**APLIKASI MENDETEKSI KATEGORI *­*USIA SESEORANG BERDASARKAN CITRA WAJAH MENGGUNAKAN ALGORITMA *HAARCASCADE CLASSIFIER* DAN *MOBILENET* BERBASIS *DESKTOP***

**SKRIPSI**

Sebagai Salah Satu Syarat untuk Meraih

Gelar Sarjana Komputer

Oleh:

William Thamrin

32160078



Fakultas Teknologi Dan Desain

Program Studi Teknik Informatika

Universitas Bunda Mulia

Jakarta

2020

|  |
| --- |
| **UNIVERSITAS BUNDA MULIA**  **FAKULTAS TEKNOLOGI DAN DESAIN**  **PROGRAM STUDI TEKNIK INFORMATIKA** |
| Pernyataan Kesiapan Ujian Pendadaran Skripsi  Saya William Thamrin, dengan ini menyatakan bahwa Skripsi yang berjudul :  **APLIKASI MENDETEKSI KATEGORI *­*USIA SESEORANG BERDASARKAN CITRA WAJAH MENGGUNAKAN ALGORITMA *HAARCASCADE CLASSIFIER* DAN *MOBILENET* BERBASIS *DESKTOP***  Merupakan hasil karya saya dan belum pernah diajukan sebagai karya ilmiah, sebagian seluruhnya, atas nama saya atau pihak lain  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  William Thamrin  32160078  Disetujui oleh Pembimbing,  Kami setuju Skripsi tersebut diajukan untuk Ujian Pendadaran  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  I Gusti Ngurah Suryantara, S.Kom, M.Kom 1 Januari 2020  Disetujui oleh Ketua Program Studi,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Lukman Hakim, ST., M.Kom 1 Januari 2020 |

|  |
| --- |
| **UNIVERSITAS BUNDA MULIA**  **FAKULTAS TEKNOLOGI DAN DESAIN**  **PROGRAM STUDI TEKNIK INFORMATIKA** |
| **Persetujuan Skripsi**  Yang bertanda tangan di bawah ini, menyatakan bahwa Skripsi dengan judul  **APLIKASI MENDETEKSI KATEGORI *­*USIA SESEORANG BERDASARKAN CITRA WAJAH MENGGUNAKAN ALGORITMA *HAARCASCADE CLASSIFIER* DAN *MOBILENET* BERBASIS *DESKTOP***  Disusun oleh :  William Thamrin 32160078  Telah disetujui dan diterima sebagai salah satu karya ilmiah mahasiswa yang bersangkutan pada Fakultas Teknologi dan Desain – Program Studi Teknik Informatika Universitas Bunda Mulia  Jakarta, 1 Januari 2020  Mengetahui  Ketua Program Studi Dosen Pembimbing  (Lukman Hakim, ST., M.Kom) (I Gusti Ngurah Suryantara, S.Kom, M.Kom) |

**PERNYATAAN**

Saya menyatakan bahwa Skripsi yang berjudul **“APLIKASI MENDETEKSI KATEGORI *­*USIA SESEORANG BERDASARKAN CITRA WAJAH MENGGUNAKAN ALGORITMA *HAARCASCADE CLASSIFIER* DAN *MOBILENET* BERBASIS *DESKTOP*”**, sepenuhnya karya saya sendiri. Tidak ada bagian di dalamnya yang merupakan plagiat dari karya orang lain dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko / sanksi yang dijatuhkan kepada saya, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Jakarta, 1 Januari 2020

Yang membuat pernyataan

William Thamrin

# ABSTRAK

Deteksi objek adalah salah satu bagian dari pengolahan citra dan *Computer Vision* yang terkenal. Deteksi objek pun dapat diaplikasikan kedalam berbagai hal salah satunya adalah dalam mendeteksi usia seseorang melalui citra wajah manusia. Faktor – faktor seperti warna kulit, biometrik wajah manusia, panjang rambut juga mempengaruhi aplikasi dalam perhitungan.

Dalam melakukan deteksi usia seseorang melalui citra wajah manusia digunakan algoritma *Haar-Cascade Classifier* dan *Mobilenet*. Algoritma *mobilenet* digunakan dalam program untuk mendeteksi bagian – bagian wajah seperti mata, telinga, hidung, mulut, pipi, rambut, dll. Sedangkan algoritma *Haar-Cascade Classifier* digunakan program untuk mengklasifikasikan suatu foto dengan kumpulan foto yang ada di *database* lalu dihitung masuk dalam kategori rentang usia mana.

Hasil yang diharapkan adalah aplikasi dapat mendeteksi usia seseorang menggunakan citra wajah manusia dengan tepat. Aplikasi juga diharapkan memiliki waktu pemrosesan yang cepat.

Kesimpulan yang diharapkan adalah aplikasi dapat dikembangkan dan diaplikasikan ke tahap yang lebih tinggi, dan juga dapat digunakan oleh orang lain, khususnya mahasiswa UBM dalam penelitian lebih tinggi.

**Kata Kunci :**

*Haar-Cascade Classifier, Mobilenet,* Pemograman Citra*, Computer Vision*.

# PRAKATA

# DAFTAR ISI

# BAB 1 PENDAHULUAN

## Latar Belakang

Pendeteksian objek adalah salah satu topik yang sangat terkenal pada pengolahan citra maupun *Computer Vision*. Penerapan pendeteksian objek dapat dilakukan dibanyak hal seperti medis, keamanan, militer, dan lain lain. Sebagai contoh, jika kita ingin mendeteksi jenis kelamin seseorang, maka kita harus mengetahui beberapa faktor seperti wajah yang dimiliki seseorang, panjang rambut, warna kulit, dan lain lain. Dengan era teknologi yang terus berkembang, pendeteksian objek yang dulu hanya dapat dilakukan secara bertatapan langsung, sekarang dapat ditempuh dengan sistem komputer yang telah dilatih. Walau begitu pendeteksian objek mengunakan sistem komputer juga memiliki banyak hambatan. Oleh sebab itu diperlukan metode-metode khusus pada sistem komputer untuk dapat melakukan pendeteksian objek [11].

Ada dua cara yang dapat dilakukan untuk melakukan pendeteksian objek, yaitu dengan mengunakan segmentasi citra dan deteksi tepi. Segmentasi citra merupakan cara untuk memisahkan antara objek utama, objek sekunder, dan objek latar. Segmentasi citra dapat dilakukan dengan memisahkan objek yang memiliki kontras dan warna yang sama. Deteksi tepi sendiri merupakan cara untuk mencari perubahan intensitas warna yang signifikan pada piksel tertentu dan lalu perbedaan piksel tersebut akan ditandai sebagai batas suatu objek.

Berdasarkan latar belakang yang terjadi, maka dibuat skripsi dengan judul “**Mendeteksi Kategori Usia Seseorang Berdasarkan Citra Wajah Mengunakan Algoritma *Haar-Cascade Classifier* dan *Mobilenet* Berbasis *Desktop***”. Dengan tujuan agar aplikasi berikut dapat bermanfaat dan memudahkan seseorang dalam melakukan pendeteksian suatu objek.

## Rumusan Masalah

Berdasarkan latar belakang skripsi maka didapatkan suatu rumusan masalah yaitu :

1. Bagaimana menerapkan metode *Haar-Cascade Classifier* pada aplikasi deteksi usia menggunakan citra wajah
2. Bagaimana keakuratan aplikasi deteksi usia dengan metode *Haar-Cascade Classifier* menggunakan citra wajah.

## 1.3 Tujuan dan Manfaat

### Tujuan

Tujuan yang didapatkan dalam skripsi ini adalah:

* + - 1. Menerapkan metode *Haar-Cascade Classifier* pada aplikasi *Desktop* dalam melakukan deteksi usia menggunakan citra wajah.
      2. Mendapatkan nilai akurasi untuk deteksi usia dengan citra wajah yang menggunakan metode *Haar-Cascade Classifier*.

### Manfaat

Manfaat yang didapatkan dalam skripsi ini adalah sebagai berikut:

1. memudahkan proses dalam melakukan deteksi usia menggunakan citra wajah.

## Ruang Lingkup

Ruang lingkup yang ada pada skripsi ini adalah sebagai berikut:

1. Objek deteksi hanya dilakukan pada citra wajah, dan rambut.
2. *Media* yang digunakan dalam deteksi usia berupa gambar.
3. Besarnya ukuran gambar yang di *training* secara keseluruhan sebesar kurang lebih *3GB*.

## Metodologi Penelitian

### Teknik Pengumpulan *Data*

*Data* yang digunakan dalam skripsi ini didapatkan dari gambar dengan format .*jpg* atau *.png* yang tergolong *free to use* atau bebas untuk digunakan yang ada pada *internet*.

### Teknik Perancangan Aplikasi

Perancangan aplikasi dibuat dengan *Visual Studio Code* sebagai aplikasi perancang utama, dan juga Algoritma *Haar-Cascade Classifier* dan *Mobilenet* sebagai algoritma utama pada aplikasi deteksi usia berikut.

### Pemodelan Sistem

Pemodelan sistem pada aplikasi informasi ini menggunakan *UML* karena pemograman ini bersifat berorientasi objek.

## Sistematika Penulisan

### BAB 1 PENDAHULUAN

Bab 1 berisi latar belakang masalah, rumusan masalah, tujuan, manfaat, ruang lingkup, metodologi penelitian, dan sistematika penulisan.

### BAB 2 LANDASAN TEORI

Bab 2 berisi teori yang berupa pengertian dan definisi yang diambil dari sumber tepercaya untuk menjadi dasar untuk digunakan dalam menyelesaikan permasalahan yang sudah dijelaskan pada bab sebelumnya.

### BAB 3 ANALISIS DAN PERANCANGAN

Bab 3 berisi tentang analisis kebutuhan yang diperlukan dalam melakukan skripsi serta rancangan sistem yang dipresentasikan menggunakan *flowchart* atau *UML*.

### BAB 4 IMPLEMENTASI

Bab 4 berisi penerapan teori kedalam bahasa pemrograman dan pengujian terhadap pengklasifikasian yang akan dilakukan.

### BAB 5 SIMPULAN DAN SARAN

Bab 5 berisi tentang simpulan dari skripsi yang dilakukan dan saran dalam melakukan skripsi terkait.

# **BAB 2** LANDASAN TEORI

## *Computer Vision*

Vision secara bahasa dapat diartikan sebagai penglihatan. Vision juga dapat diartikan sebagai suatu proses pengamatan apa yang ada pada dunia nyata melalui panca indra penglihatan manusia. Adapun computer vision adalah suatu pembelajaran menganalisis gambar dan video untuk memperoleh hasil sebagaimana yang bisa dilakukan manusia. Pada hakikatnya, computer vision mencoba meniru cara kerja sistem visual manusia (Human Vision). Manusia melihat obyek dengan indra penglihatan (mata), lalu citra obyek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti obyek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini mungkin digunakan untuk pengambilan keputusan (misalnya menghindar kalau melihat mobil melaju di depan atau menghindar ketika ada pejalan kaki ketika sedang mengendarai sebuah mobil). Analisis visual pergerakan manusia juga marupakan salah satu topik terpopuler pada computer vision [12].

## Pengolahan Citra

Di dalam bidang komputer, sebenarnya ada tiga bidang studi yang berkaitan dengan *data* citra, namun tujuan ketiganya berbeda, yaitu:

### Grafika Komputer (*Computer Graphics*)

Grafika Komputer bertujuan menghasilkan citra (lebih tepat disebut grafik atau *picture*) dengan primitif-primitif geometri seperti garis, lingkaran dan sebagainya. Primitif-primitif geometri tersebut memerlukan data deskriptif untuk melukis elemen-elemen gambar. Contoh *data* deskriptif adalah koordinat titik, panjang garis, jari-jari lingkaran, tebal garis, warna, dan sebagainya. Grafika komputer memainkan peranan penting dalam visualisasi dan *virtual reality* [12].

### Pengolahan Citra (*Image Processing*)

Pengolahan Citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan termasuk ke dalam bidang ini juga adalah pemampatan citra *(image compression*) [12].

### Pengenalan Pola (*Patern Recognition* atau *Image Compression*)

Pengenalan pola mengelompokkan *data* numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu obyek di dalam citra. Manusia bisa mengenali obyek yang dilihatnya karena otak manusia telah belajar mengklasifikasi obyek obyek di alam sehingga mampu membedakan suatu obyek dengan obyek lainnya. Kemampuan sistem *visual* manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra obyek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi obyek di dalam citra [12].

## *Object Detection*

Suatu proses yang digunakan untuk menetukan keberadaan objek tertentu di dalam suatu citra digital. Proses deteksi tersebut dapat dilakukan dengan berbagai macam metode yang umumnya melakukan pembacaan fitur-fitur dari seluruh objek pada citra *input*. Fitur dari objek pada citra *input* tersebut akan dibandingkan dengan fitur dari objek referensi atau *template*. Hasil perbandingan tersebut dapat digunakan untuk menentukan apakah suatu objek terdeteksi sebagai *template* yang dimaksud atau tidak. Proses pembacaan dan perbandingan fitur objek umumnya mengalami kesulitan pada saat suatu objek berhimpit atau tertutup oleh objek lain dalam citra. Hal ini mengakibatkan dua objek tersebut terbaca sebagai satu objek dan berakibat pada hasil pembacaan fitur yang salah terutama saat suatu bagian objek tertutup objek lain. Kejadian ini juga dikenal sebagai permasalahan *overlap* pada *digital image processing* [2].

## *Deep Learning*

*Deep Learning* adalah cabang ilmu *machine learning* berbasis Jaringan Saraf Tiruan *(JST)* atau bisa dikatakan sebagai perkembangan dari *JST*. Dalam *deep learning*, sebuah komputer belajar mengklasifikasi secara langsung dari gambar atau suara. *Convolutional Neural Network (CNN/ConvNet)* adalah salah satu algoritma *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron (MPL)* yang dirancang untuk mengolah *data* dalam bentuk dua dimensi, misalnya gambar atau suara. *CNN* dapat belajar langsung dari citra sehingga mengurangi beban dari pemrograman [9].

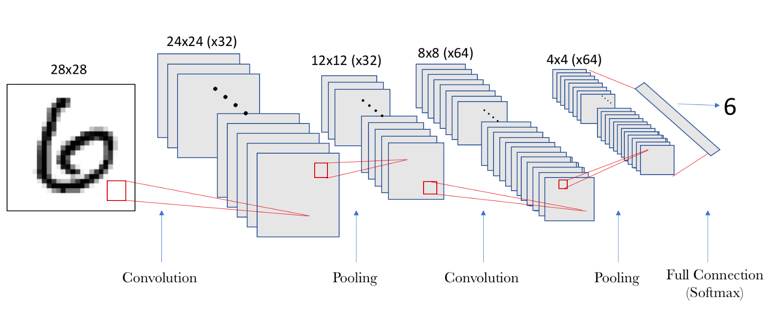
Selain itu, metode *machine learning* konvensional hanya mengandalkan *CPU* dan *RAM* dalam proses komputasi, sehingga spesifikasi *CPU* dan *RAM* menentukan kecepatan komputasi. Sedangkan metode *deep learning*, selain menggunakan *CPU* dan *RAM* dalam proses komputasi, metode ini juga memanfaatkan kemampuan *GPU* sehingga proses komputasi *data* yang besar dapat berlangsung lebih cepat [9].

## *Convolution Neural Network (CNN)*

*Convolutional Neural Network* yang juga dikenal dengan *CNN* atau *ConvNet* adalah pengembangan dari *Multilayer Perceptron (MLP)* untuk mengolah *data* dua dimensi. *Multilayer Perceptron* sendiri adalah pengembangan dari *Artificial Neural Network (ANN)* yang ditujukan untuk menutupi kekurangan *Artificial Neural Network (ANN)* dengan *Single-layer Perceptron* dalam menyelesaikan operasi logika yang kompleks. Hal tersebut dimungkinkan dengan menambahkan *hidden layer* yang membuat *ANN* kuat untuk memecahkan operasi logika yang kompleks *(universal approximation)* dan sering digunakan untuk permasalahan-permasalahan klasifikasi, *recognition*, dan prediksi [11].

*CNN* dapat dibilang sangat sukses dalam video dan *image recognition* berskala besar, *Google* bahkan menggunakan *CNN* untuk mengambil nomor rumah dari gambar *StreetView*. Akan tetapi, dari sekian banyaknya keunggulan yang ditawarkan *CNN*, untuk mengaplikasikan *CNN* pada sekumpulan gambar dalam skala besar dan berkualitas tinggi masih terhitung mahal secara waktu, *dataset* yang digunakan, dan *hardware*. Namun hal tersebut dapat diatasi saat ini dengan *dataset* gambar berukuran besar yang telah disediakan seperti *ImageNet* dan *pre-trained model* seperti yang disediakan oleh *Keras* [11].

Berbeda dengan manusia, komputer mengenali gambar dalam bentuk *array* dari nilai piksel-pikselnya. Bayangkan untuk kasus *input* gambar dengan resolusi piksel 260×260, akan terdapat 260×260×3 *array* dari angka. Angka 3 yang merupakan suku ketiga dari perkalian melambangkan 3 nilai *RGB* pada gambar. Selanjutnya, *array* angka-angka tersebut akan diproses oleh *CNN* untuk dijadikan nilai kemungkinan suatu gambar berada pada kelas tertentu [11].

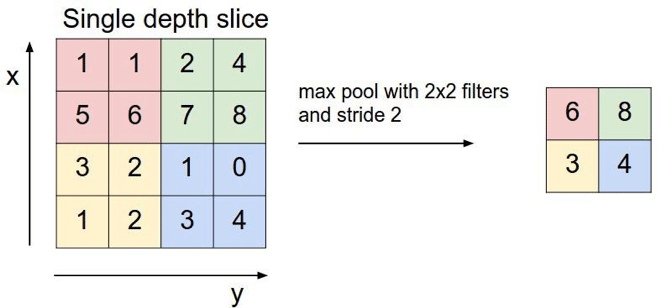


Gambar 2. 1 *Layer* Pada *Convolutional Neural Network*

Pada *CNN data* yang dikirimkan ke dalam jaringan adalah *data* dua dimensi, sehingga bobot pada hubungan antar *neuron* akan berbeda. Proses *CNN* dibagi menjadi 4 tahapan utama yaitu *Input*, *Convolution*, *Pooling*, dan *Fully-Connection*. *Layer input* adalah *layer* pertama pada proses *CNN* dimana *data* dimasukan kedalam tahap proses pertama [11].

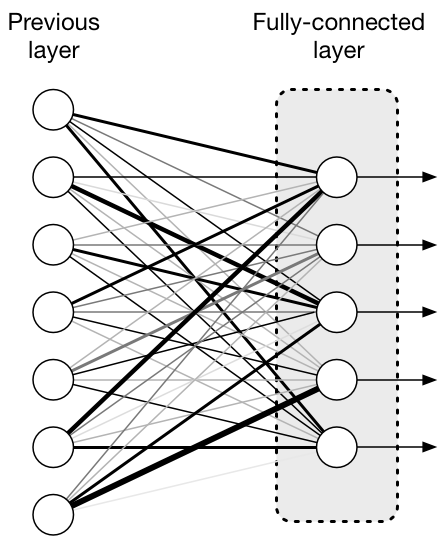
*Layer* kedua setelah *layer input* pada Gambar 2.1 dapat dibagi menjadi beberapa bagian yaitu *convolution layer*, *pooling layer*, dan *fully-connected layer* [11].

Pada *convolution layer*, dilakukan operasi konvolusi yang merupakan proses utama pada *CNN*. Konvolusi adalah istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Ketika menguji keberadaan fitur pada gambar baru, *CNN* akan mencoba semua kemungkinan posisi pada gambar. *Filter* dibuat untuk menghitung kecocokan fitur pada keseluruhan gambar [11].



Gambar 2. 2 *Max Pooling* Pada *Convolution Neura Network*

Pada *pooling layer* dilakukan operasi *subsampling*, yaitu proses mereduksi ukuran sebuah *data* citra. *Pooling* bertujuan untuk memperkecil gambar berukuran besar tetapi masih menyimpan informasi-informasi pentingnya. Salah satu pendekatan yang banyak digunakan untuk *CNN* adalah *max pooling*, *max pooling* membagi matriks gambar ke dalam beberapa bagian kecil dan memilih nilai paling besar di dalamnya untuk digunakan ketika matriks gambar baru yang sudah direduksi dibentuk. Gambar 2.2 menunjukkan konsep *max pooling* pada *Convolution Neural Network* [11].



Gambar 2. 3 *Fully-Connected Layer* Pada *Convolution Neural Network*

*Fully-connected layer* adalah *layer* yang biasa digunakan pada *Multilayer* *Perceptron (MLP)* dan bertujuan untuk melakukan transformasi pada dimensi *data* agar dapat diklasifikasi secara linear. Gambar 2.3 menggambarkan *layer* sebelumnya yang saling terkoneksi sehingga menghasilkan *layer* baru yaitu *Fully-Connected Layer*. *Output* dari *layer* ini adalah *array* dengan panjang jumlah kelas yang *model* harus pilih. Pada *fully-connected layer*, bobot yang paling besar dari *layer* sebelumnya akan menentukan fitur mana yang paling berhubungan dengan kelas atau *label* tersebut [11].

Proses *training* pada *CNN* untuk mendapatkan fitur-fitur serta *weight* dari setiap kelas adalah dengan menggunakan metode yang disebut dengan *Haar-Cascade Classifier*.

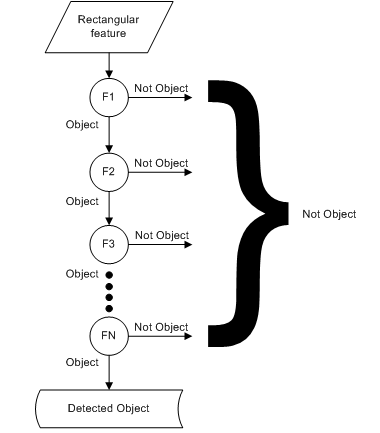
## *Mobilenet*

*MobileNet* merupakan salah satu arsitektur *Convolutional Neural Network (CNN)* yang dapat digunakan untuk mengatasi kebutuhan akan *computing data* dalam jumlah yang besar. Perbedaan mendasar antara arsitektur *MobileNet* dan arsitektur *CNN* pada umumnya adalah penggunaan lapisan atau *layer* konvolusi dengan ketebalan *filter* yang sesuai dengan ketebalan dari *input image*. *MobileNet* membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution* [6].

## *Haar Cascade Classifier*

Untuk proses pendeteksi wajah digunakan algoritma *Haar Cascade Classifier*. Secara umum, *haar-like feature* digunakan dalam mendeteksi objek pada gambar digital. Istilah *Haar* menunjukkan suatu fungsi matematika *(Haar Wavelet)* yang berbentuk kotak, prinsipnya sama seperti pada fungsi *Fourier*. Awalnya pengolahan gambar hanya dengan melihat dari nilai *RGB* setiap *pixel*, namun metode ini ternyata tidaklah efektif. *Viola* dan *Jones* kemudian mengembangkannya sehingga terbentuk *Haar-Like feature*. *Haar-like feature* memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa *pixel*. Per kotak itu pun kemudian diproses dan menghasilkan perbedaan nilai yang menandakan daerah gelap dan terang. Nilai-nilai inilah yang nantinya dijadikan dasar dalam pemrosesan gambar [1].

Cara menghitung nilai dari fitur ini adalah dengan mengurangkan nilai *pixel* pada area putih dengan *pixel* pada area hitam. Untuk mempermudah proses penghitungan nilai fitur, algoritma *Haar* menggnakan sebuah *media* berupa *Integral Image*. *Integral Image* adalah sebuah citra yang nilai tiap pikselnya merupakan penjumlahan dari nilai *pixel* kiri atas hingga kanan bawah. Sebagai contoh *pixel* (a,b) memiliki nilai akumulatif untuk semua *pixel* (x, y). Dimana x ≤ a dan y ≤ b. Dalam menggunakan metode *haar cascade* ada beberapa jenis citra gambar yang bisa diolah salah satunya yaitu *grayscale* [1].



Gambar 2. 4 *Fully-Connected Layer* Pada *Convolution Neural Network*

*Cascade Classifier* merupakan *step* untuk mendapatkan hasil yang lebih akurat dengan menghitung nilai *Haar Feature* secara banyak dan berulang. Gambar 2. 4 menampilkan alur kerja dari *Cascade Classifier*. Pada klasifikasi tahap 1, setiap sub citra akan diklasifikasikan dengan satu fitur, bila hasil tidak memenuhi kriteria, maka hasil ditolak. Pada klasifikasi tahap 2, setiap sub citra akan diklasifikasikan kembali. Jika didapatkan nilai *threshold* yang diinginkan, maka dilanjutkan ke tahap *filter* selanjutnya (klasifikasi tahap 3). Hingga sub-citra yang lolos akan berkurang hingga mendekati citra yang ada pada sampel [1].

## *OpenCV*

*OpenCV* adalah *library* yang didalamnya terdapat berbagai fungsi yang dipergunakan dalam *computer* *vision*. *OpenCV* ditulis dengan bahasa *C* dan *C++* dan dapat berjalan di *Linux*, *Windows* dan *MacOS*. *OpenCv* ini adalah *library* yang sudah sangat terkenal pada pengolahan citra komputer [10].

## *Tensorflow*

*TensorFlow* adalah *framework machine learning* yang bekerja dalam skala besar dan dalam *environment* yang *heterogeneous*. *TensorFlow* digunakan untuk melakukan eksperimen model *deep learning*, melatih model pada *dataset* yang berukuran besar, dan membuatnya layak diproduksi [11].

Selain itu *TensorFlow* juga mendukung *training* dan *inference* berskala besar dengan menggunakan ratusan *server* yang menggunakan *Graphic Processing Unit (GPU)* [11].

## *Keras*

*Keras* adalah *High Level Neural Network API* yang dapat dijalankan di atas *framework-framework machine learning* seperti *TensorFlow*, *CNTK*, atau *Theano*. *Keras* ditulis dalam bahasa *Python*. Lebih lanjut, *Keras* menyediakan *API* yang mempermudah user dalam membangun arsitektur *ANN* [11].

*Keras* juga menyediakan *Keras Applications* yang merupakan *deep learning models* yang digunakan Bersama dengan *pre-trained weights*. Model-model tersebut dapat digunakan untuk memprediksi, melakukan *feature extraction* atau *fine-tuning* [11].

Model-model tersebut di antaranya adalah *Xception*, *VGG16*, *VGG19*, *ResNet50*, *InceptionV3*, *InceptionResNetV2*, *MobileNet*, *ImageNet*, *DenseNet* & *NasNet* [11].

## *Python*

*Python* merupakan bahasa pemrograman yang berorientasi obyek dinamis, dapat digunakan untuk bermacam macam pengembangan perangkat lunak. *Python* menyediakan dukungan yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat-alat bantu lainnya. *Python* hadir dengan pustaka-pustaka standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari. Bahasa pemrograman yang interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode [5].

*Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif [5].

## *Visual Studio Code*

*Visual Studio Code* atau disebut *VS Code* adalah sebuah *text editor* ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. *Text editor* ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang di *marketplace Visual Studio Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java*, dst).

Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan *text editor*. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan *text editor* yang lain.

## *PyQt*

*PyQt* adalah seperangkat *binding Python* untuk kerangka kerja aplikasi lintas *platform* yang menggabungkan semua keunggulan *Qt* dan *Python*. Dengan *PyQt*, Anda dapat memasukkan pustaka *Qt* dalam kode *Python*, memungkinkan Anda untuk menulis aplikasi *GUI* dengan *Python*. Dengan kata lain, *PyQt* memungkinkan Anda untuk mengakses semua fasilitas yang disediakan oleh *Qt* melalui kode *Python*. Karena *PyQt* bergantung pada pustaka *Qt* untuk dijalankan, ketika Anda menginstal *PyQt*, versi *Qt* yang diperlukan juga diinstal secara otomatis pada mesin Anda.

Aplikasi *GUI* dapat terdiri dari jendela utama dengan beberapa *dialog* atau hanya satu *dialog*. Aplikasi *GUI* kecil biasanya terdiri dari setidaknya satu *dialog*. Aplikasi *dialog* berisi tombol. Itu tidak mengandung bilah menu, bilah alat, bilah status, atau *widget* pusat, sedangkan aplikasi jendela utama biasanya memiliki semua itu. *PyQt5* adalah versi *PyQt* paling akhir pada saat ini.

## *Command Line Interface (CLI)*

*CLI* adalah tipe antarmuka dimana pengguna berinteraksi dengan sistem operasi melalui *text-terminal*. Pengguna menjalankan perintah dan *program* di sistem operasi tersebut dengan cara mengetikkan baris-baris tertentu [8].

Meskipun konsepnya sama, tiap-tiap sistem operasi memiliki nama atau istilah yang berbeda untuk *CLI*-nya. *UNIX* memberi nama *CLI*-nya sebagai *bash*, *ash*, *ksh*, dan lain sebagainya. *Microsoft Disk Operating System (MS-DOS)* memberi nama *command.com* atau *Command Prompt*. Sedangkan pada *Windows Vista*, *Microsoft* menamakannya *PowerShell*. Pengguna *Linux* mengenal *CLI* pada *Linux* sebagai *terminal*, sedangkan pada *Apple* adalah *commandshell* [8].

## Biometrik Wajah Manusia

Dalam dunia teknologi informasi, biometrik relevan dengan teknologi yang digunakan untuk menganalisa fisik dan kelakuan manusia dalam autentifikasi. Biometrik secara teoritis dapat lebih efektif untuk mengidentifikasikan pribadi seseorang karena biometrik mengukur karakteristik masing-masing pribadi untuk membedakan setiap orang. Ketika digunakan untuk indentifikasi pribadi, teknologi biometrik mengukur dan menganalisa karakteristik tingkah laku dan fisiologis manusia [4].

## Kategori Umur Manusia

Secara ilmiah telah diketahui bahwa wajah mengandung banyak informasi penting dari seorang individu seperti *gender*, ras, dan umur. Banyak perubahan yang dapat terjadi pada wajah manusia, bisa bersifat sementara ataupun permanen. Salah satu contoh perubahan tersebut adalah kerutan. Menurut Hardiantara kerutan merupakan proses alami penuaan, seiring dengan bertambahnya usia maka kulit akan menjadi lebih tipis, lebih kering, dan berkurang elastisitasnya [3].

Umur manusia dapat dibagi menjadi beberapa rentang atau kelompok dimana masing-masing kelompok menggambarkan tahap pertumbuhan manusia tersebut. Salah satu pembagian kelompok umur atau kategori umur dikeluarkan oleh Departemen Kesehatan RI (2009) dalam situs resminya yaitu *depkes.go.id* sebagai berikut [3]:

1. Masa balita = 0 – 5 tahun,
2. Masa kanak-kanak = 6 – 11 tahun.
3. Masa remaja Awal = 12 – 16 tahun.
4. Masa remaja Akhir = 17 – 25 tahun.
5. Masa dewasa Awal = 26 – 35 tahun.
6. Masa dewasa Akhir = 36 – 45 tahun.
7. Masa Lansia Awal = 46 – 55 tahun.
8. Masa Lansia Akhir = 56 – 65 tahun.
9. Masa Manula = 65 – atas

## *Unified Modeling Language (UML)*

*Unified Modeling Language* *(UML)* bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh artifak sistem perangkat lunak. Penggunaan model ini bertujuan untuk mengidentifikasikan bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain di luarnya [7].

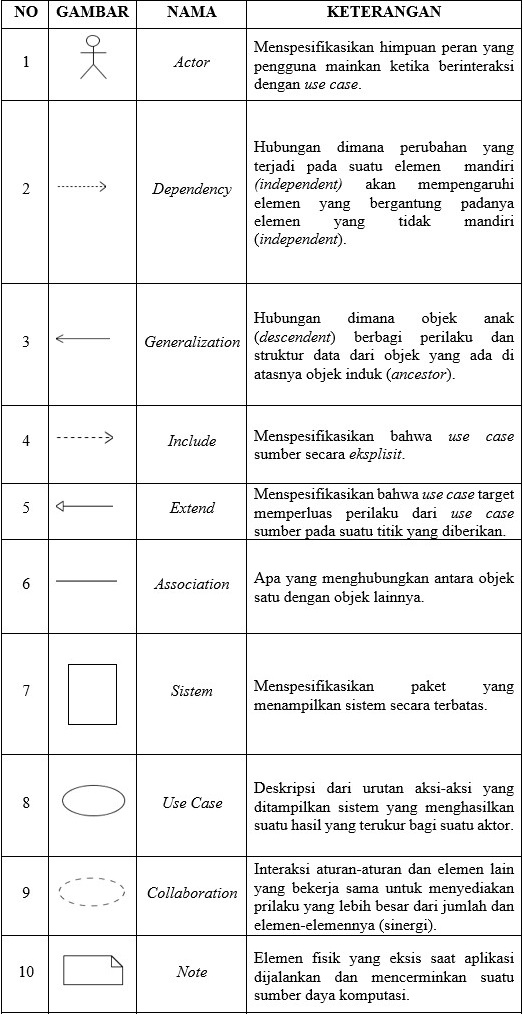
*Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan dari sebuah sistem pengembangan *software* berbasis *object oriented*.

Dari Pengertian diatas penulis menyimpulkan bahwa *Unified Modeling Language (UML)* merupakan bahasa pemodelan yang berbentuk grafis yang digunakan untuk memvisualisasi, menspesifikasikan suatu sistem perangkat lunak.

### *Use Case Diagram*

*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan. Menurut *Shalahuddin* dalam jurnal *(Umar Al Faruq, 2015)* mengungkapkan : “Diagram *use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut” [7].

Berikut adalah simbol-simbol yang ada pada *Use Case Diagram* :

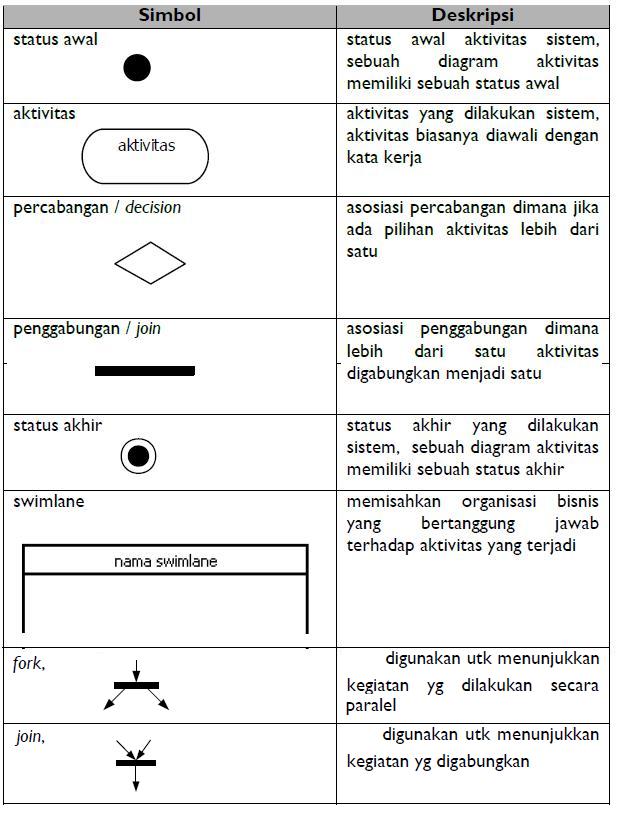


Gambar 2. 2 *Use Case Diagram Symbols*

### *Activity Diagram*

Menurut *Rosa* dalam jurnal *(Sari dan David)* mengungkapkan : “*Activity Diagram* menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem” [7].

Simbol-simbol yang digunakan dalam *activity diagram* sebagai berikut :

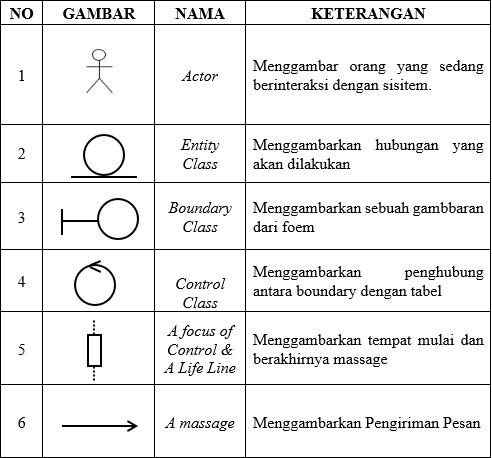


Gambar 2. 3 *Activity Diagram Symbols*

### *Sequence Diagram*

*Sequence Diagram* adalah *tool* yang sangat populer dalam pengembangan sistem informasi secara *object-oriented* untuk menampilkan interaksi antar objek. Berdasarkan definisi tersebut, dapat disimpulkan bahwa *Sequence Diagram* adalah *tool* yang digunakan dalam pengembangan sistem [7].

Berikut adalah simbol-simbol yang digunakan dalam *Sequence diagram* adalah :



Gambar 2. 4 *Sequence Diagram Symbols*

## Penelitian Terdahulu

Pada penelitian terdahulu, dapat disimpulkan bahwa:

Table 2. 1 Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| No | Penulis | Judul | Kesimpulan |
| 1 | Sayeed Al-Aidid, Daniel S. Pamungkas | Sistem Pengenalan Wajah dengan Algoritma *Haar Cascade* dan *Local Binary Pattern Histogram* [1] | Metode *Haar Cascade* digabungkan dengan metode *Local Binary Pattern Histogram* dapat digunakan untuk mendetektsi dan mengenali wajah manusia dengan keberhasilan sebesar 91.67% dari 12 kali percobaan masing-masing dengan jarak yang berbeda. |
| 2 | Al-Azzo, Fadwa  Taqi, Arwa Mohammed  Milanova, Mariofanna | *Human related-health actions detection using Android Camera based on TensorFlow Object Detection API* [2] | Kesimpulannya, tindakan terkait kesehatan dapat dideteksi dengan deteksi kecepatan tinggi dan akurasinya yang luar biasa. Ini bisa mengetahui tindakan yang benar yang diperlukan untuk menangani situasi yang sesuai menggunakan kamera ponsel. |
| 3 | Muchamad Al Amin | Klasifikasi Kelompok Umur Manusia Berdasarkan Analisis Dimensi Fraktal Box Counting dari Citra Wajah Dengan Deteksi Tepi Canny | Dari hasil percobaan dengan 12 nilai yang berbeda dan persentase akurasi optimal sebesar 98,34%, dapat disimpulkan bahwa analisis dimensi fraktal box counting cocok digunakan untuk klasifikasi kelompok umur manusia dari citra wajah dengan deteksi tepi Canny sebagai parameter untuk menganalisis intensitas kerutan wajah. |
| 4 | Rosa Andrie Asmara, Dkk | Klasifikasi Jenis Kelamin pada Citra Wajah Mengunakan Metode *Naïve Bayes* | Klasifikasi dengan metode *Naïve Bayes* hasilnya cukup akurat untuk mengklasifikasikan jenis kelamin berdasarkan nilai probabilitas yang didapatkan pada setiap *gender* laki dan perempuan dengan hasil akurasi sebesar 80%. |
| 5 | Fitri, DKK | Pengunaan Bahasa Pemrograman *Python* Sebagai Pusat Kendali Pada Robot *10-D* | Dari hasil pengujian data pergerakan robot yang diperoleh dapat disimpulkan bahwa *base* *station* dapat menerima dan mengirim *data* dengan baik. *Data* yang dikirim juga dapat dilaksanakan dengan benar pada robot. |
| 6 | Hendriyana, Yazid Hilman Maulana | Identifikasi Jenis Kayu Menggunakan Convolutional Neural Network dengan Arsitektur Mobilenet | Arsitektur Mobilenet pada penelitian ini menghasilkan akurasi training dan testing dalam melakukan identifikasi jenis kayu sebesar 98% training dan 93,3% untuk testing, selain itu untuk recall dan presisinya arsitektur Mobilenet menghasilkan persentase sebesar 28% untuk recall dan 93% untuk presisinya. |
| 7 | Yunahar Heriyanto | Perancangan Sistem Informasi Rental Mobil Berbasis *Web* pada *PT. APM Rent Car* | Dalam membangun *web* pada penelitian ini terdapat dua *web* dengan fungsi berbeda namun saling terkoneksi yaitu *web* untuk pelanggan dan *web* untuk *admin*. *Web* untuk pelanggan digunakan untuk mendapatkan informasi serta melakukan proses rental mobil dan *web* *admin* digunakan untuk mengelola *data* pada *web* rental mobil *PT.APM*. |
| 8 | Ikhsan, Hendra Kurniawan | Implementasi Sistem Kendali Cahaya dan Sirkulasi Udara Ruangan dengan Memanfaatkan *PC* dan Mikrokontroler *ATMEGA8* | Berdasarkan analisa dan hasil penilitian dari pembuatan alat ini disimpulkan bahwa penerapan sensor cahaya *LDR* dapat mendeteksi adanya perubahaan cahaya agar membuka jendela dan tirai sehingga sirkulasi udara dapat bertukar secara teratur dan cahaya dapat masuk kedalam ruangan. |
| 9 | Sarirotul Ilahiyah, Agung Nilogiri | Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network | Hasil uji 40 citra baru yang dilakukan percobaan mendapatkan nilai akurasi yang cukup tinggi. Percobaan mendapatkan prediksi benar sebanyak 36 citra dari 40 citra uji sehingga mendapatkan nilai akurasi sebesar 90 persen. |
| 10 | Muhammad Rizki Muliawan, DKK | Implementasi Pengenalan Wajah Dengan Metode *Eigenface* pada Sistem Absensi | Pada saat *database* yang terdiri dari 20 orang, hasil test menunjukkan hasil yang berbeda antara citra wajah 1 hingga citra wajah 5. Citra wajah 1 menghasilkan persentase sebesar 60%, citra wajah 2 menghasilkan persentase sebesar 50%, citra wajah 3 menghasilkan persentase sebesar 40%, citra wajah 4 menghasilkan persentase sebesar 60%, dan citra wajah 5 meng-hasilkan persentase sebesar 50%. |
| 11 | Mutfah Afrizal Pangestu, Hendra Bunyamin | Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model | Dari tiga pre-trained model dari Keras yang diujikan (ResNet50, Xception dan VGG16), Xception memiliki performa keakuratan yang lebih baik daripada dua model lainnya. Selain itu tersedianya tampilan antar muka juga akan membantu user dalam menggunakan sistem deteksi ras anjing ini. |
| 12 | Khairul Umam, Benny Sukma Negara | Deteksi Objek Manusia Pada Basis Data Video Menggunakan Metode *Backgrond Substraction* dan Operasi Morfologi | Dari pengujian yang dilakukan terlihat bahwa sistem dapat mendeteksi objek pejalan kaki dengan baik pada kondisi pencahayaan terang, dan kurang begitu baik pada saat pencahayaan gelap dan didalam ruangan. |

# **BAB 3** ANALISIS DAN PERANCANGAN

## Analisis Kebutuhan Fungsional dan Non Fungsional

### Kebutuhan Fungsional

Kebutuhan Fungsional merupakan rangkaian kebutuhan yang memiliki keterkaitan dengan sistem. Kebutuhan fungsional dari aplikasi yang dirancang ini adalah sebagai berikut:

1. Sistem dapat membedakan kategori usia seseorang berdasarkan citra wajah.
2. Sistem dapat menambahkan *data* latih gambar untuk mendeteksi kategori usia seseorang.

### Kebutuhan Non-Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang secara tidak langsung terkait dengan sistem.

1. Perangkat keras (*hardware*)

Perangkat keras (*hardware*) yang digunakan dalam perancangan aplikasi ini adalah laptop dengan spesifikasi:

1. *Operating System* : *Windows 10 Home Single Language 64-bit*
2. *Processor*  : *Intel® Core™ i5 8250U*
3. *Memory*  : *12 GB*
4. *VGA* : *NVIDIA GeForce MX130*
5. Perangkat lunak (*software*)

Perangkat lunak (*software*) yang digunakan dalam perancangan aplikasi ini adalah:

1. *Visual Studio Code*
2. *PyQt5*
3. *Terminal*
4. *CUDA* versi 9.0.176
5. *CUDNN* versi 9.0
6. *Brainware*

*Brainware* merupakan setiap orang yang memiliki keterlibatan dalam perancangan maupun penggunaan aplikasi. Untuk penggunaan aplikasi, *brainware* tidak memerlukan keahlian khusus namun setidaknya mengerti bagaimana menggunakan aplikasi berbasis *Desktop*.

1. Bahasa Pemograman

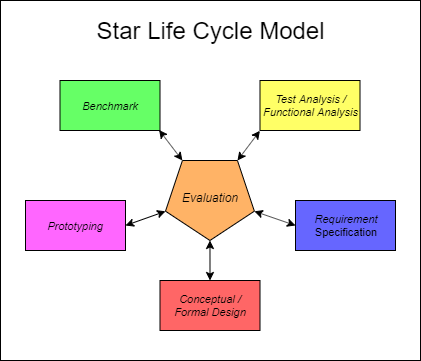
Bahasa Pemograman yang digunakan pada pembuatan *program* deteksi kategori usia manusia adalah *Python* versi 3.6.8. *Module* *python* yang digunakan dalam paket pembuatan *program* ini adalah:



Gambar 3. 1 *Module Python*

## Pemilihan Algoritma/Teknik/Metode

### Model Perancangan Sistem



Gambar 3. 2 *Star Life Cycle Model*

Model perancangan sistem yang digunakan dalam membangun atau merancang aplikasi ini adalah *Star Lifecycle Model*. Penggunaan model *Star Lifecycle* ini dikarenakan pengujian terhadap struktur *program* dilakukan pada banyak faktor sehingga proses pengujian dilakukan secara terus menerus dan banyak dalam satu waktu. Dengan menerapkan model *Star Lifecycle* ini juga menghemat waktu karena tidak perlu menunggu tahapan awal selesai untuk dapat melakukan tahapan selanjutnya sehingga tidak ada hirarki dalam perancangan.

Berikut proses-proses yang ada dan dilakukan pada perancangan sistem dengan *Star Lifecycle Model* ini:

1. *Task Analysis or Functional Analysis*

*Task Analysis* atau *Functional Analysis* adalah identifikasi kemampuan *user*, strategi yang digunakan untuk meningkatkan keterampilannya, alat yang saat ini dipakai, masalah-masalah yang dialami, perubahan yang diinginkan baik dalam ketrampilan maupun peralatan. Metode yang dapat dilakukan adalah dengan mengetahui kemampuan *user*, membuat daftar dengan skala prioritas, dan observasi keterampilan di lapangan.

1. *Requirement Specification*

*Requirement Specification* adalah menentukan kekuatan dan kelemahan suatu spesifikasi yang ada. Metode ini dapat dilakukan dengan meminta *user* untuk mencoba menggunakan berbagai produk yang ada dalam spesifikasi lalu memberitahukan apa kelebihan dan kelemahan dari masing-masing produk.

1. *Conceptual or Formal Design*

*Conceptual or Formal Design* adalah menggunakan hasil analisa untuk membuat alternatif solusi dan saran sampai dengan penentuan pilihan yang terbaik. Metode yang dapat dilakukan adalah dengan bertanya pada *user* sehubungan dengan pengalaman *user* dalam menggunakan prototipe-prototipe yang telah dibuat.

1. Evaluasi dan Validasi

Evaluasi dan Validasi merupakan pemberian masukan kepada *user* secara periodik selama pengembangan dan perancangan dan akan dilakukan terus menerus berdasarkan masukan tadi. Metode yang dapat ditempuh adalah dengan mengamati kebutuhan pokok *user* dalam menggunakan sistem.

1. *Benchmark*

Memadukan hal-hal terbaik yang dimiliki pesaing untuk diterapkan dalam sistem yang dibangun. Metode yang dapat dilakukan adalah dengan menggali informasi dari *user* hal-hal yang sebaiknya ada dan dibandingkan dengan kompetitor.

### Pemilihan Algoritma / Metode

Algoritma yang digunakan pada penulisan skripsi ini adalah

1. *Haar Cascade Classifier*

Algoritma *Haar Cascade Classifier* digunakan saat aplikasi melakukan deteksi kategori usia manusia dengan *data* hasil *training*. Langkah pertama citra gambar diambil dengan *format* warna *BGR*, lalu gambar akan dikonversi menjadi *format* warna *RGB*. Citra lalu akan disesuaikan ukurannya dengan maksimal 900x900 *pixel*. Citra yang sudah disesuaikan ukurannya lalu akan dideteksi dengan algoritma *haar cascade classifier* dan akan diulangi sebanyak 2 kali jika tidak ditemukan citra wajah pada percobaan pertama. Terakhir citra wajah yang ditemukan akan diberikan *bounding box* dan dilakukan perhitungan di tiap-tiap kategori usai manusia yang dibedakan menjadi 20 kelompok dari umur 1 sampai 100 tahun.

1. *Mobilenet Convolutional Neural Network*

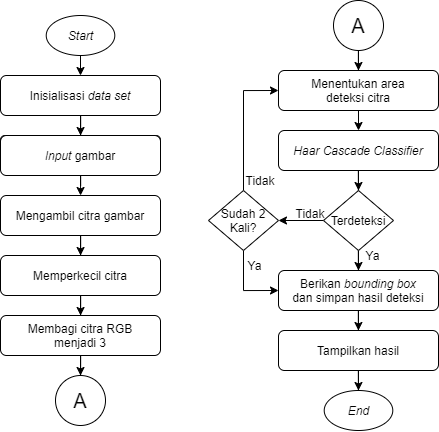
*Mobilenet convolutional neural network* atau *Mobilenet* adalah salah satu arsituktur dan metode terkenal yang ada pada *deep learning*. *Mobilenet* menerapkan algoritma *CNN* seperti konvolusi, *pooling* dalam prosesnya. Pada skripsi ini algoritma *mobilenet* akan digunakan pada *training* citra gambar. Prosesnya sama dengan *CNN* pada umumnya namun pada tahap konvolusi proses dibagi menjadi 2 bagian yaitu *depthwise convolutional* dan *pointwise convolution*.

## Pemilihan Algoritma/Teknik/Metode

### *Flowchart Diagram*

#### **Flowchart Deteksi Usia**

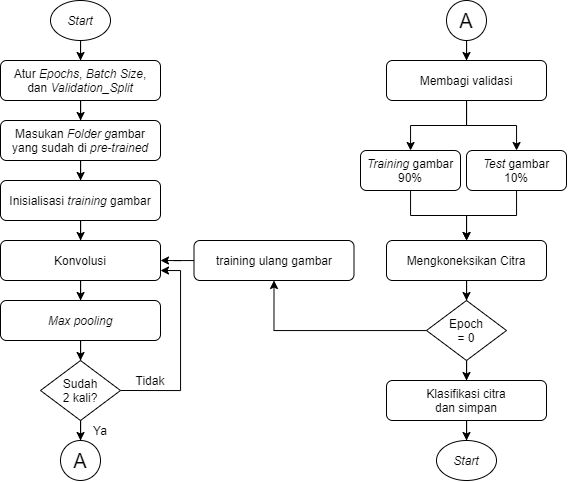
Perancangan *flowchart* *diagram* berikut bertujuan untuk menggambarkan bagaimana aliran proses deteksi kategori usia manusia dilakukan. *Flowchart diagram* dari aplikasi ini ditunjukkan pada Gambar 3.3.



Gambar 3. 3 *Flowchart* Deteksi Kategori Usia

#### **Flowchart Training Gambar**

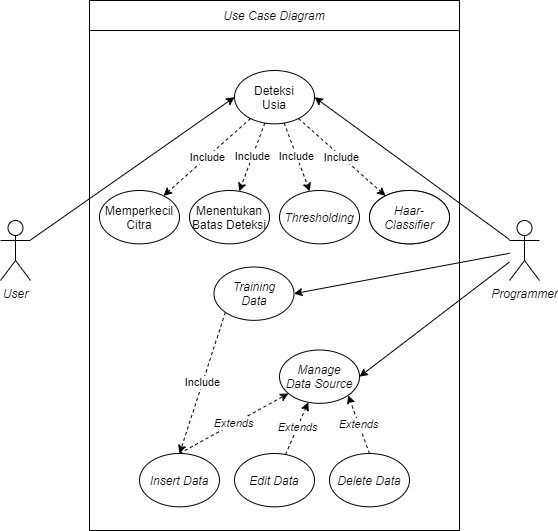
Perancangan *flowchart* *diagram* berikut bertujuan untuk menggambarkan bagaimana sistem melakukan pelatihan gambar. *Flowchart diagram* dari aplikasi ini ditunjukkan pada Gambar 3.4.



Gambar 3. 4 *Flowchart* *Data Training*

### *Use Case Diagram*

*Use* *case* *diagram* berikut mengambarkan apa saja peranan dari *user* dan *programmer* :

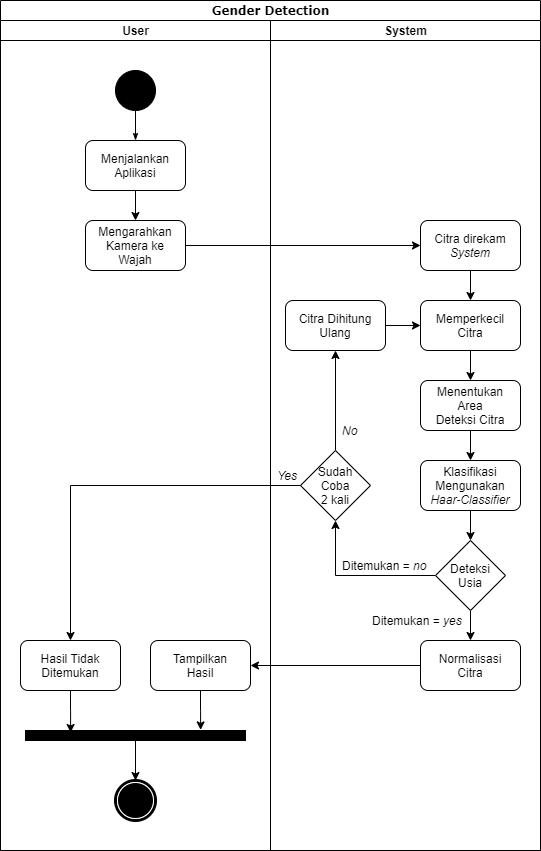


Gambar 3. 5 *Use Case Diagram*

Seperti yang terlihat pada Gambar 3.4, terdapat 2 aktor yaitu *user* dan *programmer* yang berperan aktif dalam *use-case diagram* diatas. *User* disini hanya dapat melakukan deteksi usia yang didalamnya meliputi Memperkecil Citra*, Thresholding,* dan *Classifing*. Sedangkan *programmer* dapat melakukan 3 hal yaitu, deteksi usia seperti *user*, melatih *data,* dan mengatur *database*. *Programmer* dalam melatih *data* dapat memasukan *data* baru ke *database*, dan dalam mengatur *database* dapat memasukan *data* secara manual, merevisi *data*, dan menghapus *data*.

### *Activity Diagram*

Perancangan *activity* *diagram* ini bertujuan untuk menggambarkan aktivitas aktivitas yang terjadi di dalam sistem/aplikasi secara sistematis.

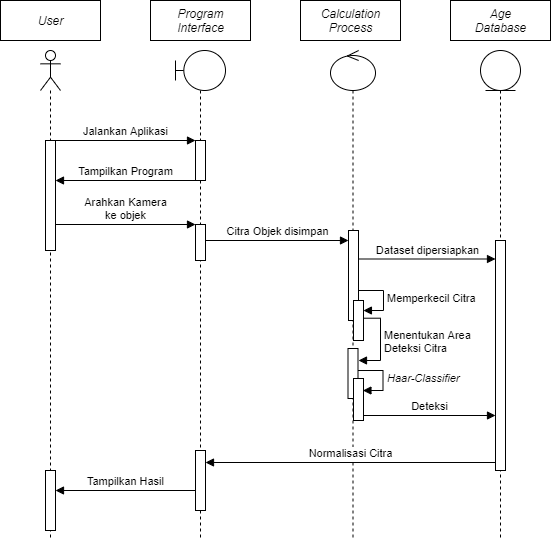


Gambar 3. 6 *Activity Diagram:* Deteksi Usia

Aktivitas yang dilakukan oleh *user* dengan sistem dalam penggunaan aplikasi digambarkan dengan *activity diagram* yang ditunjukkan pada Gambar 3.5.

1. Dalam melakukan deteksi usia, *user* awal mula harus menjalankan aplikasi terlebih dahulu.
2. Setelah aplikasi telah dibuka dan sudah siap, selanjutnya *user* mengarahkan kamera pada citra wajah seseorang.
3. Citra wajah seseorang kemudian akan direkam oleh sistem dan di *preprocessing* terlebih dulu,
4. Citra wajah selanjutnya akan di perkecil jikalau ukurannya melebihi standar pengukuran.
5. Setelah proses pengecilan selesai sistem akan menentukan daerah deteksi citra wajah untuk dihitung.
6. Setelah daerah deteksi telah ditentukan sistem akan mengklasifikasi citra yang akan dideteksi dengan citra-citra lain yang telah terekam dengan mengunakan metode *Haar-Classifier*.
7. Jika deteksi gagal dilakukan maka sistem akan melakukan *preprocessing* ulang dengan algoritma baru, lalu akan balik pada proses memperkecil citra pada langkah ke empat.
8. Jika deteksi untuk citra wajah yang sama gagal untuk dilakukan sebanyak dua kali, maka sistem akan mengeluarkan hasil tidak ditemukan pada *output interface program*.
9. Jika deteksi berhasil maka citra yang sedang diproses akan dinormalisasi sesuai dengan citra yang ditemukan pada *database*.
10. Hasil deteksi usia akan ditampilkan.

### *Sequence Diagram*



Gambar 3. 7 *Sequence Diagram:* Deteksi Usia

*Sequence* *diagram* pada Gambar 3.6 menggambarkan proses aplikasi yang dilakukan oleh *user* dalam mendeteksi usia dimana terdapat 4 objek yang saling berkaitan yaitu *user* sebagai aktor, *program interface* sebagai *object boundary*, *calculation process* sebagai *object control*, dan age *database* sebagai *object entity*.

### *Gantt Chart*



Gambar 3. 8 *Gantt Chart*

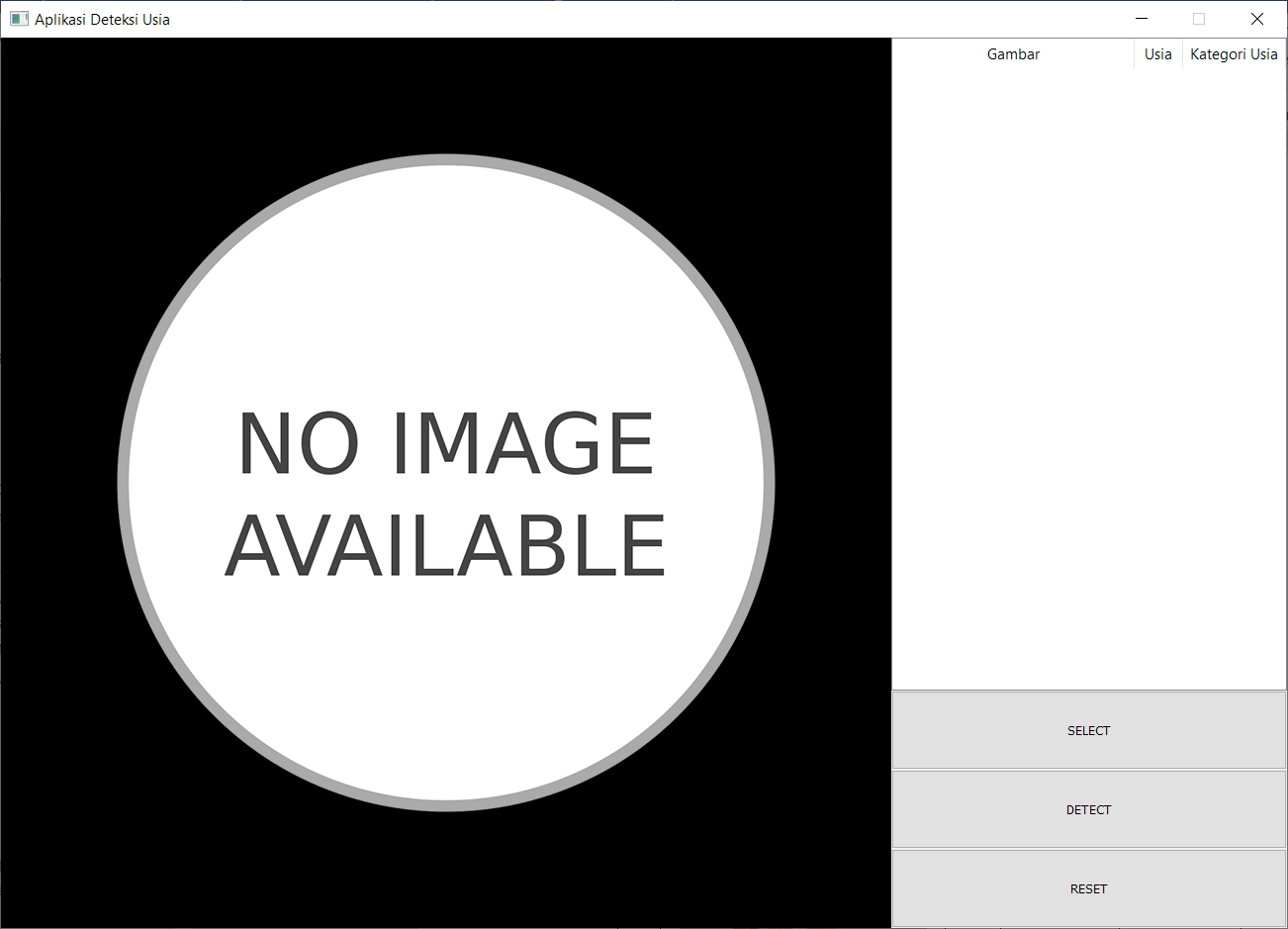


Penjadwalan dalam melakukan segala aktivitas sangatlah menjadi perhatian penting. Jika penjadwalan tidak dilakukan dengan baik, maka hasil yang akan didapatkan tidak akan sesuai dengan harapan. Oleh karena itu kami membuat penjadwalan dalam bentuk *Gantt* *Chart* untuk pencarangan aplikasi berikut.

# **BAB 4** HASIL PENELITIAN

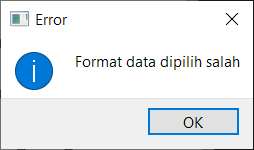
## Implementasi AntarmukaAplikasi

Berikut adalah implementasi antarmuka aplikasi pada skripsi ini,



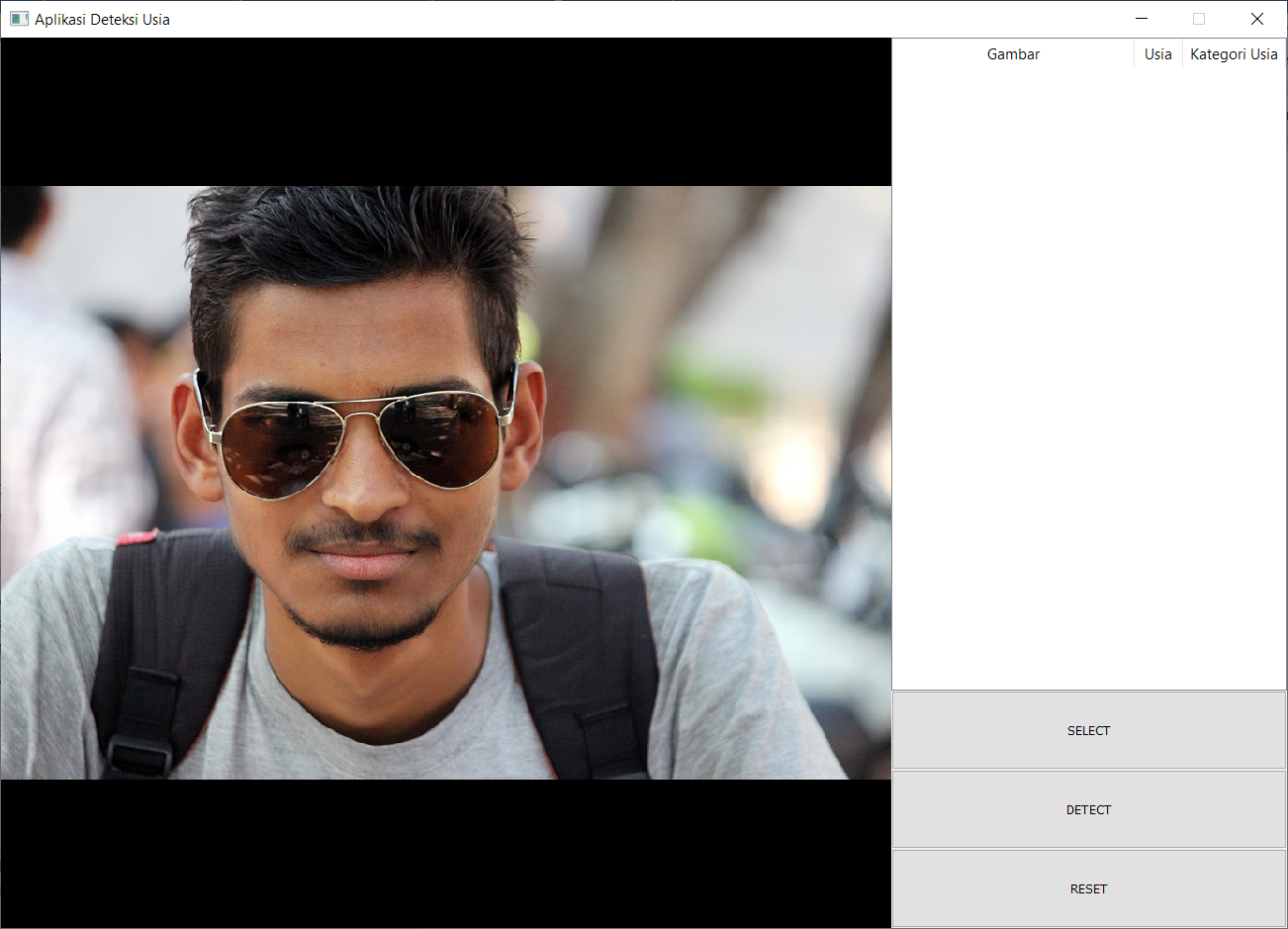
Gambar 4. 1 Tampilan Antarmuka Aplikasi

Aplikasi dibuat menggunakan bahasa pemograman *python* dengan *library* *PyQt5*. Aplikasi memiliki dimensi ukuran 900x1200 pixels dan memiliki 3 *button* pada pojok kanan bawah yaitu *button select, detect,* dan *reset* seperti pada Gambar 4.1. *Button select* memiliki fungsi untuk mengambil gambar yang ada pada *directory* dengan format *.png, .jpg,* dan *.jpeg*.



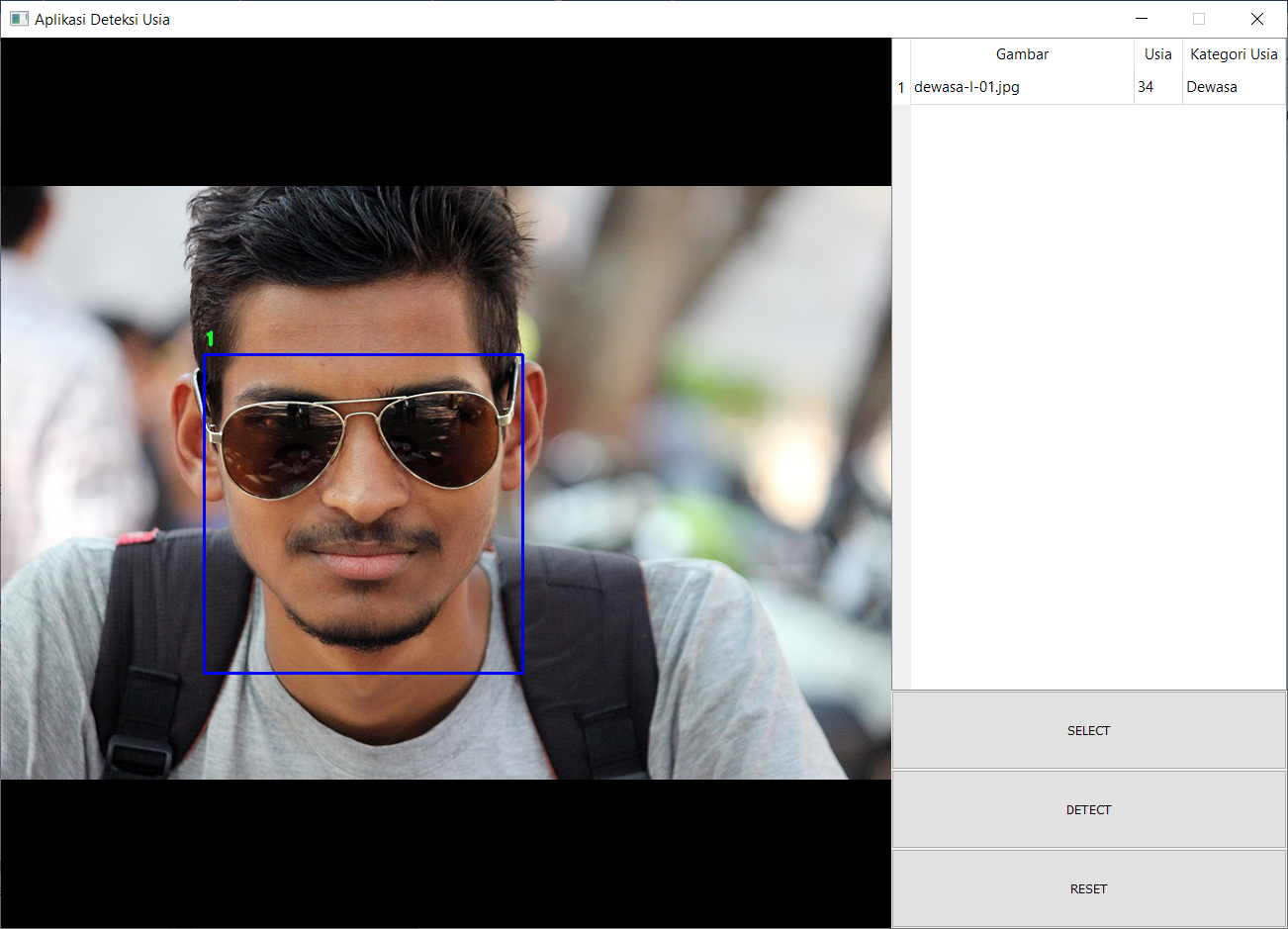
Gambar 4. 2 Tampilan Antarmuka Aplikasi 2

Jika *fomat* yang dipilih bukan *format* *.png, .jpg* dan *.jpeg* maka akan muncul *messagebox* yang berisikan *error* dan informasi mengenai kesalahan *data* seperti pada Gambar 4.2.



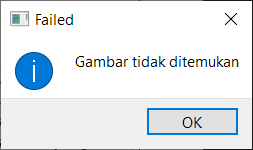
Gambar 4. 3 Tampilan Antarmuka Aplikasi 3

Jika *format* yang dipilih benar *format* *.png, .jpg* dan *.jpeg* maka *imagebox* akan berubah jadi foto yang kita pilih dan ukurannya menyesuaikan dimensi *imagebox* yaitu 900x900 piksel seperti pada Gambar 4.3.



Gambar 4. 4 Tampilan Antarmuka Aplikasi 4

Jika *button detect* ditekan setelah gambar sudah ada di *imagebox* maka akan dilakukan deteksi citra wajah dengan menggunakan algoritma *haar cascade classifier*. Jika citra wajah ditemukan maka akan tergambar *bounding box* pada sekeliling wajah beserta nomor deteksinya dan juga table yang semula kosong akan terisi dengan informasi hasil pengolahan citra wajah seperti Gambar 4.4.



Gambar 4. 5 Tampilan Antarmuka Aplikasi 5

Jika citra wajah tidak ditemukan pada saat *button detect* ditekan, maka akan muncul *messagebox* yang berisikan informasi bahwa citra wajah tidak ditemukan pada gambar seperti Gambar 4.5.

*Button reset* memiliki fungsi untuk menghilangkan isi *table* dan membalikan gambar pada *imagebox* ke asal mulanya.

## Implementasi Metode atau Algoritma

Implementasi metode dan algoritma yang dibuat pada skripsi ini dibedakan menjadi 2 bagian penting yaitu,

### *Predict.py*

*File predict.py* berisikan algoritma perhitungan dalam mendeteksi kategori usia manusia.

#### **Function detectDisplay()**

Code 4. 1 *Function detectDisplay()*

def detectDisplay(imgargs):

    face\_cascade = cv2.CascadeClassifier("C:\\Users\\WILLIAM\\AppData\\Local\\Programs\\Python\\Python36\\Lib\\site-packages\\cv2\\data\\haarcascade\_frontalface\_default.xml")

    img = cv2.imread(imgargs)

    img\_name = os.path.basename(imgargs)

    img = scaling\_image(img, 900)

    img\_data = preprocess(detect\_faces(img))

    results = model.predict(img\_data)

    if results == []:

        return None, None, "error"

    else:

        ages = np.arange(0, 21).reshape(21, 1)

        predicted\_age = str((results[1].dot(ages).flatten())\*4.76).replace('[','').replace(']','').replace('  ',' ').split(' ')

        age\_ktg, age\_int = age\_category(predicted\_age)

        data = []

        item = {"image\_name":img\_name,"age":age\_int,"age\_category":age\_ktg}

        data.append(item)

        with open('result/json/data.json', 'w', encoding='utf-8') as f:

            json.dump(data, f, ensure\_ascii=False)

        tempstring = 'tanggal : '+str(datetime.now().strftime("%d/%m/%Y %H:%M:%S"))+", image : "+img\_name+", age : "+str(age\_int)+", age\_category : "+str(age\_ktg)

        logging.info(tempstring)

        imgsave = scaling\_image(img, 200)

        cv2.imwrite('result/img/'+str(img\_name),imgsave)

        h, w, c = img.shape

        bytesPerLine = 3 \* w

        qim = QImage(img.data, w, h, bytesPerLine, QImage.Format\_RGB888).rgbSwapped()

        pix = QPixmap(qim)

        return pix, item, img\_name

*Function detectDisplay()* pada Code 4.1 merupakan fungsi utama yang menghubungkan fungsi-fungsi lainnya dalam melakaukan deteksi kategori usia manusia. Fungsi ini dipakai untuk melakukan penyimpanan gambar berserta *bounding box* kedalam *directory* *result/img*. Selain itu fungsi ini juga dipakai untuk menyimpan hasil deteksi kategori usia manusia kedalam *file .json* dan *.log*.

#### **Function scaling\_image()**

Code 4. 2 *Function scaling\_image()*

def scaling\_image(img, a):

    h,w,c = img.shape

    if h > a:

        temp = float(a/h)

        h = int(h\*temp)

        w = int(w\*temp)

        img = cv2.resize(img, dsize=(w,h), interpolation=cv2.INTER\_CUBIC)

    if w > a:

        temp = float(a/w)

        h = int(h\*temp)

        w = int(w\*temp)

        img = cv2.resize(img, dsize=(w,h), interpolation=cv2.INTER\_CUBIC)

    return img

*Function scaling\_image()* pada Code 4.2 memiliki fungsi untuk melakukan pengecilan ukuran gambar jika ukuran melebihi *container* yang telah ditetapkan.

#### **Function detect\_faces()**

Code 4. 3 *Function detect\_faces()*

def detect\_faces(img):

    input\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

    detected = detector(input\_img, 1)

    faces = np.empty((len(detected), IMG\_SIZE, IMG\_SIZE, 3))

    img\_h, img\_w, \_ = np.shape(input\_img)

    n = 1

    for i, d in tqdm(enumerate(detected)):

        x1, y1, x2, y2, w, h = d.left(), d.top(), d.right() + 1, d.bottom() + 1, d.width(), d.height()

        xw1 = max(int(x1 - 0.4 \* w), 0)

        yw1 = max(int(y1 - 0.4 \* h), 0)

        xw2 = min(int(x2 + 0.4 \* w), img\_w - 1)

        yw2 = min(int(y2 + 0.4 \* h), img\_h - 1)

        tmp = cv2.rectangle(img, (x1, y1), (x2, y2), (255, 0, 0), 2)

        faces[i, :, :, :] = cv2.resize(img[yw1:yw2 + 1, xw1:xw2 + 1, :], (IMG\_SIZE, IMG\_SIZE))

        cv2.putText(tmp, str(n),(x1, y1-10), cv2.FONT\_HERSHEY\_DUPLEX, 0.6, (36,255,12), 2)

        n+=1

    return faces

*Function detect\_faces()* pada Code 4.3 memiliki fungsi untuk melakukan perhitungan awal dalam deteksi citra wajah. Selain itu fungsi ini dapat megambarkan *bounding box* pada sekeliling citra wajah dan memberikan penomoran pada sisi atas kiri *bounding box*.

#### **Function age\_category()**

Code 4. 4 *Function age\_category()*

def age\_category(age):

    ktg = []

    ages = []

    for item in age:

        if item != "":

            temp = int(float(item))

            ages.append(str(temp))

            if temp <= 5:

                ktg.append('Balita')

            elif temp <= 11:

                ktg.append('Kanak-kanak')

            elif temp <= 25:

                ktg.append('Remaja')

            elif temp <= 45:

                ktg.append('Dewasa')

            elif temp <= 65:

                ktg.append('Lansia')

            elif temp > 65:

                ktg.append('Manula')

    return ktg, ages

*Function age\_category()* pada Code 4.4 memiliki fungsi untuk mengklasifikasikan usia seseorang dalam angka menjadi kategori usia yang dibedakan menjadi 6 kategori yaitu balita, kanak-kanak, remaja, dewasa, lansia, dan manula.

### *Aplikasi-deteksi-usia.py*

class Ui\_MainWindow(object):

    img\_path = ""

    img\_name = ""

    def setupUi(self, MainWindow):

        MainWindow.setObjectName("MainWindow")

        MainWindow.setFixedSize(1300,900)

        self.centralwidget = QtWidgets.QWidget(MainWindow)

        self.centralwidget.setObjectName("centralwidget")

        self.photo = QtWidgets.QLabel(self.centralwidget)

        self.photo.setGeometry(QtCore.QRect(0, 0, 900, 900))

        self.photo.setMinimumSize(QtCore.QSize(0, 0))

        self.photo.setMaximumSize(QtCore.QSize(16777215, 16777215))

        self.photo.setText("")

        self.photo.setPixmap(QtGui.QPixmap("src/no-image.png"))

        self.photo.setScaledContents(False)

        self.photo.setAlignment(QtCore.Qt.AlignCenter)

        self.photo.setWordWrap(False)

        self.photo.setOpenExternalLinks(False)

        self.photo.setObjectName("photo")

        self.photo.setStyleSheet("background-color: black;")

        self.table = QtWidgets.QTableWidget(self.centralwidget)

        self.table.setGeometry(QtCore.QRect(900, 0, 400, 660))

        self.table.setRowCount(0)

        self.table.setColumnCount(4)

        self.table.setObjectName("table")

        self.table.setHorizontalHeaderLabels(['No','Gambar','Usia','Kategori Usia'])

        header = self.table.horizontalHeader()

        header.setSectionResizeMode(0, QtWidgets.QHeaderView.Stretch)

        header.setSectionResizeMode(1, QtWidgets.QHeaderView.Stretch)

        header.setSectionResizeMode(2, QtWidgets.QHeaderView.ResizeToContents)

        header.setSectionResizeMode(3, QtWidgets.QHeaderView.ResizeToContents)

        self.table.setColumnHidden(0,True)

        self.table.setEditTriggers(QtWidgets.QAbstractItemView.NoEditTriggers)

        self.select = QtWidgets.QPushButton(self.centralwidget)

        self.select.setGeometry(QtCore.QRect(900, 660, 400, 80))

        self.select.setObjectName("select")

        self.detect = QtWidgets.QPushButton(self.centralwidget)

        self.detect.setGeometry(QtCore.QRect(900, 740, 400, 80))

        self.detect.setObjectName("detect")

        self.reset = QtWidgets.QPushButton(self.centralwidget)

        self.reset.setGeometry(QtCore.QRect(900, 820, 400, 80))

        self.reset.setObjectName("reset")

        MainWindow.setCentralWidget(self.centralwidget)

        self.menubar = QtWidgets.QMenuBar(MainWindow)

        self.menubar.setGeometry(QtCore.QRect(0, 0, 900, 26))

        self.menubar.setObjectName("menubar")

        MainWindow.setMenuBar(self.menubar.hide())

        self.statusbar = QtWidgets.QStatusBar(MainWindow)

        self.statusbar.setObjectName("statusbar")

        MainWindow.setStatusBar(self.statusbar.hide())

        self.retranslateUi(MainWindow)

        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):

        \_translate = QtCore.QCoreApplication.translate

        MainWindow.setWindowTitle(\_translate("MainWindow", "Aplikasi Deteksi Usia"))

        self.select.setText(\_translate("MainWindow", "SELECT"))

        self.detect.setText(\_translate("MainWindow", "DETECT"))

        self.reset.setText(\_translate("MainWindow", "RESET"))

        self.select.clicked.connect(self.select\_handler)

        self.detect.clicked.connect(self.detect\_handler)

        self.reset.clicked.connect(self.reset\_handler)

    def reset\_handler(self):

        Ui\_MainWindow.img\_name = ""

        Ui\_MainWindow.img\_path = ""

        self.photo.clear()

        self.photo.setPixmap(QtGui.QPixmap("src/no-image.png"))

        self.table.clearContents()

        self.table.setRowCount(0)

    def detect\_handler(self):

        if Ui\_MainWindow.img\_path == "":

            msg = QMessageBox()

            msg.setIcon(QMessageBox.Information)

            msg.setText("Gambar belum dipilih")

            msg.setWindowTitle("Error")

            msg.exec()

        else:

            pixmap, itemlist, Ui\_MainWindow.img\_name = detectDisplay(Ui\_MainWindow.img\_path)

            if Ui\_MainWindow.img\_name == "error":

                msg = QMessageBox()

                msg.setIcon(QMessageBox.Information)

                msg.setText("Gambar tidak ditemukan")

                msg.setWindowTitle("Failed")

                msg.exec()

            else:

                if pixmap.width() > 900 or pixmap.height() > 900:

                    pixmap = pixmap.scaled(900,900,QtCore.Qt.KeepAspectRatio)

                self.photo.setPixmap(pixmap)

                self.table.setRowCount(0)

                row = self.table.rowCount()

                for i, items in enumerate(itemlist['age']):

                    self.table.insertRow(row)

                    self.table.setItem(row, 0, QtWidgets.QTableWidgetItem(str(i+1)))

                    self.table.setItem(row, 1, QtWidgets.QTableWidgetItem(Ui\_MainWindow.img\_name))

                    self.table.setItem(row, 2, QtWidgets.QTableWidgetItem(itemlist['age'][i]))

                    self.table.setItem(row, 3, QtWidgets.QTableWidgetItem(itemlist['age\_category'][i]))

                self.table.sortItems(0, QtCore.Qt.AscendingOrder)

    def select\_handler(self):

        name = self.open\_dialog\_box()

        if name[0] == "" or name[1] == "":

            None

        elif name[0].endswith(('.png', '.jpg', '.jpeg')):

            self.table.clearContents()

            self.table.setRowCount(0)

            self.photo.clear()

            pixmap = QtGui.QPixmap(name[0], "1")

            if pixmap.width() > 900 or pixmap.height() > 900:

                pixmap = pixmap.scaled(900,900,QtCore.Qt.KeepAspectRatio)

            self.photo.setPixmap(pixmap)

            Ui\_MainWindow.img\_path = str(name[0])

        else:

            msg = QMessageBox()

            msg.setIcon(QMessageBox.Information)

            msg.setText("Format data dipilih salah")

            msg.setWindowTitle("Error")

            msg.exec()

    def open\_dialog\_box(self):

        filename = QFileDialog.getOpenFileName()

        return filename

if \_\_name\_\_ == "\_\_main\_\_":

    import sys

    app = QtWidgets.QApplication(sys.argv)

    MainWindow = QtWidgets.QMainWindow()

    ui = Ui\_MainWindow()

    ui.setupUi(MainWindow)

    MainWindow.show()

    sys.exit(app.exec\_())

*File aplikasi-deteksi-usia.py* berisikan penulisan bahasa *program* untuk menjalankan *interface* beserta *widget* yang ada didalamnya. *File* ini memiliki *class* yang berisikan *object* beserta pengaturannya dan *function.* Terdapat 3 *function* utama dalam class ini yaitu *select\_handler(), detect\_handler()* dan *reset\_handler()*.

*Select\_handler()* memiliki fungsi untuk memilih file gambar yang ingin dideteksi. *Detect\_handler()* memiki fungsi untuk melakukan deteksi pada gambar yang sudah dipilih sebelumnya pada fungsi *select\_handler()*. *Reset\_handler()* memiliki fungsi untuk melakukan pembersihan variabel dan objek dengan membalikan pada keadaan semula pada saat *program* dijalankan.

## Pengujian Aplikasi dan Algoritma

Pengujian aplikasi dan algoritma deteksi kategori usia manusia dibedakan menjadi beberapa bagian yaitu,

### Pengujian Pada Kategori Usia Balita

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **Usia** | **Kategori** | **Benar/Salah** |
| 1 |  | 14 | Remaja | Salah |
| 2 |  | 13 | Remaja | Salah |
| 3 |  | 15 | Remaja | Salah |
| 4 |  | 26 | Dewasa | Salah |
| 5 |  | 17 | Remaja | Salah |
| 6 |  | 42 | Dewasa | Salah |
| 7 |  | 19 | Remaja | Salah |
| 8 |  | 15 | Remaja | Salah |
| 9 |  | 19 | Remaja | Salah |
| 10 |  | 19 | Remaja | Salah |
| 11 |  | 13 | Remaja | Salah |
| 12 |  | 33 | Dewasa | Salah |
| 13 |  | 36 | Dewasa | Salah |
| 14 |  | 18 | Remaja | Salah |
| 15 |  | 16 | Remaja | Salah |
| 16 |  | 18 | Remaja | Salah |
| 17 |  | 17 | Remaja | Salah |
| 18 |  | 12 | Remaja | Salah |
| 19 |  | 18 | Remaja | Salah |
| 20 |  | 18 | Remaja | Salah |

Table 4. 1 *Table* Pengujian Pada Kategori Usia Balita

Table 4.1 merupakan hasil pengujian deteksi kategori usia manusia pada kategori usia balita yaitu dengan keakuratan deteksi sebesar 0% dari 20 foto yang terdiri dari 10 foto laki-laki dan 10 foto perempuan.

### Pengujian Pada Kategori Usia Kanak-kanak

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **Usia** | **Kategori** | **Benar/Salah** |
| 1 |  | 29 | Dewasa | Salah |
| 2 |  | 13 | Remaja | Salah |
| 3 |  | 16 | Remaja | Salah |
| 4 |  | 14 | Remaja | Salah |
| 5 |  | 16 | Remaja | Salah |
| 6 |  | 30 | Dewasa | Salah |
| 7 |  | 22 | Remaja | Salah |
| 8 |  | 15 | Remaja | Salah |
| 9 |  | 19 | Remaja | Salah |
| 10 |  | 23 | Remaja | Salah |
| 11 |  | 40 | Dewasa | Salah |
| 12 |  | 15 | Remaja | Salah |
| 13 |  | 16 | Remaja | Salah |
| 14 |  | 15 | Remaja | Salah |
| 15 |  | 20 | Remaja | Salah |
| 16 |  | 31 | Dewasa | Salah |
| 17 |  | 14 | Remaja | Salah |
| 18 |  | 14 | Remaja | Salah |
| 19 |  | 29 | Dewasa | Salah |
| 20 |  | 17 | Remaja | Salah |

Table 4. 2 *Table* Pengujian Pada Kategori Usia Kanak-kanak

Table 4.2 merupakan hasil pengujian deteksi kategori usia manusia pada kategori usia kanak-kanak yaitu dengan keakuratan deteksi sebesar 0% dari 20 foto yang terdiri dari 10 foto laki-laki dan 10 foto perempuan.

### Pengujian Pada Kategori Usia Remaja

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **Usia** | **Kategori** | **Benar/Salah** |
| 1 |  | 19 | Remaja | Benar |
| 2 |  | 20 | Remaja | Benar |
| 3 |  | 19 | Remaja | Benar |
| 4 |  | 19 | Remaja | Benar |
| 5 |  | 19 | Remaja | Benar |
| 6 |  | 15 | Remaja | Benar |
| 7 |  | 15 | Remaja | Benar |
| 8 |  | 17 | Remaja | Benar |
| 9 |  | 21 | Remaja | Benar |
| 10 |  | 20 | Remaja | Benar |
| 11 |  | 24 | Remaja | Benar |
| 12 |  | 22 | Remaja | Benar |
| 13 |  | 14 | Remaja | Benar |
| 14 |  | 18 | Remaja | Benar |
| 15 |  | 16 | Remaja | Benar |
| 16 |  | 24 | Remaja | Benar |
| 17 |  | 18 | Remaja | Benar |
| 18 |  | 23 | Remaja | Benar |
| 19 |  | 21 | Remaja | Benar |
| 20 |  | 24 | Remaja | Benar |

Table 4. 3 *Table* Pengujian Pada Kategori Usia Remaja

Table 4.3 merupakan hasil pengujian deteksi kategori usia manusia pada kategori usia remaja yaitu dengan keakuratan deteksi sebesar 100% dari 20 foto yang terdiri dari 10 foto laki-laki dan 10 foto perempuan.

### Pengujian Pada Kategori Usia Dewasa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **Usia** | **Kategori** | **Benar/Salah** |
| 1 |  | 34 | Dewasa | Benar |
| 2 |  | 28 | Dewasa | Benar |
| 3 |  | 22 | Remaja | Salah |
| 4 |  | 41 | Dewasa | Benar |
| 5 |  | 38 | Dewasa | Benar |
| 6 |  | 41 | Dewasa | Benar |
| 7 |  | 41 | Dewasa | Benar |
| 8 |  | 41 | Dewasa | Benar |
| 9 |  | 43 | Dewasa | Benar |
| 10 |  | 28 | Dewasa | Benar |
| 11 |  | 42 | Dewasa | Benar |
| 12 |  | 37 | Dewasa | Benar |
| 13 |  | 34 | Dewasa | Benar |
| 14 |  | 31 | Dewasa | Benar |
| 15 |  | 28 | Dewasa | Benar |
| 16 |  | 29 | Dewasa | Benar |
| 17 |  | 27 | Dewasa | Benar |
| 18 |  | 30 | Dewasa | Benar |
| 19 |  | 28 | Dewasa | Benar |
| 20 |  | 27 | Dewasa | Benar |

Table 4. 4 *Table* Pengujian Pada Kategori Usia Dewasa

Table 4.4 merupakan hasil pengujian deteksi kategori usia manusia pada kategori usia dewasa yaitu dengan keakuratan deteksi sebesar 95% dari 20 foto yang terdiri dari 10 foto laki-laki dan 10 foto perempuan. Foto laki-laki memilki keakuratan sebesar 90% dan foto perempuan memiliki keakuratan sebesar 100%.

### Pengujian Pada Kategori Usia Lansia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **Usia** | **Kategori** | **Benar/Salah** |
| 1 |  | 50 | Lansia | Benar |
| 2 |  | 55 | Lansia | Benar |
| 3 |  | 63 | Lansia | Benar |
| 4 |  | 62 | Lansia | Benar |
| 5 |  | 62 | Lansia | Benar |
| 6 |  | 55 | Lansia | Benar |
| 7 |  | 59 | Lansia | Benar |
| 8 |  | 63 | Lansia | Benar |
| 9 |  | 65 | Lansia | Benar |
| 10 |  | 55 | Lansia | Benar |
| 11 |  | 54 | Lansia | Benar |
| 12 |  | 56 | Lansia | Benar |
| 13 |  | 54 | Lansia | Benar |
| 14 |  | 61 | Lansia | Benar |
| 15 |  | 63 | Lansia | Benar |
| 16 |  | 65 | Lansia | Benar |
| 17 |  | 44 | Dewasa | Salah |
| 18 |  | 54 | Lansia | Benar |
| 19 |  | 44 | Dewasa | Salah |
| 20 |  | 61 | Lansia | Benar |

Table 4. 5 *Table* Pengujian Pada Kategori Usia Lansia

Table 4.5 merupakan hasil pengujian deteksi kategori usia manusia pada kategori usia lansia yaitu dengan keakuratan deteksi sebesar 90% dari 20 foto yang terdiri dari 10 foto laki-laki dan 10 foto perempuan. Foto laki-laki memilki keakuratan sebesar 100% dan foto perempuan memiliki keakuratan sebesar 80%.

### Pengujian Pada Kategori Usia Manula

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **Usia** | **Kategori** | **Benar/Salah** |
| 1 |  | 67 | Manula | Benar |
| 2 |  | 66 | Manula | Benar |
| 3 |  | 67 | Manula | Benar |
| 4 |  | 64 | Lansia | Salah |
| 5 |  | 64 | Lansia | Salah |
| 6 |  | 63 | Lansia | Salah |
| 7 |  | 59 | Lansia | Salah |
| 8 |  | 62 | Lansia | Salah |
| 9 |  | 68 | Manula | Benar |
| 10 |  | 67 | Manula | Benar |
| 11 |  | 57 | Lansia | Salah |
| 12 |  | 63 | Lansia | Salah |
| 13 |  | 58 | Lansia | Salah |
| 14 |  | 67 | Manula | Benar |
| 15 |  | 69 | Manula | Benar |
| 16 |  | 60 | Lansia | Salah |
| 17 |  | 62 | Lansia | Salah |
| 18 |  | 63 | Lansia | Salah |
| 19 |  | 65 | Lansia | Salah |
| 20 |  | 68 | Manula | Benar |

Table 4. 6 *Table* Pengujian Pada Kategori Usia Manula

Table 4.6 merupakan hasil pengujian deteksi kategori usia manusia pada kategori usia manula yaitu dengan keakuratan deteksi sebesar 40% dari 20 foto yang terdiri dari 10 foto laki-laki dan 10 foto perempuan. Foto laki-laki memilki keakuratan sebesar 50% dan foto perempuan memiliki keakuratan sebesar 30%.

### Pengujian Pada Banyak Orang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Gambar** | **No.**  **img** | **Program** | **Manual** |
| 1 |  | 1 | Dewasa | Dewasa |
| 2 | Dewasa | Dewasa |
|  |  |  |
| 2 |  | 1 | Dewasa | Remaja |
| 2 | Remaja | Remaja |
| 3 | Dewasa | Remaja |
| 4 | Dewasa | Remaja |
| 5 | Remaja | Kanak-kanak |
| 6 | Dewasa | Balita |
|  |  |  |
| 3 |  | 1 | Dewasa | Dewasa |
| 2 | Remaja | Kanak-kanak |
| 3 | Dewasa | Dewasa |
| 4 | Remaja | Kanak-kanak |
| 5 | Dewasa | Remaja |
| 6 | Dewasa | Balita |
|  |  |  |
| 4 |  | 1 | Dewasa | Dewasa |
| 2 | Lansia | Dewasa |
| 3 | Dewasa | Dewasa |
| 4 | Remaja | Remaja |
| 5 | Dewasa | Balita |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 |  | 1 | Dewasa | Dewasa |
| 2 | Remaja | Remaja |
| 3 | Dewasa | Kanak-kanak |
| 4 | Dewasa | Dewasa |
| 5 | Dewasa | Balita |
|  |  |  |
| 6 |  | 1 | Dewasa | Kanak-kanak |
| 2 | Dewasa | Dewasa |
| 3 | Dewasa | Dewasa |
| 4 | Remaja | Kanak-kanak |
|  |  |  |
| 7 |  | 1 | Dewasa | Remaja |
| 2 | Dewasa | Dewasa |
| 3 | Remaja | Balita |
| 4 | Remaja | Kanak-kanak |
| 5 | Remaja | Dewasa |
|  |  |  |
| 8 |  | 1 | Dewasa | Dewasa |
| 2 | Dewasa | Dewasa |
| 3 | Dewasa | Balita |
|  |  |  |
| 9 |  | 1 | Dewasa | Dewasa |
| 2 | Remaja | Kanak-kanak |
| 3 | Dewasa | Dewasa |
| 4 | Dewasa | Dewasa |
| 5 | Dewasa | Dewasa |
| 6 | Remaja | Kanak-kanak |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 |  | 1 | Dewasa | Kanak-kanak |
| 2 | Remaja | Dewasa |
| 3 | Dewasa | Lansia |
| 4 | Dewasa | Lansia |
| 5 | Dewasa | Dewasa |
| 6 | Remaja | Kanak-kanak |
|  |  |  |
| 11 |  | 1 | Remaja | Kanak-kanak |
| 2 | Dewasa | Dewasa |
| 3 | Dewasa | Dewasa |
| 4 | Remaja | Kanak-kanak |
|  |  |  |
| 12 |  | 1 | Dewasa | Remaja |
| 2 | Dewasa | Dewasa |
| 3 | Dewasa | Kanak-kanak |
| 4 | Dewasa | Remaja |
| 5 | Dewasa | Dewasa |
|  |  |  |
| 13 |  | 1 | Remaja | Dewasa |
| 2 | Dewasa | Dewasa |
| 3 | Remaja | Balita |
| 4 | Remaja | Kanak-kanak |
|  |  |  |
| 14 |  | 1 | Lansia | Dewasa |
| 2 | Dewasa | Dewasa |
| 3 | Dewasa | Dewasa |
| 4 | Dewasa | Dewasa |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15 |  | 1 | Dewasa | Dewasa |
| 2 | Remaja | Balita |
| 3 | Dewasa | Dewasa |
|  |  |  |
| 16 |  | 1 | Remaja | Dewasa |
| 2 | Remaja | Dewasa |
| 3 | Dewasa | Balita |
|  |  |  |
| 17 |  | 1 | Dewasa | Lansia |
| 2 | Dewasa | Dewasa |
| 3 | Remaja | Dewasa |
| 4 | Lansia | Lansia |
| 5 | Remaja | Kanak-kanak |
|  |  |  |
| 18 |  | 1 | Dewasa | Dewasa |
| 2 | Remaja | Kanak-kanak |
| 3 | Dewasa | Dewasa |
|  |  |  |
| 19 |  | 1 | Remaja | Dewasa |
| 2 | Remaja | Dewasa |
| 3 | Remaja | Remaja |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 20 |  | 1 | Remaja | Dewasa |
| 2 | Dewasa | Dewasa |
| 3 | Dewasa | Balita |
| 4 | Remaja | Balita |
|  |  |  |

Table 4. 7 *Table* Pengujian Pada Banyak Orang

Table 4.7 merupakan hasil pengujian deteksi kategori usia manusia pada banyak orang dan mendapatkan hasil yaitu,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Total img** | **Balita** | **Kanak kanak** | **Remaja** | **Dewasa** | **Lansia** | **Manula** | **Akurat** |
| 1 | 2 |  |  |  | 2 |  |  | 100 |
| 2 | 6 |  |  | 1 |  |  |  | 16.67 |
| 3 | 6 |  |  |  | 2 |  |  | 33.33 |
| 4 | 5 |  |  | 1 | 2 |  |  | 60.00 |
| 5 | 5 |  |  | 1 | 2 |  |  | 60.00 |
| 6 | 4 |  |  |  | 2 |  |  | 50.00 |
| 7 | 5 |  |  |  | 1 |  |  | 20.00 |
| 8 | 3 |  |  |  | 2 |  |  | 66.67 |
| 9 | 6 |  |  |  | 4 |  |  | 66.67 |
| 10 | 6 |  |  |  | 1 |  |  | 16.67 |
| 11 | 4 |  |  |  | 2 |  |  | 50.00 |
| 12 | 5 |  |  |  | 2 |  |  | 40.00 |
| 13 | 4 |  |  |  | 1 |  |  | 25.00 |
| 14 | 4 |  |  |  | 3 |  |  | 75.00 |
| 15 | 3 |  |  |  | 2 |  |  | 66.67 |
| 16 | 3 |  |  |  |  |  |  | 0 |
| 17 | 5 |  |  |  | 1 | 1 |  | 40.00 |
| 18 | 3 |  |  |  | 2 |  |  | 66.67 |
| 19 | 3 |  |  | 1 |  |  |  | 33.33 |
| 20 | 4 |  |  |  | 1 |  |  | 25.00 |
| **Persentase ketepatan** | | | | | | | | 45.58 |

Table 4. 8 *Table* Perhitungan Pengujian Pada Banyak Orang

Table 4.8 adalah hasil perhitungan pada pengujian yang ada pada Table 4.7. Pada perhitungan pengujian didapatkan persentase ketepatan sebesar 45,58% dari 86 citra wajah yang ada pada 20 foto. 37,21% diantaranya itu didapatkan dari deteksi pada kategori usia dewasa, 4665% dari kategori usia remaja, dan 1.16% dari kategori usia lansia.

# **BAB 5** KESIMPULAN DAN SARAN

## Simpulan

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan maka dapat disimpulkan sebagai berikut:

1. Aplikasi dapat melakukan deteksi kategori usia dengan baik pada rentang kategori remaja – lansia dengan persentase keberhasilan sebesar 95%.
2. Aplikasi memiliki kesulitan dalam mendeteksi citra wajah dengan kategori usia manula demgan persemtase keberhasilan sebesar 30%.
3. Aplikasi tidak dapat melakukan deteksi pada kategori usia balita dan kanak – kanak.
4. Aplikasi memiliki kesulitan dalam mendeteksi citra wajah pada banyak orang yaitu dengan persentase keberhasilan sebesar 45.58%.

## Saran

Dengan dibuatnya skripsi ini diharapkan aplikasi deteksi kategori usia sekarang dapat dikembangkan lagi menjadi lebih baik lagi, yaitu dengan saran:

1. Algoritma *Mobilenet-CNN* kurang baik digunakan dalam *training* gambar yang menghasilkan hasil berupa usia.
2. Jumlah gambar yang digunakan untuk training bisa ditingkatkan lagi.
3. Spesifikasi *GPU* untuk melakukan *training* dapat ditingkan lagi lebih besar agar dapat mengatur *epoch* dan *batch file* pada saar *training*.
4. Integrasi dengan perangkat lain seperti *android*, *web*, dan *smart device* lainnya dapat dikembangkan.

# **BAB 6** DAFTAR PUSTAKA

[1] Al-Aidid. S., dan Pamungkas. D. S. (2016). Sistem Pengenalan Wajah dengan Algoritma *Haar Cascade* dan *Local Binary Pattern* Histogram. *Jurnal Rekayasa Elektrika*. Vol. 14, No. 1, pp. 62-67. DOI: 10.17529/jre.v14i1.9799, ISSN: 1412-4785, e-ISSN: 2252-620X.

[2] Al-Azzo. F., Taqi. A.M., and Milanova. M. (2018). *Human Related-Health Actions Detection Using Android Camera Based on TensorFlow Object Detection API*. *International Journal of Advanced Computer Science and Applications (IJACSA)*. Vol. 9, No. 10, pp. 9–23. DOI: 10.14569/ijacsa.2018.091002, p-ISSN: 2156-5570.

[3] Amin. M. A., & Juniati. D. (2017). Klasifikasi Kelompok Umur Manusia Berdasarkan Analisis Dimensi Fraktal Box Counting dari Citra Wajah Dengan Deteksi Tepi Canny. *MATHunesa (Jurnal Ilmiah Matematika)*. Vol. 2, No. 6, pp. 33-42. ISSN : 2301-9115.

[4] Asmara. R. A., Andjani. B. S., Rosiani. U. D., dan Choirina. P. (2018). Klasifikasi Jenis Kelamin pada Citra Wajah Menggunakan Metode *Naïve Bayes*. *Jurnal Informatika Polinema.* Vol. 4, No. 3, pp. 212-217. e-ISSN: 2407-070X, p-ISSN: 2614-6371.

[5] Fitri, Reski. K. R., Rahmansyah. A., & Darwin. W. (2017, Juli). Pengunaan Bahasa Pemograman Python Sebagai Pusat Kendali Pada Robot 10-D. *5th Indonesian Symposium on Robotic Systems and Control*. No. 6, pp. 23-26. ISBN: 978-602-72004-3-2.

[6] Hendriyana, & Maulana. Y. H. (2020, Februari). Identifikasi Jenis Kayu Menggunakan *Convolutional Neural Network* dengan Arsitektur *Mobilenet*. *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*. Vol. 4, No. 1, pp. 70-70. e-ISSN : 2580-0760.

[7] Heriyanto. Y. (2018, Oktober). Perancangan Sistem Informasi Rental Mobil Berbasis *Web* Pada *PT. APM Rental Car*. *Jurnal Intra-Tech*, Vol. 2, No. 2, pp. 64-77, ISSN: 2549-0222.

[8] Ikhsan, dan Kurniawan. H. (2015). Implementasi Sistem Kendali Cahaya dan Sirkulasi Udara Ruangan Dengan Memanfaatkan *PC* dan Mikrokontroler *ATMEGA8*. *Jurnal TeknoIF*. Vol. 3, No. 1, pp. 12-19. ISSN: 2338-2724.

[9] Ilahiyah. S., & Nilogiri. A. (2018, Agustus). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem & Teknologi Informasi Indonesia)*. Vol. 3, No. 2, pp. 49-56. p-ISSN : 2502-5724, e-ISSN : 2541-5735.

[10] Muliawan. M. R., Irawan. B., dan Brianorman. Y. (2015). Implementasi Pengenalan Wajah Dengan Metode *Eigenface* Pada Sistem Absensi. *Jurnal Coding Sistem Komputer UNTAN*. Vol. 3, No. 1. ISSN: 2238-493X.

[11] Pangestu. M. A., & Bunyamin. H. (2018, Agustus). Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Jurnal Teknik Informatika dan Sistem Informasi*. Vol. 4, No. 2, pp. 337-334. p-ISSN: 2443-2210, e-ISSN: 2443-2229, DOI: 10.28932/jutisi.v4i2.828.

[12] Umam. K., & Negara, B. S. (2016, Desember). Deteksi Objek Manusia Pada basis Data Video Menggunakan Metode Background *Substraction* dan Operasi Morfologi. *Jurnal CoreIT*, Vol. 2, No. 2, pp. 31-40. p-ISSN: 2460-738X.