepercayaan kotak, atau nilai kepercayaan, untuk setiap kotak dijelaskan oleh gambar dan rumus dibawah yang menetapkan hubungan antara probabilitas kotak mengandung objek dan kesesuaian antara kotak prediksi dan kotak kebenaran dasar.

$$IOU = \frac{\text{area irisan}}{\text{area gabungan}} = \frac{\text{area } (BB_{prediksi} \cap BB_{groundTruth})}{\text{area } (BB_{prediksi} \cup BB_{groundTruth})}$$


**Gambar 2.9 Ilustrasi IOU**

$$IOU = \frac{\text{area } (BB_{prediksi} \cap BB_{groundTruth})}{\text{area } (BB_{prediksi} \cup BB_{groundTruth})}$$

Dalam menetapkan hasil prediksi akhir, faktor utama yang mempengaruhinya adalah nilai skor keyakinan kelas, yang dihasilkan dari probabilitas kondisional kelas dan skor keyakinan kotak. Kedua faktor ini berperan penting dalam mengukur tingkat kepercayaan. Persamaan untuk menghitung skor keyakinan kelas pada setiap kotak prediksi diperlihatkan dalam persamaan dibawah ini

$$\Pr(Class|Object) \cdot \text{box confidence score} = \Pr(class\ i) \cdot IoU_{pred}^{truth}$$

dimana :

$\Pr(Class|object)$  adalah probabilitas kondisional kelas i.

$\Pr(Class\ i)$  adalah probabilitas kelas i

## 2.6 Pengukuran Evaluasi Kinerja

Evaluasi kinerja model klasifikasi melibatkan penggunaan metrik penting yang komprehensif. Dua metrik utama, yakni presisi (*precision*) dan *recall*, sangat penting dalam menilai seberapa baik model bekerja. Presisi mengukur seberapa akurat model

dalam memprediksi hasil positif dari semua prediksi positif yang dibuat, sementara recall menunjukkan kemampuan model untuk menemukan semua hasil positif yang sebenarnya dari total hasil positif yang ada dalam dataset. Selain itu, metrik *mean average precision* (mAP) digunakan untuk memberikan penilaian keseluruhan. Metrik ini mengevaluasi kinerja model dengan membandingkan kotak deteksi model dengan kotak deteksi yang sebenarnya, memberikan skor yang mencerminkan tingkat kesesuaian (Vo et al., 2024).

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$mAP = \frac{1}{N} \sum_{i=1}^n AP_i$$

Dalam hal ini, FP mewakili *False Positive* (Hasil Salah Positif), TN menunjukkan *True Negative* (Hasil Benar Negatif), TP menyatakan *True Positive* (Hasil Benar Positif), dan FN mengindikasikan *False Negative* (Hasil Salah Negatif). AP adalah *Average Precision* (Presisi Rata-rata),  $AP_i$  menunjukkan nilai presisi rata-rata untuk kategori ke- $i$ ,  $N$  adalah jumlah kelas.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis Sistem**

Tahap ini adalah langkah kunci dalam studi mengenai suatu sistem yang sedang beroperasi. Fokus utamanya adalah untuk mengidentifikasi setiap potensi masalah yang mungkin muncul dalam sistem tersebut, sehingga memungkinkan pemahaman yang lebih baik tentang penyebab dan dampak dari setiap masalah yang terjadi. Dari analisis ini, kita dapat memperoleh wawasan yang lebih mendalam tentang latar belakang dari munculnya masalah, yang akan menjadi dasar penting untuk merancang solusi yang sesuai. Dengan mengenali akar masalah, langkah-langkah berikutnya dalam pengembangan sistem dapat dilakukan dengan lebih terfokus dan efektif..

##### *3.1.1 Analisis Masalah*

Pengkajian masalah menjadi tahap penting dalam penelitian ini, dimana tujuannya adalah untuk mengenali dan memahami hambatan utama yang dihadapi. Salah satu tantangan yang muncul adalah minimnya pengetahuan masyarakat tentang berbagai jenis luka ringan dan cara penanganannya di Indonesia. Kondisi ini sering kali mengakibatkan perlakuan yang kurang tepat terhadap luka ringan, yang pada akhirnya dapat menyebabkan komplikasi yang lebih serius.

Selain itu, dalam situasi darurat di mana tidak tersedia tenaga medis yang dapat memberikan bantuan langsung, masyarakat seringkali tidak memiliki pengetahuan yang cukup untuk mengidentifikasi jenis luka ringan dan memberikan penanganan awal yang diperlukan dengan benar. Ini menjadi masalah besar karena dapat memperburuk kondisi luka dan mempengaruhi kesehatan secara keseluruhan.

Karenanya, untuk meningkatkan pemahaman dan perawatan terhadap luka ringan di masyarakat Indonesia, diperlukan solusi yang dapat memberikan bantuan yang cepat dan tepat. Salah satu pendekatan yang diusulkan adalah pengembangan aplikasi berbasis deteksi objek *You Only Look Once* (YOLO) yang memanfaatkan arsitektur *Convolutional Neural Network* (CNN). Aplikasi ini diharapkan dapat

mengidentifikasi jenis luka ringan dari gambar digital dan memberikan informasi yang relevan mengenai penanganan awal dan rekomendasi obat. Dengan demikian, diharapkan aplikasi ini dapat memberikan kontribusi signifikan dalam meningkatkan pemahaman dan perawatan terhadap luka ringan di masyarakat Indonesia.

### *3.1.2 Analisis Kebutuhan*

Analisis kebutuhan adalah tahap krusial dalam perancangan sistem untuk merawat luka ringan, yang bertujuan untuk mengidentifikasi dan memahami elemen-elemen yang diperlukan. Proses ini terdiri dari dua tahap utama pertama, identifikasi kebutuhan pengguna, yang menitikberatkan pada kebutuhan fungsional dan non-fungsional dari perspektif pengguna kedua, penetapan kebutuhan sistem, di mana kebutuhan pengguna diubah menjadi spesifikasi teknis yang dapat diimplementasikan. Sasaran utama dari analisis kebutuhan adalah memastikan bahwa sistem dapat memenuhi harapan pengguna, mengatasi masalah perawatan luka ringan, dan mencapai hasil yang diinginkan.

#### *3.1.2.1 Kebutuhan Fungsional*

Kebutuhan fungsional merujuk pada serangkaian tindakan dan fungsi yang harus dilakukan oleh sistem untuk mendeteksi luka ringan dan memberikan manajemen perawatan luka untuk merawat luka ringan secara efektif. Fitur-fitur fungsional dari sistem ini termasuk:

1. Sistem dapat mengidentifikasi jenis-jenis luka ringan, seperti luka tusuk, luka gores, lecet, dan luka bakar.
2. Sistem dapat memberikan petunjuk langkah demi langkah tentang penanganan awal terkait luka ringan yang telah diidentifikasi.
3. Sistem harus memberikan rekomendasi obat yang relevan terkait dengan luka ringan yang telah diidentifikasi .

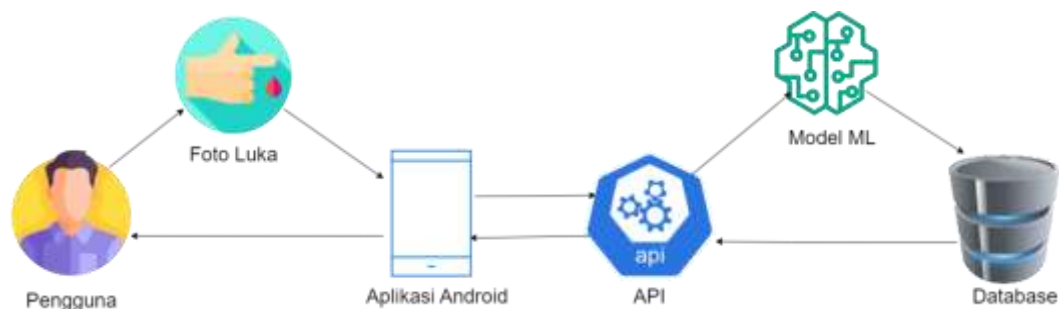
#### *3.1.2.2 Kebutuhan Non-Fungsional*

Mencakup fitur-fitur tambahan dari sistem yang tidak secara langsung berkaitan dengan fungsi inti perawatan luka ringan. Beberapa fitur tambahan dari sistem ini, di luar dari fungsi utamanya, meliputi:

1. Sistem memiliki antarmuka pengguna yang ramah dan mudah digunakan, sehingga pengguna dapat dengan cepat mengakses informasi yang mereka butuhkan tentang perawatan luka ringan.
2. Internet dibutuhkan untuk terhubung dengan sistem.
3. Sistem dapat menjawab pertanyaan *user* seputar luka ringan dan konteks kesehatan melalui fitur *chatbot*
4. Batasan luka ringan yang dapat terdeteksi yaitu luka tusuk, luka sayat, luka gores dan luka bakar yang perawatannya dapat dilakukan secara mandiri di rumah.

### 3.2 Arsitektur Umum

Arsitektur umum adalah bentuk susunan, rancangan dan desain dari sistem yang hendak dibuat. Rancangan arsitektur umum dari sistem, sebagai berikut :



**Gambar 3.1 Arsitektur Umum**

1. Pengguna memulai proses dengan membuka aplikasi di perangkat mobile mereka.
2. Pengguna menggunakan perangkat Android untuk mengambil foto luka secara langsung atau dengan cara di unggah.
3. Foto luka dikirim ke API untuk diproses lebih lanjut.
4. API menggunakan model deteksi untuk menganalisis foto luka dan mengidentifikasi jenis luka ringan
5. Berdasarkan hasil deteksi, API mencari pada *database* untuk menemukan informasi tentang penanganan awal yang diperlukan dan memberikan rekomendasi obat sesuai dengan jenis luka yang diidentifikasi.
6. Informasi penanganan awal dan rekomendasi obat dikembalikan ke API.
7. API mengirim informasi kembali ke perangkat *Android* pengguna

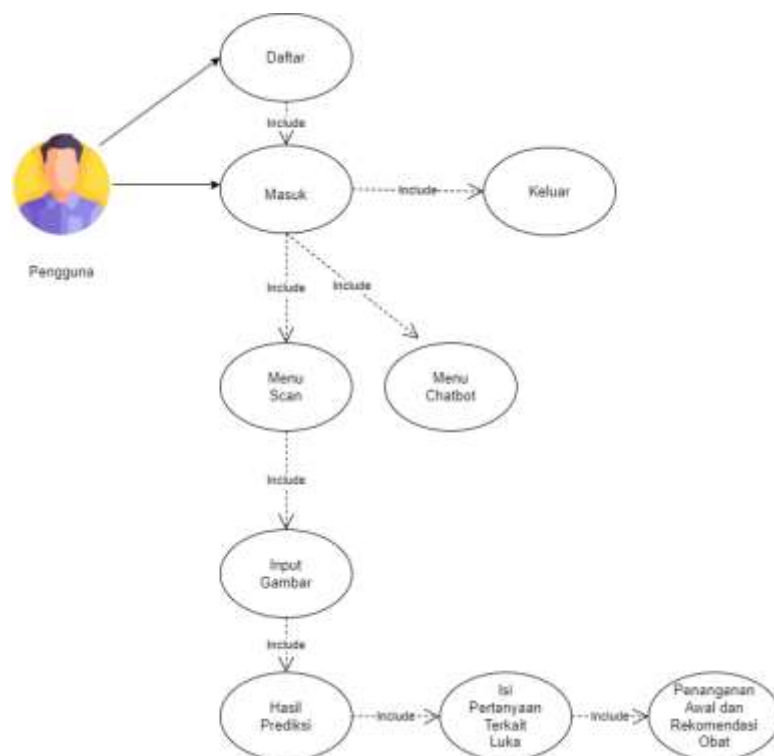
8. Aplikasi *mobile* menampilkan informasi penanganan awal dan rekomendasi obat kepada pengguna.

### 3.3 Pemodelan Sistem

Langkah-langkah dalam menguraikan interaksi antara pengguna dan aplikasi yang dibangun untuk memastikan operasional sistem secara efisien seringkali dimodelkan dalam format *Unified Modeling Language* (UML), yang merupakan bahasa yang paling umum digunakan untuk memodelkan sistem, memungkinkan penggambaran hubungan antara komponen-komponen sistem yang memfasilitasi interaksi melalui pengguna. Dalam penelitian ini, diagram yang menjadi dasar sebagai model sistem adalah *Activity Diagram*, *Sequence Diagram*, dan *Use Case Diagram*.

#### 3.3.1 Use Case Diagram

Berfungsi menampilkan gambaran konsep rancangan sistem antara interaksi pengguna (*actor*) dan sistem yang dirancang.



**Gambar 3.2 Use Case Diagram**

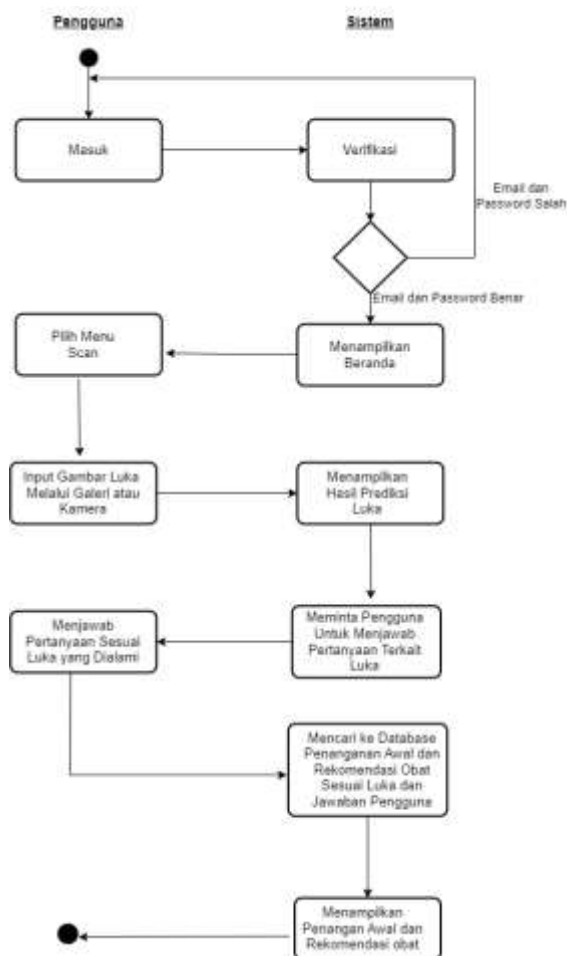
Menurut gambar 3.2, Diagram dimulai ketika pengguna membuka aplikasi. Setelah membuka aplikasi, pengguna akan disambut dengan opsi untuk mendaftar atau masuk. Setelah berhasil masuk, pengguna akan diarahkan langsung ke beranda. Di beranda, pengguna akan diberikan beberapa pilihan, salah satunya adalah "scan".

Setelah memilih opsi "*scan*", pengguna dapat mengimpor gambar dari kamera perangkat atau dari galeri. Setelah gambar diimpor, sistem akan melakukan prediksi terkait luka yang terdapat pada gambar tersebut. Selanjutnya, pengguna akan disajikan dengan pertanyaan-pertanyaan yang relevan terkait luka yang dideteksi. Pengguna dapat menjawab pertanyaan-pertanyaan tersebut.

Setelah menjawab pertanyaan, sistem akan menampilkan informasi mengenai penanganan awal yang tepat dan rekomendasi obat sesuai dengan luka yang terdeteksi.

### 3.3.2 Activity Diagram

Merupakan adalah alat visual yang memperlihatkan alur kegiatan pada rancangan sistem, mulai dari tahap awal ketika aplikasi dijalankan, selama proses berlangsung, hingga tahap selesai. Dan menjadi representasi grafis yang mengilustrasikan urutan langkah-langkah yang diambil dalam sistem, memungkinkan untuk pemahaman yang lebih baik tentang proses yang terlibat dalam penggunaan aplikasi atau sistem tersebut.



Gambar 3.3 Activity Diagram

Diagram aktivitas sistem, yang disajikan dalam Gambar 3.3, menyoroti alur langkah-langkah dari awal hingga akhir dalam penggunaan aplikasi. Pengguna pertama kali diminta untuk masuk dengan memasukkan email dan password. Setelah itu, sistem akan memverifikasi kebenaran kombinasi email dan password yang dimasukkan. Jika verifikasi berhasil, pengguna akan diarahkan ke beranda aplikasi oleh sistem.

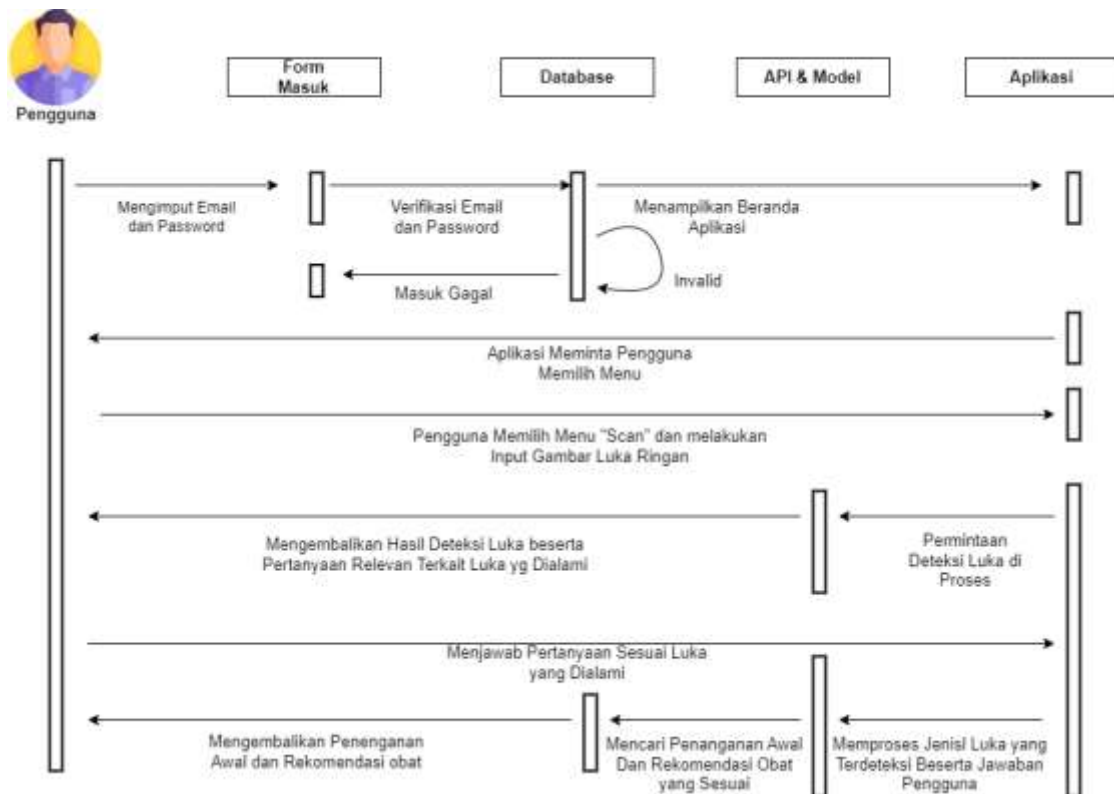
Selanjutnya, pengguna memiliki opsi untuk memilih menu "*scan*". Setelah memilih opsi ini, pengguna dapat memilih untuk mengimpor gambar melalui kamera perangkat atau langsung dari galeri. Setelah gambar diimpor, sistem akan melakukan prediksi terkait luka yang terdeteksi dan akan menampilkan pertanyaan-pertanyaan yang relevan kepada pengguna terkait luka tersebut.

Pengguna kemudian dapat menjawab pertanyaan-pertanyaan tersebut. Setelah pengguna menjawab, sistem akan mencari dalam *database* untuk menemukan informasi terkait penanganan awal dan rekomendasi obat yang sesuai dengan luka dan jawaban yang dijawab oleh pengguna. Informasi ini akan ditampilkan kepada pengguna.

### 3.3.3 Sequence Diagram

Merupakan salah satu jenis diagram interaksi dalam bahasa pemodelan UML (*Unified Modeling Language*) yang mengilustrasikan bagaimana objek-objek dalam sistem berinteraksi satu sama lain dalam urutan waktu yang ditentukan. Diagram ini fokus pada urutan pesan yang dikirim antara objek-objek tersebut, serta kapan pesan tersebut dikirimkan dan diterima. Diagram urutan sangat berguna untuk memvisualisasikan aliran kerja atau proses dalam sistem, serta untuk memahami bagaimana objek-objek berkomunikasi satu sama lain dalam konteks skenario tertentu. Dengan menggunakan diagram urutan, pengembang atau analis sistem dapat lebih mudah memahami interaksi antar objek dan mengidentifikasi potensi masalah atau kebutuhan pengembangan tambahan dalam sistem.





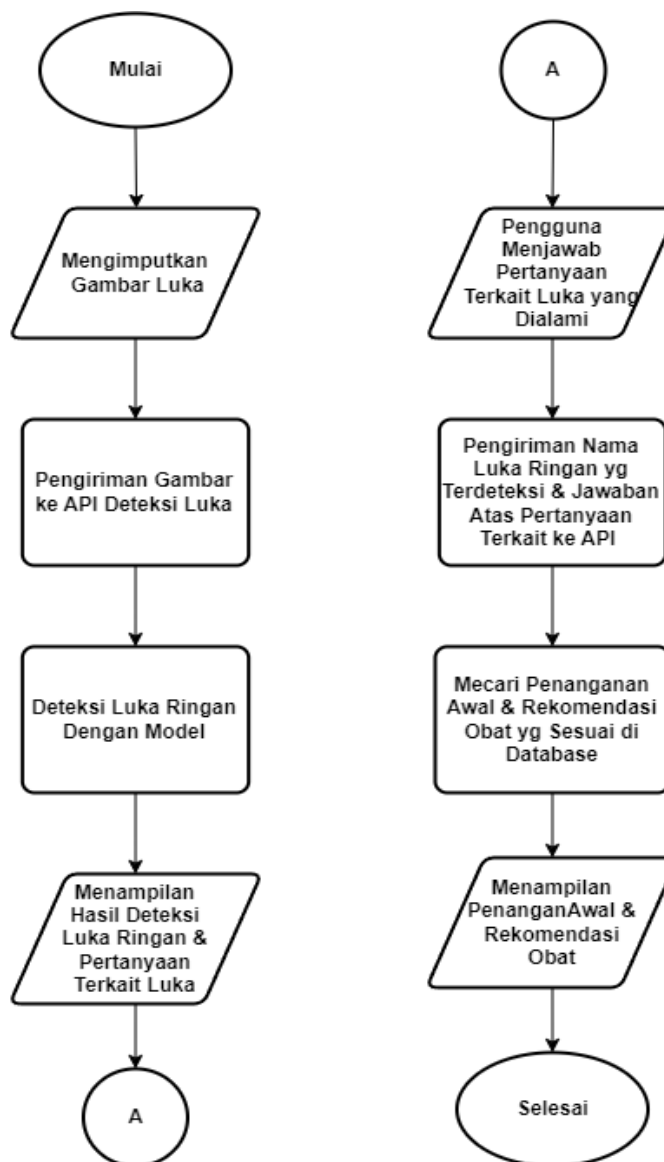
**Gambar 3.4 Sequence Diagram**

Gambar 3.4 mengilustrasikan hubungan dinamis antara lima komponen yang saling berinteraksi secara berurutan. Gambaran ini memerinci proses mulai dari pengguna masuk ke dalam aplikasi hingga permintaan diproses sepenuhnya. Dengan fokus pada komunikasi antar objek atau komponen dalam konteks skenario atau proses tertentu, visualisasi ini memberikan pemahaman yang jelas tentang aliran kerja sistem dalam konteks waktu yang ditentukan.

### 3.4 Flowchart (Diagram Alir)

Merupakan gambaran visual yang menggambarkan urutan dan output dari proses dalam suatu sistem. Pendekatan utama dalam merancang flowchart adalah untuk menyajikan solusi dari suatu masalah secara terperinci, ringkas, mudah dipahami, dan logis, menggunakan simbol-simbol yang umum digunakan dalam penggambaran proses.

### 3.4.1 Flowchart Sistem

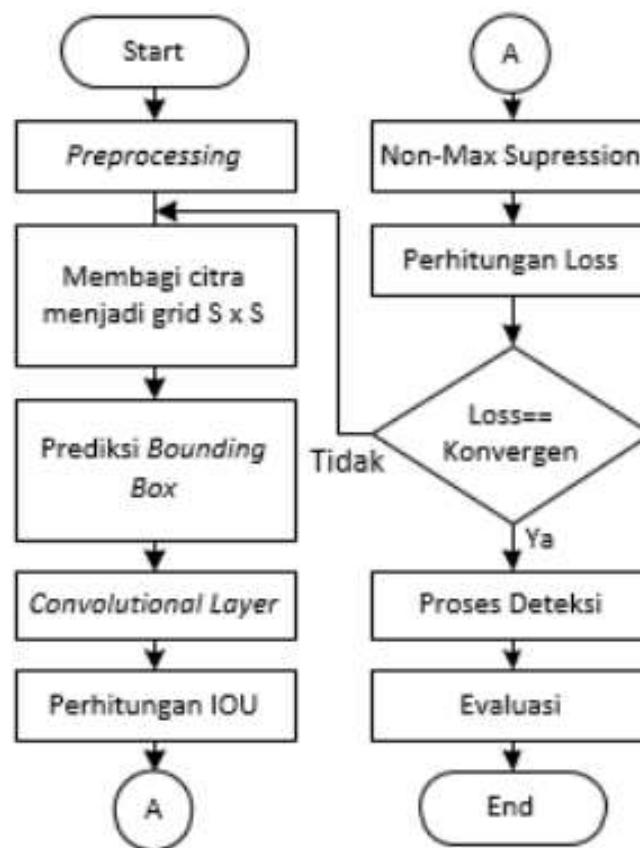


**Gambar 3.5 Flowchart Sistem**

Diagram alir pada gambar 3.5 menggambarkan proses sistem secara bertahap dengan sederhana. Tahapan pertama adalah memasukkan gambar luka ke dalam sistem. Selanjutnya, gambar tersebut akan disampaikan ke API untuk diproses dan diteruskan ke model deteksi. Model deteksi akan menghasilkan informasi mengenai luka yang terdeteksi beserta pertanyaan yang relevan terkait luka ringan yang teridentifikasi.

Setelah itu, pengguna akan memberikan jawaban atas pertanyaan yang sesuai dengan kondisi luka yang dialami. Nama luka ringan dan jawaban yang diberikan akan dikirim kembali ke API untuk mencari penanganan awal yang sesuai serta merekomendasikan obat yang tepat. Selanjutnya, hasil penanganan awal dan rekomendasi obat akan ditampilkan kepada pengguna melalui sistem.

### 3.4.2 Flowchart Deteksi Object YOLO



**Gambar 3.6 Flowchart Deteksi Object YOLO**

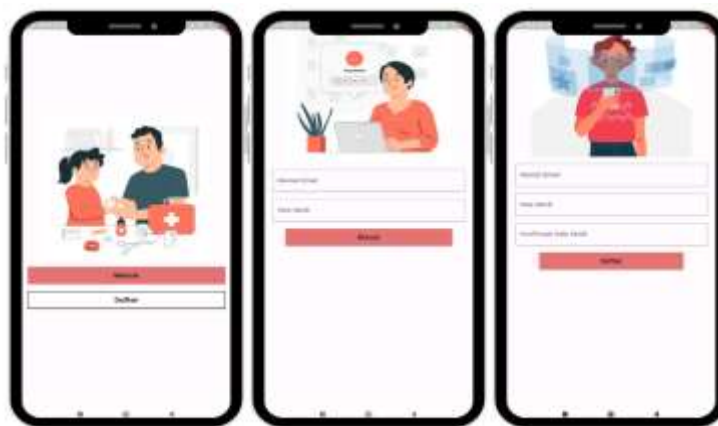
Diagram alur deteksi objek menggunakan metode YOLO (*You Only Look Once*) dimulai dengan tahapan *preprocessing*, di mana citra disiapkan untuk deteksi objek. Selanjutnya, citra dibagi menjadi *grid* dengan ukuran  $s \times s$  untuk memfasilitasi deteksi objek di setiap *grid*. Setelah itu, dilakukan prediksi *bounding box* untuk menemukan objek dalam setiap *grid*, yang diikuti oleh konvolusi untuk mengekstraksi fitur-fitur citra yang diperlukan. Tahapan selanjutnya adalah menghitung nilai IoU (*Intersection over Union*) antara *bounding box* hasil prediksi dengan *ground truth* untuk mengevaluasi akurasi deteksi. Kemudian, dilakukan seleksi *non-maximal* untuk menghapus *bounding box* yang tumpang tindih dan mempertahankan yang paling relevan. Setelah itu, nilai *loss* dihitung untuk mengevaluasi performa deteksi, dan jika nilai *loss* tidak konvergen, proses akan kembali ke tahap awal pembagian citra menjadi *grid*. Proses deteksi objek kemudian dilakukan dengan memanfaatkan hasil prediksi *bounding box* yang telah disesuaikan. Akhirnya, hasil deteksi dievaluasi untuk mengukur kinerja sistem deteksi sebelum proses diakhiri.

### 3.5 Perancangan *Interface*

Antarmuka, yang sering disebut sebagai *Interface*, telah menjadi bagian integral dari sistem dengan peran utama sebagai media interaksi antara pengguna dengan sistem. Dalam proses merancang sebuah antarmuka, penting untuk memiliki pemahaman yang jelas tentang struktur sistem yang terkait dengan antarmuka yang akan dikembangkan sebelum melangkah ke tahap implementasi.

Rancangan antarmuka sistem memerlukan pendekatan yang memperhatikan kebutuhan pengguna dengan menawarkan tampilan yang sederhana. Hal ini bertujuan untuk memberikan kenyamanan serta mempermudah pengguna dalam berinteraksi dengan sistem tersebut.

#### 3.5.1 *Desain Halaman Login (Masuk dan Daftar)*



**Gambar 3.7 Desain Halaman *Login***

Gambar 3.7 diatas menggambarkan antarmuka sederhana untuk halaman *login*, yang terdiri dari dua segmen utama yaitu "Masuk" dan "Daftar". Pada bagian "Masuk", pengguna akan diminta untuk memasukkan alamat email beserta kata sandi yang terkait dengan akun untuk mengakses platform atau layanan. Bagian ini bertujuan untuk memberikan pengalaman yang mudah dan intuitif bagi pengguna yang sudah memiliki akun.

Sementara itu, pada bagian "Daftar", pengguna diberi kesempatan untuk mendaftarkan akun baru dengan cara memasukkan alamat email yang *valid* dan membuat kata sandi yang aman. Ini memberikan pengguna yang baru kesempatan untuk bergabung dengan platform atau layanan tersebut dengan memberikan informasi yang diperlukan.

Desain halaman *login* ini memiliki fokus pada kejelasan dan kemudahan penggunaan. *Input* yang diminta (alamat email dan kata sandi) ditempatkan dengan jelas di setiap bagian untuk memudahkan pengguna dalam mengisi informasi yang diperlukan. Selain itu, pengguna juga diberikan pilihan yang jelas antara "Masuk" dan "Daftar", sehingga mereka dapat dengan mudah menentukan langkah yang ingin pengguna ambil sesuai dengan kebutuhan pengguna saat ini.

### 3.5.2 Desain Beranda Aplikasi



**Gambar 3.8 Desain Beranda Aplikasi**

Gambar 3.8 di atas memperlihatkan antarmuka sederhana untuk halaman beranda aplikasi. Pada halaman ini, pengguna disambut dengan beberapa artikel atau bacaan yang berkaitan dengan topik seperti batasan luka ringan yang dapat diobati di rumah. Artikel-Artikel ini bertujuan untuk menyediakan informasi yang bernilai dan bermanfaat kepada pengguna. Selain itu, halaman beranda juga menampilkan artikel kesehatan yang lebih umum yang dapat dibaca oleh pengguna. Artikel ini dapat mencakup berbagai topik kesehatan, mulai dari tips menjaga kesehatan hingga informasi tentang penyakit tertentu atau perawatan medis.

Pada bagian bawah halaman, terdapat *navigation bar* yang mencakup tautan menuju beranda, fitur pemindaian (*scan*), dan *chat bot*. Fitur-fitur ini dirancang untuk memberikan akses yang mudah bagi pengguna ke berbagai bagian penting dari aplikasi. Menu beranda memungkinkan pengguna untuk kembali ke halaman utama kapan saja,

fitur scan memberikan akses untuk menginput gambar untuk deteksi luka ringan, dan *chat bot* menyediakan layanan interaktif untuk menjawab pertanyaan pengguna atau memberikan bantuan dalam waktu nyata.

### 3.5.3 Desain Artikel Kesehatan



**Gambar 3.9 Desain Halaman Artikel Kesehatan**

Gambar 3.9 diatas menggambarkan desain yang bersih dan minimalis untuk halaman artikel kesehatan. Pada halaman ini, setiap artikel ditampilkan dengan judul yang jelas, gambar yang relevan, serta sumber artikel dan isi utama artikel tersebut. Setiap artikel diwakili oleh gambar yang mengilustrasikan topik yang dibahas dalam artikel tersebut. Gambar ini bertujuan untuk menarik perhatian pengguna dan memberikan gambaran visual tentang topik yang akan dibahas. Di bawah gambar, terdapat judul artikel yang ditampilkan dengan ukuran font yang cukup besar untuk memudahkan pembacaan dan penafsiran. Judul ini dirancang untuk mencerminkan inti dari artikel tersebut, sehingga pengguna dapat dengan cepat memperoleh pemahaman terkait topik yang akan dibahas.

Selain itu, setiap artikel juga mencantumkan sumbernya hal ini memberikan kredibilitas dan transparansi terhadap informasi yang disajikan kepada pengguna. Isi artikel ditampilkan di bawah judul dan sumber, dengan teks yang mudah dibaca dan dipahami. Isi artikel dirancang untuk memberikan informasi yang bermanfaat dan relevan tentang topik kesehatan yang dibahas.

### 3.5.4 Desain Artikel Batasan Luka



**Gambar 3.10 Desain Halaman Artikel Batasan Luka**

Gambar 3.10 diatas menampilkan desain sederhana dari halaman artikel batasan luka. Dimana dalam artikel ini menampilkan ilustrasi yang relevan dengan topik artikel, seperti gambar luka atau ikon yang mencerminkan jenis luka yang dibahas. Gambar ini bertujuan untuk memberikan gambaran visual singkat kepada pembaca tentang topik yang akan dibahas dalam artikel. Di bawah gambar, terdapat judul singkat yang mencakup jenis luka yang dibahas, seperti "Luka Bakar" atau "Luka Sayat", yang diikuti oleh informasi tentang batasannya. Informasi ini disajikan secara singkat dan jelas untuk memberikan pemahaman awal kepada pembaca tentang topik artikel.

Isi artikel kemudian disajikan di bawah judul dan informasi singkat. Isi artikel ini dirancang untuk memberikan informasi yang relevan dan penting tentang jenis luka yang dibahas, termasuk penjelasan mengenai batasannya

### 3.5.5 Desain Halaman Scan



**Gambar 3.11 Desain Halaman *Scan* Aplikasi**

Halaman *scan* menampilkan desain yang minimalis dengan fokus pada fungsi utama, yaitu pemindaian foto luka ringan. Terdapat dua opsi yang ditampilkan dengan jelas di tengah halaman, yaitu tombol kamera dan tombol galeri. Tombol kamera memungkinkan pengguna untuk secara langsung mengambil foto luka ringan yang sedang dialami menggunakan kamera perangkat mereka. Setelah menekan tombol kamera, aplikasi akan membuka antarmuka kamera perangkat untuk pengguna agar dapat mengambil gambar luka dengan mudah.

Tombol galeri memberikan opsi kepada pengguna untuk memilih foto luka ringan yang sudah ada di galeri perangkat mereka. Ketika tombol galeri ditekan, aplikasi akan membuka galeri perangkat pengguna, memungkinkan mereka untuk memilih foto yang ingin dipindai untuk deteksi.



### 3.5.6 Desain Halaman Hasil Deteksi Luka



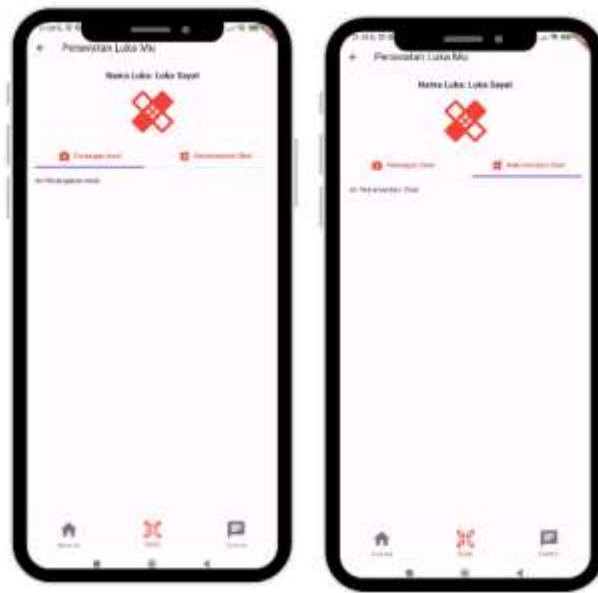
**Gambar 3.12 Desain Halaman Hasil Deteksi Luka**

Halaman deteksi luka menampilkan gambar luka yang sebelumnya diinputkan oleh pengguna, disertai dengan hasil prediksi luka ringan berdasarkan analisis gambar tersebut. Gambar luka dan hasil prediksi ditampilkan secara jelas di bagian atas halaman untuk memudahkan pengguna melihat hasilnya.

Di bawah gambar luka dan hasil prediksi, terdapat dua pertanyaan terkait kondisi luka yang terdeteksi. Pertanyaan ini dirancang untuk membantu pengguna dalam memverifikasi atau memberikan informasi tambahan tentang kondisi luka yang dialami. Pengguna akan diberikan opsi untuk menjawab pertanyaan yang diberikan sesuai dengan kondisi luka yang pengguna alami.

Pengguna diberikan kebebasan untuk menjawab pertanyaan sesuai dengan pengalaman mereka sendiri. Jawaban pengguna akan memberikan informasi tambahan yang dapat digunakan untuk meningkatkan keakuratan manajemen perawatan luka yang diberikan oleh aplikasi.

### 3.5.7 Desain Halaman Penanganan Awal dan Rekomendasi Obat



**Gambar 3.13 Desain Halaman Penanganan Awal dan Rekomendasi Obat**

Untuk halaman penanganan awal dan rekomendasi obat memiliki dua tab yang jelas yaitu "Penanganan Awal" dan "Rekomendasi Obat". Jika pengguna memilih tab "Penanganan Awal", mereka akan melihat informasi tentang langkah-langkah pertolongan pertama untuk luka ringan yang terdeteksi. Sementara itu, jika mereka memilih tab "Rekomendasi Obat", mereka akan melihat obat-obatan yang direkomendasikan untuk pengobatan luka ringan terkait. Dengan menyusun informasi dalam dua tab yang terpisah, pengguna dapat dengan mudah beralih antara panduan penanganan awal dan rekomendasi obat sesuai dengan kebutuhan mereka. Desain ini bertujuan untuk memberikan pengalaman yang mudah dipahami dan digunakan bagi pengguna tanpa tambahan detail yang rumit.

### 3.5.8 Desain Halaman Chatbot



**Gambar 3.14 Desain Halaman *Chatbot* Aplikasi**

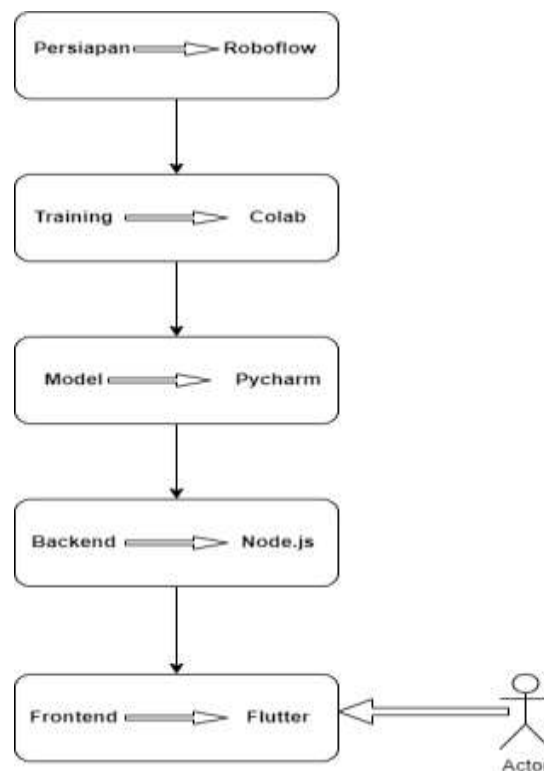
Halaman *chatbot* adalah tempat di mana pengguna dapat berinteraksi dengan bot melalui pesan teks. Ketika pengguna membuka halaman ini, mereka akan disambut dengan antarmuka chat sederhana yang memungkinkan mereka untuk mulai berkomunikasi dengan bot. Pengguna dapat mengetik pesan atau pertanyaan mereka ke dalam kotak obrolan dan mengirimkannya ke bot. Bot akan merespons dengan cepat dan memberikan jawaban atau informasi yang relevan tergantung pada pertanyaan atau perintah yang diberikan pengguna.

Antarmuka *chatbot* ini dirancang untuk mudah digunakan, dengan fokus pada kemudahan navigasi dan komunikasi yang efisien. Pengguna dapat dengan cepat mengetik pesan mereka dan menerima respons dari bot tanpa perlu meninggalkan halaman atau melakukan navigasi yang rumit.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN SISTEM

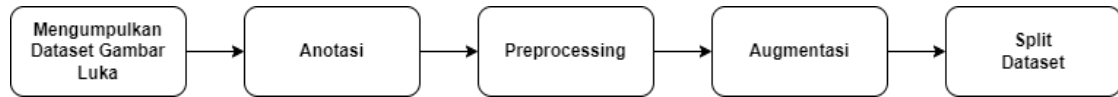
Proses implementasi dimulai dengan menyiapkan *dataset* menggunakan platform Roboflow, di mana data dikumpulkan, diproses, dan diatur agar siap untuk proses pelatihan. Kemudian, model deteksi dilatih menggunakan Google Colab, di mana algoritma pembelajaran mesin diterapkan dan model diperbaiki sesuai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



**Gambar 4.1 Implementasi Sistem**

#### 4.1 Implementasi Tahap Persiapan *Dataset*

Dataset yang digunakan dalam penelitian ini diambil dari berbagai sumber melalui platform dataset publik yaitu Kaggle, Roboflow dan digabung dengan gambar yg diambil dari internet. Proses persiapan dataset secara umum diilustrasikan oleh gambar



**Gambar 4.2** Proses Persiapan Dataset

##### 4.1.1 *Pengumpulan Dataset*

**Tabel 4.1** Pengumpulan *Dataset*

Jenis Luka	Total Gambar	Total Anotasi
Luka Sayat	123	172
Luka Lecet	140	133
Luka Bakar	96	123
Luka Tusuk	62	114
<b>TOTAL</b>	<b>421</b>	<b>542</b>

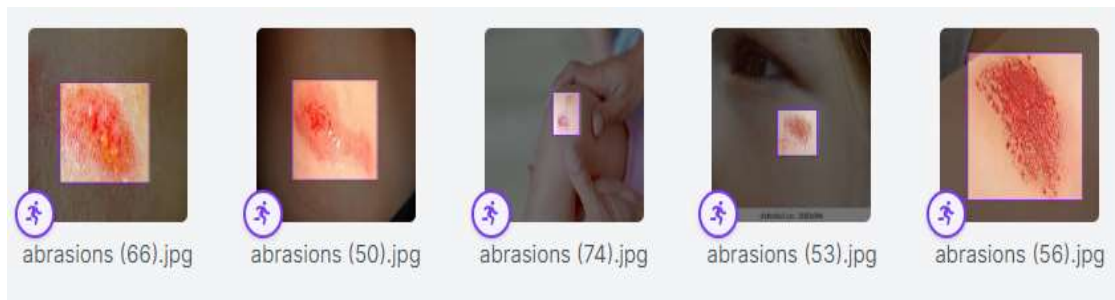
Pada penelitian ini digunakan platform Robloflow untuk mempermudah untuk melabeli gambar dengan cepat dan efesien. Dataset yang tekumpul sebanyak 542 gambar luka, dataset luka tusuk memiliki jumlah yang paling sedikit dikarenakan data dengan jenis luka tersebut sangat sulit ditemukan.Contoh dataset luka sebagai berikut :

Luka Sayat = 172 total anotasi



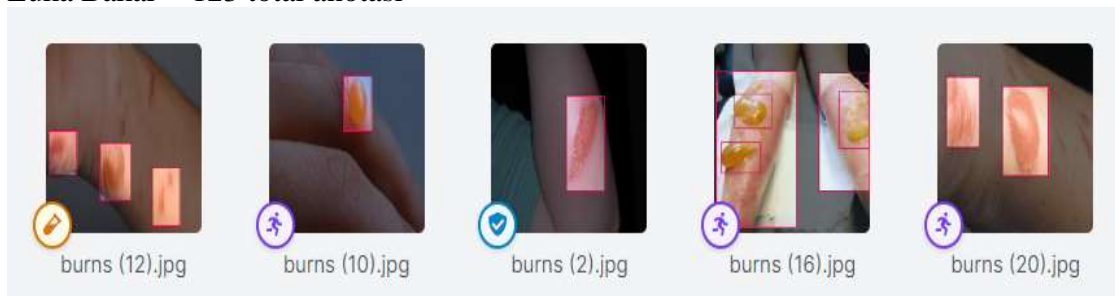
**Gambar 4.3** *Dataset Luka Sayat*

Luka Lecet = 133 total anotasi



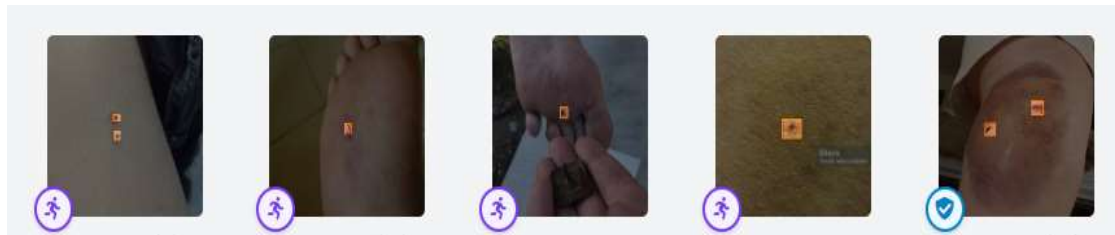
**Gambar 4.4 Dataset Luka Lecet**

Luka Bakar = 123 total anotasi



**Gambar 4.5 Dataset Luka Bakar**

Luka Tusuk = 114 total anotasi



**Gambar 4.6 Dataset Luka Tusuk**

#### 4.1.2 Anotasi Dataset

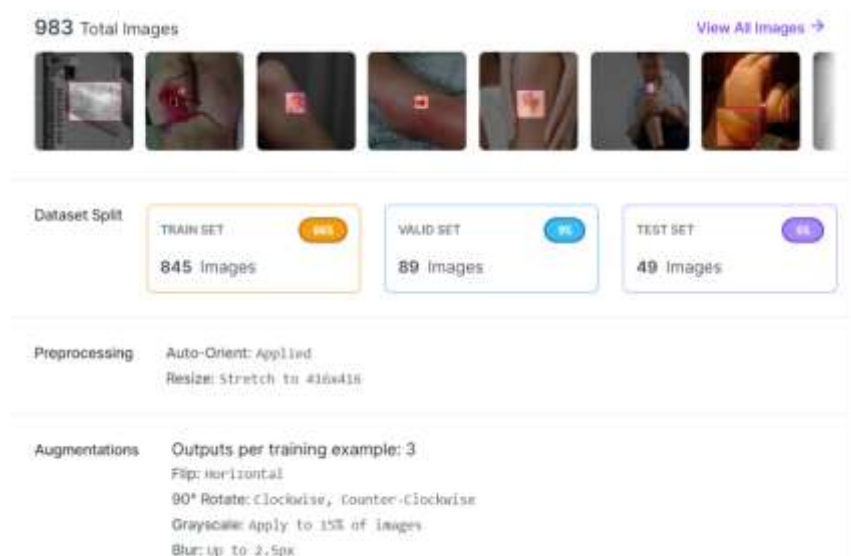
Anotasi adalah proses menandai atau menetapkan kotak pembatas (*bounding box*) untuk objek yang ingin dideteksi dalam gambar. Anotasi ini penting karena YOLO memerlukan data latihan yang sudah ditandai dengan benar untuk melakukan pelatihan model deteksi objek.



**Gambar 4.7 Proses Anotasi**

Proses anotasi dalam penelitian ini menggunakan alat *bounding box* yang disediakan oleh platform *Roboflow*. Langkah ini memungkinkan penandai untuk secara akurat menetapkan kotak pembatas di sekitar objek yang ingin dideteksi dalam gambar. Dengan menggunakan alat 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 anotasi ini memainkan peran kunci dalam memastikan akurasi dan keandalan model deteksi objek yang dihasilkan.

#### 4.1.3 Preprocessing, Augmentasi, Split Dataset



**Gambar 4.8 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 416x416. 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 *flip*, *rotate* dan *blur*. Setelah proses diatas selesai didapatkan 983 total gambar kemudian gambar tersebut dibagi menjadi beberapa bagian set yaitu *Train set* 845 gambar, *Valid Set* 89 gambar dan *Test Set* sebanyak 49 gambar, pembagian ini dipilih dengan mempertimbangkan jumlah *dataset* yang tidak terlalu banyak, dan pembagian nya dilakukan menggunakan *K-fold cross-validation* dan di dapatkan dengan kombinasi seperti ini menghasilkan model yang lebih optimal.

## 4.2 Implementasi Tahap Pelatihan Model

Setelah tahap persiapan data dan memastikan dataset siap dipakai, maka langkah selanjutnya adalah melatih .

### 4.2.1 Mengakses Dataset



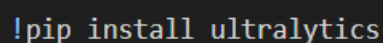
```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="ax0SHZPiSU0yGG8evpRu")
project = rf.workspace("universitas-sumatera-utara-h9u4i").project("ceklukaid")
version = project.version(3)
dataset = version.download("yolov8")
```

**Gambar 4.9 Akses ke Dataset**

Gambar 4.9 diatas merupakan proses mengakses dataset yang dihosting di *Roboflow* menggunakan pustaka *Roboflow Python*, yang kemudian dapat digunakan untuk melatih model atau melakukan berbagai tugas pengolahan data lainnya.

### 4.2.2 Mengunduh Pustaka Ultralytics



```
!pip install ultralytics
```

**Gambar 4.10 Unduh Pustaka Ultralytics**

Gambar 4.10 diatas merupakan proses mengunduh Pustaka *Ultralytics* sehingga dapat mengakses fungsionalitasnya dan menggunakan alat dan model yang disediakan.



#### 4.2.3 Import Kelas Yolo

```
from ultralytics import YOLO
```

**Gambar 4.11 Import YOLO**

Gambar 4.11 diatas merupakan proses mengimpor kelas YOLO dari modul *ultralytics*. Ini adalah proses impor yang memungkinkan untuk menggunakan fungsi-fungsi dan objek-objek yang didefinisikan dalam modul YOLO dalam proyek.

#### 4.2.4 Membuat Objek Model YOLO

```
model = YOLO('yolov8n.pt')
```

**Gambar 4.12 Membuat Objek Model YOLO**

Gambar 4.12 diatas adalah proses membuat sebuah objek model dari kelas YOLO dengan menyediakan path atau nama file dari model YOLOv8n yang telah dilatih. Dengan cara ini, objek model yang dibuat akan menggunakan model YOLOv8n yang telah disimpan dalam file 'yolov8n.pt' untuk melakukan deteksi objek pada gambar atau video yang diberikan. Selanjutnya menggunakan objek model ini untuk melakukan berbagai tugas deteksi objek, seperti mendeteksi objek dalam gambar, mengambil bounding box dari objek yang terdeteksi, dan melakukan tugas-tugas lain yang relevan dengan deteksi objek menggunakan model YOLOv8n.

#### 4.2.5 Tahap Pelatihan Model

```
results = model.train(data='/content/CekLukaID-3/data.yaml', pretrained=True, batch=10, device=0, epochs=100, imgsz=416)
```

**Gambar 4.13 Pelatihan Model**

Gambar 4.13 diatas merupakan proses yang mengacu pada pelatihan model deteksi objek YOLOv8n menggunakan pustaka *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 ukuran batch, jumlah *epochs*, dan ukuran gambar, serta parameter tambahan seperti penggunaan bobot yang telah dilatih sebelumnya. Proses pelatihan dilakukan menggunakan dataset yang telah ditentukan dan hasilnya disimpan dalam variabel *results*.

#### 4.2.6 Menyimpan Model

```
model.save('best.pt')
```

**Gambar 4.14 Menyimpan Model**

Gambar 4.14 diatas berisi baris kode yg digunakan untuk menyimpan model yang telah dilatih ke dalam file dengan format yang sesuai. Dalam contoh ini, model disimpan dalam file dengan nama 'best.pt'.

#### 4.3 Tuning Hyperparameter dan Pengujian Model

Dalam penelitian ini, penyetelan *hyperparameter* dilakukan melalui kombinasi berbagai nilai jumlah *epoch* (100, 150, 200,250) dan *learning rate* (0.01, 0.001). Kombinasi-kombinasi ini dieksplorasi secara cermat untuk menemukan model terbaik dan optimal. Hasil *tuning hyperparameter* dapat dilihat dalam tabel di bawah ini, yang menyajikan kombinasi yang diuji beserta performa hasilnya

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

epoch	box_loss	cls_loss	dfl_loss	Precision	Recall	mAP
1/100	1.856	3.371	1.77	0.45	0.157	0.184
2/100	1.869	2.706	1.74	0.242	0.308	0.151
3/100	1.878	2.673	1.808	0.666	0.159	0.206
4/100	1.939	2.643	1.794	0.194	0.175	0.123
5/100	1.97	2.555	1.812	0.26	0.125	0.117
...	...	...	...	...	...	...
96/100	0.9242	0.5958	1.108	0.612	0.546	0.55
97/100	0.9155	0.5904	1.086	0.641	0.533	0.539
98/100	0.9111	0.5899	1.094	0.69	0.519	0.573
99/100	0.9132	0.5823	1.086	0.669	0.527	0.547
100/100	0.9076	0.5806	1.101	0.669	0.533	0.547

**Tabel 4.3 Summary Training Epoch 100, Learning Rate 0.01**

Class	Precision	Recall	mAP
all	0.663	0.539	0.58
Luka Bakar	0.754	0.542	0.679
Luka Lecet	0.499	0.724	0.606
Luka Sayat	0.645	0.508	0.619
Luka Tusuk	0.754	0.381	0.415

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

<b>epoch</b>	<b>box_loss</b>	<b>cls_loss</b>	<b>dfl_loss</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>1/150</b>	1.856	3.371	1.77	0.45	0.157	0.184
<b>2/150</b>	1.884	2.725	1.756	0.227	0.235	0.139
<b>3/150</b>	1.894	2.676	1.791	0.606	0.159	0.125
<b>4/150</b>	1.926	2.677	1.81	0.45	0.169	0.149
<b>5/150</b>	1.952	2.578	1.759	0.598	0.115	0.153
...	...	...	...	...	...	...
<b>146/150</b>	0.7798	0.4967	1.001	0.545	0.598	0.593
<b>147/150</b>	0.7887	0.4882	1.001	0.662	0.526	0.592
<b>148/150</b>	0.7889	0.492	1.005	0.608	0.571	0.596
<b>149/150</b>	0.7827	0.4857	1.01	0.599	0.565	0.597
<b>150/150</b>	0.7703	0.4931	1.006	0.61	0.584	0.617

**Tabel 4.5 Summary Training Epoch 150, Learning Rate 0.01**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>all</b>	0.626	0.614	0.616
<b>Luka Bakar</b>	0.679	0.625	0.644
<b>Luka Lecet</b>	0.505	0.704	0.642
<b>Luka Sayat</b>	0.731	0.651	0.707
<b>Luka Tusuk</b>	0.59	0.476	0.471

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

<b>epoch</b>	<b>box_loss</b>	<b>cls_loss</b>	<b>dfl_loss</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>1/200</b>	1.017	0.7678	1.151	0.643	0.479	0.543
<b>2/200</b>	1.088	0.8618	1.188	0.573	0.514	0.499
<b>3/200</b>	1.191	0.9546	1.254	0.43	0.45	0.397
<b>4/200</b>	1.262	1.02	1.286	0.623	0.586	0.53
<b>5/200</b>	1.303	1.115	1.3	0.482	0.467	0.433
...	...	...	...	...	...	...
<b>196/200</b>	0.5833	0.3539	0.9075	0.633	0.571	0.561
<b>197/200</b>	0.601	0.3684	0.9185	0.609	0.575	0.553
<b>198/200</b>	0.5916	0.3642	0.9138	0.582	0.584	0.554
<b>199/200</b>	0.5822	0.3511	0.9121	0.633	0.565	0.572
<b>200/200</b>	0.5951	0.36	0.8996	0.657	0.538	0.562

**Tabel 4.7 Summary Training Epoch 200, Learning Rate 0.01**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>all</b>	0.676	0.557	0.604
<b>Luka Bakar</b>	0.734	0.542	0.679
<b>Luka Lecet</b>	0.529	0.586	0.622
<b>Luka Sayat</b>	0.829	0.721	0.719
<b>Luka Tusuk</b>	0.614	0.381	0.396

**Tabel 4.8 Training Epoch 100, Learning Rate 0.001**

<b>epoch</b>	<b>box_loss</b>	<b>cls_loss</b>	<b>dfl_loss</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>1/100</b>	1.856	3.371	1.77	0.45	0.157	0.184
<b>2/100</b>	1.869	2.706	1.74	0.241	0.308	0.152
<b>3/100</b>	1.878	2.673	1.808	0.666	0.159	0.206
<b>4/100</b>	1.939	2.643	1.794	0.193	0.175	0.123
<b>5/100</b>	1.97	2.555	1.812	0.26	0.125	0.117
...	...	...	...	...	...	...
<b>96/100</b>	0.9242	0.5958	1.108	0.612	0.546	0.55
<b>97/100</b>	0.9155	0.5904	1.086	0.641	0.533	0.54
<b>98/100</b>	0.9111	0.5899	1.094	0.69	0.519	0.573
<b>99/100</b>	0.9132	0.5823	1.086	0.669	0.527	0.547
<b>100/100</b>	0.9076	0.5806	1.101	0.669	0.533	0.548

**Tabel 4.9 Summary Training Epoch 100, Learning Rate 0.001**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>all</b>	0.664	0.538	0.58
<b>Luka Bakar</b>	0.756	0.542	0.679
<b>Luka Lecet</b>	0.5	0.724	0.606
<b>Luka Sayat</b>	0.645	0.507	0.619
<b>Luka Tusuk</b>	0.755	0.381	0.415

**Tabel 4.10 Training Epoch 150, Learning Rate 0.001**

<b>epoch</b>	<b>box_loss</b>	<b>cls_loss</b>	<b>dfl_loss</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>1/150</b>	1.856	3.371	1.77	0.45	0.157	0.184
<b>2/150</b>	1.884	2.725	1.756	0.227	0.235	0.139
<b>3/150</b>	1.894	2.676	1.791	0.606	0.159	0.125
<b>4/150</b>	1.926	2.677	1.81	0.45	0.169	0.149
<b>5/150</b>	1.952	2.578	1.759	0.598	0.115	0.153
...	...	...	...	...	...	...
<b>146/150</b>	0.7798	0.4967	1.001	0.545	0.598	0.593
<b>147/150</b>	0.7887	0.4882	1.001	0.662	0.526	0.592
<b>148/150</b>	0.7889	0.492	1.005	0.608	0.571	0.596
<b>149/150</b>	0.7827	0.4857	1.01	0.599	0.565	0.597
<b>150/150</b>	0.7703	0.4931	1.006	0.61	0.584	0.617

**Tabel 4.11 Summary Training Epoch 150, Learning Rate 0.001**

<b>Class</b>	<b>Precision</b>	<b>Recall</b>	<b>mAP</b>
<b>all</b>	0.626	0.614	0.616
<b>Luka Bakar</b>	0.679	0.625	0.644
<b>Luka Lecet</b>	0.505	0.704	0.642
<b>Luka Sayat</b>	0.731	0.651	0.707
<b>Luka Tusuk</b>	0.59	0.476	0.471

**Tabel 4.12 Training Epoch 200, Learning Rate 0.001**

epoch	box_loss	cls_loss	dfl_loss	Precision	Recall	mAP
1/200	1.856	3.361	1.762	0.403	0.207	0.214
2/200	1.84	2.715	1.724	0.334	0.235	0.193
3/200	1.897	2.662	1.804	0.327	0.154	0.121
4/200	1.949	2.63	1.791	0.255	0.306	0.18
5/200	1.99	2.548	1.794	0.214	0.28	0.167
...	...	...	...	...	...	...
196/200	0.71	0.4339	0.973	0.604	0.588	0.587
197/200	0.7068	0.4472	0.9707	0.602	0.586	0.582
198/200	0.7031	0.4409	0.9728	0.604	0.588	0.575
199/200	0.6832	0.4253	0.9577	0.594	0.584	0.578
200/200	0.7156	0.4356	0.9614	0.583	0.578	0.575

**Tabel 4.13 Summary Training Epoch 200, Learning Rate 0.001**

Class	Precision	Recall	mAP
all	0.737	0.545	0.628
Luka Bakar	0.819	0.565	0.73
Luka Lecet	0.576	0.655	0.655
Luka Sayat	0.791	0.628	0.721
Luka Tusuk	0.762	0.333	0.407

**Tabel 4.14 Training Epoch 250, Learning Rate 0.001**

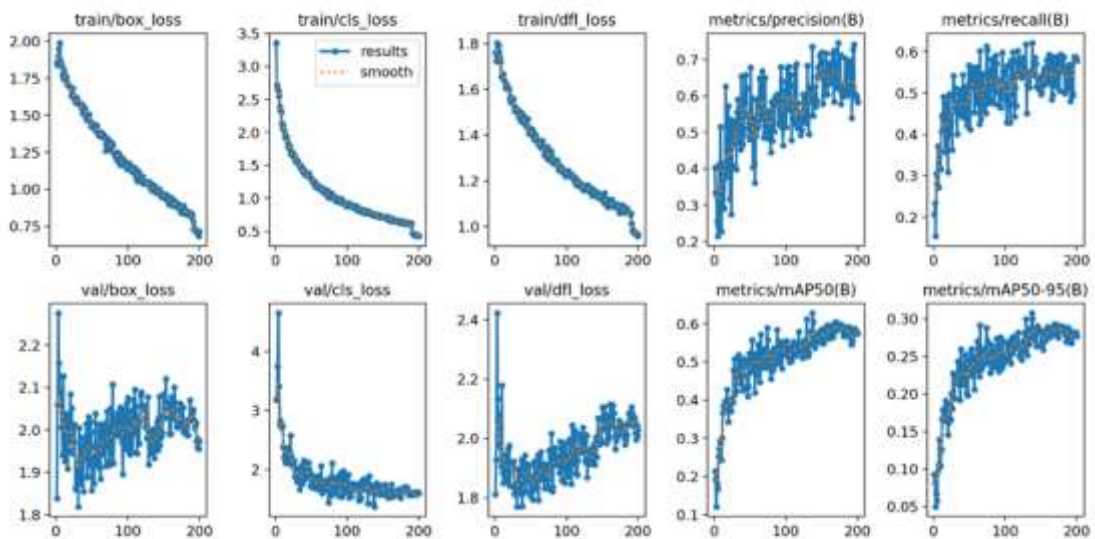
epoch	box_loss	cls_loss	dfl_loss	Precision	Recall	mAP
1/250	2.263	2.216	2.005	0.413	0.353	0.338
2/250	1.789	1.592	1.632	0.532	0.467	0.475
3/250	1.625	1.495	1.545	0.686	0.429	0.496
4/250	1.551	1.444	1.488	0.503	0.526	0.502
5/250	1.569	1.431	1.489	0.386	0.482	0.435
...	...	...	...	...	...	...
231/250	0.726	0.5099	1.007	0.788	0.461	0.549
232/250	0.7327	0.5213	1.004	0.782	0.472	0.564
233/250	0.7363	0.513	1.011	0.781	0.468	0.57
234/250	0.723	0.5252	1.007	0.747	0.466	0.555
235/250	0.7177	0.5155	0.998	0.728	0.476	0.549

**Tabel 4.15 Summary Training Epoch 250, Learning Rate 0.001**

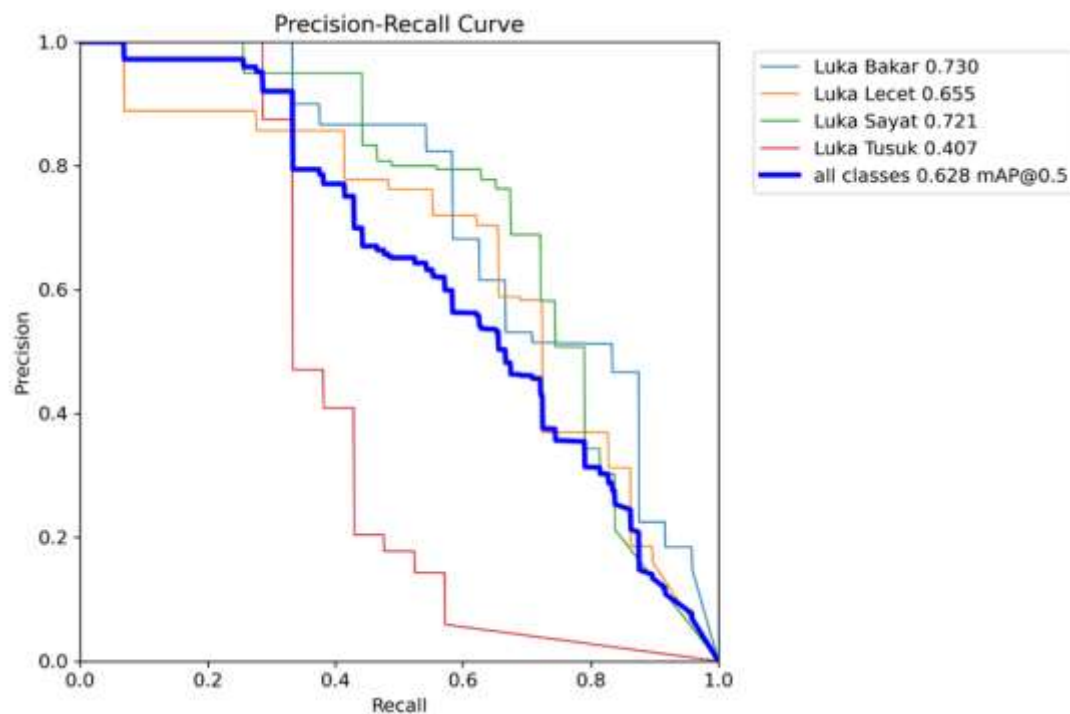
Class	Precision	Recall	mAP
all	0.619	0.625	0.614
Luka Bakar	0.775	0.667	0.653
Luka Lecet	0.507	0.708	0.684
Luka Sayat	0.676	0.698	0.684
Luka Tusuk	0.520	0.429	0.434

Setelah melakukan penyetelan *hyperparameter*, termasuk mengatur *epoch* dan *learning rate*, ditemukan kombinasi optimal pada *epoch* 200 dengan *learning rate* 0.001. Kombinasi ini menghasilkan nilai mAP tertinggi di antara kombinasi lainnya, yaitu sebesar 0.62 untuk semua kelas seperti yang terlihat pada tabel 4.13. Oleh karena itu, model yang dihasilkan dari pelatihan tersebut dipilih untuk menjadi model sistem.

Berdasarkan proses pelatihan model pada *epoch* 200 dan *learning rate* 0.001 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* yang dapat dilihat pada gambar 4.15 dan 4.16.



**Gambar 4.15 Grafik Performa Model**



**Gambar 4.16 Gambar Precision-Recall Model**

Secara umum, grafik menunjukkan bahwa model memiliki performa yang memuaskan dengan metrik seperti *box\_loss*, *class\_loss*, *dfl\_precision*, *recall*, dan mAP menunjukkan kinerja yang baik. Namun, ketika kita mengevaluasi hasil untuk setiap kelas secara spesifik, terlihat adanya variasi yang signifikan. Contohnya, nilai mAP untuk luka lecet adalah 0.655, luka bakar mencapai 0.73, luka sayat sebesar 0.721, dan luka tusuk hanya 0.407 seperti yang terlihat pada gambar 4.16. mAP atau *mean Average Precision* adalah metrik penting yang memberikan gambaran tentang seberapa baik model dapat mengenali objek dalam semua kelas.

Meskipun secara keseluruhan model menunjukkan kinerja yang memuaskan, hasil ini menunjukkan bahwa ada perbedaan dalam kemampuan model untuk mengidentifikasi dan mengklasifikasikan jenis luka secara spesifik. Luka tusuk memiliki nilai mAP paling rendah diantara kelas lain nya ini mengindikasikan bahwa model mungkin akan mengalami kesulitan mengenali luka tersebut.

Setelah melakukan pelatihan model YOLOv8 dengan menggunakan data *training*, langkah selanjutnya adalah melakukan evaluasi menggunakan *testing set* yang terdiri dari 49 gambar. Dalam tahap evaluasi ini, *confusion matrix* digunakan untuk membandingkan hasil prediksi model dengan hasil sebenarnya. Evaluasi juga melibatkan pengukuran akurasi (*accuracy*), presisi (*precision*), dan *recall* (keberhasilan) untuk mengevaluasi kinerja model. Hasil evaluasi, yang

direpresentasikan dalam bentuk confusion matrix, dari pemodelan menggunakan algoritma YOLO v8 dapat dilihat pada Tabel 4.16.

**Tabel 4.16 Pengukuran *Confusion matrix***

Nilai Prediksi	Nilai Sebenarnya			
	Luka Sayat	Luka Lecet	Luka Bakar	Luka Tusuk
Luka Sayat	15	0	0	0
Luka Lecet	0	15	1	0
Luka Bakar	0	1	9	1
Luka Tusuk	0	0	0	3
Tidak Terdeteksi	1	1	1	1

Evaluasi kinerja menggunakan *confusion matrix* memanfaatkan empat konsep untuk menggambarkan hasil dari proses klasifikasi, yaitu:

*True Positive* (TP) = Merupakan jumlah kasus positif yang benar-benar diklasifikasikan sebagai positif oleh model.

*True Negative* (TN) = Merupakan jumlah kasus negatif yang benar-benar diklasifikasikan sebagai negatif oleh model.

*False Positive* (FP) = Merupakan jumlah kasus negatif yang salah diklasifikasikan sebagai positif oleh model.

*False Negative* (FN) = Merupakan jumlah kasus positif yang salah diklasifikasikan sebagai negatif oleh model.

*Accuracy* (akurasi) adalah ukuran umum untuk mengevaluasi kinerja dari sebuah model klasifikasi. Ini mengukur seberapa sering model tersebut membuat prediksi yang benar, secara keseluruhan, dari semua kasus yang diamati. Dalam konteks *confusion matrix*, akurasi dihitung dengan rumus:

$$Accuracy = \frac{\text{Total TP}}{\text{Total Data}}$$

$$Accuracy = \frac{15+15+9+3}{49} = \frac{42}{49} = 0,85$$

*Precision* (presisi) adalah ukuran yang menggambarkan seberapa akurat model dalam mengklasifikasikan kasus positif. Dalam konteks *confusion matrix*, *precision* mengacu pada proporsi dari data yang secara tepat diklasifikasikan sebagai positif dari keseluruhan data yang diprediksi sebagai positif. Presisi dihitung dengan rumus:

$$Precision = \frac{TP}{TP+FP}$$



Precision untuk kelas “Luka Sayat”

$$TP = 15$$

$$FP = 0+0+0+0 = 0$$

Precision untuk kelas “Luka Lecet”

$$TP = 15$$

$$FP = 0+0+1+0 = 1$$

Precision untuk kelas “Luka Bakar”

$$TP = 9$$

$$FP = 0+1+0+1 = 2$$

Precision untuk kelas “Luka Tusuk”

$$TP = 3$$

$$FP = 0+0+0+0+0 = 0$$

$$Precision = \frac{15+15+9+3}{15+16+11+3} = \frac{42}{45} = 0,93$$

*Recall* adalah ukuran keberhasilan sistem dalam mendapatkan informasi yang relevan. Rumus untuk menghitung *recall* dapat ditemukan seperti pada rumus dibawah ini.

$$Recall = \frac{TP}{TP+FN}$$

Recall untuk kelas “Luka Sayat”

$$TP = 15$$

$$FN = 0+0+0+0+1 = 1$$

Recall untuk kelas “Luka Lecet”

$$TP = 15$$

$$FN = 0+0+1+0+1 = 2$$

Recall untuk kelas “Luka Bakar”

$$TP = 9$$

$$FN = 0+1+0+0+1 = 2$$

Recall untuk kelas “Luka Tusuk”

$$TP = 3$$

$$FN = 0+0+1+0+1 = 2$$

$$Recall = \frac{15+15+9+3}{16+17+11+5} = \frac{42}{49} = 0,85$$

*F1 Score* adalah nilai yang membandingkan antara *precision* dan *recall*. Cara menghitung *F1 Score* mengikuti rumus dibawah ini.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F1\ Score = 2 \times \frac{0,93 \times 0,85}{0,93 + 0,85}$$

$$F1\ Score = 2 \times \frac{0,7905}{1,78} = 0,88$$



**Gambar 4.17 Hasil Pengujian Model (1)**



**Gambar 4.18 Hasil Pengujian Model (2)**



**Gambar 4.19 Hasil Pengujian Model (3)**

Hasil pengujian model menunjukkan bahwa kemampuan model dalam mengenali luka tergantung pada kejelasan gambar. Ketika gambar luka terpapar dengan jelas, model cenderung memberikan hasil yang akurat. Namun, ketika gambar kurang berkualitas atau tidak menyoroti luka dengan baik, terdapat potensi kesalahan dalam deteksi, bahkan kemungkinan gambar tersebut tidak terdeteksi sama sekali. Dengan kata lain, kinerja model cenderung optimal saat gambar luka memiliki kualitas yang baik dan luka terlihat dengan jelas. Namun, tantangan muncul ketika gambar kurang jelas atau tidak menonjolkan luka secara memadai, yang dapat mengakibatkan kesalahan dalam proses deteksi. Oleh karena itu, untuk meningkatkan kinerja model, diperlukan pemrosesan gambar yang lebih baik dan pengembangan teknik deteksi yang lebih canggih untuk mengatasi variasi kualitas gambar.

#### **4.4 Analisa Proses Deteksi**

Berikut ini pengujian manual proses deteksi YOLO yang melalui beberapa tahapan sehingga didapatkan hasil akhir sebuah objek yang terdeteksi.

##### *4.4.1 Resize citra*

Sebelum mengolah data citra dengan arsitektur YOLO, penting untuk menyesuaikan ukuran masukan. *Meresize* data juga berguna untuk menyamakan ukuran citra yang beragam pada masukan.



**Gambar 4.20 Citra Inputan**

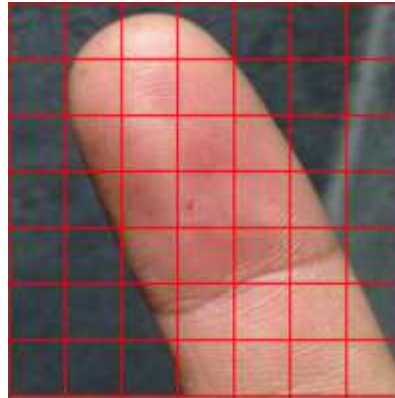
Gambar 4.20 adalah gambar asli luka sayat dengan ukuran 496x 498 pixel, dan sebelum di proses lebih lanjut maka akan di ubah menjadi ukuran 416 x 416 seperti pada gambar 4.21



**Gambar 4.21 Citra *Resize* 416x416**

Setelah didapatkan citra berukuran 416 x 416 dengan 3 *channel* proses selanjutnya adalah membagi citra kedalam kotak kotak dengan ukuran 7 x 7 seperti yang di tunjukan pada gambar 4.22 kotak kotak ini disebut sel *grid* dan setiap sel bertanggung jawab untuk memprediksi apakah terdapat objek di dalamnya atau tidak. Jika ada, akan diberi nilai 1 jika tidak diberi nilai 0. Kotak yang memiliki nilai 1 akan menghasilkan kotak pembatas (*Bounding Box*). Setiap sel terdiri dari beberapa kotak pembatas dengan 7 komponen pada setiap kotak (bx, by, bw, bh, kepercayaan, pc0, pc1).  
 bx = Merupakan koordinat horizontal (atau sumbu x)  
 by = Merupakan koordinat vertikal (atau sumbu y)  
 bw = Merupakan lebar (*width*) dari kotak pembatas  
 bh = Merupakan tinggi (*height*) dari kotak pembatas  
 pc0 dan pc1= adalah probabilitas atau skor yang diberikan oleh model untuk menunjukkan seberapa yakin model tersebut bahwa objek yang dideteksi adalah dari kelas tertentu.

Tabel 4.17 menampilkan ilustrasi dari setiap sel grid pada vektor output.



**Gambar 4.22 Citra Diberi Gridcell 7x7**

**Tabel 4.17 Ilustrasi Komponen Pada Bounding Box**

	bx	by	bw	bh	confidence	Pc0	Pc1
Bbox 1	...	...	...	...	...	....	...
Bbox2	...	...	...	...	...	...	...
Bbox3	...	...	...	...	...	...	...
Bbox4	...	...	...	...	...	...	...
Bbox...	...	...	...	...	...	...	...

#### 4.4.2 Arsitektur YOLOv8

Berikut ini arsitektur YOLOv8 yang digunakan dalam penelitian ini yang di ambil dari *summary* saat pelatihan model dengan  $F = \text{Ukuran Kernel Filter}$ ,  $P = \text{Padding}$ ,  $S = \text{Stride}$

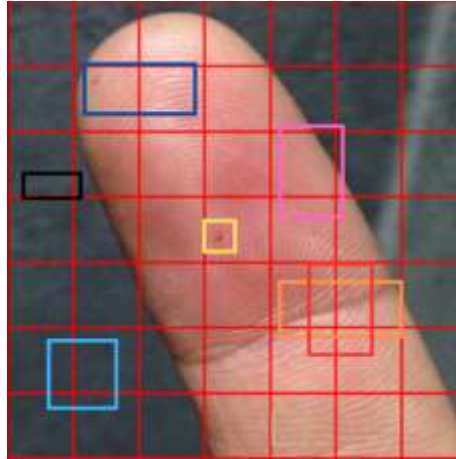
**Tabel 4.18 Arsitektur YOLOv8 Per Layer**

#	Layer Type	Input Shape	Output Shape	F	p	S
1	Conv + BatchNorm + LeakyReLU	(416, 416, 3)	(208, 208, 16)	3x3	1	2
2	Max Pooling	(208, 208, 16)	(104, 104, 16)	2x2	0	1
3	Conv + BatchNorm + LeakyReLU	(104, 104, 16)	(104, 104, 32)	3x3	1	2
4	Max Pooling	(104, 104, 32)	(52, 52, 32)	2x2	1	2
5	Conv + BatchNorm + LeakyReLU	(52, 52, 32)	(52, 52, 64)	3x3	0	1

6	Max Pooling	(52, 52, 64)	(26, 26, 64)	2x2	0	1
7	Conv + BatchNorm + LeakyReLU	(26, 26, 64)	(26, 26, 128)	3x3	1	2
8	Max Pooling	(26, 26, 128)	(13, 13, 128)	2x2	0	1
9	Conv + BatchNorm + LeakyReLU	(13, 13, 128)	(13, 13, 256)	3x3	1	2
10	SPPF	(13, 13, 256)	(13, 13, 256)	-	-	-
11	Upsample	(13, 13, 256)	(26, 26, 256)	-	-	-
12	Concatenation	(26, 26, 512)	(26, 26, 512)	-	-	-
13	Conv + BatchNorm + LeakyReLU	(26, 26, 512)	(26, 26, 128)	3x3	1	1
14	Upsample	(26, 26, 128)	(52, 52, 128)	-	-	-
15	Concatenation	(52, 52, 256)	(52, 52, 256)	-	-	-
16	Conv + BatchNorm + LeakyReLU	(52, 52, 256)	(52, 52, 64)	3x3	1	1
17	Conv + BatchNorm + LeakyReLU	(52, 52, 64)	(26, 26, 64)	3x3	1	2
18	Concatenation	(26, 26, 128)	(26, 26, 128)	-	-	-
19	Conv + BatchNorm + LeakyReLU	(26, 26, 128)	(13, 13, 128)	3x3	1	2
20	Concatenation	(13, 13, 256)	(13, 13, 256)	-	-	-
21	Conv + BatchNorm + LeakyReLU	(13, 13, 256)	(13, 13, 512)	3x3	1	2
22	Concatenation	(13, 13, 1024)	(13, 13, 1024)	-	-	-
23	Detect	(13, 13, 1024)	(13, 13, 256)	-	-	-

Berikut ini adalah penjelasan singkat mengenai arsitektur YOLO diatas. Pertama, gambar input berukuran 416x416 pixel dengan 3 saluran warna dimasukkan ke dalam model. Kemudian, gambar tersebut melewati lapisan konvolusi pertama yang menggunakan filter 3x3 untuk mengekstraksi fitur-fitur penting. Setelah itu, hasil konvolusi tersebut dinormalisasi menggunakan *batch normalization* dan diaktivasi menggunakan fungsi *Leaky ReLU* untuk memperkenalkan non-linearitas. Selanjutnya, dilakukan operasi *max pooling* dengan ukuran kernel 2x2 untuk mengurangi dimensi gambar menjadi separuh, sehingga fitur-fitur yang relevan tetap dipertahankan. Proses ini diulangi beberapa kali dengan lapisan konvolusi, normalisasi batch, dan *max pooling*, yang semuanya bertujuan untuk mengekstraksi fitur-fitur dari gambar. Pada lapisan tertentu, seperti pada layer 9, terdapat lapisan khusus yang disebut SPPF yang

melakukan penggabungan dari beberapa tingkat skala spasial untuk meningkatkan kemampuan deteksi. Terakhir, gambar yang dihasilkan dari lapisan sebelumnya digunakan dalam lapisan deteksi untuk mengidentifikasi objek dan menentukan kotak pembatas untuk setiap objek yang terdeteksi.



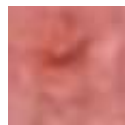
**Gambar 4.23 Ilustrasi Pencarian *Bounding Box***

Proses pencarian *bounding box* terlihat dalam gambar 4.23, menunjukkan bahwa setiap sel *grid* bertanggung jawab untuk mencari menggunakan *anchor box* dengan berbagai ukuran. Diketahui titik yang dilihat adanya luka terdapat pada kordinat Titik  $(x,y) = (228,260)$

Lebar  $(w) = 23$

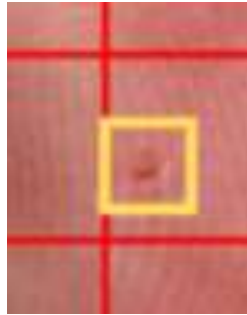
Tinggi  $(h) = 22$

Potongan gambar yang menunjukkan koordinat  $(185,193)$  dapat dilihat pada gambar 4.24



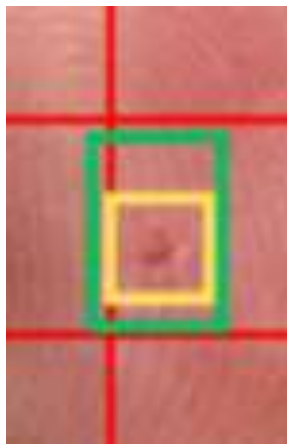
**Gambar 4.24 Potongan Gambar Koordinat  $(185,193)$**

Untuk potongan gambar *bounding box* dengan koordinat  $(228,260)$  dapat dilihat pada gambar 4.25



**Gambar 4.25 Objek yang Didapatkan Oleh *Bounding Box***

*Bounding box* yang digunakan adalah yang berwarna kuning pada ilustrasi dalam Gambar 4.26



**Gambar 4.26 Prediksi IOU**

Setelahnya, langkah-langkah perhitungan *Intersection over Union* (IOU) dapat dihitung dengan cara sebagai berikut:

$$\text{area} (BB_{\text{prediksi}} \cap BB_{\text{groundTruth}}) = (30 \times 46) = 1380$$

$$\text{area} (BB_{\text{prediksi}} \cup BB_{\text{groundTruth}}) = (23 \times 22) + (30 \times 46) = 506 + 1380 = 1886$$

$$IOU = \frac{\text{area} (BB_{\text{prediksi}} \cap BB_{\text{groundTruth}})}{\text{area} (BB_{\text{prediksi}} \cup BB_{\text{groundTruth}})}$$

$$IOU = \frac{1380}{1886} = 0.73$$

Data bounding box yang diperoleh akan digunakan hingga mencapai *fully connected layer* dan nilai 0.73 akan digunakan sebagai skor kepercayaan (*confidence score*) pada *bounding box*. Menurut Joseph Redmon dalam papernya, nilai akhir dari prediksi (kelas dari skor kepercayaan) akan didapat dengan mengalikan skor kepercayaan *bounding box* dengan probabilitas kondisional kelas. Probabilitas kondisional untuk setiap kelas  $\text{Pr}(\text{object})$  menunjukkan kemungkinan bahwa objek pada *bounding box* termasuk dalam kelas tertentu, contohnya kelas "Luka Sayat". Nilai



probabilitas ini berkisar antara 0 hingga 1, di mana 1 menunjukkan bahwa objek pada *bounding box* termasuk dalam kelas tersebut, dan 0 menunjukkan sebaliknya.

Sebagai contoh, jika probabilitas kondisional kelas "Luka Sayat" ( $\Pr$  luka sayat) adalah 1, maka skor kepercayaan kelas akan dihitung sebagai berikut:

$$\Pr(\text{Class}|\text{Object}).\text{box confidence score} = \Pr(\text{class } i).IoU_{pred}^{truth}$$

$$\Pr(\text{Class}_{Luka Sayat}).\text{box confidence score} = 1 \cdot 0.7317 = 0.731$$

#### 4.4.3 Proses CNN

Tahap pertama adalah proses konvolusi, setelah proses *resize* dan didapatkan IOU *prediction* sehingga didapatkan data kordinat maka akan masuk ke tahap konvolusi, Untuk cara kerja konvolusi yaitu dengan menjumlahkan perkalian antara matriks inputan dengan matriks filter, pada tabel 4.22 adalah contoh dari matriks filter yang digunakan dimana ukuran yg digunakan adalah 3x 3 dengan *stride* 2 dalam proses konvolusinya dengan koordinat citra sebagai berikut:

Titik (x,y) = (228,260)

Lebar (w) = 23

Tinggi (h) = 22

**Tabel 4.19 Matriks Citra Pada Koordinat (228, 260) Red**

x,y	0	1	2	...	20	21	22
0	205	208	210	...	202	207	217
1	207	207	208	...	199	205	216
2	202	209	211	...	195	200	208
...	...	...	...	...	...	...	...
19	204	208	208	...	212	207	204
20	208	210	207	...	206	201	199
21	205	208	208	...	216	213	203

**Tabel 4.20 Matriks Citra Pada Koordinat (228, 260) Green**

x,y	0	1	2	...	20	21	22
0	125	126	126	...	120	125	135
1	126	125	124	...	117	123	134
2	121	126	126	...	114	119	126
...	...	...	...	...	...	...	...
19	130	134	134	...	131	126	123
20	135	137	134	...	125	120	117
21	133	136	136	...	137	135	125

**Tabel 4.21 Matriks Citra Pada Koordinat (228, 260) Blue**

<b>x,y</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>...</b>	<b>20</b>	<b>21</b>	<b>22</b>
<b>0</b>	122	124	125	...	118	123	134
<b>1</b>	123	122	122	...	115	121	133
<b>2</b>	117	122	123	...	111	116	125
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>19</b>	132	134	132	...	128	123	121
<b>20</b>	137	137	132	...	122	117	116
<b>21</b>	132	133	133	...	132	130	120

**Tabel 4.22 Filter Konvolusi 3x3**

<b>x,y</b>	<b>1</b>	<b>2</b>	<b>3</b>
<b>1</b>	1	0	-1
<b>2</b>	1	0	-1
<b>3</b>	1	0	-1

Setelah dilakukan konvolusi dengan filter pada tabel 4.22 maka didapat hasil dengan ukuran 11 x 10 seperti pada tabel 4.23, 4.24 dan 4.25

**Tabel 4.23 Matriks Hasil Konvolusi (228, 260) Red**

<b>x,y</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>...</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>0</b>	-15	14	12	...	8	24	-45
<b>1</b>	-11	18	-7	...	-12	20	-13
<b>2</b>	-6	10	-4	...	-40	-6	25
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>7</b>	-8	3	27	....	-23	47	3
<b>8</b>	-7	18	35	...	-17	5	8
<b>9</b>	-8	15	19	...	-16	-1	20

**Tabel 4.24 Matriks Hasil Konvolusi (228, 260) Green**

<b>x,y</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>...</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>0</b>	-4	18	22	...	6	10	-44
<b>1</b>	0	21	9	...	-15	8	-11
<b>2</b>	6	12	16	...	-45	-13	25
<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>	<b>...</b>
<b>7</b>	-1	6	38	....	-25	42	5
<b>8</b>	-5	17	48	...	-17	1	9
<b>9</b>	-9	14	33	...	-17	-6	22

**Tabel 4.25 Matriks hasil konvolusi (228, 260) Blue**

x,y	0	1	2	...	8	9	10
0	-8	19	31	...	4	5	-48
1	-3	26	16	...	-17	7	-16
2	3	17	24	...	-48	-13	22
...	...	...	...	...	...	...	...
7	7	19	45	....	-27	35	0
8	6	32	49	...	-19	-8	5
9	3	29	30	...	-15	-17	17

Tahap selanjutnya akan masuk ke lapisan ReLu (*Rectified Linear Unit*) yang merupakan fungsi aktivasi yang digunakan untuk mengubah ukuran pixel dari lapisan konvolusi dimana setiap nilai negatif akan diubah menjadi nol (0), sementara nilai yang lebih besar dari 255 akan diubah menjadi 255 dan hasilnya dapat dilihat pada tabel 4.26, 4.27 dan 4.28.

**Tabel 4.26 Hasil Dari Proses RELU pada matriks (228, 260) Red**

x,y	0	1	2	...	8	9	10
0	0	14	12	...	8	24	0
1	0	18	0	...	0	20	0
2	0	10	0	...	0	0	25
...	...	...	...	...	...	...	...
7	0	3	27	....	0	47	3
8	0	18	35	...	0	5	8
9	0	15	19	...	0	0	20

**Tabel 4.27 Hasil Dari Proses RELU pada matriks (228, 260) Green**

x,y	0	1	2	...	8	9	10
0	0	18	22	...	6	10	0
1	0	21	9	...	0	8	0
2	6	12	16	...	0	0	25
...	...	...	...	...	...	...	...
7	0	6	38	....	0	42	5
8	0	17	48	...	0	1	9
9	0	14	33	...	0	0	22

**Tabel 4.28 Hasil Dari Proses RELU pada matriks (228, 260) Blue**

x,y	0	1	2	...	8	9	10
0	0	19	31	...	4	5	0
1	0	26	16	...	0	7	0
2	3	17	24	...	0	0	22
...	...	...	...	...	...	...	...
7	7	19	45	....	0	35	0
8	6	32	49	...	0	0	5
9	3	29	30	...	0	0	17

Tahapan selanjutnya adalah operasi *max pooling* yaitu penyeleksian matriks ukuran yg paling maksimal diantara nilai tetangganya, dan pada proses ini digunakan pooling berukuran 2x2 dengan *stride* 1, maka di dapatkan matriks baru seperti pada tabel 4.29, 4.30 dan 4.31

**Tabel 4.29 Hasil *Max Pooling* citra (228, 260) *Red***

<b>x,y</b>	<b>0</b>	<b>1</b>	<b>...</b>	<b>8</b>	<b>9</b>
<b>0</b>	18	18	...	24	24
<b>1</b>	18	18	...	20	25
<b>...</b>	...	...	...	...	...
<b>7</b>	18	35	...	47	47
<b>8</b>	18	35	...	5	20

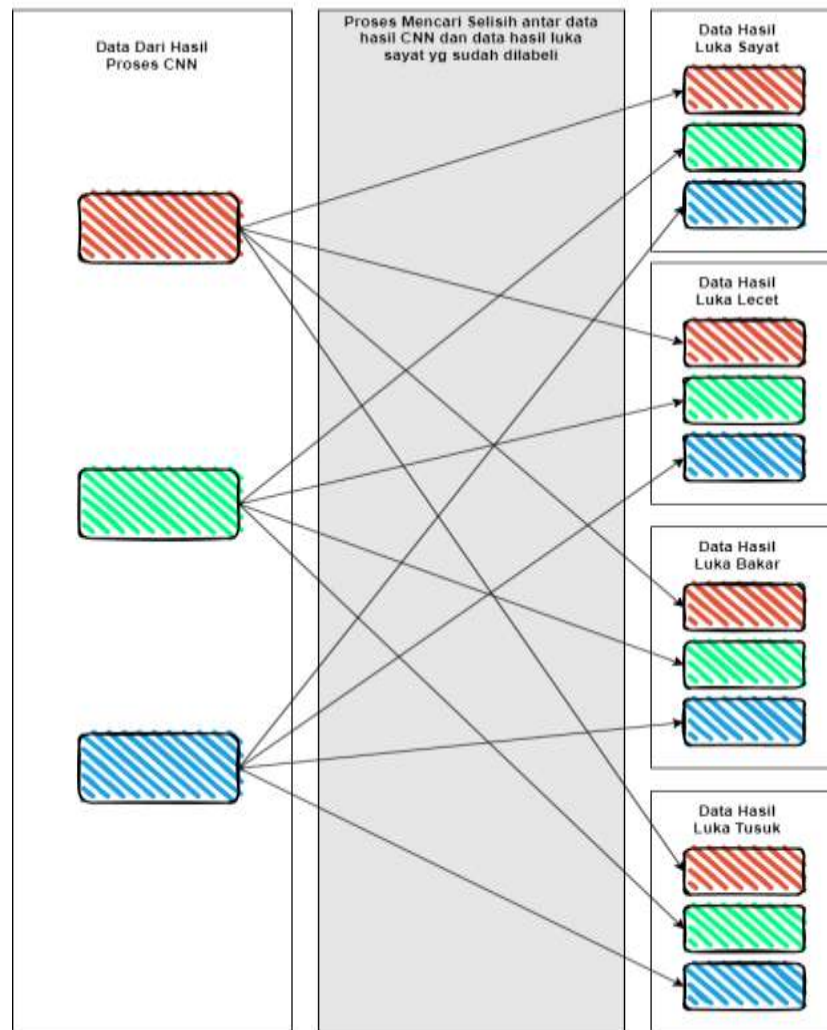
**Tabel 4.30 Hasil *Max Pooling* citra (228, 260) *Green***

<b>x,y</b>	<b>0</b>	<b>1</b>	<b>...</b>	<b>8</b>	<b>9</b>
<b>0</b>	21	22	...	10	10
<b>1</b>	21	21	...	8	25
<b>...</b>	...	...	...	...	...
<b>7</b>	17	48	...	42	42
<b>8</b>	17	48	...	1	22

**Tabel 4.31 Hasil *Max Pooling* citra (228, 260) *Blue***

<b>x,y</b>	<b>0</b>	<b>1</b>	<b>...</b>	<b>8</b>	<b>9</b>
<b>0</b>	26	31	...	7	7
<b>1</b>	26	26	...	7	22
<b>...</b>	...	...	...	...	...
<b>7</b>	32	49	...	35	35
<b>8</b>	32	49	...	0	17

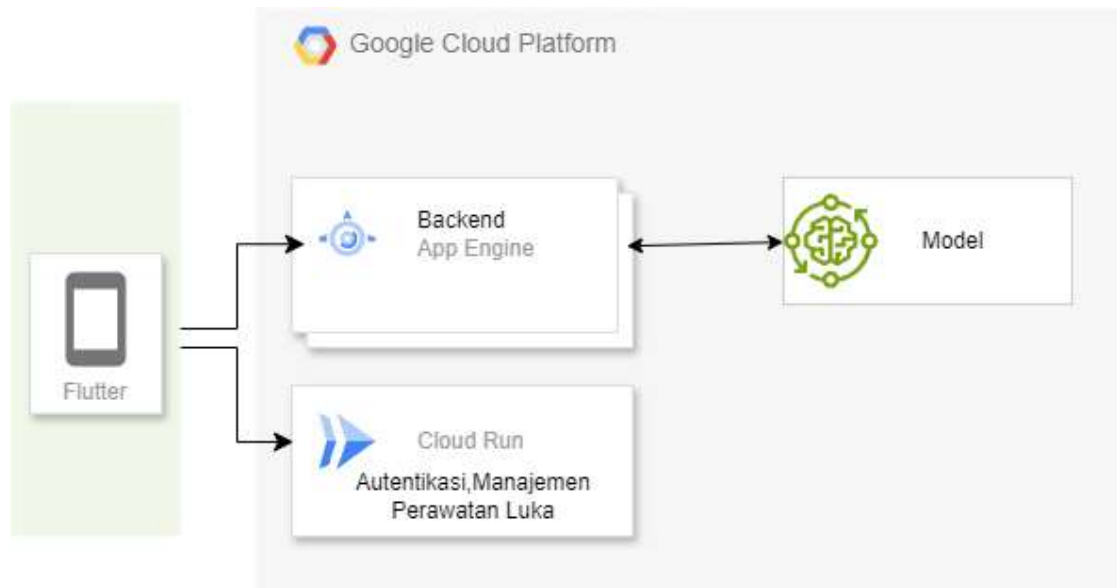
Setelah selesai melalui seluruh proses *Convolutional Neural Network* (CNN), langkah selanjutnya adalah menggunakan lapisan *fully connected*. Lapisan ini bertanggung jawab untuk melakukan klasifikasi objek. Pada tahap klasifikasi, data citra yang telah diproses melalui langkah-langkah sebelumnya dalam CNN akan dicocokkan dengan data pelatihan yang telah dilabeli. Proses pencocokan dilakukan dengan mengurangkan data satu per satu, dan hasil selisihnya akan digunakan untuk menentukan kelas objek. Kelas objek yang dipilih adalah yang memiliki selisih terkecil, yang menunjukkan tingkat kepercayaan tertinggi.



**Gambar 4.27 Fully Connected Layer Untuk Mencari Probabilitas**

Gambar 4.27 menggambarkan langkah-langkah untuk menemukan perbedaan antara hasil dari CNN dengan keempat kelas data. Dengan menemukan perbedaan terkecil, dapat disimpulkan bahwa kelas yang paling mungkin muncul dalam citra inputan adalah kelas yang sesuai.

#### 4.5 Implementasi Model ke Aplikasi Berbasis *Android*



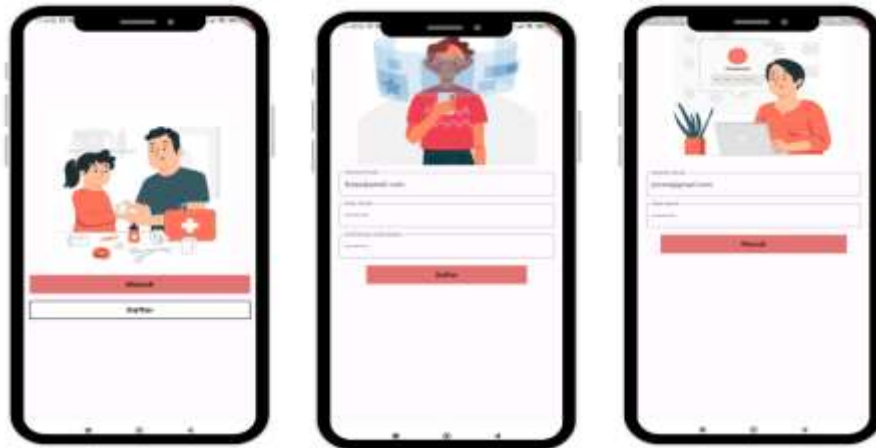
**Gambar 4.28**Arsitektur Pada *Cloud*

Dalam penelitian ini, aplikasi dikembangkan menggunakan metode deteksi objek YOLOv8, yang memanfaatkan arsitektur *Convolutional Neural Networks* (CNN). Aplikasi tersebut dirancang dengan dua bagian utama, yaitu bagian *frontend* dan *backend*. Bagian *frontend* dikembangkan menggunakan *framework* *Flutter*, sementara bagian *backend* menggunakan Node.js. Selanjutnya, aplikasi ini akan diunggah dan di-host menggunakan layanan yang disediakan oleh *Google Cloud*.

Dengan pendekatan ini, aplikasi memiliki kemampuan untuk mendeteksi objek menggunakan metode YOLOv8, yang telah terbukti efektif dalam tugas deteksi objek. Penggunaan *Flutter* untuk bagian *frontend* memberikan pengalaman pengguna yang mulus dan responsif, sementara Node.js memberikan *backend* yang andal dan efisien untuk menangani logika dan permintaan server.

Selain itu, pilihan untuk menggunakan layanan *hosting* pada *Google Cloud* menawarkan skalabilitas yang dapat diandalkan serta keamanan yang kuat untuk aplikasi tersebut. Dengan demikian, pendekatan ini menyatukan teknologi-teknologi terkini untuk menghasilkan aplikasi yang dapat memberikan pengalaman yang optimal bagi pengguna, serta memenuhi persyaratan keamanan dan skalabilitas yang diperlukan dalam pengembangan aplikasi modern.

#### 4.5.1 Halaman Login Aplikasi



**Gambar 4.29 Halaman *Login* Aplikasi CekLukaID**

Gambar 4.29 merupakan halaman login aplikasi yang terbagi menjadi dua bagian yaitu "Masuk" dan "Daftar". Bagian "Masuk" memungkinkan pengguna untuk memasukkan alamat email dan kata sandi mereka untuk mengakses *platform*. Sedangkan, bagian "Daftar" memberikan kesempatan bagi pengguna baru untuk membuat akun dengan memasukkan alamat email dan kata sandi baru.

#### 4.5.2 Halaman Beranda Aplikasi

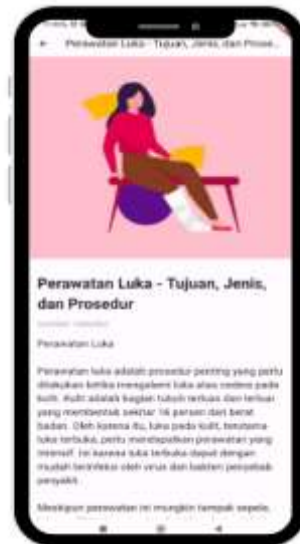


**Gambar 4.30 Beranda Aplikasi CekLukaID**

Gambar 4.30 merupakan halaman beranda aplikasi, menampilkan artikel-artikel terkait topik kesehatan, termasuk informasi tentang batasan luka ringan dan artikel kesehatan umum. Artikel-artikel ini bertujuan untuk memberikan informasi yang berguna kepada pengguna. Di bagian bawah halaman, terdapat navigation bar yang menyediakan akses

mudah ke beranda, fitur pemindaian (*scan*), dan *chat bot*. fitur ini memungkinkan pengguna untuk kembali ke halaman utama, mengakses fitur pemindaian untuk deteksi luka ringan, dan berinteraksi dengan chat bot untuk mendapatkan bantuan dalam waktu nyata.

#### 4.5.3 Halaman Artikel Kesehatan



**Gambar 4.31 Halaman Artikel Kesehatan**

Gambar 4.31 merupakan halaman artikel kesehatan, menampilkan artikel-artikel yang dilengkapi dengan gambar yang relevan, judul yang jelas, serta sumber artikel. Pengguna dapat menekan gambar artikel Kesehatan yang ingin dibaca pada dashboard dan akan menampilkan artikelnya secara lengkap



#### 4.5.4 Halaman Artikel Batasan Luka



**Gambar 4.32 Halaman Batasa Luka**

Gambar 4.31 merupakan halaman artikel tentang batasan luka menampilkan ilustrasi relevan, judul singkat yang mencakup jenis luka, dan informasi singkat tentang batasannya. Pengguna dapat menekan gambar artikel Batasan luka agar menampilkan artikel secara lengkap

#### 4.5.5 Halaman Scan

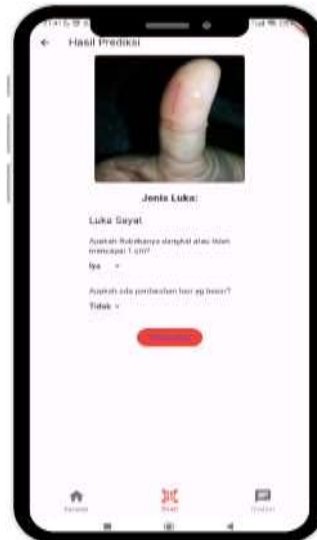


**Gambar 4.33 Halaman Scan Luka**

Gambar 4.33 merupakan halaman *scan* yang memiliki fungsi utama yaitu pemindaian foto luka ringan. Terdapat dua opsi yang jelas di tengah halaman, tombol kamera dan tombol galeri. Tombol kamera memungkinkan pengguna untuk mengambil foto luka

secara langsung, sedangkan tombol galeri memungkinkan pengguna memilih foto dari galeri perangkat mereka.

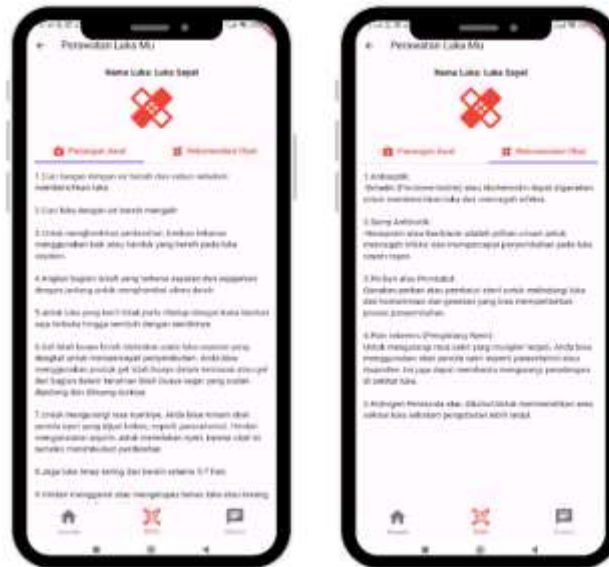
#### 4.5.6 *Halaman Hasil Deteksi Luka*



**Gambar 4.34 Halaman Hasil Deteksi**

Gambar 4.34 merupakan halaman hasil deteksi luka, saat pengguna masuk ke halaman deteksi luka, pengguna akan disambut dengan gambar yang telah diunggah sebelumnya, lengkap dengan hasil deteksi luka ringan. Setelah melihat hasilnya, secara otomatis, muncul dua pertanyaan yang terkait langsung dengan kondisi luka. Pengguna diminta untuk merespons dengan mengklik opsi "Iya" atau "Tidak" sesuai dengan apa yang Anda alami. Setelah memberikan jawaban, Pengguna dapat melanjutkan dengan menekan tombol "Selanjutnya", yang akan membawa pengguna ke halaman manajemen perawatan luka. Di sana, pengguna akan menemukan rekomendasi perawatan yang spesifik berdasarkan hasil deteksi dan respons pengguna sebelumnya, membantu pengguna dalam merawat luka dengan lebih efektif.

#### 4.5.7 Halaman Penanganan Awal dan Rekomendasi Obat



**Gambar 4.35 Halaman Manajemen Perawatan Luka**

Gambar 4.35 merupakan halaman manajemen perawatan luka, setelah melewati halaman deteksi luka, pengguna akan dipandu melalui langkah-langkah penanganan awal yang sesuai dengan jenis luka yang terdeteksi dan respons dari jawaban yang diberikan pada halaman sebelumnya. Di sini sistem akan memberikan panduan langkah demi langkah tentang cara merawat luka pengguna secara efektif. Selain itu, pengguna akan diberikan rekomendasi obat-obatan yang tersedia di apotek dan sesuai dengan jenis luka yang terdeteksi. Informasi ini dirancang untuk membantu pengguna dalam merawat luka dengan tepat dan efisien, sehingga mereka dapat mengambil langkah-langkah yang diperlukan untuk pemulihan yang optimal.

#### 4.5.8 Halaman Chatbot



**Gambar 4.36 Halaman Chatbot Aplikasi**






Gambar 4.36 merupakan halaman chatbot aplikasi, pada halaman *chatbot* pengguna dapat bertanya seputar Kesehatan dan hal yg berhubungan dengan luka.






#### 4.6 Pengujian Sistem



Berikut adalah langkah-langkah yang perlu dilakukan setelah berhasil mengimplementasikan tahapan sistem. Pengujian sistem bertujuan untuk memverifikasi bahwa sistem yang telah dibangun dapat berfungsi dengan tepat dalam mengenali gambar luka ringan. Pada tahap pengujian, akan digunakan 20 gambar yang diambil dari sumber internet yang menampilkan contoh luka ringan yang umum terjadi sehari-hari. Tujuan dari penggunaan gambar-gambar ini adalah untuk memastikan bahwa sistem dapat mengidentifikasi dengan akurat berbagai jenis luka ringan yang mungkin ditemui dalam situasi sehari-hari dan memberikan penanganan awal dan rekomendasi obat yg sesuai, serta memastikan kehandalan sistem dalam berbagai variasi gambar.

**Tabel 4.32 Pengujian Sistem**

No	Gambar Luka Ringan		Nama Luka Ringan
1			Luka Sayat
2			Luka Bakar
3			Luka Bakar
4			Luka Lecet
5			Luka Sayat

No	Gambar Luka Ringan	Nama Luka Ringan
6		Luka Lecet
7		Luka Tusuk
8		Luka Lecet
9		Luka Lecet
10		Luka Bakar

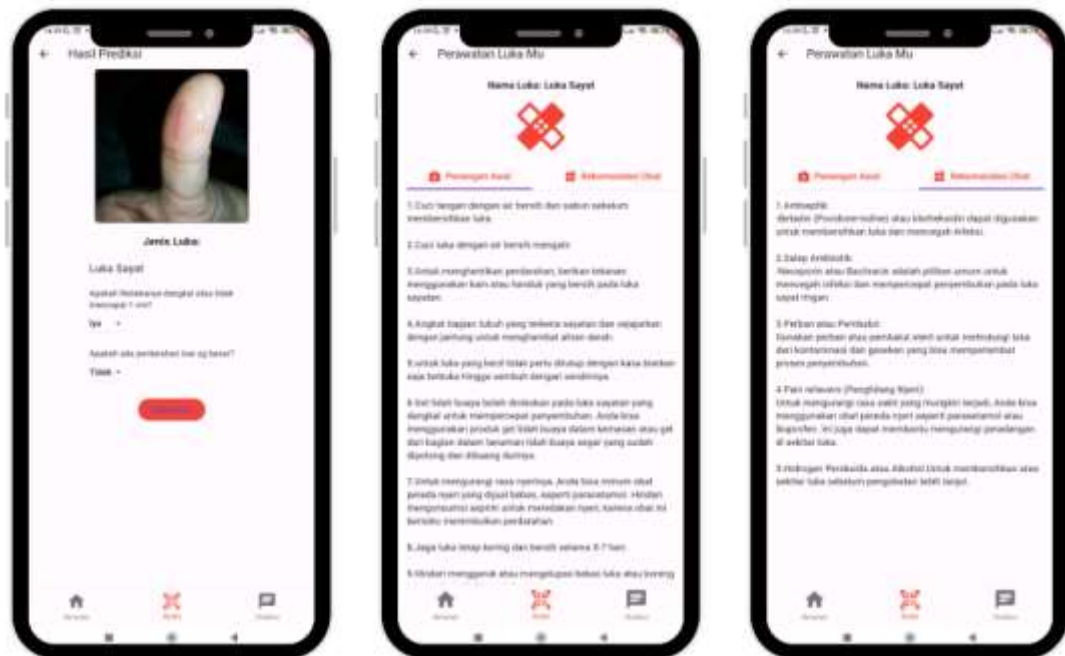
No	Gambar Luka Ringan	Nama Luka Ringan
11		Luka Bakar
12		Luka Sayat
13		Luka Sayat
14		Luka Tusuk
15		Luka Tusuk

No	Gambar Luka Ringan	Nama Luka Ringan
16		Luka Lecet
17		Luka Lecet
18		Luka Sayat
19		Luka Sayat
20		Luka Sayat

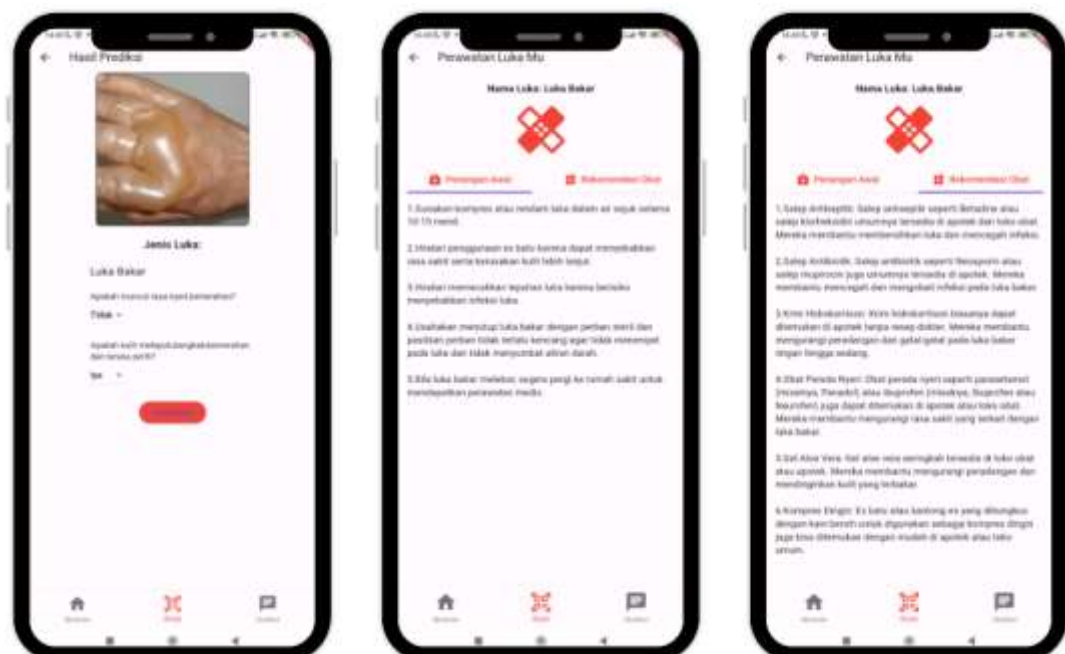


## 4.7 Hasil Pengujian Sistem

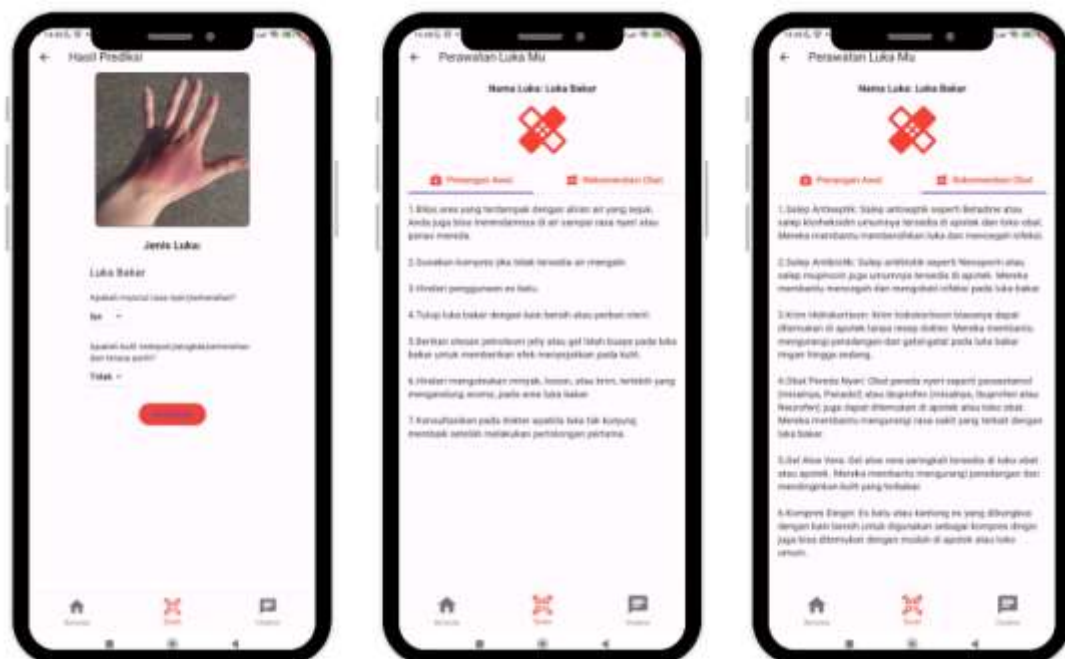
Berikut adalah hasil yang didapat dengan mencoba semua gambar luka ringan diatas:



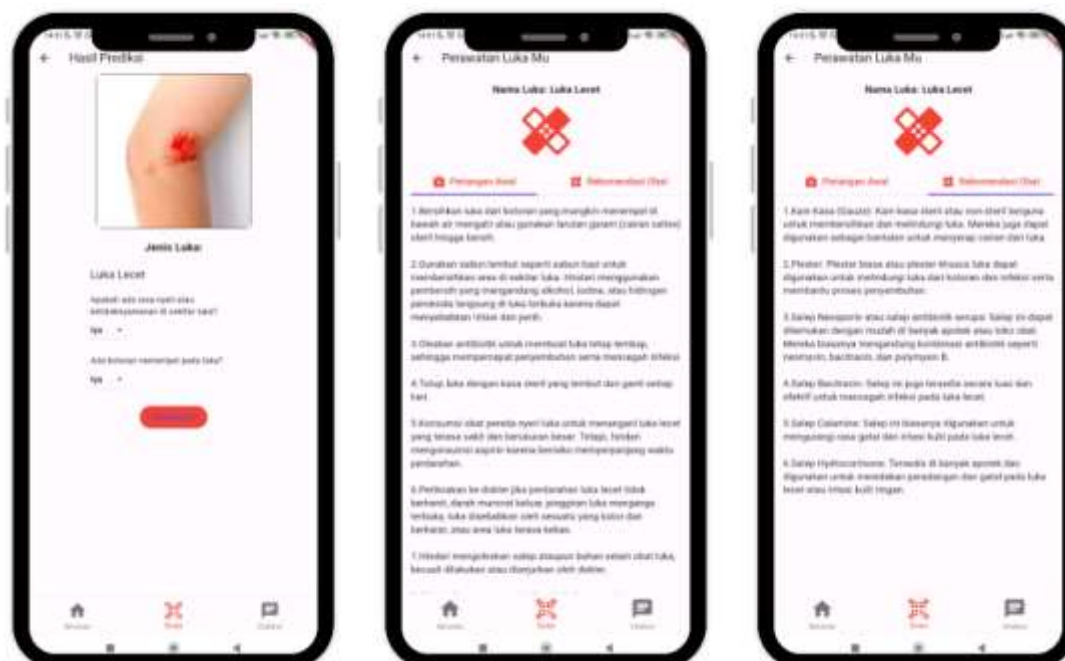
Gambar 4.37 Hasil Uji Sistem Data Uji No 1



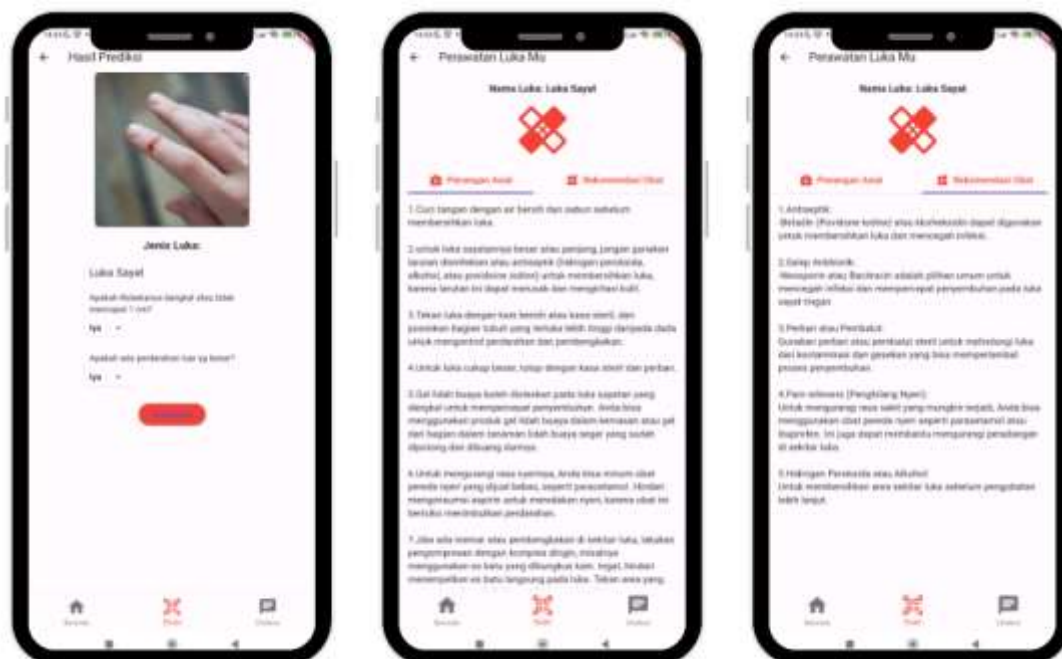
Gambar 4.38 Hasil Uji Sistem Data Uji No 2



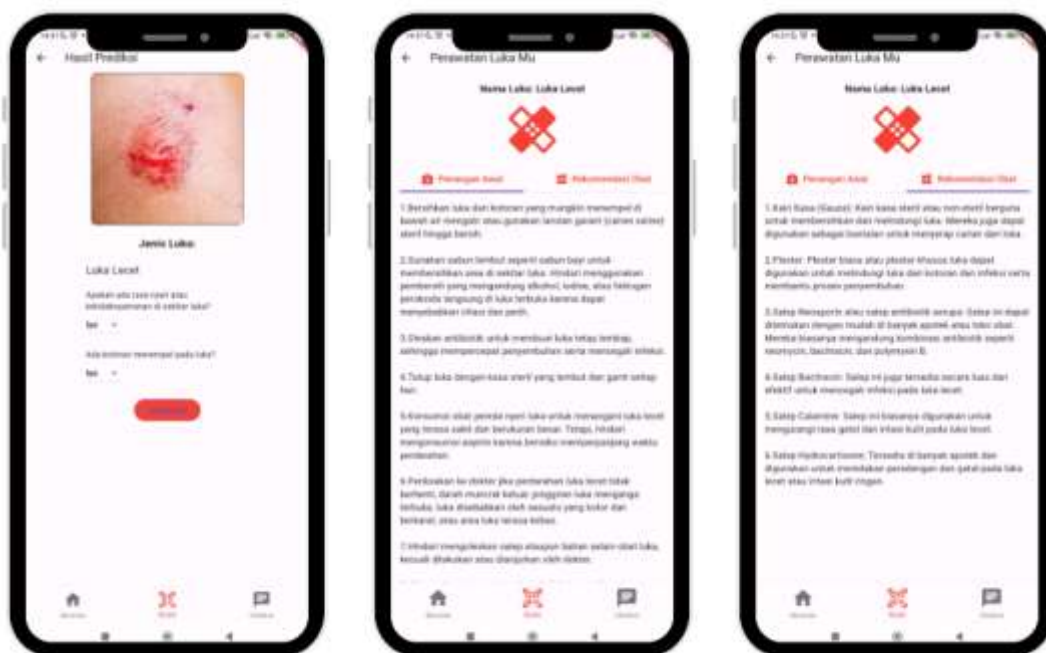
Gambar 4.39 Hasil Uji Sistem Data Uji No 3



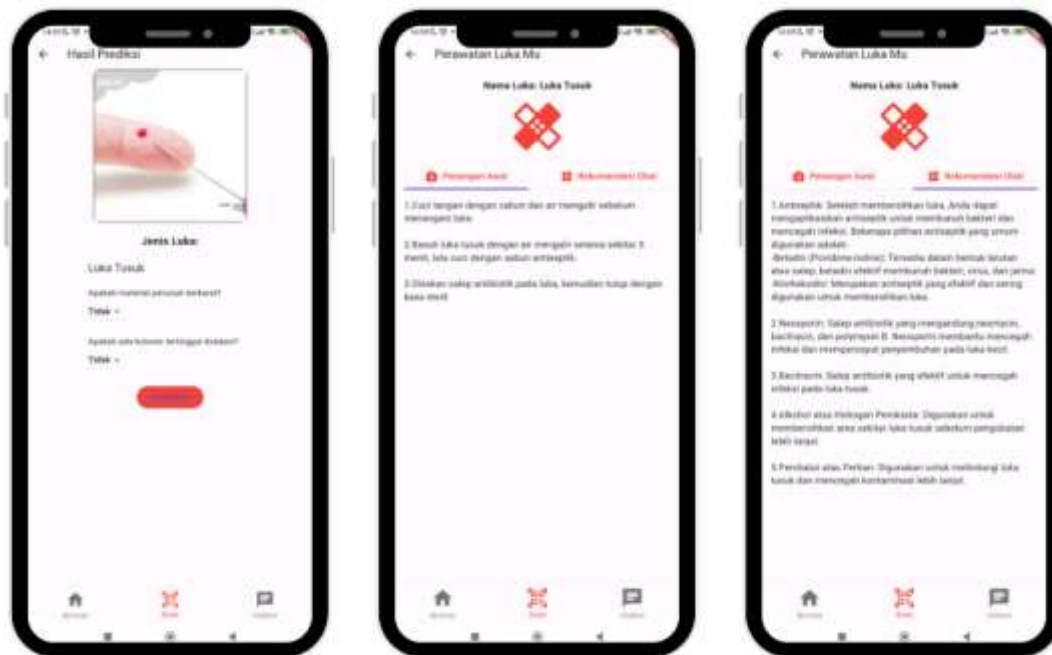
Gambar 4.40 Hasil Uji Sistem Data Uji No 4



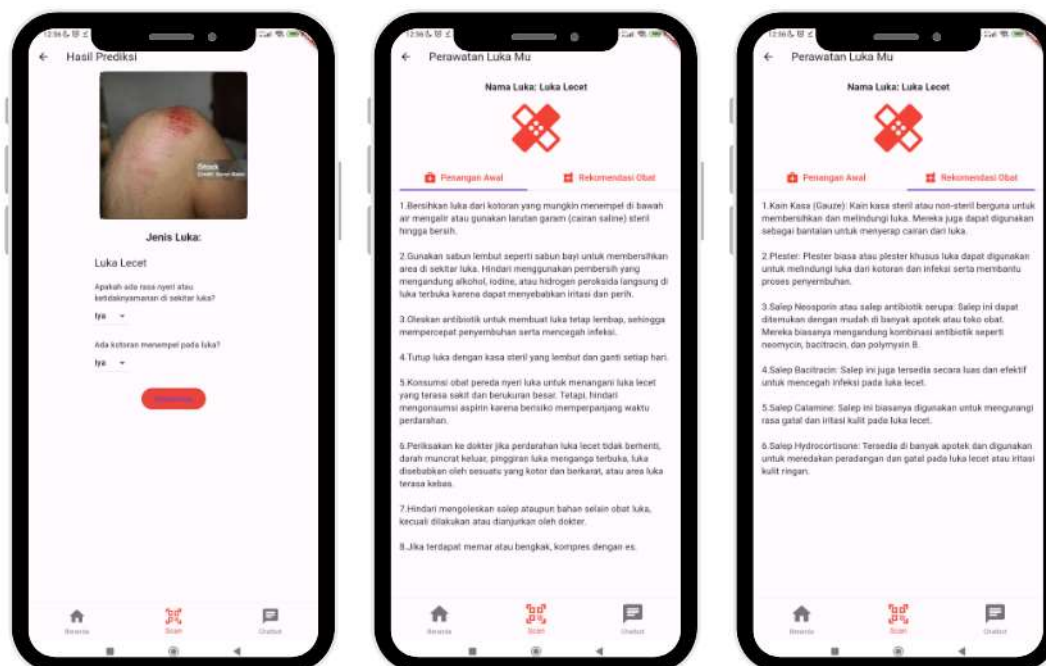
Gambar 4.41 Hasil Uji Sistem Data Uji No 5



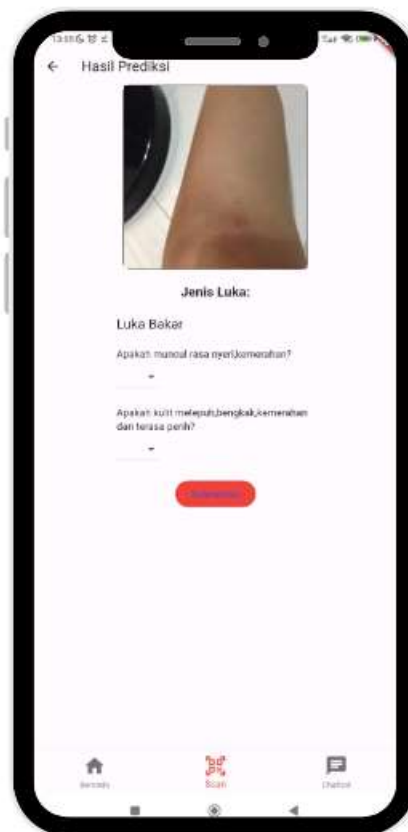
Gambar 4.42 Hasil Uji Sistem Data Uji No 6



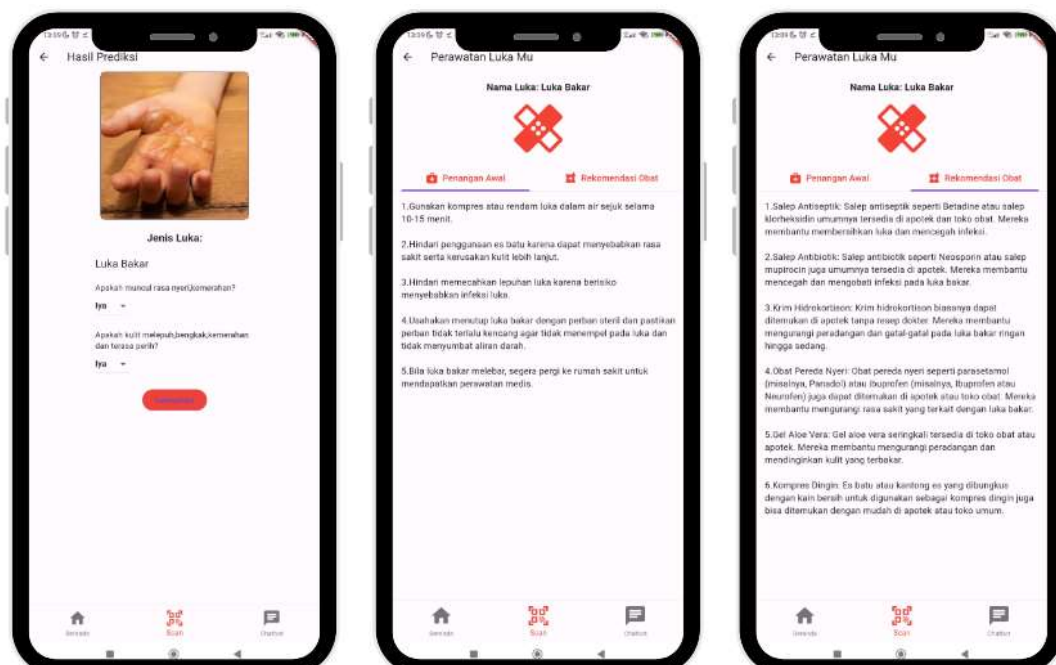
Gambar 4.43 Hasil Uji Sistem Data Uji No 7



Gambar 4.44 Hasil Uji Data Uji No 8

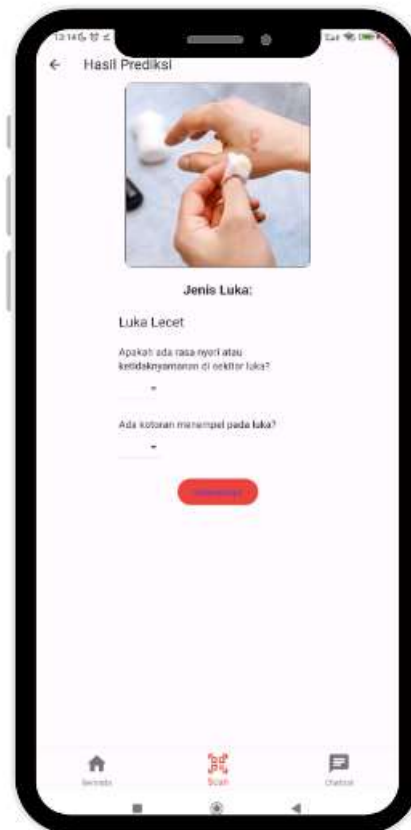


Gambar 4.45 Hasil Uji Data Uji No 9

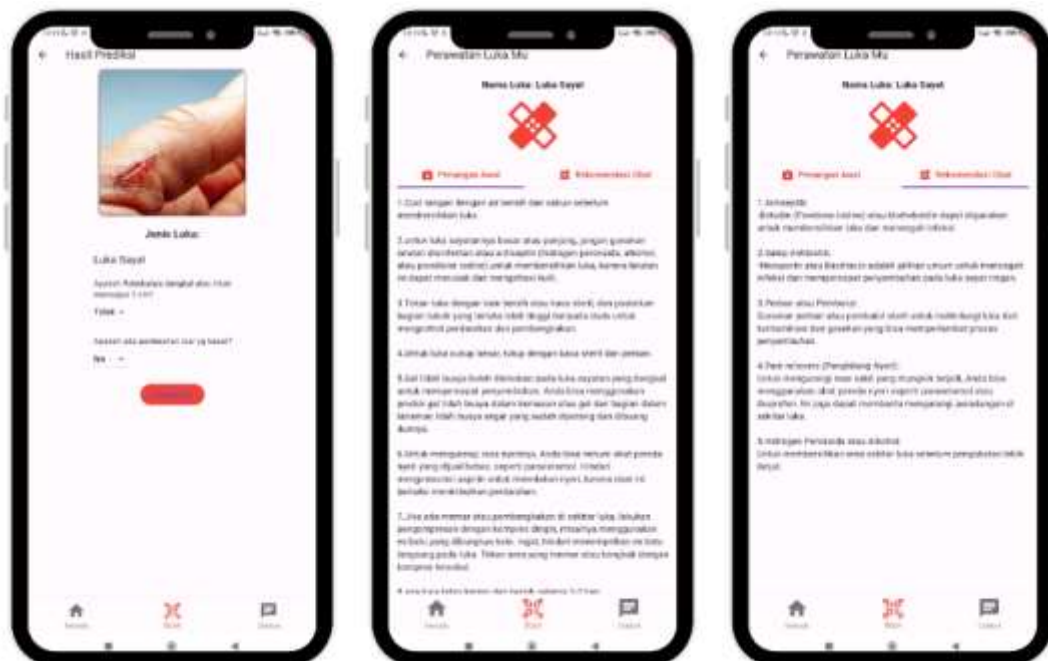


Gambar 4.46 Hasil Uji Data Uji No 10

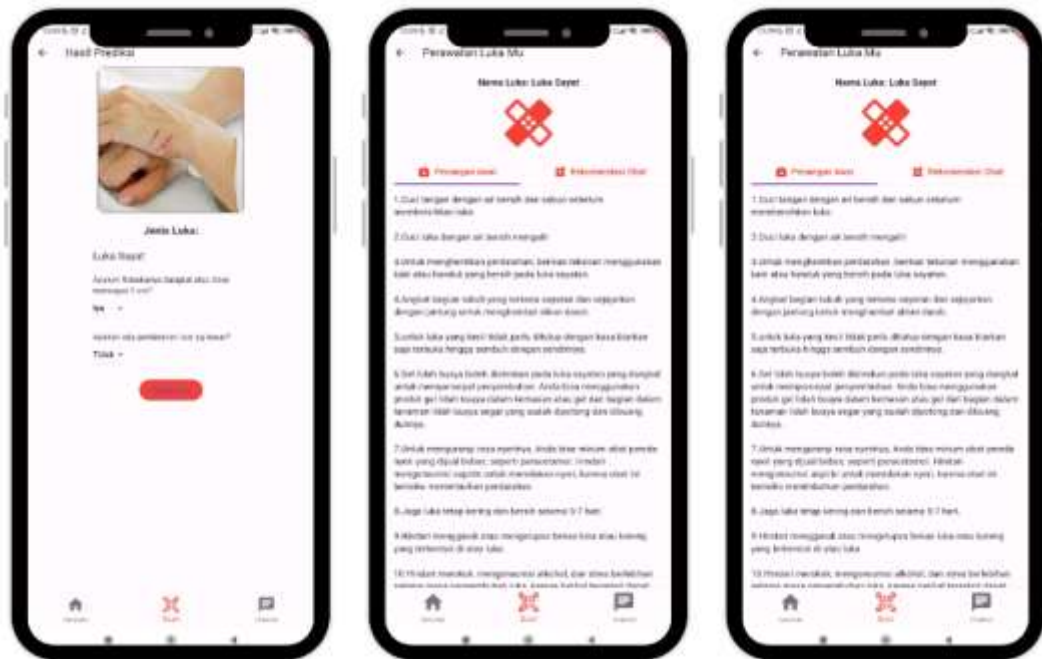




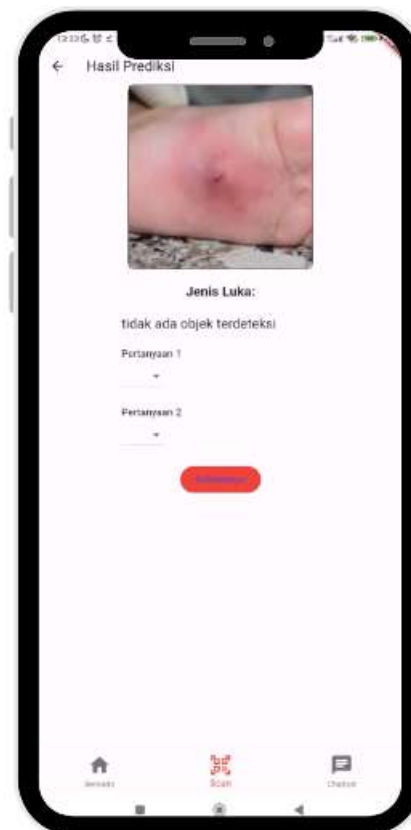
Gambar 4.47 Hasil Uji Data Uji No 11



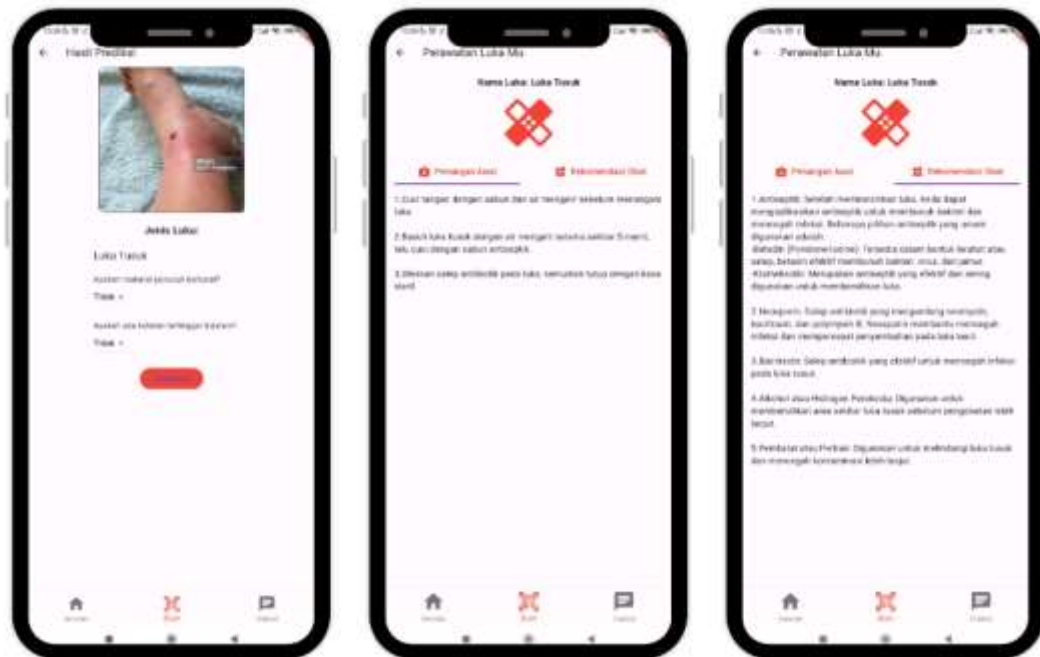
Gambar Hasil Uji Data Uji No 12



Gambar 4.48 Hasil Uji Data Uji No 13



Gambar 4.49 Hasil Uji Data Uji No 14

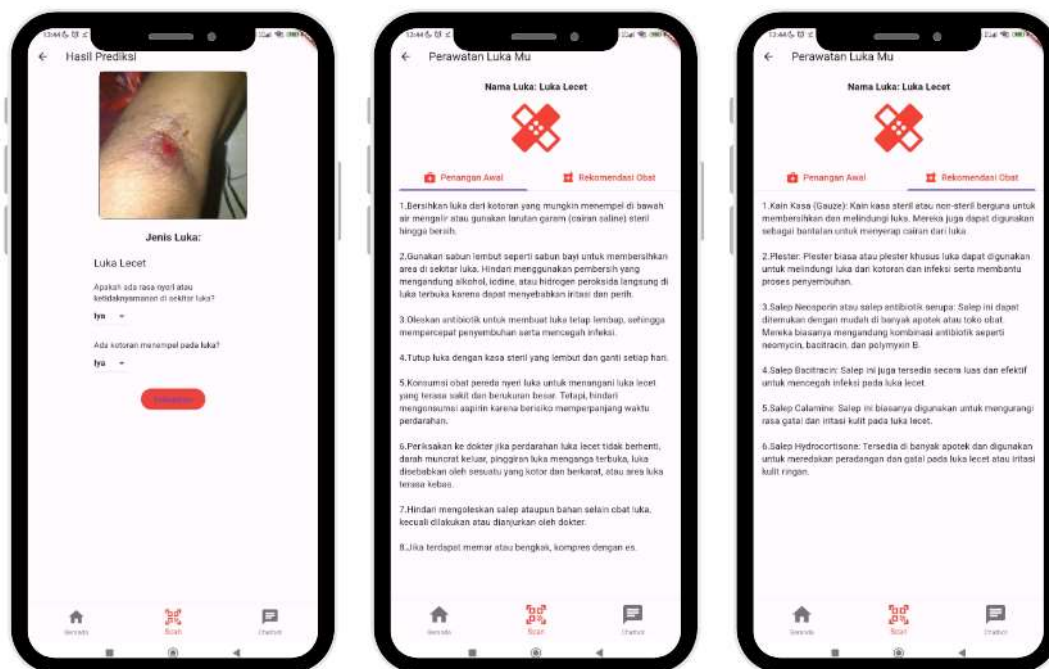


Gambar 4.50 Hasil Uji Data Uji No 15

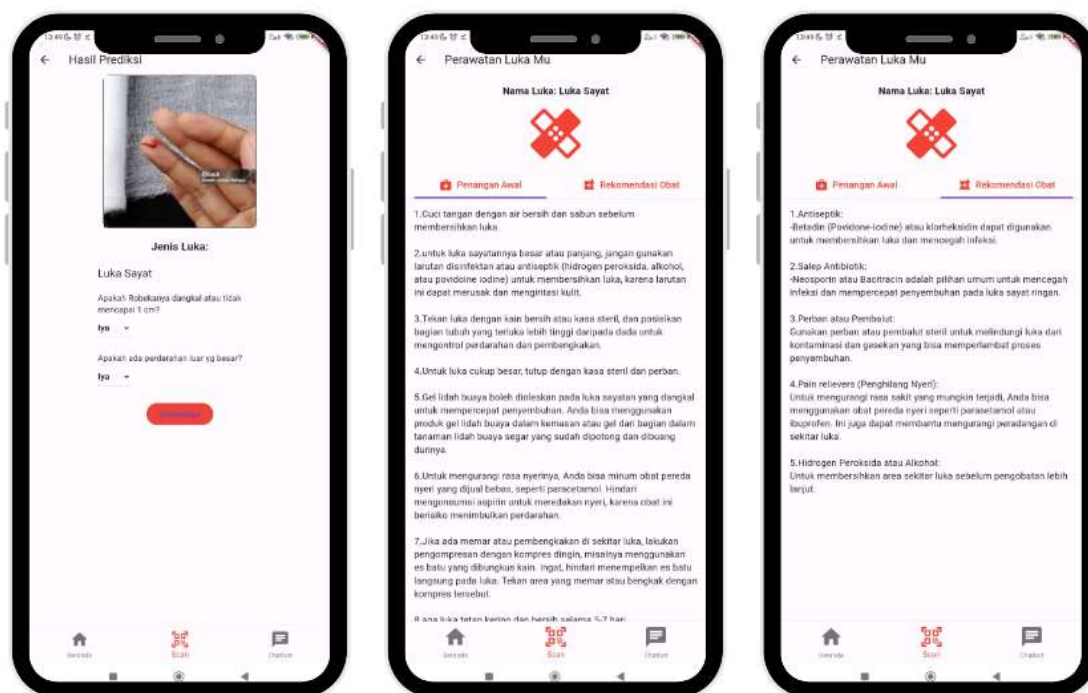


Gambar 4.51 Hasil Uji Data Uji No 16

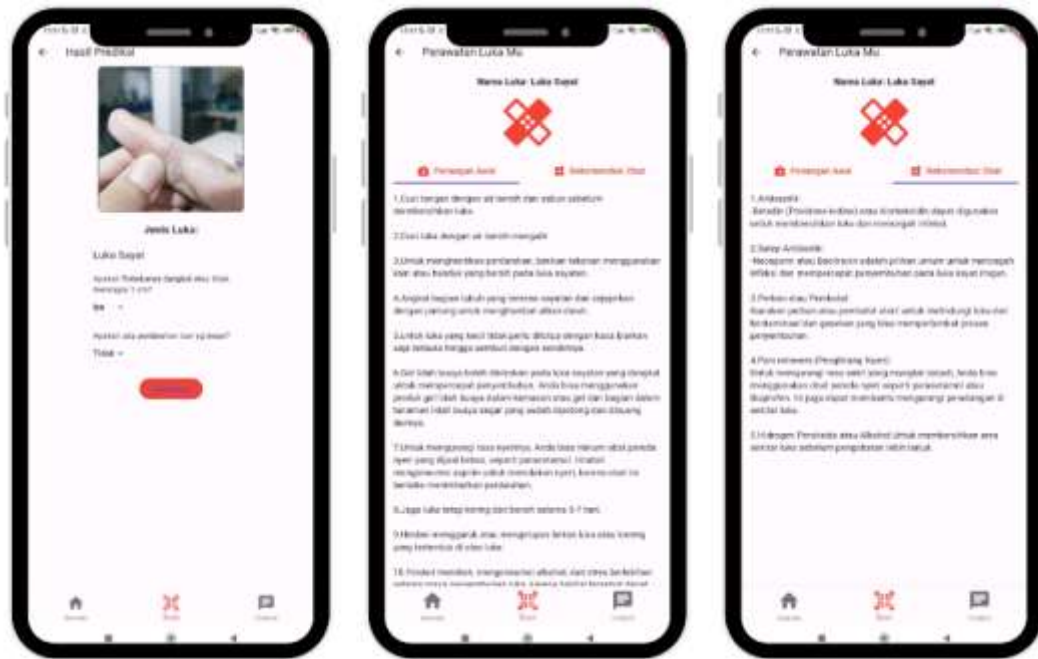




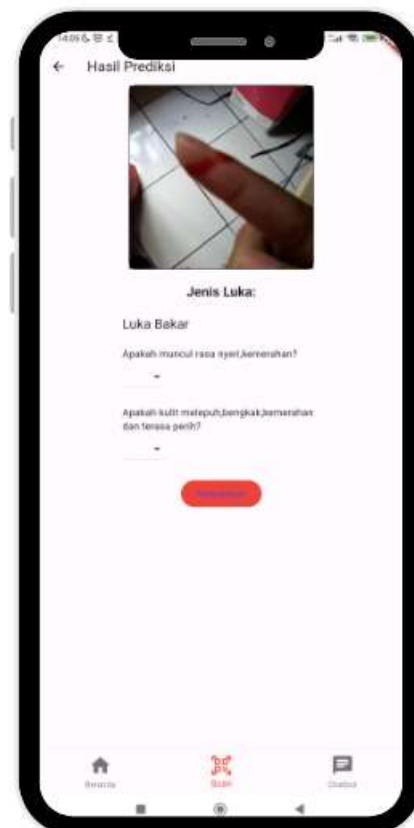
Gambar 4.52 Hasil Uji Data Uji No 17



Gambar 4.53 Hasil Uji Data Uji No 18



Gambar 4.54 Hasil Uji Data Uji No 19



Gambar 4.55 Hasil Uji Data Uji No 20

**Tabel 4.33 Hasil Uji Sistem**

<b>Data Uji</b>	<b>Nilai Sebenarnya</b>	<b>Hasil Uji Sistem</b>
Data Uji 1	Luka Sayat	Luka Sayat
Data Uji 2	Luka Bakar	Luka Bakar
Data Uji 3	Luka Bakar	Luka Bakar
Data Uji 4	Luka Lecet	Luka Lecet
Data Uji 5	Luka Sayat	Luka Sayat
Data Uji 6	Luka Lecet	Luka Lecet
Data Uji 7	Luka Tusuk	Luka Tusuk
Data Uji 8	Luka Lecet	Luka Lecet
Data Uji 9	Luka Lecet	Luka Bakar
Data Uji 10	Luka Bakar	Luka Bakar
Data Uji 11	Luka Bakar	Luka Lecet
Data Uji 12	Luka Sayat	Luka Sayat
Data Uji 13	Luka Sayat	Luka Sayat
Data Uji 14	Luka Tusuk	Tidak Ada Objek Terdeteksi
Data Uji 15	Luka Tusuk	Luka Tusuk
Data Uji 16	Luka Lecet	Tidak Ada Objek Terdeteksi
Data Uji 17	Luka Lecet	Luka Lecet
Data Uji 18	Luka Sayat	Luka Sayat
Data Uji 19	Luka Sayat	Luka Sayat
Data Uji 20	Luka Sayat	Luka Bakar

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berikut merupakan kesimpulan menggunakan analisis, perancangan, implementasi dan pengujian menggunakan metode deteksi objek YOLOv8 yang memanfaatkan arsitektur *Convolutional Neural Networks* (CNN) dalam identifikasi luka ringan untuk manajemen perawatan luka

1. Metode deteksi objek YOLOv8 yang *memanfaatkan arsitektur Convolutional Neural Networks* (CNN) dapat digunakan untuk mengidentifikasi gambar luka ringan
2. Penelitian ini menunjukkan bahwa YOLOv8 mampu mengidentifikasi luka ringan dengan variasi gambar yg beragam. Dalam pengujian ini, model YOLOv8 dilatih menggunakan dataset yang terdiri dari 845 gambar untuk training, 89 gambar untuk validasi, dan diuji menggunakan 49 gambar untuk pengujian. Dengan menggunakan *confusion matrix*, akurasi model terverifikasi mencapai 85%. Namun, untuk meningkatkan performa dan ketangguhan model, langkah-langkah lebih lanjut perlu dilakukan, termasuk pengembangan dataset yang lebih besar dan beragam.
3. Dengan adopsi YOLOv8 untuk identifikasi luka ringan, proses manajemen perawatan luka dapat ditingkatkan secara signifikan. Ini termasuk penanganan awal dan rekomendasi obat yang dapat di beli di apotek terdekat
4. Terdapat perbedaan dalam performa deteksi antar kelas. Untuk kelas "Luka Bakar" memiliki nilai mAP yang lebih tinggi yaitu 0.73 dibandingkan dengan kelas "Luka Bakar" sebesar 0.4 ini mengindikasikan bahwa luka tusuk akan sulit dikenali oleh model, ini disebabkan karena dataset kelas "Luka Tusuk" memiliki jumlah yang paling sedikit dan kurang beragam dibandingkan dengan kelas lainnya.
5. Model akan mengalami kesulitan dalam mengenali luka jika gambar yang di inputkan tidak menyoroti luka dengan jelas, atau jika terdapat efek blur dan

pencapaian yang kurang seperti yang terlihat pada uji sistem dengan data uji No 9, 11, 14, 16, 20.

## **5.2 Saran**

Berikut beberapa saran yang dapat dijadikan rekomendasi untuk bahan pertimbangan bagi penelitian lanjutan

1. Melakukan pengumpulan data yang lebih luas dan representatif dari berbagai jenis luka ringan serta kondisi lingkungan yang berbeda terutama pada luka tusuk untuk meningkatkan keberagaman dataset yang digunakan dalam pelatihan model.
2. diperlukan upaya untuk terus meningkatkan kinerja model agar dapat memberikan hasil yang lebih akurat dan berguna dalam manajemen perawatan luka.
3. Menambahkan fitur dalam aplikasi atau sistem yang memungkinkan pengguna untuk mencari rumah sakit atau tempat kesehatan terdekat berdasarkan lokasi pengguna, sehingga mereka dapat dengan mudah menemukan tempat untuk mendapatkan perawatan luka lanjutan
4. Menyediakan fitur dalam aplikasi atau sistem anda yang memungkinkan pengguna untuk melakukan konsultasi langsung dengan tenaga kesehatan, seperti dokter atau perawat, untuk mendapatkan saran dan rekomendasi terkait perawatan luka pengguna.

## DAFTAR PUSTAKA

- Ahad, M. T., Li, Y., Song, B., & Bhuiyan, T. (2023). Comparison of CNN-based deep learning architectures for rice diseases classification. *Artificial Intelligence in Agriculture*, 9, 22–35. <https://doi.org/10.1016/j.aiia.2023.07.001>
- Badan Pusat Statistik. (2022). *Jumlah Kecelakaan, Korban Mati, Luka Berat, Luka Ringan, dan Kerugian Materi 2019-2021*. Badan Pusat Statistik. <https://www.bps.go.id/indicator/17/513/1/jumlah-kecelakaan-korban-mati-luka-berat-luka-ringan-dan-kerugian-materi.html>
- Carrión, H., Jafari, M., Bagood, M. D., Yang, H., Isseroff, R. R., & Gomez, M. (2022). Automatic wound detection and size estimation using deep learning algorithms. *PLOS Computational Biology*, 18(3), e1009852. <https://doi.org/10.1371/journal.pcbi.1009852>
- Criollo-Mendoza, M. S., Contreras-Angulo, L. A., Leyva-López, N., Gutiérrez-Grijalva, E. P., Jiménez-Ortega, L. A., & Heredia, J. B. (2023). Wound Healing Properties of Natural Products: Mechanisms of Action. *Molecules*, 28(2), 598. <https://doi.org/10.3390/molecules28020598>
- Darmaputra, I. (2022). *Penanganan Awal Luka Bakar*. Kementerian Kesehatan RI. [https://yankes.kemkes.go.id/view\\_artikel/1206/penanganan-awal-luka-bakar](https://yankes.kemkes.go.id/view_artikel/1206/penanganan-awal-luka-bakar)
- Hananti, A. (2023). *Luka Sayatan, Ditangani Sendiri atau Harus oleh Dokter?* Alodokter. <https://www.alodokter.com/luka-sayatan-ditangani-sendiri-atau-harus-oleh-dokter>
- Katyusha, W. (2021). *Luka Tusuk: Pertolongan Pertama dan Cara Perawatannya*. Hellosehat. <https://hellosehat.com/hidup-sehat/pertolongan-pertama/luka-tusuk/>
- Katyusha, W. (2022). *Pertolongan Pertama Luka Lecet dan Cara Merawatnya*. Hellosehat. <https://hellosehat.com/hidup-sehat/pertolongan-pertama/luka-lecet/>
- Linggom, T., & Sihalo, K. (2022). KEMATIAN AKIBAT KEKERASAN TAJAM PADA DADA YANG MENEMBUS JANTUNG. *Majalah Ilmiah METHODA*, 12(1), 1–6. <https://doi.org/10.46880/methoda.Vol12No1.pp1-6>
- Melati, M., Wirasti, W., Nizmah, N., & Slamet, S. (2022). Uji efektivitas ekstrak etanol pelepah pisang susu terhadap penyembuhan luka sayat pada punggung kelinci. *Jurnal Ilmiah Kesehatan*, 15(2), 86–92. <https://doi.org/10.48144/jiks.v15i2.1126>
- Mesbahi, S. C., Mahraz, M. A., Riffi, J., & Tairi, H. (2023). Hand Gesture Recognition Based on Various Deep Learning YOLO Models. *International Journal of Advanced Computer Science and Applications*, 14(4), 307–319. <https://doi.org/10.14569/IJACSA.2023.0140435>

- Patel, R., & Patel, S. (2020). A comprehensive study of applying convolutional neural network for computer vision. *International Journal of Advanced Science and Technology*, 29(6 Special Issue), 2161–2174.
- Undang Undang Lalu Lintas dan Angkutan Jalan, (2009).
- Risal Wintoko, A. D. N. Y. (2020). Manajemen Terkini Perawatan Luka. *Jurnal Kesehatan Universitas Lampung*, 4, 183–189.
- Rohaziat, N., Tomari, M. R. M., Zakaria, W. N. W., & Othman, N. (2020). White Blood Cells Detection using YOLOv3 with CNN Feature Extraction Models. *International Journal of Advanced Computer Science and Applications*, 11(10), 459–467. <https://doi.org/10.14569/IJACSA.2020.0111058>
- Rostami, B., Anisuzzaman, D. M., Wang, C., Gopalakrishnan, S., Niezgoda, J., & Yu, Z. (2021). Multiclass wound image classification using an ensemble deep CNN-based classifier. *Computers in Biology and Medicine*, 134, 104536. <https://doi.org/10.1016/j.combiomed.2021.104536>
- Shah, S. M. A. H., Rizwan, A., Atteia, G., & Alabdulhafith, M. (2023). CADFU for Dermatologists: A Novel Chronic Wounds & Ulcers Diagnosis System with DHuNeT (Dual-Phase Hyperactive UNet) and YOLOv8 Algorithm. *Healthcare*, 11(21), 2840. <https://doi.org/10.3390/healthcare11212840>
- Syahputra, A., Azhary, M. F., Binti, A., Rahman, A., & Saad, A. (2024). *Occupancy Measurement in Under-Actuated Zones : YOLO-based Deep Learning Approach*. 15(2), 757–769.
- Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716. <https://doi.org/10.3390/make5040083>
- 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), 814–821. <https://doi.org/10.14569/IJACSA.2023.0141182>
- Vo, H. T., Mui, K. C., Thien, N. N., & Tien, P. P. (2024). Automating Tomato Ripeness Classification and Counting with YOLOv9. *International Journal of Advanced Computer Science and Applications*, 15(4), 1120–1128. <https://doi.org/10.14569/IJACSA.2024.01504113>



## LAMPIRAN

### Nilai Matriks Lengkap Pada Analisis Perhitungan

#### Matriks Citra Red Kordinat (228, 260)

205	208	210	208	205	204	204	209	212	207	200	200	201	202	209	210	212	212	208	204	202	207	217
207	207	208	210	208	204	198	204	205	195	185	187	193	198	205	207	207	210	207	201	199	205	216
202	209	211	206	202	204	201	209	207	200	186	176	189	200	199	202	209	211	205	195	195	200	208
206	209	209	206	203	203	206	210	205	205	197	183	189	195	198	201	204	207	209	202	201	202	205
208	207	207	207	204	201	209	212	203	205	202	191	196	202	200	201	201	205	212	211	210	208	206
206	206	206	206	204	201	207	211	202	201	199	194	203	202	191	191	199	205	211	219	218	214	209
208	212	215	213	210	208	206	207	199	202	200	193	196	180	174	186	202	211	219	228	220	211	208
198	207	212	209	204	203	204	203	197	207	209	191	184	167	175	199	215	222	232	236	220	204	201
199	206	206	202	196	192	189	183	192	206	192	171	170	167	184	208	221	225	233	228	209	196	198
201	205	206	203	200	197	189	176	173	173	161	157	165	171	184	205	215	221	229	220	202	193	198
203	205	209	208	203	199	186	175	164	160	161	169	177	182	189	203	211	216	224	214	199	193	200
203	206	214	216	205	196	192	194	191	186	187	188	187	194	198	208	212	213	216	211	198	193	200
198	202	216	222	209	199	202	208	207	205	205	203	199	200	202	210	213	209	209	210	199	194	198
198	198	213	222	210	202	200	199	200	205	209	213	215	203	202	208	213	212	212	208	198	194	197
202	200	208	214	206	201	202	202	203	206	207	210	211	205	200	205	213	217	220	206	196	193	196
201	204	203	205	207	202	200	214	220	215	217	214	204	199	194	195	204	212	214	205	197	192	195
207	209	207	206	202	194	186	193	204	212	214	208	199	193	190	193	200	205	206	205	200	196	199
209	212	211	208	202	192	188	188	193	202	200	192	189	185	186	192	200	205	206	207	204	201	202
204	209	209	208	205	195	200	196	192	195	197	199	198	188	190	196	202	206	207	213	210	206	205
204	208	208	207	204	195	196	198	194	192	199	208	207	197	199	201	204	206	208	215	212	207	204
208	210	207	204	200	190	194	207	205	197	203	210	205	201	202	203	205	209	212	210	206	201	199
205	208	208	203	193	185	194	206	213	212	206	202	200	197	202	211	219	219	214	213	216	213	203

#### Matriks Citra Green Kordinat (228, 260)

125	126	126	123	120	118	116	120	122	118	112	111	113	115	122	123	125	125	121	120	120	125	135
126	125	124	125	122	118	109	113	114	105	96	97	104	111	117	119	120	122	121	118	117	123	134
121	126	126	121	116	117	111	116	115	109	96	86	99	111	112	115	122	124	119	112	114	119	126
124	125	124	120	117	116	114	116	111	112	105	92	99	106	110	114	118	121	124	120	120	121	123
126	123	121	120	117	114	116	116	108	110	109	99	106	113	112	114	115	120	127	129	128	127	124
124	121	119	119	117	112	113	113	105	105	105	101	112	112	102	105	114	121	128	137	136	132	128
124	127	128	126	122	119	111	108	102	105	106	100	104	89	85	100	116	126	135	146	139	130	126
113	121	125	121	116	115	109	104	99	110	113	98	92	76	86	112	130	137	149	154	138	122	119
113	117	116	111	105	100	92	83	93	108	96	77	77	77	94	121	135	140	148	145	128	114	116
114	116	114	111	106	102	92	77	73	75	65	62	72	80	95	116	129	135	143	137	121	111	116
117	117	118	116	110	105	90	77	66	63	65	76	85	91	100	116	126	131	138	131	117	111	118
118	119	125	125	114	105	97	97	94	90	93	96	96	104	110	121	127	127	130	128	117	112	118
115	117	129	133	120	109	109	112	113	112	113	112	109	112	115	123	127	124	124	127	117	113	117
116	115	128	136	123	114	109	106	108	114	118	124	126	116	115	123	128	128	128	126	117	113	116
122	118	126	130	122	116	114	112	114	117	118	122	125	119	115	120	128	133	137	124	115	111	114
124	125	123	124	126	122	116	128	133	128	130	129	119	114	109	111	121	129	131	123	116	111	113
131	132	129	127	124	116	104	108	120	127	129	124	116	109	106	109	117	122	123	123	118	115	117
133	135	134	131	126	116	108	105	110	119	118	110	106	102	103	109	117	122	123	126	122	119	120
129	134	135	134	131	120	121	115	111	114	116	117	117	107	108	113	119	123	124	131	129	125	123
130	134	134	134	131	121	118	118	114	111	119	127	126	117	117	119	121	123	125	133	131	126	123
135	137	134	131	127	118	117	128	126	118	124	129	124	120	121	121	122	126	130	128	125	120	117
133	136	136	131	121	114	120	130	137	136	129	124	121	118	122	131	138	137	132	134	137	135	125



### Matriks Citra Blue Kordinat (228, 260)

122	124	125	122	120	118	112	113	116	112	106	106	108	110	117	119	120	121	117	117	118	123	134
123	122	122	123	120	116	104	105	107	98	89	91	98	106	113	115	115	118	116	115	115	121	133
117	122	123	116	111	113	104	108	106	101	88	79	92	105	107	110	118	120	116	109	111	116	125
119	120	119	115	111	110	106	106	102	103	96	84	92	100	105	110	114	118	121	117	117	118	122
119	117	116	114	110	108	106	104	97	100	99	91	99	107	107	109	112	117	124	126	126	124	123
116	114	113	111	109	105	102	101	92	94	95	92	104	105	97	100	110	118	125	135	134	130	126
117	119	120	117	113	111	100	95	88	93	95	91	96	82	79	95	113	124	134	143	136	127	125
106	113	116	111	106	105	96	90	86	98	102	88	84	69	80	108	126	134	147	152	136	120	118
110	114	112	105	97	91	80	70	80	96	85	68	69	69	89	117	132	138	147	143	125	112	115
114	115	113	107	101	95	80	63	61	64	55	53	64	72	89	113	126	133	142	135	118	109	115
118	116	116	112	105	99	79	64	53	51	56	67	77	84	94	112	122	129	137	129	115	109	117
119	119	123	121	108	97	86	85	82	80	83	87	88	97	105	117	123	125	128	126	114	109	117
115	116	126	128	113	102	99	102	103	101	103	104	102	106	110	119	124	121	121	125	115	110	115
117	114	125	131	117	107	100	96	98	105	110	116	119	110	110	118	123	123	125	123	114	110	114
123	118	124	126	116	109	105	103	104	109	111	114	117	112	109	115	123	128	133	121	112	109	113
124	125	121	120	120	113	107	119	125	120	123	121	112	109	104	105	115	123	125	119	113	108	112
132	131	127	123	117	108	96	101	113	121	123	117	109	104	100	103	111	116	118	119	116	112	116
135	135	132	127	119	107	101	99	105	114	112	104	100	96	97	103	111	116	117	122	120	117	119
131	134	133	130	124	112	114	110	106	109	111	111	110	100	102	107	113	117	119	128	126	122	122
132	134	132	129	124	112	112	114	109	107	114	122	120	110	111	113	115	117	118	129	128	123	121
137	137	132	127	120	109	112	124	121	113	119	125	118	113	114	115	116	119	122	124	122	117	116
132	133	133	127	116	107	114	124	131	129	123	118	115	112	117	125	132	132	127	129	132	130	120

### Matriks Hasil Konvolusi (228, 260) Red

-15	14	12	-21	53	-12	-30	-15	8	24	-45
-11	18	-7	1	30	11	-23	-17	-12	20	-13
-6	10	-4	18	3	6	30	-37	-40	-6	25
-28	23	11	11	-13	51	17	-105	-46	35	42
-18	22	35	35	15	2	-45	-90	-39	76	14
-35	22	37	18	9	-10	-26	-47	-13	53	-2
-39	12	21	-6	-11	-4	21	-35	-2	48	2
-8	3	27	-39	-11	24	30	-33	-23	47	3
-7	18	35	-15	-22	25	20	-36	-17	5	8
-8	15	19	-1	-8	-11	19	-20	-16	-1	20

### Matriks Hasil Konvolusi (228, 260) Green

-4	18	22	-15	47	-12	-35	-16	6	10	-44
0	21	9	7	24	6	-30	-21	-15	8	-11
6	12	16	25	-5	-2	23	-46	-45	-13	25
-19	26	31	18	-21	42	8	-116	-51	27	44
-4	27	47	42	6	-8	-55	-101	-39	63	16
-22	28	48	23	2	-19	-35	-55	-12	41	-2
-30	18	33	-3	-14	-11	15	-38	-6	40	2
-1	6	38	-33	-10	17	30	-36	-25	42	5
-5	17	48	-8	-22	24	22	-36	-17	1	9
-9	14	33	5	-8	-8	21	-16	-17	-6	22

## Matriks Hasil Konvolusi (228, 260) Blue

-8	19	31	-9	46	-15	-39	-16	4	5	-48
-3	26	16	11	22	0	-36	-25	-17	7	-16
3	17	24	31	-12	-10	16	-52	-48	-13	22
-15	32	40	22	-28	33	1	-123	-57	31	39
1	38	64	45	-2	-14	-62	-108	-46	68	11
-13	39	62	26	-4	-25	-42	-60	-17	42	-5
-20	29	42	-1	-19	-14	9	-41	-9	38	-1
7	19	45	-34	-15	19	25	-36	-27	35	0
6	32	49	-13	-22	27	20	-36	-19	-8	5
3	29	30	2	-8	-4	21	-17	-15	-17	17

## Hasil Dari Proses RELU pada matriks (228, 260) Red

0	14	12	0	53	0	0	0	8	24	0
0	18	0	1	30	11	0	0	0	20	0
0	10	0	18	3	6	30	0	0	0	25
0	23	11	11	0	51	17	0	0	35	42
0	22	35	35	15	2	0	0	0	76	14
0	22	37	18	9	0	0	0	0	53	0
0	12	21	0	0	0	21	0	0	48	2
0	3	27	0	0	24	30	0	0	47	3
0	18	35	0	0	25	20	0	0	5	8
0	15	19	0	0	0	19	0	0	0	20

## Hasil Dari Proses RELU pada matriks (228, 260) Green

0	18	22	0	47	0	0	0	6	10	0
0	21	9	7	24	6	0	0	0	8	0
6	12	16	25	0	0	23	0	0	0	25
0	26	31	18	0	42	8	0	0	27	44
0	27	47	42	6	0	0	0	0	63	16
0	28	48	23	2	0	0	0	0	41	0
0	18	33	0	0	0	15	0	0	40	2
0	6	38	0	0	17	30	0	0	42	5
0	17	48	0	0	24	22	0	0	1	9
0	14	33	5	0	0	21	0	0	0	22

## Hasil Dari Proses RELU pada matriks (228, 260) Blue

0	19	31	0	46	0	0	0	4	5	0
0	26	16	11	22	0	0	0	0	7	0
3	17	24	31	0	0	16	0	0	0	22
0	32	40	22	0	33	1	0	0	31	39
1	38	64	45	0	0	0	0	0	68	11
0	39	62	26	0	0	0	0	0	42	0
0	29	42	0	0	0	9	0	0	38	0
7	19	45	0	0	19	25	0	0	35	0
6	32	49	0	0	27	20	0	0	0	5
3	29	30	2	0	0	21	0	0	0	17

### Hasil Max Pooling citra (228, 260) Red

18	18	12	53	53	11	0	8	24	24
18	18	18	30	30	30	30	0	20	25
23	23	18	18	51	51	30	0	35	42
23	35	35	35	51	51	17	0	76	76
22	37	37	35	15	2	0	0	76	76
22	37	37	18	9	21	21	0	53	53
12	27	27	0	24	30	30	0	48	48
18	35	35	0	25	30	30	0	47	47
18	35	35	0	25	25	20	0	5	20

### Hasil Max Pooling citra (228, 260) Green

21	22	22	47	47	6	0	6	10	10
21	21	25	25	24	23	23	0	8	25
26	31	31	25	42	42	23	0	27	44
27	47	47	42	42	42	8	0	63	63
28	48	48	42	6	0	0	0	63	63
28	48	48	23	2	15	15	0	41	41
18	38	38	0	17	30	30	0	42	42
17	48	48	0	24	30	30	0	42	42
17	48	48	5	24	24	22	0	1	22

### Hasil Max Pooling citra (228,260) Blue

26	31	31	46	46	0	0	4	7	7
26	26	31	31	22	16	16	0	7	22
32	40	40	31	33	33	16	0	31	39
38	64	64	45	33	33	1	0	68	68
39	64	64	45	0	0	0	0	68	68
39	62	62	26	0	9	9	0	42	42
29	45	45	0	19	25	25	0	38	38
32	49	49	0	27	27	25	0	35	35
32	49	49	2	27	27	21	0	0	17