

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Ketika kita menjalani kehidupan sehari-hari, terutama ketika berbicara dengan orang lain, wajah sering digunakan untuk berkomunikasi. Ekspresi wajah individu menunjukkan emosinya. Salah satu ciri perilaku adalah ekspresi wajah. Wajah seseorang dapat mengungkapkan perasaan mereka, seperti kegembiraan, kesedihan, kemarahan, ketakutan, dan lainnya. Melalui ekspresi wajah, orang dapat memahami dan mengetahui perasaan orang lain. Tapi ketika emosi menjadi terlalu kuat, ekspresi wajah menjadi tidak jelas. (Agustinus, 2020). Wajah adalah penanda utama emosi manusia karena berhubungan dengan mereka. Perasaan atau efek yang mendorong perilaku atau respons individu terhadap suatu stimulus merupakan ciri emosi. Ekspresi wajah adalah cara paling umum untuk mengungkapkan emosi nonverbal dan dianggap sebagai bukti nyata apakah seseorang mengatakan yang sebenarnya atau tidak. Ekspresi wajah dan foto ekspresi wajah telah digunakan secara luas untuk mempelajari emosi dasar manusia lintas budaya.

Ekspresi wajah seringkali dikaitkan dengan emosi yang dirasakan makhluk hidup pada suatu waktu. Emosi yang berbeda mungkin muncul atau dirasakan dalam kondisi yang berbeda, seperti: Ekspresi kemarahan, kebahagiaan, ketakutan, kesedihan, dan keterkejutan. Situasi yang dialami tentunya mempengaruhi cara berpikir dan berperilaku sehingga menimbulkan

emosi. Dalam banyak kasus, orang ingin menunjukkan perasaan bahagia di depan orang yang tidak disukainya, namun terkadang tanpa sengaja mereka menunjukkan perasaan benci melalui ekspresi wajahnya. Oleh karena itu, sangat mudah untuk memprediksi apakah emosi yang dirasakan seseorang merupakan emosi yang benar. Kemarahan atau ketidaknyamanan. Sebaliknya dalam hal ini Banyak orang salah membaca ekspresi karena mereka hanya melihat ekspresi wajah. Tentu saja dalam hal ini perlu terus menerus mengamati perubahan ekspresi wajah orang tersebut. Namun hal ini tidak bisa Anda lakukan secara instan karena suatu saat Anda akan bosan dan menjadi tidak akurat.

Aplikasi pengenalan ekspresi wajah merupakan media yang dapat mendeteksi ekspresi dari ekspresi wajah. Hanya dengan menggunakan kamera ponsel atau kamera komputer, Anda sudah bisa melihat seperti apa objek yang ada di kamera tersebut. Aplikasi ini memungkinkan pengguna untuk menentukan jenis ekspresi berdasarkan keakuratan sistem aplikasi yang mereka bangun. Akurasi aplikasi berbeda dengan akurasi manusia.

Meskipun manusia dapat dengan mudah mengenali ekspresi wajah orang lain, teknologi saat ini berkembang sangat cepat, terutama dalam dunia *computer vision*. Dengan kata lain, komputer dapat dilatih untuk berpikir dan memahami manusia.

Banyak orang menyebutnya pembelajaran mesin. Ini merupakan subbidang kecerdasan buatan (AI).

Dalam hal ini, mengembangkan aplikasi pengenalan ekspresi wajah dengan menggunakan teknik *machine learning* yang merupakan bagian dari

kecerdasan buatan (AI), mungkin merupakan pilihan yang tepat. Aplikasi ini tidak hanya bisa diakses di ponsel, tapi juga di komputer yang dilengkapi kamera. Ada tiga komponen utama saat menggunakan aplikasi pengenalan ekspresi: pengenalan wajah, yang menentukan ada tidaknya fitur wajah, dan ekstraksi fitur wajah, yang menentukan fitur wajah penting dari input. Komponen terakhir adalah pengenalan ekspresi wajah. Input mengklasifikasikan kelas ekspresi tertentu.

Dengan demikian, di rancanglah “**Aplikasi Deteksi Ekspresi Pada Wajah Dengan Mesin Learning**” dimana dengan adanya aplikasi ini, diharapkan dapat membantu dalam beberapa hal seperti menentukan jenis jenis ekspresi pada wajah dengan ketelitian yang lebih akurat menyesuaikan dengan kecerdasan teknologi yang digunakan pembuat, sekaligus akan menjadi judul skripsi penulis untuk memenuhi syarat kelulusan.

### **B. Rumusan Masalah**

Berdasarkan latar belakang sebelumnya, maka didapatkan rumusan masalah yaitu bagaimana cara pembuatan aplikasi deteksi ekspresi pada wajah menggunakan mesin learning dan bagaimana hasil pembuatan aplikasi deteksi ekspresi pada wajah menggunakan mesin learning.

### **C. Tujuan Penelitian**

Tujuan dari penulisan penelitian ini yaitu untuk membuat program deteksi ekspresi pada wajah dengan mesin learning.

#### **D. Batasan Masalah**

Terdapat beberapa batasan masalah yang diberikan dalam penelitian ini oleh penulis yaitu:

1. Pendeteksi hanya dapat dilakukan dengan pencahayaan yang cukup.
2. Aplikasi yang dihasilkan hanya berfokus pada deteksi ekspresi pada wajah.
3. Jenis ekspresi yang di deteksi yaitu membelalak, terpejam, senyum, diam, senyum dan tertawa.

#### **E. Manfaat Penelitian**

Manfaat dari penelitian ini yaitu membuat sebuah aplikasi deteksi ekspresi pada wajah seperti yang diharapkan dengan adanya aplikasi ini dapat membantu dalam beberapa hal seperti menentukan jenis jenis ekspresi pada wajah dengan ketelitian yang lebih akurat menyesuaikan dengan kecerdasan teknologi yang digunakan pembuat

Aplikasi ini dapat mendeteksi ekspresi secara real-time, yang dapat digunakan dalam berbagai situasi, seperti dalam pelayanan pada pelanggan dimana Ini adalah aplikasi yang dapat digunakan untuk mendeteksi ekspresi pelanggan dan memberikan respon yang tepat. Aplikasi ini juga dapat digunakan untuk meningkatkan interaksi mesin dan manusia. Misalnya dalam pengembangan produk AI, aplikasi ini dapat digunakan untuk membuat produk yang lebih responsif terhadap ekspresi pengguna aplikasi.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. Kajian Hasil Penelitian Terdahulu**

Penelitian dengan kasus yang sama namun objek dan metode yang berbeda yaitu **Aplikasi deteksi ekspresi pada wajah dengan mesin learning** pada penelitian ini digunakan tiga penelitian terdahulu yang menjadi rujukan ilmiah yaitu:

1. Penelitian yang dilakukan oleh (Yusuf, 2023) menjelaskan sistem yang menggunakan backpropagation untuk mengidentifikasi ekspresi wajah secara real-time. Data yang dikumpulkan untuk penelitian ini berasal dari foto wajah dengan label emosi seperti sedih, senang, takut, marah, jijik, netral, dan terkejut. Mengolah dan menganalisis data foto wajah sesuai dengan tujuan penggunaan Merancang proses pelatihan dan pengujian, memutar model proses pelatihan, dan menguji sistem yang dibuat. Hasil penelitian ini meliputi pengukuran presisi, loss, MSE, dan recall dengan menggunakan teknik ekstraksi fitur backpropagation. Teknik ini menghasilkan nilai indeks tinggi dalam membedakan berbagai emosi wajah dari data yang dikumpulkan. Selain itu, penelitian ini menunjukkan bahwa teknik backpropagation untuk pengenalan emosi wajah berhasil mengembangkan sistem pengenalan emosi wajah; Model yang dibor teknik ini dapat mengidentifikasi emosi wajah dengan cukup akurat dan mampu mengklasifikasikan emosi wajah dengan baik.

2. Penelitian yang dilakukan oleh (Ihsan, 2021) merupakan alat komunikasi non verbal yang biasa digunakan oleh manusia untuk menyampaikan pesan sosial dan menggambarkan keadaan emosi atau perasaan. Penelitian ini memanfaatkan Tensorflow, perpustakaan pembelajaran mesin sumber terbuka yang menggunakan teknik jaringan saraf konvolusional (CNN). Metode ini dirancang khusus untuk mengenali dan mengklasifikasikan ekspresi wajah manusia yang dasar dan netral. Metode ini memiliki hasil yang paling signifikan untuk pengenalan gambar. Untuk meningkatkan kinerja model, data didistribusikan secara seragam. Hasil uji analisis menunjukkan bahwa parameter optimal adalah batch 32, epoch 100, dan dropout 0,6. Akurasi pelatihan 62,24 %, akurasi validasi 62,44%, training loss 4,54 %, dan validation loss 4,02 % .

3. Penelitian yang dilakukan oleh (Safiro, 2020) yaitu proses pengenalan wajah yang dilakukan secara *real time* melalui teknik *deep learning*. Hasil penerapan teknik *deep learning* yang diterapkan pada pengolahan pengenalan ekspresi wajah, tingkat akurasinya sebesar 85% pada tempat terang dan 70% pada tempat gelap. Pengenalan ekspresi wajah dapat dilakukan bahkan saat memakai kacamata, dengan akurasi 78% dalam kondisi pencahayaan terang. Berdasarkan hasil penelitian ini, ketika teknik *deep learning* diterapkan, angkanya mencapai 85%, namun di lingkungan gelap angkanya 70%, artinya wajah tidak dikenali di lingkungan gelap.

## B. Kajian Teori

### 1. Ekspresi Wajah

Wajah seseorang dapat memberikan banyak informasi. Wajah mereka dapat menunjukkan ketidakpercayaan diri, kesedihan, atau ketakutan. Seringkali kita tidak memiliki kemampuan untuk mengubah ekspresi wajah kita untuk menunjukkan emosi kita. Namun, dalam buku Joe Navarro "What Every Body Is Saying", gerakan otot di sekitar mata dan mulut yang disebut "Display Rules" menunjukkan area wajah yang menunjukkan pemikiran dan perasaan kita. (Amda & Fitriyani 2019).

Berikut beberapa gambaran ekspresi yang dapat kita baca dari wajah:

#### 1. Membelalak



**Gambar 2.1** Ekspresi Membelalak

Banyak orang yang membelalak sebagai tanda terkejut atau ketidakpercayaan terhadap sesuatu. Ini dapat dilihat dari mata yang bergerak dan melebar.

## 2. Senyum



***Gambar 2.2*** Ekspresi Senyum

Seseorang dapat tersenyum karena senang dengan sesuatu yang terjadi pada mereka. Gerakan bibir yang melebar ke samping dan tampilan mata menunjukkan senyum.

## 3. Diam



***Gambar 2.3*** Ekspresi Diam

Ketika seseorang melamun atau berpikir tentang sesuatu, mereka biasanya mengalami ekspresi diam. Ekspresi diam juga sering terjadi ketika seseorang sedang sedih dan tidak memiliki kata-kata untuk mengungkapkan perasaan

mereka. Seseorang yang diam dengan mata tidak fokus dan wajah datar dapat dibaca.

4. Tertawa



**Gambar 2.4** Ekspresi Tertawa

Ketika seseorang melihat sesuatu yang lucu di video atau dalam keadaan nyata, ekspresi tertawa biasanya ditunjukkan dengan mata yang membesar, bibir yang melebar ke samping, dan gigi yang terlihat.

5. Terpejam

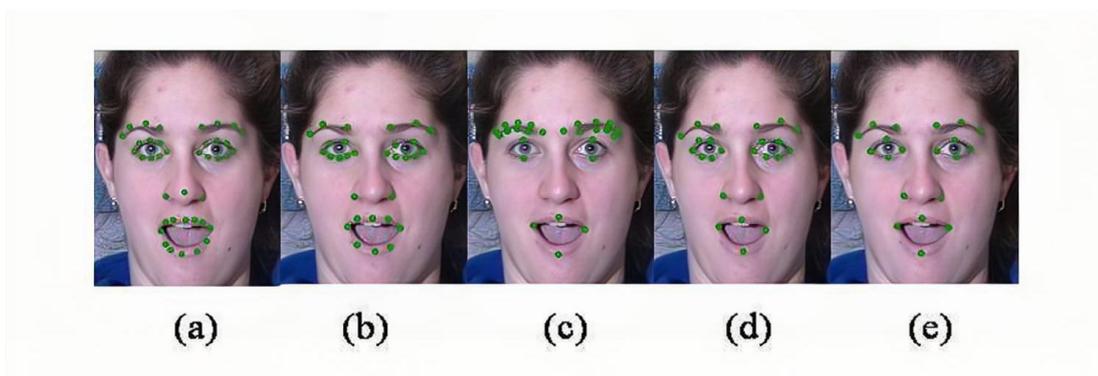


**Gambar 2.5** Ekspresi Terpejam.

Ekspresi terpejam biasanya ditunjukkan dengan mata tertutup dan bibir tidak bergerak; ini dapat terjadi pada orang yang tertidur atau ingin menenangkan diri.

## 2. Titik Deteksi

Wajah yang akan dikenali diidentifikasi melalui deteksi wajah. Dalam penelitian pengenalan wajah, berbagai algoritma digunakan untuk meningkatkan akurasi dan kecepatan pemrosesan serta memaksimalkan hasil pengenalan. (Puspanignrum & Saputra, 2018). Pelacakan area wajah harus selalu tepat agar pengamatan gerakan area wajah dapat akurat. Setelah wajah diidentifikasi dengan benar, memindai ujung hidung dan kemudian membentuk area wajah, sehingga area wajah tetap tepat meskipun kepala bergerak. (Roshiani & Choirina, 2020).



**Gambar 2.6** Contoh Titik Deteksi pada Wajah.

Identifikasi dan penandaan elemen penting di wajah seseorang dikenal sebagai deteksi titik di wajah. Mata, hidung, mulut, dagu, dan bagian wajah lainnya termasuk dalam kategori ini. Berbagai aplikasi yang menggunakan teknologi pengenalan titik wajah, seperti pengenalan wajah dan analisis ekspresi wajah, menggunakan algoritma dan teknik pemrosesan gambar untuk mengidentifikasi

dan mengatur karakteristik wajah. Ini memungkinkan efek visual interaktif pada wajah dan memahami ekspresi wajah.

### 3. *Machine Learning*

*Machine learning* merupakan bidang kecerdasan buatan yang berkembang sangat pesat untuk membantu memecahkan masalah klasifikasi, regresi, pengelompokan, dan *anomaly detection* dengan lebih efisien di berbagai bidang. (Heryadi, 2020).

*Machine learning* juga dikenal sebagai pembelajaran mesin, adalah metode AI yang sering digunakan untuk meniru atau menggantikan perilaku manusia dalam melakukan otomatisasi atau memecahkan masalah. (Ahmad, 2017).

Secara umum, proses machine learning dapat menjadi beberapa langkah, yaitu:

1. Pengumpulan data: Data digunakan sebagai bahan bakar untuk melatih model. Data ini dapat mencakup informasi yang dikumpulkan tentang masalah yang akan diselesaikan dari berbagai sumber.
2. *Preprocessing* data: Data harus diproses secara berkala sebelum dapat digunakan untuk melatih model. Prosedur ini termasuk pengelompokan, standarisasi, dan pembersihan untuk memastikan konsistensi dan kualitas data.
3. Pemilihan model: Pemilihan model adalah proses memilih arsitektur atau algoritma untuk menyelesaikan tugas tertentu. Proses ini tergantung pada jenis masalah yang dihadapi, seperti klasifikasi, regresi, atau pengelompokan.

4. Pelatihan model: Data yang telah diproses digunakan untuk melatih model. Ketika menemukan pola dalam data selama pelatihan, model mengubah parameter internalnya untuk meningkatkan kapasitas prediksi atau pengambilan keputusan.
5. Validasi dan evaluasi: Setelah pelatihan, model dievaluasi menggunakan data yang belum pernah dilihat sebelumnya (data validasi atau pengujian) untuk memastikan bahwa model dapat secara efektif menggeneralisasi data baru.
6. Pengoptimalan dan penyetelan (*Tuning*): Model dapat dibuat dengan mengubah parameter atau dengan menggunakan metode seperti regularisasi untuk mencegah overfitting jika diperlukan pengoptimalan tambahan.

#### **4. Pengolahan Citra Digital**

Pengolahan citra digital (*Digital Image Processing*) adalah bidang ilmu yang mempelajari metode pengolahan gambar. Gambar tersebut dapat berupa gambar bergerak (video) atau gambar diam (foto). Namun, dalam konteks ini, yang dimaksud dengan "digital" adalah pengolahan gambar yang dilakukan melalui komputer. Secara garis besar, pengolahan citra digital mengacu pada pengolahan data dua dimensi. Agar komputer digital dapat memproses suatu gambar, gambar tersebut harus direpresentasikan secara numerik menggunakan nilai diskrit. Digitalisasi gambar adalah proses mengubah fungsi kontinu menjadi nilai diskrit. Tujuan pengolahan gambar ini adalah meningkatkan kualitas gambar sehingga dapat diolah dengan mudah oleh manusia atau mesin komputer. Mengubah fungsi kontinu menjadi nilai diskrit dikenal sebagai digitalisasi gambar. Tujuan dari pengolahan citra adalah untuk meningkatkan kualitas gambar agar mudah diinterpretasikan oleh manusia dan mesin komputasi.

Digitalisasi adalah proses mengubah gambar, teks, atau suara dari sesuatu yang terlihat menjadi data elektronik yang dapat disimpan dan diproses kemudian. Pengolahan citra digital mempelajari bagaimana gambar diproses dalam konteks yang lebih luas, dengan fokus pada manajemen data dua dimensi. Gambar di sini mengacu pada gambar diam (foto) atau gambar bergerak (video).

Pengolahan citra digital adalah proses pengolahan gambar, terutama dengan menggunakan komputer untuk menghasilkan gambar berkualitas tinggi. Anda dapat menginterpretasikan gambar digital menggunakan fungsi dua dimensi  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat posisi dan  $f$  adalah amplitudo pada posisi  $(x,y)$ . Kumpulan nilai skala abu-abu yang disebut piksel membentuk gambar digital. Melewati gambar melalui grid untuk mengambil sampel atau memperoleh nilai koordinat diskrit  $(x, y)$  dan mengkuantisasinya merupakan langkah dalam proses digitalisasi gambar.

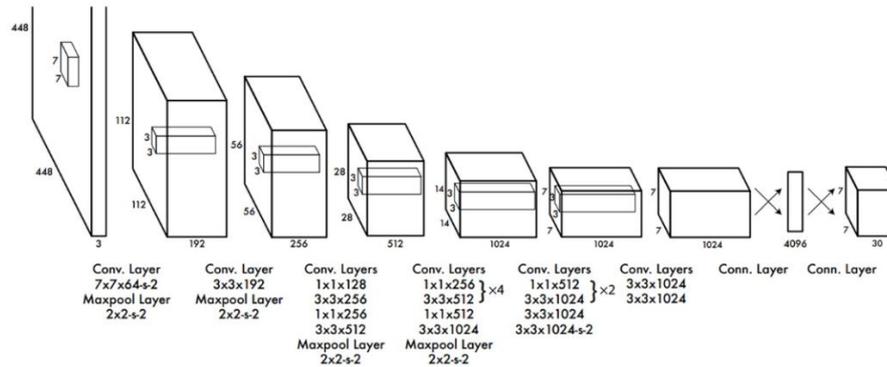
Pengolahan citra digital bertujuan untuk meningkatkan kualitas gambar atau memungkinkan untuk menggambarkan fiturnya. Kita diberkahi dengan kemampuan visual yang luar biasa, yang memungkinkan kita tidak hanya melihat, tetapi juga dengan cepat dan mudah mengidentifikasi objek yang terlihat dan membedakannya dari objek lain. Di sisi lain, komputer dan sensor pada dasarnya tidak dapat melakukan definisi tanpa menerapkan algoritma cerdas. Pembaca akan mendapatkan pemahaman awal tentang pengolahan gambar digital melalui buku Pengolahan Citra Digital ini. (Pengantar. (2020). (n.p.)).

## 5. *YOLO (You Only Look Once)*

Algoritma *YOLO (You Only Look Once)* adalah sistem deteksi objek yang menggunakan pembelajaran mendalam untuk menemukan objek secara real-time. Ini adalah sistem pemantauan end -to-end yang dapat secara langsung memprediksi kotak penghalang dan probabilitas kelas dari gambar penuh. Ini membuatnya jauh lebih cepat daripada sistem pendeteksian objek seperti Faster R-CNN. Algoritma *YOLO* sangat populer karena kecepatannya dan akurasi, dan digunakan dalam banyak aplikasi. Itu bekerja dengan kerangka kerja seperti Darknet dan Darkflow, dan memiliki kemampuan untuk menemukan objek dalam skenario baik statis maupun dinamis. (Zophie & Triharminto, 2022).

Karakteristik utama algoritma *YOLO* untuk deteksi target adalah kecepatan kalkulasi yang cepat dan ukuran model yang kecil. Struktur *YOLO* sangat sederhana dan dapat langsung menampilkan posisi dan kategori kotak pembatas melalui jaringan saraf. Karena *YOLO* hanya perlu memasukkan gambar ke dalam jaringan untuk mendapatkan hasil deteksi akhir, *YOLO* menggunakan global gambar untuk deteksi, yang dapat menyandikan informasi global dan mengurangi kesalahan mendeteksi latar belakang sebagai objek. *YOLO* dapat mempelajari sifat yang sangat umum untuk digunakan di bidang lain, yang membuatnya hebat dalam generalisasi. Dengan demikian, masalah deteksi target berubah menjadi masalah regresi; namun, deteksi akurasi masih perlu ditingkatkan. (Jiang et al., 2022)

1. Arsitektur algoritma *YOLO*



**Gambar 2.7** Jaringan Arsitektur *You Only Look Once*.

Dalam arsitekturnya, algoritma *YOLO* menggunakan *Convolutional Neural Network*. Terdapat 24 *convolutional layers* dan *fully connected layers* yang membantu mengekstraksi fitur dari gambar yang dimasukkan, dan *fully connected layers* membantu memprediksi probabilitas dan koordinat keluaran. Adapun beberapa struktur yang ada pada arsitektur algoritma *YOLO* yaitu:

a. Struktur Input

Algoritma *You Only Look Once* mengambil gambar input berukuran tetap dan membaginya menjadi grid sel. Tugas setiap sel adalah memprediksi kotak penghalang objek dan kelas objek yang ada di dalamnya.

b. Struktur Konvolusi

Algoritma *You Only Look Once* menggunakan lapisan konvolusi untuk mengekstrak fitur dari gambar yang dimasukkan; lapisan konvolusi membantu algoritma memahami konteks dan membedakan objek sesuai dengan fitur visual gambar.

c. Struktur Deteksi

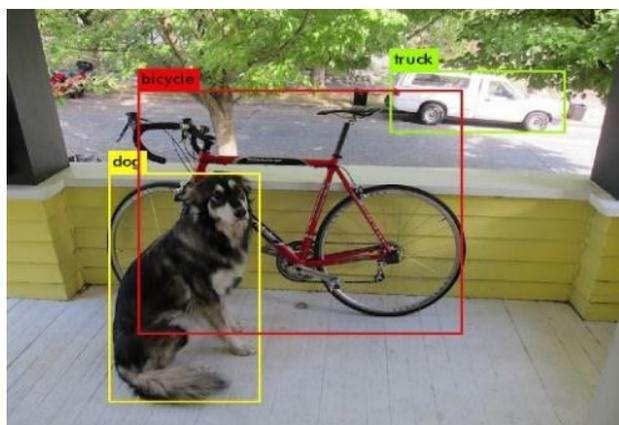
Setelah proses konvolusi, algoritma *You Only Look Once* mengidentifikasi kotak pembatas objek dan kelas objek dan menghasilkan beberapa kotak pembatas dengan tingkat kepercayaan yang sesuai untuk setiap sel dalam grid.

d. Struktur Output

Daftar kotak pembatas objek dengan label kelas dan tingkat kepercayaan adalah hasil akhir dari algoritma *You Only Look Once*. Untuk mendapatkan deteksi objek yang tepat, teknik non-maximum suppression digunakan untuk menghilangkan kotak-kotak yang tumpang tindih.

2. Proses prediksi Bounding Box

Metode YOLO menggunakan jaringan saraf pada sebuah gambar, membagi input gambar menjadi sejumlah grid, lalu menebak bounding box dan probabilitas untuk masing-masing grid. Gambar berikut menunjukkan bounding box untuk kelas dog, bicycle, dan truck.

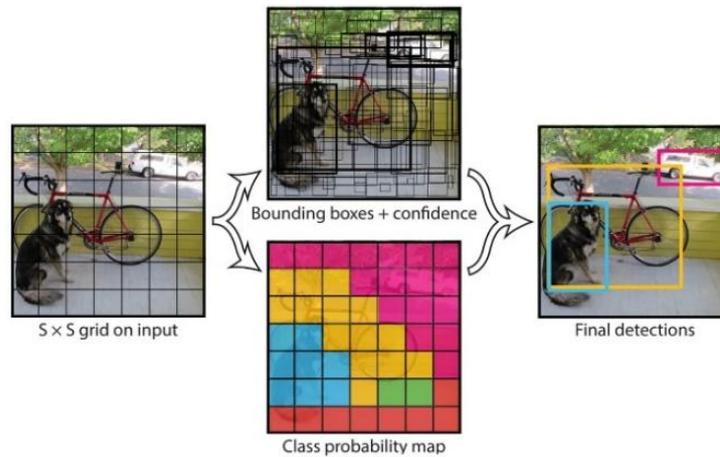


**Gambar 2.8** Bounding box kelas.

Proses konvolusi dari input gambar digunakan untuk mendapatkan bounding box. Hasilnya adalah ukuran bounding box sebesar  $SxSx(B*5+C)$ , di mana B adalah banyaknya bounding box dalam 1 grid dan C adalah banyaknya class yang dapat diidentifikasi. Nilai B dikalikan dengan 5 karena sebuah bounding box memiliki 5 nilai yang perlu disimpan di antaranya:

1. koordinat x
2. koordinat y
3. width (lebar)
4. height (tinggi), dan

*confidence score* (nilai probabilitas *bounding box* yang mendeteksi sebuah objek). Proses normalisasi mengubah semua atribut kotak batas, menghasilkan nilai antara 0 dan 1. Koordinat x dan y akan dinormalisasi dengan mengubah titik kiri atas grid yang bersangkutan, dan lebar dan tinggi juga akan dinormalisasi dengan mengubah ukuran gambar. Gambar berikut menunjukkan proses ini.



**Gambar 2.9** Proses *bounding box*.

### 3. Cara kerja algoritma YOLO

#### a. Preprocessing

Sebelum gambar dimasukkan ke dalam algoritma *You Only Look Once*, proses preprocessing dilakukan. Pada tahap ini, ukuran gambar diubah untuk menyesuaikan dengan input yang diterima oleh algoritma, dan nilai piksel gambar dinormalisasi untuk memastikan bahwa nilai tetap dalam rentang yang diharapkan.

#### b. Pembagian gambar menjadi Grid

Setelah tahap preprocessing selesai, gambar akan dibagi menjadi grid yang berbeda ukurannya. Setiap grid akan mendeteksi objek dan menghasilkan beberapa kotak pembatas untuk mengetahui posisi dan ukuran objek.

#### c. Prediksi dan NMS

Pada saat ini, algoritma *YOLO* akan memprediksi kemungkinan adanya objek dalam setiap *bounding box* yang dibuat oleh grid. *Non-Maximum Suppression*

(NMS) menghilangkan *bounding box* yang memiliki overlap yang tinggi untuk menghasilkan hasil deteksi yang lebih akurat dan mengurangi kemungkinan adanya *bounding box* yang tumpang tindih.

## 6. *Android*



**Gambar 2.10** Logo *Android*

*Android* adalah sistem operasi untuk perangkat seluler berbasis Linux yang berisi aplikasi, *middleware*, dan sistem operasi. *Android* awalnya dikembangkan oleh *Android Inc.*, sebuah perusahaan kecil di *Silicon Valley*. Pada tahun 2005, *Google* mengambil sistem operasi tersebut dan mendeklarasikannya sebagai sistem operasi "*Open Source*". Termasuk kode sumber yang digunakan untuk mengkompilasi sistem operasi tersebut, produk ini dapat digunakan secara gratis oleh siapa pun. Karena sifatnya yang terbuka bagi para *developer*, *Android* menjadi sistem operasi dengan pengguna terbanyak di dunia.

*Android* tidak membedakan antara aplikasi inti dan aplikasi pihak ketiga; sebaliknya, *Android* menawarkan berbagai lingkungan pengembangan untuk setiap aplikasi. *Application Programming Interface* (API) yang disertakan memungkinkan penghapusan aplikasi inti dan penggunaan aplikasi pihak ketiga

sebagai penggantinya. API juga memungkinkan anda mengakses perangkat keras, termasuk data sistem itu sendiri, dan ponsel cerdas.

Keuntungan utama *Android* adalah ia menawarkan pendekatan terpadu untuk pengembangan aplikasi, yang berarti aplikasi hanya perlu dibuat untuk Android dan berjalan di berbagai perangkat selama Android. Bagian terpenting dari rantai kesuksesan adalah aplikasi. (DeMarzio, 2017).

## 7. *Android Studio*



*Gambar 2.11 Logo Android Studio*

*Android Studio*, yang dikembangkan oleh *Google*, adalah *Integrated Development Environment* (IDE) yang lengkap yang dimaksudkan untuk mengembangkan aplikasi *Android*. Ini menawarkan berbagai fitur dan alat yang membantu pengembang membangun, menguji, dan menyebarkan aplikasi Android. Fitur-fitur ini meliputi:

1. Editor Kode: Dengan alat debugging, penyorotan sintaksis, dan penyelesaian kode, editor kode *Android Studio* memudahkan pengembang menulis dan memelihara kode.

2. Editor Tata Letak Visual: Dengan fitur editor tata letak visual IDE ini, pengembang dapat membuat aplikasi yang menarik secara visual dan ramah pengguna dengan menggunakan antarmuka seret dan lepas.
3. Integrasi dengan Layanan *Google*: Untuk meningkatkan kinerja aplikasi *Android*, *Android Studio* memungkinkan integrasi layanan *Google* seperti *Google Maps*, *Google Play Services*, dan *Google Cloud Platform*.
4. Alat Pengujian: *Android Studio* menawarkan alat untuk menguji aplikasi *Android*, seperti pengujian unit, pengujian UI, dan pengujian performa. Alat ini membantu pengembang memastikan bahwa aplikasi mereka stabil dan berkinerja baik.
5. Pembuatan Profil dan Pengoptimalan: Untuk membantu pengembang menemukan dan memperbaiki masalah kinerja aplikasi, IDE ini menawarkan alat pengoptimalan dan pembuatan profil.
6. Integrasi dengan *Android SDK* dan *NDK*: Dengan bantuan *Android Studio* dan *Android Software Development Kit* (SDK) dan *Native Development Kit* (NDK), pengembang dapat menulis kode dalam *Java*, *Kotlin*, atau *C++*.
8. Emulator dan Pengujian Perangkat: Dilengkapi dengan emulator, IDE ini memungkinkan pengujian aplikasi *Android* pada berbagai perangkat dan ukuran layar. Selain itu, dukungannya mencakup pengujian pada perangkat fisik yang terhubung ke mesin pengembangan.
9. Manajemen Proyek: Kontrol versi, otomatisasi pembuatan, dan manajemen ketergantungan adalah beberapa fitur *Android Studio* yang dapat digunakan untuk mengelola proyek *Android*.

10. Plugin dan Ekstensi: IDE ini mendukung banyak plugin dan ekstensi, yang memungkinkan pengembang mengubah lingkungan pengembangan mereka dan menambahkan fitur tambahan ke aplikasi mereka.

11. Integrasi dengan Google Cloud: *Android Studio* dan *Google Cloud Platform* memungkinkan pengembang mengirim dan mengelola aplikasi mereka ke cloud dengan mudah.

Secara keseluruhan, *Android Studio* menawarkan berbagai alat dan fitur untuk pengembangan aplikasi Android, menjadikannya favorit pengembang. (Mudadla et al., 2021).

## 7. ***Kotlin***

*Kotlin*, bahasa pemrograman tipe statistik yang dikembangkan oleh *JetBrains* pada tahun 2011, Ini adalah bahasa pemrograman yang menargetkan mesin *Java Virtual Machine (JVM)*, *Android*, *JavaScript*, dan *Native*. Bahasa ini cocok untuk pengembangan aplikasi *Android* yang berjalan pada platform *Java Virtual Machine (JVM)*.

## 12. **UML (*Unified Modeling Language*)**

UML (*Unified Modeling Language*) adalah alat desain sistem berorientasi objek. Secara filosofis, UML didasarkan pada konsep pemodelan *Object Oriented (OO)*, yang menganalogikan sistem ke dunia nyata yang didominasi oleh objek dan digambarkan atau dicatat dengan simbol yang cukup spesifik. proses yang independen dan standar.

Salah satu tujuan utama diagram UML adalah untuk membantu tim pengembangan proyek berkomunikasi, mengeksplorasi desain yang mungkin, dan memvalidasi penulis program atau desain arsitektur perangkat lunak. Tiga notasi sebelumnya membentuk komponen notasi UML: *Grady Booch*, *OOD (Oriented Object Design)*, *Jun Rumbaugh*, *OMT (Object Modeling Technique)*, dan *Ivar Jacobson*, *OOSE (Oriented Software Engineering)*.

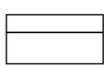
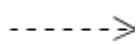
Tiga kategori utama diagram UML adalah diagram struktur, diagram perilaku, dan diagram interaksi. Setiap kategori memiliki diagram yang menjelaskan arsitektur sistem dan saling terintegrasi. Adapun daftar simbol UML yaitu :

**Tabel 2. 1** *Symbol Use Case Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
5		<i>Extend</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Association</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

NO.	GAMBAR	NAMA	KETERANGAN
7		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
8		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

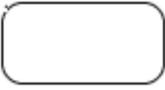
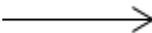
**Tabel 2. 2** *Symbol Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

**Tabel 2. 3** *Symbol Sequence Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi

**Tabel 2. 4** *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai <i>Link</i> pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

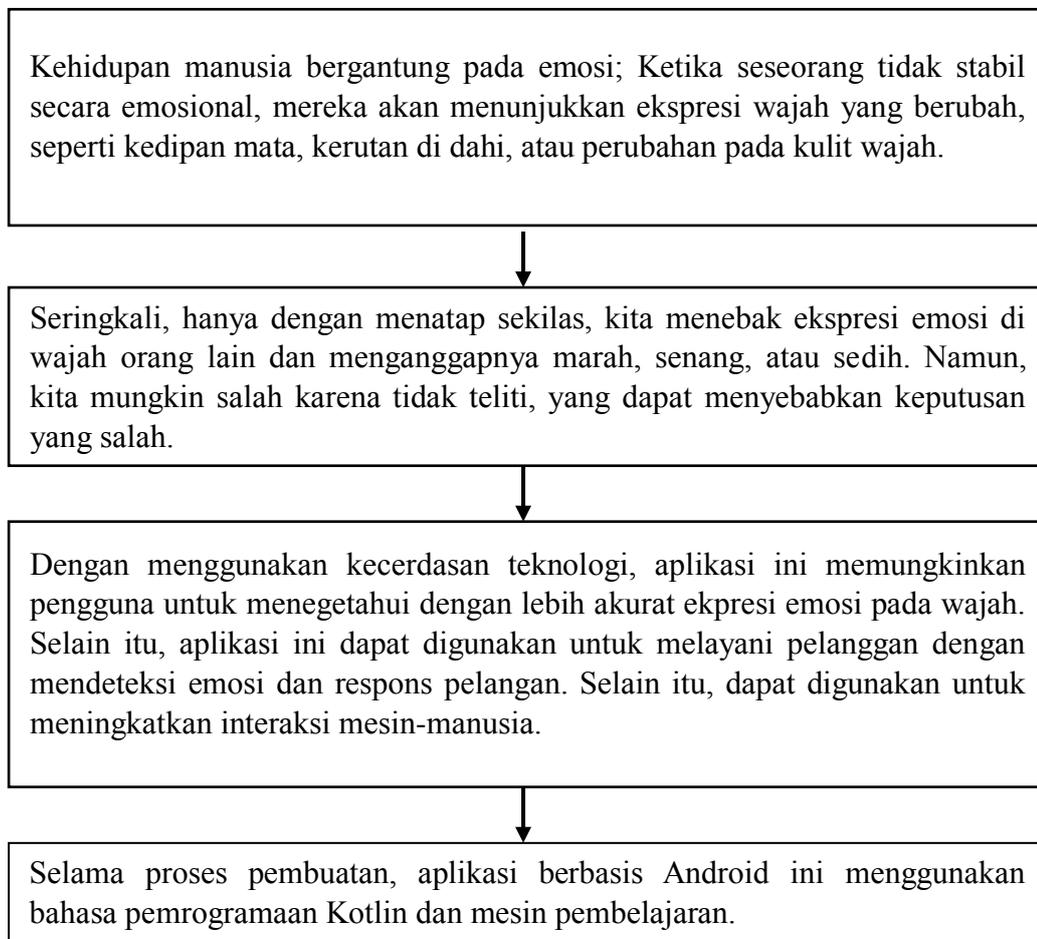
**Tabel 2. 5** *Symbol Activity Diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi

No.	GAMBAR	NAMA	KETERANGAN
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

### C. Kerangka Pikir

Untuk memahami alur penelitian diatas, diuraikan ke dalam kerangka berpikir yang akan disajikan dalam bentuk diagram ini:



## BAB III

### METODE PENELITIAN

#### A. Jenis Penelitian

Penelitian ini akan menggunakan metode penelitian dan pengembangan atau *research and development* (R&D).

#### B. Waktu Penelitian

Penelitian ini dilakukan untuk membuat aplikasi pendeteksi emosi pada wajah dengan kurung waktu selama  $\pm$  2 sampai 3 bulan di tahun 2024.

**Tabel 2. 6** Uraian Kegiatan Penelitian

No	Kegiatan	TAHUN 2024				
		Apr	Mei	Jun	Jul	Agust
1	Studi Literatur		■			
2	Perancangan Aplikasi		■			
3	Pembuatan Aplikasi		■	■		
4	Pengujian Aplikasi			■		
5	Hasil Pengujian			■	■	

#### C. Alat dan Bahan

##### 1. Alat

- a. Perangkat Keras (*Hardware*)

1) Laptop

ASUS VivoBook 14 M415DAO-FHD 323/AMD Ryzen 3-325 OU/8GB/526GB

2) *Android Mobile*

REALME 7i, RAM 8 GB, ROM 128 GB

b. Perangkat Lunak

1) *Android Studio*

2) *Kotlin*

## **2. Bahan**

Data berupa beberapa jenis ekspresi wajah yang akan dideteksi.

### **D. Teknik Pengumpulan data**

Adapun metode yang digunakan untuk menyelesaikan penelitian ini, yaitu sebagai berikut:

#### **1. Metode Pustaka**

Penelitian dengan metode pustaka, yang juga disebut studi atau tinjauan literatur, memerlukan pencarian dan evaluasi literatur yang relevan dengan topik penelitian. Metode ini tidak menggunakan data primer; sebaliknya, itu menggunakan sumber sekunder seperti buku, artikel jurnal, makalah konferensi, dan literatur lainnya.

#### **2. Metode Konsultasi**

Melakukan konsultasi dengan pihak-pihak terkait yang mengetahui lebih banyak hal-hal yang berhubungan dengan penelitian terutama dosen pembimbing

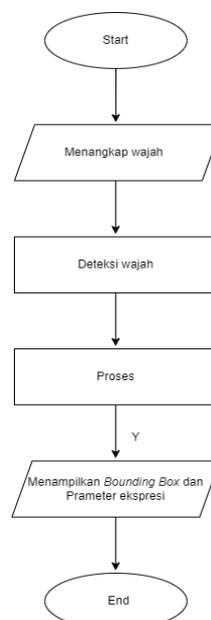
### 3. Metode Perancangan

Pada metode ini, tahap yang akan dibuat meliputi perencanaan, pembuatan rangkaian, layout komponen dan tata letak komponen.

#### E. Implementasi

Setelah data dikumpulkan, anotasi data dilakukan. Tujuan anotasi data adalah untuk membantu memberi label pada bentuk ekspresi yang sesuai dengan situasi atau kondisi sebenarnya pada data pelatihan. Selanjutnya, pelatihan data dilakukan dengan tujuan mengajar komputer untuk mengolah data beranotasi tersebut hingga membentuk karakteristik sebagai bahan pertimbangan untuk mencapai prediksi. Hasil perancangan dimasukkan ke dalam perangkat lunak yang dapat dibaca komputer untuk implementasi.

#### F. Diagram Alir

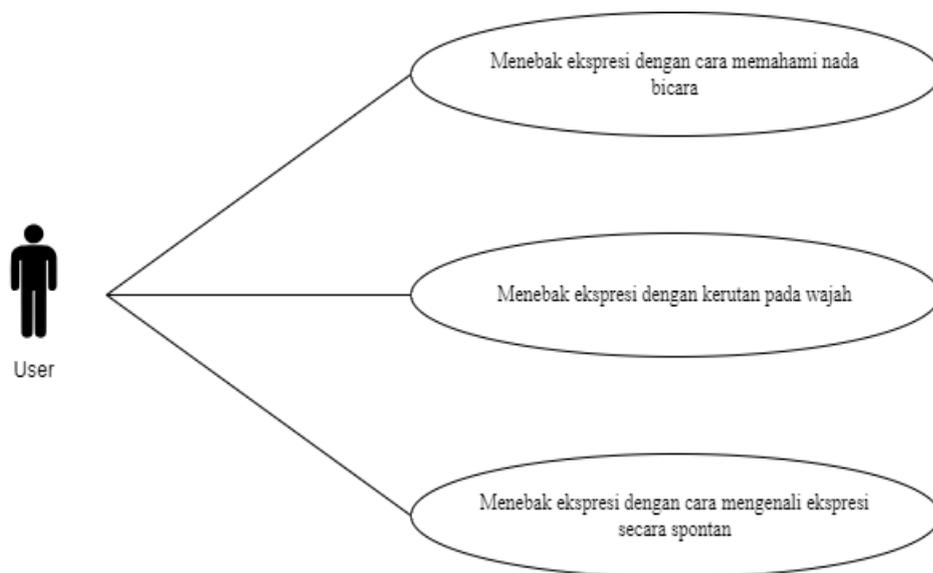


**Gambar 3.1** Flowchart

Pada Gambar 3.1 *Flowchart* tersebut menjelaskan secara singkat alur aplikasi pengenalan ekspresi wajah menggunakan pembelajaran mesin. Ketika pengguna membuka suatu aplikasi, mereka melihat tampilan aplikasi pengenalan yang umumnya menyerupai tampilan kamera ponsel. Pengguna dapat menyorot ekspresi wajah yang dikenali. Saat Anda mengarahkan kamera ke wajah Anda, akan muncul panel deteksi yang menampilkan informasi tentang ekspresi yang terdeteksi oleh aplikasi.

## G. Desain Sistem

### 1. Desain sistem yang berjalan

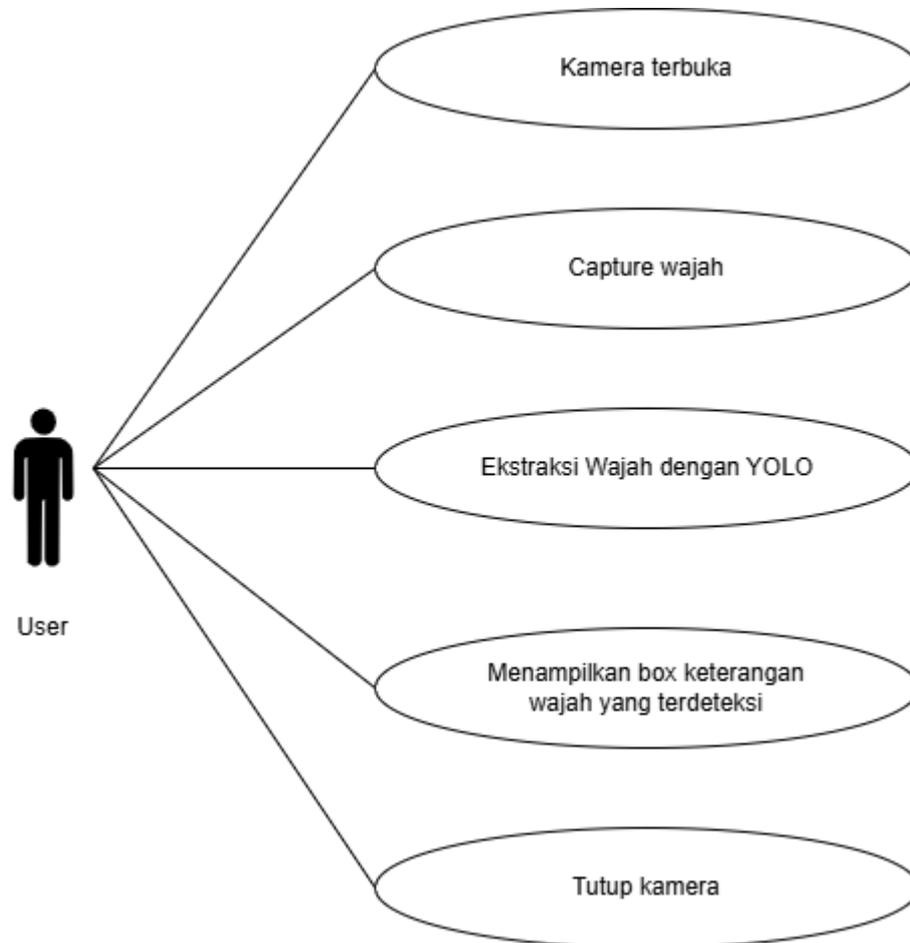


**Gambar 3.2** Usecase sistem yang berjalan.

Pada gambar *usecase* sistem yang berjalan saat ini, seseorang dapat melihat dan menebak ekspresi pada wajah seseorang secara langsung. Karena

salah perkiraan dan ketelitian yang dilakukan secara manual tanpa bantuan aplikasi, ada kemungkinan menebak ekspresi yang salah.

## 2. Desain Sistem yang Diusulkan



**Gambar 3.3** Usecase sistem yang diusulkan

Dalam *Usecase* sistem yang diusulkan, pengguna dapat melihat jenis ekspresi wajah berdasarkan persentase yang terdeteksi oleh aplikasi. Mereka dapat menggunakan aplikasi dengan membukanya dan menyoroti wajah dengan kamera atau perangkat lainnya yang dapat mengambil gambar, dan kemudian me aplikasi untuk melihat hasil yang terdeteksi dan ditampilkan olehnya.

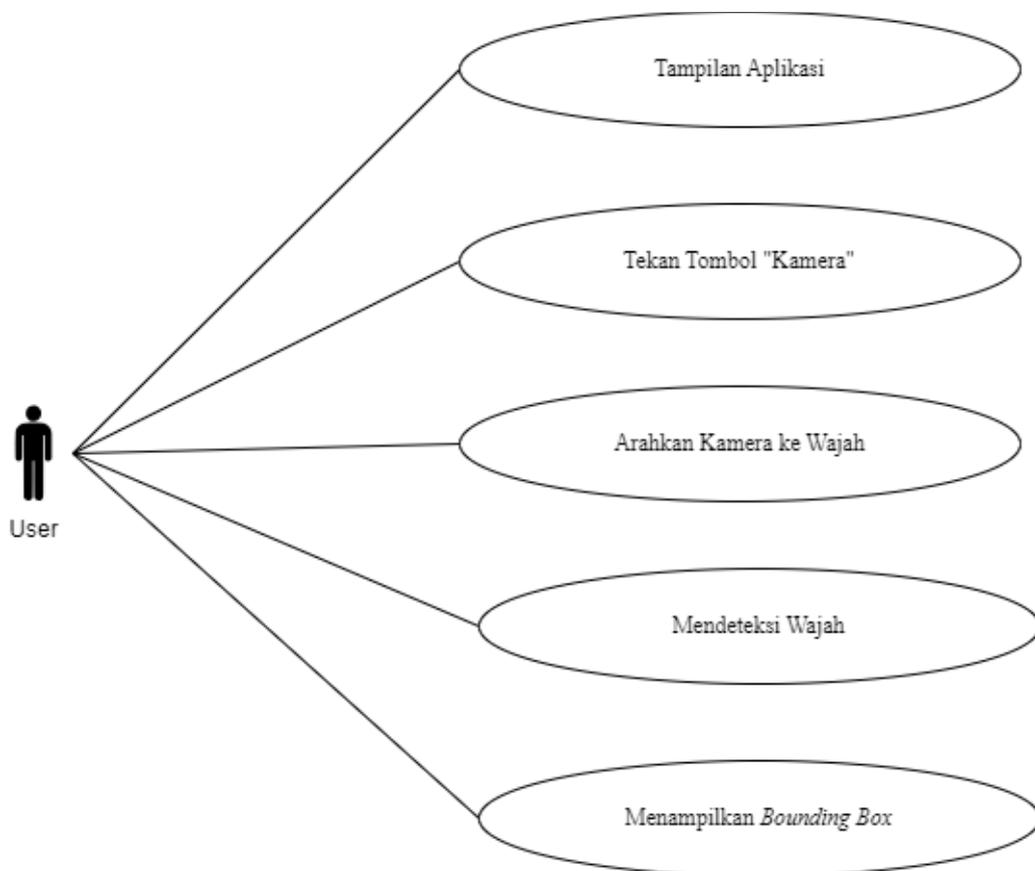
## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Analisis Aliran Data dengan UML

##### 1. *Usecase Diagram*

Jika dilihat dari sudut pandang orang yang tidak menggunakan sistem, *use case diagram* membantu menjalankan manfaat sistem.



**Gambar 4.1** Usecase Diagram.

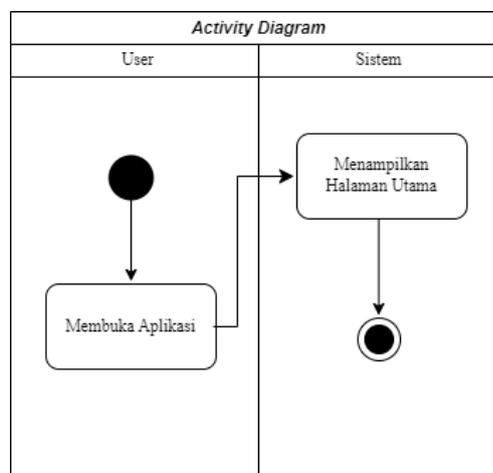
**Tabel 4.1** Tabel Keterangan *Usecase*.

<b>Nama <i>Usecase</i></b>	<b>Deskripsi <i>Usecase</i></b>
Tampilan Aplikasi	Proses pertama dalam mengakses aplikasi pendeteksi wajah
Tekan tombol “Kamera”	Merupakan tombol untuk berpindah halaman ke kamera untuk mendeteksi wajah
Arahkan Kamera ke Wajah	Proses memulainya untuk wajah
Mendeteksi Wajah	Proses pendeteksian wajah yang di sorot oleh kamera
Menampilkan <i>Bounding box</i>	Proses akhir dari pendeteksian wajah dengan menampilkan box keterangan atau bounding box yang di deteksi oleh kamera

## 2. *Activity Diagram*

*Activity Diagram* merupakan proses berisi penjelasan mengenai aktivitas aktivitas yang akan terjadi dalam sebuah aliran proses pada sistem.

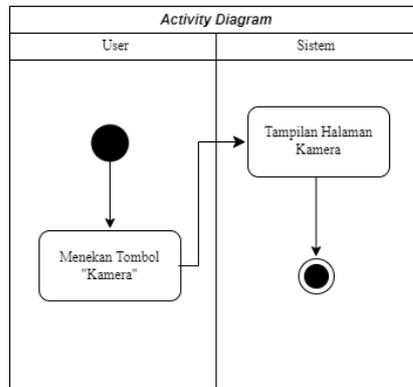
### a. *Activity Diagram* Membuka Aplikasi



**Gambar 4.2** *Activity Diagram* Membuka Aplikasi.

Tujuan dari "Activity Diagram Membuka Aplikasi" adalah untuk menunjukkan cara pengguna membuka aplikasi dan melihat halaman utama sistem.

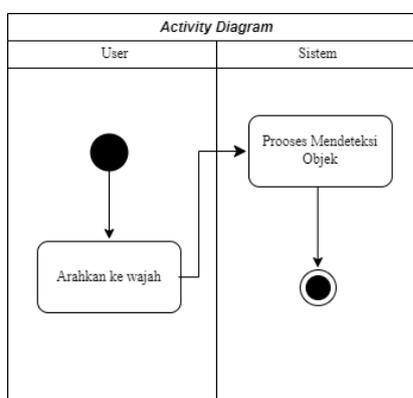
b. *Activity Diagram Menekan Tombol "Kamera"*



**Gambar 4.3** *Activity Diagram Menekan Tombol "Kamera"*

*Activity Diagram Menekan Tombol "Kamera"* memiliki tujuan untuk menggambarkan bagaimana user menekan tombol "Mulai" kemudian sistem akan menampilkan halaman kamera

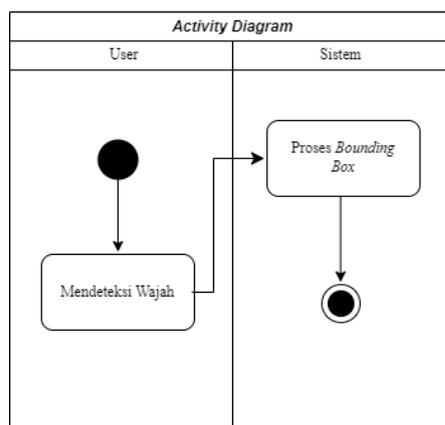
c. *Activity Diagram Mengarahkan ke Wajah*



**Gambar 4.4** *Activity Diagram Mengarahkan ke Wajah.*

*Actifity Diagram* Mengarahkan ke Wajah memiliki tujuan untuk menampilkan bagaimana pengguna dapat memulai untuk mendeteksi suatu objek dengan mengarahkan kamera ke objek yang akan di deteksi

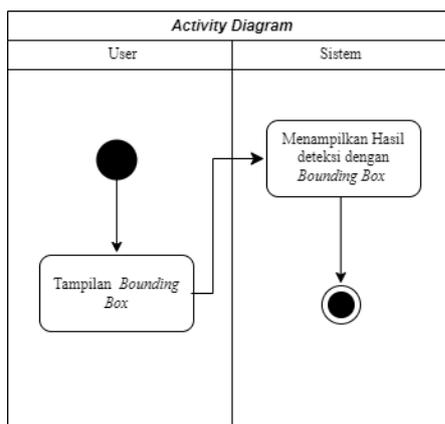
d. *Actifity Diagram* Mendeteksi Wajah.



**Gambar 4.5** *Actifity Diagram* Mendeteksi Wajah.

*Actifity Diagram* Mendeteksi Wajah memiliki tujuan unntuk mendeteksi wajah pengguna dengan cara pada halaman kamera yang ada di ponsel memiliki objek yang dapat dideteksi

e. *Actifity Diagram* Menampilkan *bounding box*



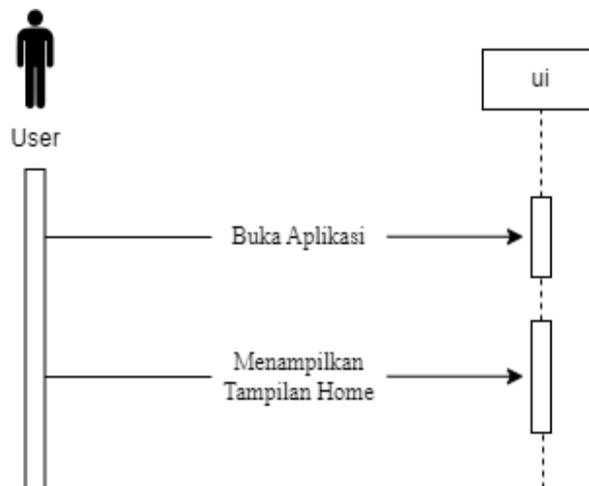
**Gambar 4.6** *Activity Diagram Menampilkan bounding box*

*Activity Diagram* Mengarahkan ke Wajah memiliki tujuan untuk menampilkan bagaimana pengguna dapat memulai deteksi suatu objek dengan mengarahkan kamera ke objek, dan kemudian sistem akan menampilkan hasil deteksi dengan *bounding box* beserta dengan keterangan persentase yang terdeteksi.

### 3. *Sequence Diagram*

Di bawah ini adalah *sequence* diagram yang menunjukkan aliran antara objek yang membentuk proses.

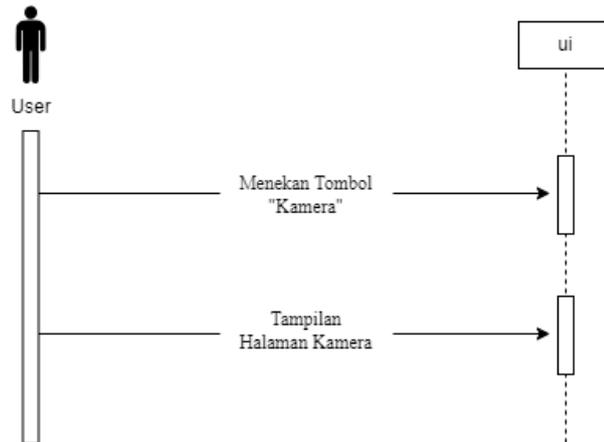
#### a. *Sequence Diagram Membuka Aplikasi*



**Gambar 4.7** *Sequence Diagram Membuka Aplikasi*

Tujuan dari *sequence* Diagram Membuka Aplikasi adalah untuk menunjukkan bagaimana pengguna meluncurkan program dan bagaimana antarmuka pengguna menampilkan halaman utama.

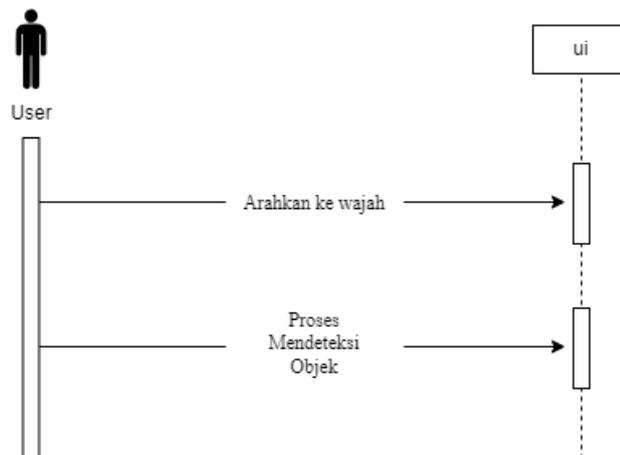
#### b. *Sequence Diagram Menekan Tombol “Kamera”*



**Gambar 4.8** Sequence Diagram Menekan Tombol “Kamera”

Sequence Diagram Menekan Tombol Kamera memiliki tujuan menampilkan halaman kamera kepada pengguna

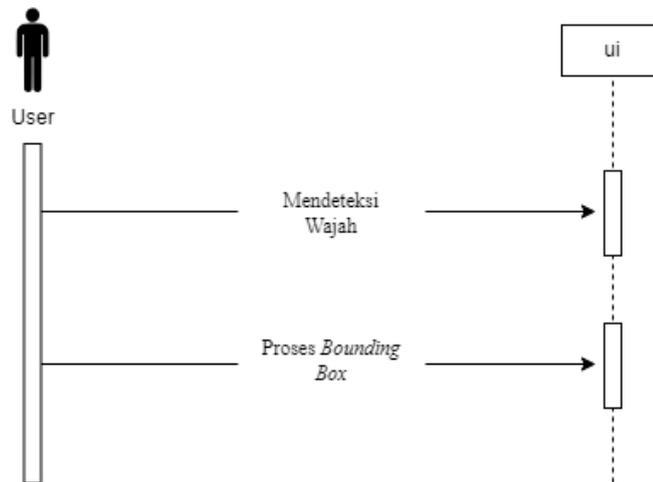
c. Sequence Diagram Mengarahkan ke Wajah



**Gambar 4.9** Sequence Diagram Mengarahkan ke Wajah

Sequence Diagram mengarahkan ke wajah memiliki tujuan untuk memberi arahan kepada pengguna untuk mengarahkan kamera ke wajah agar sistem dapat memulai proses mendeteksi objek.

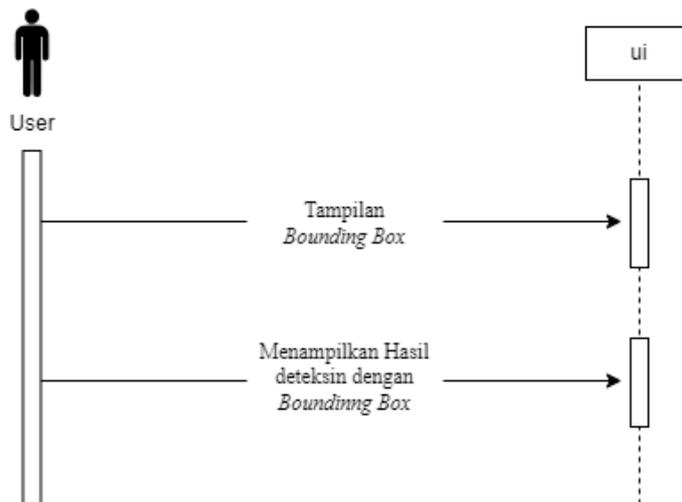
d. Sequence Diagram Mendeteksi Wajah.



**Gambar 4.10** Sequence Diagram Mendeteksi Wajah.

Sequence Diagram Mendeteksi Wajah bertujuan untuk menampilkan proses sistem dalam menampilkan *Bounding Box* beserta dengan keterangan objek yang terdeteksi kepada pengguna.

e. Sequence Diagram Menampilkan *bounding box*



**Gambar 4.11** Sequence Diagram Menampilkan *bounding box*.

Sequence Diagram Menampilkan *bounding box* memiliki untuk menampilkan bagaimana sistem akan memunculkan tampilan hasil dari yang

terdeteksi pada sistem. Hasil yang terdeteksi akan ditampilkan dengan *Bounding Box* dan persentase keterangan

### **B. Metode YOLO ( You Only Look Once ).**

*YOLO* adalah algoritma deteksi objek *real-time* yang menggunakan *convolutional neural network* (CNN). Algoritma ini bekerja dengan membagi *frame* atau gambar menjadi beberapa daerah, kemudian memberi mereka grid dan memprediksi probabilitas kotak untuk setiap daerah.

#### a. Dataset

Penelitian ini menggunakan dataset kumpulan gambar yang diperoleh melalui *Google*. Dataset tersebut juga dilengkapi dengan label yang terdiri dari empat kelas emosi: senang, sedih, marah, dan terkejut.

**Tabel 4.2** Kelas Dataset

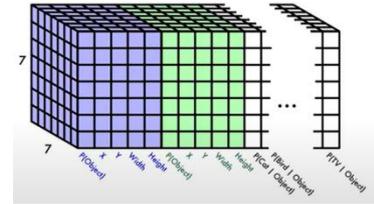
<b>Label</b>	<b>Jenis Ekspresi</b>	<b>Train</b>
0	Membelalak	50
1	Terpejam	50
2	Terdiam	50
3	Tersenyum	50
4	Tertawa	50

#### b. Processing

*Image classification* biasanya berukuran 224 x 224, tetapi objek kecil mungkin tidak terdeteksi karena detailnya tidak terlihat dengan jelas. Oleh karena itu, ukuran gambar diubah menjadi 448 x 448 x 3 pada algoritma *YOLO*.



➔ **YOLO**



**Gambar 4.12** Input Wajah

448x448x3

**Gambar 4.13** Lapisan Grid

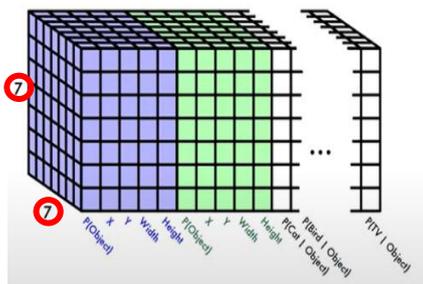
7x7x30

448x448 = Ukuran spatikal Gambar

x3 = RGB pada gambar

7x7 = Ukuran grid deteksi

x30 = Jumlah prediksi yang dihasilkan oleh setiap sel dalam grid



**Gambar 4.14** Lapisan Grid

Dari lebar x tinggi 7 x 7 pada gambar di samping bisa dilihat seolah olah ada grid 7 x 7 pada input yang dimasukkan, masing masing sel pada grid akan memprediksi bounding box beserta dengan confidence nya contoh seperti gambar berikut.

Bounding box #1, Confidence #1

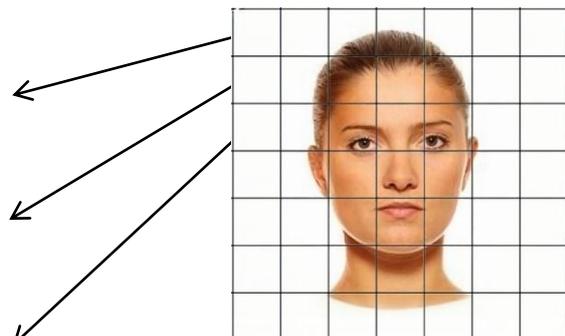
Bounding box #2, Confidence #2

Bounding box #1, Confidence #1

Bounding box #2, Confidence #2

Bounding box #1, Confidence #1

Bounding box #2, Confidence #2



**Gambar 4.15** Grid Wajah.

Bentuk Output;

Confidence #1 = channel output 1

Bounding box #1 = channel output 2 hingga 5

Confidence #2 = channel output 6

Bounding box #1 = channel 7 hingga 10

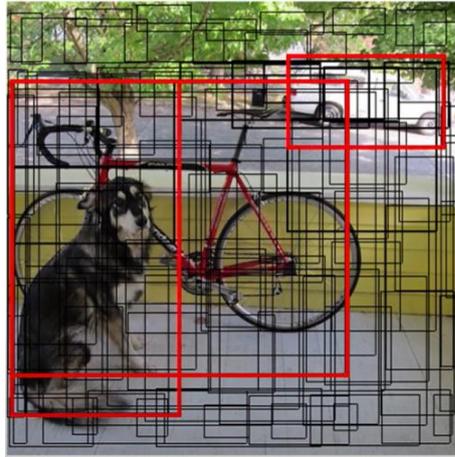
Setiap sel memiliki total 2 prediksi bounding box, total prediksi yang didapat yaitu;

$7 \times 7 \times 2 = 98$  Bounding box.

*Bounding box* yang diterima divisualisasikan; 98 dari mereka akan mendapatkan keyakinan yang akurat, yang ditunjukkan dengan garis yang lebih tebal. Contohnya dapat dilihat pada gambar di bawah.

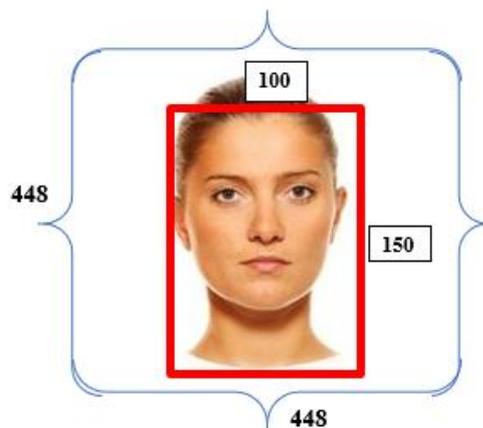


**Gambar 4.16** confidence *Bounding Box*



**Gambar 4.17** Confidence Bounding Box.

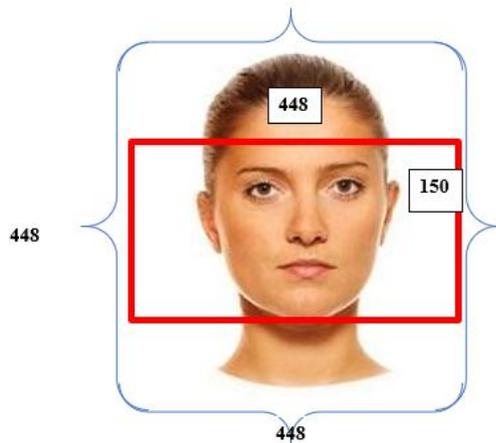
Untuk nilai  $x$ ,  $y$ ,  $width$  dan  $height$  *bounding box* nya dinormalisasikan ke 0 hingga 1. Cara menormalisasikan nilai *width* dan *height* yaitu menormalisasikan dengan tinggi dan lebar gambar. Tinggi dan lebar gambar yaitu 448x448 memprediksi bahwa kotak batas dengan lebar 100pixel dan tinggi 150pixel akan dinormalisasikan.



**Gambar 4.18** Normalisasi Bounding Box

$$W = 100/448 = 0.22$$

$$H = 150/448 = 0.33$$



**Gambar 4.19** Normalisasi *Bounding Box*

$$W = 448/448 = 1$$

$$H = 150/448 = 0.33$$

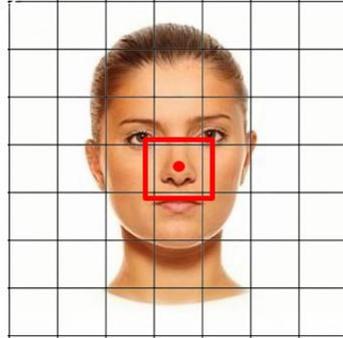
Selanjutnya, koordinat x dan y dihitung dari setiap sel yang memiliki *bounding boxnya* sendiri. Ini ditunjukkan pada gambar di bawah, di mana titik sel terletak pada titik yang relevan dan dapat diidentifikasi. Nilai x dan y pada sel tersebut kemudian dinormalisasi dari 0 hingga 1 dengan mengukur posisi kiri atas sel yang relevan. Tinggi dan lebar gambar 448 x 448 dikalikan 7 kali dari 7 grid dengan ukuran 64 x 64 sel. Kita dapat mengetahui koordinat titik kiri atas sel yang bersangkutan 320 untuk titik x dan 64 untuk titik y, sehingga hasilnya 320 dari 64 x 5 dan titik tengah di bounding box 365 untuk x dan 102 untuk y, dan kemudian dinormalisasi

$$X = (365-320) = 45/64$$

$$= 0.7$$

$$Y = (102-64) = 38/64$$

$$= 0.6$$



**Gambar 4.20** Titik Koordinat Wajah.

c. Klasifikasi

Logika klasifikasi *YOLO* menggunakan kotak batas yang bergerak melalui seluruh gambar dan memindainya hingga menghasilkan model klasifikasi yang telah dimasukkan ke *dataset*.

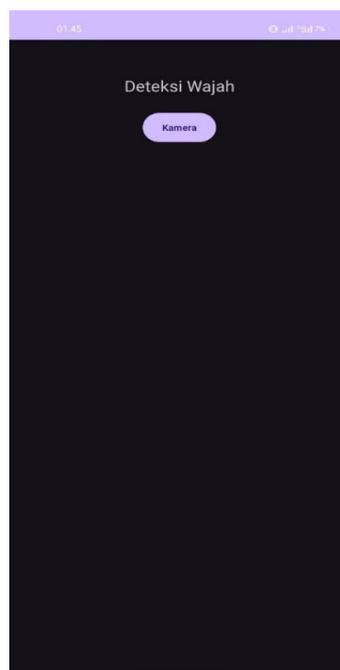


**Gambar 4.21** Klasifikasi wajah.

### C. Perancangan *Input-Output*.

#### 1. Tampilan Awal

Pada tampilan layar yang mirip dengan gambar di bawah ini akan muncul pada tampilan awal pendeteksi wajah. Ada beberapa elemen yang terlihat, termasuk latar belakang berwarna hitam dengan kalimat “Deteksi Wajah” disertai dengan *button* yang bertuliskan “Kamera”



**Gambar 4.22** Tampilan Awal.

#### 2. Halaman Kamera



**Gambar 4.23** Halaman Kamera.

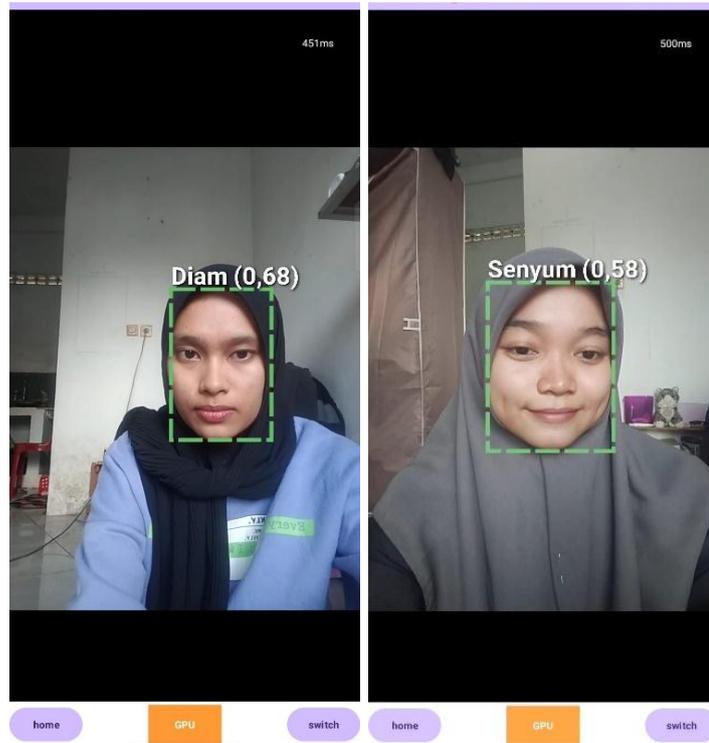
Setelah pengguna menekan *button* kamera pada halaman awal akan berpindah menampilkan halaman kamera serta mengaktifkan fitur kamera secara otomatis pada ponsel. Pengguna dapat mengarahkan kamera ponsel ke wajah menggunakan halaman kamera, kemudian aplikasi akan memulai mendeteksi ekspresi pada wajah berdasarkan langkah-langkah berikut:

- a. Dengan keadaan pencahayaan yang cukup, aplikasi dapat mendeteksi ekspresi wajah sebagai objek. Setelah itu, gambar akan diubah menjadi tensor, yang akan digunakan sebagai input untuk model.
- b. Menggunakan *TensorFlow* untuk membuat model deteksi objek *YOLO* dengan banyak layer konvolusi dan *pooling*. Model ini telah disiapkan sebelumnya melalui pelatihan pada dataset yang mengandung ekspresi terbelalak, terdiam, tersenyum, terpejam dan tertawa

- c. Praproses gambar diuji sesuai dengan spesifikasi model setelah model dibuat. Persiapan dapat mencakup mengubah ukuran gambar, mengubah nilai piksel, atau melakukan persiapan tambahan yang ditentukan oleh model.
- d. Untuk menjalankan gambar yang diproses melalui model, gunakan metode serupa. Model akan memprediksi batas kotak yang mengandung objek dan persentase ekspresi pada gambar.
- e. Langkah berikutnya adalah mengidentifikasi ekspresi dan menunjukkan hasil prediksi pada gambar aslinya dengan kotak pembatas dan label yang menunjukkan presentase ekspresi wajah. Hasil ini akan menunjukkan ekspresi wajah yang ada dalam gambar.

### **3. Pengujian Akurasi deteksi Wajah**

Halaman ini menampilkan *bounding box* yang dijelaskan pada poin sebelumnya. Perangkat lunak ini dapat mengidentifikasi empat ekspresi wajah yang berbeda: kegembiraan, kemarahan, kesedihan, dan keterkejutan. Saat kamera memfokus pada wajah, kotak pembatas yang mewakili ekspresi wajah akan muncul.



**Gambar 4.24** Pengujian Akurasi Deteksi Wajah

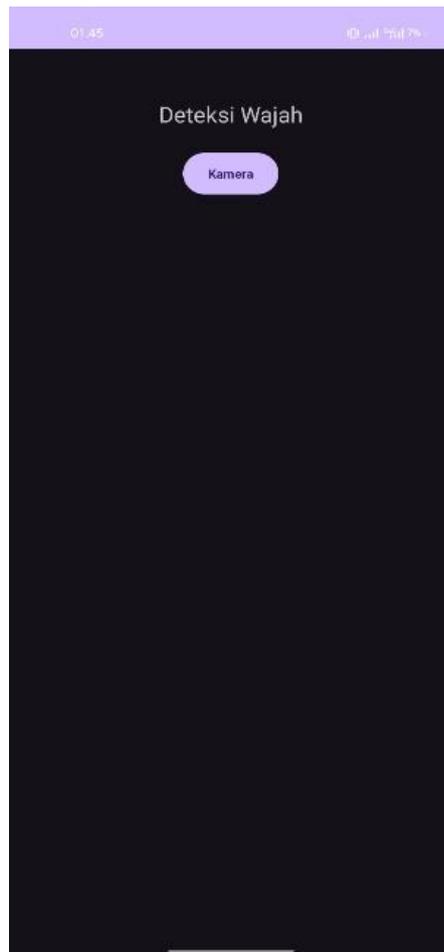
Ini adalah hasil deteksi pada ekspresi senang dan sedih dengan *bounding box* berlabel keterangan yang terdeteksi.

#### **D. Pengujian Sistem**

- 1. Pengujian *Black Box***
  - a. Pengujian *Black Box* Halaman Utama.

**Tabel 4.3** Pengujian *Black Box* Halaman Utama.

Tes Faktor	Hasil	Keterangan
Menu Home		Aplikasi terbuka dengan menampilkan menu home seperti gambar dibawah.



b. Pengujian *Black Box* Ekspresi Terbelalak

**Tabel 4.4** Pengujian *Black Box* Ekspresi Terbelalak

Tes Faktor	Hasil	Keterangan
Terbelalak	✓	Aplikasi berhasil mendeteksi ekspresi terbelalak menampilkan <i>bounding box</i> dengan keterangan terbelalak beserta dengan parameternya. Dapat diketahui bahwa yang terdeteksi pada ekspresi terbelalak yaitu mata membesar bersamaan dengan alis yang naik sesuai dengan dataset yang digunakan.



c. Pengujian *Black Box* Ekspresi Terpejam

**Tabel 4.5** Pengujian *Black Box* Ekspresi Terpejam

Tes Faktor	Hasil	Keterangan
Terpejam	✓	Aplikasi berhasil mendeteksi ekspresi terpejam menampilkan <i>bounding box</i> dengan keterangan terpejam beserta dengan parameternya. Dapat diketahui bahwa yang terdeteksi pada ekspresi terpejam yaitu struktur wajah yang datar dan poin utama terdeteksi pada mata yang tertutup sesuai dengan dataset yang digunakan.



d. Pengujian *Black Box* Ekspresi Terdiam

**Tabel 4.6** Pengujian *Black Box* Ekspresi Terdiam

Tes Faktor	Hasil	Keterangan
Terdiam	✓	Aplikasi berhasil mendeteksi ekspresi terdiam menampilkan <i>bounding box</i> dengan keterangan terdiam beserta dengan parameternya. Dapat diketahui bahwa yang terdeteksi pada ekspresi terdiam yaitu struktur wajah datar tanpa berekspresi sesuai dengan dataset yang digunakan.



e. Pengujian *Black Box* Ekspresi Tersenyum

**Tabel 4.7** Pengujian *Black Box* Ekspresi Tersenyum

Tes Faktor	Hasil	Keterangan
Tersenyum	✓	Aplikasi berhasil mendeteksi ekspresi tersenyum menampilkan <i>bounding box</i> dengan keterangan tersenyum beserta dengan parameternya. Dapat diketahui bahwa yang terdeteksi pada ekspresi tersenyum yaitu terjadinya pergerakan pada bagian mata dan pada bagian ujung bibir melebar kesamping sesuai dengan dataset yang digunakan.



f. Pengujian *Black Box* Ekspresi Tertawa

**Tabel 4.8** Pengujian *Black Box* Ekspresi Tertawa

Tes Faktor	Hasil	Keterangan
Tertawa	✓	Aplikasi berhasil mendeteksi ekspresi tertawa menampilkan <i>bounding box</i> dengan keterangan tertawa beserta dengan parameternya. Dapat diketahui bahwa yang terdeteksi pada ekspresi tertawa yaitu terjadinya pergerakan pada bagian mata dan bibir melebar disertai dengan mulut yang terbuka secara bersamaan sesuai dengan dataset yang digunakan.



g. Pendeteksian tidak berhasil

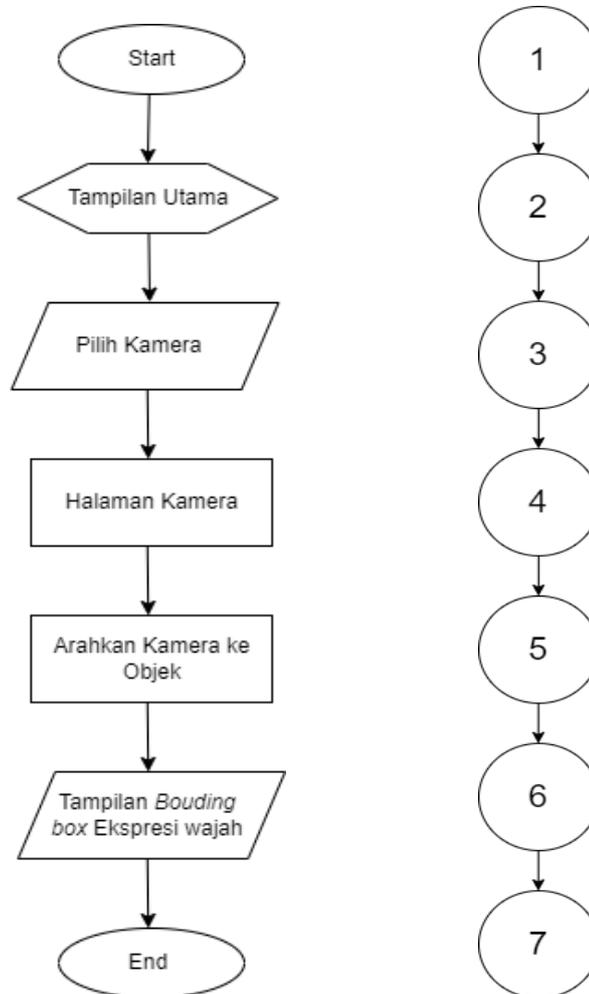
**Tabel 4.9** Pendeteksian tidak berhasil

Tes Faktor	Hasil	Keterangan
Terbelalak	<b>X</b>	Aplikasi tidak berhasil mendeteksi ekspresi terbelalak dan menampilkan <i>bounding box</i> dengan keterangan terpejam beserta dengan parameternya. Dapat diketahui bahwa yang terdeteksi pada ekspresi terbelalak yaitu mata membesar bersamaan dengan alis yang naik sesuai dengan dataset yang digunakan.



## 2. Pengujian *White Box*

a. *Flowchart* dan *Flowgraph* halaman utama.



**Gambar 4.25** *Flowchart* dan *Flowgraph* Halaman Utama.

Dari *flowchart* halaman utama di atas, dapat dilakukan proses perhitungan sebagai berikut:

1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari Edge dan Node

Dengan Rumus:  $V(G) = E - N + 2$

$N(\text{node}) = 7$

$$E (\text{edge}) = 6$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian;

$$V (G) = E - N + 2$$

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Floegraph* di atas memiliki Region = 1

3) *Independent path* pada *flowgraph* di atas adalah;

$$\text{Path 1} = 1,2,3,4,5,6,7$$

4) Grafik Matriks halaman utama

**Tabel 4.7** Matriks Halaman Utama.

	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4					1			1 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
SUM (E + 1)								0 + 1 = 1

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Bab-bab sebelumnya telah memberikan gambaran umum tentang hasil pengujian dan diskusi tentang Aplikasi Deteksi Ekspresi pada Wajah Dengan Mesin Learning. Aplikasi Deteksi Ekspresi pada Wajah Dengan Mesin Learning merupakan teknologi yang memungkinkan mengidentifikasi ekspresi wajah seseorang. Pendekatan ini memanfaatkan algoritma YOLO (*You Only Look Once*) untuk mengenali ekspresi wajah manusia. Studi ini menggunakan beragam dataset wajah yang mencakup sudut pandang, ekspresi dan kondisi pencahayaan yang berbeda untuk melati model deteksi ekspresi wajah dengan akurasi yang lebih tinggi. Penelitian ini juga menunjukkan bahwa penggunaan YOLO dalam deteksi ekspresi wajah mampu mencapai tingkat akurasi yang signifikan, yang memungkinkan aplikasi ini untuk digunakan dalam berbagai konteks.

## **B. Saran**

Di masa mendatang, menambahkan banyak fitur menarik akan membuat aplikasi ini lebih menarik. Saat ini, aplikasi ini dapat mendeteksi lebih dari ekspresi wajah dan hanya empat ekspresi wajah. Menambah fitur yang dapat mendeteksi objek selain wajah akan membuat aplikasi ini lebih berkembang dan akan lebih menarik jika tampilan awalnya sedikit lebih kreatif. Penulis dapat membuat saran ini. Beberapa fitur yang disebutkan di atas dapat digunakan sebagai titik awal untuk mengembangkan aplikasi ini secara keseluruhan, terutama untuk penulis.

## DAFTAR PUSTAKA

- Abdulrahman Alreshidi & Mohib Ullah (2020). Facial Emotion Recognition Using Hybrid Features. *Informatics* 2020, 7, 6; doi:10.3390/informatics7010006
- Ahmad, A. (2017). Mengenal artificial intelligence, machine learning, neural network, dan deep learning. *J. Teknol. Indones.*, no. October, 3.
- Akhand M. A. H., Roy, S. Siddique, N. Kamal M.A.S & Shimamura, T. Facial Emotion Recognition Using Transfer Learning in the Deep CNN. *Electronics* 2021, 10, 1036.
- Amda, K., & Fitriyani, R. (2016). *Membaca Ekspresi Wajah: Mengungkap Kepribadian Seseorang dengan Membaca Wajah* (pp. 140–155).
- Amda, K., & Fitriyani, R. (2019). *Membaca ekspresi wajah*. Penerbit Genesis.
- AZIZI, F. N. (2021). Deteksi Emosi Menggunakan Citra Ekspresi Wajah Secara Otomatis.
- DiMarzio, J. F. (2016). *WOW! eBook* [www.wowebook.org](http://www.wowebook.org). [http://solutionsproj.net/software/Beginning\\_Android\\_Programming\\_with\\_Android\\_Studio.pdf](http://solutionsproj.net/software/Beginning_Android_Programming_with_Android_Studio.pdf)
- Heryadi, Y., & Wahyono, T. KONSEP DAN IMPLEMENTASI.
- Ibrahim, A. F., Dewanta, F., & Raniprima, S. (2022). Implementasi Machine Learning Pada Alat Deteksi Emosi Untuk Sistem Kontrol Suhu Dan Pencahayaan Ruangan. *eProceedings of Engineering*, 9(2).
- Ihsan, M., Niswatin, R. K., Swanjaya, D., Informatika, T., Teknik, F., & Kediri, K. (2021). Deteksi ekspresi wajah menggunakan tensorflow. *J. Tek. UNISLA*, 6, 1-6.
- James W, Elston D, T. J. et al. (2023). Kecerdasan Emosional (TEORI DAN APLIKASI). In *Andrew's Disease of the Skin Clinical Dermatology*.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia computer science*, 199, 1066-1073.
- Mr. Shubham Babhulkar<sup>1</sup>, Prof. M. S. Chaudhari (2021). Application of Machine Learning for Emotion Classification. *ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429*
- Mudadla, K., & Kumari, M. S. (2021). Android dictionary application in Android studio. *International Journal of Advance Research, Ideas and Innovations in Technology*, 7(4), 150–156. <https://www.ijariit.com>
- Ninad Mehendale (2020). Facial emotion recognition using convolutional neural networks (FERC). *SN Applied Sciences* (2020) 2:446 |

- PRAYOGA, M. (2020). *SISTEM DETEKSI EMOSI MARAH PADA PENGEMUDI MOBIL BERBASIS ANDROID* (Doctoral dissertation, Universitas Hasanuddin).
- Puspaningrum, EY, & Saputra, WS (2018). DETEKSI WAJAH DENGAN BOOSTED CASCADE CLASSIFIER. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*
- Rahmadani, S., Rahayu, C. S., Salim, A., & Cahyo, K. N. (2022). DETEKSI EMOSI BERDASARKAN WICARA MENGGUNAKAN DEEP LEARNING MODEL. *Jurnal Informatika Teknologi dan Sains (Jinteks)*, 4(3), 220-224.
- Rosiani, UD, & Choirina, P. (2020). Pendeteksian dan Pelacakan Lokasi Wajah pada Tahap Pra-pemrosesan Pengenalan Ekspresi Mikro Menggunakan Metode Kanade-Lucas-Tomasi (KLT). *Jurnal Informatika Polinema*
- Safiro, G. A. (2020). *LKP: Deteksi Ekspresi Wajah Menggunakan Deep Learning* (Doctoral dissertation, Universitas Dinamika).
- Sakdiyah, H., Swastika, W., & Setiawan, H. RANCANG BANGUN APLIKASI INFORMASI EKSPRESI EMOSI UNTUK PENENTUAN KATEGORI FILM BERBASIS CNN.
- Septian, R., Saputra, D. I., & Sambasri, S. (2020, March). Klasifikasi Emosi Menggunakan Convolutional Neural Networks. In *Prosiding Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung* (pp. 53-62).
- Sudianto, S. (2022). Analisis Kinerja Algoritma Machine Learning Untuk Klasifikasi Emosi. *Building of Informatics, Technology and Science (BITS)*, 4(2), 1027-1034.
- Suja Palaniswamy & Shikha Tripathi (2018). Emotion Recognition from Facial Expressions using Images with Pose, Illumination and Age Variation for Human-Computer/Robot Interaction. *J. ICT Res. Appl.*, Vol. 12, No. 1, 2018, 14-34
- Ulhaq, M. R. D., Zaidan, M. A., & Firdaus, D. (2023). Pengenalan Ekspresi Wajah Secara Real-Time Menggunakan Metode SSD Mobile Net Berbasis Android. *Journal of Technology and Informatics (JoTI)*, 5(1), 48-52.
- Wildah, S. K., Agustiani, S., Mustopa, A., Wuryani, N., Nawawi, H. M., & Safitri, R. A. (2021). Pengenalan Wajah Menggunakan Pembelajaran Mesin Berdasarkan Ekstraksi Fitur pada Gambar Wajah Berkualitas Rendah Face Recognition Using Machine Learning Based on Feature Extraction on Low Quality Face Images. *INFOTECH J. Inform. Teknol*, 2(2), 95-103.
- Yusuf, R., & Huda, A. A. (2023). Deteksi Emosi Wajah Menggunakan Metode Backpropagation. *Journal Automation Computer Information System*, 3(2), 103-114.

Zophie, J., & Triharminto, H. H. (2022). 9. Implementasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis. *TNI Angkatan Udara, 1*(