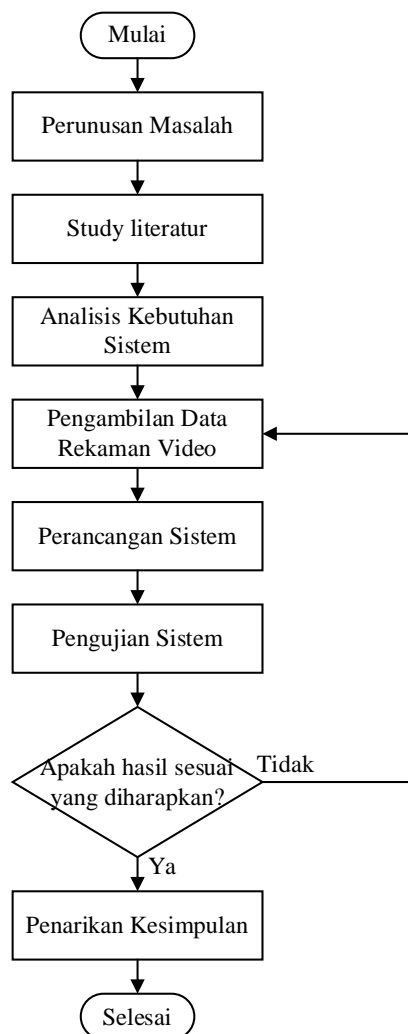


BAB III METODOLOGI PENELITIAN

3.1 Desain Penelitian

Untuk menyelesaikan sistem deteksi laju kendaraan dan plat nomor kendaraan menggunakan *DeepSORT tracking* dan *HyperLPR* diperlukan perencanaan yang matang dan teratur terhadap penelitian yang akan dibuat untuk menghasilkan sebuah sistem yang diharapkan. Langkah penelitian yang dilakukan ditunjukkan pada Gambar 3.1.



Gambar 3. 1 *Flowchart* metode penelitian

Seperti yang ditunjukkan pada Gambar 3.1, penelitian dilakukan melalui beberapa proses. Proses dimulai dengan melakukan perumusan masalah dan studi literatur dimana dilakukan pendalaman materi terkait penelitian yang dilakukan. Setelahnya melakukan pengumpulan data yang akan digunakan pada penelitian, serta melakukan perancangan sistem dengan menggunakan bahasa pemrograman python dan melakukan pengujian sistem. Hasil pengujian akan dianalisis dan dievaluasi untuk menarik kesimpulan. Uraian desain penelitian dijelaskan sebagai berikut:

3.1.1 Perumusan Masalah

Perumusan masalah merupakan tahapan perumusan masalah dan mengidentifikasi secara mendalam terkait permasalahan yang dijadikan sebagai topik penelitian. Perumusan masalah bertujuan untuk memberikan konsolidasi terkait penelitian yang dilakukan untuk bisa mendapatkan hasil yang dapat dipertanggungjawabkan. Dengan melakukan pengamatan pada jalan tol palikanci Kab. Cirebon secara langsung dan seterusnya, yang tujuannya adalah untuk mengembangkan kerangka berpikir tentang penelitian yang telah dilakukan.

3.1.2 Studi Literatur

Studi literatur bertujuan untuk mencari sumber referensi terkait penelitian yang akan digunakan sebagai acuan dalam penelitian. Eksplorasi materi dilakukan dengan mempelajari hal-hal terkait pemrograman python, *image processing*, *object detection*, *object tracking*, *deep learning*, *Convolutional Neural Network (CNN)*, *PyTorch*, penggunaan model *PyTorch*, kalman filter, karakteristik *YOLOv5*, karakteristik *DeepSORT* untuk melakukan proses *object detection* dan *object tracking* berdasarkan latar belakang permasalahan yang ada dan karakteristik metode *HyperLPR* dalam menentukan deteksi plat nomor kendaraan.

3.1.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem melibatkan pengumpulan kebutuhan mengenai hal-hal yang akan diimplementasikan, analisis dan mendefinisikan kebutuhan sistem secara keseluruhan. Tahap ini bertujuan untuk mengidentifikasi masalah dan menentukan bagaimana sistem akan memenuhi tujuan dari pengembangan sistem.

3.1.4 Pengambilan Data

Pengambilan data video pada area jalan tol dilakukan dengan kamera *smartphone* yang ditempatkan di lokasi strategis. Dalam hal ini spesifikasi kamera yang digunakan adalah 1080p dan 30 *FPS*. sehingga dapat menjangkau area plat nomor kendaraan dengan jelas. Data pada video yang diambil meliputi kendaraan, laju kendaraan dan plat nomor kendaraan. Metode pengambilan data yang diterapkan dalam penelitian ini melibatkan pemasangan kamera pengawas dengan sudut pandang yang optimal untuk mencakup area jalan tol yang relevan. Data tersebut dapat digunakan sebagai input dalam algoritma sistem deteksi laju dan plat nomor kendaraan.

3.1.5 Pengembangan Sistem

Perancangan sistem dilakukan dengan menggunakan metode eksperimen. Pengembangan sistem terdiri dari tiga proses perancangan yaitu pembuatan sistem deteksi menggunakan *YOLOv5* dan *tracking* kendaraan menggunakan algoritma *DeepSORT*, pembuatan sistem deteksi kecepatan kendaraan berdasarkan hasil *tracking* posisi piksel sebelum dan sesudah per 10 frame dan pembuatan sistem deteksi plat nomor kendaraan menggunakan *HyperLPR*. Input yang digunakan pada keseluruhan sistem penelitian ini berupa video rekaman yang telah dilakukan pada proses pengambilan data video rekaman keadaan jalan tol Cipali. Program dilakukan dengan menggunakan bahasa pemrograman *python* dan menggunakan beberapa *library* pada *python*.

3.1.6 Pengujian Sistem

Proses pengujian dilakukan untuk mengetahui tingkat keakuratan dari hasil deteksi laju dan plat nomor kendaraan. Input yang digunakan dalam pengujian menggunakan video rekaman mobil melintas yang diambil menggunakan kamera *smartphone* pada area sekitar ruas jalan tol Cipali Kabupaten Cirebon.

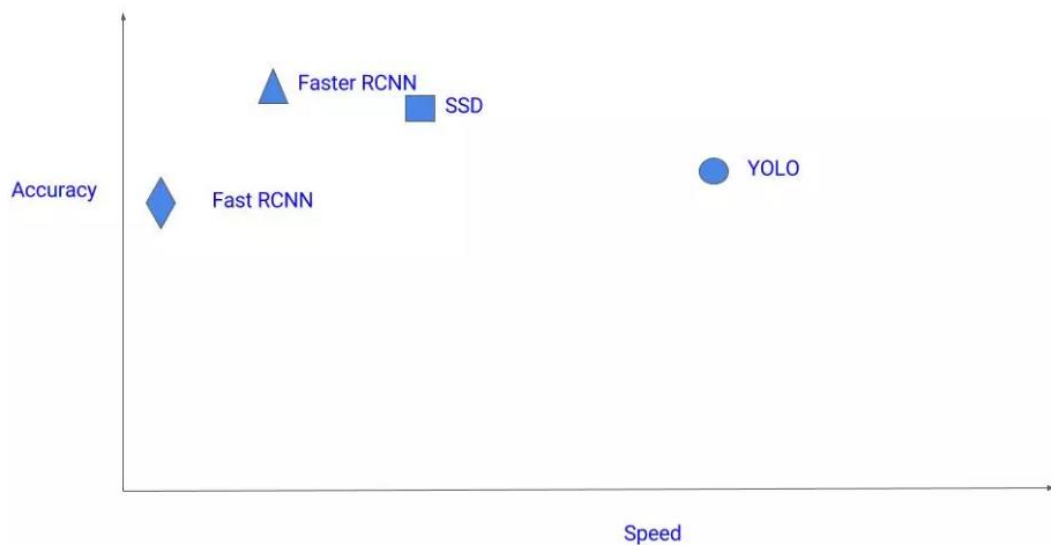
3.1.7 Penarikan Kesimpulan

Dalam tahap ini penulis membuat kesimpulan berdasarkan hasil pengujian yang telah dilakukan. Kesimpulan yang akan diambil akan menunjukkan apakah

deteksi laju dan plat nomor kendaraan menggunakan *YOLOv5 DeepSORT* dan *HyperLPR* akurat dan dapat dijalankan secara bersamaan.

3.2 Desain Prototyping

Dalam proses perancangan sistem ini dilakukan dengan menerapkan lima tahap perancangan yang meliputi tahap deteksi, tahap pelacakan, tahap estimasi laju kendaraan, tahap deteksi plat nomor kendaraan. Pada tahap deteksi kendaraan, penulis memilih menggunakan model *deep learning YOLOv5*. Keputusan ini diambil karena model deteksi *YOLO* memiliki kecepatan yang lebih tinggi dibandingkan dengan model deteksi lainnya, sehingga proses deteksi objek dapat dilakukan dengan cepat, dan tingkat akurasi yang cukup baik. Oleh karena itu, model ini sesuai dengan penelitian ini, di mana akurasi deteksi laju kendaraan dipengaruhi oleh kecepatan deteksi dan *tracking* pada kendaraan tersebut. Dikutip dari *website* (Sachan dan Sinhal., 2017) perbandingan grafik antara model deteksi *YOLO* dan model deteksi lainnya dapat dilihat pada Gambar 3.2.

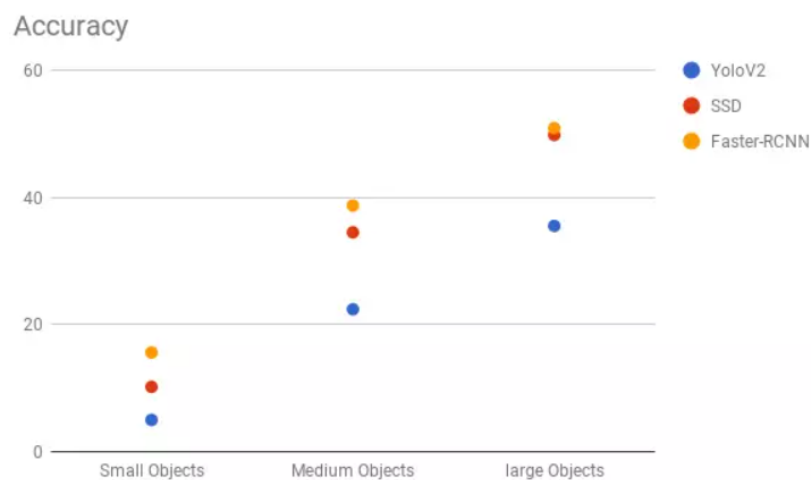


Gambar 3. 2 Perbandingan grafik antara model deteksi *YOLO*, *SSD*, *Fast-RCNN* dan *Faster-RCNN*

Sumber. Sachan dan Sinhal., 2017

Berdasarkan gambar tersebut, dapat terlihat bahwa model objek deteksi dengan menggunakan *YOLO* memiliki laju paling tinggi dibandingkan dengan model lainnya dalam mendeteksi objek. Dalam tingkat akurasi model *YOLO* dikatakan

cukup baik untuk mendeteksi objek dan masih lebih baik dibandingkan dengan *Fast RCNN* namun masih dibawah *SSD*, dan *Faster RCNN*. Berdasarkan ukuran objek deteksi model deteksi *YOLO* memiliki tingkat akurasi paling rendah dibandingkan dengan model *SSD*, dan *Faster-RCNN* baik dalam ukuran objek kecil, sedang dan besar. Namun dalam deteksi objek besar model deteksi *YOLO* masih memiliki tingkat akurasi yang mumpuni sehingga masih bisa diandalkan dalam mendeteksi kendaraan. Gambar 3.3 menampilkan perbandingan antara model objek deteksi *YOLO*, *SSD* dan *Faster-RCNN* berdasarkan ukuran objek.



Gambar 3. 3 Perbandingan antara model objek deteksi *YOLO*, *SSD* dan *Faster-RCNN* berdasarkan ukuran objek

Sumber. Sachan dan Sinhal., 2017

Dalam tahap *tracking*, penulis menerapkan algoritma *DeepSORT*. *DeepSORT* adalah algoritma *tracking* objek yang menggunakan *deep neural network* untuk mendeteksi objek dalam video dan melacak pergerakan dari objek berdasarkan *frame to frame*. Penerapan algoritma *DeepSORT* pada penelitian ini dikarenakan algoritma ini dapat mengidentifikasi objek yang sama pada pada frame yang berbeda, meskipun terjadi perubahan posisi, ukuran dan orientasi sehingga akan efektif untuk diterapkan pada sistem *tracking* objek. Pada penelitian ini penggunaan *DeepSORT* dapat membantu untuk menghindari terjadinya tumpang tindih antara kendaraan yang terdeteksi jika jarak dari masing-masing kendaraan tersebut berada pada posisi yang berdekatan.

Selanjutnya dalam menghitung estimasi dari kecepatan pada setiap kendaraan Salah satu aspek yang paling penting dalam deteksi laju kendaraan berbasis video adalah mengkonversi jarak pada video menjadi jarak sebenarnya. Dalam konteks ini, jarak piksel dalam video harus disesuaikan dengan jarak sebenarnya agar laju dapat dihitung secara akurat. Prinsip utama dalam estimasi laju kendaraan secara *real-time* adalah dengan merekam perpindahan posisi piksel pada *tracking ID* kendaraan yang dilakukan pada tahap sebelumnya, selain itu ukuran dari setiap kendaraan dapat berpengaruh dalam mendeteksi laju dari kendaraan. Perekaman perpindahan piksel pada kondisi sebelum dan sesudahnya dilakukan berdasarkan setiap per 10 frame. Selanjutnya estimasi laju kendaraan ditentukan berdasarkan rumus perpindahan jarak berdasarkan *ID* frame sebelum dan sesudah pada setiap 10 frame tersebut. Adapun rumus perhitungannya adalah sebagai berikut:

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

X_2 = Posisi mobil sesudahnya pada bidang horizontal

X_1 = Posisi mobil sebelumnya pada bidang horizontal

Y_2 = Posisi mobil sesudahnya pada bidang vertikal

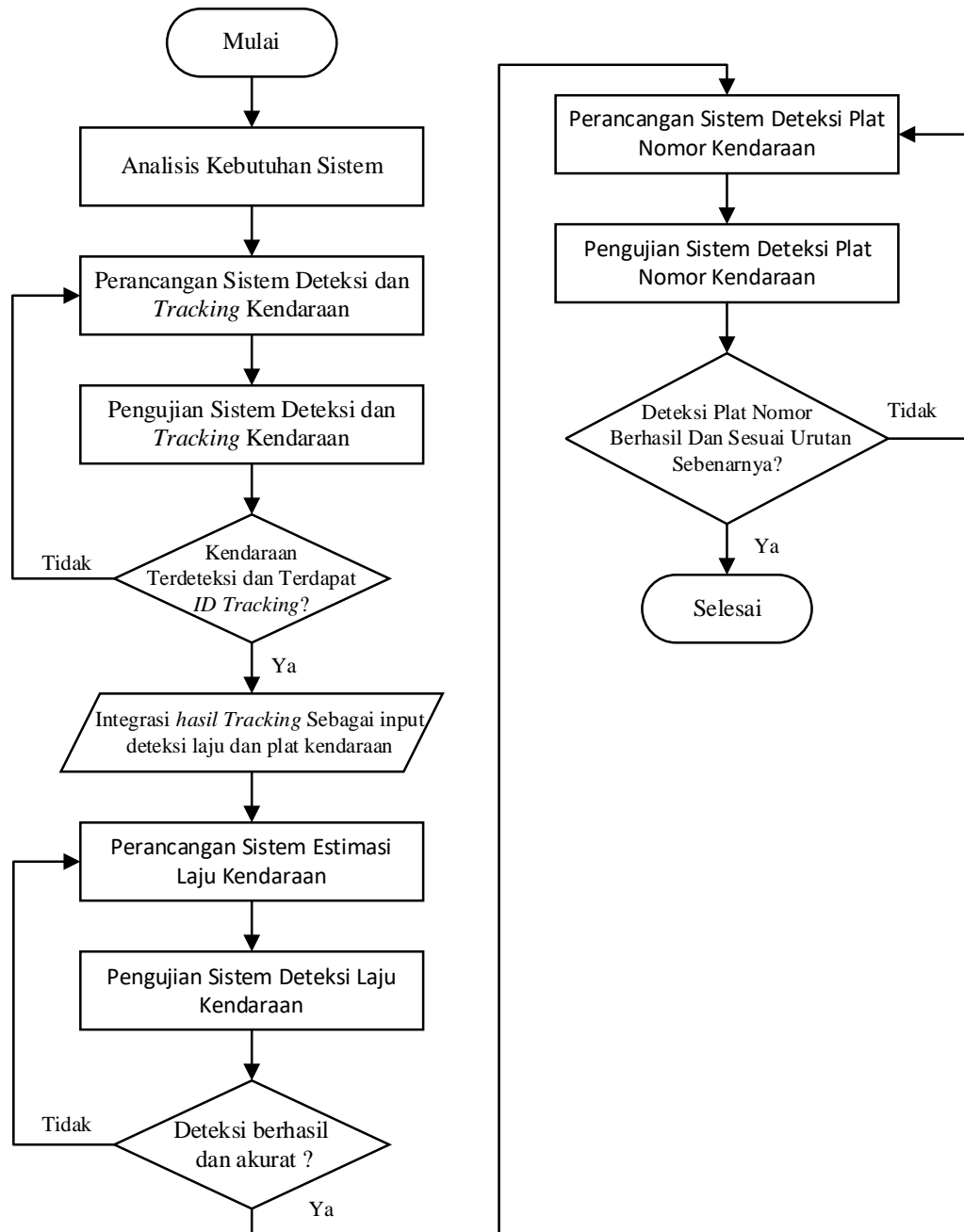
Y_1 = Posisi mobil sebelumnya pada bidang vertikal

Deteksi plat nomor kendaraan pada dasarnya bekerja dengan dua tahapan, yaitu tahap mendeteksi objek plat nomor kendaraan dan tahap mendeteksi karakter dari hasil deteksi plat nomor kendaraan tersebut. Pada *HyperLPR* metode deteksi objek plat nomor kendaraan menggunakan *cascade classifier* sehingga dalam mendeteksi objek relatif cepat namun dalam hal akurasi tidak terlalu akurat. Sedangkan dalam mendeteksi karakter pada plat nomor kendaraan *HyperLPR* melakukan proses training model dataset pengenalan karakter dengan menerapkan *CTC Loss*. *CTC Loss* adalah metode untuk melatih model jaringan saraf dalam tugas seperti pengenalan ucapan dan pengenalan tulisan tangan. *CTC Loss* memungkinkan model untuk mempelajari pemetaan sekuensial tanpa pasangan input-output yang sekuensial. Ini dilakukan dengan menghitung probabilitas jalur pemetaan menggunakan forward-backward algorithm dan menggunakan backpropagation untuk mengoreksi model. *HyperLPR* akan mengganti karakter yang ditampilkan

pada plat nomor pada *ID tracking* kendaraan akan mengalami pembaruan tergantung pada nilai *confidence*, jika nilai *confidence* plat nomor terdeteksi lebih besar dari *confidence* sebelumnya maka hasil yang ditampilkan akan diganti, hal ini bertujuan untuk mendapatkan hasil deteksi mendekati karakter plat nomor sebenarnya.

3.3 Metode Pengembangan Sistem

Dalam penelitian ini digunakan metode eksperimen berbasis pemrosesan video dengan menerapkan serangkaian langkah pemrograman dengan menerapkan YOLOv5, DeepSORT dan HyperLPR. Metode ini dipilih karena dapat diterapkan untuk melakukan deteksi pelanggaran batas laju dan plat nomor kendaraan dengan memanfaatkan kamera pengawas area jalan Tol. Gambar 3.4 menunjukkan *flowchart* dalam pengembangan sistem tersebut.



Gambar 3. 4 *Flowchart* rancangan sistem deteksi laju dan plat nomor kendaraan

1. Analisis kebutuhan sistem

Pada tahap ini, langkah yang dilakukan adalah melakukan pengumpulan, analisis, dan definisi kebutuhan sistem secara keseluruhan. Fokus tahap ini adalah

untuk mengidentifikasi masalah yang ada dan menentukan bagaimana sistem akan mencapai tujuan yang telah ditetapkan dalam pengembangan sistem.

2. Perancangan sistem

Pada tahap ini, seluruh desain akan dijalankan dalam bentuk kode program. Setiap kode pemrograman masih terdiri dari modul-modul yang nantinya akan digabungkan menjadi sistem yang dapat digunakan untuk melakukan estimasi kecepatan kendaraan dan deteksi plat nomor kendaraan melalui kamera pengawas lalu lintas. Perancangan sistem terdiri dari tiga tahap, yaitu:

- a. Perancangan deteksi dan *tracking* kendaraan
- b. Perancangan sistem estimasi laju kendaraan
- c. Perancangan sistem deteksi plat nomor kendaraan

3. Pengujian sistem

Pada tahap ini, dilakukan penggabungan modul-modul yang telah dibuat. Dilakukan pengujian untuk menganalisis sistem yang telah dibuat sesuai dengan desainnya, serta memeriksa adanya kesalahan dalam semua fungsi sistem. Pengujian dilakukan dengan menggunakan proses pengujian individual atau komponen sebelum semua bagian terintegrasi dengan sistem lain secara keseluruhan. Unit pengujian terdiri dari pengujian objek deteksi dan pelacakan, pengujian estimasi kecepatan kendaraan, serta pengujian deteksi plat nomor kendaraan.

4. Evaluasi sistem

Setelah semua sistem diimplementasikan dan dimodifikasi berulang kali hingga sesuai dengan tujuan yang diharapkan, dilakukan evaluasi sistem. Tujuan dari evaluasi sistem adalah untuk menemukan kesalahan-kesalahan dan melakukan perbaikan terhadap setiap fungsi yang mungkin terjadi, baik dalam pemrograman maupun di luar pemrograman.

3.4 Perangkat Keras dan Perangkat Lunak

Lingkungan komputasi bertujuan untuk menentukan kebutuhan perangkat lunak dan perangkat keras dalam perancangan sistem. Lingkungan dari penelitian ini dilampirkan pada Tabel 3.1 dan 3.2 sebagai berikut.

3.4.1 Kebutuhan Perangkat Keras

Tabel 3. 1 Perangkat keras yang digunakan dalam penelitian

Perangkat keras yang digunakan	Spesifikasi
Kamera	48MP Ukuran sensor 1/2” f/1,79
<i>CPU</i>	Ryzen 5 4600H
<i>GPU</i>	NVIDIA GFORCE GTX 1650
<i>Memory</i>	8 GB DDR4
<i>Storage</i>	512 GB SSD
Aksesoris	<i>Mouse dan Keyboard</i>

3.4.2 Kebutuhan perangkat lunak

Tabel 3. 2 Perangkat lunak yang digunakan dalam penelitian

Nama Perangkat Lunak	Perangkat Lunak Yang Digunakan
Sistem Operasi	Windows 11
Text Editor	PyCharm
Bahasa Pemrograman	Python
<i>Framework dan library</i>	<ul style="list-style-type: none"> • OpenCV • Numpy • <i>DeepSORT</i> • <i>PyTorch</i> • PIL • <i>HyperLPR</i> • <i>YOLOv5</i>

