

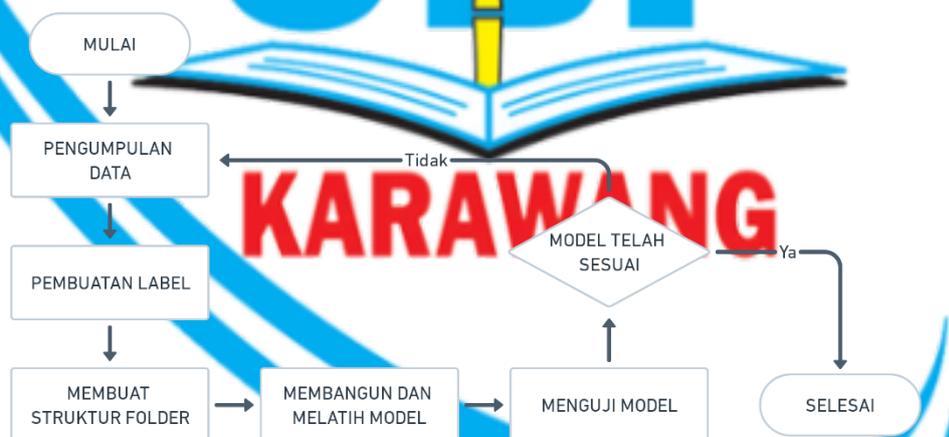
BAB III METODE PENELITIAN

3.1 Objek Penelitian

Gambaran secara umum tentang objek penelitian yang digunakan berupa individu yang memakai masker dan tidak memakai masker. Sesuai dengan protokol kesehatan yang berlaku bahwa masyarakat masih wajib menggunakan masker pada beberapa tempat umum atau terbuka. Kemudian untuk *dataset* yang digunakan berupa kumpulan foto atau citra wajah seseorang menggunakan masker dan tidak menggunakan masker. Objek ini didapatkan dari *website dataset* yaitu *kaggle*.

3.2 Prosedur Penelitian

Penelitian ini memiliki beberapa tahapan sesuai dengan prosedur yang digunakan pada YOLOv5.



Gambar 3. 1. Diagram Alur Prosedur Penelitian

Berdasarkan Gambar 3.1, alur penelitian dimulai dengan pengumpulan data, kemudian pembuatan label dan membuat struktur folder. Tahap selanjutnya adalah membangun dan melatih model, setelah itu menguji model tersebut. Jika model yang diuji belum sesuai maka proses dimulai kembali dari pengumpulan data.

3.2.1 Pengumpulan Data

Pada proses pengumpulan data, yang dikerjakan adalah melakukan pengumpulan data berupa citra atau gambar wajah manusia yang dibagi dalam 2 kategori atau kelas. Kelas pertama citra wajah menggunakan masker dan kelas kedua citra wajah tanpa menggunakan masker. Citra yang dikumpulkan didapatkan dari *website* Kaggle.

3.2.2 Pembuatan Label

Pembuatan label yang dapat disebut juga anotasi merupakan proses pemberian informasi pada objek. Proses pembuatan label dilakukan pada tiap objek yang ada dalam gambar yang telah dikumpulkan. Seluruh gambar yang sudah terkumpul sebelumnya diberikan label sebagai identitas agar dapat memuat nama gambar. Proses pelabelan dapat dilihat pada Gambar 3.2, dilakukan dengan cara membuat nama kelas dan *bounding box* pada tiap-tiap objek gambar. Kelas yang dibuat pada penelitian ini adalah 'mask' dan 'nomask'. Pemberian label pada data gambar yang nantinya dipakai untuk *training* pada penelitian dilakukan dengan bantuan *web application* yaitu *makesense.ai* agar mempercepat proses pelabelan dan menjadikannya dataset yang siap digunakan di YOLOv5. Setelah seluruh gambar selesai diberi label, export hasil labeling tadi dengan pilihan *export annotation* dan pilih YOLO format, nantinya hasil labeling akan berbentuk text dengan format .txt.



Gambar 3. 2. Proses Labeling

3.2.3 Pembuatan Struktur Folder

Pembuatan struktur folder dilakukan untuk mengelompokkan data. Data gambar dan hasil labelling sebelumnya disusun serta dibagi kedalam struktur folder YOLO. Tujuannya untuk kebutuhan training dan validasi. Data gambar dimasukkan pada folder *images* dan data label dimasukkan pada folder dengan nama *labels*. Folder yang sudah dibuat juga sudah berisikan data yang nantinya akan digunakan selanjutnya diubah kedalam bentuk .zip untuk memudahkan ketika diunggah pada *Google Colab*.

3.2.4 Membangun dan Melatih Model

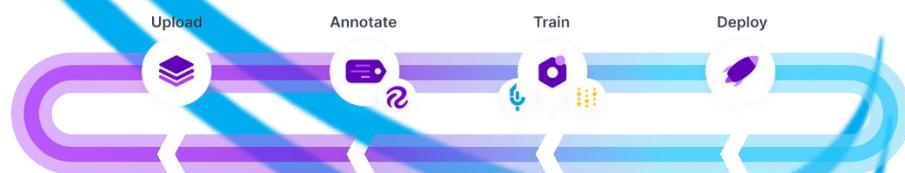
Data yang sudah diberi label pada tahap sebelumnya akan dilatih untuk membentuk suatu pola dengan *output* berbentuk Bobot. Bobot tersebut kemudian digunakan dalam mendeteksi objek di dalam gambar. *Training* akan dilakukan dengan *You Only Look Once* (YOLO) dimana metode ini berbasis CNN. Pada bagian ini gambar yang telah dimasukkan selanjutnya dilakukan proses CNN yang meliputi pemecahan *citra*, *weights sharing*, pembentukan hasil *array*, dan *downsampling*. Proses pembangunan dan pelatihan model dilakukan dengan bantuan *Google Colaboratory* atau yang sering dikenal sebagai *Google Colab*. *Training* data dilakukan pada *Google Colab* karena diberikannya GPU sebesar 12 GB dengan bantuan nvidia yang dapat membuat tahap *training* data dapat dilakukan lebih cepat. Tahap pertama yang dilakukan adalah melakukan *cloning repository* YOLOv5 dari github, kemudian *install requirements* yang dibutuhkan. Selanjutnya unggah data yang telah disiapkan sebelumnya dalam bentuk .zip pada *Google Colab* dan lakukan proses unzip. Tahap berikutnya adalah mengubah file *coco128.yaml* yang ada pada direktori YOLOv5 untuk mengatur jumlah kelas dan nama kelas yang telah dibuat sebelumnya agar sesuai. Kemudian terakhir melakukan proses pelatihan dengan menjalankan fungsi *train* pada YOLOv5. Pada proses pelatihan ini dipilih varian model YOLOv5 yang digunakan. Terdapat

5 varian model pada YOLOv5, pada penelitian ini model yang akan digunakan merupakan varian model YOLOv5s. Model YOLOv5s memiliki parameter yang ringan sehingga deteksi dan proses komputasi dapat dilakukan lebih cepat dengan akurasi mAP (*mean-average precision*) tetap baik. Jenis dan perbandingan varian model YOLOv5 dapat dilihat pada Gambar 3.3.

Nano YOLOv5n	Small YOLOv5s	Medium YOLOv5m	Large YOLOv5l	XLarge YOLOv5x
4 MB _{FP16} 6.3 ms _{V100} 28.4 mAP _{COCO}	14 MB _{FP16} 6.4 ms _{V100} 37.2 mAP _{COCO}	41 MB _{FP16} 8.2 ms _{V100} 45.2 mAP _{COCO}	89 MB _{FP16} 10.1 ms _{V100} 48.8 mAP _{COCO}	166 MB _{FP16} 12.1 ms _{V100} 50.7 mAP _{COCO}

Gambar 3. 3. Varian Model YOLOv5

Pelatihan model YOLOv5 dilakukan dengan menentukan dataset, ukuran batch, ukuran gambar, dan model YOLOv5 atau *-weights*. Semua hasil pelatihan disimpan ke `run/train/` dengan direktori `run` yang bertambah, yaitu `run/train/exp2`, `run/train/exp3` etc. Proses training dilakukan dengan 20 *epoch*. Hasil training yang muncuk akan berupa hasil *training* terbaik (*best.pt*) dan hasil *training* terakhir (*last.pt*).



Gambar 3. 4. Alur Training Custom Data

Berdasarkan Gambar 3.4. Alur pelatihan pada YOLOv5 dalam membuat model untuk mendeteksi objek adalah proses yang berulang. Dalam mengumpulkan dan mengatur gambar, memberi label pada objek yang dijadikan model, melatih model, menerapkan untuk membuat prediksi, dan kemudian menggunakan model yang diterapkan.

3.2.1 Pengujian Model

Pengujian dilakukan dengan menggunakan hasil training yang dilakukan menggunakan metode CNN dengan arsitektur YOLO sebelumnya. Keluaran dari hasil training adalah akurasi dengan hasil terbaik dan hasil *epoch* terakhir. Tahap pengujian model dilakukan dengan memuat file model yang telah dilatih atau dilakukan *training* dengan hasil yang terbaik (*best*). Menerapkan klasifikasi menggunakan model yang telah dilatih. Bertujuan mengetahui apakah gambar yang terdeteksi nantinya menggunakan masker atau tidak menggunakan masker. Output yang diperoleh berupa *confidence*, nama kelas, serta posisi koordinat *bounding box* dari manusia yang terdeteksi.

