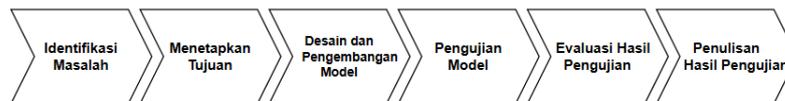


BAB III METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian merupakan tahapan atau struktur yang tersusun secara sistematis pada penelitian yang dilakukan. Penelitian ini menggunakan metode penelitian *Design and Development (D&D)*. Menurut Richey dan Klein (2014) dapat dikatakan bahwa metode D&D merupakan penelitian sistematis yang mencakup desain, pengembangan, dan evaluasi. Metode ini bertujuan untuk menciptakan atau meningkatkan produk tertentu. Tahapan dari metode penelitian *Design and Development (D&D)* dapat dilihat pada gambar 3.1.



Gambar 3.1 Tahapan Metode *Design and Development* (Richey & Klein, 2014)

3.1.1 Tahapan Penelitian

Tahapan penelitian ini mengadopsi metode penelitian *Design and Development (D&D)* yang terdiri dari beberapa tahapan sistematis dari awal hingga akhir yang di jelaskan pada gambar *flowchart* 3.2 berikut.



Gambar 3.2 Flowchart Tahapan Penelitian

Berdasarkan Gambar 3.2, penelitian dilakukan melalui beberapa langkah. Langkah penelitian dimulai dengan mengidentifikasi masalah dan studi literatur untuk pendalaman materi terkait penelitian yang akan dilakukan. Selanjutnya, pada tahap desain dan pengembangan model mengambil dan mengolah dataset yang berupa kumpulan gambar plat nomor kendaraan, serta melakukan perancangan model menggunakan bahasa pemrograman Python dan melakukan pengujian model. Jika pada evaluasi model hasil yang didapatkan kurang maksimal, akan kembali ke langkah pengambilan dan pengolahan dataset tetapi jika hasil yang didapatkan sudah maksimal akan dilanjutkan dengan penulisan laporan. Tahap-tahap dalam metode *Design and Development (D&D)* dijelaskan dalam poin-poin berikut:

3.1.2 Identifikasi Masalah

Tahap ini bertujuan untuk menemukan fokus dari masalah yang akan dibahas dan diatasi oleh model yang akan dikembangkan. Oleh karena itu, diperlukan berbagai bahan bacaan dan referensi yang diperoleh dari jurnal, skripsi, dan sumber lainnya. Pada penelitian ini masalah yang diidentifikasi adalah mendeteksi plat nomor dan membedakannya menjadi ganjil atau genap. Pada tanggal merah atau masa liburan jalan raya seringkali mengalami kemacetan. Salah satu cara yang sering digunakan untuk mengatasi kemacetan yaitu menerapkan sistem ganjil atau genap. Sistem tersebut bertujuan membatasi kendaraan yang ingin melewati jalan tertentu pada waktu tertentu berdasarkan angka terakhir dari plat nomor. Akan tetapi, pihak kepolisian yang menegakan sistem tersebut bisa jadi melewatkan kendaraan yang tidak sesuai dengan aturan yang ada atau biasa disebut sebagai *human error*.

3.1.3 Menentukan Tujuan

Tahap ini dilakukan berdasarkan pada masalah yang telah diidentifikasi. Pada penelitian ini, tujuan yang ingin dicapai adalah merancang model deteksi plat nomor yang dapat membedakan plat nomor ganjil dan genap dan mengamankan informasi yang terdapat dalam plat nomor. Model yang dirancang diharapkan dapat bermanfaat bagi pihak instansi atau lembaga terkait agar dapat

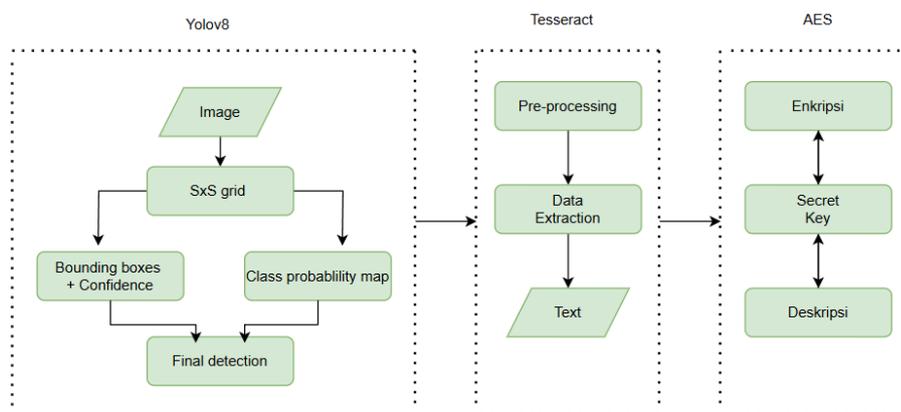
membedakan plat nomor ganjil atau genap dan mengamankan informasi didalamnya secara efektif dan efisien.

3.1.4 Desain dan Pengembangan Sistem

Tahap ini bertujuan untuk merancang dan membangun model pendeteksian plat nomor untuk sistem ganjil atau genap. Proses ini mencakup pembuatan langkah penelitian dan *flowchart* model yang akan digunakan. Pengembangan sistem melibatkan pembuatan kode program menggunakan python untuk pengembangan model, model deteksi YOLO-v8 untuk mendeteksi objek, Tesseract untuk mengekstrak karakter, dan enkripsi AES untuk mengamankan data. Detail perancangan dan tahapan model dijelaskan pada poin-poin berikut.

1. Arsitektur Model

Pada gambar 3.3 merupakan arsitektur model yang dirancang pada penelitian ini.



Gambar 3.3 Arsitektur Model

Gambar 3.3 merupakan arsitektur keseluruhan model yang menggunakan YOLO-v8, Tesseract, dan AES. *Input* berupa gambar yang akan di *grid* menjadi ukuran *SxS*. Untuk setiap sel dalam *grid*, YOLO-v8 memprediksi beberapa *bounding boxes* (kotak pembatas) serta tingkat kepercayaan (*confidence*) untuk setiap kotak tersebut. YOLO-v8 juga menghasilkan peta probabilitas kelas (*class probability map*) yang menunjukkan probabilitas objek tertentu berada di setiap sel pada *grid*. Selanjutnya Tesseract akan melakukan *pre-processing* hasil deteksi YOLO-v8 berupa *grayscale* dan binerisasi. Data yang ada pada gambar kemudian

di ekstrak menjadi teks. Terakhir, AES akan mengenkripsi data yang telah di ekstrak oleh Tesseract menjadi *chipher text* menggunakan *secret key* yang memiliki panjang 16 karakter. Deskripsi dilakukan untuk mengubah *chipher text* menjadi teks aslinya.

2. Perancangan Model

Perancangan model perlu dibuat secara matang, sehingga mampu berfungsi dengan baik dan memberikan hasil yang akurat. Proses ini melibatkan pengembangan deteksi plat nomor dengan YOLO-v8, implementasi Tesseract untuk pengenalan karakter, dan implementasi enkripsi data dengan AES yang dijelaskan sebagai berikut:

2.1 Pengembangan Deteksi Plat Nomor dengan YOLO-v8

Pada proses pengembangan deteksi plat nomor dengan YOLO-v8 melibatkan beberapa langkah yang dijelaskan sebagai berikut:

2.1.1 Persiapan Dataset

Mempersiapkan gambar plat nomor dengan dataset publik yang tersedia secara online pada *platform* Roboflow. Dataset yang digunakan memiliki 9158 gambar oleh TA, yang bernama plat nomor dataset. Dataset tersebut mencakup gambar plat nomor yang diambil dari jarak dekat hingga jarak jauh. Selain itu, kondisi pencahayaan juga beragam dari gelap hingga terang. Dataset akan dilatih pada *platform* Google Collaboratory. Pada dataset *test* akan diganti dengan gambar publik yang di *screenshots* dari *channel youtube* milik Motomobi.

2.1.2 Preprocessing Data

Melakukan *Preprocessing* pada dataset untuk meningkatkan kualitas data menjadi data yang siap untuk diolah. Pada proses ini menggunakan fungsi *resize* untuk mengubah ukuran gambar menjadi ukuran yang seragam. Selain itu, menerapkan filter *null* untuk menghapus gambar yang memiliki nilai kosong atau tidak memiliki label.

2.1.3 Augmentasi Data

Meningkatkan keberagaman pada data dengan menerapkan beberapa perubahan. Teknik ini dapat membantu menambah jumlah dan variasi data. Pertama menggunakan *Flip Horizontal* dan vertikal untuk merubah rotasi gambar secara *Horizontal* dan vertikal. Kedua, *Crop Minimum Zoom* dan *Maximum zoom* untuk memotong bagian dari gambar. Tujuannya dapat membantu model belajar fokus pada berbagai bagian gambar. Ketiga, *rotation* untuk memutar gambar searah atau berlawanan arah jarum jam.

2.1.4 Training Model

Training model YOLOv8 menggunakan dataset yang telah dipersiapkan. Pada *training* terdapat parameter *training* yang dapat disesuaikan tergantung kepada kebutuhan seperti *learning rate*, *batch size*, dan jumlah *epoch*. Parameter tersebut perlu disesuaikan agar menghasilkan model yang bagus. Semakin tinggi parameter tersebut diterapkan maka akan semakin baik pula model. Tetapi hal itu akan memerlukan daya komputasi yang besar.

2.1.5 Validasi Model

Setelah model dilatih akan divalidasi menggunakan dataset yang memiliki gambar berbeda. Tujuannya adalah untuk mengukur akurasi dan performa mode menggunakan Confusion Matrix.

2.1.6 Prediksi Model

Model akan diberikan gambar yang berbeda dari tahap *training* dan validasi untuk memprediksi objek terdeteksi atau tidak. Pada tahap ini menggunakan gambar publik yang di *screenshots* dari video milik *channel* Youtube Motomobi.

2.2 Implementasi Tesseract untuk Pengenalan Karakter

Pada proses implementasi Tesseract untuk pengenalan karakter melibatkan beberapa langkah yang dijelaskan sebagai berikut:

2.2.1 *Pre-processing* Gambar Deteksi

Gambar plat nomor yang terdeteksi oleh model YOLO-v8 kemudian di *Crop* dan dilakukan proses *Pre-processing* untuk

meningkatkan kualitas gambar agar lebih mudah dikenali oleh Tesseract. Langkah tersebut mencakup merubah gambar ke *grayscale* dan binerisasi.

2.2.2 Konfigurasi Tesseract

Menyesuaikan pengaturan Tesseract untuk pengenalan karakter seperti bahasa yang digunakan.

2.2.3 Uji Coba Pengenalan Karakter

Melakukan uji coba pengenalan karakter pada gambar plat nomor hasil deteksi untuk memastikan bahwa Tesseract dapat mengenali karakter dengan baik dan akurat.

2.3 Implementasi Enkripsi Data dengan AES

2.3.1 Desain Skema Enkripsi

Merancang skema enkripsi yang akan digunakan untuk mengamankan data hasil pengenalan karakter. Ini mencakup penggunaan algoritma AES dengan panjang kunci 128 bit.

2.3.2 Pengembangan Modul Enkripsi

Mengembangkan modul enkripsi yang akan mengimplementasikan skema enkripsi yang telah dirancang. Modul ini harus mampu mengamankan data dengan aman.

2.3.3 Integrasi Modul Enkripsi

Mengintegrasikan modul enkripsi dengan sistem deteksi dan pengenalan karakter. Setelah karakter plat nomor ekstraksi, kemudian dilakukan proses enkripsi.

3.1.5 Pengujian

Pada tahap pengujian, model akan diuji untuk melihat fungsionalitas secara keseluruhan. Apakah YOLO-v8 mampu untuk mendeteksi plat nomor pada setiap kendaraan, Tesseract mampu untuk mengenali karakter pada plat nomor, dan AES mampu untuk melakukan enkripsi dengan baik atau tidak. Umpan balik yang diperoleh dari tahap pengujian akan digunakan untuk meningkatkan kualitas model lebih lanjut. Pengujian mencakup akurasi model YOLO-v8, akurasi Tesseract, enkripsi AES, dan klasifikasi plat nomor.

3.1.6 Evaluasi Hasil Pengujian

Pada tahap ini, hasil dari pengujian performa model dianalisis untuk mengevaluasi kinerja dan keandalan model. Berdasarkan hasil evaluasi ini, dilakukan perbaikan dan penyempurnaan pada model agar dapat meningkatkan performa model. Berikut merupakan metrik evaluasi yang digunakan untuk mengukur performa dari YOLO-v8 dan Tesseract.

1. Metrik Mean Average Precision (mAP)

Metrik ini digunakan untuk mengukur performa model deteksi YOLO-v8 dalam deteksi objek dengan rentang *threshold* 50% dan 50-95%. Cara menghitung mAP adalah sebagai berikut:

$$\text{mAP} = \frac{1}{N} \sum_{k=1}^N \text{AP}_k \quad (3.1)$$

mAP adalah rata-rata dari AP (area dibawah kurva *precision* dan *recall*) untuk semua kelas dalam dataset, memberikan gambaran umum tentang performa model deteksi objek secara keseluruhan. N adalah jumlah total kelas dalam dataset.

2. Metrik Akurasi Pengenalan Teks

Metrik ini digunakan untuk mengukur seberapa banyak karakter benar daripada seluruhnya yang terbaca oleh Tesseract. Cara menghitung akurasi Tesseract adalah sebagai berikut:

$$\text{Persentase Hasil} = \frac{\text{Karakter benar}}{\text{Jumlah karakter}} \times 100 \quad (3.2)$$

Semakin baik nilai yang diperoleh maka akan semakin pula model mengenali karakter pada plat nomor dengan tingkat kesalahan yang rendah.

3.1.7 Penulisan Laporan Hasil Pengujian

Pada tahap ini, seluruh proses dan hasil penelitian di dokumentasikan ke dalam bentuk laporan. Laporan ini berisi latar belakang penelitian, identifikasi masalah, batasan masalah, tujuan penelitian, metode penelitian, desain dan pengembangan sistem, hasil pengujian, evaluasi hasil pengujian, dan kesimpulan dan rekomendasi untuk penelitian selanjutnya.