



TESIS - EE235401

**PENGEMBANGAN POINT OF INTEREST PADA
INDOOR DRONE MENGGUNAKAN FEATURE
MATCHING**

MAHARANI WISUDAWATI SWISTA
6022222013

DOSEN PEMBIMBING
Ronny Mardiyanto, S.T., M.T., Ph.D.
Dr. Rudy Dikairono, S.T., M.Sc.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK ELEKTRONIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025



TESIS - EE235401

PENGEMBANGAN POINT OF INTEREST PADA INDOOR DRONE MENGGUNAKAN FEATURE MATCHING

MAHARANI WISUDAWATI SWISTA
6022222013

DOSEN PEMBIMBING
Ronny Mardiyanto, S.T., M.T., Ph.D.
Dr. Rudy Dikairono, S.T., M.Sc.

PROGRAM MAGISTER
BIDANG KEAHLIAN TEKNIK ELEKTRONIKA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2025

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

MAHARANI WISUDAWATI SWISTA

NRP: 6022222013

Tanggal Ujian: 11 Juli 2025

Periode Wisuda: September 2025

Disetujui oleh:

Pembimbing:

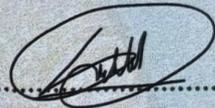
1. Ronny Mardiyanto, ST., MT., Ph.D.

NIP: 19810118 200312 1 003


.....

2. Dr. Rudy Dikairono, S.T., M.Sc.

NIP: 19810325 200501 1 002


.....

Penguji:

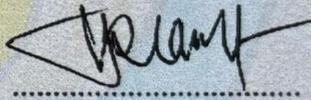
1. Dr.Ir. Djoko Purwanto, M.Eng..

NIP: 19651211 199002 1 001


.....

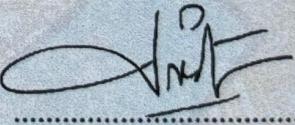
2. Prof. Dr. Tri Arief Sardjono, S.T., M.T.

NIP: 19700212 199512 1 001


.....

3. Dr. Rachmad Setiawan, S.T., M.T.

NIP: 19690529 199512 1 001


.....

Kepala Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas



Ronny Mardiyanto, S.T., M.T., Ph.D.

NIP: 19810118 200312 1 003

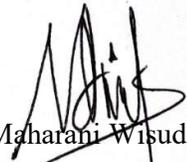
Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **“PENGEMBANGAN POINT OF INTEREST PADA INDOOR DRONE MENGGUNAKAN FEATURE MATCHING”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 5 Mei 2025



Maharani Wisudawati Swista
NRP. 6022222013

Halaman ini sengaja dikosongkan

PENGEMBANGAN POINT OF INTEREST PADA INDOOR DRONE MENGGUNAKAN FEATURE MATCHING

Nama mahasiswa : Maharani Wisudawati Swista
NRP : 6022222013
Pembimbing : 1. Ronny Mardianto, S.T., M.T., Ph.D.
2. Dr. Rudy Dikairono, S.T., M.Sc.

ABSTRAK

Penelitian ini membahas sistem navigasi drone dalam ruangan menggunakan teknologi Point of Interest (POI) berbasis penginderaan visual. Drone jenis MAV (Micro Aerial Vehicle) digunakan untuk mengorbit objek target secara otonom tanpa bergantung pada GPS, yang sering terganggu dalam lingkungan tertutup. Algoritma ORB dipilih untuk mendeteksi dan mencocokkan citra objek 3D secara waktu nyata karena ketahanannya terhadap noise. Target pengujian berupa lingkaran berdiameter 10 cm, 20 cm, dan 30 cm, dengan variasi menggunakan kontur dan tanpa kontur. Hasil menunjukkan bahwa penggunaan kontur membantu sistem visual mempertahankan struktur distribusi titik estimasi, meskipun akurasi tetap rendah. Untuk diameter 10 cm, rata-rata error mencapai 0,24 meter dengan RMSE 0,28 meter. Sementara itu, pengukuran tanpa kontur menghasilkan distribusi titik yang lebih acak dan sulit dikenali. Kamera Intel RealSense T265 yang digunakan untuk pemetaan posisi menunjukkan adanya penyimpangan (drift), terutama dalam kondisi tanpa kontur. Pendeteksian menggunakan ORB cukup efektif, ditunjukkan dengan akurasi tertinggi pada frame ke-16 (error 7,07%). Namun, frame lain seperti ke-39 dan ke-59 menunjukkan error besar hingga 102,94%. Kesimpulannya, sistem navigasi visual ini masih membutuhkan peningkatan untuk aplikasi presisi di lingkungan dalam ruangan.

Kata kunci : Point of Interest, pencocokan fitur, objek, drone, T265 Realsense

Halaman ini sengaja dikosongkan

POINT OF INTEREST MODE FOR INDOOR DRONE BASED ON FEATURE MATCHING

By : Maharani Wisudawati Swista
Student Identity Number : 6022222013
Supervisor(s) : 1. Ronny Mardianto, S.T., M.T., Ph.D.
2. Dr. Rudy Dikairono, S.T., M.Sc.

ABSTRACT

This study explores an indoor drone navigation system utilizing Point of Interest (POI) technology based on visual sensing. A Micro Aerial Vehicle (MAV) is employed to autonomously orbit a target object without relying on GPS, which is often disrupted in enclosed environments. The ORB algorithm is selected for detecting and matching 3D object images in real time due to its robustness against noise. The test targets are circular objects with diameters of 10 cm, 20 cm, and 30 cm, with variations using contour and no contour. The results show that the use of contours helps the visual system maintain a more structured distribution of estimated position points, although the overall accuracy remains low. For the 10 cm diameter target, the average error reached 0.24 meters with an RMSE of 0.28 meters. In contrast, measurements without contours resulted in more scattered and unrecognizable point distributions. The Intel RealSense T265 camera used for position mapping exhibited significant drift, especially in the absence of contours. ORB-based detection proved to be relatively effective, with the highest accuracy observed at frame 16 (7.07% error). However, other frames such as frame 39 and frame 59 showed much higher errors, up to 102.94%. In conclusion, while the visual-based navigation system shows potential, further optimization is required for precise indoor applications.

Key words: Point of Interest, feature matching ORB, drone, object, T265
Realsense

Halaman ini sengaja dikosongkan

KATA PENGANTAR

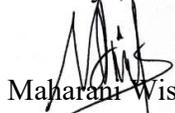
Puji dan syukur kepada Allah SWT atas berkat, rahmat dan karunianya sehingga penulis dapat menyelesaikan laporan Tesis dengan judul “PENGEMBANGAN POINT OF INTEREST PADA INDOOR DRONE MENGGUNAKAN FEATURE MATCHING”. Buku Tesis ini merupakan salah satu persyaratan dalam menyelesaikan pendidikan program Magister di Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.

Dalam penyusunan dan penyelesaian laporan Tesis ini penulis tidak lupa dari bantuan dan doa berbagai pihak. Untuk itu penulis akan mengucapkan banyak terimakasih kepada:

1. Kedua orang tua dan seluruh anggota keluarga yang selalu menjadi motivasi utama penulis dalam penyelesaian laporan Tesis ini.
2. Bapak Ronny Mardiyanto, S.T., M.T., Ph.D. selaku dosen pembimbing 1 atas usulan judul topik Tesis, bimbingan serta arahan yang telah diberikan kepada penulis selama proses pengerjaan laporan Tesis ini.
3. Bapak Dr. Rudy Dikairono, S.T., M.Sc. selaku dosen pembimbing 2 atas evaluasi, bimbingan serta arahan yang telah diberikan kepada penulis selama proses pengerjaan laporan Tesis ini.
4. Teman-teman perkuliahan yang banyak memberikan bantuan baik dalam pengujian ataupun penyusunan laporan Tesis ini.

Penulis menyadari bahwa pada laporan Tesis ini masih terdapat kekurangan dan banyak bagian yang dapat dikembangkan lebih lanjut. Oleh karena itu penulis dengan senang hati menerima kritik maupun saran yang diberikan pembaca dan meminta maaf atas segala kekurangan yang masih terdapat pada laporan Tesis ini.

Surabaya, 5 Mei 2025



Maharani Wisudawati Swista

NRP. 6022222013

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN TESIS.....	iii
PERNYATAAN KEASLIAN TESIS.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Kontribusi.....	3
BAB 2 KAJIAN PUSTAKA	5
2.1 Kajian Penelitian Terkait	5
2.2 Teori Dasar.....	6
2.2.1 DJI Tello.....	6
2.2.2 Deteksi Object Berbasis Visi Komputer.....	8
2.2.3 Feature Matching.....	9
2.2.4 Algoritma ORB.....	10
2.2.5 Point of Interest.....	11
2.2.6 Euclidean Distance.....	13
2.2.7 ROS (Robot Operating System).....	15
2.2.8 T265 Realsense.....	16
2.2.8.1 Modul Pelacakan Intel Realsense T261.....	17
2.2.8.2 Modul Pelacakan Intel Realsense T261.....	18
2.2.8.3 Unit Pengukuran Inersia.....	18
2.2.8.4 Fisheye Imagers.....	18
BAB 3 METODE PENELITIAN.....	21

3.1	Rancangan Penelitian.....	21
3.2	Sistem Komunikasi Antara Drone dan Pengendali.....	21
3.3	Pengujian Stabilitas Drone.....	23
3.4	Pergerakan Drone Pada Point of Interest.....	25
3.5	Deteksi Point of Interest.....	26
3.6	Pengujian Sistem Feature Matching.....	27
3.7	Evaluasi Deteksi dan Identifikasi Feature Matching.....	28
3.8	Intel Realsense T265 Sebagai Sistem Navigasi Indoor.....	30
BAB 4 HASIL DAN PEMBAHASAN.....		32
4.1	Pengumpulan Dataset.....	32
4.2	Pengujian Deteksi Berdasarkan Nilai Error Posisi.....	33
4.3	Evaluasi Kinerja Pelacakan Berdasarkan Kecepatan Gerak.....	34
4.4	Evaluasi Akurasi Pose Kamera T265.....	35
4.4.1	Hasil akurasi pose kamera T265 terhadap contour target dengan diameter 10 cm.....	36
4.4.2	Hasil akurasi pose kamera T265 terhadap contour target dengan diameter 20 cm.....	38
4.4.3	Hasil akurasi pose kamera T265 terhadap contour target dengan diameter 30 cm.....	40
4.4.4	Hasil akurasi pose kamera T265 tanpa contour target dengan diameter 10 cm.....	42
4.4.5	Hasil akurasi pose kamera T265 tanpa contour target dengan diameter 10 cm.....	44
4.4.6	Hasil akurasi pose kamera T265 tanpa contour target dengan diameter 30 cm.....	46
BAB 5 KESIMPULAN.....		48
5.1	Kesimpulan.....	48
DAFTAR PUSTAKA.....		50
LAMPIRAN.....		55

DAFTAR GAMBAR

Gambar 2.2.1 DJI Tello.....	7
Gambar 2.2.1.2 Drone tampak belakang.....	8
Gambar 2.2.2 Deteksi objek berbasis visi komputer.....	9
Gambar 2.2.3 Feature Matching.....	10
Gambar 2.2.5 Point of Interest.....	12
Gambar 2.2.7 ROS (Robot Operating System).....	16
Gambar 2.2.8 T265 Realsense.....	16
Gambar 2.2.8.1. Deskripsi SKU Produk Modul Pelacakan.....	17
Gambar 2.2.8.2. Properti Modul Pelacakan.....	18
Gambar 2.2.8.3. Spesifikasi Pengukuran Inersia.....	18
Gambar 2.2.8.4. Properti Sensor Gambar Fisheye.....	19
Gambar 3.1 Visualisasi Navigasi Drone Indoor Menuju Target Dengan Bantuan Operator Dalam Deteksi Objek.....	21
Gambar 3.2 Drone DJI Ryze Tello.....	22
Gambar 3.2.1 Mekanisme komunikasi antara drone dan komputer.....	22
Gambar 3.3 Pengujian drone terbang awal.....	23
Gambar 3.3.1 Drone turun 20 cm.....	24
Gambar 3.3.2 Drone naik 30 cm.....	24
Gambar 3.2.3 Drone turun 30 cm.....	25
Gambar 3.3 Visualisasi Pergerakan Drone Menuju Titik Target.....	26
Gambar 3.4 Objek yang digunakan sebagai objek target.....	27
Gambar 3.5 Pengujian Sistem Feature Matching.....	28
Gambar 3.4 Evaluasi deteksi dan identifikasi feature matching.....	29
Gambar 3.6 Blok diagram Desain Eksperimen Pengujian Kamera T265.....	30
Gambar 4.4 Evaluasi Kinerja Pelacakan Berdasarkan Kecepatan Gerak.....	35
Gambar 4.4.1 Scatter diameter 10 cm dengan contour.....	37
Gambar 4.4.2 Scatter diameter 20 cm dengan contour.....	39
Gambar 4.4.3 Scatter diameter 30 cm dengan contour.....	41

Gambar 4.4.4 Scatter diameter 10 cm tanpa contour.....	43
Gambar 4.4.5 Scatter diameter 20 cm tanpa contour.....	45

DAFTAR TABEL

Tabel 4.2 Hasil Data Pengujian.....	34
Tabel 4.2.2 Hasil Data Pengujian.....	34
Tabel 4.1.1 Akurasi Diameter 10 cm Dengan Contour.....	36
Tabel 4.2.2 Akurasi Diameter 20 cm Dengan Contour.....	38
Tabel 4.3.3 Akurasi Diameter 30 cm Dengan Contour.....	40
Tabel 4.4.4 Akurasi Diameter 10 cm Tanpa Contour.....	42
Tabel 4.4.5 Akurasi Diameter 20 cm Tanpa Contour.....	44
Tabel 4.4.6 Akurasi Diameter 30 cm Tanpa Contour.....	46

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Mikro Aerial Vehicles (MAVs) adalah kendaraan udara tak berawak (UAV) kecil yang dirancang untuk terbang dekat dengan objek dan bernavigasi di area yang sulit diakses atau tidak memiliki jangkauan GPS. Namun memerlukan navigasi otonom tingkat tinggi dan navigasi berbasis visi komputer [1][2]. Pencocokan fitur gambar untuk UAV adalah bagian dasar komputer vision dalam pengenalan pola visual navigasi[3] dan tugas yang menantang dalam fotografi pencocokan [4]. Drone banyak digunakan saat ini untuk berbagai tujuan, seperti operasi militer, pengiriman paket, bantuan darurat, misi penyelamatan, pemantauan lingkungan, pengawasan [5][6] dan termasuk pertanian[7].

Drone DJI Tello memiliki dua mode operasi: kendali jarak jauh dan navigasi mandiri. Sensor Inertial Measurement Unit (IMU) merupakan komponen standar drone DJI Tello. Ini mengukur kecepatan sudut, orientasi absolut, dan percepatan linier [8]. Sedangkan, untuk drone dji Mavic Air memiliki beberapa mode operasi : kendali navigasi secara mandiri yang lebih maju, menggunakan GPS dan GLONASS untuk posisi yang lebih akurat. Serta terdapat Intelligent Flight Modes seperti ActiveTrack, Point of Interest, Waypoints, dan QuickShot [9].

Fitur drone berkembang pesat, salah satunya adalah fitur Point of Interest (POI) pada drone outdoor. Fitur ini digunakan dalam fotografi untuk mendeteksi lokasi tertentu yang dijadikan fokus pengambilan data atau gambar[10]. Tugasnya mencakup setiap titik di area tertentu[11]. Dengan fitur tersebut, drone dapat mengumpulkan informasi dari lokasi dan kemudian menentukan apakah memenuhi standar kualitas yang dipersyaratkan. Perbedaan mode pada drone dinilai kurang efektif jika diterapkan pada drone indoor, karena pada fitur POI harus menggunakan GPS.

Dalam penelitian ini, kami mengusulkan pendekatan *point of interest* untuk mode drone indoor melalui pencocokan fitur dan deteksi gerakan berdasarkan perubahan piksel dalam urutan video[13]. Akurasi ditingkatkan dengan membandingkan gambar sebelum dan sesudah *cropping* menggunakan deskriptor

jarak Euclidean. Untuk mendukung pemetaan dan estimasi posisi secara real-time, digunakan sensor Intel RealSense T265 berbasis visual-inertial odometry yang menggabungkan data kamera stereo dan IMU, sehingga meningkatkan stabilitas pelacakan serta akurasi navigasi dan pencocokan fitur di lingkungan indoor.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, beberapa rumusan masalah dari topik penelitian tersebut adalah:

1. Bagaimana cara mengidentifikasi dan mendeteksi point of interest (POI) didalam ruangan menggunakan drone?
2. Sejauh mana efektivitas teknik ORB dalam pencocokan fitur untuk pengenalan objek di lingkungan indoor?
3. Apa saja tantangan yang dihadapi dalam pengolahan citra dan pencocokan fitur pada drone yang beroperasi di dalam ruangan?

1.3 Tujuan

Berdasarkan perumusan masalah yang telah dipaparkan, tujuan dari penelitian ini adalah sebagai berikut

1. Mengembangkan metode untuk mendeteksi dan mengidentifikasi point of interest (POI) di dalam ruangan menggunakan drone.
2. Mengevaluasi efektivitas metode ORB dalam proses pencocokan fitur untuk mendukung pengenalan objek di dalam ruangan.
3. Mengidentifikasi dan mengatasi tantangan dalam pengolahan citra serta pencocokan fitur untuk meningkatkan akurasi dan efisiensi sistem navigasi drone di dalam ruangan.

1.4 Batasan Masalah

Untuk memfokuskan permasalahan yang diangkat maka dilakukan pembatasan masalah. Batasan-batasan masalah tersebut diantaranya adalah

1. Penelitian ini hanya akan fokus pada penggunaan drone di lingkungan indoor, tanpa mempertimbangkan penggunaan di luar ruangan.
2. Penelitian ini membatasi analisis pencocokan fitur pada metode ORB (Oriented FAST and Rotated BRIEF) karena efisiensinya dalam kondisi nyata serta kesesuaiannya dengan keterbatasan komputasi.

3. Penelitian ini tidak akan membahas aspek fisik drone, seperti desain dan spesifikasi teknis, tetapi lebih pada navigasi dan pemetaan dibantu oleh sensor Intel RealSense T265, tanpa menggunakan metode pemetaan eksternal lainnya.

1.5 Kontribusi

Penelitian ini diharapkan dapat memberikan pemahaman yang lebih baik tentang teknik feature matching yang dapat digunakan untuk mendeteksi point of interest di dalam ruangan, serta menghasilkan metode yang dapat diimplementasikan pada drone untuk meningkatkan kemampuan navigasi dan pengenalan objek di lingkungan indoor. Selain itu, penelitian ini juga menyediakan dasar bagi penelitian lebih lanjut dalam pengembangan teknologi drone dan aplikasi di bidang pemetaan, pengawasan, dan layanan dalam ruangan, sehingga dapat membuka peluang untuk inovasi dan penerapan yang lebih luas di masa depan

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

Pada bab 2 ini akan dibahas terlebih dahulu tentang penelitian terkait yang telah dilakukan peneliti lain. Kemudian akan disajikan teori penunjang yang mendasari penelitian tesis ini. Teori penunjang dijabarkan secara ringkas untuk menjadi dasar dalam menyelesaikan penelitian yang lebih terarah.

2.1 Kajian Penelitian Terkait

Drone DJI Tello memiliki dua mode operasi, yaitu kendali jarak jauh dan navigasi mandiri dengan sensor IMU untuk mengukur kecepatan sudut, orientasi, dan percepatan linier [1]. Sementara itu, drone DJI Mavic Air memiliki mode navigasi mandiri yang lebih canggih menggunakan GPS dan GLONASS untuk akurasi posisi, serta fitur Intelligent Flight Modes seperti ActiveTrack dan Point of Interest (POI) [2]. Fitur POI, yang sering digunakan dalam fotografi outdoor untuk mendeteksi lokasi tertentu, memungkinkan pengumpulan data dari area fokus. Namun, penggunaan fitur ini dinilai kurang efektif pada drone indoor karena memerlukan GPS.

Penelitian [3] menyajikan perbaikan pada algoritma ORB. Setelah mengelompokkan fitur-fitur yang cocok secara berurutan, peneliti menggunakan skala titik fitur yang dikombinasikan dengan sudut dan jarak fitur-fitur yang cocok untuk menyaring ketidakcocokan fitur. Hasil penelitian menunjukkan bahwa algoritma ORB yang ditingkatkan lebih baik dibandingkan dengan algoritma ORB, dalam hal akurasi pencocokan dan jumlah perbandingan yang benar. Kompleksitas dari algoritma yang diusulkan, menghasilkan algoritma yang diusulkan menjadi sekitar 30% lebih lambat dari algoritma ORB dan 2,5 kali lebih cepat dari algoritma LATCH. Algoritme ORB yang ditingkatkan mengekstrak deskriptor ORB dengan membandingkan dua piksel. Hasil percobaan menunjukkan bahwa algoritma ORB yang ditingkatkan lebih efektif meningkatkan akurasi pencocokan dan meningkatkan jumlah perbandingan yang benar.

Penelitian[4] ORB SLAM diusulkan karena kinerjanya dan bekerja dengan baik untuk sistem monokuler. Fitur ORB dapat diekstraksi dari gambar tanaman tomat, buah-buahan, dan tanaman merambat yang ditangkap oleh kamera

onboard drone. Hasil percobaan juga menunjukkan bahwa cloud ORB SLAM dapat membuat peta presisi yang dapat digunakan untuk memperkirakan posisi drone dan dapat digunakan untuk menavigasi drone ke lingkungan yang diketahui.

Penelitian ini [5] mengevaluasi penggunaan kamera Intel RealSense T265 berbasis *visual odometry* pada robot Four Wheel Omni Mobile Robot (FWOMR) untuk navigasi. Berbeda dengan odometri roda biasa, visual odometry tidak terpengaruh oleh slip roda, sehingga lebih akurat dalam menentukan posisi robot.

Kamera T265 memberikan data posisi 2D secara real-time untuk mengendalikan pergerakan robot. Saat diuji tanpa dipasang pada robot, kesalahan pembacaan posisi (RMSE) terbesar tercatat 1,61 cm pada sumbu z. Saat digunakan untuk mengontrol gerakan dengan PID ($K_p = 2.0$, $K_i = 0.1$, $K_d = 0.1$), RMSE terbesar adalah 3,47 cm pada sumbu x.

FWOMR juga berhasil mengikuti lintasan berbentuk persegi dan segitiga, dengan RMSE maksimum 3,24 cm pada sumbu x. Hasil ini menunjukkan bahwa kamera T265 efektif digunakan untuk pelacakan dan pengendalian navigasi dasar robot.

2.2 Teori Dasar

Beberapa teori dasar yang digunakan terkait penelitian ini adalah pengetahuan umum tentang UAV (drone), membuat pengembangan fitur point of interest pada drone indoor dan menggunakan feature matching meliputi pengenalan pola, analisis citra, dan algoritma pencocokan fitur. Konsep ini penting untuk mengidentifikasi dan memetakan objek dalam lingkungan tertutup secara akurat.

2.2.1 DJI Tello

Memiliki kemampuan di udara, live streaming, dan mampu memprogram badan pesawat quadcopter. DJI Tello Ryzen memiliki spesifikasi dengan Panjang baling-baling 3 inchi. Memiliki kaki rotor yang kecil membuatnya lebih mudah untuk terbang didalam ruangan. DJI tello memiliki fitur stabilisasi gambar elektronik (EIS) sehingga hasil foto dan video lebih stabil. DJI Tello bisa terbang dalam 13 menit, dengan pengambilan jarak gambar transmisi 100 meter. Gambar atau rekaman video dengan kualitas 720 HD dan dilengkapi dengan kamera 5 megapixel [6].

Untuk mengontrol pergerakan drone DJI Tello dari komputer menggunakan keyboard sebagai antarmuka. tampilan sistem pengolahan gambar menggunakan OpenCV yang telah diinstal PySimpleGUI dan telah dilengkapi dengan fitur tombol "Simpan" untuk menyimpan gambar. Dalam konteks deteksi objek berbasis drone dan laptop, penting untuk memahami jarak kendali maksimum drone dan jarak deteksi objek maksimum. Kontrol drone maksimum pada laptop biasanya melibatkan kehadiran saluran komunikasi nirkabel antara drone dan pendali, seperti Wi-Fi atau tautan radio [7].



Gambar 2.2.1 DJI Tello

Dji Tello drone ini juga sudah dilengkapi dengan Vision Positioning System yang letakkan di bawah badan drone, komponen tersebut berbentuk kamera inframerah yang digunakan untuk mendeteksi relatif terhadap permukaan bawah. VPS pada drone digunakan untuk membantu menstabilkan posisi saat terbang dan menjaga ketinggian. Selain itu, VPS bekerja optimal jika permukaannya di bawah memiliki pola visual yang jelas, seperti lantai bermotif atau bergaris, namun akan menjadi tidak efektif di permukaan reflektif, terlalu gelap, atau terlalu terang, misalnya lantai putih polos atau kaca.



Gambar 2.2.1.2 Drone tampak belakang

2.2.2 Deteksi Object Berbasis Visi Komputer

Visi komputer adalah bidang ilmu yang memungkinkan komputer untuk memperoleh dan memahami gambar atau video dengan meniru kemampuan penglihatan manusia. Aplikasi utamanya meliputi pencarian titik korespondensi antara foto yang sama, melalui proses registrasi gambar, kalibrasi kamera, pengenalan objek, dan pencocokan fitur. Proses ini terdiri dari tiga tahap utama dan memanfaatkan algoritma pembelajaran mesin serta pembelajaran mendalam untuk mengenali pola visual berdasarkan atribut seperti warna, ukuran, bentuk, dan kilauan. Dengan demikian, visi komputer tidak hanya mereplikasi penglihatan manusia, tetapi juga memungkinkan komputer untuk secara otomatis dan efisien memahami serta bertindak berdasarkan informasi visual [8]

Visi komputer dan pembelajaran mendalam (deep learning) telah merevolusi berbagai bidang dengan memungkinkan mesin untuk memahami dan menafsirkan informasi visual. Deteksi objek dalam visi komputer telah mengalami kemajuan pesat berkat pembelajaran mendalam, yang mengarah pada pengembangan dua jenis utama algoritma: detektor dua tahap (two-stage) dan detektor satu tahap (single-stage). Meskipun detektor dua tahap umumnya menghasilkan akurasi yang lebih tinggi, detektor satu tahap seperti arsitektur You Only Look Once (YOLO) menawarkan waktu inferensi yang jauh lebih cepat, sehingga sangat cocok untuk aplikasi yang membutuhkan pemrosesan secara real-time[9].

YOLOv8 menggabungkan algoritma pelacakan untuk pemantauan objek secara terus-menerus dengan mengasosiasikan objek yang terdeteksi antar frame berturut-turut, menciptakan pelacakan yang stabil bahkan untuk objek yang bergerak cepat. Sistem ini mengandalkan fitur seperti koordinat bounding box, label kelas, dan informasi objek seperti ukuran dan penampilan untuk menjaga identitas objek. Algoritma pelacakan ini juga memprediksi posisi objek dalam kondisi menantang, seperti saat objek terhalang, serta meminimalkan pergeseran posisi (drift) dengan mengkalibrasi ulang lokasi objek secara berkala. Salah satu kekuatan utama YOLOv8 adalah efisiensinya, yang menggabungkan deteksi dan pelacakan dalam satu kerangka, mengurangi beban komputasi dan meningkatkan kecepatan pemrosesan, memungkinkan pelacakan real-time yang sangat penting dalam lingkungan dinamis, seperti kendaraan otonom atau sistem robotik [10].

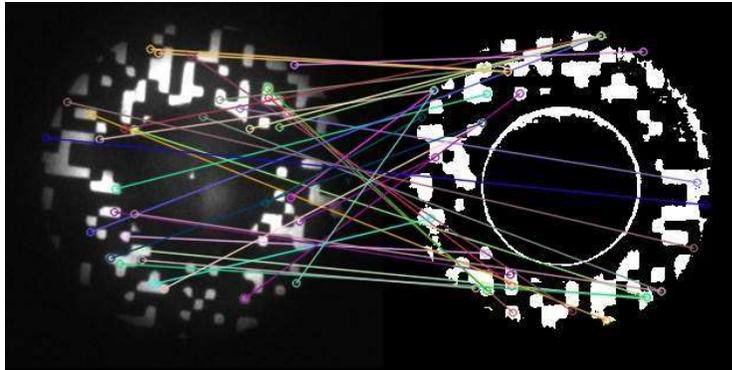


Gambar 2.2.2 Deteksi objek berbasis visi komputer

2.2.3 Feature Matching

Pencocokan fitur adalah metode untuk menyelaraskan gambar yang diambil dari pemandangan yang sama dengan menggunakan fitur-fitur yang andal. Karena memiliki tingkat ketahanan yang tinggi terhadap perubahan identitas dan distorsi gambar, teknik ini banyak digunakan dalam aplikasi dengan perubahan gambar yang kompleks. Prosesnya melibatkan pencocokan titik-titik fitur berdasarkan deskriptor lokal yang serupa untuk membentuk kumpulan korespondensi awal, yang mencakup sebagian besar inlier (pencocokan yang benar)

serta sejumlah outlier (pencocokan yang salah). Tahap berikutnya adalah menghilangkan outlier dari kumpulan korespondensi awal tersebut[11].



Gambar 2.2.3 Feature Matching

2.2.4 Algoritma ORB

Algoritma ORB (Oriented FAST and Rotated BRIEF) adalah metode ekstraksi fitur yang efisien secara komputasi dan cepat, menggabungkan deteksi titik fitur FAST dan deskripsi fitur BRIEF yang telah dimodifikasi agar tahan terhadap rotasi. ORB menghasilkan deskriptor biner yang memudahkan pencocokan dan memiliki kecepatan puluhan kali lebih tinggi dibandingkan algoritma SIFT, sehingga sangat cocok untuk aplikasi pemrosesan waktu nyata. Namun, ORB tidak dapat langsung digunakan untuk pencocokan gambar tanpa risiko tingkat ketidakcocokan yang tinggi dan akurasi rendah[12].

ORB digunakan untuk menetapkan korespondensi antara titik kunci yang diidentifikasi dalam wilayah kotak pembatas melalui deteksi tepi.

Untuk mencapai invariansi rotasi, algoritma ORB menggunakan teknik *intensity centroid* dengan menghitung momen spasial dari distribusi intensitas dalam area lokal sekitar titik fitur. Pada setiap keypoint yang terdeteksi :

1. Perhitungan momen patch

Momen spasial m_{pq} dari patch gambar dihitung menggunakan rumus:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

Dalam rumus [13] diatas, di tuliskan bahwa x dan y mewakili nilai koordinat piksel dari petak gambar, dan menunjukkan nilai skala abu-abu. Centroid skala abu-abu dapat ditentukan sebagai berikut :

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

Titik ini merepresentasikan pusat distribusi intensitas dalam patch. Dimana m_{00} mewakili orde nol dari patch gambar, dan m_{01} dan m_{10} mewakili momen orde pertama, setiap orde sesuai dengan titik pusat. Orientasi ini kemudian digunakan untuk memutar patch ke posisi kanonik sebelum ekstraksi deskriptor:

$$\theta = \arctan(m_{01}, m_{10})$$

Orientasi ini kemudian digunakan untuk memutar patch ke posisi kanonik sebelum ekstraksi deskriptor, sehingga pola sampling pada patch dapat disesuaikan dengan arah orientasi tersebut agar deskriptor menjadi tahan terhadap rotasi. Deskriptor BRIEF sendiri terdiri dari string biner 0 dan 1 yang dibuat dengan memilih secara acak pasangan titik piksel di sekitar titik kunci sebagai pusat lingkaran, kemudian membandingkan intensitas kedua titik tersebut. Untuk setiap set fitur n pengujian biner di lokasi (x_i, y_i) , mendefinisikan matrix $2 \times n$:

$$Q = \begin{pmatrix} x_1, x_2, \dots, x_n \\ y_1, y_2, \dots, y_n \end{pmatrix}$$

Menggunakan orientasi titik fitur θ , dan matrix rotasi yang sesuai R_θ , ORB membangun versi yang dikendalikan [13] yaitu Q_θ of $Q: Q_\theta = R_\theta Q$, maka deskriptor titik fitur invarian rotsi pada ORB menjadi :

$$g_n(p, \theta) = f_n(p) | (x_i, y_i) \in Q_\theta$$

2.2.5 Point of Interest

Point of Interest (POI) adalah representasi digital dari entitas geografis di dunia nyata yang sangat relevan dengan kehidupan sehari-hari penduduk kota, seperti toko, restoran, dan rumah sakit. Setiap POI biasanya memiliki beberapa atribut, termasuk nama, koordinat, dan kategori. Karena POI menunjukkan hubungan antara lokasi geografis dan aktivitas manusia yang terjadi di sekitarnya,

data POI banyak digunakan dalam pengembangan digital twin dan dalam mendukung pengelolaan kota.



Gambar 2.2.5 Point of Interest

Point of Interest (interest points) memiliki peranan yang sangat penting dalam cara manusia mempersepsikan bentuk dan menjadi fitur utama yang banyak dimanfaatkan dalam berbagai tugas pemrosesan dan pemahaman citra. Tugas-tugas tersebut meliputi kalibrasi kamera, pendaftaran citra, pengenalan objek, rekonstruksi 3D, interaksi manusia-komputer, mengemudi otonom, pengenalan wajah, analisis lalu lintas, pelacakan multi-objek, bedah berbantuan robot, pengawasan, dan realitas tertambah. Tidak hanya itu, telah banyak dilakukan penelitian mengenai deteksi titik ketertarikan dalam konteks tertentu, seperti deteksi blob, fitur lokal invarian, pelacakan visual, kontur tepi, pencocokan citra, hingga rekonstruksi 3D [14].

Dalam fitur khusus Point of Interest (PoI) sebuah gambar yang menonjol dari yang lainnya. PoI sering kali mewakili puncak, objek, atau tepi dari objek dalam foto. "Ketertarikan" di sini berarti bahwa lokasi tersebut dapat memiliki fitur yang tidak berubah meskipun terjadi perubahan pada gambar, seperti saat pengambilan gambar, perubahan ukuran, atau perubahan intensitas pencahayaan. Metode identifikasi dan ekstraksi membantu dalam mengidentifikasi dan mengkodekan informasi penting untuk mendukung pemrosesan dan segmentasi gambar [15].

PoI [16] ditentukan dengan mengacu pada nilai maksimum dalam fungsi energi, dengan cara memindahkan titik acak awal ke lokasi dengan energi tertinggi di sekitarnya, yang selanjutnya membantu membentuk area dengan fitur-fitur yang diperlukan untuk proses segmentasi.

Penentuan PoI berdasarkan fungsi energi adalah sebagai berikut:

Biarkan $E(x, y)$ menjadi fungsi energi dari gambar.

Titik ketertarikan P ditentukan dengan mencari nilai maksimum dalam $E(x, y)$.

- 1) Inisialisasi titik N Acak

$$P_0 = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

Dimana n adalah jumlah titik $((x_i, y_i) \in [0, width] \times [0, height])$

- 2) Ulangi lebih dari t iterasi

Untuk setiap titik $(x_i, y_i) \in p_t$:

- a) Penentuan lingkungan $N(x_i, y_i)$
- b) Temukan lokasi dengan energi paling banyak di lingkungan sekitar :

$$(x'_i, y'_i) = \operatorname{argmax}(E(x, y)) \text{ dengan } (x, y) \in N(x_i, y_i)$$

- c) Perbarui lokasi baru untuk point:

$$(x_i, y_i) = (x'_i, y'_i)$$

- 3) Hasil akhir

$P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ setelah perulangan t.

2.2.6 Euclidean Distance

Metode jarak Euclidean umumnya diterapkan untuk menghitung jarak antara titik data dalam algoritma K-means [17] karena merupakan metode perhitungan jarak yang paling mudah dipahami, berasal dari rumus jarak antara dua titik dalam ruang Euclidean. Jarak ini mengukur jarak absolut atau garis lurus terpendek antara dua titik dalam ruang multidimensi, baik dalam dua dimensi maupun tiga dimensi, dan dapat dianggap sebagai jarak sebenarnya antara dua titik tersebut [18]. Namun, ketika diterapkan langsung pada data deret waktu, karakteristik unik dari deret waktu tersebut dapat hilang, sehingga mempengaruhi hasil pengelompokan. Dalam eksperimen [19], jarak Euclidean digunakan untuk

menghitung skor kesamaan dengan mendapatkan bobot skor melalui fungsi Gaussian, dan hasil akhir diperoleh berdasarkan bobot tersebut. Rumus perhitungan perluasan dimensi lebih tinggi sebagai berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^n \beta_i (x_i - y_i)^2}$$

Dalam ruang n-dimensi, terdapat dua titik yaitu x dan y [20] Dimana $x(i)$ adalah koordinat ke-1 dari titik x , sama dengan, $y(i)$ adalah koordinat ke-2 dari titik y , n mewakili dimensi, dan $dist$ mewakili jarak antara dua titik.

Dalam eksperimen [19], jarak Euclidean digunakan untuk menghitung skor kesamaan antara data. Metode perhitungannya dilakukan dengan memperoleh bobot skor melalui fungsi Gaussian, dan hasil akhirnya ditentukan berdasarkan bobot tersebut. Proses perhitungannya adalah sebagai berikut:

$$W = a \cdot \exp\left[-\frac{(d_i - b)^2}{2c^2}\right] \quad (4)$$

$$Y = \frac{\sum_{i=1}^k W_i \cdot y_i}{\sum_{i=1}^k W_i} \quad (5)$$

Dalam persamaan (4), a merupakan tinggi kurva, b adalah offset garis tengah kurva pada sumbu x , dan C adalah setengah lebar puncak kurva. Sedangkan dalam persamaan (5), w_i menunjukkan bobotnya, y adalah hasil prediksi, y_i merupakan nilai hasil yang sebenarnya, dan K menunjukkan jumlah data yang digunakan untuk mengevaluasi kebenaran data.

Jarak Euclidean juga digunakan untuk menentukan jarak terpendek antara dua titik dalam ruang Euclidean atau jarak antara node dalam suatu klaster, dan dapat dihitung dalam berbagai dimensi. Berdasarkan [21], jarak Euclidean $d_{x,y}$ untuk titik dua dimensi dapat ditentukan sebagai berikut:

$$d_{x,y} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

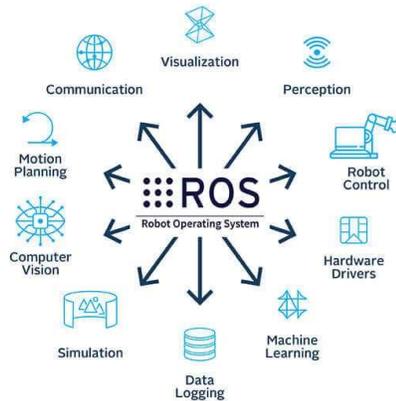
Rumus ini merupakan penerapan langsung dari teorema Pythagoras, di mana jarak antara dua titik dihitung sebagai panjang sisi miring segitiga siku-siku yang dibentuk oleh selisih koordinat kedua titik tersebut. $d(x, y)$ merupakan jarak Euclidean antara titik x dan y dan y , sedangkan x_1, x_2 koordinat titik pertama pada sumbu x dan y , dan untuk y_1, y_2 koordinat titik kedua pada sumbu x dan y .

2.2.7 ROS (Robot Operating System)

ROS (Robot Operating System) merupakan perangkat lunak sumber terbuka yang menyediakan berbagai pustaka dan alat bantu untuk pengembangan, pengendalian, dan visualisasi sistem robotik [22]. Platform ini mempermudah proses pemrograman serta koordinasi antar komponen dalam sistem robot, seperti yang diterapkan pada AUV, di mana ROS digunakan untuk mengintegrasikan beragam perangkat lunak, termasuk sistem pendeteksian dan modul output PPM berbasis Arduino [23].

ROS mendukung pemodelan struktur fisik robot melalui URDF serta memungkinkan integrasi antara robot fisik dan virtual. Makalah ini juga membahas penerapan ROS dalam sistem teleoperasi berbasis Virtual Reality (VR), yang memungkinkan pengendalian robot dari jarak jauh di berbagai skenario, seperti area berbahaya, lingkungan bawah laut, hingga aplikasi medis seperti operasi jarak jauh [24]. ROS sendiri merupakan arsitektur perangkat lunak tingkat atas yang tersedia dalam dua versi, yakni ROS1 dan ROS2. Desain arsitekturnya ditujukan untuk mempermudah proses analisis serta verifikasi interaksi antar komponen sistem, sehingga memastikan fungsi sistem berjalan secara optimal [25].

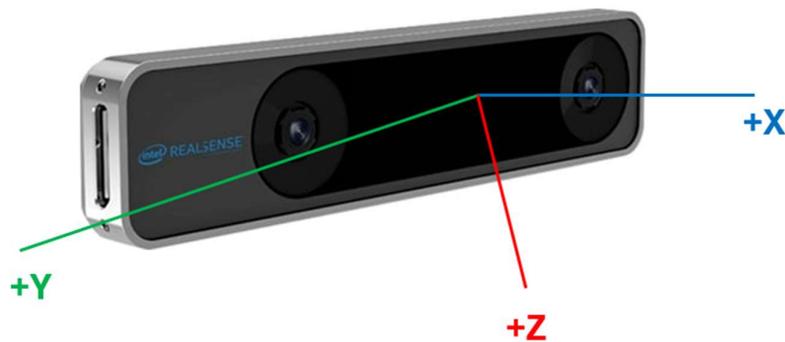
ROS1 dan ROS2 adalah middleware open-source yang banyak digunakan dalam bidang robotika. ROS1 menawarkan pustaka dan sistem komunikasi antar node, sementara ROS2 dikembangkan sebagai proyek baru dengan struktur arsitektur yang diperbarui untuk memenuhi tuntutan aplikasi industri. ROS 2 bukanlah versi lanjutan dari ROS 1, melainkan sebagai proyek terpisah dengan pendekatan arsitektur yang sepenuhnya baru. Pengembangannya mencakup rekonstruksi menyeluruh, mulai dari komponen inti tingkat rendah hingga pustaka aplikasi pada tingkat yang lebih tinggi. ROS 1 dan ROS 2 adalah platform yang efektif dalam pengembangan testbed robotik. Meskipun demikian, testbed robotik yang dirancang untuk kebutuhan Industri 4.0 menghadirkan dua persyaratan tambahan [26].



Gambar 2.2.7 ROS (Robot Operating System)

2.2.8 T265 Realsense

Intel RealSense T265 merupakan perangkat pelacak berukuran kecil dan efisien dalam konsumsi daya, yang dirancang untuk memberikan estimasi posisi serta orientasi tiga dimensi secara langsung (real-time) dengan tingkat akurasi yang tinggi. Perangkat ini memadukan kamera stereo fisheye terkalibrasi, unit IMU, dan prosesor Intel® Movidius™ Myriad™ 2 VPU yang secara internal menangani proses Visual SLAM dan Visual Odometry [27]. Dengan implementasi Visual Inertial Odometry (VIO) yang optimal [28], T265 dapat menjadi alternatif pengganti sistem lokalisasi visual seperti ORB-SLAM2 pada robot berukuran kecil, dengan keunggulan utama pada efisiensi energi dibandingkan metode SLAM konvensional yang memerlukan sumber daya komputasi besar [29].



Gambar 2.2.8 T265 Realsense

Penggunaan Intel RealSense T265 sebagai pengganti GPS pada drone indoor menawarkan banyak keunggulan karena perangkat ini menyediakan

kemampuan Visual-Inertial Odometry (VIO) dan Visual Simultaneous Localization and Mapping (V-SLAM). Hal ini sangat cocok untuk navigasi di lingkungan dalam ruangan di mana sinyal GPS tidak tersedia atau sangat tidak stabil. T265 menggabungkan dua kamera fish-eye dan sensor IMU untuk menghasilkan estimasi posisi dan orientasi enam derajat kebebasan secara real-time tanpa perlu sensor eksternal atau infrastruktur tambahan seperti GPS. Melalui algoritma V-SLAM, T265 mampu membangun peta lingkungan sekaligus melacak posisi drone secara akurat dalam ruang tiga dimensi, sehingga memungkinkan drone melakukan navigasi otomatis, menjaga posisi, dan fungsi lain yang biasanya bergantung pada GPS dalam aplikasi outdoor.

Keunggulan utama T265 dibandingkan GPS indoor adalah kemampuannya bekerja tanpa bergantung pada sinyal eksternal, melainkan menggunakan data visual dan inertial dari lingkungan sekitar dan pergerakan drone sendiri. Namun, keberhasilan penggunaan T265 juga tergantung pada kondisi lingkungan, seperti pencahayaan yang memadai dan keberadaan fitur visual yang cukup, karena performanya dapat menurun jika kondisi sangat gelap, terlalu terang, atau fitur visual minim. Selain itu, meskipun secara umum akurat, T265 dapat mengalami drift posisi jika tracking visual terganggu. Dengan demikian, Intel RealSense T265 merupakan solusi efektif dan praktis untuk substitusi GPS pada drone indoor, memungkinkan navigasi dan pengendalian yang lebih baik di dalam ruangan, asalkan integrasi sistem dan kondisi lingkungan diperhatikan dengan baik.

2.2.8.1 Modul Pelacakan Intel® RealSense™ T261

Tabel di bawah ini menjelaskan komponen utama yang membentuk SKU modul

Component	T261
Movidius Myriad 2 ASIC	✓
Fisheye Imagers	✓
IMU	✓

Gambar 2.2.8.1. Deskripsi SKU Produk Modul Pelacakan

2.2.8.2 Modul Pelacakan Intel® RealSense™ T261

H – FOV Horizontal, V – FOV Vertikal, D – FOV Diagonal, X – Panjang, Y – Lebar, Z – Ketebalan.

Imaging Module	Intel® RealSense™ Tracking Module T261
Baseline (mm)	64±0.15
Left/Right Fisheye Imagers	OV9282
Shutter Type	Global
Fisheye FOV (degrees)	D:173
Module Dimensions (mm)	X=93.35 (+0.15 -0.25) Y=17.60±0.15 Z=7.13±0.30

Gambar 2.2.8.2. Properti Modul Pelacakan

2.2.8.3 Unit Pengukuran Inersia

IMU merupakan sistem dalam paket untuk mendeteksi percepatan dalam 3 dimensi dan rotasi dalam 3 dimensi.

Parameter	Properties
Degrees of Freedom	6
Acceleration Range	±4g
Accelerometer Sample Rate	62.5Hz
Gyroscope Range	±2000 Deg/s
Gyroscope Sample Rate	200Hz

Gambar 2.2.8.3. Spesifikasi Pengukuran Inersia

2.2.8.4 Fisheye Imagers

Fisheye imagers digunakan dalam proses produksi data 6DoF yang dialirkan ke platform host. Imagers menyediakan gambar monokrom pada 30FPS.

Properti Sensor Gambar Fisheye

Parameter	Camera Sensor Properties
Active Pixels	848 X 800
Sensor Aspect Ratio	1.06
Format	8bit, 10-bit RAW
Filter Type	IR Cut Filter
Focus	Fixed
Shutter Type	Global Shutter
Signal Interface	MIPI CSI-2, 2 X Lanes

Gambar 2.2.8.4. Properti Sensor Gambar Fisheye

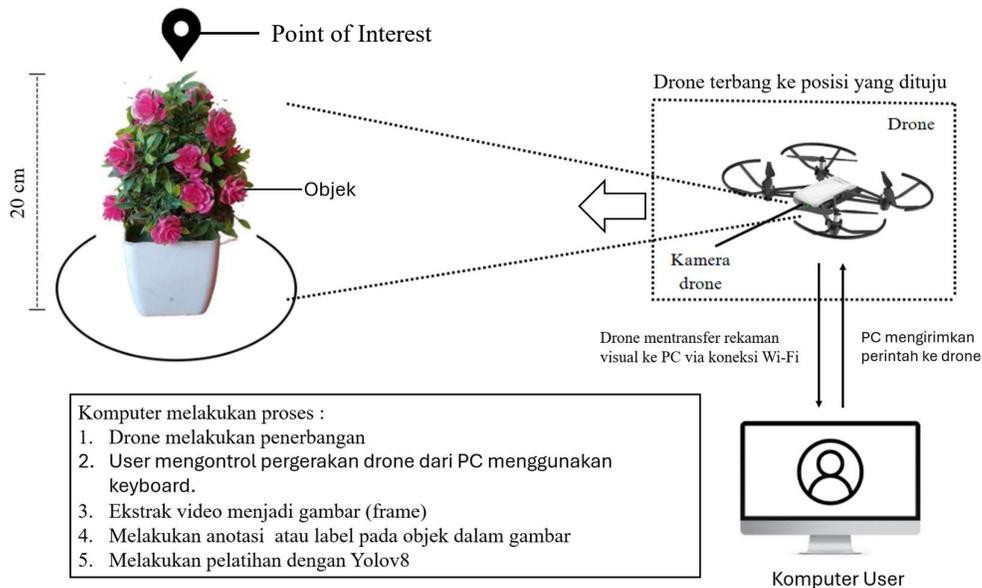
Halaman ini sengaja dikosongkan

BAB 3

METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian dilakukan untuk membuat pengembangan point of interest pada drone di dalam ruangan dengan cara mendeteksi objek yang memiliki banyak feature, disini peneliti memilih objek bunga sebagai acuan dalam memperkirakan posisi drone. Kemudian drone diperintahkan untuk terbang oleh operator menuju posisi target dengan mengandalkan pergerakan posisi dengan keyboard. Pendeteksian objek dilakukan menggunakan kamera depan, Dimana proses pengolahan gambar untuk mendeteksi objek, hingga menghasilkan perintah pergerakan drone semua melalui computer user. Visualisasi navigasi drone secara umum, dapat dilihat pada Gambar 3.1.



Gambar 3.1 Visualisasi Navigasi Drone Indoor Menuju Target Dengan Bantuan Operator Dalam Deteksi Objek.

3.2 Sistem Komunikasi Antara Drone dan Pengendali

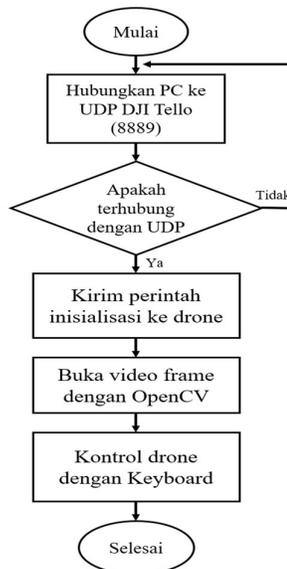
Dalam penelitian ini, drone yang digunakan adalah DJI Ryze Tello, sebagaimana ditampilkan pada Gambar 3.3. Drone ini merupakan salah satu jenis mini drone komersial yang dilengkapi dengan kamera, yang memungkinkan

pengambilan gambar maupun video. Pengoperasian drone ini dapat dilakukan melalui aplikasi di smartphone maupun melalui perintah pemrograman menggunakan komputer.

Drone Tello didukung oleh Tello SDK (Software Development Kit), yaitu seperangkat antarmuka pemrograman yang disediakan oleh DJI untuk memungkinkan pengendalian drone melalui kode program yang dikirimkan melalui jaringan Wi-Fi berbasis UDP (User Datagram Protocol). Dalam proses komunikasinya, DJI Tello memanfaatkan beberapa port UDP yang memiliki fungsi khusus, seperti untuk mengirimkan perintah kontrol, melakukan streaming video, dan menerima informasi status drone. Status ini mencakup berbagai data dari sensor internal yang terpasang pada drone.



Gambar 3.2 Drone DJI Ryze Tello



Gambar 3.2.1 Mekanisme komunikasi antara drone dan komputer

3.3 Pengujian Stabilitas Drone

Penelitian ini dilakukan untuk mengevaluasi kestabilan vertikal drone DJI Tello dalam kondisi hover rendah di dalam ruangan. Drone ini dilengkapi dengan sensor visual bawah (downward vision sensor) yang memungkinkannya untuk mempertahankan posisi saat terbang tanpa menggunakan GPS. Penelitian difokuskan pada pengujian respon drone terhadap gerakan naik-turun kecil secara berulang serta kestabilannya saat diam di udara dalam waktu yang cukup lama.

Pengujian dilakukan dengan menggunakan skrip Python yang mengontrol drone melalui library djitellopy. Drone dihubungkan ke komputer melalui jaringan WiFi, kemudian diinisialisasi dan dicek status baterainya. Setelah itu, drone diterbangkan secara otomatis sekitar 50 cm untuk mendekati lantai, namun tetap berada dalam ketinggian aman bagi sensor bawah.



Gambar 3.3 Pengujian drone terbang awal

Selanjutnya, dilakukan simulasi gerakan vertikal berupa naik dan turun sebanyak 10 siklus dengan variasi ketinggian, setelah terbang pada ketinggian 50 cm, drone turun sebanyak 20 cm, hingga berada di posisi 30 cm.



Gambar 3.3.1 Drone turun 20 cm

Kemudian drone akan naik pada ketinggian 30 cm, hingga berada berada pada posisi 60 cm.



Gambar 3.3.2 Drone naik 30 cm

Pada gambar 3.2.3 drone turun sebanyak 30 cm, pada ketinggian 30 cm dari permukaan drone terlihat cukup stabil dalam mempertahankan posisi tanpa perintah gerakan tambahan.



Gambar 3.2.3 Drone turun 30 cm

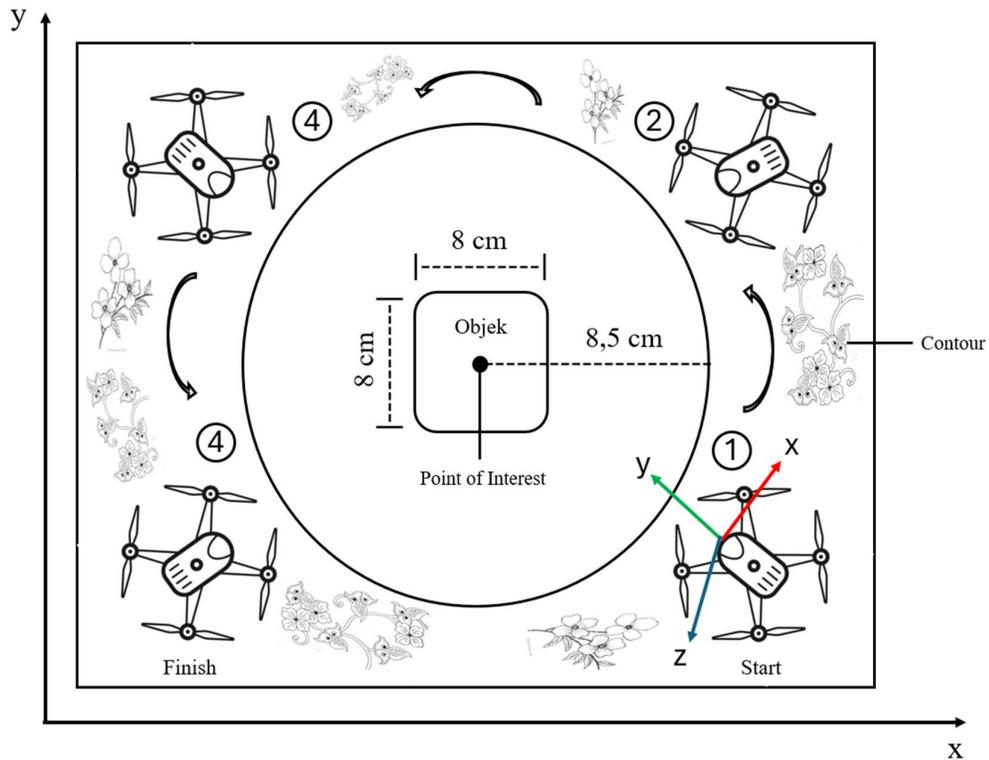
Selama proses pengujian, permukaan lantai diberi alas bermotif seperti kertas bercorak atau karpet berpola. Hal ini penting karena DJI Tello mengandalkan pola visual pada permukaan untuk menjaga kestabilan posisi. Jika permukaan terlalu polos atau mengkilap, sensor visual akan kesulitan mendeteksi pergerakan relatif terhadap lantai, sehingga dapat menyebabkan drone kehilangan posisi dan mengalami pergeseran (drift). Dengan memberikan alas bermotif, sistem visual positioning dapat bekerja lebih optimal, sehingga drone tetap stabil saat melakukan hover maupun saat bergerak naik-turun.

3.4 Pergerakan Drone Pada Point of Interest

Posisi target dalam konteks ini merujuk pada titik awal drone yang berada tepat di depan objek dan dijadikan sebagai set point atau acuan awal pergerakan. Pengaturan posisi drone dilakukan dengan mengirimkan perintah kecepatan dari komputer ke drone untuk mengatur arah gerak berdasarkan koordinat estimasi

posisi saat ini. Pengendalian dilakukan pada tiga sumbu, yaitu sumbu-X (gerakan ke kanan/kiri), sumbu-Y (maju/mundur), dan sumbu-Z (naik/turun).

Sebelum memulai navigasi, terlebih dahulu dilakukan perencanaan jalur (*trajectory planning*) di dalam ruangan. Dalam eksperimen ini, digunakan satu objek utama yang diletakkan di tengah sebagai pusat lintasan. Setelah drone lepas landas (*takeoff*) dan mencapai ketinggian sekitar 30 cm, drone mulai bergerak ke arah kanan (sumbu-X) dan mengikuti lintasan melingkar mengelilingi objek. Pergerakan dilakukan seperti arah jarum jam melalui beberapa titik sudut hingga drone kembali ke posisi finish yang berdekatan dengan titik awal. Setelah lintasan selesai, drone kemudian berhenti dan melakukan pendaratan (*landing*).



Gambar 3.3 Visualisasi Pergerakan Drone Menuju Titik Target

3.5 Deteksi Point of Interest

Point of Interest dalam penelitian ini merujuk pada benda yang dijadikan titik menarik atau pusat perhatian sistem. Untuk keperluan penelitian ini, digunakan objek nyata berupa bunga tiga dimensi (3D) yang memungkinkan untuk tetap

terdeteksi dari berbagai sudut pandang kamera drone. Bunga dipilih sebagai objek deteksi karena secara alami memiliki banyak titik fitur visual yang kompleks, sehingga mendukung proses identifikasi dan pelacakan. Dimensi bunga yang digunakan dalam penelitian ini adalah sekitar $8 \times 8 \times 20$ cm untuk panjang, lebar, dan tingginya. Objek pada gambar 3.4 merupakan objek fisik yang digunakan selama proses penelitian :



Gambar 3.4 Objek yang digunakan sebagai objek target

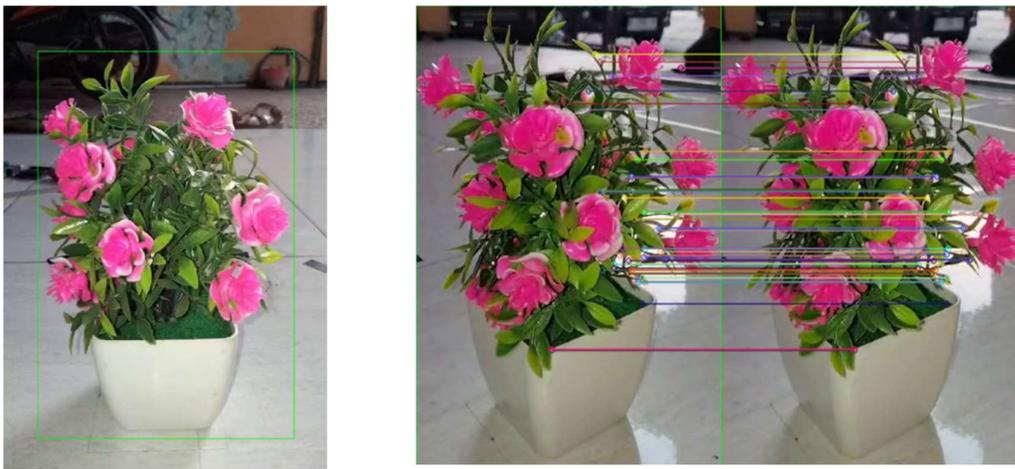
Deteksi objek menggunakan YOLOv8 berbasis deep learning dipilih dalam penelitian ini karena pendekatannya yang efisien dengan sistem deteksi satu tahap (*one-stage detection*), yang meningkatkan baik akurasi maupun kecepatan. Dengan versi anchor-free, model ini dapat memprediksi posisi pusat objek secara langsung, sehingga memberikan akurasi yang lebih tinggi, terutama untuk objek dengan ukuran atau bentuk yang tidak biasa.

Dalam penelitian ini, bunga dipilih sebagai target deteksi karena memiliki karakteristik tiga dimensi, yang memungkinkan drone mendeteksi dan mencapai posisi yang diinginkan secara lebih presisi. Selain itu, YOLOv8 juga mampu mendeteksi beberapa objek sekaligus dalam satu frame, yang turut mendukung efisiensi dan akurasi proses deteksi.

3.6 Pengujian Sistem Feature Matching

Selama proses operasi, kami merekam video yang berisi objek-objek dengan anotasi. Langkah pertama yang dilakukan adalah mempersiapkan video dan file anotasi, yang berisi informasi mengenai posisi objek. Selanjutnya, kami memuat video dan membaca file anotasi untuk menandai objek pada setiap frame.

Pada setiap frame, citra dikonversi ke format RGB, dan anotasi digambar menggunakan kotak pembatas (bounding box) yang didefinisikan oleh dua koordinat (x, y) dan label objek. Setelah itu, kami mengekstrak keypoint dan deskriptor menggunakan metode ORB, lalu mencocokkan deskriptor dari frame video dengan deskriptor dari citra Point of Interest (POI) menggunakan BFMatcher. Hasil pencocokan ditampilkan dengan menghubungkan titik-titik kunci yang sesuai, sehingga memungkinkan analisis terhadap seberapa baik algoritma mendeteksi dan mencocokkan objek yang dianotasi, seperti yang ditunjukkan pada gambar.



Gambar 3.5 Pengujian Sistem Feature Matching

3.7 Evaluasi Deteksi dan Identifikasi Feature Matching

Pada tahap ini, proses dimulai dengan memasukkan video yang berisi objek target Bunga. Alih-alih memproses setiap frame secara penuh, sistem menerapkan teknik ekstraksi keyframe, yaitu memilih frame-frame penting yang merepresentasikan perubahan atau informasi signifikan dalam video. Hal ini bertujuan untuk mengurangi redundansi dan mempercepat proses analisis dengan memfokuskan pemrosesan hanya pada frame yang relevan.

Pada keyframe awal, sistem melakukan deteksi tepi (edge detection) untuk mengidentifikasi kontur objek. Setelah objek dikenali, proses anotasi dilakukan untuk menandai objek tersebut, biasanya dengan bounding box atau label identitas.

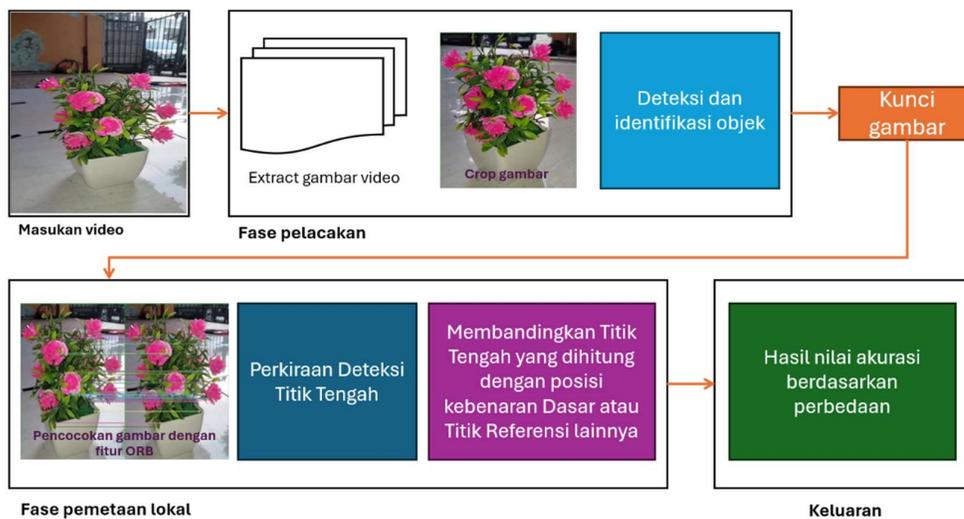
Frame ini selanjutnya digunakan sebagai referensi untuk proses pelacakan di frame-frame berikutnya. Setelah itu, gambar objek yang telah diidentifikasi akan di-crop untuk mendapatkan bagian penting dari citra. Gambar hasil crop tersebut kemudian disimpan sebagai gambar kunci (key image) yang akan digunakan sebagai referensi dalam tahap pencocokan visual.

Pada tahap akhir, nilai akurasi dihitung berdasarkan perbedaan posisi antara hasil estimasi dan referensi menggunakan jarak Euclidean. Jarak Euclidean adalah metrik yang digunakan untuk menghitung jarak langsung antara dua titik dalam ruang, dan sering diterapkan dalam visi komputer dan pembelajaran mesin untuk menilai perbedaan atau kesamaan antara posisi. Dalam penelitian ini, jarak Euclidean digunakan untuk membandingkan fitur-fitur yang terdeteksi pada objek target dengan titik referensi yang telah diketahui, sehingga memungkinkan drone untuk memperkirakan posisi target secara akurat selama navigasi otonom.

Di sini, nilai d merepresentasikan jarak Euclidean, atau jarak garis lurus antara dua titik dalam ruang dua dimensi, yang dihitung dengan rumus:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

di mana (x_1, y_1) adalah koordinat titik referensi, dan (x_2, y_2) adalah koordinat titik hasil deteksi objek. Dengan menerapkan rumus ini, jarak antara posisi aktual dan posisi estimasi dapat dihitung secara presisi, sehingga memberikan dasar kuantitatif dalam mengevaluasi akurasi pelacakan objek.

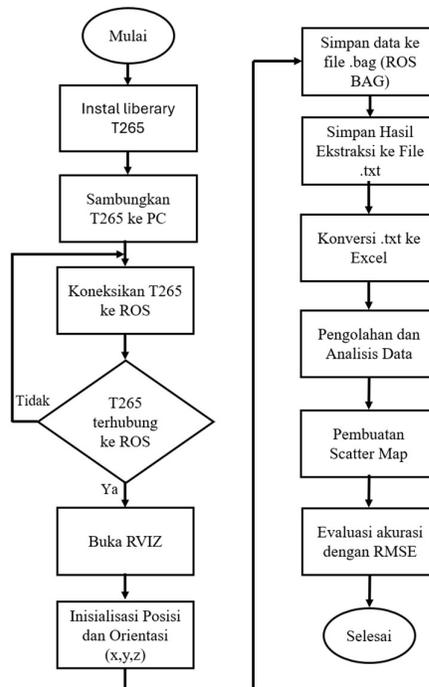


Gambar 3.4 Evaluasi deteksi dan identifikasi feature matching

3.8 Intel RealSense T265 sebagai Sistem Navigasi Indoor

Dalam penelitian ini, kamera Intel RealSense T265 digunakan sebagai pengganti GPS untuk merekam lintasan pergerakan sistem secara akurat di lingkungan indoor. Eksperimen dilakukan dengan menggerakkan sistem secara mengelilingi target berbentuk lingkaran berdiameter 10 cm, 20 cm, dan 30 cm, pada dua kondisi lantai yang berbeda, yaitu dengan dan tanpa kontur. Kontur yang diletakkan di lantai berfungsi sebagai referensi visual statis untuk membantu validasi dan evaluasi akurasi estimasi posisi yang diperoleh selama pergerakan.

Kamera T265 dipasang menghadap ke bawah pada ketinggian sekitar 10 cm dari permukaan lantai, dan merekam data posisi dalam koordinat x, y, z serta orientasi (roll, pitch, yaw) secara kontinu dengan frekuensi 30 Hz. Data yang direkam dalam format .bag (ROS BAG) kemudian diekstrak menjadi file .txt, dikonversi ke format Excel, dan diolah lebih lanjut untuk analisis kuantitatif serta visualisasi lintasan dalam bentuk scatter map. Scatter map ini digunakan untuk menunjukkan seberapa representatif lintasan hasil estimasi terhadap jalur lingkaran ideal yang ditempuh sistem.



Gambar 3.6 Blok diagram Desain Eksperimen Pengujian Kamera T265

Halaman ini sengaja dikosongkan

BAB 4

HASIL DAN PEMBAHASAN

Bagian ini membahas hasil dan analisis dari pengujian yang difokuskan pada pengolahan citra gambar yang ditangkap oleh drone. Pengolahan data dilakukan menggunakan metode feature matching, dengan langkah-langkah yang meliputi pengumpulan data, pra- pemrosesan, mengenali, mencocokkan, dan melacak objek target. Analisis dilakukan untuk menelusuri hubungan antara kualitas citra, efektivitas pencocokan fitur, serta kontribusinya terhadap stabilitas pelacakan selama observasi.

4.1 Pengumpulan Dataset

Pengumpulan dataset bunga dilakukan dengan memanfaatkan platform Roboflow, di mana gambar-gambar bunga diunggah, dianotasi, dan diproses lebih lanjut untuk keperluan pelatihan model deteksi atau klasifikasi citra bunga. Proses pengumpulan dataset bunga dilakukan dengan memanfaatkan platform Roboflow, proses ini diawali dengan pengambilan gambar secara langsung menggunakan perangkat rekaman video dari drone. Gambar maupun cuplikan video yang diambil difokuskan pada satu jenis bunga dengan variasi warna terbatas, yaitu merah dan putih pada kelopak, serta hijau pada bagian daun, sesuai dengan kebutuhan penelitian dalam pengembangan sistem deteksi objek.

Setelah proses akuisisi citra, data visual diunggah ke Roboflow untuk dilakukan anotasi. Anotasi dilakukan secara menyeluruh pada bagian bunga dan daun, baik secara manual maupun semi-otomatis, guna memastikan objek target terdefinisi dengan jelas. Proses ini bertujuan untuk memberi label pada setiap elemen penting yang akan digunakan sebagai acuan pencocokan dan pelacakan dalam tahap selanjutnya.

Data yang telah dianotasi kemudian melalui proses pra-pemrosesan dan augmentasi untuk meningkatkan kualitas serta memperluas variasi citra. Tahapan ini meliputi penghilangan latar belakang yang tidak relevan, penyesuaian ukuran citra (640x640 piksel), serta penerapan augmentasi seperti rotasi, flipping, dan translasi objek. Langkah-langkah ini bertujuan untuk memperkuat daya tahan

sistem terhadap perubahan sudut pandang dan pencahayaan selama proses observasi dan pelacakan menggunakan video drone.

Dataset yang digunakan dalam penelitian ini terdiri dari 60 citra yang telah dianotasi dan diproses menggunakan Python, kemudian diklasifikasikan ke dalam tiga subset utama dengan proporsi umum: 60% untuk pelatihan (training), 20% untuk validasi (validation), dan 20% untuk pengujian (testing). Setelah proses pelabelan selesai, dataset diekspor dalam format kompatibel seperti YOLOv5 dan digunakan untuk melatih model deteksi objek YOLOv8 dalam lingkungan Python. Model yang telah dilatih ini kemudian diimplementasikan untuk mendeteksi objek target secara otomatis pada setiap frame yang diambil oleh kamera drone.

Model deteksi yang telah dilatih menggunakan algoritma YOLO selanjutnya dievaluasi dengan menggunakan metrik performa akurasi. Dalam penelitian ini, akurasi dihitung berdasarkan pendekatan *Euclidean Distance* untuk mengukur jarak antara titik prediksi dan titik referensi (ground truth) pada objek yang terdeteksi. Evaluasi ini dilakukan untuk menilai sejauh mana efektivitas model dalam mengidentifikasi dan melacak objek bunga secara otomatis, termasuk saat diaplikasikan pada rekaman video yang diambil oleh drone dalam skenario pengujian di lapangan.

4.2 Pengujian Deteksi Berdasarkan Nilai Error Posisi

Pengujian ini dilakukan untuk mengevaluasi tingkat akurasi sistem dalam mendeteksi posisi objek berdasarkan dua titik referensi pada setiap frame, yang masing-masing ditandai dengan koordinat (x_1, y_1) dan (x_2, y_2) . Hasil data pengujian yang ditampilkan pada Tabel I menunjukkan nilai Error yang mengindikasikan perbedaan posisi antara kedua titik tersebut. Semakin besar nilai Error, maka semakin besar kemungkinan terjadi ketidaksesuaian atau ketidakakuratan posisi. Di antara semua frame, frame 16 dan 19 menunjukkan nilai yang relatif stabil baik dari segi posisi titik maupun kecepatan. Hal ini menunjukkan bahwa pada frame-frame tersebut, pergerakan objek berlangsung dengan mulus serta memiliki variasi atau gangguan yang minimal. Frame 16 memiliki nilai Error terendah yaitu sebesar 7,07%, yang menandakan bahwa posisi kedua titik tersebut cukup akurat dan menunjukkan sedikit penyimpangan.

TABEL 4.2 HASIL DATA PENGUJIAN

Frame	Jarak titik koordinat (pixel)				Error (%)	V (fps)	Error (%)
	X1	Y1	X2	Y2			
16	218	513	217	506	7,07107	921,954	9,21954
19	296	529	298	513	16,1245	707,107	7,07107
20	272	531	270	510	21,095	670,82	6,7082
27	242	528	238	507	21,3776	707,107	7,07107
29	228	530	213	498	35,3412	806,226	8,06226
41	187	551	187	519	32	424,264	4,24264

Nilai Euclidean error bervariasi di setiap frame, mulai dari 9,21% (pada frame 59) hingga 36,12% (pada frame 39). Nilai error yang tinggi, seperti pada frame 39, menunjukkan tingkat ketidakakuratan posisi yang lebih besar. Kolom "Error Y (%)" merepresentasikan kesalahan atau ketidakakuratan pada sumbu vertikal (Y), dengan nilai yang berfluktuasi antara 36,49% hingga 102,94%. Nilai error yang tinggi pada sumbu Y (seperti 102,94% pada frame 59) menunjukkan adanya ketidakakuratan yang signifikan dalam pengukuran vertikal, yang kemungkinan disebabkan oleh kecepatan objek yang tinggi atau perubahan posisi objek. Pada kecepatan yang sangat tinggi (seperti pada frame 59), error pada sumbu Y mencapai nilai tertinggi (102,94%), yang mengindikasikan bahwa pengukuran koordinat Y menjadi lebih sulit dilakukan secara akurat pada kondisi tersebut.

TABEL 4.2.2 HASIL DATA PENGUJIAN

Frame	Jarak titik koordinat (pixel)				Error	V (fps)	Error (Y)
	X1	Y1	X2	Y2			
18	240	521	246	506	16,1555	3649,66	36,4966
39	186	542	183	506	36,1248	3832,75	38,3275
43	215	551	218	527	24,1868	3712,14	37,1214
55	200	502	209	486	18,3576	3667,42	36,6742
57	228	520	212	490	34	4148,49	41,4849
59	264	516	271	510	9,21954	10294,2	102,942

4.3 Evaluasi Kinerja Pelacakan Berdasarkan Kecepatan Gerak

Dalam penelitian ini, kami menganalisis 60 frame untuk mengkaji hubungan antara kecepatan drone (dalam satuan frame per detik atau fps) dengan tingkat error (%) dalam proses pemantauan atau pelacakan objek oleh drone.

Seperti yang ditampilkan pada "Gambar 5", grafik tersebut menggambarkan sejauh mana error terjadi saat drone bergerak dengan kecepatan yang berbeda-beda. Nilai error yang lebih tinggi menunjukkan tingkat akurasi pelacakan objek yang lebih rendah, yang berarti bahwa semakin cepat pergerakan drone, semakin sulit bagi drone untuk “melihat” atau mengenali objek dengan efektif.

Pada kecepatan sedang (antara 424,26 hingga 921,95 fps), nilai error berfluktuasi tetapi masih berada dalam rentang tertentu, yang menunjukkan bahwa pada kecepatan ini, drone masih dapat melacak objek dengan cukup baik meskipun terdapat sedikit variasi dalam akurasi. Hal ini mengindikasikan bahwa pada kecepatan sedang, drone mampu mengikuti objek tanpa mengalami kesalahan yang signifikan.

Namun, pada akhir grafik terlihat lonjakan tajam dalam tingkat error pada kecepatan sekitar 10294,17 fps. Pada titik ini, nilai error meningkat secara drastis, menandakan bahwa pada kecepatan tinggi, drone kesulitan untuk melacak objek secara akurat. Kecepatan yang terlalu tinggi membuat sistem kamera atau program feature matching tidak mampu mengenali objek dengan tepat.



Gambar 4.4 Evaluasi Kinerja Pelacakan Berdasarkan Kecepatan Gerak

4.4 Evaluasi Akurasi Pose Kamera T265

Evaluasi ini dilakukan untuk menilai ketepatan estimasi posisi kamera Intel RealSense T265 saat mengelilingi target berbentuk lingkaran berdiameter 10 cm yang dilengkapi kontur visual kontras tinggi. Selama proses pengambilan data,

kamera dipasang pada drone dan bergerak membentuk lintasan melingkar 360° mengelilingi target. Data posisi dua dimensi (x, y) yang dihasilkan T265 kemudian dibandingkan terhadap posisi pusat target (ground truth), yang telah ditentukan sebelumnya berdasarkan nilai rata-rata koordinat p_x dan p_y .

Untuk menghitung tingkat akurasi, digunakan pendekatan matematis berdasarkan jarak Euclidean antara posisi estimasi dengan titik pusat target. Dari jarak r_i yang diperoleh, dilakukan pengurangan terhadap nilai radius ideal sebesar 0,05 meter (setengah dari diameter target). Selisih tersebut mencerminkan nilai error pada masing-masing titik. Dua parameter statistik digunakan dalam evaluasi ini: Mean Error yang menunjukkan rata-rata penyimpangan estimasi posisi dari radius ideal, dan Root Mean Square Error (RMSE) yang merepresentasikan tingkat deviasi keseluruhan, termasuk memperbesar dampak error ekstrem.

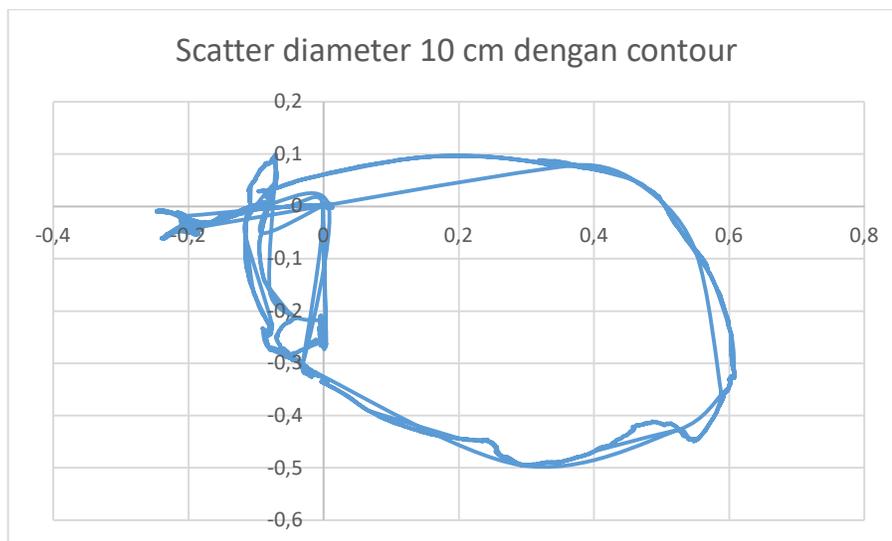
4.4.1 Hasil akurasi pose kamera T265 terhadap contour target dengan diameter 10 cm.

Tabel 4.1.1 akurasi diameter 10 cm dengan contour

Segmen	p_x	p_y	x_0	y_0	r_i (Jarak ke pusat)	Error ($r_i - 0.05$)	$Error^2$
1	0,11903	-0,01039	0,06728	0,15967	0,23874	0,18874	0,03562
2	0,00616	0,00623	0,06728	0,15967	0,17680	0,12680	0,01607
...
1000	0,44024	0,059884	0,06728	0,15967	0,43278	0,38278	0,14652
1001	0,4413	0,05986	0,06728	0,15967	0,43375	0,38375	0,14726
...
10196	0,00360	-0,26330	0,06728	0,15967	0,12163	0,07163	0,00513
10197	0,00360	-0,26325	0,06728	0,15967	0,12158	0,07158	0,00512
Mean Error						0,24231	
RMSE							0,27753

Hasil pengujian akurasi pose kamera Intel RealSense T265 terhadap kontur target berdiameter 10 cm menunjukkan bahwa estimasi posisi kamera

seringkali meleset dari posisi sebenarnya, namun lebih terstruktur dan konsisten. Dari 10.197 sampel data yang dianalisis, diperoleh rata-rata error sebesar 0,24 meter, yang berarti posisi yang diperkirakan kamera meleset sekitar 24 cm dari pusat lingkaran. Selain itu, nilai RMSE sebesar 0,28 meter mengindikasikan adanya beberapa estimasi dengan error yang lebih besar, sehingga penyebaran error cukup luas. Error yang cukup besar ini kemungkinan disebabkan oleh keterbatasan sensor, kualitas deteksi kontur, atau kalibrasi kamera yang belum optimal. Pada gambar 4.4.1 merupakan hasil scatter.



Gambar 4.4.1 Scatter diameter 10 cm dengan contour

Berdasarkan scatter map hasil pengukuran dengan target lingkaran berdiameter 10 cm menggunakan kontur, terlihat bahwa distribusi titik estimasi posisi kamera Intel RealSense T265 tidak sepenuhnya membentuk pola lingkaran yang simetris. Terdapat penyimpangan yang mencolok, terutama pada area kiri atas dan kanan bawah, di mana bentuk kontur tampak melebar dan tidak konsisten. Hal ini menunjukkan adanya variasi error yang cukup besar, baik secara rata-rata maupun dalam sebaran data. Pola yang tidak rapat dan menyimpang dari bentuk lingkaran ideal mengindikasikan bahwa kamera mengalami kesulitan menjaga estimasi posisi yang stabil terhadap target kecil. Kemungkinan penyebabnya meliputi noise sensor, ketidaktepatan deteksi kontur, drift pada sistem odometri, atau kalibrasi yang belum optimal. Temuan visual ini sejalan dengan data statistik, di mana rata-rata error sebesar 0,24 meter dan RMSE 0,28 meter menunjukkan

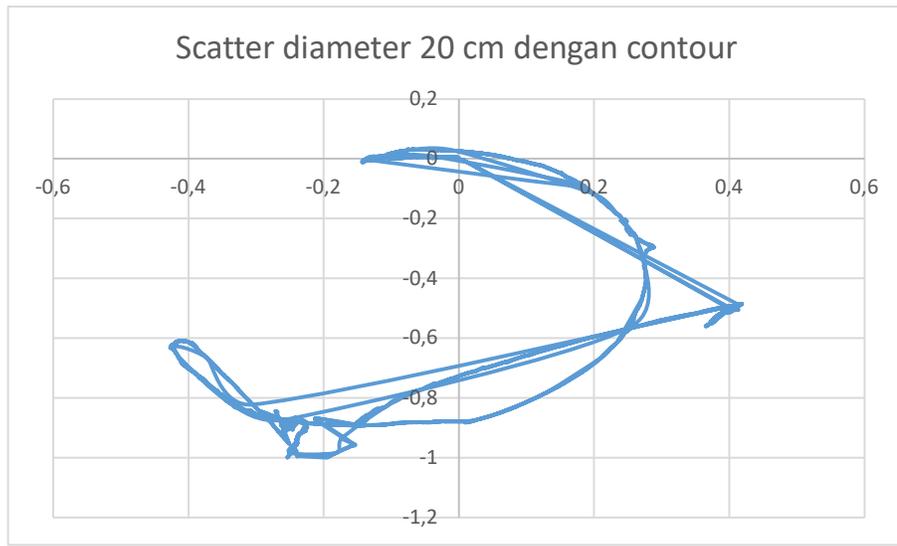
tingkat deviasi yang tinggi. Oleh karena itu, akurasi estimasi pose kamera terhadap target kecil tergolong rendah dan kurang dapat diandalkan untuk aplikasi presisi.

4.4.2 Hasil akurasi pose kamera T265 terhadap contour target dengan diameter 20 cm.

Tabel 4.2.2 akurasi diameter 20 cm dengan contour

Segmen	p_x	p_y	x_0	y_0	r_i (Jarak ke pusat)	Error ($r_i - 0.05$)	$Error^2$
1	- 0,13711	-0,00310	-0,01727	- 0,44991	0,46259	0,41259	0,17023
2	0,18621	-0,09340	-0,01727	- 0,44991	0,41048	0,36048	0,12995
...
1000	0,15922	-0,74602	-0,01727	- 0,44991	0,34472	0,29472	0,08686
1001	0,15804	-0,74736	-0,01727	- 0,44991	0,34527	0,29527	0,08719
...
8027	0,40618	-0,50527	-0,01727	- 0,44991	0,42706	0,37706	0,14217
8028	0,40635	-0,50521	-0,01727	- 0,44991	0,42722	0,37722	0,14229
Mean Error						0,38156	
RMSE							0,38685

Berdasarkan hasil pengukuran kontur pada objek berbentuk lingkaran berdiameter 20 cm, diperoleh data berupa koordinat pusat, titik-titik pengukuran, jarak tiap titik ke pusat, error terhadap radius ideal 0,05 m, serta kuadrat error untuk perhitungan RMSE. Nilai rata-rata error sebesar 0,38156 dan RMSE 0,38685, yang jauh lebih besar dari radius ideal, menunjukkan deviasi signifikan terhadap bentuk lingkaran sempurna. Variasi radius antara 0,34 m hingga 0,46 m mengindikasikan kemungkinan kesalahan satuan atau tingginya noise dalam pengukuran. Error yang cenderung besar dan positif memperkuat indikasi penyimpangan dari radius yang diharapkan. Selain itu, kontur yang terbentuk kemungkinan tidak simetris atau mengalami pergeseran pusat secara sistematis. Pada gambar 4.4.2 merupakan hasil scatter.



Gambar 4.4.2 Scatter diameter 20 cm dengan contour

Berdasarkan gambar scatter hasil pengukuran kontur pada objek berbentuk lingkaran berdiameter 20 cm, terlihat bahwa titik-titik pengukuran tidak membentuk pola lingkaran yang simetris atau konsisten. Sebagian besar titik terkumpul di area tertentu, sementara area lain tampak lebih kosong dan garis kontur tidak beraturan, menunjukkan deviasi signifikan dari bentuk lingkaran ideal. Selain itu, terdapat beberapa outlier yang menonjol keluar dari pola utama, mengindikasikan adanya noise tinggi atau gangguan dalam proses deteksi kontur dan akuisisi data. Kontur yang terbentuk juga tampak mengalami pergeseran dari pusat yang diharapkan, serta memiliki rentang nilai radius antara 0,34 hingga 0,46 meter, jauh melebihi radius ideal 0,1 meter untuk diameter 20 cm.

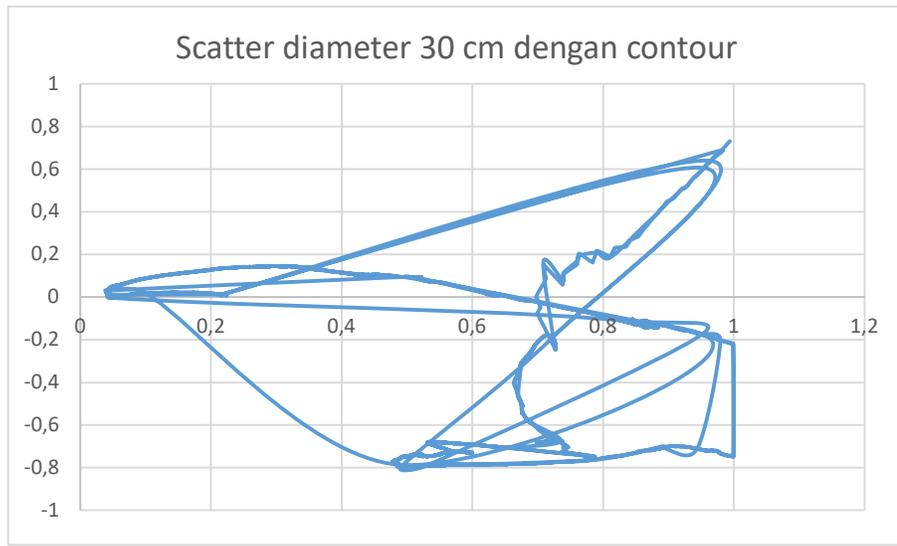
4.4.3 Hasil akurasi pose kamera T265 terhadap contour target dengan diameter 30 cm.

Tabel 4.3.3 akurasi diameter 30 cm dengan contour

Segmen	p_x	p_y	x_0	y_0	r_i (Jarak ke pusat)	Error ($r_i - 0.05$)	$Error^2$
1	0,04397	-0,00345	0,54969	- 0,25173	0,56337	0,51337	0,26355
2	0,95951	-0,18473	-0,01727	- 0,44991	0,41526	0,36526	0,13341
...
1000	0,94025	-0,16472	-0,01727	- 0,44991	0,40013	0,35013	0,12259
1001	0,94067	-0,16570	-0,01727	- 0,44991	0,40033	0,35033	0,12273
...
6700	0,99145	0,72302	-0,01727	- 0,44991	1,07018	1,02018	1,04078
6701	0,99390	0,73152	-0,01727	- 0,44991	1,07893	1,02893	1,05871
Mean Error						0,38063	
RMSE							0,40052

Tabel menunjukkan hasil pengukuran kontur dengan 6701 segmen pengukuran, dimana setiap segmen memiliki koordinat titik pengukuran (p_x , p_y), koordinat pusat (x_0 , y_0), jarak ke pusat (r_i), error terhadap nilai radius ideal, dan kuadrat error. Nilai jarak ke pusat (r_i) bervariasi secara signifikan, mulai dari sekitar 0.40013 m hingga 1.07893 m, jauh melebihi radius ideal yang seharusnya 0.15 m (untuk diameter 30 cm). Hal ini mengindikasikan adanya deviasi yang sangat besar dari bentuk lingkaran ideal. Error yang dihitung ($r_i - 0.05$) juga konsisten positif dan besar, dengan nilai mean error mencapai 0.38063.

Nilai RMSE (Root Mean Square Error) sebesar 0.40052 menunjukkan tingkat ketidakpresisian yang tinggi dalam pengukuran kontur. Nilai ini hampir tiga kali lipat dari radius ideal objek, mengindikasikan bahwa bentuk kontur yang diukur sangat berbeda dari lingkaran sempurna dengan diameter 30 cm. Pada gambar 4.4.3 merupakan hasil scatter.



Gambar 4.4.3 Scatter diameter 30 cm dengan contour

Berdasarkan hasil scatter plot pengukuran kontur objek berdiameter 30 cm, terlihat bahwa distribusi titik sangat tidak simetris dan menyimpang dari bentuk lingkaran ideal. Titik-titik tersebar secara acak dan membentuk pola yang tidak teratur, dengan banyak deviasi ke arah dalam maupun luar dari pusat. Beberapa area menunjukkan penumpukan titik atau garis-garis saling berpotongan, menandakan adanya distorsi atau noise yang cukup tinggi. Pola ini konsisten dengan nilai RMSE dan error rata-rata yang besar dari hasil analisis numerik sebelumnya.

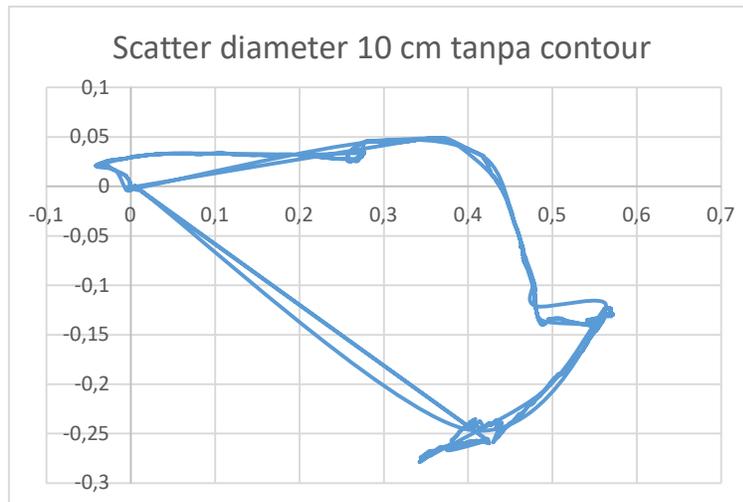
Ketidakteraturan tersebut menunjukkan adanya permasalahan dalam proses pengukuran, seperti kesalahan kalibrasi alat, gangguan lingkungan, atau bahkan drift yaitu pergeseran estimasi posisi secara bertahap akibat akumulasi kesalahan dalam sistem. Meski tidak ditemukan outlier ekstrem, penyimpangan bersifat menyeluruh dan sistematis.

4.4.4 Hasil akurasi pose kamera T265 tanpa contour target dengan diameter 10 cm.

Tabel 4.4.4 akurasi diameter 10 cm tanpa contour

Segmen	p_x	p_y	x_0	y_0	r_i (Jarak ke pusat)	Error ($r_i - 0.05$)	$Error^2$
1	0,34893	0,04820	0,31617	- 0,06255	0,11550	0,06550	0,00429
2	0,00556	-0,00010	0,31617	- 0,06255	0,31682	0,26682	0,07119
...
1000	0,00484	-0,00012	0,31617	- 0,06255	0,31753	0,26753	0,07157
1001	0,00483	-0,00012	0,31617	- 0,06255	0,31754	0,26754	0,0715
...
4004	0,42363	-0,25691	0,31617	- 0,06255	0,22208	0,17208	0,02961
4005	0,42363	-0,25688	0,31617	- 0,06255	0,22205	0,17205	0,02960
Mean Error						0,16803	
RMSE							0,18437

Analisis data diameter 10 cm tanpa contour cenderung kurang akurat karena scatter map yang dihasilkan tidak menunjukkan bentuk objek yang jelas atau pola lingkaran yang terdefinisi dengan baik. Ketidakjelasan pola ini menyebabkan kesulitan dalam menentukan batas diameter yang tepat, sehingga pengukuran menjadi rentan terhadap error dan variasi yang tinggi. Selain itu, distribusi titik yang tidak merata atau adanya outlier pada scatter map dapat menyebabkan bias dalam hasil analisis. Oleh karena itu, metode tanpa contour kurang cocok digunakan karena gambar scatter tidak beraturan. Pada gambar 4.4.4 merupakan hasil scatter.



Gambar 4.4.4 Scatter diameter 10 cm tanpa contour

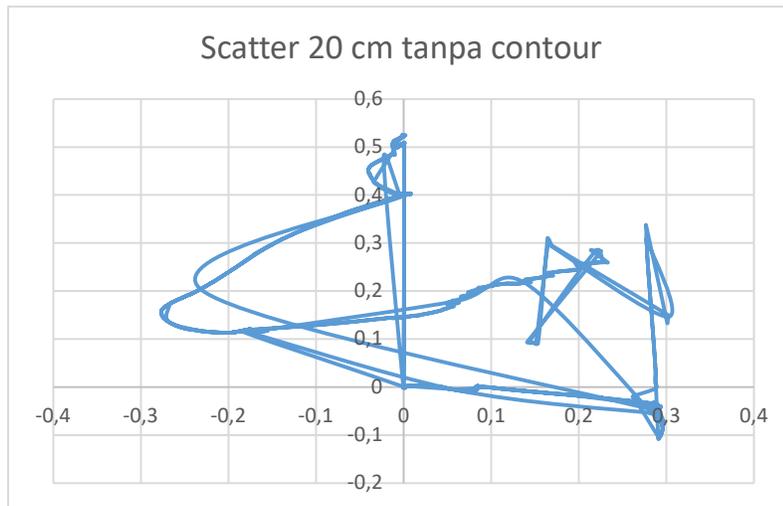
Berdasarkan hasil scatter plot dan tabel analisis numerik, dapat disimpulkan bahwa metode pengambilan data tanpa menggunakan kontur menghasilkan representasi bentuk objek yang kurang akurat. Scatter plot menunjukkan distribusi titik yang tidak beraturan dan tidak membentuk pola lingkaran yang jelas, sehingga menyulitkan identifikasi batas diameter secara visual. Hal ini diperkuat oleh analisis kuantitatif pada tabel, di mana nilai rata-rata error mencapai 0,16803 meter dan RMSE sebesar 0,18437 meter terhadap radius ideal 0,05 meter. Nilai error yang tinggi ini menunjukkan adanya deviasi signifikan dari bentuk lingkaran yang diharapkan, serta mengindikasikan bahwa banyak titik berada jauh dari posisi seharusnya.

4.4.5 Hasil akurasi pose kamera T265 tanpa contour target dengan diameter 10 cm.

Tabel 4.5.5 akurasi diameter 20 cm tanpa contour

Segmen	p_x	p_y	x_0	y_0	r_i (Jarak ke pusat)	Error ($r_i - 0.05$)	$Error^2$
1	0,00122	0,00023	0,03775	0,16405	0,16784	0,11784	0,01388
2	- 0,18412	0,11690	0,03775	0,16405	0,22682	0,17682	0,03126
...
1000	0,00169	0,00015	0,03775	0,16405	0,16782	0,11782	0,01388
1001	0,00169	0,00016	0,03775	0,16405	0,16781	0,11781	0,01388
...
3806	0,00066	0,50788	0,03775	0,16405	0,34581	0,29581	0,08750
3807	0,00063	0,50783	0,03775	0,16405	0,34577	0,29577	0,08748
Mean Error						0,20440	
RMSE							0,21685

Berdasarkan tabel data numerik, terlihat bahwa nilai Mean Error sebesar 0,20440 meter dan RMSE sebesar 0,21685 meter, yang menunjukkan bahwa deviasi jarak titik ke pusat terhadap radius ideal (0,1 meter untuk diameter 20 cm) sangat besar. Error ini mengindikasikan bahwa sebagian besar titik memiliki jarak lebih dari dua kali lipat dari radius ideal, yang memperkuat bahwa penyebaran titik sangat menyimpang dari bentuk lingkaran yang seharusnya. Beberapa nilai error bahkan mendekati 0,3 meter, menunjukkan adanya outlier signifikan dalam data. Dengan RMSE yang tinggi, dapat disimpulkan bahwa metode pengambilan tanpa kontur menghasilkan penyimpangan sistematis, bukan hanya error acak. Pada gambar 4.4.5 merupakan hasil scatter.



Gambar 4.4.5 Scatter diameter 20 cm tanpa contour

Scatter map yang ditampilkan pada Gambar 4.4.5 menunjukkan pola distribusi titik yang tidak membentuk bentuk lingkaran secara visual. Sebaran data terlihat acak, tidak simetris, dan terdapat banyak garis lintas antar titik yang menandakan inkonsistensi pengambilan data. Ketidakteraturan ini memperkuat indikasi bahwa metode tanpa kontur menghasilkan visualisasi bentuk yang tidak representatif terhadap objek sebenarnya. Titik-titik juga menyebar secara tidak merata dengan konsentrasi data yang tidak stabil, sehingga menyulitkan proses estimasi diameter dan bentuk. Pola tersebut menunjukkan kemungkinan adanya noise tinggi atau outlier, dan mendukung argumen bahwa pendekatan visual ini tidak dapat diandalkan untuk pengukuran presisi tanpa bantuan metode pemrosesan tambahan seperti filtering, smoothing, atau fitting.

4.4.6 Hasil akurasi pose kamera T265 tanpa contour target dengan diameter 30 cm.

Tabel 4.6.6 akurasi diameter 30 cm tanpa contour

Segmen	p_x	p_y	x_0	y_0	r_i (Jarak ke pusat)	Error ($r_i - 0.05$)	$Error^2$
1	2.081	-21.053	3.772	-10.181	Nan	Nan	Nan
2	0,0274	0,00997	3.772	-10.181	Nan	Nan	Nan
...
1000	21.800	-2.130	3.772	-10.181	Nan	Nan	Nan
1001	2.182	-21.307	3.772	-10.181	Nan	Nan	Nan
...
5221	2.434	-1.537	3.772	-10.181	Nan	Nan	Nan
5222	24.349	-1.536	3.772	-10.181	Nan	Nan	Nan
Mean Error						Nan	
RMSE							Nan

Analisis terhadap data diameter 30 cm tanpa kontur menunjukkan kegagalan sistem dalam menghitung jarak ke pusat objek, sebagaimana terlihat dari nilai NaN di seluruh kolom hasil. Hal ini menunjukkan bahwa metode tanpa kontur tidak hanya menghasilkan hasil yang tidak akurat, tetapi juga berisiko gagal sepenuhnya dalam proses analisis numerik ketika diterapkan pada objek berukuran besar. Ketidaksesuaian ini semakin menegaskan perlunya penggunaan kontur atau metode pendukung lain agar hasil yang diperoleh valid dan dapat diinterpretasikan secara ilmiah.

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN

Dari hasil percobaan dan Analisa yang dilakukan pada penelitian berjudul “PENGEMBANGAN POINT OF INTEREST PADA INDOOR DRONE MENGGUNAKAN FEATURE MATCHING” ini, penulis mendapatkan beberapa kesimpulan yang diharapkan dapat berguna untuk pengembangan penelitian ini.

5.1 Kesimpulan

Pendeteksian drone menggunakan metode ORB terbukti sangat membantu dalam mencocokkan gambar secara akurat. Hasil pengujian menunjukkan bahwa frame ke-16 memiliki tingkat akurasi tertinggi dengan nilai error hanya sebesar 7,07%, yang menandakan bahwa posisi titik pada frame tersebut relatif stabil. Sebaliknya, frame ke-39 menunjukkan nilai error yang jauh lebih tinggi, yakni mencapai 36,12%, yang mengindikasikan adanya ketidakakuratan posisi yang cukup signifikan. Variasi nilai error juga tampak pada frame-frame lainnya, seperti pada frame ke-59 yang mencatatkan error tertinggi pada sumbu Y sebesar 102,94%.

Berdasarkan hasil pengujian T265 realSense untuk mendapatkan map scatter dengan diameter 10 cm, 20 cm, dan 30 cm, baik dalam kondisi menggunakan kontur maupun tanpa kontur, ditemukan bahwa estimasi posisi yang dihasilkan oleh kamera Intel RealSense T265 masih belum sepenuhnya akurat. Kontur di lantai memang membantu memberikan acuan visual statis, sehingga distribusi titik estimasi menjadi sedikit lebih terstruktur. Namun demikian, titik-titik tersebut tetap tidak membentuk pola lingkaran yang rapi dan simetris, dan beberapa di antaranya menyimpang jauh dari posisi yang seharusnya.

Tanpa adanya kontur, hasil estimasi posisi justru menjadi lebih tidak teratur dan sulit dikenali bentuk lintasannya. Hal ini menunjukkan bahwa meskipun T265 mampu digunakan sebagai pengganti GPS dalam skenario indoor, akurasi sangat bergantung pada keberadaan fitur visual di lingkungan sekitar. Dalam konteks penelitian ini yang berfokus pada Point of Interest (POI), penggunaan T265 tanpa bantuan kontur menyebabkan terjadinya drift atau penyimpangan posisi dari lintasan ideal, sehingga mengurangi ketepatan pelacakan target secara keseluruhan.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] R. A. Firmansyah, R. Mardiyanto, and T. A. Sardjono, "Improvement Monocular ORB SLAM for Indoor Drone 3D localization using Inertial Navigation System," in *Proceedings of the 2024 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 348–353. doi: 10.1109/IAICT62357.2024.10617662.
- [2] *The 23rd International Conference on Advanced Communications Technology: "On-Line security in Pandemic Era!" : Phoenix Park, Pyeongchang, Korea (South), (On-line Conference), Feb. 07-10, 2021 : proceeding & journal*. The Institute of Electrical and Electronics Engineers, Inc., 2021.
- [3] Z. Wang, Z. Li, L. Cheng, and G. Yan, "An improved ORB feature extraction and matching algorithm based on affine transformation," in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 1511–1515. doi: 10.1109/CAC51589.2020.9327165.
- [4] K. Sukvichai, N. Thongton, and K. Yajai, "Implementation of a Monocular ORB SLAM for an Indoor Agricultural Drone," in *2023 3rd International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics, ICA-SYMP 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 45–48. doi: 10.1109/ICA-SYMP56348.2023.10044953.
- [5] M. Y. Mustar, B. S. Tryatmojo, S. G. Kusuma, Y. K. M. Putra, H. A. Sasono, and R. O. Wiyagi, "Exploring the Potential of Intel Realsense T265 Camera as Visual Odometry and Tracking System for Autonomous Navigation of Four-Wheel Omnidirectional Mobile Robot," in *Proceedings - ICE3IS 2024: 4th International Conference on Electronic and Electrical Engineering and Intelligent System: Leading-Edge Technologies for Sustainable Societies*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 115–120. doi: 10.1109/ICE3IS62977.2024.10775680.
- [6] Fahmizal, D. Afidah, S. Istiqphara, and N. S. Abu, "Interface Design of DJI Tello Quadcopter Flight Control," *Journal of Fuzzy Systems and Control*, vol. 1, no. 2, pp. 49–54, Aug. 2023, doi: 10.59247/jfsc.v1i2.35.
- [7] M. D. B. Pranoto, M. I. Sani, and M. I. Sari, "Aerial Object Tracking System on Micro Quadrotor Drone for Crowd Detection in Small-Scale Area," 2023, pp. 992–1006. doi: 10.2991/978-2-38476-132-6_84.
- [8] C. C. Zhang, K. Zhang, R. Ni, H. Liu, and J. Shen, "Unleashing the Potential of Machine Learning: An Exploration of State-of-the-Art Algorithms and

- Real-World Applications in Computer Vision,” in *Proceedings - 2023 Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 422–425. doi: 10.1109/CSCE60160.2023.00075.
- [9] C. A. Canitan, J. E. C. Rosal, and D. I. E. Hisola, “Development of a Computer Vision Application for Copra Meat Cooking Detection and Classification Using YOLOv8 Architecture,” in *6th IEEE International Conference on Artificial Intelligence in Engineering and Technology, IICAJET 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 442–447. doi: 10.1109/IICAJET62352.2024.10730597.
- [10] J. M. Phillips, S. Shue, and J. M. Conrad, “A Design Architecture for Steering an Autonomous All-Terrain Vehicle Using Object Tracking via Computer Vision and YOLOv8,” in *2024 IEEE 21st International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT, HONET 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 242–247. doi: 10.1109/HONET63146.2024.10822935.
- [11] W. L. Du and X. L. Tian, “An Automatic Evaluation Platform for Feature Matching Algorithms Based on an Orbital Optical Pushbroom Stereo Imaging System,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 359–363, Mar. 2018, doi: 10.1109/LGRS.2017.2787743.
- [12] Z. Wang, Z. Li, L. Cheng, and G. Yan, “An improved ORB feature extraction and matching algorithm based on affine transformation,” in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 1511–1515. doi: 10.1109/CAC51589.2020.9327165.
- [13] D. Ran, C. Huang, Y. Zhang, and W. Ren, “Fast relocation method for indoor mobile robots based on ORB feature matching,” in *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology, AINIT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 64–70. doi: 10.1109/AINIT61980.2024.10581417.
- [14] J. Chen *et al.*, “An Automatic Point-of-Interest Generation Method Based on Open Street View Images,” in *International Conference on Big Data and Information Analytics, BigDIA*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 52–59. doi: 10.1109/BigDIA63733.2024.10808290.
- [15] J. Jing, T. Gao, W. Zhang, Y. Gao, and C. Sun, “Image Feature Information Extraction for Interest Point Detection: A Comprehensive Review,” *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 4, pp. 4694–4712, Apr. 2023, doi: 10.1109/TPAMI.2022.3201185.

- [16] T. P. Huynh and H. X. Huynh, “ESS-PoI: Energy-Driven Semantic Segmentation with Points of Interest,” in *Proceedings - 2025 5th Asia Conference on Information Engineering, ACIE 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 111–116. doi: 10.1109/ACIE64499.2025.00025.
- [17] L. Jiang, W. Wang, S. Zheng, S. Xu, J. Mao, and L. Ding, “A Novel Data Edge Mining Algorithm With Euclidean Distance Weighted Optimization for Integrated Energy Station,” in *ITOEC 2023 - IEEE 7th Information Technology and Mechatronics Engineering Conference*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 67–71. doi: 10.1109/ITOEC57671.2023.10291381.
- [18] X. Liu and F. Li, “Research on Fault Detection Method Based on Improved Euclidean Distance Control,” in *ICIEA 2022 - Proceedings of the 17th IEEE Conference on Industrial Electronics and Applications*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 106–111. doi: 10.1109/ICIEA54703.2022.10006143.
- [19] R. Li, S. Yang, and W. Xie, “Cardiovascular disease prediction model based on logistic regression and euclidean distance,” in *Proceedings - 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 711–715. doi: 10.1109/AEMCSE51986.2021.00147.
- [20] F. Cheng, “A Comparative Study of the Performance of Spark-based k-Means Algorithm Based on Euclidean Distance and Manhattan Distance,” in *Proceedings - 2024 3rd International Conference on Big Data, Information and Computer Network, BDICN 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 15–20. doi: 10.1109/BDICN62775.2024.00010.
- [21] M. A. P. Putra, D. S. Kim, and J. M. Lee, “DB-BiLSTM: Euclidean Distance-Based Sensor Data Prediction for IoT Applications,” in *International Conference on ICT Convergence*, IEEE Computer Society, 2021, pp. 814–817. doi: 10.1109/ICTC52510.2021.9620877.
- [22] N. Zivkovic, J. Vidakovic, S. Mitrovic, and M. Lazarevic, “Implementation of Dual Quaternion-based Robot Forward Kinematics Algorithm in ROS,” in *2022 11th Mediterranean Conference on Embedded Computing, MECO 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/MECO55406.2022.9797160.
- [23] Y. K. Mali, V. U. Rathod, S. Dargad, V. N. Kapure, N. Vyawahare, and S. Gajarlewar, “Development of ROS(Robotic Operating System) for Fault Detection of Underwater Cables,” in *IEEE Region 10 Humanitarian Technology Conference, R10-HTC*, Institute of Electrical and Electronics

- Engineers Inc., 2023, pp. 956–961. doi: 10.1109/R10-HTC57504.2023.10461858.
- [24] S. Vairavasamy, J. Noel Innocent, M. J. Hamin, S. Ahmed, N. Seenu, and R. Mm Dean, “Simulation And Real Time Of VR Controlled Robotic Manipulator Using ROS,” in *IBSSC 2022 - IEEE Bombay Section Signature Conference*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IBSSC56953.2022.10037520.
- [25] N. Shaukat *et al.*, “Trustworthy ROS Software Architecture for Autonomous Drones Missions: From RoboChart Modelling to ROS Implementation,” in *MESA 2024 - 20th International Conference on Mechatronic, Embedded Systems and Applications, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/MESA61532.2024.10704818.
- [26] Q. D. Nguyen, S. Dhouib, Y. Huang, and P. Bellot, “An Approach to Bridge ROS 1 and ROS 2 Devices into an OPC UA-based Testbed for Industry 4.0,” in *1st IEEE Industrial Electronics Society Annual On-Line Conference, ONCON 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ONCON56984.2022.10126783.
- [27] M. Y. Mustar, B. S. Tryatmojo, S. G. Kusuma, Y. K. M. Putra, H. A. Sasono, and R. O. Wiyagi, “Exploring the Potential of Intel Realsense T265 Camera as Visual Odometry and Tracking System for Autonomous Navigation of Four-Wheel Omnidirectional Mobile Robot,” in *Proceedings - ICE3IS 2024: 4th International Conference on Electronic and Electrical Engineering and Intelligent System: Leading-Edge Technologies for Sustainable Societies*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 115–120. doi: 10.1109/ICE3IS62977.2024.10775680.
- [28] *2020 International Conference on Unmanned Aircraft Systems : ICUAS'20 : June 9-12, 2020 (moved to September 1-4, 2020), Divani Caravel Hotel, Athens, Greece GR-16121*. IEEE, 2020.
- [29] Libor. Přeučil, Sven. Behnke, and Miroslav. Kulich, *European Conference on Mobile Robots : Prague, Czech Republic 2019 : conference proceedings : September 4-6, 2019*. IEEE, 2019.

Halaman ini sengaja dikosongkan

LAMPIRAN

Pengujian Stabilitas Drone

```
from djitellopy import Tello
import time

def main():
    tello = Tello()
    tello.connect()
    print(f"Battery: {tello.get_battery()}%")

    time.sleep(5)
    tello.takeoff()
    time.sleep(2)

    # Turun 50 cm agar mendekati lantai tapi tetap stabil
    tello.move_down(50)
    time.sleep(2)

    print("Simulasi naik-turun kecil selama 10 siklus...")
    for _ in range(10):
        tello.move_up(30)
        time.sleep(1)
        tello.move_down(30)
        time.sleep(1)
        tello.move_up(20)
        time.sleep(1)
        tello.move_down(20)
        time.sleep(1)
```

```

print("Hover selama 5 menit (~1 meter dari lantai)...")
time.sleep(300) # 5 menit

tello.land()
print("Mendarat dengan aman.")

if __name__ == "__main__":
    main()

```

Program Pergerakan Drone Pada Point of Interest

```

from djitellopy import Tello
import cv2, math, time

tello = Tello()
tello.connect()

tello.streamon()
frame_read = tello.get_frame_read()

tello.takeoff()

while True:
    img = frame_read.frame
    cv2.imshow("drone", img)

    key = cv2.waitKey(1) & 0xff
    if key == 27: # ESC
        break
    elif key == ord('w'):
        tello.move_forward(30)

```

```

elif key == ord('s'):
    tello.move_back(30)
elif key == ord('a'):
    tello.move_left(30)
elif key == ord('d'):
    tello.move_right(30)
elif key == ord('e'):
    tello.rotate_clockwise(30)
elif key == ord('q'):
    tello.rotate_counter_clockwise(30)
elif key == ord('r'):
    tello.move_up(30)
elif key == ord('f'):
    tello.move_down(30)

tello.land()

```

Program Pengujian dengan YOLO

```

import cv2
from ultralytics import YOLO
import numpy as np

# Load the YOLOv8n model
model = YOLO('D:\TESIS BISMILLAH\yolov8n.pt')

video_path = 'D:\TESIS BISMILLAH\deteksibunga .mp4'

# Create a VideoCapture object
cap = cv2.VideoCapture(video_path)

```

```

# Check if the video opened successfully
if not cap.isOpened():
    print("Error: Could not open video.")
    exit()

# Initialize the ORB detector
orb = cv2.ORB_create()

# Read the first frame
ret, frame_before = cap.read()
results = model(frame_before)

# Initialize previous center
prev_center = None

# Accessing the results for the first frame
for result in results:
    boxes = result.boxes # Bounding boxes
    for box in boxes:
        x1, y1, x2, y2 = box.xyxy[0] # Coordinates
        print(box.cls)

        if box.cls == 58:
            cv2.rectangle(frame_before, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255,
0), 1)
            cropped_before = frame_before[int(y1):int(y2), int(x1):int(x2)]

# Calculate the center of the bounding box
prev_center = ((x1 + x2) / 2, (y1 + y2) / 2)

```

```

# Read and display the video frames
while True:

    # Capture frame-by-frame
    ret, frame = cap.read()

    # If the frame was not retrieved successfully, break the loop
    if not ret:
        break

    # Run inference on the current frame
    results = model(frame)

    # Accessing the results
    for result in results:
        boxes = result.boxes # Bounding boxes
        for box in boxes:
            x1, y1, x2, y2 = box.xyxy[0] # Coordinates
            print(box.cls)

            if box.cls == 58:
                cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 1)
                cropped_now = frame[int(y1):int(y2), int(x1):int(x2)]

                # Detect keypoints and descriptors
                kp1, des1 = orb.detectAndCompute(cropped_now, None)
                kp2, des2 = orb.detectAndCompute(cropped_before, None)

                # Initialize Brute Force Matcher
                bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)

```

```

# Match descriptors
matches = bf.match(des1, des2)

# Sort matches by distance
matches = sorted(matches, key=lambda x: x.distance)

# Draw matches
frame_matches = cv2.drawMatches(cropped_now, kp1, cropped_before,
kp2, matches[:50], None,
flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)

# Calculate the center of the current bounding box
current_center = ((x1 + x2) / 2, (y1 + y2) / 2)

# Calculate pixel displacement if previous center exists
if prev_center is not None:
    displacement = np.sqrt((current_center[0] - prev_center[0])** 2 +
(current_center[1] - prev_center[1])** 2)
    print(f"Displacement: {displacement:.2f} pixels")

# Update previous center to current center
prev_center = current_center

# Display the frame
cv2.imshow('Video', frame)

# Display the matches and cropped frames
cv2.imshow('Matches', frame_matches)
cv2.imshow('Cropped frame', cropped_now)

```

```
# Press 'q' to exit the video display
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

```
cropped_before = cropped_now
```

```
# Release the video capture object
cap.release()
# Close all OpenCV windows
cv2.destroyAllWindows()
```

Program rosbag ke .txt.

```
1 import rosbag
2 |
3 bag_file = '/home/ubuntu-desk/30cm tanpa conture.bag'
4 topic_name = '/camera/odom/sample'
5
6 output_file = '30 cm tanpa contur new_output.txt'
7
8 with open(output_file, 'w') as f:
9     f.write("time px py pz ox oy oz ow\n")
10
11     with rosbag.Bag(bag_file, 'r') as bag:
12         for topic, msg, t in bag.read_messages(topics=[topic_name]):
13             p = msg.pose.pose.position
14             o = msg.pose.pose.orientation
15             f.write(f"{t.to_sec()}, {p.x}, {p.y}, {p.z}, {o.x}, {o.y}, {o.z}, {o.w}\n")
16
17 print(F"Data berhasil diekspor ke: {output_file}")
```

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Maharani Wisudawati Swista lahir di Surabaya, Jawa Timur, pada tahun 2000. Penulis menyelesaikan pendidikan Sarjana (S1) di Universitas PGRI Adi Buana Surabaya pada Program Studi Teknik Elektro, dan lulus pada tahun 2022. Selama masa studi sarjana, penulis menunjukkan minat yang besar terhadap bidang robotika dan sistem kontrol, yang menjadi dasar untuk pengembangan karier akademiknya. Setelah menyelesaikan studi sarjana, penulis melanjutkan pendidikan Magister (S2) di Institut Teknologi Sepuluh Nopember (ITS), Jurusan Teknik Elektro. Dalam perjalanan pendidikan magisternya, penulis aktif terlibat dalam penelitian di bidang robotika dan sistem kontrol, dengan fokus utama pada pengembangan teknologi navigasi drone. Penulis berhasil menyelesaikan studi magisternya pada tahun 2025. Untuk berkomunikasi lebih lanjut, penulis dapat dihubungi melalui email: swistamaharani@gmail.com.