

# **DETEKSI ANTUSIASME SISWA DALAM PEMBELAJARAN DARING DENGAN YOLOV8**



Disusun Oleh:

N a m a : Kartika Salma  
NIM : 20523172

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

DETEKSI ANTUSIASME SISWA DALAM PEMBELAJARAN  
DARING DENGAN YOLOV8

TUGAS AKHIR



## HALAMAN PENGESAHAN DOSEN PENGUJI

**DETEKSI ANTUSIASME SISWA DALAM PEMBELAJARAN  
DARING DENGAN YOLOV8****TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia  
Yogyakarta, 15 Mei 2024

Tim Penguji

Dr. Syarif Hidayat, S.T., M.I.T.

**Anggota 1**Chanifah Indah Ratnasari, S.Kom.,  
M.Kom.**Anggota 2**

Arrie Kurniawardhani, S.Si., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana  
Fakultas Teknologi Industri  
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. )

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Kartika Salma

NIM : 20523172

Tugas akhir dengan judul: Deteksi Antusiasme Siswa dalam Pembelajaran Daring dengan YOLOV8

**DETEKSI ANTUSIASME SISWA DALAM PEMBELAJARAN  
DARING DENGAN YOLOV8**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 15 Mei 2024



( Kartika Salma )

## **HALAMAN PERSEMBAHAN**

Laporan tugas akhir ini dengan penuh rasa syukur peneliti dedikasikan untuk keluarga tercinta, terutama kepada diri peneliti yang telah bersedia dan mampu menyelesaikan tugas akhir ini. Terima kasih kepada orang tua yang selalu memberikan dukungan tanpa henti, serta kepada kakak tercinta yang selalu memberikan bantuan dan selalu mengingatkan agar tugas akhir ini segera terselesaikan. Tidak lupa, laporan ini juga dipersembahkan sebagai ungkapan terima kasih kepada jurusan Informatika sebagai wadah pembelajaran dan sumbangsih ilmu yang tak ternilai.

## HALAMAN MOTO

“Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum, sebelum mereka mengubah keadaan diri mereka sendiri.”

– QS Ar Rad 11

“Jika sedang merasa kecewa, jangan menyalahkan kebahagiaan atau pencapaian orang lain. Sebenarnya, ini hanyalah masalah waktu; mereka mungkin mendapatkannya lebih dulu, dan kamu juga akan mendapatkannya, hanya saja mungkin belum saatnya. Siapa tahu, apa yang sedang Allah persiapkan untukmu adalah yang terbaik dan telah lama kamu idamkan. Jangan bersikap jahat terhadap pencapaian orang lain, tidak perlu menyalahkan keadaan, dan jangan kehilangan keyakinanmu kepada Allah.”

– Charisma Putri Sulaiman

## KATA PENGANTAR

Bismillahirrahmanirrahim,

Assalamu'alaikum Warahmatullahi Wabarakatuh.

Dengan penuh rasa syukur, penulis ingin mengucapkan puji dan terima kasih kepada Allah SWT, karena dengan rahmat dan karunia-Nya, penulis berhasil menyelesaikan laporan tugas akhir berjudul "Deteksi Antusiasme Siswa Dalam Pembelajaran Daring dengan YOLOv8". Penulisan skripsi ini bertujuan untuk memenuhi persyaratan sidang skripsi di Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penulis menyadari bahwa tanpa dukungan dan motivasi, laporan ini tidak akan terselesaikan tepat waktu. Oleh karena itu, penulis mengucapkan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang senantiasa memberikan rahmat, petunjuk, dan kekuatan dalam menyelesaikan penelitian ini.
2. Bapak dan Ibu penulis, Bapak Kusmono dan Ibu Titik Riyani, yang telah memberikan dukungan secara materiil dan emosional.
3. Kakak dan kakak ipar penulis, Nino Setyo Aji, Dessy Eka Setyaningrum, Budi Sulistyio Jati, dan Rakhma Shafrida Kurnia. Terima kasih atas bantuan, semangat, serta pengingatannya untuk menyelesaikan penelitian ini.
4. Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
5. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
6. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
7. Bapak Dr. Syarif Hidayat, S.T., M.I.T., selaku dosen pembimbing tugas akhir, yang memberikan bimbingan dan masukan berharga.
8. Bapak dan Ibu dosen Program Studi Informatika, yang telah berjasa dalam memberikan ilmu yang bermanfaat selama penelitian ini.
9. Gabriel Imam Andaru, yang turut andil dalam penelitian ini dengan dukungan dan waktunya.
10. Teman-teman penulis, Biandra Anone Aryantoputri, Fatimah Azzahra Kusuma Dewi, Intan Nabila Azmi, Mayla Ayuni Sonya, Nadira Adiva Wibowo, Sonya

Ainurohmah, dan Syuhda Fakhrunnisa yang selalu memberikan semangat dan menjadi teman yang baik.

11. Seluruh pihak yang tidak dapat peneliti sebutkan satu per satu, terima kasih atas semua dukungan baik secara langsung maupun tidak langsung dalam menyelesaikan penelitian ini.

Semoga segala ilmu, dukungan, bimbingan, dan doa yang telah diberikan kepada peneliti memperoleh balasan yang lebih baik dari Allah SWT. Peneliti menyadari bahwa laporan Tugas Akhir ini belum mencapai tingkat kesempurnaan dan masih memiliki kekurangan. Oleh karena itu, peneliti sangat mengharapkan saran dan kritik yang membangun. Harapannya, laporan Tugas Akhir ini dapat memberikan manfaat bagi semua pihak.

Yogyakarta, 15 Mei 2024

A handwritten signature in black ink, appearing to read 'Kartika Salma' in a cursive style.

( Kartika Salma )



## SARI

Pada era pembelajaran daring, ekspresi wajah siswa menjadi indikator penting sebagai respon emosional terhadap materi pembelajaran. Ekspresi wajah memiliki peran yang signifikan dalam memahami tingkat pemahaman dan keterlibatan siswa dalam pembelajaran. Namun, penelitian sebelumnya hanya mencapai tingkat akurasi sebesar 87%. Penelitian ini bertujuan untuk meningkatkan akurasi menjadi 95,3% dengan menerapkan teknologi *Face Expression Recognition* (FER) menggunakan model YOLOv8. Dengan hasil yang lebih akurat, teknologi ini berpotensi memperkuat efektivitas pembelajaran daring dengan memungkinkan guru untuk secara tepat mengawasi respon emosional siswa dan menyusun pendekatan pembelajaran yang lebih sesuai.

Metode yang digunakan terdiri dari beberapa tahap yang terstruktur. Tahap pertama adalah pengumpulan *dataset* ekspresi wajah, yang kemudian dilakukan pelabelan dan pembagian menjadi subset pelatihan dan pengujian. Selanjutnya, dilakukan pengaturan *environment* dan *hyperparameter* yang optimal. Penelitian ini dilaksanakan dengan tujuan meningkatkan akurasi dari penelitian sebelumnya, dengan fokus pada deteksi ekspresi wajah siswa dalam konteks pembelajaran daring.

Hasil dari penelitian ini menunjukkan tingkat akurasi yang signifikan, mencapai 95,3%, dibandingkan dengan penelitian sebelumnya yang hanya mencapai 87%. Selain itu, waktu inferensi yang diperoleh dari model yang dikembangkan sangatlah cepat, hanya 0,62ms. Kedua pencapaian ini menandakan peningkatan yang substansial dalam kinerja sistem deteksi objek dan pengenalan ekspresi wajah siswa dalam konteks pembelajaran daring. Peningkatan akurasi memastikan bahwa respon emosional siswa terhadap materi pembelajaran dapat dideteksi dan diinterpretasikan dengan lebih baik, memungkinkan guru untuk merespons dengan lebih tepat sesuai dengan kebutuhan siswa. Hal ini berpotensi meningkatkan efektivitas pembelajaran secara keseluruhan. Selain itu, waktu inferensi yang sangat cepat juga memberikan keuntungan praktis yang besar, dengan waktu inferensi yang singkat, sistem dapat bekerja secara *real-time*, memungkinkan interaksi yang lebih lancar antara guru dan siswa dalam situasi pembelajaran yang sesungguhnya.

Kata kunci: ekspresi wajah, respon emosional, pembelajaran daring, antusiasme siswa, YOLOv8, *face emotion recognition*.

## GLOSARIUM

<i>Antusias</i>	ketertarikan yang kuat terhadap sesuatu yang tercermin dalam tingginya tingkat kebahagiaan dan semangat yang membara.
<i>Batch size</i>	seberapa banyak data yang dimasukkan ke dalam model saat melatih.
<i>Epoch</i>	satu putaran lengkap model <i>machine learning</i> atau <i>deep learning</i> melalui seluruh <i>dataset</i> pelatihan.
<i>Hyperparameter</i>	parameter eksternal yang perlu diatur sebelum proses pelatihan model, dan memengaruhi kinerja serta proses pembelajaran model.
<i>Loss</i>	nilai yang mengukur seberapa besar perbedaan antara prediksi model dan nilai sebenarnya (label) dari data pelatihan.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Penelitian .....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metode Penelitian .....	4
1.7 Sistematika Penulisan .....	5
<b>BAB II LANDASAN TEORI.....</b>	<b>6</b>
2.1 Deteksi Wajah .....	6
2.2 Emosi.....	7
2.3 Definisi Antusias dan Tidak Antusias .....	8
2.4 CNN.....	10
2.4.1 Konvolusi.....	11
2.4.2 <i>Max Pooling</i> .....	11
2.4.3 SiLU.....	12
2.5 YOLO.....	13
2.5.1 YOLO V8 .....	16
2.5.2 IoU .....	18
2.5.3 <i>Precision</i> .....	19
2.5.4 mAP.....	19
2.6 Penelitian Sebelumnya .....	20
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>24</b>
3.1 Data.....	24
3.2 Perancangan Penelitian.....	25
3.2.1 Mengumpulkan <i>Dataset</i> .....	25
3.2.2 Melabeli <i>Dataset</i> .....	25
3.2.3 Membagi <i>Dataset</i> .....	29
3.2.4 <i>Environment Setting</i> .....	30
3.2.5 <i>Hyperparameter Setting</i> .....	30
3.2.6 Melatih Model .....	31
3.2.7 Menguji Model .....	33
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>35</b>
4.1 Implementasi .....	35
4.2 Hasil Pengolahan <i>Dataset</i> .....	35
4.2.1 Hasil Proses Pelabelan <i>Dataset</i> .....	35

4.2.2 Hasil Proses Pembagain <i>Dataset</i> .....	38
4.3 Hasil Pelatihan .....	38
4.4 Hasil Pengujian.....	43
4.4.1 Hasil Pengujian pada Ukuran Gambar 48x48 .....	44
4.4.2 YOLOv8n .....	45
4.4.3 YOLOv8s.....	47
4.4.4 YOLOv8m .....	49
4.4.5 YOLOv8l.....	50
4.5 Pembahasan .....	52
4.5.1 Pembahasan Hasil Pengujian pada Video Pembelajaran Daring.....	53
BAB V KESIMPULAN DAN SARAN .....	63
5.1 Kesimpulan.....	63
5.2 Saran .....	64
DAFTAR PUSTAKA .....	65

**DAFTAR TABEL**

Tabel 2.1 Indikator Antusias dan Tidak Antusias.....	9
Tabel 2.2 Penelitian mengenai deteksi dan pengenalan ekspresi emosi wajah. ....	21
Tabel 3.1 Kelas emosi pada <i>dataset</i> FER2013 .....	24
Tabel 4.1 Tabel Perbandingan <i>Box Loss</i> dan <i>Class Loss</i> .....	42
Tabel 4.2 Hasil Pengujian YOLOv8 ukuran n, s, m, dan l .....	52
Tabel 4.3 Rekapitulasi Video Uji Pertama .....	58
Tabel 4.4 Rekapitulasi Video Uji Kedua .....	60
Tabel 4.5 Rekapitulasi Video Uji Ketiga.....	61

## DAFTAR GAMBAR

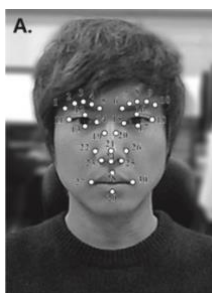
Gambar 1.1 Titik fitur wajah yang digunakan untuk ekspresi wajah .....	1
Gambar 2.1 Tujuh Ekspresi Dasar Manusia .....	8
Gambar 2.2 Ilustrasi Operasi Konvolusi.....	11
Gambar 2.3 Operasi Max Pooling .....	12
Gambar 2.4 Grafik Perbandingan SiLU dan ReLU .....	13
Gambar 2.5 Sistem Deteksi YOLO .....	13
Gambar 2.6 Bounding Box pada YOLO.....	14
Gambar 2.7 Proses Deteksi Model Yolo.....	15
Gambar 2.8 Arsitektur YOLO .....	15
Gambar 2.9 Arsitektur YOLOv5 .....	16
Gambar 2.10 Arsitektur YOLOv8 .....	17
Gambar 2.11 <i>Intersection-over-Union</i> (IoU).....	18
Gambar 3.1 <i>Flowchart</i> perancangan penelitian .....	25
Gambar 3.2 Kode Program Proses Pelabelan .....	27
Gambar 3.3 Proses Pelabelan dengan LabelImg.....	28
Gambar 3.4 Kode Program Proses Pembagian <i>Dataset</i> .....	30
Gambar 3.5 Gambar Data Pelatihan .....	31
Gambar 3.6 Kode untuk Impor Kelas YOLO .....	32
Gambar 3.7 Gambar Data Pengujian .....	33
Gambar 4.1 Hasil Pelabelan oleh Lima Partisipan .....	36
Gambar 4.2 Hasil Pelabelan dengan Kode Program.....	37
Gambar 4.3 Hasil Pelabelan Gambar dengan LabelImg.....	37
Gambar 4.4 Hasil Pembagian <i>Dataset</i> .....	38
Gambar 4.5 Hasil Pelatihan YOLOv8n Menggunakan Ukuran Gambar 48x48 .....	39
Gambar 4.6 Hasil Pelatihan Menggunakan Ukuran Gambar 320x320.....	41
Gambar 4.7 Hasil <i>Testing</i> dengan Ukuran Gambar 48x48 .....	44
Gambar 4.8 <i>Confusion Matrix</i> Model dengan Ukuran Gambar 48x48 .....	44
Gambar 4.9 Hasil <i>testing</i> YOLOv8n .....	45
Gambar 4.10 <i>Confusion Matrix</i> YOLOv8n .....	46
Gambar 4.11 Hasil <i>testing</i> YOLOv8s.....	48
Gambar 4.12 <i>Confusion Matrix</i> YOLOv8s .....	48
Gambar 4.13 Hasil <i>testing</i> YOLOv8m .....	49

Gambar 4.14 <i>Confusion Matrix</i> YOLOv8m .....	50
Gambar 4.15 Hasil <i>testing</i> YOLOv8l .....	51
Gambar 4.16 <i>Confusion Matrix</i> YOLOv8l .....	51
Gambar 4.17 Hasil Deteksi dengan Model yang Dilatih dengan Ukuran Gambar 48x48 .....	54
Gambar 4.18 Model Mendeteksi Foto Profil .....	55
Gambar 4.19 Hasil Deteksi YOLOv8n yang dilatih dengan Gambar Ukuran 320x320 .....	56
Gambar 4.20 Hasil Voting Analisis Ketepatan Prediksi.....	57
Gambar 4.21 Video Uji Pertama.....	58
Gambar 4.22 Video Uji Kedua .....	59
Gambar 4.23 Video Uji Ketiga .....	61

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Ekspresi wajah adalah salah satu sinyal paling kuat, alami, dan universal bagi manusia untuk menyampaikan keadaan emosional dan niat mereka (C. Darwin and P. Prodger, 1998; Tian et al., n.d.). Ekman (1994) dan Friesen (1971) berpendapat bahwa manusia merasakan emosi dasar tertentu dengan cara yang sama terlepas dari latar belakang budaya mereka dan mereka mendefinisikan ekspresi wajah yang khas menjadi tujuh kategori, yaitu, jijik, takut, bahagia, netral, sedih, dan terkejut (P. Ekman, 1994; P. Ekman and W. V. Friesen, 1971). Ekman dan Friesen mengusulkan Facial Action Coding System (FACS) untuk menentukan ekspresi wajah, yang didasarkan pada fakta bahwa ekspresi dihasilkan dari perubahan bagian wajah dengan titik fitur dapat dilihat pada Gambar 1.1.



Gambar 1.1 Titik fitur wajah yang digunakan untuk ekspresi wajah

Sumber: (Y. Bin Kim et al., 2015)

FER (*Facial Expression Recognition*) merupakan bidang dalam pengenalan pola dan pengolahan citra yang berfokus pada mengenali dan menganalisis ekspresi wajah manusia. Seiring berkembangnya kecerdasan buatan dan pembelajaran mesin, banyak algoritma FER telah digunakan untuk mempelajari informasi yang terdapat dalam gambar dan video sehingga dapat mengenali berbagai macam ekspresi wajah dan tingkat keakuratannya semakin meningkat jauh dibandingkan dengan metode tradisional (Agrawal et al., 2020). Dalam upaya meningkatkan akurasi tersebut, algoritma deep learning seperti CNN (*Convolutional Neural Network*) dan YOLO (*You Only Look Once*) telah diimplementasikan dengan hasil yang sangat baik dalam mendeteksi objek dan ekspresi wajah (Redmon et al., 2016). Ketika membandingkan Yolov5 dengan CNN, masing-masing model mendapatkan akurasi sebesar



0.938 dan 0.853 pada kasus pengenalan wajah (Lam et al., 2022). Hal tersebut menjadikan YOLOv5 menjadi salah satu algoritma deteksi yang memiliki tingkat akurasi yang baik. Dalam perbandingan dengan YOLOv8x yang dievaluasi pada *dataset* uji MS COCO 2017, YOLOv5 mencapai nilai AP sebesar 50,7% pada ukuran input yang sama, sedangkan YOLOv8x berhasil mencapai nilai AP sebesar 53,9% dengan ukuran gambar 640 piksel, menunjukkan peningkatan kinerja yang signifikan. Selain itu, YOLOv8x juga memiliki kecepatan lebih tinggi yakni 280 FPS pada perangkat NVIDIA A100 dan TensorRT (Terven et al., 2023). Pengaplikasian kasus FER terhadap berbagai aspek juga telah banyak dilakukan, khususnya dalam bidang Pendidikan.

Kelas daring yang saat ini telah banyak bermunculan dan semakin berkembang, baik dari universitas maupun kursus, membuka peluang yang luas untuk pengaplikasian FER (Allen et al., 2017). Fenomena ini memberikan peluang bagi pengaplikasian teknologi pengenalan emosi (FER) dalam pembelajaran daring. Dengan FER, pengajar dapat memantau respons emosional siswa terhadap materi yang diajarkan dan menyesuaikan pendekatan mereka sesuai dengan kebutuhan siswa. Hal ini dapat membantu meningkatkan efektivitas pembelajaran daring dan memberikan pengalaman belajar yang lebih personal bagi siswa. Adapun respon emosional yang akan diambil dalam penelitian ini adalah respon antusias dan tidak antusias.

Penelitian ini menawarkan solusi yang menggabungkan kelas daring dengan teknologi pengenalan ekspresi wajah (FER) yang menggunakan algoritma YOLOv8. YOLOv8 dipilih karena memiliki kecepatan deteksi yang tinggi tanpa mengorbankan akurasi. Algoritma ini dirancang dengan mempertimbangkan kinerja *real-time*, yang sangat penting dalam konteks kelas daring di mana deteksi cepat sangat dibutuhkan. Selain itu, YOLOv8 mencapai akurasi yang lebih tinggi dalam deteksi ekspresi wajah dibandingkan dengan YOLOv5. Perbandingan ini dilakukan atas beberapa dasar, yaitu karena YOLOv8 dan YOLOv5 dikeluarkan oleh perusahaan yang sama, yaitu Ultralytics dan memiliki arsitektur dasar yang mirip dengan menggunakan *convolutional neural networks* (CNN) untuk mendeteksi objek dalam gambar atau video. Keduanya juga menggunakan CSPNet sebagai bagian penting dari arsitekturnya. Selain itu, Ultralytics juga mengklaim bahwa YOLOv8 memiliki performa yang lebih baik dibanding YOLOv5. Oleh karena itu, pada penelitian ini dilakukan perbandingan dengan YOLOv5 untuk membuktikan apakah YOLOv8 memiliki performa yang lebih baik dibandingkan dengan YOLOv5. Selain itu tujuan utamanya yaitu untuk mendapatkan mAP terbaik. Pada penelitian ini, YOLOv8 dilatih dengan *Dataset Facial Expression Recognition* (FER) 2013 yang berasal dari situs Kaggle dengan tautan

<https://www.kaggle.com/datasets/msambare/fer2013> dengan jumlah 2000 gambar untuk mendeteksi ekspresi wajah yang dimunculkan. Dengan demikian, solusi ini bertujuan untuk memantau respons emosional siswa terhadap materi yang diajarkan dan memberikan evaluasi yang akurat. Selain itu, penelitian ini juga akan menguji apakah dengan menggunakan YOLOv8 dapat mendeteksi suatu ekspresi wajah dengan kecepatan dan keakuratan yang sesuai dengan yang ditampilkan. Diharapkan solusi ini dapat membantu meningkatkan efektivitas pembelajaran daring.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, rumusan masalah pada penelitian ini yaitu bagaimana mengembangkan model deteksi antusiasme siswa dalam pembelajaran daring menggunakan YOLOv8 untuk memperoleh mAP tertinggi dengan membandingkan beberapa ukuran model YOLOv8.

## 1.3 Batasan Penelitian

Dalam rangka memfokuskan isu yang dibahas, penelitian ini mengadopsi beberapa pembatasan masalah. Beberapa batasan masalah yang diterapkan di antaranya adalah:

- a. Penelitian ini akan difokuskan pada deteksi ekspresi wajah yang dikelaskan menjadi antusias dan tidak antusias.
- b. *Dataset* ekspresi wajah yang digunakan berasal dari FER 2013 dan hanya diambil 2000 gambar.
- c. Penelitian akan menggunakan algoritma YOLOv8 sebagai alat untuk mendeteksi ekspresi wajah siswa dalam lingkungan pembelajaran daring.
- d. Pengujian performa sistem deteksi menggunakan metrik *mean Average Precision* (mAP).

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan sebuah model *object detection* dengan arsitektur YOLOv8 untuk mendeteksi respon emosional siswa khususnya antusias dan tidak antusias dalam kelas daring.

## 1.5 Manfaat Penelitian

Manfaat yang diharapkan dapat diperoleh dari penelitian ini adalah sebagai berikut:

- a. Manfaat Teoritis

Penelitian ini akan menyumbangkan pemahaman baru dalam bidang teknologi pengenalan ekspresi wajah (FER) dan integrasi algoritma YOLOv8 dalam konteks pembelajaran daring, yang akan memperkaya pengetahuan dalam domain ini.

b. Manfaat Praktis

Integrasi YOLOv8 dalam penelitian ini akan membantu mengurangi keterlibatan manusia dalam proses analisis deteksi ekspresi wajah siswa, yang akan meningkatkan efisiensi dan waktu respon dalam mendeteksi respons emosional siswa.

## 1.6 Metode Penelitian

Berikut adalah serangkaian langkah yang diterapkan dalam penelitian ini untuk mencapai tujuan yang diinginkan:

- a. Mengumpulkan data, merupakan proses pengumpulan data berupa gambar yang akan digunakan dalam pengembangan program pendeteksian ekspresi.
- b. Melabeli *dataset*, tahap yang dilakukan untuk memberikan informasi tentang nama kelas pada gambar. Tahap ini dilakukan dengan tujuan memberikan tanda terhadap objek yang telah ditetapkan sebelum dilakukan proses pelatihan.
- c. Membagi *dataset*, merupakan proses menentukan komposisi data yang akan digunakan untuk proses pelatihan dan pengujian.
- d. *Environment Setting*, tahap ini mencakup konfigurasi perangkat keras dan perangkat lunak yang diperlukan untuk menjalankan kode, termasuk sistem operasi, library *machine learning*, dan dependensi lainnya.
- e. *Hyperparameter*, dilakukan untuk mengoptimalkan kinerja model. Dengan mengatur hyperparameter secara optimal, pengembang dapat mencari kombinasi nilai yang memberikan hasil terbaik.
- f. Melatih Model, merupakan proses melatih model dengan pengaturan *hyperparameter*-nya. Dalam tahap ini, model belajar mengenali pola dan relasi dalam data yang diberikan untuk meningkatkan kemampuannya dalam melakukan tugas yang diminta.
- g. Menguji Model, tahap ini dilakukan untuk memastikan pencapaian tujuan penelitian, yaitu mendeteksi emosi melalui proses pelabelan ekspresi yang menunjukkan tingkat antusiasme siswa.

## **1.7 Sistematika Penulisan**

Bagian ini berisi panduan mengenai urutan dan sistematika penulisan yang digunakan. Berikut adalah sistematika penulisan yang diterapkan:

### **BAB I PENDAHULUAN**

Berisi latar belakang masalah yang menjadi dasar penelitian pentingnya deteksi emosi siswa dan solusi yang diupayakan untuk mengatasi permasalahan tersebut. Berdasarkan latar belakang yang ada, kemudian dibuat rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Berisi teori-teori dan konsep yang berkaitan dengan penelitian mengenai deteksi ekspresi, pembelajaran mesin, dan algoritma YOLO. Teori-teori dan konsep bersumber dari jurnal, buku, dan artikel.

### **BAB III METODOLOGI PENELITIAN**

Berisi tahapan dan kebutuhan penelitian yang meliputi pengumpulan data serta perancangan sistem menggunakan diagram alir.

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

Berisi hasil dari penelitian dan pembahasan setiap proses yang ada serta pengujian kinerja sistem.

### **BAB V KESIMPULAN DAN SARAN**

Berisi kesimpulan dari hasil penelitian dan saran yang dapat diterapkan untuk penelitian selanjutnya.

## BAB II

### LANDASAN TEORI

#### 2.1 Deteksi Wajah

Deteksi wajah merupakan sebuah subdomain dari deteksi objek, dengan beragam algoritma yang telah ditingkatkan dari metode-metode deteksi objek (Adhitama, 2022). Sistem identifikasi wajah bekerja dengan prinsip yang serupa dengan teknologi identifikasi biometrik lainnya, seperti pengenalan telapak tangan dan sidik jari (Azizi, 2021). Langkah awal dalam identifikasi wajah adalah melakukan deteksi wajah, yang mencakup pengenalan ekspresi wajah. Tujuan dari deteksi wajah adalah untuk memastikan keberadaan wajah dalam sebuah gambar, serta menentukan lokasi dan ukuran masing-masing wajah dalam gambar tersebut (Ming-Hsuan Yang et al., 2002). Pada proses deteksi wajah, terdapat beberapa kendala yang perlu dihadapi, termasuk posisi wajah, ukuran wajah, ekspresi wajah, wajah yang tertutup objek lain, dan kondisi pencahayaan (Yang et al., 2015). Beberapa metode yang sering digunakan dalam deteksi wajah meliputi *knowledge-based methods*, *feature invariant approaches*, *template matching methods*, dan *appearance-based methods* (Ming-Hsuan Yang et al., 2002).

Dalam penelitian yang dilakukan oleh Azhari dan Fitriyani (2020), menunjukkan hasil bahwa penggunaan *Deep Learning* dengan metode *Convolutional Neural Network* (CNN) untuk mendeteksi emosi berdasarkan ekspresi wajah secara *real time* menghasilkan kinerja yang memuaskan. Namun, di sisi lain, metode ini membutuhkan sumber daya *hardware* yang cukup besar untuk melakukan prediksi nilai emosi. *Dataset* yang digunakan dalam penelitian ini diambil dari kaggle FER 2013. Hasil dari kombinasi metode CNN, Viola-Jones, dan Haar Cascade menunjukkan tingkat akurasi yang cukup baik, dengan rentang antara 67-83% (I. Azhari, 2021).

Omar, dkk. (2021) menyimpulkan bahwa dalam penelitiannya, model *Convolutional Neural Network* (CNN) dilatih menggunakan library tensorflow-gpu versi 2.3.0. *Dataset* yang digunakan untuk pelatihan adalah *dataset* FER-2013 yang diperoleh dari kaggle. Hasil yang diperoleh menunjukkan bahwa model berhasil melakukan klasifikasi ekspresi wajah manusia, dengan tingkat akurasi sebesar 73,56%. Pendekatan menggunakan algoritma Haar Cascade juga berhasil menghasilkan akurasi yang sama. Selain itu, sistem ini juga mampu mengenali berbagai ekspresi wajah pada gambar tangkapan layar dari *virtual meeting* Zoom. Analisis

kurva *training* dan validasi menunjukkan bahwa model CNN dalam penelitian ini tidak mengalami *overfitting* (Badrulhisham et al., 2021).

Seandrio, dkk. (2021) menyimpulkan bahwa dalam penelitian mereka, jumlah lapisan pada lapisan konvolusi tidak menjamin hasil akurasi yang baik atau buruk. Penggunaan arsitektur *Convolutional Neural Network* dengan model *Visual Group Geometry* (VGG) yang memiliki 16 lapisan terbukti efektif dalam mengklasifikasikan data emosi wajah. Hal ini didukung oleh hasil pengujian *confusion matrix* yang menunjukkan rata-rata akurasi sebesar 70,47%, presisi sebesar 70,86%, dan recall sebesar 69,57%. Pengujian dilakukan menggunakan data uji yang terdiri dari 3589 *dataset*, dengan komposisi 491 data marah, 55 data jijik, 528 data takut, 879 data senang, 594 data sedih, 416 data kaget, dan 626 data biasa. Hasil klasifikasi yang kurang tepat dalam penelitian ini dipengaruhi oleh faktor-faktor tertentu, seperti buram pada kamera dan kemiripan antara emosi-emosi tersebut (Seandrio et al., 2021).

Majeed, dkk. (2021) melakukan penelitian dalam pengenalan wajah menggunakan algoritma YOLOv5 di bawah kondisi lingkungan yang kompleks. Dalam konteks ini, pengenalan wajah dalam *real time* memegang peranan kunci dalam sejumlah aplikasi, terutama dalam sistem *surveilans*. Pengenalan beberapa wajah dalam situasi *real time* yang penuh tantangan, seperti efek penyalauan dan kondisi lingkungan yang berubah-ubah, merupakan masalah yang perlu dicari solusinya. Penelitian ini melibatkan pelatihan YOLOv5 dari awal dan pengujian terhadap *dataset* FDDB dan *dataset* yang telah disesuaikan dengan aliran video *real time*. Hasil eksperimen menunjukkan tingkat akurasi mencapai 87% pada *dataset* FDDB, sementara *dataset* yang disesuaikan mendapat akurasi sebesar 94%. Penelitian ini juga melakukan analisis perbandingan hasil dengan versi YOLOv5 sebelumnya (YOLOv3 dan YOLOv4) untuk menyoroti perkembangan yang dicapai oleh algoritma yang diusulkan. Selain eksperimen di lingkungan yang terkendali, algoritma ini juga diuji dalam situasi *real time*, di mana berhasil mendeteksi wajah dengan akurasi yang sangat baik (Majeed et al., 2021).

## 2.2 Emosi

Emosi manusia merujuk pada sensasi atau reaksi psikis yang timbul di individu sebagai respons terhadap stimulus, baik yang bersumber dari internal atau eksternal (Anggraini et al., 2019). Emosi seseorang dapat dikenali melalui pengamatan ekspresi mikro mereka, yang mana ekspresi mikro tersebut adalah sesuatu yang berlaku secara universal bagi seluruh manusia (Ekman, 2016). Ekspresi wajah atau mimik muka adalah jenis komunikasi nonverbal yang memungkinkan individu untuk menyampaikan emosi mereka kepada orang lain yang

mengamati ekspresi tersebut (Yin et al., 2018). Ekspresi wajah manusia biasanya diklasifikasikan ke dalam tujuh kategori, yaitu marah, jijik, takut, bahagia, netral, sedih, dan terkejut (Lundqvist, D., Flykt, A., & Öhman, A. (1998). *The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.*, n.d.). Jenis ekspresi dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tujuh Ekspresi Dasar Manusia

Sumber: <https://www.kaggle.com/datasets/msambare/fer2013/data>

FER atau *Facial Expression Recognition* (Pengenalan Ekspresi Wajah) merupakan bidang dalam pengenalan pola dan pengolahan citra yang berfokus pada pengenalan dan analisis ekspresi wajah manusia. Tujuan utama FER adalah untuk mengidentifikasi ekspresi emosi seseorang berdasarkan ekspresi wajah yang ditangkap melalui gambar atau video.

Sistem Pengenalan Ekspresi Wajah (FER) dapat dibagi menjadi dua kategori utama berdasarkan representasi fitur, yaitu gambar statis dan urutan dinamis (D. H. Kim et al., 2019). Selain itu, FER terdiri dari dua pendekatan yang berbeda. Dalam pendekatan tradisional, FER terdiri dari tiga langkah, yaitu deteksi komponen wajah, ekstraksi fitur, dan klasifikasi ekspresi. Secara umum, terdapat dua jenis metode ekstraksi fitur, yaitu parameterisasi berbasis geometri dan berbasis penampilan (Uddin et al., 2017). Dengan kata lain, ekstraksi fitur dan klasifikasi dilakukan secara simultan (Luh et al., 2019).

### 2.3 Definisi Antusias dan Tidak Antusias

Antusiasme adalah ketertarikan yang kuat terhadap sesuatu yang tercermin dalam tingginya tingkat kebahagiaan dan semangat yang membara (Ohoirat, 2020). Tanda antusiasme seseorang dapat tercermin melalui ekspresi wajah mereka, yaitu ketika mereka tersenyum dengan semangat yang ditunjukkan melalui pergerakan otot-otot pipi yang mengangkat sudut bibir, menciptakan ekspresi senyuman. Ekspresi ini merupakan tanda positif yang menunjukkan bahwa seseorang merasakan kebahagiaan atau kesenangan terhadap sesuatu (Adhitama, 2022).

Ketidakantusiasan adalah suatu bentuk emosi negatif yang mencerminkan sikap kurang peduli atau kurang berminat. Biasanya, emosi tidak antusias ini muncul ketika seseorang kehilangan minat untuk berpartisipasi dalam aktivitas sosial. Tanda-tandanya dapat dilihat melalui ekspresi wajah, seperti sisi alis bagian dalam menyatu dan condong kebawah, bibir menyempit, dan pandangan mata yang acuh (Adhitama, 2022).

Hasil dari studi yang dilakukan oleh Ohoirat (2020) menyajikan sejumlah indikator yang dapat digunakan untuk mengidentifikasi tingkat antusiasme, serta kurangnya antusiasme. Indikator-indikator tersebut meliputi gestur wajah dan perilaku nonverbal. indikator penting yang diidentifikasi dalam studi ini adalah kontak mata, kedipan mata, gestur pada pipi, gestur pada bibir, gestur pada alis mata, dan gestur pada dagu. Semua indikator ini dapat dilihat dengan jelas pada Tabel 2.1, yang menyajikan data secara komprehensif.

Tabel 2.1 Indikator Antusias dan Tidak Antusias

<b>Indikator</b>	<b>Pernyataan</b>	
Kontak mata	Antusias	Pandangan mata yang tertuju ke layar komputer, laptop, atau gawai.
	Tidak Antusias	Pandangan mata yang tidak terfokus pada layer computer, laptop, atau gawai.
Kedipan mata	Antusias	Kedipan mata yang berulang dengan pola dan frekuensi yang teratur.
	Tidak Antusias	Mengedipkan mata dengan pola yang tidak konsisten terlalu sering.
Gestur pada pipi	Antusias	Adanya variasi pergerakan otot pada pipi dari ujung mulut hingga ke arah telinga secara simultan.
	Tidak Antusias	Tidak adanya variasi pergerakan otot pada pipi dari



		ujung mulut hingga ke arah telinga secara simultan.
Gestur pada bibir	Antusias	Perubahan ukuran pupil mata yang mengalami perluasan.
	Tidak Antusias	Perubahan ukuran pupil mata yang mengalami penyempitan atau tidak mengalami perubahan.
Gestur pada alis mata	Antusias	Alis mata bagian dalam yang santai dan perlahan mengarah ke atas.
	Tidak Antusias	Alis mata bagian dalam yang kaku dan tidak mengarah ke atas.
Gestur pada dagu	Antusias	Terjadi pelebaran pada bagian dagu secara perlahan.
	Tidak Antusias	Tidak terjadi pelebaran pada bagian dagu secara perlahan atau malah mengalami pengecilan.

## 2.4 CNN

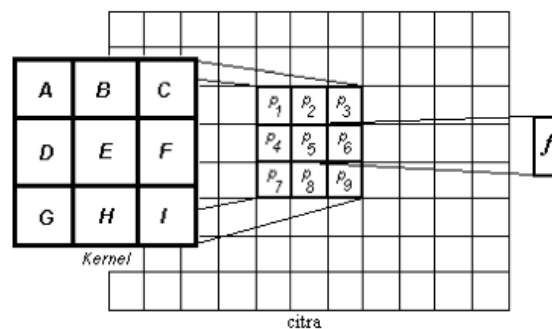
*Convolutional Neural Networks* (CNN) adalah sebuah metode yang berkembang dari multilayer perceptron (MLP) dan dirancang khusus untuk memproses data berdimensi dua. CNN adalah salah satu jenis jaringan saraf yang dalam, dan sering digunakan dalam konteks pengolahan citra karena tingkat kedalaman yang tinggi yang dimilikinya serta penerapannya yang luas pada berbagai tugas pengolahan citra (Azizi, 2021). Pada *Convolutional Neural Network* (CNN), terdapat dua langkah kunci, yakni tahap pembelajaran fitur (*feature learning*) dan tahap klasifikasi. Tahap pembelajaran fitur terdiri dari lapisan konvolusi, fungsi aktivasi ReLU, dan lapisan penggabungan (*pooling*), sementara tahap klasifikasi terdiri dari penyatuan (*flatten*), lapisan terhubung penuh (*fully-connected layer*), dan tahap prediksi. Prinsip kerjanya

mirip dengan Multilayer Perceptron (MLP), yang membedakan CNN adalah representasinya dalam bentuk dua dimensi (Yusuf et al., 2019).

### 2.4.1 Konvolusi

Konvolusi merupakan operasi fundamental dalam *Convolutional Neural Network* (CNN) yang melibatkan pengaplikasian kernel pada output dari layer sebelumnya, yang dikenal sebagai Convolution layer. Proses konvolusi melibatkan pergeseran kernel konvolusi satu piksel demi satu piksel pada citra input, di mana setiap pergeseran menghasilkan perhitungan piksel keluaran  $f(i, j)$ , yang kemudian dicatat dalam sebuah matriks baru seperti persamaan (2.1) dan gambar yang dapat dilihat pada Gambar 2.2 (Novita Wardhani et al., n.d.). Hasil konvolusi ini memiliki tujuan untuk mengekstraksi fitur dari citra input dengan menghasilkan transformasi linier sesuai dengan informasi spasial dalam data. Dalam CNN, bobot pada lapisan konvolusi menspesifikasikan kernel yang digunakan, dan kernel ini dapat dilatih berdasarkan input untuk memahami dan mengekstraksi fitur-fitur penting dalam citra (Hanafi et al., n.d.).

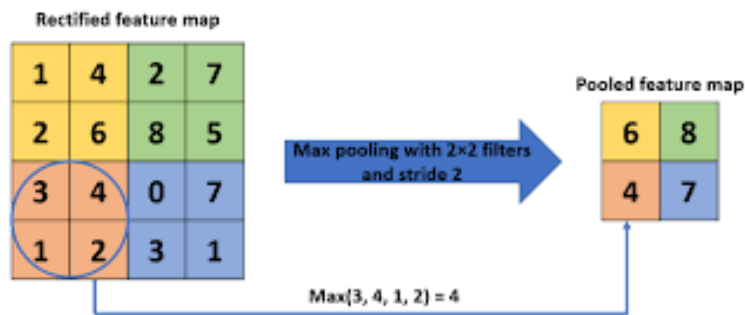
$$f(i, j) = A.P1 + B.P2 + C.P3 + D.P4 + E.P5 + F.P6 + G.P7 + H.P8 + I.P9 \quad (2.1)$$



Gambar 2.2 Ilustrasi Operasi Konvolusi

### 2.4.2 Max Pooling

*Layer Pooling* adalah metode yang digunakan untuk mengurangi dimensi matriks dengan memanfaatkan sebuah filter yang memiliki ukuran dan nilai tertentu (Hakim et al., 2019). Filter tersebut akan bergerak melintasi seluruh area *feature map*. Contohnya, apabila kita menggunakan *pool* terbesar dengan kernel *pool* berukuran 2x2, maka pada setiap perpindahan filter, nilai tertinggi dalam daerah piksel 2x2 akan dipilih, sebagaimana yang diperlihatkan dalam gambar Gambar 2.3. Setelah mengaplikasikan lapisan ini, baik ukuran maupun jumlah parameter akan berkurang, yang pada gilirannya akan meningkatkan efisiensi perhitungan.



Gambar 2.3 Operasi Max Pooling

Sumber: (Gholamalinezhad et al., 2020)

### 2.4.3 SiLU

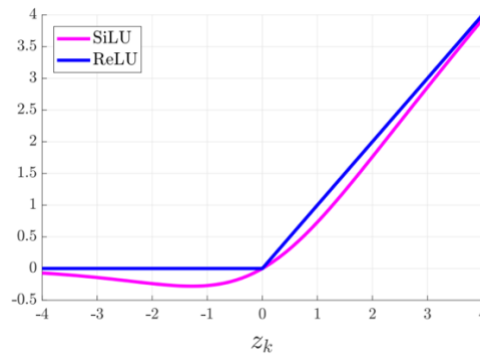
Fungsi SiLU merupakan variasi nonlinear lain dari fungsi aktivasi ReLU. Perhitungannya dilakukan dengan mengalikan inputnya dengan fungsi sigmoid, sehingga menyerupai varian ReLU yang bersifat kontinu dan "undershooting" (Job et al., 2022). Secara matematis, fungsi ini dapat dijelaskan dengan persamaan (2.2) sebagai berikut:

$$R_5(z) = \begin{cases} z, & \text{for } z > 0, \\ z\sigma(z), & \text{for } z \leq 0, \end{cases} \quad (2.2)$$

Di mana  $z$  merupakan input dan  $\sigma(z)$  adalah fungsi sigmoid seperti pada persamaan (2.3) yang didefinisikan sebagai berikut:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.3)$$

Para peneliti sebelumnya menyatakan bahwa fungsi SiLU lebih unggul daripada fungsi ReLU, seperti yang terlihat pada Gambar 2.4 (Nwankpa et al., 2018).



Gambar 2.4 Grafik Perbandingan SiLU dan ReLU

Sumber: (Elfwing et al., 2017)

Dalam penelitian sebelumnya oleh Wang, dkk. 2022, ditemukan bahwa YOLOv7-tiny-SiLU memiliki kecepatan 127 fps lebih tinggi dan tingkat akurasi yang 10.7% lebih tinggi pada skor AP jika dibandingkan dengan YOLOv5-N (r6.1) (Wang et al., 2022).

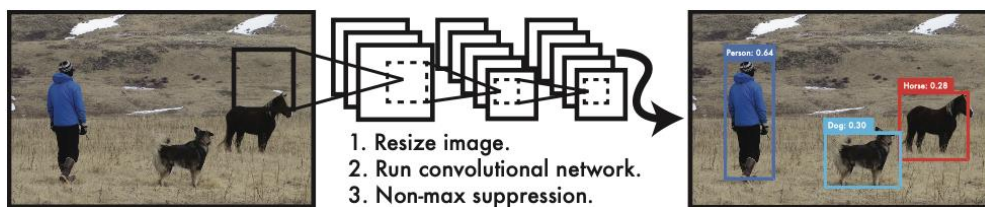
## 2.5 YOLO

YOLO (*You Only Look Once*) adalah metode identifikasi objek *real time* yang dirancang untuk mengidentifikasi objek dengan cepat dan akurat dalam gambar atau video. Metode ini memproses seluruh gambar dengan satu *convolutional neural network* (CNN) dan menghasilkan sejumlah kotak pembatas di sekitar setiap objek yang terdeteksi, bersama dengan label kelas dan *confidence* yang sesuai dengan setiap kotak (Redmon et al., 2015).

YOLO memiliki tiga langkah utama dalam melakukan deteksi objek. Tiga langkah yang dilakukan oleh YOLO adalah sebagai berikut:

- Mengubah ukuran gambar masukan menjadi  $448 \times 448$ ;
- Menjalankan satu jaringan konvolusi pada gambar;
- Menetapkan ambang batas pada deteksi yang dihasilkan oleh nilai *confidence* model.

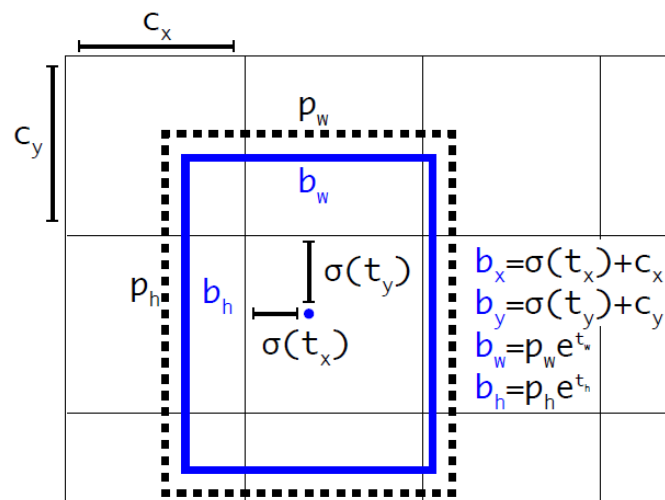
Secara visual, tiga langkah tersebut dapat dilihat pada Gambar 2.5.



Gambar 2.5 Sistem Deteksi YOLO

Sumber: (Redmon et al., 2015)

Dalam tahap awal, YOLO melakukan deteksi objek dengan menggunakan pendekatan *unified detection* yang mengintegrasikan komponen deteksi objek ke dalam satu jaringan saraf tunggal. Desain YOLO memungkinkan *end-to-end training* dan *real time speed*, sambil tetap menjaga tingkat presisi yang tinggi. Sistem dalam YOLO membagi gambar masukan ke dalam *grid* berukuran  $S \times S$ . Jika pusat sebuah objek berada dalam sel *grid*, sel *grid* tersebut bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel *grid* memprediksi bounding box B dan nilai *confidence* untuk setiap kotak. Nilai *confidence* mengindikasikan sejauh mana model yakin bahwa dalam kotak pembatas tersebut berisi objek dan seberapa akurat kotak tersebut menurut prediksinya. Gambar 2.6 menjelaskan bahwa, setiap *bounding box* memiliki lima parameter prediksi, yaitu  $x$ ,  $y$ ,  $w$ ,  $h$  dan *confidence*. Koordinat  $(x, y)$  mengacu pada pusat kotak pembatas relatif terhadap gambar, dan *confidence* menggambarkan *intersection over Union* (IoU) antara *predicted box* dan *ground-truth box*. Setiap sel *grid* juga memprediksi probabilitas kelas C. Probabilitas kelas ini diprediksi berdasarkan sel *grid* yang berisi objek, sehingga setiap sel *grid* hanya memiliki probabilitas untuk satu kelas tanpa memperhatikan jumlah kotak pembatas B. Saat melakukan deteksi, probabilitas kelas dikalikan dengan nilai *confidence* sesuai dengan persamaan (2.4) (Redmon et al., 2015).



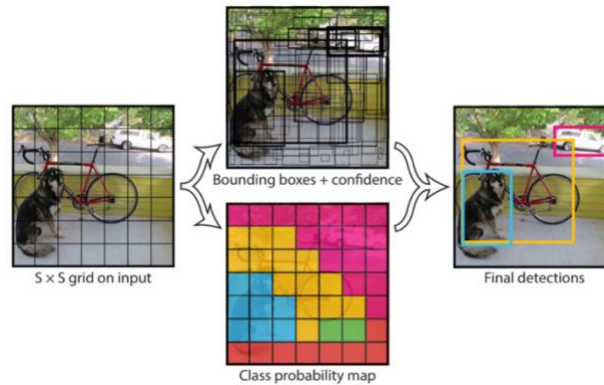
Gambar 2.6 Bounding Box pada YOLO

Sumber: (Redmon et al., 2018)

$$Pr(Class_i|Object).Pr(Object).IOU_{pred}^{truth} = Pr(Class_i).IOU_{pred}^{truth} \quad (2.4)$$

Dari persamaan tersebut, diperoleh skor *confidence* yang berhubungan dengan kelas tertentu. Skor ini mengindikasikan probabilitas kelas muncul dalam kotak dan tingkat

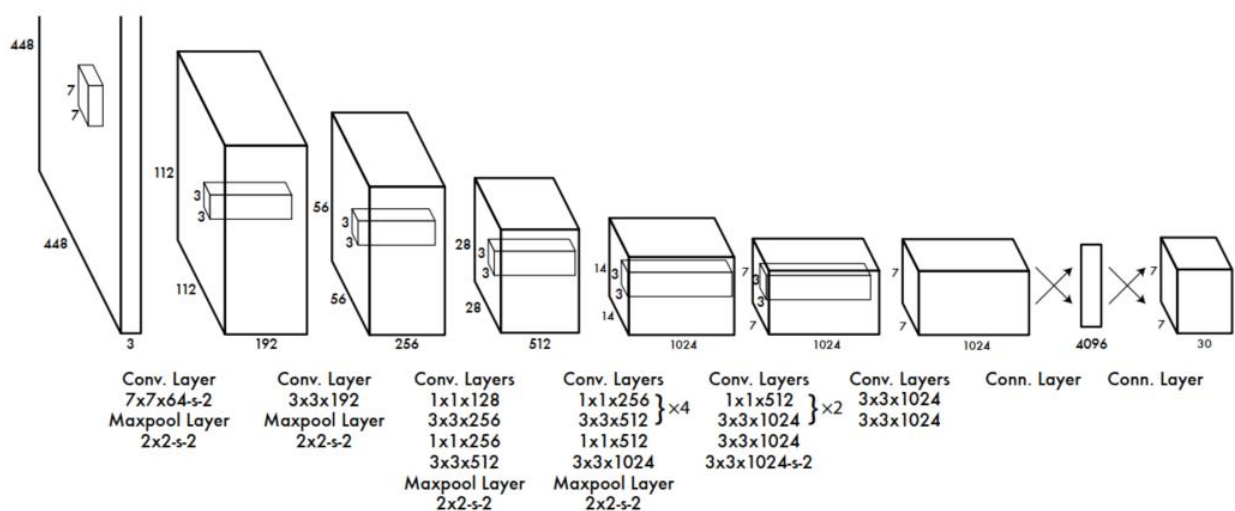
ketepatan kotak yang diprediksi terhadap objek. Sebagaimana diperlihatkan dalam ilustrasi pada Gambar 2.7, YOLO mendeteksi masalah ini sebagai regresi. YOLO mengurai gambar menjadi grid dan secara bersamaan memprediksi *bounding box*, skor *confidence* pada *bounding box*, dan kelas probabilitas (Hanafi et al., n.d.).



Gambar 2.7 Proses Deteksi Model Yolo

Sumber: (Redmon et al., 2015)

YOLO memiliki jaringan deteksi yang terdiri dari 24 lapisan konvolusi, diikuti oleh 2 *fully connected layer*. Lapisan konvolusi  $1 \times 1$  yang bergantian digunakan untuk mengurangi *features space* dari lapisan sebelumnya. Sebelum digunakan untuk deteksi, YOLO melatih lapisan konvolusi pada tugas klasifikasi ImageNet dengan resolusi setengah ( $224 \times 224$  gambar masukan), dan kemudian meningkatkan resolusi menjadi dua kali lipat seperti pada Gambar 2.8. Keunggulan YOLO dibandingkan dengan model deteksi objek lainnya terletak pada ukuran yang kecil dan kemampuan komputasi yang cepat (Jiang et al., 2022).



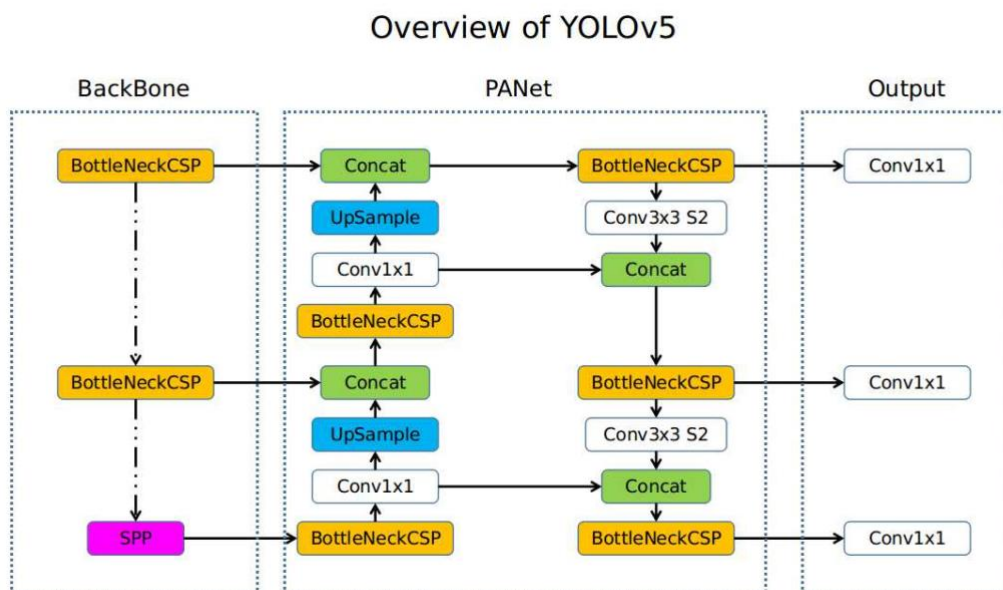
Gambar 2.8 Arsitektur YOLO

Sumber: (Redmon et al., 2015)

### 2.5.1 YOLO V8

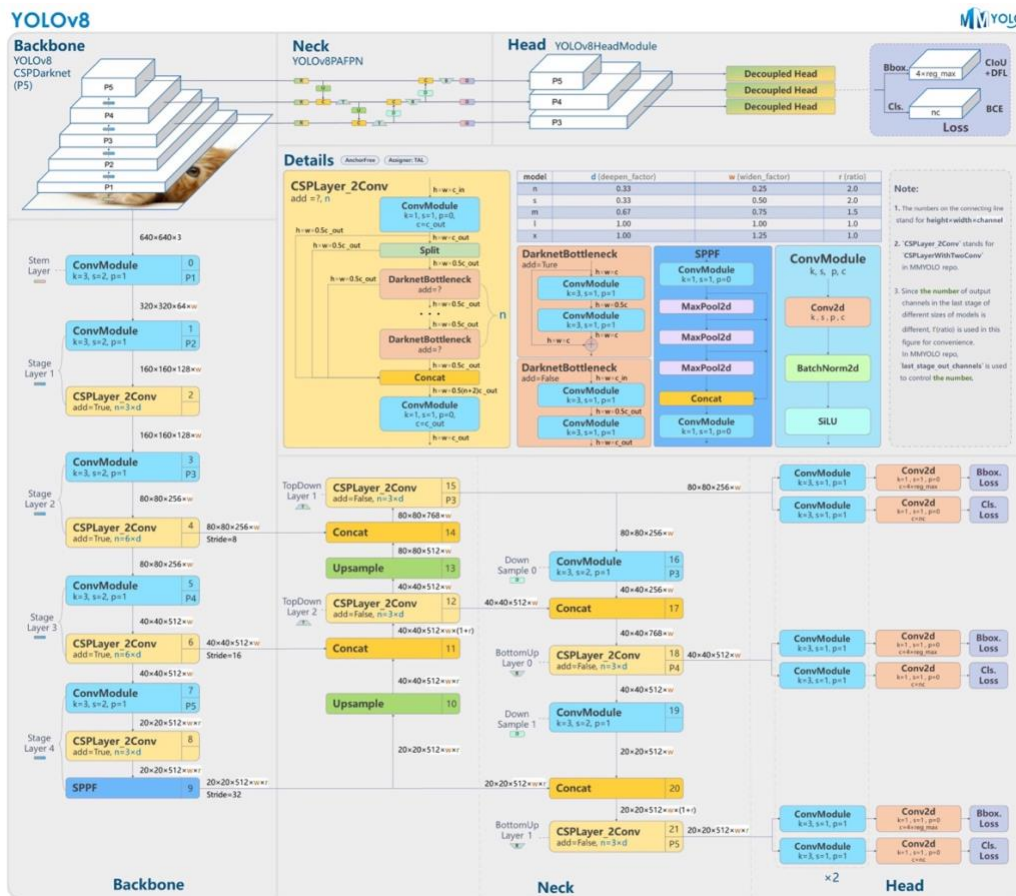
YOLOv8 dirilis oleh Ultralytics pada Januari 2023, yang merupakan kelanjutan dari karya mereka setelah merilis YOLOv5 pada tahun 2020. YOLOv8 menyediakan lima versi yang berbeda ukuran: YOLOv8n (*nano*), YOLOv8s (*small*), YOLOv8m (*medium*), YOLOv8l (*large*), dan YOLOv8x (*extra large*). Meskipun belum terdapat *paper* resmi yang dirilis untuk YOLOv8, model ini sudah dapat digunakan untuk pelatihan dan pengujian oleh publik (Terven et al., 2023).

Model ini mempertahankan struktur utama yang mirip dengan YOLOv5, dengan beberapa perubahan pada CSPLayer, yang kini disebut modul C2f. Modul C2f, yang merupakan singkatan dari *cross-stage partial bottleneck with two convolutions*, bertujuan untuk menggabungkan fitur-fitur tingkat tinggi dengan informasi kontekstual guna meningkatkan akurasi deteksi. YOLOv8 menggunakan model bebas anchor dengan kepala terpisah untuk memproses secara independen tugas objek, klasifikasi, dan regresi. Desain ini memungkinkan setiap cabang untuk fokus pada tugasnya dan meningkatkan akurasi keseluruhan model. Pada lapisan output YOLOv8, mereka menggunakan fungsi sigmoid sebagai fungsi aktivasi untuk *objectness scor*, yang mencerminkan probabilitas bahwa *bounding box* mengandung objek (Terven et al., 2023). Gambar arsitektur YOLOv5 dan YOLOv8 dapat dilihat pada Gambar 2.9 dan Gambar 2.10. Gambar 2.10 bukan berasal dari situs resmi, melainkan dari pengguna GitHub bernama RangeKing.



Gambar 2.9 Arsitektur YOLOv5

Sumber: (Jocher, 2020)



Gambar 2.10 Arsitektur YOLOv8

Sumber: (M. Contributors, 2023)

YOLOv8 dapat dijalankan melalui antarmuka baris perintah (CLI), atau juga dapat diinstal sebagai paket PIP. Selain itu, model ini dilengkapi dengan berbagai integrasi untuk pelabelan, pelatihan, dan implementasi. Beberapa perubahan telah dilakukan untuk meningkatkan akurasi model ini dengan meningkatkan augmentasi selama proses pelatihan. YOLOv8 melakukan augmentasi gambar secara online selama pelatihan. Pada setiap *epoch*, model melihat variasi gambar yang sedikit berbeda dari gambar yang telah diberikan. YOLOv8 menggunakan augmentasi mosaik selama pelatihan. Namun, karena ditemukan bahwa augmentasi ini dapat mengurangi presisi jika digunakan selama seluruh proses pelatihan, maka augmentasi ini dinonaktifkan untuk sepuluh *epoch* terakhir. Augmentasi mosaik adalah teknik di mana 4 gambar pelatihan disatukan untuk membentuk gambar baru. Teknik augmentasi ini membantu model melatih dan belajar dengan lebih efisien (W. Hao et al., 2020). YOLOv8 lebih efisien daripada versi sebelumnya karena menggunakan peta fitur yang lebih besar dan jaringan konvolusi yang lebih efisien. Hasil evaluasi pada kumpulan data MS COCO test-dev



2017 menunjukkan bahwa YOLOv8x mencapai AP sebesar 53,9% dengan ukuran gambar 640 piksel (dibandingkan dengan 50,7% dari YOLOv5 pada ukuran masukan yang sama) dengan kecepatan 280 FPS pada perangkat NVIDIA A100 dan TensorRT (Terven et al., 2023).

### 2.5.2 IoU

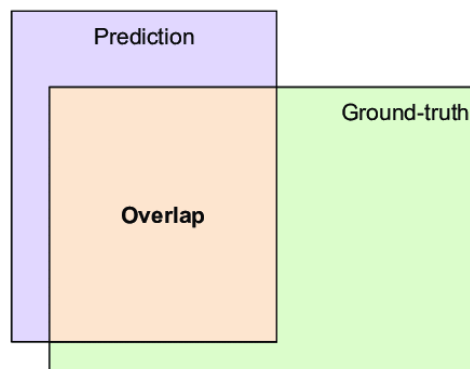
IoU adalah rasio antara hasil irisan dan gabungan antara *prediction frame* dan *ground truth* (Zhong et al., 2023) . Karena irisan adalah bagian dari gabungan, nilai IoU biasanya berada dalam rentang [0, 1]. Jika IoU sama dengan 1, ini menunjukkan bahwa irisan *ground truth* dan hasil segmentasi model tumpang tindih dengan sempurna, sedangkan jika IoU adalah 0, itu berarti tidak ada tumpang tindih sama sekali. Secara matematis, IoU dapat ditulis seperti persamaan (2.5) dan digambarkan seperti Gambar 2.11.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (2.5)$$

Dalam kasus klasifikasi biner, persamaan (2.5) dapat ditulis seperti persamaan (2.6):

$$IoU = \frac{TP}{TP + FN + FP} = \frac{irisan}{gabungan} \quad (2.6)$$

- TP = *True Positive*, yaitu total data kelas *true* yang berhasil diprediksi *true*.
- FN = *False Negative*, yaitu total data kelas *true*, tetapi diprediksi *false*.
- FP = *False Positive*, yaitu total data kelas *false*, tetapi diprediksi *true*.



Gambar 2.11 *Intersection-over-Union* (IoU)

Sumber: (Fernandes Junior et al., 2021)

### 2.5.3 Precision

Nilai presisi diperoleh dengan membagi jumlah keseluruhan hasil deteksi positif yang diklasifikasikan secara benar dengan jumlah keseluruhan hasil deteksi yang diprediksi sebagai positif, seperti yang dinyatakan dalam persamaan (2.7). Presisi yang tinggi mengindikasikan bahwa hasil deteksi yang benar-benar positif sesuai dengan prediksi positif (dengan sedikit *false positive*) (Hasan et al., 2023).

$$Precision = \frac{TP}{TP + FP} \quad (2.7)$$

### 2.5.4 mAP

mAP atau *mean Average Precision* (rata-rata Presisi) adalah metrik evaluasi yang umum digunakan dalam tugas-tugas seperti deteksi objek dan pengenalan wajah. mAP mengukur sejauh mana sebuah sistem dapat mengenali objek atau entitas tertentu dengan tingkat keakuratan yang baik.

Metrik ini menghitung rata-rata dari nilai Presisi yang dihitung pada setiap kelas atau kategori objek yang diuji. Setiap nilai Presisi mengukur sejauh mana sistem dapat mengenali objek dalam kategori tersebut dengan benar. Kemudian, rata-rata semua nilai Presisi ini memberikan gambaran keseluruhan tentang seberapa baik sistem tersebut dalam mengenali objek dari berbagai kategori. Semakin tinggi nilai mAP, semakin baik sistem dalam mengenali objek dengan benar. mAP 50 atau dapat ditulis dengan [mAP@\[.5\]](#), berarti bahwa model dapat memberikan presisi tinggi pada tingkat *confidence* 50%. Penulisan ini digunakan untuk menunjukkan bahwa mAP dihitung pada *threshold* 0.5. Penulisan [mAP@\[.5\]](#) adalah cara singkat untuk mengekspresikan ambang batas yang digunakan dalam perhitungan mAP. Angka dalam kurung siku [ ] menunjukkan nilai *threshold* IoU yang digunakan. *Threshold* 0.5 merujuk pada ambang batas *Intersection over Union* (IoU) yang digunakan untuk menentukan apakah prediksi model dianggap benar atau salah. Jika IoU antara kotak prediksi dan kotak *ground truth* lebih besar dari atau sama dengan 0.5, prediksi tersebut dianggap benar. Perhitungan mAP dapat ditulis seperti persamaan (2.8).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.8)$$

## 2.6 Penelitian Sebelumnya

Banyak peneliti telah menjalankan penelitian mengenai pengenalan ekspresi wajah dengan berbagai pendekatan dan metode yang berbeda. Seperti penelitian yang telah dilakukan oleh Prasetyawan, dkk. (2020), yang bertujuan untuk mengidentifikasi emosi melalui pengenalan ekspresi wajah manusia dan merekam perubahan ekspresi yang terjadi. Metodenya melibatkan klasifikasi terhadap 6 ekspresi dasar wajah manusia dan ekspresi netral dengan menggunakan *Convolutional Neural Network* (CNN) dengan *dataset* yang berasal dari FER-2013. Data hasil pemodelan dievaluasi dengan menggunakan confusion matrix, dan hasil evaluasi menunjukkan tingkat akurasi sebesar 74,17% dan rata-rata presisi sebesar 74,07% (Prasetyawan et al., 2020).

Penelitian Azizi (2021), menguji tiga metode optimasi, yaitu ADAM, SGD, dan RMS, serta mengadakan eksperimen dengan variasi jumlah *epoch* untuk mendeteksi emosi berdasarkan ekspresi. *Dataset* yang digunakan berasal dari FER2013, KDEF, dan CK+. Hasil studi tersebut mengindikasikan bahwa optimasi dengan ADAM memberikan tingkat akurasi paling tinggi dibandingkan dengan metode optimasi lainnya, mencapai akurasi tertinggi sebesar 82% pada *dataset* KDEF (Azizi, 2021).

Penelitian Adhitama (2022), menggunakan metode yang melibatkan algoritma YOLOv5 (*You Only Look Once*) dan kerangka kerja PyTorch untuk *deep learning* dalam mendeteksi ekspresi wajah (*Facial Expression Recognition*). Dalam penelitian ini, dibuat dua kelas *dataset* pelatihan khusus, yaitu "antusias" dan "tidak antusias," yang diambil dari *dataset* FER2013. Penelitian ini menggunakan tiga model YOLOv5, yaitu YOLOv5s, YOLOv5m, dan YOLOv5x. Hasil evaluasi menunjukkan bahwa semua model YOLOv5 mencapai akurasi tertinggi dalam mendeteksi ekspresi antusias sebesar 87%, sedangkan model YOLOv5m memberikan akurasi tertinggi dalam mendeteksi ekspresi tidak antusias sebesar 85% (Adhitama, 2022).

Penelitian Zhong, dkk. (2023), mengidentifikasi ekspresi wajah guru secara *real time* menggunakan model YOLOv5 yang telah ditingkatkan dengan *attention* model yang memperkaya model *Backbone* YOLOv5 dengan *dataset* yang berasal dari RAF-DB. Hasil penelitian menunjukkan bahwa model jaringan yang mengintegrasikan *attention* model berkoordinat setelah modul CBS dalam struktur CSP1\_X mampu mencapai waktu deteksi sekitar 25 ms dengan tingkat akurasi sebesar 77,1%. Ini menunjukkan peningkatan signifikan sebesar 3,5% dibandingkan dengan YOLOv5 yang tidak menerapkan *attention* model (Zhong et al., 2023).

Dalam penelitian terdahulu oleh Gautam, dkk. (2023), fokus penelitian adalah pada Gangguan Spektrum Autisme (ASD). Mereka mencoba mengidentifikasi ASD dengan mengamati ciri-ciri wajah menggunakan model YoloV8 dan *dataset* Kaggle. Hasilnya luar biasa, dengan akurasi klasifikasi mencapai 89,64% dan F1-score 0,89. Temuan ini mendukung pengamatan klinis mengenai perbedaan ciri-ciri wajah antara anak-anak dengan ASD, dan menunjukkan potensi model deep learning dalam skrining ASD.

Berdasarkan uraian di atas, dapat diambil kesimpulan bahwa penelitian sebelumnya telah mencoba berbagai metode dan teknik dalam deteksi dan identifikasi ekspresi wajah. Beberapa penelitian menggunakan *Convolutional Neural Network* (CNN), model YOLOv5, dan YOLOv8, sementara yang lain menguji metode optimasi seperti ADAM, SGD, dan RMS. Hasil akhirnya mencakup akurasi yang bervariasi, dengan beberapa penelitian mencapai tingkat akurasi yang signifikan. Penelitian-penelitian ini memberikan dasar yang kuat untuk melanjutkan penelitian dalam bidang deteksi ekspresi wajah, serta menggarisbawahi pentingnya teknologi deep learning dalam peningkatan akurasi dalam tugas ini. Berikut ringkasan penelitian terdahulu yang disajikan dalam Tabel 2.2.

Tabel 2.2 Penelitian mengenai deteksi dan pengenalan ekspresi emosi wajah.

No	Peneliti dan Tahun	Tujuan Penelitian	Metode Penelitian	Sumber Data	Hasil Akhir
1.	Daru Prasetyawan dan Shofwatul 'Uyun (2020)	Mengidentifikasi emosi melalui pengenalan ekspresi wajah manusia dan merekam perubahan ekspresi yang terjadi.	Klasifikasi terhadap 6 ekspresi dasar wajah manusia dan ekspresi netral dengan menggunakan <i>Convolutional Neural Network</i> (CNN).	FER2013	Akurasi 74,17% dan rata-rata presisi 74,07%.
2.	Faza Nur Azizi	Menguji tiga metode optimasi (ADAM, SGD, dan RMS) untuk	Metode optimasi, variasi jumlah <i>epoch</i> .	<i>Dataset</i> dari FER2013,	Akurasi tertinggi 82% pada <i>dataset</i> KDEF dengan

		mendeteksi emosi berdasarkan ekspresi.		KDEF, dan CK+.	optimasi ADAM.
3.	Gunawan Adhitama (2022)	Mendeteksi ekspresi wajah ( <i>Facial Expression Recognition</i> ) menggunakan algoritma YOLOv5 (You Only Look Once) dan kerangka kerja PyTorch.	Pembuatan dua kelas <i>dataset</i> pelatihan khusus ("antusias" dan "tidak antusias") dari <i>dataset</i> FER2013.	<i>Dataset</i> dari FER2013.	Model YOLOv5 mencapai akurasi tertinggi 87% untuk ekspresi antusias dan 85% untuk ekspresi tidak antusias.
4.	Hongmei Zhong, Tingting Han, Wei Xia, Yan Tian <sup>1</sup> , dan Libao Wu (2023)	Mengidentifikasi mengidentifikasi ekspresi wajah guru secara <i>real time</i> menggunakan model YOLOv5 yang telah ditingkatkan dengan attention model yang memperkaya model Backbone YOLOv5.	Integrasi <i>attention</i> model berkoordinat setelah modul CBS dalam struktur CSP1_X.	FER2013	Waktu deteksi sekitar 25 ms dengan tingkat akurasi 77,1%, meningkat 3,5% dibandingkan dengan YOLOv5 tanpa <i>attention</i> model
5.	Subash Gautam, Prabin Sharma, Kisan Thapa, Mala Deep	Mengidentifikasi Gangguan Spektrum Autisme (ASD) melalui ciri-ciri wajah menggunakan	Identifikasi ekspresi menggunakan YOLOv8	Kaggle	Akurasi klasifikasi 89,64% dan F1-score 0,89, mendukung pengamatan klinis

	Upadhaya, Dikshya Thapa, Salik Ram Khanal, dan Vítor Manuel de Jesus Filipe (2023)	model YoloV8 dan <i>dataset</i> Kaggle.			mengenai perbedaan ciri- ciri wajah anak-anak dengan ASD.
--	--	--	--	--	---

## BAB III METODOLOGI PENELITIAN

### 3.1 Data

Pengembangan model YOLOv8 dalam penelitian ini memanfaatkan *dataset Facial Expression Recognition 2013 (FER-2013)*. *Dataset Facial Expression Recognition 2013 (FER-2013)* diperkenalkan dalam International Conference on Machine Learning (ICML) 2013 dan memuat 35.887 gambar wajah berformat *grayscale* berukuran 48x48 piksel yang mencakup 7 variasi ekspresi emosi yang berbeda (Goodfellow et al., 2015). Setiap gambar dalam *dataset* ini telah dilabeli dan diklasifikasikan ke dalam 7 kategori dengan indeks dari 0 hingga 6 seperti yang tertera di Tabel 3.1.

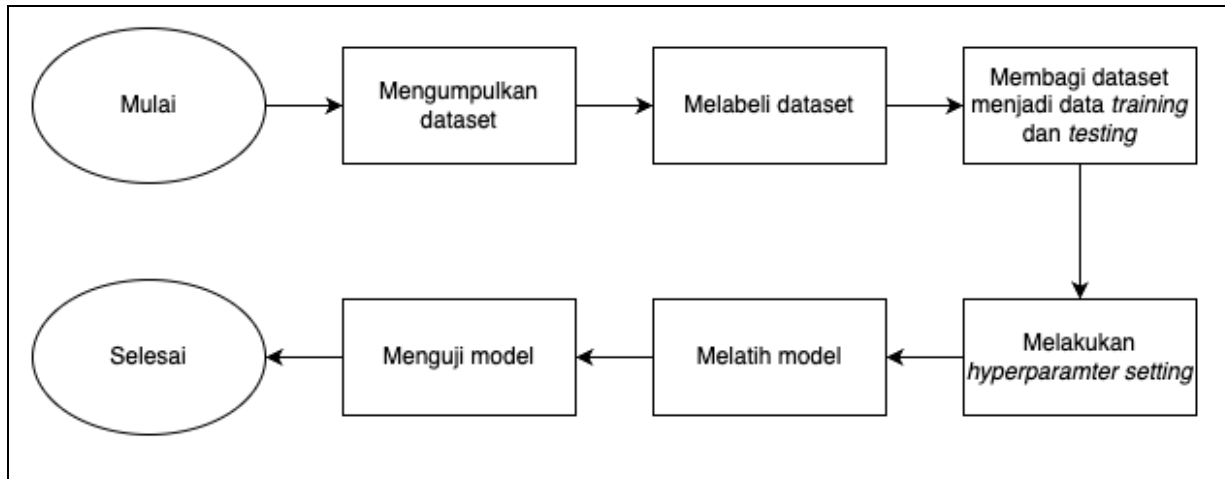
Tabel 3.1 Kelas emosi pada *dataset* FER2013

Label	Jenis Emosi	Jumlah
0	Marah	4593
1	Jijik	547
2	Takut	5121
3	Bahagia	8989
4	Sedih	6077
5	Terkejut	4002
6	Netral	6198

*Dataset* ini awalnya digunakan dalam kompetisi "Challenges in Representation Learning: Facial Expression Recognition Challenge" di platform Kaggle dan dipersiapkan oleh Pierre-Luc Carrier dan Aaron Courville. FER-2013 dibagi menjadi dua subset, dengan 80% data digunakan untuk pelatihan dan 20% untuk pengujian. Data pelatihan terdiri dari 28.709 gambar, sementara data pengujian mencakup 7.178 gambar. Sejumlah variasi emosi termasuk senang dan jijik, dengan jumlah gambar emosi tertinggi sebanyak 8.989 pada emosi senang, dan yang paling sedikit dengan 547 gambar pada emosi jijik. Meskipun *dataset* mencakup gambar dengan berbagai jenis kelamin, informasi tentang perbandingan jumlah antara jenis kelamin tidak disertakan dalam *dataset*. Selain itu, data ini juga tidak mencantumkan informasi tentang ras atau rentang usia subjek dalam gambar.

### 3.2 Perancangan Penelitian

Pada bagian ini akan dijelaskan mengenai perancangan penelitian mencakup gambaran mengenai alur program yang digunakan dalam penelitian ini. Alur tersebut diilustrasikan dalam bentuk flowchart, yang dapat dilihat dalam Gambar 3.1. *Flowchart* tersebut merinci alur perancangan penelitian ini.



Gambar 3.1 *Flowchart* perancangan penelitian

#### 3.2.1 Mengumpulkan *Dataset*

Dalam penelitian ini, penggunaan *dataset* berasal dari *Facial Expression Recognition 2013 (FER-2013)*. *Dataset* ini terdiri dari tujuh kelas dan berisi 28.709 gambar yang siap digunakan untuk proses pelatihan. Ekspresi tersebut dibagi menjadi ekspresi marah, jijik, takut, senang, netral, sedih, dan terkejut. Dari 28.709 gambar yang tersedia, penelitian ini hanya menggunakan 2000 gambar dengan pembagian 10% dari masing-masing ekspresi. Diharapkan 2000 gambar dapat memberikan keseimbangan antara akurasi dan efisiensi komputasi. Model yang dilatih dengan *dataset* yang lebih kecil dapat dilatih lebih cepat dan hemat sumber daya komputasi, tetapi tetap diharapkan mencapai akurasi yang memadai.

#### 3.2.2 Melabeli *Dataset*

Pelabelan atau anotasi merupakan proses penting dalam pengembangan model deteksi objek. Tujuan utamanya adalah untuk memberikan informasi tentang nama kelas yang sesuai dengan objek yang perlu diidentifikasi dan juga tentang posisi yang dinamakan *bounding box* yang mengelilingi objek tersebut. Dalam konteks ini, fokus utama pelabelan adalah untuk menunjukkan letak wajah dan mengelompokkan gambar menjadi antusias dan tidak antusias. Tujuannya adalah untuk memastikan bahwa saat aplikasi berjalan, sistem deteksi akan



berkonsentrasi pada identifikasi wajah, dan tidak akan terganggu atau membuang waktu untuk mendeteksi objek atau area lain dalam citra yang tidak relevan.

Proses pelabelan data antusias dan tidak antusias dilakukan dengan melibatkan lima orang partisipan yang merupakan aktor di bidang Pendidikan. Lima orang partisipan ini terdiri dari dua orang dosen dan tiga orang mahasiswa, yaitu Bapak Budi Sulistiyo Jati, S.Kom., M.Eng., Ibu Rakhma Safrida Kurnia, S.Kom., M.Kom., Mayla Ayuni Sonya, Sonya Ainurohmah, dan Gabriel Imam Andaru. Setiap partisipan bertugas melabeli 2000 gambar yang ditampilkan melalui *platform* Zoom. Peneliti membagikan layar yang menampilkan gambar dari *dataset* satu per satu kepada para partisipan. Ketika mengamati gambar, para partisipan diminta untuk menilai apakah gambar tersebut menunjukkan ekspresi antusias atau tidak antusias. Jika menurut mereka antusias, mereka diminta untuk menuliskan angka "1" pada lembar Google Sheet yang telah disediakan. Sebaliknya, jika mereka menilai gambar tersebut tidak antusias, mereka diminta untuk menuliskan angka "0". Setelah proses pelabelan 2000 gambar dengan angka 1 dan 0 selesai, modus dari setiap gambar dipilih. Jika modus dari gambar tersebut adalah 1, maka gambar tersebut dikategorikan sebagai antusias. Sebaliknya, jika modulusnya 0, maka gambar tersebut dikategorikan sebagai tidak antusias. Setelah proses pelabelan bersama lima partisipan selesai dilakukan, kemudian langkah selanjutnya adalah menyimpan data dalam format CSV dengan nama labels.csv untuk kemudian dilabeli dengan menggunakan kode program.

Data tersebut kemudian digunakan untuk mengubah anotasi gambar pada FER2013 yang semula memiliki label dari 0 sampai dengan 6 menjadi hanya berlabel 0 dan 1. Gambar 3.2 merupakan kode program yang digunakan untuk melabeli gambar.

```
import os

def change_class_in_annotation(annotation_file_path, old_class_id, new_class_id):
    with open(annotation_file_path, 'r') as file:
        lines = file.readlines()

    updated_lines = []
    for line in lines:
        parts = line.strip().split(' ')
        class_id = int(parts[0])
        print("Before: ",parts)

    # if class_id == old_class_id:
        parts[0] = str(new_class_id)
        updated_lines.append(' '.join(parts))
        print("Updated: ",updated_lines)
    with open(annotation_file_path, 'w') as file:
        file.write("\n".join(updated_lines))
        print("File {} has been updated.".format(annotation_file_path))
```

```

# Define the paths and class IDs
csv_file_path = 'label.csv'
annotation_folder = './labels/'
old_class_id = 0 # The old class ID you want to change
# new_class_id = 1 # The new class ID you want to assign

# Read the CSV file and process each row
import csv
with open(csv_file_path, mode='r') as file:
    csv_reader = csv.reader(file)
    for row in csv_reader:
        image_name = row[0] # Assuming the image name is in the first column
        annotation_file_path = os.path.join(annotation_folder, f"{os.path.splitext(image_name)[0]}.txt")
        new_class_id = row[1]
        print("Annotation Path: ", annotation_file_path)
        print("Class ID: ", new_class_id)
        # Check if the annotation file exists
        if os.path.exists(annotation_file_path):
            change_class_in_annotation(annotation_file_path, old_class_id, new_class_id)
        else:
            print(f"Annotation file not found for image: {image_name}")

```

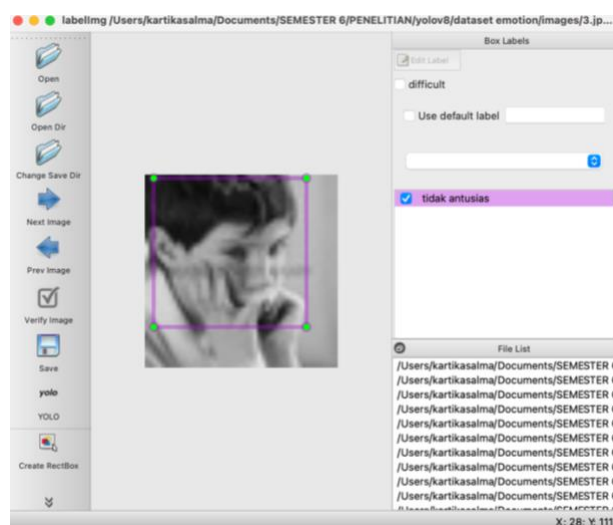
Gambar 3.2 Kode Program Proses Pelabelan

Singkatnya kode program ini bekerja dengan membaca file CSV yang berisi nama gambar dan ID kelas baru untuk setiap gambar. Kemudian, kode program ini mencari file anotasi yang sesuai untuk setiap gambar dan mengubah ID kelas dalam file tersebut. Kode program ini terdiri atas empat langkah. Langkah pertama adalah mengimport modul `os` dan modul `csv`. Mengimport modul `os` untuk operasi sistem file dan modul `csv` untuk membaca file CSV. Kemudian, langkah kedua dimulai pada baris ketiga dengan mendefinisikan fungsi `change_class_in_annotation` yang berfungsi untuk mengubah ID kelas dalam file anotasi. Fungsi ini mengambil tiga argument, yaitu `annotation_file_path` (jalur ke file anotasi), `old_class_id` (ID kelas lama), dan `new_class_id` (ID kelas baru). Fungsi ini pertama-tama membaca isi file anotasi ke dalam daftar baris. Kemudian, fungsi ini mengulangi setiap baris dalam daftar dan membagi baris menjadi daftar bagian. Bagian pertama dari daftar adalah ID kelas. Jika ID kelas sama dengan ID kelas lama, fungsi ini mengubah ID kelas menjadi ID kelas baru dan menambahkan baris yang diperbarui ke daftar baris yang diperbarui. Terakhir, fungsi ini menulis isi daftar baris yang diperbarui ke file anotasi dan mencetak pesan bahwa file telah diperbarui.

Langkah ketiga adalah menentukan *path* dan ID kelas. Langkah ini dimulai dari baris ke-22 yang berisi `csv_file_path` (jalur ke file CSV yang berisi data gambar dan ID kelas baru), `annotation_folder` (jalur ke folder yang berisi file anotasi), dan `old_class_id` (ID kelas lama

yang ingin diubah). Kemudian langkah keempat dimulai pada baris ke-28, dengan nama import csv. Bagian kode ini membaca file CSV yang berisi nama gambar dan ID kelas baru untuk setiap gambar. Kemudian, kode program ini mengulangi setiap baris dalam file CSV dan memproses informasi berikut, `image_name` (nama gambar), `annotation_file_path` (jalur ke file anotasi untuk gambar), `new_class_id` (ID kelas baru untuk gambar). Kode program ini kemudian memeriksa apakah file anotasi untuk gambar tersebut ada. Jika ada, kode program ini memanggil fungsi `change_class_in_annotation` untuk mengubah ID kelas dalam file tersebut. Jika file anotasi tidak ada, kode program ini mencetak pesan bahwa file anotasi tidak ditemukan untuk gambar tersebut. Sedikit catatan, kode program ini perlu dijalankan sebanyak tujuh kali sesuai dengan jumlah kelas pada FER2013 dan terdapat baris yang harus diubah tiap dijalankan, yaitu baris ke-24 (`old_class_id = 0`). Baris ini berisi informasi terkait ID kelas lama yang akan diganti, karena pada *dataset* FER2013 terdapat tujuh kelas dengan anotasi 0 sampai dengan 6, maka baris kode tersebut juga harus disesuaikan nilainya. Pada proses penjalanan pertama `old_class_id` diisi dengan nilai 0, lalu pada penjalanan kedua `old_class_id` diisi dengan nilai 1, hingga pada penjalanan ketujuh `old_class_id` diisi dengan nilai 6.

Setelah proses merubah label dengan kode program selesai dilakukan, langkah selanjutnya adalah melabeli gambar satu per satu yang dilakukan dengan bantuan *tool* LabelImg seperti pada Gambar 3.3. Proses pelabelan dilakukan dengan cara memberi “Bounding Box” sebagai pembatas setiap objek. Hasil pelabelan akan disimpan dalam file .txt dengan format data YOLO.



Gambar 3.3 Proses Pelabelan dengan LabelImg

### 3.2.3 Membagi Dataset

Dalam tahap pengolahan *dataset*, *dataset* awal telah dibagi secara proporsional menjadi dua bagian utama, yakni data *training* dan data *testing*. Pembagian *dataset* dilakukan dengan menerapkan komposisi 90%, 10%. Berdasarkan hasil pelabelan 2000 gambar, diperoleh hasil sejumlah 1013 gambar dilabeli sebagai antusias dan 987 gambar dilabeli sebagai tidak antusias. Pembagian 90% dan 10% ini dilakukan pada masing-masing label agar pembagian bisa lebih merata. Dari masing-masing kategori, diambil 90% untuk data *training*, sehingga diperoleh sebanyak 1790 gambar. Data *training* ini digunakan untuk melatih model, agar model dapat mempelajari pola-pola yang terdapat dalam gambar-gambar tersebut.

Sebaliknya, 10% sisanya (210 gambar) dipisahkan sebagai data *testing*. Pembagian ini juga dilakukan secara proporsional, yaitu 10% dari 1013 gambar antusias dan 10% dari 987 gambar tidak antusias sehingga total data *testing* adalah 210 gambar. Data *testing* ini berperan penting dalam mengevaluasi kinerja model yang telah dilatih, untuk mengukur kemampuan model dalam mengenali ekspresi wajah dengan akurat dan generalisasi yang baik.

Proses pembagian dataset dilakukan dengan menggunakan kode program. Adapun kode program yang digunakan dapat dilihat pada Gambar 3.4 Kode Program Proses Pembagian Dataset

```

from tqdm import tqdm
from pathlib import Path
import random
import os

DATASETS_DIR = Path(".")

IMG_FORMATS = 'bmp', 'dng', 'jpeg', 'jpg', 'mpo', 'png', 'tif', 'tiff', 'webp', 'pfm' # include image suffixes

def img2label_paths(img_paths):
    # Define label paths as a function of image paths
    sa, sb = f'{os.sep}images{os.sep}', f'{os.sep}labels{os.sep}' # /images/, /labels/ substrings
    return [sb.join(x.rsplit(sa, 1)).rsplit('.', 1)[0] + '.txt' for x in img_paths]

def autosplit(path=DATASETS_DIR / 'images', weights=(0.9, 0.1, 0.0), annotated_only=False):

    path = Path(path) # images dir
    files = sorted(x for x in path.rglob('*.*') if x.suffix[1:].lower() in IMG_FORMATS) # image files only
    n = len(files) # number of files
    random.seed(0) # for reproducibility
    indices = random.choices([0, 1, 2], weights=weights, k=n) # assign each image to a split

    txt = ['train.txt', 'valid.txt', ] # 2 txt files
    for x in txt:
        if (path.parent / x).exists():
            (path.parent / x).unlink() # remove existing

```

```

print(f'Autosplitting images from {path}' + ', using *.txt labeled images only' * annotated_only)
for i, img in tqdm(zip(indices, files), total=n):
    if not annotated_only or Path(img2label_paths([str(img)]))[0].exists(): # check label
        with open(path.parent / txt[i], 'a') as f:
            f.write(f'./{img.relative_to(path.parent).as_posix()} + '\n') # add image to txt file

autosplit()

```

Gambar 3.4 Kode Program Proses Pembagian *Dataset*

Pertama, kode tersebut mengimpor beberapa *library* yang diperlukan seperti `tqdm` untuk menampilkan progress bar, dan `pathlib` untuk memanipulasi *path* file. Kemudian, didefinisikan variabel `DATASETS_DIR` yang merupakan *path* direktori *dataset*. Selanjutnya, terdapat *tuple* `IMG_FORMATS` yang berisi format-format gambar yang akan disertakan dalam dataset.

Fungsi `img2label_paths` digunakan untuk mendapatkan *path*-label berdasarkan *path* gambar. Selanjutnya, terdapat fungsi `autosplit` yang bertugas untuk membagi *dataset* menjadi subset pelatihan dan validasi. Di dalam fungsi ini, dilakukan pengambilan file-file gambar dari direktori yang ditentukan, kemudian dilakukan pembagian dengan bobot tertentu. Setelah itu, dilakukan pengecekan apakah gambar tersebut memiliki label atau tidak, tergantung pada parameter `annotated_only`. Gambar-gambar yang memenuhi syarat kemudian akan ditambahkan ke dalam file teks yang sesuai dengan subsetnya.

Setelah proses pembagian *dataset* dilakukan, hasilnya akan disimpan dalam dua file teks yang berbeda, yaitu 'train.txt' dan 'valid.txt'. Kedua file ini akan berisi daftar path relatif dari gambar-gambar yang sudah dibagi ke dalam subset pelatihan dan validasi. Proses penulisan path gambar ke dalam file teks dilakukan menggunakan operasi file *writing* pada Python, di mana setiap path gambar akan ditambahkan sebagai baris baru dalam file teks yang sesuai dengan subsetnya. Kemudian, kedua file tersebut dipindahkan ke file dengan ekstensi `yaml` dengan nama "FaceEmotion.yaml" untuk memudahkan dalam proses pelatihan dan pengujian.

### 3.2.4 *Environment Setting*

*Setup* komputer yang digunakan dalam penelitian terdiri dari MacOS Monterey sebagai sistem operasi dengan CPU Apple M1 dan memori 8 GB. Perangkat lunak yang digunakan adalah Python 3.8.18 dan *library* yang digunakan adalah Torch 2.0.1 serta Ultralytics 8.0.187.

### 3.2.5 *Hyperparameter Setting*

Pemilihan *hyperparameter* yang sesuai merupakan langkah yang sangat penting untuk meningkatkan kinerja selama tahap pelatihan model. Tiga *hyperparameter* inti yang digunakan dalam penelitian ini yaitu *batch size*, jumlah *epoch*, dan *image size* atau resolusi gambar. *Batch size* sebanyak 32 dipilih dengan alasan memungkinkan untuk secara efisien mengelola sejumlah data tertentu dalam setiap iterasi pelatihan sesuai dengan argumen yang dikemukakan oleh peneliti terdahulu. (Lin, 2023), menyatakan bahwa orang-orang dapat memilih ukuran *batch* 32 sebagai awal dan kemudian menyesuaikannya untuk meningkatkan akurasi atau efisiensi. Jumlah *epoch* yang dipilih adalah 300 *epoch*. Jumlah tersebut dipilih berdasarkan saran dari Ultralytics dalam artikelnya yang berjudul “Tips for Best Training Result”. Ultralytics menyarankan untuk mencoba memulai dengan 300 *epoch*. Kemudian, resolusi gambar yang digunakan adalah 320x320. Resolusi ini dipilih karena Resolusi 320x320 cukup tinggi untuk menangkap detail yang penting dalam gambar. Pada awalnya, *image size* yang digunakan adalah 48x48, yaitu ukuran asli dari *dataset* FER2013. Akan tetapi, ketika diuji dengan video pembelajaran daring, model justru sama sekali tidak dapat mendeteksi objek. Oleh karena itu, ukuran gambar kemudian diubah menjadi 320x320. Penelitian terdahulu juga menyatakan bahwa semakin tinggi resolusi yang digunakan, semakin baik deteksi objeknya, tetapi juga semakin besar biaya komputasi dan penyimpanannya (Y. Hao et al., 2022).

### 3.2.6 *Melatih Model*

Setelah mendapatkan anotasi kelas objek dan melakukan pembagian data, langkah berikutnya adalah memproses anotasi tersebut dalam proses yang disebut pelatihan (*training*). Tujuan dari pelatihan ini adalah melatih komputer dengan cara memproses gambar dan anotasi yang telah disiapkan sehingga setiap kelas akan memiliki pola atau karakteristiknya sendiri yang akan menjadi referensi bagi komputer dalam membuat keputusan atau prediksi. Proses pelatihan ini akan menggunakan 1790 gambar yang diperoleh dari hasil pembagian *dataset*. Contoh gambar yang digunakan dalam proses *training* dapat dilihat pada Gambar 3.5.





Gambar 3.5 Gambar Data Pelatihan

Proses pelatihan YOLOv8 dilakukan dengan metode *transfer learning*. Kode program ditulis pada Jupyter Notebook. Kode program untuk proses pelatihan dapat dilihat pada Gambar 3.6

```

from ultralytics import YOLO

# Load a model
model = YOLO('yolov8n.pt') # load a pretrained model (recommended for training)

# Train the model
results = model.train(data='./dataset emotion/faceEmotion.yaml', epochs=300, imgsz=48)

```

Gambar 3.6 Kode untuk Impor Kelas YOLO

Langkah awal yang dilakukan adalah dengan mengimpor *library* ultralytics. *Library* ini khusus dirancang untuk bekerja dengan model YOLO (*You Only Look Once*) untuk berbagai keperluan, salah satunya pengenalan objek. *Library* ultralytics menyediakan berbagai *tools* yang dapat mempermudah dalam membuat model YOLO sesuai dengan kebutuhan, melatih model, mengevaluasi model, serta melakukan prediksi.

Setelah proses impor *library* ultralytics selesai dilakukan, langkah selanjutnya adalah mengimpor kelas YOLO pada *library* ultralytics. Kemudian, model YOLO *pretrained* dimuat dari file yolov8n.pt untuk melatih YOLO dengan ukuran nano. Selanjutnya, model dilatih menggunakan data yang terdapat dalam file faceEmotion.yaml yang berisi data train.txt dan valid.txt yang diperoleh dari hasil pembagian *dataset*. Proses pelatihan dilakukan dengan dua ukuran gambar, yang pertama menggunakan ukuran gambar 48x48 piksel (ukuran asli gambar FER2013) dan yang kedua menggunakan ukuran gambar 320x320 piksel sesuai dengan *hyperparameter* yang sudah ditentukan pada penelitian ini. Perubahan ukuran gambar cukup dengan mengubah nilai pada kode program “imgsz” menjadi ukuran gambar yang diinginkan.

Proses pelatihan pertama dilakukan selama 300 *epoch* dengan gambar berukuran 48x48 piksel. Hasil pelatihan disimpan dalam variabel results yang berisi informasi tentang akurasi model dan waktu pelatihan. Pelatihan dengan menggunakan ukuran gambar 320x320 piksel dilakukan setelah model dengan ukuran gambar 48x48 selesai diuji pada video pembelajaran

daring asli. Hal ini bertujuan untuk mengecek terlebih dahulu, apakah perlu mengubah ukuran gambar ke resolusi yang lebih besar.

### 3.2.7 Menguji Model

Pengujian model dilakukan dengan 10% dari total *dataset* yang telah dipisah pada proses pembagian data sebelumnya atau sejumlah 210 gambar. Model yang akan diuji pada penelitian ini adalah model YOLOv8n yang dilatih menggunakan ukuran gambar 48x48 dan YOLOv8n, YOLOv8s, YOLOv8m, serta YOLOv8l yang dilatih menggunakan ukuran gambar 320x320 dengan menggunakan sejumlah 210 gambar yang diperoleh dari hasil pembagian *dataset*. Adapun contoh gambar yang digunakan dalam proses pengujian ini dapat dilihat pada Gambar 3.7.



Gambar 3.7 Gambar Data Pengujian

Dari hasil pengujian, beberapa metrik yang akan diambil adalah mAP, presisi, dan waktu inferensi. Selain itu, pengujian ini juga akan menampilkan *confusion matrix* untuk melihat sebaik apa model tersebut dalam memprediksi kelas antusias dan tidak antusias. Akan tetapi, penelitian ini menggunakan nilai *mean average precision* (mAP) sebagai tolok ukur. Alasannya, mAP merupakan metrik yang umum dan populer untuk mengevaluasi model deteksi objek. mAP memberikan gambaran rata-rata akurasi model dalam mendeteksi objek di semua kelas. Pengujian juga dilakukan pada beberapa video pembelajaran daring. Tujuan utama dari menguji model adalah untuk mengetahui seberapa baik model dapat mendeteksi ekspresi dan menerjemahkannya ke dalam kelas antusias atau tidak antusias.

Pengujian pertama akan dilakukan pada model YOLOv8n yang dilatih menggunakan gambar berukuran 48x48 piksel untuk melihat seberapa baik performanya terhadap video pembelajaran daring. Setelah didapatkan hasil bahwa YOLOv8n yang dilatih menggunakan gambar berukuran 48x48 tidak dapat mendeteksi antusiasme siswa, kemudian proses pengujian dilakukan pada model YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l yang dilatih dengan



ukuran gambar 320x320. Proses ini nantinya akan membandingkan hasil mAP dari masing-masing model. Kemudian hasil terbaik tersebut nantinya akan dianalisis lebih dalam. Hasil pengujian akan dibahas di bab selanjutnya.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi

Tahapan yang telah tertera pada *flowchart* seperti pada Gambar 3.1. *Flowchart* diimplementasikan ke dalam kode program dengan bahasa pemrograman Python versi 3.8.18. Implementasi ini dilakukan secara sistematis sesuai dengan urutan langkah yang digambarkan pada *flowchart*. Setelah kode program selesai ditulis, proses eksekusi dilakukan menggunakan perangkat lunak Jupyter Notebook. Jupyter Notebook dipilih karena kemampuannya dalam mendukung penulisan dan menjalankan kode Python secara interaktif.

#### 4.2 Hasil Pengolahan *Dataset*

Sub bab ini menjelaskan terkait hasil dari pelabelan *dataset* dan pembagian *dataset*. Data yang telah diolah pada tahap ini, selanjutnya digunakan dalam proses *training*.

##### 4.2.1 Hasil Proses Pelabelan *Dataset*

Proses pelabelan dilakukan secara terperinci dengan membagi data menjadi dua kelas yang berbeda, yaitu “antusias” dan “tidak antusias” di mana angka 0 mewakili “tidak antusias” dan angka 1 mewakili “antusias”. Proses pelabelan ini melibatkan partisipasi dari sejumlah individu sebanyak lima orang dengan latar belakang serta jenis kelamin yang berbeda dan disesuaikan dengan penelitian yang dilakukan, untuk memastikan bahwa pendefinisian kelas antusias tidak hanya bergantung pada sudut pandang penulis sehingga diharapkan pendefinisian antusias dan tidak antusias dapat lebih objektif dan akurat. Individu yang terlibat dalam proses pelabelan di antaranya adalah dua orang dosen dan tiga mahasiswa. Lima orang individu ini terdiri dari dua orang dosen dan tiga orang mahasiswa, yaitu Bapak Budi Sulistiyo Jati, S.Kom., M.Eng. dosen informatika Universitas Mercu Buana, Ibu Rakhma Safrida Kurnia, S.Kom., M.Kom. dosen informatika Universitas Amikom, serta Mayla Ayuni Sonya, Sonya Ainurohmah, dan Gabriel Imam Andaru mahasiswa informatika Universitas Islam Indonesia.

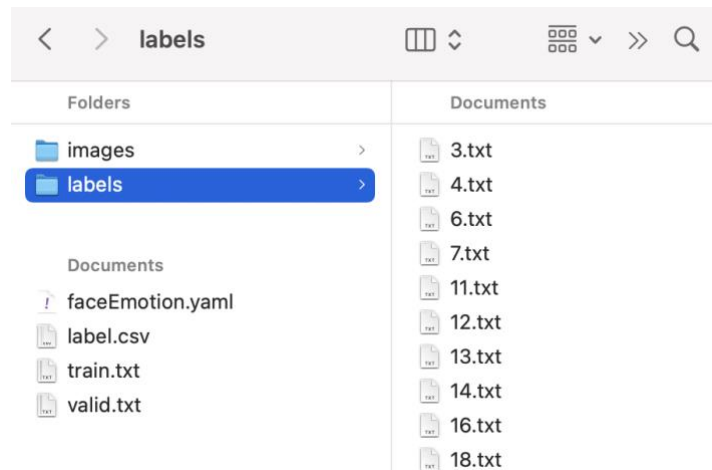
Proses pelabelan berlangsung selama dua kali pertemuan karena jumlah gambar yang tidak sedikit, sehingga di pertemuan pertama partisipan hanya dapat melabeli 1000 gambar saja. Kemudian, di pertemuan kedua langkah yang sama juga dilakukan, yaitu dengan tetap memanfaatkan *platform* Zoom untuk berbagi layar menampilkan 1000 gambar sisa dari keseluruhan *dataset* yang ada. Hasil dari proses pelabelan ini akan termuat dalam Google Sheet

yang dapat dilihat pada Gambar 4.1. Kemudian, masing-masing gambar akan diambil modulusnya untuk menentukan apakah gambar tersebut termasuk ke dalam kelas “antusias” atau “tidak antusias”. Berdasarkan proses tersebut, didapatkan hasil bahwa sejumlah 1013 gambar dinilai sebagai “antusias” dan 987 gambar dinilai sebagai “tidak antusias” yang kemudian hasil tersebut disimpan dalam bentuk *file* csv.

Nomor gambar	File Name	Budi Sulistiyo Jati, S.Kom, M.Eng.	Rakhma Shafrida Kurnia, S.Kom, M.Kom.	Mayla Ayuni Sonya	Gabriel Imam Andaru	Sonya Ainurohmah	Hasil		Jumlah Gambar	Label
1	3.jpg	0	0	0	0	0	0		987	0
2	4.jpg	0	0	0	0	0	0		1013	1
3	6.jpg	0	0	0	0	0	0			
4	7.jpg	1	1	1	1	1	1			
5	11.jpg	0	0	1	1	1	1			
6	12.jpg	0	0	0	1	0	0			
7	13.jpg	0	0	0	1	0	0			
8	14.jpg	1	1	1	1	1	1			
9	16.jpg	1	0	1	1	1	1			
10	18.jpg	0	0	0	0	0	0			

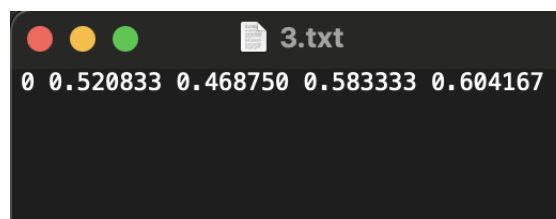
Gambar 4.1 Hasil Pelabelan oleh Lima Partisipan

Setelah mendapatkan hasil pelabelan dari lima partisipan, proses selanjutnya pelabelan menggunakan kode program. Kendala yang terjadi pada proses ini adalah harus dilakukannya tujuh kali *running* kode program karena proses pengubahan label yang dilakukan satu per satu. Label awal yang semula terdiri dari angka 0 sampai dengan 6, harus diubah hanya menjadi 0 dan 1. Proses ini dilakukan dengan mengecek masing-masing label. Kemudian keseluruhan gambar yang sudah dilabeli disimpan dalam folder labels dengan format txt. Masing-masing gambar memiliki nama tersendiri, sehingga nantinya ketika dilakukan pelabelan dengan LabelImg, penulis dapat mengecek kembali gambar mana yang harus diubah menjadi 0 dan 1 sesuai dengan nama gambar dan hasil pelabelan dengan kode program. Bentuk hasil pelabelan kode program dapat dilihat pada Gambar 4.2



Gambar 4.2 Hasil Pelabelan dengan Kode Program

Setelah itu, langkah selanjutnya adalah pelabelan menggunakan *tool* LabelImg dengan cara membuat kotak yang berisi nama kelas, koordinat dua titik (x, y) dengan koordinat x menunjukkan sudut kiri atas dan y menunjukkan sudut kiri bawah kotak. Titik koordinat ini berfungsi sebagai informasi lokasi dan ukuran objek dalam gambar. Proses ini dilakukan dengan menyamakan gambar yang diberi kotak dengan label yang dibuat menggunakan kode program. Langkah dalam proses pelabelan ini memang memakan waktu dan tahap yang tidak terlalu efisien, tetapi mampu memberikan hasil yang baik. Hasil dari proses pelabelan dapat dilihat pada Gambar 4.3.



Gambar 4.3 Hasil Pelabelan Gambar dengan LabelImg

Berdasarkan hasil pelabelan tersebut, terdapat empat kolom yang terdiri atas:

- a. Kolom pertama merupakan id kelas objek. Id kelas 0 merepresentasikan objek dengan label “tidak antusias”.
- b. Kolom kedua dan ketiga menggambarkan pusat titik pada sumbu x dan pusat titik pada sumbu y.
- c. Kolom keempat dan kelima menggambarkan lebar dan tinggi dari kotak pembatas label.

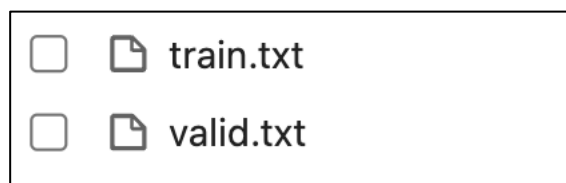
#### 4.2.2 Hasil Proses Pembagian *Dataset*

Berdasarkan pembagian *dataset* dengan komposisi 90% untuk data *training* dan 10% untuk data *testing*, diperoleh hasil, 1790 gambar untuk *training* dan 210 gambar untuk *testing*. Proses pembagian dilakukan secara proporsional pada masing-masing label untuk memastikan distribusi antara label antusias dan tidak antusias terdistribusi dengan baik.

Secara rinci, *dataset* awal terdiri dari 1013 gambar yang termasuk dalam kelas antusias dan 987 gambar yang termasuk dalam kelas tidak antusias. Untuk membagi *dataset* menjadi data *training* dan data *testing*, 90% dari masing-masing kelas diambil sebagai data *training* dan 10% sisanya sebagai data *testing*. Untuk membagi *dataset* menjadi data *training* dan data *testing*, skrip yang digunakan menghasilkan pembagian dengan sedikit deviasi dari proporsi 90:10. Deviasi tersebut akan baik untuk model karena mensimulasikan ketidakaturan yang akan menjadikan model tersebut bagus.

Untuk data *training*, dari 1013 gambar antusias, sebanyak 906 gambar diambil untuk data *training*. Demikian pula, dari 987 gambar tidak antusias, sebanyak 884 gambar diambil untuk data *training*. Sehingga, total gambar yang digunakan untuk data *training* adalah 906 gambar antusias dan 884 gambar tidak antusias, yang jumlah keseluruhannya mencapai 1790 gambar.

Sedangkan untuk data *testing*, dari 1013 gambar antusias, sebanyak 107 gambar diambil untuk data *testing*. Dari 987 gambar tidak antusias, sebanyak 103 gambar diambil untuk data *testing*. Dengan demikian, total gambar yang digunakan untuk data *testing* adalah 107 gambar antusias dan 103 gambar tidak antusias, yang jumlah keseluruhannya mencapai 210 gambar. Hasil proses pembagian *dataset* ini langsung disimpan dalam file `train.txt` dan `valid.txt` seperti pada Gambar 4.4. Kedua file ini kemudian disimpan dalam file dengan format `yaml` dengan nama `faceEmotion.yaml` untuk memudahkan dalam proses pelatihan dan pengujian model.



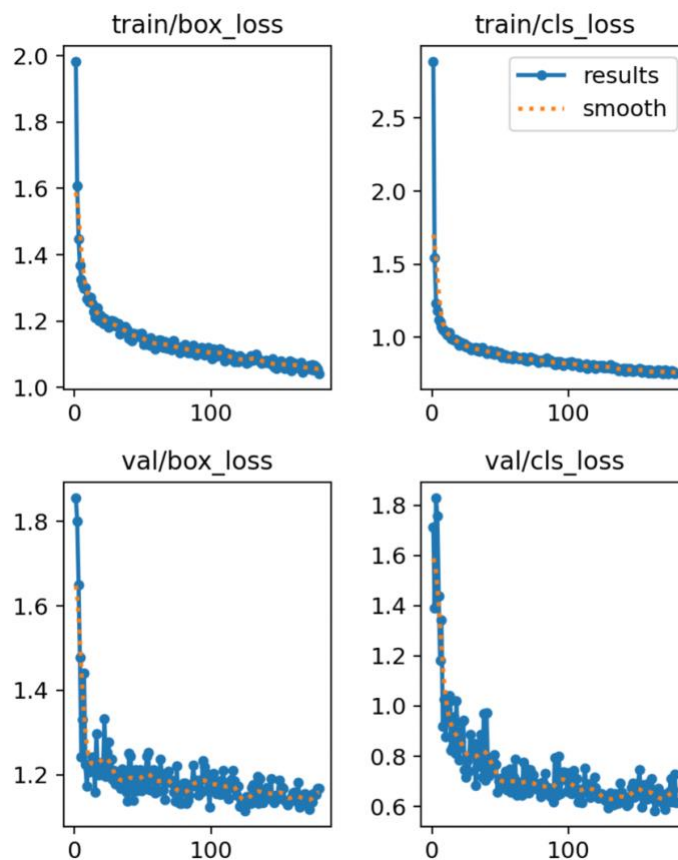
Gambar 4.4 Hasil Pembagian *Dataset*

#### 4.3 Hasil Pelatihan

Bagian ini menjelaskan hasil dari pelatihan model YOLOv8n yang dilatih dengan ukuran gambar 48x48, serta YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l yang dilatih dengan ukuran gambar 320x320. Masing-masing model dilatih menggunakan *epoch* dan *batch size*

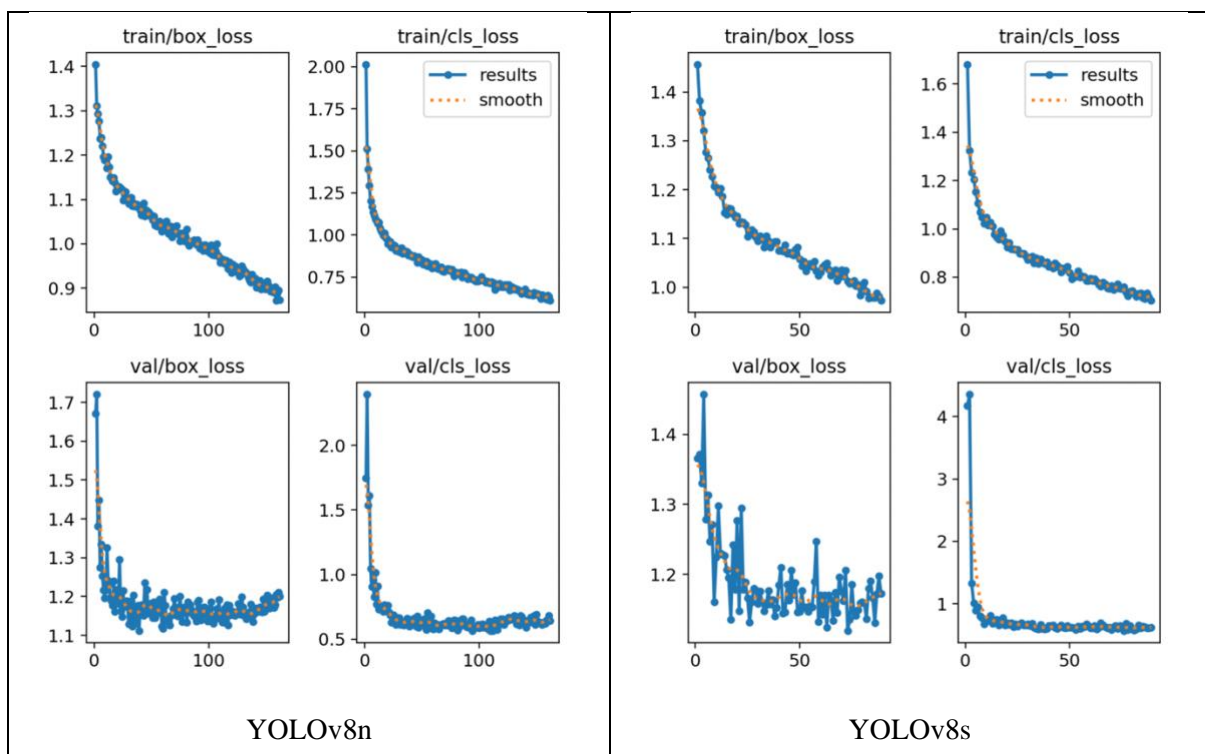
yang sama. Jumlah *epoch* yang digunakan dalam pelatihan model adalah sebanyak 300 *epoch*, *batch size* yang digunakan adalah sebanyak 32, dan *image size* yang digunakan adalah pada pelatihan pertama adalah 48.

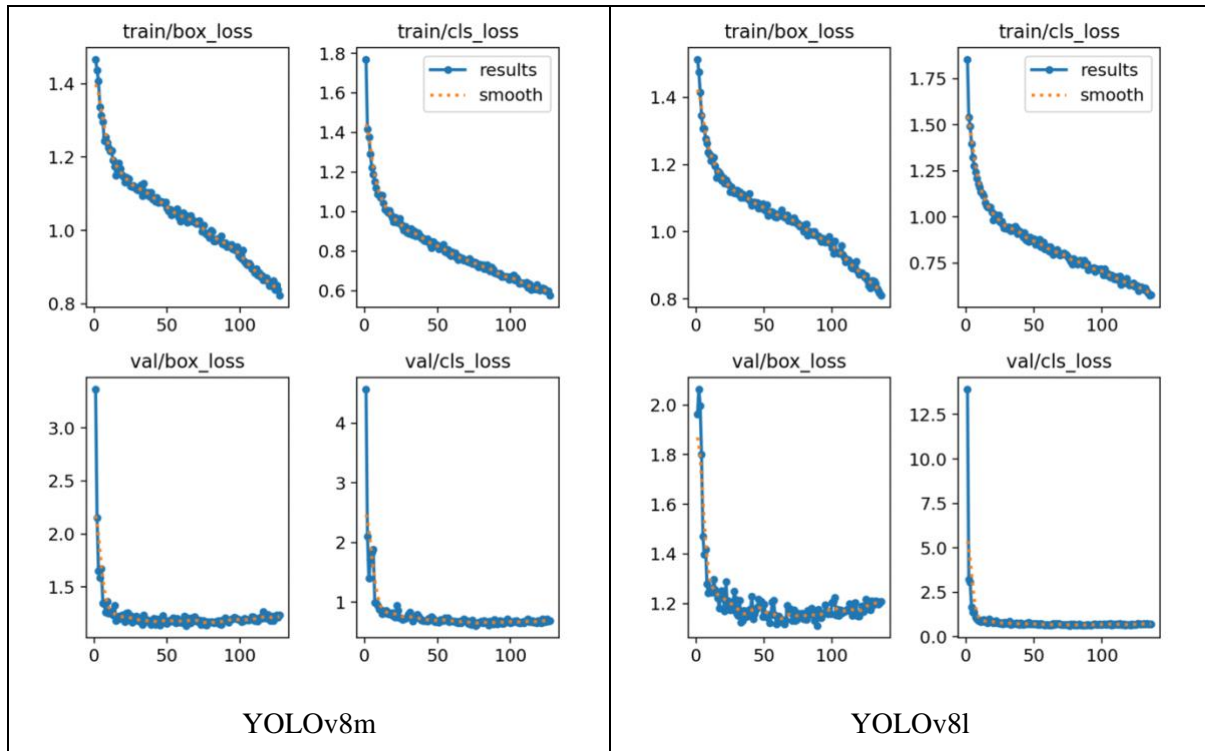
Hasil pelatihan model YOLOv8n ditunjukkan dalam Gambar 4.5. Gambar 4.5 menampilkan grafik *loss*. *Loss function* sendiri digunakan untuk mengukur seberapa baik model dapat memprediksi posisi kotak objek (*box*) dan mengenali kelas objek (*cls*). Sumbu X mewakili jumlah *epoch*, yang menandakan berapa lama model telah dilatih, sedangkan sumbu Y mewakili nilai *loss*. Semakin kecil nilai *loss*, semakin akurat model dalam memprediksi posisi *box* dan kelas objek. *Box\_loss* adalah *loss function* yang mengukur perbedaan antara prediksi lokasi kotak objek yang dihasilkan oleh model dan lokasi kotak yang sebenarnya dalam *dataset*. Semakin kecil nilai *box\_loss*, semakin dekat prediksi kotak model dengan kotak yang sebenarnya, menandakan kemampuan model dalam menentukan posisi objek dengan akurat. Di sisi lain, *cls\_loss* adalah *loss function* yang mengukur perbedaan antara prediksi kelas objek yang dihasilkan oleh model dan kelas objek yang sebenarnya dalam *dataset*. Semakin kecil nilai *cls\_loss*, semakin akurat model dalam mengenali kelas objek dengan benar.



Gambar 4.5 Hasil Pelatihan YOLOv8n Menggunakan Ukuran Gambar 48x48

Berdasarkan hasil pelatihan YOLOv8n dengan ukuran gambar 48x48, diketahui bahwa model terus mengalami penurunan nilai *loss*. Nilai *loss* pada grafik train/box\_loss menurun dari 2.0 ke 1.0, sementara grafik train/cls\_loss menurun dari 2.5 ke 1.0. Hal ini berarti model terus belajar untuk memprediksi posisi kotak objek dan kelas objek dengan lebih akurat seiring waktu. Kemudian juga terjadi penurunan pada grafik val/box\_loss dan val/cls\_loss, di mana grafik val/box\_loss nilainya turun dari 1.8 ke 1.2 dan val/cls\_loss turun dari 1.8 ke 0.6. Selain itu, pada grafik val/box\_loss dan val/cls\_loss terjadi fluktuasi. Fluktuasi pada grafik menunjukkan bahwa model mencoba menyesuaikan dengan data pelatihan. Nilai akhir *box loss* dan *class loss* relatif rendah, menunjukkan bahwa model telah mencapai tingkat akurasi yang cukup baik.





Gambar 4.6 Hasil Pelatihan Menggunakan Ukuran Gambar 320x320

Gambar 4.6 merupakan hasil pelatihan untuk model YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l yang dilatih menggunakan ukuran gambar 320x320. Sama seperti model sebelumnya yang diuji menggunakan ukuran gambar 48x48, model-model ini juga akan dilihat grafik *loss*-nya. Berdasarkan hasil pelatihan YOLOv8n, terlihat bahwa model menunjukkan penurunan yang stabil dalam nilai *loss*. Grafik *train/box\_loss* menurun dari 1,4 ke 0,9, sementara grafik *train/cls\_loss* menurun dari 2,0 ke 0,75. Hal ini menunjukkan peningkatan signifikan dalam akurasi model dalam memprediksi posisi kotak objek dan mengenali kelas objek pada data pelatihan. Nilai pada grafik *val/box\_loss* menurun dari angka 1,7 ke 1,1 dan nilai pada grafik *val/cls\_loss* turun dari angka 2,0 ke 0,5. Dalam hal ini, grafik *val/box\_loss* dan grafik *val/cls\_loss* merupakan grafik yang menunjukkan nilai *loss* untuk memprediksi posisi box objek dan memprediksi kelas objek dalam data validasi.

Hasil pelatihan YOLOv8s, menunjukkan bahwa terjadi fluktuasi pada grafik *loss*. Meskipun demikian, nilai *loss* pada kedua grafik *train/box\_loss* dan *train/cls\_loss* menurun dari awal pelatihan hingga *epoch* ke-50. Grafik *train/box\_loss* menurun dari 1,4 ke 1,0, dan grafik *train/cls\_loss* menurun dari 1,6 ke 0,8. Nilai pada grafik *val/box\_loss* dan *val/cls\_loss* juga mengalami penurunan dari nilai 1,4 ke 1,2 dan dari nilai 4,0 ke 1,0. Kemudian, dapat dilihat bahwa grafik *val/box\_loss* menunjukkan perubahan yang relatif tidak stabil pada setiap



*epochnya*. Grafik yang fluktuatif dapat mengindikasikan bahwa model sedang mencoba untuk menyesuaikan diri terlalu dekat dengan data pelatihan, yang dapat mengarah ke *overfitting*.

Sementara itu, pada hasil pelatihan YOLOv8m, grafik train/box\_loss menurun dari 1,4 ke 0,8 dan grafik train/cls\_loss menurun dari 1,8 ke 0,6. Nilai pada grafik val/box\_loss dari 3,0 ke 1,5, dan val/cls\_loss dari 4,0 ke 1,0. Proses pelatihan YOLOv8m berhenti pada angka di atas 100, jauh lebih tinggi dibandingkan dengan YOLOv8s yang berarti waktu yang dibutuhkan jauh lebih lama dibanding dengan YOLOv8s. Meskipun memerlukan waktu pelatihan yang lebih lama, YOLOv8m memberikan hasil yang lebih akurat, menunjukkan kemampuan yang baik dalam memprediksi posisi box dan kelas objek pada data pelatihan.

Grafik dari YOLOv8l menunjukkan hasil proses pelatihan berhenti pada *epoch* ke-100. Meskipun jumlah *epoch* yang digunakan adalah 300, tetapi YOLOv8 akan berhenti secara otomatis ketika model tidak menunjukkan peningkatan performa yang signifikan selama beberapa *epoch* berturut-turut. Oleh karena itu, pelatihan pada YOLOv8l terhenti pada *epoch* ke-100. Kemudian, grafik train/box\_loss dan train/cls\_loss mengalami penurunan dari 1,4 ke 0,8 dan 1,75 ke 0,75. Nilai pada grafik val/box\_loss dari 2,0 ke 1,2, dan val/cls\_loss dari 12,5 ke 0,0.

Perbandingan nilai *loss* antar model memungkinkan evaluasi kinerja relatif dari masing-masing model. Semakin kecil nilai *loss*, semakin akurat model dalam memprediksi posisi kotak dan kelas objek. Namun, perlu diperhatikan bahwa nilai *loss* pada data pelatihan tidak selalu mencerminkan kinerja yang baik pada data yang belum pernah dilihat sebelumnya (data validasi). Oleh karena itu, grafik *loss* pada data validasi (val/box\_loss dan val/cls\_loss) juga perlu diperhatikan untuk menghindari *overfitting*, yaitu kondisi di mana model terlalu disesuaikan dengan data pelatihan dan kinerjanya menurun pada data baru. Hasil perbandingan akan disajikan dalam bentuk tabel yang dapat dilihat pada Tabel 4.1. Nilai *loss* terkecil akan ditandai dengan tinta warna merah untuk membedakan dengan hasil lainnya.

Tabel 4.1 Tabel Perbandingan *Box Loss* dan *Class Loss*

Model	Box_loss	Cls_loss
YOLOv8n	1,1	0,5
YOLOv8s	1,2	1,0
YOLOv8m	1,5	1,0
YOLOv8l	1,2	0,0

#### 4.4 Hasil Pengujian

Sub bab ini akan menjelaskan hasil pengujian dari masing-masing ukuran model dan pada gambar dengan ukuran 48x48 piksel dengan fokus pada penggunaan metrik presisi, mAP (*mean Average Precision*), dan *inference time*. Selain itu, juga akan dipaparkan hasil dalam bentuk *confusion matrix*. Pemilihan metrik ini didasarkan pada kebutuhan untuk mendapatkan pemahaman yang holistik tentang performa model deteksi objek, khususnya dalam konteks pengenalan ekspresi wajah. Akan tetapi, fokus utamanya tetap pada metrik mAP.

Pertama, presisi menjadi relevan untuk mengukur sejauh mana model memberikan respon yang tepat terhadap deteksi objek. Presisi menunjukkan rasio antara jumlah *true positive* dengan total hasil positif yang diberikan oleh model. Dalam konteks deteksi ekspresi wajah, presisi menjadi indikator sejauh mana hasil deteksi yang diberikan oleh model merupakan ekspresi wajah yang sesungguhnya.

Kedua, mAP digunakan sebagai metrik komprehensif yang menggabungkan presisi dan *recall* dalam satu nilai. mAP memberikan ukuran akurasi deteksi objek dengan mempertimbangkan *precision-recall curve* untuk setiap kelas. Penggunaan mAP dapat memberikan gambaran yang lebih lengkap tentang performa model deteksi, terutama dalam tugas yang melibatkan *multiple object detection* seperti identifikasi ekspresi wajah.

Ketiga, *inference time* menjadi parameter penting untuk mengevaluasi efisiensi model. Waktu inferensi mengukur berapa lama model memproses satu data input. Dalam konteks ini, evaluasi waktu inferensi dapat memberikan informasi tentang seberapa cepat dan efisien model dapat mengenali ekspresi wajah.

Dari keseluruhan proses pengujian deteksi ekspresi menggunakan algoritma YOLOv8, didapatkan hasil yang menunjukkan bahwa kelas antusias tidak pernah dikenali sebagai *background*. Hal ini dibuktikan dengan *confusion matrix* yang dihasilkan, di mana tidak terdapat satupun gambar dari kelas antusias yang diprediksi sebagai *background*. *Confusion matrix* ini secara jelas menunjukkan bahwa model YOLOv8 mampu membedakan kelas antusias dengan sangat baik dari kelas *background*, menunjukkan keandalan model dalam mengidentifikasi ekspresi antusias secara akurat.

Selain itu, pada proses pengujian ini, terdapat satu gambar yang mengalami korup. Hal ini disebabkan oleh galat yang terjadi pada library *ultralytics*. Saat melakukan inferensi, library tersebut gagal memproses gambar tersebut dengan benar. Akibatnya, gambar tidak dapat dianalisis seperti yang diharapkan. Kendala ini mengakibatkan data yang dihasilkan dari

gambar tersebut menjadi tidak valid. Oleh karena itu, gambar korup ini tidak digunakan dalam analisis lebih lanjut sehingga gambar yang digunakan hanya berjumlah 209 gambar.

#### 4.4.1 Hasil Pengujian pada Ukuran Gambar 48x48

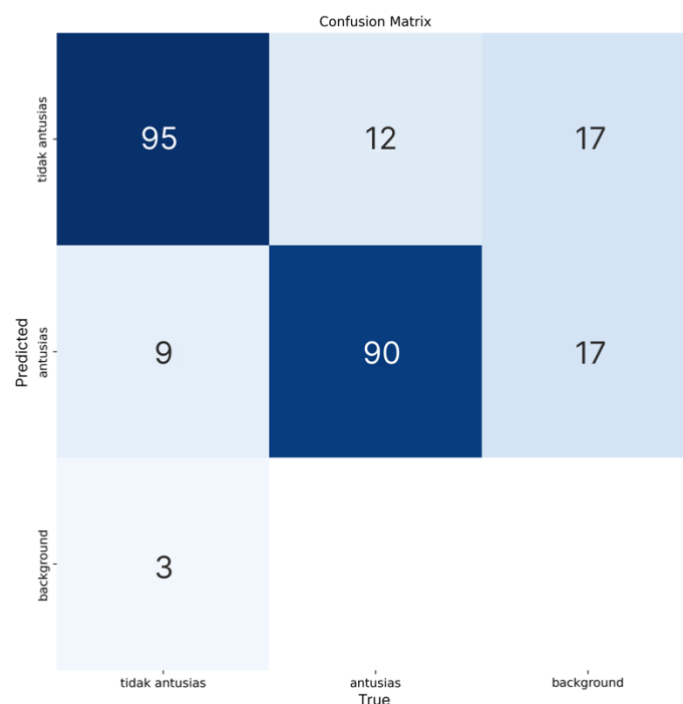
```

3 exceeds dataset class count 2. Possible class labels are 0-1
Class      Images  Instances  Box (P  R      mAP50  mAP50-95): 100%|
-----
all        209     209       0.837  0.87   0.92   0.602
tidak antusias  209     107       0.819  0.888  0.903  0.589
antusias   209     102       0.854  0.853  0.938  0.615
Speed: 0.7ms preprocess, 5.6ms inference, 0.0ms loss, 9.9ms postprocess per image
Results saved to runs/detect/val15
(yolov8) jupyter-20523239@cds-tesla:~/yolov8/data$

```

Gambar 4.7 Hasil *Testing* dengan Ukuran Gambar 48x48

Gambar 4.7 merupakan hasil pengujian yang dilakukan menggunakan YOLOv8 dengan ukuran *nano* yang diujikan pada gambar dengan ukuran 48x48. Ukuran gambar ini merupakan ukuran gambar asli dari *dataset* FER2013. Berdasarkan hasil pengujian, didapatkan hasil presisi 0,837, mAP@.5 0,92, dan *inference time* 5,6 ms. Hasil pengujian menunjukkan bahwa model memiliki kinerja yang sangat baik dalam mendeteksi dua kelas emosi pada *dataset* yang digunakan.



Gambar 4.8 *Confusion Matrix* Model dengan Ukuran Gambar 48x48

Performa model ini juga dapat dilihat berdasarkan *confusion matrix* pada Gambar 4.8. Pada kelas tidak antusias, model mampu memprediksi 95 gambar dengan benar sebagai tidak

antusias, sembilan gambar dengan salah sebagai antusias, dan tiga gambar dengan salah sebagai *background*. Sedangkan pada kelas antusias, model mampu memprediksi 90 gambar dengan benar sebagai antusias dan 12 gambar dengan salah sebagai tidak antusias. Selain itu, pada pengujian ini juga diperoleh hasil bahwa model memprediksi sejumlah 17 gambar sebagai kelas tidak antusias dan 17 gambar sebagai kelas antusias yang padahal gambar tersebut sebenarnya termasuk dalam *background class*. *Background class* dalam *object detection* adalah kelas khusus yang digunakan untuk mewakili semua area dalam gambar yang tidak mengandung objek target. Dengan kata lain, *background class* mencakup semua piksel yang tidak termasuk dalam salah satu kelas objek yang dilatih oleh model. Oleh karena itu, meskipun jumlah gambar pada *confusion matrix* terdapat 243 gambar, tetapi jumlah gambar sebenarnya hanya 209 gambar karena 34 gambar merupakan *background class*.

#### 4.4.2 YOLOv8n

Pengujian yang dilakukan pada model YOLOv8n dengan *epoch* sebanyak 300, *batch size* 32, dan *image size* 320 mendapatkan hasil yang baik. Hasil dapat dilihat pada Gambar 4.9.

```

Ultralytics YOLOv8.0.187 Python-3.8.18 torch-2.0.1 CPU (Apple M1)
Model summary (fused): 168 layers, 3006038 parameters, 0 gradients
val: Scanning /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset
val: WARNING Δ /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset emotion/
l class 3 exceeds dataset class count 2. Possible class labels are 0-1

```

Class	Images	Instances	Box(P)	R	mAP50	m
all	209	209	0.888	0.914	0.953	0.666
tidak antusias	209	107	0.924	0.906	0.945	0.624
antusias	209	102	0.851	0.922	0.961	0.709

```

Speed: 0.2ms preprocess, 69.2ms inference, 0.0ms loss, 0.2ms postprocess per image
Results saved to runs/detect/val3

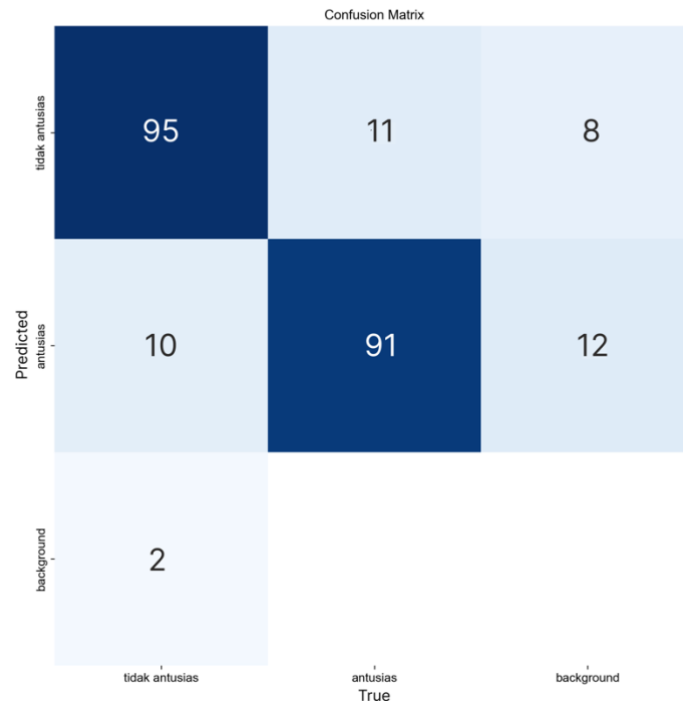
```

Gambar 4.9 Hasil *testing* YOLOv8n

Performa dari model ini mendapatkan tingkat presisi sebesar 0.888. Berdasarkan tingkat presisi yang diperoleh, dapat dikatakan bahwa model YOLOv8n mampu menghasilkan deteksi emosi yang akurat sesuai dengan tujuan utama pengaplikasian model. Selanjutnya, hasil pengujian juga mencakup ukuran mAP. mAP merupakan hasil kalkulasi antara nilai *precision*, *recall*, dan *Average Precision* (AP). Pada mAP@[.5], YOLOv8n mampu menunjukkan kinerja sebesar 0,953. Hal ini berarti bahwa model dapat memberikan presisi tinggi pada tingkat *confidence* 50% dan mampu mendeteksi objek dengan baik.

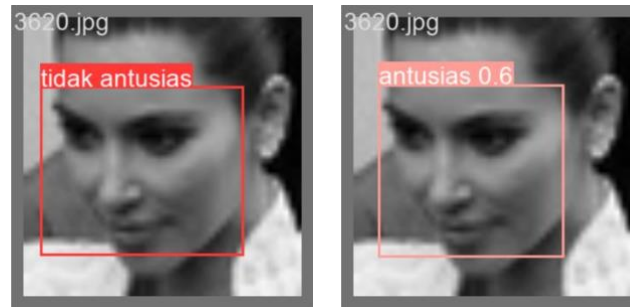
Selain tingkat presisi dan mAP, aspek lain yang diperhatikan dalam penelitian ini adalah *inference time* atau kecepatan model dalam melakukan prediksi. *Inference time* pada pengujian

model YOLOv8n relative rendah, yaitu mencapai 69,2ms. Waktu inferensi yang relatif rendah, dapat mendukung implementasi deteksi emosi secara *real time*.



Gambar 4.10 *Confusion Matrix* YOLOv8n

*Confusion matrix* pada Gambar 4.10 menunjukkan performa model YOLOv8n dalam mendeteksi kelas tidak antusias dan antusias. Matriks ini menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas. Pada kelas tidak antusias, model mampu memprediksi 95 gambar dengan benar sebagai tidak antusias, 10 gambar dengan salah sebagai antusias, dan dua gambar dengan salah sebagai *background*. Sedangkan pada kelas antusias, model mampu memprediksi 91 gambar dengan benar sebagai antusias dan 11 gambar dengan salah sebagai tidak antusias. Contoh kesalahan prediksi dapat dilihat pada Gambar 4.11. Sisi kiri menunjukkan label asli gambar yang dilabeli sebagai tidak antusias sedangkan sisi kanan merupakan hasil prediksi model yang salah mengenali gambar menjadi antusias. Kemudian Gambar 4.12 merupakan contoh kesalahan prediksi di mana model seharusnya mengenali model sebagai tidak antusias, tetapi malah dikenali sebagai *background*. Sisi kiri merupakan gambar yang dilabeli sebagai tidak antusias dan sisi kanan merupakan hasil prediksi model, di mana model tidak mampu memprediksi gambar sebagai tidak antusias.



Gambar 4.11 Gambar yang Salah dikenali



Gambar 4.12 Gambar yang Gagal Dikenali

*Background class* dalam *object detection* adalah kelas khusus yang digunakan untuk mewakili semua area dalam gambar yang tidak mengandung objek target. Dengan kata lain, *background class* mencakup semua piksel yang tidak termasuk dalam salah satu kelas objek yang dilatih oleh model. Pada pengujian ini diperoleh hasil bahwa model memprediksi sejumlah delapan gambar sebagai kelas tidak antusias dan 12 gambar sebagai kelas antusias yang padahal gambar tersebut sebenarnya termasuk dalam *background class* atau bisa dikatakan bahwa model mendeteksi sesuatu yang tidak termasuk ke dalam objek yang sudah ditentukan. Oleh karena itu, meskipun jumlah gambar pada *confusion matrix* terdapat 229 gambar, tetapi jumlah gambar sebenarnya hanya 209 gambar karena 20 gambar merupakan *background class*.

#### 4.4.3 YOLOv8s

Hasil pengujian menunjukkan bahwa model YOLOv8s memiliki kinerja yang sangat baik dalam mendeteksi dua kelas emosi pada *dataset* yang digunakan. Model ini memiliki presisi yang tinggi, yaitu 0,882. Waktu yang dibutuhkan model untuk melakukan prediksi (*inference time*) juga relatif rendah, yaitu 224,7ms masih dapat dikatakan cepat, meskipun tidak secepat YOLOv8n. [mAP@\[.5\]](#) yang diperoleh juga sangat bagus, yaitu 0,952, hanya berbeda

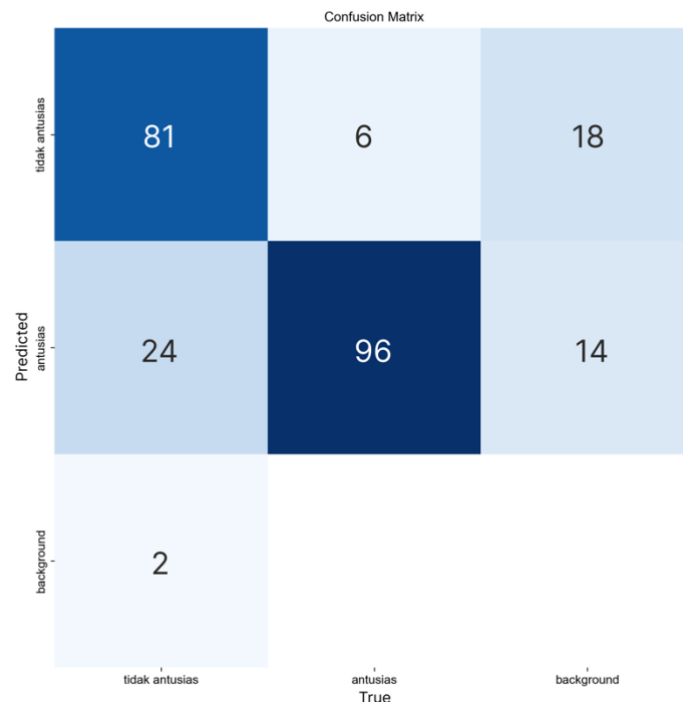
0,001 dengan YOLOv8s. Nilai presisi, [mAP@\[.5\]](#), dan *inference time* dapat dilihat pada Gambar 4.13.

```

Ultralytics YOLOv8.0.187 Python-3.8.18 torch-2.0.1 CPU (Apple M1)
Model summary (fused): 168 layers, 11126358 parameters, 0 gradients
val: Scanning /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset
val: WARNING Δ /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset emotion/i
l class 3 exceeds dataset class count 2. Possible class labels are 0-1
orted hardware.
Class      Images  Instances  Box(P      R      mAP50  m[W NNPACK.c
Class      Images  Instances  Box(P      R      mAP50  m
all        209     209        0.882     0.925  0.952  0.667
tidak antusias  209     107        0.931     0.889  0.944  0.623
antusias  209     102        0.832     0.961  0.96   0.71
Speed: 0.3ms preprocess, 224.7ms inference, 0.0ms loss, 0.3ms postprocess per image
Results saved to runs/detect/val2

```

Gambar 4.13 Hasil *testing* YOLOv8s



Gambar 4.14 Confusion Matrix YOLOv8s

Model YOLOv8s menunjukkan akurasi yang baik dalam mendeteksi objek antusias dan tidak antusias. Hal ini dibuktikan dengan *confusion matrix* pada Gambar 4.14. Pada kelas tidak antusias, model mampu memprediksi 81 gambar dengan benar, 24 gambar dengan salah sebagai antusias, dan dua gambar dengan salah sebagai *background*. Sedangkan pada kelas antusias, model mampu memprediksi 96 gambar dengan benar dan 6 gambar dengan salah.

Namun, model masih memiliki kekurangan dalam membedakan objek target dengan *background*. Hal ini terlihat dari 18 gambar yang diprediksi sebagai kelas tidak antusias dan 14 gambar yang diprediksi sebagai kelas antusias, padahal gambar tersebut sebenarnya tidak termasuk dalam objek target. Oleh karena itu, meskipun jumlah gambar pada *confusion matrix* terdapat 241 gambar, tetapi jumlah gambar sebenarnya hanya 209 gambar karena 32 gambar merupakan *background class*.

#### 4.4.4 YOLOv8m

Berdasarkan hasil pengujian pada model YOLOv8m, ditemukan bahwa model ini menunjukkan performa yang cukup baik berdasarkan nilai presisi, mAP, dan *inference time*. Dengan nilai presisi sebesar 0,911, model mampu memberikan tingkat akurasi yang baik dalam mengklasifikasikan objek positif. Nilai mAP sebesar 0,939 mencerminkan kemampuan model dalam memberikan prediksi dengan tingkat keakuratan yang tinggi, mempertimbangkan presisi dan recall secara bersamaan. Terakhir, waktu yang dibutuhkan model untuk mendeteksi (*inference time*) relatif tinggi, yaitu 602,5ms jauh lebih lama dibanding dengan ukuran *small* dan *nano*. Nilai presisi, mAP, dan *inference time* dapat dilihat pada Gambar 4.15.

```

Ultraalytics YOLOv8.0.187 Python-3.8.18 torch-2.0.1 CPU (Apple M1)
Model summary (fused): 218 layers, 25840918 parameters, 0 gradients
val: Scanning /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset
val: WARNING /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset emotion/
l class 3 exceeds dataset class count 2. Possible class labels are 0-1

```

Class	Images	Instances	Box(P)	R	mAP50	m
all	209	209	0.911	0.878	0.939	0.666
tidak antusias	209	107	0.903	0.874	0.925	0.632
antusias	209	102	0.919	0.882	0.953	0.701

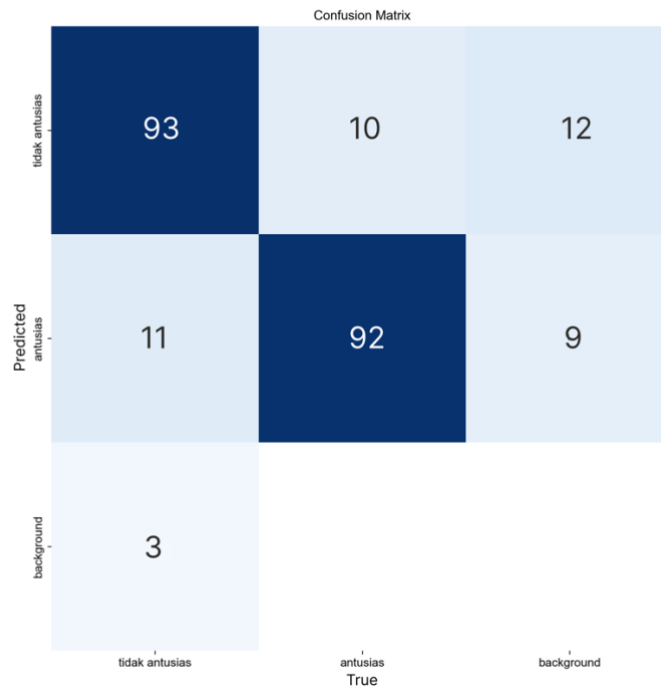
```

Speed: 0.5ms preprocess, 602.5ms inference, 0.0ms loss, 0.2ms postprocess per image
Results saved to runs/detect/val4

```

Gambar 4.15 Hasil *testing* YOLOv8m





Gambar 4.16 *Confusion Matrix* YOLOv8m

Performa model YOLOv8m juga dapat dilihat berdasarkan *confusion matrix* pada Gambar 4.16. Pada kelas tidak antusias, model mampu memprediksi 93 gambar dengan benar, 11 gambar dengan salah sebagai antusias, dan tiga gambar dengan salah sebagai *background*. Sedangkan pada kelas antusias, model mampu memprediksi sejumlah 92 gambar dengan benar dan 10 gambar dengan salah. Selain itu, pada pengujian ini diperoleh hasil bahwa model memprediksi sejumlah 12 gambar sebagai kelas tidak antusias dan sembilan gambar sebagai kelas antusias yang padahal gambar tersebut sebenarnya termasuk dalam *background class*. *Background class* dalam *object detection* adalah kelas khusus yang digunakan untuk mewakili semua area dalam gambar yang tidak mengandung objek target. Dengan kata lain, *background class* mencakup semua piksel yang tidak termasuk dalam salah satu kelas objek yang dilatih oleh model. Oleh karena itu, meskipun jumlah gambar pada *confusion matrix* terdapat 230 gambar, tetapi jumlah gambar sebenarnya hanya 209 gambar karena 21 gambar merupakan *background class*.

#### 4.4.5 YOLOv8l

Hasil pengujian pada YOLOv8l menghasilkan nilai presisi, mAP, dan *inference time* yang cukup tinggi. Hal ini menunjukkan bahwa YOLOv8l memiliki performa yang cukup baik dalam mendeteksi objek. Nilai presisi yang diperoleh adalah 0,855, mAP 0,941, dan *inference time* 1302,8 ms. Meskipun memiliki nilai presisi dan mAP yang cukup tinggi, tetapi ukuran ini

memerlukan waktu lebih lama dalam melakukan prediksi (*inference time*) jika diterapkan pada aplikasi *real-time*. Selain itu, nilai mAP yang diperoleh juga tidak lebih bagus dibanding dengan versi *small* dan *nano*. Hasil pengujian dapat dilihat pada Gambar 4.17.

```

Ultralytics YOLOv8.0.187 Python-3.8.18 torch-2.0.1 CPU (Apple M1)
Model summary (fused): 268 layers, 43608150 parameters, 0 gradients
val: Scanning /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset
val: WARNING /Users/kartikasalma/Documents/SEMESTER 6/PENELITIAN/yolov8/dataset emotion/
l class 3 exceeds dataset class count 2. Possible class labels are 0-1

```

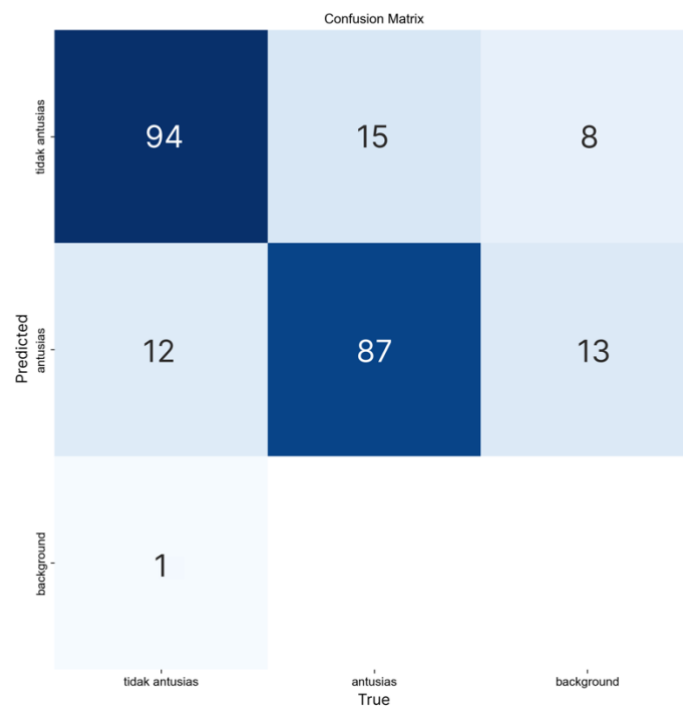
Class	Images	Instances	Box(P	R	mAP50	m
all	209	209	0.855	0.896	0.941	0.666
tidak antusias	209	107	0.873	0.9	0.928	0.625
antusias	209	102	0.836	0.892	0.955	0.707

```

Speed: 0.7ms preprocess, 1302.8ms inference, 0.0ms loss, 0.2ms postprocess per image
Results saved to runs/detect/val5

```

Gambar 4.17 Hasil *testing* YOLOv8l



Gambar 4.18 *Confusion Matrix* YOLOv8l

Gambar 4.18 menunjukkan performa YOLOv8l dalam mendeteksi antusias dan tidak antusias. Dapat dilihat bahwa sejumlah 94 gambar berhasil dideteksi dengan benar untuk kelas tidak antusias, sejumlah 12 gambar dideteksi dengan salah sebagai antusias, dan satu gambar dengan salah sebagai *background*. Pada kelas antusias, 87 gambar berhasil dideteksi dengan benar dan 15 gambar dideteksi dengan salah. Selain itu, sejumlah 8 gambar dideteksi sebagai tidak antusias dan 13 gambar dideteksi sebagai antusias padahal gambar tersebut sebenarnya termasuk dalam *background class*. *Background class* dalam *object detection* adalah kelas khusus yang digunakan untuk mewakili semua area dalam gambar yang tidak mengandung

objek target. Dengan kata lain, *background class* mencakup semua piksel yang tidak termasuk dalam salah satu kelas objek yang dilatih oleh model. Oleh karena itu, meskipun jumlah gambar pada *confusion matrix* terdapat 230 gambar, tetapi jumlah gambar sebenarnya hanya 209 gambar karena 21 gambar merupakan *background class*.

#### 4.5 Pembahasan

Berdasarkan hasil penelitian yang telah dilakukan pada model yang dilatih dengan gambar ukuran 48x48, didapatkan hasil presisi 0,837, mAP 0,92, dan *inference time* 5,6 ms. Hasil yang didapatkan dari pengujian berdasarkan *dataset* FER2013 dengan ukuran 48x48 memang dapat dikatakan baik, tetapi ketika diuji coba pada video rekaman pembelajaran daring, model tersebut memiliki performa yang buruk. Saat diuji coba pada video rekaman pembelajaran daring, model sama sekali tidak mampu mendeteksi wajah yang terdapat dalam video. Berdasarkan hasil pengujian ini, maka kemudian peneliti mencoba untuk mengubah ukuran gambar menjadi 320x320 sesuai dengan saran Y. Hao dkk (2022) yang mengatakan bahwa ukuran gambar 320x320 merupakan ukuran yang bagus untuk memulai pelatihan pada model. Berdasarkan proses pelatihan dan pengujian menggunakan ukuran gambar 320x32, didapatkan hasil untuk masing-masing ukuran dari YOLOv8 yang dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Pengujian YOLOv8 ukuran n, s, m, dan l

Model YOLOv8	Presisi	mAP	<i>Inference Time</i>
YOLOv8n (48x48)	0,837	0,92	5,6 ms
YOLOv8n (320x320)	0,888	0,953	69,2 ms
YOLOv8s (320x320)	0,882	0,952	224,7 ms
YOLOv8m (320x320)	0,911	0,939	602,5 ms
YOLOv8l (320x320)	0,855	0,941	1302,8 ms

Dari hasil yang telah diperoleh, dapat dilihat pada Tabel 4.2 bahwa model YOLOv8 dengan ukuran *nano* menunjukkan nilai mAP tertinggi, mencapai 0,953 atau 95,3%. Meskipun perbedaan nilai antara ukuran *nano* dan ukuran *small* sangat kecil, tetapi model dengan ukuran *nano* unggul 0,001 dan memiliki *inference time* yang sangat kecil, berbeda jauh dengan ukuran lainnya. Perbandingan mAP antar ukuran model merupakan hal yang penting dalam evaluasi

kinerja, karena mAP mencerminkan seberapa baik model dalam mendeteksi objek secara keseluruhan. Selain itu, perhatian terhadap *inference time* juga krusial, karena menandakan seberapa cepat model dapat melakukan prediksi.

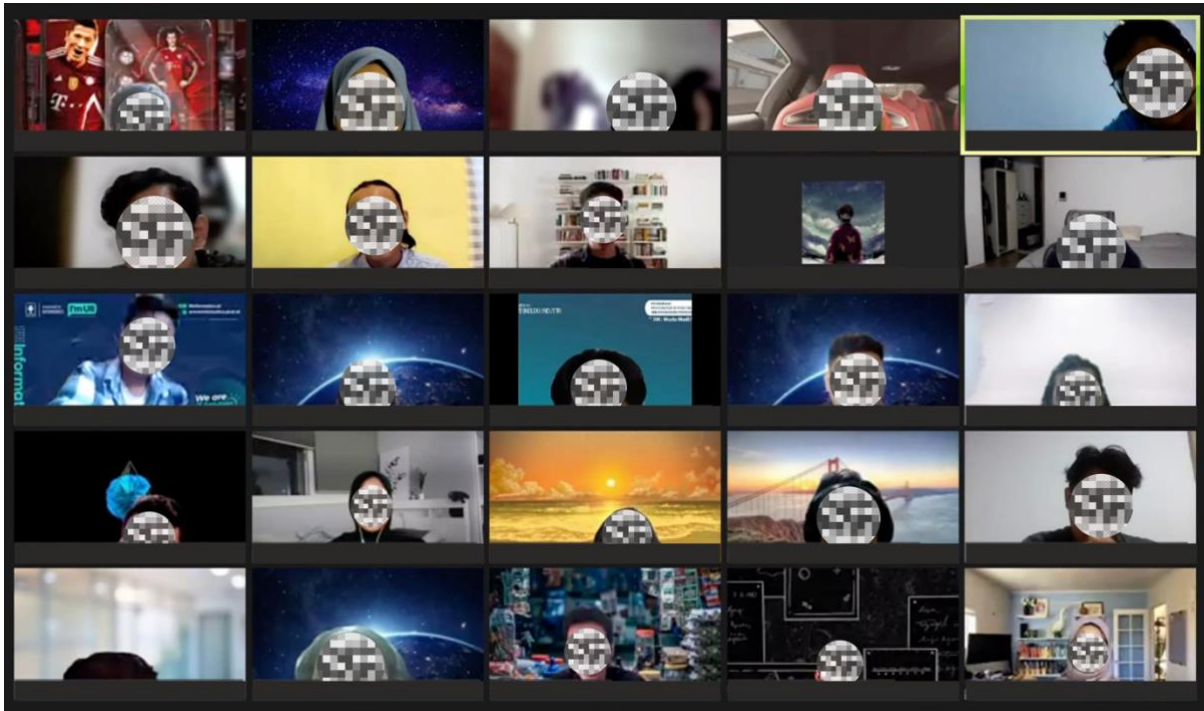
Kemampuan model dalam mendeteksi antusiasme siswa dapat memiliki dampak signifikan dalam evaluasi proses pembelajaran. Model dengan nilai mAP yang lebih tinggi dan *inference time* yang lebih rendah dapat memungkinkan guru untuk secara cepat dan akurat mengidentifikasi tanggapan emosional siswa terhadap materi pembelajaran. Hal ini dapat membantu guru dalam menyusun strategi pembelajaran yang lebih efektif, dengan mengadaptasi pendekatan mereka sesuai dengan kebutuhan siswa secara *real-time*.

Tidak hanya itu, pada pengujian ini terlihat bahwa semakin besar ukuran model, nilai mAP yang dihasilkan semakin kecil. Kemungkinan hal ini terjadi karena penggunaan kelas yang lebih banyak pada pengujian YOLOv8 dengan menggunakan *dataset* MS COCO, yang memiliki 80 kelas, sedangkan dalam penelitian ini hanya terdapat 2 kelas. Perbedaan jumlah kelas ini dapat menyebabkan terjadinya *overfitting*, di mana model terlalu disesuaikan dengan data pelatihan dan kinerjanya menurun pada data yang belum pernah dilihat sebelumnya.

#### **4.5.1 Pembahasan Hasil Pengujian pada Video Pembelajaran Daring**

Pada sub bab ini, model yang dibahas adalah model YOLOv8n yang dilatih dengan ukuran gambar 48x48 dan YOLOv8n yang dilatih dengan ukuran gambar 320x320. Model diuji menggunakan video pembelajaran daring sinkron yang diadakan oleh dosen. Seluruh video yang digunakan untuk pengujian sudah mendapatkan izin dari dosen yang bersangkutan. Selain itu, untuk tetap menjaga privasi dari orang-orang yang terdapat dalam video, maka wajah yang tertampil dalam video akan diblur dalam laporan ini.

Berdasarkan hasil pengujian pada model yang dilatih dengan ukuran gambar 48x48, model tidak dapat mendeteksi wajah mahasiswa dalam video. Hal ini terjadi karena ukuran gambar 48x48 cukup kecil dan mungkin tidak memiliki detail yang cukup untuk mendeteksi fitur wajah secara akurat, terutama ketika diterapkan pada video dengan resolusi lebih tinggi, karena model mungkin kehilangan informasi penting yang diperlukan untuk mendeteksi wajah. Hasil deteksi menggunakan model yang dilatih dengan ukuran gambar 48x48 dapat dilihat pada Gambar 4.19.



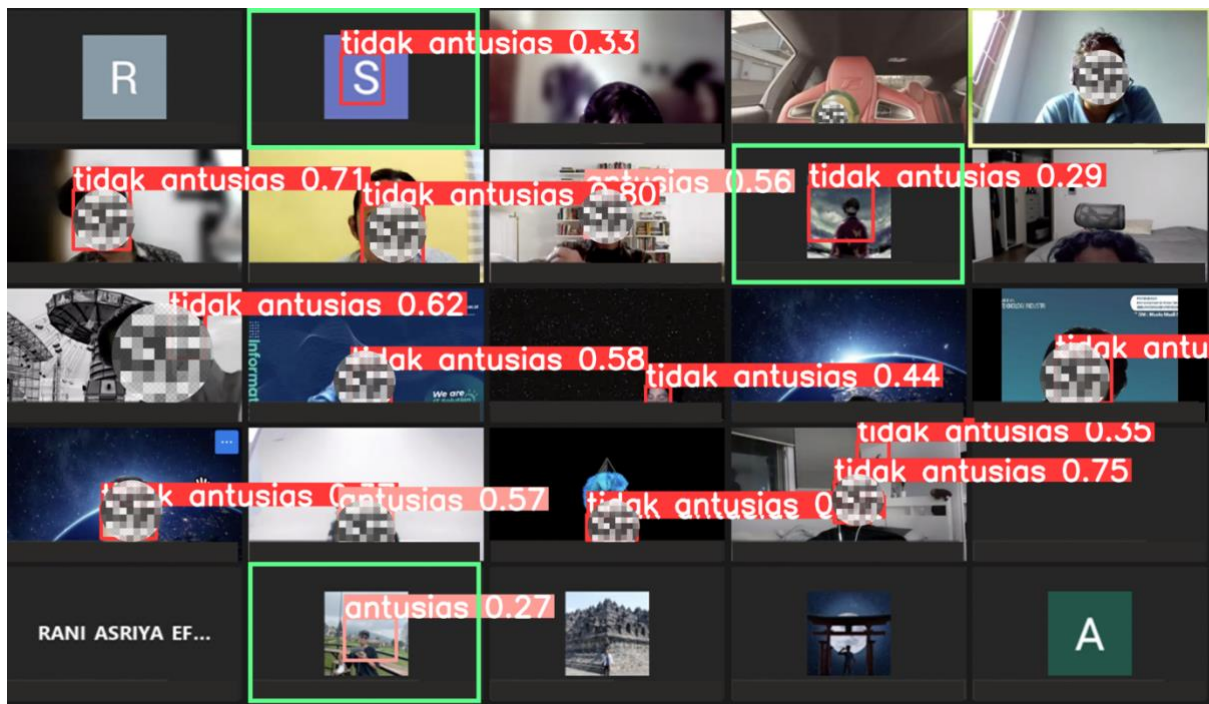
Gambar 4.19 Hasil Deteksi dengan Model yang Dilatih dengan Ukuran Gambar 48x48

Meskipun mAP yang diperoleh pada pengujian dengan *dataset* FER2013 menghasilkan nilai yang tinggi, tetapi pada kenyataannya ketika model yang dilatih dengan ukuran gambar 48x48 diuji pada video asli memberikan performa yang buruk. Model sama sekali tidak mampu mendeteksi wajah siswa sehingga tidak mampu mendeteksi antusiasme mereka. Oleh karena itu, karena hasil yang didapatkan dari model yang dilatih dengan ukuran gambar 48x48 tidak mampu memenuhi tujuan penelitian, maka pada proses pelatihan selanjutnya ukuran gambar diubah menjadi 320x320 piksel.

Berdasarkan hasil pelatihan dan pengujian pada beberapa ukuran model dengan data latih gambar berukuran 320x320 piksel diperoleh hasil mAP tertinggi pada model dengan ukuran *nano*. Model ini kemudian diuji coba pada video pembelajaran daring sinkron yang diadakan dosen untuk memastikan apakah model mampu melakukan deteksi antusiasme atau tidak. Video yang digunakan untuk menguji model YOLOv8n yang dilatih dengan ukuran gambar 320x320 adalah video yang sama dengan video yang digunakan untuk menguji model YOLOv8n yang dilatih dengan ukuran gambar 48x48.

Berdasarkan hasil pengujian yang dapat dilihat pada Gambar, didapatkan hasil bahwa model YOLOv8n yang dilatih dengan ukuran gambar 320x320 mampu mendeteksi antusiasme siswa. Akan tetapi, terdapat batasan sistem yang baru diketahui ketika proses pengujian ini selesai dilakukan. Batasan tersebut adalah ketidakmampuan model untuk membedakan kondisi

siswa yang sedang menyalakan kamera dengan kondisi siswa yang sedang mematikan kamera. Sehingga ketika siswa tidak menyalakan kamera, tetapi mereka menggunakan foto profil atau menggunakan background yang berisi wajah, model tetap mendeteksi antusiasme mereka. Hal ini dapat dilihat pada Gambar 4.20 dengan tanda persegi panjang berwarna hijau. Namun, ketika siswa hanya menggunakan foto profil, model akan tetap mendeteksi antusiasme siswa, tetapi dengan *confidence score* yang cenderung statis. *Confidence score* adalah nilai yang menunjukkan seberapa yakin model bahwa *bounding box* (kotak pembatas) yang diprediksi berisi objek yang diminati. Ketika label deteksi dan *confidence score* tidak mengalami perubahan pada rentang waktu lebih dari lima menit, maka terdapat kemungkinan bahwa siswa tersebut tidak menyalakan kamera dan hanya menggunakan foto profil.

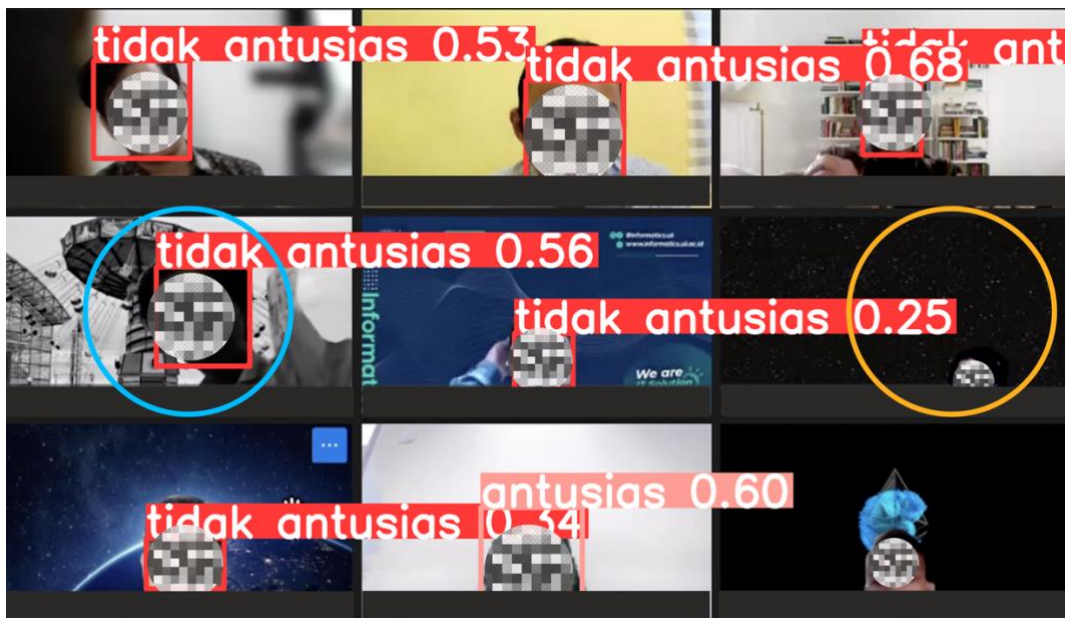


Gambar 4.20 Model Mendeteksi Foto Profil

*Confidence score* umumnya berkisar antara nol sampai dengan satu. Nilai tinggi atau mendekati angka satu berarti bahwa model sangat yakin *bounding box* berisi objek yang diminati. Pada penelitian ini, model dilatih untuk mendeteksi dua kelas yaitu antusias dan tidak antusias. Ketika *bounding box* menggambarkan tidak antusias dengan *confidence score* 0,71 berarti model sangat yakin bahwa *bounding box* berisi ekspresi tidak antusias. Sedangkan ketika *confidence score* hanya 0,44, berarti model kurang yakin bahwa *bounding box* berisi ekspresi tidak antusias dan tidak akan diklasifikasikan ke dalam kelas antusias. Hal ini terjadi

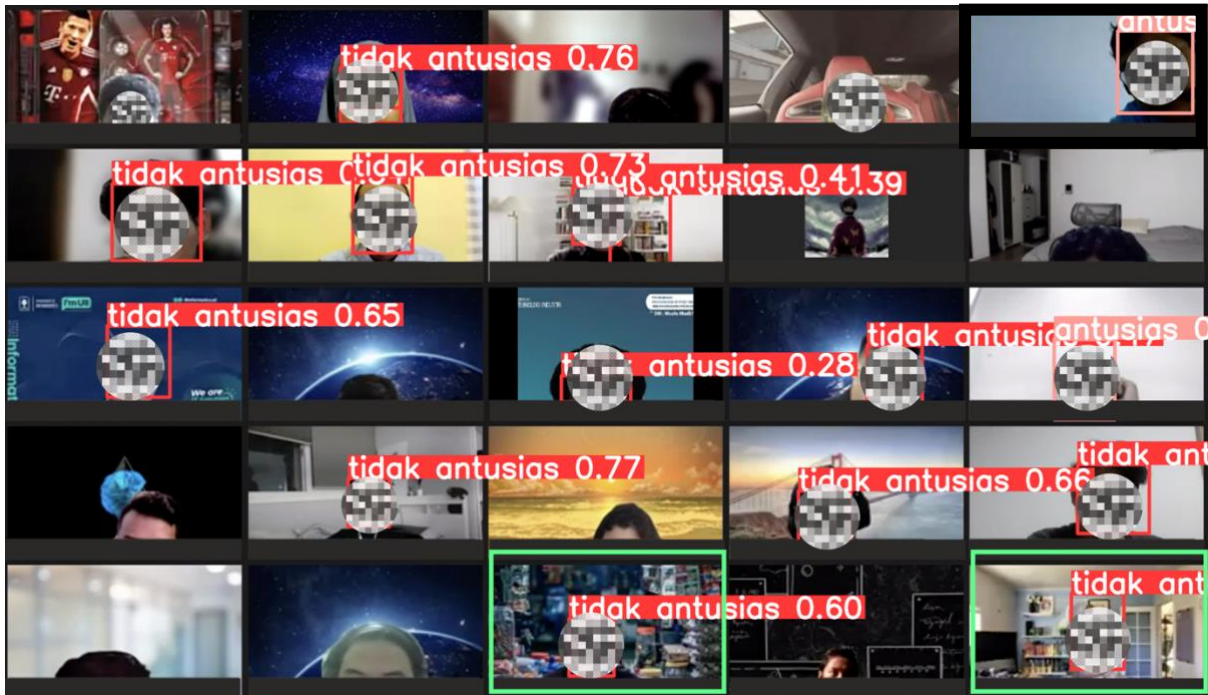
karena model *object detection* biasanya memiliki ambang klasifikasi untuk setiap kelas. Jika *confidence score* objek di bawah ambang batas, objek tersebut tidak akan diklasifikasikan ke dalam kelas apapun. Hal ini bertujuan untuk mengurangi deteksi palsu, karena model mungkin mendeteksi objek yang tidak jelas atau tidak sesuai dengan kelas yang dilatih.

Pada pengujian menggunakan video pembelajaran daring sinkron yang diadakan oleh dosen, diketahui bahwa ketika wajah siswa yang tertampil di layar terlalu kecil, model tidak mampu mendeteksi wajah tersebut seperti yang dapat dilihat pada Gambar 4.21 yang ditandai dengan lingkaran kuning. Di sisi lain, intensitas cahaya justru tidak terlalu berpengaruh. Model tetap dapat mendeteksi antusiasme sekalipun wajah yang tertampil di layar minim cahaya yang dapat dilihat pada Gambar 4.21 dengan lingkaran biru.



Gambar 4.21 Hasil Deteksi YOLOv8n yang dilatih dengan Gambar Ukuran 320x320

Kemudian, untuk menganalisis hasil prediksi, peneliti kembali meminta tolong lima partisipan yang terlibat dalam proses pelabelan untuk kembali memutuskan apakah hasil prediksi sudah tepat atau belum. Proses analisis dilakukan dengan menggunakan hasil tangkapan layar di mana mayoritas mahasiswa sedang menyalakan kamera. Perhitungan dilakukan dengan melakukan voting dan diambil suara terbanyak. Berdasarkan hasil voting, terdapat dua gambar yang dianalisis prediksinya tidak tepat. Dua gambar tersebut ditandai dengan persegi panjang berwarna hijau. Menurut lima partisipan gambar tersebut seharusnya dideteksi sebagai antusias. Hasil voting dapat dilihat pada Gambar 4.22.

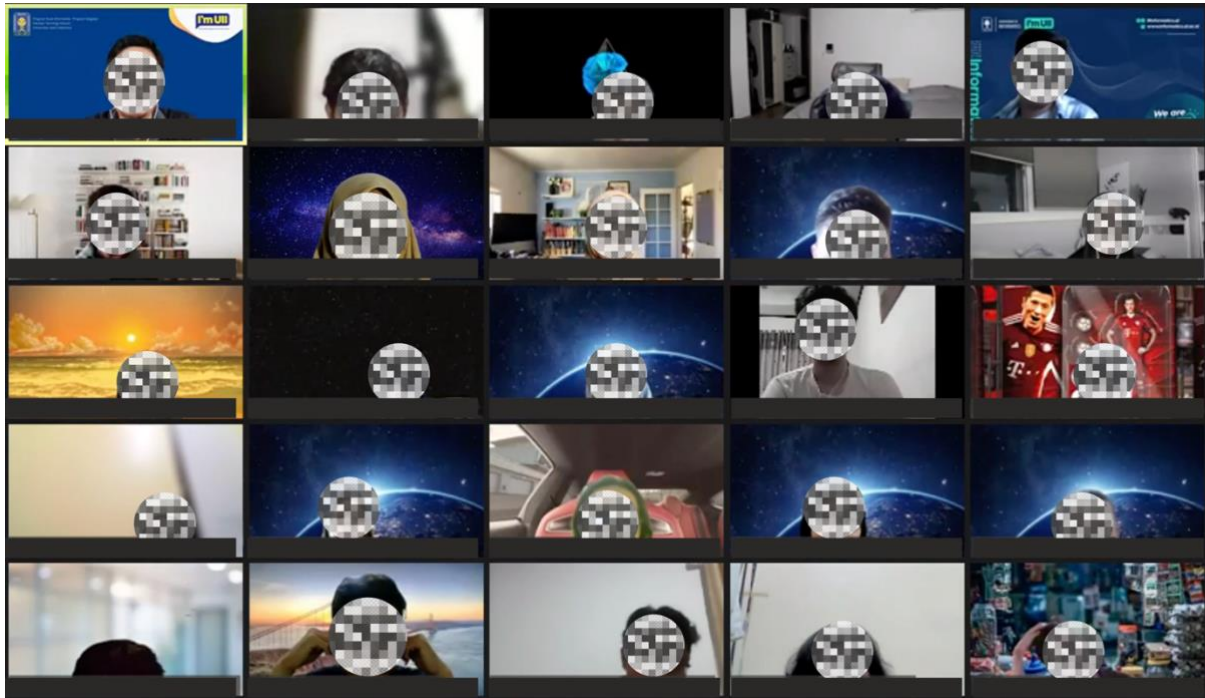


Gambar 4.22 Hasil Voting Analisis Ketepatan Prediksi

Selain itu, model YOLOv8n (320x320) akan diuji dengan tiga video pembelajaran untuk mengetahui respon antusiasme siswa selama proses pembelajaran. Hasil pengujian tersebut kemudian akan direkapitulasi. Rekapitulasi deteksi antusiasme akan dilakukan setiap 5 menit. Hal ini dilakukan karena menurut penelitian yang dilakukan oleh Bunce, D. M., Flens, dkk (2010) menunjukkan bahwa perhatian siswa sering berfluktuasi selama kuliah, dengan rentang perhatian penuh yang berlangsung sekitar 5-10 menit. Rekapitulasi ini dilakukan untuk memantau respon emosional siswa khususnya antusias dan tidak antusias dalam kelas daring.

Pengujian pertama menggunakan video dengan durasi 1,5 jam dengan jumlah wajah yang terlihat di layar sebanyak 25 wajah. Video pertama yang digunakan dapat dilihat pada Gambar 4.23. Rekapitulasi akan dilakukan setiap lima menit sehingga nantinya akan terdapat 18 data. Tabel rekapitulasi dapat dilihat pada





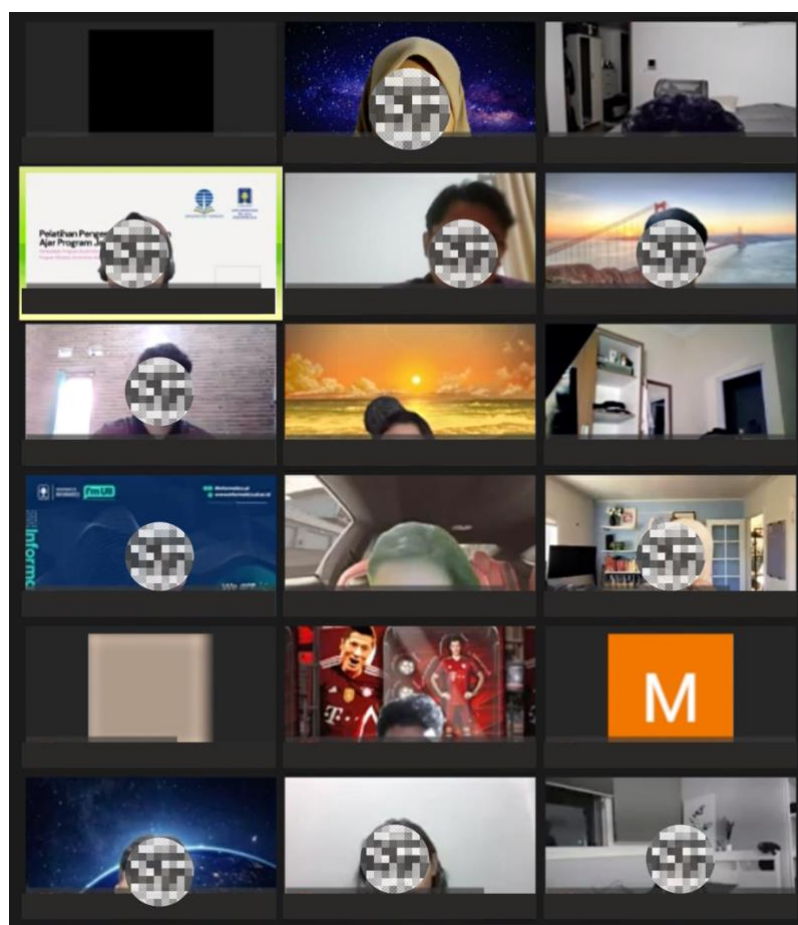
Gambar 4.23 Video Uji Pertama

Tabel 4.3 Rekapitulasi Video Uji Pertama

Menit ke-	Antusias	Tidak Antusias
5	10	1
10	4	14
15	5	9
20	2	14
25	6	13
30	4	7
35	2	9
40	6	15
45	4	12
50	1	12
55	3	13
60	4	10
65	5	15
70	4	15
75	8	13
80	3	17
85	16	2
90	4	15

Berdasarkan Tabel 4.3 diperoleh informasi bahwa pada menit ke-5 model mendeteksi 10 objek sebagai antusias dan satu objek sebagai tidak antusias. Menit ke-5 merupakan menit awal ketika kelas dimulai dan ketika pengajar memulai interaksi dengan siswa. Kemudian di menit-menit selanjutnya antusiasme lebih sedikit terdeteksi, karena setelah menit ke-5 pengajar lebih fokus menjelaskan materi dan minim interaksi dengan siswa. Lalu pada menit ke-75 model mendeteksi delapan objek sebagai antusias. Menit ini merupakan menit ketika dosen kembali membangun interaksi dengan siswa dengan melakukan sesi tanya jawab. Kemudian kembali terjadi penurunan antusiasme hingga pada menit ke-85 model mendeteksi 16 sebagai antusias. Menit ini merupakan menit ketika pengajar menyampaikan bahwa kelas telah selesai dan dibukakan sesi presensi untuk siswa.

Kemudian pengujian kedua menggunakan video dengan durasi satu jam dengan jumlah wajah yang tertampil sebanyak 18 wajah. Video kedua dapat dilihat pada Gambar 4.24. Sama seperti video pertama, rekapitulasi akan dilakukan setiap lima menit sehingga nantinya akan terdapat 12 data.



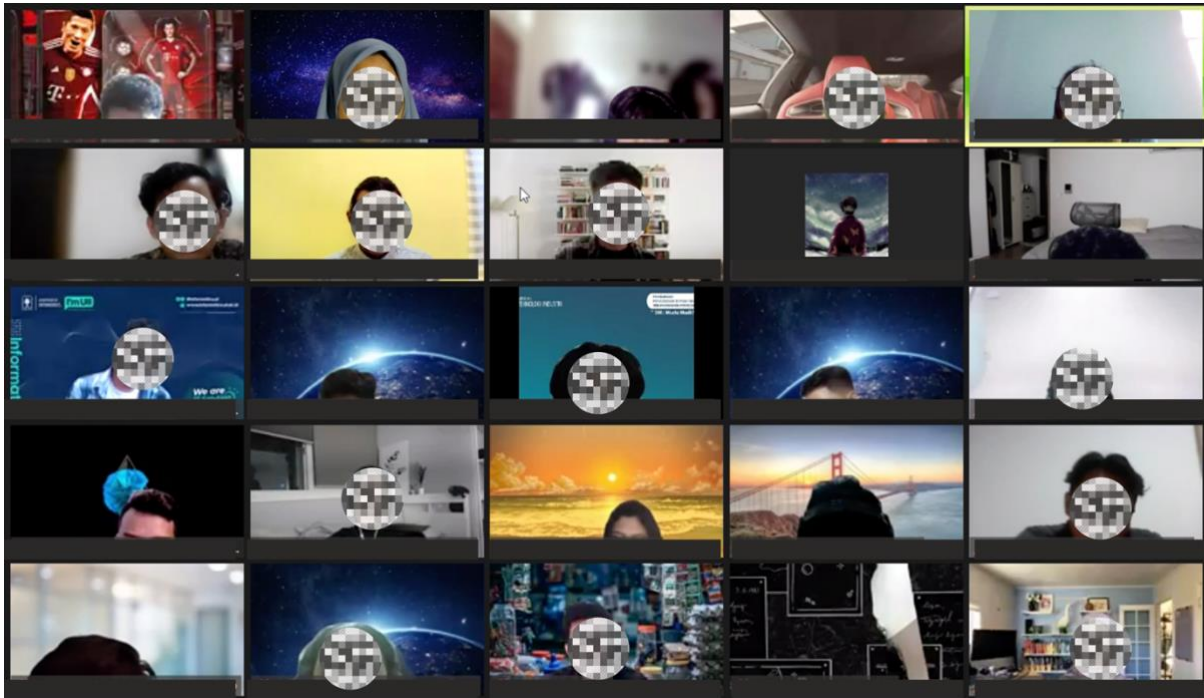
Gambar 4.24 Video Uji Kedua

Tabel 4.4 Rekapitulasi Video Uji Kedua

Menit ke-	Antusias	Tidak Antusias
5	8	4
10	4	9
15	2	11
20	8	10
25	5	10
30	7	11
35	5	9
40	4	11
45	2	13
50	4	10
55	14	3
60	2	14

Tabel 4.4 menunjukkan hasil rekapitulasi pengujian model YOLOv8n pada video pembelajaran daring. Berdasarkan pengujian, didapatkan informasi bahwa pada awal pembelajaran, yaitu pada menit ke-5, model mendeteksi delapan objek sebagai antusias. Kemudian terjadi penurunan pada 10 menit berikutnya hingga pada menit ke-20 model mendeteksi delapan objek sebagai antusias. Pada menit ini aktivitas yang terjadi adalah pengajar memanggil salah satu nama siswa yang dan melontarkan beberapa guyonan untuk mencairkan suasana kelas. Kemudian di menit-menit selanjutnya pengajar lebih fokus untuk menjelaskan materi dan sesekali berusaha tetap berinteraksi dengan siswa seperti pada menit ke-30, beliau berusaha bertanya kepada siswa tentang siapa karakter gim yang mereka suka, sehingga pada menit ini model mendeteksi tujuh antusiasme. Akan tetapi, ketika pengajar mulai menjelaskan materi dan sedikit berinteraksi, antusiasme siswa juga ikut menurun. Lalu pada menit ke-55, objek yang dideteksi antusias kembali tinggi dengan jumlah 14. Menit ini merupakan menit-menit akhir ditutupnya kelas, pengajar memberi salam dan membuka presensi untuk siswa.

Selanjutnya pada pengujian dengan video ketiga yang juga berdurasi satu jam dengan jumlah wajah yang tertampil sebanyak 25 wajah, diperoleh hasil seperti pada Tabel 4.5. Video yang digunakan pada pengujian ketiga ini dapat dilihat pada Gambar 4.25.



Gambar 4.25 Video Uji Ketiga

Tabel 4.5 Rekapitulasi Video Uji Ketiga

Menit ke-	Antusias	Tidak Antusias
5	4	9
10	5	9
15	3	11
20	2	13
25	3	17
30	1	18
35	2	9
40	4	8
45	1	16
50	5	9
55	4	11
60	2	9

Berdasarkan hasil rekapitulasi yang tertera pada Tabel 4.5 diperoleh informasi bahwa kelas daring yang diuji cenderung memberikan hasil deteksi tidak antusias yang dapat dilihat dari data banyaknya objek yang dideteksi antusias setiap lima menit memiliki nilai jauh lebih kecil dibanding tidak antusias. Hal ini terjadi karena ternyata ketika proses pembelajaran berlangsung, pengajar tidak melakukan interaksi yang cukup dengan siswa, pengajar hanya fokus menjelaskan materi, dan sempat beberapa kali keluar dari kelas dikarenakan jaringan

sinyal yang kurang bagus. Sehingga kondisi pembelajaran yang berlangsung saat ini terkesan membosankan. Oleh karena itu, deteksi yang dihasilkan oleh model menangkap lebih banyak tidak antusias dibanding dengan antusias.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

- a. Penelitian ini telah berhasil mengembangkan model YOLOv8 dengan mAP tertinggi yang mampu melakukan deteksi antusiasme siswa dalam proses pembelajaran daring dan mampu melakukan deteksi pada video pembelajaran daring.
- b. Pemilihan *hyperparameter* dengan *batch size* 32, *image size* 320x320, dan *epoch* 300 mampu menghasilkan mAP tertinggi pada model YOLOv8n.
- c. YOLOv8n mampu memberikan akurasi lebih tinggi mencapai 95,3%. Akurasi yang tinggi ini akan memberikan dampak yang lebih baik karena memungkinkan pengajar dapat lebih akurat dalam mengidentifikasi tanggapan emosional siswa terhadap materi pembelajaran.
- d. Model yang diuji dalam penelitian ini terdiri dari YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l, dengan hasil deteksi terbaik dihasilkan oleh YOLOv8n dengan [mAP@\[.5\]](#) 95,3%, presisi 0.888, dan *inference time* 69,2ms.
- e. Model YOLOv8n memiliki waktu inferensi paling cepat dibanding dengan ukuran yang lain dengan lama waktu yang dibutuhkan adalah 69,2ms. Waktu tersebut mencukupi untuk melakukan deteksi objek secara *real-time*, karena dapat dieksekusi dalam kurun waktu kurang dari 1 detik sehingga pengajar dapat dengan lebih cepat mengidentifikasi tanggapan emosional siswa.
- f. Model YOLOv8l memiliki [mAP@\[.5\]](#) lebih rendah dibanding dengan YOLOv8n, yaitu 94,1% dengan waktu inferensi yang jauh lebih lama yaitu 1302,8ms. YOLOv8l yang diklaim memiliki akurasi lebih bagus dibanding dengan YOLOv8n, ternyata pada penelitian ini menghasilkan mAP yang lebih rendah.
- g. Model tidak mampu mendeteksi kondisi ketika siswa sedang menyalakan kamera atau sedang mematikan kamera, model cenderung tetap mendeteksi foto profil sebagai wajah. Selain itu, model juga tidak mampu mendeteksi wajah pada video pembelajaran jika ukuran wajah yang tertampil sangat kecil.

## 5.2 Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran perbaikan yang dapat diimplementasikan pada pengembangan berikutnya. Beberapa saran tersebut adalah sebagai berikut:

- a. Meskipun YOLOv8n telah menunjukkan hasil yang baik, penelitian selanjutnya dapat mencoba model lain dengan mengoptimalkan hyperparameter. Tujuannya adalah untuk memperbaiki atau mengatasi potensi *overfitting* dan meningkatkan performa deteksi secara keseluruhan.
- b. Mengingat adanya indikasi *overfitting* karena perbedaan jumlah kelas antara *dataset* dan model, penelitian selanjutnya dapat lebih mendalam memeriksa strategi untuk mengatasi *overfitting*, seperti menggunakan teknik data augmentation atau penambahan data lebih lanjut pada kelas yang terbatas.
- c. Penelitian selanjutnya dapat memfokuskan pada implementasi model ini secara langsung dalam *platform* atau sistem pembelajaran daring untuk menguji kinerjanya dalam situasi nyata dan memahami dampak penggunaan model ini pada pengalaman belajar siswa secara keseluruhan.

## DAFTAR PUSTAKA

- Adhitama, G. (2022). *MODEL MACHINE LEARNING PENGENALAN EKSPRESI WAJAH SISWA MENGGUNAKAN METODE YOLO (YOU ONLY LOOK ONCE)*. Universitas AMIKOM Yogyakarta.
- Agrawal, A., & Mittal, N. (2020). Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. *Visual Computer*, 36(2), 405–412. doi: 10.1007/s00371-019-01630-9
- Allen, I. E., & Seaman, J. (2017). *Distance Education Enrollment Report 2017*.
- Anggraini, N. A., & Fadillah, N. (2019). Analisis Deteksi Emosi Manusia dari Suara Percakapan Menggunakan Matlab dengan Metode KNN. *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 3(2), 176–179. doi: 10.30743/infotekjar.v3i2.1041
- Azizi, F. N. (2021). *2021-Deteksi Emosi Menggunakan Citra*.
- Badrulhisham, N. A. S., & Mangshor, N. N. A. (2021). Emotion Recognition Using Convolutional Neural Network (CNN). *Journal of Physics: Conference Series*, 1962(1), 012040. doi: 10.1088/1742-6596/1962/1/012040
- C. Darwin and P. Prodger. (1998). *The expression of the emotions in man and animals*. Oxford University Press, USA, 1.
- Ekman, P. (2016). *Nonverbal Messages: Cracking the Code: My Life's Pursuit*.
- Elfwing, S., Uchibe, E., & Doya, K. (2017). *Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning*.
- Fernandes Junior, F. E., Nonato, L. G., Ranieri, C. M., & Ueyama, J. (2021). Memory-Based Pruning of Deep Neural Networks for IoT Devices Applied to Flood Detection. *Sensors*, 21(22), 7506. doi: 10.3390/s21227506
- Gholamalinezhad, H., & Khosravi, H. (2020). *Pooling Methods in Deep Neural Networks, a Review*.
- Goodfellow, I. J., Erhan, D., Luc Carrier, P., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shave-Taylor, J., Milakov, M., Park, J., ... Bengio, Y. (2015). Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64, 59–63. doi: 10.1016/j.neunet.2014.09.005
- Hakim, D. M., & Rainarli, E. (2019). Convolutional Neural Network untuk Pengenalan Citra Notasi Musik. *Techno.Com*, 18(3), 214–226. doi: 10.33633/tc.v18i3.2387



- Hanafi, Y. U., Pembimbing, D., Mardi, S., Nugroho, S., Rachmadi, R. F., Departemen, M. T., & Komputer, T. (n.d.). *DETEKSI PENGGUNAAN HELM PADA PENGENDARA BERMOTOR BERBASIS DEEP LEARNING*.
- Hao, W., & Zhili, S. (2020). Improved Mosaic: Algorithms for more Complex Images. *Journal of Physics: Conference Series*, 1684(1), 012094. doi: 10.1088/1742-6596/1684/1/012094
- Hao, Y., Pei, H., Lyu, Y., Yuan, Z., Rizzo, J.-R., Wang, Y., & Fang, Y. (2022). *Understanding the Impact of Image Quality and Distance of Objects to Object Detection Performance*.
- Hasan, M. A., & Lazem, A. H. (2023). *FACIAL HUMAN EMOTION RECOGNITION BY USING YOLO FACES DETECTION ALGORITHM*. Retrieved from <https://cajmtcs.centralasianstudies.org>
- I. Azhari. (2021). Implementasi Algoritma Convolutional Neural Network Dalam Deteksi Emosi Manusia. *EProsiding Teknik Informatika (PROTEKTIF) 2020, 1*, 112–118.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, 199, 1066–1073. doi: 10.1016/j.procs.2022.01.135
- Job, M. S., Bhateja, P. H., Gupta, M., Bingi, K., & Prusty, B. R. (2022). Fractional Rectified Linear Unit Activation Function and Its Variants. *Mathematical Problems in Engineering*, 2022, 1–15. doi: 10.1155/2022/1860779
- Jocher, G. (2020). *YOLOv5 by Ultralytics*. doi: 10.5281/zenodo.3908559
- Kim, Y. Bin, Kang, S. J., Lee, S. H., Jung, J. Y., Kam, H. R., Lee, J., Kim, Y. S., Lee, J., & Kim, C. H. (2015). Efficiently detecting outlying behavior in video-game players. *PeerJ*, 3, e1502. doi: 10.7717/peerj.1502
- Kim, D. H., Baddar, W. J., Jang, J., & Ro, Y. M. (2019). Multi-Objective Based Spatio-Temporal Feature Representation Learning Robust to Expression Intensity Variations for Facial Expression Recognition. *IEEE Transactions on Affective Computing*, 10(2), 223–236. doi: 10.1109/TAFFC.2017.2695999
- Lam, K. N., Nguyen, K.-N. T., Nguy, L. H., & Kalita, J. (2022). *Facial Expression Recognition and Image Description Generation in Vietnamese*. doi: 10.3233/faia210176
- Luh, G.-C., Wu, H.-B., Yong, Y.-T., Lai, Y.-J., & Chen, Y.-H. (2019). Facial Expression Based Emotion Recognition Employing YOLOv3 Deep Neural Networks. *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, 1–7. doi: 10.1109/ICMLC48188.2019.8949236

- Lundqvist, D., Flykt, A., & Öhman, A. (1998). *The Karolinska Directed Emotional Faces - KDEF, CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet, ISBN 91-630-7164-9.* (n.d.).
- M. Contributors. (2023). *YOLOv8 by MMYOLO*, <https://github.com/open-mmlab/mmyolo/tree/main/configs/yolov8>.
- Majeed, F., Khan, F. Z., Iqbal, M. J., & Nazir, M. (2021). Real-Time Surveillance System based on Facial Recognition using YOLOv5. *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, 1–6. doi: 10.1109/MAJICC53071.2021.9526254
- Ming-Hsuan Yang, Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 34–58. doi: 10.1109/34.982883
- Novita Wardhani, R., & Kartika Delimayanti, M. (n.d.). *ANALISIS PENERAPAN METODE KONVOLUSI UNTUK REDUKSI DERAU PADA CITRA DIGITAL.*
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning.*
- Ohoirat, A. L. E. (2020). *PENGEMBANGAN INSTRUMEN ANALISIS VIDEO UNTUK MENENTUKAN PROFIL KARAKTER CALON GURU DALAM MEMBUKA PELAJARAN DITINJAU DARI EKSPRESI WAJAH.*
- P. Ekman. (1994). Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique. *Psychological Bulletin*, 15(2), 268–287.
- P. Ekman and W. V. Friesen. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2), 124–129.
- Prasetyawan, D., & Informatika UIN Sunan Kalijaga Yogyakarta Jl Marsda Adi Sucipto Yogyakarta, M. (2020). Penentuan Emosi pada Video dengan Convolutional Neural Network. In *JISKa* (Vol. 5, Issue 1). MEI.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection.*
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. doi: 10.1109/CVPR.2016.91
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement.*

- Seandrio, A. L., Pratomo, A. H., & Florestiyanto, M. Y. (2021). Implementation of Convolutional Neural Network (CNN) in Facial Expression Recognition. *Telematika*, 18(2), 211. doi: 10.31315/telematika.v18i2.4823
- Terven, J., & Cordova-Esparza, D. (2023). A Comprehensive Review of YOLO: From YOLOv1 and Beyond. *ACM COMPUTING SURVEYS*, 4. Retrieved from <http://arxiv.org/abs/2304.00501>
- Tian, Y.-L., Kanade, T., & Cohn, J. F. (n.d.). *Recognizing Action Units for Facial Expression Analysis*.
- Uddin, Md. Z., Hassan, M. M., Almogren, A., Zuair, M., Fortino, G., & Torresen, J. (2017). A facial expression recognition system using robust face features from depth videos and deep learning. *Computers & Electrical Engineering*, 63, 114–125. doi: 10.1016/j.compeleceng.2017.04.019
- Wang, C.-Y., Liao, H.-Y. M., & Bochkovskiy, A. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. doi: 10.48550/arXiv.2207.02696
- Yang, S., Luo, P., Loy, C. C., & Tang, X. (2015). *WIDER FACE: A Face Detection Benchmark*.
- Yin, X., & Liu, X. (2018). Multi-Task Convolutional Neural Network for Pose-Invariant Face Recognition. *IEEE Transactions on Image Processing*, 27(2), 964–975. doi: 10.1109/TIP.2017.2765830
- Yusuf, A., Cahya Wihandika, R., & Dewi, C. (2019). *Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network* (Vol. 3, Issue 11). Retrieved from <http://j-ptiik.ub.ac.id>
- Zhong, H., Han, T., Xia, W., Tian, Y., & Wu, L. (2023). Research on real-time teachers' facial expression recognition based on YOLOv5 and attention mechanisms. *EURASIP Journal on Advances in Signal Processing*, 2023(1), 55. doi: 10.1186/s13634-023-01019-w