

Sistem Deteksi Marka Jalan Berbasis *Convolutional Neural Network*

Oddy Virgantara Putra¹, Irfan Nanda Gustri²

¹Teknik Informatika, Universitas Darussalam Gontor, Ponorogo, Indonesia

²Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Email: oddy@unida.gontor.ac.id, irfananda2212@gmail.com

Abstrak

Keselamatan dalam berkendara sangatlah penting. Akan tetapi pengendara kurang begitu peduli dengan jalanan, salah satunya marka jalan yang bisa mengakibatkan hal yang tidak diinginkan seperti kecelakaan hingga kemacetan lalu lintas. Dengan begitu perlunya pengingat agar pengemudi bisa mengambil tindakan pencegahan, seperti mendeteksi marka yang ada di jalan yang berguna untuk mengontrol dan mempertimbangkan posisi kendaraan. Deteksi marka jalan yang menggunakan Convolutional Neural Network (CNN) sebagai pengindraan kondisi marka jalan sehingga dapat membaca objek badan marka jalan yang mendeteksi dan mengenali bentuk yang ditangkap menggunakan kamera, lalu citra tersebut diterima pada input, diproses, dan output, yang diolah dalam dua dimensi hingga menghasilkan proses yang diharapkan. Library yang digunakan pada bidang CNN ini menggunakan TensorFlow, dibantu dengan algoritma You Only Look Once (YOLO). TensorFlow dan YOLO digunakan untuk mengeksekusi perintah dan mengenali objek yang berbeda. Dengan begitu diharapkan kesalahan yang terjadi di lalu lintas semakin kecil.

Kata kunci : Marka Jalan, Convolutional Neural Network (CNN), TensorFlow, You Only Look Once (YOLO)

Diterima Redaksi: 20-11-2022, Selesai Revisi: 09-12-2022, Diterbitkan Online: 27-02-2023

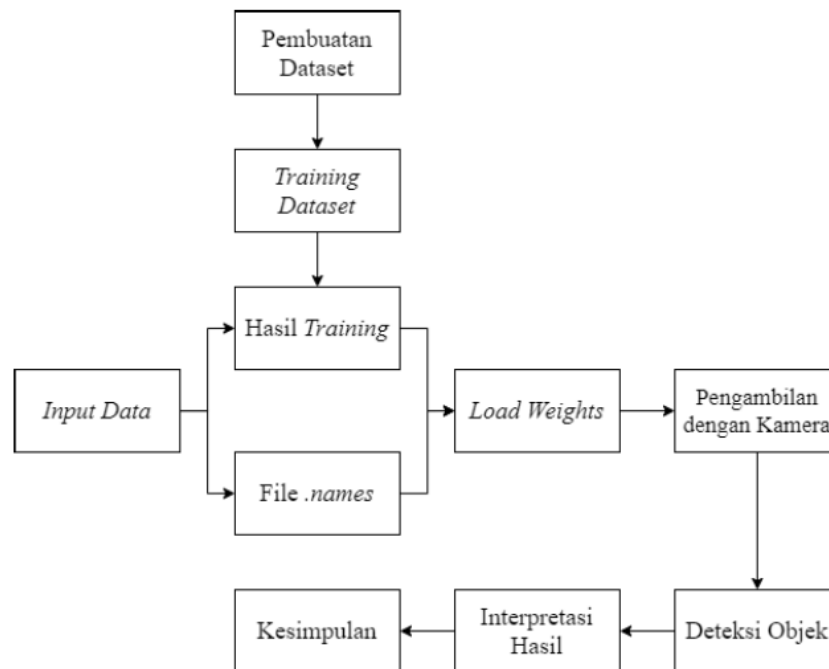
DOI: <https://doi.org/10.59378/jcenim.v1i1.2>

I. PENDAHULUAN

Dengan perkembangan pembangunan jalan dan semakin banyak kendaraan, kecelakaan lalu lintas juga cenderung meningkat saat berkendara di jalan. Tentunya banyak kecelakaan bisa dihindari jika ada sistem untuk mengingatkan pengemudi agar mengambil tindakan pencegahan lebih awal. Karena itu, pengembangan sistem ini telah dikembangkan secara signifikan. Hingga saat ini, penelitian dan pengembangan sistem pendukung keselamatan berkendara telah menjadi topik yang digemari untuk dikembangkan, mengingat kebutuhan ke depan akan semakin pesat.

Pengembangan kendaraan cerdas adalah sistem terintegrasi untuk menggabungkan seperangkat model teknologi seperti pengindraan kondisi sekitar, perencanaan pengambilan keputusan, fungsi bantuan pengemudi dan sebagainya. Kendaraan cerdas menggabungkan kemampuan untuk merasakan kondisi lingkungan kendaraan saat berkendara seperti jarak antara kendaraan di jalan raya serta membaca rambu-rambu lalu lintas khususnya marka jalan raya dan sistem kontrol yang diproses dari hasil penginderaan tersebut untuk membantu pengendara lebih aman dan nyaman saat berkendara.

Deteksi marka jalan merupakan suatu metode untuk mengetahui lokasi dari marka jalan tanpa diketahui terlebih dahulu noise yang terdapat pada lingkungan sekitarnya. Deteksi ini telah menjadi penelitian yang sering dilakukan oleh banyak orang agar bisa menjadi intelligent transportation system [1] [2]. Namun, penelitian ini masih berlanjut sampai sekarang karena masih terdapat banyak masalah-masalah yang belum bisa diselesaikan. Contohnya adalah sulitnya menentukan marka jalan pada berbagai kondisi, terutama terhadap noise yang ada di lingkungan sekitarnya, seperti bayangan, garisgaris lain yang terdapat pada jalan, dan juga marka jalan yang sudah pudar warnanya dan tidak jelas bentuknya. Dalam deteksi ini terdapat metode yang bisa digunakan dengan deteksi tepi pada objek, setelah deteksi dengan menggunakan bidang datar yang disebut hough transform [3]. Sistem ini dapat mendeteksi marka jalan



Gambar 1: Alur Deteksi Sistem Marka Jalan

lebih akurat dengan menggunakan metode Convolutional Neural Network (CNN) sebagai pengindraan kondisi sekitar, terutama marka jalan.

II. DESAIN SISTEM

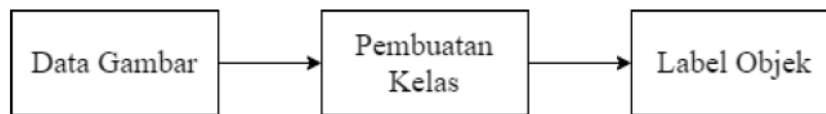
Pada penelitian ini merupakan penelitian dalam bidang Pengolahan Citra dan Video yang bertujuan untuk mendeteksi sistem marka jalan berbasis Convolutional Neural Network (CNN). Analisa ini dilakukan dengan memanfaatkan sistem pendeteksi marka jalan yang diproses pada objek dan input yang ditangkap. Diagram alur dari proses analisis sistem ditunjukkan oleh Gambar 1.

Berdasarkan alur pada Gambar 1, proses pertama yang dilakukan adalah pembuatan dataset. Untuk melakukannya, diperlukan mengumpulkan data-data gambar yang berfungsi untuk mempersiapkan data berupa marka jalan yang akan dilatih. Setelah mengumpulkan data gambar, data tersebut diberi label atau object bounding boxes pada gambar marka jalan yang ditunjuk pada objek beserta kelas dan data. Setelah proses pelabelan, perlu adanya konversi berkas-berkas gambar dan informasi label dari gambar yang dilabel dalam bentuk file .txt dengan berbasis CNN, yaitu You Only Look Once (YOLO) sebagai algoritmanya untuk proses training.

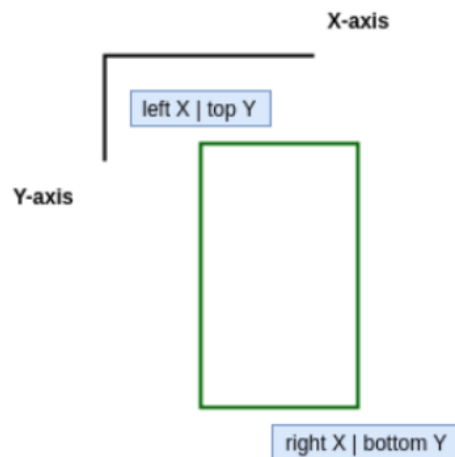
Dari hasil training diperoleh weights. Hasil training tersebut nantinya dapat digunakan untuk membantu mendeteksi penelitian marka jalan yang dikonversi dari kumpulan data gambar marka jalan dan informasi labelnya. Kemudian mengimplementasikan hasil training dengan menggunakan Python, dan Tensorflow sebagai open source, dengan memuat input data berupa hasil training dan file .names dengan load weights yang mengubah YOLO weights menjadi file model checkpoint (.ckpt) tensorflow. Setelah itu melakukan deteksi marka jalan dengan menjalankan file model tensorflow secara real-time menggunakan akses kamera yang membantu pengambilan objek dengan command line interpreter sebagai perintah untuk menjalankan program deteksi marka jalan dengan dilakukan pengambilan di 5 lokasi di Pekanbaru, 4 waktu, dan 3 cuaca untuk membandingkan kondisi ideal mengenali objek dengan baik. Saat menjalankan program deteksi, terlihat objek yang diambil menggunakan kamera, hingga akhirnya dari proses-proses yang dilakukan dapat mengenali bentuk pada marka jalan. Penggunaan GPU dengan kinerja yang tinggi dapat memaksimalkan hasil deteksi objek yang diuji.

A. Proses Pembuatan Dataset

Pada proses ini diawali dengan mengumpulkan data gambar marka jalan dengan cara mengambil foto dari internet dan video yang diubah menjadi foto. Agar dataset dapat dikonversi dengan training



Gambar 2: Tahapan pembuatan dataset



Gambar 3: Koordinat BBOX

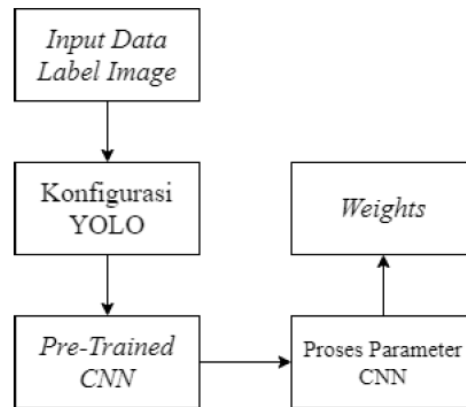
menggunakan YOLO, dataset perlu diberi label untuk mendapatkan informasi koordinat agar ketika melakukan proses training dapat bekerja dengan baik. Alur prosesnya dijelaskan pada Gambar 2.

gambar yang dikumpulkan sesuai kelas yang ada dengan menghasilkan informasi objek. Kelas tersebut berisi jenis-jenis pada gambar yang akan dilabel dengan format .txt. Dari beberapa data gambar, dilakukan pelabelan satu per satu secara manual sesuai jenis-jenis pada kelas dengan membuat koordinat *bounding box* (BBOX) yang berisi posisi koordinat pada kotak, yaitu X-kiri atas, Y-kiri atas, X-kanan bawah, Y-kanan bawah, yang ilustrasi lokasi koordinatnya bisa dilihat pada Gambar 3, dengan memperhatikan fleksibilitas yang maksimum.

Memiliki koordinat BBOX ini nantinya berguna untuk mengetahui identifikasi citra saat diuji. Gambar 4 merupakan proses pelabelan objek gambar dengan membuat koordinat BBOX sesuai pada jenis-jenis kelasnya dengan data yang dilabeli sebanyak 121 gambar. Informasi koordinat label kemudian disimpan dalam bentuk file .txt. Opsional konversi anotasi setiap file .txt agar memisahkan anotasi yang tidak perlu dengan yang perlu agar bisa diproses dalam tahapan selanjutnya.



Gambar 4: Pemberian label pada dataset



Gambar 5: Alur proses YOLOv3

B. Training

Proses training merupakan tahapan dari dataset yang telah dilabel dengan CNN yang kemudian dilatih untuk memperoleh weights. Weights akan digunakan untuk membantu mengenali objek dalam citra. Training yang digunakan berbasis CNN menggunakan You Only Look Once (YOLO) sebagai algoritmanya. Untuk YOLO yang digunakan yaitu YOLOv3. Dataset ini memiliki 10 kelas. Alur prosesnya terlihat seperti pada Gambar 5

1. Pengolahan Input: Pengolahan input merupakan tahapan persiapan sebelum melakukan proses training. Sebelum melakukan proses training, ada baiknya mengkonfigurasi dengan benar beberapa file, dengan mengatur file yolov3- marking.cfg. File yolov3-marking.cfg ini adalah pengaturan konfigurasi yang mempengaruhi hasil training yang nantinya akan digunakan untuk mendeteksi marka jalan. Adapun parameter yang diperlukan dalam mengkonfigurasi file yolov3- marking.cfg sebagai berikut:
 - (a) Mengatur nilai batch menjadi 64 dan nilai subdivisions menjadi 32. Jadi dalam proses iterasi pertama training, ada 64 sampel gambar yang diproses kemudian dibagi dengan 32 proses, sehingga ada 2 gambar yang diproses sebanyak 32 kali untuk mencapai proses iterasi yang pertama hingga mencapai batas maksimal proses training. Nilai subdivisions bisa diganti menjadi lebih kecil bila tidak memiliki permasalahan terhadap memori GPU pada perangkat yang digunakan.
 - (b) Kemudian mengatur max batches, yang merupakan batas maksimal proses iterasi training, yang didapat dari jumlah kelas $\times$ 2000. Jumlah kelas yang ada pada penelitian ini ada 10, maka nilai *max batches* yang digunakan adalah 20000.
 - (c) Mengatur nilai steps yang memiliki 2 variabel. Variabel pertama didapat adalah 80% dari nilai max batches dan variabel kedua adalah 90% dari nilai max batches. Sehingga nilai steps yang digunakan adalah 16000 dan 18000.
Setiap layer [yolo], mengubah nilai classes menjadi berapa banyak kelas yang akan dideteksi. Jadi nilai classes ialah sebanyak 10.
 - (d) Untuk setiap layer [convolutional], terdapat pengaturan nilai filters. Filters adalah berapa banyak informasi yang dapat diperoleh ketika memproses layer. Nilai filters didapatkan dari 4 box coordinates, 1 object confidence dan nilai jumlah kelas, kemudian dikalikan dengan 3 nilai anchors. Nilai anchors adalah rasio nilai bounding boxes dengan height dan weight. Sehingga perhitungan nilai filters adalah $(4 + 1 + \text{jumlah kelas}) \times 3 = (5 + \text{jumlah kelas}) \times 3$. Nilai filters yang digunakan adalah 45.

Kemudian melakukan konfigurasi file .names dan file .data. File .names berisi nama-nama kelas dan file .data berisi berapa jumlah kelas dan lokasi penyimpanan *file weights*. File konfigurasi terakhir yang diperlukan sebelum melakukan training adalah file train.txt yang berguna untuk menahan jalur relatif ke setiap gambar yang akan di-training.

2. Feature Extraction dan Object Classification: YOLOv3 menggunakan feature extractor yang bernama Darknet53, yang memiliki 53 layer menggunakan convolutional weights yang telah dilatih

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	$3 \times 3 / 2$	64×64
2x	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
	Convolutional	256	$3 \times 3 / 2$	32×32
8x	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
	Convolutional	512	$3 \times 3 / 2$	16×16
8x	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
	Convolutional	1024	$3 \times 3 / 2$	8×8
4x	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 6: Arsitektur Darknet54 [17]

di ImageNet. Untuk tugas pendeteksian, ditambahkan lagi 53 layer ke dalam jaringan, sehingga terdapat 106 layer yang mendasari arsitektur convolutional dari YOLOv3. Hal ini menyebabkan proses pendeteksian lebih lambat, namun memberikan hasil yang lebih akurat. Gambar 6 menunjukkan susunan layer dari Darknet53 [8]. Sedangkan untuk prediksi kelasnya, YOLO generasi awal menggunakan fungsi softmax yang menghitung probabilitas setiap kelas target dari semua kemungkinan kelas target. Namun pada YOLOv3 digunakan multilabel classification pada setiap objek yang terdeteksi pada gambar. Hal ini dilakukan karena fungsi softmax hanya memberikan satu kelas saja pada objek yang dideteksi. Sedangkan pada multilabel classification, setiap skor kelas diprediksi dengan menggunakan logistic regression dan threshold untuk memprediksi beberapa label dari suatu objek.

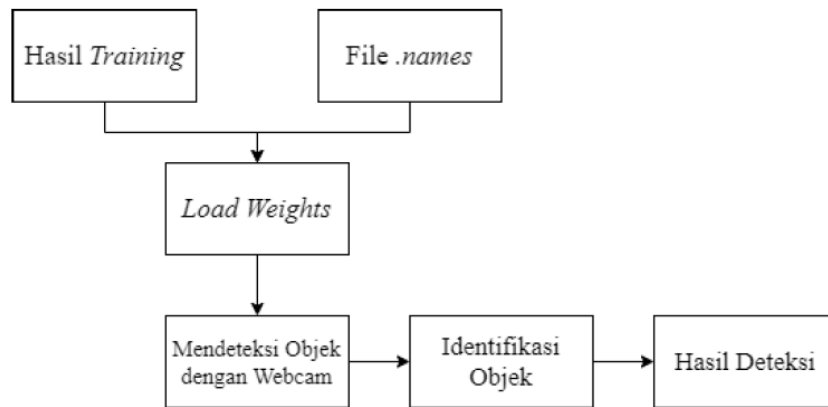
Setelah semua konfigurasi sudah terpenuhi, selanjutnya melakukan training dengan menggunakan Windows Power-Shell sebagai command interpreter hingga akhirnya hasil proses training menjadi weights. Proses training dilakukan 4 hari. File weights digunakan untuk melakukan proses selanjutnya.

C. Proses Testing

Proses testing merupakan proses klasifikasi menggunakan weights dan kelas dari hasil proses training. Proses ini memasukkan input data yang berupa hasil training dan kelas yang berada di file .names melakukan proses load weights, memuat file weights dan file .names mengubah dari YOLO weights menjadi file checkpoint (.ckpt) tensorflow. Sehingga hasil akhir dari proses ini menghasilkan akurasi yang baik dari klasifikasi yang dilakukan untuk dapat membantu proses deteksi, khususnya deteksi secara real-time marka jalan. Alur proses testing terlihat pada Gambar 7.

1. *Mekanisme Pengambilan Objek*: Dengan weights yang diubah menjadi file .tf, membuat proses deteksi dapat berjalan dengan baik. Sehingga dapat dilakukan proses pengambilan objek dengan kamera secara real-time yang diletakkan di arah

depan dalam mobil yang mengarah ke objek, yaitu marka jalan. Lokasi pengambilan objek yang dilakukan berada di area jalan yang memiliki marka jalan, meliputi ruas jalan yang ada di Pekanbaru,



Gambar 7: Alur Proses Testing

Kecepatan (km/jam)	FPS	IB	IS	B%
20	1.71	76	1	98.70
40	1.69	87	1	98.86
60	1.68	105	10	91.30
Total		268	12	95.71

Gambar 8: Peta Lokasi Rute Pengujian di Jl. Jenderal Sudirman, Pekanbaru

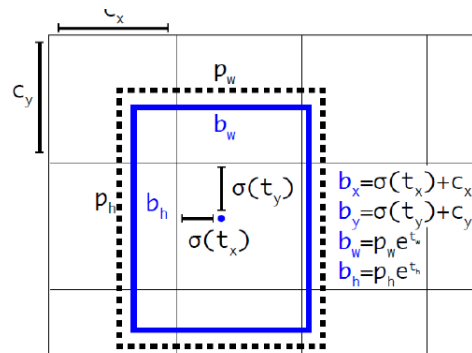
diantaranya:

- Jl. Jenderal Sudirman, Pekanbaru, dengan panjang rute sekitar 2.8 km yang ditunjukkan rutenya pada Gambar 8.
- Jl. Pattimura, Pekanbaru, dengan panjang rute sekitar 1.7 km yang ditunjukkan rutenya pada Gambar 10.
- Jembatan Siak IV Pekanbaru, dengan panjang rute sekitar 5.3 km yang ditunjukkan rutenya pada Gambar 11.
- Jl. Bandara SSK II, Pekanbaru, dengan panjang rute sekitar 400 m yang ditunjukkan rutenya pada Gambar 12.

Pengambilan dilakukan pada waktu pagi, siang, sore dan malam hari agar dapat membandingkan kemampuan pendeteksian marka jalan di waktu yang berbeda

- Mendeteksi Objek: Pada tahap awal, menjalankan file model tensorflow dengan command line interpreter sebagai perintah untuk mendeteksi marka jalan. Pengambilan dilakukan dengan perbandingan 5 lokasi di Pekanbaru seperti yang sudah dijelaskan sebelumnya, 4 waktu, dan 3 cuaca. Hal itu demikian dilakukan untuk membandingkan kondisi ideal saat mendeteksi objek dengan harapan mengenali objek dengan baik.
- Pengolahan Output: YOLO menghasilkan prediksi berupa koordinat bounding box ($tx; ty; tw; th$), confidence, dan class probability. Koordinat bounding box akan digunakan sebagai identifikasi citra, confidence merupakan tingkat keyakinan ada atau tidaknya objek pada gambar, sedangkan class probability merupakan jenis kelas dari objek yang terdeteksi. YOLO membuat bounding box pada lokasi feature map dengan menggunakan logistic activation sigmoid. Pengolahan bounding box dari hasil deteksi diilustrasikan pada Gambar 12. Dimana ($cx; cy$) merupakan lokasi dari grid. ($bx; by$) merupakan lokasi bounding box yang didapatkan dari lokasi ($cx; cy$) dengan hasil fungsi sigmoid dari tx dan ty , ($pw; ph$) merupakan anchor box yang didapatkan dari proses clustering, sedangkan ($bw; bh$) merupakan dimensi akhir dari bounding box yang didapatkan dari perkalian anchor box ($pw; ph$) dengan ($tw; th$). Pada deteksi karakter, diperlukan hasil deteksi pada karakter awal dan karakter akhir yang baik.

Apabila terdapat kesalahan pada deteksi, maka marka jalan yang dideteksi tidak dapat teridentifikasi dengan benar bahkan tidak dapat terdeteksi.



Gambar 9: Prediksi Lokasi Bounding Box

Tabel 1: Spesifikasi Laptop Asus TUF Gaming FX504 Series

Merk Laptop	ASUS TUF Gaming FX504GE
Prosesor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
Sistem Operasi	Windows 10 Home
RAM	16GB
Grafis	NVIDIA GeForce GTX 1050 Ti, dengan 4GB GDDR5
HDD/SSD	1TB/500GB
Berat	2.30 kg

Dengan menggunakan tensorflow yang memiliki fleksibilitas yang mendalam, mudah digunakan dan memiliki performa yang tinggi, membuat hasil dari deteksi marka jalan yang dilakukan menghasilkan informasi yang akurat. Terlihat dari saat deteksi dijalankan, informasi marka jalan apa yang dideteksi sudah berhasil untuk terbaca. Walaupun masih ada kekurangan pembacaan karena dataset yang di-train masih belum begitu banyak, perlu penambahan dataset dan label image yang lebih baik.

D. Analisis Hasil Deteksi Marka Jalan

Untuk mengetahui hasil dari deteksi marka jalan bisa dilihat pada perangkat yang melakukan pemrosesan yang berpengaruh pada hasil frame per second (fps) dan akurasi keberhasilan mendeteksi marka jalan. Pada fps, semakin besar fps yang dipunya, maka pergerakan visual yang ditampilkan semakin halus dan lancar. Menghitung nilai fps diawali dengan nilai awal fps berupa 0 sebagai nilai fps sebelumnya, kemudian dijumlahkan dengan nilai fps masa sekarang, yaitu nilai frame per waktu ($\frac{1}{waktuakhir - waktuawal}$), dan kemudian dirataratakan dari fps sebelumnya dengan fps sekarang dengan dibagi 2 untuk menghasilkan perhitungan fps yang sebenarnya.

Sehingga perhitungan nilai fps direalisasikan pada Persamaan 2.

$$fps = \frac{1}{WaktuAkhir - WaktuAwal} \quad (1)$$

$$fpsSebenarnya = \frac{fpsAwal + fpsSekarang}{2} \quad (2)$$

Lalu untuk mengetahui akurasi keberhasilan dan akurasi kesalahan dari deteksi marka jalan menggunakan Persamaan 4 pada akurasi keberhasilan dan Persamaan 5 pada akurasi kesalahan. Dua Persamaan tersebut berpengaruh pada hasil pengujian untuk mengetahui objek yang dideteksi dapat mengenali objek atau tidak. Adapun Persamaan 4 dan Persamaan 5 sebagai berikut.

$$\sum D = \sum IB \sum IS \quad (3)$$

$$B\% = \frac{\sum IB}{\sum ID} \% \quad (4)$$



Gambar 10: Peletakkan Kamera Tampak Depan



Gambar 11: Peletakkan Kamera Tampak Belakang

$$S\% = 100\% - \frac{\sum IB}{\sum ID} \% \quad (5)$$

dengan fps = Frame per second, D = Deteksi, IB = Identifikasi Benar, IS = Identifikasi Salah, $B\%$ = Akurasi Berhasil dan $S\%$ = Akurasi Kesalahan

III. HASIL PENELITIAN DAN PEMBAHASAN

Pada penelitian ini dilakukan beberapa pengujian.

1. Pengujian Berdasarkan Lokasi.
2. Pengujian Berdasarkan Waktu.
3. Pengujian Berdasarkan Cuaca.
4. Pengujian Berdasarkan Setiap Objek Kelas.
5. Pengujian Berdasarkan Kecepatan Kendaraan.

Pengujian dilakukan sebanyak 26 kali pengujian dengan algoritma YOLO (You Only Look Once) dengan open source TensorFlow, GPU NVIDIA GeForce GTX 1050 Ti 4GB yang dibantu dengan CUDA yang mempermudah kegunaan GPU. Menggunakan perangkat proses, yaitu Laptop ASUS TUF Gaming FX504 Series dengan spesifikasi yang dijelaskan pada Tabel 1 berikut.

Pada penelitian ini menggunakan kamera webcam, yaitu Logitech C270 HD Webcam, yang diletakkan antara kaca mobil depan dan di belakang kaca spion depan mobil. Webcam ini diletakkan dengan dibantu

Tabel 2: Spesifikasi Webcam Logitech c270

Merk Kamera	Logitech C270 HD Webcam
Max Resolution	720p/30fps
Focus	Fixed
Lens	Standard
Field of View	60°
Microphone	Mono
Focal Length	4.0 mm
Video Capture	Up to 1280 × 720 pixels
Photos	Up to 3.0 megapixels
Ports	High Speed USB 2.0
Cable Length	1.5 m
Compatible with	Windows(R) 7, Windows 8, or Windows 10 or later
Weight	500g

Tabel 3: Tabel Pengujian Berdasarkan pada Lokasi

Lokasi	FPS				IB	IS	B%
	P	Si	So	M			
Jl. Jend. Sudirman	1.49	1.85	1.89	1.69	1005	100	90.95
Jl. Arifin Ahmad	1.87	1.87	1.73	1.67	1577	99	94.09
Jl. Pattimura	1.59	1.47	1.57	1.69	698	46	93.82
Jembatan Siak IV	1.69	1.73	1.48	1.89	748	46	94.21
Jl. Bandara SSK II	1.47	1.87	1.63	1.72	21	3	87.50

car camera holder yang berguna agar saat pengambilan objek lebih stabil dan hasilnya terlihat lebih sempurna. Peletakkan kamera bisa dilihat pada Gambar 10 dan 11.

Adapun spesifikasi kamera webcam Logitech C270 HD seperti pada Tabel 2.

Dalam pengujian berdasarkan pada lokasi bertujuan untuk mengetahui pengaruh lokasi pada hasil deteksi marka jalan. Pengujian dilakukan di waktu yang berbeda tetapi lokasi yang sama. Bertujuan untuk mengetahui dari pengambilan yang dilakukan, lokasi mana yang lebih baik. Ada jenis perbedaan waktu yang diuji, P = pagi, Si = siang, So = sore, dan M = malam. Berikut Tabel 3 yang merupakan tabel pengujian berdasarkan pada lokasi.

Dari perbandingan Tabel 3, nilai fps tertinggi berada di lokasi Jl. Arifin Ahmad dan akurasi berhasil tertinggi berada di lokasi Jembatan Siak IV sebesar 94.21%. Lokasi Jl. Arifin Ahmad dan Jembatan Siak IV memiliki badan marka jalan yang jelas, membuat pada saat pengujian dapat mengenali objek dengan baik.

Dalam pengujian berdasarkan pada waktu bertujuan untuk mengetahui pengaruh waktu pada hasil pengambilan deteksi marka. Perbandingan didapatkan dengan cara membandingkan tiap lokasi pada waktu yang sama. Berikut Tabel 4 yang merupakan tabel pengujian berdasarkan pada waktu.

Dari Tabel 4, nilai fps dan akurasi keberhasilan tertinggi adalah pada waktu siang hari dengan 1.76 fps dan 99.31%. Pada siang hari memiliki tingkat cahaya yang lebih tinggi dibanding waktu lain. Untuk kasus pada waktu malam, kondisi pencahayaan tidak sebanyak pada waktu pagi, siang, dan sore, karena saat pengujian, perangkat yang digunakan dalam keadaan performa yang tinggi menghasilkan fps yang

Tabel 4: Tabel Pengujian Berdasarkan pada Waktu

Waktu	FPS	IB	IS	B%
Pagi	1.62	1120	34	97.05
Siang	1.76	1151	8	99.31
Sore	1.66	897	37	96.04
Malam	1.73	881	215	80.38

Tabel 5: Tabel Perbandingan Waktu dengan Cuaca

Waktu	Cuaca	FPS	IB	IS	B%
Pagi	Berawan	1.52	496	12	97.64
	Cerah Berawan	1.78	624	22	96.59
Siang	Cerah Berawan	1.80	643	1	99.84
	Berawan	1.73	508	7	98.64
Sore	Cerah	1.53	205	1	99.51
	Berawan	1.75	692	36	95.05

Tabel 6: Tabel Hasil Pengujian Berdasarkan pada Cuaca

Cuaca	FPS
Cerah	1.53
Cerah Berawan	1.79
Berawan	1.67

tidak jauh berbeda dengan fps pada siang hari. Alhasil saat melakukan pengujian di malam hari, sistem dapat mengenali marka jalan dengan akurasi 80.93%. Walaupun pada marka yang kekurangan cahaya tidak bisa dikenali.

Dalam pengujian berdasarkan pada cuaca bertujuan untuk mengetahui pengaruh cuaca pada hasil pengambilan deteksi marka. Perbandingan didapatkan dengan cara membandingkan tiap lokasi pada cuaca yang sama. Ada 3 kategori cuaca yang diuji pada penelitian ini, yaitu Cerah, Cerah Berawan, dan Berawan. Kategori Cerah meliputi waktu pada sore hari, yaitu lokasi Jl. Pattimura, dan Jembatan Siak IV. Kategori Cerah Berawan ada 2 waktu yang mengalami cuaca ini, meliputi pagi, dan siang, lokasi Jl. Arifin Ahmad, dan Jembatan Siak IV untuk waktu pagi, dan Jl. Arifin Ahmad, dan Jembatan Siak IV untuk waktu siang. Kategori Berawan ada 3 waktu yang mengalami, diantaranya pagi, siang, dan sore, dengan lokasi Jl. Pattimura, Jl. Jend. Sudirman, dan Jl. Bandara SSK II untuk waktu pagi, masih lokasi yang sama pada waktu siang, dan Jl. Jend. Sudirman, Jl. Bandara SSK II, Jl. Arifin Ahmad untuk waktu sore. Pada waktu malam tidak ikut sertakan karena berada pada keadaan gelap, tidak bisa dibandingkan. Berikut Tabel 5 yang merupakan tabel perbandingan waktu dengan cuaca.

Kemudian rata-ratakan setiap kategori cuaca yang sama hingga mendapatkan hasil dari pengujian berdasarkan pada cuaca seperti Tabel 6 .

Dari Tabel 5, nilai fps dan akurasi yang lebih tinggi adalah pada waktu siang hari dengan cuaca cerah berawan. Kemudian pada Tabel 6, nilai fps yang lebih tinggi dari hasil rata-rata setiap cuaca adalah pada cuaca cerah berawan. Cerah Berawan merupakan cuaca yang dialami salah satunya pada siang hari menurut pengujian yang dilakukan, sedangkan cuaca cerah merupakan bagian dari cuaca yang satu-satunya pada sore hari. Cahaya pada siang hari lebih besar dari pada cahaya pada sore hari karena pada saat pengujian tingkat kecerahan siang hari lebih besar dari pada sore hari, membuat cuaca cerah berawan lebih besar cahayanya dibandingkan dengan cahaya pada cuaca cerah dipengujian yang dilakukan.

Dalam pengujian berdasarkan pada setiap objek kelas bertujuan untuk mengetahui kelas pada marka jalan yang dapat dideteksi dengan benar dan deteksi yang salah. Kelas berisi objek-objek yang

Tabel 7: Tabel Pengujian Berdasarkan pada Setiap Objek Kelas dengan Cahaya Matahari

Objek	IB	IS	S%
Ganda Utuh	-	-	-
Lajur Sepeda	-	-	-
Panah Dua Arah	0	1	100
Panah Kanan	-	-	-
Panah Kiri	-	-	-
Panah Lurus	5	0	0
Putus-Putus	2779	71	2.49
Utuh	354	6	1.67
Utuh dan Putus-Putus	-	-	-
Zebra Cross	30	1	3.23
Total	3168	79	2.43

Tabel 8: Tabel Pengujian Berdasarkan pada Setiap Objek Kelas dengan Cahaya Buatan

Objek	IB	IS	S%
Ganda Utuh	-	-	-
Lajur Sepeda	-	-	-
Panah Dua Arah	0	1	100
Panah Kanan	-	-	-
Panah Kiri	-	-	-
Panah Lurus	1	0	0
Putus-Putus	843	213	20.17
Utuh	34	1	2.86
Utuh dan Putus-Putus	-	-	-
Zebra Cross	3	0	0
Total	881	215	19.62

Tabel 9: Tabel Pengujian Berdasarkan Kecepatan Kendaraan pada Siang Hari

Kecepatan (km/jam)	FPS	IB	IS	B%
20	1.71	76	1	98.70
40	1.69	87	1	98.86
60	1.68	105	10	91.30
Total		268	12	95.71

mendeteksi marka jalan, dengan memiliki 10 objek yang akan dideteksi

Berdasarkan hasil Tabel 7 dan 8, pengujian saat menggunakan cahaya matahari memiliki persentase kesalahan yang lebih kecil dibandingkan saat menggunakan cahaya buatan. Menurut tabel tersebut objek yang meliputi garis ganda utuh, lajur sepeda, garis panah kanan, garis panah kiri, dan garis utuh dan putus-putus kosong karena objek tidak terlihat di lokasi yang diuji. Objek yang lain dapat terdeteksi, namun ada beberapa badan objek yang memiliki kesalahan identifikasi pada objek-objek yang dideteksi. Kesalahan disebabkan karena kondisi objek yang bentuknya sudah tidak sempurna, pencahayaan yang kurang yang dibuktikan pada pengujian pada malam hari, bahkan ada objek yang menyerupai objek yang lain.

Dalam pengujian berdasarkan pada kecepatan kendaraan bertujuan untuk mengetahui pengaruh kecepatan kendaraan (mobil) pada hasil pengambilan deteksi marka. Dengan melakukan uji coba kecepatan kendaraan, diantaranya ada 3 kecepatan yang diuji di waktu siang hari dan malam hari. Siang hari dipilih karena memiliki pencahayaan tertinggi, sedangkan malam hari dipilih karena pencahayaan terendah dan dibantu cahaya buatan, yaitu lampu sekitar jalan.

Pada pengujian waktu siang hari lokasi setiap kategori kecepatan berbeda, diantaranya pada kecepatan 20 km/jam dan 40 km/jam, lokasi berada di Jl. Pattimura. Sedangkan kecepatan 60 km/jam melakukan pengujian di Jembatan Siak IV. Berikut Tabel 9 yang merupakan tabel pengujian berdasarkan kecepatan kendaraan pada waktu siang hari.

Dari Tabel 9, nilai fps tertinggi adalah pada kecepatan kendaraan 20 km/jam. Pada kecepatan 20 km/jam sampai 40 km/jam mengalami peningkatan akurasi keberhasilan, dengan akurasi tertinggi pada kecepatan 40 km/jam sebesar 98.86%. Jumlah keseluruhan akurasi keberhasilan pada pengujian ini sebesar 95.71%.

Dari pengujian ini juga dapat disimpulkan jarak terjauh dan jarak terdekat dari jarak kendaraan dengan objek yang dideteksi. Jarak terjauh atau jarak minimum dapat dilakukannya pendeteksian adalah pada jarak 36.04 meter, dan jarak terdekat atau jarak maksimum dapat dilakukan deteksi objek adalah pada jarak 3.50 meter.

Pada pengujian waktu malam hari melakukan pengujian di lokasi yang sama, yaitu Jembatan Siak IV. Adapun Tabel 10 yang merupakan tabel pengujian berdasarkan kecepatan kendaraan pada waktu malam hari.

Dari Tabel 10, nilai fps tertinggi adalah pada kecepatan kendaraan 20 km/jam. Pada kecepatan 20

Tabel 10: Tabel Pengujian Berdasarkan Kecepatan Kendaraan pada Malam Hari

Kecepatan (km/jam)	FPS	IB	IS	B%
20	1.95	90	22	80.36
40	1.89	97	23	80.83
60	1.91	106	23	82.17
Total		293	68	81.16



Gambar 12: Contoh Hasil Pengujian Deteksi Marka Jalan

km/jam sampai 60 km/jam mengalami peningkatan akurasi keberhasilan, dengan akurasi tertinggi pada kecepatan 60 km/jam sebesar 82.17%. Jumlah keseluruhan akurasi keberhasilan pada pengujian ini sebesar 81.16%.

Dari pengujian ini jarak terjauh atau jarak minimum dapat dilakukannya pendeteksian adalah pada jarak 35.09 meter, dan jarak terdekat atau jarak maksimum dapat dilakukan deteksi objek adalah pada jarak 3.52 meter.

Dari pengujian yang telah dilakukan, didapatkan hasil dari pengujian deteksi marka jalan seperti yang terlihat pada salah satu pengujian di siang hari dengan akurasi persentase benar keseluruhan sebesar 92.50%, artinya pengujian ini dapat mendeteksi objek marka jalan dengan baik.

IV. Kesimpulan

Dari hasil pengujian yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut :

- Pengujian mendeteksi dan mengidentifikasi marka jalan telah berhasil mengenali objek.
- Pengujian berdasarkan lokasi, nilai fps tertinggi berada di Jl. Arifin Ahmad, dan akurasi keberhasilan tertinggi berada di Jembatan Siak IV sebesar 94.21%.
- Pada pengujian berdasarkan waktu, nilai fps tertinggi dan akurasi keberhasilan tertinggi pada waktu siang hari dengan nilai 1.76 fps dan sebesar 99.31%.
- Pada pengujian berdasarkan cuaca, nilai fps dan akurasi tertinggi adalah pada waktu siang hari dengan cuaca cerah berawan dengan nilai 1.80 fps dan 99.84%.
- Pada pengujian berdasarkan setiap objek kelas, dapat mengenali objek dengan baik dengan akurasi 97.57% pada objek yang menggunakan cahaya matahari, dan menggunakan cahaya buatan dengan 80.38%. Pada pengujian berdasarkan kecepatan kendaraan, menghasilkan total akurasi sebesar 95.71% pada siang hari, dan 81.16% pada malam hari.
- Pengujian jarak menunjukkan dapat mendeteksi objek dari jarak 3.50 meter hingga 36.04 meter pada pengujian siang hari. Sedangkan pada malam hari dapat mendeteksi objek dari jarak 3.54 meter hingga 35.09 meter.

Daftar Pustaka

- [1] S. Kumar Vishwakarma, Akash, and D. Singh Yadav, "Analysis of Lane Detection Techniques using openCV," 2015.
- [2] T. Chen, Z. Chen, Q. Shi, and X. Huang, "Road Marking Detection and Classification Using Machine Learning Algorithms," 2015.
- [3] W. Zhang, Z. Mi, Y. Zheng, Q. Gao, and W. Li, "Road Marking Segmentation Based on Siamese Attention Module and Maximum Stable External Region," China, 2019.

- [4] Y. Bengio et al., "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016.
- [6] I Wayan Suartika Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," Institut Teknologi Sepuluh Nopember, 2016.
- [7] Dewi, S. Rosita, et al., "Deep learning object detection pada video menggunakan tensorflow dan convolutional neural network," 2018.
- [8] M. Nada, "Penerapan Deep Learning Menggunakan Convolutional Neural Network (CNN)," 2 Januari 2019. [Online]. Available: <https://medium.com/@mukhlisatunnada02/penerapan-deep-learningmenggunakan-convolutional-neural-network-cnn-d02dc6532f5b>. Terakhir diakses pada tanggal 12 April 2020.
- [9] S. Sena, "Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN)," 13 November 2017 [Online]. Available: <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>. Terakhir diakses pada tanggal 12 April 2020.
- [10] R. Zain Fadillah, "Model Penerjemah Bahasa Isyarat Indonesia (Bisindo) menggunakan Convolutional Neural Network," 2020.
- [11] M. Nada, "Kegunaan Layar Pooling Pada Penerapan Deep Learning menggunakan Convolutional Neural Network," 2 Januari 2019. [Online]. Available: <https://medium.com/@mukhlisatunnada02/kegunaan-layarpoolin-pada-penerapan-deep-learning-menggunakan-convolutionalneural-network-140146078f28>. Terakhir diakses pada tanggal 12 April 2020.
- [12] T. Nurhikmat, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek," 2018.
- [13] R. Dwi Novyantika, "Deteksi Tanda Nomor Kendaraan Bermotor Pada Media Streaming dengan Algoritma Convolutional Neural Network Menggunakan Tensorflow," 2018.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- [15] Menteri Perhubungan Republik Indonesia, Keputusan Menteri Perhubungan Nomor : KM 60 Tahun 1993 Tentang Marka Jalan, Jakarta, 1993.
- [16] Menteri Perhubungan Republik Indonesia, Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 34 Tahun 2014 Tentang Marka Jalan, Jakarta, 2014.
- [17] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [18] M. Shahid Setiawan, "Deteksi dan Identifikasi Nomor Registrasi Plat Kendaraan Indonesia Berbasis Convolutional Neural Network (CNN)," 2020.