**BAB III**

**METODE PENELITIAN**

1. **Analisis Kebutuhan**

Pada bagian ini akan membahas tentang apa saja kebutuhan sistem yang akan digunakan nantinya untuk melakukan pendeteksian kelengkapan pakaian keselamatan para pekerja dengan metode You Look Only You. Kebutuhan sistem yang akan digunakan haruslah memiliki spesifikasi yang cukup untuk menjalankan aplikasi yang akan digunakan, sehingga bisa berjalan dengan baik. Berikut analisa kebutuhan sistem yang dipakai yaitu:

1. Perangkat Keras

Pada pembuatan sistem dibutuhkan perangkat keras untuk menunjang perangkat lunak pada sistem yang akan dibuat, sehingga sistem yang diharapkan akan berjalan seperti yang diharapkan, perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut :

|  |  |  |
| --- | --- | --- |
| No. | Perangkat Keras | Tipe Perangkat Keras |
| 1. | Prosesor | AMD A8- 7410 APU |
| 2. | Memori | 8.00 GB |
| 3. | VGA | Radeon Grahpics |
| 4. | Hardisk | 240 GB |
| 5. | Handphone | Infinix HOT 9 Play |

1. Perangkat Lunak

Perangkat lunak digunakan untuk mendukung sistem yang akan dibuat, perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut

|  |  |  |
| --- | --- | --- |
| No. | Perangkat Lunak | Tipe Perangkat Lunak |
| 1. | Sistem Operasi | Windows 10 Pro 64 bit |
| 2. | Bahasa Program | Python version |
| 3. | Aplikasi / Web yang digunakan | Sublime Text 3  Kaggle.com  Colab.reserch.google.com  www.makesense.ai |
| 4. | Android | Android 10 |

1. Kebutuhan Data

Data yang digunakan untuk penelitian ini ialah data yang dihasilkan dari data training dan data test. Pada pengumpulan dataset dalam penelitian ini menggunakan google images untuk mencari gambar pekerja berpakaian keselamatan lengkap dan pekerja tidak berpakaian keselamatan lengkap yang memiliki format foto jpg atau jpeg.

Pada data set mempunyai 2 folder yaitu folder images dan labels, dimana folder images berisi gambar pekerja berpakaian keselamatan pakaian lengkap dan tidak lengkap sebanyak 100 gambar. Sedangkan folder labels berisi label dari gambar folder image. Semua data diambil dari google images dan bentuk format data png.

GAMBAR FLOWCHART KEBUTUHAN DATA

1. **Tahap Penelitian**

Tahap penelitian dibagi menjadi tiga tahap yaitu proses pengumpulan data, proses training dan proses testing. Adapun alur kerja sistem yang dibangun dalam penelitian ini akan digambarkan pada tahap-tahap berikut.

* 1. **Pengumpulan Data**

Data yang dikumpulkan antara lain adalah gambar pekerja berpakaian keselamatan lengkap dan pekerja tidak berpakaian lengkap. Pada proses pengumpulan data ini dilakukan langsung oleh peneliti.

Gambar flowchart pengumpulan data

Pengumpulan data gambar dilakukan pengunduhan di google images dengan ketentuan gambar pekerja berpakaian keselamatan lengkap dan pekerja tidak berpakaian lengkap. Kemudian setelah terunduh dan terkumpul di dalam folder data train, pada data train terdapat 2 folder yaitu folder images dan folder labels. Selanjutnya akan masuk ke sesi labelling data. Data gambar yang sudah dikumpulkan akan masuk proses labelling menggunakan website make sense. Kemudian untuk gambar yang sudah di labelling akan di ekspor ke YOLO format. Setelah terunduh di ekstrak dan dipindahkan dalam folder labels.

* 1. **Proses Training**

**Gambar flowchart proses training**

Pada tahap ini peneliti menggunakan google colab untuk proses training. Google colab digunakan untuk upload data set dengan fomat file zip. Kemudia diakukan modifikasi file data yaml. Selanjutnya data yaml diubah jalur file tempat gambar train dan validasi disimpan. Mengubah jumlah class menjadi 4 dengan nama helm, pakaian, celana dan sepatu. Kemudian disimpan dengan nama custom data yaml. Data yaml yang sudah dicustom diupload ke folder data. Peneliti memilih model YOLOv5s. Data yang sudah di upload di cari dan di salin jalur, kemudian di copy paste dan dijalankan di program train model YOLOv5s.

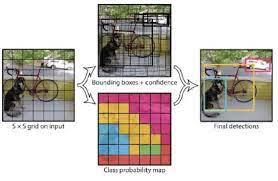
* 1. **Proses Testing**

**Gambar flowchart proses testing**

Pada tahap ini dilakukan pencarian dan penyalinan jalur folder data images kemudian di copy paste dan dijalankan pada program deteksi menggunakan model YOLOv5s. Semua hasil proses testing disimpan ke run/train/ dengan direktori run yang bertambah, yaitu run/train/exp2, run/train/exp3 dll. Setelah proses deteksi selesai kemudian download hasil deteksi yang ada di dalam folder run detect.

1. **Model YOLO v5**

You Look Only You (YOLO) merupakan sistem yang dibuat berdasarkan prinsip bahwa manusia dapat melihat dan akan mengetahui secara langsung apa saja yang terlihat oleh mata dan di mana letaknya. Pada YOLO, sistem akan membingkai ulang deteksi objek sebagai masalah regresi tunggal, langsung dari piksel gambar hingga koordinat bounding box dan probabilitas kelas. Prinsip dalam YOLO sangat sederhana, satu jaringan konvolusional secara bersamaan memprediksi banyak bounding box dan probabilitas kelas untuk kotak itu. YOLO melatih gambar penuh dan secara langsung mengoptimalkan deteksi kinerja. Model terpadu ini memiliki beberapa manfaat lebih dari metode tradisional deteksi objek (Naufal, M, R dan Rahmi Eka Putri, 2020).



Gambar 3.1 Diagram Algoritma YOLO

Metode ini dilatih melalui gambar yang sudah disediakan dan langsung mencari cara yang terbaik untuk mengoptimalkan performanya sendiri. YOLO bertugas untuk membagi gambar menjadi grid berukuran S x S. Apabila suatu objek berada dalam sel grid, sel tersebut bertanggung jawab untuk memprediksi objek tersebut. Setiap grid sel memprediksi B Bound Boxes dan nilai daripada bounding boxes tersebut. Nilai tersebut berfungsi sebagai apakah terdapat model di dalam kotak tersebut dan seberapa tinggi tingkat akurasinya. Setiap bounding boxes berisi 5 prediksi yaitu x,w,y,h dan confidence score. Koordinat (x,y) mewakili pusat berisi dari kotak yang terbatasi oleh sel grid. Sedangkan w dan h merupakan hasil prediksi lebar dan tinggi suatu objek terhadap gambar keseluruhan. Confidence score bertugas mewakili IOU (Intersection Over Union). Setiap sel grid juga akan memprediksi 1 set class probabilitas Pr(Class|Object). Skor yang dihasilkan mengkodekan kedua probabilitas class tersebut seberapa baik kotak yang diprediksi dan cocok dengan objek (Redmon et al, 2016).