

Pengaturan Lampu Lalu Lintas Berdasarkan Deteksi Volume Kendaraan Menggunakan Metode YOLOv3

Traffic Light Setting Based On Vehicle Volume Detection Using The YOLOv3 Method

1st Muhammad Irfaan Hadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
irfaanhadi@student.telkomuniversity.ac.id

2nd Desri Kristina Silalahi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
desrikristina@telkomuniversity.ac.id

3rd IG. Prasetya Dwi Wibawa
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
prasdwiwibawa@telkomuniversity.ac.id

Abstrak—*Traffic light* adalah salah satu teknologi yang perlu dikembangkan di Indonesia, dimana pengelolaan arus lalu lintas yang berdampak pada kelancaran atau kemacetan jalan. Penggunaan teknologi terbaru pada *traffic light* masih sangat minim, penggunaan sensor maupun pengembangan sistem pada objek masalah ini harus dikembangkan sedemikian rupa agar dapat memenuhi pencapaian arus lalu lintas yang efektif dan efisien. Metode yang digunakan pada penelitian ini adalah *YOLO* (*You Only Look Once*), yang penerapannya berupa penerapan pada skenario *traffic light* simpang empat menggunakan data rekaman video arus lalu lintas. Penggunaan metode ini diharapkan dapat digunakan secara maksimal karena algoritma yang dikembangkan pada metode ini dikhususkan untuk mendeteksi objek secara *real-time*. Berdasarkan hasil pengolahan data yang diambil sebagai skenario simpang empat, didapatkan kesimpulan bahwa dengan empat rekaman video arus lalu lintas sebagai data utama didapat hasil tes data *sample* berupa konfigurasi akurasi(%), presisi(%), *recall*(%), dan *F1 Score*. Data hasil yang diambil dibuktikan pada skenario arus jalan kedua dengan hasil Akurasi tertinggi 69%, Presisi 89%, *Recall* 75%, dan *F1 Score* 0,8. Dan juga penerapan *prototype* sederhana berjalan cukup baik menggunakan *serial communication* antar OpenCV Python dengan Arduino Mega yang menghasilkan sistem perempatan yang otomatis dan efisien berdasarkan jumlah kendaraan pada sebuah ruas jalan.

Kata Kunci : *traffic light*, Deteksi Objek, YOLO, Pengembangan Teknologi

Abstract—*Traffic light* is one of the technologies that need to be developed in Indonesia, where traffic flow management has an impact on road smoothness or congestion. The use of renewable technology in traffic light is still very minimal, the use of sensors and system development on the object of this problem must be developed in such a way as to meet the achievement of effective and efficient traffic flow. The method used in this study is *YOLO* (*You Only Look Once*), whose application is in the form of an application to the four intersection traffic light scenario using video recording data of traffic flow. The use of this method is expected to be used optimally because the algorithm developed in this method is specifically for detecting objects in *real-time*. Based on the results of data processing taken as an intersection scenario, it was concluded that with four video recordings of traffic flow as the main data, the results of the sample data test were in the form of configuration accuracy(%), precision(%), recall(%), and *F1 Score*. The result data taken is proven in the second road flow scenario with the highest accuracy of 69%, precision 89%, recall 75%, and *F1 Score* 0.8. And also the implementation of a simple prototype went quite well using *serial communication* between OpenCV Python and Arduino Mega which resulted in an automatic and efficient intersection system based on the number of vehicles on a road segment.

Keywords: *traffic light*, object detection, YOLO, technology development

I. PENDAHULUAN

Alat Pemberi Syarat Lalu Lintas (APILL) yang biasa disebut lampu lalu lintas adalah lampu yang dapat mengatur arus lalu lintas disebuah persimpangan jalan. Di Indonesia penerapan lampu lalu lintas yang sudah ada masih perlu dikembangkan agar dapat memenuhi kebutuhan para pengendara, dimana dibutuhkan sistem yang efisien sebagai upaya penertiban arus lalu lintas. Sebagai ibu kota Indonesia, Jakarta termasuk kota termacet peringkat ke 10 dengan tingkat kemacetan 53% pada survei tahun 2019.[1]

Solusi yang sudah ada untuk mengurangi kemacetan untuk saat ini pada lampu lalu lintas yaitu menggunakan sistem yang disebut ATCS (*Area Traffic Control System*). Sistem ini mengontrol lampu lalu lintas secara otomatis dengan menggunakan kamera yang dikontrol oleh mikrokontroler dengan sistem algoritma kontrol *fuzzy logic*. Kekurangan pada sistem ATCS terdapat pada bagian sistem pengaturan lama lampu lalu lintas yang masih manual, dimana kamera CCTV yang tersedia berfungsi sebagai pemantau arus lalu lintas saja. Fungsinya belum dimaksimalkan untuk penggunaan kamera CCTV sebagai sensor pada bagian sistem lampu lalu lintas ATCS tersebut.

Dampak yang terjadi jika sistem ini masih bersifat manual untuk pengaturan lama lampu lalu lintas pada persimpangan jalan tersebut, bisa terjadi adanya keteledoran atau biasa disebut *human error*. Maka dari itu digunakannya penggunaan teknologi sistem program metode deteksi objek kendaraan menggunakan metode YOLO (*You Only Look Once*) yang nantinya dapat menjadi salah satu solusi untuk memaksimalkan fungsi kamera sebagai pendeteksi jumlah kendaraan bermotor yang dapat mengatur lampu lalu lintas sesuai kepadatan setiap arus jalan.

ATCS (*Area Traffic Control System*) adalah sebuah sistem pengaturan lalu lintas bersinyal terkoordinasi yang diatur mencakup satu wilayah secara terpusat. ATCS dapat dilakukan upaya manajemen rekayasa lalu lintas yang mengoordinasikan semua titiktitik persimpangan bersinyal melalui pusat kontrol ATCS, sehingga diperoleh suatu kondisi pergerakan lalu lintas secara efisien (Irwanti, Samadikun, & Huboyo, 2017) [2]. Maka solusi pengembangan dari alat ATCS dengan sistem yang sudah ada, diberikanlah solusi yaitu pemaksimalan sistem kamera CCTV sebagai sensor pengaturan lama lampu lalu lintas dan penggunaan sistem *Deep Learning* yang berhubungan dengan ANN (*Artificial Neural Network*) atau jaringan syaraf tiruan. *Deep learning* dapat mempelajari metode komputasinya sendiri menggunakan 'otak'nya sendiri. Teknologi *deep learning* ini salah satu teknologi yang paling populer untuk mengenali suatu kegiatan atau objek yang memiliki tingkat keakuratan lebih tinggi dibanding dengan metode mesin sebelumnya [3]

Proses pembelajaran mesin dilakukan pada komputer yang berfungsi untuk mengklasifikasinya data citra menjadi hasil klasifikasi berupa prediksi. Teknologi *deep learning* merepresentasikan suatu konsep yang kompleks sebagai rangkaian konsep-konsep yang lebih sederhana. *Deep learning* merupakan algoritma jaringan saraf tiruan yang menggunakan data sebagai input dan memprosesnya dengan menggunakan sejumlah lapisan tersembunyi (*hidden layer*). Setelah itu melakukan transformasi non linier dari data masukan untuk menghitung nilai output [4]

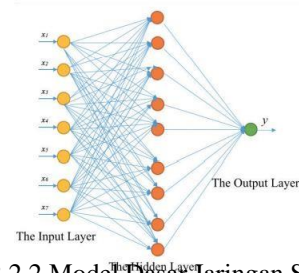
Maka dari itu solusi pada tugas akhir ini yaitu pada pengerjaannya, data utama yang diambil adalah rekaman kamera CCTV dari beberapa simpang jalan yang ada di Bandung, yang nantinya akan diproses oleh program deteksi citra menggunakan metode YOLOv3, lalu jika jumlah kendaraan pada sebuah rekaman video CCTV telah dihitung, maka data informasi jumlah kendaraan akan dikirimkan ke Arduino Mega yang telah dipasang beberapa lampu LED sebagai skenario *prototype* alat sederhana yang membentuk perempatan jalan lampu lalu lintas.

II. KAJIAN TEORI

A. Artificial Neural Network

Artificial Neural Network atau yang dikenal dengan nama Jaringan Syaraf Tiruan (JST) merupakan sistem proses komputasi yang terinspirasi dari sistem saraf biologis manusia seperti otak. JST terdiri dari sejumlah interkoneksi node komputasi yang disebut neuron. Struktur dasar dari JST terdiri dari 3 jenis layer utama, yaitu input, hidden, dan output layer. [5] Permasalahan yang harus diselesaikan saat menerapkan jaringan syaraf tiruan ke masalah yang diteliti adalah menentukan prosedur pelatihan mana yang harus diadopsi dan diaplikasikan. Ada banyak alternatif prosedur lainnya untuk dipilih.

Algoritma backpropagation merupakan metode yang cocok yang diaplikasikan dalam metode ini. Training adalah teknik pelatihan yang paling banyak digunakan untuk masalah yang serupa dengan penelitian saat ini [11]. Ilustrasi model dasar jaringan syaraf tiruan dapat dilihat pada Gambar 2.1.

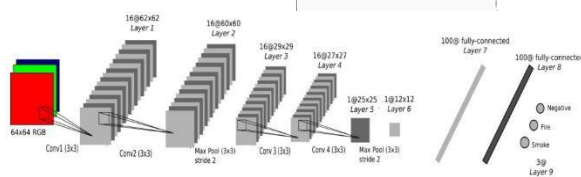


GAMBAR 2.2 Model Dasar Jaringan Syaraf Tiruan

Tahap pemodelan, yaitu perancangan arsitektur ANN. Tahapan ini merupakan proses yang kompleks dan dinamis yang memerlukan penentuan struktur permodelan dan aturan internal (yaitu, jumlah lapisan tersembunyi dan neuron dan jenis fungsi aktivasi). Model ini dirancang sesuai dengan jenis data dan respon yang dibutuhkan oleh aplikasi ANN. Model ANN saat ini telah dirancang untuk memasukkan input layer dari tujuh elemen pemrosesan (neuron) yang sesuai dengan tujuh parameter input dan output layer dari satu elemen pemrosesan (neuron) sebagai target. Satu hidden layer dari sembilan elemen proses dipilih setelah beberapa percobaan selama tahap pengujian. Fungsi hidden layer adalah mengekstrak serta mengingat fitur yang berguna. Subfitur dari pola input kemudian digunakan untuk memprediksi hasil jaringan atau nilai lapisan output [12].

B. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan JST yang neuron di dalam layer-nya tersusun menjadi 3 dimensi [5]. Berbeda dengan JST pada umumnya, neuron dalam layer tertentu pada CNN hanya akan terhubung pada ke wilayah kecil pada layer sebelumnya. O'Shea dan Nash juga menjelaskan bahwa penggunaan CNN difokuskan pada neural network yang menerima input berupa citra (image). Ilustrasi arsitektur CNN dapat dilihat pada Gambar 2.2



GAMBAR 2.3 Arsitektur Convolutional Neural Network

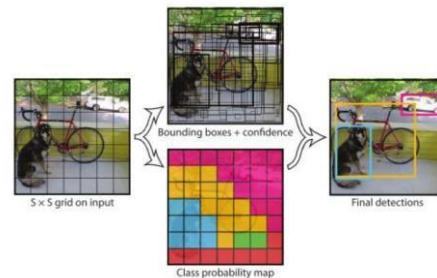
Convolutional Neural Network mengklasifikasikan sebuah gambar menjadi bagian bit layer terkecil, dimana bagian terdetail akan diklasifikasikan oleh sistem CNN yang mendeteksi sebuah gambar dari bit gambar yang saling terhubung dan diteruskan untuk menentukan sebuah objek melewati pooling layers.

C. You Only Look Once

You Only Look Once atau YOLO akan membagi inputan gambar menjadi grid berukuran $S \times S$, dimana nilai S adalah 7 dengan input gambar berukuran 448×448 . Untuk mendapatkan bounding box, akan dilakukan konvolusi dari inputan gambar, sehingga hasil akhirnya akan mendapat ukuran bounding box sebesar $S \times S \times (B \times 5 + C)$ dimana B adalah banyaknya bounding box (umumnya 2) dalam 1 grid dan C adalah banyaknya class yang dapat diklasifikasi. Nilai B dikalikan dengan 5 karena sebuah bounding box memiliki 5 nilai yang perlu disimpan, koordinat x , koordinat y , lebar

(width), tinggi (height), dan confidence score (nilai probabilitas bounding box yang bersangkutan memiliki sebuah objek) [6].

Untuk semua atribut pada bounding box akan dilakukan normalisasi sehingga nilainya menjadi antara 0 hingga 1. Koordinat x dan y akan dinormalisasi menyesuaikan titik kiri atas dari grid yang bersangkutan. Dan tinggi dan lebar akan dinormalisasi sesuai dengan ukuran gambar (width dan height). Nilai koordinat x dan y pada sebuah bounding box pada setiap grid merupakan titik tengah grid yang bersangkutan [7]. Ilustrasi YOLO dapat dilihat pada Gambar 3.



GAMBAR 2.4 You Only Look Once

1. YOLOv3

YOLO (You Only Look Once) memiliki beberapa tingkatan versi yang berbeda, dimana penggunaan pada metode kali ini adalah menggunakan metode versi ketiga. YOLO versi ketiga memiliki beberapa kelebihan :

- Prediksi Bounding Box :** YOLOv3 seperti halnya YOLOv2 menggunakan kluster dimensi untuk menghasilkan Anchor Box. Sekarang karena YOLOv3 adalah jaringan tunggal, kerugian untuk tujuan dan klasifikasi perlu dihitung secara terpisah tetapi dari jaringan yang sama.
- Prediksi Kelas :** YOLOv3 menggunakan pengklasifikasi logistik independen untuk setiap kelas, bukan lapisan softmax biasa. Hal ini dilakukan untuk membuat klasifikasi multi-label. Apa artinya dan bagaimana itu menambah nilai? Ambil contoh, di mana seorang wanita ditunjukkan pada gambar dan model dilatih pada orang dan wanita, memiliki softmax di sini akan menyebabkan probabilitas kelas dibagi antara 2 kelas ini dengan probabilitas 0,4 dan 0,45.
- Prediksi lintas skala :** Untuk mendukung deteksi skala yang bervariasi, YOLOv3 memprediksi kotak pada 3 skala berbeda. Kemudian fitur diekstraksi dari setiap skala dengan menggunakan metode yang mirip dengan jaringan piramida fitur.
- Ekstraktor Fitur :** YOLOv2 menggunakan Darknet-19 sebagai ekstraktor fitur tulang punggungnya,

YOLOv3 menggunakan jaringan baru-Darknet-53! Darknet-53 memiliki 53 lapisan konvolusi, lebih dalam dari YOLOv2 dan juga memiliki sisa atau koneksi pintasan. Ini kuat dari Darknet-19 dan lebih efisien daripada ResNet-101 atau ResNet-152.

	backbone	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
<i>Two-stage methods</i>							
Faster R-CNN++ [3]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [6]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [8]	Inception-ResNet-v2 [9]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [10]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
<i>One-stage methods</i>							
YOLOv2 [13]	DarkNet-19 [13]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [5, 2]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DS513 [2]	ResNet-101-SSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [7]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [7]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2
YOLOv3 608 × 608	DarkNet-53	33.0	57.9	34.4	18.3	35.4	41.9

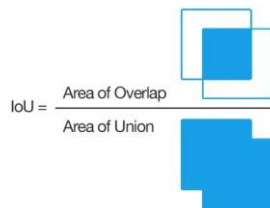
Mean Average Precision Comparisons 2018 (Source: YOLOv3 paper)

GAMBAR 2.5 Average Precision Comparisons

Dari data diatas 57,9 setara dengan semua dua detektor tahap. YOLO608 (YOLO terbaik dengan gambar input berdimensi tinggi) masih hampir 4x kali lebih cepat dari RetinaNet terbaik dan 2x lebih cepat dari RetinaNet terbaik kedua. YOLO320 memiliki akurasi yang sama dengan RetinaNet dengan backbone ResNet50 menjadi 4x kali lebih cepat. Ini membuat YOLOv3 jelas sangat efisien untuk kasus penggunaan deteksi objek umum. [14]

D. Intersection Over Union

Intersection over Union (IoU) merupakan metode evaluasi untuk mengukur akurasi deteksi objek terhadap suatu dataset. IoU membutuhkan 2 area yang akan di-intersect dan di-union, 2 area tersebut adalah area ground-truth bounding box yang merupakan bounding box aktual dan area yang dideteksi dari model yang dibangun [8]. Ilustrasi persamaan IoU dapat dilihat pada Gambar 4.



GAMBAR 2.6 Intersection Over Union

Intersection Over Union merupakan pembagian 2 area yang akan diseleksi pada vision computer yang nantinya akan ditampilkan pada video secara langsung dan akan menentukan sebuah objek benda, dimana setelah tertangkap setiap bit gambar pada video secara langsung akan menentukan tipe objek tertangkap.

Intersection Over Union merupakan pembagian 2 area yang akan diseleksi pada vision computer yang nantinya akan ditampilkan pada video secara langsung dan akan menentukan sebuah objek benda, dimana setelah tertangkap setiap bit

gambar pada video secara langsung akan menentukan tipe objek tertangkap.

E. Confusion Matrix

Confusion matrix adalah sebuah tabel yang menunjukkan kinerja dari sebuah model klasifikasi yang memiliki data jawaban benar (supervise). Dari tabel yang didapatkan, untuk model klasifikasi yang dimiliki dapat dihitung akurasi, presisi, F-Score dan masih banyak lagi variabel yang dapat dihitung berdasarkan kondisi data yang diprediksi atau diklasifikasikan [10].

Isi dari tabel confusion matrix ada 4, yaitu

1. True Positive (TP), kondisi dimana model mengklasifikasikan data sebagai ya (TRUE) dan jawaban aktualnya adalah ya (TRUE)
2. True Negative (TN), kondisi dimana model mengklasifikasikan data sebagai tidak (FALSE) dan jawaban aktualnya adalah tidak (FALSE)
3. False Positive (FP), kondisi dimana model mengklasifikasikan data sebagai ya (TRUE) dan jawaban aktualnya adalah tidak (FALSE)
4. False Negative (FN), kondisi dimana model mengklasifikasikan data sebagai tidak (FALSE) dan jawaban aktualnya adalah ya (TRUE). Persamaan yang akan digunakan berdasarkan data dari confusion matrix adalah akurasi, presisi, recall dan f-score.

- a. Akurasi merupakan pengukuran seberapa benar sebuah sistem dapat mengklasifikasi dari keseluruhan. Akurasi dapat dihitung menggunakan persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

- b. Presisi merupakan perbandingan jumlah data yang kategori positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Presisi dapat dihitung menggunakan persamaan (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- c. Recall merupakan pengukuran untuk data dengan klasifikasi positif yang benar oleh sistem. Recall dapat dihitung menggunakan persamaan (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- d. F-Score bertujuan untuk menghitung kombinasi dari presisi dan recall. F-Score akan menggunakan harmonic mean dari presisi dan recall. F-score dapat dihitung menggunakan persamaan (4).

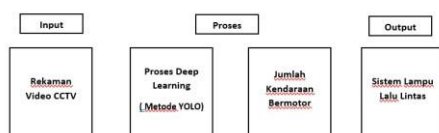
$$F\text{-Score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

III. METODE

Tugas akhir ini adalah sebuah sistem lampu lalu lintas untuk mendeteksi jumlah kendaraan bermotor, yang nantinya akan menentukan lama lampu lalu lintas. Sistem ini menggunakan kamera sebagai sensor. Video secara langsung akan ditangkap secara langsung dan akan menampilkan sebuah video pada mesin komputer dan akan diproses dari mesin komputer tersebut yang akan diolah melalui metode *deep learning* menggunakan *computer vision* yang akan diteruskan pada processor yang nantinya akan menentukan lama waktu penyalan lampu merah dan lampu hijau.

A. Diagram Kerja Sistem

Pada tugas akhir ini akan dirancang sebuah sistem lampu lalu lintas yang dapat mengatur kepadatan arus lalu lintas dengan menggunakan teknologi computer vision dan metode deep learning. Sistem Traffic Light ini terbagi ke dalam 2 bagian utama yakni bagian perangkat keras dan perangkat lunak. Perangkat keras berfungsi sebagai sensor untuk memberi masukan kepada sistem utama dan juga sebagai aktuator berupa pengendalian keluaran sistem. Bagian perangkat lunak berfungsi sebagai backend system perangkat keras untuk mengolah input dan output dan juga sebagai antarmuka bagi pengguna agar dapat menjalankan sistem.



GAMBAR 3.1 Blok Diagram Kerja Sistem

Berdasarkan sistem kerja diatas, terdiri dari rekaman video sebagai bahan utama deteksi jumlah kendaraan, arduino MEGA, lampu trafik sebagai output sistem. Arduino MEGA akan mengolah data yang telah diolah dari video ke computer vision, dengan menggunakan *enviromtent python* berperan penting karena pengolahan data video secara langsung yang nantinya akan diteruskan ke Arduino MEGA. Proses pengolahan data dan penghitungan lama waktu lampu lalu lintas dengan metode deep learning merupakan bagian perangkat keras dan bekerja di dalam sistem Arduino MEGA.

1. Fungsi dan Fitur

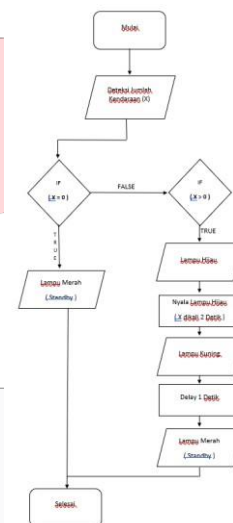
Berikut adalah fungsi dan fitur dari sebuah sistem :

- Mengambil video secara langsung disebuah ruas jalan
- Mengolah video secara langsung menggunakan computer vision

- Data yang telah diolah diteruskan ke lampu trafik untuk menentukan lama lampu lalu lintas

B. Flow Chart Sistem

Analisis sistem dibawah terdapat Inisiasi data sensor kamera yang menampilkan video secara langsung pada ruas jalan yang inputnya berupa jumlah kendaraan bermotor yang nantinya diproses oleh *Computer Vision* yang akan diubah menjadi bahasa programnya tersendiri untuk dilanjutkan pada sistem lampu lalu lintas yang akan menentukan lama lampu lalu lintas. Desain flow chart dapat dilihat pada diagram Gambar 3.



GAMBAR 3.2 Flow Chart Keseluruhan Sistem

Pada deteksi jumlah kendaraan terjadi proses deteksi kendaraan menggunakan sistem kerja YOLO, dimana YOLO tersebut akan mendeteksi dan menghitung jumlah kendaraan bermotor yang melewati garis horizontal pada sebuah video rekaman yang dimunculkan, yang dimana hanya mobil dan motor saja yang terdeteksi akan jumlahnya. Yang nantinya informasi jumlah kendaraan bermotor yang didapat akan dikirimkan ke Arduino Mega yang telah diatur untuk mengatur sistem lampu lalu lintas yang dibuat menggunakan LED dan alat-alat sederhana lainnya.

Untuk lampu LED Berwarna Hijau memiliki waktu 2 detik dari setiap kendaraan yang terdeteksi, contoh jika kendaraan yang terdeteksi berjumlah 6, maka lama lampu LED hijau yaitu 12 detik, lama lampu hijau akan disesuaikan berdasarkan jumlah kendaraan yang terdeteksi. Demi mengatasi nyala lampu LED hijau yang *overcount*, pada sistem lampu warna hijau telah diberikan batas paling lama menyala yaitu 60 detik.

Selanjutnya pada lampu warna kuning hanya menjadi lampu LED transisi dari merah ke hijau, lama lampu kuning diberikan waktu 1 detik saja. Untuk lampu LED merah dia akan *standby* menyala menunggu sistem lampu hijau diberikan

sesuai instruksi dari program python yang dilanjutkan ke arduino.

IV. HASIL DAN PEMBAHASAN

Pada bab ini, dipaparkan tentang hasil pengujian dan analisis dari sistem perhitungan jumlah kendaraan bermotor menggunakan metode YOLOv3. Pengujian sistem terdiri atas dua rancangan, yaitu pengambilan data, dan pengujian data.

A. Pengujian Sistem

Setelah merancang desain sistem Tugas Akhir ini, selanjutnya dilakukan pengujian dan analisis terkait. Beberapa tujuan dari pengujian ini antara lain:

1. Mengetahui performansi sistem dalam mendeteksi dan menghitung objek kendaraan bermotor.
2. Mendapatkan parameter performansi dengan hasil terbaik yang dapat dicapai oleh sistem dalam deteksi objek.
3. Mencari konfigurasi terbaik untuk mengoptimisasikan sistem.
4. Menganalisis hasil dari pengujian sistem dari confusion matrix dimana positif untuk mobil, dan negatif untuk motor

B. Skenario Pengujian Data

Hasil dari skenario data uji pada sistem sangat mempengaruhi keberhasilan model YOLOv3 dalam mendeteksi objek menggunakan file yolov3.weights dan yolov3.cfg, yang dimana terdapat 80 klasifikasi model yang berbeda, salah satunya yaitu mobil, dan motor. Dari proses uji data, akan didapatkan parameter akurasi, presisi, recall, F1 score, IoU, dan mAP. Dan hardware sederhana yang digunakan yaitu Arduino Mega yang akan memproses lama lampu lalu lintas menggunakan led merah, kuning, hijau sebagai output akhir pada sistem.

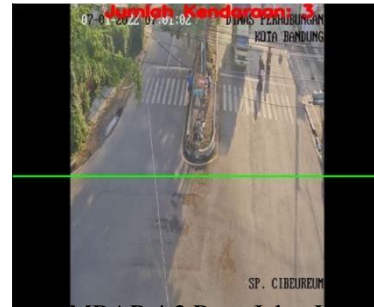
Data uji yang diambil pada skenario ini berasal dari CCTV *realtime* yang di rekam, bertujuan untuk memposisikan pengaplikasian software dan hardware sederhana secara *realtime* seperti di lapangan.

File data uji yang digunakan yaitu yolov3.weights dan yolov3.cfg yang dapat dilihat seperti Gambar 4.1



GAMBAR 4.1 File Pengujian Data

1. Skenario Ruas Jalan Pertama



GAMBAR 4.2 Ruas Jalan Ke-1

Data uji pertama menggunakan rekaman video CCTV pada ruas jalan SP. Cibereum 1 kota Bandung, yang diambil pada tanggal 07 Januari 2022 pada pukul 07:00:21 hingga 07:01:01. Pengujian data rekaman video CCTV menggunakan deteksi berdasarkan klasifikasi dari kelas mobil, dan motor, dimana sistem akan mendeteksi kendaraan bermotor yang berada pada rekaman tersebut.

Dari data pengujian menggunakan data latih yolov3.weights didapat data Confusion Matrix dan Performansi Data Uji yang terdapat pada Tabel 4.1 & Tabel 4.2

TABEL 4.1 Confusion Matrix Ruas Jalan Pertama

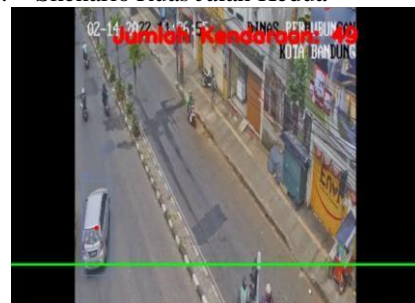
Class : Mobil & Motor		Actual Class	
		Positif	Negatif
Predicted Class	Positif	TP = 3	FP = 3
	Negatif	FN = 15	TN = 0

TABEL 4.2 Parameter Performansi Hasil Data Uji Pertama

Konfigurasi	Nilai
Akurasi (%)	14 %
Presisi (%)	50 %
Recall (%)	16 %
F1 Score	0,24

Percobaan data uji pertama mendapatkan nilai konfigurasi yang sangat rendah, dimana nilai klasifikasi *True Positive* pada sistem deteksi sangat minim. Dikarenakan pengaruh FPS yang rendah dan juga kualitas video yang berbeda pada setiap simpang jalan yang menyebabkan kualitas video juga akan berbeda. Nilai Akurasi didapat hanya 14%, yang dimana data uji pertama dinilai kurang memuaskan.

2. Skenario Ruas Jalan Kedua



GAMBAR 4.3 Ruas Jalan Ke-2

Data uji kedua menggunakan rekaman video CCTV pada ruas jalan SP. Jamika kota Bandung, yang diambil pada tanggal 14 Februari 2022 pada pukul 11:26:14 hingga 11:27:21. Pengujian data rekaman video CCTV menggunakan deteksi berdasarkan klasifikasi dari kelas mobil, dan motor, dimana sistem akan mendeteksi kendaraan bermotor yang berada pada rekaman tersebut.

Dari data pengujian menggunakan data latih yolov3.weights didapat data Confusion Matrix dan Performansi Data Uji yang terdapat pada Tabel 4.3 & Tabel 4.4

TABEL 4.3 Confusion Matrix Ruas Jalan Kedua

Class : Mobil & Motor		Actual Class	
Predicted Class	Positif	TP = 49	FP = 6
	Negatif	FN = 16	TN = 0

TABEL 4.4 Parameter Performansi Hasil Data Uji Kedua

Konfigurasi	Nilai
Akurasi (%)	69 %
Presisi (%)	89 %
Recall (%)	75 %
F1 Score	0,8

Untuk data uji kedua walaupun dalam FPS yang rendah akan tetapi dengan kualitas video yang cukup baik, didapat akurasi sebesar 69%, yang dimana nilai tersebut lebih baik dari nilai data uji sebelumnya, yang membuktikan bahwa kualitas video yang diambil cukup berpengaruh dengan hasil konfigurasi yang didapatkan yang dapat dilihat di tabel 4.4.

3. Skenario Ruas Jalan Ketiga



GAMBAR 4.4 Ruas Jalan Ke-3

Data uji ketiga menggunakan rekaman video CCTV pada ruas jalan SP. Lingkar - Bubat kota Bandung, yang diambil pada tanggal 14 Februari 2022 pada pukul 11:39:37 hingga 11:40:39. Pengujian data rekaman video CCTV menggunakan deteksi berdasarkan klasifikasi dari kelas mobil, dan motor, dimana sistem akan mendeteksi kendaraan bermotor yang berada pada rekaman tersebut.

Dari data pengujian menggunakan data latih yolov3.weights didapat data Confusion Matrix

dan Performansi Data Uji yang terdapat pada Tabel 4.5 & Tabel 4.6

TABEL 4.5 Confusion Matrix Ruas Jalan Ketiga

Class : Mobil & Motor		Actual Class	
Predicted Class	Positif	TP = 11	FP = 2
	Negatif	FN = 17	TN = 0

TABEL 4.6 Parameter Performansi Hasil Data Uji Ketiga

Konfigurasi	Nilai
Akurasi (%)	36 %
Presisi (%)	84 %
Recall (%)	39 %
F1 Score	0,53

4.



GAMBAR 4.5 Ruas Jalan Ke-4

Data uji keempat menggunakan rekaman video CCTV pada ruas jalan SP. Cibereum 2 kota Bandung, yang diambil pada tanggal 07 Januari 2022 pada pukul 07:00:21 hingga 07:01:01. Pengujian data rekaman video CCTV menggunakan deteksi berdasarkan klasifikasi dari kelas mobil, dan motor, dimana sistem akan mendeteksi kendaraan bermotor yang berada pada rekaman tersebut.

Dari data pengujian menggunakan data latih yolov3.weights didapat data Confusion Matrix dan Performansi Data Uji yang terdapat pada Tabel 4.7 & Tabel 4.8

TABEL 4.7 Confusion Matrix Ruas Jalan Keempat

Class : Mobil & Motor		Actual Class	
Predicted Class	Positif	TP = 1	FP = 4
	Negatif	FN = 16	TN = 0

TABEL 4.8 Parameter Performansi Hasil Data Uji Keempat

Konfigurasi	Nilai
Akurasi (%)	4 %
Presisi (%)	2 %
Recall (%)	5 %
F1 Score	0,08

5. Skenario Data Uji Tambahan



GAMBAR 4.6 Percobaan Video Resolusi Tinggi

Data uji kelima menggunakan rekaman video yang diambil dari internet, yang memiliki kualitas resolusi tinggi, dimana pada pengujian ini

ditujukan untuk membuktikan bahwasannya kualitas video dapat mempengaruhi hasil deteksi pada deteksi YOLOv3. Pengujian deteksi berdasarkan klasifikasi dari kelas mobil, dan motor, dimana sistem akan mendeteksi kendaraan bermotor yang berada pada video tersebut.

Dari data pengujian menggunakan data latih yolov3.weights didapat data Confusion Matrix dan Performansi Data Uji yang terdapat pada Tabel 4.9 & Tabel 4.10

TABEL 4.9 Confusion Matrix Video Resolusi Tinggi

Class : Mobil & Motor		Actual Class	
Predicted Class	Positif	TP = 61	FP = 6
	Negatif	FN = 3	TN = 0

TABEL 4.10 Parameter Performansi Hasil Data Uji Video Resolusi Tinggi

Konfigurasi	Nilai
Akurasi (%)	87 %
Presisi (%)	91 %
Recall (%)	95 %
F1 Score	0,92

C. Perbandingan Hasil Pengujian

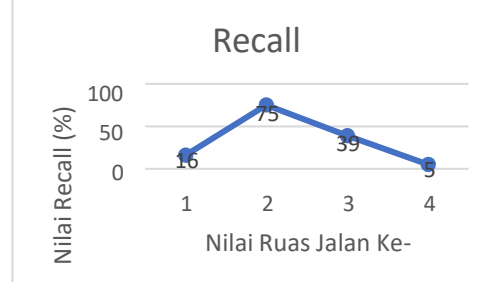
Dari hasil empat skenario ruas jalan yang telah diuji, didapatkan perbandingan data hasil konfigurasi yang telah di uji oleh *weights file*, pada setiap ruas jalan yang diuji memiliki nilai konfigurasi yang berbeda, dimana nilai tertinggi konfigurasi didapat pada skenario ruas jalan kedua yang mendapatkan nilai akurasi 69%, seperti yang disajikan pada data grafik keseluruhan ruas jalan sebagai berikut:



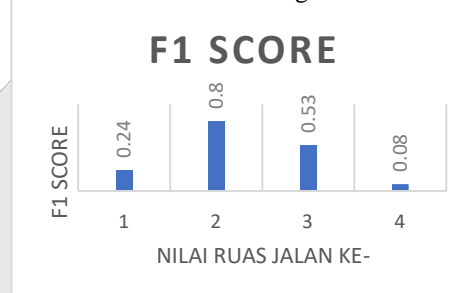
GAMBAR 4.7 Perbandingan Nilai Akurasi



GAMBAR 4.8 Perbandingan Nilai Presisi



GAMBAR 4.9 Perbandingan Nilai Recall



GAMBAR 4.10 Nilai F1 Score

Pada setiap skenario ada beberapa yang mempengaruhi nilai konfigurasi, salah satunya kualitas video, jarak antar kendaraan, dan jarak pandang dari kamera ke objek terdeteksi. Saat kualitas video yang diambil cukup baik, jarak antar kendaraan yang tidak bertumpuk, dan jarak pandang kamera ke objek tidak terlalu jauh, maka nilai konfigurasi akan semakin baik.

Untuk skenario pengujian ruas jalan keempat terdapat nilai konfigurasi yang sangat kecil, pengaruh kualitas video pada deteksi ruas jalan keempat sangat mempengaruhi nilai konfigurasi.

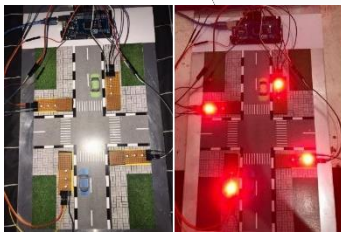
D. Skenario Pengujian Sistem dengan Alat Sederhana

Setelah mendapatkan data hasil pengujian pada keempat jalur sebagai perumpamaan simulasi simpang empat menggunakan pengujian YOLOv3 *weights*. Diperlukan pengujian sistem secara keseluruhan untuk mencoba simulasi sederhana menggunakan Arduino mega 2560 dan beberapa lampu LED sebagai media komunikasi dan output pada lampu diantara Python dan Arduino itu sendiri. Pengujian dimulai dari program python yang digunakan untuk menghitung jumlah kendaraan pada sejumlah video, setelah program mengambil data berapa jumlah kendaraan yang terdeteksi seperti yang terlihat pada gambar 4.10



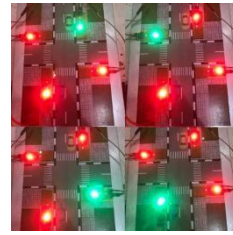
GAMBAR 4.11 Deteksi Skenario 4 Titik

Setelah informasi jumlah kendaraan bermotor didapatkan, informasi selanjutnya akan dikirimkan pada Arduino Mega 2560 dalam bentuk *string*, data yang telah ditangkap oleh Arduino akan diproses pada lima lampu lalu lintas yang telah di program pada Arduino itu sendiri, dimana setiap lampunya memiliki lima lampu lalu lintasnya sendiri.



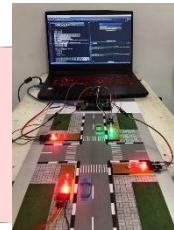
GAMBAR 4.12 Lampu LED Merah Standby

Pada Gambar 4.11 terlihat kondisi awal pada lampu lalu lintas sebelum deteksi objek pada rekaman video dimulai. Lampu akan *standby* pada lampu warna merah menunggu instruksi dari deteksi objek yang akan dilakukan pada proses deteksi program perangkat lunak Spyder(anaconda3).



GAMBAR 4.13 Sistem Lampu LED Hijau

Terlihat pada Gambar 4.12 lampu hijau mulai menyala sesuai instruksi dari kodingan program secara bergantian dengan lama lampu hijau menyala sesuai dengan jumlah kendaraan bermotor pada setiap arus lalu lintas yang terdeteksi.



GAMBAR 4.14 Tampilan Keseluruhan Alat

Berikut merupakan dokumentasi keseluruhan alat yang digunakan baik perangkat keras dan juga perangkat lunak.

V. KESIMPULAN

Berdasarkan hasil simulasi dan analisis penelitian yang telah dilakukan sebelumnya, dapat ditarik kesimpulan sebagai berikut:

- Konfigurasi keseluruhan terbaik dengan penggunaan pengujian menggunakan *weights file* terdapat pada percobaan skenario ruas jalan kedua dengan nilai akurasi 69%, presisi 89%, *recall* 75%, dan *F1 Score* 0.8.
- Pengaruh kualitas video terhadap deteksi objek sangat menentukan hasil akurasi, lebih jernih hasil video, semakin mudah deteksi jumlah objek kendaraan bermotor.
- Nilai akurasi, presisi, *recall*, dan *F1 Score* terendah didapat oleh skenario ruas jalan keempat, dikarenakan terlihat kualitas pada video lebih rendah dari video yang lainnya.
- Jarak pandang dari kamera ke kendaraan bermotor terlalu jauh menghasilkan objek tidak terdeteksi, dan tidak bisa menjadi penentu jumlah kendaraan bermotor yang ada.

REFERENSI

- [1] Shalihah, N. F. (2020, 1 31). *Survei 2019, Jakarta Masuk Peringkat 10 Kota Termacet di Dunia*. Retrieved from <https://www.kompas.com/https://www.kompas.com/tren/read/2020/01/31/052816565/survei-2019-jakarta-masuk-peringkat-10-kota-termacet-di-dunia?page=all#:~:text=KOMPAS.com%20%2D%20>

20Lembaga%20pemantau%20kemacetan,dengan%
20tingkat%20kemacetan%2053%20persen.

[2] Irwanti, I., Samadikun, B P., & Huboyo, H. S.(2017). *Pengaturan Manajemen Waktu Area Traffic Control System (Atcs) dalam Rangka Pengurangan Emisi Pencemar Udara Kendaraan Bermotor di Beberapa Persimpangan Jalan Kota Semarang*. Retrieved from Jurnal Teknik Lingkungan, 6(3), 1–10:

<https://www.neliti.com/id/publications/190556/pengaturanmanajemen-waktu-area-traffic-controlsystem-atcs-dalam-rangka-pengura>

[3] L. Deng dan D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7,no. 3-4, hal. 197-387, 2013.

[4] Primartha, R. (2018). *Belajar machine learning teori dan praktek*. Bandung: Informatika, 2018.

[5] O'Shea, K. & Nash, R. (2015). An Introduction to Convolutional Neural Networks. Diambil kembali dari arXiv: <https://arxiv.org/abs/1511.08458>

[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. *arXiv*, 1-10.

[7] Menegaz, M. Understanding YOLO – Hacker Noon. Diambil kembali dari HackerNoon: <https://hackernoon.com/understanding-yolo-f5a74bbc7967>

[8] Rosebrock, A. 2016. *Intersection over Union (IoU) for object detection*. Diambil kembali dari PYImageSearch: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

[9] DelSole, M. *What is One Hot Encoding and How to Do It*. Diambil kembali dari Medium: <https://medium.com/@michaeldelsole/what-is-one-hot-encoding-and-how-to-do-it-f0ae272f1179>

[10] DataSchool.io. (2014). *Simple guide to confusion matrix terminology*. Diambil kembali dari DataSchool: <https://www.dataschool.io/simple-guide-to-confusion-matrixterminology/>

[11] Hegazy, T. and A. Ayed., 1998. A Neural Network Model for Parametric Cost Estimation of Highway Projects.

[12] Kusumadewi, S., 2004, *Membangun Jaringan Saraf Tiruan Menggunakan MATLAB & Excel Link*. Yogyakarta: Penerbit Graha Ilmu.

[13] Kevin Adiputra Shianto, K. G. (n.d.). Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN.

[14] Anand Sonawane (2018) YOLOv3: A Huge Improvement <https://sonawaneanand.medium.com/yolo3-a-huge-improvement-2bc4e6fc44c5>

