

BAB II TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan teori-teori terkait penelitian yang bersumber dari jurnal dan sumber terpercaya lainnya. Tujuan dari penulisan bab ini adalah untuk memahami konsep dan teori penyelesaian masalah yang digunakan. Teori yang akan dibahas dalam bab ini yaitu definisi kendaraan, *artificial intelligence*, *deep learning*, YOLOv3, *openCv*, *computer vision*, serta penelitian terdahulu.

2.1 Kendaraan

Menurut Undang-undang Nomor 22 Tahun 2009 pasal 1 angka 7 Tentang Lalu Lintas dan Angkutan Jalan, kendaraan adalah suatu sarana angkut di jalan yang terdiri atas kendaraan bermotor dan kendaraan tidak bermotor. Berdasarkan pengertian tersebut, maka kendaraan dibagi menjadi:

1. **Kendaraan bermotor**, adalah setiap kendaraan yang digerakkan oleh peralatan mekanik berupa mesin selain kendaraan yang berjalan di atas rel. yang termasuk kedalam jenis kendaraan bermotor yaitu sepeda motor, mobil penumpang, mobil bus, mobil barang, dan kendaraan khusus.
2. **Kendaraan tidak bermotor**, adalah setiap kendaraan yang digerakkan oleh tenaga manusia atau hewan.

Berdasarkan fungsinya, kendaraan bermotor dibedakan menjadi kendaraan bermotor pribadi/perorangan dan kendaraan bermotor umum. Kendaraan bermotor umum adalah setiap kendaraan yang digunakan untuk angkutan barang dan/atau orang dengan dipungut biaya.

2.2 Kecerdasan Buatan (*Artificial Intelligence*)

Artificial intelligence (AI) adalah simulasi dari kecerdasan yang dimiliki oleh manusia yang dimodelkan di dalam mesin dan diprogram agar bisa berpikir seperti manusia. Menurut *McLeod* dan *Schell*, kecerdasan buatan adalah aktivitas penyediaan mesin seperti komputer dengan kemampuan untuk menampilkan perilaku yang akan dianggap sama cerdasnya dengan jika kemampuan tersebut ditampilkan oleh manusia.

Secara garis besar sebuah kecerdasan buatan dapat melakukan salah satu dari keempat faktor dibawah ini:

1. *Acting humanly*, sistem yang dapat bertindak layaknya manusia.
2. *Thinking humanly*, sistem yang bisa berpikir seperti halnya manusia.
3. *Think rationally*, sistem yang mampu berpikir secara rasional.
4. *Act rationally*, sistem yang mampu bertindak secara rasional.

Berdasarkan (Budiarjo, 2020) Beberapa contoh penerapan AI yaitu:

1. Sistem kemudi otomatis pada mobil, contohnya mobil Tesla (Amerika) yang mempunyai sistem *autopilot* pada mobil produksinya.
2. Sistem penguncian wajah (*Face Unlock*) pada *smartphone* yang dapat mengenali wajah pemilikinya.
3. *Virtual Assistant* seperti SIRI dari *Apple*, Cortana dari *Microsoft*, dan *Google Assistant* dari *google* yang dapat menjadi asisten virtual sesuai dengan preferensi pengguna.

2.3 *Deep Learning*

Deep learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar (Nurhikmat, 2020). *Deep learning* adalah metode pembelajaran yang dilakukan oleh mesin dengan cara meniru bagaimana sistem dasar otak manusia bekerja. Sistem dasar otak manusia bekerja ini disebut *neural networks*. Itulah kenapa *deep learning* disebut menggunakan *artificial neural networks* yang dengan kata lain menggunakan *neural networks* buatan.

Deep learning semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *computer vision*, *speech recognition*, dan *natural language processing*. *Feature engineering* adalah salah satu fitur utama dari *deep learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi namun sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga (Nurhikmat, 2020). Adapun penerapan teknologi *deep learning*

sendiri sudah cukup banyak dijumpai seperti *google photos*, *face unlock* di *smartphone*, sistem tilang otomatis dan sebagainya.

2.4 *You Only Look Once V3 (YOLOv3)*

YOLO adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. YOLO menggunakan pendekatan dengan menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Pendeteksian objek dilakukan dengan membingkai objek yang akan dideteksi sebagai *regression problem* dan memisahkan *special* pada *bounding boxes* dan *class probabilities*. Dengan menggunakan *single neural network* untuk memprediksi *bounding boxes* dan *class probabilities* dari seluruh gambar pada satu kali evaluasi (Isnaini, 2020).

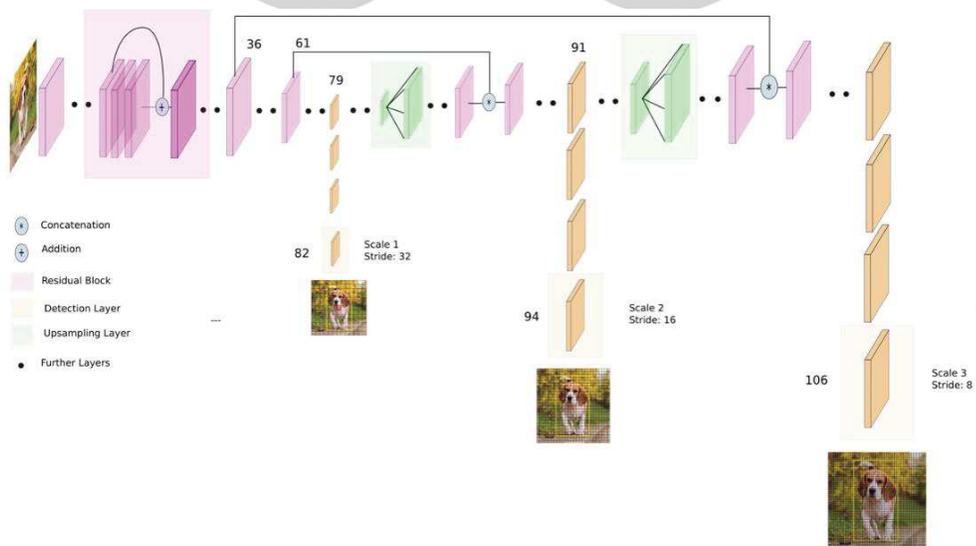
YOLOv3 merupakan versi ketiga dari algoritma *object detection* YOLO yang dikembangkan oleh Joseph Redmon. YOLOv3 memprediksi skor objektivitas (keyakinan) untuk setiap kotak pembatas menggunakan regresi logistik. Tidak seperti YOLO dan YOLOv2 yang memprediksi keluaran pada lapisan terakhir, YOLOv3 memprediksi kotak pada 3 skala berbeda. YOLOv3 sendiri mengekstraksi fitur pada skala tersebut dengan konsep yang mirip dengan fitur jaringan piramida. YOLOv3 juga memiliki dataset yaitu *coco.names* yang berisi 80 kelas objek yang sudah dilatih oleh YOLO. Kode kelas dalam YOLO dimulai dari 0. Objek-objek yang telah dilatih oleh YOLOv3 dapat dilihat pada Gambar 2.1.



1	person	28	tie	55	donut
2	bicycle	29	suitcase	56	cake
3	car	30	frisbee	57	chair
4	motorbike	31	skis	58	sofa
5	aeroplane	32	snowboard	59	pottedplant
6	bus	33	sports ball	60	bed
7	train	34	kite	61	diningtable
8	truck	35	baseball bat	62	toilet
9	boat	36	baseball glove	63	tvmonitor
10	traffic light	37	skateboard	64	laptop
11	fire hydrant	38	surfboard	65	mouse
12	stop sign	39	tennis racket	66	remote
13	parking meter	40	bottle	67	keyboard
14	bench	41	wine glass	68	cell phone
15	bird	42	cup	69	microwave
16	cat	43	fork	70	oven
17	dog	44	knife	71	toaster
18	horse	45	spoon	72	sink
19	sheep	46	bowl	73	refrigerator
20	cow	47	banana	74	book
21	elephant	48	apple	75	clock
22	bear	49	sandwich	76	vase
23	zebra	50	orange	77	scissors
24	giraffe	51	broccoli	78	teddy bear
25	backpack	52	carrot	79	hair drier
26	umbrella	53	hot dog	80	toothbrush
27	handbag	54	pizza	81	

Gambar 2.1 Kelas objek YOLOv3 (Penulis, 2021)

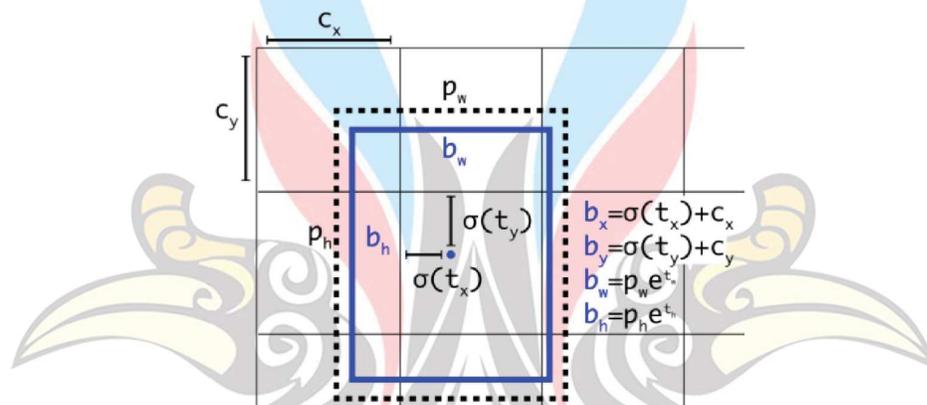
Pada dataset COCO dapat diprediksi 3 kotak pada setiap skala sehingga tensornya adalah $N \times N \times [3 \times (4 + 1 + 80)]$ untuk 4 *offset* kotak pembatas, 1 prediksi objek dan 80 prediksi kelas (Redmon & Farhadi, 2018). *Architecture* dari YOLOv3 dapat dilihat pada gambar 2.1.



Gambar 2.2 YOLOv3 *architecture* (lestari et al, 2019)

Dalam proses deteksi, YOLO menggunakan *input* gambar dengan *network size* atau ukuran gambar nilai kelipatan 32. Semakin besar *network size* maka semakin akurat komputer dalam memprediksi objek, akan tetapi proses komputasi akan semakin lambat. Sebaliknya jika *network size* yang diberikan semakin kecil maka proses komputasi menjadi lebih cepat namun akurasi dalam deteksi menjadi kurang baik. Dalam hal ini, YOLOv3 menggunakan *network size* sebesar 416 x 416 (Redmon & Farhadi, 2018).

YOLOv3 memberikan hasil prediksi berupa 6 nilai yang terdiri dari titik koordinat *bounding box* (t_x, t_y, t_w, t_h), *confidence* dan *class probability* yang dapat dilihat pada gambar 2.2.



Gambar 2.3 Bounding Box YOLOv3 (Redmon & Farhadi, 2018)

YOLOv3 memprediksi skor kelas setiap *bounding box* menggunakan *logistic regression* untuk menghindari *bounding box* yang tumpang tindih dengan *bounding box* yang lain. Selain itu, terdapat *threshold* untuk memprediksi kelas pada suatu objek yang terdeteksi. Jika terdapat suatu skor kelas yang mempunyai nilai tinggi melebihi *threshold* maka kelas tersebut diberikan pada objek yang terdeteksi (Redmon & Farhadi, 2018).

2.5 OpenCv

OpenCv adalah sebuah *library* yang berisi fungsi-fungsi pemrograman untuk teknologi *computer vision* secara *real time*. *OpenCv* bersifat *open source*, bebas digunakan untuk hal-hal yang bersifat akademis maupun komersial. Di dalam *OpenCv*, terdapat *interface* untuk bahasa pemrograman C, C++, *Python*,

dan nantinya *java* yang dapat berjalan pada *windows*, *linux*, *android* dan *mac* (Kurniawan, 2015).

OpenCv memiliki lebih dari 2500 algoritma yang dioptimalkan. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak pergerakan kamera, melacak objek bergerak, dan lain-lain. *Library* ini digunakan secara luas di perusahaan, kelompok penelitian, dan badan pemerintah (Kurniawan, 2015).

Fitur-fitur yang terdapat pada *OpenCv* (Kurniawan, 2015) antara lain:

1. Manipulasi data *image* (alokasi, rilis, duplikasi, pengaturan, konversi).
2. *Image* dan I/O video (masukan berbasis *file* dan kamera, keluaran *image video file*).
3. Manipulasi matriks dan vektor serta aljabar linear (produk, solusi, *eigenvalues*, SVD).
4. Beragam struktur data dinamis (daftar, baris, grafik).
5. Dasar pengolahan citra (filter, deteksi tepi, deteksi sudut, pengambilan sampel, interpolasi, konversi warna, operasi morfologi, *histogram*).
6. Analisis struktur (komponen yang berhubungan, pengolahan kontur, transformasi jarak, variasi momen, transformasi *Hough*, perkiraan *polygonal*, penyesuaian garis, *delaunay triangulation*).
7. Kalibrasi kamera (menemukan dan menelusuri pola kalibrasi, dasar estimasi matriks, estimasi homografi, korespondensi stereo).
8. Analisis gerakan (*optical flow*, segmentasi gerakan, penelusuran).
9. Pengenalan objek (metode *eigen*, HMM).
10. Dasar *Graphical User Interface* atau GUI (menampilkan *image video*, penanganan *mouse* dan *keyboard*, *scroll-bars*).
11. Pelabelan *image* (garis, poligon, gambar teks).

Modul-modul yang terdapat pada *OpenCv* (Kurniawan, 2015) antara lain:

1. *Cv*-fungsi utama *OpenCv*.
2. *Cvaux*-fungsi penolong *OpenCv*.
3. *Excore*-pendukung struktur data dan aljabar linear.
4. *Highui*-fungsi *Graphical User Interface* (GUI).

2.6 Computer Vision

Computer vision adalah sebuah disiplin ilmu yang bertujuan untuk membuat suatu keputusan dalam mengenali objek fisik nyata dan keadaan berdasarkan sebuah citra. *Computer vision* berkaitan dengan pemodelan dan meniru penglihatan manusia dengan menggunakan perangkat lunak dan perangkat keras pada komputer (Dewi, 2018). Kemampuan dari *computer vision* sendiri meliputi:

1. *Object detection*, mengenali sebuah objek ada pada *scene* dan mengetahui dimana batasannya.
2. *Recognition*, menempatkan label pada objek
3. *Description*, menugaskan properti kepada objek.
4. *3D Inference*, menafsirkan adegan 3D dari 2D.
5. *Interpreting motion*, menafsirkan gerakan.

2.6 Akurasi, Presisi, Sensitivitas, dan Spesifitas

Pengukuran kinerja suatu sistem klasifikasi merupakan hal yang penting. Salah satu metode pengukuran kinerja suatu sistem klasifikasi adalah *confusion matrix* yang memiliki 4 istilah sebagai representasi hasil proses klasifikasi yaitu:

1. *True Positive (TP)*
True positive merupakan data positif yang terdeteksi dengan benar.
2. *True Negative (TN)*
True Negative merupakan data negatif yang terdeteksi dengan benar.
3. *False Positive (FP)*
False positive merupakan data negatif namun terdeteksi positif.
4. *False Negative (FN)*
False negative merupakan data positif namun terdeteksi sebagai data negatif.

Berdasarkan nilai TP, TN, FP, FN dapat diperoleh nilai akurasi, presisi, sensitivitas, dan spesifisitas (Solichin, 2017).

Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Nilai akurasi dapat dilihat pada persamaan (2.1)

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.1)$$

Nilai presisi adalah jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasikan positif. Nilai presisi didapatkan dengan persamaan (2.2).

$$\text{Presisi} = \frac{TP}{(FP+TP)} \times 100\% \quad (2.2)$$

Nilai sensitivitas merupakan sebuah persentase jumlah sensor mendeteksi dengan benar saat terdapat objek diseluruh percobaan. Nilai sensitivitas didapatkan dengan persamaan (2.3).

$$\text{Sensitivitas} = \frac{TP}{(FN+TP)} \times 100\% \quad (2.3)$$

Nilai spesifisitas merupakan sebuah persentase jumlah sensor mendeteksi dengan benar saat tidak terdapat objek diseluruh percobaan. Nilai spesifisitas didapatkan dengan persamaan (2.3).

$$\text{Spesifisitas} = \frac{TN}{(FP+TN)} \times 100\% \quad (2.4)$$

2.7 Penelitian Terdahulu

Rangkuman dari penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang dilakukan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

Peneliti	Judul	Metode	Hasil
Akbar et al,	<i>Traffic IP Camera</i> untuk Menghitung Roda Empat Menggunakan Metode Luasan Pikel	Luasan Pikel	Dapat menghitung kendaraan roda empat dengan tingkat akurasi 82,18% pada waktu siang hari dan tingkat akurasi 88,30% pada waktu malam hari.
Purnomo et al, 2018	<i>Intellegence Vehicle Counting</i> Menggunakan Metode <i>Combination Value Saturation</i> Pada Video Lalu Lintas	<i>Combination Value Saturation</i>	Mampu mendeteksi kendaraan yang melintas serta melakukan counting dengan akurasi 63,38% dari 2 video yang memiliki aspect ratio berbeda

Peneliti	Judul	Metode	Hasil
Premamayudu, 2020	<i>Object Detecting Model for Autonomus Car Driving Using YOLOv3</i>	Algoritma YOLOv3	Dapat mendeteksi mobil sesuai dengan model deteksi objek
Ershadi et al, 2017	<i>Vehicle Tracking and Counting System in Dusty Weather with Vibrating Camera Conditions</i>	<i>Improved Background subtraction</i>	Sistem bekerja dengan baik di semua kondisi. Tingkat deteksi dari metode yang diusulkan mencapai 98,9% pada kondisi normal, 81,1% pada kondisi berdebu, 79,8% pada kondisi kamera bergetar, dan 83,4% pada kondisi salju.
Hakim et al, 2020	Perekaman Otomatis Berdasarkan Deteksi Objek Manusia Pada CCTV Menggunakan Metode <i>You Only Look Once V3 (YOLOv3)</i>	YOLOv3	Akurasi model yang dilatih memiliki akurasi 100% dalam mendeteksi manusia. <i>Sampling</i> yang diuji dapat dikenali sebagai <i>person</i> . Hasil <i>real time</i> dengan mendeteksi manusia menghasilkan akurasi 100%.
(Penulis, 2021)			

Pada penelitian ini akan membangun sistem klasifikasi dan *counting* kendaraan menggunakan metode YOLOv3. Selanjutnya akan dicari tingkat akurasi, presisi, sensitivitas, dan spesifisitas dari sistem yang belum dilakukan pada penelitian terdahulu.