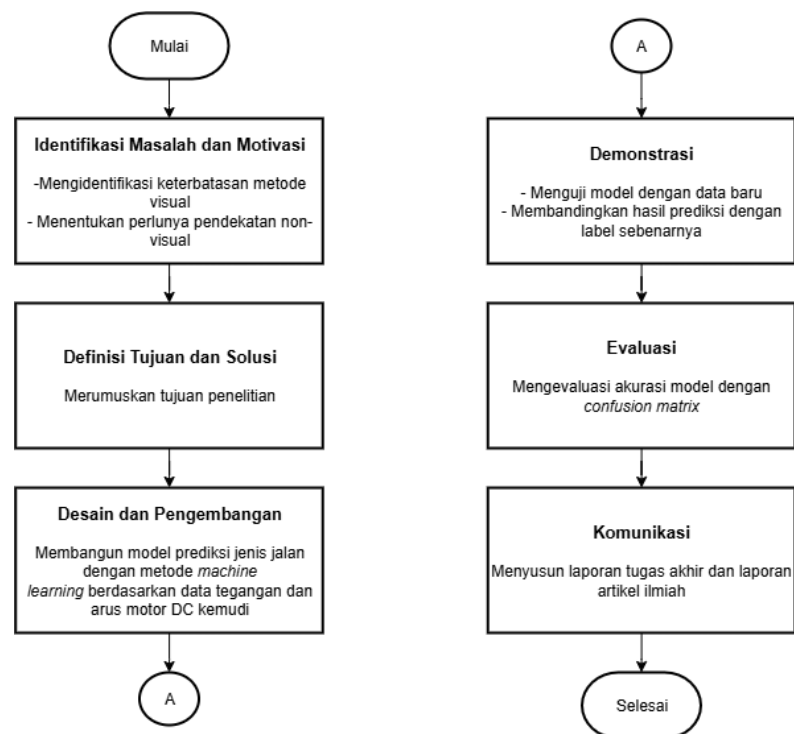


BAB III

METODE PENELITIAN

Bab ini menjelaskan langkah-langkah yang dilakukan dalam penelitian. Mulai dari desain penelitian yang menggambarkan alur kegiatan penelitian, kemudian dijelaskan arsitektur sistem yang digunakan untuk memproses data dan menghasilkan prediksi. Selanjutnya dibahas perancangan model prediksi, yang mencakup cara membangun model *machine learning* berdasarkan data yang diperoleh. Pada bagian sumber data, dijelaskan asal data yang digunakan, serta instrumen penelitian yang mencakup metrik evaluasi serta alat dan bahan yang mendukung proses penelitian.

3.1 Desain Penelitian



Gambar 3. 1 Flowchart DSRM pada Penelitian

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan pendekatan *Design Science Research Methodology* (DSRM). Pemilihan metode R&D didasarkan pada tujuan utama penelitian, yaitu menghasilkan suatu produk

berupa model prediksi jenis jalan. Metode R&D relevan karena tidak hanya menekankan pada proses penelitian, tetapi juga pada penciptaan, pengembangan, serta pengujian produk agar dapat digunakan secara efektif dalam konteks yang sesuai. Dengan kata lain, R&D mendukung orientasi penelitian yang berfokus pada pembuatan sistem prediksi jalan yang dapat langsung diimplementasikan.

Agar proses pengembangan lebih terstruktur, penelitian ini menggunakan pendekatan DSRM. Berbeda dengan R&D yang lebih bersifat umum dalam menciptakan dan menguji produk, DSRM memberikan kerangka kerja yang sistematis dengan enam tahapan, yaitu: identifikasi masalah dan motivasi, penentuan tujuan, desain dan pengembangan, demonstrasi, evaluasi, dan komunikasi. Pendekatan ini dipilih karena DSRM tidak hanya menekankan pada hasil berupa artefak teknologi pada penelitian ini model prediksi jalan, tetapi juga pada proses ilmiah bagaimana artefak tersebut dirancang, dibangun, diuji, dan dievaluasi secara komprehensif.

Dengan mengombinasikan R&D dan DSRM, penelitian ini memastikan bahwa model prediksi yang dihasilkan tidak hanya berfungsi sebagai produk yang bekerja, tetapi juga memiliki dasar metodologis yang kuat, teruji secara ilmiah, dan relevan dengan permasalahan nyata yang diteliti (sesuai karakteristik DSRM) [62], [63].

3.1.1 Identifikasi Masalah dan Motivasi (*Problem Identification and Motivation*)

Prediksi jenis permukaan jalan merupakan elemen penting dalam sistem pendukung pengendalian kendaraan. Selama ini, pendekatan yang paling umum digunakan adalah berbasis citra visual, seperti kamera atau citra satelit. Namun, metode ini memiliki beberapa keterbatasan, terutama ketika menghadapi kondisi lingkungan yang tidak ideal, seperti pencahayaan rendah, gambar buram, atau gangguan cuaca seperti hujan dan kabut.

Kondisi tersebut mendorong perlunya pendekatan alternatif yang lebih andal dan tidak bergantung pada data visual. Dalam penelitian ini, dipilih pendekatan berbasis data arus dan tegangan yang dihasilkan dari motor DC kemudi. Dengan memanfaatkan karakteristik data non-visual tersebut, model prediksi yang dibangun diharapkan mampu bekerja lebih konsisten dalam berbagai kondisi jalan.

3.1.2 Definisi Tujuan dan Solusi (*Define the Objectives for a Solution*)

Tujuan dari penelitian ini adalah membangun model prediksi jenis permukaan jalan dengan memanfaatkan data arus dan tegangan dari motor DC kemudi sebagai variabel utama. Model ini dirancang untuk mengklasifikasikan jenis jalan secara akurat tanpa bergantung pada sensor visual. Sistem ini diharapkan mampu memberikan kontribusi dalam pengembangan model prediksi jenis jalan yang lebih andal dan responsif terhadap berbagai kondisi jalan yang berbeda.

3.1.3 Desain dan Pengembangan (*Design and Development*)

Pada tahap ini, artefak solusi membangun sistem prediksi jenis permukaan jalan berbasis *machine learning* yang memanfaatkan data arus dan tegangan dari motor DC kemudi. Sistem ini dirancang sebagai alternatif terhadap metode berbasis visual yang memiliki keterbatasan pada kondisi lingkungan tertentu, sehingga diharapkan dapat digunakan secara andal di berbagai skenario jalan.

Tiga algoritma *supervised learning*, yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Gradient Boosting Decision Tree* (GBDT), diimplementasikan pada sistem prediksi. Hasil kinerja dari masing-masing algoritma kemudian dibandingkan untuk menentukan algoritma dengan performa terbaik dalam mengklasifikasikan jenis permukaan jalan berdasarkan data arus dan tegangan motor DC kemudi.

3.1.4 Demonstrasi (*Demonstration*)

Tahap demonstrasi dalam penelitian ini dilakukan untuk menguji dan mendemonstrasikan kemampuan model dalam memprediksi jenis permukaan jalan berdasarkan data arus dan tegangan dari motor DC kemudi. Pengujian dilakukan menggunakan data uji yang belum pernah digunakan dalam proses pelatihan, sehingga sekaligus berfungsi sebagai demonstrasi keandalan model terhadap data baru.

Data uji diperoleh dengan merekam topik ROS `/sensor_data`, yang memuat *array* nilai tegangan dan arus selama pergerakan kemudi. Arah kemudi dikendalikan melalui topik ROS `/yaw`, menghasilkan perubahan nilai arus dan tegangan sesuai dengan arah gerakan. Empat skenario pergerakan kemudi

diterapkan, yaitu dari tengah ke kanan, kanan ke tengah, tengah ke kiri, dan kiri ke tengah. Setiap skenario dijalankan pada dua jenis permukaan jalan yang berbeda, yakni permukaan epoxy dan beton.

Model klasifikasi yang diuji adalah hasil pelatihan dari tiga algoritma *machine learning* yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Gradient Boosting Decision Tree* (GBDT). Masing-masing model disimpan dalam format .pkl (*pickle*) dan digunakan untuk memprediksi label jenis jalan pada data uji.

3.1.5 Evaluasi (*Evaluation*)

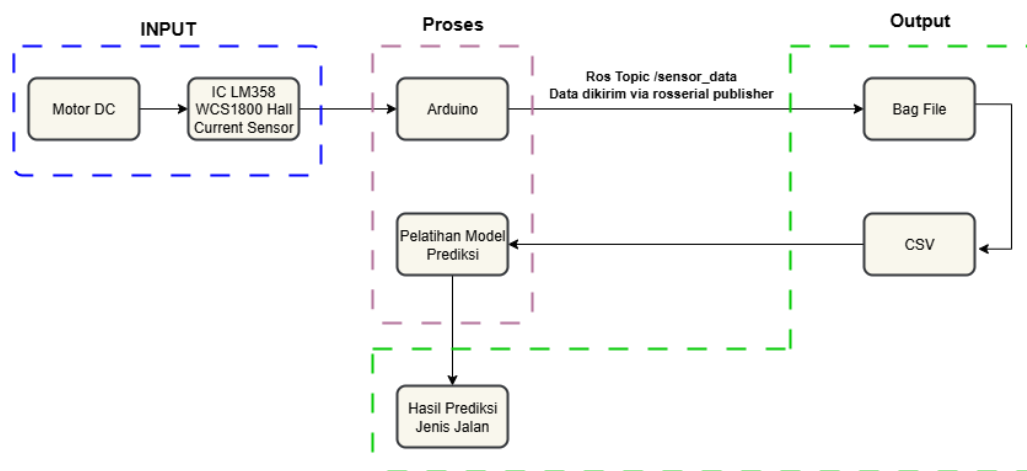
Evaluasi model dilakukan untuk mengukur tingkat akurasi dan keandalan dalam mengklasifikasikan jenis permukaan jalan. Pengujian performa model menggunakan *confusion matrix* untuk menilai hasil klasifikasi berdasarkan jumlah prediksi benar dan salah. Selain itu, metode *k-fold cross validation* digunakan untuk meminimalkan risiko *overfitting* dan memastikan evaluasi model yang lebih stabil dan representatif.

3.1.6 Komunikasi (*Communication*)

Hasil dari penelitian ini akan dikomunikasikan dalam penulisan laporan tugas akhir dan penulisan artikel ilmiah untuk dipublikasikan di jurnal nasional maupun internasional.

3.2 Arsitektur Sistem

Arsitektur sistem secara keseluruhan pada penelitian ini dirancang untuk merekam dan mengolah data tegangan dan arus dari motor DC kemudi, yang selanjutnya digunakan dalam pelatihan model *machine learning* untuk memprediksi jenis permukaan jalan.



Gambar 3. 2 Arsitektur Sistem Prediksi Jenis Jalan

Bedasarkan Gambar 3.2, sistem terdiri dari tiga tahapan utama yaitu *input*, *proses*, dan *output*. Pada tahap *input*, motor DC berperan sebagai aktuator untuk menggerakkan kemudi kendaraan, sementara IC LM358 dan WCS1800 *Hall Current Sensor* digunakan untuk perubahan tegangan dan arus yang terjadi selama tersebut. Kedua komponen ini menghasilkan sinyal analog sesuai dengan besarnya tegangan dan arus yang mengalir, yang merepresentasikan beban kerja motor saat bergerak di atas permukaan jalan yang berbeda.

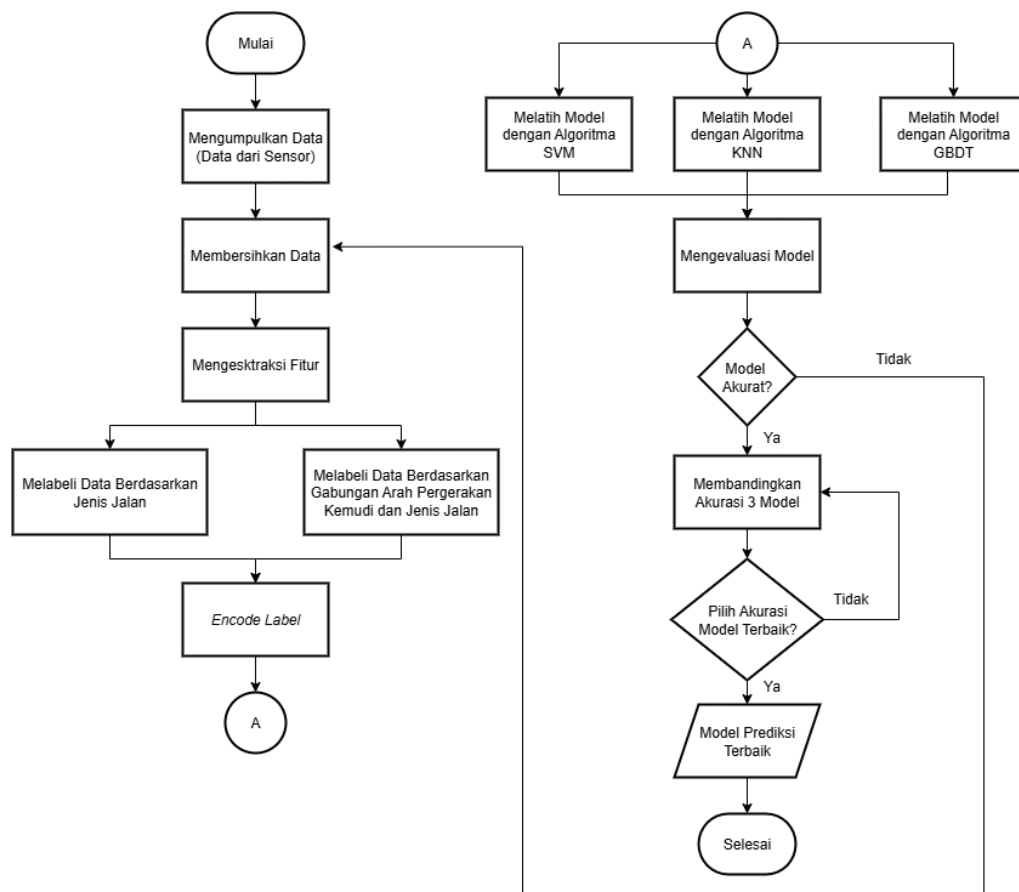
Selanjutnya, sinyal analog dari sensor dikirim ke mikrokontroler Arduino, yang berfungsi untuk mengubah sinyal analog menjadi data digital. Arduino kemudian mengirimkan data tersebut ke komputer menggunakan protokol komunikasi *rosserial*, yaitu sebuah paket di *Robot Operating System* (ROS) yang memungkinkan komunikasi antara mikrokontroler dan sistem ROS.

Data yang diterima oleh ROS dipublikasikan melalui *topic* tertentu dan direkam menggunakan *rosvag record*, yang kemudian menghasilkan *file* dalam format *bag file*. File ini menjadi sumber utama yang merekam seluruh proses pengujian dalam bentuk terstruktur. Untuk memudahkan dalam proses analisis dan pelatihan model *machine learning*, *bag file* dikonversi ke dalam format CSV. Terakhir, data CSV digunakan untuk melatih model *machine learning*. Model ini akan dilatih untuk mengenali pola-pola tertentu dalam data sensor dan mengklasifikasikan jenis permukaan jalan berdasarkan *input* tersebut.

File CSV ini kemudian dijadikan sebagai *input* pada tahap pelatihan model prediksi. Dalam tahap ini, model *machine learning* dilatih untuk mengenali pola-pola dalam data tersebut yang berkaitan dengan karakteristik jenis permukaan jalan. Setelah model selesai dilatih, sistem mampu melakukan prediksi terhadap jenis permukaan jalan berdasarkan data sensor tegangan dan arus yang baru. Hasil dari proses ini menjadi *output* akhir berupa klasifikasi jenis jalan seperti beton atau epoxy.

3.3 Perancangan Model Prediksi

Perancangan model prediksi jenis jalan berbasis *machine learning* ditunjukkan pada Gambar 3.3 berupa *flowchart* yang dimulai dari proses akuisisi hingga evaluasi model.



Gambar 3.3 *Flowchart* Perancangan Model Prediksi Jenis Jalan

Proses diawali dengan pengumpulan data sensor, di mana data tegangan dan arus dari motor DC kemudi dikumpulkan menggunakan *rosbag record* pada sistem

ROS, sehingga menghasilkan *file* dalam format *.bag*. *File* tersebut kemudian dikonversi ke dalam format *.csv* agar dapat dianalisis dan diproses lebih lanjut.

Tahap berikutnya adalah pra-olah data. Proses ini diawali dengan pembersihan data, yaitu menghapus nilai-nilai statis atau data yang terekam saat kemudi tidak bergerak. Setelah pembersihan, dilakukan analisis awal terhadap distribusi data, yang menunjukkan ketidakseimbangan jumlah data untuk setiap arah pergerakan kemudi. Untuk mengatasi hal tersebut, dilakukan ekstraksi fitur untuk menyeimbangkan data.

Ekstraksi fitur dilakukan dengan menggunakan perhitungan statistik dasar terhadap data tegangan dan arus. Fitur-fitur yang dihasilkan antara lain, rata-rata tegangan (*voltage_avg*), rata-rata arus (*current_avg*), standar deviasi tegangan (*voltage_std*), standar deviasi arus (*current_std*), nilai maksimum dan minimum dari tegangan dan arus (*voltage_max*, *voltage_min*, *current_max*, *current_min*), serta jumlah data yang terekam dalam setiap pergerakan kemudi (*n_data*). Dengan demikian, dari data mentah yang awalnya hanya berisi tegangan dan arus, diperoleh total sembilan parameter hasil ekstraksi fitur.

Setelah ekstraksi fitur selesai, dilakukan proses pelabelan data, yaitu pemberian label pada setiap entri data sesuai dengan jenis jalan yang dilalui. Dua jenis dataset disiapkan yaitu, *dataset* dengan label umum misalnya epoxy dan beton, serta *dataset* yang dilabeli lebih spesifik berdasarkan arah pergerakan kemudi dan jenis jalan misalnya *ka_epoxy*, *ka_beton*, dan sebagainya.

Selanjutnya, data dibagi menjadi dua subset, yaitu data latih dan data uji, untuk keperluan pelatihan dan validasi model. Kolom target (jenis jalan) kemudian dikonversi dari tipe *object* menjadi numerik menggunakan metode *Label Encoding*, agar dapat diproses oleh algoritma *machine learning*.

Pada tahap pelatihan dan validasi, digunakan tiga algoritma *supervised learning*, yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Gradient Boosting Decision Tree* (GBDT). Evaluasi performa model dilakukan menggunakan dua pendekatan, yakni *train-test split* dan *k-fold cross-validation*, guna meminimalkan risiko *overfitting*.

Setelah pelatihan, dilakukan evaluasi performa model menggunakan *confusion matrix*, dengan menghitung nilai *precision*, *recall*, dan *f1-score* dari masing-masing model. Jika hasil akurasi belum memadai, dilakukan evaluasi ulang terhadap tahapan sebelumnya. Namun, jika hasilnya telah memenuhi kriteria, proses dilanjutkan dengan membandingkan akurasi dari ketiga model guna menentukan model dengan performa terbaik. Dengan demikian, seluruh proses perancangan sistem prediksi jenis jalan berbasis *machine learning* dinyatakan selesai.

3.4 Sumber Data

3.4.1 Pengumpulan Data



Gambar 3. 4 *Flowchart* pengumpulan data dengan sensor dari motor DC kemudi

Pengumpulan data dilakukan yang ditunjukkan pada Gambar 3.4 diawali dengan dengan merekam *topic* ROS. Untuk sensor tegangan dan arus topik yang direkam adalah *topic /sensor_data*, yang mengirimkan dua buah pesan dalam bentuk array berisi data *voltage* (tegangan) dan *current* (arus).

Untuk mendapatkan data arus dan tegangan, diperlukan perancangan perangkat keras tambahan berupa rangkaian sensor yang terhubung ke motor DC kemudi. Rangkaian ini terdiri dari rangkaian sensor tegangan dan sensor arus WCS1800, yang masing-masing dihubungkan ke sistem mikrokontroler yang menjalankan sistem ROS. Rangkaian tersebut kemudian tersebut dipasang pada protipe kendaraan listrik yaitu *Micro Electric Vehicle* (MEVi) sebagai alat bantu dalam pengambilan data. MEVi merupakan sebuah protipe mobil listrik otonom yang dirancang untuk satu atau dua penumpang.

Proses pengambilan data dilakukan dalam kondisi statis, di mana kendaraan tidak digerakkan maju maupun mundur, melainkan hanya sistem kemudi yang dioperasikan. Selama proses pengambilan data, stir MEVi dikendalikan melalui sinyal dari *topic* ROS /yaw, yang mengatur arah kemudi. Untuk memastikan konsistensi posisi, sudut tengah stir ditetapkan pada 0° . Perubahan sudut kemudi ke kanan diatur sebesar 15° , sementara ke kiri adalah -15° , sehingga gerakan kemudi dapat dicatat dengan jelas dan konsisten. Setiap perubahan arah pada stir akan menghasilkan variasi data arus dan tegangan yang dapat mencerminkan karakteristik permukaan jalan.



Gambar 3. 5 Skenario pergerakan kemudi untuk pengambilan data

Proses pengambilan data dilakukan dalam empat skenario pergerakan kemudi kendaraan seperti yang ditunjukkan pada Gambar 3.5, yakni dari posisi

tengah ke kanan (1), dari kanan kembali ke tengah (2), dari tengah ke kiri (3), dan dari kiri kembali ke tengah (4). Masing-masing skenario dilakukan pada dua jenis permukaan jalan yang berbeda, yaitu permukaan epoxy dan permukaan beton seperti yang ditunjukkan pada Gambar 3.6. Pemilihan kedua jenis permukaan ini didasarkan pada perbedaan karakteristik fisik yang cukup kontras. Epoxy memiliki tekstur yang lebih halus dan licin, sedangkan beton memiliki permukaan yang lebih kasar dengan tingkat gesekan yang lebih tinggi. Perbedaan tersebut memengaruhi beban kerja motor kemudi dan tercermin pada variasi arus serta tegangan yang terekam oleh sensor.



Gambar 3. 6 Jenis permukaan jalan yang digunakan dalam penelitian

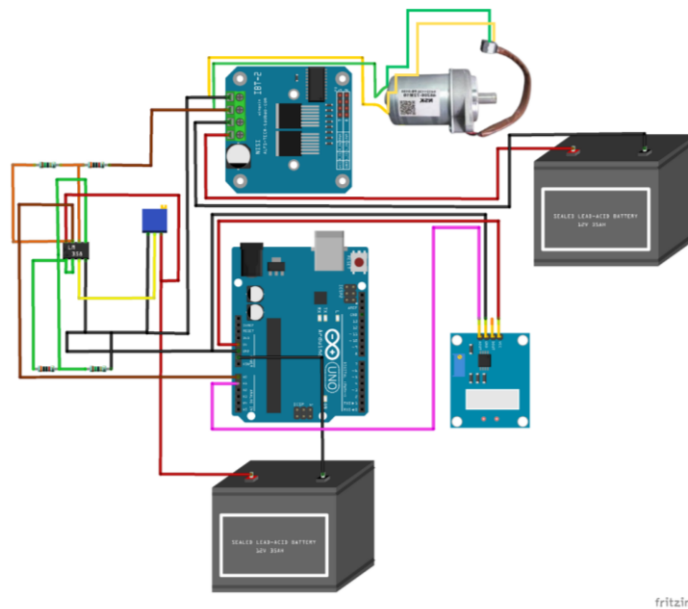
Tujuan dari pengambilan data dengan variasi ini adalah untuk memperoleh dataset yang mampu merepresentasikan perbedaan respon motor DC kemudi terhadap jenis permukaan jalan.

3.4.1.1 Perancangan Wiring Diagram

Proses perancangan sistem diawali dengan tahapan inisiasi, analisis sistem, serta analisis kebutuhan yang mencakup identifikasi komponen dan penyusunan kebutuhan teknis. Selanjutnya dilakukan perancangan elektrik yang mencakup penyusunan *wiring diagram* dan konfigurasi pin. Setelah seluruh tahapan perancangan selesai dan tervalidasi, proses perancangan perangkat keras dinyatakan selesai.

Hasil dari tahap perancangan elektrikan kemudian diwujudkan dalam bentuk *wiring diagram* yang ditunjukkan pada Gambar 3.8. *Wiring diagram* ini digunakan untuk proses pengambilan data tegangan dan arus dari motor DC kemudi. *Wiring diagram* berfungsi untuk memastikan bahwa setiap komponen,

mulai dari sensor hingga *actuator* terintegrasi dengan baik dan sistem dapat beroperasi secara efisien serta stabil.



Gambar 3. 7 Perancangan *Wiring Diagram* Sistem Prediksi Jenis Jalan

Pada sistem ini, Arduino Uno berperan sebagai mikrokontroler utama yang menjadi pusat kendali jalannya keseluruhan sistem. Komponen lainnya meliputi motor DC sebagai aktuator yang dikendalikan melalui driver motor L298N, sensor arus *WCS1800 Hall Current Sensor* untuk membaca besarnya arus dari motor, serta rangkaian sensor tegangan yang terdiri dari IC LM358, resistor, dan trimpot untuk mengukur tegangan keluaran motor. Sistem ini didukung oleh dua buah baterai yang berfungsi sebagai sumber daya utama bagi motor dan seluruh rangkaian.

Koneksi antar pin dalam rangkaian ini merepresentasikan integrasi antar komponen elektronik yang digunakan dalam sistem. Koneksi ini mencakup pin *input* dan *output* yang saling terhubung untuk memastikan pengoperasian sistem secara menyeluruh. Pin-pin dari modul pendukung seperti sensor arus, rangkaian sensor tegangan, driver motor dihubungkan secara langsung ke mikrokontroler. Untuk informasi lengkap mengenai hubungan antar pin dan komponen disajikan dalam Tabel 3.1.

Tabel 3. 1 Konfigurasi pin sensor

Module	Pin Module	Pin Arduino / Pin Driver Motor / Pin Trimpot	Fungsi
Rangkaian sensor tegangan	Pin 4	GND (Pin Arduino)	<i>Ground</i>
	Pin 8	IBT M+ (Pin Driver Motor)	Untuk mengecek tegangan dari motor
	Pin 7	A0 (Pin Arduino)	Koneksi untuk mengirimkan data ke arduino
	Pin 3	Output trimpot (Pin Trimpot)	Mengatur rentang pembacaan tegangan
WCS1800 Hall Current Sensor	VCC	5V (Pin Arduino)	<i>Input</i> tegangan dari Arduino ke modul WCS1800
	GND	GND (Pin Arduino)	<i>Ground</i>
	AOUT	A1 (Pin Arduino)	Koneksi untuk mengirimkan data ke arduino

3.4.2 Penyimpanan Data

Data dari sensor tegangan dan arus disimpan sebagai *bag file* dengan format .bag, yang diperoleh dengan merekam *topic* tertentu di ROS. Selanjutnya, data dalam format bag dapat di ekstrak menjadi format csv untuk memudahkan pengolahan data.

3.5 Instrumen Penelitian

3.5.1 Metrik Evaluasi

Instrumen penelitian yang digunakan dalam penelitian ini yaitu *confusion matrix* untutuk mengevaluasi tingkat keberhasilan model prediksi jenis jalan.

Tabel 3. 2 Confusion Matrix

		<i>True Class</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Predicted Class</i>	<i>Positive</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
	<i>Negative</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Pengukuran kinerja dengan menggunakan *confusion matrix* melibatkan empat kategori yang menggambarkan hasil dari proses prediksi sebagai berikut :

1. TP – *True Positive* : Diprediksi positif dan benar.
2. TN – *True Negative* : Diprediksi negatif dan benar.
3. FP – *False Positive* : Diprediksi positif dan salah.
4. FN – *False Negative* : Diprediksi negatif dan salah.

Dengan menggunakan kombinasi *confusion matrix*, terdapat beberapa metrik evaluasi yang berguna untuk mengukur *precision*, *recall*, dan *F1-score*.

Precision menggambarkan seberapa sering prediksi positif model benar dari seluruh prediksi positif yang dibuat. Nilai *precision* dapat diperoleh melalui persamaan berikut.

$$\frac{TP}{TP + FP} \quad (3.1)$$

Recall menggambarkan seberapa sering model mampu mendeteksi dalam menemukan kembali sebuah informasi. Nilai *recall* dapat diperoleh melalui persamaan berikut.

$$\frac{TP}{TP + FN} \quad (3.2)$$

F1-score menggambarkan penggabungan *precision* dan *recall* untuk memberikan nilai tentang kinerja model untuk mendeteksi sebanyak mungkin dengan tepat. Dengan kata lain, *f1-score* merupakan nilai rata-rata dari *precision* dan *recall*, yang kemudian memberikan penilaian antara kedua metrik tersebut. Nilai *f1-score* dapat diperoleh melalui persamaan berikut.

$$\frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3.3)$$

3.5.2 Alat dan Bahan

Pada penelitian ini digunakan alat dan bahan sesuai dengan kebutuhan penelitian. Alat dan bahan tersebut dikelompokkan dalam bentuk perangkat keras dan perangkat lunak. Adapun perangkat keras yang digunakan pada penelitian ini seperti pada tabel 3.3.

Tabel 3. 3 Perangkat keras yang digunakan dalam penelitian

Perangkat Keras	
Nama Perangkat	Deskripsi
WCS1800 Hall Current Sensor	Digunakan untuk mengukur arus
Rangkaian sensor tegangan : - Resistor - Trimpot 203 Multiturn - IC LM358 - Baterai 12V	Digunakan untuk mengukur tegangan
Arduino Uno	Digunakan sebagai mikrokontroler
Motor DC	Digunakan sebagai aktuator
Driver Motor IBT-2	Digunakan sebagai kontrol motor
Baterai 12V	Digunakan sebagai sumber daya

Tabel 3.4 menampilkan perangkat lunak yang digunakan dalam penelitian ini. Perangkat lunak ini memiliki berbagai fungsi untuk mendukung seluruh proses penelitian, mulai dari akuisisi data sensor hingga pengolahan dan analisis data.

Tabel 3. 4 Perangkat lunak yang digunakan dalam penelitian

Perangkat Lunak	
Nama Perangkat	Deskripsi
Robot Operating System Noetic Ninjemys	Perangkat lunak untuk mengolah, mengirim dan menerima data dari sensor tegangan dan arus.
Arduino IDE	Perangkat lunak untuk menulis, mengompilasi, dan mengunggah kode ke mikrokontroler
Visual Studio Code	Digunakan untuk menuliskan dan menjalankan kode program.
Google Colab	Digunakan untuk menuliskan dan menjalankan kode program.
Fritzing	Perangkat lunak untuk mendesain rangkaian alat
Python	Bahasa Pemrograman yang digunakan pada penelitian ini

Tabel 3.5 menampilkan spesifikasi komputer untuk mendukung seluruh proses penelitian.

Tabel 3. 5 Spesifikasi komputer yang digunakan dalam penelitian

Spesifikasi Komputer	
Nama Perangkat	Deskripsi
Jetson AGX Xavier	Model: Jetson AGX Xavier Developer Kit RAM: 32 GB
Linux Ubuntu	Ubuntu 20.04.6 LTS Release: 20.04 Codename: focal System Architecture: aarch64
Windows 11	Device Name: Asus Vivobook OS Edition: Windows 11 Home Single Language Processor: 12th Gen Intel® Core™ i7-12700H @ 2.30 GHz RAM: 16 GB GPU : Intel(R) Iris(R) Xe Graphics System Type: 64-bit Operating System, x64-based processor Version: 24H2