



PROPOSAL TUGAS AKHIR - EC234701

PENGEMBANGAN ROBOT QUADRUPED-LEGGED UNTUK ESTIMASI POSISI KOMPONEN OVERHEAT PADA GARUDU LISTRIK BERBASIS KAMERA TERMAL

I Wayan Agus Darmawan

NRP 5024211070

Dosen Pembimbing

Prof.Dr.Ir. Mauridhi Hery Purnomo, M.Eng.

NIP 19580916198601 1 001

Muhtadin, S.T., M.Sc.

NIP 19810609200912 1 003

Program Studi Strata 1 (S1) Teknik Komputer

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2025

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENGEMBANGAN ROBOT *QUADRUPED-LEGGED* UNTUK ESTIMASI POSISI KOMPONEN *OVERHEAT* PADA GARDU LISTRIK BERBASIS KAMERA TERMAL

PROPOSAL TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik pada Program Studi S-1
Teknik Komputer
Departemen Teknik Komputer
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh: **I Wayan Agus Darmawan**
NRP. 5024211070

Disetujui Oleh:

Prof.Dr.Ir. Mauridhi Hery Purnomo, M.Eng.
NIP: 19580916198601 1 001

(Pembimbing)

Muhtadin, S.T., M.Sc.
NIP: 19810609200912 1 003

(Ko-Pembimbing)

Mengetahui,
Kepala Departemen Teknik Komputer FTEIC-ITS

Dr. Supeno Mardi Susiki Nugroho,S.T.,M.T.
NIP 19700313199512 1 001

SURABAYA
Januari, 2025

[Halaman ini sengaja dikosongkan]

ABSTRAK

PENGEMBANGAN ROBOT *QUADRUPED-LEGGED* UNTUK ESTIMASI POSISI KOMPONEN *OVERHEAT* PADA GARDU LISTRIK BERBASIS KAMERA TERMAL

Nama Mahasiswa / NRP: I Wayan Agus Darmawan 5024211070

Departemen : Teknik Komputer FTEIC - ITS

Dosen Pembimbing : 1. Prof.Dr.Ir. Mauridhi Hery Purnomo, M.Eng.

2. Muhtadin, S.T., M.Sc.

Abstrak Peningkatan konsumsi listrik di Indonesia mendorong pengembangan infrastruktur kelistrikan, terutama gardu listrik yang rentan terhadap masalah *overheating*. Masalah ini dapat menyebabkan kerusakan serius dan gangguan distribusi listrik. Penelitian ini mengusulkan sistem pemantauan *overheat* berbasis *autonomous quadruped legged robot* yang dilengkapi *thermal camera* dan model deteksi objek *YOLOv8*. Robot ini dirancang untuk mendeteksi suhu abnormal pada komponen gardu listrik secara *real-time*, dengan sistem lokalisasi ganda yang menggabungkan GPS RTK dan *Direct LiDAR-Inertial Odometry (DLIO)*. Data dari IMU dan odometri difilter menggunakan *Extended Kalman Filter (EKF)* untuk mengurangi noise, sehingga menghasilkan estimasi posisi yang stabil sebelum digunakan dalam proses DLIO. Integrasi ini memungkinkan robot untuk mengikuti rute patroli dengan akurasi tinggi, menghindari hambatan menggunakan *Braitenberg controller*, dan memastikan navigasi aman di lingkungan yang penuh tantangan. Sistem dilengkapi antarmuka *web* untuk memudahkan operator dalam memantau dan mengontrol robot dari jarak jauh. Hasil penelitian ini diharapkan berkontribusi pada pengembangan teknologi robotika untuk pemantauan infrastruktur kelistrikan yang lebih efisien dan andal.

Kata Kunci: *Overheat, Quadruped robot, thermal camera, YOLOv8, Direct LiDAR-Inertial Odometry*

[Halaman ini sengaja dikosongkan]

ABSTRACT

Development of a Quadruped-Legged Robot for Position Estimation of Overheat Components in Electrical Substations Using a Thermal Camera

Student Name / NRP: I Wayan Agus Darmawan / 5024211070

Department : Computer Engineering FTEIC - ITS

Advisor : 1. Prof.Dr.Ir. Mauridhi Hery Purnomo, M.Eng.

2. Muhtadin, S.T., M.Sc.

Abstract The increasing electricity consumption in Indonesia drives the development of electrical infrastructure, particularly electrical substations, which are vulnerable to *overheating* issues. These problems can cause severe damage and disrupt power distribution. This study proposes an *overheat* monitoring system based on an *autonomous quadruped legged robot* equipped with a *thermal camera* and an object detection model using *YOLOv8*. The robot is designed to detect abnormal temperatures in substation components in *real-time*, with a dual localization system that combines GPS RTK and *Direct LiDAR-Inertial Odometry (DLIO)*. Data from the IMU and odometry are filtered using the *Extended Kalman Filter (EKF)* to reduce noise, providing a stable position estimate before being used in the DLIO process. This integration allows the robot to follow patrol routes with high accuracy, avoid obstacles using a *Braitenberg controller*, and ensure safe navigation in challenging environments. The system is equipped with a *web* interface to facilitate remote monitoring and control by the operator. The results of this research are expected to contribute to the development of more efficient and reliable robotic technologies for electrical infrastructure monitoring.

Keywords: *Overheat, Quadruped robot, thermal camera, YOLOv8, Direct LiDAR-Inertial Odometry*

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	v
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Manfaat Penelitian	3
1.5.1 Manfaat Praktis	3
1.5.2 Manfaat Teoritis	3
2 TINJAUAN PUSTAKA	5
2.1 Hasil penelitian/perancangan terdahulu	5
2.2 Hasil Penelitian/Perancangan Terdahulu	5
2.2.1 <i>Autonomous Thermal Vision Robotic System for Victims Recognition in Search and Rescue Missions</i>	5
2.2.2 <i>Image Processing Technique Applied to Electrical Substations Based on Drones With Thermal Vision for Predictive Maintenance</i>	6
2.2.3 <i>Direct LiDAR Odometry: Fast Localization With Dense Point Clouds</i>	6
2.3 Teori/Konsep Dasar	7
2.3.1 Gardu listrik	7
2.3.2 <i>DeepRobotics Jueying Lite 3 Quadruped Robot</i>	10

2.3.3	<i>Robot Operating System (ROS)</i>	10
2.3.4	<i>Kamera Termal</i>	12
2.3.5	<i>Computer Vision</i>	12
2.3.6	Metode Evaluasi dalam <i>Computer Vision</i>	13
2.3.7	PID Controller	14
2.3.8	<i>Direct LiDAR-Inertial Odometry (DLIO)</i>	15
2.3.9	WebSocket	15
3	METODOLOGI	17
3.1	Metode Penelitian	17
3.1.1	<i>Perencanaan (Planning)</i>	17
3.1.2	<i>Produksi (Production)</i>	17
3.1.3	<i>Evaluasi (Evaluation)</i>	17
3.2	Analisis Sistem	18
3.2.1	Analisis Kebutuhan	18
3.2.2	Analisis Tujuan	18
3.3	Desain Mekanikal	18
3.4	Desain Elektrikal	19
3.5	Arsitektur Perangkat Lunak Robot	20
3.5.1	<i>Hardware Interface Package</i>	21
3.5.2	<i>Perception</i>	21
3.5.3	<i>Overheat pose estimation Node</i>	22
3.5.4	<i>Control</i>	22
3.6	Desain Antarmuka Pengguna (<i>User Interface</i>) Sistem Kontrol	23
3.6.1	<i>Authentication Page</i>	24
3.6.2	<i>Robot Information</i>	24
3.6.3	<i>Patrol Controller</i>	24
3.7	<i>Computer Vision</i>	24
3.7.1	Pembuatan Dataset	24
3.7.2	Pelatihan Model <i>YOLOv8</i>	24

3.7.3	Konversi ke <i>TensorRT</i>	24
3.7.4	Deteksi Suhu Komponen	25
4	HASIL YANG DIHARAPKAN	27
4.1	Hasil yang Diharapkan dari Penelitian	27
4.2	Hasil Pendahuluan	27
4.2.1	Pengujian <i>Localization Node</i>	27
4.2.2	Pengujian Model <i>Computer Vision</i>	28
4.2.3	Pengujian <i>Overheat Detection Node</i>	30
4.2.4	Pengujian <i>Path Following Node</i>	31
4.2.5	Pengujian <i>Obstacle Avoidance Node</i>	31
5	JADWAL PENELITIAN	33
	DAFTAR PUSTAKA	35

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

2.1	<i>Quadruped robot</i> dengan kamera termal untuk deteksi korban	5
2.2	Drone dengan kamera termal untuk pemantauan gardu induk	6
2.3	Implementasi <i>DLO</i> pada <i>quadruped ledggged robot Boston Dynamics Spot</i> . . .	7
2.4	Transformator gardu listrik	7
2.5	Arester gardu listrik	8
2.6	Disconnecter gardu listrik	8
2.7	Pemutus sirkuit gardu listrik	9
2.8	Suhu berdasarkan warna pada kamera thermal	12
2.9	Contoh <i>confusion matrix</i> untuk klasifikasi biner	13
2.10	Arsitektur <i>Direct LiDAR-Inertial Odometry (DLIO)</i>	15
3.1	Diagram alur penelitian	17
3.2	Desain komponen mekanikal robot	19
3.3	Desain <i>block diagram</i> sistem distribusi daya dan komunikasi elektrikal robot .	19
3.4	Desain arsitektur perangkat lunak robot	20
3.5	Representasi error pada <i>WaypointControl</i>	22
4.1	<i>DLIO</i> pada jalan kampus ITS	27
4.2	Perbandingan <i>DLIO</i> dan GPS RTK	28
4.3	<i>Confusion Matrix YOLOv8n 100 Epoch SGD 8 Batch Size</i>	29
4.4	<i>Loss Function YOLOv8n 100 Epoch SGD 8 Batch Size</i>	29
4.5	FPS Model <i>YOLOv8n</i> pada <i>Jetson Nano</i>	30
4.6	Pengujian <i>Overheat Detection</i> pada citra thermal	30
4.7	Pengujian <i>Path Following</i> pada Robot	31
4.8	Pengujian <i>Obstacle Avoidance</i> pada Robot	31

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

4.1	10 Model Teratas Berdasarkan <i>mAP50</i>	28
5.1	Timeline Kegiatan Pengembangan Robot	33

[Halaman ini sengaja dikosongkan]

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Sektor kelistrikan di Indonesia memiliki peranan yang sangat penting dalam mendukung kehidupan masyarakat. Populasi penduduk yang terus meningkat, bersamaan dengan perkembangan industri yang pesat, menyebabkan kebutuhan akan listrik mengalami lonjakan yang signifikan. Menurut Laporan Statistik PLN tahun 2023, konsumsi listrik di Indonesia mencapai 288.435,78 GWh dengan jumlah pelanggan sebanyak 89.153.278. Angka ini mencerminkan tren peningkatan yang konsisten dibandingkan tahun-tahun sebelumnya. Sebagai perbandingan, pada tahun 2022 konsumsi listrik tercatat sebesar 273.761,48 GWh dengan jumlah pelanggan sebanyak 85.636.198, sedangkan pada tahun 2021 mencapai 257.634,25 GWh dengan 82.543.980 pelanggan [1]. Data dari Kementerian Energi dan Sumber Daya Mineral (ESDM) juga menunjukkan bahwa konsumsi listrik per kapita pada tahun 2023 mencapai 1.285 kWh, meningkat dari 1.173 kWh pada tahun 2022, dan diproyeksikan akan terus meningkat hingga mencapai 1.408 kWh pada tahun 2024[2]. Peningkatan konsumsi listrik yang signifikan ini menegaskan perlunya penguatan infrastruktur distribusi tenaga listrik agar dapat memenuhi permintaan yang terus berkembang. Salah satu elemen infrastruktur yang sangat vital dalam sistem distribusi tenaga listrik adalah gardu listrik. Gardu listrik berfungsi sebagai penghubung antara pembangkit listrik dan konsumen. Gardu ini tidak hanya mendistribusikan tenaga listrik, tetapi juga mengatur dan mengontrol aliran listrik, menjaga kualitas daya, serta melindungi sistem dari gangguan.

Salah satu masalah yang sering terjadi pada gardu listrik adalah *overheat* pada komponen kritis di dalamnya. *Overheat* terjadi ketika suhu komponen melebihi batas aman, yang dapat disebabkan oleh berbagai faktor, termasuk beban berlebih, kondisi lingkungan yang ekstrem, dan kurangnya pemeliharaan yang tepat[3]. Ketika suhu komponen meningkat, risiko kerusakan menjadi lebih tinggi, yang dapat mengakibatkan penurunan efisiensi operasional dan potensi pemadaman listrik yang tidak diinginkan[4]. Studi kasus menunjukkan bahwa *overheat* pada transformator dapat menyebabkan kerusakan serius pada isolasi minyak, yang berfungsi untuk mendinginkan dan melindungi komponen internal dari arus listrik. Ketika suhu minyak isolasi meningkat, sifat dielektriknya dapat terdegradasi, yang berpotensi menyebabkan kegagalan isolasi dan kebakaran[5]. Selain itu, *overheat* juga dapat memengaruhi komponen lain dalam gardu listrik, seperti isolator dan *disconnector*. Ketika isolator mengalami *overheat*, material isolasi dapat terdegradasi, sehingga kemampuan menahan tegangan menurun dan meningkatkan risiko terjadinya *arc* atau percikan listrik[6].

Untuk mencegah *overheat* pada komponen gardu listrik, sangat penting untuk melakukan pemeliharaan yang tepat dan pemantauan suhu secara berkala. Dalam konteks ini, penerapan solusi otomatisasi berbasis robotika dapat menjadi alternatif yang efektif untuk meningkatkan efektivitas pemantauan dan pemeliharaan. Salah satu teknologi yang relevan adalah *autonomous mobile robot*, yaitu perangkat yang dirancang untuk melaksanakan berbagai tugas secara otomatis, termasuk tugas yang membutuhkan ketelitian tinggi atau pengawasan di area yang sulit dijangkau manusia. Salah satu jenis robot yang saat ini sedang berkembang pesat adalah robot *quadruped-legged*. Robot jenis ini dilengkapi dengan empat kaki untuk bergerak dengan meniru cara gerak hewan seperti anjing atau kuda. *Quadruped-legged* memiliki keunggulan dibandingkan robot beroda, khususnya dalam hal mobilitas dan kemampuan bermanuver di medan yang tidak rata. Beberapa riset terkait *quadruped-legged robot* untuk pemantauan

garu listrik telah dilakukan, seperti pada proyek *ASUMO* (*Advanced Substation Monitoring*). Proyek ini menunjukkan bahwa penggunaan robot berkaki empat dapat menjadi alternatif yang efektif untuk meningkatkan efisiensi operasional dan menjaga kestabilan pasokan listrik [7]. Berdasarkan hal tersebut, penggunaan robot *quadruped-legged* dapat menjadi solusi yang efektif. Robot ini dapat dilengkapi dengan *thermal camera* untuk mendeteksi suhu panas pada komponen gardu listrik, sehingga memungkinkan deteksi dini terhadap *overheat* dan pencegahan kerusakan yang lebih serius. Oleh karena itu, penelitian lebih lanjut dan pengembangan teknologi robotika dalam konteks pemantauan infrastruktur kelistrikan sangat diperlukan untuk memastikan sistem kelistrikan Indonesia dapat berfungsi secara optimal dan berkelanjutan di masa depan serta mendukung automatisasi dalam pemantauan dan pemeliharaan gardu listrik.

1.2 Permasalahan

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, penelitian ini berfokus pada pengembangan robot otonom untuk pemantauan *overheat* pada komponen gardu listrik. Namun, beberapa permasalahan teknis perlu diselesaikan dalam proses pengembangan ini. Pertama, diperlukan sistem *computer vision* berbasis kamera termal yang mampu mendeteksi dan menganalisis suhu komponen gardu listrik secara akurat untuk mengidentifikasi potensi *overheat*. Kedua, navigasi otonom menjadi tantangan besar karena robot harus mampu bergerak secara mandiri di lingkungan gardu listrik yang dinamis, kompleks, dan terkadang memiliki medan tidak rata. Robot harus dilengkapi dengan sistem navigasi yang dapat menghindari rintangan dan memastikan patroli berjalan dengan aman dan efisien. Ketiga, untuk mendukung proses pemantauan secara *real-time*, dibutuhkan *control station* yang dapat memantau pergerakan robot, mengontrol fungsi robot secara jarak jauh, serta memberikan informasi visual dan estimasi posisi komponen yang mengalami *overheat*.

1.3 Tujuan

Penelitian ini bertujuan untuk mengembangkan sistem pemantauan *overheat* pada komponen gardu listrik menggunakan robot otonom yang dilengkapi dengan kamera termal dan sistem pendukung lainnya. Secara terperinci, tujuan penelitian ini meliputi:

1. Mengembangkan sistem *computer vision* pada robot yang mampu mendeteksi dan mengidentifikasi komponen gardu listrik yang mengalami *overheat* secara akurat.
2. Merancang dan mengimplementasikan sistem navigasi otonom untuk memungkinkan robot bergerak secara mandiri di lingkungan gardu listrik, menghindari rintangan, serta menjalankan tugas patroli secara efisien.
3. Merancang *control station* untuk memantau pergerakan robot, mengontrol operasi robot secara *real-time*, dan menampilkan hasil analisis suhu serta estimasi posisi komponen yang mengalami *overheat*.

1.4 Batasan Masalah

Penelitian ini memiliki batasan yang disebabkan oleh keterbatasan sumber daya, waktu, dan faktor lainnya. Batasan-batasan tersebut dijelaskan sebagai berikut:

1. Robot yang digunakan dalam penelitian ini adalah *DeepRobotics Jueying Lite 3*, yang dilengkapi dengan kamera termal untuk mendeteksi suhu pada komponen gardu listrik.
2. Algoritma *kinematic* yang digunakan untuk pergerakan robot memanfaatkan program bawaan tanpa melakukan pengembangan atau modifikasi algoritma baru.

3. Lingkungan gardu listrik yang menjadi target penelitian adalah gardu induk atau gardu pembangkit jenis *outdoor*, yang berada dalam jangkauan sinyal *BTS provider* untuk mendukung transmisi data kontrol.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat baik secara praktis maupun teoritis yang mendukung pengembangan teknologi robotika dan pemantauan komponen di gardu listrik.

1.5.1 Manfaat Praktis

Hasil penelitian ini diharapkan dapat memberikan solusi otomatis yang aplikatif dalam industri kelistrikan, khususnya dalam sistem pemantauan suhu berlebih (*overheat*) pada komponen gardu listrik.

1.5.2 Manfaat Teoritis

Secara teoritis, penelitian ini berkontribusi pada pengembangan ilmu pengetahuan di bidang robotika, terutama dalam navigasi otonom dan sistem deteksi berbasis computer vision. Selain itu, penelitian ini memperkaya literatur tentang integrasi teknologi IoT dan robotika dalam pemantauan infrastruktur kelistrikan.

[Halaman ini sengaja dikosongkan]

BAB 2 TINJAUAN PUSTAKA

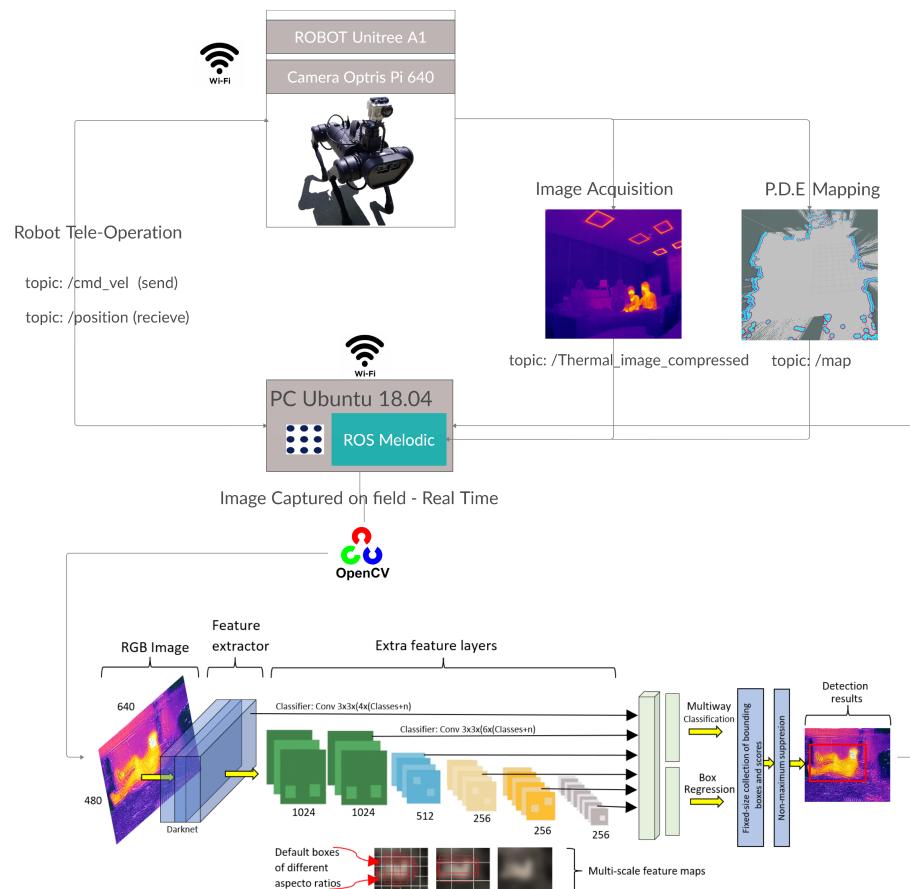
2.1 Hasil penelitian/perancangan terdahulu

Dalam penelitian ini, penulis merujuk pada beberapa studi sebelumnya yang relevan. Penelitian-penelitian tersebut memiliki hubungan dengan topik yang sedang diteliti, sehingga dapat digunakan sebagai dasar untuk penelitian ini.

2.2 Hasil Penelitian/Perancangan Terdahulu

2.2.1 Autonomous Thermal Vision Robotic System for Victims Recognition in Search and Rescue Missions

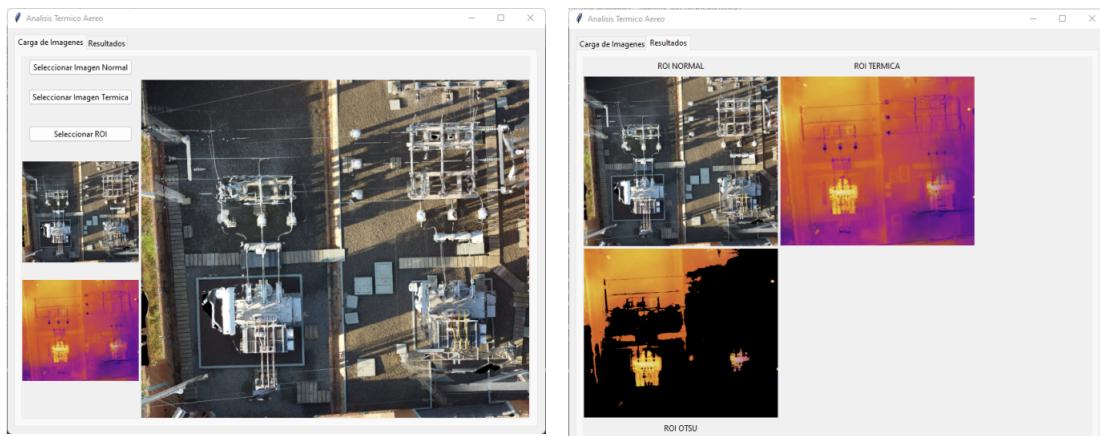
Penelitian oleh Cruz Ulloa mengembangkan robot berkaki empat (*quadruped*) menggunakan *Unitree A1* yang dilengkapi dengan kamera termal *Opitris Pi640* dan *Convolutional Neural Network (CNN)* untuk mendeteksi korban dalam misi pencarian di lingkungan pasca-bencana. Penelitian ini berhasil mencapai akurasi lebih dari 90% dalam kondisi lingkungan sulit, seperti minim cahaya dan puing-puing. Robot ini mampu bergerak secara otonom dan mendeteksi korban dengan cepat di medan yang tidak rata, sehingga dapat membantu tim pencarian dalam menemukan korban yang terperangkap di lokasi bencana[8]. Sistem robot dibangun menggunakan *ROS 1 Melodic*. Penelitian ini memiliki kesamaan dengan topik kami yang juga menggunakan *quadruped* dan kamera termal.



Gambar 2.1: *Quadruped robot* dengan kamera termal untuk deteksi korban

2.2.2 Image Processing Technique Applied to Electrical Substations Based on Drones With Thermal Vision for Predictive Maintenance

Penelitian ini mengusulkan penggunaan VANT (Vehículo Aéreo No Tripulado) atau drone, yang dilengkapi dengan dua jenis kamera: *kamera tradisional* untuk menangkap gambar visual dan *kamera termografik* untuk memperoleh gambar inframerah yang dapat menunjukkan suhu komponen di gardu induk. Drone ini dilengkapi dengan sistem navigasi dan pengendalian yang memungkinkan operasi otonom di sekitar gardu induk. Data gambar yang diambil oleh drone diproses menggunakan teknik *image processing* untuk mengidentifikasi *hot spots* atau titik panas pada komponen gardu induk. Hasil analisis ini dapat digunakan untuk memprediksi potensi kerusakan pada komponen dan mengambil tindakan pencegahan yang diperlukan. Penelitian ini menunjukkan bahwa penggunaan drone dengan kamera termal dapat meningkatkan efisiensi dan akurasi dalam pemantauan gardu induk, serta meminimalkan risiko keselamatan bagi petugas yang harus melakukan inspeksi langsung di lokasi gardu [9]. Penelitian ini memiliki kesamaan dengan topik kami yang juga menggunakan kamera termal untuk pemantauan komponen gardu listrik.



Gambar 2.2: Drone dengan kamera termal untuk pemantauan gardu induk

2.2.3 Direct LiDAR Odometry: Fast Localization With Dense Point Clouds

Penelitian ini berfokus pada pengembangan *Direct LiDAR Odometry* (DLO), sebuah solusi odometri berbasis LiDAR yang *ringan* dan *efisien* untuk robot otonom yang bekerja di lingkungan tanpa sinyal GPS. Sistem ini mengelola informasi peta historis secara efisien dengan cara menyimpan data yang relevan dan mengurangi beban komputasi. DLO memungkinkan penggunaan kembali informasi peta sebelumnya tanpa mengorbankan *kecepatan* dan *akurasi*. DLO menggunakan solver iteratif yang cepat dalam mencocokkan data titik LiDAR, memungkinkan pendaftaran data yang lebih efisien. Teknik ini dipadukan dengan penggunaan kembali struktur data, yang mengurangi waktu pemrosesan dan meningkatkan efisiensi. Sebagai hasilnya, DLO lebih *akurat* dalam menentukan posisi robot dibandingkan dengan metode LiDAR odometry yang ada.

Selain itu, DLO memiliki *beban komputasi yang lebih rendah*, menjadikannya lebih cocok untuk platform robot yang memiliki keterbatasan dalam sumber daya komputasi. Metode ini telah diuji secara ekstensif dalam berbagai lingkungan menantang pada robot udara dan berkaki, termasuk dalam konteks penelitian tim NASA JPL CoSTAR untuk *DARPA Subterranean Challenge* [10]. Penelitian ini memiliki keterkaitan dengan sistem yang sedang dikembangkan, yang menawarkan sistem lokalisasi tanpa hanya menggunakan GPS, namun tetap *akurat* dan *ringan*.

Sistem ini juga diimplementasikan pada robot berkaki (*quadruped robot*), dengan tujuan untuk memberikan solusi navigasi yang efisien dalam lingkungan yang memiliki medan yang tidak rata dan tidak memiliki sinyal GPS yang memadai.



Gambar 2.3: Implementasi *DLO* pada *quadruped ledggged robot Boston Dynamics Spot*

2.3 Teori/Konsep Dasar

Bagian ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam tugas akhir. Pada bab ini akan dijelaskan mengenai teori-teori yang digunakan dalam penelitian ini.

2.3.1 Gardu listrik

Gardu listrik merupakan fasilitas yang berfungsi sebagai titik penghubung antara pembangkit listrik dan jaringan distribusi. Gardu berperan penting dalam mengatur dan mendistribusikan tenaga listrik ke konsumen akhir. Terdapat berbagai jenis gardu listrik beberapa diantaranya adalah gardu induk dan gardu pembangkit. Gardu induk adalah fasilitas yang berfungsi untuk mengubah tegangan tinggi menjadi tegangan rendah, sehingga dapat didistribusikan ke konsumen dengan aman. Di sisi lain, gardu pembangkit adalah fasilitas yang terletak di dekat sumber energi, seperti pembangkit listrik tenaga air atau pembangkit listrik tenaga uap, yang berfungsi untuk mengubah energi dari sumber tersebut menjadi energi listrik dan mendistribusikannya ke jaringan listrik. Gardu dilengkapi dengan berbagai peralatan, seperti transformator, pemutus sirkuit, dan isolator, yang berfungsi untuk menjaga kestabilan dan keandalan pasokan listrik.

2.3.1.1 Transformator

Transformator merupakan komponen esensial dalam sistem kelistrikan, yang berfungsi untuk mengubah tegangan listrik serta memastikan distribusi energi berlangsung secara efisien.



Gambar 2.4: Transformator gardu listrik

Prinsip kerja transformator ini didasarkan pada induksi elektromagnetik, yaitu mengubah tegangan listrik dari tinggi ke rendah atau sebaliknya, sesuai dengan kebutuhan sistem distribusi. Secara umum, transformator gardu induk terdiri dari dua bagian utama, yaitu bagian inti besi dan gulungan kumparan. Selain itu, transformator ini juga dilengkapi dengan komponen-komponen penting seperti *bushing*, *tap changer*, dan *cooling system*, yang berperan penting dalam menjaga kinerja serta keandalan transformator. Suhu operasi normal transformator umumnya berkisar antara 40°C hingga 80°C, tergantung pada kapasitas dan spesifikasi teknis transformator tersebut.[11].

2.3.1.2 Arrester

Arrester, atau *lightning arrester*, adalah perangkat yang melindungi sistem kelistrikan dari lonjakan tegangan akibat petir. *Arrester* berfungsi untuk mengalihkan arus petir ke tanah, sehingga melindungi peralatan listrik dari kerusakan.



Gambar 2.5: Arrester gardu listrik

Penelitian menunjukkan bahwa untuk memastikan efektivitas *arrester*, nilai resistansi pembumian harus rendah, idealnya di bawah 10Ω . Selain itu, pemeliharaan rutin terhadap *arrester* sangat penting untuk memastikan kinerjanya tetap optimal, termasuk pengujian terhadap kondisi lingkungan yang dapat mempengaruhi performanya [12]. Suhu operasi *arrester* biasanya berkisar antara -40°C hingga 60°C. Suhu melebihi batas maksimum dapat menyebabkan kerusakan permanen pada perangkat tersebut [13].

2.3.1.3 Disconnector

Disconnecter, atau pemisah, adalah perangkat yang digunakan untuk memutuskan arus listrik dalam sistem distribusi.



Gambar 2.6: Disconnector gardu listrik

Disconnecter berfungsi untuk memisahkan bagian dari sistem kelistrikan untuk keperluan pemeliharaan atau perbaikan tanpa mempengaruhi bagian lain dari jaringan. Pemeliharaan dan pengujian berkala terhadap *disconnecter* sangat penting untuk memastikan bahwa perangkat ini berfungsi dengan baik saat dibutuhkan. Suhu operasi *disconnecter* biasanya berada dalam rentang 20°C hingga 40°C, dan overheating dapat terjadi jika suhu melebihi 85°C, yang dapat mengakibatkan kegagalan fungsi [14].

2.3.1.4 Busbar

Busbar adalah komponen penting dalam gardu listrik yang berfungsi sebagai penghubung antara berbagai peralatan listrik. *Busbar* memungkinkan distribusi arus listrik yang efisien dan aman di dalam gardu. *Busbar* biasanya terbuat dari bahan konduktif yang baik, seperti tembaga atau aluminium, dan dirancang untuk menampung arus listrik dalam jumlah besar. Pemeliharaan dan pengujian *busbar* secara rutin diperlukan untuk mencegah kerusakan dan memastikan keandalan sistem distribusi listrik. Suhu operasi *busbar* dapat bervariasi, tetapi umumnya tidak boleh melebihi 90°C untuk mencegah overheating yang dapat merusak isolasi dan struktur busbar itu sendiri [15].

2.3.1.5 Isolator

Isolator adalah perangkat yang berfungsi untuk memisahkan bagian dari sistem kelistrikan, sehingga mencegah arus listrik mengalir ke bagian yang tidak diinginkan. *Isolator* digunakan untuk menjaga keamanan dan keandalan sistem kelistrikan, terutama saat pemeliharaan dilakukan. *Isolator* dirancang untuk menahan tegangan tinggi dan memiliki karakteristik dielektrik yang baik. Pemeliharaan *isolator* juga penting untuk memastikan bahwa tidak ada keboceoran arus yang dapat menyebabkan kerusakan pada peralatan lainnya. Suhu operasi *isolator* biasanya berkisar antara -30°C hingga 50°C, dan overheating dapat terjadi jika suhu melebihi 70°C, yang dapat mengakibatkan kerusakan pada material isolasi [16].

2.3.1.6 Pemutus Sirkuit (*Circuit Breaker*)

Pemutus sirkuit adalah perangkat yang berfungsi untuk melindungi sistem kelistrikan dari arus lebih (*overcurrent*) dan hubung singkat (*short circuit*). *Pemutus sirkuit* dapat secara otomatis memutuskan aliran listrik ketika terdeteksi adanya gangguan, sehingga mencegah kerusakan pada peralatan dan menjaga keselamatan sistem. Terdapat berbagai jenis *pemutus sirkuit*, termasuk *pemutus sirkuit otomatis* (*automatic circuit breaker*) dan *pemutus sirkuit manual* (*manual circuit breaker*), yang masing-masing memiliki aplikasi dan karakteristik yang berbeda.



Gambar 2.7: Pemutus sirkuit gardu listrik

Pemeliharaan dan pengujian berkala terhadap *pemutus sirkuit* sangat penting untuk memastikan bahwa perangkat ini berfungsi dengan baik saat dibutuhkan. Suhu operasi *pemutus sirkuit* biasanya berkisar antara -25°C hingga 55°C, dan overheating dapat terjadi jika suhu melebihi 85°C, yang dapat menyebabkan kerusakan pada mekanisme pemutus [17].

2.3.2 DeepRobotics Jueying Lite 3 Quadruped Robot

DeepRobotics Jueying Lite 3 adalah robot *quadruped* yang dikembangkan oleh Deep-Robotics, sebuah perusahaan teknologi robotik di China. Robot ini dirancang untuk berbagai aplikasi, termasuk eksplorasi, pemantauan, dan layanan di berbagai lingkungan. Dengan desain yang ringkas dan kuat, *Jueying Lite 3* mampu bergerak dengan cepat dan stabil di medan yang beragam, menjadikannya ideal untuk aplikasi di luar ruangan. Untuk menjaga stabilitas, *Jueying Lite 3* menerapkan model *3D-SLIP (Three-Dimensional Spring-Loaded Inverted Pendulum)* yang dirancang untuk menjaga keseimbangan saat robot bergerak di medan tidak rata.[18].

Jueying Lite 3 dilengkapi dengan *neural network* yang telah dilatih sebelumnya untuk meningkatkan kinerja gerakan *bounding* dan kontrol umpan balik. Dengan fungsi penghargaan yang mempertimbangkan titik kontak dan fase gerakan, robot mencapai simetri dan periodisitas gerakan yang lebih baik, sehingga meningkatkan efisiensi mobilitas. Estimasi keadaan robot dilakukan melalui metode *Extended Kalman Filter (EKF)* yang dimodifikasi, yang meningkatkan akurasi estimasi di medan tidak stabil, memungkinkan robot mengatasi slip pada kaki dan beradaptasi terhadap berbagai jenis medan. Selain itu untuk kontrol eksternal, *Jueying Lite 3* dapat dikendalikan melalui *ROS URI* dengan mengirimkan topik `cmd_vel`[19].

2.3.3 Robot Operating System (ROS)

Robot Operating System (ROS) adalah *framework* perangkat lunak *open-source* yang dirancang untuk memfasilitasi pengembangan aplikasi robotik. ROS menyediakan berbagai *tools* dan *libraries* untuk mengelola berbagai tugas robotik, seperti kontrol aktuator, pemrosesan data sensor, serta navigasi dan kontrol gerak. Salah satu keunggulan ROS adalah arsitekturnya yang modular, memungkinkan pengembang untuk membagi sistem robotik kompleks menjadi bagian-bagian kecil yang saling terhubung, dikenal sebagai *node*. ROS mendukung berbagai aplikasi robotik, mulai dari robot industri hingga robot otonom. Berikut adalah beberapa komponen dan konsep penting dalam ROS:

2.3.3.1 Node

Node adalah unit dasar dalam ROS yang berfungsi sebagai proses independen untuk menjalankan tugas spesifik dalam sistem robotik. Setiap *node* memiliki fungsionalitas tertentu, seperti mengambil data sensor atau mengontrol aktuator, dan dapat berkomunikasi dengan *node* lainnya melalui berbagai mekanisme komunikasi, seperti *topic*, *service*, atau *action*. *Node* bekerja secara paralel dan saling berinteraksi untuk membentuk aplikasi robotik yang kompleks.

2.3.3.2 Topic

Topic adalah mekanisme komunikasi *publish/subscribe* antara *node* untuk pertukaran data secara asinkron. *Node* yang berfungsi sebagai *publisher* mengirimkan data ke *topic*, sementara *node* yang berfungsi sebagai *subscriber* menerima data dari *topic* tersebut. Data yang dikirim melalui *topic* biasanya berbentuk pesan (*message*) yang memiliki format tertentu yang disepakati oleh pengembang.

2.3.3.3 Service

Service adalah mekanisme komunikasi sinkron antara dua *node*, di mana satu *node* melakukan permintaan (*request*) dan *node* lainnya memberikan respon (*response*) secara langsung. Layanan ini berguna untuk tugas-tugas yang memerlukan hasil segera setelah permintaan dilakukan, seperti pengaturan parameter robot atau pembacaan nilai sensor tertentu.

2.3.3.4 Action

Action digunakan untuk tugas-tugas yang memerlukan umpan balik berkelanjutan dan memungkinkan komunikasi dua arah antara *node*. Berbeda dengan *service*, yang bersifat sinkron dan langsung, *action* mendukung komunikasi asinkron dengan menyediakan umpan

balik selama operasi berlangsung, seperti pada tugas pergerakan robot yang memerlukan estimasi waktu dan status secara dinamis.

2.3.3.5 *Launch File*

Launch File adalah file berformat XML yang digunakan untuk menginisiasi beberapa *node* dan parameter konfigurasi secara bersamaan. *Launch file* memungkinkan pengembang untuk menyusun dan mengelola konfigurasi sistem robotik dengan lebih efisien, menghindari pengaturan manual yang berulang kali. File ini dapat digunakan untuk menentukan urutan *node* yang dijalankan serta parameter yang diperlukan.

2.3.3.6 *Bag File*

Bag File adalah format file yang digunakan untuk merekam dan memutar ulang data *topic*. *Bag file* sangat berguna dalam pengujian dan debugging karena memungkinkan pengembang untuk memeriksa data yang dikumpulkan sebelumnya tanpa perlu menjalankan robot secara langsung. Dengan menggunakan *bag file*, data dapat diputar ulang untuk melakukan analisis atau pengujian lebih lanjut.

2.3.3.7 *Package*

Package adalah unit dasar dalam ROS yang mengorganisir *node*, *libraries*, dan file konfigurasi yang diperlukan oleh aplikasi robotik. Setiap paket dapat berisi kode sumber, file eksekusi, dan file konfigurasi yang mendukung tugas tertentu dalam sistem robot. Paket memungkinkan pengelolaan dan pemeliharaan proyek robotik menjadi lebih terstruktur dan modular.

2.3.3.8 *Parameter*

Parameter adalah variabel dinamis yang disimpan dalam server parameter ROS. *Parameter* ini memungkinkan *node* untuk berbagi informasi konfigurasi secara global dalam sistem. Dengan menggunakan *parameter*, pengembang dapat mengubah konfigurasi robot secara lebih fleksibel tanpa perlu mengubah kode sumber. *Parameter* biasanya digunakan untuk menyimpan nilai-nilai yang digunakan oleh beberapa *node*, seperti kecepatan robot atau batasan sensor.

2.3.3.9 *Workspace*

Workspace adalah direktori kerja dalam ROS tempat pengembang mengelola *package*, *build*, dan konfigurasi pengembangan. *Workspace* umumnya terdiri dari folder seperti *src* untuk menyimpan kode sumber, *devel* untuk hasil *build*, dan *install* untuk menampung hasil instalasi. Dengan menggunakan *workspace*, pengembang dapat mengelola proyek ROS secara terstruktur dan efisien.

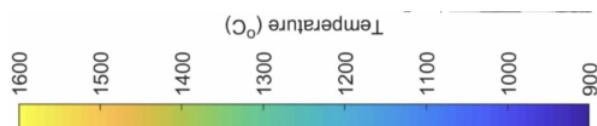
2.3.3.10 *tf*

tf adalah sebuah paket dalam ROS yang digunakan untuk melacak hubungan koordinat antara berbagai frame dalam sistem robotik. *tf* memungkinkan pengembang untuk melakukan transformasi antara berbagai frame koordinat (misalnya, frame robot dan frame dunia) secara efisien, sehingga koordinat objek dan sensor dapat diubah sesuai dengan referensi yang dibutuhkan. Misalnya, dalam robot otomotif, *tf* digunakan untuk mengetahui posisi dan orientasi robot dalam peta atau untuk menghitung posisi objek yang terdeteksi oleh sensor.

Dengan menggunakan *tf*, pengembang dapat menghindari kesalahan dalam kalkulasi posisi dan orientasi, yang sangat penting dalam aplikasi robotik yang melibatkan navigasi dan interaksi dengan lingkungan. Paket *tf* menyediakan berbagai fungsi untuk mengatur dan mengelola transformasi koordinat, seperti `tf::TransformListener` yang digunakan untuk mendengarkan perubahan transformasi, dan `tf::TransformBroadcaster` untuk mempublikasikan transformasi koordinat [20].

2.3.4 Kamera Termal

Kamera termal adalah perangkat yang digunakan untuk mendeteksi distribusi suhu secara non-kontak melalui radiasi inframerah yang dipancarkan oleh objek. Setiap objek dengan suhu di atas nol absolut memancarkan radiasi inframerah, dan kamera termal mengonversi radiasi tersebut menjadi gambar yang menampilkan variasi suhu dalam bentuk skala warna. Citra termal memvisualisasikan suhu dengan menggunakan skala warna, di mana warna biru dan ungu menandakan suhu rendah, hijau dan kuning menunjukkan suhu sedang, serta merah dan putih menunjukkan suhu tinggi atau overheating, seperti yang ditunjukkan pada gambar 2.8.



Gambar 2.8: Suhu berdasarkan warna pada kamera thermal

Penggunaan skala warna ini memudahkan interpretasi visual, terutama dalam aplikasi industri, seperti pemantauan transformator untuk mendeteksi overheating yang dapat menyebabkan kegagalan operasional. Selain itu, kamera termal banyak digunakan dalam pemantauan industri untuk mendeteksi panas berlebih pada peralatan listrik, serta dalam bidang medis dan konstruksi untuk deteksi masalah suhu atau kelembaban[21].

2.3.5 Computer Vision

Computer vision merupakan cabang dari kecerdasan buatan yang memungkinkan komputer memahami dan menganalisis informasi yang diperoleh dari gambar atau video digital. Berbeda dengan manusia yang secara alami ”belajar” untuk menginterpretasikan konteks, jarak objek, serta mendeteksi apakah objek tersebut bergerak atau diam, *computer vision* membutuhkan pemrograman yang kompleks untuk meniru kemampuan tersebut. Salah satu kemajuan terbesar dalam bidang *computer vision* adalah pengembangan arsitektur *Convolutional Neural Network* (CNN), yang menjadi dasar untuk berbagai aplikasi seperti klasifikasi citra, segmentasi, hingga deteksi objek. Salah satu algoritma yang dibangun berdasarkan CNN adalah YOLOv8. YOLOv8 (You Only Look Once versi 8) merupakan algoritma deteksi objek yang terkenal dengan efisiensinya dalam melakukan deteksi objek secara real-time. Algoritma ini dirancang untuk mendeteksi objek hanya dengan satu kali melihat gambar secara menyeluruh, tanpa perlu membaginya menjadi beberapa bagian, yang menjadikannya lebih cepat dibandingkan metode deteksi objek lainnya.

YOLOv8 membawa sejumlah peningkatan signifikan dibandingkan pendahulunya, seperti optimasi dalam deteksi multi-skala, yang memungkinkan algoritma ini mengenali objek dengan berbagai ukuran dalam satu citra. Selain itu, YOLOv8 juga dilengkapi dengan model yang lebih ringan dan efisien, sehingga dapat diterapkan pada perangkat dengan kemampuan komputasi terbatas. Model ini juga menggunakan *backbone* yang lebih efisien dan *neck* yang ditingkatkan untuk meningkatkan ekstraksi fitur, memungkinkan deteksi yang lebih akurat dan cepat pada berbagai jenis objek dan skenario aplikasi. Dengan kemampuan *deep learning* yang lebih maju, YOLOv8 tidak hanya mampu mendeteksi objek secara cepat, tetapi juga memberikan hasil yang lebih akurat. Hal ini membuatnya sangat cocok untuk berbagai aplikasi yang membutuhkan deteksi objek real-time, seperti sistem pemantauan, kendaraan otonom, dan pemrosesan video untuk keamanan [22].

2.3.6 Metode Evaluasi dalam Computer Vision

Dalam bidang *computer vision*, evaluasi kinerja model sangat penting untuk menilai kemampuan sistem dalam mengidentifikasi dan mengklasifikasikan objek. Beberapa metrik yang sering digunakan untuk mengevaluasi performa model *computer vision* antara lain adalah *confusion matrix*, *precision*, *recall*, *accuracy*, *F1 score*, dan *loss function*. Metrik-metrik ini memberikan informasi yang lebih mendalam mengenai kinerja model selain hanya menggunakan tingkat akurasi.

2.3.6.1 Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mendeskripsikan kinerja model klasifikasi. Matriks ini mencatat jumlah prediksi yang benar dan salah, baik dalam kategori positif maupun negatif. Sebuah matriks konfusi untuk klasifikasi biner dapat digambarkan sebagai berikut:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Gambar 2.9: Contoh *confusion matrix* untuk klasifikasi biner

2.3.6.2 Precision, Recall, dan Accuracy

Precision dan *recall* adalah dua metrik penting dalam evaluasi model klasifikasi. *Precision* mengukur seberapa akurat prediksi positif yang dilakukan oleh model, sedangkan *recall* mengukur seberapa banyak contoh positif yang berhasil ditemukan oleh model.

Precision dihitung sebagai:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.1)$$

Recall dihitung sebagai:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.2)$$

Accuracy mengukur seberapa banyak prediksi yang benar dilakukan oleh model, dan dihitung sebagai:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.3)$$

2.3.6.3 F1 Score

F1 score adalah metrik yang merupakan rata-rata harmonis dari *precision* dan *recall*. Metrik ini memberikan gambaran yang lebih seimbang mengenai kinerja model, terutama ketika terdapat ketidakseimbangan antara kelas-kelas dalam data.

F1 score dihitung sebagai:

$$F_1 = 2 \times \frac{P \times R}{P + R}. \quad (2.4)$$

2.3.6.4 Loss Function

Loss function adalah metrik yang digunakan untuk menghitung seberapa jauh prediksi model dari nilai yang sebenarnya. Dalam *computer vision*, loss function sangat penting untuk mengoptimalkan model selama pelatihan. Salah satu loss function yang umum digunakan dalam masalah klasifikasi adalah *cross-entropy loss*, yang menghitung perbedaan antara distribusi probabilitas yang diprediksi dan distribusi probabilitas yang sebenarnya.

Dalam klasifikasi multi-kelas, *cross-entropy loss* dihitung sebagai:

$$L = - \sum_{i=1}^C y_i \log(p_i) \quad (2.5)$$

Di mana C adalah jumlah kelas, y_i adalah label asli untuk kelas ke- i , dan p_i adalah probabilitas yang diprediksi untuk kelas ke- i .

2.3.6.5 IMU dan Odometry Fusion dengan Extended Kalman Filter (EKF)

Pada tahap prediksi, model dinamik digunakan untuk memprediksi keadaan berdasarkan kontrol yang diterima. Misalkan vektor keadaan $\hat{x}_{k|k-1}$ pada waktu $k-1$ dan kontrol u_{k-1} :

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_{k-1}) \quad (2.6)$$

Di mana $f(\hat{x}_{k-1}, u_{k-1})$ adalah fungsi non-linear yang menggambarkan evolusi keadaan sistem. Pengukuran dari IMU dan odometry digunakan untuk memperbarui estimasi keadaan. Kalman gain dihitung dengan:

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k \right)^{-1} \quad (2.7)$$

Estimasi posisi dan orientasi diperbarui dengan:

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1})) \quad (2.8)$$

Kovarians pembaruan dihitung dengan:

$$P_k = (I - K_k H_k) P_{k|k-1} \quad (2.9)$$

Fusion data IMU dan odometry dengan EKF meningkatkan akurasi estimasi posisi dan orientasi dengan mengurangi ketidakpastian pada masing-masing sumber data [23].

2.3.7 PID Controller

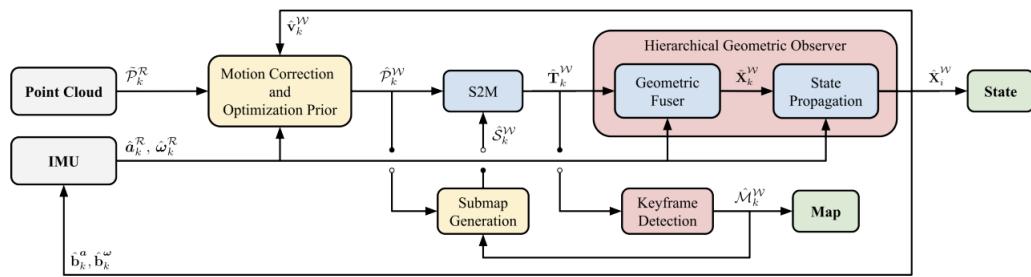
PID controller merupakan salah satu metode kontrol yang paling sering digunakan dalam berbagai aplikasi, karena memiliki struktur yang sederhana dengan hanya tiga parameter utama. Hal tersebut membuat PID controller lebih mudah dioperasikan dan dipahami oleh teknisi dibandingkan dengan metode kontrol lain yang lebih kompleks. PID controller terdiri dari tiga komponen utama, yaitu aksi *proportional*, *integral*, dan *derivative*. Ketiga komponen ini saling melengkapi dalam memperbaiki *error* pada sistem kontrol. *Proportional* memberikan koreksi berdasarkan *error* saat ini, aksi *integral* memperhitungkan *error* masa lalu untuk mengeliminasi offset, dan aksi *derivative* memperkirakan *trend error* untuk respons prediktif.

$$u(t) = u_P(t) + u_I(t) + u_D(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t), \quad (2.10)$$

di mana K_p , K_i , dan K_d adalah koefisien *gain* yang terkait dengan aksi *proportional*, *integral*, dan *derivative* masing-masing.

2.3.8 Direct LiDAR-Inertial Odometry (DLIO)

Algoritma *Direct LiDAR-Inertial Odometry* (DLIO) menawarkan solusi dengan pendekatan *coarse-to-fine* untuk menghasilkan koreksi gerakan waktu kontinu secara lebih akurat. DLIO memanfaatkan persamaan analitik yang dirancang untuk memperbaiki setiap titik data secara paralel dan efisien, sekaligus mengintegrasikan data LiDAR dan *Inertial Measurement Unit* (IMU) secara ketat untuk menghasilkan estimasi keadaan robot yang lebih akurat. Pendekatan ini memungkinkan koreksi distorsi pada *point cloud* secara *real-time*, bahkan dalam kondisi pergerakan dinamis yang kompleks [24].



Gambar 2.10: Arsitektur *Direct LiDAR-Inertial Odometry* (DLIO)

Keunggulan DLIO tidak hanya terletak pada akurasi tinggi dalam estimasi posisi, tetapi juga pada efisiensi komputasi yang lebih baik dibandingkan algoritma-algoritma terkini lainnya seperti LIO-SAM dan FAST-LIO2. DLIO menggabungkan koreksi gerakan dan pembuatan *prior* dalam satu langkah, menghilangkan kebutuhan akan *scan-to-scan matching* yang biasanya memakan waktu. Melalui eksperimen pada *dataset* publik dan *dataset* lapangan yang dikumpulkan secara mandiri, DLIO menunjukkan keunggulan dalam menghasilkan peta 3D yang detail dengan kesalahan posisi minimum. Sistem ini menjadi alternatif yang menjanjikan untuk platform robot bergerak, termasuk drone, dalam navigasi dan pemetaan di lingkungan yang tidak terstruktur.

2.3.9 WebSocket

WebSocket adalah protokol komunikasi dua arah yang beroperasi di atas koneksi TCP yang persisten, memungkinkan pertukaran pesan antara klien dan server secara efisien. Protokol ini dirancang untuk mengatasi kekurangan teknologi komunikasi dua arah yang ada sebelumnya, yang sering menggunakan *HTTP* sebagai lapisan transportasi. Dengan memanfaatkan mekanisme *upgrade* dari *HTTP*, WebSocket dapat membuka saluran komunikasi bidirectional yang lebih efisien dibandingkan dengan pendekatan tradisional seperti *polling* atau *long polling* [25].

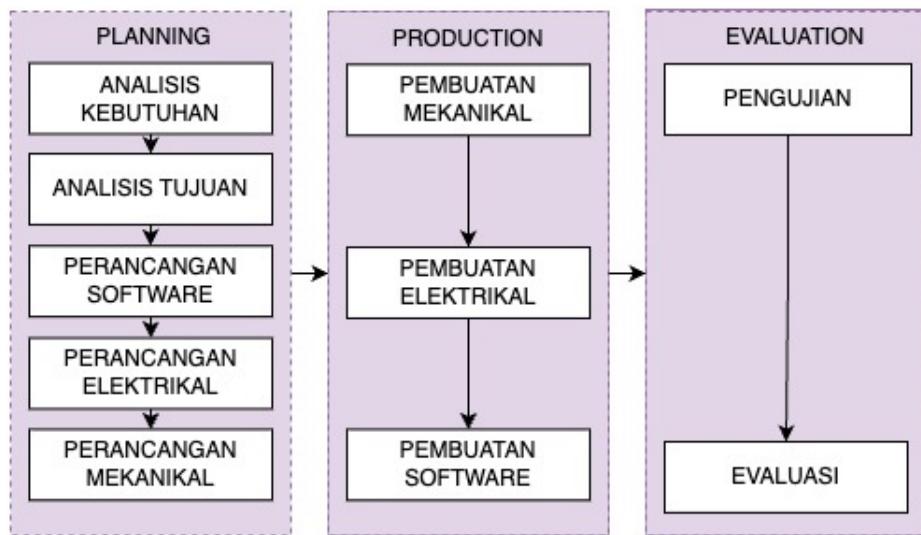
Keunggulan utama dari WebSocket terletak pada kemampuannya untuk mengurangi overhead komunikasi, yang sangat penting dalam aplikasi yang memerlukan pembaruan real-time, seperti aplikasi *chat*, permainan daring, dan sistem *IoT*. Penelitian menunjukkan bahwa WebSocket tidak hanya meningkatkan efisiensi komunikasi, tetapi juga memungkinkan pengembangan aplikasi kolaboratif yang lebih responsif dan interaktif [26]. Dengan demikian, WebSocket menjadi solusi yang ideal untuk aplikasi yang membutuhkan latensi rendah dan kecepatan tinggi dalam pertukaran data.

[Halaman ini sengaja dikosongkan]

BAB 3 METODOLOGI

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah penelitian pengembangan dengan model pengembangan Richey dan Klein. Model ini terdiri dari tiga tahapan, yaitu *Planning*, *Production*, dan *Evaluation* [27]. Setiap tahapan memiliki langkah-langkah spesifik yang harus diikuti untuk mencapai tujuan penelitian seperti pada diagram dibawah.



Gambar 3.1: Diagram alur penelitian

3.1.1 Perencanaan (*Planning*)

Pada tahap ini, peneliti melakukan analisis dan perancangan prototipe. Analisis kebutuhan dilakukan untuk mengidentifikasi fitur dan spesifikasi yang diperlukan. Selain itu, analisis tujuan dilakukan untuk menentukan sasaran yang ingin dicapai dengan teknologi yang akan diterapkan. Berdasarkan hasil analisis kebutuhan dan tujuan, dilakukan perancangan prototipe yang mencakup perancangan perangkat lunak (*software*), perancangan elektrikal, dan perancangan mekanikal.

3.1.2 Produksi (*Production*)

Tahap produksi mencakup pengembangan robot berdasarkan rencana yang telah dibuat. Pengembangan ini meliputi pembuatan komponen mekanikal, elektrikal, dan perangkat lunak (*software*) sesuai dengan desain yang telah ditetapkan. Setiap komponen robot dikembangkan secara terintegrasi untuk memastikan kesesuaian dan kinerja sistem secara keseluruhan.

3.1.3 Evaluasi (*Evaluation*)

Pada tahap evaluasi, dilakukan pengujian secara bertahap untuk memastikan bahwa robot dapat beroperasi sesuai dengan tujuan yang telah ditetapkan. Pertama, dilakukan pengujian sub-sistem untuk memastikan setiap komponen sistem berfungsi dengan baik. Selanjutnya, pengujian terhadap fungsi-fungsi utama robot, seperti navigasi, deteksi *overheat*, dan pengendalian jarak jauh, dilakukan untuk memastikan performa sistem secara keseluruhan. Terakhir, dilakukan pengujian di lingkungan nyata untuk menilai kinerja robot dalam kondisi operasional sesungguhnya. Jika ditemukan kekurangan atau masalah, perbaikan akan dilakukan untuk memastikan kinerja robot optimal dan sesuai dengan tujuan penelitian.

3.2 Analisis Sistem

Analisis dibagi menjadi analisis kebutuhan dan analisis tujuan seperti penjelasan berikut.

3.2.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan melalui pengumpulan data dari studi literatur dan wawancara dengan pihak-pihak terkait. Berdasarkan hasil analisis, beberapa kebutuhan sistem yang harus dipenuhi adalah sebagai berikut:

1. Sistem harus mampu mengidentifikasi komponen yang mengalami *overheat* pada gardu listrik dengan menggunakan kamera termal, serta memberikan estimasi posisi dan jenis komponen yang terdeteksi.
2. Sistem harus mampu melakukan patroli secara otonom di area gardu listrik, mengikuti jalur inspeksi yang telah ditentukan, dan menghindari rintangan fisik yang terdapat pada sekitar robot.
3. Sistem harus dapat berkomunikasi secara *online* dengan *control station* untuk mengirimkan data posisi dan informasi deteksi robot secara real-time, serta menerima perintah kontrol seperti penetapan rute, jadwal patroli, serta perintah untuk melakukan *wireless emergency stop* apabila terjadi keadaan darurat.

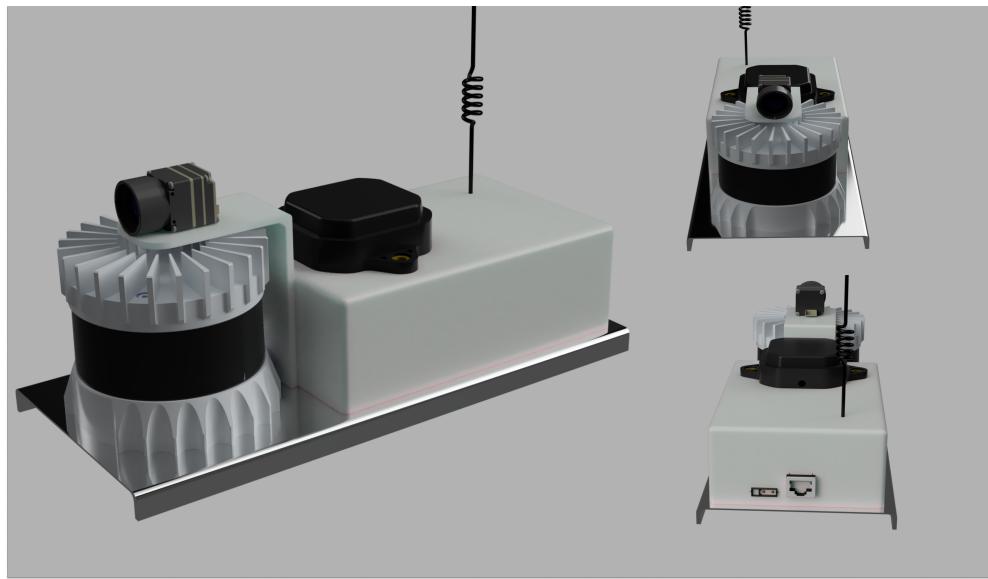
3.2.2 Analisis Tujuan

Berdasarkan hasil analisis kebutuhan adapun tujuan yang ingin dicapai dalam pengembangan sistem ini adalah sebagai berikut:

1. Mengembangkan sistem *computer vision* untuk mendeteksi dan mengidentifikasi komponen yang mengalami *overheat* menggunakan kamera termal. Sistem ini juga harus mampu memperkirakan posisi dan jenis komponen yang terdeteksi mengalami *overheat*.
2. Merancang dan mengimplementasikan sistem navigasi otonom pada robot, sehingga robot dapat melakukan patroli secara mandiri di area gardu listrik, menghindari rintangan fisik, dan mengikuti jalur inspeksi yang telah ditentukan. Sistem navigasi ini harus beroperasi dengan stabil dan akurat di lingkungan yang memiliki gangguan elektromagnetik serta interferensi dari peralatan listrik.
3. Merancang dan membangun *control station* yang dapat memantau lokasi robot dan hasil deteksi *overheat* secara *real-time*. Sistem ini harus dapat mengirim perintah kontrol *emergency stop* secara *wireless* ke robot. Selain itu, sistem harus dapat beroperasi secara online dan dapat diakses dari jarak jauh.

3.3 Desain Mekanikal

Penelitian ini menggunakan robot *quadruped legged* tipe *DeepRobotics Jueying Lite 3 Pro* sebagai basis robot. Modifikasi dilakukan dengan menambahkan plat aluminium pada bagian atas robot sebagai tempat peletakan komponen. Plat aluminium ini dilapisi dengan cat isolator untuk mencegah terjadinya konduksi listrik, mengingat robot akan beroperasi di area gardu listrik. Komponen tambahan yang dipasang meliputi kamera termal dan sensor LiDAR yang diletakkan di bagian depan robot, serta modul GPS, *switch*, dan modem yang ditempatkan di dalam *electrical box*. *Electrical box* ini dirancang menggunakan teknologi *3D printing* sebagai pelindung komponen elektronik dari benturan dan kerusakan akibat lingkungan eksternal. Desain mekanikal robot dirancang sedemikian rupa agar komponen tambahan dapat terpasang dengan aman tanpa mengganggu pergerakan robot. Seperti pada gambar 3.2,

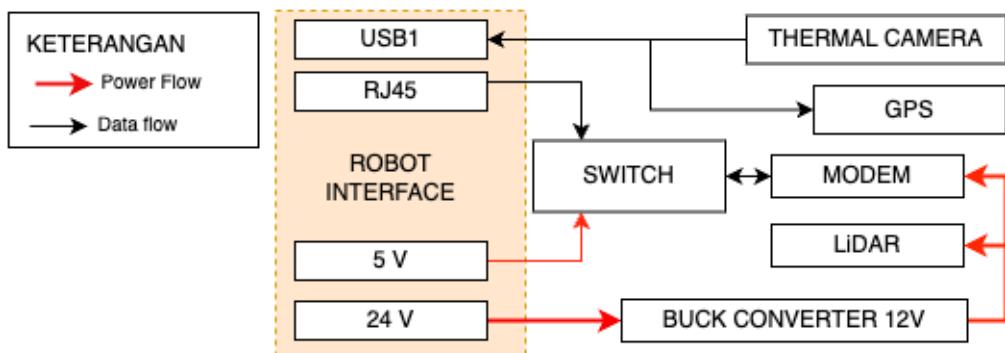


Gambar 3.2: Desain komponen mekanikal robot

Penempatan sensor LiDAR di bagian atas robot memungkinkan sensor untuk memperoleh pandangan yang lebih luas, sedangkan kamera termal ditempatkan di posisi strategis untuk mendeteksi perubahan suhu pada komponen gardu listrik. *Electrical box* dipasang di bagian belakang robot untuk mengoptimalkan distribusi berat dan memastikan kestabilan robot.

3.4 Desain Elektrikal

Desain elektrikal robot mencakup perancangan sistem distribusi daya dan komunikasi yang bertujuan untuk mengintegrasikan komponen-komponen tambahan ke dalam sistem robot yang sudah ada. Komponen-komponen tambahan tersebut dirancang untuk mendukung kemampuan operasi otonom robot. Desain elektrikal lengkap dari robot ini dapat dilihat pada Gambar 3.3.

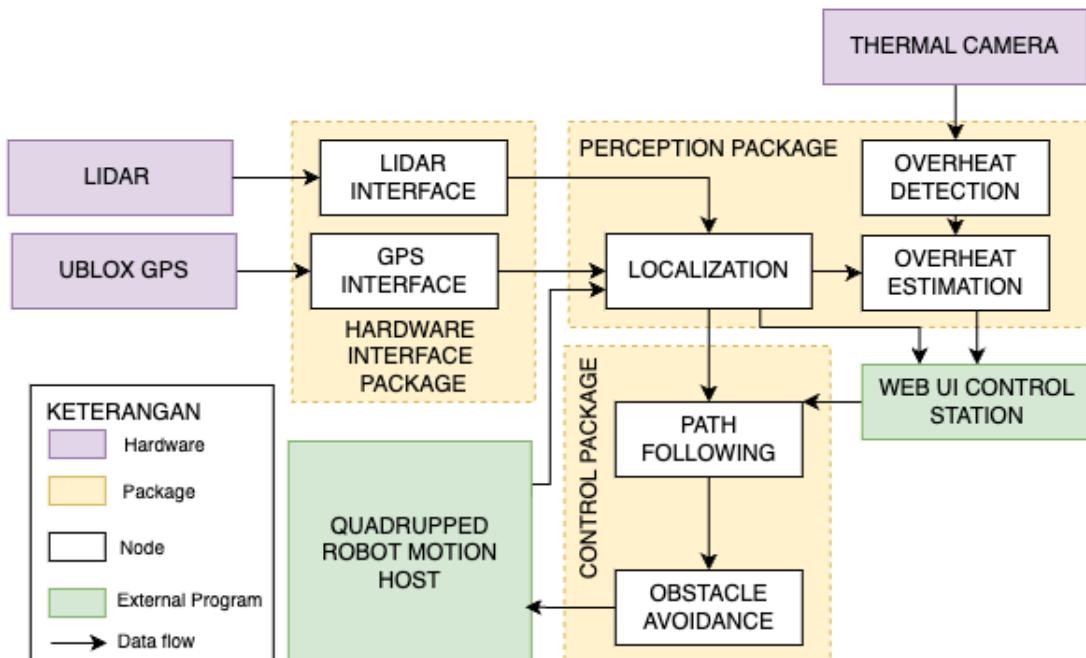


Gambar 3.3: Desain *block diagram* sistem distribusi daya dan komunikasi elektrikal robot

Dalam hal distribusi daya, tegangan 24V yang disuplai oleh robot diturunkan menjadi 12V menggunakan modul *buck converter* untuk memenuhi kebutuhan daya LiDAR dan modem. Komponen lain yang memerlukan tegangan 5V memperoleh catu daya dari sumber 5V yang tersedia pada robot. Selain itu, terdapat modul GPS dan *thermal camera* yang terhubung melalui port USB. Sementara itu, LiDAR dan modem terhubung melalui switch dan kemudian masuk ke dalam sistem robot melalui LAN. Sistem komunikasi menggunakan protokol *wireless* melalui modem yang terhubung ke jaringan *BTS provider* untuk mengirimkan data posisi dan informasi deteksi *overheat* ke *control station*.

3.5 Arsitektur Perangkat Lunak Robot

Arsitektur perangkat lunak robot menggunakan *Robot Operating System (ROS) 1 Noetic*, ROS berfungsi sebagai *middleware* untuk mendukung pengembangan program secara modular. ROS memfasilitasi pengembangan dengan menggunakan *package* dan *node* yang saling berinteraksi, sehingga memungkinkan pengembangan yang efisien dan terstruktur. Pemilihan ROS 1 Noetic didasarkan pada kompatibilitasnya dengan robot *DeepRobotics Jueying Lite 3*. Integrasi ini memungkinkan sistem navigasi yang dirancang untuk memerintahkan robot bergerak pada sumbu translasi *x*, *y*, dan sumbu rotasi *z* (*angular z*) dengan nilai kecepatan tertentu tanpa perlu mengembangkan ulang program *kinematic* atau kontrol pergerakan robot yang sudah ditangani oleh program bawaan *DeepRobotics Jueying Lite 3*. Arsitektur perangkat lunak pada robot dapat dibagi dibagi menjadi tiga *package*, yaitu *hardware interface*, dan *control*, *perception*. Desain arsitektur perangkat lunak robot dapat dilihat pada Gambar 3.4.



Gambar 3.4: Desain arsitektur perangkat lunak robot

Berdasarkan gambar diatas dapat dilihat *HARDWARE INTERFACE* akan memberikan data GPS, LiDAR, dan IMU yang diperoleh dari sensor GPS dan LiDAR ke *PERCEPTION*. *PERCEPTION* akan menggabungkan data dari *HARDWARE INTERFACE* dengan *odometry* yang diperoleh dari *MOTION HOST* untuk mendeteksi posisi dan orientasi robot dalam peta. Selain itu, *PERCEPTION* juga akan memproses data dari kamera termal untuk mendeteksi adanya *overheat* pada komponen gardu listrik serta memperkirakan posisi komponen yang mengalami *overheat*. Hasil deteksi dari *PERCEPTION* akan dikirim ke *CONTROL STATION* melalui *websocket*, dan *PERCEPTION* juga akan melakukan *broadcast* posisi robot (TF) dan marker LiDAR ke map. Disisi lain *CONTROL STATION* akan menerima data posisi dan lokasi dari *PERCEPTION* serta rute yang dikirim oleh *CONTROL STATION*. Ketika perintah *start* diterima dari *CONTROL STATION*, robot akan merencanakan trajektori untuk melakukan patroli sambil menghindari *obstacle* yang ada pada rute. Data trajektori kemudian akan dikirim ke *MOTION HOST* untuk menggerakkan robot.

3.5.1 Hardware Interface Package

Hardware interface mengelola interaksi dengan sensor. *Package* ini terdiri dari dua *node* dengan tugas masing-masing. Node pertama, *GPS Interface*, membaca data dari sensor GPS dan menjalankan *Real Time Kinematic* (RTK) untuk menghubungkan sensor GPS robot dengan stasiun basis, memberikan koreksi sinyal satelit dan memungkinkan pengukuran posisi yang lebih presisi. Node kedua, *LiDAR Interface*, membaca data sensor LiDAR dan mengirimkan *point cloud* ke *perception*, yang merepresentasikan objek di sekitar robot.

3.5.2 Perception

Perception adalah *package* yang berfungsi untuk memproses data sensor guna memungkinkan robot memahami lingkungan sekitarnya. *Package* ini terdiri dari beberapa *node* utama, yang masing-masing memiliki fungsi tertentu diantaranya adalah sebagai berikut.

3.5.2.1 Localization Node

Dalam penelitian ini, robot menggunakan *predefined topological map*, yang diambil dari Google Maps atau aplikasi lainnya. Posisi awal robot atau *initial state* ditentukan pada titik tetap (*fixed position*) dalam peta topologi yang telah ditentukan sebelumnya. Dalam menentukan orientasi, sistem ini menggunakan data *IMU*, *GPS*, dan *LiDAR* yang didapat dari *HARDWARE INTERFACE* untuk menghitung orientasi robot dalam peta topologi. Untuk menentukan posisi robot, digunakan dua metode utama, yaitu *GPS* dan *Direct LiDAR Inertial Odometry* (DLIO). Metode *GPS* digunakan untuk menentukan posisi robot dalam koordinat geografis. Namun, di area yang dipenuhi dengan interferensi elektromagnetik, seperti gardu listrik, sinyal *GPS* dapat terpengaruh, sehingga menurunkan kualitas akurasi di beberapa kondisi. Untuk mengatasi hal ini, jika sinyal *GPS* terdeteksi berada di bawah ambang batas akurasi yang dibutuhkan, sistem secara otomatis beralih menggunakan metode *Direct LiDAR Inertial Odometry* (DLIO) seperti pada Algoritma 1.

Algorithm 1 Seleksi Posisi dan Orientasi Robot

```
1: Input: IMU data, LiDAR data, GPS data, Odometry data
2: Output: Robot position and orientation
3: GPS_Accuracy  $\leftarrow$  GPS.status.service
4: EKF_Fusion  $\leftarrow$  Run Extended Kalman Filter (IMU + Odometry)
5: Fused_IMU  $\leftarrow$  EKF_Fusion.output
6: DLIO_Pos, DLIO_Orientation  $\leftarrow$  Run DLIO (LiDAR + Fused_IMU)
7: if GPS_Accuracy  $\geq$  Threshold then
8:     Position  $\leftarrow$  Position_GPS
9:     Orientation  $\leftarrow$  Orientation_IMU
10: else
11:     Position  $\leftarrow$  Position_DLIO
12:     Orientation  $\leftarrow$  Orientation_DLIO
13: end if
```

DLIO menggabungkan informasi dari sensor *LiDAR* dan *IMU* (*Inertial Measurement Unit*) yang telah digabungkan dengan odometri dengan menggunakan *Extended Kalman Filter* (EKF) untuk menghitung posisi dan orientasi robot.

3.5.2.2 Overheat Detection Node

Node ini berfungsi untuk mendeteksi komponen yang mengalami suhu tinggi (*overheat*) menggunakan kamera termal yang terpasang pada robot. Citra termal yang diperoleh diklasifikasikan menggunakan *YOLOv8* untuk mendeteksi komponen gardu listrik. Setelah kompo-

nen terdeteksi, analisis suhu dilakukan pada bounding box menggunakan teknik kontur dengan threshold warna pada ruang warna HSV (*Hue, Saturation, Value*). Thresholding dilakukan pada saluran *HSV* untuk mengidentifikasi suhu tinggi, dan kontur diterapkan untuk memetakan area yang menunjukkan suhu berlebih. Jika suhu komponen melebihi ambang batas, komponen tersebut diklasifikasikan sebagai *overheat*.

3.5.3 Overheat pose estimation Node

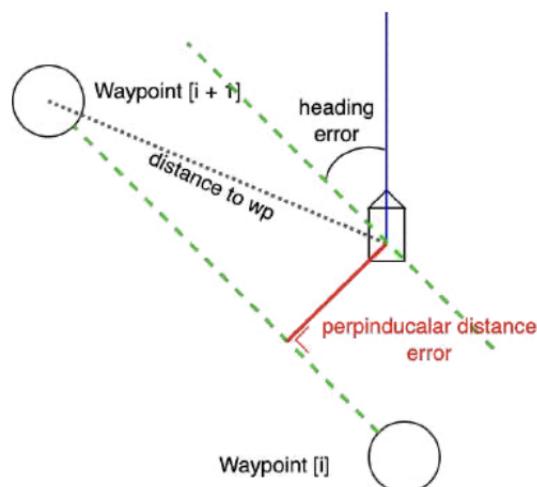
Node ini berfungsi untuk mendeteksi posisi komponen yang mengalami *overheat*. Setelah komponen yang mengalami *overheat* terdeteksi, *node* ini akan memperkirakan posisi komponen tersebut dalam peta topologi. Posisi komponen diestimasi berdasarkan posisi robot yang diperoleh dari *Localization Node* dan data LiDAR yang diperoleh dari *HARDWARE INTERFACE*. Data LiDAR digunakan untuk memetakan area sekitar robot dan mengidentifikasi posisi relatif komponen yang mengalami *overheat* terhadap posisi robot. Hasil estimasi posisi kemudian dikirim ke *CONTROL STATION* untuk ditampilkan pada *graphical user interface (GUI)*.

3.5.4 Control

Control adalah *package* yang berfungsi untuk mengelola pergerakan robot. *Package* ini terdiri dari beberapa *node* utama, diantaranya adalah sebagai berikut.

3.5.4.1 Patrol Control Node

Node ini berfungsi sebagai otak dari robot yang bertanggung jawab untuk mengatur pergerakan robot berdasarkan perintah yang diterima dari *CONTROL STATION*. Perintah yang diterima meliputi perintah untuk memulai patroli, menghentikan patroli, dan mengubah rute patroli. *Patrol Control Node* akan merencanakan trajektori robot berdasarkan rute yang diterima berupa *waypoint* dan mengirimkan perintah ke *MOTION HOST* untuk menggerakkan robot. Dalam mengikuti *waypoint*, robot akan menggunakan algoritma *path following* yang dibuat oleh Akbar 2022. Algoritma ini menggunakan posisi dan orientasi dari *Localization Node* sebagai referensi kontrol. Pada kontrol ini terdapat *WaypointControl*, yang berperan untuk mengarahkan robot menuju titik yang ditentukan. Adapun rumus error yang digunakan dapat dilihat pada Persamaan 3.1. Representasi error yang dimaksud dijelaskan melalui Gambar 3.5 [28].



Gambar 3.5: Representasi error pada *WaypointControl*

$$\text{error} = \text{heading}_{\text{error}} + \text{perpendicular distance}_{\text{error}}. \quad (3.1)$$

Adapun kondisi dimana waypoint telah tercapai adalah apabila kondisi berikut ini terpenuhi:

$$\text{distance to wp} < \text{threshold distance}. \quad (3.2)$$

$$\text{angle to wp} > \text{threshold angle}. \quad (3.3)$$

Error yang didapat kemudian di proses dengan menggunakan *PID controller* untuk mendapatkan output pergerakan robot yang akan dikirim ke *MOTION HOST*. Secara detail algoritma path following dapat dijabarkan sebagai berikut:

Algorithm 2 Algoritma Path Following Robot

```

1: Input: Current position (x, y, yaw), Path (array of poses: x, y)
2: Output: Velocity in x-direction ( $v_x$ ) and Angular velocity ( $\omega_z$ )
3: while not reached final waypoint do
4:    $wp \leftarrow path[path\_idx]$ 
5:    $prev\_wp \leftarrow path[path\_idx - 1]$ 
6:   Calculate Perpendicular Distance Error:
7:    $y_2 - y_1 \leftarrow wp.y - prev\_wp.y$ 
8:    $x_2 - x_1 \leftarrow wp.x - prev\_wp.x$ 
9:    $perpendicular\_distance \leftarrow \frac{|(y_2 - y_1) \cdot current\_pose.x - (x_2 - x_1) \cdot current\_pose.y + wp.x \cdot prev\_wp.y - wp.y \cdot prev\_wp.x|}{\sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}}$ 
10:  Calculate Angle Error:
11:   $angle\_error \leftarrow atan2(wp.y - prev\_wp.y, wp.x - prev\_wp.x) - current\_pose.yaw$ 
12:   $angle\_error \leftarrow (angle\_error + \pi) \bmod (2\pi) - \pi$             $\triangleright$  Normalize angle to  $[-\pi, \pi]$ 
13:  if  $perpendicular\_distance < threshold\_distance$  and  $|angle\_error| < threshold\_angle$ 
then
14:     $path\_idx \leftarrow path\_idx + 1$ 
15:  else
16:     $v_x \leftarrow PID_v(perpendicular\_distance)$ 
17:     $\omega_z \leftarrow PID_\omega(angle\_error)$ 
18:  end if
19: end while

```

3.5.4.2 *Obstacle Avoidance Node*

Obstacle Avoidance Node bertugas untuk mendeteksi obyek pada rute dan menghindari tabrakan saat mengikuti *waypoint*. Robot menggunakan data LiDAR dari *HARDWARE INTERFACE*, yang diproses dengan algoritma *Braitenberg* untuk mendeteksi obyek. Algoritma ini menghasilkan nilai *gain* yang mengatur kecepatan robot dalam menghindari obyek. Jika obyek terdeteksi, robot akan mengurangi kecepatan atau mengubah arah. Setelah obyek tidak terdeteksi, robot akan kembali ke rute dan melanjutkan pergerakan menuju *waypoint* berikutnya.

3.6 Desain Antarmuka Pengguna (*User Interface*) Sistem Kontrol

Sistem ini menggunakan antarmuka berbasis web GUI yang dibangun dengan *Next.js* ¹⁴ dan *roslibjs* untuk integrasi dengan ROS. Sistem akan di-deploy pada internet menggunakan *Node.js*. *PostgreSQL* digunakan sebagai database, sementara *Prisma* berfungsi sebagai *ORM* untuk mengelola data. Sistem GUI dapat dibagi menjadi tiga bagian utama yang memiliki fungsi sebagai berikut:

3.6.1 Authentication Page

Authentication Page adalah halaman yang ditampilkan kepada pengguna yang ingin mengakses sistem. Sistem ini menggunakan *next/auth* untuk mengelola autentikasi pengguna. Pengguna dapat mendaftarkan akun melalui formulir pendaftaran. Setelah proses pendaftaran, *next/auth* akan mengelola autentikasi dan menyimpan data pengguna di *PostgreSQL*.

3.6.2 Robot Information

Robot Information adalah halaman yang menampilkan informasi terkait robot, seperti nama, versi, dan spesifikasi dari robot yang terdaftar dalam sistem. Halaman ini juga menampilkan status real-time robot, termasuk posisi, kecepatan, dan arah gerak. Pengguna dapat mendaftarkan robot baru melalui halaman ini, sehingga robot dapat terdaftar dan digunakan dalam sistem. Setiap robot yang didaftarkan akan memiliki ID unik yang digunakan untuk mengidentifikasi robot tersebut di seluruh sistem.

3.6.3 Patrol Controller

Patrol Controller adalah halaman utama yang digunakan untuk mengontrol robot dan memantau operasi patroli. Sistem ini mendukung operasi multi-robot, sehingga setiap robot yang terdaftar akan memiliki *Patrol Controller* masing-masing. Halaman ini menampilkan peta interaktif yang menunjukkan posisi robot, rute patroli, dan status *real-time* robot. Selain itu, citra termal robot juga ditampilkan untuk mendeteksi komponen yang mengalami *overheat*.

Pengguna dapat memilih robot yang ingin dikontrol menggunakan ID robot yang telah didaftarkan di *Robot Information*. Setiap robot memiliki kontrol manual dan otomatis melalui tombol yang tersedia di halaman ini. Di *Patrol Controller*, pengguna juga dapat membuat atau mengubah rute patroli dan menetapkan jadwal patroli untuk setiap robot secara terpisah. Dengan sistem multi-robot ini, pengguna dapat memantau dan mengelola beberapa robot secara bersamaan, masing-masing dengan kontrol dan informasi yang terpisah berdasarkan ID robot yang dipilih.

3.7 Computer Vision

Dalam mendeteksi komponen yang mengalami *overheat*, peneliti menggunakan model *YOLOv8*. Pemilihan model ini didasarkan pada kemampuannya dalam deteksi objek yang akurat dan kecepatan proses yang optimal. Proses deteksi komponen *overheat* dilakukan melalui beberapa tahapan sebagai berikut:

3.7.1 Pembuatan Dataset

Dataset citra termal dari komponen gardu listrik dikumpulkan dan diproses untuk melatih model deteksi. Pengumpulan data dilakukan melalui platform *Roboflow*, di mana setiap citra diberi label secara akurat untuk mengidentifikasi jenis komponen yang terdeteksi.

3.7.2 Pelatihan Model *YOLOv8*

Model *YOLOv8* dilatih menggunakan dataset yang telah disiapkan. Proses pelatihan mencakup penyesuaian hiperparameter dan penggunaan augmentasi data untuk meningkatkan akurasi model. Evaluasi kinerja model dilakukan dengan menggunakan metrik *mean Average Precision* (mAP) dan *Intersection over Union* (IoU) untuk menilai kemampuan deteksi. Setelah pelatihan, model diuji menggunakan dataset validasi untuk memastikan kinerja yang optimal.

3.7.3 Konversi ke *TensorRT*

Untuk meningkatkan kecepatan deteksi dan mengurangi beban komputasi, model *YOLOv8* dikonversi ke format *TensorRT* (TRT). Model yang awalnya dalam format *PyTorch* diekspor ke format *Open Neural Network Exchange* (ONNX) menggunakan perintah dari *library Ultralytics*. Setelah model diekspor ke format ONNX, konversi ke format TRT dilakukan dengan perintah `trtexec` dari NVIDIA untuk mempercepat proses inferensi.

3.7.4 Deteksi Suhu Komponen

Setelah komponen berhasil terdeteksi oleh *YOLOv8*, tahap selanjutnya adalah menganalisis suhu pada komponen dengan menggunakan segmentasi warna dalam ruang warna *HSV* (*Hue*, *Saturation*, *Value*) pada *bounding box* yang terdeteksi. Metode ini digunakan untuk mengidentifikasi area dalam citra termal yang memiliki suhu tinggi, biasanya ditunjukkan dengan warna seperti merah, kuning, atau oranye, sesuai dengan dataset kamera termal yang digunakan.

Deteksi komponen terlebih dahulu sangat penting karena setiap komponen memiliki ambang batas suhu *overheat* yang berbeda. Tanpa mengetahui jenis komponen yang terdeteksi, sulit untuk menentukan apakah suhu tersebut melebihi batas aman atau tidak. Oleh karena itu, identifikasi komponen terlebih dahulu memastikan bahwa segmentasi suhu yang dilakukan pada *bounding box* yang terdeteksi relevan dan akurat. Berdasarkan data ambang batas suhu masing-masing komponen.

[Halaman ini sengaja dikosongkan]

BAB 4 HASIL YANG DIHARAPKAN

4.1 Hasil yang Diharapkan dari Penelitian

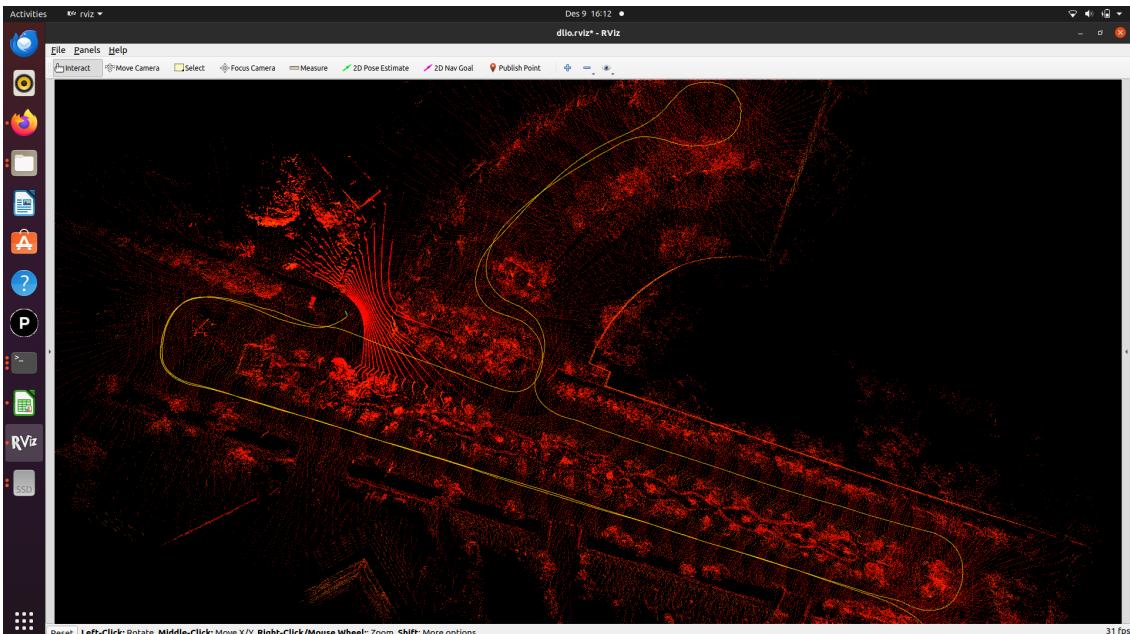
Penelitian ini diharapkan menghasilkan sistem pemantauan komponen *overheat* pada gardu listrik menggunakan robot *quadruped legged* yang dilengkapi dengan kamera termal. Sistem ini diharapkan dapat mendeteksi overheating secara *real-time*, meningkatkan efisiensi dan keamanan pemantauan gardu listrik. Robot diharapkan mampu melakukan patroli mandiri, menghindari rintangan, serta memberikan estimasi posisi dan jenis komponen yang terdeteksi mengalami *overheat*. Selain itu, sistem ini juga diharapkan memungkinkan kontrol jarak jauh melalui antarmuka *web* interaktif, sehingga memudahkan operator untuk mengawasi kondisi gardu listrik tanpa perlu berada di lokasi. Secara keseluruhan, hasil penelitian ini diharapkan dapat meningkatkan keamanan dan efisiensi dalam pemantauan gardu listrik serta memberikan kontribusi pada pengembangan teknologi robotika di sektor kelistrikan.

4.2 Hasil Pendahuluan

Sampai saat ini, peneliti telah melakukan pengujian terhadap berbagai sistem robot. Adapun hasil yang telah diperoleh adalah sebagai berikut:

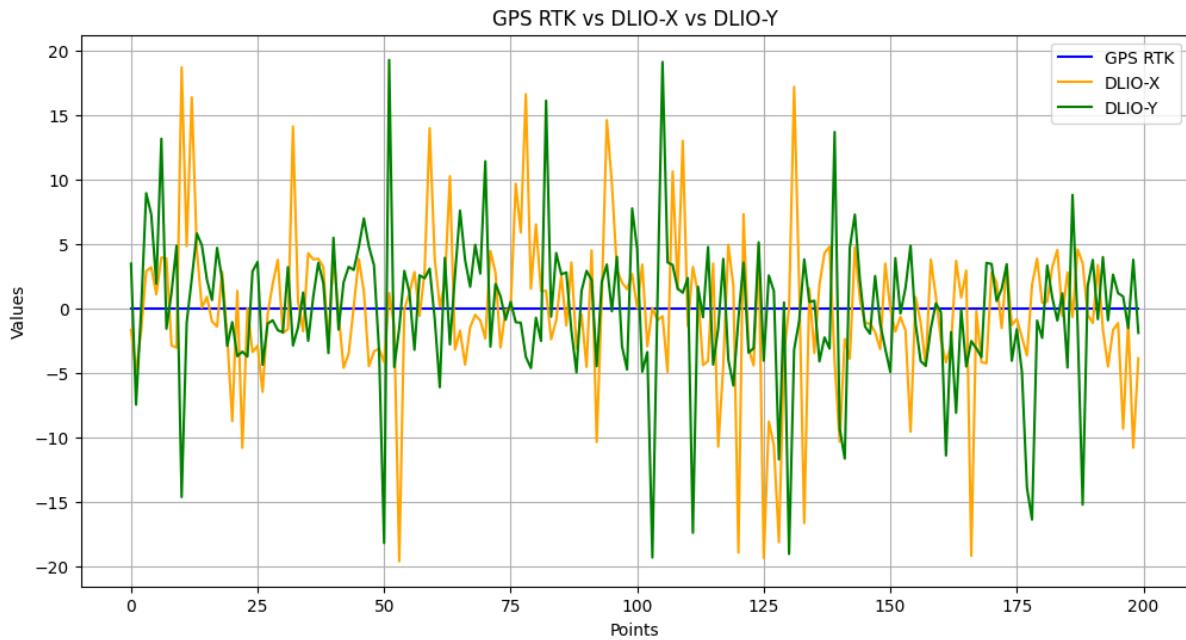
4.2.1 Pengujian *Localization Node*

Implementasi program *Localization Node* diuji menggunakan robot *iCar* di kampus ITS. Pengujian dilakukan dengan mengulang rute yang sama sebanyak dua kali untuk memverifikasi presisi estimasi posisi. Hasil pengujian menunjukkan bahwa robot mampu mendeteksi posisinya dengan sangat akurat, terlihat dari kedua pengulangan yang memberikan hasil konsisten dalam hal posisi dan orientasi.



Gambar 4.1: *DLIO* pada jalan kampus ITS

Data hasil lokalisasi menggunakan *DLIO* dibandingkan dengan data *GPS RTK* yang terdapat pada *iCar*, menghasilkan grafik seperti pada Gambar 4.2.



Gambar 4.2: Perbandingan *DLIO* dan *GPS RTK*

Grafik tersebut menunjukkan bahwa hasil lokalisasi *DLIO* memiliki deviasi yang relatif kecil dibandingkan dengan *GPS RTK*, yakni tidak melebihi 20 cm. Meskipun demikian, akurasi dapat ditingkatkan lebih lanjut dengan melakukan *fusion* data IMU dengan *odometry*. Meski begitu, hasil yang diperoleh saat ini sudah cukup memadai untuk digunakan sebagai acuan dalam pengendalian robot.

4.2.2 Pengujian Model *Computer Vision*

Dataset citra termal gardu listrik yang diperoleh dari platform *Roboflow* dilatih menggunakan *YOLOv8* dengan berbagai konfigurasi parameter. Model yang digunakan adalah *YOLOv8s*, dengan jumlah *epoch* 50, 100, 200, dan 300, *batch size* 2, 4, dan 8, serta *optimizer* *Adam* dan *SGD*. Dari kombinasi tersebut, diperoleh 48 model, dan 10 model terbaik berdasarkan nilai *mAP50* disajikan pada Tabel 4.1.

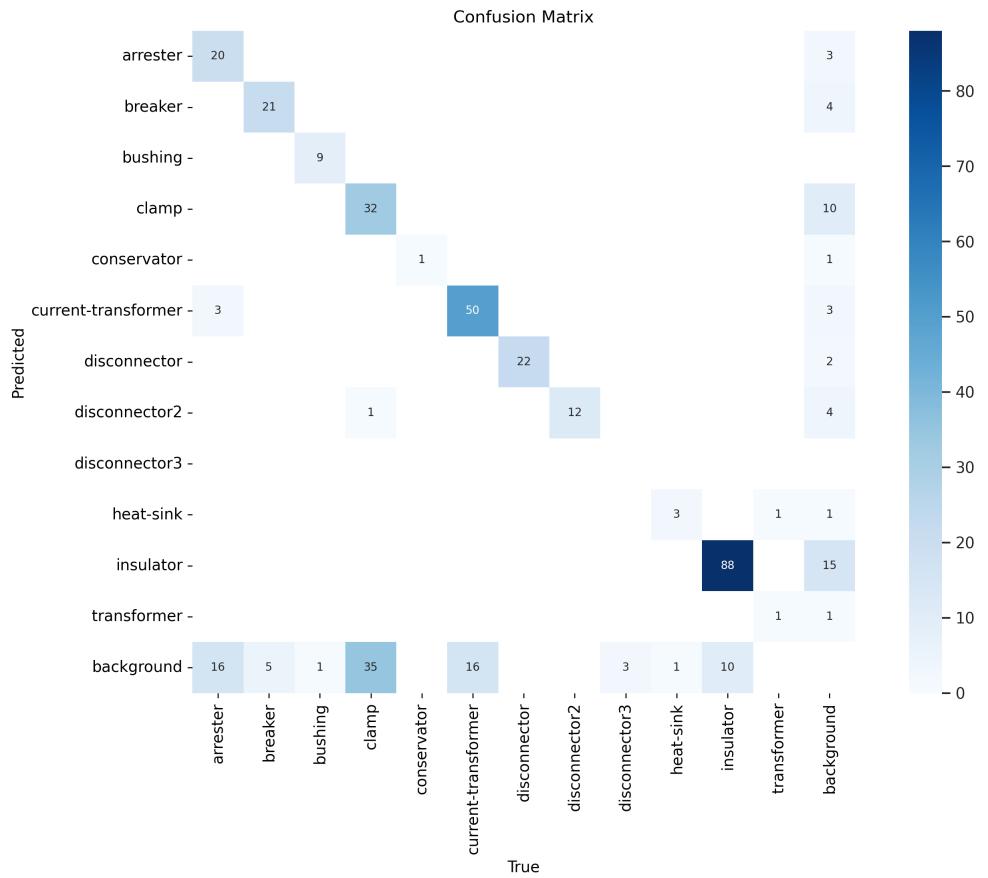
Name	Batch Size	Epochs	Optimizer	<i>mAP50</i>	<i>Precision(B)</i>	<i>Recall(B)</i>
yolov8s	8	100	SGD	0.810282	0.795060	0.711395
yolov8s	8	300	SGD	0.786689	0.702299	0.787149
yolov8s	4	50	SGD	0.773952	0.772349	0.701915
yolov8s	2	50	SGD	0.773948	0.741667	0.737056
yolov8s	4	300	SGD	0.772399	0.796335	0.679174
yolov8s	2	100	SGD	0.765646	0.859258	0.645118
yolov8s	2	300	SGD	0.764409	0.838419	0.654519
yolov8n	8	300	SGD	0.757128	0.726358	0.721584
yolov8s	8	200	SGD	0.754204	0.826324	0.652525
yolov8s	8	300	Adam	0.752367	0.829013	0.670934

Tabel 4.1: 10 Model Teratas Berdasarkan *mAP50*

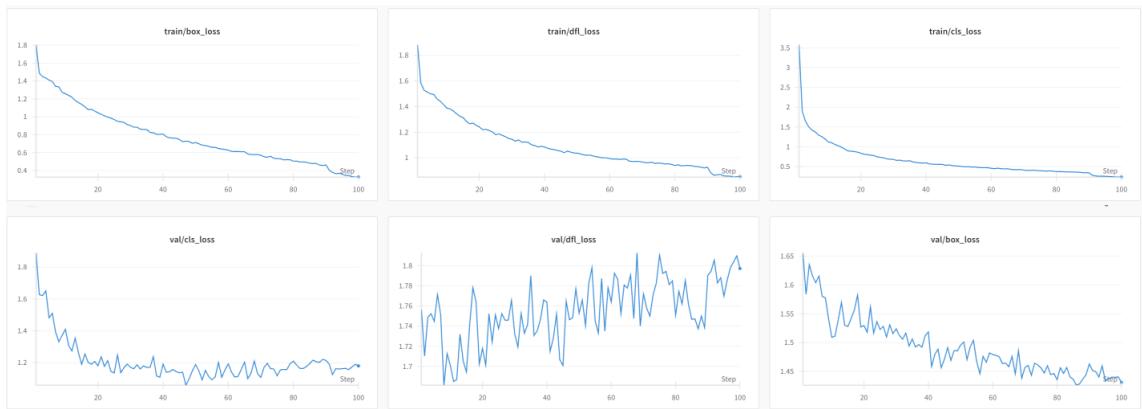
Hasil ini menunjukkan bahwa model *YOLOv8s* dengan konfigurasi *batch size* 8, *epochs* 100, dan *optimizer* *SGD* memberikan performa terbaik dengan nilai *mAP50* sebesar 0.810282

dan nilai *precision* serta *recall* yang cukup tinggi. Pengujian ini memberikan wawasan penting untuk memilih konfigurasi optimal dalam pelatihan model deteksi objek berbasis citra termal.

Adapun hasil *confusion matrix*, *lost function*, serta grafik akurasi dari model tersebut tersebut dapat dilihat pada Gambar 4.3 dan Gambar 4.4.



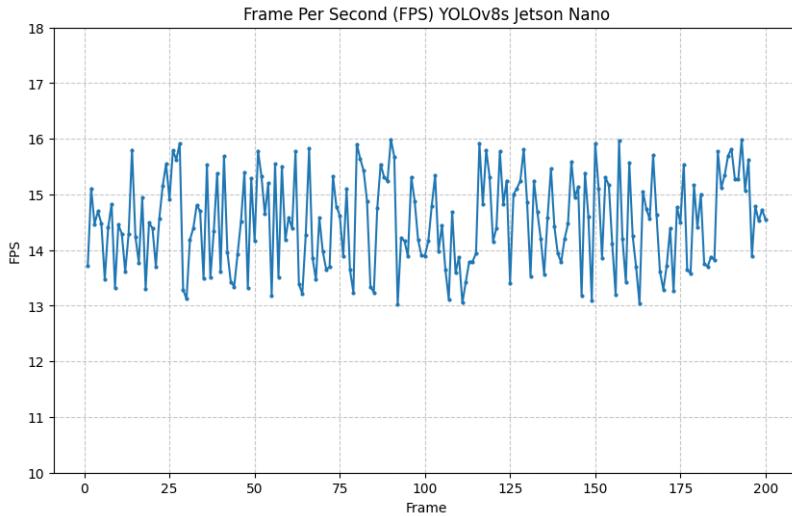
Gambar 4.3: *Confusion Matrix YOLOv8n 100 Epoch SGD 8 Batch Size*



Gambar 4.4: *Loss Function YOLOv8n 100 Epoch SGD 8 Batch Size*

Hasil deteksi dari model cukup baik. Ke depannya, performa model akan ditingkatkan dengan pengambilan data langsung dari lingkungan nyata serta penerapan augmentasi pada dataset.

Selain itu, untuk memastikan bahwa model *YOLOv8s* dapat beroperasi pada robot yang menggunakan *mini PC* Jetson Xavier NX 16 GB, dilakukan pengembangan model *YOLO* sederhana dengan konfigurasi parameter pelatihan yang sama, namun dengan kelas yang berbeda. Model tersebut diuji pada *Jetson Nano*, yang merupakan seri di bawah *Jetson Xavier NX*. Hasilnya, model mampu berjalan dengan baik dengan *frame rate per second* (FPS) sebagaimana ditunjukkan pada grafik di bawah.



Gambar 4.5: FPS Model *YOLOv8n* pada *Jetson Nano*

Dari grafik diatas didapatkan hasil model dapat berjalan di 13 hingga 16 *frame per second* (FPS) pada *Jetson Nano* dengan menggunakan *TensorRT YOLOv8 CPP*. Hasil ini menunjukkan bahwa model akan dapat berjalan dengan baik pada *Jetson Xavier NX* yang akan digunakan pada robot.

4.2.3 Pengujian *Overheat Detection Node*

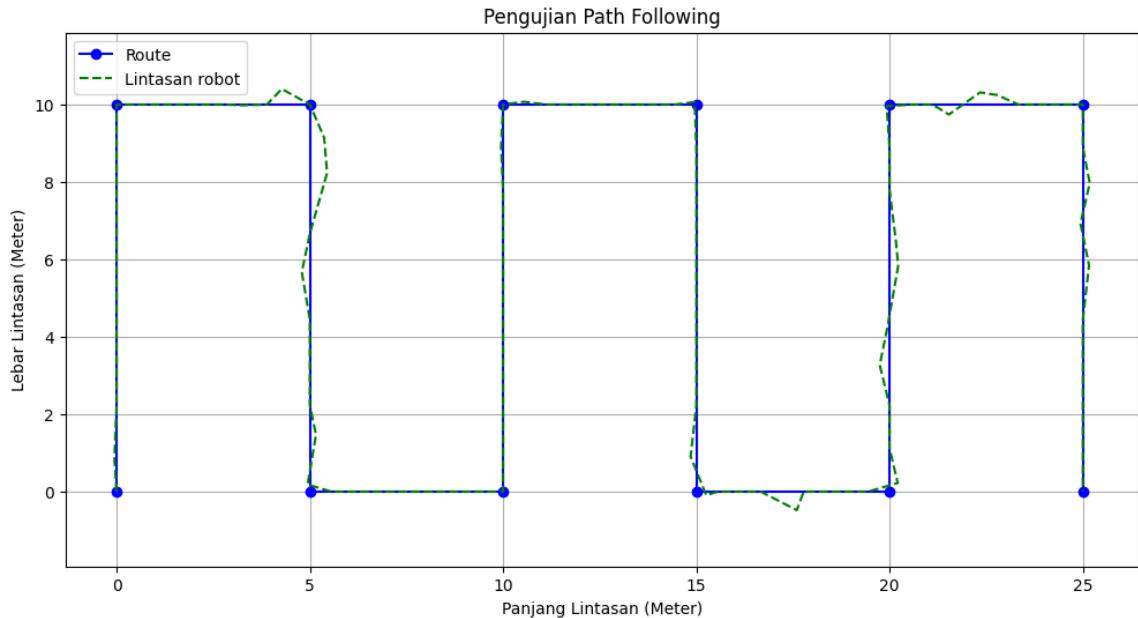
Pengujian *overheat detection* dilakukan dengan menggunakan segmentasi HSV pada *bounding box* yang terdeteksi oleh *YOLOv8s*. Pengujian dilakukan pada citra termal gardu listrik yang diperoleh dari platform *Roboflow*. Hasil pengujian menunjukkan bahwa algoritma segmentasi dapat memisahkan area *overheat* dengan baik seperti pada gambar dibawah.



Gambar 4.6: Pengujian *Overheat Detection* pada citra thermal

4.2.4 Pengujian *Path Following Node*

Pengujian *path following* dilakukan untuk menguji kemampuan algoritma dalam menggerakkan robot mengikuti lintasan yang telah ditentukan. Pengujian dilakukan di Taman Alumni ITS dengan rute yang sudah direncanakan sebelumnya. Hasil pengujian menunjukkan bahwa robot dapat mengikuti lintasan dengan baik, meskipun terdapat beberapa deviasi yang dapat diperbaiki dengan penyesuaian konstanta PID. Hasil pengujian dapat dilihat pada Gambar 4.7.

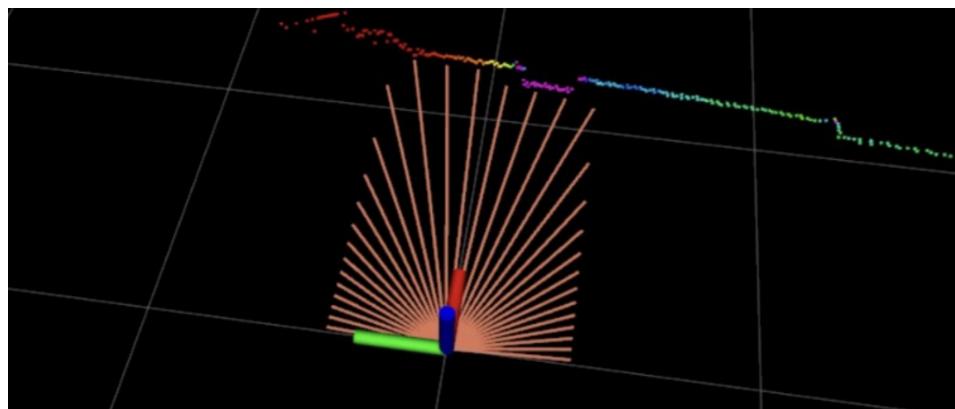


Gambar 4.7: Pengujian *Path Following* pada Robot

Namun, pengujian ini hanya menggunakan data *odometry*, sehingga hasil yang diperoleh belum optimal. Kedepanya data *DLIO* akan digunakan untuk meningkatkan akurasi posisi robot.

4.2.5 Pengujian *Obstacle Avoidance Node*

Pengujian *obstacle avoidance* dilakukan untuk menguji kemampuan algoritma dalam menghindari objek yang ada di lintasan. Pengujian dilakukan dengan mendekatkan sensor pada benda dan sensor dapat memberikan respon untuk menghindar sesuai dengan objek yang terdeteksi seperti pada gambar



Gambar 4.8: Pengujian *Obstacle Avoidance* pada Robot

[Halaman ini sengaja dikosongkan]

BAB 5 JADWAL PENELITIAN

Penelitian ini dilaksanakan selama 16 minggu, dimulai dari pembuatan robot hingga penulisan laporan seperti pada tabel dibawah ini.

Tabel 5.1: Timeline Kegiatan Pengembangan Robot

Kegiatan	Minggu															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pembuatan mekanikal robot																
Pembuatan elek-trikal robot																
Pembuatan <i>website control station</i>																
Pembuatan <i>hard-ware interface</i>																
Pembuatan program <i>lokalisasi</i>																
Pembuatan <i>com-puter vision</i>																
Pembuatan sistem <i>control</i>																
Pengujian navigasi																
Pengujian <i>computer vision</i>																
Pengujian <i>control station</i>																
Pengujian dan eval-uasi sistem di lapan-gan																
Pembuatan laporan																

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] P. L. N. (PLN), *Buku Statistik PLN 2023*. Jakarta: Perusahaan Listrik Negara (PLN), Jun. 2024.
- [2] K. E. dan Sumber Daya Mineral (ESDM), *Konsumsi listrik masyarakat meningkat, tahun 2023 capai 1.285 kWh/kapita*, REPUBLIK INDONESIA, Siaran Pers, NOMOR: 28.Pers/04/SJI/2024, Tanggal: 15 Januari 2024, 2024. Accessed: Oct. 4, 2024. [Online]. Available: <https://www.esdm.go.id/siaran-pers/konsumsi-listrik-masyarakat-2023>.
- [3] R. C. Bailey *et al.*, “Forecasting gics and geoelectric fields from solar wind data using lstms: Application in austria,” *Space Weather*, vol. 20, no. 3, e2021SW002907, 2022. doi: [10.1029/2021SW002907](https://doi.org/10.1029/2021SW002907).
- [4] T. Aksenovich, V. Bilin, Y. Saharov, and V. N. Selivanov, “Wavelet analysis of geomagnetically induced currents during the strong geomagnetic storms,” *Russian Journal of Earth Sciences*, pp. 1–12, 2022. doi: [10.2205/2022es000825](https://doi.org/10.2205/2022es000825).
- [5] H. Kalathiripi and S. Karmakar, “Analysis of transformer oil degradation due to thermal stress using optical spectroscopic techniques,” *International Transactions on Electrical Energy Systems*, vol. 27, e2346, 9 2017. doi: [10.1002/etep.2346](https://doi.org/10.1002/etep.2346).
- [6] B. Li, N. Lu, and Y. He, “The contact performance analysis of disconnecting switch’s fixed contact,” *Proceedings of the 2017 International Conference on Electronic Industry and Automation (EIA 2017)*, 2017. doi: [10.2991/eia-17.2017.13](https://doi.org/10.2991/eia-17.2017.13).
- [7] Redeia, “Asumo project: All the way from design to maintenance,” May 2023, Diakses: 29 Desember 2025. [Online]. Available: <https://www.elewit.ventures/en/news/asumo-project>.
- [8] C. Cruz Ulloa, G. Prieto Sánchez, A. Barrientos, and J. Del Cerro, “Autonomous thermal vision robotic system for victims recognition in search and rescue missions,” *Sensors*, vol. 21, no. 21, 2021, ISSN: 1424-8220. doi: [10.3390/s21217346](https://doi.org/10.3390/s21217346). [Online]. Available: <https://www.mdpi.com/1424-8220/21/21/7346>.
- [9] A. Prieto *et al.*, “Image processing technique applied to electrical substations based on drones with thermal vision for predictive maintenance,” in *2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, 2022, pp. 1–6. doi: [10.1109/ICA-ACCA56767.2022.10006138](https://doi.org/10.1109/ICA-ACCA56767.2022.10006138).
- [10] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, “Direct lidar odometry: Fast localization with dense point clouds,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022. doi: [10.1109/LRA.2022.3142739](https://doi.org/10.1109/LRA.2022.3142739).
- [11] J. M. Tambunan, A. Hariyanto, and W. K. Tindra, “Kerja pembebanan dan temperatur terhadap susut umur transformator tenaga 150/20 kv 60 mva,” *Teknik Elektro, STT-PLN*, 2023.
- [12] I. G. Suputra Widharma, P. Alit WS, D. Indah Cahya Dewi, I. Sajayasa, I. Sunaya, and I. Sangka, “Studi pengaruh pemasangan sistem counterpoise pada lightning arrester di gardu distribusi berbasis software etap,” *Jurnal Ilmiah Vastuwidya*, vol. 7, pp. 08–14, 1 2024. doi: [10.47532/jiv.v7i1.1011](https://doi.org/10.47532/jiv.v7i1.1011).

- [13] M. Kartika and I. A. Bangsa, “Pemeliharaan preventif kubikel cbog 20 kv di pt pln (persero) up3 bekasi uid jawa barat,” *Je-Unisla*, vol. 7, p. 63, 2 2022. doi: [10.30736/je-unisla.v7i2.835](https://doi.org/10.30736/je-unisla.v7i2.835).
- [14] H. Henriana and E. Permata, “Pemeliharaan dan perbaikan hot spot (titik panas) pada ds (disconnector switch) rel 2,” *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 1, pp. 93–104, 4 2022. doi: [10.55606/juprit.v1i4.869](https://doi.org/10.55606/juprit.v1i4.869).
- [15] W. M. Telaumbanua, R. Bondar, and J. Napitupulu, “Studi pemeliharaan dan pengoperasian pada gardu distribusi di pt. pln (persero) ulp gunungsitoli,” *Jurnal Teknologi Energi Uda: Jurnal Teknik Elektro*, vol. 13, p. 1, 1 2024. doi: [10.46930/jteu.v13i1.4216](https://doi.org/10.46930/jteu.v13i1.4216).
- [16] G. Moreno, “Power electronics thermal management research: Annual progress report,” 2017. doi: [10.2172/1404874](https://doi.org/10.2172/1404874).
- [17] S. Ilomets, D. Heim, I. Chodák, D. Czarny, and T. Kalamees, “A method to develop energy activated etics,” *E3S Web of Conferences*, vol. 172, p. 21006, 2020. doi: [10.1051/e3sconf/202017221006](https://doi.org/10.1051/e3sconf/202017221006).
- [18] D. Robotics, *Jueying lite3 perception development manual (beta)*, version V2.1.1-0, Accessed: 2024-07-26, 2024. [Online]. Available: <https://deeprobotics.com/manuals/jueying-lite3-perception>.
- [19] D. Robotics, *Jueying lite3 motion host communication interface (beta)*, version V1.0.7-0, Accessed: 2024-05-15, 2024. [Online]. Available: <https://deeprobotics.com/manuals/jueying-lite3-motion>.
- [20] Open Source Robotics Foundation, *Ros noetic ninjemys documentation*, Accessed: 2024-10-16, 2020. [Online]. Available: <http://wiki.ros.org/noetic>.
- [21] K. Kartono, A. Suryanto, and E. B. Apriyanto, “Pemanfaatan teknologi infrared thermography untuk deteksi dini kegagalan isolasi jaringan kabel listrik,” *Fakultas Teknik, Universitas Negeri Semarang*, 2023.
- [22] Y. Contributors, *Yolov8: Object detection and image segmentation*, Accessed: 2024-10-17, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [23] D. Barbosa, A. Lopes, and R. E. Araújo, “Sensor fusion algorithm based on extended kalman filter for estimation of ground vehicle dynamics,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 1049–1054. doi: [10.1109/IECON.2016.7793145](https://doi.org/10.1109/IECON.2016.7793145).
- [24] K. Chen, R. Nemiroff, and B. T. Lopez, “Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction,” *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3983–3989, 2023. doi: [10.1109/ICRA48891.2023.10160508](https://doi.org/10.1109/ICRA48891.2023.10160508).
- [25] I. Fette and A. Melnikov, *The websocket protocol*, 2011. doi: [10.17487/rfc6455](https://doi.org/10.17487/rfc6455).
- [26] J. Milsap *et al.*, “Bci2000web and webfm: Browser-based tools for brain computer interfaces and functional brain mapping,” *Frontiers in Neuroscience*, vol. 13, p. 1030, 2019. doi: [10.3389/fnins.2018.01030](https://doi.org/10.3389/fnins.2018.01030).
- [27] Sugiyono, *Metode Penelitian dan Pengembangan: Research and Development (R&D)*, Indonesian. Bandung: Alfabeta, 2019, ISBN: 978-602-289-158-1.

- [28] A. Z. Akbar, “Autonomous surface vehicle pencari korban kecelakaan laut berbasis computer vision,” Institut Teknologi Sepuluh Nopember, Surabaya, 2022.