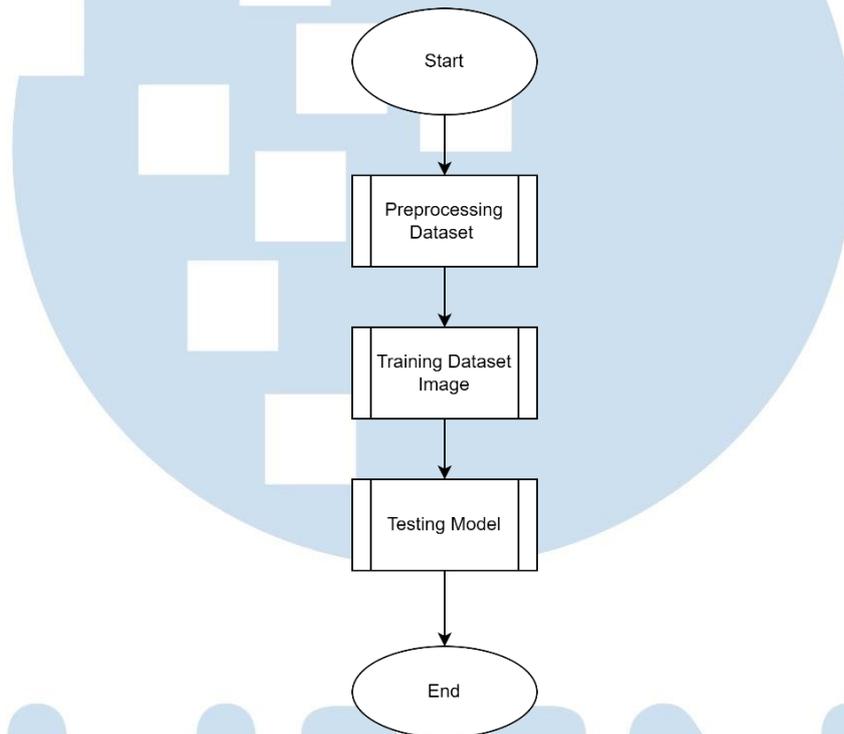


BAB III

ANALISIS DAN PERANCANGAN SISTEM

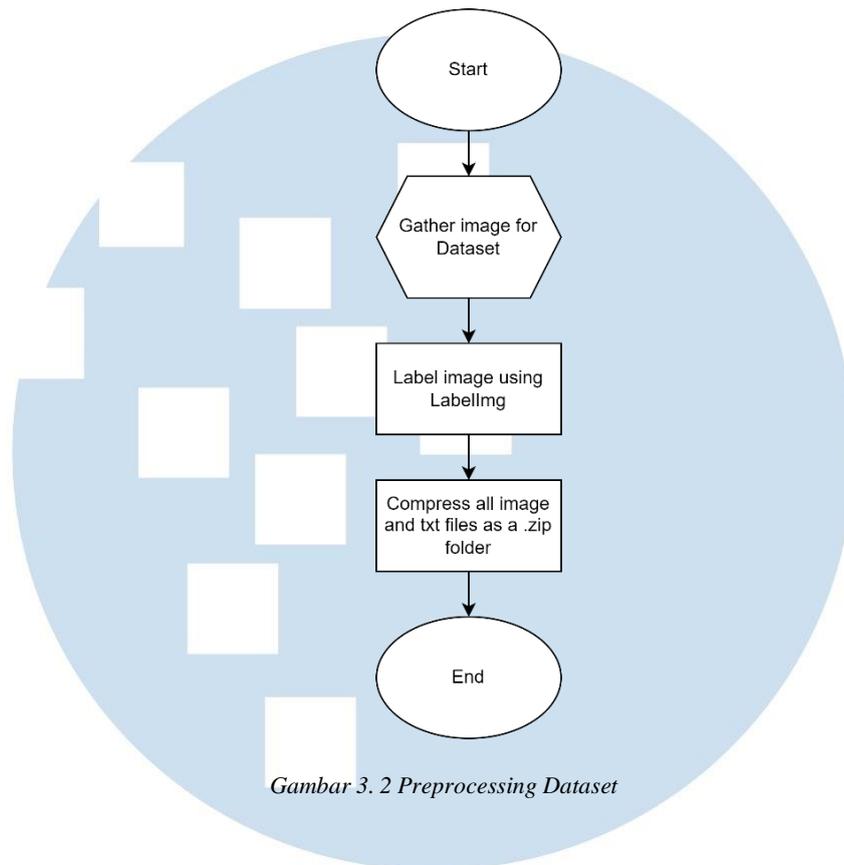
3.1 Perancangan Sistem



Gambar 3.1 Flowchart sistem secara keseluruhan

Gambar 3.1 menunjukkan *flowchart* keseluruhan sistem secara garis besar, yaitu *Preprocessing Dataset*, *Training Dataset Image* dan *Testing model*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.2 Preprocessing Dataset

Gambar 3.2 menunjukkan *flowchart preprocessing dataset*. Pada langkah pertama dilakukan pengumpulan data – data yang berupa gambar dan *install software* yang di perlukan. Dataset berupa bentuk gambar dengan format .jpg. Gambar tersebut terdiri dari gambar mobil ambulans. Jumlah data yang akan di ambil kurang lebih akan berjumlah 1109 gambar. Kemudian dataset tersebut di bagi menjadi data untuk training dan data untuk testing. Pembagian dataset dilakukan secara random. Setiap gambar yang di gunakan untuk training nanti nya akan memiliki kelas “ambulans” setelah di lakukan anotasi gambar.

Setelah data terkumpul selanjutnya proses yang dilakukan adalah anotasi gambar. Anotasi gambar bertujuan untuk memberikan *boundary box* pada gambar dan nama kelas pada setiap masing-masing objek. Pada tahap ini peneliti menggunakan tools LabelImg untuk membantu dalam proses membuat *boundary box* pada gambar, sehingga koordinat suatu gambar dapat diperoleh dengan

mudah. Setelah itu hasil dari anotasi suatu gambar akan di simpan 1 tempat dengan gambar tersebut. Contoh penerapan nya seperti ini.



Gambar 3. 3 Tampilan LabelImg

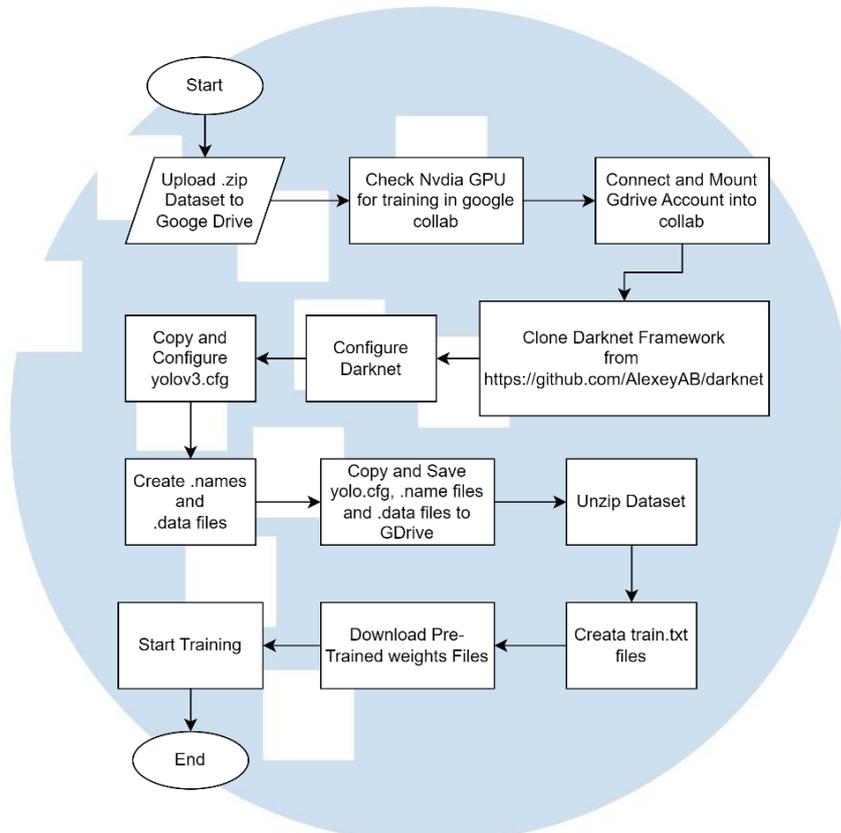
Hasil dari anotasi gambar adalah berupa image dan file .txt yang berisi koordinat bounding box yang sebelumnya dilakukan pada labelimg. Berikut hasil dari anotasi gambar.



Gambar 3. 4 Hasil Anotasi dengan LabelImg

Proses kompres gambar hasil anotasi dilakukan agar dapat di upload di google drive yang akan di gunakan untuk proses training di google collab.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



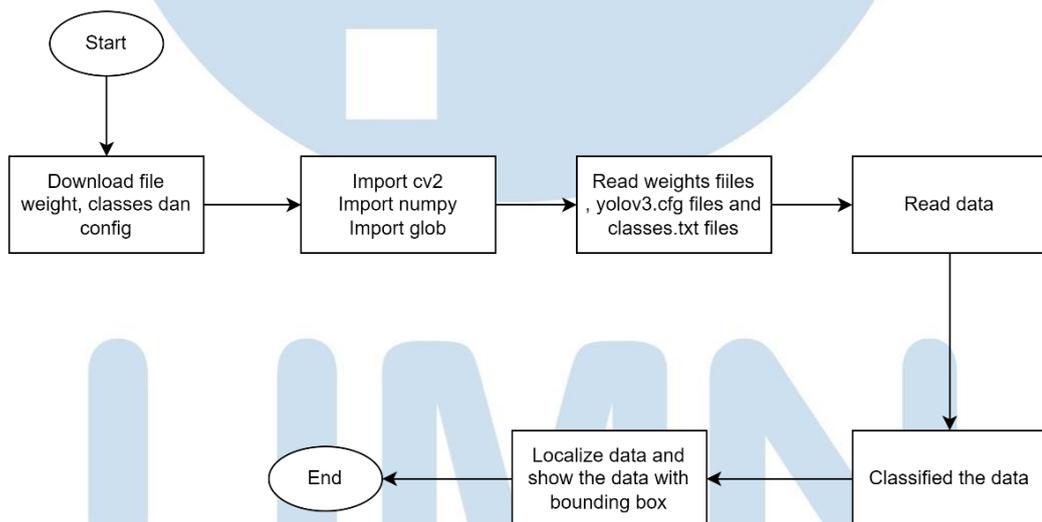
Gambar 3. 5 Flowchart proses training

Setelah melalui tahap preprocessing, tahap selanjut nya adalah proses training model. Gambar 3.5 merupakan flowchart training model. Sebelum training, dilakukan proses upload file .zip ke dalam google drive. Setelah file .zip terupload di google drive, hal yang perlu di lakukan pertama kali adalah memeriksa apakah GPU Nvidia tersedia untuk dilakukan training. Jika tersedia, maka selanjut nya adalah mengintegrasikan Google Drive dengan Google Colab, agar file .zip yang berisi image training dan classes.txt dapat diakses oleh Google Colab. Setelah itu framework Darknet perlu di clone dari github. Setelah darknet ter clone dari github, maka directory darknet akan tersedia di google colab, dan langkah selanjutnya yang perlu di lakukan adalah meng konfigurasi darknet. Setelah itu, hal yang perlu dikonfigurasi adalah file yolo.cfg. Di dalam file tersebut, yang perlu di konfigurasi adalah bagian max batches dan jumlah filter sesuai dengan rumus dibawah ini :

Max Batches = 2000 * n

Filter (n + 5) * 3

Langkah selanjut nya adalah membuat file obj.names dan obj.data yang berisi nama kelas dan jumlah kelas yang akan di gunakan untuk training. Kemudian dilakukan copy file yolov3.cfg dan obj.names yang sudah di buat sebelum nya kedalam google drive. Setelah itu, unzip dataset training yang berada di google drive kedalam google colab dan buat file train.txt. File train.txt akan mengambil lokasi unzipped dataset ketika proses training berlangsung. Selanjut nya adalah mendownload pretrained weight agar proses training tidak perlu di lakukan dari awal. Setelah semua proses tersebut selesai, kemudian dilakukan proses training model. Training model akan di simpan ke dalam google drive.



Gambar 3. 6 Flowchart proses testing

Pada gambar 3.6 merupakan sebuah flowchart program untuk testing berbasis python yang berguna untuk menjalankan object detection. Langkah pertama sebelum menjalankan program tersebut adalah mendownload hasil training, yaitu berupa file .weight, classes dan .cfg. Setelah itu, import library yang di butuhkan, seperti cv2 dan numpy. Kemudian program tersebut akan membaca dan memproses file hasil training yang berupa file .weight, classes dan

.cfg. Setelah itu data input yang berupa foto atau video akan di baca dan di klasifikasikan berdasarkan kelas yang sudah di tentukan. Lalu objek tersebut akan dilokalisasi menggunakan bounding box dan seterusnya sampai objek tersebut di berhentikan.

3.2 Metode Pengujian

Metode pengujian dalam penelitian ini pertama adalah menggunakan confusion matrix yang bertujuan untuk mencari nilai Recall, Precision dan F1Score. Confusion matrixs yang digunakan merupakan matrix berbentuk 2x2 yang berisi kelas ground truth bounding box dan hasil prediksi. Hasil perhitungan confusion matriks akan di gunakan untuk dimasukan kedalam rumus Recall, Precision dan F1 Score. Penulis juga menggunakan Average Precision karena metrix tersebut merupakan metrix yang sering digunakan untuk mengukur performance dari YOLO V3. karena dalam penelitian ini hanya terdapat satu kelas yang di deteksi, sehingga dapat di katakan bahwa Average Precision dan Mean Average Precision sama saja.

