

BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Objek pada penelitian ini menggunakan citra dua dimensi wajah manusia. Selanjutnya, dibagi menjadi dua kelas yaitu menggunakan masker atau tidak menggunakan masker. Data tersebut didapat dari web Kaggle dengan jumlah 3.857 citra. Pada (Kumar 2021).



Gambar 3. 1 Objek Penelitian

3.2. Lokasi dan Waktu Penelitian

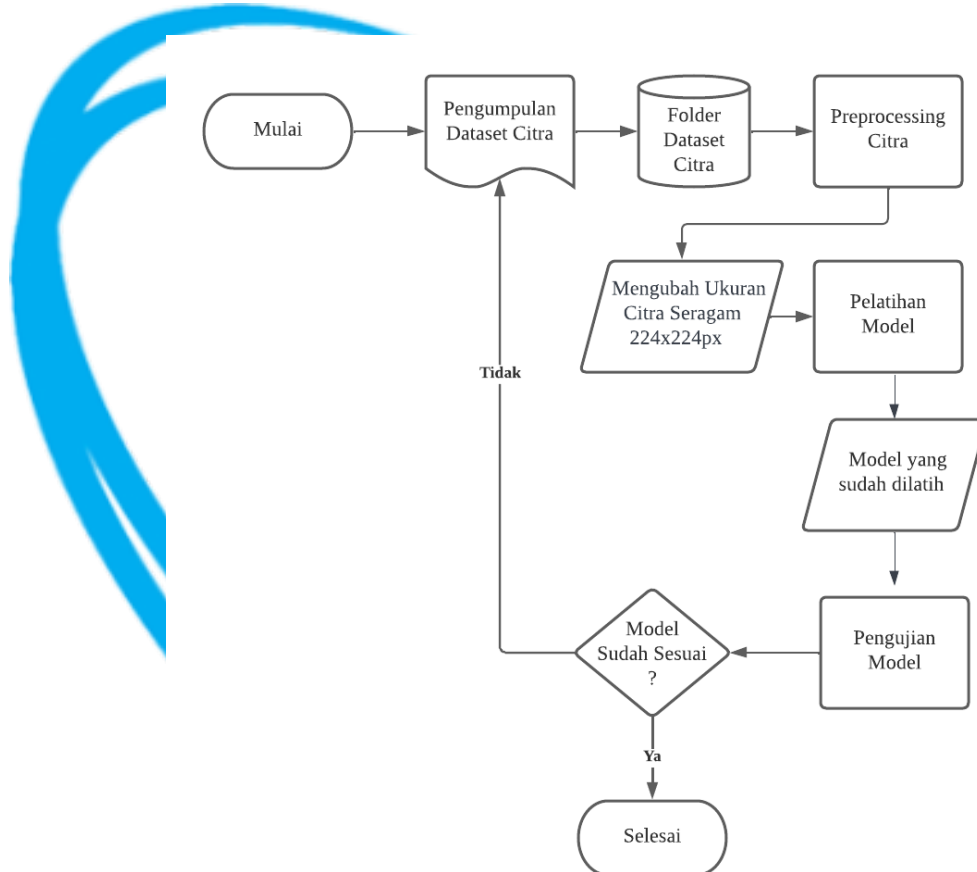
Lokasi pada penelitian ini yaitu menggunakan Laboratorium Riset Universitas Buana Perjuangan Karawang. Dimulai pada tanggal 31 Juni 2021 sampai dengan 8 Oktober 2021. Perincian ini berada pada Tabel 3.1.

Tabel 3. 1 Kegiatan dan Waktu Penelitian

No	Kegiatan	2021					2022	
		Maret	April	Mei	Juni	Juli	Maret	April
1	Studi Literatur							
2	Pengumpulan Data set							
3	<i>Preprocessing</i> dan Pelatihan Model							
4	<i>Testing</i> Deteksi Masker							
5	Pengujian Akurasi							
6	Pembuatan Laporan							

3.3. Prosedur Penelitian

Tiga tahapan pada prosedur penelitian ini dilakukan, tahapan tersebut meliputi persiapan gambar lalu pelatihan model dan pengujian model yang sudah dilatih. Citra atau gambar yang telah dikumpulkan sebelumnya lebih di fokuskan pada persiapan model. Citra tersebut dikumpulkan dalam bentuk *Dataset* yang nantinya proses selanjutnya dilakukan *preprocessing* citra.

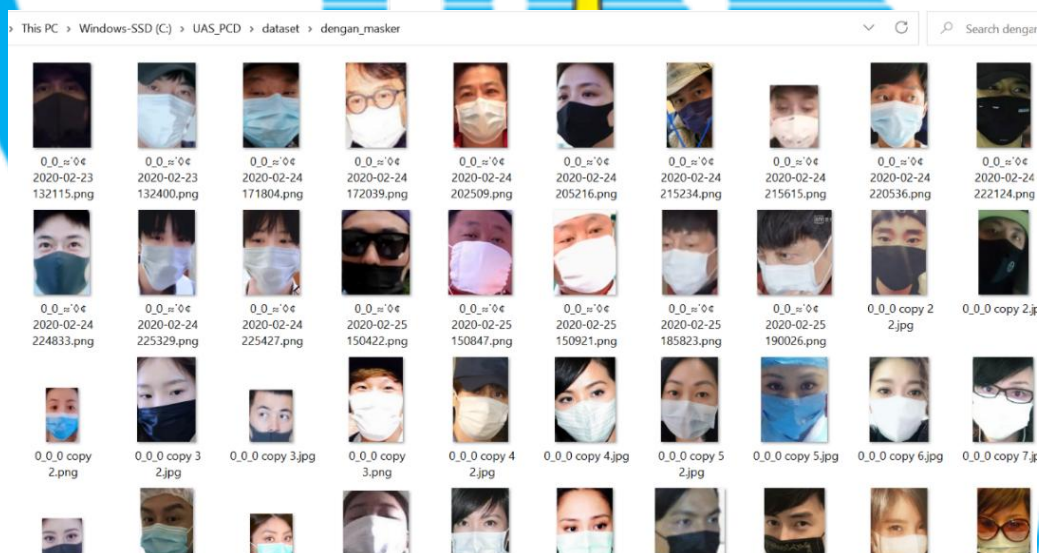


Gambar 3. 2 Diagram Alir Prosedur Penelitian

Dapat dilihat diagram alir pada Gambar 3.2, setelah dilakukan *preprocessing* data lalu dilatih dan mendapatkan sebuah model deteksi yang diperlukan pada penelitian ini. Fungsi dari model deteksi tersebut menandakan klasifikasi terhadap wajah yang terdeteksi. Untuk mengetahui masker digunakan pada wajah atau tidak menggunakan masker.

1.3.1. Pengumpulan data dan *Preprocessing* Gambar

Persiapan data diperlukan pada tahap ini, dengan mengumpulkan gambar wajah orang yang menggunakan masker dan tidak menggunakan masker. Bersumber dari data Kaggle *Dataset* diperoleh dan dilakukan pelatihan. Pelatihan menggunakan bahasa pemrograman *Python*, total 1924 gambar yang memakai masker dan 1933 gambar tidak menggunakan masker. Contoh dari *Dataset* berada pada Gambar 3.3 dan Gambar 3.4. Kemudian, data yang sudah terkumpul disimpan dalam folder yang sudah terstruktur. Selanjutnya dilakukan tahap *preprocessing* pada gambar dengan mengubah ukuran menjadi 224x224 piksel. *Library* yang digunakan dari tensorflow, tujuannya agar saat proses *input* dan klasifikasi menjadi seragam serta mengatasi kehilangan tingkat akurasi pada proses pelatihan.

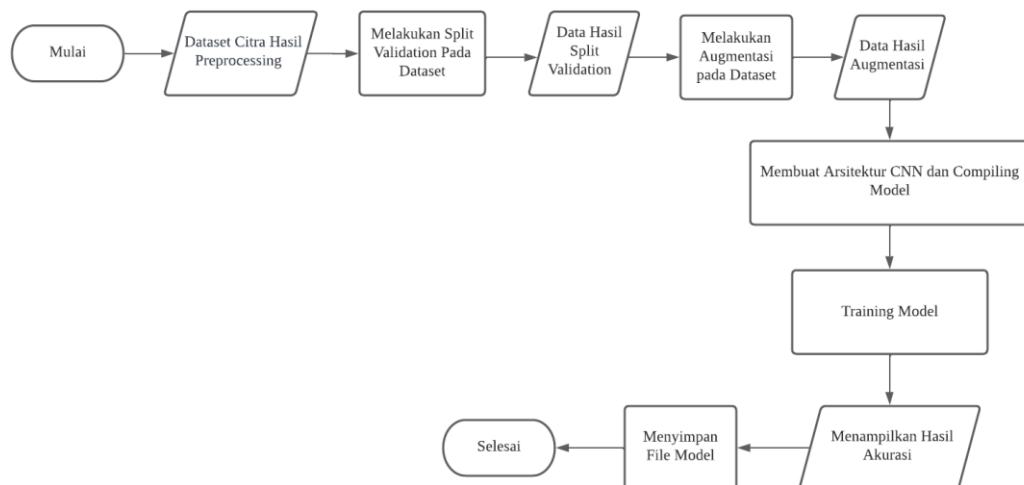


Gambar 3. 3 *Dataset* Wajah Menggunakan Masker



Gambar 3. 4 Dataset Wajah Tidak Menggunakan Masker

1.3.1. Pelatihan Model



Gambar 3. 5 Diagram Alir Pelatihan Model

Dikarenakan model deteksi masker ini merupakan bagian dari supervised learning, membuat deteksi memerlukan kumpulan data atau *Dataset* dari suatu citra yang berlabel sebagai data pelatihan untuk mendeteksi objek pada citra masukan. Ada tiga tahapan pada proses pengembangan model deteksi objek, tahapan tersebut meliputi *Dataset* citra, pelatihan model deteksi objek dan menampilkan hasil akurasi. *Convolutional neural network* digunakan, fungsinya sebagai algoritma pendeteksian objek. Berdasarkan pada Gambar 3.5 memiliki penjelasan sebagai berikut:

- **Data set *Preprocessing***

Pada proses ini yaitu memuat data set yang sebelumnya telah dilakukan tahap *preprocessing* (mengubah ukuran citra dan mengkonversi ke bentuk *array*).

- ***Split Validasi***

Dataset pada gambar dilakukan *split* validasi menggunakan *library* dari *skicit-learn*. Tujuannya agar model yang dihasilkan memberikan hasil maksimal dan akurat.

- **Data Augmentasi**

Augmentasi dilakukan agar citra dapat dimanipulasi sehingga memperbanyak citra pada data set. Proses augmentasi ini dilakukan menggunakan *library ImageDataGenerator* dari Tensorflow Keras. Augmentasi yang dilakukan yaitu *zoom*, rotasi, menggeser, memotong dan membalikan citra secara acak dari citra yang ada pada data set.

- **Membuat arsitektur CNN dan *Compiling Model***

Dalam membuat arsitektur CNN terdapat dua tahapan yaitu *Feature Learning* dan *Classification*. Tahapan tersebut tentunya memiliki langkah-langkah yang harus dilakukan, pada *Feature Learning* terdapat *input* yaitu data set hasil *preprocessing*. Kemudian, dilanjutkan dengan *convolution* untuk melakukan proses *stride* yang didalamnya memiliki operasi matematika dengan hasil dimasukan pada peta fitur atau *feature map*. Selanjutnya fungsi aktivasi dilakukan agar menghapus nilai negatif yang terdapat pada data, lapisan penyatuan atau *pooling layer* digunakan untuk mengurangi komputasi pada proses pelatihan.

Tahap *Classification* berfungsi untuk hasil dari proses *feature learning* bergantung pada *subclass* yang ditentukan, sistem kategorisasi akan digunakan. Langkah pada tahap ini yaitu *flatten* yang dilakukan untuk mengubah data multi dimensi *array* pada *feature map*, menjadi vector agar dapat digunakan sebagai *input* pada dari *fully connected layer*. Kemudian, lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron selanjutnya. Fungsi

aktivasi yang digunakan tidak RELU tetapi menjadi softmax (mencari nilai maksimum dalam suatu klasifikasi).

- **Training Model**

Tahap ini data pada hasil *processing* akan di *inputkan* ke dalam arsitektur tersebut untuk memulai proses pelatihan. Tujuan dari proses ini agar arsitektur CNN dapat mengenali *Dataset* untuk membentuk model berdasarkan pelatihan. Beberapa parameter ditentukan seperti *epoch* dan *learning rate*. *Epoch* tersebut berguna untuk melakukan banyaknya perulangan yang pada proses pelatihan. Adapun *learning rate* berfungsi agar parameter yang akan digunakan untuk mengupdate bobot pada proses pelatihan di setiap kali iterasi (*epoch*). Pada tahap ini juga metode data augmentasi dikombinasikan. Bertujuan untuk memperbanyak data set dan meningkatkan hasil akurasi.

1.3.1. Pengujian Model



Gambar 3. 6 Diagram Alir Pengujian Model

Pada Gambar 3.6, langkah pertama adalah menginstal semua komponen yang diperlukan seperti mengunggah *caffemodel* yang berfungsi sebagai pendeteksi wajah dan langkah terakhir adalah mengunggah fail model yang kita latih sebelumnya. Selama fase pengujian ini, dapat memilih untuk mendeteksi melalui *webcam* gambar atau video. Kemudian *caffemodel* akan mencari wajah di sumbernya, jika ada wajah, kurangi *Region of Interest* (ROI) untuk setiap wajah yang terdeteksi.