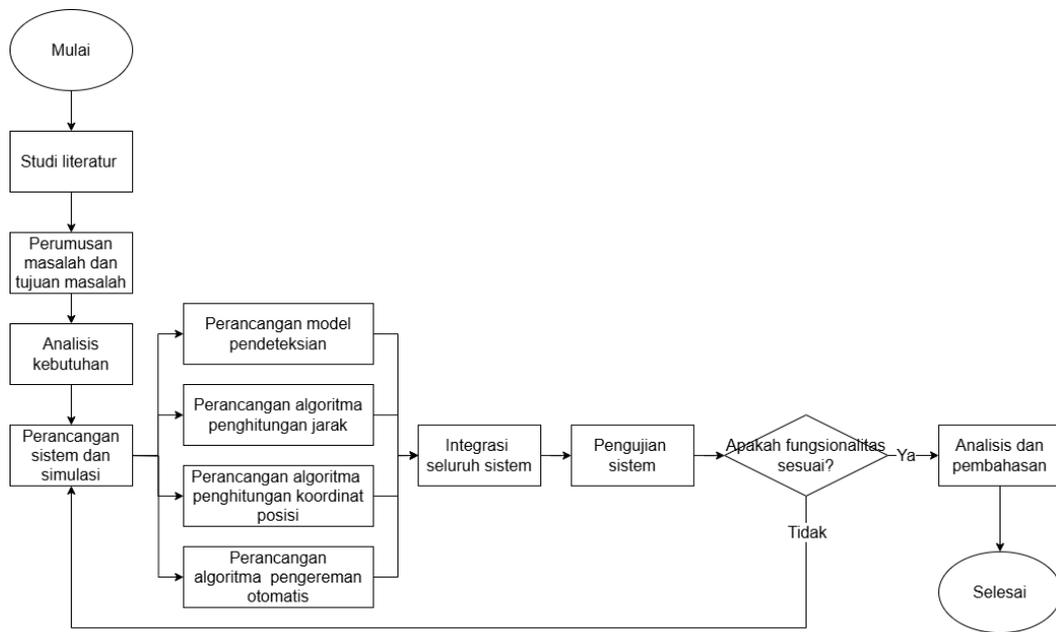


## BAB III METODE PENELITIAN

### 3.1. Desain Penelitian

Berdasarkan analisis kebutuhan, penelitian dengan judul "Implementasi Model YOLOv8m untuk Deteksi dan Estimasi Posisi Halangan Berbasis Kamera RGBD pada Kendaraan Otonom" terstruktur dalam empat komponen utama, yaitu model deteksi halangan, algoritma pengukuran jarak, algoritma penghitungan koordinat posisi, dan algoritma pengereman otomatis. Perencanaan penelitian ini secara rinci dapat dilihat pada sebagai berikut.



Gambar 3. 1 Desain penelitian

Pada Gambar 3. 1 diperlihatkan bahwa tahapan awal penelitian akan dimulai melalui studi literatur, di mana berbagai penelitian sebelumnya dieksplorasi secara mendalam untuk mengidentifikasi permasalahan yang masih relevan dalam konteks teknologi kendaraan otonom. Studi literatur ini berfokus pada pengimplementasian model untuk sistem pendeteksian halangan, pengukuran jarak dan posisi, serta implementasi sistem pengereman otomatis menggunakan berbagai teknologi pendukung seperti kamera RGBD, model YOLOv8, dan *framework* ROS2. Hasil eksplorasi ini, akan memberikan dasar pemahaman untuk memahami perkembangan terkini sekaligus menemukan celah penelitian yang dapat dijadikan sebagai kontribusi baru.

Langkah selanjutnya adalah melakukan analisis kebutuhan sistem, yang bertujuan untuk merancang solusi yang sesuai dengan tujuan penelitian. Analisis ini, dimulai dengan identifikasi kebutuhan perangkat keras dan perangkat lunak. Pada tahap ini, diputuskan bahwa perangkat utama yang akan digunakan adalah kamera RGBD yang akan digunakan untuk mendeteksi sekaligus mengukur jarak dan posisi dari kamera ke halangan. Selain itu, mini komputer NUC11 digunakan untuk menjalankan *framework* ROS2 di Ubuntu 22.04, yang berfungsi sebagai platform utama untuk integrasi sistem. Kebutuhan perangkat lunak mencakup implementasi model YOLOv8m untuk algoritma pendeteksian halangan, algoritma khusus untuk posisi dan jarak, serta komunikasi antar-komponen yang terintegrasi dalam *framework* ROS2 versi Humble Hawksbill.

Berdasarkan hasil analisis kebutuhan, penelitian ini akan merencanakan perancangan sistem deteksi halangan yang mengintegrasikan model YOLOv8m dengan masukan gambar dari kamera RGBD. Perancangan ini akan mencakup pengolahan data dari kamera RGBD, di mana informasi dari data RGB (*color*) akan digunakan untuk mendeteksi halangan, sedangkan data kedalaman (*depth*) akan dimanfaatkan untuk memperoleh informasi jarak dan mengestimasi koordinat objek berbagai kondisi lingkungan. Model YOLOv8m yang digunakan direncanakan akan dilatih menggunakan data set *Common Objects in Context* (COCO) karena data set ini mencakup berbagai jenis objek yang relevan untuk kebutuhan deteksi. Dengan perencanaan ini, sistem deteksi halangan diharapkan akan mampu memberikan kontribusi dalam pengembangan teknologi yang memerlukan pendeteksian objek khususnya halangan pada kendaraan otonom dan penentuan jarak dan posisi yang mendekati kondisi dunia nyata.

Tahap berikutnya adalah perancangan algoritma estimasi posisi dan jarak. Algoritma ini dirancang untuk mengolah data kedalaman dari kamera RGBD, yang kemudian digunakan untuk menghitung jarak antara kamera dengan halangan yang terdeteksi. Data ini akan diolah secara *real-time* sehingga dapat memberikan respons yang sesuai untuk mendukung sistem pengereman otomatis. Setelah seluruh sistem dirancang, dilakukan integrasi keseluruhan sistem berbasis ROS2. Integrasi ini mencakup penghubungan antara kamera RGBD, perangkat lunak pendeteksi halangan, algoritma estimasi jarak, dan pengereman otomatis.

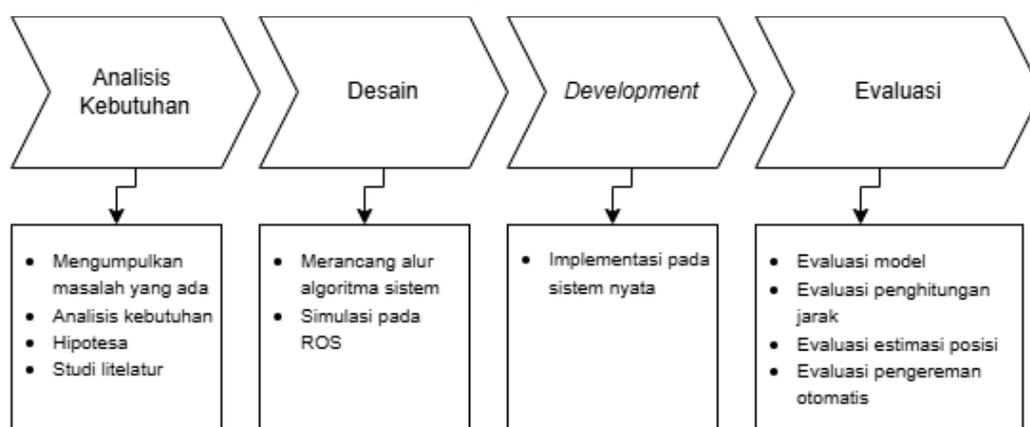
*Framework* ROS2 dipilih karena kemampuannya dalam menangani komunikasi antar-komponen secara efisien dalam waktu nyata. Integrasi ini dirancang untuk memastikan bahwa data yang dihasilkan oleh sistem deteksi halangan dapat langsung digunakan oleh algoritma estimasi jarak dan diteruskan ke sistem pengereman.

Sistem yang telah diintegrasikan kemudian diuji dalam skenario simulasi dan dunia nyata. Pengujian mendalam ini mencakup berbagai kondisi, seperti lingkungan luar ruangan dan dalam ruangan dengan tingkat pencahayaan yang berbeda dan diuji dengan sebuah halangan untuk mengetahui estimasi jarak dan posisinya. Tujuan utama pengujian adalah memastikan bahwa sistem mampu mendeteksi halangan dengan akurat pada berbagai kondisi, mendapatkan nilai jarak dan koordinat posisi dengan mendekati, dan mengaktifkan pengereman otomatis dalam waktu respons yang cepat. Rumus *Mean Absolute Error* (MAE) akan digunakan sebagai metrik utama untuk menilai akurasi estimasi jarak, koordinat posisi, dan pengereman otomatis. Sedangkan pada sistem pengereman otomatis akan dilakukan analisis dari selisih antara jarak yang ditentukan untuk berhenti pada algoritma pengereman otomatis dengan jarak yang dihasilkan oleh kamera pada saat kendaraan berhenti. Hasil pengujian ini, kemudian akan dianalisis untuk mengevaluasi keandalan dan kinerja sistem secara menyeluruh, baik dalam aspek deteksi objek, nilai jarak dan posisi, sampai respons dari pengereman otomatis. Dengan mengikuti desain penelitian yang sistematis dan terstruktur, penelitian ini diharapkan dapat menghasilkan sistem deteksi halangan dan estimasi jarak dan posisi berbasis kamera RGBD, yang dapat digunakan sebagai dasar untuk mengembangkan teknologi kendaraan otonom yang lebih aman dan efisien. Penelitian ini juga memberikan kontribusi signifikan dalam pengembangan algoritma deteksi halangan dan sistem pengereman otomatis berbasis teknologi visi komputer.

### **3.2. Metode Penelitian**

Penelitian ini mengambil pendekatan metode penelitian *Design and Development Research*. Menurut Richey dan Klein (2009) *Design and Development Research* didefinisikan sebagai studi sistematis tentang proses desain, pengembangan, dan evaluasi dengan tujuan untuk membangun dasar empiris bagi

penciptaan produk dan alat instruksional serta model baru atau yang ditingkatkan. Metode ini mencakup analisis kebutuhan, perencanaan, pengembangan, dan evaluasi pada produk atau alat. Tujuan utama dari metode penelitian ini adalah untuk memperluas basis pengetahuan di bidang desain dan pengembangan serta menciptakan fondasi untuk teori desain dan pengembangan yang lebih baik. Maka dari itu, metode *Design and Development Research* cocok digunakan pada penelitian ini yang bertujuan dengan bertujuan untuk menghasilkan solusi praktis dan teoritis melalui desain, pengembangan, dan evaluasi sistem. Dengan memilih metode *Design and Development Research*, penelitian ini terbagi ke dalam empat tahap utama yang saling terintegrasi. Metode *Design and Development Research* sangat cocok untuk penelitian ini karena bertujuan untuk menghasilkan solusi praktis dan teoritis melalui pendekatan desain, pengembangan, dan evaluasi sistem. Berikut adalah tahapan sistematis yang diadopsi dari metode *Design and Development Research* dalam konteks penelitian ini.



Gambar 3. 2 Tahapan metode *Design and Development Research*

Pada Gambar 3. 2 metode *Design and Development Research* ditunjukkan terdapat beberapa tahapan yang mendukung penelitian ini. Berikut adalah penjelasan tahapan pada metode *Design and Development Research*.

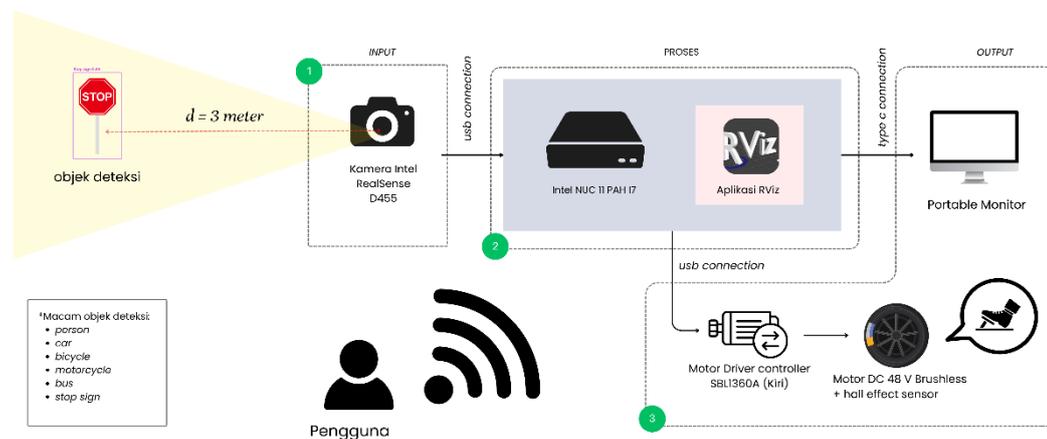
### 1. Analisis Kebutuhan

Pada fase ini, dilakukan studi literatur untuk menggali penelitian terdahulu yang relevan terkait dengan penggunaan YOLOv8m dalam deteksi objek, estimasi jarak dan posisi menggunakan kamera RGBD, serta penerapan pengereman otomatis pada kendaraan otonom. Analisis kebutuhan sistem juga dilakukan, mencakup identifikasi dan eksplorasi perangkat keras serta perangkat lunak yang

akan digunakan seperti kamera RGBD, ROS, dan model YOLOv8m. Selain itu, masalah yang dihadapi dalam pengembangan sistem pengereman otomatis pada kendaraan otonom akan diidentifikasi dan dijadikan dasar bagi perancangan sistem yang lebih baik.

## 2. Desain

Setelah analisis kebutuhan selesai, tahap desain berfokus pada perancangan sistem deteksi objek berbasis YOLOv8m dan integrasi dengan kamera RGBD serta algoritma estimasi posisi dan jarak. Di sini, desain perangkat lunak akan difokuskan pada integrasi dengan ROS untuk menghubungkan sistem deteksi objek, pengolahan data, dan sistem pengereman. Maka dari itu, terdapat diagram arsitektur sistem untuk merepresentasikan komponen dari sistem yang dirancang.



Gambar 3. 3 Diagram arsitektur sistem

Pada Gambar 3. 3 ditampilkan diagram arsitektur sistem yang menggunakan kamera Intel RealSense D455 sebagai sumber *input* utama. Kamera ini menyediakan data berupa gambar berwarna (RGB) dan gambar kedalaman (*depth*), yang akan digunakan untuk mendeteksi dan menganalisis objek di sekitar. Data dari kamera kemudian diproses oleh Intel NUC 11 PAH i7, sebuah unit pemrosesan yang bertugas menjalankan algoritma deteksi, pemrosesan data, serta pengambilan keputusan berdasarkan informasi yang diterima. Hasil pemrosesan data divisualisasikan menggunakan aplikasi RViz, yang berguna untuk memantau informasi secara langsung dalam bentuk representasi grafis, termasuk posisi objek dalam ruang tiga dimensi. Selain itu, sistem juga memiliki *output* berupa layar

monitor, yang digunakan untuk menampilkan sistem operasi dan proses yang sedang dijalankan oleh Intel NUC 11 PAH i7.

Untuk mengendalikan sistem penggerak, data hasil pemrosesan dari Intel NUC 11 PAH i7 digunakan untuk memberikan perintah referensi kecepatan kepada motor *driver controller* SBL1360A. Dalam penerapan fitur penggerak pada sistem pengereman otomatis, ketika sensor menerima instruksi untuk berhenti, motor *driver controller* SBL1360A akan mengalirkan arus sesuai referensi yang diperintahkan. Arus tersebut kemudian disalurkan ke motor *brushless* DC 48V, yang akan mengaktifkan mekanisme pengereman. Data yang diterima oleh motor *brushless* DC 48V selanjutnya dikomunikasikan ke modul sensor *hall effect*. Modul ini bertugas memverifikasi apakah kecepatan motor yang sebenarnya telah sesuai dengan perintah yang diberikan, memastikan pengereman berjalan sesuai dengan kebutuhan sistem.

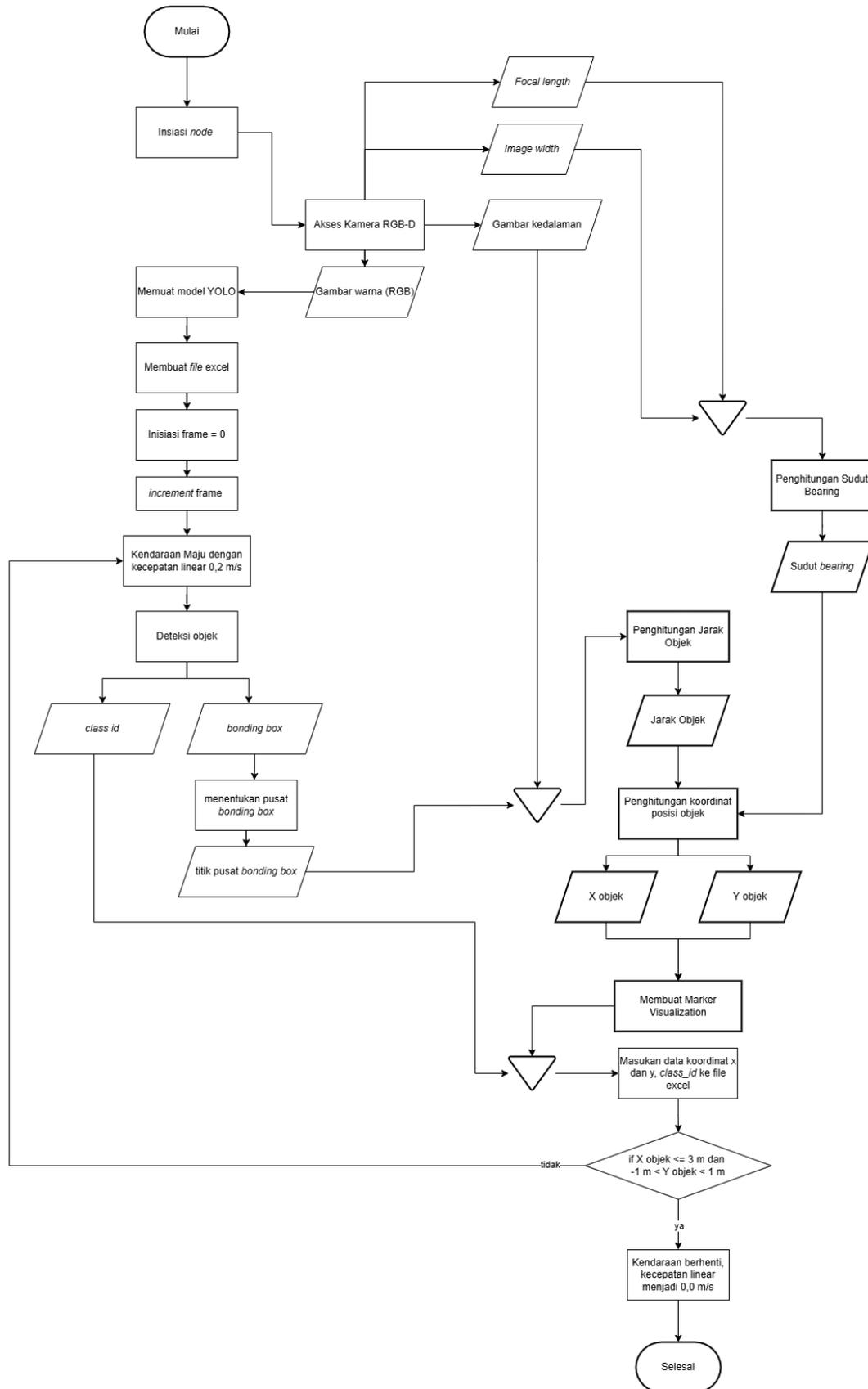
Selain diagram arsitektur sistem, pada tahap ini juga di rancang alur algoritma sistem yang tertuang pada diagram alir sistem pada Gambar 3. 4 yang terdiri dari algoritma jarak, estimasi posisi, sampai pengereman otomatis. Pada Gambar 3. 4 Alur ini mencakup serangkaian proses mulai dari algoritma penghitungan jarak, estimasi posisi, hingga implementasi pengereman otomatis. Gambar 3.4 menunjukkan struktur alur sistem yang diawali dengan inisiasi *node* dalam ROS untuk mengakses kamera RGBD. Kamera ini kemudian melakukan *streaming* data berupa gambar warna (RGB) dan gambar kedalaman (*depth*).

Data gambar warna (RGB) akan diproses lebih lanjut dengan memuat model YOLOv8m. Model ini menggunakan gambar warna sebagai input untuk mendeteksi objek. Selanjutnya, sistem membuat berkas excel untuk menyimpan data yang dikumpulkan selama proses pengujian. Pada tahap ini, *frame* yang akan diambil datanya diinisialisasi dan ditambahkan secara bertahap sesuai kebutuhan. Setelah itu, sistem akan mempublikasikan nilai kecepatan linear sebesar 0,2 m/s untuk menggerakkan kendaraan maju secara lurus.

Model YOLOv8m kemudian melakukan deteksi objek, menghasilkan informasi berupa kelas *ID* dan *bounding box* untuk setiap objek yang terdeteksi. Data kelas *ID* dicatat ke dalam berkas excel, sementara *bounding box* diproses lebih lanjut untuk menentukan titik pusatnya. Di sisi lain, kamera kedalaman (*depth*)

digunakan untuk mengukur jarak objek dari kamera. Selain jarak, data tambahan seperti panjang fokus (*focal length*) dan lebar gambar juga diambil untuk menghitung *sudut bearing*, yaitu sudut horizontal dari posisi objek terhadap kamera.

Koordinat objek dalam ruang dua dimensi, yaitu  $x$  dan  $y$ , dihitung berdasarkan jarak dan sudut bearing yang diperoleh. Hasil koordinat ini kemudian diteruskan ke marker visualization pada aplikasi RViz. Ketika posisi objek yang terdeteksi berada pada jarak kurang dari 3 meter di sumbu  $x$  dan dalam rentang 1 meter di sumbu  $y$  (baik ke kiri maupun ke kanan), sistem akan mempublikasikan nilai kecepatan linear sebesar 0 m/s. Hal ini menandakan bahwa kendaraan akan berhenti untuk menghindari potensi tabrakan dengan objek tersebut. Alur ini memastikan sistem bekerja secara terstruktur dan responsif dalam mendeteksi dan menghindari tabrakan pada objek secara otomatis.



Gambar 3. 4 Diagram alir sistem

### 3. *Development*

Pada tahap *Development*, desain yang telah dibuat akan diimplementasikan dalam bentuk sistem yang nyata. Model YOLOv8m akan diterapkan untuk deteksi objek dalam lingkungan nyata, sementara kamera RGBD digunakan untuk mengestimasi jarak objek yang terdeteksi. Integrasi sistem menggunakan ROS2 untuk komunikasi antara komponen dan pengereman akan dikembangkan. Seluruh perangkat keras dan perangkat lunak diimplementasikan dan diintegrasikan untuk membangun sistem deteksi objek, estimasi jarak, dan pengereman otomatis pada kendaraan otonom. Kendaraan yang dimaksud adalah SEATER.

### 4. Evaluasi

Setelah pengembangan selesai, sistem yang telah dibangun akan diuji secara menyeluruh dalam simulasi dan lingkungan dunia nyata. Pengujian ini mencakup berbagai kondisi seperti pencahayaan yang berbeda dan berbagai jenis halangan. Evaluasi akan dilakukan untuk mengukur akurasi deteksi objek, estimasi jarak, serta efektivitas sistem pengereman otomatis dalam merespons halangan. Kinerja sistem akan dievaluasi menggunakan metrik seperti MAE (*Mean Absolute Error*) pada penghitungan jarak, estimasi posisi, serta waktu respons pengereman. Hasil pengujian akan digunakan untuk menilai keandalan dan efektivitas keseluruhan sistem, serta mengidentifikasi area yang perlu diperbaiki. Dengan mengikuti fase-fase ini, penelitian dapat secara sistematis mengembangkan dan mengevaluasi solusi yang komprehensif untuk masalah deteksi dan estimasi jarak pada kendaraan otonom.

#### 3.2.1. Instrumen Penelitian

Pada sub bab instrumen penelitian akan menjelaskan data primer, data sekunder, dan perangkat penunjang yang dipakai. Berikut adalah penjelasan lanjut pada data primer, data sekunder, dan perangkat penunjang pada penelitian ini.

##### 3.2.1.1. Data Primer

Data primer dalam penelitian ini diperoleh dengan menggunakan data set dari COCO (*Common Objects in Context*) sebagai sumber utama untuk melatih dan menguji model YOLOv8m. COCO adalah COCO adalah data set yang

dikembangkan dengan mengumpulkan gambar-gambar dari berbagai situasi sehari-hari yang kompleks, di mana objek-objek umum ditampilkan dalam konteks alami untuk mendukung tugas-tugas seperti deteksi objek, segmentasi, dan pengenalan. Data set ini, secara keseluruhan mencakup lebih dari 328.000 gambar dengan total label 2,5 juta, dari 91 jenis objek (Lin, dkk., 2015). Dengan memilih data set ini, memungkinkan model untuk mendeteksi dan mengenali objek dengan tingkat presisi tinggi dalam berbagai kondisi lingkungan.

Namun, untuk penelitian ini, akan diambil beberapa kategori objek yang dianggap relevan dengan tempat dan lingkungan uji. Objek yang digunakan dalam penelitian ini meliputi manusia (*person*), mobil (*car*), sepeda (*bicycle*), sepeda motor (*motorcycle*), bus, dan tanda berhenti (*stop sign*). Pemilihan kategori objek ini didasarkan pada relevansinya dengan skenario pengujian yang dirancang, yaitu lingkungan yang berkaitan dengan tempat dilakukannya eksperimen yakni Kawasan Sains dan Teknologi Badan Riset dan Inovasi Nasional Samaun Samadikun. Dengan fokus pada objek-objek ini, penelitian bertujuan untuk mengevaluasi performa model YOLOv8m dalam mendeteksi dan mengenali objek-objek tersebut secara akurat dalam kondisi dan situasi lingkungan yang bervariasi.

### 3.2.1.2. Data Sekunder

Studi literatur dilakukan sebagai langkah awal untuk mengumpulkan data sekunder yang relevan dengan topik penelitian. Proses ini melibatkan pencarian referensi dari berbagai sumber yang dapat dipercaya, seperti perpustakaan daring, jurnal ilmiah, artikel daring, dan berbagai sumber lainnya di internet. Selama pencarian ini, peneliti mencari berbagai literatur yang dapat memberikan wawasan atau informasi yang relevan dengan topik yang sedang diteliti.

### 3.2.1.3. Perangkat Penunjang Penelitian

Penelitian ini didukung oleh berbagai perangkat keras dan perangkat lunak yang memainkan peran penting dalam memastikan kelancaran eksperimen serta analisis data secara menyeluruh. Rincian mengenai perangkat pendukung yang digunakan dapat dilihat pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Perangkat Keras Pendukung

---

#### Perangkat Keras

No	Nama	Spesifikasi
1	Laptop pribadi	Prosesor intel <i>core</i> i7, RAM sebesar 16 GB, serta VGA RTX 4060 yang dilengkapi dengan VRAM 8 GB. Kecepatan dan kapasitas penyimpanan didukung oleh SSD 512 GB.
2	Intel RealSense D455	Resolusi RGB sampai dengan $1280 \times 800$ px, 30 fps, 1 MP, RGB sensor FOV ( $H \times V$ ) $90 \times 65^\circ$ . Jangkauan ideal 0,6 m sampai 6 m, <i>Depth</i> FOV ( $H \times V$ ) $87 \times 58^\circ$ .
3	<i>Mini Computer</i> NUC i7 GEN 11 NUC11PAHi7	Prosesor <i>Core</i> i7 1165G7 2.8Ghz dengan RAM 32 GB dan penyimpanan NVME 1TB

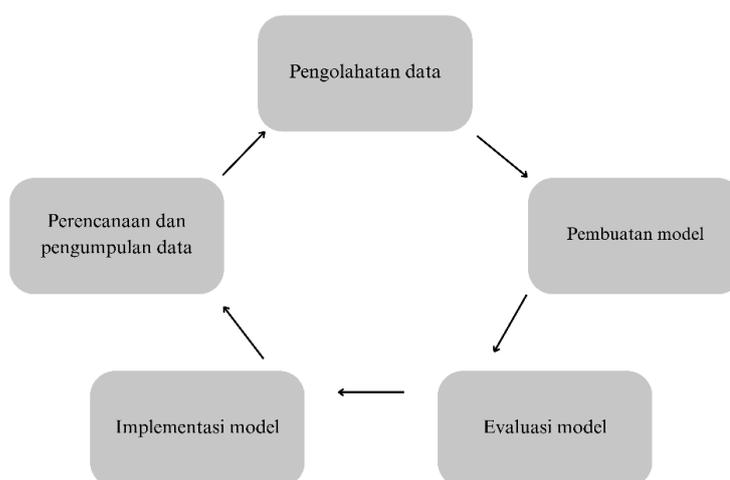
Tabel 3.2 Perangkat Lunak Pendukung

Perangkat Lunak		
No	Nama	fungsi
1	Ubuntu 22.04.5 LTS	Sebagai sistem operasi yang mendukung ROS humble hawkbill
2	Gazebo versi 11	Untuk menyimulasikan robot dan kamera serta membuat lingkungan dalam ruang tiga dimensi.
3	RVIZ2	Untuk memvisualisasikan dalam <i>framework</i> ROS ( <i>Robot Operating System</i> ) yang digunakan untuk menampilkan data dari kamera, visualisasi <i>marker</i> , serta status robot dalam sebuah lingkungan virtual.
4	Visual Studio Code	Untuk editor teks pada <i>script</i> untuk mengembangkan perangkat lunak sistem deteksi sampai dengan pengereman otomatis.

- 5 Roboflow Untuk mengunduh data set dan memvisualisasikan data set yang sudah terlabelisasi.

### 3.2.2. Metode Perancangan Model

*Artificial Intelligence Life Cycle* adalah kerangka kerja komprehensif untuk desain, pengembangan, dan penerapan sistem atau solusi AI. Siklus ini mencakup semua tahapan utama, mulai dari konsepsi, penilaian risiko awal, desain, pengembangan, hingga implementasi. Metode ini dirancang untuk mengatasi kekurangan dalam pendekatan tradisional, seperti kurangnya fokus pada risiko adopsi AI, etika, dan tata kelola, serta kebutuhan akan pendekatan yang spesifik untuk AI daripada hanya memodifikasi kerangka kerja umum (De Silva, & Alahakoon, 2022).



Gambar 3. 5 Metode perancangan model AILC

Pada Gambar 3. 5 Metode perancangan model deteksi objek ini didasarkan pada siklus hidup kecerdasan buatan (*Artificial Intelligence Life Cycle*), yang terdiri dari beberapa tahap utama, yaitu perencanaan, pengumpulan data, pengolahan data, pengembangan model, evaluasi model, dan implementasi. Model yang digunakan adalah YOLOv8m, yang merupakan salah satu varian dari YOLO (*You Only Look Once*) dengan keseimbangan optimal antara kecepatan dan akurasi. Berikut adalah rincian setiap tahap pada metode AILC.

#### 3.2.2.1. Perencanaan dan Pengumpulan Data

Tahap perencanaan dimulai dengan menetapkan tujuan utama proyek, yaitu mendeteksi objek-objek tertentu dalam sebuah *frame* menggunakan algoritma YOLOv8m. Varian medium dari model YOLOv8 dipilih berdasarkan hasil penelitian yang menunjukkan bahwa model ini memiliki akurasi tertinggi dibandingkan varian lainnya. Sebagai contoh, penelitian yang dilakukan oleh Jamal dkk. (2024) menyimpulkan bahwa di antara beberapa varian YOLOv8, YOLOv8m adalah yang paling unggul dalam mendeteksi cacat produk berupa gelembung di perusahaan manufaktur. Selain itu, Panja dkk. (2024) juga mengaplikasikan YOLOv8m dalam klasifikasi kendaraan, dan setelah melalui proses augmentasi data, model YOLOv8m terbukti memberikan hasil terbaik. Data yang akan digunakan bersumber dari data set COCO (*Common Objects in Context*), dengan varian data set YOLOv8-640 dengan resolusi 640 x 640 *pixel* sebanyak 123.272 total gambar sebanyak 80 kelas dengan total 854.991 anotasi yang terunggah pada laman web Roboflow. Dengan struktur data yang terorganisir ini, proses pelatihan, validasi, dan pengujian model dapat berjalan lebih terstruktur dan efisien.

### 3.2.2.2. Pengolahan Data

Setelah data set terkumpul, dilakukan proses pengolahan data agar siap digunakan dengan memisahkan kelas-kelas objek yang relevan. Data set COCO difilter menggunakan pemisahan berdasarkan berkas JSON bernama *\_annotation.coco*, yang berisi informasi tentang data set. berkas ini mencakup *metadata* untuk setiap gambar, seperti ID unik, nama berkas, dan dimensi gambar. Selain itu, berkas ini juga memuat anotasi objek dalam gambar, termasuk *bonding box*, ID kategori, serta ID gambar tempat objek ditemukan. Dalam proyek ini, deteksi difokuskan pada enam objek, yaitu manusia, mobil, sepeda, sepeda motor, bus, dan tanda setop. Pemilihan kelas-kelas ini dilakukan untuk mendukung penerapan sistem pada kendaraan otonom. Setelah pengolahan selesai, data set dibagi menjadi tiga bagian, yaitu data latih sebanyak 80 persen, data validasi sebanyak 10 persen, dan data uji sebanyak 10 persen. Proporsi yang besar untuk sub set ini memastikan model dapat mempelajari pola dan fitur yang ada pada data secara optimal. 10 persen sub set validasi digunakan selama proses pelatihan untuk mengevaluasi kinerja model. Validasi ini membantu dalam proses pemantauan *overfitting* dan penyesuaian parameter model. 10 persen sub set pengujian

digunakan setelah pelatihan selesai untuk mengevaluasi generalisasi model pada data baru. Pembagian data set dengan proporsi ini bertujuan untuk menciptakan keseimbangan antara pelatihan, validasi, dan pengujian.

### 3.2.2.3. Pembuatan Model

Pada tahap ini, proses pelatihan model YOLOv8m dilakukan dengan tujuan untuk memaksimalkan kinerja deteksi objek berdasarkan data set yang telah disiapkan dan difilter sebelumnya. YOLOv8m dipilih karena kemampuannya dalam mendeteksi objek dengan akurasi tinggi sekaligus efisiensi komputasi yang baik. Untuk memaksimalkan kinerja model beberapa *hyperparameter* pada Tabel 3. 1 diatur dan akan dipakai pada pelatihan model sebagai berikut.

Tabel 3. 1 *Hyperparameter* yang akan digunakan

Jenis <i>hyperparameter</i>	Kegunaan	Nilai yang digunakan
<i>Learning Rate</i>	Mengontrol kecepatan pembelajaran model	0.0001
<i>Weight Decay</i>	Regularisasi untuk mencegah <i>overfitting</i>	0.0005
<i>Image Size</i>	Ukuran gambar <i>input</i>	640
<i>Epochs</i>	Jumlah iterasi pelatihan	100
<i>Batch Size</i>	Jumlah data yang diproses dalam satu iterasi	16
<i>Worker</i>	Jumlah pekerja untuk <i>loading</i> data	4
<i>AMP (Automatic Mixed Precision)</i>	Mengurangi konsumsi memori dan mempercepat pelatihan menggunakan <i>mixed precision</i>	<i>True</i>
<i>Freeze</i>	Membekukan lapisan awal model selama pelatihan	5

<i>mosaic</i>	Augmentasi gambar dengan metode <i>mosaic</i>	<i>True</i>
<i>Label Smoothing</i>	Regularisasi untuk mengurangi kepercayaan berlebihan pada prediksi	0.1
<i>Patience</i>	Batas untuk <i>early stopping</i> jika tidak ada peningkatan	10

Selain penggunaan *hyperparameter*, mekanisme *early stopping* dengan *patience* 10 akan diterapkan untuk mencegah proses pelatihan berjalan terlalu lama ketika model tidak menunjukkan peningkatan performa yang signifikan pada metrik evaluasi. Dengan mekanisme ini, pelatihan akan berhenti secara otomatis jika tidak terjadi peningkatan selama 10 *epoch* berturut-turut, sehingga dapat menghemat waktu dan sumber daya komputasi, serta mencegah model menjadi *overfitting* akibat pelatihan yang terlalu lama.

Selanjutnya, *optimizer* SGD (*Stochastic Gradient Descent*) dipilih untuk digunakan dalam proses pelatihan model. *Optimizer* ini dikenal mampu memberikan stabilitas pada proses optimisasi dengan cara memanfaatkan gradien secara bertahap pada setiap iterasi, meskipun menggunakan sub set data (*mini-batch*) yang lebih kecil. Dalam kasus ini, penggunaan SGD akan membantu mempercepat proses konvergensi menuju nilai minimum *loss* yang optimal.

#### 3.2.2.4. Evaluasi Model

Evaluasi hasil pelatihan model deteksi objek dilakukan dengan menggunakan metrik-metrik tertentu untuk mengukur performa model. Metrik-metrik yang akan digunakan seperti *Mean Average Precision* (mAP), *precision*, *recall*, *F1-score*, *confusion matrix*. Berikut adalah persamaan untuk mendukung evaluasi model.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (6)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (7)$$

Dimana:

$TP$  = Jumlah prediksi positif yang benar

$FP$  = Jumlah prediksi positif yang salah (objek tidak ada, tetapi terdeteksi ada)

$FN$  = Jumlah kasus positif yang tidak terdeteksi (objek ada tetapi tidak terdeteksi)

$AP_i$  = *Average Precision* untuk kelas ke- $i$

*Precision* mengukur seberapa banyak prediksi yang benar dibandingkan dengan total prediksi yang dilakukan model, sedangkan *recall* mengukur seberapa banyak objek yang benar-benar ada dan berhasil ditemukan oleh model. *F1 Score* digunakan untuk mengukur keseimbangan antara *precision* dan *recall*, terutama jika keduanya tidak seimbang. Pada penelitian ini *Mean Average Precision* (mAP) digunakan untuk mengukur performa keseluruhan dari model deteksi atau klasifikasi dalam mengidentifikasi objek yang benar dengan akurasi tinggi pada berbagai kelas dan *threshold Intersection over Union* (IoU).

### 3.2.2.5. Implementasi Model

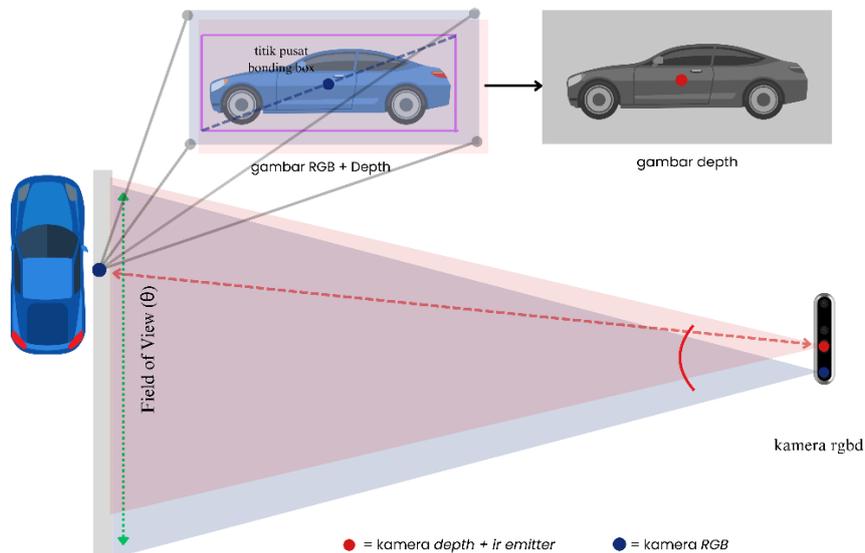
Model yang telah dievaluasi akan diimplementasikan ke dalam sistem untuk mendukung deteksi objek. Implementasi ini direncanakan dilakukan dalam dua skenario, yaitu simulasi dan pengujian pada sistem nyata. Pengujian simulasi akan digunakan untuk menilai performa sistem dalam lingkungan yang terkendali, sedangkan pengujian pada sistem nyata akan bertujuan mengukur kinerja sistem di dunia nyata. Setelah deteksi objek selesai dan bounding box terbentuk untuk setiap objek, koordinat pusat dari bounding box akan digunakan sebagai representasi data kedalaman. Informasi ini akan membantu menghitung jarak objek dan menentukan posisi dalam koordinat x dan y. Sistem ini, direncanakan untuk mendukung pengoperasian kendaraan otonom dengan menggunakan data jarak dan posisi objek untuk mengambil keputusan pada pengereman otomatis. Dengan pendekatan ini, sistem diharapkan mampu mengintegrasikan deteksi objek, pengukuran jarak, dan pengambilan keputusan secara efektif dalam pengoperasian kendaraan otonom.

### 3.3. Metode Pengujian Sistem

Pada tahap ini, akan dirancang metode pengujian untuk sistem yang dikembangkan dalam penelitian ini dengan langkah-langkah sebagai berikut.

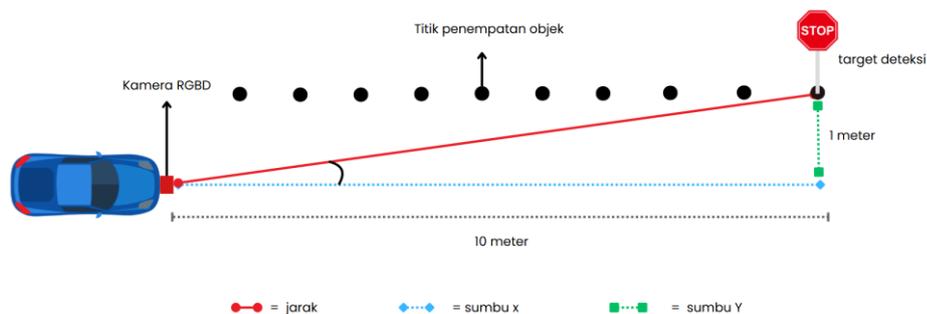
#### 3.3.1. Metode Pengujian Pengukuran Jarak

Pada pengujian jarak, penelitian akan dilakukan pada empat jenis lingkungan yang berbeda, yaitu di dalam ruangan dan di luar ruangan dengan tiga waktu berbeda, yaitu pagi hari, siang hari, dan sore hari. Untuk memastikan data yang diperoleh representatif terhadap berbagai intensitas cahaya (*lux*). Hal ini bertujuan untuk mengevaluasi pengaruh intensitas cahaya terhadap hasil pengukuran jarak dan estimasi posisi menggunakan kamera RGBD.



Gambar 3. 6 Visualisasi pengambilan nilai jarak dari kamera RGBD

Pada Gambar 3. 6 menjelaskan konsep pengambilan nilai data jarak dijelaskan sebagai berikut: data jarak diperoleh dari informasi kedalaman (*depth data*) yang dihasilkan oleh kamera RGBD. Prosesnya dimulai dengan pendeteksian objek menggunakan model YOLOv8m, yang menghasilkan *bounding box* untuk objek yang terdeteksi. Nilai kedalaman yang digunakan untuk menghitung jarak diambil dari titik spesifik, yaitu pusat *bounding box* yang dihasilkan. Dengan kata lain, nilai kedalaman di pusat *bounding box* tersebut menjadi representasi jarak objek dari kamera. Metode ini memastikan bahwa data jarak yang diperoleh relevan dengan lokasi objek yang terdeteksi oleh model.



Gambar 3. 7 Visualisasi pengambilan data jarak

Pada Gambar 3. 7 dilihat kamera RGBD akan diarahkan langsung ke target yang akan ditempatkan pada posisi tertentu sesuai dengan pengaturan awal. Pengujian akan dimulai dari jarak 2 meter dari kamera, kemudian akan dilanjutkan dengan peningkatan jarak secara bertahap sebesar 0,5 meter hingga mencapai jarak maksimum 10 meter. Pada setiap jarak yang telah ditentukan, kamera akan mengambil data pengukuran sebanyak 20 *frame* berturut-turut dan disimpan pada berkas excel. Setelah itu, data jarak akan dirata-ratakan (*mean*) untuk menghasilkan nilai yang lebih representatif. Untuk memastikan keakuratan dalam pengujian pengukuran jarak, penelitian ini akan berencana menggunakan objek statis sebagai target, guna meminimalkan potensi gangguan dari pergerakan seperti tanda setop.

Selain itu, pengujian ini akan dilakukan dalam berbagai kondisi pencahayaan untuk mengevaluasi performa kamera RGBD di lingkungan yang berbeda. Nilai rata-rata dari setiap interval jarak akan dianalisis untuk menilai tingkat akurasi dan keandalan kamera dalam mengukur jarak objek. Selanjutnya, data jarak akan dievaluasi terkait perbedaan antara jarak aktual dan jarak yang diukur oleh kamera. Melalui langkah-langkah tersebut, penelitian ini berencana untuk memahami pengaruh intensitas cahaya terhadap hasil pengukuran jarak. Dengan membandingkan nilai *Mean Absolute Error* (MAE) di berbagai kondisi pencahayaan. *Mean Absolute Error* (MAE) berperan penting dalam pengukuran jarak dalam model prediksi, untuk mengukur rata-rata kesalahan absolut antara nilai yang diprediksi dan nilai aktual, memberikan gambaran yang jelas tentang seberapa jauh prediksi menyimpang dari kenyataan tanpa memberikan bobot lebih pada kesalahan yang lebih besar. *Absolute* pada *Mean Absolute Error* (MAE) dapat diartikan memberikan bobot yang sama untuk semua kesalahan, sehingga tidak ada satu kesalahan pun yang mendominasi hasil. Pengujian ini diharapkan memberikan

wawasan mendalam tentang performa kamera RGBD, termasuk pengaruh terhadap perubahan intensitas cahaya serta kemampuannya dalam mempertahankan akurasi pada berbagai kondisi lingkungan. Berikut adalah rumus beserta keterangan pada metrik evaluasi data yang akan diuji pada sistem penghitungan jarak.

$$MAE_{distance} = \frac{1}{n} \sum_{i=1}^n |d_i - \hat{d}_i| \quad (8)$$

Dimana:

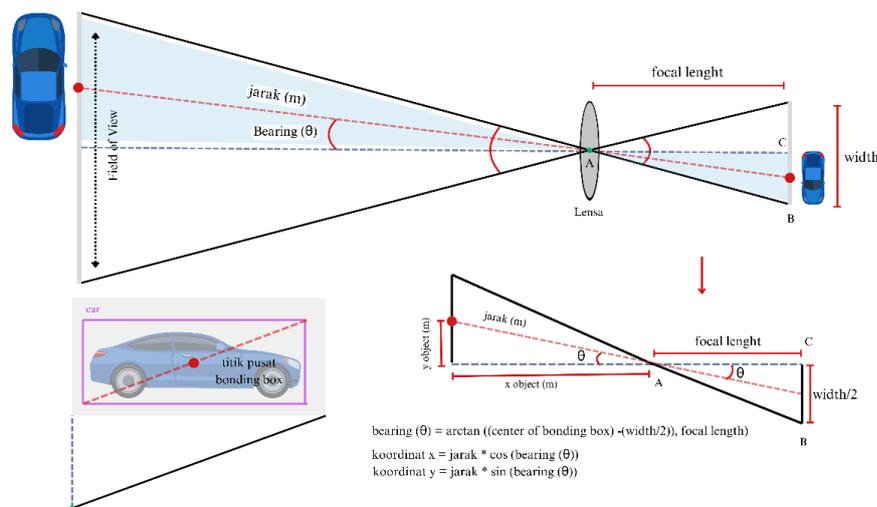
$n$  = jumlah total data pengukuran

$d_i$  = nilai jarak aktual

$\hat{d}_i$  = nilai jarak prediksi dari kamera

### 3.3.2. Metode Pengujian Estimasi Posisi

Dalam pengujian estimasi posisi ini, beberapa jenis objek direncanakan untuk dideteksi, dengan fokus pada kombinasi objek statis dan dinamis, seperti manusia, mobil, sepeda, sepeda motor, bus, dan rambu setop dengan koordinat yang sudah ditentukan. Setelah jarak objek diperoleh melalui data kedalaman yang dihasilkan oleh kamera RGBD, langkah berikutnya direncanakan untuk mendapatkan nilai sudut *bearing* guna agar dapat menentukan posisi objek terhadap kamera. Sudut *bearing* akan dihitung menggunakan informasi posisi objek dalam bidang pandang kamera. Jika objek berada di sisi kiri kamera, nilai *bearing* akan diberi tanda negatif, sedangkan jika objek berada di sisi kanan, nilai *bearing* akan diberi tanda positif. Proses ini bertujuan untuk memperjelas posisi objek terhadap kamera.



Gambar 3. 8 Visualisasi pengambilan nilai koordinat posisi

Pada **Error! Reference source not found.** ditunjukkan untuk menghitung nilai *bearing*, data lebar gambar (*width*) akan digunakan dan dibagi menjadi dua agar menjadi titik tengah gambar sebagai referensi nol derajat. Berdasarkan pergeseran posisi objek dari titik tengah tersebut, sudut *bearing* akan dihitung menggunakan fungsi trigonometri yaitu arctan atau invers dari tangen. Dengan diketahuinya nilai jarak, nilai kosinus digunakan untuk merepresentasikan koordinat x objek dengan mengalikan variabel jarak dengan kosinus dari sudut *bearing*, yang akan menunjukkan proyeksi jarak objek pada sumbu horizontal. Sementara itu, nilai sinus digunakan untuk merepresentasikan koordinat y objek, dengan mengalikan variabel jarak dengan kosinus dari sudut *bearing*, yang menunjukkan proyeksi jarak objek pada sumbu vertikal. Setelah nilai posisi x dan y objek diperoleh, nilai diambil evaluasi akan dilakukan menggunakan *Mean Absolute Error* (MAE) untuk membandingkan posisi yang diestimasi dengan posisi aktual. Evaluasi ini diharapkan memberikan wawasan mengenai tingkat akurasi estimasi posisi objek berdasarkan data kedalaman dan sudut *bearing*. Berikut adalah rumus beserta keterangan pada metrik evaluasi data yang akan diuji pada sistem penghitungan estimasi posisi.

$$MAE_{x,y} = \frac{1}{n} \sum_{i=1}^n |(x, y)_i - (x, y)^{\wedge}_i| \quad (9)$$

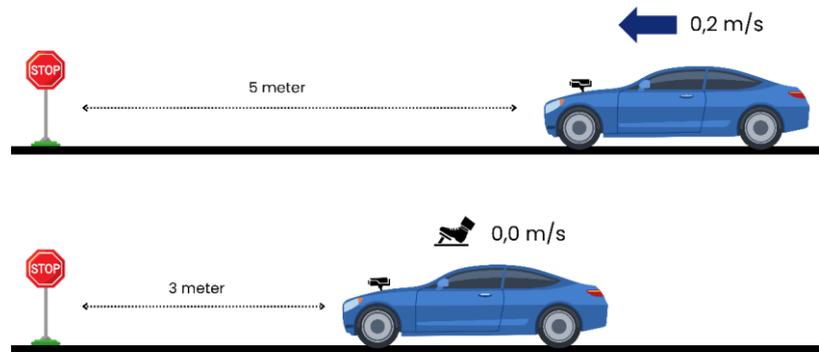
Dimana:

$n$  = jumlah total data pengukuran

$(x, y)_i$  = nilai koordinat x dan y aktual

$(x, y)^{\wedge}_i$  = nilai koordinat x dan y prediksi dari kamera

### 3.3.3. Metode Pengujian Pengereman Otomatis



Gambar 3. 9 Visualisasi pengujian pengereman otomatis

Pada ditunjukkan pengujian sistem pengereman otomatis, kendaraan akan diset dengan kecepatan kendaraan 0,2 m/s untuk maju linear secara konstan. Lalu, ketika data dari jarak sudah koordinat x menyentuh 3 meter dan koordinat y 1 meter dari kamera ke objek, maka kendaraan akan melakukan pengereman otomatis untuk mengevaluasi kemampuannya menghentikan kendaraan sebelum menabrak. Setelah itu, dilakukan pengukuran jarak secara manual sebagai acuan, serta jarak yang terdeteksi oleh kamera untuk dibandingkan. Data yang diambil meliputi kecepatan kendaraan, jarak awal kendaraan terhadap objek, jarak deteksi kamera, serta jarak akhir kendaraan dari objek setelah berhenti, waktu komputasi. Pengujian dilakukan sebanyak 10 kali untuk memastikan konsistensi hasil. Hasil dari setiap percobaan kemudian dianalisis untuk menghitung rata-rata *error*, yaitu selisih antara jarak deteksi kamera dan jarak manual dengan jarak yang diharapkan untuk pengereman. Rata-rata *error* yang diperoleh dari 10 percobaan ini digunakan untuk mengevaluasi keakuratan sistem deteksi jarak dan efektivitas pengereman otomatis. Dengan kecepatan rendah sebesar 0,2 m/s, sistem diharapkan dapat menghentikan kendaraan secara tepat waktu. Jika terdapat kesalahan yang signifikan atau kendaraan tetap menabrak, hal ini menunjukkan perlunya perbaikan pada algoritma deteksi atau sistem pengereman untuk meningkatkan performanya. Pada sistem ini juga diimplementasikan metrik *Mean Absolute Error* (MAE) untuk mengevaluasi data yang diambil. Berikut adalah rumus beserta keterangan pada metrik evaluasi data yang akan diuji pada sistem pengereman otomatis.

$$MAE_{x,y} = \frac{1}{n} \sum_{i=1}^n |(x,y)_i - (x,y)^{\wedge}_i| \quad (10)$$

Dimana:

$n$  = jumlah total data pengukuran

$(x,y)_i$  = nilai koordinat x dan y aktual

$(x,y)^{\wedge}_i$  = nilai koordinat x dan y prediksi dari kamera