

## BAB III

### METODOLOGI PENELITIAN

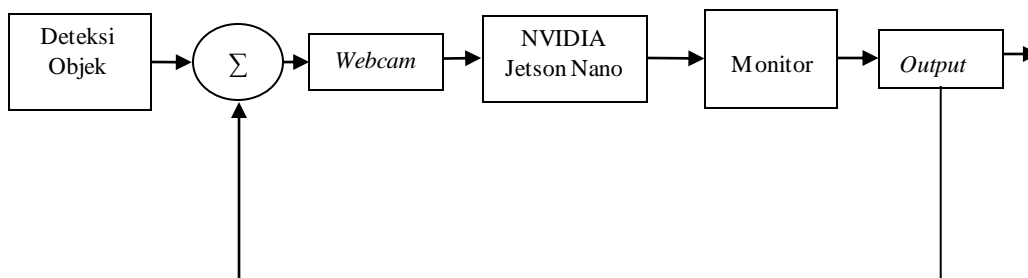
#### 3.1. Alur penelitian

Metode penelitian ini menggunakan NVIDIA Jetson Nano yang di rancang untuk mendeteksi tubuh manusia kemudian menghitung jarak antara tubuh manusia pada saat yang sama. Pada penelitian ini memiliki dua kondisi, yaitu kondisi jarak tidak aman yang memiliki jarak kurang dari 1 meter antara tubuh satu dan lainnya. kondisi yang kedua, yaitu kondisi aman dimana tubuh yang terdeteksi memiliki jarak lebih dari 1 meter dari tubuh yang lainnya. Ada beberapa tahapan yang digunakan untuk menyelesaikan penelitian tersebut yaitu sebagai berikut:

- a. Mendata kebutuhan alat dan bahan yang dipakai untuk perancangan alat deteksi tubuh manusia
- b. Mendesain hardware penelitian agar lebih efisien dan mudah dioperasikan
- c. Merakit hardware yang terdiri dari NVIDIA Jetson Nano, *webcam* dan *Monitor*
- d. Melatih software dengan cara melakukan training pada yolo V4 menggunakan dataset gambar tubuh manusia dan mengukur jaraknya pada Google Collab
- e. Menguji hardware serta software yang sudah dilatih apakah bisa mendeteksi tubuh manusia dan mengukur jaraknya.

#### 3.2. Diagram blok perancangan sistem

Diagram blok alat berikut memberikan penjelasan tentang proses alat pendeteksi *social distancing*. Diagram blok alat dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Diagram blok pemantauan jarak

Dapat dilihat pada Gambar 3.1 diketahui bahwa alat dimulai dari deteksi objek dengan webcam lalu NVIDIA Jetson Nano memproses objek yang dideteksi dan ditampilkan ke monitor.

### 3.3. Komponen penelitian

Komponen yang digunakan pada penelitian ini terdiri dari *hardware* dan *software*. Berikut adalah komponen yang digunakan pada penelitian.

#### a. NVIDIA Jetson Nano

NVIDIA Jetson Nano adalah perangkat tepi komputasi dimana seluruh sistem pengembangan Artificial Intelligence (AI) dapat diimplementasikan. Selain itu, NVIDIA Jetson Nano dilengkapi dengan *library* yang dapat memiliki kemampuan tinggi pada proses pemodelan *Artificial Intelligence* (AI), seperti aplikasi pada deteksi objek, pengenalan gambar, segmentasi, pengenalan suara, dan sebagainya.

Penelitian ini menggunakan NVIDIA Jetson Nano sebagai mini computer yang digunakan untuk memproses data setelah pembacaan jarak antara manusia dengan manusia lainnya. NVIDIA Jetson Nano memiliki bentuk dan ukuran yang kecil, akan tetapi komputer mini ini dapat menghasilkan kinerja komputasi hingga 472 GFLOPS (Proses per detik). Namun daya yang diserap oleh alat ini hanya sebesar 5 watt saja dan itupun sudah cukup untuk menjalankan beban kerja AI modern.

Pada penelitian ini NVIDIA Jetson nano berfungsi sebagai *mikrokontroler* untuk memproses hasil deteksi jarak antara beberapa orang.

#### b. Monitor

*Monitor* merupakan perangkat keras yang dihubungkan ke CPU/GPU sebagai sarana keluaran data grafis. *Monitor* sering digunakan sebagai layar komputer, monitor digunakan sebagai *output* dari video atau tampilan yang dihasilkan oleh CPU/GPU, pada penelitian yang dilakukan menggunakan NVIDIA Jetson Nano sebagai mini PC GPU. Monitor juga disebut dengan istilah VDU (*Visual Display Unit*).



*Gambar 3. 1 Monitor*

Pada gambar 3.3 merupakan *monitor* yang pakai untuk menampilkan hasil dari pengolahan dari NVIDIA Jetson Nano.

c. *Flashdisk 8GB*

Pada penelitian ini *Flashdisk* digunakan untuk menyimpan program yang dijalankan. *Flashdisk* yang digunakan sebesar 8GB.



*Gambar 3. 2 Flashdisk*

Pada gambar 3.4 merupakan *Flashdisk* yang berfungsi sebagai alat untuk memindahkan data antuk di latih pada NVIDIA Jetson Nano.

d. *Micro SD Card 64GB*

Pada penelitian ini *Micro SD* digunakan untuk menjadi tempat OS serta penyimpanan dari NVIDIA Jetson Nano. *Micro SD* yang digunakan sebesar 64GB.



Gambar 3. 3 *Micro SD 64GB*

Gambar 3.5 menggambarkan *memori* yang di pakai untuk menyimpan hasil data latih yang ada pada penelitian.

e. *Webcam NYK A-96*

Pada Penelitian ini *webcam* digunakan untuk mengambil *input* objek. *Webcam* yang digunakan adalah *webcam NYK A-96* dengan resolusi QHD 2k serta memiliki *frame rate* 60 fps.



Gambar 3. 4 *Webcam NYK A-96*

Dapat dilihat pada Gambar 3.6 diketahui bahwa *Webcam* yang digunakan adalah *NYK A-96* dengan resolusi QHD 2k serta memiliki *frame rate* 60 fps.

f. *Komputer*

Pada penelitian ini, komputer digunakan untuk sarana dalam memasukan, mengedit, serta menjalankan program pada penelitian.

*g. Adaptor*

*Adaptor* digunakan sebagai catu daya yang merubah sumber tegangan *AC* yang tinggi yaitu 220V menjadi *DC* sumber tegangan yang rendah dari 1-12V. Pada penelitian ini diperlukan adaptor sebagai sumber tegangan untuk menghidupkan *NVIDIA Jetson Nano*. Sumber tegangan yang diperlukan *NVIDIA Jetson Nano* merupakan tegangan sebesar 5V dan arus 2A untuk menghidupkan komponen *NVIDIA Jetson Nano*, adaptor yang digunakan pada penelitian ini adalah tipe barrel jack yang memiliki tegangan 5V dan arus 4A.



Gambar 3. 5 *Adaptor*

Dapat dilihat pada Gambar 3.7 diketahui bahwa Adaptor digunakan sebagai catu daya yang merubah sumber tegangan *AC* yang tinggi yaitu 220v menjadi *DC* sumber tegangan yang rendah dari 1 s.d. 12v, adaptor yang digunakan pada penelitian ini adalah tipe barrel jack yang memiliki tegangan 5V dan arus 4A.

*h. Kabel HDMI*

Kabel *HDMI* digunakan sebagai penghubung untuk menampilkan *output*. Pada penelitian ini diperlukan kabel *HDMI* untuk menampilkan *output* ke monitor.



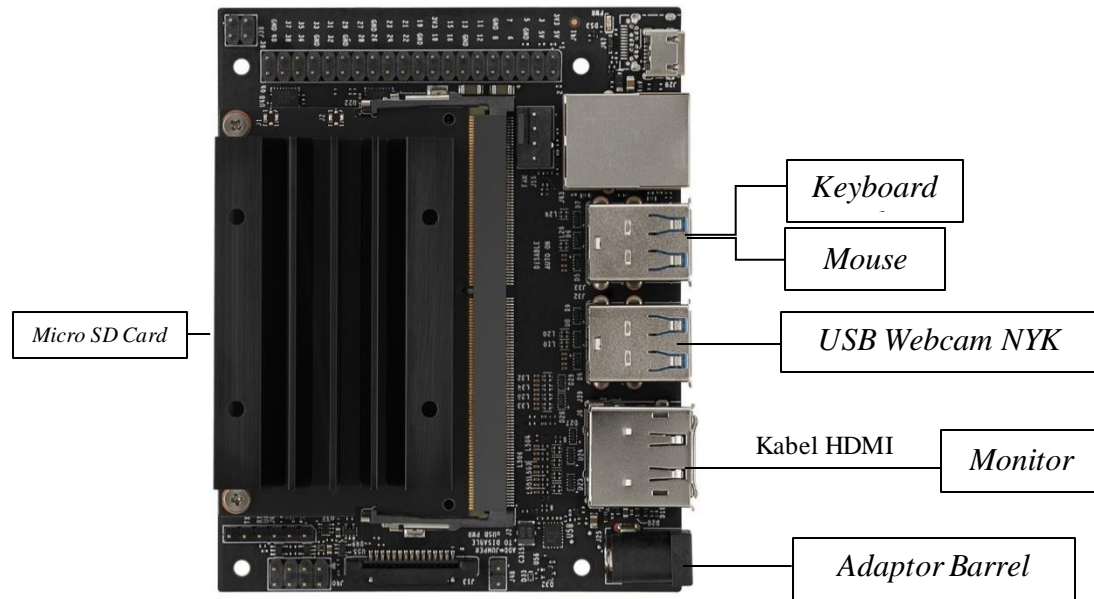
Gambar 3. 6 Kabel HDMI

Dapat dilihat pada Gambar 3.10 diketahui bahwa Kabel HDMI digunakan sebagai penghubung untuk menampilkan output. Kabel HDMI yang digunakan merek vention dengan versi HDMI 2.0 dengan panjang 2 meter dan memiliki konektor male to male.

### 3.1. Perancangan Sistem (*Hardware dan Software*)

Pembuatan perangkat keras (*hardware*) dan tahap perancangan perangkat lunak (*software*). Perancangan perangkat keras yang dimaksud adalah merangkai perangkat keras *NVIDIA Jetson Nano* serta menghubungkannya ke *webcam*, *monitor*, *keyboard* dan *mouse*. Sedangkan perancangan perangkat lunak adalah pembuatan program untuk mengatur perangkat keras.

Pada pembuatan alat ini menjelaskan *port* yang dipakai pada *NVIDIA Jetson Nano*. *NVIDIA Jetson Nano* harus terhubung dengan beberapa komponen agar alat berfungsi dengan baik. Berikut perancangan alat pada *NVIDIA Jetson Nano* yang dipakai dapat dilihat pada gambar 3.9.



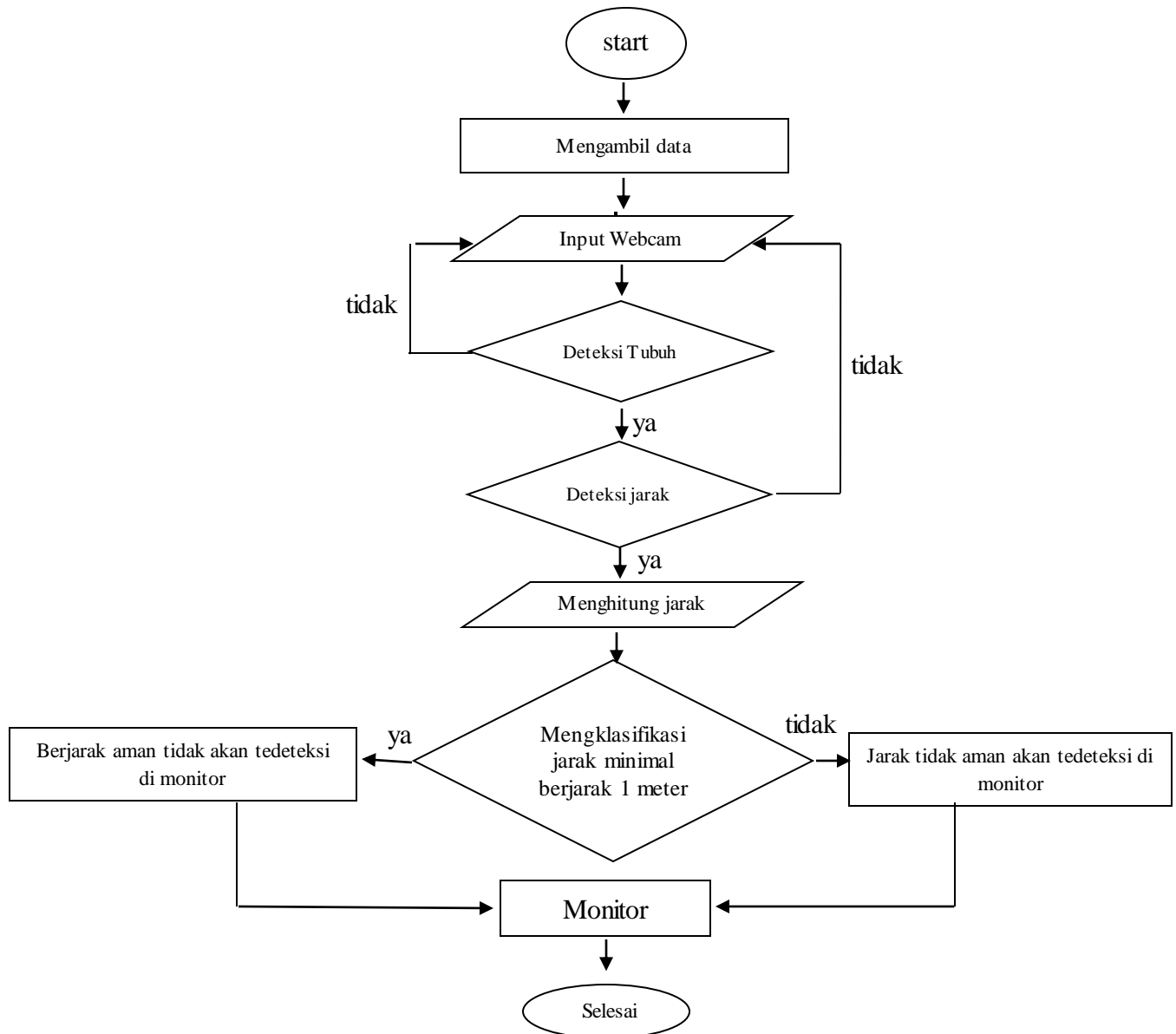
Gambar 3. 7 Perancangan Alat

Pada perakitan perangkat keras dengan menggunakan beberapa alat, antara lain *NVIDIA Jetson Nano* yang mendapat catu daya dari *adaptor* yang di hubungkan dengan kabel *Micro USB*, dan di pasang *sensor* berupa *webcam NYK A-96*, dan memiliki *output* berupa gambar yang di tampilkan pada *monitor* yang di hubungkan dengan menggunakan Kabel *HDMI*.

### 3.4. Pengujian sistem

Pengujian *hardware* meliputi penklasifikasi jarak antara tubuh beberapa orang pada *webcam* yang terpasang pada *NVIDIA Jetson Nano*. Alat akan mendeteksi adanya tubuh manusia dan menghitung jarakaknya. Ada dua jenis kondisi jarak pada percobaan ini, yaitu aman dan tidak aman yang di bedakan melalui jarak yang terdeteksi. Apabila alat mendeteksi jarak pada antara tubuh manusia mencapai 1 meter, alat akan mendeteksi kondisi aman, atau sebaliknya maka alat akan mendeteksi kondisi tidak aman. Jika pada pengujian semua komponen berfungsi sesuai dengan yang diinginkan, maka akan dilanjutkan tahap selanjutnya.

### 3.2. Algoritma Sistem



Gambar 3. 8 Algoritma Sistem

Berdasarkan *flowchart* sistem seperti pada Gambar 3.10 skema cara kerja dari sistem pendeteksi masker wajah dapat dijelaskan sebagai berikut:

- Start* dilakukan dengan menyalakan *NVIDIA Jetson Nano*.
- Kamera sebagai *sensor* akan mengambil data dan mendeteksi dan terus bekerja secara terus-menerus.



- c. *NVIDIA Jetson Nano* akan memproses data dengan mendeteksi tubuh manusia.
- d. Mendeteksi jarak yang ditangkap oleh kamera, apabila jarak antara objek sangat kecil, alat akan mengidentifikasi objeknya hanya satu.
- e. Mengklasifikasi terhadap jarak yang ditangkap oleh kamera, dan memisah antara jarak aman dan tidak menjaga jarak.
- f. Hasil klasifikasi tersebut dapat dibedakan pada monitor dengan tanda yang berbeda yang berjarak kurang dari 1 meter akan pertanda merah pada layar, apabila jaraknya aman atau lebih dari 1 meter akan muncul warna hijau pada layar.
- g. Jumlah pelanggaran akan di tampilkan pada sudut kiri-bawah pada layar.

### **3.5. Klasifikasi Objek**

Klasifikasi objek dilakukan untuk membedakan jenis objek yaitu tubuh manusia, karna itu alat ini hanya mendeteksi tubuh manusai dan tidak akan mendeteksi selain tubuh manusia pada saat pendeteksian. Jika ada objek lain yang masuk pada saat pengambilan gambar, alat tidak akan mendeteksinya. Jenis klasifikasi jarak pada objek ada 2, yaitu kurang dari 1 meter dan lebih atau sama dengan 1 meter.

### **3.6. Instalasi Awal *NVIDIA Jetson Nano***

Hal yang paling pertama dilakukan agar penelitian ini bisa berjalan yaitu *NVIDIA Jetson Nano* harus mempunyai *Operating System* terlebih dahulu agar program yang dibutuhkan bisa berjalan dengan baik. Apabila sudah memiliki *Operating system* maka selanjutnya instalasi library yang dibutuhkan. Langkah instalasi awal *NVIDIA Jetson Nano* yang di butuhkan, yaitu.

#### **3.6.1. Komponen *NVIDIA Jetson Nano***

Komponen yang dibutuhkan untuk *NVIDIA Jetson Nano* ada 2, yaitu:

- a. Penyimpanan: *MicroSD Card* dengan minimal 16 GB penyimpanan, pada penelitian ini menggunakan *MicroSD Card* merek Sandisk yang mempunyai penyimpanan 64GB.

- b. *Adaptor* digunakan sebagai catu daya yang merubah sumber tegangan AC yang tinggi yaitu 220V menjadi DC sumber tegangan yang rendah dari 1-12V.

### 3.6.2. *Instalasi Operating System menggunakan Etcher*

Pada tahap ini *MicroSD* akan di *format* ulang untuk memasukan *Operating System* pada *NVIDIA Jetson Nano*. Tahap ini menggunakan *OS Windows*, berikut ini adalah cara *Instalasi Operating System*:

- a. Mengunduh *Jetson Nano Developer Kit SD Card Image*
- b. Mengunduh, *Menginstal* dan menjalankan aplikasi *Etcher*
- c. Klik “*Select image*” dan pilih *zip Jetson Nano Developer Kit SD Card Image*.
- d. Masukkan *MicroSD Card* jika belum dimasukkan.
- e. Klik “*Select drive*” dan pilih target perangkat yang benar.
- f. Klik “*Flash!*” dan menunggu prosesnya selama 10-15 menit, kecepatan *MicroSD Card* sangat berpengaruh pada saat proses *Flash*.
- g. Setelah proses *Etcher* selesai, keluarkan *MicroSD Card*
- h. *MicroSD Card* sudah siap dipasang di *Jetson Nano*

### 3.6.3. *Setup NVIDIA Jetson Nano*

Pada Tahap ini *NVIDIA Jetson Nano* akan pertama kali dinyalakan. Hal yang diperlukan untuk setup *NVIDIA Jetson Nano* yaitu:

- a. *Monitor*
- b. *Keyboard*
- c. *Mouse*
- d. *Adaptor*

Terdapat beberapa langkah selanjutnya setelah hal yang dibutuhkan sudah tersedia, langkahnya yaitu:

- a. Memasukkan *MicroSD Card* (dengan *system image* sudah didalamnya) ke dalam slot *MicroSD Card* pada modul *Jetson Nano*.
- b. Menyalakan layar *monitor* dan menyambungkannya ke modul *Jetson Nano*.

- c. Menghubungkan *keyboard* dan *mouse* ke modul *Jetson Nano*.
- d. Menghubungkan catu daya tambahan atau *Micro-USB Power Supply*, lalu *Developer Kit*-nya menyala dan *boot* secara otomatis.

#### **3.6.4. Tahap Instalasi Jetson Nano Developer Kit**

Ini merupakan tahap yang penting, karena tahap ini dilakukan segera setelah *Developer Kit*-nya menyala, hal itu ditandai dengan menyalnya LED hijau yang ada disebelah konektor *Micro-USB*. Saat sudah menyala semua, sistem melakukan *boot* untuk pertama kali dan *Developer Kit* melakukan pengaturan awal, diantaranya yaitu:

- a. Meninjau dan menerima EULA (*End-User License Agreement*) perangkat lunak NVIDIA Jetson Nano.
- b. Memilih bahasa sistem, *keyboard layout*, dan zona waktu.
- c. Membuat *username*, *password*, dan nama komputer.
- d. Memilih ukuran partisi, langkah ini disarankan untuk menggunakan ukuran maksimal yang telah disarankan.
- e. Jetson Nano akan masuk ke desktop Linux.
- f. Setelah itu, tahap instalasi Jetson Nano Developer Kit selesai.

#### **3.6.5. Tahap Instalasi Library**

Pada tahap ini yang dilakukan dengan tujuan agar program yang telah dibuat dapat melakukan deteksi tubuh manusia. *Library* yang dibutuhkan yaitu *library cv2*, *numpy*, *math*, *time*, *sys*, *gpio*, *zip unzip* dan *tensorflow*. Masing-masing *library* mempunyai fungsinya, diantaranya yaitu pada *cv2* yang mempunyai fungsi untuk melakukan *computer vision task*, *numpy* yang berfungsi untuk proses komputasi numerik, *math* yang bisa memperluas fungsi matematika, *time* untuk menyediakan modul waktu, *sys* yang digunakan untuk mengakses konfigurasi interpreter pada saat *runtime*, *gpio* untuk menginisiasi pin-pin pada Jetson Nano, *zip unzip* untuk membuka dan mengarsipkan *file* dalam bentuk *zip* dan *tensor flow* yang berfungsi untuk menghitung komputasi numerik secara mudah dan cepat.

### 3.7. Melatih NVIDIA Jetson Nano

NVIDIA Jetson Nano perlu dilatih dahulu agar bisa mengidentifikasi suatu objek. *Dataset* yang digunakan berupa gambar tubuh manusia, dan banyak banyak objek lain yang bukan tubuh manusia untuk pembandingan, agar tidak *error* saat muncul objek yang mirip dengan tubuh manusia. Pada penelitian ini NVIDIA Jetson Nano dilatih dengan 253 gambar yang sudah diberi label dengan anotasi kotak pembatas.

### 3.8. Data Processing

*Dataset* yang digunakan pada penelitian ini berasal dari website *Crowdhuman Dataset Introduce* dengan nama *Crowd Human*. Setiap gambar memiliki *file anotasi* yang sesuai dalam format xml yang menentukan koordinat empat sudut kotak pembatas yang menunjukkan bentuk tubuh manusia yang berbeda-beda, serta kelas yang sesuai untuk setiap kotak. Berikut ini adalah cara melakukan persiapan *Dataset*:

#### 3.8.1. Persiapan *Dataset*

Langkah awal persiapan dataset yaitu unduh *Crowd Human* dari website *Crowdhuman Dataset Introduce* lalu buat *folder dataset* dan *unzip* arsip *dataset* yang sudah di unduh. di dalamnya terdapat dua *folder*, *folder* pertama berisi anotasi dari *dataset* gambar, *folder* kedua berisi gambar yang akan dilatih.

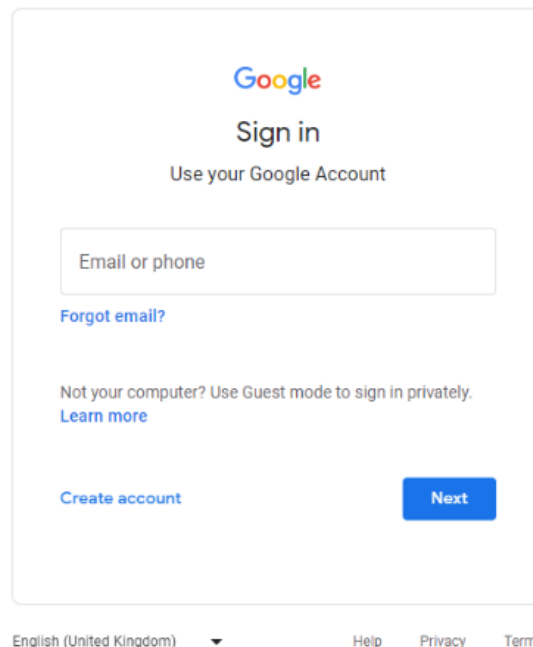
#### 3.8.2. Merubah *format xml* ke *format yolo* pada *Dataset*

Proses perubahan format xml ke yolo dilakukan setelah arsip *dataset* di *unzip*. Tujuan dari perubahan format ini karena *dataset* masih berupa *Pascal VOC* maka dari itu harus dirubah formatnya agar terbaca oleh yolo. Setelah dirubah dataset akan dibagi menjadi 80%-20%, 80% untuk data latih sedangkan 20% untuk data uji.

### 3.8.4. Melatih data dengan Google Collab

Melatih data dilakukan setelah *data processing* selesai. Pada proses *training* data ini, prosesnya dilakukan di *Google Colab* karena terdapat super GPU (*Graphics Processing Unit*), sehingga bisa mempercepat prosesnya. *Google Colab* yang digunakan ialah *Google Colab Pro* yang terbatas dimana mempunyai *benefit* diantaranya dapat mengakses ke GPU (*Graphics Processing Unit*) dan TPU (*Tensor Processing Unit*) yang lebih cepat, memiliki lebih banyak memori RAM dan *disk*, dan mendapatkan waktu proses yang lebih lama sehingga lebih jarang memutuskan sambungan. Terdapat beberapa tahapan dalam melaksanakan proses *training* data dengan menggunakan *Google Colab*, tahapannya yaitu:

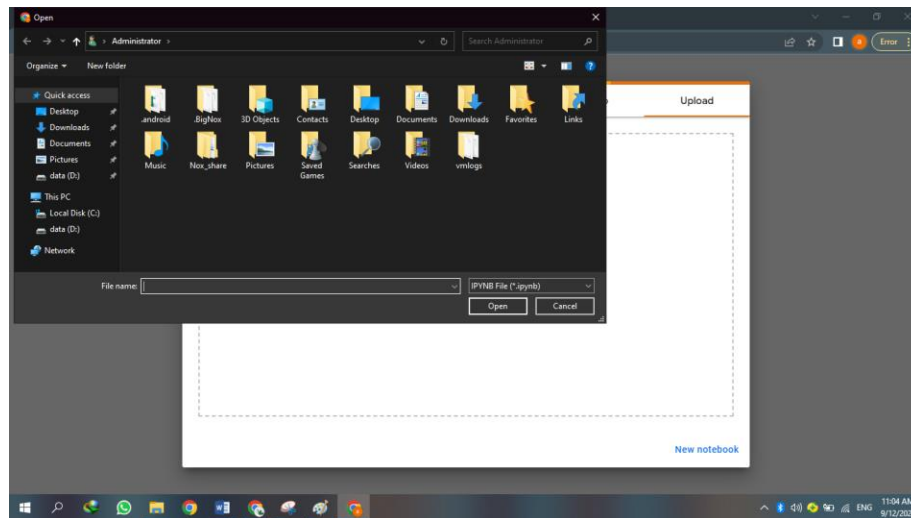
- a. Masuk ke *website* <https://colab.research.google.com/>.
- b. *Login* melalui akun Google, jika belum mempunyai akun Google, maka buat akun terlebih dahulu.



Gambar 3. 9 Tahap Unggah File *Google Collab*

Gambar 3.11 merupakan langkah pertama untuk mengunggah *file* di *google collab*, untuk mengakses *google collab* kita harus masuk menggunakan akun *google*,

- c. Jika sudah unggah file *Google Collab* dengan format ipnya.



Gambar 3. 10 Tahap Unggah File *Google Collab*

Pada gambar 3.12 adalah tahap pemilihan *file* yang akan di unggah pada *Google Collab*, kita harus memilih dari penyimpanan komputer kita.

- d. Setelah *file* berhasil terunggah, maka akan muncul beberapa *listing code* yang akan dieksekusi diantaranya yaitu *mount drive*, ekstrak *file model*, *update* sistem terbaru dan *install library* yang dibutuhkan. Langkah eksekusi tersebut wajib dilakukan sebelum *training data* dimulai.

```

Clone "yolov4_crowdhuman" source code from GitHub.

[ ] %cd /content
    if not Path('yolov4_crowdhuman').is_dir():
        !git clone https://github.com/jkjung-avt/yolov4_crowdhuman.git

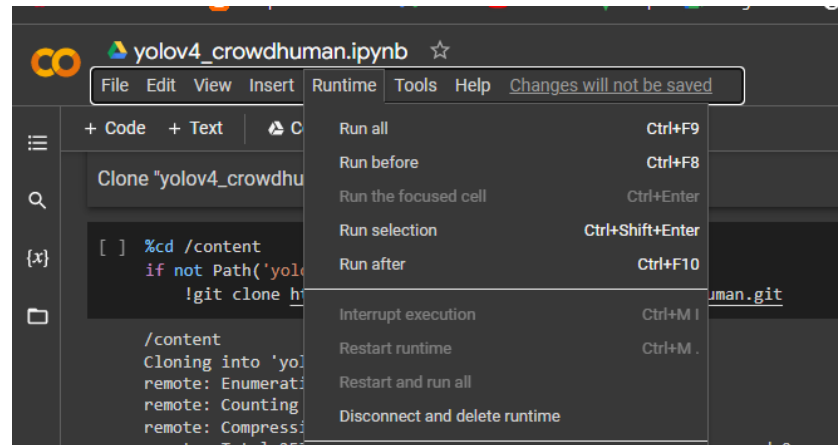
/content
Cloning into 'yolov4_crowdhuman'...
remote: Enumerating objects: 253, done.
remote: Counting objects: 100% (253/253), done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 253 (delta 146), reused 167 (delta 66), pack-reused 0
Receiving objects: 100% (253/253), 2.17 MiB | 30.91 MiB/s, done.
Resolving deltas: 100% (146/146), done.

```

Gambar 3. 11 Isi *File Google Collab* Yang Sudah Diunggah

Dapat dilihat pada gambar 3.13 merupakan hasil dari *file Google Collab* yang sudah terunggah, ini adalah isi *file* yang sudah terunggah.

- e. Jalankan semua script di *Google Collab* dengan cara ke menu *runtime* lalu klik *Run all*.



Gambar 3. 12 Menjalankan Semua *Script* Pada *Google Collab*

Pada gambar 3.14 tergambar proses terakhir yang kita jalankan, yaitu menjalankan semua *script* di *google collab* dengan menekan tombol *run all*.

- f. Jika sudah selesai maka di dalam *google drive* ada *folder backup* di dalam folder *yolov4-mask*, di dalamnya terdapat bobot yang telah dilatih.
- g. Ubah nama *file* *yolov4-mask\_best.weights* ke *yolov4-mask.weights* lalu *download*, *file* ini merupakan *file* yang akan digunakan.

### 3.9. Tempat dan Waktu Penelitian

Tempat penelitian dilakukan di Kontrakan yang berada di Jl. Lembang Raya, Kubang Bale, Citangkil. Waktu Pengerjaan Skripsi ini berlangsung dari bulan Mei 2022.