

## **BAB III**

### **METODE KERJA**

#### **3.1 Waktu dan Tempat**

Dalam program MSIB di PT. Orbit Ventura (*Orbit Future Academy*) dilaksanakan pada waktu dan tempat sebagai berikut :

Waktu Pelaksanaan : 21 Februari 2022 - 22 Juli 2022

Tempat Pelaksanaan : *online meeting*

#### **3.2 Alat dan Bahan**

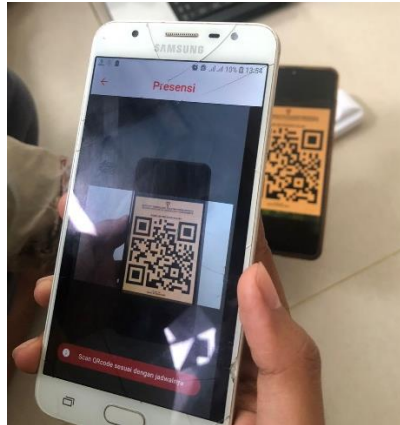
1. Perangkat Keras (*Hardware*)
  - a. PC atau Laptop
2. Perangkat Lunak (*Software*)
  - a. *Visual Studio Code*
  - b. *Google Colab*
  - c. *Android Studio*
3. Kartu Tanda Mahasiswa (KTM)

#### **3.3 Metode dan Proses Kerja**

Pelaksanaan project akhir dimulai dengan melakukan tinjauan permasalahan yang terjadi dilingkungan kami. Setelah kami berdiskusi secara daring kami memutuskan untuk mengambil topik permasalahan absensi di IT Telkom Purwokerto. Kemudian kami melakukan *exploring* masing-masing terlebih dahulu untuk menentukan metode-metode perangkat apa saja yang diperlukan. Berikut tahap pelaksanaan dalam pengerjaan project akhir.

##### 1. Pengerjaan *AI Project Cycle*

Permasalahan yang timbul adalah Mahasiswa yang kesulitan dalam melakukan presensi yang diharuskan dengan menggunakan 2 devices (penggunaan aplikasi Igracias). Pada permasalahan ini pihak yang terlibat yaitu Mahasiswa, Dosen dan Pihak Administrasi Kampus. Keterlibatan mereka dalam mencatat kehadiran mahasiswa harus melakukan absensi yang nanti di catat ke data kampus dan sebagai bahan penilaian dosen.



Gambar 3.1 Proses Absensi yang menggunakan 2 devices

Penyelesaian dari permasalahan tersebut nanti pada pengembangan sistem absensi dilakukan dengan menscan KTM (Kartu Tanda Mahasiswa) Pendeteksi ini akan menggunakan AI lalu yang pada aplikasinya sudah terdapat jadwal mata kuliah. Pada aplikasi sudah terdapat menu “*Take a picture*” dan “*Launch Galery*”, kemudian kita pilih salah satu pada menu tersebut. Setelah itu terdapat fitur “*scan KTM*”, kemudian arahkan kamera tepat pada KTM sampai foto wajah kalian yang terdapat di KTM bisa terdeteksi.

Dataset yang kami pilih adalah dataset KTM Mahasiswa yang didalamnya terdapat identitas mahasiswa (foto, nama, NIM) Data set yang digunakan adalah KTM. Dataset ini bisa didapatkan di Kampus memperoleh data tersebut atas persetujuan pihak yang memiliki KTM terkait ketersediaannya. Hasil pengumpulan dataset didapatkan 11 class dengan total jumlah dataset sebanyak 110.

Pembuatan modeling menggunakan arsitektur CNN, dengan menggunakan Metode evaluasi menggunakan metode akurasi serta *matrix confusion* untuk menentukan *recall* dan presisi. *Deployment* ini akan diberikan kepada Android dengan menggunakan server.

## 2. Pembagian Tugas

Pembagian tugas dari kelompok kami sebagai berikut:

Tabel 3.1 Pembagian Tugas

<b>Nama</b>	<b>Tugas</b>
Rizki Dwi Putranto	Data set ( <i>Data Acquisition</i> )
Irga	Back-end ( <i>Modeling CNN</i> )

Tennov Caesar Alwi	Back-end (Modeling CNN)
Nur Badriatun Muntamah	Front-end (Dessain UI/UX dan <i>Deployment</i> )
Rizky Satrya Nugraha	Front-end (Dessain UI/UX dan <i>Deployment</i> )

### 3. Pengumpulan Data set

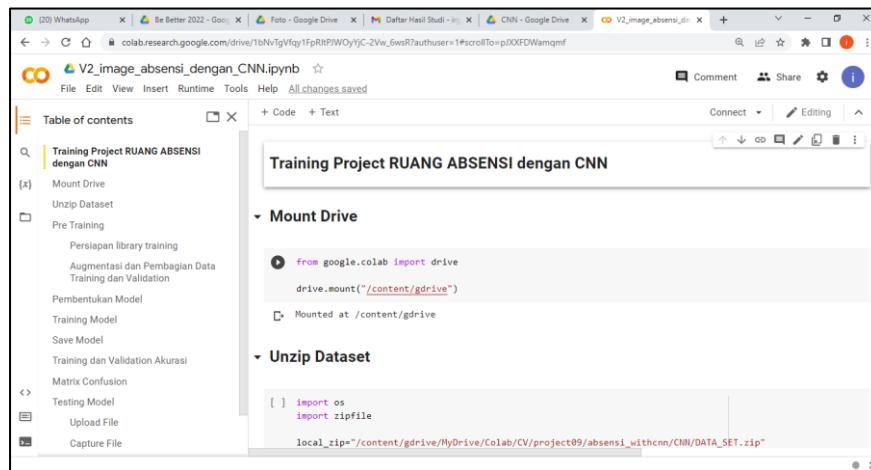
Pengumpulan data set dilakukan kepada mahasiswa Institut Teknologi Telkom Purwokerto yang telah memberikan ijin dan diperoleh sebanyak 11 *class* dengan total data set sebesar 110.

### 4. Labeling

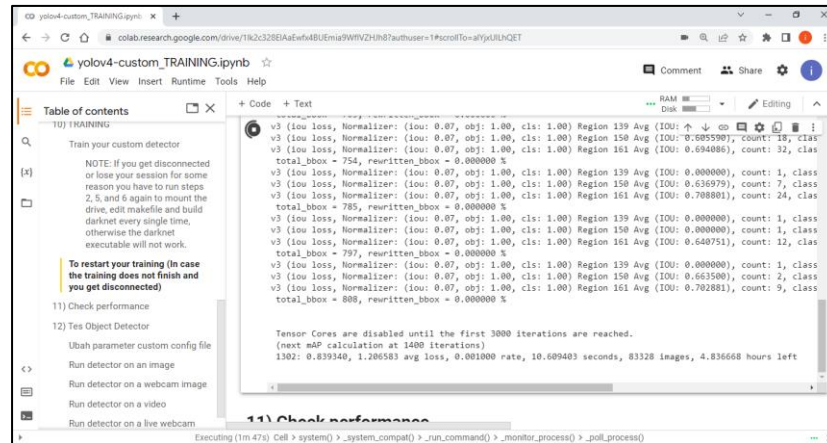
Labeling merupakan tahapan untuk menentukan koordinat citra yang akan di deteksi atau dengan kata lain melakukan persiapan untuk klasifikasi dan lokalisasi image, pada tahapan pemrograman di lakukan di *jupyternotebook* dengan menggunakan *library labelling*, setiap objek yang akan di deteksi akan di *bounding* dan beri label.

### 5. Training dataset

Proses training data set yang dilakukan pada arsitektur YoloV4 membutuhkan waktu yang lama untuk proses tersebut dilakukan secara bergantian dengan kelompok PA setiap harinya. Sedangkan untuk CNN dapat dilakukan dilakukan dalam waktu sehari.



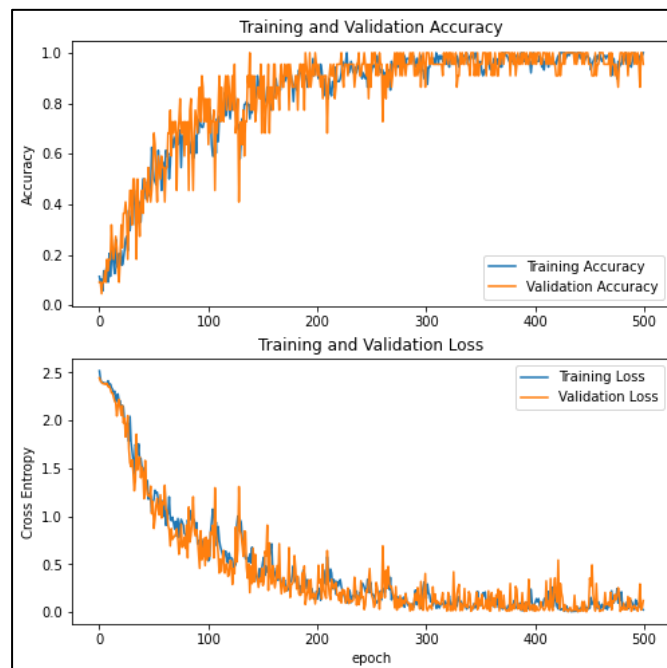
Gambar 3.2 Proses Pembuatan Training Dataset dengan Arsitektur CNN



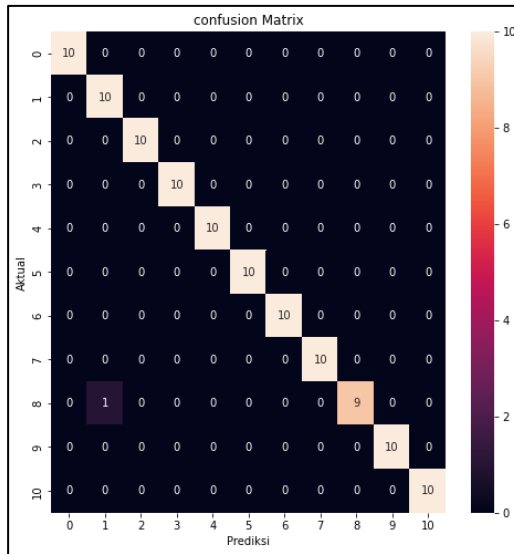
Gambar 3.3 Proses Training Data Set Menggunakan CNN

## 6. Evaluasi

Evaluasi Dilakukan untuk menentukan performa model setelah dilatih. Evaluasi yang tinjau pada YoloV4 meliputi akurasi presisi recall, ketiga parameter tersebut berasal dari nilai *True Positive (TP)*, *False Positive (FP)*, *False Negative (FN)*. Hasil ini ditunjukkan berupa teks deskripsi. Kemudian Pada CNN referentansi akurasi presisi *recall* ditunjukkan dalam bentuk grafik dan *heatmap confusion matrix*.



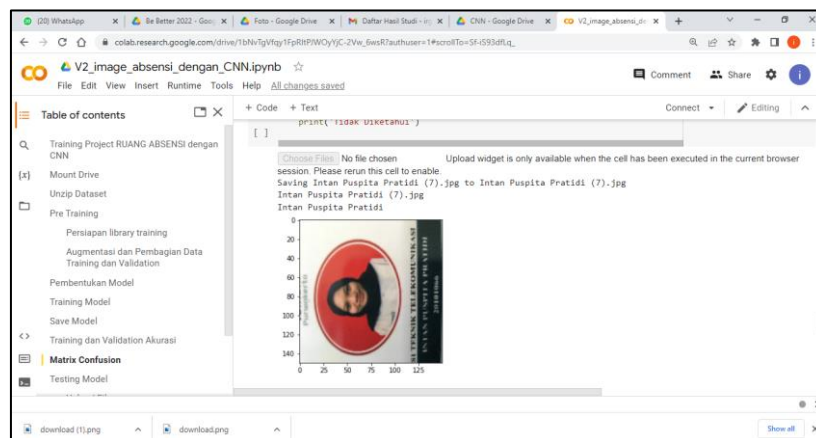
Gambar 3.4 Grafik Training Loss Dan Loss Akurasi Model CNN



Gambar 3.5 Matrix Confusion Model CNN

### 7. Melakukan Testing Training Dataset dan Save Model

Model YoloV4 dan model CNN dapat di test pada IDE, testing dilakukan untuk memeriksa apakah model sudah sesuai dengan apa yang kita inginkan. Setelah di dapatkan model terbaik model disimpan yang nanti akan dilanjutkan ke tahap deployment, untuk model save pada CNN berextension h5 dan yolov4 berextension weight. Converting model hal ini dilakukan agar model dapat dijalankan sesuai dengan deployment yang kita mau. Aplikasi ini akan berjalan pada android maka model di convert ke tflite.



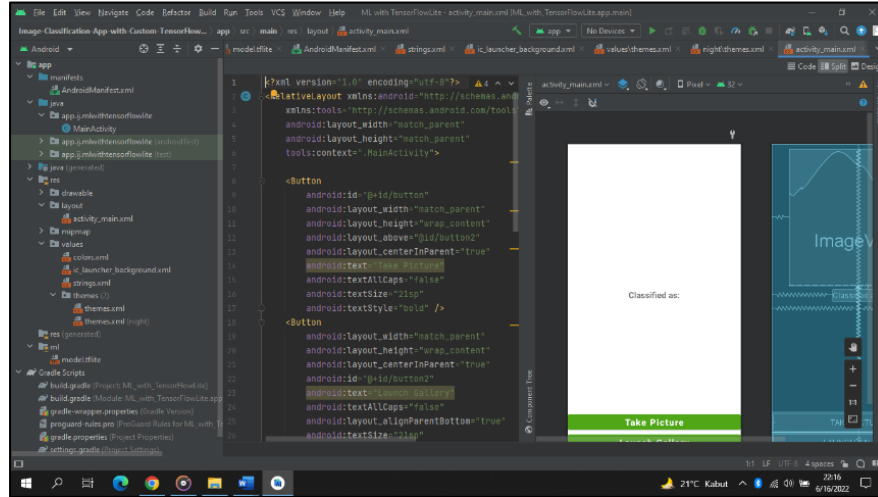
Gambar 3.6 Hasil dari Testing Model dengan CNN

### 8. Deployment

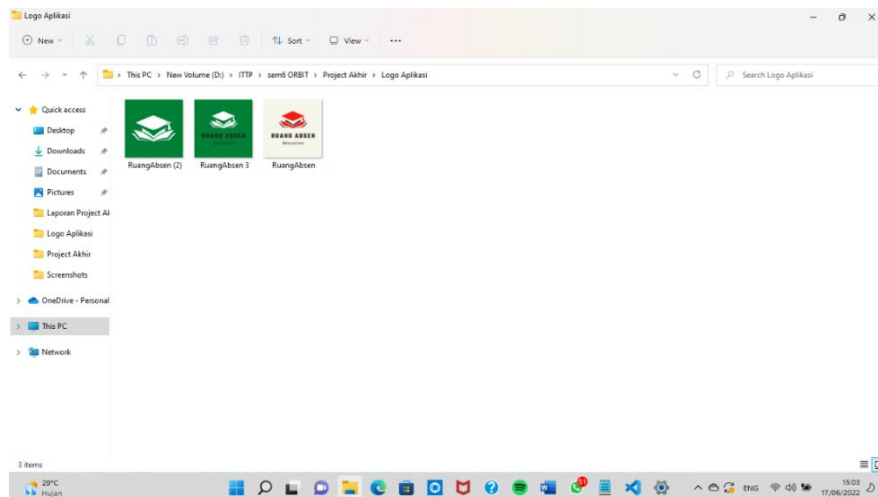
Deployment model pada tahapan ini model di deployment ke android, proses deployment menggunakan android studio deployment ini sebagai tahapan untuk mengetahui apakah mudah dapat digunakan pada aplikasi android.

## 9. Desain UI/UX

Tahapan melakukan desain mengenai *user interface* seperti tata letak *tools* dan pembuatan icon atau logo aplikasi.



Gambar 3.7 Proses Desain UI/UX pada Android Studio



Gambar 3.8 Pembuatan Logo Aplikasi

## 10. Hasil Parameter

Parameter pada project akhir ini menggunakan akurasi dan *matrix confusin* dari pengujian dua arsitektur didapatkan sebagai berikut pada model yoloV4 hasil akhir bahwa model cenderung ke recall dengan akurasi berada pada 0.77 dengan rincian nilai recall 1.00 dan presisi 0.77, serta FP 10, FP 3, dan FN 0. hasil yang di peroleh ini didapatkan pada training data ke 1300.

```

class_id = 5, name = Rafi Naufal Rizqi, ap = 100.00% (TP = 1, FP = 0)
class_id = 6, name = Raufina Rian Aurelia, ap = 0.00% (TP = 0, FP = 0)
class_id = 7, name = Amara Chairunissa, ap = 100.00% (TP = 1, FP = 0)
class_id = 8, name = Rizal Ardiansyah, ap = 100.00% (TP = 1, FP = 0)
class_id = 9, name = Rizki Dwi Putranto, ap = 100.00% (TP = 1, FP = 1)
class_id = 10, name = Rizky Satrya Nugraha, ap = 100.00% (TP = 1, FP = 1)

for conf_thresh = 0.25, precision = 0.77, recall = 1.00, F1-score = 0.87
for conf_thresh = 0.25, TP = 10, FP = 3, FN = 0, average IoU = 51.82 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.863636, or 86.36 %
Total Detection Time: 8 Seconds

Set -points flag:
'-points 101' for MS COCO
'-points 11' for PascalVOC 2007 (uncomment 'difficult' in voc.data)
'-points 0' (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
  
```

Gambar 3.9 Hasil Parameter

Parameter dari arsitektur CNN sebagai perbandingan memperoleh nilai akurasi sebesar 0.99, presisi sebesar 0.99 dan recall 0.99. jika dilihat pada gambar 3.9, angka terisi secara diagonal, maka dengan begitu model dikatakan sangat bagus karena tidak ada kecenderungan lebih ke recall atau lebih ke presisi, ditunjukkan juga nilai recall sama dengan nilai presisi.

```

from sklearn.metrics import classification_report
print(classification_report(test_data_generator.classes, y_pred))
  
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.91	1.00	0.95	10
2	1.00	1.00	1.00	10
3	1.00	1.00	1.00	10
4	1.00	1.00	1.00	10
5	1.00	1.00	1.00	10
6	1.00	1.00	1.00	10
7	1.00	1.00	1.00	10
8	1.00	0.90	0.95	10
9	1.00	1.00	1.00	10
10	1.00	1.00	1.00	10
accuracy			0.99	110
macro avg	0.99	0.99	0.99	110
weighted avg	0.99	0.99	0.99	110

Gambar 3.10 Hasil dari Presisi dan recall