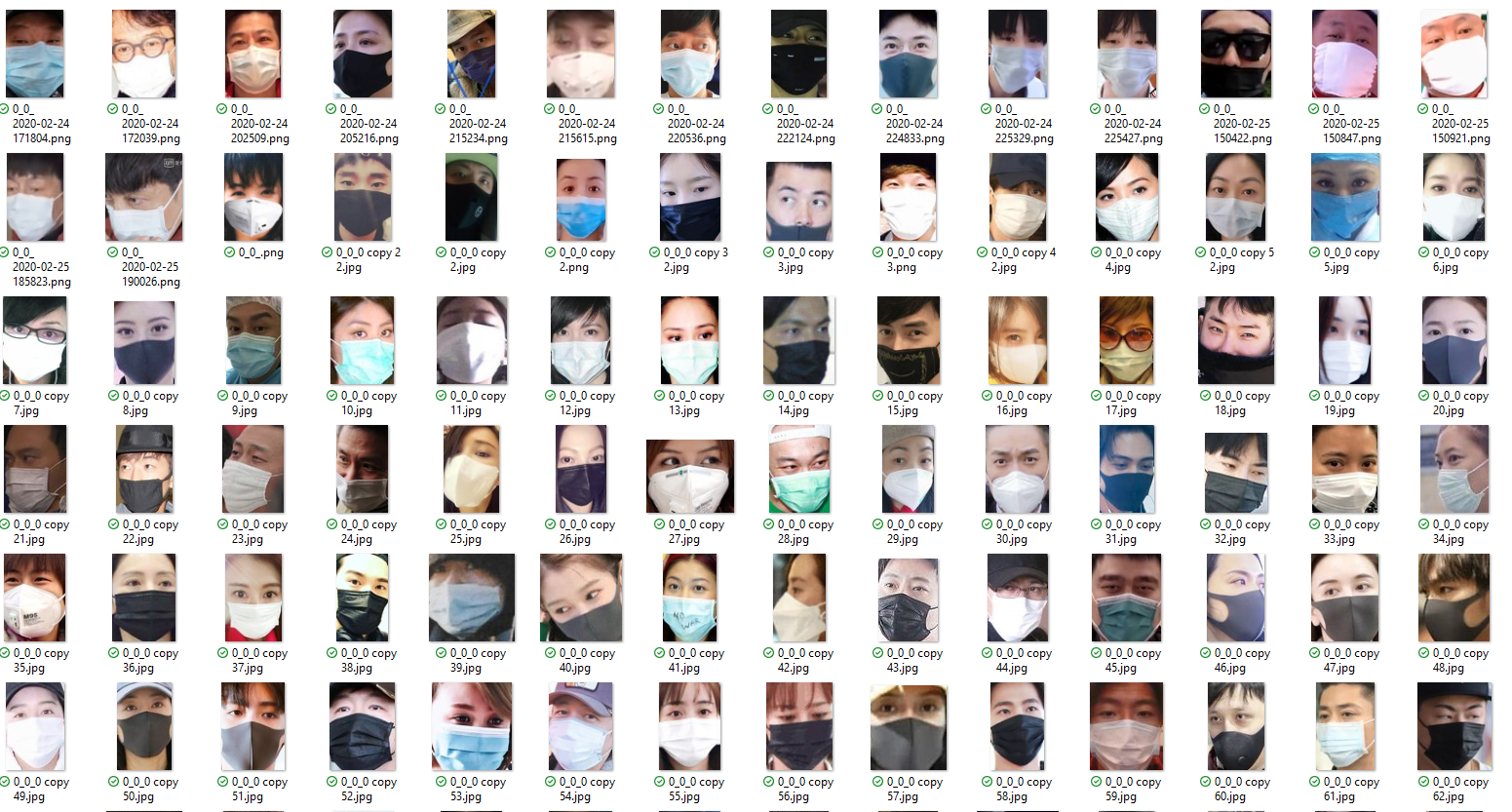
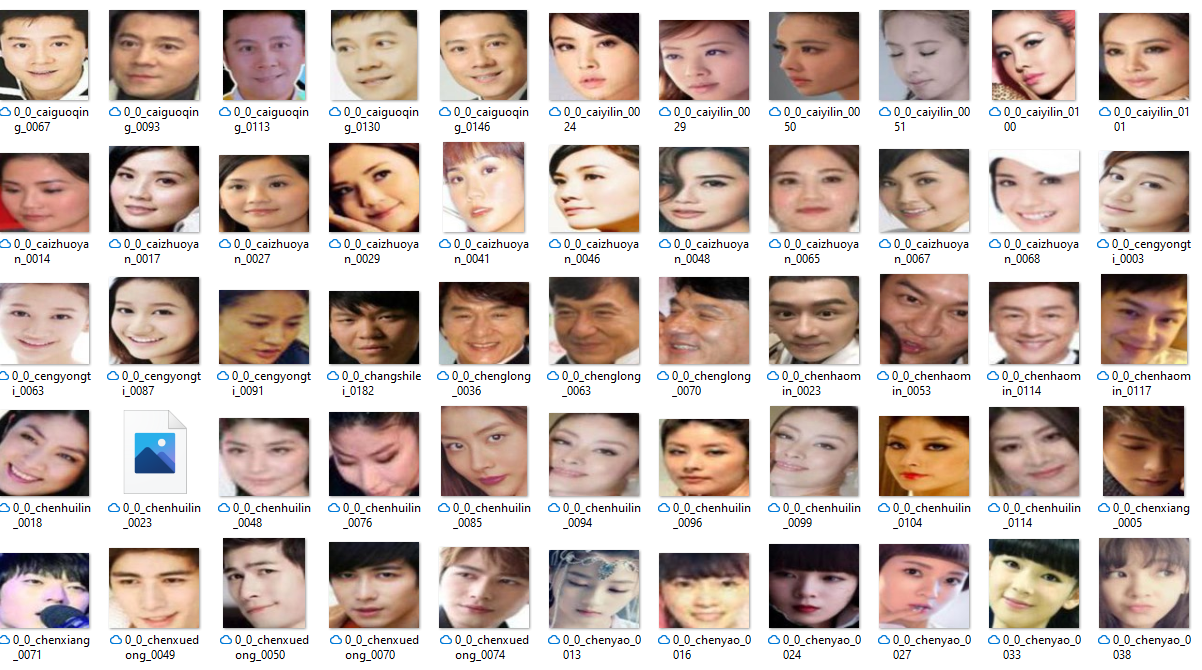
# BAB III METODE PENELITIAN

## 3.1 Deskripsi Data

Data yang digunakan dalam penelitian ini diperolah dari pengumpulan data citra orang yang menggunakan masker dan tidak menggunakan masker, yang merupakan data citra digunakan untuk melatih *machine learning.* Data ini memiliki jumlah 3.833 data citra yang terdiri dari 1.918 tanpa masker dan 1.915 menggunakan masker. Gambar 3.1 merupakan isi dari dataset yang digunakan.



Gambar 3.1 Data penggunaan masker

****

Gambar 3.2 Data penggunaan tanpa masker

## 3.2 Kebutuhan Perangkat *Hardware* dan *Software*

1. Spesifikasi Hardware
   1. Processor AMD Ryzen 5 3550H 4 Core 8 Threads
   2. RAM 16 GB DDR4 Dual-Channel 2400 Mhz
   3. Nvidia GTX 1050 3GB
2. Spesifikasi Software
   1. Sistem Operasi Windows 10 Home 64—bit
   2. Google Collab dan Visual Studio Code

## 3.3 Perancangan Sistem

A screenshot of a computer

Description automatically generated with medium confidence

Gambar 3.3 Diagram Perancangan Sistem

Berdasarkan Gambar 3.3 diagram perancangan sistem dibagi menjadi 5 tahap yaitu *Preprocessing, Data Augmentation,* Latih Model CNN, Deteksi Masker dan Hitung Akurasi. Berikut penjelasan masing-masing tahapan tersebut.

### 3.3.1 *Data Augmentation*

proses data augmentasi untuk membuat variasi baru dalam dataset. Terdapat beberapa parameter variasi data augmentasi yang meliputi rotasi, pergeseran, *cropping, zooming,* dan beberapa variasi lainnya yang bertujuan membantu meningkatkan performa model pada *deep learning*.

### 3.3.2 Preprocessing

Pada proses *Preprocessing* ini bertujuan untuk mempersiapkan data sebelum dilakukan *Data Augmentation*. Tahap ini bertujuan untuk melakukan normalisasi, transformasi, dan pengubahan ukuran citra sebagai untuk gambar agar sesuai dengan syarat yang diperlukan oleh arsitektur VGG16Net sehingga lebih mudah untuk diproses oleh model *deep learning*. Berikut adalah tahapan preprocessing yang akan dilakukan.

#### **3.3.2.1 Rezise**

*Rezise* merupakan proses untuk mengubah ukuran citra menjadi dimensensi tertentu yang diperlukan oleh arsitektur jaringan VGG16Net. Pada penelitian ini ukuran citra akan di ubah ukurannya menjadi 224 x 224 piksel. Hal ini dilakukan dengan tujuan terdapat beberapa data citra yang ukuran pikselnya berbeda-beda sehingga kemungkinan bisa menimbulkan hasil akurasi berkurang atau rendah.

#### **3.3.2.2 Konversi BGR**

Setelah melakukan proses *rezise*, langkah berikutnya adalah melakukan proses konversi yang dimana arsitektur VGG16Net menggunakan saluran warna dalam urutan BGR (*Blue, Green, Red*) sebagai *input*. Jadi, jika awal citra dalam format RGB, maka perlu melakukan konversi saluran warna dari RGB ke BGR.

#### **3.3.2.3 Mean Subtraction**

Setelah dilakukan konversi maka nilai dari BGR akan dihitung rata-rata piksel dari setiap saluran BGR tersebut dari kumpulan data pelatihan *VGG16Net*. Setelah dihitung, nilai rata-rata tersebut akan dikurangkan dari setiap piksel citra. Hal ini bertujuan untuk membantu dalam memusatkan data dan menghilangkan komponen cahaya tidak relevan. Dalam tahap ini, citra menjadi *zero-centered*

#### **3.3.2.4 Normalisasi**

Setelah memasuki tahap *Mean Subtraction*. Tahap normalisasi ini dilakukan untuk memperoleh rentang piksel yang seragam dan mempermudah proses pelatihan untuk *VGG16Net*. Normalisasi dilakukan dengan membagi setiap piksel dengan standar deviasi dari setiap saluran warna. Hal ini bertujuan menghasilkan piksel dengan rata-rata nol dan standar deviasi 1. Hal ini untuk mengontrol variasi piksel dan mempermudah konvergen selama pelatihan

Dalam tahap *Preprocessing*, penelitian ini menggunakan *framework* yang sudah di sediakan oleh *framework* Keras untuk melakukan proses *preprocessing* dalam arsitektur VGG16Net. Salah satu fungsi yang digunakan yaitu “*preprocess\_input()”* yang tersedia dalam modul “*keras.application.vgg16*”

### 3.3.4 Latih Model CNN

Dalam tahap ini akan dilakukan pelatihan dengan *data training* yang telah diaugmentasi. Pada penelitian ini dilakukan dengan menggunakan metode *convolutional neural network* (CNN).

### 3.3.5 Evaluasi Model

Setelah tahap latih model CNN, akan dilakukan evaluasi performa dalam model pada data testing. Dalam hal ini model akan diukur menggunakan *confusion matrix* yang meliputi *akurasi, presisi, recall dam F1-score.*

### 3.3.5 Model CNN

Setelah model dinyatakan memenuhi kriteria performa *confusion matrix,* model dapat disimpan untuk digunakan pada tahap untuk mendeteksi masker pada citra baru.

## 3.4 Desain Antarmuka

Icon

Description automatically generated

Gambar 3.4 Desain Antar Muka

Desain antar muka pada Gambar 3.4 digunakan untuk melakukan pendeteksi uji data citra menggunakan masker dan tidak menggunakan masker. Kemudian pada panel dalam *frame window* langsung menampilkan output dari *webcam* yang menerima input berupa video dan foto secara *realtime* langsung menampilkan tingkat akurasi pada pendeteksian tersebut.

## 3.5 Skenario Pengujian

Dalam penelitian ini dari jumlah dataset akan dibagi menjadi dua percobaan dari dataset awal berjumlah 3.833 (1.918 tanpa masker dan 1.915 menggunakan masker) dan 800 ( 400 tanpa masker dan 400 menggunakan masker)

### 3.5.1 Skenario K-Fold Cross Validation

Skenario pengujian ini dilakukan seleksi k-fold, dalam pengujian ini akan dilakukan percobaan menggunakan tiga jenis k-fold yaitu 3-fold, 5-fold, dan 10-fold, dari 3 percobaan tersebut akan dicari hasil akurasi yang optimal.

### 3.5.2 Skenario Fungsi Pelatihan CNN

Dalam Skenario ini fungsi pelatihan pada *Convolutional Neural Network* akan menggunakan percobaan 3 jenis fungsi aktivasi yaitu *Adaptive Moment Estimation*, *Gradient Descent*, dan *Root Mean* *Square Propagation* untuk membuat sebuah model CNN yang dibuat panjang iterasi atau epochs sebesar 30, 50, dan 100 iterasi.

### 3.5.3 Skenario Uji Data Tunggal

Dalam Skenario uji data tunggal untuk menguji seberapa baik model yang sudah dilatih untuk mengetahui seberapa akurat model tersebut mendeteksi data citra yang baru secara realtime dengan variasi jenis masker seperti masker beda, masker kain, dan N95.