

BAB III

METODOLOGI PENELITIAN

3.1 Gambaran Umum Objek Penelitian

Objek penelitian ini berfokuskan kepada implementasi sebuah sistem *object detection* untuk menghitung jumlah penumpang yang masuk kedalam transportasi umum bus. *Object detection* ini memanfaatkan *deep learning* untuk membaca sebuah objek berupa orang atau *person* yang akan dideteksi melalui sebuah video. Video yang digunakan adalah video saat penumpang memasuki sebuah bus melalui pintu masuk. Kemudian, dari sistem tersebut akan mencoba untuk melakukan *object detection person* yang sedang masuk kedalam bus. Sistem yang dibentuk memanfaatkan tools seperti *Visual Studio Code* dan *Google Collaboratory*. *Visual Studio Code* adalah sebuah aplikasi editor kode sumber yang dibuat oleh Microsoft dan tersedia secara gratis untuk sistem operasi Windows, Linux, dan MacOS. Dengan aplikasi ini, pengguna dapat menulis kode dengan mudah dan aplikasi ini juga mendukung berbagai jenis bahasa pemrograman, termasuk C++, C#, Java, Python, PHP, dan GO [38]. Sedangkan *Google Collaboratory* atau disingkat *Colab* adalah sebuah layanan *cloud* yang didasarkan pada *Jupyter Notebooks* dan digunakan untuk mendistribusikan pengetahuan dan penelitian tentang *machine learning*. Layanan ini memungkinkan pengguna untuk mengakses lingkungan pengembangan yang telah disiapkan dan di-hosting oleh Google sehingga pengguna dapat melakukan eksperimen, melakukan pengolahan data, dan menjalankan kode *machine learning* tanpa memerlukan instalasi lokal pada komputer mereka. Layanan ini gratis untuk digunakan dan dapat diakses oleh siapa saja yang memiliki akun Google [39].

Video penumpang yang masuk dan keluar transportasi bus diambil pada tahun 2016. Video penumpang terdiri dari 4 yaitu video penumpang masuk dalam kondisi tidak ramai, video penumpang masuk dengan kondisi ramai, video penumpang keluar dengan kondisi tidak ramai, dan video penumpang keluar dengan kondisi ramai. Video tersebut rata-rata memiliki durasi 10 – 20 detik dikarenakan hanya mengambil *moment* dimana penumpang akan memasuki dan

keluar dari transportasi bus. Data berupa video ini, didapatkan dari PT. Teknologi Karya Digital Nusa dalam rangka untuk pengembangan *object detection* yang diharapkan dapat diimplementasikan kedalam projek bus mereka.

3.1.1 Video Penumpang Masuk Bus

a) Penumpang Ramai



Gambar 3. 1 Video Pertama Penumpang Masuk Ramai

Gambar 3.1 diatas merupakan *screenshot* dari sebagian *frame* video pertama penumpang masuk kedalam bus dalam kondisi ramai. Dalam kondisi ini, volume penumpang yang masuk kedalam bus memiliki jumlah yang besar dan penumpang cenderung tidak mengantri saat masuk kedalam bus.



Gambar 3. 2 Video Kedua Penumpang Masuk Ramai

Gambar 3.2 adalah video kedua dari penumpang ramai saat masuk ke dalam bus. Dalam kondisi tersebut, maka penumpang tidak akan mengantri dengan rapih untuk masuk kedalam bus dan tingkat volume penumpang masuk ke dalam bus besar.

b) Penumpang Tidak Ramai



Gambar 3. 3 Video Pertama Penumpang Masuk Tidak Ramai

Gambar 3.3 diatas adalah *screenshot* dari salah satu *frame* pada video pertama penumpang masuk bus dalam kondisi tidak ramai. Dalam kondisi tersebut, volume penumpang masuk ke bus rendah dan penumpang cenderung menunggu atau mengantri saat masuk ke dalam bus.

U M V I N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3. 4 Video Kedua Penumpang Masuk Tidak Ramai

Gambar 3.4 diatas merupakan video kedua dari penumpang tidak ramai saat masuk ke dalam bus. Dalam kondisi tersebut, penumpang cenderung masuk ke dalam bus dalam kondisi satu per satu atau penumpang tidak masuk secara bersama-sama.

3.1.2 Video Penumpang Keluar Bus

a) Penumpang Ramai



Gambar 3. 5 Video Pertama Penumpang Keluar Ramai

Gambar 3.5 merupakan *screenshot* dari video penumpang saat keluar bus dalam kondisi ramai. Dalam kondisi tersebut, penumpang akan keluar dari bus secara bersamaan tanpa mengantri dan volume dari penumpang keluar itu besar dan terkesan tidak rapih.



Gambar 3.6 Video Pertama Penumpang Keluar Tidak Ramai

Gambar 3.6 adalah video kedua penumpang saat keluar dari bus dalam kondisi ramai. Dalam kondisi tersebut, penumpang cenderung keluar secara bersamaan tanpa mengantri dan berhimpitan atau berdekatan.

b) Penumpang Tidak Ramai



Gambar 3.7 Video Pertama Penumpang Keluar Tidak Ramai

Gambar 3.7 merupakan *screenshot* dari video pertama penumpang keluar dalam kondisi tidak ramai. Dalam kondisi tersebut, penumpang keluar satu per satu tanpa berdekatan atau berhimpitan. Penumpang keluar secara mengantri dan tidak keluar secara bersamaan.



Gambar 3. 8 Penumpang Keluar Melakukan Antri

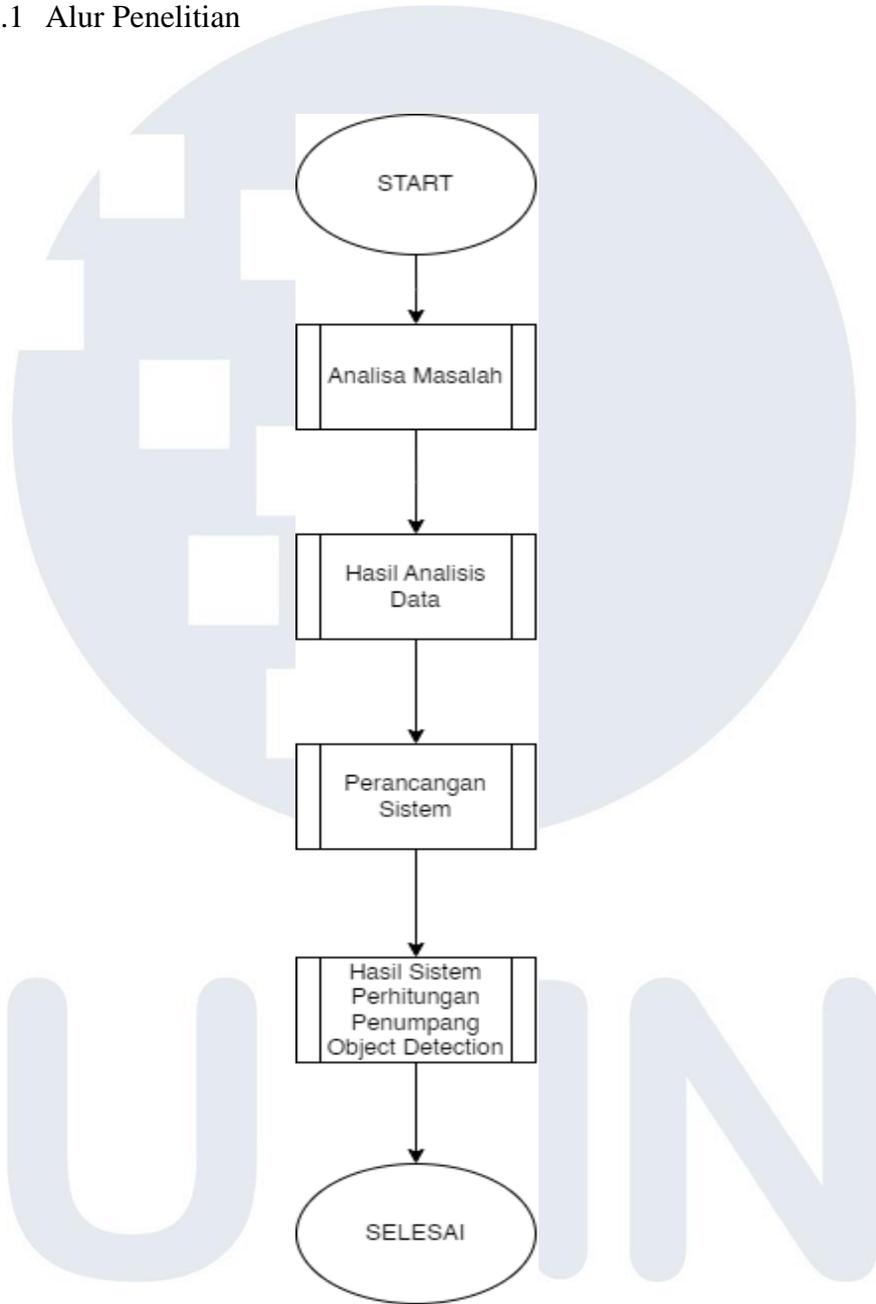
Gambar 3.8 merupakan gambaran lain ketika penumpang keluar dalam kondisi tidak ramai. Penumpang melakukan antri untuk keluar dan terlihat lebih rapih dibandingkan penumpang keluar saat kondisi ramai.

Dari kedua jenis video yang digunakan dalam objek penelitian, masing-masing video akan digunakan dalam pengimplementasian *object detection* untuk melakukan perhitungan jumlah penumpang yang masuk dan keluar. Dikarenakan video yang diberikan secara terpisah yaitu video saat masuk dan video saat keluar, maka jumlah penumpang yang dihitung tidak dapat dilakukan secara bersamaan seperti pengurangan antara penumpang yang masuk ke bus dengan penumpang yang keluar dari bus.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

3.2 Metode Penelitian

3.2.1 Alur Penelitian



Gambar 3. 9 Flowchart Penelitian

Gambar 3.9 memaparkan alur penelitian untuk melakukan *object detection* untuk menghitung jumlah penumpang pada bus. Alur penelitian *object detection* merupakan kerangka uji salah satu penelitian Implementasi Deteksi *Real-time* Pada Jenis Kendaraan di Indonesia menggunakan YOLOv5

[19]. Berikut penjelasan terkait dengan alur penelitian seperti pada gambar diatas:

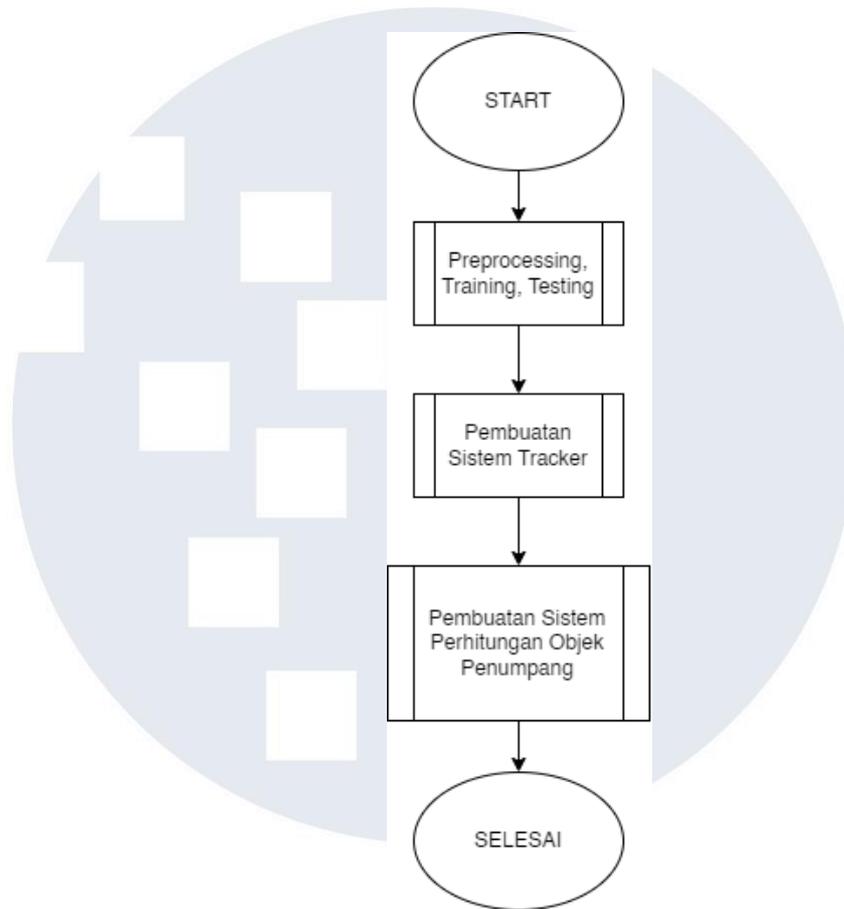
1. Analisa Masalah

Masalah yang sering dihadapi oleh penumpang pada saat menggunakan transportasi umum adalah masalah keselamatan. Banyak kasus kecelakaan pada transportasi umum yang disebabkan oleh berbagai hal seperti *human error*, kehilangan fungsi pada transportasi, dan kelebihan muatan pada transportasi. Kelebihan muatan atau *over capacity* menjadi hal yang diteliti oleh peneliti dikarenakan hal tersebut masih sering dilupakan oleh masyarakat yang dapat mengancam keselamatan penumpang. Oleh karena itu, peneliti ingin menyelesaikan masalah tersebut dengan menciptakan sebuah sistem *object detection* untuk menghitung jumlah penumpang yang masuk dan keluar dari sebuah transportasi yang diharapkan dapat memberikan informasi tentang penuh atau tidak penuhnya sebuah transportasi.

2. Hasil Analisis Data

Dalam melakukan sebuah analisa, perlu memerlukan data pendukung seperti kuesioner atau wawancara. Dalam penelitian ini, peneliti melakukan analisis data dengan melakukan wawancara secara langsung dengan Bpk. Suwardi selaku *Head of IT Department* dari perusahaan PT. Teknologi Karya Digital Nusa. Perusahaan tersebut bergerak pada bidang pembuatan sistem dan aplikasi terhadap transportasi umum seperti bus yang sudah diimplementasikan di Indonesia dan sering melakukan kerjasama dengan Kementerian Perhubungan Indonesia. Wawancara dilakukan peneliti untuk mengali informasi terhadap masalah-masalah pada transportasi umum seperti bus dan ingin mencari tau apakah perusahaan memerlukan sebuah sistem yang baru untuk diimplementasikan pada transportasi bus.

3. Perancangan Sistem



Gambar 3. 10 Flowchart Perancangan Sistem

Gambar 3.10 merupakan *flowchart* dari perancangan sistem untuk melakukan *object detection* perhitungan penumpang bus. Tahap-tahap tersebut merupakan proses yang harus dilalui untuk membuat sistem *object detection* dengan memanfaatkan beberapa *tools* seperti *Roboflow*, Visual Studio Code, dan Google Collaboratory. Berikut penjelasan singkat mengenai proses-proses tersebut:

- Preprocessing, Training, Testing

Pada tahap ini, peneliti melakukan persiapan pada data yang digunakan untuk melakukan *object detection* seperti upload, annotate, generate, dan deploy. Hal-hal tersebut

dapat dilakukan dengan memanfaatkan *Roboflow* yang merupakan sebuah website yang dapat melakukan *preprocessing* data, *training*, dan *testing* data dengan mudah dan cepat.

- Pembuatan Sistem *Tracker*

Tracker pada *object detection* diciptakan untuk melacak objek secara berkelanjutan di dalam video yang diteliti. Dengan memanfaatkan informasi pergerakan dan hubungan spasial antar objek yang terdeteksi, *tracker* memastikan pelacakan dengan akurat dan konsisten. Selain itu, *tracker* dapat mengatasi kehilangan deteksi sementara, meningkatkan efisiensi komputasi, dan memberikan konsistensi dalam pemantauan objek dari waktu ke waktu.

- Pembuatan Sistem Perhitungan Objek Penumpang

Peneliti membuat 3 sistem *object detection* yang berbeda untuk menghitung penumpang bus. Sistem pertama yaitu sistem *object detection* dengan menggunakan seluruh class pada *coco.txt* yang dimanfaatkan untuk melihat apakah ada objek lain yang terdeteksi oleh area pada saat penumpang masuk dan keluar bus. Sistem kedua adalah sistem *object detection* yang hanya menggunakan class *person* pada *coco.txt* yang memberitahu sistem untuk selalu mendeteksi objek *person* dalam video. Sistem ketiga adalah sistem *object detection* yang menggunakan model *Best.pt* yang didapatkan dari hasil custom training dari dataset. Peneliti ingin membandingkan dari hasil ketiga sistem *object detection* dan dari hasil perbandingan tersebut akan dicari yang terbaik.

4. Hasil Sistem Perhitungan Penumpang *Object detection*

Pada tahap ini akan diperlihatkan hasil dari 3 sistem yang telah dibuat apakah berjalan dengan lancar dan baik. Pada setiap hasil sistem tersebut, akan dibandingkan dengan hasil *real count* yang menjadi acuan bahwa hasil *object detection* memiliki tingkat akurasi yang baik atau tidak.

3.2.2 Metode Pengembangan Sistem

YOLO sebagai pilihan metode dalam menciptakan sebuah sistem *Object detection* untuk menghitung jumlah penumpang merupakan metode yang baik. Terdapat beberapa keunggulan YOLO dibandingkan algoritma *object detection* lainnya yaitu sebagai berikut:

1. YOLO vs SSD (Single Shot Multibox Detector)

Tabel 3. 1 YOLO vs SSD

| YOLO | SSD |
|---|---|
| Kecepatan deteksi lebih tinggi daripada SSD[40] | Kecepatan lebih lambat dibandingkan YOLO[40] |
| Jumlah inference lebih rendah. [40] | Jumlah inference lebih besar dari pada YOLO. [40] |
| Performa FPS lebih tinggi dan lebih baik [40] | Performa FPS lebih rendah. [40] |

2. YOLO vs Faster R-CNN

Tabel 3. 2 YOLO vs Faster R-CNN

| YOLO | Faster R-CNN |
|---|---|
| Mengambil gambar sekali dan memprosesnya secara keseluruhan, kemudian | Melakukan deteksi pada beberapa wilayah gambar dan memprediksi berulang |

| | |
|--|---|
| memberikan prediksi sekaligus.[41] | kali pada setiap wilayah yang terdeteksi.[41] |
| Dapat melatih dan memproses gambar secara end-to-end dengan sangat efisien.[41] | Membutuhkan perangkat dengan GPU yang kuat karena membutuhkan lebih banyak perhitungan.[41] |
| Membuat lebih sedikit kesalahan latar belakang dibandingkan dengan Faster R-CNN.[41] | Memiliki akurasi yang baik tetapi kinerja waktu nyata terbatas.[41] |

3. YOLO vs RetinaNet

Tabel 3. 3 YOLO vs RetinaNet

| YOLO | RetinaNet |
|--|--|
| MAP (Mean Average Precision) (80.69%) [42] | MAP (Mean Average Precision) yang tinggi (82.89%)[42] |
| Kecepatan deteksi (FPS: 17) tidak cukup cepat untuk aplikasi real-time[42] | kecepatan deteksi dan mencapai kinerja real-time (FPS: 51)[42] |
| Ukuran model relatif kecil [42] | Ukuran model besar [42] |

Dari perbandingan diantara algoritma lainnya, menunjukkan bahwa YOLO memiliki keunggulan dalam hal kecepatan dan kesederhanaanya dalam melakukan pendeteksian objek. Namun, masing-masing algoritma memang memiliki kelebihan dan kekurangan tergantung pada kondisi tujuan yang diinginkan oleh peneliti dalam hal *object detection*. Pada penelitian ini,

pemanfaatan metode YOLO sudah tepat dalam melakukan *object detection* penumpang dan melakukan perhitungan jumlahnya.

3.3 Teknik Pengumpulan Data

3.3.1 Data Penumpang Masuk dan Keluar Transportasi Bus

Teknik pengumpulan data yang dilakukan oleh peneliti adalah dengan meminta data sample atau data contoh dari perusahaan PT. Teknologi Karya Digital Nusa berupa video penumpang yang masuk dan keluar dari transportasi bus. Video yang berdurasi kurang lebih 10 sampai 20 detik memiliki resolusi yang memadai untuk dijadikan sebagai objek penelitian. Terdapat 8 video yang digunakan dalam penelitian meliputi 2 video penumpang masuk dalam kondisi ramai, 2 video penumpang masuk dalam kondisi tidak ramai, 2 video penumpang keluar dalam kondisi ramai, dan 2 video penumpang keluar dalam kondisi tidak ramai.

3.3.2 Periode Pengumpulan Data

Data diambil dari tanggal 20 – 22 Maret 2023 dengan diberikannya akses berupa Google Drive kepada peneliti. Video yang diambil merupakan hasil rekaman video dari bus-bus di China selama periode tahun 2016.

3.4 Variabel Penelitian

3.4.1 Variabel Bebas

Variabel independen atau bebas adalah kondisi dimana sebuah variabel yang mempengaruhi timbulnya perubahan pada variabel terikat (dependen) [43]. Pada penelitian ini, yang menjadi variabel bebas adalah gambar yang terdapat pada video yang akan dilakukan *object detection* nya. Video merupakan keluar dan masuknya penumpang kedalam bus.

3.4.2 Variabel Terikat

Variabel dependen atau variabel terikat adalah variabel yang nilainya bergantung kepada variabel bebas atau independen variabel. Pada penelitian ini, variabel terikat adalah output dari model *object detection* YOLO, yaitu koordinat bounding box, kelas objek yang terdeteksi yaitu person atau penumpang.

3.5 Teknik Analisis Data

3.5.1 Precision, Recall, dan F1-Score

Teknik analisis data yang sudah umum digunakan untuk menganalisa hasil *object detection* dari algoritma YOLO adalah *precision*, *recall*, dan *F1-Score*. Teknik ini digunakan untuk mengukur seberapa baik kinerja model untuk *object detection* dengan benar dan menghindari kesalahan deteksi. *Precision* adalah ukuran seberapa akurat model dalam *object detection* positif. *Precision* dihitung dengan membagi jumlah *object detection* yang benar (*true positive*) dengan total jumlah deteksi yang dilakukan (*true positive* + *false positive*). *Recall* adalah ukuran seberapa baik model dapat menemukan semua objek positif yang ada. *Recall* dihitung dengan membagi jumlah *object detection* yang benar (*true positive*) dengan total jumlah objek positif yang sebenarnya ada dalam gambar (*true positive* + *false negative*). *F1 score* adalah ukuran rata-rata harmonis antara *precision* dan *recall*. *F1 score* dapat membantu menilai seberapa seimbang antara *precision* dan *recall* pada model *object detection*. *F1 score* dihitung dengan menggunakan rumus: $2 \times (\textit{precision} \times \textit{recall}) / (\textit{precision} + \textit{recall})$.

