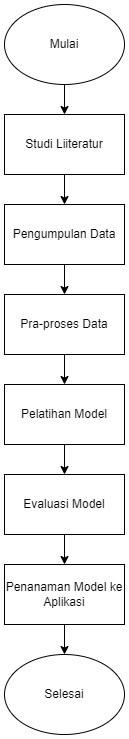
# **BAB III METODOLOGI PENELITIAN**

## **Tahapan Penelitian**

Dalam pelaksanaan penelitian ini, berikut merupakan metode yang akan digunakan untuk melaksanakan eksperimen penelitian, yaitu.



Gambar 3.1 Diagram Alur Tahapan Penelitian

Gambar 3.1 merupakan diagram alur tahapan penelitian yang akan dilaksanakan oleh penulis. Tahapan dimulai dari tahapan melakukan studi literatur kemudian melakukan pengumpulan data-data yang dibutuhkan yang akan diproses pada proses utama dalam penelitian yang dilakukan. Proses utama dalam penelitian ini dimulai dari tahapan pra-proses data, selanjutnya dilakukan perancangan dan pembuatan algoritma YOLOv4 untuk dilakukan pelatihan model dari hasil pra-proses data. Jika model telah dilakukan pelatihan, maka dilanjutkan melakukan evaluasi model untuk mengetahui model yang terbaik. Setelah diketahui model yang terbaik berdasarkan hasil evaluasi, maka dilanjutkan ke dalam tahap penanaman model ke dalam aplikasi.

## **Studi Literatur**

Pada tahapan studi literatur, kegiatan yang dilakukan, yaitu mencari, membaca, dan mempelajari publikasi artikel ilmiah, makalah ilmiah, dan penelitian-penelitian sebelumnya dengan tema yang serupa dengan penelitian yang dilakukan oleh penulis yang tersedia secara terbuka di internet. Studi literatur dilakukan dengan tujuan mendapatkan dasar teori pendukung dalam melaksanakan sebuah penelitian seperti teori mengenai pengolahan citra, algoritma YOLOv4, hingga penanaman model ke dalam aplikasi.

## **Pengumpulan Data**

Dataset yang digunakan dalam penelitian ini diambil dari situs *online* terbuka yang menyediakan beragam dataset, yaitu situs Kaggle dengan berbagai sumber akun yang memublikasikan ke dalam Kaggle. Dataset dilakukan pengunduhan dan dikumpulkan menjadi sebuah folder untuk disimpan dan diolah pada tahapan berikutnya. Dari dataset yang telah diunduh melalui Kaggle, perlu dilakukan tahapan pra-proses data untuk dilakukan penyesuaian dengan kebutuhan seperti dilakukan pengolahan terhadap dataset menyesuaikan dengan kebutuhan penelitian.

## **Pra-proses Data**

Pada tahapan pra-proses data, akan dilakukan beberapa tahapan untuk mengolah dataset yang telah dikumpulkan. Pada tahapan ini terdiri dari dua bagian dalam melakukan pra-proses data. Langkah pertama, yaitu melakukan anotasi data sesuai dengan format YOLOv4. Dari hasil dataset yang telah dilakukan anotasi data dilanjutkan melakukan akuisisi data.

### **Anotasi Data**

Dari dataset yang telah dikumpulkan sebelumnya, dilakukan proses anotasi pada dataset, yaitu melakukan pelabelan pada setiap gambar sesuai dengan format YOLOv4. Setiap gambar pada dataset dilakukan pelabelan menjadi dua kategori, yaitu memakai rompi keselamatan kerja dan tidak memakai rompi keselamatan kerja.

Pada tahapan anotasi dara menggunakan aplikasi labelIMG untuk melakukan pelabelan. LabelIMG dibuat menggunakan bahasa pemrograman *python* yang dijalankan menggunakan terminal untuk membuka *interface* dan menjalankan aplikasi LabelIMG. LabelIMG menyediakan beberapa jenis anotasi yang disimpan di antaranya CreateML, file XML dalam format PASCAL VOC, yaitu format yang digunakan oleh ImageNet. Selain itu, LabelIMG juga mendukung format YOLO yang akan digunakan pada penelitian ini. sebelum menjalankan aplikasi LabelIMG dibutuhkan pemasangan *library*,yaitu *python* sebagai *library* dasar dan Qt5 sebagai *library interface*, kemudian unduh kumpulan file dari aplikasi LabelIMG pada github milik tzutalin. Pada kumpulan file aplikasi Labellmg yang telah diunduh, dilakukan perubahan pendefinisian class atau kategori sesuai dengan kebutuhan penelitian yaitu pada folder data dengan nama file predefined classes yang di isi sesuai dengan kategori class dari penelitian.

Cara melakukan anotasi menggunakan Labellmg yaitu pertama, membuka terminal dan menjalankan file Labellmg.py. Setelah interface aplikasi terbuka, ubah format anotasi menjadi YOLO dan mengatur folder untuk menyimpan hasil anotasi. Langkah selanjutnya yaitu membuka folder dari dataset dan memuat seluruh gambar yang ada didalam folder tersebut. Setelah gambar yang akan di anotasi sudah ditampilkan maka dilakukan proses pemberian bounding box pada setiap bagian gambar yang di inginkan dan pilih class kategori yang sesuai kemudian simpan. Hasil anotasi yang disimpan berupa file text dengan format txt. Berikut merupakan format anotasi dari YOLOv4:

Objet class adalah nilai integer dari claus kategori objek yang bernilai mulai dari 0 hingga total dari claxs-1,x dan y merupakan titik tengah dari hounding box, width merupakan lebar dari bounding box dan height merupakan tinggi dari bounding box.

### **Akuisisi Data**

Tahapan akuisisi data yang dilakukan, yaitu melakukan pembagian atau split dataset yang telah dianotasi menjadi data latih, data validasi dan data tes. Data latih adalah dataset yang digunakan dalam pelatihan model yang dibuat sedangkan data validasi adalah dataset yang digunakan untuk melakukan validasi pada model dan mengetahui performa dari model dan data tes adalah data yang digunakan untuk menguji hasil dari model yang telah dilakukan pelatihan dan validasi.

Dalam tahapan akuisisi data dibuatkan sebuah program menggunakan bahasa pemrograman python yang dapat melakukan pembagian dataset secara otomatis untuk mempersingkat dan mempermudah proses split data. Data yang telah dilakukan pembagian disimpan berupa file text dengan format txt yang berisi dari nama gambar sesuai proses split

Pada penelitian ini dataset dibagi dengan rasio 80% dari dataset untuk data latih. 10% untuk data validasi dan 10% untuk data tes.

Tabel 3.1 Tabel Pembagian Dataset

|  |  |  |
| --- | --- | --- |
| **Jenis Data** | **Rasio** | **Jumlah Data** |
| Total Data | | 5000 |
| Data Latih | 80% | 4000 |
| Data Validasi | 10% | 500 |
| Data Tes | 10% | 500 |

## **Pelatihan Model**

Pelatihan model adalah tahapan yang dilakukan untuk melatih model menggunakan data latih yang telah disiapkan pada tahapan sebelumnya. Dalam pelatihan model algoritma yang digunakan adalah YOLOV4. Setelah dilakukan pra-proses data, maka dilanjutkan pada tahapan pelatihan model untuk digunakan dalam melakukan deteksi objek. Tahapan pelatihan model di antaranya:

### **Membangun Arsitektur YOLOv4**

Sebelum melakukan tahapan pelatihan model maka diperlukan pembuatan arsitektur YOLOv4. YOLOV4 menggunakan koneksi CSP yang dibangun dengan Darknet-53 di sebagai backbone dalam ekstraksi fitur sesuai yang dijelaskan pada bab sebelumnya.

Dasar dari arsitektur YOL.Ov4 diperoleh melalui cloning atau unduhan dari pembuat algoritma YOLOV4 yang disimpan di folder yang telah ditentukan. Dari arsitektur dasar yang telah diunduh dilakukan improvisasi untuk menyesuaikan kebutuhan dan keterbatasan dari media yang digunakan dalam melakukan pelatihan model. Improvisasi yang dilakukan yaitu mengubah proses penyimpanan bobot saat proses pelatihan model.

Pada arsitektur dasar dari pembuat algoritma YOLOv4 bobot selalu disimpan adalah bobot dengan mAP paling tinggi dengan nama file yolov4- obj\_best.weights, kemudian setiap 1000 epoch dengan nama file yolov4- obj <epoch weights dan akan menyimpan bobot dengan nama yolov4- obj last.weights disetiap 100 iterasi. Pada arsitektur dasar dilakukan improvisasi yaitu menyimpan setiap 100 iterasi dengan nama file yolov4-obj\_<epoch>.weights agar dapat digunakan melanjutkan pelatihan model jika proses pelatihan berhenti mendadak.

Tools yang digunakan dalam tahapan pelatihan model yaitu Google Colaboratory sebagai media untuk melakukan pengkodean dan menjalankan program yang dibuat dengan Google Drive sebagai media penyimpanannya. Improvisasi yang dilakukan sangat diperlukan karena dalam proses pelatihan model menggunakan GPU pada Google Colaboratory dan memanfaatkan koneksi internet yang tidak dapat dipastikan kestabilannya sehingga dapat terputus sewaktu-waktu dalam proses pelatihan model. Dengan adanya improvisasi tersebut mengurangi kehilangan bobot yang di inginkan ketika sudah mencapai batas penggunaan GPU, atau terputusnya koneksi internet sehingga dapat melanjutkan proses pelatihan model dilain waktu.

### **Konfigurasi Hyperparameter YOLOv4**

Dalam metode pembelajaran mesin salah satunya pada Algoritma YOLOv4 terdapat nilai yang harus di lakukan penyesuaian diantaranya adalah parameter dan hyperparameter. Pembeda antara parameter dan hyperparameter adalah pada parameter nilai yang telah ditetapkan akan berubah dengan sendirinya dalam proses pelatihan sedangkan nilai dari hyperparameter yang telah ditetapkan tidak akan terjadi perubahan selama proses pelatihan berjalan. Nilai yang dapat diatur pada hyperparameter YOLOV4 di antaranya adalah:

Tabel 3.2 Hyperparameter pada YOLOv4

|  |  |
| --- | --- |
| **Hyperparameter** | **Deskripsi** |
| Batch | Variabel yang berisi total data latih yang akan di proses dalam pelatihan model setiap satu iterasi pelatihan. |
| Subdivision | Variabel yang mengatur jumlah mini batch di setiap batch. Mini batch adalah jumlah data yang akan diproses GPU dalam kurun satu waktu. Mini batch dihitung dengan rumus: mini batch-batch/subdivision. |
| Width dan height | Width dan height merupakan ukuran resolusi dari lapisan input yang akan dilalui gambar sehingga gambar berubah resolusi sesuai nilai width dan height yang ditetapkan. Width merupakan lebar sedangkan height merupakan tinggi dari ukuran resolusi. |
| Max\_batches | Variabel yang mengatur jumlah maksimal iterasi pada proses pelatihan model. |
| Steps | Variabel yang mengatur iterasi pada saat learning rate mengecil sesuai dengan scale factor. |
| Scale | Variabel yang berisi nilai scale factor yang berfungsi sebagai pengecil learning rate. |
| Momentum | Variabel yang berfungsi membantu mengetahui Langkah berikutnya berdasarkan pengetahuan pada Langkah sebelumnya pada proses pelatihan model. |
| Decay | Variabel yang berisi nilai untuk menghilangkan ketidakseimbangan data pada dataset. |
| Learning rate | Variabel yang berfungsi mengatur kecepatan jaringan dalam memproses bobot pada saat pelatihan model. |
| Filters | Variable yang berisi nilai jumlah dari filter yang digunakan dan sesuai dengan jumlah dari class. Jumlah dari class kategori objek yang akan dilakukan pelatihan model. |
| Classes | Jumlah dari class kategori objek yang akan dilakukan pelatihan model |

Hyperparameter pada algoritma YOLOv4 tersimpan pada file konfigurasi dengan ekstensi file (.cfg) pada folder cfg dalam arsitektur YOLOv4 yang sudah dibangun. Pada penelitian ini improvisasi yang dilakukan adalah melakukan tunning terhadap hyperparameter yang akan dibahas pada sub bab skenario peneliti. Tujuan dari hyperparameter adalah untuk menemukan nilai hyperparameter dari skenario penelitian yang memberikan hasil performa model terbaik.

### **Memuat Bobot Pre-trained YOLOv4 dan Dataset**

Setelah arsitektur YOLOV4 telah selesai dibangun, proses berikutnya ialah melakukan *transfer learning* menggunakan bobot yang tela dilatih sebelumnya oleh pembuat algoritma YOLOv4 menggunakan dataset MS COCO. *Transfer learning* merupakan teknik yang menggunakan model sebelumnya yang sudah dilatih untuk digunakan sebagai titik mulai pada proses pelatihan yang baru.

Bobot pre-trained didapatkan dari proses unduh dari github pembuat algoritma YOLOv4 dan disimpan dalam dub folder dengan nama “data” di satu folder yang sama dengan arsitektur YOLOv4 yang telah dibuat sebelumnya pada Google Drive. Selain memuat Pre-trained YOLOv4, dataset juga diunggah ke dalam google drive berupa file archive berformat zip dan dilakukan ekstraksi ke dalam folder dengan nama "data" di satu folder yang sama dengan arsitektur YOLOv4.

### **Pelatihan Dataset pada Model YOLOv4**

Setelah semua skenario dari hyperpatameter sudah dikonfigurasikan dan telah diterapkan di setiap model YOLOv4, Langkah selanjutnya yaitu melakukan pelatihan pada dataset. Proses pelatihan dilakukan menggunakan Google Colaboratory menggunakan akselerator GPU dari Google Colaboratory untuk mempersingkat waktu proses pelatihan dataset. Akan tetapi, penggunaan akselerator GPU dari Google Colaboratory memiliki batasan penggunaan dan harus menggunakan koneksi internet sehingga improvisasi yang dilakukan pada model YOLOV4 dapat mempermudah untuk melanjutkan proses pelatihan data di sesi berikutnya tanpa kehilangan iterasi bobot yang paling mendekati batasan penggunaan Google Colaboratory atau ketika koneksi internet terputus. Pada proses pelatihan model, bobot baru akan otomatis tersimpan yaitu setiap 100 iterasi, bobot terbaik, dan bobot terakhir. Pelatihan dataset dilakukan sesuai jumlah skenario penelitian. Dari hasil pelatihan dataset disetiap skenario pelatihan dipilih bobot dengan mAP tertinggi untuk dilakukan pengujian dengan data tes dan melakukan evaluasi model.

## **Evaluasi Model**

Setelah pelatihan model selesai dilakukan dan telah mendapat bobot dari setiap skenario, tahapan selanjutnya yaitu melakukan evaluasi pada setiap model. Evaluasi yang dilakukan diantaranya adalah menggunakan Mean Average Precision (mAP), Intersection Over Union (IoU), Confusion Matrix, Precision, Recall, dan F1 Score.

## **Penanaman Model ke dalam Aplikasi**

Setelah diketahui hasil dari evaluasi model terhadap semua skenario, penulis akan memilih satu bobot model yang memiliki performa paling baik di antara semua skenario untuk dilakukan penanaman model ke dalam aplikasi. Akan tetapi, dikarenakan dalam penelitian ini fokus yang diteliti bukan pada pembuatan sistem. maka penulis tidak membahas proses perancangan dan pembuatan sistem aplikasi secara lebih detail.

Model yang telah ditanamkan ke dalam aplikasi dapat berguna untuk membantu pengawasan terhadap keselamatan di area kerja kritis yang mengharuskan menggunakan Alat Pelindung diri (APD) yaitu safety helmet. Gambaran umum dari cara kerja sistem aplikasi yaitu pengguna dapat menghubungkan komputer ke dalam kamera atau cctv sebelum menjalankan aplikasi. Aplikasi yang telah dibuka dan terhubung dengan kamera atau cctv maka akan langsung mendeteksi area yang ditangkap oleh kamera atau cctv, selanjutnya ditampilkan kedalam layar komputer dengan hasil deteksi yang telah dilakukan.

Aplikasi yang dirancang untuk penanaman model dibangun dengan library dari python yaitu menggunakan Flask. Untuk memuat model dari YOLOv4, penulis akan menggunakan fungsi yang telah ada pada bahasa pemrograman python.

## **Skenario Penelitian**

Skenario penelitian yang dilakukan yaitu membandingkan hasil evaluasi model berdasarkan improvisasi tuning hyperparameter pada YOLOv4 untuk mengetahui model dengan tuning hyperparameter terbaik. Tuning hyperparameter yang dilakukan hanya pada dua variabel hyperparameter yaitu pada ukuran resolusi input (width dan height) dan learning rate dari setiap model. Pada variabel hyperparameter yang lain akan bernilai sama pada setiap model. Berikut adalah skenario penelitian yang dilakukan:

Tabel 3.3 Skenario Penelitian

|  |  |  |
| --- | --- | --- |
| **Skenario Penelitian** | **Ukuran Resolusi Input** | **Learning Rate** |
|  | 512 x 512 | 1 |
|  | 0.1 |
|  | 0.01 |
|  | 416 x 416 | 1 |
|  | 0.1 |
|  | 0.01 |
|  | 320 x 320 | 1 |
|  | 0.1 |
|  | 0.01 |

Dalam skenario penelitian ini, diberikan 9 skenario yang terbagi menjadi 3 macam ukuran resolusi input dan 3 macam *learning rate.* Seluruh skenario penelitian akan dilatih menggunakan bobot pre-trained dan dataset yang sama.