

## **BAB III**

### **METODE PENELITIAN**

Pada bab 3 akan dijelaskan mengenai metode penelitian yang membahas mengenai *tools* yang digunakan untuk melakukan penelitian serta pengembangan sistem. Pada bagian ini dibagi menjadi beberapa tahap yaitu, alat dan bahan untuk mendukung proses pembuatan sistem, alur penelitian, perancangan model sistem presensi berbasis teknologi *scan* wajah dan ai pada bidang cv dengan arsitektur *single shot multibox detector (ssd)*, metode pengujian, *flowchart system*, dan blok diagram. Metode penelitian digunakan dengan melakukan serangkaian uji coba untuk dapat mengetahui hasil akhir yang didapatkan dari penelitian.

#### **3.1 ALAT YANG DIGUNAKAN**

Dalam penelitian ini, terdapat beberapa perangkat keras dan perangkat lunak yang digunakan, yaitu:

##### **3.1.1 Perangkat Keras**

Berikut ini akan dijelaskan mengenai perangkat keras yang akan digunakan dalam penelitian beserta fungsinya. Informasinya dapat dilihat pada Tabel 3.1.

**Tabel 3.1 Penggunaan perangkat keras dan fungsinya**

No.	Nama Perangkat Keras	Fungsi
1	Laptop	Merancang perangkat lunak dengan menggunakan aplikasi berbasis <i>Windows</i>
2	Kamera <i>Handphone</i>	Mengumpulkan <i>dataset</i> berupa gambar wajah manusia.

##### **3.1.2 Perangkat Lunak**

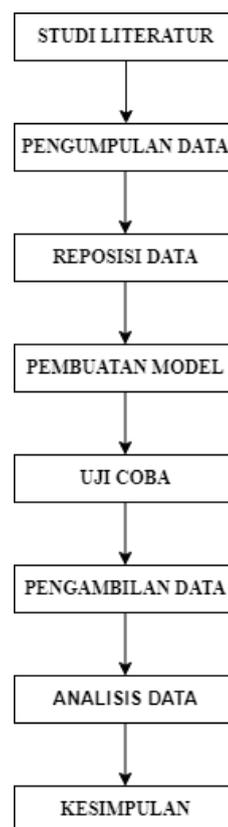
Pada penelitian ini, digunakan beberapa perangkat lunak yang berjalan di sistem operasi *Windows*. Berikut adalah daftar perangkat lunak yang digunakan

dalam penelitian ini:

1. *Google Colaboratory*
2. *LabelImg*
3. *Command Prompt (CMD)*

### 3.2 ALUR PENELITIAN

Penelitian ini melibatkan serangkaian tahapan untuk mencapai hasil yang diharapkan. Salah satunya adalah mengimplementasikan arsitektur *Single Shot Multibox Detector (SSD)* dengan menggunakan arsitektur *MobileNetV2* sebagai *backbones* dan *FPN lite* yang memiliki resolusi input 320x320, dilatih pada *dataset COCO 2017* dalam kecerdasan buatan pada bidang komputer vision untuk metode presensi. Tahapan-tahapan ini akan dijelaskan melalui diagram alur gambar 3.1.



**Gambar 3.1 Diagram alur penelitian**

Gambar 3.1 di atas menggambarkan tahapan alur penelitian yang meliputi studiliteratur, pengumpulan data, reposisi data, pembuatan model, uji coba, analisis data, hasil dan pembahasan, kesimpulan. Penjelasan untuk masing-masing tahapan

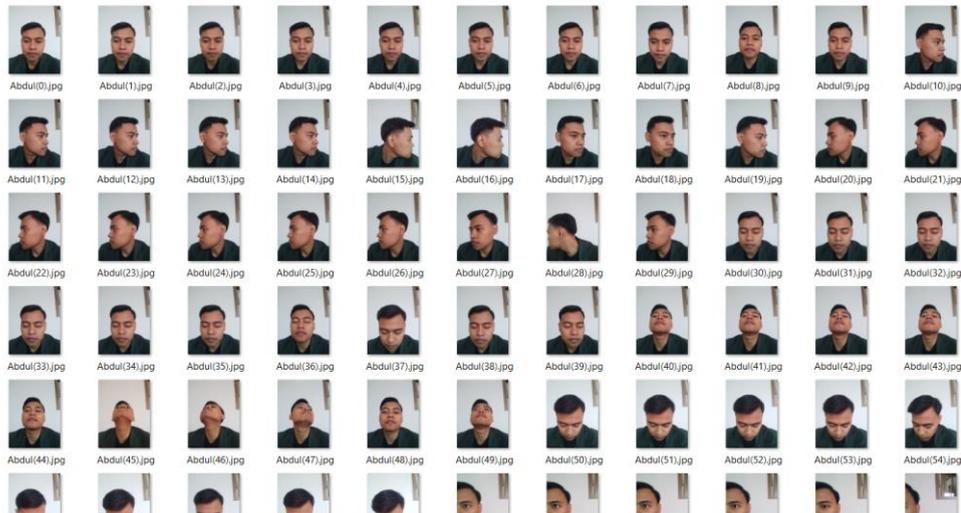
penelitian dari diagram alur tersebut adalah sebagai berikut:

### 3.2.1 Studi Literatur

Pada tahap studi literatur, dilakukan pengumpulan, pemahaman, dan literatur terkait dengan arsitektur algoritma *Single Shot Multibox Detector* (SSD) dengan menggunakan arsitektur *MobileNetV2* sebagai *backbones* FPN *lite*, yang memiliki resolusi input 320x320, dilatih pada *dataset* COCO 2017, perangkat lunak yang diperlukan untuk mendukung penelitian, serta persiapan pengumpulan data yang diperlukan dalam implementasi arsitektur SSD-*MobileNetV2*. Sumber literatur yang digunakan mencakup jurnal, artikel, dan situs web yang terpercaya dan relevan dalam bidang penelitian ini.

### 3.2.2 Pengumpulan Data

Pada tahap ini, dilakukan proses pengumpulan data dengan melakukan pencarian 200 foto citra wajah individu, termasuk Abdul Zulfiantiko, dan Eko Prasetyo. Pengumpulan data dilakukan melalui media kamera *handpohone* yang diambil secara langsung dengan menghasilkan format gambar berupa JPG. Pengumpulan gambar dapat dilihat pada gambar 3.2 dan gambar 3.3.



Gambar 3.2 Abdul Zulfiantiko



**Gambar 3.3 Eko Prasetyo**

Hasil pengumpulan data terdiri dari 200 *image* wajah, dimana 100 wajah Abdul Zulfiantiko, dan 100 wajah Eko Prasetyo. Komposisi data pada tahap ini dapat dilihat pada Tabel 3.2.

**Tabel 3.2 Komposisi Data**

No	List Nama	Jumlah Gambar
1	Abdul Zulfiantiko	100
2	Eko Prasetyo	100

### 3.2.3 Preposisi Data

Pada tahap ini, setelah berhasil mengumpulkan seluruh data, selanjutnya dilakukan proses preposisi data. Preposisi data merupakan tahapan krusial yang dilakukan untuk mempersiapkan data sebelum dilakukan pengolahan lebih lanjut. Dalam penelitian ini, proses preposisi data terdiri dari empat langkah utama yang saling berhubungan, yaitu *resize* gambar, pemisahan *data train* dan *test*, anotasi gambar, dan konversi menjadi format CSV dan TFRecord.

#### 1. *Resize* Gambar.

Dalam proses preposisi data, langkah *resize* atau mengubah ukuran data menjadi kritis karena mayoritas gambar memiliki resolusi yang sangat tinggi, yang dapat menyulitkan proses pengolahan dan analisis. Dalam penelitian ini, secara konsisten mengubah semua gambar menjadi ukuran 320x320 piksel. Hal

ini dilakukan dengan beberapa pertimbangan yang perlu diperhatikan.

*Resize* atau perubahan ukuran gambar menjadi 320x320 piksel bertujuan untuk mencapai beberapa tujuan penting:

- 1) Efisiensi Pengolahan: Dengan mengubah ukuran gambar menjadi standar 320x320 piksel, kami dapat meningkatkan efisiensi dan kecepatan dalam proses pengolahan data. Ukuran yang lebih kecil memungkinkan komputasi yang lebih ringan dan cepat, yang dapat membantu mengurangi waktu pemrosesan secara keseluruhan.
- 2) Konsistensi Data: Proses *resize* dilakukan untuk menjaga konsistensi ukuran gambar dalam *dataset*. Dengan mengubah semua gambar menjadi ukuran yang sama, dipastikan bahwa setiap data memenuhi persyaratan yang seragam, sehingga dapat diolah dengan metode analisis yang sama tanpa ada bias ukuran yang mempengaruhi hasil.
- 3) Penghematan Penyimpanan: Gambar dengan resolusi tinggi biasanya memerlukan lebih banyak ruang penyimpanan. Dengan mengubah ukuran menjadi 320x320 piksel, maka dapat menghemat ruang penyimpanan tanpa mengorbankan informasi esensial dari gambar tersebut.
- 4) Meningkatkan Kinerja Model: Ukuran gambar yang terlalu besar dapat membebani model dan algoritma yang digunakan dalam proses pengenalan dan analisis. Dengan mengurangi ukuran gambar, model dapat berfokus pada fitur- fitur penting tanpa terlalu terbebani oleh ukuran yang besar.

Penting untuk mencatat bahwa dalam melakukan proses *resize* ini, faktor aspek *ratio* perlu diperhatikan agar gambar tetap proporsional dan tidak mengalami distorsi yang signifikan. Selain itu, perubahan ukuran juga mempertimbangkan kualitas visual yang memadai sehingga informasi pada gambar tetap dapat dikenali dengan baik.

Semua langkah dalam proses preposisi data ini bertujuan untuk memastikan kualitas dan reliabilitas data sebelum dilakukan tahap pengolahan dan analisis lebih lanjut.

## 2. Pemisahan Data Uji dan Latih.

Dalam tahapan preposisi data yang dilakukan setelah proses *resize* seluruh data, selanjutnya dilakukan proses pemisahan data menjadi dua bagian utama,

yaitu data uji dan data latih. Proses ini dilakukan dengan memperhatikan rasio yang telah ditentukan yaitu 80:20. Langkah-langkah yang dilakukan untuk pemisahan data uji dan latih adalah sebagai berikut:

- 1) Pengaturan Rasio Data: Pertama-tama, kami menentukan rasio 80:20 atau persentase yang akan digunakan untuk memisahkan data menjadi data uji dan data latih. Rasio ini berdasarkan kebutuhan penelitian dan pertimbangan tertentu untuk mencapai tujuan analisis yang diinginkan.
- 2) Pemisahan Data: Setelah menentukan rasio yang sesuai, dilakukan pemisahan data menjadi dua kelompok, yaitu data uji (*test*) dan data latih (*train*). Data latih akan digunakan untuk melatih model atau algoritma yang akan digunakan dalam analisis data. Sementara itu, data uji akan digunakan untuk menguji performa dan keakuratan dari model yang telah dilatih.
- 3) Randomisasi Pemisahan: Saat melakukan pemisahan data, kami memastikan bahwa proses pemilihan data untuk masing-masing kelompok (data uji dan data latih) bersifat acak. Dengan randomisasi ini, dapat menghindari bias dalam data yang digunakan untuk melatih atau menguji model.
- 4) Validasi dan Evaluasi: validasi dilakukan untuk memastikan bahwa pemisahan data telah dilakukan dengan benar dan sesuai dengan rasio yang ditentukan yaitu 80:20. Evaluasi dilakukan untuk memastikan bahwa data uji dan data latih mewakili seluruh *dataset* secara representatif.

Pemisahan data menjadi data uji dan data latih ini memungkinkan kami untuk melatih model dengan data yang relevan dan menguji performa model dengan data yang belum pernah "dilihat" sebelumnya. Dengan demikian, hasil analisis menjadi lebih objektif dan dapat diandalkan dalam mengambil kesimpulan dari penelitian ini.

### 3. Anotasi Gambar.

Dalam tahap anotasi data gambar, kami menggunakan perangkat lunak *labeling* untuk memberi label pada setiap gambar yang terdapat dalam bagian data *train* dan *test*. Pada setiap gambar yang menggambarkan wajah Abdul Zulfiantiko, kami memberi label "Abdul\_Zulfiantiko", untuk gambar wajah

Eko Prasetyo diberilabel "Eko\_Prasetyo. Proses Anotasi gambar terdapat pada Gambar 3.4



**Gambar 3.4 Pemberian Anotasi pada gambar**

Proses anotasi dilakukan secara cermat dan teliti untuk memastikan setiap gambar menerima label yang sesuai dan akurat. Setelah selesai melakukan anotasi pada setiap gambar yang menggambarkan wajah, hasilnya disimpan dalam format *file* yang berekstensi XML.

Dalam format *file* XML ini, setiap informasi anotasi akan terstruktur dengan baik untuk setiap gambar wajah yang telah diberi label. Informasi yang disimpan meliputi label yang ditetapkan untuk masing-masing gambar, posisi atau *bounding box* dari wajah yang diberi label.

Tahap anotasi data ini merupakan langkah penting dalam mempersiapkan data sebelum proses pelatihan model atau algoritma. Dengan adanya informasi anotasi yang lengkap dan tepat, diharapkan dapat memberikan hasil yang akurat dan dapat diandalkan dalam pengenalan wajah dan klasifikasi objek dalam gambar.

#### 4. Konversi Menjadi Format CSV dan TFRecord.

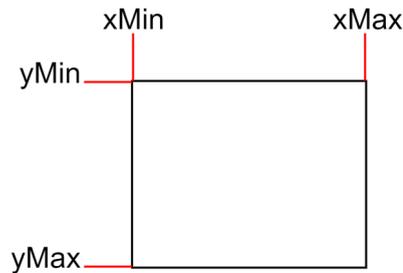
Setelah proses anotasi, data akan disimpan dalam format .xml dan selanjutnya dikonversi menjadi format CSV (*Comma-Separated Values*).

Data CSV yang dihasilkan berisi kumpulan data yang telah dianotasi sebelumnya, disusun dalam bentuk tabel. Tabel 3.4 menunjukkan contoh data CSV dengan kolom-kolom yang berisi informasi penting mengenai setiap gambar yang telah dianotasi.

Setiap baris mewakili satu objek yang terdapat dalam suatu gambar, dan

kolomnya memberikan informasi tentang objek tersebut. Sebagai contoh, pada baris pertama, objek terdapat dalam file gambar "10.jpg" dengan lebar 1728 piksel dan tinggi 2304 piksel. Objek tersebut memiliki kelas "Abdul\_Zulfiantiko" dan terletak dengan koordinat pojok kiri atas (xmin, ymin) = (5, 257) serta pojok kanan bawah (xmax, ymax) = (850, 2062).

Gambar 3.5 kordinat xmin, ymin, xmax, dan ymax membentuk sebuah kotak pembatas (*bounding box*) yang mengelilingi objek di dalam gambar. Dengan menggunakan nilai-nilai ini, kita dapat menentukan posisi dan ukuran relatif objek tersebut dalam gambar. Sebagai contoh tabel 3.3, jika kita memiliki nilai xmin=5, ymin=257, xmax=850, ymax=2062, maka objek tersebut akan memiliki kotak pembatas yang dimulai dari piksel (5,257) di pojok kiri atas dan berakhir di piksel (850,2062) di pojok kanan bawah.



**Gambar 3.5 Kordinat xmin, ymin, xmax, dan ymax**

**Tabel 3.3 Data CSV**

filename	width	height	class	xmin	ymin	xmax	ymax
10.jpg	1728	2304	Abdul_Zulfiantiko	5	257	850	2062
107.jpg	3072	4096	Eko_Prasetyo	697	539	2543	3407
109.jpg	3072	4096	Eko_Prasetyo	829	439	2365	2852
117.jpg	3072	4096	Eko_Prasetyo	1388	7	3072	4096
12.jpg	1728	2304	Abdul_Zulfiantiko	1	267	865	2062
123.jpg	3072	4096	Eko_Prasetyo	1374	21	3072	4096
124.jpg	3072	4096	Eko_Prasetyo	1038	16	3070	3802
126.jpg	3072	4096	Eko_Prasetyo	974	21	3072	3907
135.jpg	3072	4096	Eko_Prasetyo	15	7	1824	3966
162.jpg	1728	2304	Abdul_Zulfiantiko	277	587	1412	1937
168.jpg	1728	2304	Abdul_Zulfiantiko	315	564	1420	1977
173.jpg	1728	2304	Abdul_Zulfiantiko	315	552	1382	1947
175.jpg	1728	2304	Abdul_Zulfiantiko	362	619	1292	2082
196.jpg	1728	2304	Abdul_Zulfiantiko	355	522	1267	1924
2.jpg	1728	2304	Abdul_Zulfiantiko	215	524	1270	2104
41.jpg	1728	2304	Abdul_Zulfiantiko	280	889	1467	2302

Berikut adalah penjelasan isi dari setiap kolom pada tabel CSV:

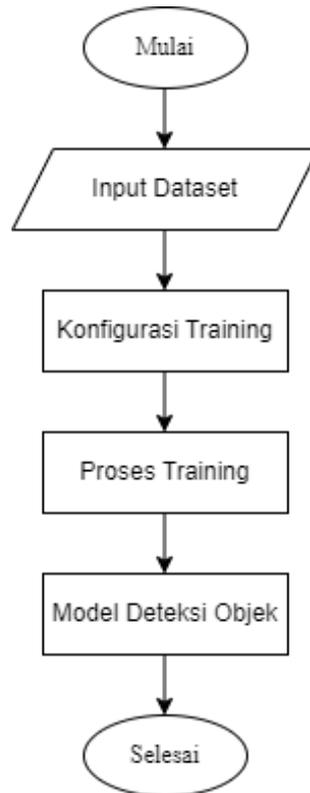
- 1) *Filename*: Kolom ini menyimpan nama *file* gambar yang telah dianotasi.
- 2) *Width*: Kolom ini berisi informasi mengenai lebar (*width*) dari gambar yang bersangkutan.
- 3) *Height*: Kolom ini berisi informasi mengenai tinggi (*height*) dari gambar yang bersangkutan.
- 4) *Class*: Kolom ini mencantumkan nama objek pada gambar yang telah dianotasi.
- 5) *x\_min*: Kolom ini menyimpan nilai jarak koordinat x-min dari letak objek yang telah dianotasi sebelumnya.
- 6) *y\_min*: Kolom ini menyimpan nilai jarak koordinat y-min dari letak objek yang telah dianotasi sebelumnya.
- 7) *x\_max*: Kolom ini menyimpan nilai jarak koordinat x-max dari letak objek yang telah dianotasi sebelumnya.
- 8) *y\_max*: Kolom ini menyimpan nilai jarak koordinat y-max dari letak objek yang telah dianotasi sebelumnya.

Setelah data diekstrak ke dalam data CSV, maka data akan dikonversi lagi untuk menghasilkan data *biner tfrecord* dengan menjalankan *script python generate\_tfrecord.py*.

Setelah dijalankan, akan menghasilkan berkas *test.record* dan *train.record* yang dapat digunakan untuk *input dataset* pada saat proses *training*.

### 3.2.4 Pembuatan Model

Proses pembuatan model deteksi objek dilakukan dengan menggunakan metode *Single Shot Multibox Detector* (SSD) dengan menggunakan arsitektur *MobileNetV2* sebagai *backbones*, melalui tahapan *input dataset*, konfigurasi *training*, proses *training*, dan menghasilkan model deteksi objek yang siap digunakan untuk tugas deteksi objek pada data baru. Gambar 3.6 merupakan alur pembuatan model. Tampak bahwa proses pembuatan model terdiri dari 4 tahapan, yaitu *input dataset*, konfigurasi *training*, proses *training*, dan model deteksi objek.



**Gambar 3.6 Flowchart proses pembuatan model**

Berikut ini merupakan penjelasan dari tahap pembuatan model :

1. *Input Dataset.*

Pada tahap awal pembuatan model, persiapan *dataset* dilakukan dengan ekstraksi dari gambar. Proses ini melibatkan beberapa langkah, dimulai dari ekstraksi informasi label dan koordinat objek dari file XML. Selanjutnya, *dataset* diubah menjadi format CSV. Proses terakhir melibatkan konversi *dataset* ke dalam format *tfrecord*. *Output* dari tahap ini adalah dua *file*, yaitu *train.record* dan *test.record*, yang akan menjadi *input dataset* pada tahap pelatihan model.

2. *Konfigurasi Training.*

Tahap ini berfokus pada konfigurasi model pelatihan. Konfigurasi ini mencakup pemilihan model deteksi objek, penyesuaian *hyperparameter* seperti *learning rate*, jumlah langkah pelatihan (*num\_train\_steps*), dan lainnya. Penggunaan *pre-trained model* seperti *ssd\_mobilenet\_v2\_fpnlite\_320x320\_coco17\_tpu-8* dari *TensorFlow Model Zoo* juga dapat terjadi pada tahap ini.

3. *Proses Training.*

Pada proses Training menggunakan *framework Tensorflow object detection* dan pengolahan citra dengan metode CNN-SSD.



#### 4. Model Deteksi Objek.

Selama proses pelatihan, model akan memperbarui parameter-parameter yang diperlukan untuk meningkatkan kemampuannya dalam mendeteksi objek. *Output* dari tahap ini adalah model yang telah dilatih dan dapat digunakan untuk melakukan deteksi objek pada data baru.

#### 3.2.5 Uji Coba

Pada tahap ini, dilakukan pengujian pada mekanisme pembelajaran dengan membandingkan rasio data latih (*train data*) dan data uji (*test data*). Pengujian ini bertujuan untuk mengukur performa model deteksi objek yang dihasilkan dari proses pembelajaran. Evaluasi model juga dilakukan menggunakan *confusion matrix* untuk mendapatkan informasi tentang akurasi, presisi, *recall*, dan F1-score.

*Confusion matrix* tabel 3.4 digunakan untuk mengevaluasi performa model klasifikasi. Tabel ini berisi empat komponen utama.

**Tabel 3.4 Confusion Matrix**

	Prediksi Positif (Model Deteksi)	Prediksi Negatif (Model Tidak Deteksi)
Aktual Positif	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Aktual Negatif	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

1. *True Positive* (TP): Jumlah sampel positif yang benar-benar terdeteksi dengan benar oleh model.
2. *True Negative* (TN): Jumlah sampel negatif yang benar-benar terdeteksi dengan benar oleh model.
3. *False Positive* (FP): Jumlah sampel negatif yang keliru terdeteksi sebagai positif oleh model.
4. *False Negative* (FN): Jumlah sampel positif yang keliru terdeteksi sebagai negatif oleh model.

Matrics :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

*Metrics* ini memberikan gambaran menyeluruh tentang seberapa baik model dapat membedakan antara kelas positif dan negatif, serta seberapa baik model dalam menghindari kesalahan klasifikasi.

### **3.2.6 Pengambilan Data**

Setelah pengujian sistem, hasil data yang diperoleh mencakup evaluasi performa model deteksi objek, termasuk tingkat akurasi dan performa lainnya. Data ini nantinya dapat diambil untuk mendukung analisis dan kesimpulan yang berkaitan dengan arsitektur model deteksi objek yang telah dikembangkan.

### **3.2.7 Analisis Data**

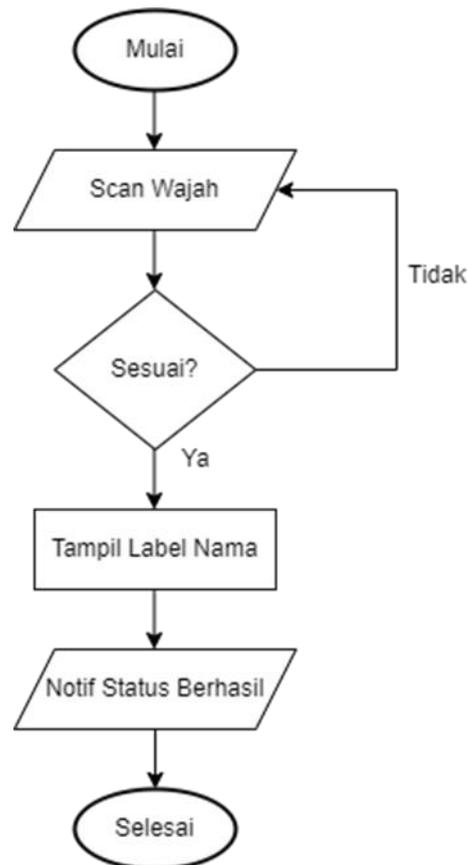
Pada tahap ini, data yang telah dikumpulkan pada tahap pengambilan data akan diolah dan dianalisis. Jika ditemukan bahwa hasil pengambilan data tidak mencapai tingkat akurasi yang diharapkan, langkah-langkah yang dapat dilakukan untuk meningkatkan akurasi antara lain menambahkan *dataset*, melakukan perbaikan pada perintah *coding* yang digunakan, atau melakukan analisis lebih lanjut tergantung pada permasalahan yang dihadapi. Tujuannya adalah untuk mencari solusi yang sesuai guna meningkatkan akurasi dan kualitas data yang digunakan dalam penelitian.

### **3.2.8 Kesimpulan**

Pada tahap terakhir ini, dilakukan penyusunan kesimpulan berdasarkan hasil penelitian yang telah dilaksanakan. Proses studi literatur, pengumpulan data, preposisi data, dan pembuatan model memberikan kontribusi positif terhadap akurasi hasil deteksi. Uji coba sistem presensi menunjukkan hasil yang memuaskan dengan menggunakan perbandingan data latih dan uji serta evaluasi menggunakan *confusion matrix*. Penggunaan teknologi ini diharapkan dapat meningkatkan keefektifan dalam kegiatan belajar mengajar. Model deteksi objek yang telah dilatih menggunakan *Single Shot Multibox Detector* (SSD) dengan menggunakan arsitektur *MobileNetV2* sebagai *backbones*, nantinya akan diterapkan ke dalam sistem presensi berbasis teknologi *scan* wajah dan ai pada bidang *computer vision* dengan arsitektur *single shot multibox detector* (SSD).

### 3.3 FLOWCHART SYSTEM

Sebelumnya sudah dilakukan pemanfaatan metode *Single-Shot MultiBox Detector* untuk mendeteksi nama, langkah selanjutnya adalah mengaplikasikan metode tersebut dalam metode presensi. Gambar 3.8 ini bertujuan untuk memantau keaslian kehadiran wajah yang terkait dengan nama yang melakukan presensi, terutama dalam konteks kegiatan belajar mengajar di lembaga pendidikan.



**Gambar 3.8 Flowchart sistem kerja presensi**

Gambar 3.8 merupakan ilustrasi proses untuk melakukan pendeteksian objek wajah, berikut ini merupakan penjelasan alur prosesnya :

1. *Scan Wajah.*

Foto gambar wajah akan dimanfaatkan untuk mengenali wajah mahasiswa/i yang akan melakukan presensi.

2. *Sesuai?*

Setelah itu, sistem akan memproses informasi tersebut untuk mengidentifikasikan subjek pada gambar. Menggunakan model yang telah

dilatih menjadi file *checkpoint*, serta menggunakan *Label Map* untuk mendapatkan hasil dari prediksi aktivitas. Setiap *frame* gambar akan dibaca oleh sistem, lalu hasil prediksinya berupa label nama akan ditampilkan menggunakan model yang sudah dilatih pada setiap *frame* tersebut. Jika belum berhasil maka akan kembali membaca *frame* gambar.

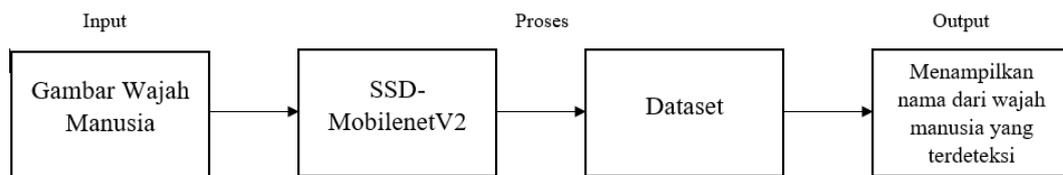
3. Notifikasi Status Berhasil.

Pesan notifikasi “Berhasil Presensi” jika data sesuai.

4. Selesai.

### 3.3.1 Blok Diagram

Pada penelitian ini, gambar 3.9 memiliki alur pengerjaan yang dijelaskan melalui blok diagram dengan poin berupa *input*, proses, dan *output*. Blok diagram tertampil padagambar berikut.



**Gambar 3.9 Diagram blok penelitian**

Gambar 3.7 merupakan ilustrasi blok diagram dari penelitian yang bertujuan untuk memanfaatkan arsitektur algoritma SSD yang menggunakan *MobileNetV2* dan *FPN lite*, memiliki resolusi *input* 320x320, dilatih pada *dataset* COCO 2017 untuk mendeteksi wajah manusia. Input dari penelitian ini berupa gambar foto wajah manusia yang telah diambil menggunakan kamera *handphone*. Setelah foto tersebut diambil, gambar akan diproses menggunakan arsitektur algoritma SSD yang menggunakan *MobileNetV2* dan *FPN lite*, memiliki resolusi *input* 320x320, dilatih pada *dataset* COCO 2017.

Arsitektur algoritma *SSD-MobileNetV2* akan melakukan proses pengolahan berdasarkan *dataset* yang sebelumnya telah dimasukkan ke dalamnya dan telah melalui tahap *training dataset*. Proses *training* ini bertujuan untuk mengenalkan algoritma dengan berbagai pola dan fitur wajah manusia sehingga algoritma dapat belajar mengidentifikasi wajah dengan lebih baik.

Setelah gambar *input*-an diolah dengan arsitektur SSD, algoritma akan

menghasilkan *output* berupa nama dari pemilik wajah yang telah difoto sebelumnya. Nama yang dihasilkan dari proses ini akan digunakan untuk mengidentifikasi wajah dan memungkinkan pemanfaatan lebih lanjut dari data wajah yang telah teridentifikasi.

Penelitian ini diharapkan dapat memberikan kontribusi dalam meningkatkan kemampuan deteksi wajah manusia menggunakan algoritma SSD-*MobilenetV2*. Dengan kemampuan identifikasi yang lebih baik, penggunaan algoritma ini dapat diaplikasikan dalam berbagai bidang, seperti keamanan, pengenalan wajah untuk akses, dan berbagai aplikasi lain yang memanfaatkan identifikasi individu berdasarkan wajah.