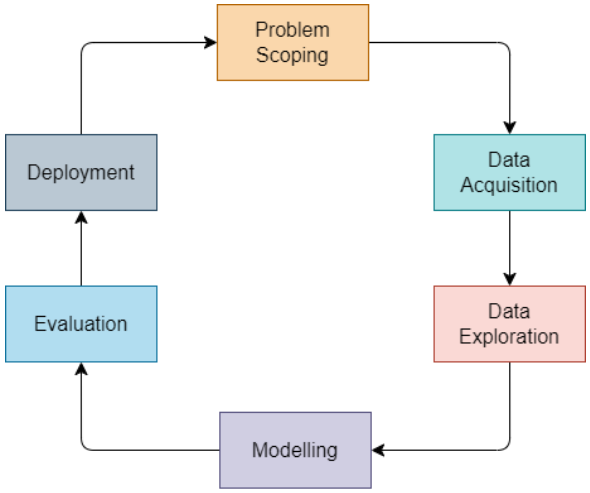
BAB III
METODE PENELITIAN

3.1. Objek Penelitian

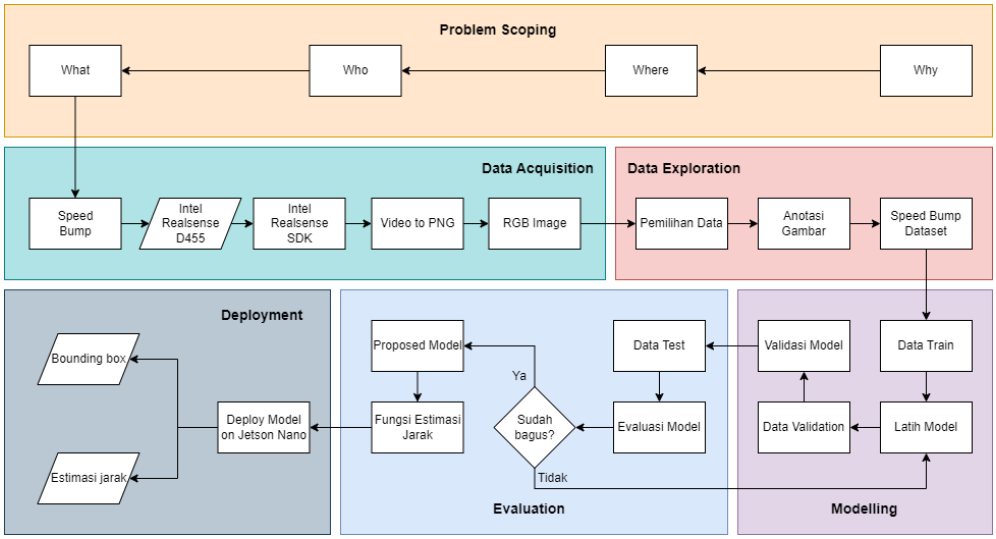
Objek penelitian dalam penelitian skripsi ini adalah model sistem deteksi dan estimasi jarak yang dapat mendeteksi *speed bump* dan melakukan pengukuran jarak antara kendaraan otonom terhadap objek yang dideteksi menggunakan algoritma *deep learning* dengan arsitektur YOLOv8.

3.2. Metode Penelitian

Pada penelitian ini metode penelitian yang digunakan yaitu AI Project Cycle. AI Project Cycle adalah suatu proses dalam membuat proyek AI sehingga AI yang dikembangkan bersifat utuh dan memiliki tingkat keberhasilan yang siginifikan. Berdasarkan Gambar 3.1, AI Project Cycle memiliki 6 tahapan yaitu *problem scoping*, *data acquisition*, *data exploration*, *modelling*, *evaluation* dan *deployment* (Azimah dkk., 2022). Gambar 3.2 menjelaskan lebih detail bagaimana penelitian akan dilakukan.



Gambar 3.1 AI Project Cycle



Gambar 3.2 Alur Pengerjaan

3.2.1 *Problem Scoping*

Pada tahap ini, peneliti memetakan batasan masalah yang akan diselesaikan. Untuk mempermudah proses *problem scoping* dapat menggunakan metode 4W, yaitu:

a. *Who*: siapa yang terlibat dalam masalah tersebut.
b. *What*: masalah tersebut tentang apa.
c. *Where*: dimana masalah tersebut terjadi.
d. *Why*: mengapa masalah tersebut bisa terjadi dan mengapa harus diselesaikan.

Berdasarkan metode 4W tersebut, batasan masalah pada penelitian ini adalah *who*: kendaraan otonom, *what*: kemampuan deteksi *speed bump* bermarka, *where*: *speed bump* di jalan raya di Indonesia, dan *why*: tidak memiliki kemampuan deteksi *speed bump* dan berisiko menyebabkan kecelakaan.

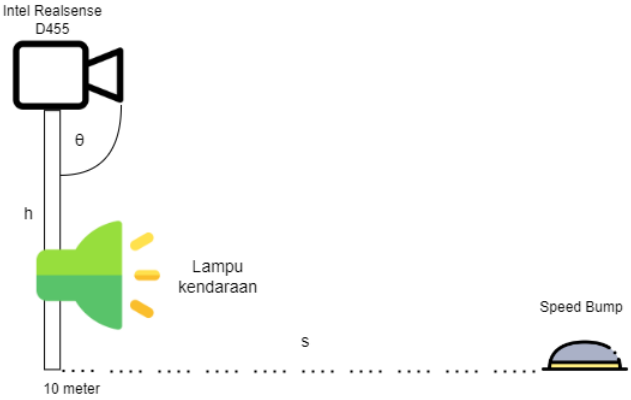
3.2.2 *Data Acquisition*

Pada tahap ini, peneliti melakukan pengumpulan data yang akan digunakan sebagai dataset untuk melatih model. Pengumpulan data dilakukan dengan melakukan pengambilan video *speed bump* di depan SLB Negeri Cileunyi. Pengambilan video dilakukan dengan menggunakan kamera Intel RealSense D455 dan Intel RealSense SDK sebagai perangkat lunak untuk mengoperasikan kameranya. Proses pengambilan video akan dilakukan pada siang dan malam hari.

Pada siang hari pengambilan video dilakukan ketika cuaca cerah, tidak terlalu terik juga tidak hujan, dengan intensitas cahaya ada pada rentang 20.000-30.000 lux. Pada malam hari terdapat 2 kondisi pengambilan video, yaitu pengambilan video tanpa dibantu penerangan dari kendaraan dan pengambilan video yang dibantu penerangan dari kendaraan, dalam kasus ini menggunakan lampu depan sepeda motor. Pengambilan video pada malam hari tanpa bantuan penerangan dari kendaraan memiliki nilai intensitas cahaya sebesar 0-5 lux. Sedangkan untuk pengambilan video pada malam hari dengan bantuan penerangan dari kendaraan memiliki nilai intensitas cahaya sebesar 5-50 lux. Oleh karena itu, terdapat 3 kondisi pengambilan video, 1 kondisi pada siang hari dan 2 kondisi pada malam hari. Dalam setiap kondisi terdapat 4 skenario pengambilan video dengan ketinggian dan sudut kamera yang berbeda-beda. Pengambilan video dilakukan dalam rentang jarak 1-10 meter terhadap objek. Pengambilan video akan dilakukan setiap jarak 1 meter sehingga dalam satu skenario ada 10 video yang telah diambil. Total video yang diambil yaitu 120 video dari 12 skenario yang ada.

Kamera akan diatur pada ketinggian yang berbeda-beda untuk mendapatkan gambar yang berbeda-beda pula. Pertama, kamera akan diatur setinggi 0.5 meter, kemudian 1 meter, 1.5 meter, dan terakhir 2 meter sesuai dengan penelitian (Yang, 2022). Sudut kamera diatur berdasarkan pada tinggi kamera dan jarak maksimal terhadap objek sejauh 10 meter. Gambar 3.3 dan Gambar 3.4 menjelaskan ilustrasi pengambilan video. Variabel s merupakan jarak kamera terhadap objek, h merupakan tinggi kamera, dan θ merupakan sudut kemiringan kamera terhadap permukaan tanah. Pengambilan video dilakukan selama 10 detik dengan pengaturan FPS sebesar 10 FPS. Gambar 3.5 dan Gambar 3.6 menunjukkan proses pengambilan video baik pada siang hari maupun malam hari.

Gambar 3.3 Ilustrasi Pengambilan Video ketika Siang dan Malam Hari tanpa Penerangan



Gambar 3.4 Ilustrasi Pengambilan Video ketika Malam Hari dengan Penerangan

(a) (b)

Gambar 3.5 Pengambilan Video pada (a) Siang Hari dan (b) Malam Hari

(a)

(b)

Gambar 3.6 Pengambilan Video pada Malam Hari (a) tanpa Penerangan dan (b) dengan Penerangan

Video yang telah diambil akan dikonversi menjadi gambar berformat PNG. Proses konversi gambar menggunakan Robot Operating System (ROS) karena video hasil perekaman dengan menggunakan Intel RealSense D455 memiliki format yang hanya dapat dibuka menggunakan ROS. Setelah melakukan instalasi ROS selanjutnya mengonversi video menjadi gambar. Sebelumnya telah disebutkan bahwa selama proses pengambilan video, FPS kamera diatur menjadi 10 FPS sehingga hasil konversi menunjukkan bahwa terdapat 100 gambar (10 FPS x 10 detik) dari satu video yang diambil. Terdapat total 12 skenario pengambilan video maka total video yang telah diambil yaitu sebanyak 120 video. Oleh karena itu, jumlah gambar hasil konversi terdapat 12000 gambar. Teknik pengambilan video pada kondisi siang hari, malam hari dengan penerangan, dan malam hari tanpa penerangan adalah sama. Informasi mengenai teknis pengambilan video pada 3 kondisi tersebut dijelaskan pada Tabel 3.1 berikut.

Tabel 3. 1
Skenario Pengambilan Data

Skenario	**Tinggi Kamera (m)**	**Sudut Kamera (°)**	**Jarak (m)**	**Durasi Video (detik)**	**Jumlah *Frame***
1	0.5	87	1	10	100
			2	10	100

Skenario	Tinggi Kamera (m)	Sudut Kamera (°)	Jarak (m)	Durasi Video (detik)	Jumlah *Frame*
			3	10	100
			4	10	100
			5	10	100
			6	10	100
			7	10	100
			8	10	100
			9	10	100
			10	10	100
2	1	85	1	10	100
			2	10	100
			3	10	100
			4	10	100
			5	10	100
			6	10	100
			7	10	100
			8	10	100
			9	10	100
			10	10	100
3	1.5	81	1	10	100
			2	10	100
			3	10	100
			4	10	100
			5	10	100
			6	10	100
			7	10	100
			8	10	100
			9	10	100
			10	10	100

Skenario	Tinggi Kamera (m)	Sudut Kamera (°)	Jarak (m)	Durasi Video (detik)	Jumlah *Frame*
4	2	79	1	10	100
			2	10	100
			3	10	100
			4	10	100
			5	10	100
			6	10	100
			7	10	100
			8	10	100
			9	10	100
			10	10	100
				Total (Gambar)	4000

3.2.3 *Data Exploration*

Data yang telah dikumpulkan akan dieksplorasi guna mendapatkan informasi penting pada data. Gambar yang telah dikonversi akan dipilih sebagai sampel agar *dataset* beragam. Dari 100 gambar akan diambil sampel sebanyak 50 gambar dengan interval satu gambar, sehingga total terdapat 2000 gambar sampel dari masing-masing kondisi pengambilan video (4 skenario). Semua sampel dari 12 skenario pengambilan video akan digabungkan sehingga data menjadi lebih beragam. Total sampel *dataset* sebanyak 6000 gambar (3 kondisi pengambilan video: 1 kondisi siang dan 2 kondisi malam). Pada penelitian ini gambar yang diperlukan adalah gambar yang berformat JPG. Oleh karena itu, 6000 gambar PNG tersebut akan dikonversi menjadi gambar berformat JPG menggunakan program konversi berbahasa Python.

Dataset akan dibagi menjadi data *train*, data *val*, dan data *test* dengan masing-masing perbandingan sebesar 70% : 20% : 10% sehingga terdapat 4200 gambar untuk *train*, 1200 gambar untuk *validation*, dan 600 gambar untuk *testing*. Setelah itu, data *train* dan data *val* akan dilakukan anotasi berbentuk *polygon*

menggunakan perangkat lunak VGG Image Annotator (VIA). *Dataset* hasil anotasi akan disimpan dalam *format* JSON yang selanjutnya akan digunakan untuk melatih model dengan arsitektur Mask R-CNN. Arsitektur YOLOv4 Tiny dan YOLOv8 tidak mendukung *format* JSON dalam *dataset* yang akan digunakan. Oleh karena itu, hasil anotasi menggunakan VIA harus dikonversi ke *format* yang mendukung. *Format file* hasil anotasi menggunakan VIA berupa VIA JSON sedangkan YOLO memerlukan *file* anotasi berformat YOLO TXT. Oleh karena itu, *file* hasil anotasi harus dikonversi menggunakan fungsi konversi yang telah disediakan oleh Ultralytics atau bisa juga dengan menggunakan Roboflow.

3.2.4 *Modelling*

Pada tahap ini, peneliti akan membuat model deteksi objek menggunakan algoritma *deep learning*. Sebelum itu, *dataset* yang akan digunakan dibagi menjadi 70% data untuk *training, 20%* data untuk *validation*, dan 10% data untuk *testing*. Arsitektur yang akan digunakan ialah arsitektur Mask R-CNN, YOLOv4 Tiny, dan YOLOv8n. Perancangan model deteksi *speed bump* menggunakan arsitektur Mask R-CNN akan dilakukan secara lokal pada komputer dengan spesifikasi OS Linux Ubuntu 18.04, CPU Intel i7-9700F, GPU NVIDIA GeForce GTX 1060, RAM 32 GB, CUDA Toolkit 11.8, cuDNN 8.6, Tensorflow 2.8, Anaconda 3, dan Python 3.9. Perancangan model deteksi *speed bump* menggunakan arsitektur YOLOv4 Tiny akan dilakukan secara *online* menggunakan *platform* Google Colaboratory dengan *framework* Darknet, Python 3.10, dan GPU T4. Perancangan model deteksi *speed bump* menggunakan arsitektur YOLOv8n akan dilakukan secara *online* menggunakan *platform* Kaggle. Kaggle merupakan suatu *platform data science* dan *machine learning* yang bukan hanya menyediakan *dataset* publik tetapi juga menyediakan layanan pelatihan model secara gratis. *Environment* Kaggle yang digunakan memiliki spesifikasi GPU T4, Ultralytics 8.1.44, OpenCV 4.6.0, Torch 1.8.0, Torchvision 0.9.0, dan Python versi 3.10. Model pertama kali akan dilatih menggunakan *dataset training* kemudian akan dilakukan validasi menggunakan *dataset validation*. Untuk mendapatkan performa model yang sesuai dengan yang diinginkan perlu dilakukan *hyperparameter tuning* ketika proses pelatihan model. Daftar *hyperparameter* yang digunakan adalah jumlah *epochs, batch size,* jenis *optimizer* dan *learning rate*.

3.2.5 *Evaluation*

Pada tahapan evaluasi, model akan diuji menggunakan *dataset testing* untuk mengetahui performa model hasil pelatihan. Setelah model memiliki performa yang mumpuni, model akan diintegrasikan dengan program estimasi jarak sehingga ketika model mendeteksi objek dapat terlihat estimasi jarak terhadap objek yang dideteksi. Model akan dievaluasi menggunakan 5 metrik evaluasi, yaitu Akurasi, Presisi, mAP, *Recall*, dan F1 *Score*. Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai. Presisi (*precision*) merupakan presentase kasus yang diprediksi model yang terjadi sesuai dengan kenyataannya. mAP merupakan rata-rata dari persentase nilai presisi. Nilai *recall* berasal dari pecahan kasus yang terjadi dan diprediksi dengan tepat oleh model. Sedangkan F1 *Score* merupakan nilai rata-rata dari perbandingan presisi dan *recall*. Selain itu, evaluasi model juga akan dilakukan dengan menguji model untuk melakukan deteksi dan estimasi jarak objek pada lingkungan nyata atau *speed bump* yang ada di jalan

3.2.6 *Deployment*

Pada tahap ini model akan diimplementasikan pada sebuah perangkat keras sehingga *output* akhirnya berupa sistem tertanam. Jetson Nano digunakan sebagai *Single Board Computer* yang berfungsi sebagai perangkat komputasi model deteksi objek. *Single Board Computer* (SBC) merupakan sebuah komputer yang memiliki ukuran relatif lebih kecil dan hemat daya dibandingkan komputer pada umumnya. Meskipun begitu, SBC memiliki fungsi yang sama seperti komputer pada umumnya. Jetson Nano dipilih karena mendukung pengembangan model *machine learning* dan kecerdasan buatan, terlihat dari tersedianya modul GPU. Jetson Nano yang digunakan memiliki spesifikasi CPU Quad-core ARM Cortex-A57 @ 1.43 GHz, GPU 128-core NVIDIA Maxwell, dan RAM 4 GB. *Output* dari model yang telah diimplementasikan pada Jetson Nano berupa kotak deteksi dan nilai estimasi jarak.

Setelah dilakukan *deployment*, model deteksi dan estimasi jarak *speed bump* akan dilakukan pengujian secara langsung di lapangan. Pengujian validitas dilakukan dengan mengukur sejauh mana model dapat mendeteksi objek dengan baik dan benar, serta kemampuannya mendeteksi objek dengan benar ketika *input* data yang masuk berbeda dengan data yang digunakan pada saat pelatihan model.

Selain itu, dilakukan pula pengukuran akurasi fitur estimasi jarak dalam menghitung jarak objek dengan selisih nilai yang kecil terhadap jarak asli di lapangan. Pengujian reliabilitas dilakukan dengan menilai konsistensi model deteksi objek ketika mendapatkan data baru di lokasi yang berbeda, serta mengukur konsistensi fitur estimasi jarak pada jarak yang berbeda dengan yang diterapkan pada proses pengambilan data.