

## BAB III

### METODE PENELITIAN

#### A. Metode dan Desain Penelitian

Metode penelitian yang digunakan oleh peneliti adalah Metode *System Development Life Cycle (SDLC)*. Menurut Sukamto dan Shalahuddin (dalam Firmansyah & Udi, 2017) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya, berdasarkan *best practice* atau cara-cara yang sudah teruji baik. SDLC juga merupakan proses evolusioner yang diikuti dalam menerapkan sistem dan subsistem informasi berbasis komputer (Sutabri dalam Permana & Sahara, 2018). SDLC memanfaatkan pengembangan sistem secara bertahap dari mulai tahap perencanaan hingga tahap pemeliharaan. SDLC adalah tahapan kerja yang bertujuan untuk menghasilkan sistem berkualitas tinggi yang sesuai dengan keinginan pelanggan atau tujuan dibuatnya sistem tersebut (Binus, 2020).

Model SDLC yang dipilih oleh peneliti merupakan model *waterfall* dimana setiap tahap dilakukan satu persatu secara berurutan dan tidak dapat dikerjakan bila tahap sebelumnya belum diselesaikan. Urutan dalam metode SDLC model *Waterfall* bersifat serial yang dimulai dari proses perencanaan, analisa, desain, implementasi dan pemeliharaan pada sistem (Satriawan, 2022). Model *waterfall* melakukan teknik pengembangan tim untuk merincikan dengan mengumpulkan dan menentukan kebutuhan sistem sebelum sistem tersebut dikembangkan (Simamarta Dalam Pratama & Meilinda, 2018).

Menurut Rizky (2019) Beberapa kelebihan model *waterfall* diantaranya, pengembangan sistem paling handal dan kompleks yang digunakan, model ini cocok untuk sistem yang memiliki kompleksitas rendah, serta pengerjaan akan terjadwal dan mudah dikontrol. Namun disamping itu, terdapat kekurangan pada model ini antara lain, pengerjaan akan memakan waktu yang cukup lama, pembiayaan akan lebih besar karena dikerjakan pertahap, dan tidak cocok untuk projek berulang dengan kompleksitas yang tinggi.

## **B. Populasi dan Sampel**

Populasi adalah suatu kesatuan individu atau subyek pada wilayah dan waktu dengan kualitas tertentu yang akan diamati/diteliti. Pada penelitian ini, peneliti memilih populasi siswa sekolah SMKN 2 Bandung kelas X Animasi Multimedia (AMM).

Sampel merupakan bagian dari populasi yang ingin diteliti. Menurut Sugiyono (dalam OKTAVIAN, 2018) “Sampel adalah bagian dari jumlah dan karakteristik yang dimiliki oleh populasi tersebut. Sampel yang akan digunakan oleh peneliti adalah 32 siswa kelas X AMM yang akan melewati pintu masuk lab sekolah untuk dideteksi apakah mereka menggunakan masker atau tidak menggunakan masker.

## **C. Instrumen Penelitian**

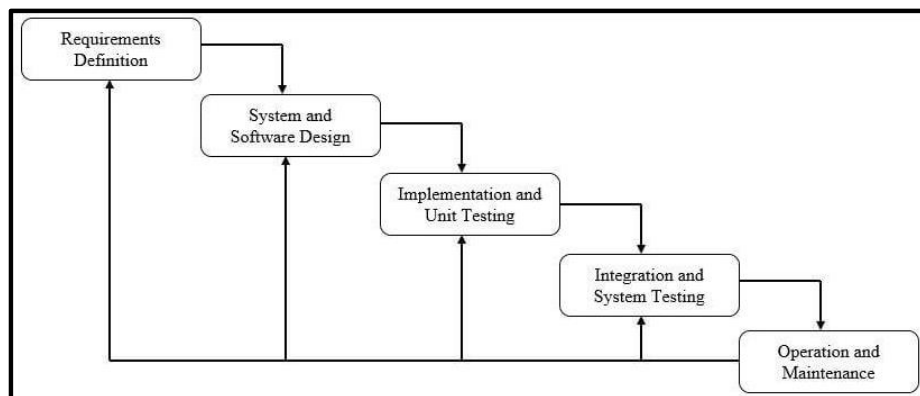
Instrumen penelitian adalah alat-alat yang diperlukan atau dipergunakan untuk mengumpulkan data (Alhamid & Anufia, 2019). Instrumen berfungsi sebagai alat bantu dalam mengumpulkan data yang di perlukan. Bentuk instrumen berkaitan dengan metode pengumpulan data, misal metode wawancara yang instrumennya pedoman wawancara. Instrumen adalah alat atau fasilitas yang digunakan penelitian dalam mengumpulkan data agar pekerjaannya lebih mudah dan hasilnya lebih baik, sehingga mudah diolah (Arikunto dalam, Alhamid & Anufia, 2019). Menyusun instrumen pada dasarnya adalah menyusun alat evaluasi, karena mengevaluasi adalah memperoleh data tentang sesuatu yang diteliti, dan hasil yang diperoleh dapat diukur dengan menggunakan standar yang telah ditentukan sebelumnya oleh peneliti (Aisyah, 2018).

Pada penelitian ini, instrumen penelitian adalah peneliti itu sendiri. Bentuk instrumen pada penelitian ini menggunakan instrumen observasi. Peneliti akan mengamati hasil penelitian di lapangan sebagai output dari program deteksi masker yang telah dibuat. Objek penelitian ini adalah siswa kelas X AMM SMKN 2 Bandung.

## D. Prosedur Penelitian

Metode SDLC Waterfal merupakan salah satu metode yang mempunyai ciri khas bahwa pengerjaan setiap fase harus dikerjakan terlebih dahulu sebelum melanjutkan ke fase berikutnya (Nugraha et al., 2018).

Ian Sommerville menjelaskan (Dalam Hidayat, 2020) lima tahapan model *waterfall* diantaranya, *Requirements Analysis and Definition*, *System and Software Design*, *Implementation and Unit Testing*, *Integration and System Testing*, dan *Operation and Maintenance*.



Gambar 3. 1 Tahapan penelitian model *waterfall* oleh Ian Sommerville

### 1. *Requirements Analysis and Definition*

Peneliti merancang dan menganalisis kebutuhan program yang diperlukan untuk penelitian. Tahap ini adalah proses awal mengumpulkan data dan menentukan penggunaan bahasa program yang akan dibangun untuk program deteksi makser.

Perancangan program deteksi masker memanfaatkan teknologi *Artificial Intellegence* (AI) atau kecerdasan buatan dan *machine learning* yang sebagian besar menggunakan bahasa pemrograman python. Penggunaan bahasa python memiliki beberapa kelebihan diantaranya :

- *Open source*
- Memiliki *library* yang dapat digunakan untuk *machine learning*
- Penulisan sintaks sederhana dan mudah dimengerti

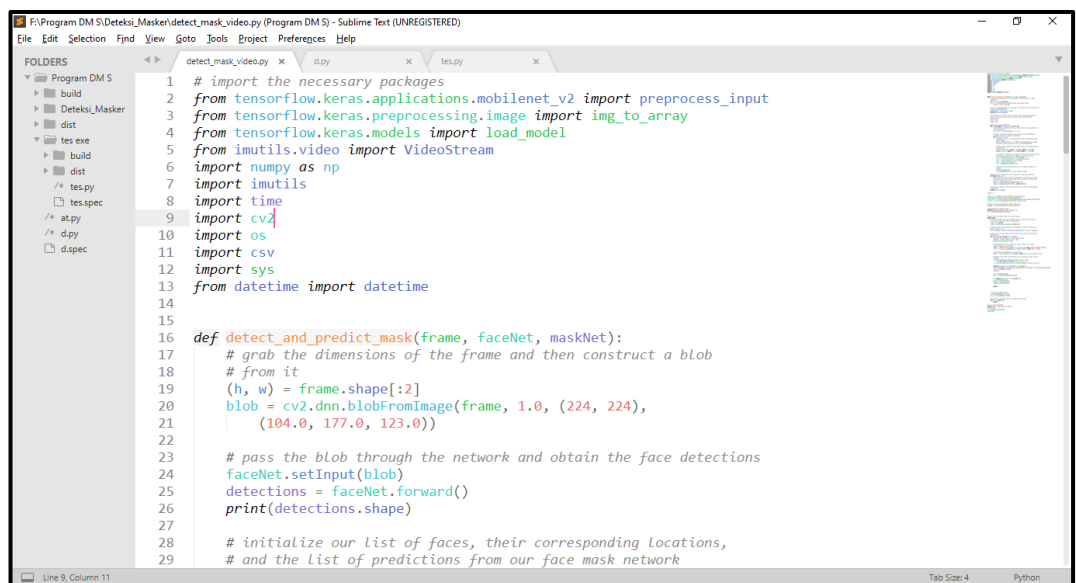
Dalam pengembangan program deteksi masker dengan python, terdapat beberapa *library* utama yang digunakan untuk dapat menangkap gambar melalui external webcam yaitu opencv dan tensorflow.

Peneliti juga memerlukan *dataset* foto orang menggunakan masker dan tidak menggunakan masker sebagai data yang akan dilatih oleh program dalam mendeteksi masker.

## 2. System and Software Design

Setelah melakukan analisis yang dibutuhkan oleh program, peneliti membuat desain program yang akan dibangun menggunakan prosedur pengkodean yang akan digunakan.

Bahasa pemrograman yang digunakan adalah bahasa pemrograman python dengan beberapa *library* tambahan yaitu tensorflow dan opencv. Penulisan sintaks pemrograman dilakukan pada aplikasi *code editor* sublime text dan dijalankan menggunakan command prompt.



```

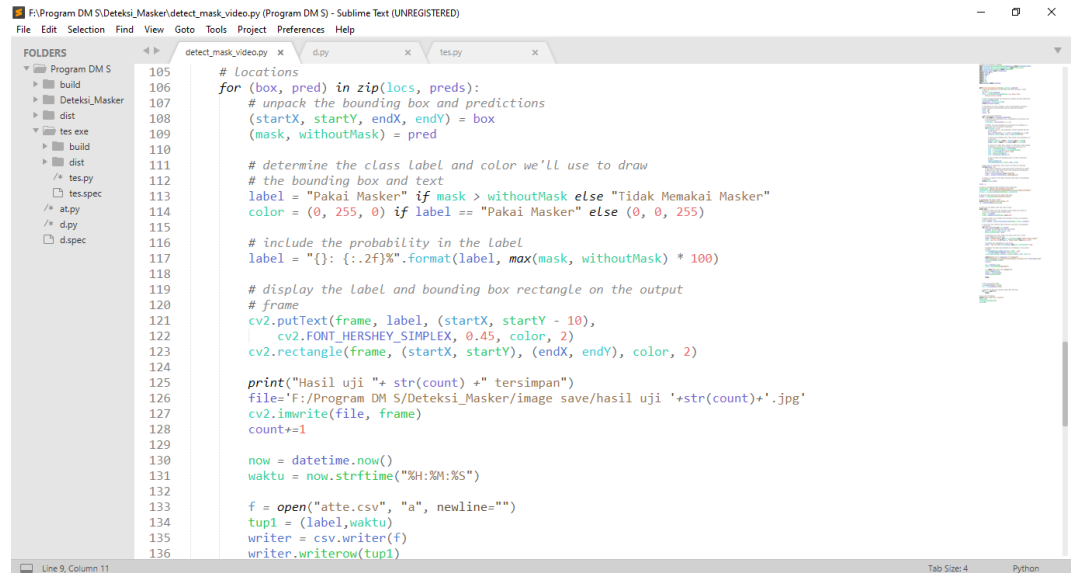
1 # import the necessary packages
2 from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
3 from tensorflow.keras.preprocessing.image import img_to_array
4 from tensorflow.keras.models import load_model
5 from imutils.video import VideoStream
6 import numpy as np
7 import imutils
8 import time
9 import cv2
10 import os
11 import csv
12 import sys
13 from datetime import datetime
14
15
16 def detect_and_predict_mask(frame, faceNet, maskNet):
17     # grab the dimensions of the frame and then construct a blob
18     # from it
19     (h, w) = frame.shape[:2]
20     blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
21     | (104.0, 177.0, 123.0))
22
23     # pass the blob through the network and obtain the face detections
24     faceNet.setInput(blob)
25     detections = faceNet.forward()
26     print(detections.shape)
27
28     # initialize our list of faces, their corresponding locations,
29     # and the list of predictions from our face mask network

```

Gambar 3. 2 aplikasi code editor sublime text dan penggalan sintaks program Penulisan sintaks program menggunakan fitur bahasa python sehingga sintaks akan otomatis terstruktur dengan warna yang membedakannya. Penggunaan *code editor* sublime text dikarenakan aplikasi ini ringan dibandingkan dengan aplikasi *code editor* lain.

### 3. Implementation and Unit Testing

Tahap ini merupakan tahap pengkodean dan pembuatan program yang telah dirancang sebelumnya pada tahap *System and Software Design*.



```

105 # Locations
106 for (box, pred) in zip(locs, preds):
107     # unpack the bounding box and predictions
108     (startX, startY, endX, endY) = box
109     (mask, withoutMask) = pred
110
111     # determine the class label and color we'll use to draw
112     # the bounding box and text
113     label = "Pakai Masker" if mask > withoutMask else "Tidak Memakai Masker"
114     color = (0, 255, 0) if label == "Pakai Masker" else (0, 0, 255)
115
116     # include the probability in the label
117     label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
118
119     # display the label and bounding box rectangle on the output
120     # frame
121     cv2.putText(frame, label, (startX, startY - 10),
122                cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
123     cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
124
125     print("Hasil uji "+ str(count) +" tersimpan")
126     file='F:/Program DM 5/Deteksi_Masker/image save/hasil uji '+str(count)+'.jpg'
127     cv2.imwrite(file, frame)
128     count+=1
129
130     now = datetime.now()
131     waktu = now.strftime("%H:%M:%S")
132
133     f = open("atte.csv", "a", newline="")
134     tup1 = (label,waktu)
135     writer = csv.writer(f)
136     writer.writerow(tup1)

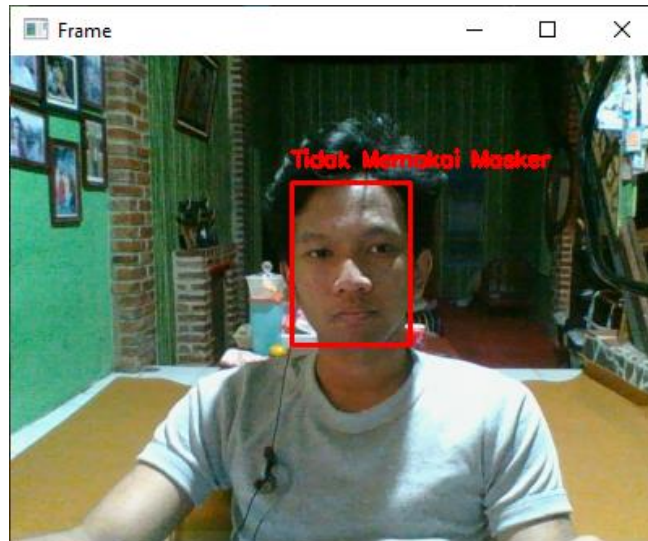
```

Gambar 3. 3 Sintaks program

Pada tahap ini, sintaks program mulai disusun dan jalankan agar memberikan hasil output program yang sesuai. Pengkodean sintaks perlu memperhatikan posisi *indent* atau penulisan menjorok untuk menghindari *error* pada program dan bahasa pemrograman *python* merupakan bahasa pemrograman dengan *case sensitive*. Artinya penulisan sintaks huruf kecil dan huruf besar berbeda.

### 4. Integration and System Testing

Setelah menyelesaikan tahap pengkodean, program melalui tahap pengujian yang telah dibuat untuk memastikan program beroperasi dengan semestinya tanpa ada kesalahan. Program dijalankan dengan *command prompt* pada *operating system windows* dan menggunakan kamera *external* yang dipasang di laptop. Setelah dijalankan, program akan menampilkan jendela yang menangkap wajah lewat kamera *external*. Tampilan program saat dijalankan dapat dilihat pada gambar 3.4.



Gambar 3. 4 uji running program

Tampilan jendela program berukuran 100x100 pixel. Saat program dijalankan, jendela program tidak dapat diubah ukurannya. Program akan secara otomatis menangkap wajah yang terdeteksi di jendela program.

Jendela program dijalankan bersamaan dengan *command prompt*. Apabila jendela *command prompt* tertutup, maka jendela program juga akan ikut tertutup. Hal ini dikarenakan program dipanggil menggunakan *command prompt*. Deklarasi pemanggilan program pada *command prompt* dapat dilihat pada gambar 3.5.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1706]
(c) Microsoft Corporation. All rights reserved.

F:\Program DM S\Deteksi_Masker>python detect_mask_video.py
2022-06-16 21:53:31.721619: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dLError: cudart64_110.dll not found
2022-06-16 21:53:31.745602: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine.
2022-06-16 21:54:27.023364: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dLError: nvcuda.dll not found
2022-06-16 21:54:27.036460: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-06-16 21:54:27.057853: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: Darwin
2022-06-16 21:54:27.058327: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: Darwin
2022-06-16 21:54:27.163013: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
[INFO] starting video stream...
(1, 1, 200, 7)
(1, 1, 200, 7)
Hasil uji 0 tersimpan
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)
(1, 1, 200, 7)

```

Gambar 3. 5 menjalankan program dengan command prompt

Setelah dijalankan, program akan menangkap wajah yang terdeteksi. Program telah dirancang untuk mendeteksi wajah memakai masker dan wajah tidak memakai masker. Output dari program berupa tangkapan gambar dapat dilihat seperti pada gambar 3.6.



Gambar 3. 6 Output program deteksi masker

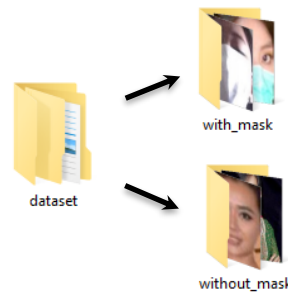
Output program merupakan tangkapan gambar wajah memakai masker dan wajah tidak memakai masker. Output disimpan di dalam folder program yang dituliskan alamatnya pada sintaks program.

##### 5. *Operation and Maintenance*

Pada tahap ini, program yang telah dibuat akan dianalisis kembali dan ditambahkan atau diubah sesuai kebutuhan.

## E. Teknik Analisis Data

Pada penelitian ini, peneliti membagi *dataset* menjadi dua bagian yaitu data wajah menggunakan masker dan data wajah tidak menggunakan masker. Setiap *dataset* terdiri dari 1980 data foto wajah yang akan di *training* sebelum memasuki tahap pre-proses. Dataset tersebut dibagi kedalam masing- masing direktori seperti gambar dibawah ini.



Gambar 3. 7 Direktori *Dataset*

Data yang telah di *training* akan diuji akurasi menggunakan data testing untuk menghitung akurasi hasil uji coba data testing dapat menggunakan rumus pada gambar dibawah ini.

$$\text{Akurasi} = \frac{\text{Jumlah Hasil Uji Coba yang Benar}}{\text{Total Uji Coba}} \times 100 \%$$

Gambar 3. 8 Rumus Akurasi Uji Coba Program

Untuk memperoleh hasil penelitian yang diinginkan, program akan diimplementasikan secara *realtime* untuk mendeteksi tingkat penggunaan masker di sekolah. Real time maksudnya adalah wajah yang berhasil dikenali akan terdata dan masuk ke dalam database sesuai tanggal dan jam pada saat melakukan proses absensi, dimana proses absensi dilakukan secara langsung melalui webcam (JURJAWI, 2020). Setelah dilakukan *training* dan *testing* pada program, dilakukan pendekatan statistik yang berhubungan dengan efektivitas program dengan melakukan perbandingan variabel pengujian pada tabel 1.



Tabel 3. 1. Variabel Uji Berdasarkan Kondisi Data Sebenarnya dan Hasil Identifikasi

Data Sebenarnya		Hasil Identifikasi	
Dikenal (Ada di Dataset)	Tidak Dikenal (Tidak Ada di Dataset)	Sistem Berhasil Dalam Mengenali	Sistem Gagal Dalam Mengenali
<i>True Positif (TP)</i>	<i>True Negatif (TN)</i>	<i>True Positif (TP)</i>	<i>False Positif (FP)</i>
<i>False Positif (FP)</i>	<i>False Negatif (FN)</i>	<i>True Negatif (TN)</i>	<i>False Negatif (FN)</i>

Untuk menghitung akurasi secara keseluruhan berdasarkan data uji, data dihitung menggunakan rumus dibawah ini.

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100 \%$$

Gambar 3. 9 Rumus Akurasi Uji Keseluruhan

Keterangan :

1. *True Positif (TP)* : Kondisi dimana masker berhasil diidentifikasi oleh program sesuai yang ada pada dataset.
2. *False Positif (FP)* : Kondisi dimana masker gagal diidentifikasi oleh program sesuai yang ada pada dataset, atau program salah dalam mengidentifikasi masker tersebut.
3. *True Negatif (TN)* : Kondisi dimana tidak memakai masker berhasil diidentifikasi oleh program sebagai tidak ada masker.
4. *False Negatif (FN)* : Kondisi dimana tidak memakai masker gagal diidentifikasi oleh program sebagai tidak ada masker, atau justru teridentifikasi dengan masker yang ada pada *dataset*.

Variabel-variabel pengujian diatas akan digunakan untuk menentukan akurasi secara keseluruhan berdasarkan hasil pengujian dari sampel data uji.

Persentase yang telah diperoleh kemudian diolah menggunakan skala penilaian (*rating scale*). *Rating scale* dilakukan untuk mengetahui kualitas dari produk yang telah dibuat dalam bentuk persentase. Pembagian skala dilakukan

dengan memperhatikan bilangan dengan ukuran yang telah ditetapkan. Nilai maksimal atau kondisi maksimal yang diharapkan yaitu 100%. Maka skala 0% hingga 100% dibagi rata untuk menghasilkan kategori kelayakan. Kategori kelayakan dapat dilihat pada tabel 3.2.

Tabel 3. 2 Kategori Penilaian Produk Multimedia Menggunakan Rating scale (Ernawati, 2017)

Kategori	Persentase
Sangat Layak	81% - 100%
Layak	61% - 80%
Cukup Layak	41% - 60%
Tidak Layak	21% - 40%
Sangat Tidak Layak	0% - 20%

Kriteria penilaian diatas menjadi acuan dari hasil penilaian yang digunakan untuk mengetahui baik tidaknya program yang telah dikembangkan.