

**SISTEM DETEKSI RAMBU LALU LINTAS
MENGUNAKAN ARSITEKTUR NEURAL
NETWORK YOLO**

***TRAFFIC SIGN DETECTION SYSTEM USING YOLO NEURAL
NETWORK ARCHITECTURE***

TUGAS AKHIR

Program Studi S1 Teknik Komputer

Disusun oleh:

NANDA HARRY NURBUDYANA

1103198222



**Universitas
Telkom**

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

BANDUNG

2021

LEMBAR PENGESAHAN

TUGAS AKHIR

**SISTEM DETEKSI RAMBU LALU LINTAS
MENGUNAKAN ARSITEKTUR NEURAL
NETWORK YOLO**

***TRAFFIC SIGN DETECTION SYSTEM USING YOLO
NEURAL NETWORK ARCHITECTURE***

Telah disetujui dan disahkan sebagai Buku Tugas Akhir

Program Studi Teknik Komputer

Fakultas Teknik Elektro

Universitas Telkom

Disusun oleh:

NANDA HARRY NURBUDYANA

1103198222

Bandung, 24 Juli 2021

Menyetujui,

Pembimbing I



Ir. Agus Virgono, M.T.

NIP. 93660027-1

Pembimbing II



Fussy Mentari Dirgantara, S.T., M.T.

NIP. 20950005-3

LEMBAR PERNYATAAN ORISINALITAS

Nama : Nanda Harry Nurbudyana
NIM : 1103198222
Alamat : Jl. Desa Cipatik-Kp. Picung Gede, Rt 01 Rw 05, Desa Cipatik,
Kecamatan Cihampelas, Kabupaten Bandung Barat, Jawa Barat
No. Tlp/HP : 082216649714
Email : harrynurbudyana@gmail.com

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri dengan judul:

SISTEM DETEKSI RAMBU LALU LINTAS MENGGUNAKAN ARSITEKTUR NEURAL NETWORK YOLO

TRAFFIC SIGN DETECTION SYSTEM USING YOLO NEURAL NETWORK ARCHITECTURE

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.



Bandung, 24 Juli 2021

Nanda Harry Nurbudyana

1103198222

ABSTRAK

Deteksi rambu lalu lintas telah menjadi masalah tradisional untuk kendaraan cerdas, terutama sebagai langkah pendahuluan untuk pengenalan rambu lalu lintas yang memberikan informasi berguna seperti petunjuk arah dan peringatan untuk *autonomus driving* atau *driver assistance system*. Saat ini, deteksi rambu lalu lintas telah mendapatkan perhatian dalam sistem kendaraan cerdas, dimana rambu lalu lintas dapat digunakan sebagai petunjuk berbeda untuk menunjukkan pemetaan dan lokalisasi. Berbeda dari petunjuk alam seperti titik sudut atau tepi yang memiliki tampilan sembarangan, rambu lalu lintas memiliki tampilan standar seperti bentuk, warna, dan pola yang ditentukan oleh peraturan ketat.

Tujuan penelitian ini dilakukan adalah untuk membuat sistem deteksi rambu lalu lintas yang dapat mendeteksi jenis rambu lalu lintas tertentu yang ada di Indonesia. Sistem deteksi rambu lalu lintas ini disertai dengan sistem estimasi jarak untuk memprediksi jarak kendaraan terhadap rambu lalu lintas. Arsitektur sistem deteksi objek yang digunakan adalah YOLOv4. Arsitektur estimasi jarak yang digunakan adalah Dense Depth dan MiDaS(*mixing datasets for zero-shot cross-dataset transfer*).

Berdasarkan hasil pengujian yang dilakukan dalam penelitian ini akurasi deteksi rambu lalu lintas yang diperoleh adalah 96% dan rata-rata waktu deteksi yang diperoleh adalah 32.76 ms. Hasil pengujian estimasi jarak dari dua arsitektur yang diuji berdasarkan rata-rata *error* yang diperoleh dari setiap pengujian, yaitu Dense Depth mempunyai *error* 2.52 meter dan MiDaS mempunyai *error* 4.07 meter.

Kata Kunci: Rambu Lalu Lintas, *Image Recognition*, *Neural Network*. Estimasi Jarak.

ABSTRACT

Traffic sign detection has become a traditional problem for intelligent vehicles, especially as a preliminary step to traffic sign recognition which provides useful information such as directions and warnings for autonomous driving or driver assistance systems. Nowadays, traffic sign detection has received attention in intelligent vehicle systems, where traffic signs can be used as different clues to indicate mapping and localization. Different from natural clues such as corner points or edges that have a haphazard appearance, traffic signs have a standard appearance such as shape, color, and pattern determined by strict regulations.

The purpose of this research is to create a traffic sign detection system that can detect certain types of traffic signs in Indonesia. This traffic sign detection system is accompanied by a distance estimation system to predict the vehicle's distance to traffic signs. The object detection system architecture used is YOLOv4. The distance estimation architecture used is Dense Depth and MiDaS(mixing datasets for zero-shot cross-dataset transfer).

Based on the results of the tests carried out in this study, the accuracy of traffic sign detection obtained was 96% and the average detection time obtained was 32.76 ms. The results of the distance estimation test of the two architectures tested are based on the average error obtained from each test, namely Dense Depth has an error of 2.52 meters and MiDaS has an error of 4.07 meters.

Keywords: *Traffic Signs, Image Recognition, Neural Networks. Distance Estimation.*

KATA PENGANTAR

Puji syukur kita panjatkan Kepada Allah SWT yang hingga saat ini masih memberikan kita nikmat iman dan kesehatan, sehingga penulis diberi kemudahan dan kelancaran dalam menyelesaikan Buku Tugas Akhir. Dalam Buku Tugas Akhir ini yang berjudul “Sistem Deteksi Rambu Lalu Lintas Menggunakan Arsitektue Neural Network YOLO” disusun untuk melengkapi salah satu syarat kelulusan program studi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom. Buku Tugas Akhir ini diharapkan bisa bermanfaat bagi pembaca dan bisa menjadi referensi untuk pembuatan tugas akhir berikutnya.

Penulis menyadari bahwa dalam Tugas Akhir ini masih banyak kekurangan, oleh karena itu kritik dan saran yang membangun sangat dibutuhkan dan diharapkan penulis untuk kedepannya dalam penyusunan Buku Tugas Akhir yang serupa bisa jauh lebih baik. Demikian buku Tugas Akhir ini disusun, penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi semua.

Bandung, 24 Juli 2021

Penulis

UCAPAN TERIMA KASIH

Dengan menyelesaikan Tugas Akhir di program studi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom. Penulis banyak mendapatkan bantuan baik materi, informasi dan dukungan dari berbagai pihak. Pada kesempatan ini penulis ingin mengucapkan rasa terima kasih sebesar-besarnya kepada:

1. Allah SWT, atas segala petunjuk, rahmat, karunia dan hidayah-Nya penulis dapat menyelesaikan tugas akhir ini.
2. Nabi Muhammad SAW, yang telah menjadi panutan bagi seluruh umat muslim dalam menjalani kehidupan di dunia ini.
3. Orang tua tersayang, Bapak Thuryana dan Ibu Nurelah yang tidak pernah henti memberikan segalanya kepada penulis dan selalu mendukung serta memberikan perhatian.
4. Saudari tersayang, Nabila Wilma Nubudyani, yang selalu memberikan doa.
5. Saudari sepupu, Hilda Afrianthy yang telah membantu dalam penyelesaian Tugas Akhir ini.
6. Bapak Ir. Agus Virgono, M.T. selaku dosen pembimbing satu Tugas Akhir yang telah berkenan menyediakan waktu untuk memberikan pengarahan dan bimbingan dalam penyusunan Tugas Akhir ini.
7. Ibu Fussy Mentari Dirgantara, S.T., M.T., selaku dosen pembimbing dua Tugas Akhir yang telah berkenan menyediakan waktu untuk memberikan pengarahan dan bimbingan dalam penyusunan Tugas Akhir ini.
8. Bapak Umar Ali Ahmad, S.T., M.T., PhD. selaku dosen wali yang selalu memberikan dukungan dalam penyelesaian Tugas Akhir ini.
9. Segenap dosen program studi S1 Teknik Komputer atas segala jasa serta ilmunya selama ini.
10. Ahmad Alfi Adz-Dzikri dan Muhammad Muchlis Kurnia selaku rekan satu kelompok penulis dalam Tugas Akhir ini yang selalu mendukung dan saling memberikan masukan dan motivasinya.

11. Rekan-rekan TK-43-01 ekstensi yang telah berjuang bersama dari awal memasuki program studi S1 Teknik Komputer hingga saat ini.
12. Rekan-rekan ATOMIC MAN Insan Cendekia Jambi yang selalu memberi semangat, motivasi dan dukungan kepada penulis.
13. Fina Najma Amalia, yang selalu memberikan dukungan dan semangat di saat senang maupun susah.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
LEMBAR PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR.....	vi
UCAPAN TERIMA KASIH.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	2
1.5. Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1. Image Recognition.....	4
2.2. Image Classification	5
2.3. Neural Network	5
2.4. YOLO	6
2.5. Depth Estimation.....	7
BAB III PERANCANGAN SISTEM	8
3.1. Gambaran Umum Sistem	8
3.1.1. Proses Deteksi Rambu Lalu Lintas	9
3.1.2. Proses Depth Prediction.....	10
3.2. Analisis Sistem	11
3.2.1. Analisis Kebutuhan Perangkat Lunak.....	11
3.2.2. Analisis Kebutuhan Perangkat Keras	11
3.2.3. Analisis Kebutuhan Pengguna	11
3.2.4. Analisis Kebutuhan Data	11

3.3. Perancangan Sistem.....	12
3.3.1. Blok Diagram.....	13
3.3.2. Penyusunan Dataset.....	13
3.3.3. Pembuatan Replika Rambu Lalu Lintas	14
3.3.4. Validasi Data Estimasi Jarak	14
3.4 Perancangan Pengujian.....	15
3.4.1 Pengujian Deteksi Rambu Lalu Lintas	15
3.4.2 Pengujian Estimasi jarak.....	16
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	17
4.1. Implementasi	17
4.1.1. Implementasi Deteksi Rambu Lalu Lintas.....	17
4.1.2. Implementasi Estimasi Jarak	17
4.2. Pengujian Sistem	18
4.2.1. Pengujian Deteksi Rambu Lalu Lintas	18
4.2.2 Pengujian Estimasi Jarak	22
BAB V KESIMPULAN DAN SARAN	28
5.1. Kesimpulan.....	28
5.2. Saran	28
DAFTAR PUSTAKA	29
LAMPIRAN A	31
LAMPIRAN B	43

DAFTAR GAMBAR

Gambar 2.1 Image Recognition	4
Gambar 2.2 Image Classification.....	5
Gambar 2.3 Perceptron	6
Gambar 2.4 Objek Deteksi YOLO	7
Gambar 3.1 Gambaran Umum Sistem.....	8
Gambar 3.2 Proses Deteksi Objek.....	9
Gambar 3.3 Proses Depth Prediction	10
Gambar 3.4 Jenis Rambu Lalu Lintas	12
Gambar 3.5 Blok Diagram Sistem.....	13
Gambar 4.1 Implementasi Deteksi Rambu Lalu Lintas.....	17
Gambar 4.2 Depth Image	18

DAFTAR TABEL

Tabel 4.1 Akurasi Pengujian Deteksi Objek.....	19
Tabel 4.2 Gambar Hasil Deteksi Rambu Lalu Lintas	20
Tabel 4.3 Pengujian Waktu Deteksi Rambu Lalu Lintas.....	21
Tabel 4.4 Pengujian Estimasi Jarak.....	23
Tabel 4.5 Gambar Hasil Pengujian Dense Depth.....	24
Tabel 4.6 Hasil Pengujian MiDaS.....	25
Tabel 4.7 Gambar Hasil Pengujian MiDaS	26
Tabel 4.8 Perbandingan Depth Image.....	27

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Deteksi rambu lalu lintas telah menjadi masalah tradisional untuk kendaraan cerdas, terutama sebagai langkah pendahuluan untuk pengenalan rambu lalu lintas yang memberikan informasi berguna seperti petunjuk arah dan peringatan untuk *autonomus driving* atau *driver assistance systems*[1]. Saat ini, deteksi rambu lalu lintas telah mendapatkan perhatian dalam sistem kendaraan cerdas, dimana rambu lalu lintas dapat digunakan sebagai petunjuk berbeda untuk menunjukkan pemetaan dan lokalisasi. Berbeda dari petunjuk alam seperti titik sudut atau tepi yang memiliki tampilan sembarangan, rambu lalu lintas memiliki tampilan standar seperti bentuk, warna, dan pola yang ditentukan oleh peraturan ketat.

Deteksi rambu lalu lintas memainkan peran penting dalam *Driver Assistance System* dan *Automated Driving*. Namun, tugas ini tidak mudah untuk sebuah komputer karena variasi besar dalam tampilan visual gambar rambu lalu lintas karena oklusi parsial, sudut pandang yang berbeda, iluminasi dan kondisi cuaca. Untuk mengenali rambu lalu lintas pada sebuah gambar, metode paling populer mencakup dua langkah: Deteksi dan Klasifikasi[2].

Rambu lalu lintas kini dideteksi menggunakan convolutional neural network (CNN)[3], sebuah metode pembelajaran mendalam yang menggunakan konvolusi dengan menggerakkan pengganda konvolusi (filter) dengan ukuran tertentu di atas gambar dan komputer menggunakannya untuk mendapatkan informasi representatif baru. Metode jaringan saraf manusia (CNN) digunakan untuk memungkinkan pemrosesan informasi visual untuk meniru sistem pengenalan gambar dari korteks visual manusia. Namun, metode ini memiliki kelemahan dalam membentuk model lama. Oleh karena itu, untuk mempercepat dan menyederhanakan proses komputasi, penelitian ini menggunakan arsitektur jaringan saraf YOLO untuk memastikan dapat bekerja dengan baik pada perangkat kelas bawah.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang dibahas adalah sebagai berikut.

1. Bagaimana cara mendeteksi rambu lalu lintas menggunakan arsitektur?
2. Bagaimana cara mengklasifikasikan gambar atau citra rambu-rambu lalu lintas?
3. Bagaimana cara mengetahui estimasi jarak antara kendaraan dan rambu lalu lintas yang terbaca?

1.3. Tujuan

Adapun tujuan dan manfaat yang diperoleh dari penelitian ini adalah.

1. Membuat model deteksi rambu lalu lintas menggunakan arsitektur YOLO.
2. Membuat klasifikasi citra dari jenis-jenis rambu lalu lintas tertentu.
3. Membuat sistem estimasi jarak yang dapat memprediksi jarak antar kendaraan dengan rambu lalu lintas.

1.4. Batasan Masalah

Untuk membatasi meluasnya bahasan masalah yang akan diteliti, maka dibatasi masalah yang berkaitan dengan perancangan dan implementasi sistem ini, yaitu sebagai berikut :

1. Penelitian ini menggunakan metode CNN (*Convolutional Neural Network*) dengan arsitektur YOLOv4.
2. Penelitian ini mendeteksi dan mengenali rambu-rambu lalu lintas.
3. Penelitian ini memanfaatkan *dataset* yang sudah tersedia.
4. Klasifikasi rambu lalu lintas yang digunakan terdapat sembilan kelas, yaitu: lurus, belok kanan, belok kiri, berhenti, persimpangan, penyeberangan, bundaran, hati-hati dan lajur dua arah.

1.5. Sistematika Penulisan

Pada penulisan Tugas Akhir ini, dibagi dalam beberapa bagian sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi penjelasan gambaran secara umum tentang Tugas Akhir yang dikerjakan. Bab ini terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan.

BAB 2 DASAR TEORI

Bab ini berisi tentang landasan teori dan sumber-sumber terkait sebagai penunjang dalam penelitian.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas mengenai pemodelan sistem, analisis sistem, dan perancangan aplikasi web tes gangguan perilaku.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas proses pengujian yang dilakukan pada sistem dan hasil pengujian akan dianalisis agar dapat dilakukannya penarikan kesimpulan dari penelitian yang telah dilakukan.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan akhir dari penelitian Tugas Akhir ini dan saran untuk Tugas Akhir ini.

LAMPIRAN

BAB II

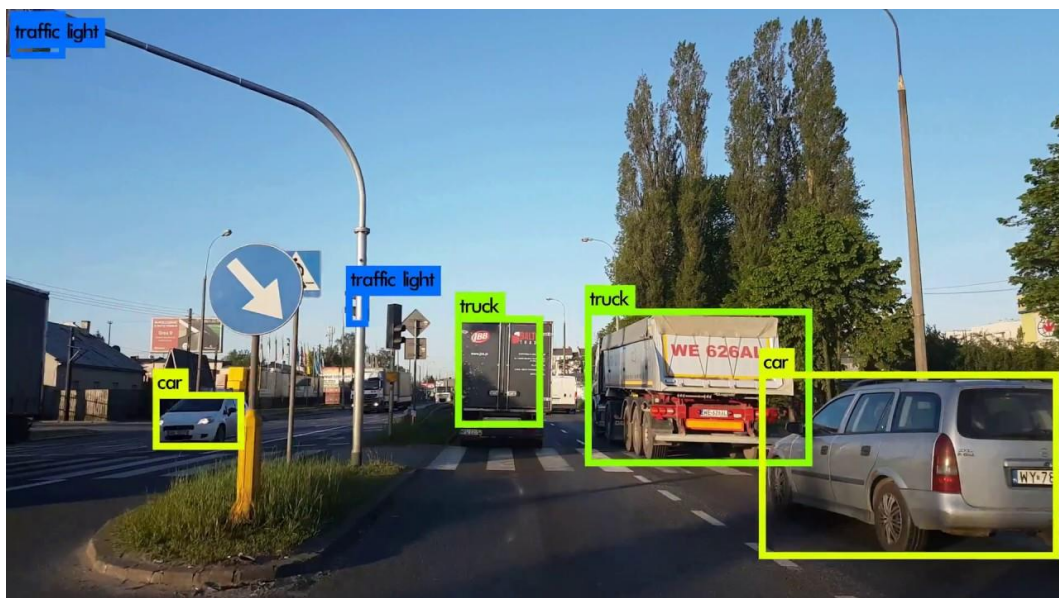
TINJAUAN PUSTAKA

Penelitian ini terbagi menjadi beberapa aspek yaitu: *Image Recognition*, *Image Classification* dan *Neural Network*.

2.1. Image Recognition

Pengenalan citra adalah sebuah proses mengidentifikasi dan mendeteksi objek atau fitur dalam citra atau video digital [4]. Konsep seperti ini digunakan sebagai langkah untuk melakukan sejumlah besar tugas penglihatan berbasis mesin, seperti menandai konten gambar dengan tag meta, mencari konten gambar, dan memandu robot otonom, kendaraan otonom, dan sistem penghindaran tabrakan[5].

Sebagai bagian dari proses pembelajaran, pengenalan gambar bekerja dengan belajar melacak hubungan atau pola dalam gambar yang diberikan. Setelah proses pelatihan selesai, kinerja sistem akan diverifikasi berdasarkan data uji.

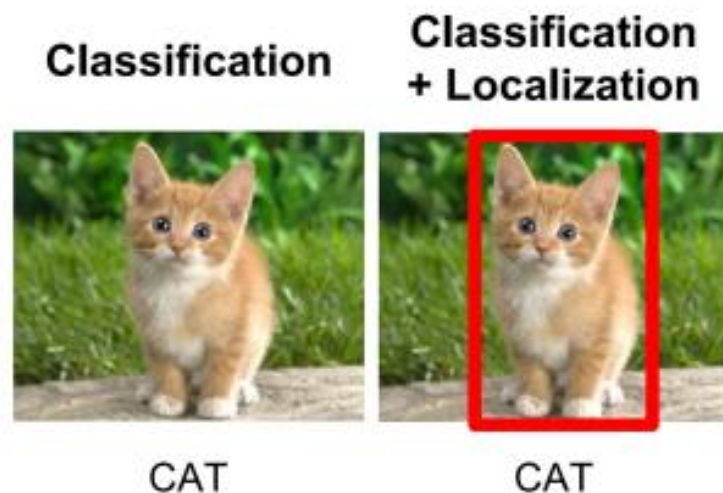


Gambar 2.1 Image Recognition[6]

Gambar 2.1 mengilustrasikan hasil dari proses *image recognition* yang digunakan untuk mendeteksi kendaraan dan rambu lalu lintas.

2.2. Image Classification

Klasifikasi citra terdiri dari pengklasifikasian citra ke dalam kategori tertentu[7]. Klasifikasi citra juga erat kaitannya dengan positioning citra, yaitu perkembangan klasifikasi citra, keluaran yang dihasilkan bukan lagi kategori-kategori citra, melainkan posisi objek kategori gambar dalam gambar, yang biasanya berupa kotak pembatas.



Gambar 2.2 Image Classification[8]

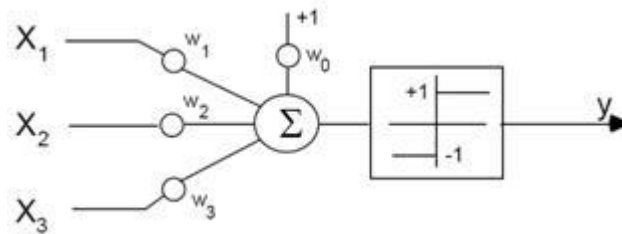
Gambar 2.2 menunjukkan hasil dari *image classification* dengan menggunakan *bounding box* pada objek yang berhasil terdeteksi.

2.3. Neural Network

Jaringan syaraf tiruan atau *Neural Network* merupakan salah satu kategori ilmu komputasi lunak. Jaringan saraf sebenarnya memanfaatkan kekuatan otak manusia untuk memberikan stimulasi, pemrosesan, dan keluaran. Keluaran tersebut diperoleh dari rangsangan dan proses perubahan yang berubah pada otak manusia[9].

Perkembangan ilmu jaringan syaraf tiruan dimulai pada tahun 1943, saat Warren McCulloch beserta dengan Walter Pitts memperkenalkan model jaringan syaraf komputasional yang pertama[10]. Mereka menggabungkan beberapa unit pemrosesan sederhana untuk memberikan peningkatan daya komputasi secara keseluruhan. Selanjutnya, Rosenblatt melakukan survei di tahun 1950, saat itu

Rosenblatt berhasil menemukan jaringan dua lapis yang disebut perceptron. Perceptron melakukan proses klasifikasi dengan menambahkan bobot ke setiap koneksi jaringan.



Gambar 2.3 *Perceptron*[11]

Gambar 2.3 menggambarkan perceptron yang memungkinkan beberapa klasifikasi pembelajaran bekerja dengan menambahkan bobot ke setiap koneksi antar jaringan. Perceptrons telah berhasil menjadi dasar untuk penelitian masa depan di bidang jaringan saraf. Penelitian tentang jaringan saraf mulai berkembang pada awal 1980-an, dan para peneliti menemukan banyak bidang minat baru di bidang ilmu jaringan saraf.

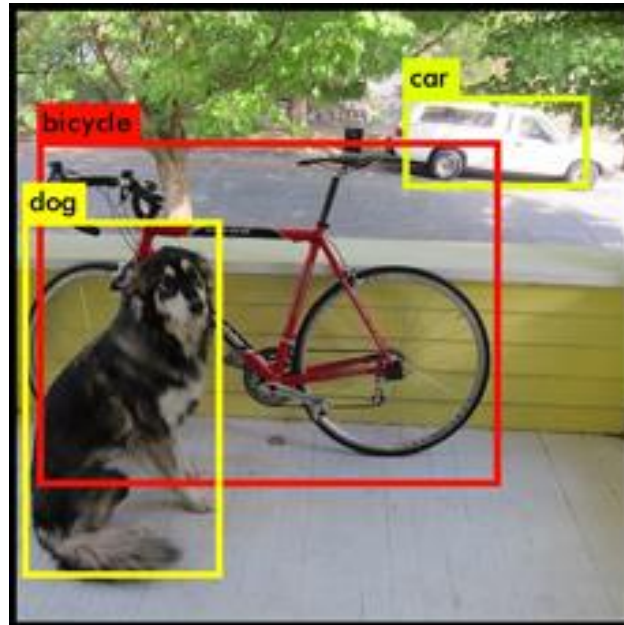
Sekarang, jaringan neuronal dapat diterapkan pada berbagai tugas, seperti klasifikasi, kesadaran, pendekatan, prediksi, pengelompokan, simulasi memori dan banyak tugas lain dari berbagai tugas.

2.4. YOLO

YOLO (*You Only Look Once*) dibuat pertama kali pada tahun 2015 oleh Joseph Redmon. Sistem ini adalah sistem deteksi objek *real-time* berbasis CNN (*Convolutional Neural Network*)[12]. Pada konferensi CVPR (*Conference on Computer Vision and Pattern Recognition*) tahun 2017, Joseph Redmon dan Ali Farhadi merilis YOLO v2, yang meningkatkan akurasi dan kecepatan algoritma[13]. Joseph Redmon dan Ali Farhadi meluncurkan YOLO v3 versi terbaru pada bulan April tahun 2018, yang meningkatkan kinerja deteksi objek[14].

Algoritma Yolo adalah salah satu algoritma untuk mendeteksi objek yang sangat unik. Ini akan memproses dan memproses gambar hanya sekali. Sesuai

dengan namanya, *You Only Looked Once*. Yolo adalah algoritma berdasarkan regresi (regresi) yang mengeksekusi proses eksekusi untuk klasifikasi setiap objek dan output prediktif dari kotak pembatas. Algoritma Yolo berjalan dengan *framework* darknet[15] dan darkflow dan didukung oleh GPU.



Gambar 2.4 Objek Deteksi *YOLO*[16]

Gambar 2.4 mengilustrasikan deteksi objek menggunakan arsitektur *YOLO* yang menghasilkan gambar objek yang sudah diklasifikasi dan ditandai dengan *bounding box*.

2.5. Depth Estimation

Mengukur jarak objek relatif terhadap kamera masih sulit, tetapi sangat penting untuk membuka kunci aplikasi keren seperti mengemudi otonom, rekonstruksi adegan 3D, dan augmented reality (AR)[17]. Dalam robotika, kedalaman adalah prasyarat utama untuk tugas-tugas seperti persepsi, navigasi, dan perencanaan.

Dalam penelitian ini, estimasi kedalaman digunakan untuk mengukur jarak antara rambu lalu lintas dengan jarak kendaraan. Algoritma estimasi kedalaman yang digunakan adalah Dense Depth[18] dan MiDAS (*Mixing Datasets for Zeroshot Crossdataset Transfer*)[19].

BAB III PERANCANGAN SISTEM

3.1. Gambaran Umum Sistem

Berikut adalah diagram alir sistem yang akan dibuat terlampir pada Gambar 3.1.



Gambar 3.1 Gambaran Umum Sistem

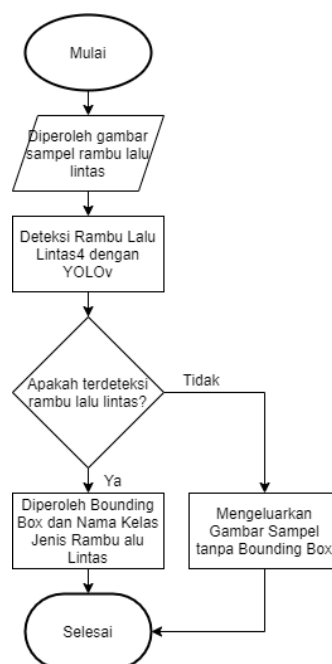
Kondisi pada saat dilakukan deteksi objek dan estimasi jarak difokuskan pada kondisi cahaya yang baik yaitu pada pagi hari dan siang hari, posisi kamera diletakkan pada dashboard mobil.

Algoritma yang digunakan pada deteksi objek adalah YOLO (*You Only Look Once*), memiliki tingkat akurasi dan kecepatan proses yang tinggi, sehingga diharapkan algoritma ini dapat berjalan dengan baik pada saat dijalankan secara *real-time*.

Gambar 3.1, sistem dimulai dengan menangkap gambar dengan format tiga channel yaitu format RGB (*Red, Green, Blue*), proses selanjutnya adalah melakukan proses deteksi objek dan menentukan jenis objek. Setelah diperoleh keluaran berupa jenis objek yang terdeteksi, dan koordinat objek sebagai tujuan untuk lokalisasi gambar. Proses terakhir adalah menggabungkan beberapa output yaitu, koordinat, label objek yang terdeteksi dan jarak pada objek yang terdeteksi yang diperlukan untuk menampilkan hasil kepada pengguna.

3.1.1. Proses Deteksi Rambu Lalu Lintas

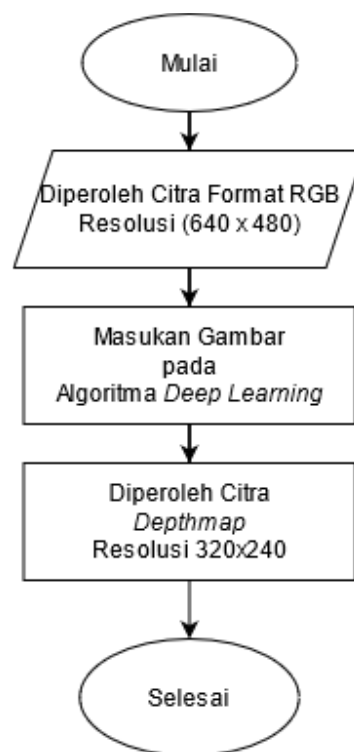
Berikut adalah *flowchart* proses deteksi rambu lalu lintas menggunakan YOLOv4.



Gambar 3.2 Proses Deteksi Objek

Gambar 3.2 menampilkan proses deteksi rambu lalu lintas menggunakan YOLOv4, dimulai dari memasukkan gambar sampel rambu lalu lintas, kemudian YOLO akan mendeteksi rambu lalu lintas pada gambar sampel yang telah dimasukkan. Jika tidak terdapat rambu lalu lintas, sistem akan mengeluarkan gambar tanpa *bounding box*, jika sampel terdeteksi rambu lalu lintas maka gambar akan dikeluarkan dengan *bounding box* pada gambar yang menunjukkan jenis rambu lalu lintas yang terdeteksi.

3.1.2. Proses Depth Prediction



Gambar 3.3 Proses Depth Prediction

Gambar 3.3 menunjukkan proses *depth prediction*, tujuan dari proses ini adalah untuk mengetahui estimasi jarak objek terhadap kamera. Sistem dimulai dari memperoleh citra RGB dengan resolusi sebesar 640 x 480, citra RGB tersebut dimasukkan ke dalam algoritma deep learning yang khusus untuk memecahkan masalah depth prediction, hasil dari prediksi ini berupa matriks dengan dimensi 320 x 240 hal ini bertujuan untuk mengurangi beban komputasi pada saat proses upsampling citra, dikarenakan pada proses konvolusi pada algoritma deep learning, proses ini mengurangi dimensi data masukan. Hasil dari prediksi ini disebut *Depth Map* berupa matriks, matriks ini berisi kumpulan-kumpulan jarak dalam satuan meter.

3.2. Analisis Sistem

Analisis sistem yang dilakukan pada sistem deteksi rambu lalu lintas dan estimasi jarak terbagi menjadi empat yaitu analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, analisis kebutuhan pengguna, dan analisis kebutuhan data.

3.2.1. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam perancangan dan implementasi sistem ini memiliki spesifikasi sebagai berikut.

1. Google Colab
2. Darknet *framework*
3. Open CV
4. Numpy
5. TensorFlow
6. Github
7. Matplotlib

3.2.2. Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam perancangan dan implementasi sistem ini memiliki spesifikasi sebagai berikut.

1. GPU Tesla T4(Colab GPU)
2. CPU Intel® Xeon® CPU @ 2.00GHz (Colab CPU)

3.2.3. Analisis Kebutuhan Pengguna

Pengguna sistem ini diharapkan adalah pengguna kendaraan bermotor jenis mobil.

3.2.4. Analisis Kebutuhan Data

Kebutuhan data yang diperlukan oleh sistem agar dapat memberikan *output* yang sesuai dengan tujuan sistem. Data yang dibutuhkan adalah dataset rambu lalu lintas dengan sembilan jenis rambu lalu lintas yang ada di Indonesia. Rambu lalu lintas yang digunakan dipilih berdasarkan tingkat krusial yang berpengaruh pada pengguna jalan.

Berikut adalah jenis-jenis rambu lalu lintas yang digunakan pada sistem deteksi rambu lalu lintas.



Gambar 3.4 Jenis Rambu Lalu Lintas

Gambar 3.4 menunjukkan jenis-jenis rambu lalu lintas yang digunakan pada sistem objek deteksi rambu lalu lintas. Berdasarkan Gambar 3.4 jenis rambu lalu lintas yang digunakan adalah sepuluh jenis rambu lalu lintas yang ada di Indonesia. Jenis rambu lalu lintas tersebut adalah sebagai berikut.

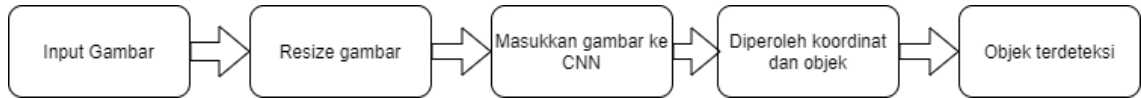
1. Lajur Dua Arah
2. Penyeberangan Kuning
3. Hati-hati
4. Belok Kanan
5. Belok Kiri
6. Penyeberangan Biru
7. Persimpangan (*Give Away*)
8. Berhenti
9. Lurus.
10. Bundaran.

3.3. Perancangan Sistem

Perancangan sistem deteksi rambu lalu lintas, sistem dimodelkan menggunakan blok diagram.

3.3.1. Blok Diagram

Berikut adalah blok diagram sistem deteksi rambu lalu lintas.

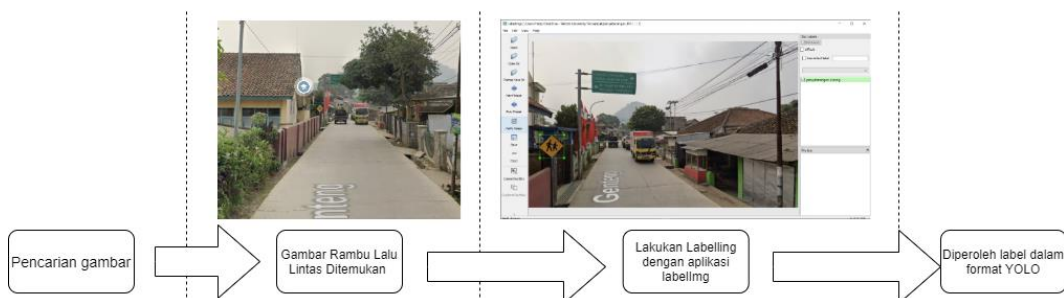


Gambar 3.5 Blok Diagram Sistem

Gambar 3.5 menggambarkan cara kerja sistem yang akan dibangun dimulai dari input gambar kemudian gambar diolah atau di-*resize* untuk mengurangi beban pada saat proses komputasi berlangsung. Setelah dilakukan penyesuaian, kemudian gambar dimasukkan ke dalam algoritma CNN dan kemudian diperoleh koordinat dan objek yang terdeteksi. Keluaran yang didapatkan akan berupa gambar dengan *bounding box* yang menunjukkan bahwa objek sudah terdeteksi.

3.3.2. Penyusunan Dataset

Dataset yang digunakan pada penelitian ini adalah dataset hasil kustomisasi manual dari pencarian menggunakan fitur *street view* dari aplikasi Google Maps. Dataset dicari dengan menelusuri jalan raya, jalan tol, dan jalan perkampungan yang mempunyai jenis rambu lalu lintas sesuai dengan kebutuhan penelitian ini. Proses penyusunan dataset dapat dilihat pada Gambar 3.6 berikut ini.



Gambar 3.6 Proses Penyusunan Dataset

Gambar 3.6 menunjukkan proses penyusunan dataset yang dimulai dari pencarian gambar rambu lalu lintas dengan menelusuri Google Maps, setelah gambar diperoleh, dilakukan *labelling* sesuai dengan nama rambu lalu lintas pada

gambar yang ditemukan menggunakan aplikasi LabelImg, setelah selesai labelling akan diperoleh label berbentuk format YOLO.

3.3.3. Pembuatan Replika Rambu Lalu Lintas

Pembuatan replika rambu lalu lintas bertujuan untuk validasi data pada pengujian estimasi jarak. Replika dibuat dengan tiang kayu sepanjang 2 meter dan gambar rambu lalu lintas dicetak menggunakan bahan kain spanduk (*Tetoron Cotton*).



Gambar 3.7 Replika Rambu Lalu Lintas

Gambar 3.7 menunjukkan gambar replika rambu lalu lintas yang dibuat dengan kain TC(*Tetoron Cotton*) dan dicetak di sebuah percetakan spanduk di daerah Batujajar.

3.3.4. Validasi Data Estimasi Jarak

Validasi data estimasi jarak menggunakan replika rambu lalu lintas yang sudah dipasang pada tiang kayu setinggi 2 meter. Pengambilan data validasi dilakukan di bahu jalan pada jam 16.00 sampai jam 17.30.



Gambar 3.8 Pengambilan Data Validasi Jarak

Gambar 3.8 menunjukkan proses pengambilan data validasi jarak menggunakan replika rambu lalu lintas. Pengambilan data validasi jarak dilakukan dengan menarik alat ukur meteran pita sepanjang 20 meter kemudian gambar replika rambu lalu lintas diambil berdasarkan empat jarak yaitu, 5 meter, 10 meter, 15 meter, dan 20 meter.

3.4 Perancangan Pengujian

Perancangan pengujian dibagi menjadi dua bagian meliputi pengujian deteksi rambu lalu lintas dan pengujian estimasi jarak.

3.4.1 Pengujian Deteksi Rambu Lalu Lintas

Pengujian deteksi objek meliputi pengujian akurasi deteksi rambu lalu lintas dan pengujian waktu deteksi rambu lalu lintas. Pengujian akurasi deteksi rambu lalu lintas diuji berdasarkan nilai akurasi yang terdapat pada *bounding box* pada *output* sampel. Pengujian akurasi deteksi rambu lalu lintas dilakukan

sebanyak sepuluh kali menggunakan 23 sampel gambar rambu lalu lintas yang berbeda. Pengujian waktu deteksi rambu lalu lintas diuji berdasarkan waktu sistem untuk mendeteksi rambu lalu lintas pada sampel gambar rambu lalu lintas yang dimasukkan.

3.4.2 Pengujian Estimasi jarak

Pengujian estimasi jarak meliputi dua pengujian yaitu pengujian estimasi jarak menggunakan Dense Depth dan pengujian estimasi jarak menggunakan MiDaS. Kedua pengujian ini dilakukan dengan membandingkan data jarak sebenarnya pada sampel dengan jarak prediksi yang didapatkan dari *depth image* yang berisi matriks-matriks jarak yang diproses oleh Dense Depth dan MiDaS. Setelah jarak sebenarnya dan jarak prediksi dibandingkan akan diperoleh *error* dari hasil perbandingan jarak sebenarnya dan jarak prediksi.

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi

Implementasi sistem deteksi rambu lalu lintas meliputi implementasi deteksi rambu lalu lintas dan implementasi estimasi jarak pada rambu lalu lintas.

4.1.1. Implementasi Deteksi Rambu Lalu Lintas

Implementasi deteksi rambu lalu lintas adalah implementasi *ouput* berupa *bounding box* pada gambar sampel yang terdeteksi memiliki rambu lalu lintas tertentu yang dihasilkan dari model yang telah di-*training* menggunakan dataset hasil kustomisasi dan labeling yang diperoleh dari pencarian menggunakan fitur *street view* pada Google Maps.

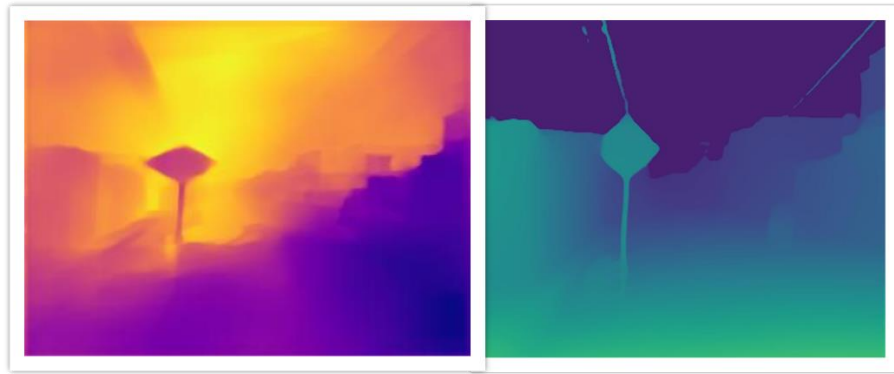


Gambar 4.1 Implementasi Deteksi Rambu Lalu Lintas

Gambar 4.1 menunjukkan hasil implementasi deteksi rambu lalu lintas pada gambar sampel yang memiliki rambu lalu lintas jenis bundaran. Gambar sampel yang sudah terdeteksi akan memiliki *bounding box* pada rambu lalu lintas yang terdapat pada gambar.

4.1.2. Implementasi Estimasi Jarak

Implementasi estimasi jarak menggunakan dua arsitektur yang berbeda, yaitu Dense Depth dan MiDaS(*mixing datasets for zero-shot cross-dataset transfer*).



Gambar 4.2 Depth Image

Gambar 4.2 menunjukkan hasil implementasi estimasi jarak rambu lalu lintas yang berupa *depth image*. Hasil yang berupa *depth image* kemudian dikonversikan kedalam matriks-matriks yang merepresentasikan jarak objek dalam satuan meter.

4.2. Pengujian Sistem

Pada penelitian ini, pengujian sistem yang dilakukan adalah pengujian akurasi deteksi rambu lalu lintas, pengujian waktu deteksi rambu lalu lintas, dan pengujian estimasi jarak.

4.2.1. Pengujian Deteksi Rambu Lalu Lintas

Pengujian deteksi rambu lalu lintas meliputi dua skenario pengujian, yaitu pengujian akurasi deteksi rambu lalu lintas dan pengujian waktu deteksi rambu lalu lintas.

4.2.1.1. Pengujian Akurasi Deteksi Rambu Lalu Lintas

Pengujian akurasi deteksi pada deteksi objek diuji menggunakan dataset hasil kustomisasi yang didapatkan dari beberapa dataset yang tersedia pada penelitian sebelumnya dan kumpulan dataset yang didapatkan dari hasil pencarian menggunakan aplikasi *google street view* di jalanan Indonesia. Berikut adalah tabel hasil pengujian yang dilakukan terhadap sembilan kelas rambu lalu lintas.

Tabel 4.1 Akurasi Pengujian Deteksi Objek

Jenis Rambu	Akurasi
Belok Kanan	100%
Belok Kiri	100%
Lurus	98%
Berhenti	73%
Hati-hati	100%
Persimpangan	100%
Penyeberangan	98%
Lajur Dua Arah	95%
Bundaran	100%

Tabel 4.1 menunjukkan hasil pengujian akurasi dan waktu deteksi system saat mendeteksi rambu lalu lintas. Sampel rambu lalu lintas yang diuji menggunakan sembilan jenis rambu lalu lintas yang sudah ditentukan yaitu, belok kanan, belok kiri, lurus, berhenti, hati-hati, persimpangan, penyeberangan, lajur dua arah dan bundaran. Setiap sampel diuji sebanyak sepuluh kali dan diambil rata-rata dari akurasi setiap pengujian. Pengujian waktu deteksi diuji berdasarkan waktu sistem saat mendeteksi rambu lalu lintas. Sampel diuji sebanyak sepuluh kali dan didapatkan waktu deteksi dari rata-rata setiap pengujian dan dinyatakan dalam satuan *millisecond*.

Hasil pengujian pada tabel 4.1 menunjukkan akurasi yang didapatkan dari pengujian masing-masing sampel pada dataset. Jika dirata-ratakan setiap persentase akurasi dari setiap jenis sampel akan didapatkan nilai persentase akurasi sebesar 96% dengan perhitungan sebagai berikut.




$$\begin{aligned}
 \text{rata - rata persentase akurasi} &= \frac{\text{jumlah persentase setiap sampel}}{\text{jumlah sampel}} \\
 &= \frac{100 + 100 + 98 + 73 + 100 + 100 + 98 + 95 + 100}{9}
 \end{aligned}$$

$$= 96\%$$

Perhitungan di atas digunakan untuk mendapatkan persentase rata-rata akurasi deteksi rambu lalu lintas, dapat disimpulkan bahwa akurasi deteksi objek adalah 96%.

Berikut ini adalah gambar hasil pengujian akurasi deteksi rambu lalu lintas yang telah diuji pada *platform* Google Colab.

Tabel 4.2 Gambar Hasil Deteksi Rambu Lalu Lintas

Jenis Rambu	Gambar
Penyeberangan(kuning)	
Belok Kanan	
Belok Kiri	

Tabel 4.2 menyajikan gambar hasil pengujian deteksi rambu lalu lintas. Tabel tersebut menyajikan tiga jenis rambu lalu lintas dari total sepuluh jenis rambu lalu

lintas yang menjadi sampel uji. Gambar sampel didapatkan dari hasil penelusuran fitur *street view* pada aplikasi Google Maps.

4.2.1.2. Pengujian Waktu Deteksi Rambu Lalu Lintas

Pengujian akurasi dan waktu deteksi pada deteksi objek diuji menggunakan dataset hasil kustomisasi yang didapatkan dari beberapa dataset yang tersedia pada penelitian sebelumnya dan kumpulan dataset yang didapatkan dari hasil pencarian menggunakan aplikasi *google street view* di jalanan Indonesia. Berikut adalah tabel hasil pengujian yang dilakukan terhadap sembilan kelas rambu lalu lintas.

Tabel 4.3 Pengujian Waktu Deteksi Rambu Lalu Lintas

Jenis Rambu	Waktu Deteksi (ms)
Belok Kanan	32.83
Belok Kiri	32.74
Lurus	32.75
Berhenti	32.75
Hati-hati	32.75
Persimpangan	32.79
Penyeberangan	32.71
Lajur Dua Arah	32.75
Bundaran	32.73

Pengujian waktu deteksi diuji berdasarkan waktu sistem saat mendeteksi adanya rambu lalu lintas sampai dengan mengeluarkan output gambar disertai dengan *bounding box*. Pengujian dilakukan sebanyak sepuluh kali dan menggunakan satuan *millisecond*. Hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “belok kanan” diperoleh waktu deteksi sebesar 32.83ms, hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “belok kiri” diperoleh waktu sebesar 32.74ms, hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “lurus” diperoleh waktu deteksi sebesar 32.75ms, hasil

pengujian waktu deteksi yang diperoleh pada pengujian sampel rambu lalu lintas jenis “berhenti” sebesar 32.75ms, hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “hati-hati” diperoleh waktu deteksi sebesar 32.75ms, hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “persimpangan” diperoleh waktu deteksi sebesar 32.79ms, hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “penyeberangan” diperoleh waktu deteksi sebesar 32.71 ms, hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “lajur dua arah” diperoleh waktu deteksi sebesar 32.75ms dan hasil pengujian waktu deteksi pada sampel rambu lalu lintas jenis “bundaran” diperoleh waktu deteksi sebesar 32.73 ms.

Hasil dari setiap pengujian waktu deteksi dihitung rata-ratanya menggunakan perhitungan yang sama seperti perhitungan rata-rata persentase akurasi yang hasilnya sebagai berikut.

$$\begin{aligned} \text{rata - rata waktu deteksi} &= \frac{\text{jumlah waktu deteksi setiap sampel}}{\text{jumlah sampel}} \\ &= \frac{32.83 + 32.74 + 32.75 + 32.75 + 32.75 + 32.79 + 32.71 + 32.75 + 32.73}{9} \\ &= 32.76 \end{aligned}$$

Hasil perhitungan menggunakan rumus di atas menunjukkan bahwa rata-rata waktu deteksi sistem untuk mendeteksi rambu lalu lintas pada sembilan sampel adalah 32.76 *millisecond*.

4.2.2 Pengujian Estimasi Jarak

Pengujian metode estimasi jarak menggunakan algoritma *dense depth* dan MiDaS. Pengukuran jarak sebenarnya menggunakan alat ukur meteran pita yang memiliki panjang maksimal 30 meter.

Pengukuran jarak sebenarnya menggunakan empat parameter jarak yaitu, 5 meter, 10 meter, 15 meter dan 20 meter. Pengukuran jarak dilakukan menggunakan sepuluh jenis sampel rambu lalu lintas yang berbeda, yaitu, belok kanan, belok kiri, lurus, hati-hati, berhenti, persimpangan, lajur dua arah, penyeberangan(warna biru), penyeberangan(warna kuning) dan bundaran. Proses

pengambilan gambar dilakukan pada waktu sore hari pada pukul 16.30 WIB dan berakhir pada pukul 17.30 WIB.

4.2.2.1. Pengujian Estimasi Jarak Menggunakan Dense Depth

Pengujian estimasi jarak menggunakan *Dense Depth* dan *pre-trained model* Kitti. Pengukuran jarak sebenarnya dilakukan menggunakan alat ukur meteran pita yang memiliki panjang maksimal 30 meter. Sampel pengukuran jarak sebenarnya diuji menggunakan sepuluh jenis rambu lalu lintas belok kanan, belok kiri, lurus, hati-hati, berhenti, persimpangan, lajur dua arah, penyeberangan(warna biru), penyeberangan(warna kuning) dan bundaran. Pengukuran dilakukan dengan berpatokan pada titik nol alat ukur yang disejajarkan dengan titik tengah tiang rambu lalu lintas. Hasil pengukuran jarak sebenarnya kemudian dibandingkan dengan hasil prediksi pada sistem yang telah dibuat dan disajikan pada tabel di bawah ini.

Tabel 4.4 Pengujian Estimasi Jarak

No	Jarak Sebenarnya	Jarak Prediksi	Absolute Error
1	5 meter	8.1788 meter	3.17 meter
2	10 meter	13.834 meter	3.83 meter
3	15 meter	16.601 meter	1.60 meter
4	20 meter	21.47 meter	1.47 meter
Average Error			2.52 meter



Tabel 4.2 menyajikan data hasil perbandingan antara pengukuran jarak sebenarnya dan pengukuran jarak hasil prediksi *dense depth* yang telah diuji menggunakan sampel rambu lalu lintas jenis “belok kanan”. Perbandingan hasil pengukuran jarak pada jarak sebenarnya dengan jarak hasil prediksi pada jarak 5 meter memiliki perbandingan sebesar 3.177 meter yang berarti hasil prediksi memiliki galat sebesar 3.177 meter. Hasil perbandingan pengukuran pada jarak 10

meter memiliki perbedaan sebesar 3.834 meter, maka hasil pengujian jarak prediksi memiliki galat sebesar 3.834 meter. Hasil pengujian prediksi jarak pada jarak 15 meter memiliki perbandingan sebesar 1.601 meter, maka hasil prediksi jarak memiliki galat sebesar 1.601 meter. Hasil pengujian prediksi jarak pada jarak 20 meter memiliki perbandingan sebesar 1.47 meter, maka hasil prediksi memiliki galat sebesar 1.47 meter. Galat pada setiap pengujian kemudian dicari rata-ratanya menggunakan perhitungan berikut ini.

$$\begin{aligned} \text{average error} &= \frac{\text{jumlah error setiap pengujian}}{\text{jumlah sampel jarak}} \\ &= \frac{3.177 + 3.834 + 1.601 + 1.47}{4} \\ &= 2.5205 \end{aligned}$$

Hasil perhitungan di atas menghasilkan nilai 2.5205 meter, maka rata-rata galat yang diperoleh adalah 2.5205 meter dari empat pengujian yang sudah dilakukan.

Tabel 4.5 Gambar Hasil Pengujian Dense Depth

Gambar sampel	Gambar <i>Depth</i>
	

Tabel 4.5 menyajikan gambar sampel yang diambil menggunakan kamera *handphone* sebelum diubah menjadi *depth image*. Gambar sampel pada Tabel 4.5 merupakan sampel gambar yang diambil dari jarak 5 meter, 10 meter, 15 meter, dan 20 meter.

4.2.2.2. Pengujian Estimasi Jarak Menggunakan MiDaS

Pengujian estimasi jarak menggunakan MiDaS menggunakan 10 dataset yang digabungkan yaitu RedWeb, DIML, Movies, MegaDepth, WSVD, TartanAir, HRWSI, ApolloScape, BlendedMVS dan IRS dengan optimisasi multi objek. Pada awalnya MiDaS menggunakan lima campuran dataset yaitu DIW, KITTI, NYU, ETH3D dan TUM, kemudian dirilis MiDaSv2.1 yang memiliki akurasi 21% lebih akurat daripada model sebelumnya yang menggunakan lima campuran dataset.

Tabel 4.6 Hasil Pengujian MiDaS







No	Jarak Sebenarnya	Jarak Prediksi	Absolute Error
1	5 meter	5.53 meter	0.53
2	10 meter	9.75 meter	0.25
3	15 meter	9.75 meter	5.25
4	20 meter	9.75 meter	10.25
Average Error			4.07

Tabel 4.5 menyajikan data hasil perbandingan antara pengukuran jarak sebenarnya dan pengukuran jarak hasil prediksi *dense depth* yang telah diuji menggunakan sepuluh jenis sampel rambu lalu lintas. Perbandingan hasil pengukuran jarak pada jarak sebenarnya dengan jarak hasil prediksi pada jarak 5 meter memiliki perbandingan sebesar 0.53 meter yang berarti hasil prediksi memiliki galat sebesar 0.53 meter. Hasil perbandingan pengukuran pada jarak 10 meter memiliki perbedaan sebesar 0.25 meter, maka hasil pengujian jarak prediksi memiliki galat sebesar 0.25 meter. Hasil pengujian prediksi jarak pada jarak 15 meter memiliki perbandingan sebesar 5.25 meter, maka hasil prediksi jarak memiliki galat sebesar 5.25 meter. Hasil pengujian prediksi jarak pada jarak 20 meter memiliki perbandingan sebesar 10.25 meter, maka hasil prediksi memiliki galat sebesar 10.25 meter. Galat pada setiap pengujian kemudian dicari rata-ratanya menggunakan perhitungan berikut ini.

$$\begin{aligned}
 \text{average error} &= \frac{\text{jumlah error setiap pengujian}}{\text{jumlah sampel jarak}} \\
 &= \frac{0.53 + 0.25 + 5.25 + 10.25}{4} \\
 &= 4.07
 \end{aligned}$$

Hasil perhitungan di atas menghasilkan nilai 4.07 meter, maka rata-rata galat yang diperoleh adalah 4.07 meter dari empat pengujian yang sudah dilakukan.

Tabel 4.7 Gambar Hasil Pengujian MiDaS

Gambar Sampel	Gambar <i>Depth</i>
	
	
	




Tabel 4.6 menyajikan gambar sampel yang diambil menggunakan kamera *handphone* sebelum diubah menjadi *depth image*. Gambar sampel pada Tabel 4.6 merupakan sampel gambar yang diambil dari jarak 5 meter, 10 meter, 15 meter, dan 20 meter.

4.2.2.3. Perbandingan Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan pada sub-bab sebelumnya, Dense Depth memiliki rata-rata *error* 2.52 meter dan MiDaS memiliki rata-rata *error* 4.07 meter. Maka dapat disimpulkan bahwa Dense Depth memiliki rata-rata *error* lebih kecil dari MiDaS.

Berikut adalah tabel perbandingan *depth image* dari Dense Depth dan MiDaS pada Tabel 4.7.

Tabel 4.8 Perbandingan Depth Image

Gambar Sampel	Dense Depth	MiDaS
		

Tabel 4.7 menunjukkan hasil konversi gambar sampe menjadi *depth image* dari Dense Depth dan MiDaS. Berdasarkan pengujian yang telah dilakukan, semakin jauh jarak yang diuji maka semakin kecil *error* yang diperoleh oleh Dense Depth. Sebaliknya, pada pengujian MiDaS semakin jauh jarak yang diuji, semakin besar *error* yang diperoleh.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada pengujian persentase akurasi deteksi dan waktu deteksi pada rambu lalu lintas dan pengujian estimasi jarak, maka dapat disimpulkan bahwa:

1. Berdasarkan model yang telah dibuat, sistem dapat mendeteksi jenis-jenis rambu lalu lintas dengan waktu deteksi rata-rata 32.76 ms.
2. Berdasarkan hasil deteksi rambu lalu lintas yang telah diuji, model yang dibuat dengan mengklasifikasikan sepuluh jenis rambu lalu lintas diperoleh akurasi deteksi rata-rata sebesar 96%.
3. Berdasarkan sistem estimasi jarak yang telah diuji menggunakan arsitektur Dense Depth dan MiDaS setiap sistem estimasi jarak yang diuji mempunyai *error* terhadap jarak yang sebenarnya. Dense Depth mempunyai rata-rata *error* 2.52 meter dan MiDaS mempunyai rata-rata *error* 4.07 meter. Maka dapat disimpulkan bahwa Dense Depth memiliki rata-rata *error* lebih kecil dari MiDaS.

5.2. Saran

Berdasarkan hasil pengujian yang dilakukan dan kesimpulan yang didapatkan, untuk pengembangan lebih lanjut maka disarankan:

1. Menambah jumlah data *train* pada dataset yang akan digunakan.
2. Menambah jumlah data validasi pada dataset yang akan digunakan.
3. Menggunakan alat ukur yang presisi untuk pengukuran jarak sebenarnya untuk estimasi jarak.

DAFTAR PUSTAKA

- [1] H. S. Lee and K. Kim, "Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network," *IEEE Trans. Intell. Transp. Syst.*, 2018, doi: 10.1109/TITS.2018.2801560.
- [2] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network," *IEEE Trans. Intell. Transp. Syst.*, 2018, doi: 10.1109/TITS.2017.2714691.
- [3] A. I. Djihad, S. M. Nasution, and A. L. Prasasti, "Implementation of Illegal Parking Detection on Road Traffic using Neural Networks," vol. xx, no. xx, pp. 45–50, 2020.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, doi: 10.1109/CVPR.2016.90.
- [5] N. H. Tsani, B. Dirgantoro, and A. L. Prasasti, "Impelementasi Deteksi Kecepatan Kendaraan Menggunakan Kamera Webcam dengan Metode Frame Difference The Implementation of Vehicle Speed Detection using Webcam with Frame Difference Method," *e-Proceeding Eng.*, vol. 4, no. 2, pp. 2373–2381, 2017.
- [6] F. Shaikh, "Understanding and Building an Object Detection Model from Scratch in Python," 2018.
<https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/>.
- [7] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," 2019, doi: 10.1109/CVPR.2019.00065.
- [8] A. Ouaknine, "Review of Deep Learning Algorithms for Object Detection," *Medium*, 2018. <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852> (accessed Nov. 29, 2020).
- [9] A. L. Caterini and D. E. Chang, "Recurrent neural networks," in *SpringerBriefs in Computer Science*, 2018.
- [10] T. R. Cook, "Neural Networks," in *Advanced Studies in Theoretical and*

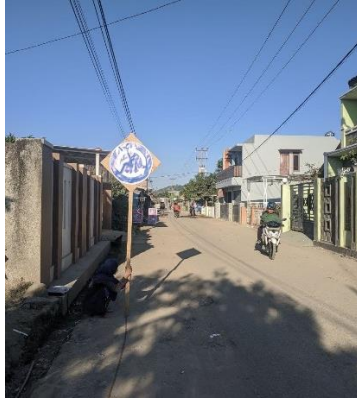
Applied Econometrics, 2020.

- [11] D. Suhartono, “Dasar Pemahaman Neural Network,” *Binus*, 2012.
<https://socs.binus.ac.id/2012/07/26/konsep-neural-network/>.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2016, doi: 10.1109/CVPR.2016.91.
- [13] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” 2017, doi: 10.1109/CVPR.2017.690.
- [14] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv*. 2018.
- [15] J. Redmon, “Darknet: Open Source Neural Networks in C,”
<http://pjreddie.com/darknet/>. .
- [16] I. Panteleyev, “How To Implement Object Recognition on Live Stream,” *IoT for All*, 2017. <https://www.iotforall.com/objects-recognition-live-stream-yolo-model>.
- [17] F. M. Dirgantara, A. S. Rohman, and L. Yulianti, “Object distance measurement system using monocular camera on vehicle,” *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, pp. 122–127, 2019, doi: 10.23919/EECSI48112.2019.8976939.
- [18] I. Alhashim and P. Wonka, “High Quality Monocular Depth Estimation via Transfer Learning,” 2018, [Online]. Available: <http://arxiv.org/abs/1812.11941>.
- [19] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020, doi: 10.1109/tpami.2020.3019967.

LAMPIRAN A

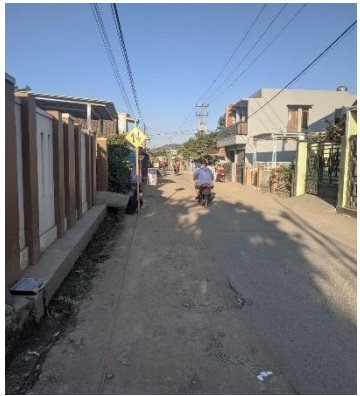
SAMPEL ESTIMASI JARAK

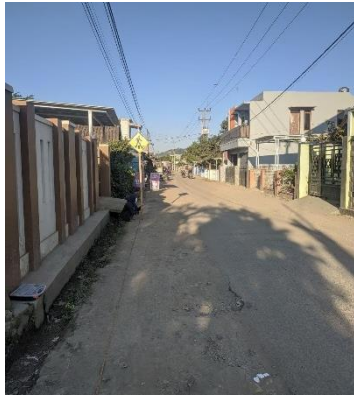
Jarak	Gambar
5 meter	
	
	






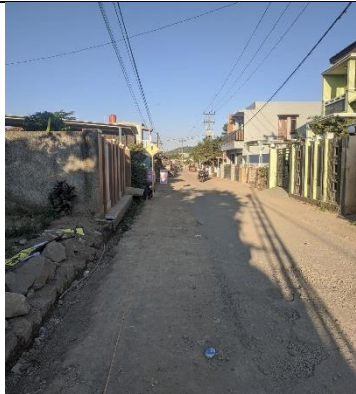
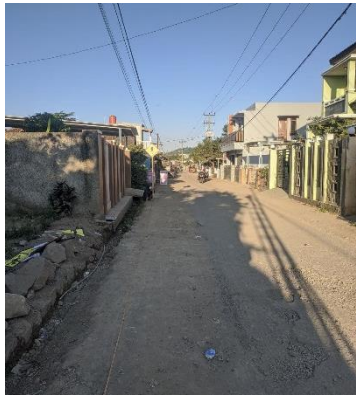
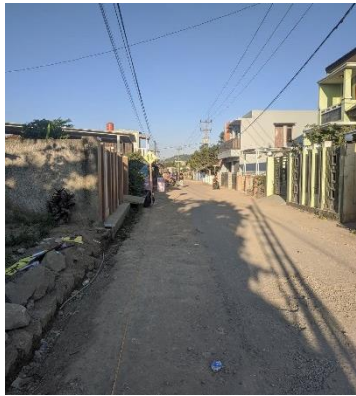


Jarak	Gambar
10 meter	
	
	



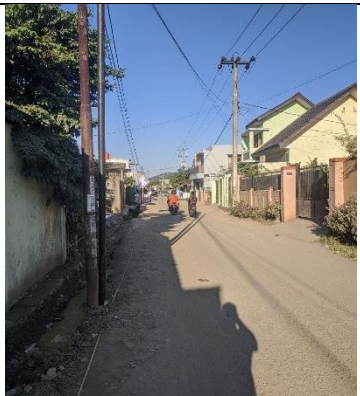
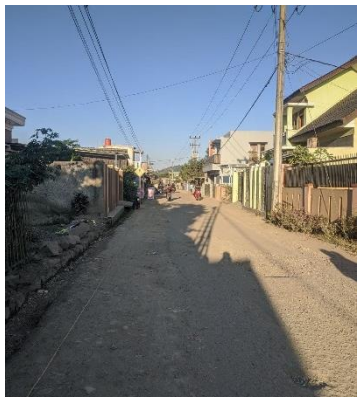
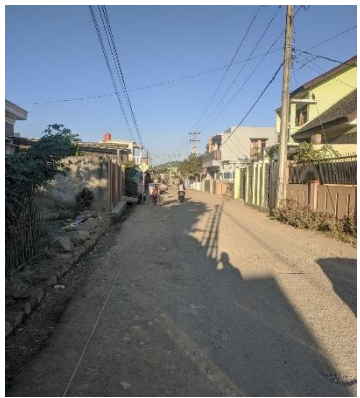


Jarak	Gambar
15 meter	
	
	








Jarak	Gambar
20 meter	
	
	





LAMPIRAN B

SAMPEL DETEKSI RAMBU LALU LINTAS

Jenis Rambu Lalu Lintas	Gambar Sampel
Belok Kiri	
Belok Kanan	
Hati-hati	

Bundaran



Penyeberangan Kuning



Penyeberangan Biru



Berhenti



Persimpangan



Lajur Dua Arah dan Hati-hati



Lurus

