

BAB III

METODE PENELITIAN

3.1. Objek Penelitian

Pada penelitian ini objek yang diteliti ialah pembuatan sebuah Software yang dapat mendeteksi penggunaan masker sebelum memasuki ruangan, dengan penggunaan *Image Processing* yang didasari kebutuhan pada saat pandemi ini agar dapat mengingatkan penggunaan masker sebelum masuk ruangan. Dalam penelitian ini berfokus pada deteksi wajah yang menggunakan masker, tidak menggunakan masker dengan baik dan tidak menggunakan masker. Yang telah dikumpulkan kedalam dataset untuk kemudian melakukan tahapan dataset *training* dan dataset uji. Untuk mengetahui tingkat keakurasian terhadap Software yang akan dibuat.

3.2 Peralatan Penelitian

Pada penelitian ini, menggunakan beberapa alat untuk menunjang dalam melaksanakan penelitian tugas akhir tersebut antara lain :

3.2.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

A. Laptop

perangkat keras yang digunakan untuk penelitian ini adalah sebagai berikut :

- 1) Model : X450EA
- 2) Processor : AMD Quad-Core E2-3800
- 3) Sistem Operasi
- 4) Ram : 2 GB
- 5) Hardisk

B. WebCam

WebCam digunakan untuk menangkap gambar atau menampilkan input kedalam monitor laptop adalah sebagai berikut :

- 1) WebCam 720p

3.2.2 Perangkat Lunak (Software)

Sedangkan perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

a) Visual Studio

3.3 Lokasi Dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium Riset Universitas Buana Perjuangan Karawang, Kecamatan teluk jambe, Kabupaten Karawang. Terhitung sejak November 2021 hingga April 2022. Rincian pelaksanaan dimasukan kedalam tabel sebagai berikut :

Tabel 3.1 Waktu Penelitian

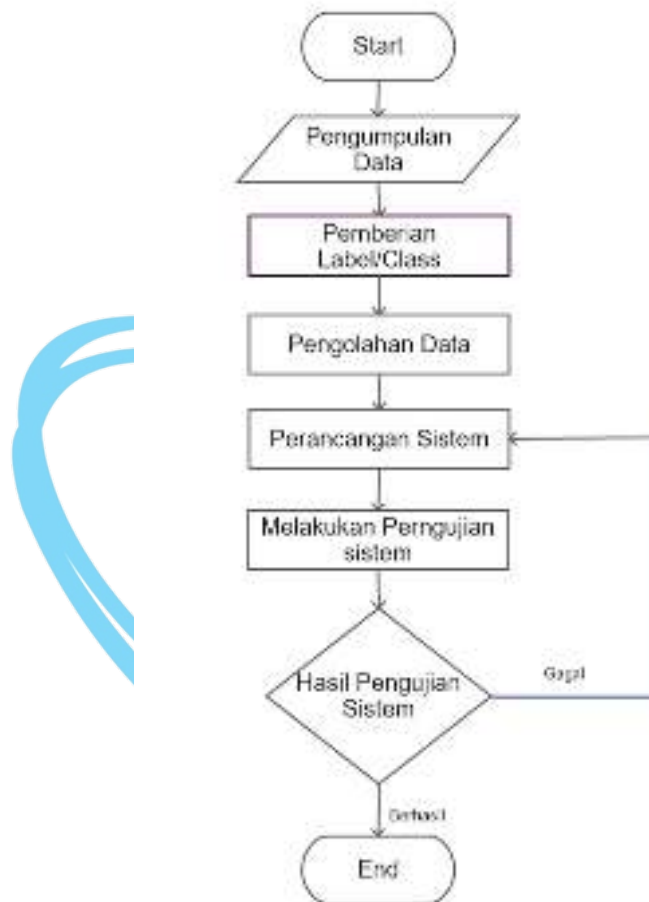
No	kegiatan	Nov 2021				Des 2021				Jan 2022				Maret 2022				April 2022			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Litelatur																				
2	Perumusan masalah dan analisis kebutuhan																				
3	Pembuatan proposal																				
4	Pembuatan Software																				
5	Pengujian dan Perbaikan																				
6	Pengumpulan TA																				

3.4 Prosedur Penelitian

Prosedur yang digunakan akan di tuangkan dalam sub-sub bab yang akan menjelaskan setiap proses, metode dan kebutuhan yang akan dilakukan dalam penelitian ini.

Pada penilitian ini akan menggunakan Algoritma *You Only Look Once* (YOLO) dengan menggunakan YOLO. Langkah pertama melakukan pengumpulan data atau citra yang diambil berdasarkan data yang telah di kumpulkan oleh pembimbing untuk kemudian melakukan penelitian. Kemudian dilanjutkan dengan pengolahan data dan melakukan pemberian label/class terhadap data yang telah

dikumpulkan. Kemudian melakukan pengolahan data agar bisa melakukan data *training* dan data uji pada sistem. Langkah selanjutnya melakukan perancangan sistem. Sistem yang akan dibuat menggunakan algoritma YOLO akan berfokus terhadap identifikasi pemakaian masker dan tingkat keakurasian dari hasil identifikasi tersebut. Berikut alur diagram pada penelitian ini sebagai berikut.



Gambar 3.1 Prosedur Penelitian

3.4.1 Pengumpulan Data

Pada tahapan ini akan dilakukan pengumpulan data untuk membantu dalam penelitian yang kemudian menguji dan melatih model deteksi penggunaan masker. Antara lain sebagai berikut :

3.4.1.1 Dataset

Pada proses pembuatan Dataset dengan melakukan pengambilan gambar untuk kemudian diolah dengan pengolahan Citra Digital. Data yang akan diambil

berupa gambar dengan dibagi kedalam tiga kelas. yaitu menggunakan masker, menggunakan masker tidak benar dan tidak menggunakan masker. Data yang diambil berdasarkan data yang telah di kumpulkan oleh pembimbing untuk kemudian melakukan penelitian.

3.4.1.2 Pemberian label/class

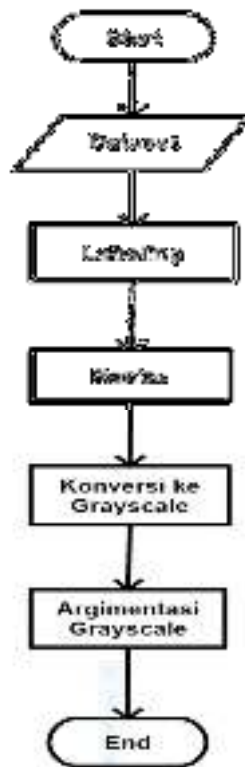
Setelah mengumpulkan data dan kemudian menjadikan sebuah database untuk kemudian diberikan pelabelan atau *class* untuk menentukan objek yang akan di deteksi berdasarkan kebutuhan dalam penelitian ini. pelabelan akan diberikan empat kelas yaitu, dengan nama label antara lain, wajah, hidung, mulut dan masker kemudian digunakan untuk menentukan deteksi objek dengan output sebagai berikut: menggunakan masker, tidak menggunakan masker dengan benar dan tidak menggunakan masker.

3.4.2 Pengolahan Data

Dengan melakukan pengolahan data yang akan dilakukan untuk membantu dalam proses pembuatan sistem ini akan dituangkan kedalam alur diagram sebagai berikut.

3.4.2.1 Preprocessing

Pada tahapan ini Dataset yang telah dikumpulkan kemudian akan melewati beberapa tahapan *Preprocessing* dimensi citra pada Dataset. Kemudian akan dirubah menjadi 422x422 pixel. Karena memiliki berbagai dimensi yang beragam. Hal ini dibutuhkan untuk menyamakan sebuah ukuran pada citra agar mempermudah sistem dalam memproses citra tersebut. Berikut alur diagram proses *preprocessing*.



Gambar 3.2 Alur Diagram Preprocessing

3.4.2.2 Ekstrasi

Melakukan ekstrasi terhadap citra yang akan dibuat dengan melakukan ekstrasi fitur wajah untuk melakukan deteksi wajah yang kemudian akan mendeteksi pada area wajah saat menggunakan masker, menggunakan masker tidak benar, dan tidak menggunakan masker. Dengan menggunakan *Algoritma YOLO* yang dapat membantu untuk mendeteksi batasan-batasan pada fitur deteksi masker ini secara lebih akurat. Dengan merubah citra kedalam bentuk grayscale menggunakan HSV dalam merubah citra objek menjadi grayscale

3.4.2.3 Klasifikasi

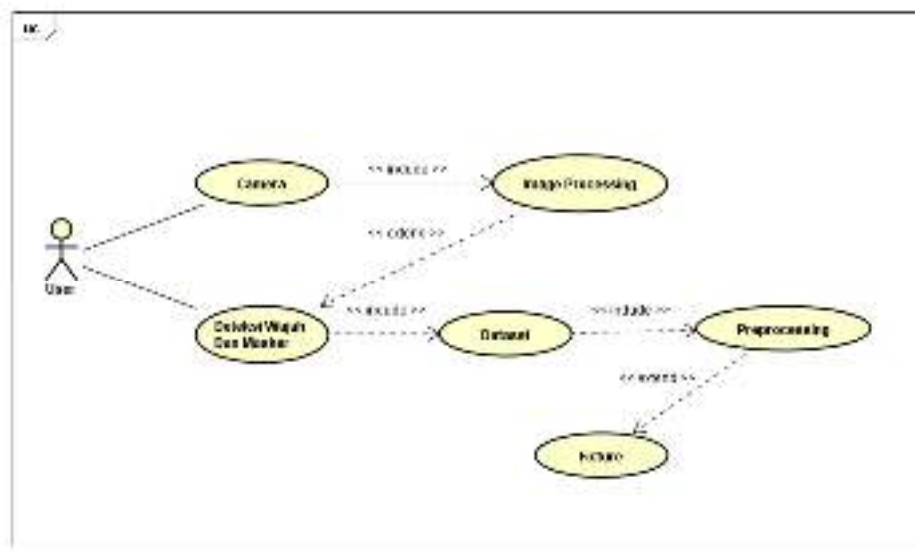
Melakukan klasifikasi pada citra sesuai dengan tiga kelas yang di beri label, yaitu menggunakan masker, tidak menggunakan masker dengan benar, dan tidak menggunakan masker. Dan melakukan klasifikasi terhadap penggunaan masker, antara masker medis dan non medis. Dan kemudian akan melakukan deteksi pada fitur deteksi masker.

3.4.3 Perancangan Sistem

Pada tahapan ini melakukan perancangan sistem yang akan di lakukan untuk pengembangan sistem yang akan dibuat berupa Use Case Diagram.

a) Use Case Diagram

Use Case Diagram ini digunakan untuk mengetahui apa saja yang dapat digunakan oleh pengguna atau user terhadap rancangan fungsional yang berada pada Software yang akan di rancang;



Gambar 3.2 Diagram Use Case

b). Indikator Actor

Actor merupakan abstraksi antara user atau orang dengan sistem lain yang dapat menggunakan fungsi dari sistem. Berikut adalah actor yang akan berperan dalam menjalankan sebuah sistem.

Tabel 3.2 Actor Use Case

No	Actor	Deskripsi
A-00	User	Merupakan Actor dari Software yang dibangun atau pengguna aplikasi

Penggambaran pada Use Case Diagram sistem ini yaitu Use Case menunjukan interaksi antara User dan sistem. User yaitu yang menjalankan sistem pada aplikasi.

c). Indentifikasi Use Case

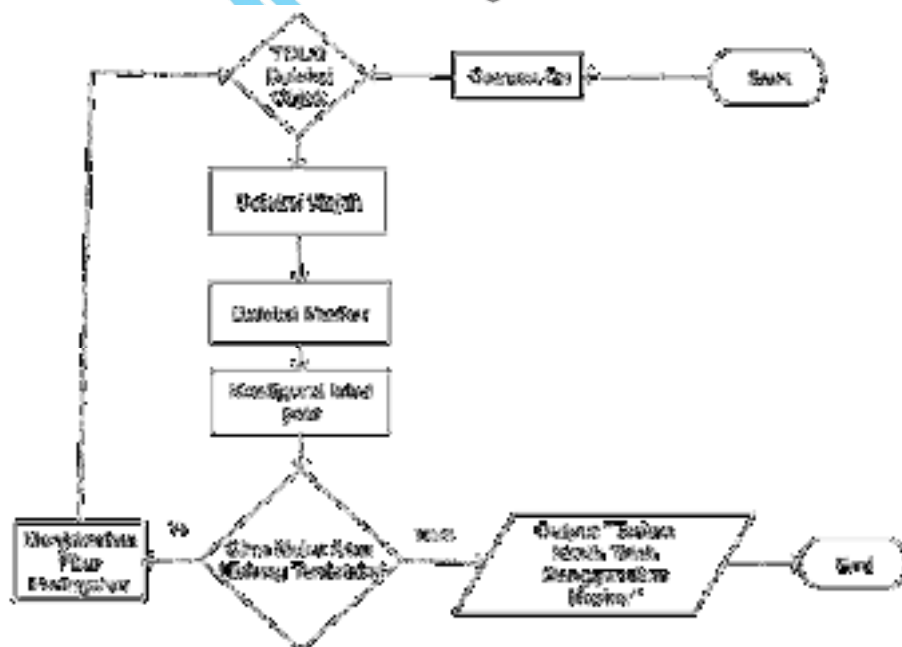
Berikut merupakan tabel indentifikasi Use Case yang terdapat pada Software.

Use Case Sistem User		
No	Use Case	Deskripsi
UC-00	Camera	Fungsional mengambil gambar menggunakan camera WebCam
UC-01	Deteksi Wajah Dan Masker	Fungsional memungkinkan pengguna atau user mendeteksi penggunaan masker

Tabel 3.3 Use Case Software

3.4.4 Implementasi

Dalam tahapan ini merupakan tahapan menetralisasikan dari tahapan sebelumnya kedalam bahasa pemrograman yang akan dimasukan kedalam sistem pada perangkat software yang akan dibuat, proses ini dimasukan kedalam flowchart kemudian melakukan pengcodingan. Berikut tahapan implementasi pada sistem ini sebagai berikut :



Gambar 3.4 FlowChart Sistem Dengan YOLO

Flowchart di atas merupakan alur dari proses sistem yang akan dibuat dimana pada tahapan pertama menyalanya kamera kemudian melakukan konfigurasi deteksi dengan menggunakan YOLO. Kemudian mendeteksi wajah dan penggunaan masker. Tahapan selanjutnya melakukan konfigurasi label pada YOLO, bila citra mulut atau hidung terdeteksi maka akan muncul berupa peringatan untuk menggunakan masker, apabila tidak terdeteksi maka akan muncul output berupa “Terima Kasih Telah Menggunakan Masker” proses deteksi selesai. Tahapan YOLO pada proses pendeteksian masker adalah sebagai berikut:

a) Membagi Gambar Menjadi Grid

Pada objek yang akan dideteksi algoritma YOLO akan membagi citra kedalam sebuah grid yang berukuran $s \times s$. Grid tersebut akan bertanggung jawab untuk memprediksi objek tersebut. Pada setiap grid akan di prediksi juga pada setiap *bounding box*nya serta menghitung nilai *confidence*nya. Untuk menunjukan seberapa yakin dan seberapa akurat dalam mendeteksi sebuah objek tersebut.

b) Bounding Box

Memprediksi *bounding box* yang telah diberi nama *class* pada setiap objek citra. Yang dimana tiap *bounding box* memiliki 5 nilai informasi yaitu, x , y , w , h , dan c . nilai x dan y merupakan kordinat titik tengah pada *bounding box* yang terprediksi, kemudian nilai w dan h merupakan rasio untuk ukuran lebar dan tinggi yang relative terhadap grid, sengangkan untuk nilai c nilai *confidence* untuk nilai pada *bounding box* tersebut.

c) Nilai pada prediksi

Pada tahapan ini YOLO akan memprediksi sebuah nilai *class probability*, jika terdapat objek yang terprediksi didalamnya. Pada saat pengujian nilai *probability* akan memngkalikan nilai *class cofidence* yang yang terdapat dari *bounding box*, untuk menunjukan seberapa akurat *bounding box* untuk memprediksi sesuai dengan objek yang dituju.

d) Perancangan Tampilan Pada Sistem (Mockup)

Pada tampilan aplikasi ini pengguna atau user cukup melihat kearah kamera dan wajah menghadap ke kamera kemudian sistem akan mengidentifikasi wajah dan penggunaan masker. Kemudian setelah

teridentifikasi sistem akan mengirimkan sebuah pesan informasi apakah penggunaan masker sudah sesuai atau belum.



Gambar 3.5 Mockup Interface Sistem

3.4.5 Training

training yang akan digunakan pada metode yolo ini dengan menggunakan arsitektur dari darknet53, yang dimana memiliki *convolutional layer* yang memiliki 53 lapisan konvolusional yang digunakan pada metode yolo ini. Pada darknet53 ini terdiri dari struktur *convolutional* dan *residual*. Pada metode model ini menggunakan yolov3 *convolutional layer* yang selalu di ikuti oleh batch normalization dan leaky ReLu. Sebagai metode dalam melatih model yolo ini agar dapat mendeteksi sebuah objek berupa penggunaan masker.

	Type	Filters	Size	Output
1x	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
2x	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
8x	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
8x	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
4x	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
				Softmax

Gambar 3.6 Arsitektur Darknet53

3.4.6 Testing

Pada tahap pengujian dalam penelitian ini yang dilakukan memiliki dua aspek pengujian yaitu, sistem dapat berjalan dengan baik, dan yang kedua ialah tingkat keakurasian sistem dalam mengidentifikasi objek. Pada tahapan pengujian pertama semua sistem di uji dengan menyalakan kamera dan mendeteksi sebuah objek hingga melakukan proses deteksi pada objek atau citra. Jika semua berfungsi dengan baik dan dapat mengidentifikasi sebuah objek maka sistem dapat dikategorikan berhasil. Sedangkan untuk pengujian tahap kedua tingkat keakurasian hasil dari indentifikasi penggunaan masker akan diperhatikan pada sistem yang telah di buat. Hasil dari indentifikasi ini kemudian akan dibandingkan dengan hasil indentifikasi penggunaan masker secara manual. Apabila hasil dari indentifikasi sama, maka dalam pengujian ini sistem dapat dikatakan berhasil.

3.4.7 Evaluasi

Pada tahap evaluasi ini menggunakan keakurasian dengan menggunakan metode *Confusion Matriks* untuk mengetahui seberapa akurat dalam mendeteksi sebuah objek atau citra yang telah di deteksi berdasarkan dataset yang ada dengan berbagai uji tes yang di lakukan apakah intensitas cahaya berpengaruh terhadap deteksi objek dan jarak antara kamera dan objek. Untuk itu dengan adanya evaluasi ini dapat mengetahui seberapa keakuratan aplikasi dalam mendeteksi sebuah objek dengan menggunakan Rumus *Confusion Matriks* sebagai berikut:

Tabel 3.4 Confusion Matrix

		Prediksi	
		Positif	Negatif
Aktual	Positif	TP	FN
	Negatif	FP	TN

Keterangan:

TP (*True Positive*) adalah banyaknya data dari *class* aktualnya yang merupakan *class* positif dengan kelas prediksinya adalah *class* positif.

FN (*False Negative*) adalah banyaknya data dari *class* aktualnya yang merupakan *class* positif dengan *class* prediksinya adalah *class* negatif.

FP (*False Positive*) adalah banyaknya data dari *class* aktualnya yang merupakan *class* negatif dengan *class* prediksinya adalah *class* positif.

TN (*True Negative*) adalah banyaknya data dari *class* aktualnya yang merupakan *class* negative dengan *class* prediksinya adalah *class* negative.

1. Presisi

Presisi adalah rasio dari hasil jumlah objek yang dapat dideteksi dengan benar atau *true positive* yang dibandingkan dengan jumlah data yang terprediksi secara positif. Semakin besar semakin besar nilai *false positive*. Maka nilai dari presisi semakin rendah dan begitu sebaliknya

$$precision = \frac{TP}{TP+FP} \times 100\%$$

2. Recall

Recall adalah jumlah hasil dari rasio objek yang dideteksi dengan benar atau *true positive* yang dibandingkan dengan jumlah data yang positif. Nilai *recall* yang tinggi dapat menunjukkan bahwa sistem tersebut dapat mengklasifikasikan *class* objek dengan benar. Dengan menghitung nilai *recall* dengan rumus sebagai berikut:

$$recall = \frac{TP}{TP + FN} \times 100\%$$

3. Akurasi

Akurasi adalah sebuah parameter untuk menguji kehandalan sistem dalam mengklasifikasikan sebuah objek. Akurasi untuk menentukan kebenaran dalam pengujian sistem terhadap objek yang. Dengan menghitung nilai akurasi dengan menggunakan rumus sebagai berikut:

$$accuracy = \frac{Total\ TP}{Total\ Data\ uji}$$