

BAB II PROSEDUR KERJA

2.1 Deskripsi Penugasan Kerja

Berikut adalah deskripsi pekerjaan student sebelum pengerjaan Project Akhir :

- a. Mengikuti *pre-test*.
- b. Mengikuti kelas sesi pagi pada pukul 08.00 hingga 11.30 WIB.
- c. Mengikuti kelas sesi siang pada pukul 13.00 hingga 16.30 WIB.
- d. Mengulang materi yang telah disampaikan di kelas sesi pagi dan siang, setelah kelas sesi siang, selama 1 jam (*self-study*).
- e. Mengerjakan latihan individu atau kelompok yang diberikan oleh homeroom atau domain coach saat kelas berlangsung.
- f. Mengerjakan tugas yang diberikan *homeroom* atau *domain coach* hingga batas waktu tertentu.
- g. Mengerjakan *mini project* yang diberikan homeroom atau domain coach hingga batas waktu tertentu
- h. Mengikuti *post-test*.

Mahasiswa dapat mempelajari setiap materi yang disajikan LMS Orbit Future Academy – Orbit Guru. Dalam pembelajaran MSIB Orbit Future Academy mahasiswa mempelajari lebih detail terkait materi yang dipelajari sebagai berikut.

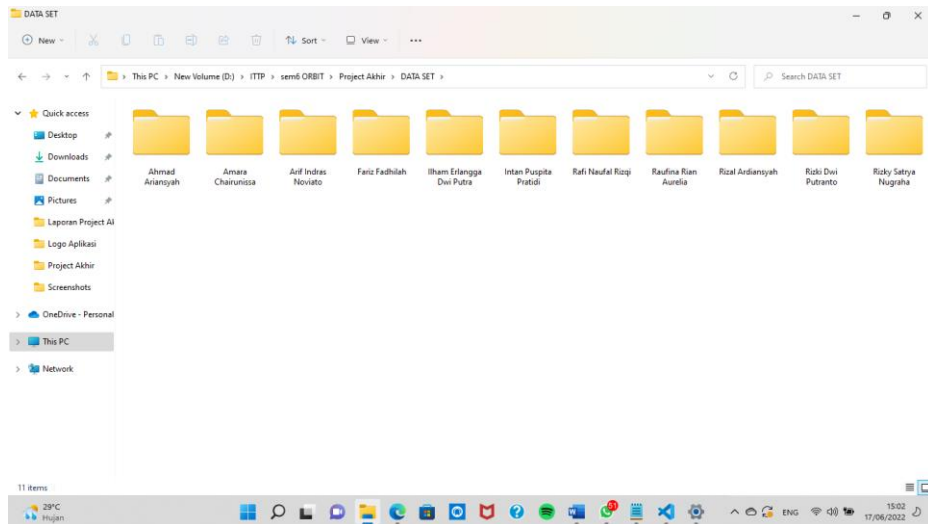
Pada saat pembuatan project akhir student memiliki peran *Data Acquisition* dan *Exploration* selama pengerjaan PA, dengan deskripsi pekerjaan sebagai berikut:

- a. Mencari dataset berupa KTM (Kartu Tanda Mahasiswa)
Pencarian datasetnya dengan cara kita meminta izin ke teman untuk meminjam KTM yang akan digunakan untuk tugas project akhir.



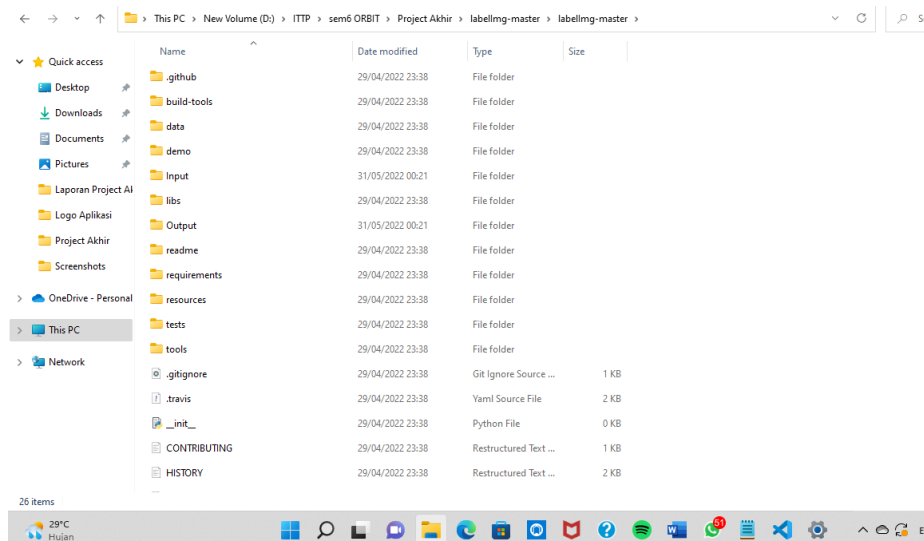
Gambar 2.1 Contoh Data Set berupa KTM

- b. Pengumpulan Data set yang dijadikan dalam satu file
 Pengumpulan data set berupa ktm dikelompokkan sesuai dengan nama masing masing KTM mahasiswa.



Gambar 2.2 Pengumpulan File Data set

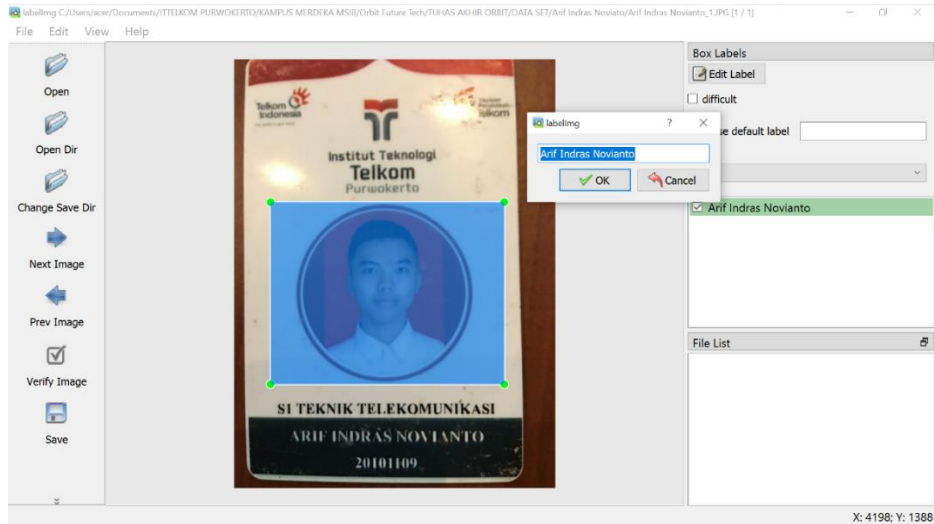
- c. Menentukan pembagian presentase varian
 Pembagian presentase varian bisa berupa *gender*, hijab atau tidak, dan rambut panjang atau pendek.
- d. Mempersiapkan file file untuk melakukan *labelling*
 File file yang akan digunakan untuk labelling disiapkan terlebih dahulu dan juga mempersiapkan data set berupa KTM untuk diproses ke *labelling*.



Gambar 2.3 Proses Pengumpulan File-File yang Diperlukan pada *Labelling*

e. Melakukan *labelling* pada data setnya

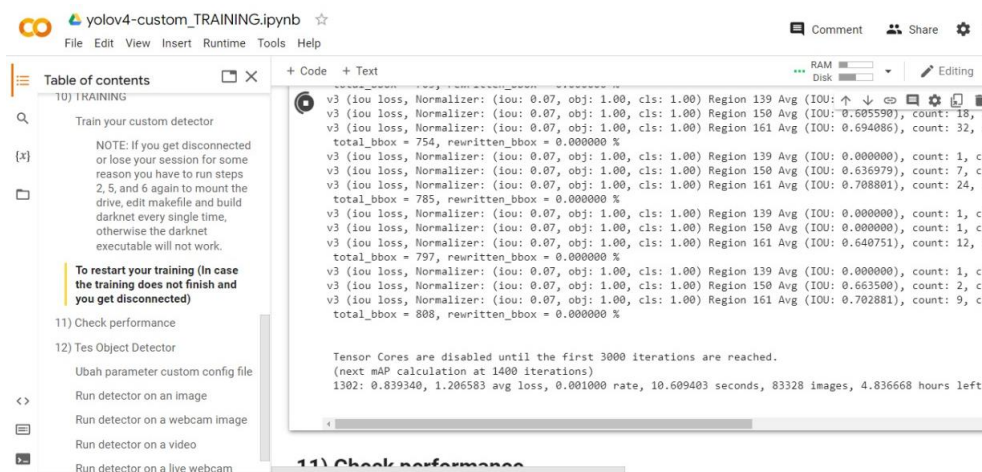
Proses labeling merupakan tahapan untuk menentukan koordinat citra yang akan di deteksi atau dengan kata lain melakukan persiapan untuk klasifikasi dan lokalisasi *image*, pada tahapan pemrograman di lakukan di *jupyternotebook* dengan menggunakan *library labelling*, setiap objek yang akan di deteksi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



Gambar 2.4 Proses *Bounding box* dan *Labeling Image*

f. Melakukan training data dengan menggunakan *google collab*

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



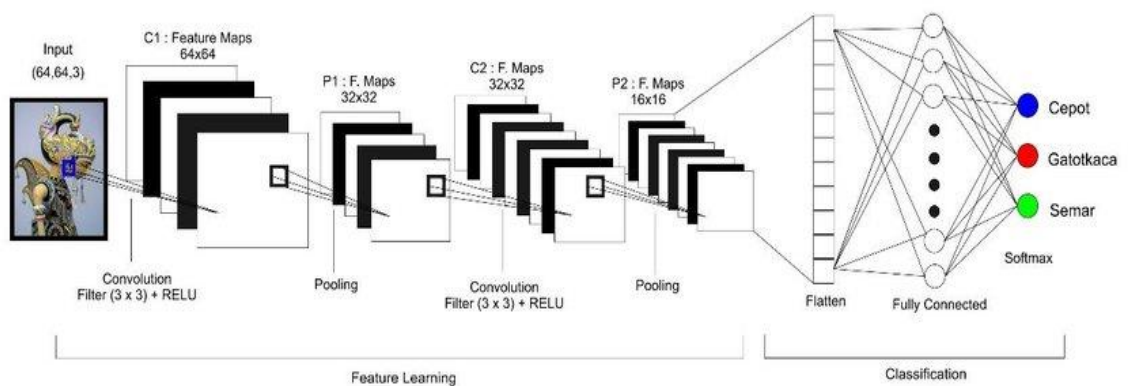
Gambar 2.5 Proses Training Data Set Menggunakan CNN

2.2 Teori Dasar Pendukung

2.2.1 Convolutional Neural Network

Metode CNN atau Convolutional Neural Network merupakan metode fitur hybrid yang memanfaatkan kamera *real-time*. CNN ini juga merupakan metode yang saat ini sedang populer dalam bidang *computer vision* dan nilai akurasi yang dimilikinya telah banyak diakui oleh peneliti di bidang *computer vision*. Gambar yang diambil merupakan gambar mentah berupa gambar wajah dan latar belakang. Dalam metode ini dibuat menggunakan kombinasi deteksi kerangka wajah dan model kerangka dalam pengenalan wajah [2].

Secara umum tipe layer CNN terdiri dari feature learning/extraction dan classification layer. Dimana *feature extraction* terdiri dari *convolution layer*, activation dan *pooling layer*. Sementara *classification layer* terdiri dari *flatten*, *fully connected layer* dan *output layer*. CNN memiliki kinerja yang sangat baik dalam mengatasi masalah pembelajaran mesin (*machine learning*) dan *deep learning*. Khususnya aplikasi yang berhubungan dengan data gambar, seperti kumpulan data klasifikasi gambar terbesar (Image Net), *computer vision* dan *natural language processing*. Arsitektur CNN dapat dikatakan mirip dengan pola hubungan neuron atau sel saraf pada otak manusia. CNN terinspirasi dari *Visual Cortex*, yaitu bagian pada otak yang bertugas untuk memproses informasi dalam bentuk visual [3].



Gambar 2.6 Contoh arsitektur *Convolutional Neural Network*

a. *Convolutional Layer*

Lapisan konvolusi adalah lapisan untuk proses manipulasi gambar agar menghasilkan gambar baru. Lapisan-lapisan pada CNN bekerja secara hierarki, artinya *output* dari lapisan pertama akan digunakan sebagai input pada lapisan selanjutnya. Tujuan utama dari konvolusi adalah untuk mengekstrak fitur dari citra masukan. Konvolusi mempertahankan hubungan spasial antara input dengan mempelajari fitur input [4].

b. *Activation Function*

Sebelum memasuki ke proses pooling, maka harus menggunakan fungsi aktivasi, fungsi aktivasi bertujuan untuk menentukan apakah sebuah neuron harus aktif atau tidak berdasarkan dari weighted sum dari input. Fungsi aktivasi yang digunakan pada penelitian ini yaitu ReLU (*Rectifier Linear Unit*), karena ReLU memiliki rentang nilai dari 0 sampai tak hingga. Nilai yang ada pada hasil konvolusi yang bernilai negative akan diubah dengan aktivasi ini menjadi nol, sehingga dengan menggunakan ReLU, hasil akhir dari matriks yang didapatkan adalah tetap karena nilai pada elemen matriks tidak ada yang bernilai kurang dari nol [4].

c. *Pooling Layer*

Pooling layer merupakan salah satu teknik untuk melakukan proses pencarian fitur yang sudah didapat di tahap sebelumnya (*Convolution*). *Pooling* bekerja dengan mengurangi ukuran matriks menggunakan operasi pooling (penggabungan). Dari hasil *convolution layer* pada lapisan sebelumnya, teknik pooling ini akan mencari fitur dengan tetap mempertahankan fleksibilitas yang tinggi. Ia tidak peduli apakah gambarnya miring ke kanan, kiri, diputar, dibesarkan, dkecilkan, berbeda teksturnya, warnanya dan lain sebagainya. Ia hanya peduli dengan fitur yang dicari [4].

d. *Flatten Layer*

Flatten merupakan tahapan yang lebih sederhana jika dibanding dengan *convolution* dan *pooling layer*. Tahapan *flattening* bekerja dengan mengubah dari matriks yang ada di *pooling layer* menjadi satu kolom saja (sebuah vektor tunggal). Dimana vektor ini nantinya akan menjadi bagian

dari input layer di lapisan *fully connected layer* atau ANN (*Artificial Neural Network*) [3].

e. *Fully Connected Layer*

Pada lapisan ini, hasil dari proses *flattening* dimasukkan ke dalam struktur ANN (*artificial neural networks*) yang utuh. Seperti biasa, ANN terdiri dari 3 bagian, yaitu input layer, hidden layer, dan output layer. Hal yang perlu diperhatikan adalah di dalam model deep learning ANN (*artificial neural networks*) bahwa setiap *neuron (nodes)* tidaklah harus saling terhubung dengan nodes di depannya. Dalam konteks CNN, maka semua nodes dari ANN harus terhubung dengan nodes di depan dan di belakangnya, oleh karena itu disebut dengan *Fully connected* [3].

f. *Output Layer*

Output layer merupakan lapisan terakhir pada CNN maupun model deep learning lainnya. Pada lapisan ini banyak metode ataupun algoritma yang bisa digunakan diantaranya *binary*, *sigmoid*, *softmax* dan lain-lain. Pada penelitian ini, peneliti menggunakan fungsi aktivasi softmax sebagai algoritma outputnya karena jumlah kelas yang lebih dari dua (*multi-class*). Fungsi *softmax* yaitu untuk menghitung probabilitas pada semua label untuk semua kelas. Probabilitas tertinggi dari suatu label itu nantinya yang akan ditampilkan sebagai output pada saat melakukan prediksi [4].

2.2.2 Definisi Citra

Citra (*image*) adalah gambar pada bidang dua dimensi yang disusun oleh banyak piksel, dimana piksel (*pixel*) merupakan singkatan dari *Picture Elements*, dan dapat diartikan sebagai ribuan hingga jutaan titik kecil yang menyusun sebuah foto digital. Setiap piksel membawa informasi yang menentukan warna, kekuatan warna tersebut (*saturation*), dan seberapa terang warna tersebut ditampilkan (*brightness*) sehingga dapat dikatakan bahwa piksel bagian terkecil dari citra. Pada umumnya, citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal antara piksel sama pada seluruh bagian citra. Sebuah gambar disebut dengan citra digital bila gambar yang diperoleh dari hasil proses pada kamera, komputer, mesin scan, atau perangkat elektronik lainnya.

2.2.3 Bahasa Pemrograman *Python*

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* merupakan salah satu bahasa pemrograman tingkat tinggi (*high level language*). *Python* dirancang untuk memberikan kemudahan bagi programmer melalui segi efisiensi waktu, kemudahan dalam pengembangan dan kompatibilitas dengan sistem. *Python* bisa digunakan untuk membuat aplikasi standalone (berdiri sendiri) dan pemrograman script (*scripting programming*) [5].

2.2.4 *Google Colab*

Google Colab atau *Google Colaboratory*, adalah sebuah *executable document* yang dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah ditulis melalui *Google Drive*. Software ini pada dasarnya serupa dengan *Jupyter Notebook* gratis berbentuk cloud yang dijalankan menggunakan browser, seperti *Mozilla Firefox* dan *Google Chrome*. Semua akan diserahkan ke *cloud*. Kita dapat juga menggunakan berbagai fungsionalitas yang dimiliki oleh *Python* serta memanfaatkan *built-in library* yang disediakan oleh *Colab* [6].

2.2.5 *Android Studio*

Android Studio yang merupakan *Integrated Development Environment* (IDE) untuk pengembangan aplikasi *Android*, berdasarkan *IntelliJ IDEA*. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, *Android Studio* menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi *Android*, misalnya sistem versi berbasis *Gradle* yang fleksibel, *emulator* yang cepat dan kaya fitur, lingkungan yang menyatu untuk pengembangan bagi semua perangkat *Android*, *instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat *APK* baru, template kode dan *integrasi GitHub* untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh, alat

pengujian dan kerangka kerja yang ekstensif, alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain [7].

2.2.6 Open CV (Computer Vision)

Open CV (*Open Computer Vision*) merupakan sebuah pustaka (*library*) perangkat lunak *open source* yang digunakan untuk pengolahan citra (*image processing*) dinamis secara *real-time*, yang dibuat oleh Intel. Tujuan diciptakannya Open CV yaitu sebagai *tool* yang dibutuhkan untuk mengatasi masalah yang berkaitan dengan visi computer (*computer vision*). Open CV dapat digunakan pada bahasa pemrograman C, C++ maupun *Python*, serta memiliki fungsi untuk pengolahan citra (*image processing*) tingkat rendah dan algoritma tingkat tinggi [3].