

BAB III METODE PENELITIAN

3.1 Objek Penelitian

Objek penelitian yang digunakan dalam penelitian ini adalah gambar dua dimensi wajah manusia. Kemudian dibagi menjadi tiga kelas yaitu menggunakan masker, masker tidak sesuai atau salah dan tidak menggunakan masker yang didapatkan dari website Kaggle berjumlah 8.982 gambar. Pada Gambar 3.1 adalah contoh dari gambar yang dijadikan objek pada penelitian ini.



Gambar 3. 1 Objek Penelitian

3.2 Lokasi dan Waktu Penelitian

Lokasi penelitian dilakukan di Laboratorium Riset Universitas Buana Perjuangan Karawang sejak Maret 2021 sampai dengan April 2022. Perincian penelitian ditunjukkan pada Tabel 3.1.

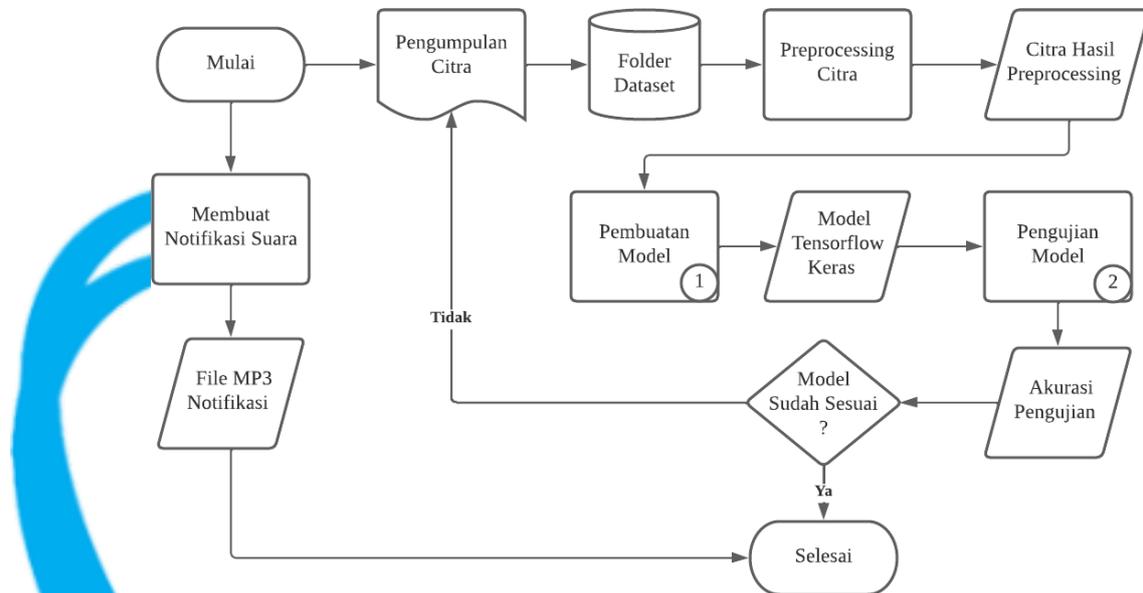
Tabel 3. 1 Kegiatan dan Waktu Penelitian

No	Kegiatan	2021					2022	
		Maret	April	Mei	Juni	Juli	Maret	April
1	Studi Literatur	■						
2	Pengumpulan Dataset		■					
3	Preprocessing dan Pelatihan Model			■				
4	Testing Deteksi Masker dan Fitur Suara				■			
5	Pengujian Akurasi					■		
6	Pembuatan Laporan						■	■

3.3 Prosedur Penelitian

Prosedur dalam penelitian ini dibagi menjadi tiga tahapan yaitu mempersiapkan citra lalu membuat model deteksi masker wajah dan pengujian

model yang telah dilatih. Persiapan model lebih berfokus untuk melakukan akuisisi citra yang telah dikumpulkan sebelumnya dalam bentuk *dataset* yang nantinya akan dilakukan *preprocessing* citra.



Gambar 3. 2 Diagram Alir Prosedur Penelitian

Berdasarkan diagram alir yang ditunjukkan pada Gambar 3.2, penelitian terdiri dari dua tahapan yaitu membuat file mp3 berisi notifikasi suara dan membuat model deteksi. Model deteksi berfungsi sebagai klasifikasi terhadap wajah yang terdeteksi apakah menggunakan masker wajah, masker tidak tepat atau tidak menggunakan masker. Selanjutnya menguji model yang sebelumnya sudah dibuat dengan ditambahkan fitur notifikasi suara.

3.2.1 Notifikasi Suara

Tahap awal yaitu menciptakan sebuah file yang berformat MPEG-1 Audio Layer 3 (MP3) yang berisi notifikasi suara. Dengan menggunakan metode *text-to-speech* atau mengubah sebuah teks menjadi suara. Proses ini dibantu dengan menggunakan *library* Google *Text-to-Speech* (gTTS) yang nanti outputnya berupa file MP3 yang digunakan ketika melakukan pengujian deteksi masker.

3.2.2 Pengumpulan Data Citra

Tahap ini bukan lanjutan dari tahapan notifikasi suara, tahapan ini merupakan tahap pengolahan citra yaitu mempersiapkan data yang diperlukan dengan mengumpulkan gambar wajah manusia yang menggunakan masker, masker salah dan tanpa masker. *Dataset* pelatihan menggunakan *dataset* yang didapatkan dari

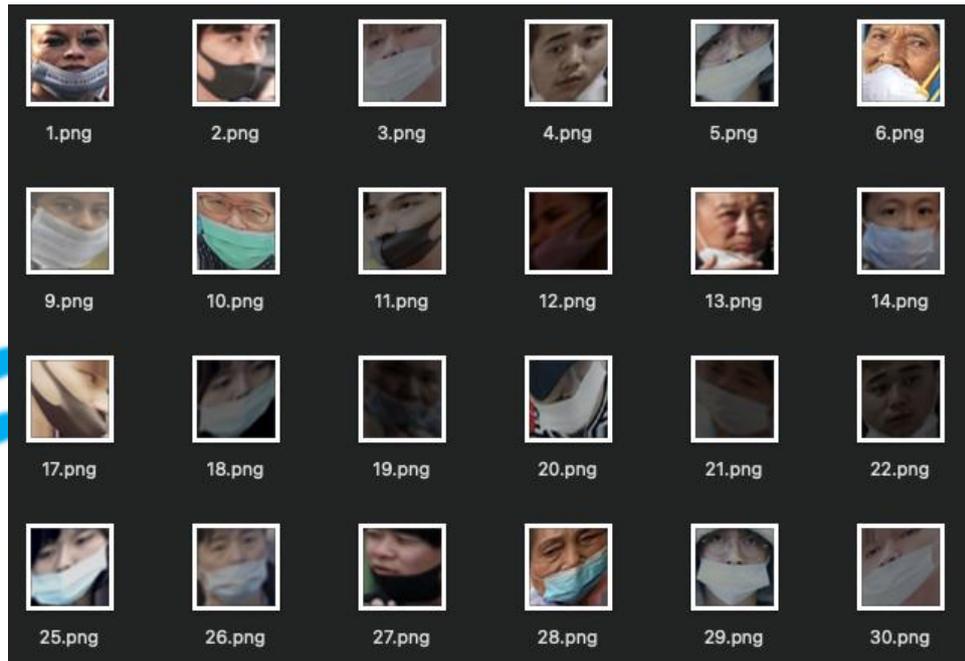
Kaggle. Dengan jumlah 2994 gambar memakai masker, 2994 gambar memakai masker tidak tepat yaitu masker yang tidak menutupi hidung dengan baik dan 2994 gambar tanpa menggunakan masker. Contoh gambar *dataset* dapat dilihat pada Gambar 3.3, Gambar 3.4 dan Gambar 3.5. Setelah itu, kumpulan data citra akan ditempatkan di folder struktur.



Gambar 3. 3 Dataset Wajah Menggunakan Masker



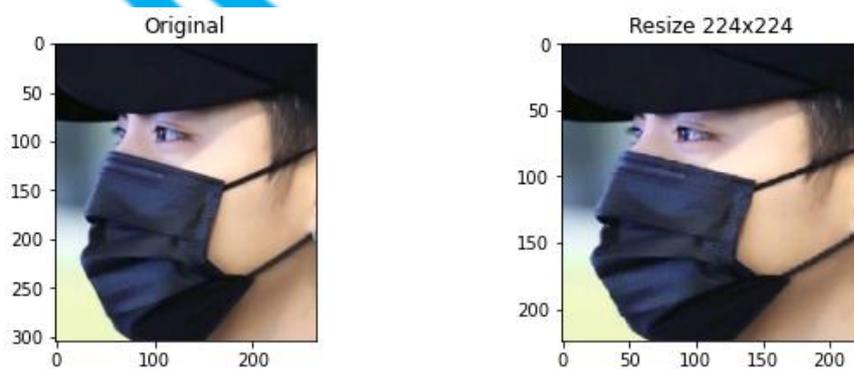
Gambar 3. 4 Dataset Wajah Tidak Menggunakan Masker



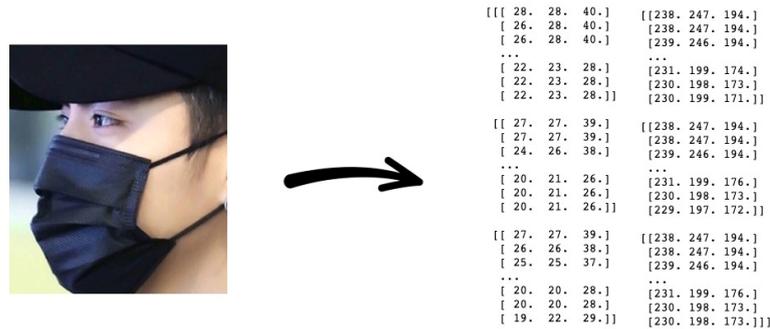
Gambar 3. 5 Dataset Wajah Menggunakan Masker Salah

3.2.1 Preprocessing Citra

Kemudian akan dilakukan tahap *preprocessing* pada citra dengan mengubah ukurannya menjadi 224×224 piksel seperti pada Gambar 3.6. Dilanjutkan dengan proses konversi ke dalam bentuk *array* menggunakan bahasa pemrograman python dan *library preprocessing* dari tensorflow keras seperti pada Gambar 3.7. Hal ini bertujuan agar saat penginputan dan proses klasifikasi pada menjadi seragam dan mengatasi kehilangan tingkat akurasi pada proses pelatihan.



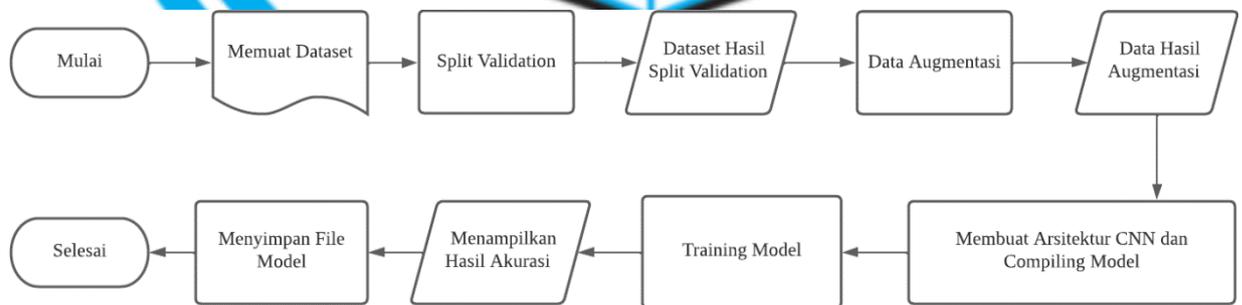
Gambar 3. 6 Mengubah Ukuran Citra



Gambar 3. 7 Konversi ke Array

3.2.1 Implementasi dan Pelatihan Model

Sebagai bagian dari *supervised learning* maka model deteksi memerlukan kumpulan data (*dataset*) dari citra berlabel sebagai data pelatihan untuk mendeteksi objek atau kelas pada citra masukan. Berdasarkan diagram alur pada Gambar 3.8 proses pengembangan model deteksi objek dibagi menjadi tiga tahap yaitu memuat dataset citra, pelatihan model deteksi objek dan menampilkan hasil akurasi pelatihan. *Convolutional neural network* digunakan sebagai algoritma pendeteksian objek dalam penelitian ini.



Gambar 3. 8 Diagram Alur Pelatihan Model

1) **Memuat *Dataset***

Pada tahap ini adalah memuat *dataset* yang sebelumnya telah dilakukan tahap *preprocessing* yaitu mengubah ukuran citra dan mengkonversinya ke bentuk *array*.

2) ***Split Validation***

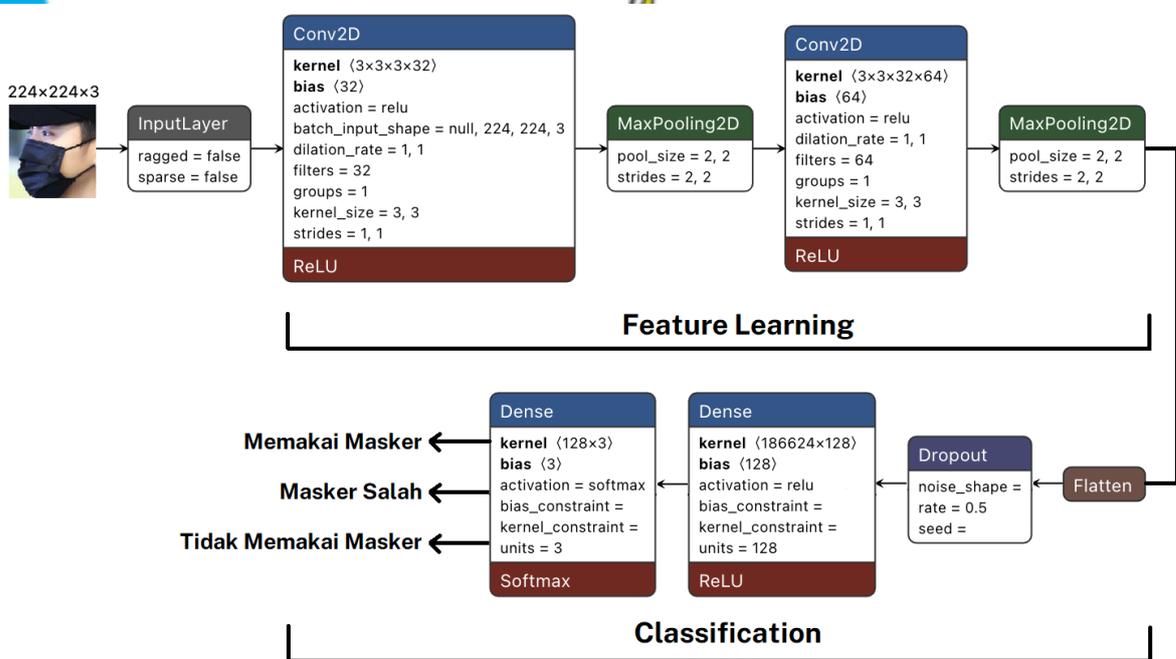
Kemudian dataset citra akan dilakukan *split validation* dengan komposisi 80% data untuk pelatihan dan 20% untuk pengujian dengan

menggunakan library dari skicit-learn. Hal ini bertujuan agar model yang dihasilkan memberikan hasil maksimal dan akurat.

1) **Data Augmentasi**

Proses augmentasi dilakukan ini untuk memanipulasi data citra sehingga terdapat keberagaman dan memperbanyak citra pada *dataset*. Proses augmentasi dilakukan menggunakan library *ImageDataGenerator* dari Tensorflow Keras. Augmentasi yang dilakukan seperti berupa *zoom*, merotasi, menggeser, kemiringan (*shear*), memotong dan membalikkan citra secara acak dari gambar yang ada di dalam *dataset*.

2) **Membuat Arsitektur CNN dan *Compiling* Model**



Gambar 3. 9 Rancangan dari Arsitektur CNN

Berdasarkan Gambar 3.9 bahwa terdapat dua tahapan yaitu *Feature Learning* dan *Classification*. Pada tahap *Feature Learning* adalah proses untuk menentukan representasi dari sebuah gambar menjadi *features* yang berupa nilai-nilai yang merepresentasikan gambar tersebut dan nantinya akan digunakan pada tahap klasifikasi. Tahap *Classification* adalah tahap dimana hasil dari proses *feature learning* akan digunakan untuk proses klasifikasi berdasarkan subclass yang suda ditentukan.

Berikut adalah proses-proses yang ada didalam rancangan arsitektur CNN :

- a) Pada konvolusi pertama jumlah filter yang digunakan yaitu sebanyak 32 dengan kernel matrix 3×3 . Hasil dari tahap ini adalah matrix dengan ukuran $224 \times 224 \times 32$.
- b) Lalu pada setiap proses konvolusi akan digunakan aktivasi *ReLU* yaitu menghapus nilai negatif yang terdapat pada data.
- c) Selanjutnya dilakukan proses *MaxPooling* dengan ukuran 2×2 dengan *stride* atau pergeseran sebanyak dua langkah.
- d) Kemudian pada tahap konvolusi yang kedua dengan menggunakan 64 filter dengan kernel matrix 3×3 .
- e) Dilakukan kembali proses *MaxPooling* yang kedua dengan ukuran 2×2 dan *stride* dua langkah.
- f) Tahap terakhir pada *feature learning* yaitu layer *flatten* dengan merubah *output* dari proses konvolusi yang berupa *matrix* menjadi sebuah bentuk *vector* dengan ukuran 1×128 .
- g) Selanjutnya akan diteruskan pada tahap klasifikasi dengan menggunakan *Multi Layer Perceptron (Neural Network)*. Kelas dari citra hasil *flatten* kemudian diklasifikasikan berdasarkan nilai dari *neuron* pada *hidden layer* dengan menggunakan fungsi aktivasi *softmax*.

Setelah arsitektur CNN telah dibangun tahap selanjutnya file model akan di-*compile* menggunakan *Adam optimizer*.

1) **Pelatihan Model**

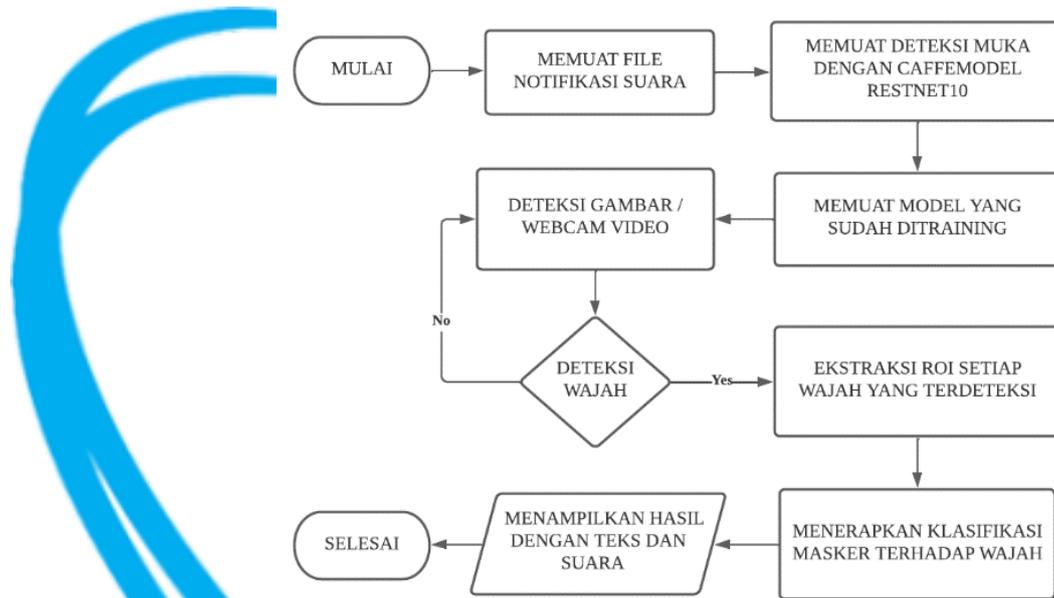
Sesudah arsitektur CNN dibuat dan di-*compile*, selanjutnya pada tahap ini data hasil prosesing akan diinputkan ke dalam arsitektur tersebut untuk memulai proses pelatihan. Proses ini bertujuan agar arsitektur CNN dapat mengenali datasetnya dan membentuk model berdasarkan pelatihan tersebut. Lalu menentukan beberapa parameter seperti *epoch* dan *learning rate*.

Epoch adalah banyaknya perulangan yang akan dilakukan pada proses pelatihan. Sedangkan *learning rate* adalah parameter yang

digunakan untuk mengupdate bobot pada proses *training* di setiap kali iterasi (*epoch*).

Saat pelatihan juga mengkombinasikan metode data augmentasi pada proses sebelumnya. Bertujuan untuk memperbanyak dan memperagam *dataset* dan meningkatkan tingkat akurasi dari model.

3.2.1 Pengujian Model



Gambar 3. 10 Diagram Alur Pengujian Model

Pada tahap pengujian melalui video menggunakan perangkat *webcam* dengan spesifikasi 720 piksel FaceTime HD kamera. kemudian seperti pada Gambar 3.10, tahap awal yaitu memuat semua komponen yang dibutuhkan, seperti memuat *file* “.mp3” notifikasi suara yang sebelumnya telah dibuat. Selanjutnya memuat *file* *caffemodel* yang berfungsi sebagai pendeteksi wajah, dan yang terakhir yaitu memuat *file* model yang sebelumnya telah dilakukan *training*. Pada tahap pengujian ini terdapat pilihan apakah mendeteksi melalui citra atau *webcam* video. Berikutnya, *caffemodel* akan melakukan pencarian wajah pada sumber jika terdapat wajah maka akan melakukan ekstraksi *Region of Interest* (ROI) pada setiap wajah yang terdeteksi. Kemudian, menerapkan klasifikasi menggunakan model yang sudah dilatih. Tujuannya agar mengetahui apakah menggunakan masker atau tidak menggunakan masker dengan *output* berupa *bounding box* di area wajah disertai keterangan dan suara notifikasi.