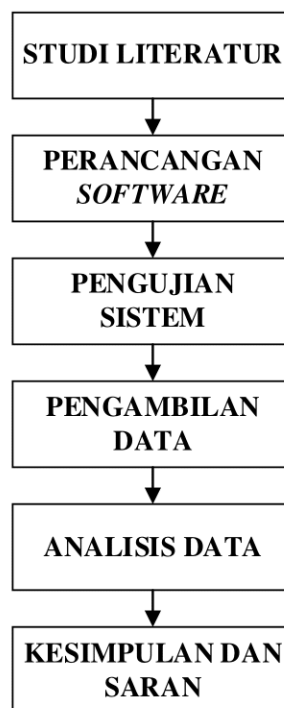


BAB III METODOLOGI PENELITIAN

Perancangan yang dilakukan dalam Skripsi ini meliputi pengumpulan *dataset* berupa gambar wajah dan perancangan perangkat lunak. Perancangan penyelesaian masalah dimulai dengan merancang diagram blok dan alur kerja sistemnya.

3.1 ALUR PENELITIAN

Pada penelitian ini, ada beberapa tahapan-tahapan yang akan dilakukan demi mewujudkan hasil yang diharapkan. Berikut ini merupakan tahapan pelaksanaan pengimplementasian arsitektur YOLO dalam kecerdasan buatan pada bidang *vision computer* untuk metode presensi dengan tambahan fitur menampilkan lokasi GPS pada aplikasi berbasis Android yang akan dijelaskan melalui diagram alur sebagai berikut.



Gambar 3.1 Diagram alur penelitian.

Gambar 3.1 di atas merupakan diagram mengenai tahapan alur penelitian, tahapan alur penelitian tersebut meliputi studi literatur, perancangan *software*, pengujian sistem, pengambilan data, analisis data, serta yang terakhir adalah

kesimpulan dan saran. Penjelasan untuk tahapan penelitian dari diagram alur di atas adalah sebagai berikut:

1. Studi Literatur

Pada tahap studi literatur bertujuan untuk mengumpulkan, mempelajari dan mengolah bahan penelitian mengenai arsitektur algoritma YOLO, serta *software* yang digunakan untuk mendukung penelitian serta alat dan bahan yang perlu dikumpulkan pada saat pengimplementasian dari arsitektur YOLO tersebut. Untuk sumber referensi yang digunakan berasal dari jurnal, artikel dan *website* yang terpercaya (bukan diambil dari blogspot atau wordpress).

2. Perancangan *Software*

Tahapan kedua yang dilakukan yaitu perancangan *software*. Pada tahap ini akan disusun program yang akan mendukung berjalannya penelitian ini, perancangan *software* yang akan dibuat meliputi pembuatan *software* untuk arsitektur YOLO itu sendiri termasuk pengambilan *dataset* berupa gambar wajah manusia yang digunakan untuk *training* arsitektur. Kemudian, membuat *web* untuk manajemen akun pengguna yang akan melakukan presensi dengan dilengkapi pembuatan *database* dan perancangan tampilan UI dari aplikasi berbasis Android.

3. Pengujian Sistem

Setelah perancangan *software* sudah dilakukan, maka langkah selanjutnya adalah pengujian sistem, pada tahap ini, pengujian dilakukan untuk mengetahui apakah *software* yang telah dirancang sudah sesuai dengan hasil yang diharapkan atau tidak. Pengujian yang dilakukan pertama kali tentunya adalah arsitektur algoritma YOLO, apakah arsitektur tersebut dapat memprediksi objek wajah atau bukan, dilihat juga hasil keyakinan arsitektur tersebut, seberapa yakin sistem terhadap objek yang sedang diprediksi. Kemudian, dilakukan juga pengujian metode presensi yang berbasis *web* dengan memasukkan data ke *database*, apakah bisa dilakukan atau tidak, serta pengujian lokasi yang ditampilkan pada aplikasi dengan menggunakan API dari

Google Maps. Apabila hasilnya sudah dirasa sesuai dan cukup, maka dapat dilakukan tahapan selanjutnya yaitu pengambilan data dari hasil pengujian sistem.

4. Pengambilan Data

Setelah pengujian sistem, terdapat sebuah hasil data yang menampilkan seberapa akurat arsitektur YOLO yang sudah digunakan, dari hasil data tersebut akan dikumpulkan untuk dapat dianalisis ke tahapan selanjutnya.

5. Analisis Data

Data yang sudah diperoleh pada tahap pengumpulan data, akan diolah dan dianalisis pada tahapan ini, data tersebut dianalisis sehingga dapat diketahui alasan dari perolehan nilai yang didapatkan ketika proses *training* data dilakukan.

6. Kesimpulan dan Saran

Tahapan yang terakhir adalah penyusunan kesimpulan dan saran terkait hasil penelitian yang sudah dilaksanakan.

3.2 ALAT DAN BAHAN

Alat dan bahan yang digunakan dalam penelitian ini, meliputi perangkat keras dan perangkat lunak, yaitu diantaranya:

3.2.1 Perangkat Keras

Di bawah ini akan dijelaskan mengenai perangkat keras yang akan digunakan pada penelitian ini beserta dengan fungsinya. Tertampil dengan menggunakan tabel 3.1 di bawah ini:

Tabel 3.1 Penggunaan perangkat keras dan fungsinya

No.	Nama Perangkat Keras	Fungsi
1	Laptop	Merancang <i>software</i> dengan menggunakan aplikasi berbasis <i>Windows</i>
2	Kamera <i>Webcam</i>	Mengumpulkan <i>dataset</i> berupa gambar wajah manusia, dan mengambil citra gambar untuk mendeteksi objek berupa wajah manusia

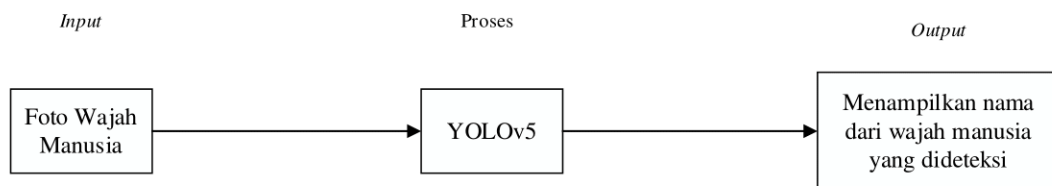
3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian kali ini, merupakan aplikasi yang berjalan di sistem operasi Windows dengan beberapa aplikasi yang digunakan bersifat *multi-platform*, berikut perangkat lunak yang digunakan untuk penelitian kali ini, yaitu:

1. Android Studio
2. Visual Studio Code
3. Postman
4. Google Colab
5. Roboflow

3.3 BLOK DIAGRAM

Pada penelitian ini, memiliki alur pengerjaan yang dijelaskan melalui blok diagram dengan poin berupa *input*, proses dan *output*. Blok diagram tertampil pada gambar berikut.

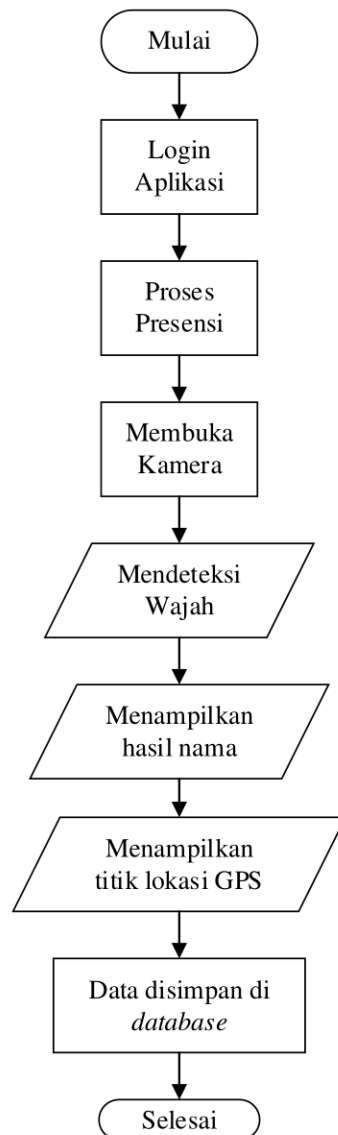


Gambar 3.2 Blok diagram penelitian.

Gambar 3.2 adalah ilustrasi dari blok diagram penelitian mengenai pemanfaatan arsitektur algoritma YOLO untuk mendeteksi wajah manusia. *Input* yang diberikan yaitu berupa gambar foto wajah manusia yang sebelumnya diambil menggunakan kamera. Setelah foto tersebut diambil, maka akan diproses menggunakan arsitektur algoritma YOLO versi kelima atau YOLOv5, dari arsitektur YOLO akan diproses berdasarkan *dataset* yang sebelumnya telah dimasukkan ke arsitektur tersebut dengan sebelumnya telah melewati proses *training*. Setelah gambar *input* atau masukkan tersebut diolah dengan arsitektur YOLO, maka akan menghasilkan *output* berupa nama dari pemilik wajah yang telah difoto sebelumnya. Nama ini yang akan digunakan untuk identifikasi wajah untuk proses pemanfaatan lebih lanjut.

3.4 FLOWCHART SYSTEM

Penelitian ini memanfaatkan YOLO untuk mendeteksi objek wajah dan mengenali nama dari pemilik wajah tersebut. Selanjutnya, akan digunakan untuk metode presensi agar wajah dari nama yang hendak melakukan presensi akan terpantau keasliannya bahwa hadir dalam kegiatan belajar mengajar. Untuk *flowchart* atau sistem kerja dari pengimplementasian YOLO, tertampil pada *flowchart* di bawah ini:



Gambar 3.3 *Flowchart system* kerja aplikasi presensi

Pada gambar 3.3, diilustrasikan *flowchart system* dari alur kerja aplikasi presensi yang akan dibuat, diawali dengan “*login aplikasi*” sesuai dengan akun pengguna, sebelumnya akun telah dibuat oleh *administrator* dan data akun

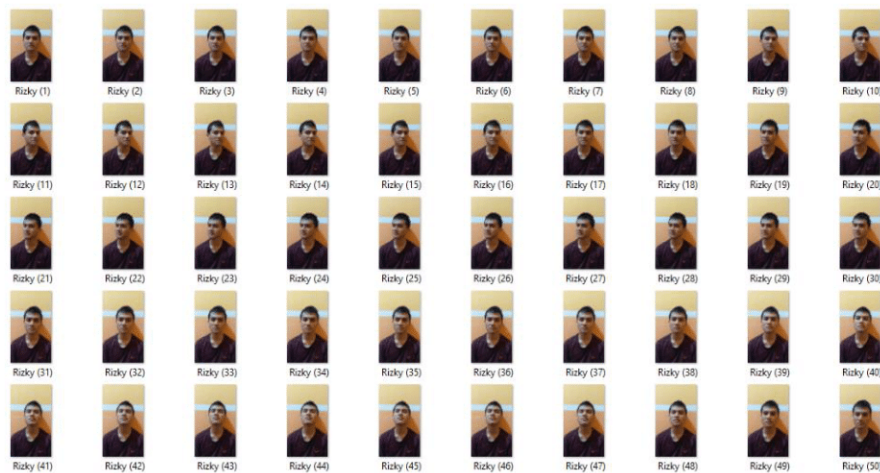
tersebut disimpan pada *database* berbasis MySQL. Selanjutnya, “proses presensi” dengan “membuka kamera” untuk “mendeteksi wajah” pengguna yang hendak melakukan presensi. Ketika sudah mengambil gambar maka akan tertampil wajah dari pengguna yang sedang melakukan presensi, kemudian akan terlihat juga titik koordinat dari lokasi pengguna agar diketahui posisi dari pengguna tersebut. Setelah itu, pengguna menekan tombol *check in* dan data tersebut disimpan di *database* untuk diaudit oleh *administrator*.

3.5 PENGUJIAN SISTEM

Pada pengujian system, terdapat beberapa poin yang perlu diperhatikan, diantaranya adalah:

3.5.1 *Data Acquisition*

Sebelum melatih model *deep learning*, diperlukan pengambilan data terlebih dahulu, tahap ini diberi nama dengan akuisisi data atau *data acquisition*. Data yang digunakan pada penelitian ini diambil dengan cara manual dari dokumentasi pribadi kepada beberapa mahasiswa/i. Data yang dikumpulkan terdiri dari lima kelas objek dari lima orang dengan gender laki-laki sebanyak tiga orang dan perempuan sebanyak dua orang, yaitu dengan nama Rizky, Tenvov, Dimas, Maharani dan Amel. *Dataset* yang sudah dikumpulkan dari masing-masing kelas yaitu berjumlah 100 gambar, dengan beberapa posisi wajah untuk membedakan gambar satu dengan gambar lainnya. Berikut merupakan dua contoh *dataset* hasil dari dokumentasi pribadi:



Gambar 3.4 Pengumpulan *dataset* gambar wajah Rizky

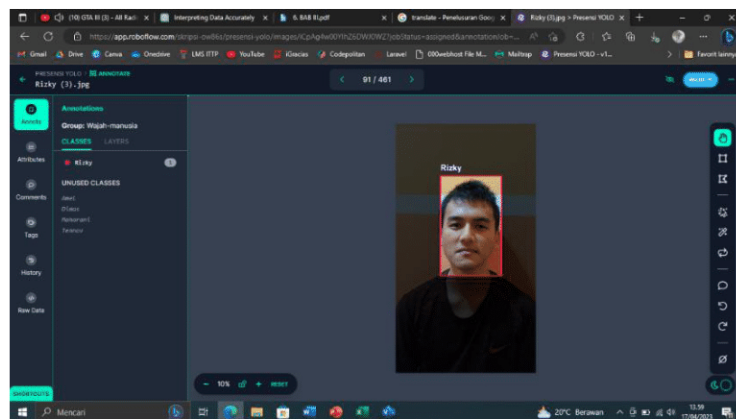


Gambar 3.5 Pengumpulan *dataset* gambar wajah Maharani

Dataset yang terkumpul dari dokumentasi pribadi tersebut berjumlah 100 gambar untuk masing-masing kelas. Foto yang diambil sepenuhnya dilakukan dengan kamera *smartphone*.

3.5.2 *Data Exploration*

Proses ini dilakukan untuk mengolah data yang telah terkumpul dengan cara memberi label dari setiap gambar pada masing-masing kelas, pemberian label tersebut menggunakan aplikasi Roboflow yang berbasis *web*. Proses pelabelan data ini bertujuan untuk memberikan BBOX atau *Bounding BOX* pada sebuah objek yang perlu dikenali oleh sistem. Berikut contoh saat pelabelan data pada Roboflow untuk sebuah foto yang akan dijadikan *dataset*.



Gambar 3.6 Proses *labelling* objek pada foto Rizky menggunakan Roboflow

Pada gambar 3.6 adalah contoh dalam memberikan label pada gambar foto Rizky, dimana pelabelan data dilakukan untuk memberikan label atau nama dari objek yang dianotasikan.

Setelah dilakukan proses pelabelan gambar sebanyak jumlah *dataset* yaitu 500 gambar, kemudian dari 500 gambar tersebut diolah dengan cara augmentasi data, agar mendapatkan gambar *dataset* yang lebih beragam. Sebelum dilakukan augmentasi, data tersebut dibagi untuk *training*, *test* dan *validation* dengan *settingan* 80%, 10% dan 10%. Sehingga, gambar yang digunakan untuk *training* berjumlah 400 gambar, untuk *test* 50 gambar dan untuk *validation* 50 gambar. Untuk penelitian ini, dilakukan *setup* augmentasi data dengan Roboflow seperti gambar di bawah.

PREPROCESSING	Auto-Orient: Applied Resize: Stretch to 640×640 Filter Null: Require all images to contain annotations.
AUGMENTATIONS	Outputs per training example: 3 Flip: Horizontal Rotation: Between -5° and +5° Bounding Box: Flip: Horizontal Bounding Box: Rotation: Between -5° and +5°
DETAILS	Version Name: 2023-04-16 1:18pm Version ID: 1 Generated: Apr 16, 2023 Annotation Group: Wajah-manusia

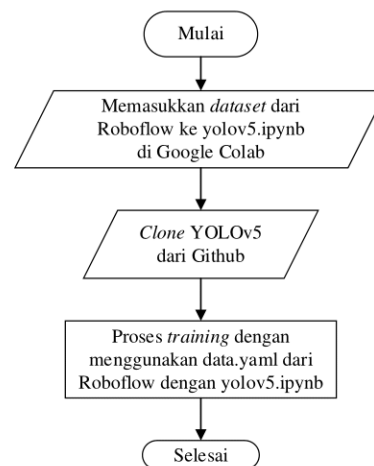
Gambar 3.7 Proses augmentasi data menggunakan Roboflow

Pada proses augmentasi data tersebut, dibagi menjadi beberapa tahap. Pertama yaitu *preprocessing*, pada tahap ini, dilakukan perubahan resolusi gambar menjadi 640×640 agar sesuai dengan ketentuan *training* data pada YOLOv5. Kedua yaitu augmentasi, pada tahap ini, dilakukan proses agar gambar yang digunakan untuk *training* yang sebelumnya berjumlah 400 gambar, dikalikan menjadi 3, sehingga menghasilkan 1200 gambar. Namun dari perkalian tersebut, gambar keseluruhan dan BBOX-nya dilakukan proses augmentasi berupa di-*flip* secara *horizontal* dan dirotasi antara -5 derajat hingga 5 derajat.

3.5.3 Modelling

Pada tahap *modelling*, dilakukan dengan melakukan *training dataset* yang telah diberi label sebelumnya. Proses *training* data ini dilakukan dengan YOLOv5 yang dikembangkan oleh Ultralytics. Pada tahap *modelling* ini, dilakukan setelah tahap *preprocessing* dan augmentasi data selesai dilakukan. Pada Roboflow, tersedia fitur ekspor *dataset* yang telah diolah (seperti terlihat pada tahap *data*

exploration) agar menjadi format yang didukung oleh sistem YOLOv5. Pada Roboflow sendiri, akan membuat sebuah API, sehingga dapat diakses di Google Colab tanpa perlu mengunduh terlebih dahulu *dataset* yang telah diolah tersebut. Setelah *dataset* diekspor, langkah selanjutnya adalah mengekstrak *dataset* tersebut sehingga didapatkan sebuah *file* bernama “data.yaml” yang nantinya digunakan untuk proses *training* data di YOLOv5. Proses *training* tersebut diilustrasikan pada *flowchart* di bawah ini.



Gambar 3.8 *Flowchart* alur tahapan *modelling* untuk *training*

Pada tahap *modelling* ini, dilakukan seperti alur pada *flowchart* di atas, kemudian, dari tahapan tersebut, akan diatur *epoch* dan *batch size* yang akan digunakan pada tahap pelatihan model. *Epoch* adalah jumlah perulangan atau iterasi yang dilakukan oleh sistem ketika semua *batch* sudah diproses, sedangkan *batch size* adalah banyaknya data yang dilakukan pada satu *epoch*, misalkan 50 *epoch* 16 *batch*, itu artinya dalam satu kali *training* dilakukan 50 kali iterasi, dengan tiap iterasi membawa 16 jumlah *batch*. Pada penelitian ini, akan dilakukan 10 kali pengetesan dengan jumlah *batch* dan *epoch* yang berbeda-beda. Yaitu *epoch* 25, 50, 100 dan 150, dengan *batch* sebesar 16, 32 dan 64.

3.5.4 *Evaluation*

Dalam melakukan *training model*, hal yang perlu diperhatikan selanjutnya adalah melakukan analisis performa model yang telah dilatih. Dalam hal itu, diperlukan sebuah metrik evaluasi. Matrik evaluasi yang diamati untuk menganalisis performa model adalah *confusion matrix*, *Recall*, *Precision* dan akurasi.

3.5.5 Confusion Matrix

Pada *machine learning*, terdapat sebuah tabel yang sering digunakan untuk mengukur kinerja dari model klasifikasi, yaitu *Confusion Matrix*. Terdapat beberapa istilah pada *Confusion Matrix*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Skema dari *Confusion Matrix* tersebut, diilustrasikan seperti pada gambar berikut:

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Gambar 3.9 Peta skema *Confusion Matrix*

Pada gambar 3.9 terdapat empat skema yang digunakan, yang pertama adalah TP atau *True Positive*, yang merupakan suatu skema di mana nilai prediksi positif dengan kenyataannya adalah benar, contoh dalam deteksi wajah manusia, maka TP diartikan bahwa terdeteksi wajah manusia dan kenyataan memang ada wajah manusia. Kedua, yaitu FP atau *False Positive*, merupakan suatu skema dimana kenyataan bernilai negatif tetapi diprediksi sebagai positif, contoh seperti kasus sebelumnya, maka FP adalah keadaan ketika terdeteksi wajah manusia, padahal kenyataannya tidak ada wajah manusia. Ketiga, yaitu FN atau *False Negative*, merupakan skema dimana kenyataan bernilai positif tetapi diprediksi sebagai negatif, artinya adalah ketika prediksi menunjukkan tidak ada wajah manusia, namun kenyataannya ada wajah manusia. Keempat, yaitu TN atau *True Negative*, merupakan skema dimana prediksi negatif dan kenyataannya benar sebagai negatif, contohnya berarti keadaan ketika terdeteksi tidak adanya wajah manusia dan kenyataannya memang tidak ada wajah manusia.

3.5.6 Recall atau Sensitivity (*True Positive Rate*)

Recall adalah suatu cara sistem untuk menggambarkan sebuah keberhasilan model dalam menemukan kembali sebuah informasi yang penting atau dengan kata lain, *recall* merupakan rasio prediksi *True Positive* yang dibandingkan dengan keseluruhan data yang *True Positive*. Nilai *recall* dapat diperoleh dengan persamaan sebagai berikut.

$$R = \frac{TP}{TP+FN}$$

Keterangan:

TP = *True Positive*

FN = *False Negative*

3.5.7 Precision (Positive Predictive Value)

Sistem dalam menggambarkan tingkat akurasi yang berhubungan antara data yang diminta dengan hasil prediksi yang diberikan oleh model, dinamakan dengan *Precision*. Dengan kata lain, *Precision* adalah suatu rasio prediksi *True Positive* yang dibandingkan dengan keseluruhan hasil yang diprediksi positif. Artinya, dari semua kelas yang bernilai positif dan telah diprediksi dengan benar, digambarkan dengan *precision*, berapa banyak data yang benar-benar bernilai positif. Nilai *precision* dapat diperoleh dengan persamaan sebagai berikut.

$$P = \frac{TP}{TP+FP}$$

Keterangan:

FP = *False Positive*

3.5.8 Akurasi

Sistem dapat menggambarkan seberapa akurat model yang digunakan dengan dapat mengklasifikasikan dengan benar. Secara sederhana, berarti akurasi merupakan tingkat kedekatan nilai yang diprediksi oleh sistem dengan nilai aktual dari kenyataan yang manusia nilai. Nilai akurasi dapat diperoleh dengan persamaan sebagai berikut.

$$A = \frac{TP+TN}{TP+TN+FP+FN}$$

Keterangan:

TN = *True Negative*

3.5.9 Pengujian Data

Pada proses ini dilakukan pengujian pada *training dataset* untuk mengetahui seberapa baik dengan menggunakan arsitektur YOLOv5 dengan objek yang telah dilatih dengan menghasilkan nilai terbaik. Pengujian ini menggunakan parameter evaluasi berupa *precision*, *recall* dan akurasi.