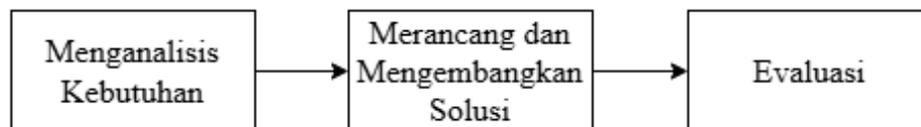


BAB III

METODE PENELITIAN

3.1 Metode Penelitian

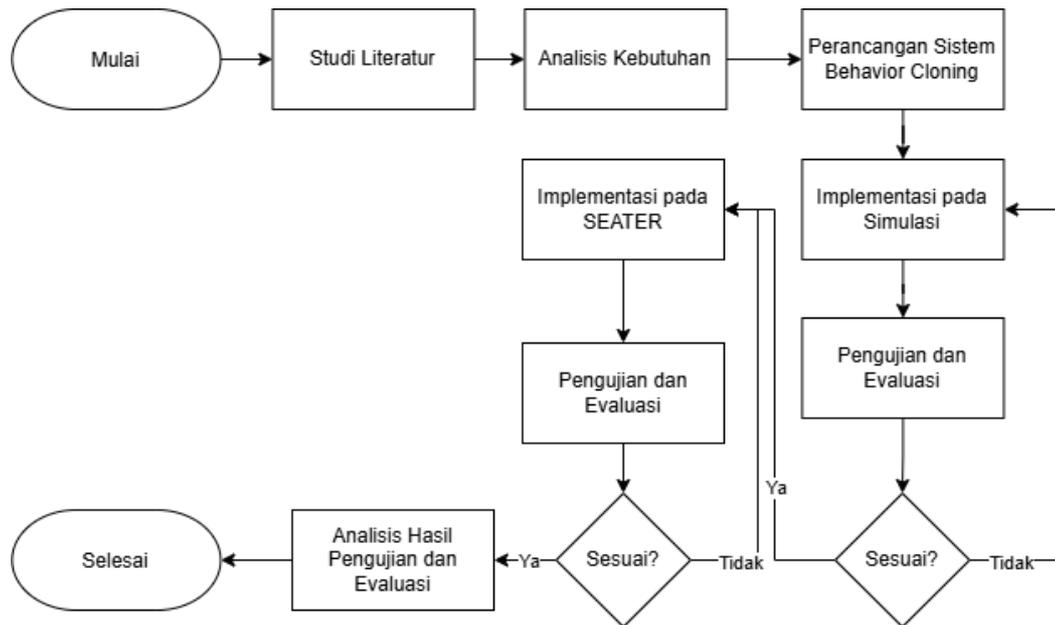
Metode penelitian yang digunakan dalam tugas akhir ini adalah *Design and Development Research* (DDR). *Design and Development Research* (DDR) didefinisikan sebagai studi sistematis mengenai proses desain, pengembangan, dan evaluasi yang bertujuan untuk membangun dasar empiris bagi penciptaan produk dan alat yang efektif (Richey & Klein, 2013). Pendekatan ini menekankan integrasi antara teori dan aplikasi praktis dalam konteks dunia nyata (Richey & Klein, 2013). Proses DDR mencakup tiga langkah utama, yaitu menganalisis kebutuhan, merancang dan mengembangkan solusi, serta melakukan evaluasi (Padzil dkk., 2021). Gambar 3.1 menyajikan ilustrasi yang menggambarkan tahapan-tahapan dalam metode *Design and Development Research* (DDR).



Gambar 3.1. Tahapan dari *design and development research*

Perancangan model *behavior cloning* untuk kendaraan otonom dengan satu penumpang memiliki satu fokus utama, yaitu pengembangan *lane keeping assist*. Proses perencanaan penelitian ini disusun dengan cara yang sistematis, seperti yang dapat dilihat pada Gambar 3.2.

Penelitian ini dimulai dengan kajian pustaka yang menyeluruh, di mana penelitian-penelitian terdahulu akan dianalisis untuk mengidentifikasi masalah-masalah yang masih relevan dengan penelitian terkini. Langkah selanjutnya adalah analisis kebutuhan sistem, yang bertujuan untuk menentukan model dan perangkat pendukung terbaik guna mendukung tujuan penelitian. Analisis ini akan dilakukan dengan rinci untuk merancang dua sistem utama yang diperlukan, baik pada tahap perancangan simulasi maupun implementasi dunia nyata pada kendaraan O-SEATER.



Gambar 3.2. Desain penelitian yang digunakan sebagai dasar penelitian

Setelah perancangan selesai, implementasi simulasi akan dilakukan menggunakan ROS gazebo. Setelah implementasi, pengujian model akan dilakukan untuk menilai sejauh mana solusi yang dihasilkan dapat memenuhi tujuan dan kebutuhan yang telah ditetapkan. Jika solusi belum sesuai, maka perbaikan akan dilakukan pada implementasi simulasi. Namun, jika sudah sesuai, implementasi akan dilanjutkan ke dunia nyata, yaitu pada O-SEATER. Terakhir, tahap analisis dan pembahasan akan dilakukan secara mendalam untuk mengevaluasi hasil penelitian, mengidentifikasi keberhasilan serta potensi perbaikan, dan memastikan bahwa solusi yang dihasilkan dapat memberikan kontribusi signifikan terhadap tujuan penelitian yang telah ditetapkan sebelumnya. Proses ini bertujuan untuk mengkonfirmasi apakah implementasi yang telah dilakukan pada simulasi maupun dunia nyata dapat mencapai hasil yang diinginkan atau perlu dilakukan revisi lebih lanjut.

3.2 Analisis Kebutuhan

Pada bagian ini akan dilakukan analisis kebutuhan sistem yang diperlukan untuk mendukung proses serta pencapaian tujuan. Selain itu, bagian ini mencakup penentuan kebutuhan penelitian, termasuk pengumpulan data primer dan sekunder yang dibutuhkan. Pembahasan juga akan meliputi berbagai perangkat yang mendukung pelaksanaan penelitian.

3.2.1 Data Primer

Data primer dalam penelitian ini diperoleh langsung dari sumber asli dengan menggunakan metode pengumpulan data berbasis sensor. Data yang dikumpulkan terdiri dari tiga kategori utama, yaitu data kamera, *linear velocity* (kecepatan kendaraan), dan *angular velocity* (kecepatan sudut), yang semuanya dikenal sebagai data peneliti. Data kamera berfungsi untuk memberikan representasi visual dari kondisi (*state*) O-SEATER di dunia nyata serta pada kendaraan simulasi di Gazebo ROS2 pada setiap titik waktu tertentu. Data ini menggambarkan lingkungan sekitar kendaraan yang digunakan untuk pengambilan keputusan. Visualisasi ini sangat penting dalam proses pelatihan karena membantu model *behavior cloning* untuk mengenali pola lingkungan yang relevan dengan aksi kendaraan yang dilakukan dalam situasi tertentu.

Kategori berikutnya, yaitu *linear velocity*, digunakan dengan menjaga kecepatan kendaraan tetap konstan. Tujuannya adalah untuk mencegah *linear velocity* mempengaruhi prediksi model terhadap perubahan *angular velocity* atau *angular velocity*. Dalam konteks *behavior cloning*, menjaga stabilitas *linear velocity* memastikan bahwa model hanya fokus pada hubungan antara kondisi lingkungan dan pengambilan keputusan terkait *angular velocity*, tanpa adanya gangguan dari variasi kecepatan. Dengan cara ini, model dapat belajar lebih baik dari data yang konsisten, sehingga menghasilkan prediksi *angular velocity* yang lebih akurat.

Angular velocity, sebagai komponen ketiga, memegang peranan penting sebagai data pelatihan dan validasi dalam model *behavior cloning*. Dalam pendekatan ini, model belajar untuk meniru aksi yang diambil oleh kendaraan berdasarkan kondisi yang ditangkap oleh kamera, untuk memprediksi *angular velocity* yang tepat dalam berbagai situasi. Data *angular velocity* ini memungkinkan model untuk menyesuaikan *angular velocity* O-SEATER dan kendaraan simulasi di Gazebo ROS2 sesuai dengan kondisi lingkungan, dengan tujuan mendekati perilaku yang diinginkan pada kendaraan otonom. Dengan ketiga jenis data ini, model memiliki dasar yang kuat untuk belajar meniru keputusan pengemudi secara end-to-end, memprediksi aksi yang tepat berdasarkan keadaan saat ini, dan

mencapai kinerja yang optimal pada sistem O-SEATER dan kendaraan simulasi di Gazebo ROS2.

Pengumpulan data dilakukan dalam periode waktu antara Oktober 2024 hingga November 2024, dengan kegiatan berkendara yang dilakukan pada pagi, siang, dan sore hari untuk pengambilan data di dunia nyata, serta simulasi di Gazebo untuk model kendaraan. Peneliti dapat mengumpulkan data yang sangat akurat mengenai data peneliti melalui pendekatan ini.

3.2.2 Data Sekunder

Studi literatur dilakukan untuk mengumpulkan data sekunder dengan mencari referensi yang relevan dari berbagai sumber, seperti perpustakaan dan internet. Data yang ditemukan kemudian dianalisis untuk menghasilkan informasi yang diperlukan, termasuk dalam penyusunan kajian pustaka, metode penelitian, dan pengembangan sistem yang telah dilakukan.

3.2.3 Perangkat Penunjang Penelitian

Penelitian ini didukung oleh berbagai perangkat keras dan perangkat lunak yang krusial untuk mendukung kelancaran eksperimen dan analisis data secara menyeluruh.

Selain O-SEATER, penelitian ini juga melibatkan sejumlah perangkat pendukung lainnya. Perangkat-perangkat pendukung ini mencakup berbagai komponen, baik perangkat keras maupun perangkat lunak, yang digunakan dalam proses simulasi, pengolahan data, dan pengujian model. Semua perangkat pendukung yang digunakan dalam penelitian ini tercatat secara rinci dan dapat dilihat pada Tabel 3.1.

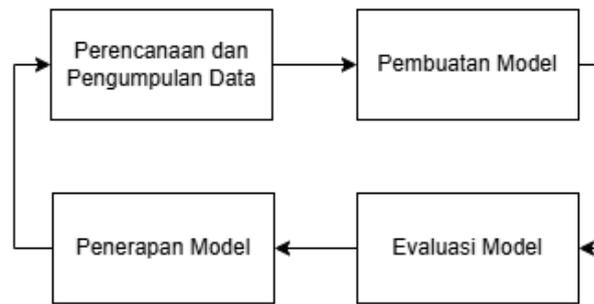
Tabel 3.1. Perangkat pendukung tambahan yang digunakan pada penelitian.

Perangkat Keras		
No	Nama	Spesifikasi
1	Personal Computer (PC)	Prosesor AMD Ryzen 5 5600X, RAM sebesar 16 GB, serta VGA RTX 2060 yang dilengkapi dengan VRAM 6 GB. Kecepatan dan kapasitas penyimpanan didukung oleh SSD 256 GB
2	Intel Realsense D455	1280 × 800 px, 30 fps, 1 MP

3	Intel NUC	Intel core i7, RAM sebesar 32 GB, dan GPU Intel TigerLake-LP GT2	
4	GPS - Xsens Vision Navigator	Akurasi Posisi dengan RTK: 1 cm + 1 ppm	
5	Roda & Hall Effect Sensor	16 Inch Hub Motor 48V 500W	
Perangkat Lunak			
No	Nama	Versi	Fungsi
1	WSL2	Ubuntu-20.04	<i>Virtual machine</i> yang menyediakan lingkungan Ubuntu dalam <i>windows</i> .
2	ROS2	Foxy Fitzroy	Menyediakan berbagai pustaka dan <i>tool</i> yang memudahkan pengembang untuk membuat aplikasi robotik yang kompleks, mengontrol sistem robot, dan melakukan integrasi perangkat keras dengan perangkat lunak.
3	Gazebo	11	Digunakan sebagai <i>platform</i> simulasi untuk merancang dan menguji kendaraan dalam berbagai kondisi, memungkinkan pengujian algoritma sebelum diterapkan di dunia nyata.

3.3 Metode Perancangan dan Pengembangan Model

Selain memanfaatkan metode *Design and Development Research* (DDR), penelitian ini juga mengintegrasikan *Artificial Intelligence Life Cycle* (AILC) sebagai kerangka kerja utama dalam pengembangan kecerdasan buatan. AILC merupakan pendekatan yang Berpotensi meningkatkan efisiensi produksi industri dengan mendukung pencapaian Tujuan Pembangunan Berkelanjutan serta menitikberatkan pada aspek desain dan pemeliharaan sistem produksi (Rahman dkk., 2022). Siklus Hidup AILC terdiri dari tiga tahap utama, yaitu Perancangan, Pengembangan, dan Penerapan (Silva & Alahakoon, 2022), yang kemudian digambarkan pada gambar Gambar 3.3.

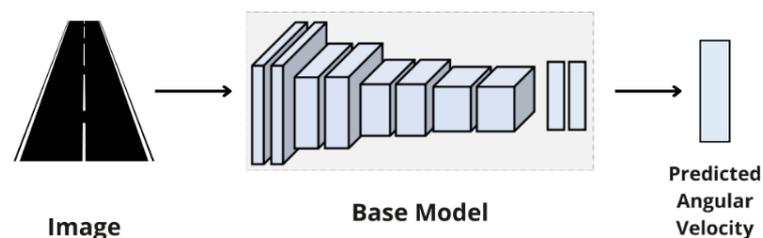


Gambar 3.3. Tahapan dalam AILC yang digunakan untuk membuat model

Dalam penelitian ini, *Artificial Intelligence Life Cycle* (AILC) digunakan untuk menjamin bahwa pengembangan model *Behavior cloning* berjalan secara sistematis dan berkelanjutan, mulai dari tahap awal hingga penerapannya. AILC mencakup berbagai proses penting yang memastikan setiap langkah dalam pengembangan dilakukan dengan tepat.

3.3.1 Perencanaan dan Pengumpulan Data

Pada tahap ini, masalah utama yang perlu dipecahkan oleh model AI dan tujuan yang ingin dicapai diidentifikasi dengan jelas. Dalam konteks pengembangan kendaraan otonom satu penumpang, *input* dari kamera memainkan peran krusial untuk menggambarkan kondisi terkini kendaraan, yang kemudian diolah untuk menentukan tindakan yang harus diambil, yang diilustrasikan pada Gambar 3.4.



Gambar 3.4. Ilustrasi *input* dan *output* model yang diharapkan

Angular velocity merupakan laju perubahan perpindahan sudut yang diukur dalam radian per detik (Bird & Chivers, 1983), dan dapat dinyatakan dengan Persamaan 3-1.

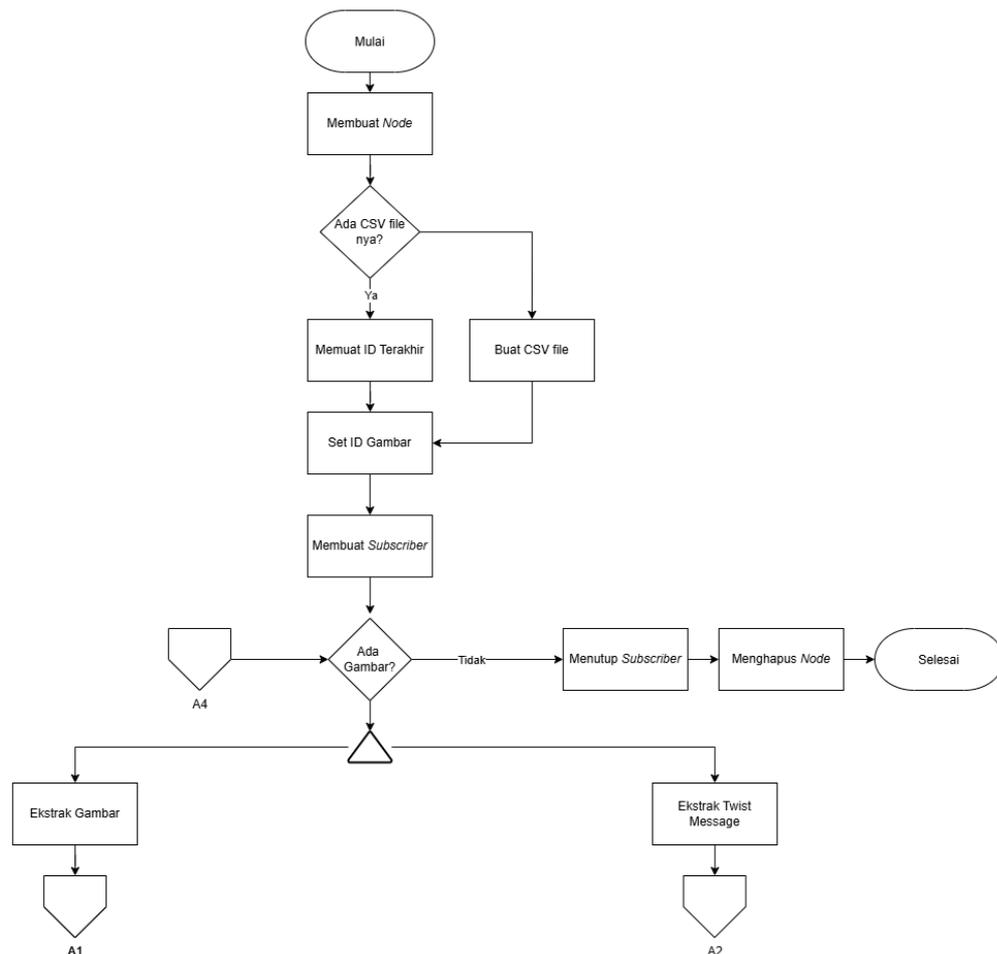
$$\omega = \frac{\theta}{t} \quad 3-1$$

Dalam persamaan tersebut, ω menggambarkan *angular velocity*, θ menunjukkan perubahan posisi sudut yang terjadi, dan t adalah interval waktu yang diperlukan untuk perubahan tersebut. Secara umum, konvensi yang digunakan

adalah bahwa *angular velocity* positif menunjukkan rotasi berlawanan arah jarum jam, sementara *angular velocity* negatif menunjukkan rotasi searah jarum jam.

Untuk memungkinkan model AI dalam kendaraan otonom membuat keputusan yang tepat berdasarkan kondisi lingkungan, sangat penting untuk memiliki data yang kaya dan relevan. Dalam hal ini, data tersebut hanya diperoleh melalui kamera yang terpasang pada kendaraan. Gambar yang diambil oleh kamera yang terletak di sisi kanan, kiri, dan tengah kendaraan memberikan informasi visual yang penting mengenai kondisi jalan dan pergerakan kendaraan. Selain itu, data yang mengandung informasi mengenai *linear velocity* dan *angular velocity* juga digunakan sebagai dasar untuk melatih model.

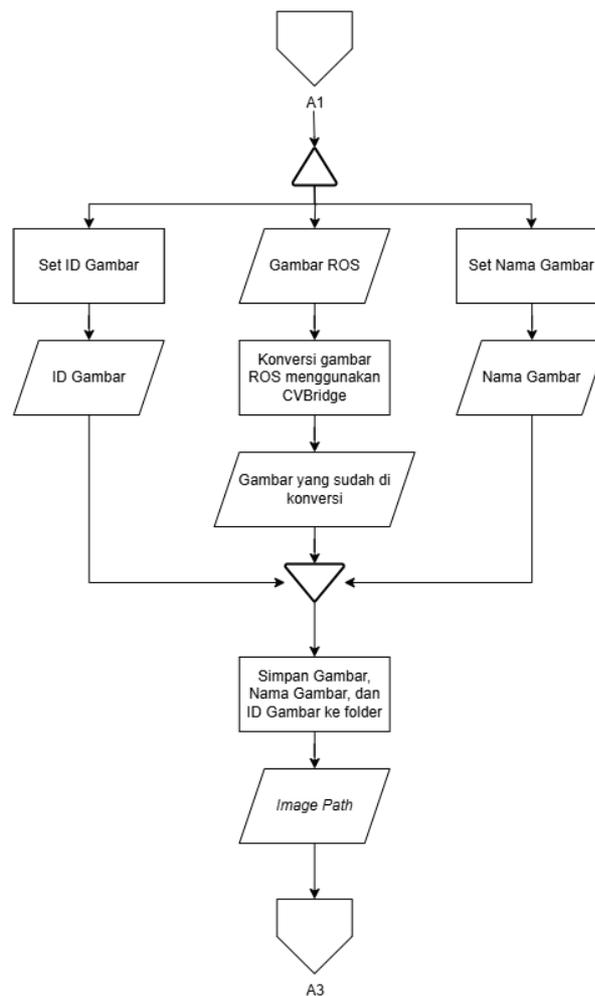
Proses pengumpulan data ini dijelaskan lebih rinci dalam *flowchart* yang terlihat pada Gambar 3.5.



Gambar 3.5. Flowchart alur pengumpulan data untuk menjadi data latih

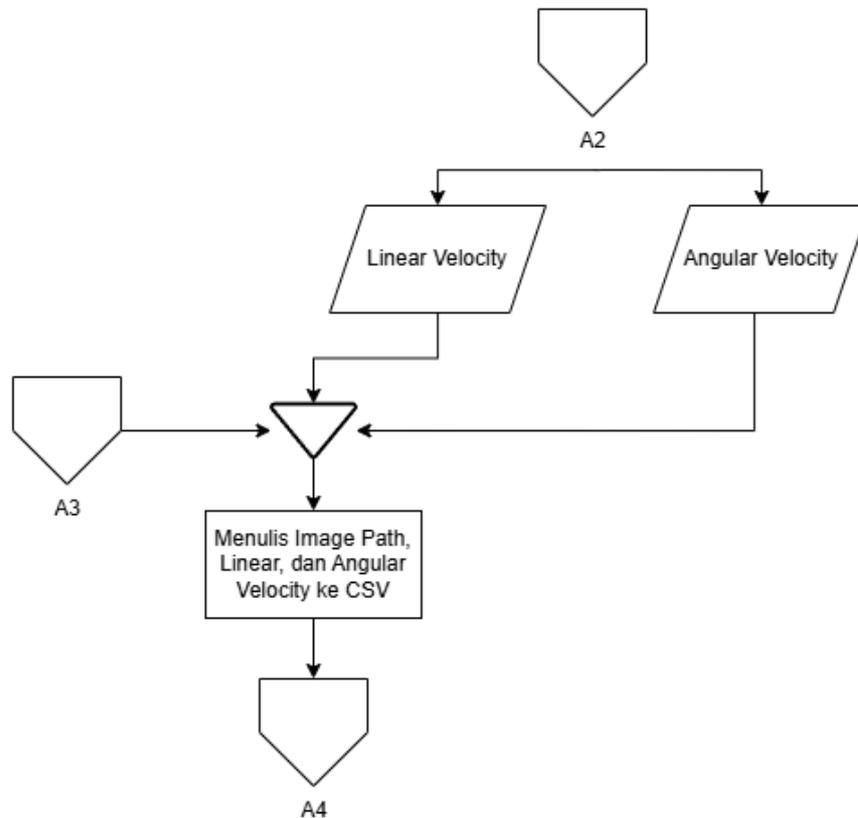
Flowchart pada Gambar 3.5 menggambarkan alur proses yang dimulai dengan pembuatan *node* di ROS (*Robot Operating System*). *Node* berperan sebagai modul program yang saling berkomunikasi dalam sistem. Selanjutnya, program memeriksa keberadaan file CSV. Jika file CSV telah tersedia, program akan memuat data terakhir dari file tersebut. Namun, jika file belum ada, program akan membuat file CSV baru. Setelah itu, ID terakhir akan disimpan ke dalam variabel program, diikuti dengan pembuatan *subscriber* untuk kamera dan pesan *twist* yang berisi informasi *linear velocity* dan *angular velocity*.

Proses kemudian memasuki tahap *looping* untuk menyimpan data, yang terbagi menjadi dua sub proses utama: ekstraksi gambar dan ekstraksi pesan *twist*. *Looping* diawali dengan memeriksa keberadaan gambar. Jika gambar tersedia, proses *looping* akan dimulai. Sebaliknya, jika gambar tidak ditemukan, program akan menghapus *node* dan mengakhiri proses.



Gambar 3.6. Alur pengumpulan data bagian gambar

Flowchart pada Gambar 3.6 menjelaskan alur proses ekstraksi gambar yang terdiri dari tiga langkah utama. Pertama, gambar diambil dari ROS. Selanjutnya, gambar tersebut dikonversi menggunakan *CVBridge* agar dapat digunakan untuk pelatihan model AI. Di sisi lain, setiap gambar diberi nama dan ID secara otomatis. Setelah itu, gambar, nama, dan ID disimpan dalam sebuah folder, menghasilkan *output* berupa jalur file gambar (*image path*).



Gambar 3.7. Alur pengumpulan data bagian *twist message*

Flowchart pada Gambar 3.7 menggambarkan proses ekstraksi *twist message*, yang merupakan salah satu langkah penting dalam pengumpulan data untuk pelatihan model. Pada tahap ini, dua nilai float utama diambil, yaitu *angular velocity* dan *linear velocity*. Data tersebut, bersama dengan *Image path* yang merujuk pada lokasi gambar hasil tangkapan kamera, disimpan secara terstruktur dalam file CSV. Kumpulan data yang dihasilkan dari proses ini dikenal sebagai data peneliti, yang berfungsi sebagai referensi untuk melatih model AI agar mampu meniru perilaku pengemudi manusia. Struktur data peneliti ini dapat dilihat lebih

rinci pada ilustrasi di Tabel 3.2, yang menunjukkan bagaimana data tersebut diorganisasi.

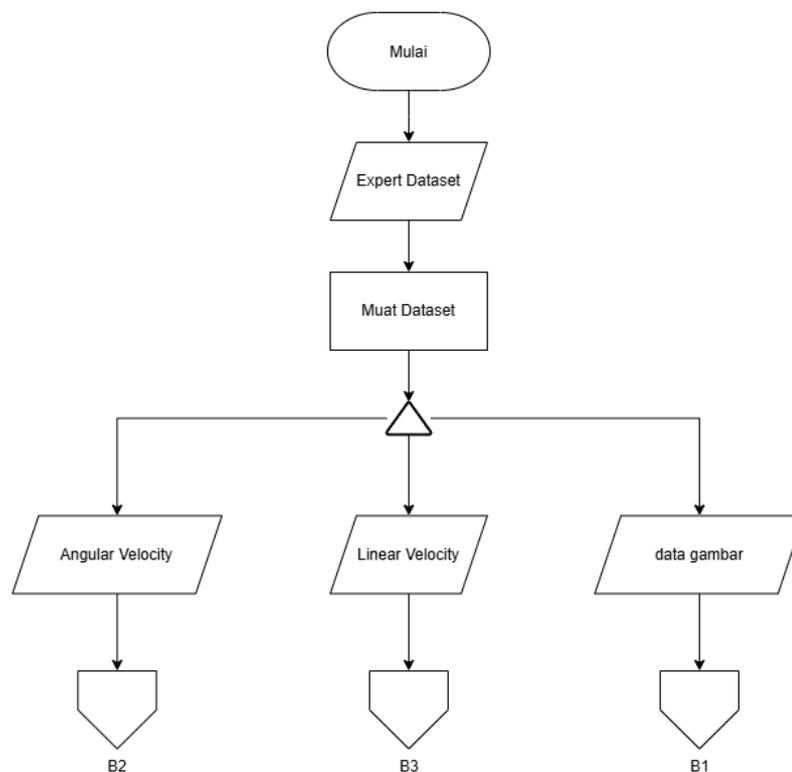
Tabel 3.2. Struktur data peneliti yang direncanakan

<i>Right Image</i>	<i>Center Image</i>	<i>Left Image</i>	<i>Linear velocity</i> (m/s)	<i>Angular velocity</i> (rad/s)
<i>Image/right_1</i>	<i>Image/center_1</i>	<i>Image/left_1</i>	0.1003	0.2

3.3.2 Pembuatan Model

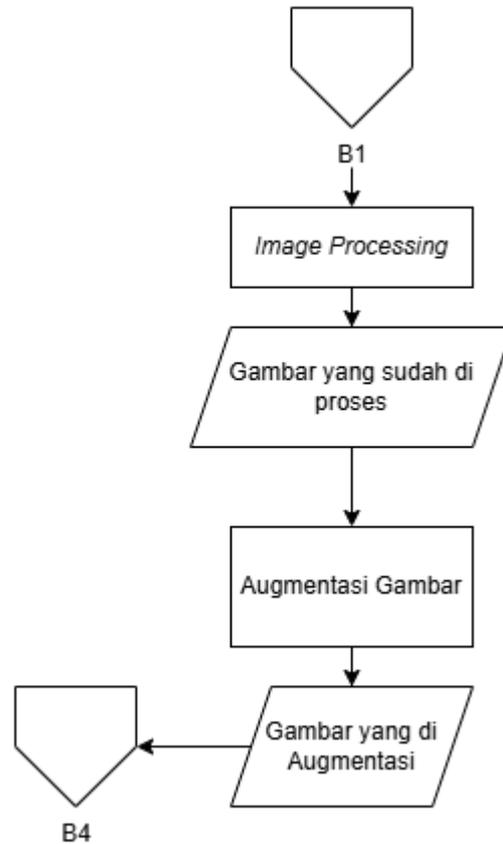
Tahap berikutnya adalah merancang dan melatih model *Behavior cloning* menggunakan pendekatan *Imitation Learning*. Model ini dilatih dengan algoritma pembelajaran mesin yang bertujuan untuk mengubah data perilaku pengemudi manusia menjadi kontrol sistem, berupa *angular velocity*. Proses pelatihannya melibatkan tiga tahap utama, yaitu *preprocessing* data, pelatihan model, dan evaluasi.

Pada tahap *preprocessing data*, dataset yang diperoleh dari pengemudi ahli diproses dan dibagi menjadi tiga komponen utama, yakni data gambar, *linear velocity*, dan *angular velocity*, seperti yang digambarkan dalam *flowchart* pada Gambar 3.8.



Gambar 3.8. Flowchart tahap awal *preprocessing* data

Gambar 3.9 menunjukkan alur pemrosesan data gambar, yang melibatkan tahapan *image processing* dan augmentasi. Proses ini dirancang untuk meningkatkan kualitas dan variasi data yang digunakan dalam pelatihan model, sehingga model mampu mengenali berbagai kondisi lingkungan secara lebih baik.

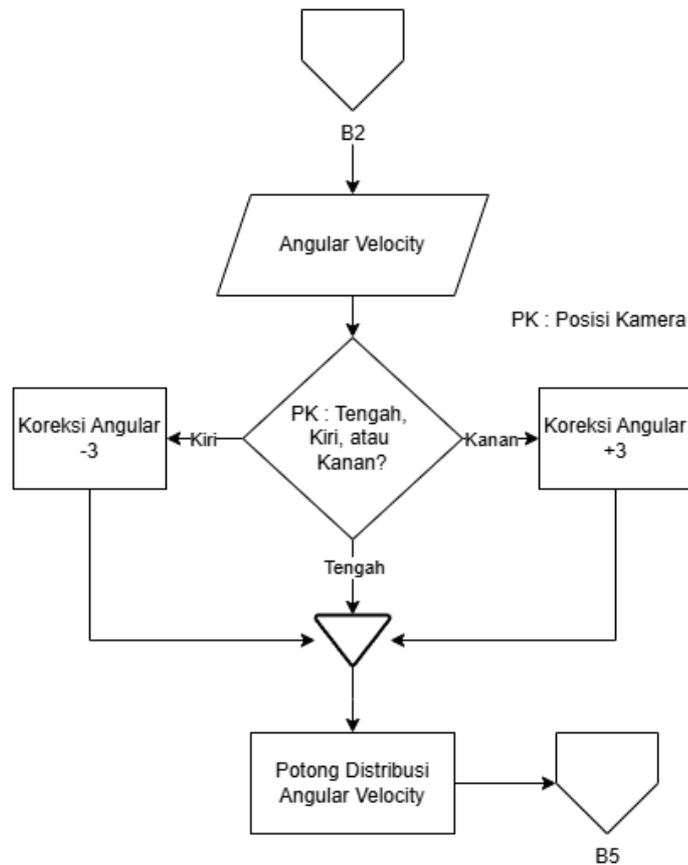


Gambar 3.9. Flowchart bagian *preprocessing* gambar

Langkah pertama dalam pemrosesan gambar adalah memotong bagian tertentu dari gambar asli untuk memastikan bahwa fokus visual tertuju pada area jalan yang relevan dengan jalur kendaraan. Setelah itu, data akan langsung melalui tahap augmentasi.

Pada tahap ini, dua teknik utama diterapkan yaitu *random brightness* dan *random shadow*. Teknik *random brightness* digunakan untuk mensimulasikan perubahan intensitas cahaya, seperti saat berkendara di bawah sinar matahari atau bayangan. Sementara itu, *random shadow* menciptakan bayangan acak pada gambar untuk mereplikasi kondisi visual yang lebih realistis di jalan. Hasil akhirnya adalah gambar yang telah mengalami augmentasi, yang lebih beragam dan sesuai

untuk melatih model agar lebih tangguh terhadap berbagai kondisi pencahayaan dan lingkungan.

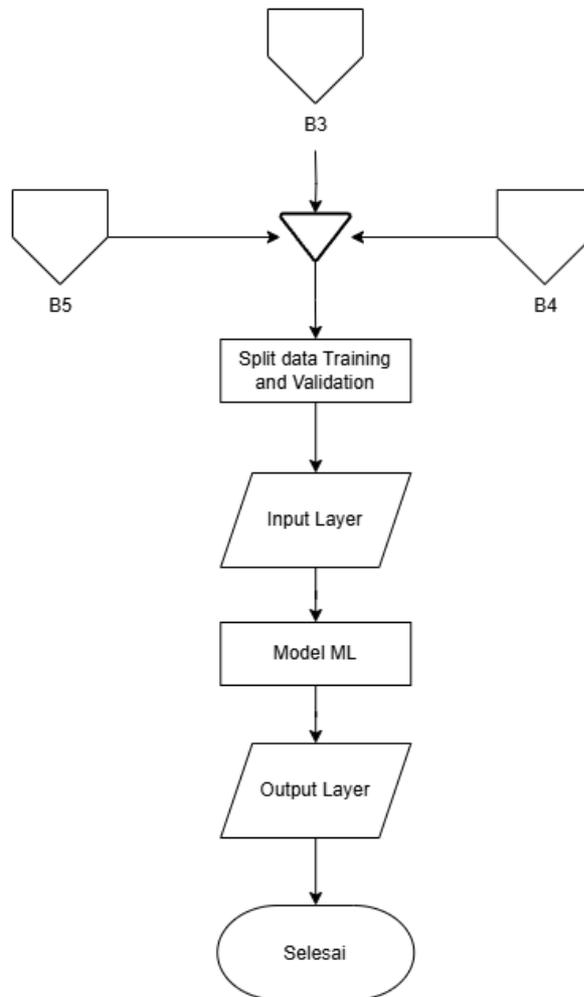


Gambar 3.10. Preprocessing data angular velocity

Proses pengolahan *angular velocity* ditunjukkan dalam diagram pada Gambar 3.10, di mana terdapat pengambilan keputusan berdasarkan jenis kamera yang digunakan, yaitu kamera tengah, kanan, atau kiri. Jika data berasal dari kamera kanan, nilai *angular velocity* akan dikoreksi dengan menambahkan +0.3 untuk menyesuaikan perspektif arah belok. Sebaliknya, jika berasal dari kamera kiri, koreksi dilakukan dengan mengurangi -0.3. Untuk data dari kamera tengah, tidak ada koreksi yang diterapkan karena data tersebut sudah dianggap representatif tanpa penyesuaian tambahan.

Selanjutnya, distribusi data akan disesuaikan untuk menghindari dominasi oleh satu jenis data tertentu, sehingga model dapat belajar secara lebih seimbang dari seluruh variasi data yang ada. Setelah semua proses penyesuaian selesai, data yang telah diproses akan dilanjutkan ke tahap pengolahan berikutnya untuk digunakan dalam pelatihan model.

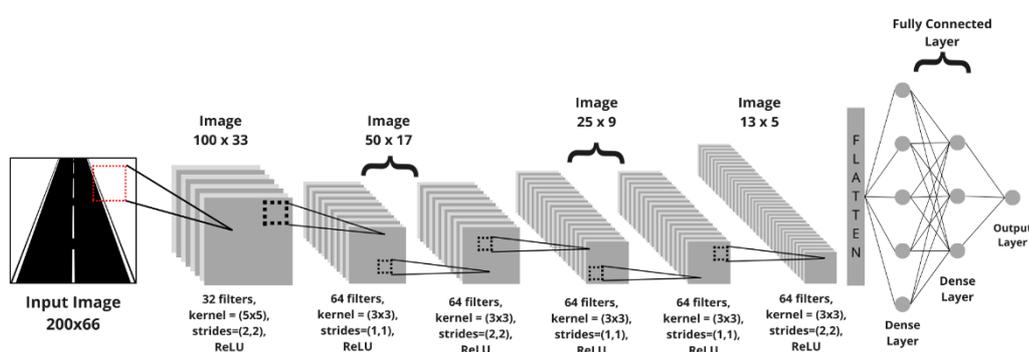
Pendekatan ini dirancang untuk meningkatkan kemampuan model dalam memahami dan mereplikasi perilaku kendaraan dengan lebih akurat. Dengan mempertimbangkan kondisi visual dari berbagai perspektif kamera, model diharapkan dapat bekerja lebih adaptif terhadap skenario yang kompleks di dunia nyata.



Gambar 3.11. *Flowchart* bagian akhir *preprocessing* data

Setelah melalui tahapan *image processing* dan koreksi *angular velocity*, data yang telah diproses kemudian dibagi menjadi dua kelompok, yaitu data untuk pelatihan (*training data*) dan data untuk validasi (*validation data*). Proses pembagian ini bertujuan untuk memastikan bahwa model dapat belajar dari data pelatihan dan diuji kemampuannya pada data validasi yang tidak digunakan selama pelatihan. Dengan cara ini, performa model dapat dievaluasi secara objektif pada data baru yang tidak pernah dilihat sebelumnya.

Setelah data dibagi, langkah berikutnya adalah melatih (*training*) model menggunakan algoritma pembelajaran yang telah dirancang. Model ini memanfaatkan data pelatihan untuk mempelajari pola hubungan antara masukan berupa gambar hasil *image processing* dan keluaran berupa koreksi *angular velocity*. Proses pelatihan mencakup iterasi berulang untuk mengoptimalkan parameter model agar dapat memprediksi keluaran dengan akurasi yang tinggi. Rincian arsitektur model dan mekanisme pelatihannya dapat dilihat pada diagram di Gambar 3.12, yang memberikan gambaran visual mengenai desain dan alur kerja model yang digunakan.



Gambar 3.12. Model layer yang terdiri dari CNN dan FCN

Model ini dilatih menggunakan 6 lapisan *Convolutional Neural Networks* (CNN) dan 2 lapisan *Fully Connected Networks* (FCN), yang secara keseluruhan bertujuan untuk memprediksi *angular velocity* berdasarkan citra kamera yang diambil dari kendaraan otonom. Pada bagian CNN, model dirancang khusus untuk mengekstraksi fitur-fitur penting dari gambar *input*, seperti garis jalan, tepi jalan, atau objek lain yang relevan untuk navigasi. Ini dilakukan melalui kombinasi lapisan Conv2D yang bertugas melakukan konvolusi terhadap gambar, lapisan *BatchNormalization* yang membantu menormalkan *output* dari lapisan konvolusi untuk meningkatkan stabilitas dan mempercepat pelatihan, serta fungsi aktivasi ReLU yang memperkenalkan non-linearitas ke dalam model agar mampu menangkap pola yang lebih kompleks.

Setelah fitur-fitur penting diproses oleh lapisan CNN, model mengirimkan *output* ke lapisan FCN. Pada bagian ini, fitur yang telah diekstraksi diratakan melalui lapisan *Flatten*, dan kemudian diproses oleh lapisan *Dense*. Di sinilah keputusan akhir mengenai *angular velocity* diambil. Lapisan FCN juga dilengkapi

dengan mekanisme *dropout*, yang berfungsi untuk mencegah *overfitting* dengan secara acak mengabaikan sebagian neuron selama pelatihan. Dengan demikian, model tidak hanya mengandalkan beberapa fitur saja, tetapi belajar generalisasi yang lebih baik terhadap data.

Kombinasi lapisan CNN untuk ekstraksi fitur dan FCN untuk pengambilan keputusan akhir membuat model ini mampu memetakan gambar *input* dari kamera langsung menjadi perintah kendali berupa *angular velocity*.

```

Fungsi behavior_cloning():
  Inisialisasi model sebagai Sequential()

  Tambahkan InputLayer dengan bentuk (66, 200, 3) ke model

  Tambahkan lapisan Conv2D dengan 32 filter, ukuran kernel (5, 5), stride (2, 2),
    padding 'same', dan aktivasi 'relu'
  Tambahkan lapisan BatchNormalization

  Untuk setiap indeks dari 0 hingga 4 (5 iterasi):
    - Setel ukuran kernel menjadi (3, 3)
    - Jika indeks genap, setel stride menjadi (2, 2), jika tidak, setel stride menjadi (1, 1)
    - Tambahkan lapisan Conv2D dengan 64 filter, ukuran kernel saat ini, stride saat ini,
      padding 'same', dan aktivasi 'relu'
    - Tambahkan lapisan BatchNormalization

  Tambahkan lapisan Flatten ke model

  Tambahkan lapisan Dense dengan 128 unit dan aktivasi 'relu'
  Tambahkan lapisan Dropout dengan tingkat 0.3
  Tambahkan lapisan BatchNormalization

  Tambahkan lapisan Dense dengan 64 unit dan aktivasi 'relu'
  Tambahkan lapisan Dropout dengan tingkat 0.3
  Tambahkan lapisan BatchNormalization

  Tambahkan lapisan Dense dengan 1 unit (lapisan output)

  Inisialisasi optimizer Adam dengan laju pembelajaran 1e-4
  Kompilasi model dengan loss 'mse' dan optimizer Adam

  Kembalikan model

Inisialisasi model dengan memanggil behavior_cloning()
Cetak ringkasan model

```

Gambar 3.13. *Pseudo Code Layer Behavior cloning*

Gambar 3.13 merupakan *Pseudo Code* dari *layer behavior cloning* yang dibangun. *pseudo code* ini memberikan gambaran rinci tentang bagaimana algoritma *behavior cloning* diterapkan dalam model ini, termasuk bagaimana data diproses melalui lapisan-lapisan CNN dan FCN serta bagaimana keputusan pengendalian akhir dibuat berdasarkan fitur yang diekstraksi.

3.3.3 Evaluasi Model

Setelah proses pelatihan selesai, langkah selanjutnya adalah mengevaluasi performa model dengan menghitung nilai *Mean Squared Error* (MSE). MSE adalah salah satu metrik evaluasi yang digunakan untuk mengukur seberapa dekat prediksi model terhadap nilai sebenarnya. Metrik ini dihitung menggunakan persamaan 3-2.

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=0}^N (a_i - \hat{a}_i)^2 \quad 3-2$$

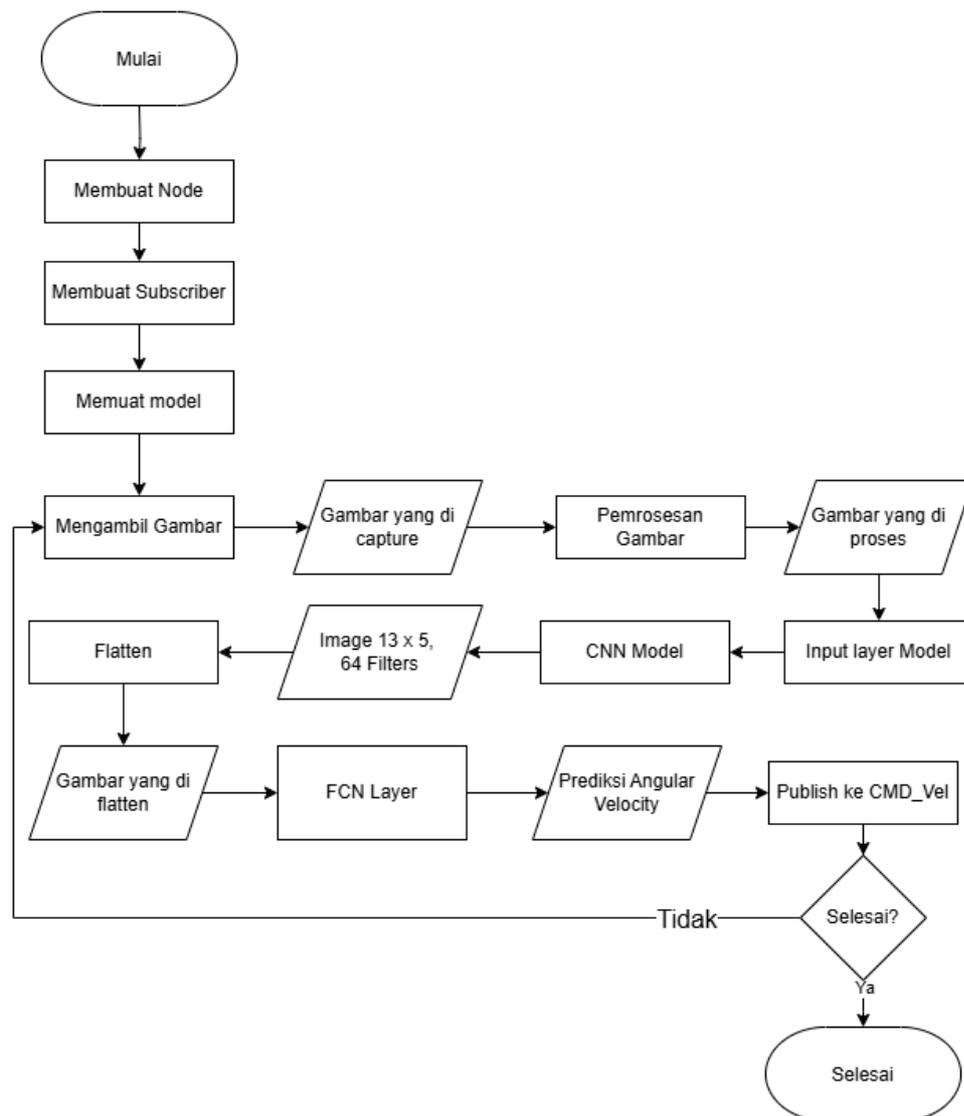
Pada persamaan tersebut, a_i mewakili nilai aktual atau aksi sebenarnya yang diambil, sementara \hat{a}_i adalah nilai prediksi yang dihasilkan oleh model. Variabel N mengacu pada jumlah total data yang digunakan dalam evaluasi.

MSE bekerja dengan menghitung rata-rata dari kuadrat selisih antara nilai aktual dan prediksi. Nilai MSE yang lebih kecil menunjukkan bahwa model memiliki performa yang lebih baik, karena prediksi yang dihasilkan semakin mendekati nilai sebenarnya. Sebaliknya, nilai MSE yang besar menunjukkan bahwa prediksi model memiliki deviasi yang signifikan dari nilai sebenarnya, yang menunjukkan perlunya perbaikan pada model atau data pelatihan. Jika model tidak sesuai atau nilai MSE masih tinggi, maka data pada model akan diperbaiki dan pelatihan akan dilakukan kembali.

Penggunaan MSE sebagai metrik evaluasi sangat relevan dalam konteks ini, karena memberikan wawasan kuantitatif mengenai akurasi model dalam memprediksi aksi berbasis data yang telah diproses. Dengan menghitung MSE, pengembang dapat mengevaluasi kualitas model secara objektif dan mengidentifikasi area yang memerlukan optimasi lebih lanjut.

3.3.4 Penerapan Model

Tahap selanjutnya adalah menguji model secara menyeluruh untuk memastikan kinerjanya dalam berbagai skenario dan kondisi jalan dengan cara diterapkan pada kendaraan. Proses pengujian ini bertujuan untuk mengevaluasi kemampuan model dalam meniru perilaku pengemudi manusia, sehingga kendaraan otonom dapat beroperasi dengan baik. Ilustrasi tahapan pengujian model dapat dilihat pada *flowchart* yang disajikan di Gambar 3.14.



Gambar 3.14. *Flowchart* untuk melakukan pengujian model

Proses pengujian dimulai dengan program membuat *node* pada *platform* ROS (*Robot Operating System*). Setelah *node* dibuat, sistem kemudian menetapkan *subscriber* untuk menerima data dari kamera yang terpasang pada kendaraan otonom. Langkah berikutnya adalah memuat model yang telah dilatih. Kamera kemudian mengambil gambar secara langsung dari lingkungan sekitar, yang selanjutnya diproses melalui tahap *image processing*.

Hasil dari *image processing* ini kemudian dijadikan *input* ke dalam model Convolutional Neural Network (CNN). Model CNN menghasilkan *output* berupa citra berukuran 13x5 piksel yang kemudian diubah menjadi bentuk satu dimensi (*flatten*). Setelah itu, data hasil *flattening* diproses oleh Fully Connected Network

(FCN) untuk menghasilkan prediksi nilai *angular velocity*. Nilai prediksi ini kemudian dipublikasikan melalui topik `cmd_vel`, yang digunakan untuk mengontrol gerakan roda kendaraan otonom.

Penerapan dilakukan melalui simulasi dengan satu skenario di Gazebo untuk kendaraan simulasi, kemudian dilanjutkan dengan tiga skenario berkendara pada pagi, siang, dan sore hari untuk percobaan di O-SEATER.

3.4 Evaluasi

Pada bagian ini, akan dibahas mengenai beberapa kekurangan yang terdapat dalam penelitian yang telah dilaksanakan. Identifikasi terhadap kekurangan-kekurangan tersebut bertujuan untuk memberikan gambaran yang lebih jelas mengenai area yang masih dapat diperbaiki atau dikembangkan lebih lanjut dalam penelitian selanjutnya. Evaluasi ini mencakup beberapa aspek penting, seperti metodologi yang digunakan, keterbatasan dalam pengumpulan data, serta variabel-variabel yang mungkin belum sepenuhnya tercakup atau dieksplorasi dalam penelitian ini.

Selain itu, berbagai faktor eksternal yang dapat mempengaruhi hasil penelitian, seperti keterbatasan waktu, sumber daya, atau hambatan teknis lainnya, juga akan diulas untuk memberikan pemahaman yang lebih menyeluruh. Adapun rekomendasi yang diajukan diharapkan dapat menjadi pedoman bagi peneliti selanjutnya untuk memperbaiki dan mengembangkan penelitian ini, baik dari segi konsep, pendekatan, maupun implementasinya.

Dengan adanya evaluasi ini, diharapkan penelitian berikutnya dapat mengatasi kendala yang ada dan memperkaya temuan-temuan yang lebih relevan serta aplikatif dalam konteks yang lebih luas.