

TUGAS AKHIR - EE 184801

**PENGEMBANGAN SISTEM AUTOPILOT PESAWAT
TANPA AWAK MENGGUNAKAN ALGORITMA GENETIKA
UNTUK MEMINIMALISIR ZONA *BLANK SPOT* PADA
APLIKASI PEMETAAN WILAYAH**

Muhammad Ichlasul Salik
NRP 0711164000021

Dosen Pembimbing
Ronny Mardiyanto, ST., MT., Ph.D.
Dr. Ir. Djoko Purwanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



TUGAS AKHIR - EE 184801

**PENGEMBANGAN SISTEM AUTOPILOT PESAWAT
TANPA AWAK MENGGUNAKAN ALGORITMA GENETIKA
UNTUK MEMINIMALISIR ZONA *BLANK SPOT* PADA
APLIKASI PEMETAAN WILAYAH**

Muhammad Ichlasul Salik
NRP 0711164000021

Dosen Pembimbing
Ronny Mardiyanto, ST., MT., Ph.D.
Dr. Ir. Djoko Purwanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020



FINAL PROJECT - EE 184801

***DEVELOPMENT OF UNMANNED AERIAL VEHICLE
AUTOPILOT SYSTEM USING GENETIC ALGORITHM TO
MINIMIZE BLANK SPOT ZONE IN AERIAL MAPPING
APPLICATION***

Muhammad Ichlasul Salik
NRP 0711164000021

Supervisor(s)
Ronny Mardiyanto, ST., MT., Ph.D.
Dr. Ir. Djoko Purwanto, M.Eng.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology and Informatics Intelligent
Institut Teknologi Sepuluh Nopember
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Pengembangan Sistem Autopilot Pesawat Tanpa Awak Menggunakan Algoritma Genetika untuk Meminimalisir Zona *Blank Spot* pada Aplikasi Pemetaan Wilayah**” 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, Juni 2020

Muhammad Ichlasul Salik
NRP. 0711 16 4000 0021


**PENGEMBANGAN SISTEM AUTOPILOT PESAWAT
TANPA AWAK MENGGUNAKAN ALGORITMA
GENETIKA UNTUK MEMINIMALISIR ZONA
BLANK SPOT PADA APLIKASI PEMETAAN
WILAYAH**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I


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

NIP. 198101182003121003

SURABAYA

Juli, 2020

**PENGEMBANGAN SISTEM AUTOPILOT PESAWAT
TANPA AWAK MENGGUNAKAN ALGORITMA
GENETIKA UNTUK MEMINIMALISIR ZONA
BLANK SPOT PADA APLIKASI PEMETAAN
WILAYAH**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing II



Dr. Ir. Djoko Purwanto, M.Eng.
NIP. 196512111990021002

SURABAYA
Juli, 2020

PENGEMBANGAN SISTEM AUTOPILOT PESAWAT TANPA AWAK MENGGUNAKAN ALGORITMA GENETIKA UNTUK MEMINIMALISIR ZONA *BLANK SPOT* PADA APLIKASI PEMETAAN WILAYAH

Nama : Muhammad Ichlasul Salik
Pembimbing I : Ronny Mardiyanto, ST., MT., Ph.D.
Pembimbing II : Dr. Ir. Djoko Purwanto, M.Eng.

ABSTRAK

Dalam tugas akhir ini akan dilakukan optimasi terhadap sistem autopilot pada UAV dalam melakukan misi pemotretan lahan dari udara. Optimasi dilakukan dengan memberikan kemampuan kepada UAV dalam menganalisa zona *blank spot* yang dihasilkan selama misi pemotretan. Data koordinat *blank spot* akan menjadi suatu rute penerbangan baru untuk UAV dalam melakukan pemotretan ulang. Untuk menghasilkan rute penerbangan baru dilakukan analisis koordinat *blank spot* dengan menerapkan algoritma genetika dalam melakukan penyusunan jalur penerbangan tersebut. Dengan menerapkan algoritma genetika diharapkan akan didapatkannya rute penerbangan baru yang menghasilkan jarak tempuh total terpendek dalam melakukan pemotretan ulang. Dengan demikian waktu dan daya baterai yang dibutuhkan untuk menghasilkan foto peta yang sempurna akan lebih sedikit dan efisien. Hasil dari pengujian yang didapatkan pada tugas akhir ini adalah sistem autopilot yang dapat menganalisa zona *blank spot* dalam misi pemetaan wilayah dan melakukan optimasi terhadap rute penerbangan untuk mendapatkan jalur penerbangan dengan total jarak tempuh terpendek. Dalam satu kali pengujian, UAV terbang menggunakan baterai lipo 4S 4200 mAh 25C pada ketinggian 120 m dengan nilai sudut kontrol *roll* maksimum sebesar 45 derajat dan persentase *cruise throttle* sebesar 30% maka penggunaan sistem autopilot optimasi dapat menghemat 47% dari waktu dan 41% dari kapasitas baterai yang dibutuhkan apabila dibandingkan dengan penggunaan sistem autopilot konvensional pada umumnya.

Kata kunci: Autopilot, Pemetaan Udara dan Algoritma Genetika

Halaman ini sengaja dikosongkan

DEVELOPMENT OF UNMANNED AERIAL VEHICLE AUTOPILOT SYSTEM USING GENETIC ALGORITHM TO MINIMIZE BLANK SPOT ZONE IN AERIAL MAPPING APPLICATION

Name : Muhammad Ichlasul Salik
Supervisor I : Ronny Mardiyanto, ST., MT., Ph.D.
Supervisor II : Dr. Ir. Djoko Purwanto, M.Eng.

ABSTRACT

In this thesis, an optimization of autopilot system on the UAV is carried out for aerial mapping mission. Optimization is done by giving the ability to UAV in analyzing the blank spot zones that generated during the mission. The blank spot coordinate data become a new flight route for the UAV to re-take the picture. To generate new flight routes, analysis of all blank spot coordinate data points is carried out by applying genetic algorithms in preparing the flight path. With this method, it is expected that a new flight route that produced which results in the shortest total mileage for UAV to re-take the picture. Thus, it decreases the time and battery power needed to produce a perfect aerial map, making it more efficient. The results of the test obtained in this thesis is an autopilot system that can analyze blank spot zones in aerial mapping missions and optimize flight routes to get flight paths with the shortest total mileage. In onetime trial, UAV flight using 4200 mAh 4s 25C lipo battery at 120 m altitude with 45 degrees for roll control angle value and 30% throttle cruise percentage. Therefore, with the use of this autopilot system, it can minimize the blank spot zones that generated in a map and save 47% of the time and 41% of the battery capacity needed when compared to the use of conventional autopilot systems in general.

Keywords: *Autopilot, Aerial Mapping and Genetic Algorithm*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji dan syukur kepada ALLAH SWT atas berkat, rahmat dan karunianya sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul **“PENGEMBANGAN SISTEM AUTOPILOT PESAWAT TANPA AWAK MENGGUNAKAN ALGORITMA GENETIKA UNTUK MEMINIMALISIR ZONA *BLANK SPOT* PADA APLIKASI PEMETAAN WILAYAH”**. Laporan tersebut merupakan salah satu persyaratan dalam menyelesaikan pendidikan program Strata-Satu(S1) di Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.

Dalam penyusunan dan penyelesaian laporan Tugas Akhir ini penulis tidak luput 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 tugas akhir ini.
2. Bapak Ronny Mardiyanto, ST., MT., Ph.D. selaku dosen pembimbing 1 atas usulan judul topik tugas akhir, bimbingan serta arahan yang telah diberikan kepada penulis selama proses pengerjaan tugas akhir ini.
3. Bapak Dr. Ir. Djoko Purwanto, M.Eng selaku dosen pembimbing 2 atas usulan evaluasi judul topik tugas akhir, bimbingan serta arahan yang telah diberikan kepada penulis selama proses pengerjaan tugas akhir ini.
4. Teman-teman perkuliahan : Mar'ie, Rasyid, Aldifa dan Dion yang banyak memberikan bantuan baik dalam pengujian ataupun penyusunan laporan pada tugas akhir ini.
5. Teman-teman tim BAYUCARAKA ITS yang membantu penulis mendapatkan pinjaman berbagai komponen yang dibutuhkan dalam merealisasikan UAV untuk melakukan tahap uji coba pada tugas akhir ini.
6. Seluruh dosen dan karyawan Departemen Teknik Elektro ITS yang telah memberikan banyak ilmu dan menciptakan suasana belajar yang kondusif selama masa proses perkuliahan.

Dengan terselesaikannya tugas akhir ini penulis berharap dapat memberikan banyak manfaat kepada pembaca dan ikut berkontribusi dalam meningkatkan perkembangan teknologi UAV (*Unmanned Aerial Vehicle*) khususnya di Indonesia. Penulis menyadari bahwa pada tugas akhir 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 tugas akhir ini.

Surabaya, 2020

Muhammad Ichlasul Salik

DAFTAR ISI

HALAMAN JUDUL
PERNYATAAN KEASLIAN TUGAS AKHIR.....
LEMBAR PENGESAHAN
ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	3
1.3. Tujuan Penelitian.....	3
1.4. Batasan masalah	3
1.5. Metodologi Penelitian	4
1.6. Sistematika Penulisan.....	7
1.7. Relevansi	7
BAB 2 TEORI PENUNJANG DAN TINJAUAN PUSTAKA	9
2.1. Teori Penunjang	9
2.1.1. Fotogrametri	10
2.1.2. <i>Flight Controller</i>	10
2.1.3. Mission Planner	13
2.1.4. Komunikasi Serial	14
2.1.5. OpenCV	15
2.1.5.1. <i>Contour</i>	16
2.1.6. <i>Genetic Algorithm</i>	16
2.1.6.1. <i>Fungsi Fitness</i>	17
2.1.6.2. <i>Crossover</i>	18
2.1.6.3. <i>Mutasi</i>	19
2.1.6.4. <i>Seleksi</i>	19
2.1.6.5. <i>Travelling Salesman Problem (TSP)</i>	21
2.2. Tinjauan Pustaka	23

2.2.1.	<i>Master-Slave Parallel Vector-Evaluated Genetic Algorithm for Unmanned Aerial Vehicle's Path Planning</i>	23
2.2.2.	<i>Improved GASA Algorithm for Mutation Strategy UAV Path Planning</i>	23
2.2.3.	<i>Path Planning Based on Immune Genetic Algorithm for UAV</i>	24
2.2.4.	<i>Vibrational Genetic Algorithm Based Path Planner for Autonomous UAV In Spatial Data Based Environments</i> 24	
2.2.5.	<i>2D Path Planning of UAVs with Genetic Algorithm in a Constrained Environment</i>	25
2.2.6.	<i>The Comparison of Four UAV Path Planning Algorithms Based on Geometry Search Algorithm</i>	26
2.2.7.	<i>Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm</i>	26
2.2.8.	<i>UAV Path Planning based on Bidirectional Sparse A* Search Algorithm</i>	27
2.2.9.	<i>Autonomous Path Planning by Unmanned Aerial Vehicle (UAV) for Precise Monitoring of Large-Scale PV plants</i>	28
2.2.10.	<i>A Path Planning for One UAV Based on Geometric Algorithm</i>	28
2.2.11.	<i>Path planning of mobile robot with path rule mining based on GA</i>	29
2.2.12.	<i>Ship Path Planning Based on Improved Particle Swarm Optimization</i>	29
2.2.13.	<i>UAV Path Planning Based on Chaos Ant Colony Algorithm</i>	30
2.2.14.	<i>3D Environment Modeling with Height Dimension Reduction and Path Planning for UAV</i>	30
2.2.15.	<i>Experimental performance evaluation of various path planning algorithms for obstacle avoidance in UAVs</i> ..	31
BAB 3 PERANCANGAN SISTEM		33
3.1.	Perencanaan Jalur Terbang	35
3.1.1.	Faktor Kamera	35
3.1.2.	Perhitungan Jalur Terbang	38
3.1.3.	Penyimpanan Rencana Terbang	39
3.2.	Perancangan Optimasi	40

3.2.1.	Menganalisa Zona <i>Blank Spot</i>	43
3.2.2.	Implementasi Algoritma Genetika	55
3.2.2.1.	<i>Pengkodean</i>	55
3.2.2.2.	<i>Membangkitkan Populasi</i>	56
3.2.2.3.	<i>Evaluasi</i>	57
3.2.2.4.	<i>Elitisme</i>	62
3.2.2.5.	<i>Kawin Silang</i>	63
3.2.2.6.	<i>Mutasi</i>	65
3.2.2.7.	<i>Repetisi</i>	65
3.2.3.	Konversi koordinat	66
3.3.	Perancangan Wahana	66
3.3.1.	Perancangan Mekanik	67
3.3.2.	Perancangan Elektronik	72
3.3.3.	Perakitan	76
3.4.	Komunikasi GCS dan UAV	77
BAB 4 PENGUJIAN DAN ANALISIS		81
4.1.	Realisasi Alat	81
4.2.	Pengujian Karakteristik UAV	83
4.3.	Pengujian Sistem	84
4.3.1.	Simulasi	84
4.3.2.	Realisasi	88
4.3.2.1.	<i>Tanpa Penerapan Algoritma Pengembangan Sistem Autopilot</i>	88
4.3.2.2.	<i>Penerapan Algoritma Pengembangan Sistem Autopilot</i>	97
4.3.2.3.	<i>Analisa Hasil Pengujian</i>	106
BAB 5 PENUTUP		109
5.1.	KESIMPULAN	109
5.2.	SARAN	109
DAFTAR PUSTAKA		111
LAMPIRAN A		117
LAMPIRAN B		129
LAMPIRAN C		131
LAMPIRAN D		133
BIOGRAFI PENULIS		149

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1. <i>Flight Controller</i> Pixhawk	10
Gambar 2.2. Sistem Autopilot Konvensional.....	11
Gambar 2.3. <i>Software Mission Planner</i>	13
Gambar 2.4. Pengaturan Jalur Terbang	13
Gambar 2.5. Hasil Perencanaan Jalur Terbang	14
Gambar 2.6. Logo OpenCV	15
Gambar 2.7. Ilustrasi Tahap Kawin Silang	18
Gambar 2.8. Ilustrasi Tahap Mutasi	19
Gambar 2.9. Roulette Wheel.....	20
Gambar 3.1. Ilustrasi Proses Pengambilan Foto.....	33
Gambar 3.2. Ilustrasi Penerapan Analisa Blank Spot dan Algoritma Genetika	33
Gambar 3.3. Pola <i>Simple Grid</i>	38
Gambar 3.4. Prosedur Penyimpanan <i>Waypoint</i>	39
Gambar 3.5. Diagram Blok Keseluruhan Sistem	40
Gambar 3.6. Diagram Blok Mekanisme Pemetaan Menggunakan Sistem Autopilot Hasil Optimasi.....	41
Gambar 3.7. Ilustrasi Koordinat Jalur Penerbangan.....	41
Gambar 3.8. Ilustrasi Koordinat Zona <i>Blank spot</i>	42
Gambar 3.9. Ilustrasi Susunan Koordinat Misi Penerbangan Baru	43
Gambar 3.10. Hasil <i>Canvas</i>	44
Gambar 3.11. Titik Koordinat Batas Area Cakupan	45
Gambar 3.12. Penyesuaian Batas Titik Koodinat <i>Canvas</i>	46
Gambar 3.13. Koordinat Titik Pusat Area Cakupan.....	47
Gambar 3.14. Hasil Representasi Koordinat Bumi pada <i>Canvas</i>	47
Gambar 3.15. Ilustrasi Metode <i>Plotting</i>	48
Gambar 3.16. Ilustrasi Hasil Metode <i>Plotting</i>	49
Gambar 3.17. Ilustrasi <i>Canvas</i> dengan Zona <i>Blanspot</i>	50
Gambar 3.18. Ilustrasi <i>Canvas</i> Tanpa Zona <i>Blank spot</i>	50
Gambar 3.19. Hasil <i>Grayscale Canvas</i>	52
Gambar 3.20. Hasil <i>Thresholding Canvas</i>	53
Gambar 3.21. Implementasi <i>FindContour</i> Pada <i>Canvas</i>	54
Gambar 3.22. Ilutrasi Metode <i>Equidistance</i>	57

Gambar 3.23. Metode Pythagoras.....	58
Gambar 3.24. Metode Perhitungan Jarak Dua Titik Koordinat	58
Gambar 3.25. <i>Design</i> 3D UAV.....	67
Gambar 3.26. <i>Design</i> 3D UAV Tampak Samping.....	68
Gambar 3.27. <i>Design</i> 3D UAV Tampak Depan	68
Gambar 3.28. <i>Design</i> 3D UAV Tampak Atas	69
Gambar 3.29. <i>Design</i> 2D UAV Tampak Atas	69
Gambar 3.30. <i>Design</i> 2D UAV Tampak Samping.....	70
Gambar 3.31. <i>Design</i> Cetak UAV 2D	70
Gambar 3.32. Tahap Pembuatan Sayap UAV.....	71
Gambar 3.33. <i>Wiring</i> Komponen Elektronik UAV	73
Gambar 3.34. <i>Wiring</i> Komponen Elektronik GCS	74
Gambar 3.35. Peletakan Komponen Pada UAV	76
Gambar 4.1. UAV Tampak Atas	81
Gambar 4.2. UAV Tampak Samping.....	81
Gambar 4.3. UAV Tampak Belakang.....	82
Gambar 4.4. <i>Ground Control Systems</i>	82
Gambar 4.5. Nilai Parameter Kontrol UAV	83
Gambar 4.6. Pola Rute Penerbangan	84
Gambar 4.7. Simulasi Jalur Pergerakan UAV Saat Mengudara	85
Gambar 4.8. Hasil Analisa Zona <i>Blank spot</i>	86
Gambar 4.9. Hasil Implementasi Algoritma Genetika.....	86
Gambar 4.10. Keseluruhan Jalur Pergerakan UAV	87
Gambar 4.11. Grafik Ketinggian Autopilot Konvensional Misi 1	89
Gambar 4.12. Grafik 1 Tegangan Baterai Autopilot Konvensional Misi 1	89
Gambar 4.13. Grafik 2 Tegangan Baterai Autopilot Konvensional Misi 1	90
Gambar 4.14. Hasil <i>Stitching</i> Misi 1.....	91
Gambar 4.15. Ilustrasi Rute Penerbangan Hasil Analisa Manual	92
Gambar 4.16. Rute Penerbangan Misi 2	93
Gambar 4.17. Grafik Ketinggian Autopilot Konvensional Misi 2.....	94
Gambar 4.18. Grafik 1 Tegangan Baterai Autopilot Konvensional Misi 2	94
Gambar 4.19. Grafik 2 Tegangan Baterai Autopilot Konvensional Misi 2	95

Gambar 4.20. Hasil <i>Stitching</i> Penggabungan Foto Misi 1 dan Misi 2 ..	96
Gambar 4.21. Realisasi Hasil Pencatatan Pergerakan UAV	98
Gambar 4.22. Realisasi Hasil Analisa Zona <i>Blank Spot</i>	99
Gambar 4.23. Rute Penerbangan Dalam Koordinat <i>Canvas</i>	100
Gambar 4.24. Rute Penerbangan Dalam Koordinat Bumi	100
Gambar 4.25. Rute Penerbangan Hasil Implementasi Algoritma Genetika.....	100
Gambar 4.26. Rute Penerbangan Hasil Analisa <i>Blank Spot</i>	101
Gambar 4.27. Jalur Pergerakan UAV Dalam Penggunaan Sistem Autopilot Hasil Pengembangan	102
Gambar 4.28. Grafik Ketinggian Sistem Autopilot Hasil Pengembangan	103
Gambar 4.29. Grafik 1 Tegangan Baterai Sistem Autopilot Hasil Pengembangan.....	103
Gambar 4.30. Grafik 2 Tegangan Baterai Sistem Autopilot Hasil Pengembangan.....	104
Gambar 4.31. Hasil <i>Stitching</i> Penggunaan Sistem Autopilot Hasil Pengembangan.....	105
Gambar 4.32. Perbandingan Hasil <i>Stitching</i> Dalam Perbedaan Penggunaan Sistem Autopilot.....	107

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1. Spesifikasi Kamera	36
Tabel 3.2. Data Koordinat Zona <i>Blank spot</i>	54
Tabel 3.3. Tabel Hasil Pengkodean Koordinat Zona <i>Blank spot</i>	56
Tabel 3.4. Tabel Hasil Pembangkitan Populasi.....	57
Tabel 3.5. Tabel Hasil Evaluasi	59
Tabel 3.6. Hasil Perhitungan Nilai Fitness.....	59
Tabel 3.7. Tabel Hasil Normalisasi	61
Tabel 3.8. Tabel Populasi Sebelum Tahap Elitisme.....	62
Tabel 3.9. Tabel Populasi Setelah Tahap Elitisme	62
Tabel 3.10. <i>Wiring</i> Komponen Elektronik	74
Tabel 3.11. Spesifikasi Komponen Elektronik.....	76
Tabel 3.12. Protokol Komunikasi	79
Tabel 4.1. Tabel Kalkulasi Proses <i>Stitching</i> Penggunaan Autopilot Konvensional Misi 1	90
Tabel 4.2. Tabel Kalkulasi Proses <i>Stitching</i> Penggunaan Autopilot Konvensional Misi 2	95
Tabel 4.3. Tabel Kalkulasi Proses Pemetaan Dengan Sistem Autopilot Konvensional	97
Tabel 4.4. Nilai Variabel Dalam Implementasi Algoritma Genetika	99
Tabel 4.5. Tabel Kalkulasi Proses <i>Stitching</i> Dengan Sistem Autopilot Hasil Pengembangan.....	104
Tabel 4.6. Tabel Kalkulasi Proses Pemetaan Dengan Sistem Autopilot Hasil Optimasi.....	105
Tabel 4.7. Hasil Perbandingan Autopilot Konvensional dan Autopilot Optimasi	107

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai hal-hal yang mendasari dalam pembuatan Tugas Akhir ini. Dimana hal-hal dasar tersebut meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, sistematika penulisan dan relevansi.

1.1. Latar Belakang

Pada awalnya sebuah UAV diimplementasikan pada bidang militer. Bermula saat USAF (Angkatan Udara Amerika Serikat) menggunakan sebuah UAV dalam medan perang. Penggunaan UAV terus meluas dikarenakan banyaknya pilot yang tewas dalam medan perang, sehingga penggunaan UAV dilakukan pada misi-misi yang beresiko tinggi sebagai jalan untuk mengurangi adanya korban jiwa[1]. Seiring perkembangan zaman, teknologi UAV banyak diimplementasikan diberbagai sektor. Kemudahan penggunaan pesawat tanpa awak yang mana dapat dikontrol secara jarak jauh memberikan fleksibilitas bagi penggunaanya dalam melakukan suatu misi di area tertentu yang tidak mudah dijangkau oleh manusia. Fungsinya yang mampu diterapkan ke dalam berbagai misi menjadi suatu pertimbangan utama. Hal ini ternyata dirasa dapat membantu dalam peningkatan produktivitas, penekanan biaya operasional dan meminimalisir resiko kecelakaan.

Salah satu fungsi pesawat tanpa awak yang akan dibahas pada tugas akhir ini adalah pemanfaatan UAV dalam pembuatan sebuah peta dengan metode yang dinamakan fotogrametri. Teknik fotogrametri semakin berkembang pesat sejak foto udara mulai digunakan dalam pemetaan topografi pada tahun 1849 oleh Colonel Aime Laussedat yang merupakan Bapak Fotogrametri. Pada tahun itu fotogrametri dilakukan dengan memanfaatkan wahana yang paling modern dizamannya yaitu balon udara dan layang-layang. Pemetaan dilakukan dengan membuat balon udara ataupun layang-layang terbang dengan membawa beberapa perangkat yang dibutuhkan untuk melakukan pemotretan. Kemudian seiring perkembangan zaman, teknologi-teknologi baru bermunculan. Pada tahun 1902 ditemukannya pesawat terbang oleh Wright Bersaudara.

Penemuan teknologi pesawat ini diiringi dengan berkembangnya kamera metrik sebagai sensor dalam pemotretan[2]. Dengan ditemukannya pesawat dan teknologi kamera metrik membuat teknik pemetaan fotogrametri semakin berkembang. Dimana terjadi penerapan kamera metrik pada pesawat terbang untuk melakukan pemetaan fotogrametri. Penerapan tersebut mampu menghasilkan kualitas foto udara yang baik. Namun biaya operasional yang perlu dikeluarkan menjadi sangat mahal. Berkembangnya teknologi UAV atau pesawat tanpa awak menjadi solusi baru dalam hal teknologi fotogrametri. Penggunaan UAV dalam fotogrametri memberikan beberapa keuntungan diantaranya adalah efektif dan efisien baik dari segi waktu dan biaya operasional, serta dapat menghasilkan gambar yang lebih jelas. Hal ini dikarenakan UAV dapat terbang kurang lebih 300 meter diatas permukaan laut sehingga dapat meminimalisir terjadinya gangguan awan pada saat proses pemotretan[3]. Dengan menggunakan UAV, pemetaan juga dapat diaplikasikan untuk wilayah area yang kecil.

Dalam hal pengambilan data foto udara pada suatu rute penerbangan, sebuah UAV memproses koordinat GPS aktual saat terbang terhadap koordinat *waypoint* yang telah di tetapkan[4]. Namun, terdapat beberapa faktor yang dapat membuat hasil peta menjadi tidak sempurna. Salah satu faktor yang mempengaruhi ialah kecepatan udara. Kecepatan udara dapat mempengaruhi UAV saat terbang mengudara. Akibatnya beberapa titik koordinat *flight plan* yang direncanakan tidak terlewati sempurna oleh wahana UAV tersebut. Faktor ini akan membuat beberapa bagian wilayah tidak dapat terfoto sempurna oleh wahana UAV dan mengakibatkan munculnya zona *blank spot* pada peta yang dihasilkan.

Dari uraian latar belakang tersebut maka pada tugas akhir ini akan dibuat suatu pengembangan sistem autopilot yang dapat menganalisa koordinat-koordinat yang terlewati oleh wahana UAV dan mengeksekusi kembali koordinat tersebut untuk melakukan pengambilan foto ulang sehingga tidak adanya zona *blank spot*. Dengan mengadopsi algoritma optimasi *Genetic Algorithm Travel Sales Planner* membuat wahana UAV dapat menentukan susunan jalur terdekat yang harus ditempuh dalam mengeksekusi koordinat-koordinat yang tidak terlewati sebelumnya, dengan demikian dapat mengurangi biaya operasional dan mempercepat

waktu dalam pengambilan data. Sehingga diharapkan penggunaan UAV dalam pemetaan udara dapat lebih efektif dan efisien.

1.2. Perumusan Masalah

Perumusan masalah pada Tugas Akhir ini adalah :

1. Munculnya zona *blank spot* pada hasil pemetaan dengan menggunakan UAV.
2. Dengan munculnya zona *blank spot* waktu yang dibutuhkan dalam melakukan pemetaan untuk mendapatkan hasil peta yang optimal relatif lama.
3. Dengan munculnya zona *blank spot* konsumsi daya baterai yang dibutuhkan untuk mendapatkan hasil peta yang optimal relatif besar.

1.3. Tujuan Penelitian

1. Meminimalkan zona *blank spot* yang dihasilkan pada peta.
2. Mengurangi durasi waktu yang dibutuhkan untuk menghasilkan gambar peta yang optimal.
3. Mengurangi konsumsi daya baterai yang dibutuhkan untuk menghasilkan gambar peta yang optimal.

1.4. Batasan masalah

1. Diasumsikan wahana UAV dapat melakukan manuver dengan cepat dan tajam.
2. Wilayah cakupan pemetaan seluas 500x500 m.
3. Tidak memperhitungkan ketinggian pesawat dalam pemrosesan algoritma genetika.
4. Pengujian dilakukan dengan melakukan simulasi.

1.5. Metodologi Penelitian



Gambar 1.1 Metodologi Penelitian

1. Studi Literatur

Studi literatur bertujuan untuk mempelajari teori dasar atau penelitian tentang hal terkait yang sudah ada guna menunjang pengerjaan tugas akhir ini. Literatur yang dicari bisa diperoleh dari buku, jurnal, paper, atau artikel yang terpercaya. Topik yang akan dipelajari nantinya mengenai sebuah *library* bahasa pemrograman python yaitu drone-kit. Dimana drone-kit mengemas komunikasi protokol MAVLINK sebagai perantara komunikasi antara komputer dengan *flight controller* pada UAV dan topik selanjutnya mengenai algoritma optimasi yaitu *genetic algorithm: travel sales planner*.

2. Pemodelan Sistem

Pada tahap ini terdapat 3 hal utama yang diteliti.

a. Perkiraan Perencanaan Jalur Terbang

Pada tahap ini dilakukan penyesuaian jalur terbang dengan menentukan tipe pola jalur terbang yang akan digunakan serta memperhitungkan spesifikasi kamera, nilai *overlap* dan *sidelap*. Keluaran yang didapatkan berupa nilai-nilai

parameter yang dapat dimasukkan pada *flight controller* sehingga UAV dapat terbang sesuai dengan harapan.

b. Perancangan optimasi

Pada tahap ini dilakukan perancangan logika optimasi yang digunakan serta mekanisme pengaplikasian pada UAV. Pada bagian optimasi digunakan bahasa pemrograman python dengan mengadopsi logika dari algoritma genetika yaitu *travel sales planner*. Logika optimasi dibuat sedemikian rupa agar tidak mengganggu sistem kontrol pada UAV saat mengudara. Dalam hal pengaplikasian, sistem kontrol UAV akan dilakukan pada komputer dengan menggunakan *library drone-kit* sebagai protokol komunikasi terhadap *flight controller*. Kemudian kedua sub program tersebut akan disatukan untuk menjadi sebuah program autopilot yang terintegrasi.

c. Perancangan Wahana

Pada tahap ini dilakukan perancangan *Design* wahana UAV yang akan digunakan. Perancangan tersebut akan dibagi menjadi dua bagian yaitu perancangan mekanik dan perancangan elektronik. Perancangan mekanik akan berkaitan mengenai bentuk dan jenis wahana UAV yang akan dibuat. Sedangkan perancangan elektronik berkaitan dengan kebutuhan komponen-komponen elektronik yang digunakan UAV tersebut dalam menunjang menyelesaikan misi penerbangan yang diinginkan.

3. Uji Simulasi

Pada tahap ini akan dilakukan uji simulasi kesesuaian sistem yang ada. Simulasi yang akan dilakukan sebanyak 2 tahap. Tahap pertama, melakukan simulasi program optimasi pada *software* Jupyter untuk mendapatkan nilai-nilai parameter optimasi yang paling optimum. Tahap kedua, melakukan simulasi sistem kontrol UAV yang terintegrasi dengan algoritma optimasi. Pengujian dilakukan untuk memastikan algoritma optimasi dapat

terimplementasi pada sistem autopilot yang berjalan. Simulasi akan dilakukan menggunakan *software* Mission Planner.

4. Perakitan Alat

Setelah uji simulasi menghasilkan keluaran yang sesuai dengan yang diharapkan, selanjutnya dilakukan perakitan alat. Pada segi *hardware*, peletakan *hardware* akan mempertimbangkan dari volume *payload* yang tersedia pada wahana UAV yang akan digunakan. Peletakan *hardware* tersebut dipertimbangkan terhadap titik CG (*central of gravity*) wahana UAV. Sehingga peletakan *hardware* tidak mengganggu sistem kontrol dan karakteristik wahana UAV saat mengudara. Ketika faktor-faktor tersebut telah terpenuhi maka dilakukan perakitan dan pemasangan komponen-komponen penunjang pada wahana UAV.

5. Pengujian dan Pengembangan Sistem

Pada tahap ini dilakukan pengecekan kesesuaian sistem autopilot yang telah dikembangkan terhadap keluaran yang diinginkan. Dimana pengujian dilakukan dengan melakukan proses *mapping* wilayah seluas luasan daerah yang telah ditentukan. Kemudian dilakukan identifikasi dan perbandingan hasil *mapping* ketika UAV menggunakan sistem autopilot yang telah dikembangkan pada tugas akhir ini dengan sistem autopilot konvensional pada umumnya. Banyaknya tahap pengujian diharapkan akan mendapatkan algoritma pemrograman optimasi yang paling optimum dengan menganalisa error-error yang terjadi selama tahap pengujian.

6. Penyusunan Laporan

Penyusunan laporan dikerjakan secara beriringan dengan pengerjaan tahap – tahap lainnya. Laporan akan disajikan dalam bentuk buku untuk dijadikan laporan hasil tugas akhir yang telah dikerjakan. Isinya meliputi pendahuluan, studi literatur, pengujian, analisa, dan penutup. Kesimpulan yang didapat merupakan jawaban dari topik permasalahan yang dianalisa pada tugas akhir ini.

1.6. Sistematika Penulisan

Dalam penyusunan laporan buku tugas akhir ini disesuaikan dengan peraturan Buku Pedoman Tugas Akhir Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Pada buku ini pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- **BAB I: Pendahuluan**

Pada bab ini penulis memuat tentang latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, manfaat, metodologi penelitian dan sistematika penulisan.

- **BAB II: Teori Penunjang dan Tinjauan Pustaka**

Pada bab ini penulis memuat tentang teori-teori yang menunjang dalam pengembangan sistem pada tugas akhir ini. Penjelasan yang diberikan memuat mengenai komponen perangkat keras, perangkat elektronik dan bahasa pemrograman yang akan digunakan.

- **BAB III: Perancangan Sistem**

Pada bab ini penulis memuat tentang perancangan sistem yang akan digunakan dalam menyelesaikan permasalahan yang diangkat pada tugas akhir ini. Perancangan meliputi perancangan perangkat keras (*hardware*), perangkat elektronik dan logika pemrograman.

- **BAB IV: Pengujian dan Analisis**

Pada bab ini penulis memuat tentang pengujian sistem pada UAV beserta analisisnya.

- **BAB V: Penutup**

Pada bab ini penulis memuat tentang kesimpulan yang didapatkan dari pengimplementasian pengembangan sistem autopilot yang dihasilkan pada tugas akhir ini beserta saran untuk pengembangan lebih lanjut.

1.7. Relevansi

Luas dan banyaknya wilayah berpotensi di Negara Indonesia membuat tahap pemetaan merupakan suatu langkah utama dalam hal perencanaan pembangunan suatu wilayah. Teknik pemetaan menggunakan UAV bukanlah suatu metode yang asing lagi dalam hal pemetaan wilayah. Dengan dikembangkannya sistem autopilot dalam hal misi pemetaan yang dilakukan pada tugas akhir ini diharapkan

dapat meningkatkan efisiensi waktu dan konsumsi baterai yang digunakan pada UAV sehingga dapat menurunkan biaya operasional dalam hal proses pemetaan dan meningkatkan luas cakupan pemetaan.

BAB 2

TEORI PENUNJANG DAN TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai teori-teori dasar yang menunjang dalam pembuatan Tugas Akhir ini. Dimana terdapat teori-teori yang dijadikan acuan sebagai sumber pemecahan masalah dalam penyelesaian Tugas Akhir ini. Selain itu terdapat juga tinjauan pustaka yang membahas hasil-hasil riset yang pernah dilakukan sebelumnya dan dijadikan sebagai referensi.

2.1. Teori Penunjang

Pesawat tanpa awak atau pesawat nirawak (*Unmanned Aerial Vehicle* atau UAV), adalah sebuah mesin terbang yang berfungsi dengan kendali jarak jauh oleh pilot atau secara mandiri tanpa pilot. Dimana sebuah UAV menggunakan hukum aerodinamika untuk mengangkat dirinya bisa mengudara. Pada awalnya secara garis besar penggunaan dari pesawat tanpa awak ini diimplementasikan dibidang militer.

Perkembangan teknologi membuat *drone* juga mulai banyak diterapkan untuk kebutuhan sipil, terutama di bidang bisnis, industri dan logistik. Pada dunia industri bisnis, *drone* telah diterapkan dalam berbagai layanan seperti pengawasan infrastruktur, pengiriman paket barang, pemadam kebakaran hutan, eksplorasi bahan tambang, pemetaan daerah pertanian, dan pemetaan daerah industri[5]. Penggunaan UAV dalam berbagai keperluan sipil dan militer masing-masing akan membutuhkan spesifikasi UAV yang berbeda. Oleh karenanya penggunaan komponen pada UAV dapat menyesuaikan dengan misi penerbangan yang direncanakan. Dengan demikian faktor tersebut akan mempengaruhi waktu terbang, kemampuan jarak jelajah maupun ketinggian terbang sebuah UAV[6].

Jangkauan radio *remote control* sebuah UAV memiliki jarak yang terbatas, sedangkan pada umumnya sebuah UAV diimplementasikan untuk misi- misi penerbangan jarak jauh seperti halnya pada Tugas Akhir ini yaitu penggunaan UAV untuk pemetaan lahan. Oleh karenanya dibutuhkan sebuah sistem autopilot pada sebuah UAV. Penggunaan sistem autopilot akan memberikan kemampuan UAV untuk terbang secara mandiri dengan mengelola seluruh nilai-nilai sensor yang terkandung pada *flight controller*.

2.1.1. Fotogrametri

Fotogrametri adalah cabang ilmu geoinformasi yang dapat didefinisikan sebagai seni, sains dan teknologi dalam memperoleh informasi yang dapat dipercaya tentang objek fisik dan lingkungan melalui proses pencatatan, pengukuran dan menafsirkan gambar dan pola fotografi citra pancaran elektromagnetik dan fenomena lainnya[7]. Metode ini memiliki keunggulan diantaranya dapat menghasilkan data *geo-spatial* yang relatif lebih cepat dan efektif untuk cakupan wilayah yang luas. Terdapat dua tipe UAV yang dapat digunakan untuk proses *mapping* yaitu jenis *fixed wing* dan *rotary wing*[8].

2.1.2. Flight Controller

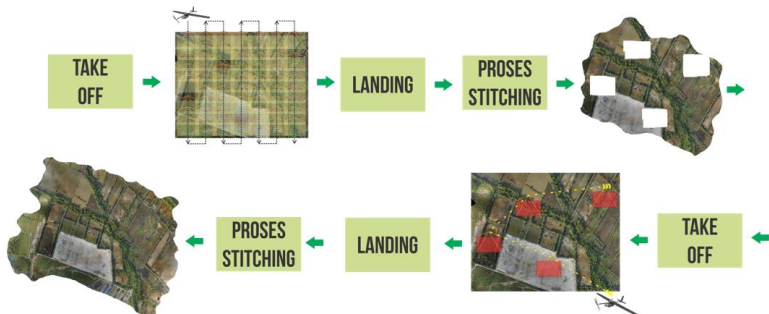


Gambar 2.1. *Flight Controller Pixhawk*

Flight Controller (FC) merupakan sebuah komponen elektronika yang diaplikasikan pada dunia aviasi dimana dapat membuat pesawat terbang secara autonomus. Pada *flight controller* memuat berbagai sensor seperti *Gyro*, *Accelormeter*, *Magnetometer*, *Barometer*, dan GPS. Dengan membaca dan melakukan kalkulasi terhadap seluruh sinyal keluaran dari berbagai sensor diatas, hal tersebut memungkinkan sebuah wahana terbang dapat mengudara secara mandiri dengan stabil dengan mengoreksi gerakannya. Hal tersebut pula memungkinkan sebuah wahana terbang dapat melakukan perintah sesuai dengan instruksi pengguna atau pengendara.

UAV (*Unmanned Aerial Vehicle*) pada umumnya digunakan untuk misi penerbangan jarak jauh diluar jarak pandang pengendali, oleh sebab itu sebuah komponen *flight controller* dibutuhkan. Berbagai jenis *flight controller* telah banyak dikembangkan, pada umumnya komponen yang terkandung memiliki jenis yang sama namun setiap *flight controller* mempunyai algoritma tersendiri sehingga akan berpengaruh pada karakteristik terbang yang dihasilkan. Dikarenakan penggunaan jenis *flight controller* berpengaruh pada karakteristik terbang suatu UAV, oleh karena itu penggunaan *flight controller* perlu disesuaikan dengan misi penerbangan yang akan dilakukan oleh UAV.

Perkembangan *flight controller* terus meningkat seiring dengan kebutuhan pasar yang beragam. Pada misi pemetaan wilayah, *flight controller* digunakan untuk membuat UAV dapat terbang sesuai dengan *flight plan* yang telah ditentukan tanpa perlu dikendalikan oleh seorang pilot. Namun, dalam misi pemetaan wilayah terdapat permasalahan akan terjadinya kemunculan fenomena zona *blank spot*. Pada umumnya untuk mengatasi permasalahan tersebut dilakukan dengan melakukan penerbangan secara berulang. Sehingga membutuhkan konsumsi daya baterai yang lebih besar dan waktu pemetaan yang lebih lama. Berikut pada Gambar 2.2 merupakan ilustrasi tahap-tahap proses pemetaan dalam menggunakan sistem autopilot konvensional pada umumnya.



Gambar 2.2. Sistem Autopilot Konvensional

Pada tugas akhir ini digunakan sebuah *flight controller* jenis Pixhawk 3DR PX4 Advanced Autopilot. *Flight controller* jenis ini diproduksi oleh 3D Robotics.

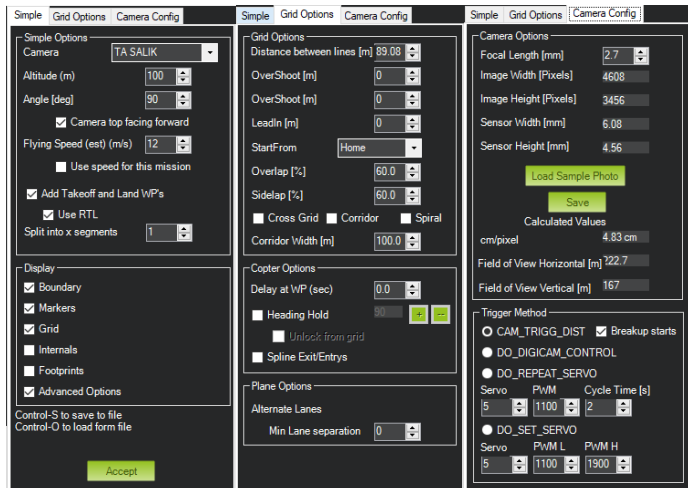
Berikut Spesifikasi yang ditawarkan[9] :

1. Prosesor
 - 32-bit ARM Cortex M4 core dengan FPU
 - 168 Mhz/256 KB RAM/2 MB Flash
 - 32-bit failsafe co-processor
2. Sensor
 - MPU6000 sebagai sensor accel dan gyro utama
 - ST Micro 16-bit gyroscope
 - ST Micro 14-bit accelerometer/compass (magnetometer)
 - MEAS barometer
3. Daya
 - Penggunaan kontroler dioda ideal dengan proteksi otomatis
 - Suplai tegangan dan arus motor servo yang tinggi (7V)
 - Seluruh keluaran dilindungi proteksi arus berlebih, semua input dilindungi ESD
4. Antarmuka
 - PPM sum signal
 - RSSI (PWM or voltage) input
 - I2C, SPI, 2x CAN, USB
 - 3.3V and 6.6V ADC input
 - Futaba S.BUS input
 - Spektrum DSM/DSM2/DSM-X Satellite input
 - 5x UART port serial

2.1.3. Mission Planner



Gambar 2.3. Software Mission Planner



Gambar 2.4. Pengaturan Jalur Terbang

Mission Planner adalah salah satu *software user interface* yang dikembangkan oleh Michael Osborne dimana berfungsi sebagai *interface ground control station* untuk sebuah *drone/UAV, copter* dan *rover*[10].

Software ini hanya cocok dijalankan untuk sistem operasi windows saja. Pada tugas akhir ini penggunaan *software* mission planner dilakukan dalam mengatur setiap konfigurasi kontrol UAV dan perencanaan jalur terbang dalam melakukan pemetaan. Dengan penggunaan *software* ini dalam melaksanakan pemetaan, nilai perhitungan skala, resolusi spasial, ketinggian terbang, nilai *overlap* dan *sidelap*, dan lainnya bisa didapatkan dengan perhitungan secara otomatis. Dapat dilihat pada Gambar 2.4 merupakan beberapa variabel yang perlu dimasukkan dalam melakukan perencanaan jalur terbang.

Dengan melakukan *input* nilai data terhadap variabel-variabel yang dibutuhkan seperti Gambar 2.4 diatas, *software* akan melakukan pembuatan jalur terbang secara otomatis berdasarkan kalkulasi data variabel yang telah dimasukkan sebelumnya seperti Gambar 2.5 dibawah ini.



Gambar 2.5. Hasil Perencanaan Jalur Terbang

2.1.4. Komunikasi Serial

Komunikasi serial adalah salah satu metode komunikasi data dimana pengiriman data dilakukan per-bit secara bergantian dan berurutan. Komunikasi serial memiliki kelebihan dibandingkan dengan metode komunikasi parallel dimana hanya membutuhkan satu jalur dan sedikit kabel. Namun komunikasi serial memiliki kecepatan pengiriman

data yang lebih lambat dibandingkan dengan komunikasi parallel dikarenakan pengiriman data yang dilakukan secara per-bit[11].

Dalam komunikasi serial terbagi dalam 2 jenis yaitu *asynchronous serial* dan *synchronous serial*. *Synchronous serial* adalah komunikasi dimana *clock* hanya dihasilkan oleh satu pihak (pengirim atau penerima) dan *clock* dikirim bersamaan dengan data yang akan ditransfer. Sedangkan *Asynchronous serial* adalah komunikasi dimana data yang dikirimkan tidak diikuti dengan *clock*, masing-masing pengirim atau penerima menghasilkan *clock* sendiri. Oleh sebab itu dibutuhkan sinkronisasi frekuensi *clock* antara pihak pengirim dan penerima sehingga data yang diterima sesuai dengan data yang dikirim.

2.1.5. OpenCV



Gambar 2.6. Logo OpenCV¹

OpenCV merupakan suatu *library open source* dalam hal *computer vision*. Salah satu tujuan dari OpenCV adalah untuk menyediakan infrastruktur *computer vision* yang mudah dalam penggunaannya sehingga dapat digunakan dengan cepat membangun aplikasi *vision* yang cukup canggih. *Library* OpenCV dibuat dalam bahasa C dan C++ dan dapat dijalankan dalam sistem operasi Linux, Windows dan MAC OS X. OpenCV memiliki 500 fungsi yang dapat digunakan dalam banyak hal pada sistem *vision* seperti melakukan

¹OpenCV team, "OpenCV", diakses dari <https://opencv.org/>, pada tanggal 07 juli 2020, pukul 13.42 WIB

inspeksi produk suatu pabrik, sekuritas, *medical imaging*, kalibrasi kamera, *stereo vision*, antar muka pengguna (*user interface*) dan robotik[12].

2.1.5.1. Contour

Kontur adalah garis lurus atau kurva yang menggambarkan persimpangan satu atau lebih bidang horizontal dengan lebar permukaan nyata atau hipotesis[13]. Sedangkan menurut [14] Kontur adalah sebuah kumpulan titik-titik yang membentuk sebuah kurva yang membatasi daerah dimana memiliki perbedaan intensitas yang signifikan. Dalam hal ini OpenCV mempunyai sebuah fungsi yang disebut dengan *cv::FindContours* (). Dengan menggunakan fungsi tersebut kita dapat melakukan analisa bentuk objek yang akan diteliti. Dalam implementasinya terdapat beberapa hal yang perlu diperhatikan.

- a. Pertama, sumber gambar yang dianalisa harus berjenis *binary image*.
- b. Kedua, *library cv::FindContours* () akan melakukan modifikasi pada sumber gambar. Sehingga apabila sumber gambar dibutuhkan, lakukan penyimpanan sumber gambar ke variabel lain.
- c. Ketiga, dalam OpenCV *library cv::FindContours* () berfungsi sebagaimana diilustrasikan seperti mencari objek putih dalam *background* hitam. Jadi perlu diingat dalam menggunakan *library* tersebut objek pada sumber gambar yang ingin dianalisa haruslah berwarna putih dan objek yang ingin diabaikan/*background* pada sumber gambar haruslah berwarna hitam[15].

2.1.6. Genetic Algorithm

Genetic algorithm atau algoritma genetika merupakan suatu algoritma optimasi yang dibuat dengan mengadopsi mekanisme evolusi dalam perkembangan makhluk hidup dimana konsep utama pada algoritma genetika yaitu individu-individu yang paling unggul akan bertahan hidup sedangkan individu-individu yang lemah akan punah[16]. Pada penggunaannya, algoritma genetika dianggap menjadi suatu metode yang tepat untuk menyelesaikan masalah optimasi kompleks yang sulit dilakukan apabila menggunakan metode konvensional. Sifat algoritma

genetika adalah mencari nilai optimum dari kumpulan calon solusi dalam penyelesaian masalah. Calon-calon solusi tersebut dikumpulkan dalam suatu ruang cakupan yang disebut ruang pencarian/populasi. Suatu calon solusi yang layak direpresentasikan sebagai sebuah titik/individu dalam ruang pencarian yang disebut *chromosome*. Kelayakan setiap solusi/*chromosome* dalam ruang pencarian dibedakan oleh nilai fitness dari calon-calon solusi tersebut[17]. Kemudian *chromosome* tersebut akan mengalami evolusi dalam proses iterasi yang berkelanjutan sehingga menghasilkan generasi dengan solusi-solusi baru[18]. Dalam menghasilkan generasi baru dilakukan pemrosesan menggunakan 3 buah operator yaitu seleksi, *crossover* dan mutasi.

2.1.6.1. Fungsi Fitness

Fitness merupakan suatu nilai yang dimiliki masing-masing *chromosome* solusi. Nilai tersebut dijadikan parameter tingkat kelayakan *chromosome* terhadap kriteria atau nilai keluaran yang ingin dicapai. Sehingga penggunaan algoritma genetika dapat menghasilkan nilai keluaran yang optimal. *Chromosome* yang memiliki nilai fitness rendah akan mati atau berpeluang kecil untuk bertahan digenerasi selanjutnya. Begitupun sebaliknya, *chromosome* dengan nilai fitness tinggi akan bertahan hidup dan berpeluang tinggi untuk tetap ada pada generasi selanjutnya[18].

Pada umumnya fungsi fitness terbagi atas dua yaitu fungsi maksimasi untuk mendapatkan nilai maksimum dan fungsi minimasi untuk mendapatkan nilai minimum. Penggunaan fungsi fitness harus memperhatikan fungsi objektifnya. Pada tugas akhir ini fungsi objektifnya adalah meminimalkan jarak total UAV dalam mengeksekusi seluruh koordinat *waypoint blank spot*. Sehingga pada tugas akhir ini menggunakan fungsi minimasi. Berikut rumusan dari kedua jenis fungsi fitness tersebut:

A. Fungsi Maksimasi

$$\text{Fungsi Fitness} = \text{Fungsi Tujuan}$$

B. Fungsi Minimasi

$$\text{Fungsi Fitness} = 1 / (\text{Fungsi Tujuan} + \text{Bilangan Kecil})$$

Bilangan kecil pada fungsi minimasi berguna untuk mencegah nilai fungsi fitness menjadi nilai tak hingga ketika fungsi tujuan bernilai 0[19].

2.1.6.2. Crossover

Crossover merupakan salah satu operator dalam algoritma genetika yang melakukan proses kombinasi antara dua buah *chromosome* menjadi satu buah *chromosome* keturunan baru. Proses kombinasi dilakukan dengan cara menukarkan nilai gen tertentu dari kedua *chromosome* terpilih secara acak[18]. Metode iterasi *crossover* merupakan salah satu proses paling penting dalam tahapan algoritma genetika. Metode iterasi *crossover* memiliki perbedaan dengan metode iterasi konvensional pada umumnya. Pada metode konvensional, iterasi bersifat *hill climbing*. Sedangkan pada metode *crossover*, solusi yang dihasilkan akan menuju konvergen pada suatu titik tertentu secara acak[19].

Kromosom INDUK 1 :

1	2	3	5	4	6	7	8
---	---	---	---	---	---	---	---

Kromosom INDUK 2 :

2	1	5	6	8	4	7	3
---	---	---	---	---	---	---	---

Kromosom ANAK :

1	2	5	6	8	4	7	3
---	---	---	---	---	---	---	---

Gambar 2.7. Ilustrasi Tahap Kawin Silang

2.1.6.3. Mutasi

Tahap mutasi berfungsi untuk memunculkan individu-individu baru yang bukan berasal dari hasil kawin silang/*crossover* sehingga memungkinkan didapatkannya suatu solusi baru yang dapat keluar dari kondisi *local optimum*. Mutasi dilakukan dengan melakukan perubahan urutan atau penggantian elemen pada *chromosome* solusi. Pemilihan elemen pada *chromosome* yang mengalami perubahan atau penggantian dilakukan secara random. Sehingga terdapat sebuah parameter penting dalam melakukan tahap mutasi yaitu probabilitas mutasi. Nilai probabilitas ini akan mempengaruhi dalam penentuan elemen pada *chromosome* mana yang akan mengalami perubahan/pergantian. Apabila nilai probabilitas mutasi semakin besar maka semakin banyak pula *chromosome* dalam populasi yang akan mengalami mutasi. Begitu pula sebaliknya[20]. Contoh proses mutasi dapat dilihat pada gambar dibawah ini.

Kromosom X :



Titik elemen yang terpilih untuk melakukan mutasi

Kromosom X setelah dilakukan mutasi :



Gambar 2.8. Ilustrasi Tahap Mutasi

2.1.6.4. Seleksi

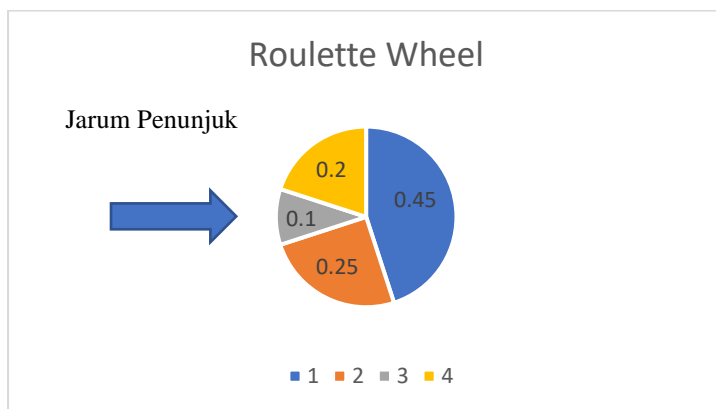
Setelah setiap *chromosome* solusi dilakukan perhitungan nilai fitness maka tahap selanjutnya adalah melakukan proses seleksi terhadap *chromosome*. Seleksi merupakan pemilihan individu terbaik yang akan dijadikan induk untuk generasi selanjutnya [21]. Tujuannya dilakukan proses seleksi untuk memperoleh solusi terbaik dalam memperoleh individu dengan sifat paling bagus, sehingga perlu dilakukan seleksi

populasi baru dan reproduksi[18]. Terdapat dua buah metode yang dapat digunakan dalam proses seleksi yaitu metode *elitism* dan metode *rouellete wheel*.

A. Metode Elitism

Pada metode ini individu-individu yang akan mengalami regenerasi, yaitu mutasi dan crossover, didasarkan pada nilai fitness yang rendah, sedangkan individu yang memiliki nilai fitness tinggi akan dipertahankan untuk dibandingkan dengan individu hasil proses regenerasi[22]. Dengan metode ini individu-individu terbaik akan tetap muncul dipopulasi berikutnya.

B. Metode Roulette Wheel



Gambar 2.9. Roulette Wheel

Pada metode ini proses seleksi mengadaptasi dari sebuah permainan bernama *Roulette Wheel*. Masing-masing *chromosome* diilustrasikan akan menempati potongan lingkaran secara proporsional berdasarkan nilai fitness yang dimilikinya[18]. *Chromosome* yang memiliki nilai fitness tertinggi akan mempunyai porsi terbesar pada *Roulette Wheel*, begitu pula sebaliknya. Dalam metode ini setiap individu akan memiliki kesempatan untuk terpilih. Namun, semakin besar luas porsi yang dimiliki pada *Roulette Wheel*

akan memperbesar peluang individu tersebut terpilih. Ilustrasi *Roulette Wheel* dapat dilihat pada Gambar 2.9 di atas.

Pada Gambar 2.9 di atas terlihat terdapat 4 buah *chromosome* dengan nilai fitnessnya masing-masing. Setiap *chromosome* mempunyai porsi penempatan pada *Roulette Wheel* yang sesuai dengan besar nilai fitness yang dimilikinya. Ketika roda *Roulette Wheel* diputar maka setiap *chromosome* memiliki kesempatan untuk terpilih. Namun, *chromosome* dengan porsi paling besar pada *Roulette Wheel* akan mempunyai peluang terbesar untuk terpilih oleh jarum penunjuk. Begitupula sebaliknya.

2.1.6.5. Travelling Salesman Problem (TSP)

Travelling Salesman Problem merupakan masalah klasik yang mana mencoba mencari rute terpendek yang bisa dilalui *salesman* yang ingin mengunjungi beberapa kota tanpa harus mendatangi kota yang sama lebih dari satu kali[23]. Menurut [24] permasalahannya adalah bagaimana *salesman* tersebut dapat mengatur rute perjalanannya sehingga jarak yang ditempuhnya merupakan jarak minimum. Terdapat berbagai metode yang dapat digunakan untuk menyelesaikan permasalahan TSP antara lain *Hill Climbing Method*, *Ant Colony System*, *Dynamic Programming* dan *Genetic Algorithm*.

Dalam perancangan algoritma pemrograman pada Tugas Akhir ini digunakan *genetic algorithm* pada implementasi TSP (*Travelling Salesman Problem*). Dimana pada implementasi TSP ini menyelesaikan permasalahan yaitu menentukan rute perjalanan dengan jarak terpendek yang dilewati oleh seorang kurir barang dalam mengantarkan seluruh paket yang ada. Dimana jika kita lihat, permasalahan ini memiliki inti permasalahan yang sama terhadap rumusan masalah dalam menentukan rute penerbangan UAV dengan jarak terpendek untuk mengeksekusi seluruh koordinat titik *blank spot* yang dihasilkan. Sehingga dapat disimpulkan *Genetic Algorithm* untuk TSP ini bisa diimplementasikan pada algoritma optimasi yang akan digunakan pada Tugas Akhir ini.

Berikut adalah langkah-langkah algoritma genetika dalam menyelesaikan TSP pada permasalahan tugas akhir ini:

1. Inisialisasi
Menentukan jumlah gen dalam satu kromosom, yaitu sejumlah koordinat *waypoint* yang diindikasikan sebagai titik *blank spot*, menentukan ukuran populasi, menentukan probabilitas *crossover* dan menentukan probabilitas mutasi.
2. Bangkitkan populasi awal
Membangkitkan sejumlah rute sebanyak ukuran populasi secara acak.
3. Melakukan penghitungan nilai fitness
Perhitungan nilai fitness dilakukan dengan menghitung jarak total dari kemungkinan solusi rute UAV yang dihasilkan. Memilih individu terbaik dengan cara memilih solusi rute UAV terpendek pada populasi yang ada.
4. Menyalin individu terbaik
Tahap ini dilakukan untuk menjaga agar individu terbaik akan tetap ada pada populasi berikutnya tanpa mengalami kawin silang atau mutasi.
5. Melakukan proses kawin silang
Tahapan pertama yang dilakukan yaitu dengan memilih individu-individu untuk berperan sebagai induk. Kemudian antar induk dilakukan kawin silang.
6. Mutasi
Melakukan proses mutasi dengan menukarkan urutan gen pada *chromosome* dalam suatu individu. Langkah ini bertujuan untuk memunculkan kemungkinan solusi baru. Namun diusahakan tidak terlalu banyak individu yang mengalami mutasi dengan menetapkan nilai peluang mutasi yang cukup kecil.

2.2. Tinjauan Pustaka

Pada bagian ini akan memuat beberapa penelitian yang telah ada dan dikembangkan sebelumnya. Penelitian-penelitian tersebut akan menjadi pembanding terhadap rancangan penelitian pada Tugas Akhir ini. Berikut beberapa judul *paper* ataupun penelitian yang berkaitan:

2.2.1. *Master-Slave Parallel Vector-Evaluated Genetic Algorithm for Unmanned Aerial Vehicle's Path Planning*

Pada penelitian ini menyatakan bahwa permintaan UAV dalam melakukan pemantauan bencana alam mengalami perluasan pemanfaatan menjadi diperuntukkan dalam memenuhi kebutuhan sipil. Penggunaan UAV dikatakan dapat mengurangi adanya korban manusia dalam melakukan misi dilingkungan berbahaya. Namun, banyak informasi mengenai adanya kecelakaan UAV disebabkan oleh faktor kesalahan manusia. Pada penelitian ini mengusulkan pengembangan otomasi UAV dalam hal perencanaan jalur menggunakan algoritma genetika. Pada penelitian ini algoritma genetika yang digunakan adalah *Master-Slave Parallel Vector-Evaluated Genetic Algorithm* (MSPVEGA). Hasil pada penelitian ini menyatakan bahwa algoritma MSPVEGA membutuhkan jumlah populasi dan generasi yang banyak. Dimana pengumpulan informasi membutuhkan ukuran populasi yang besar sedangkan kecepatan program sebagian besar dipengaruhi oleh jumlah generasi[25].

2.2.2. *Improved GASA Algorithm for Mutation Strategy UAV Path Planning*

Pada penelitian ini dirancang sebuah algoritma dalam melakukan optimasi perencanaan jalur. Optimasi dilakukan dengan melakukan penggabungan beberapa algoritma untuk menutupi kekurangan yang terjadi pada algoritma lainnya. Dimana dilakukan penggabungan antara algoritma genetika, algoritma *Simulated Annealing* (SA) dan algoritma *Targeted Mutation* (TM). Pada algoritma genetika mempunyai fitur yang baik dalam paralelisme dan optimasi *global*. Namun terdapat kekurangan dimana kecepatannya melambat ketika mendekati solusi optimal dan mudah terperangkap dalam *local optimum*. Pada algoritma *Simulated*

Annealing (SA) mempunyai kemampuan untuk terhindar dari *local optimum* dengan menggunakan teknologi pencarian acak didalamnya. *Targeted Mutation* (TM) merupakan strategi variabilitas evolusi diferensial yang baru dimana digunakan untuk meningkatkan strategi mekanisme mutasi pada algoritma genetika. Pada algoritma ini mampu meningkatkan keragaman variasi genetik dan mempercepat pencarian dalam kondisi mendekati solusi optimal. Penggabungan algoritma genetika dengan algoritma SA dan secara bersamaan dengan menambahkan strategi TM dapat mempercepat tercapainya kondisi optimum *global* dan menghambat prematuritas dari algoritma genetika[26]

2.2.3. Path Planning Based on Immune Genetic Algorithm for UAV

Pada penelitian ini membahas mengenai pengembangan dalam algoritma genetika. Dimana algoritma genetika pada umumnya mempunyai kekurangan dimana kecepatan mencapai konvergen yang rendah dan fenomena *premature* yang membuat terperangkap dalam *local optimum*. Untuk mengatasi kekurangan tersebut dilakukan penerapan operator *Immune* pada algoritma genetika. Proses seleksi individu dalam algoritma genetika tidak lagi hanya mengandalkan nilai fitness saja tetapi menggunakan operator *Immune*. Dimana operator *Immune* terdapat dua buah yaitu *Vaccination* dan *Immune selection*. Dimana pada *Vaccination* dilakukan modifikasi pada beberapa gen pada individu sedangkan pada *Immune Selection* dilakukan perbandingan nilai fitness antara individu anak dan individu kedua induk untuk menentukan kelayakan individu anak untuk bertahan. Pada penelitian ini menunjukkan bahwa dengan menambahkan operator *Immune* pada algoritma genetika dapat dengan efektif meningkatkan kecepatan konvergensi dan mencegah terjadinya fenomena *premature* pada algoritma genetika[27].

2.2.4. Vibrational Genetic Algorithm Based Path Planner for Autonomous UAV In Spatial Data Based Environments

Pada penelitian ini membahas mengenai algoritma yang efisien dalam melakukan perencanaan jalur terbang untuk UAV. Dimana

dilakukan pengembangan jenis algoritma genetika untuk menutupi kekurangan dari algoritma genetika pada umumnya dikarenakan fenomena *premature* yang mencegah tercapainya *global optimum*. Pengembangan dilakukan dengan menggunakan *Vibrational genetic algorithm*. Peneliti menyatakan bahwa tingkat populasi rendah dan siklus regenerasi yang pendek merupakan manfaat utama dari *Vibrational genetic algorithm*. Dengan menerapkan operator mutasi *Vibrational* kepada seluruh individu pada populasi, membuat terjadinya penyebaran individu. Oleh karena itu, membuat terjadinya kemungkinan kemampuan dapat keluar dari kondisi *local optimum* dan mencapai kondisi *global optimum* lebih cepat. Meskipun operator mutasi *vibrational* menyebabkan tingkat mutasi yang tinggi dan mengakibatkan distribusi individu secara acak, konsep elitism membuat distribusi individu berada pada jalan yang benar. Disisi lain, penyebaran individu yang luas memungkinkan penggunaan ukuran populasi yang kecil dalam proses genetik[28].

2.2.5. 2D Path Planning of UAVs with Genetic Algorithm in a Constrained Environment

Pada penelitian ini membahas mengenai perencanaan jalur untuk UAV dalam menghindari zona-zona yang terlarang pada suatu rute penerbangan. Untuk mengatasi permasalahan tersebut, peneliti melakukan optimasi dalam hal perencanaan jalur terbang. Namun, terdapat kesulitan utama dalam melakukan optimasi tersebut yaitu penyelesaian yang kompleks dan sangat susah apabila dilakukannya evaluasi kelayakan pada setiap vektor solusi yang dilakukan satu demi satu. Permasalahan tersebut bisa terselesaikan dengan melakukan optimasi menggunakan algoritma genetika. Menemukan solusi optimal untuk jalur penerbangan UAV sama halnya dengan memecahkan permasalahan *Travelling Salesman Problem* (TSP). Dimana dalam TSP yaitu untuk melakukan optimasi pada suatu rute perjalanan untuk mengunjungi beberapa titik kota yang mana setiap kota hanya dikunjungi satu kali dan kembali ke titik awal keberangkatan. Optimasi dilakukan untuk menemukan solusi jalur yang menghasilkan jarak tempuh total terpendek. Jika dilihat terdapat titik permasalahan yang sama antara

perencanaan jalur terbang UAV dengan TSP. Dimana setiap titik koordinat *waypoint* merepresentasikan titik masing-masing kota pada TSP dan UAV harus melakukan perjalanan seperti *salesman* dan kembali ke titik awal dengan menghasilkan panjang jalur total terpendek. Untuk teknik dalam menghindari zona terlarang peneliti mengilustrasikan zona tersebut dalam sebuah area kotak dengan luas tertentu. Kemudian dilakukan analisa keberadaan titik potong antara jalur penerbangan UAV dengan ilustrasi area berbentuk kotak yang telah dibuat. Apabila terdapat titik potong maka akan dilakukan pergantian atau penambahan titik baru yang berada diluar area berbentuk kotak tersebut. Dengan demikian peneliti berhasil melakukan optimasi jalur penerbangan UAV dalam menghindari zona-zona terlarang atau berbahaya[29].

2.2.6. *The Comparison of Four UAV Path Planning Algorithms Based on Geometry Search Algorithm*

Pada penelitian ini penulis melakukan perbandingan terhadap suatu algoritma perencanaan rute perjalanan berdasarkan pencarian geometri. Algoritma yang dilakukan perbandingan adalah algoritma Dijkstra, algoritma Floyd, algoritma A* dan algoritma *ant colony*. Proses perhitungan dimulai dengan mengubah model jalur 3D dengan model jalur 2D dengan algoritma yang memperhitungkan karakteristik medan suatu jalur. Dalam penelitian ini keempat algoritma tersebut dilakukan perbandingan terhadap variabel waktu dan panjang jalur. Dari hasil simulasi, pada penelitian ini menghasilkan kesimpulan bahwa algoritma Dijkstra lebih baik dari ketiga algoritma lainnya. Namun, pada tugas akhir ini tidak menggunakan algoritma Dijkstra dikarenakan algoritma tersebut menggunakan prinsip *greedy*, yaitu mencari solusi optimum pada setiap langkah yang dilalui. Hal tersebut akan membuat kompleksitas waktu menjadi cukup besar[30].

2.2.7. *Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm*

Pada penelitian ini penulis melakukan perencanaan jalur terbang UAV dalam menghindari halangan-halangan yang ada. Penulis

melakukan perencanaan optimasi jalur menggunakan algoritma ACO (*ant colony optimization*). Algoritma ACO didasarkan oleh tiga buah perilaku cerdas yaitu komunikasi semut berdasarkan feromon, perilaku memori semut dan kegiatan pengelompokan oleh semut. Peneliti melakukan percobaan perencanaan jalur dengan meningkatkan jumlah halangan dari setiap percobaan. Dengan demikian didapatkan hasil, algoritma ACO dapat menghasilkan perencanaan jalur yang optimal seiring dengan meningkatnya jumlah halangan atau tingkat kompleksitas yang dihadapi. Terdapat perbedaan dengan penelitian yang dilakukan pada Tugas Akhir ini, dimana pada Tugas Akhir ini UAV tidak dihadapi oleh peningkatan kompleksitas yang diakibatkan karena adanya rintangan dalam jalur penerbangan UAV[31].

2.2.8. UAV Path Planning based on Bidirectional Sparse A* Search Algorithm

Pada penelitian ini peneliti melakukan perencanaan jalur menggunakan algoritma BSAS (*Bidirectional Sparse A* Search*). Algoritma BSAS merupakan pengembangan dari algoritma SAS. Algoritma SAS mempunyai kemampuan untuk mengurangi waktu dan lingkup area pencarian dalam menghasilkan perencanaan jalur yang optimal. Namun, terdapat beberapa kekurangan pada algoritma SAS dimana dalam beberapa kasus hasil perencanaan jalur tidak optimal ketika adanya penambahan persyaratan berupa munculnya zona terlarang yang tidak boleh dilewati. Dengan penggunaan algoritma BSAS, algoritma SAS dikembangkan sehingga perencanaan jalur dapat optimal dalam kondisi adanya zona terlarang. Algoritma BSAS mempunyai kemampuan penambahan titik *node* dalam mensiasati penghindaran zona terlarang tersebut. Hasil penelitian menunjukkan algoritma BSAS dapat menghasilkan perencanaan jalur yang optimal yang dapat menghindari zona terlarang[32].

2.2.9. *Autonomous Path Planning by Unmanned Aerial Vehicle (UAV) for Precise Monitoring of Large-Scale PV plants*

Pada penelitian ini dilakukan perencanaan jalur autopilot untuk melakukan *monitoring* pada pembangkit listrik tenaga surya. *Monitoring* dimaksudkan untuk melakukan inspeksi terhadap pembangkit tersebut. Perencanaan jalur dilakukan dengan mengambil foto keseluruhan area pembangkit dan dilakukan pengolahan gambar. Pengolahan gambar dilakukan dengan menggunakan bahasa pemrograman python dan menggunakan fungsi OpenCV seperti *filter Gaussian*, *filter HSV* dan metode *Canny*. Dengan dilakukan pengolahan gambar akan didapatkannya titik-titik koordinat jalur dimana UAV akan terbang atau bermanuver. Dimana terdapat perbedaan metode yang dilakukan dibandingkan pada Tugas Akhir ini. Pada penelitian ini pengolahan gambar dilakukan untuk perencanaan keseluruhan jalur penerbangan UAV, sedangkan pada Tugas Akhir ini pengolahan gambar dilakukan untuk mendapatkan titik koordinat pada gambar yang diindikasikan sebagai zona *blank spot*[33].

2.2.10. *A Path Planning for One UAV Based on Geometric Algorithm*

Pada penelitian ini dilakukan perencanaan jalur menggunakan algoritma *geometric*. Algoritma *geometric* yang diterapkan pada penelitian ini dapat mengintegrasikan informasi geometris jarak dan model risiko lingkungan UAV dalam membuat model perencanaan jalur. Dalam menyelesaikan permasalahan tersebut dilakukan penerapan metode fungsi ancaman berkelanjutan dan untuk mengurangi waktu pemrosesan algoritma dilakukan metode diskritisasi peta. Hasil percobaan menunjukkan bahwa algoritma perencanaan jalur penerbangan UAV berdasarkan *geometric* memiliki ketahanan yang baik dan kemampuan adaptif. Dimana terdapat perbedaan permasalahan apabila dibandingkan pada Tugas Akhir ini, dimana tidak adanya model risiko lingkungan sehingga tidak diperlukannya fungsi ancaman berkelanjutan[34].

2.2.11. Path planning of mobile robot with path rule mining based on GA

Pada penelitian ini dilakukan perencanaan jalur untuk aplikasi pada *mobile robot*. Perencanaan jalur dilakukan menggunakan algoritma genetika dengan mempertimbangkan peningkatan evolusi-GA untuk mendapatkan solusi yang lebih baik. Skema *path rule mining* diusulkan pada penelitian ini dengan maksud untuk meningkatkan metode evolusi dalam proses penerapan algoritma genetika. Dari hasil pengujian simulasi didapatkan bahwa penggunaan algoritma genetika modifikasi akan membutuhkan waktu yang lebih sedikit apabila dibandingkan dengan algoritma genetika pada umumnya dalam menghasilkan solusi yang konvergen. Kesimpulan dari hasil penelitian ini bahwa algoritma genetika modifikasi dapat meningkatkan kemampuan pencarian ruang lokal dalam masalah pencarian global. Dimana terdapat perbedaan objek pengimplementasian apabila dibandingkan dengan penelitian pada Tugas Akhir ini[35].

2.2.12. Ship Path Planning Based on Improved Particle Swarm Optimization

Pada penelitian ini dilakukan analisa mendalam mengenai permasalahan perencanaan jalur kapal. Dimana dalam perencanaan jalur kapal akan fokus dalam menyelesaikan dua isu antara lain kapal dapat menghindari segala bentuk halangan dari titik keberangkatan sampai titik tujuan dan menghasilkan jalur kapal terpendek dan menghabiskan konsumsi bahan bakar yang lebih sedikit. Untuk menyelesaikan permasalahan tersebut dilakukan pengembangan algoritma PSO (*particle swarm optimization*) berdasarkan algoritma CMPSO (*chaos multi-population particle swarm optimization*). Pengembangan dilakukan dengan menerapkan model pemetaan logistik yang diadopsi dari teori *chaotic*. Penerapan tersebut dimaksudkan untuk membangkitkan populasi awal yang memiliki sifat *uniform* sehingga meningkatkan efisiensi eksekusi algoritma tersebut. Kesimpulan dari penelitian ini adalah algoritma CMPSO akan membutuhkan waktu kalkulasi yang lebih sedikit dan dapat terhindar dari keadaan lokal optimum dalam keadaan lingkungan yang kompleks. Pada penelitian ini terdapat perbedaan objek

pengimplementasian dan tidak adanya isu dalam menghindari suatu halangan apabila dibandingkan pada pengerjaan Tugas Akhir ini[36].

2.2.13. UAV Path Planning Based on Chaos Ant Colony Algorithm

Pada penelitian ini dilakukan perencanaan jalur dengan melakukan peningkatan performa dari algoritma ACA (*ant colony algorithm*). Dimana pada algoritma ACA mempunyai kelemahan yaitu tingkat konvergensi yang rendah dan mudah terjatuh dalam ekstrem lokal. Untuk mengatasi permasalahan tersebut pada penelitian ini diusulkan suatu pengembangan algoritma ACA yaitu algoritma CACA (*chaos ant colony algorithm*). Pengembangan dilakukan dengan melakukan penggabungan strategi pencarian *chaotic* berupa metode sistem pemetaan logistik dan faktor penuntun tujuan. Kesimpulan dari penelitian ini bahwa penerapan algoritma CACA dapat secara efektif mengatasi kekurangan pada algoritma ACA dan penerapan algoritma CACA dapat dengan cepat menemukan arah jalur pencarian, dapat secara efektif meningkatkan efisiensi pencarian, dan hasil perencanaan CACA dapat memenuhi kendala ancaman radar dan radius putaran minimum UAV. Pada penelitian ini terdapat faktor area ancaman yang dijadikan variabel perhitungan dalam kalkulasi perencanaan jalur dimana faktor tersebut tidak diperlukan dalam implementasi perencanaan jalur yang dilakukan pada Tugas Akhir ini[37].

2.2.14. 3D Environment Modeling with Height Dimension Reduction and Path Planning for UAV

Pada penelitian ini dilakukan perencanaan jalur menggunakan algoritma A* dengan melakukan pemodelan lingkungan secara 3D. Pemodelan dilakukan dengan menggunakan metode *grids* dimana adanya konversi lingkungan 3D menjadi 2D. Dengan cara ini, jumlah *grid* selama pemodelan di lingkungan 3D dan ukuran grid bisa berkurang banyak, dan terjadinya peningkatan akurasi pemodelan lingkungan. Dalam melakukan konversi tersebut dilakukan penerapan metode pemodelan lingkungan dengan metode reduksi dimensi tinggi. Kesimpulan pada penelitian ini adalah metode pemodelan lingkungan

dengan reduksi dimensi tinggi dapat mengatasi konflik antara akurasi model lingkungan dan efisiensi algoritma perencanaan. Namun, pada Tugas Akhir ini tidak dilakukan pemodelan lingkungan seperti yang dilakukan pada penelitian diatas, dikarenakan lingkungan tempat pengujian terbebas dari gedung-gedung ataupun variabel lain yang dapat mengganggu proses perencanaan optimasi rute perjalanan[38].

2.2.15. *Experimental performance evaluation of various path planning algorithms for obstacle avoidance in UAVs*

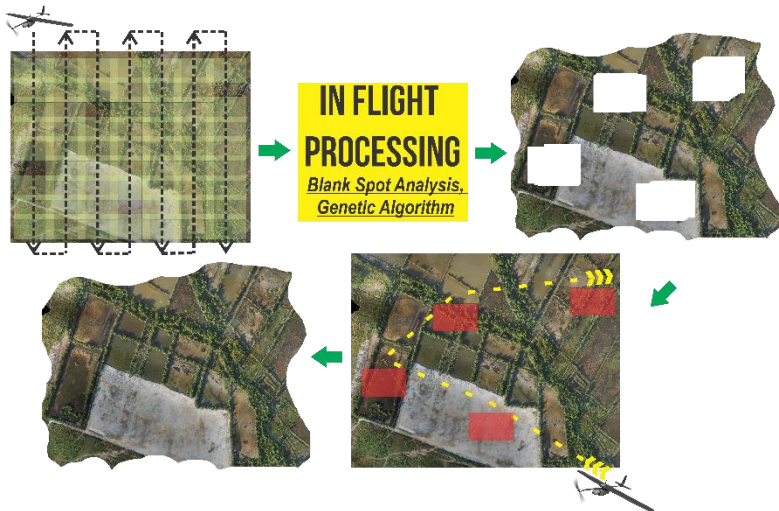
Pada penelitian ini dilakukan komparasi terhadap beberapa algoritma perencanaan rute perjalanan dalam menghindari rintangan statis. Algoritma yang dilakukan komparasi antara lain algoritma Dijkstra, algoritma A*, algoritma *artificial potential field*, algoritma RRT dan algoritma RRT*. Parameter yang dilakukan komparasi adalah panjang jalur, maksimum sudut deviasi, dan jarak aman antara jalur penerbangan dengan area rintangan. Dalam melakukan pengujian peneliti menggunakan UAV jenis *fixed wing*. Hasil dari penelitian ini adalah dimana dalam pengujian didapatkan algoritma Dijkstra menghasilkan waktu terlalu lama dalam menghasilkan rute dan menghasilkan rute yang memiliki jarak sangat dekat dengan zona rintangan, algoritma A* menghasilkan rute yang paling layak dikarenakan jumlah titik koordinat yang banyak, algoritma *artificial potential field* menghasilkan rute penerbangan yang paling aman dikarenakan jarak rute penerbangan dengan zona rintangan yang jauh dan algoritma RRT menghasilkan rute yang lancar dan jelas tetapi menghasilkan jarak terpanjang. Di antara lima algoritma yang dievaluasi, algoritma RRT* menunjukkan hasil yang paling menguntungkan secara keseluruhan, karena kelancaran jalur dan jarak terkecil yang dilalui oleh UAV. Terdapat perbedaan variabel permasalahan yang diangkat pada penelitian ini dengan pengerjaan pada Tugas Akhir ini, dimana tidak adanya variabel penghindaran rintangan yang dimasukkan pada perancangan sistem optimasi pada Tugas Akhir ini[39].

Halaman ini sengaja dikosongkan

BAB 3 PERANCANGAN SISTEM



Gambar 3.1. Ilustrasi Proses Pengambilan Foto



Gambar 3.2. Ilustrasi Penerapan Analisa *Blank Spot* dan Algoritma Genetika

Pada bab ini akan dibahas mengenai perancangan pengembangan sistem autopilot UAV yang akan dikerjakan pada Tugas Akhir ini. Sistem autopilot yang dimaksud pada Tugas Akhir ini adalah sistem kontrol otomatis yang digunakan pada UAV untuk melakukan misi berupa pemotretan lahan dari udara. UAV akan dilengkapi sebuah kamera untuk melakukan misi penerbangan diatas lahan dengan luasan tertentu dan melakukan pemotretan terhadap lahan dibawahnya. Keluaran dari misi tersebut adalah didapatkannya kumpulan foto-foto lahan yang tercakup. Dimana foto-foto tersebut selanjutnya akan melalui proses *stitching* dan menghasilkan sebuah foto peta daerah tercakup. Tingkat kesempurnaan peta yang dihasilkan bergantung terhadap kelengkapan foto yang didapatkan selama misi penerbangan. Dalam misi penerbangan tersebut, UAV harus mampu terbang sesuai jalur yang direncanakan agar didapatkan data foto seluruh bagian daerah cakupan yang diinginkan. Ketidaklengkapan foto yang didapatkan selama misi penerbangan akan menghasilkan ketidaksempurnaan pada foto peta yang dihasilkan yang diakibatkan karena adanya *zona blank spot*. Untuk meminimalisir *zona blank spot*, pada Tugas Akhir ini dilakukan pengembangan sistem autopilot UAV dan pengimplementasian logika algoritma genetika terhadap mekanisme penyusunan jalur penerbangan UAV. Selama proses penerbangan, UAV akan terus berkomunikasi terhadap GCS untuk menerima dan memberikan data-data yang dibutuhkan. Dengan adanya komunikasi dua arah pada UAV dan GCS(*Ground Control System*), UAV akan melakukan segala perintah yang diberikan oleh GCS dan GCS akan mendapatkan segala data yang diinginkan pada UAV. Dengan memanfaatkan komunikasi dua arah ini, GCS dapat mengetahui kesesuaian pergerakan UAV terhadap jalur penerbangan yang telah direncanakan dengan mengolah data GPS pada UAV selama proses misi penerbangan. Dengan menganalisa setiap pergerakan UAV, maka akan didapatkan titik-titik koordinat dimana terjadinya ketidaksesuaian akan jalur yang telah direncanakan terhadap pergerakan aktual UAV. Titik-titik koordinat yang dianggap belum terlewati oleh UAV akan dikumpulkan dan dicatat sampai UAV telah menyelesaikan misi penerbangan. Kemudian kumpulan titik koordinat tersebut dilakukan pemrosesan menggunakan algoritma genetika. Algoritma genetika digunakan untuk melakukan penyusunan terhadap

kumpulan koordinat yang dianggap tidak terlewat oleh UAV sehingga didapatkan susunan koordinat yang menghasilkan jarak tempuh terpendek. Setelah misi penerbangan awal telah selesai dieksekusi dan sebelum UAV melakukan misi pendaratan, GCS akan mengirimkan perintah kepada UAV untuk mengeksekusi kumpulan koordinat yang telah melalui pemrosesan menggunakan algoritma genetika tersebut untuk dapat dilakukan pemotretan ulang. Kemudian apabila seluruh koordinat tersebut berhasil dieksekusi maka UAV dapat melakukan misi pendaratan. Dengan dilakukan pengimplementasian menggunakan algoritma genetika ini, diharapkan UAV dapat melakukan misi penerbangan dengan menghabiskan waktu dan daya baterai yang lebih efisien dan dapat meminimalisir terjadinya fenomena zona *blank spot* pada foto peta yang dihasilkan.

3.1. Perencanaan Jalur Terbang

Dalam melakukan suatu misi penerbangan secara otomatis, UAV akan dilengkapi dengan kumpulan-kumpulan data yang menunjangnya. Salah satu data yang dibutuhkan adalah data koordinat atau *waypoint*. Dengan dilengkapi data koordinat, UAV akan dapat menentukan kearah mana ia akan melakukan misi penerbangan. Dimana perencanaan ini disesuaikan dengan misi penerbangan yang diinginkan pengguna. Perbedaan misi penerbangan akan menghasilkan perencanaan jalur terbang yang berbeda pula. Dalam Tugas Akhir ini UAV yang digunakan akan melakukan misi penerbangan berupa pemotretan lahan dibawahnya melalui udara. Maka jika kita cermati terdapat *point* penting dalam melakukan misi penerbangan ini yaitu UAV harus dapat terbang melewati seluruh lahan yang diinginkan untuk mendapatkan foto setiap bagian dari lahan dibawahnya.

3.1.1. Faktor Kamera

Dalam melakukan misi penerbangan ini kamera merupakan komponen yang berperan sangat penting pada UAV. Oleh karenanya kamera merupakan faktor yang harus dipertimbangkan dalam melakukan perencanaan jalur terbang pada misi penerbangan ini. Salah satu nilai yang diperhitungkan dalam melakukan perencanaan jalur terbang ini adalah luas cakupan foto sebuah kamera yang digunakan. Dengan memperhitungkan nilai luas cakupan foto tersebut maka kita akan dapat

menentukan arah ataupun pola dan ketinggian untuk UAV melakukan misi penerbangan.

Tabel 3.1. Spesifikasi Kamera

No	Parameter	Xiaomi Yi Cam
1	Focal Length (mm)	2.73
2	Pixel Size (μm)	1.12
3	Tinggi Terbang (m)	100
4	Sensor Width (mm)	6.08
5	Sensor Height (mm)	4.56
6	Image Width (pixel)	4608
7	Image Height (pixel)	3456
8	Overlap	60%
9	Sidelap	60%

Untuk mendapatkan luas cakupan sebuah foto dilakukan perhitungan menggunakan rumus *Ground Sampling Distance* (GSD)[40].

$$GSD(\text{cm/pixel}) = \frac{\text{Sensor Width (mm)} \times \text{Flight Height(m)} \times 100}{\text{Focal Length(mm)} \times \text{Image Width(pixel)}} \quad (3.1)$$

$$GSD = \frac{6.08 \text{ mm} \times 100 \text{ m} \times 100}{2.73 \text{ mm} \times 4608 \text{ pixel}} = 4.83 \text{ cm/pixel}$$

Maka dari nilai diatas akan didapatkan nilai luasan cuplikan area dalam satu buah foto, seperti dibawah ini:

$$\text{Panjang(m)} = \text{Image Width(pixel)} \times GSD \quad (3.2)$$

$$\text{Panjang} = 4608 \times 4.83 = 222.71 \text{ m}$$

$$\text{Lebar(m)} = \text{Image Height(pixel)} \times GSD \quad (3.3)$$

$$\text{Lebar} = 3456 \times 4.83 = 167.03 \text{ m}$$

$$\begin{aligned} \text{Luasan}(m^2) &= P(m) \times L(m) \quad (3.4) \\ &= 222.71 \times 167.03 = 37200.01 \text{ m}^2 \end{aligned}$$

Kemudian dilakukan perhitungan jumlah stasiun titik pemotretan untuk mendapatkan jumlah total perkiraan foto yang akan dihasilkan seperti dibawah ini:

$$\begin{aligned} W(m) &= \text{Panjang}(m) - (\text{Sidelap} \times \text{Panjang}(m)) \quad (3.5) \\ &= 222.71 - (60\% \times 222.71) = 89.084 \text{ m} \end{aligned}$$

$$\begin{aligned} B(m) &= \text{Lebar}(m) - (\text{Overlap} \times \text{Lebar}(m)) \quad (3.6) \\ &= 167.03 - (60\% \times 167.03) = 66.8 \text{ m} \end{aligned}$$

$$\begin{aligned} \Sigma \text{Foto}(\text{Horizontal}) &= \frac{\text{Panjang Wilayah}}{B} \quad (3.7) \\ &= \frac{500}{66.8} = 7.48 \approx 8 \end{aligned}$$

$$\begin{aligned} \Sigma \text{Foto}(\text{Vertikal}) &= \frac{\text{Lebar Wilayah}}{W} \quad (3.8) \\ &= \frac{500}{89.084} = 5.61 \approx 6 \end{aligned}$$

$$\Sigma \text{Foto} = \Sigma \text{Foto}(\text{horizontal}) \times \Sigma \text{Foto}(\text{vertikal}) \quad (3.9)$$

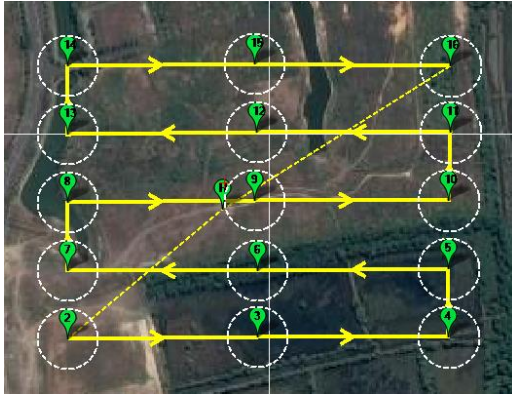
$$\text{Total Foto} = 6 \times 8 = 48 \text{ Foto}$$

Ket : B = Jarak antar stasiun pemotretan

W = Jarak antar jalur terbang yang berhimpitan

3.1.2. Perhitungan Jalur Terbang

Seperti yang dijelaskan sebelumnya bahwa setiap misi penerbangan UAV yang berbeda akan membutuhkan perencanaan jalur terbang yang berbeda pula. Dalam Tugas Akhir ini, UAV memiliki misi penerbangan untuk melakukan pemotretan lahan dari udara. Dalam melakukan misi ini, terdapat beberapa jenis pola penerbangan yang disediakan dalam *mission planner*. Namun pada Tugas Akhir ini digunakan jenis pola *simple grid* Seperti Gambar 3.3 dibawah ini.



Gambar 3.3. Pola *Simple Grid*

Dalam pola rute penerbangan *simple grid* seperti Gambar 3.3 diatas perhitungan dilakukan untuk mendapatkan jarak antar jalur terbang *horizontal*. Nilai perhitungan tersebut akan mengacu pada luas cakupan kamera yang didapat dan nilai *sidelap* yang diinginkan. Seperti contoh apabila persentase nilai *sidelap* yang diinginkan sebesar 60%. Berikut penjelasan mengenai perhitungan jalur terbang:

$$W(m) = Panjang(m) - (Sidelap \times Panjang(m)) \quad (3.10)$$

$$= 222.71 - (60\% \times 222.71) = 89.084 \text{ m}$$

$$Jumlah \text{ Jalur Horizontal} = \frac{Lebar \text{ Wilayah}}{w} \quad (3.11)$$

$$= \frac{500}{89.084} = 5.61 \approx 6$$

Maka pola jalur penerbangan yang akan dibuat mempunyai 6 jalur *horizontal* dengan jarak masing-masing jalur sejauh 89.084 m.

3.1.3. Penyimpanan Rencana Terbang



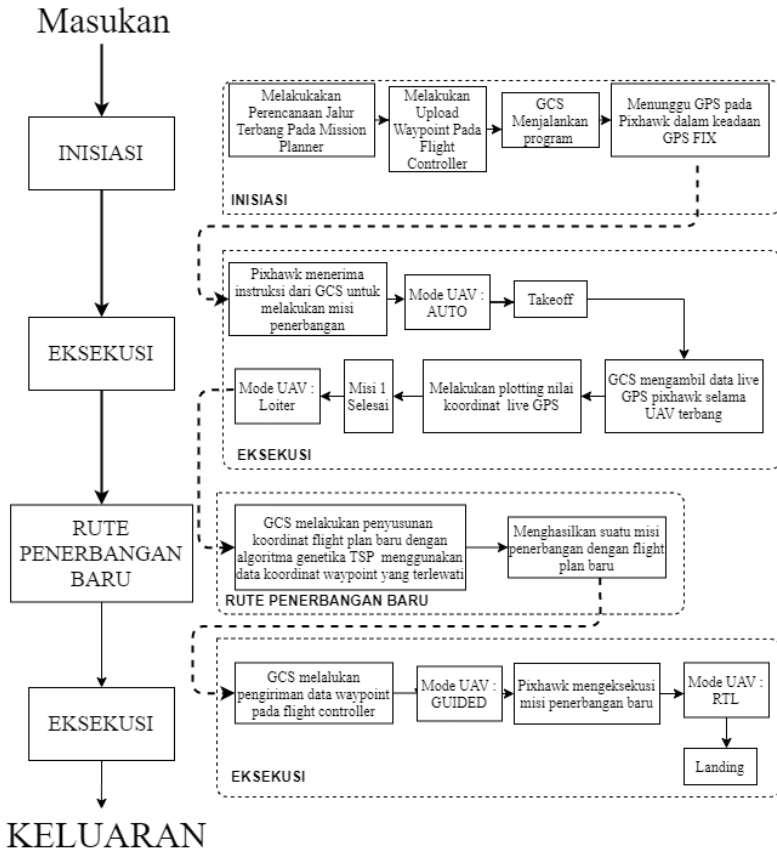
Gambar 3.4. Prosedur Penyimpanan *Waypoint*

Setelah dilakukan perencanaan rute penerbangan berdasarkan perhitungan seperti yang dibahas sebelumnya, yaitu melakukan penyimpanan rencana rute penerbangan. Hal pertama yang perlu dilakukan adalah merealisasikan rencana rute penerbangan pada *software* Mission Planner. Kemudian dilakukan penyimpanan rute penerbangan seperti yang dijelaskan pada Gambar 3.4 diatas.

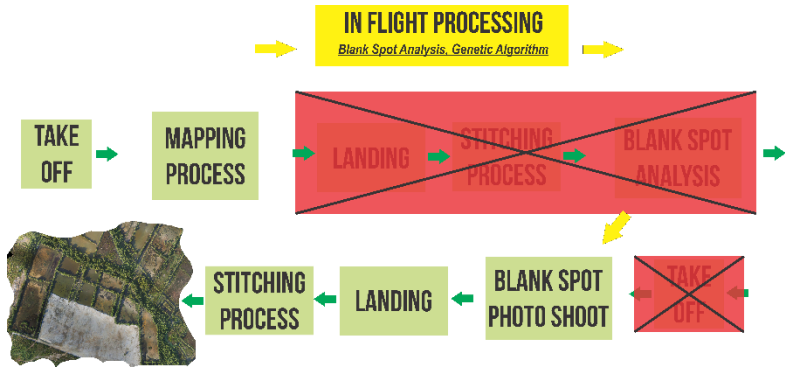
Kemudian rencana rute penerbangan akan tersimpan dalam format *.txt*. Agar file dapat terbaca oleh program yang dikerjakan pada tugas akhir ini, dilakukan konversi format *file* menjadi *.xlsx* dan disimpan dalam satu *folder* yang sama dengan *file* program optimasi.

3.2. Perancangan Optimasi

Pada tahap ini dilakukan pemodelan sistem optimasi yang akan diterapkan pada pengembangan sistem autopilot pada Tugas Akhir ini, seperti yang dapat dijelaskan di bawah ini.

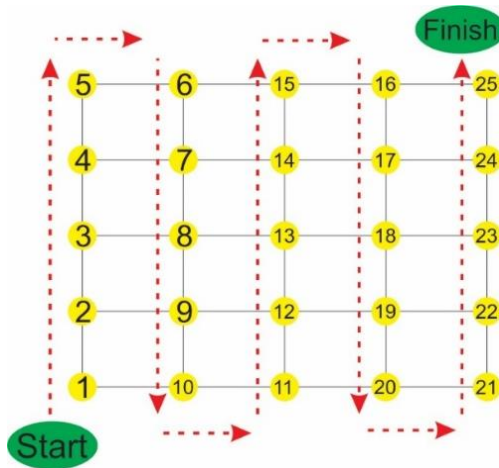


Gambar 3.5. Diagram Blok Keseluruhan Sistem



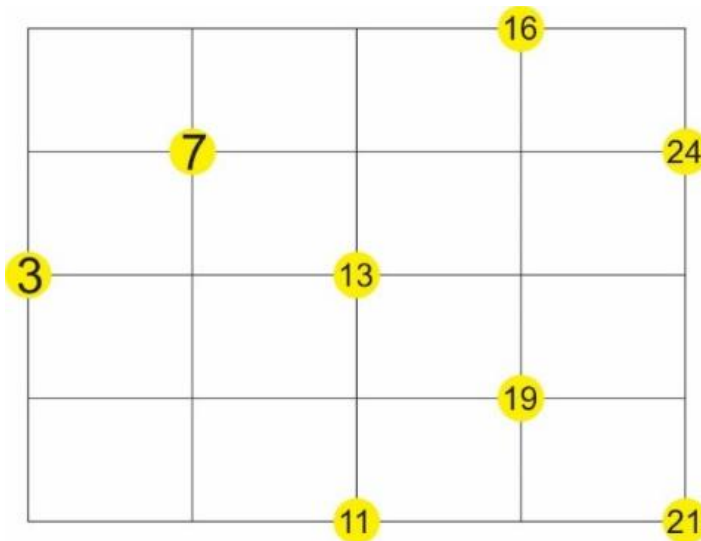
Gambar 3.6. Diagram Blok Mekanisme Pemetaan Menggunakan Sistem Autopilot Hasil Optimasi

- Dalam melakukan misi pemetaan, sebuah wahana UAV akan mengeksekusi seluruh *waypoint* (*flight plan*) yang diperintahkan oleh GCS. Diibaratkan sebuah *flight plan* berbentuk *mesh* 5x5 seperti pada Gambar 3.7 dibawah ini. Lingkaran berwarna kuning menunjukkan titik-titik koordinat pada *flight plan* dan angka pada lingkaran berwarna kuning menunjukkan urutan koordinat yang akan dilewati UAV.



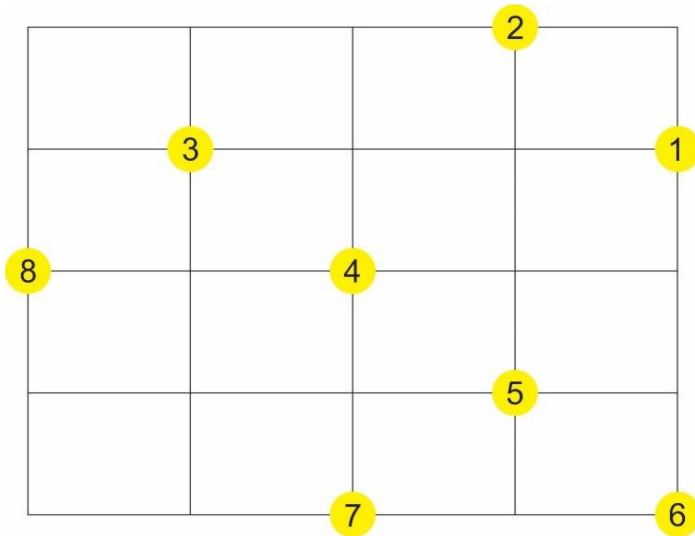
Gambar 3.7. Ilustrasi Koordinat Jalur Penerbangan

- Setelah misi pemetaan tahap pertama selesai dimana UAV sudah sampai pada titik koordinat *waypoint* terakhir. Kemudian GCS akan melakukan proses tahap menganalisa keberadaan zona *blank spot*. Hasil dari pemrosesan tersebut akan didapatkan data titik-titik koordinat yang dianggap tidak terlewati(diindikasikan sebagai zona *blank spot*) oleh wahana UAV. Diibaratkan pada Gambar 3.8 *mesh* dibawah ini.



Gambar 3.8. Ilustrasi Koordinat Zona *Blank spot*

- Seluruh data koordinat yang dihasilkan pada proses sebelumnya akan disimpan dan dilanjutkan pada proses tahap optimasi menggunakan algoritma genetika. Dalam tahap ini akan didapatkan urutan koordinat yang harus dilewati UAV dalam misi penerbangan selanjutnya.



Gambar 3.9. Ilustrasi Susunan Koordinat Misi Penerbangan Baru

3.2.1. Menganalisa Zona *Blank Spot*

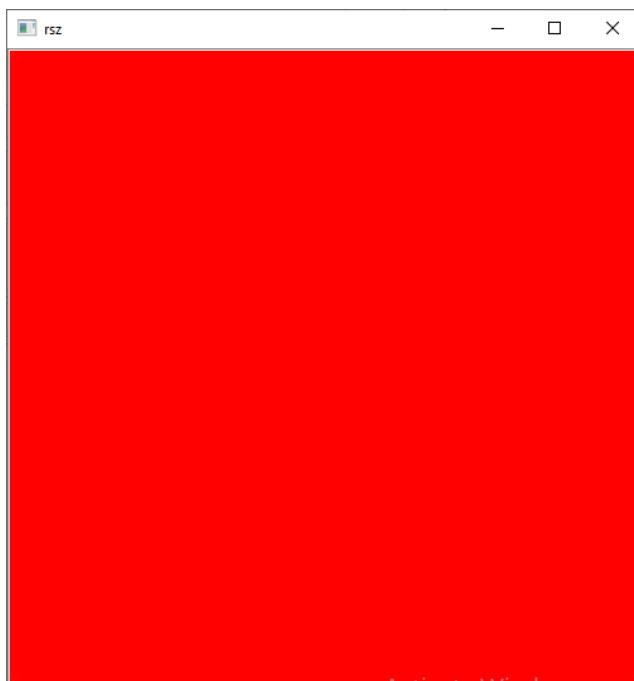
Pada pemrosesan tahap ini, pergerakan UAV menjadi suatu titik fokus utama. Dimana pergerakan UAV dapat diamati apabila terdapat data posisi selama UAV melakukan misi penerbangan. Data posisi UAV didapatkan dengan mengolah data GPS yang terdapat pada UAV. Untuk mendapatkan data gps tersebut dilakukan komunikasi menggunakan protokol MAVLINK yang telah dikemas dalam *library* bernama *dronekit* selama UAV mengudara. Dengan memanfaatkan data gps tersebut analisa pergerakan UAV akan menghasilkan data ada atau tidaknya keberadaan zona *blank spot*. Tahap analisa dilakukan dengan menggunakan pemrograman berbahasa python dan memanfaatkan *library* OpenCV. Dalam melakukan analisa tersebut terdapat beberapa tahapan yang akan dijelaskan seperti dibawah ini.

- Tahap 1

Pergerakan UAV selama misi penerbangan akan dicatat dengan menggambarkan kembali seluruh rute perjalanan yang ditempuh oleh UAV tersebut. Untuk menggambarkan kembali rute perjalanan UAV

tersebut, area lahan yang akan dilakukan pemotretan diilustrasikan menjadi sebuah *canvas* dengan ukuran terskala dari luasan area lahan aktual. Pada Tugas Akhir ini area yang dilakukan pemotretan berluas 500x500 m^2 oleh karenanya ukuran *canvas* yang akan dibuat berluas 500x500 pixel. Pembuatan *canvas* dilakukan dengan memanggil gambar yang terdapat pada *folder* program dan dilakukan *resize* terhadap ukuran gambar tersebut. Berikut program untuk membuat *canvas* beserta hasilnya:

```
im = cv2.imread("merah.jpg")
rsz= cv2.resize(im,(500, 500),interpolation =
cv2.INTER_NEAREST)
```



Gambar 3.10. Hasil *Canvas*

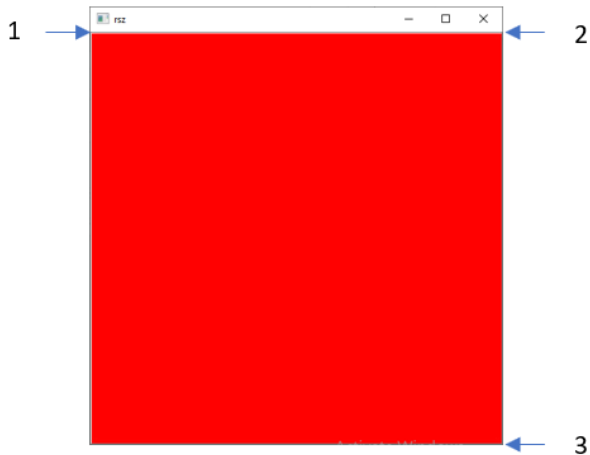
- Tahap 2

Pada tahap ini dilakukan penyimpanan dan penyesuaian titik koordinat aktual batas area lahan pemetaan terhadap titik koordinat batas pada *canvas* yang dibuat. Berikut program beserta ilustrasinya pada Gambar 3.11 dan 3.12.

```
lat1=-7.2821460
long1=112.8179169
lat2=-7.2821460
long2=112.8224490
lat3=-7.286642
long3=112.8224490
deltalong=long2-long1
deltalat=lat2-lat3
```



Gambar 3.11. Titik Koordinat Batas Area Cakupan



Gambar 3.12. Penyesuaian Batas Titik Koodinat *Canvas*

- Tahap 3

Pada tahap ini dilakukan pengambilan data GPS pada UAV selama misi penerbangan berlangsung. Berikut program yang digunakan untuk mengambil data GPS pada UAV:

```
live_gps = vehicle.location.global_frame
latitude = live_gps.lat
longitude = live_gps.lon
```

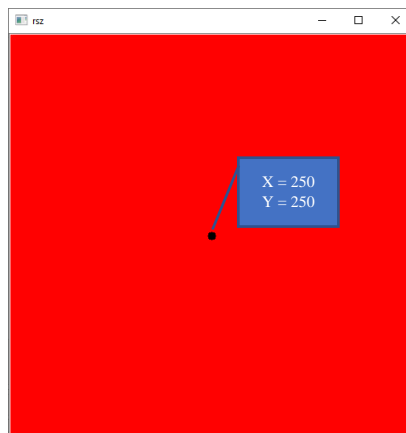
Kemudian data GPS yang didapatkan dari UAV selama misi penerbangan akan dilakukan perhitungan untuk dapat direpresentasikan dalam *canvas* yang telah dibuat. Koordinat bumi yang didapatkan akan diubah menjadi nilai koordinat kartesian pada *canvas*. Berikut program dan ilustrasi dari pengimplementasiannya:

```
def bumiTOcanvas(latlive, longlive):
    xx=(500*(longlive-long1))/deltalong
    yy=(500*(lat1-latlive))/deltalat
    return xx,yy
```

Representasi nilai koordinat bumi pada *canvas* dikatakan benar apabila posisi titik koordinat kartesian pada *canvas* memiliki kesesuaian posisi pada koordinat bumi di area lahan yang diinginkan. Berikut Gambar 3.13 dan Gambar 3.14 sebagai ilustrasi percobaan.



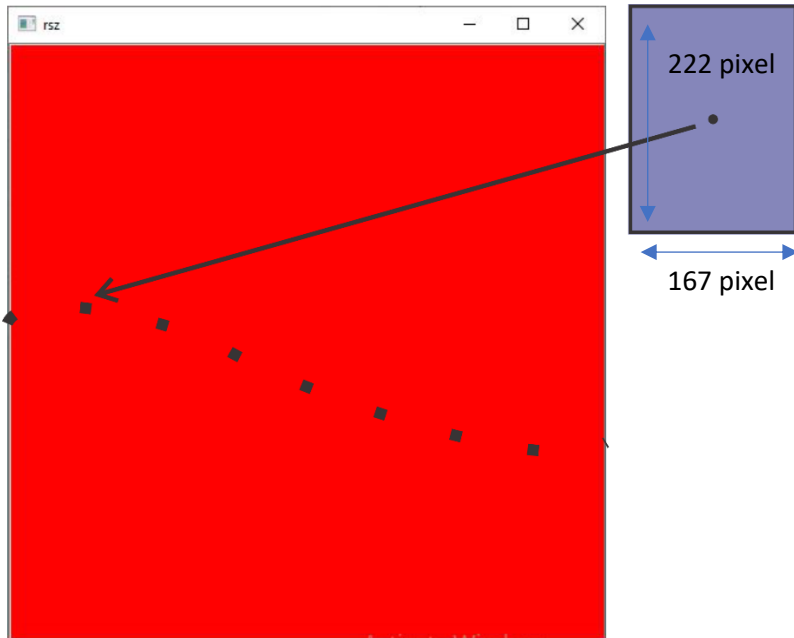
Gambar 3.13. Koordinat Titik Pusat Area Cakupan



Gambar 3.14. Hasil Representasi Koordinat Bumi pada *Canvas*

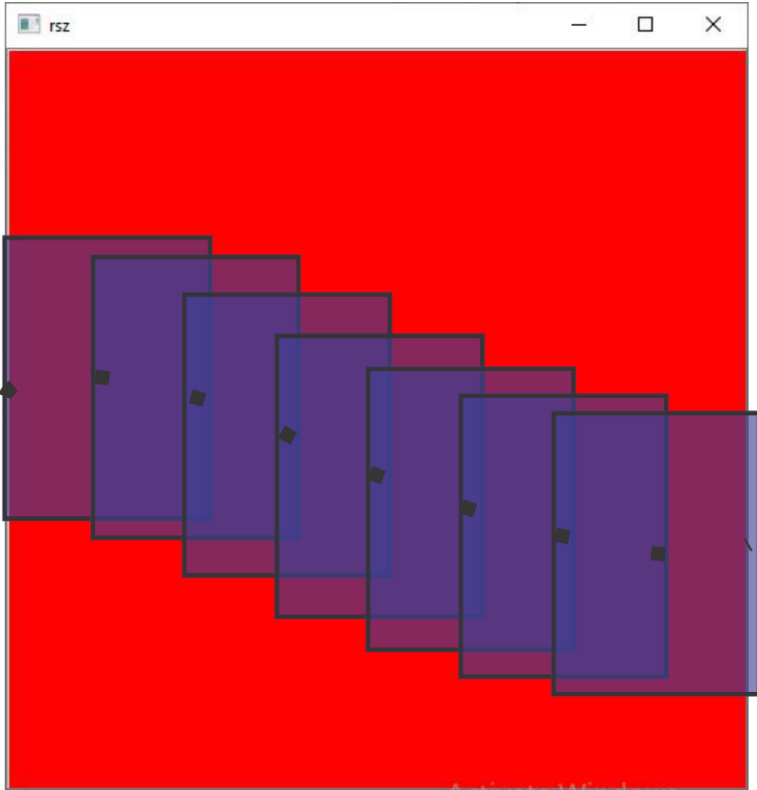
Setelah didapatkan nilai koordinat kartesian tersebut dilakukan penggambaran jalur pergerakan UAV pada *canvas*. Penggambaran jalur direpresentasikan menggunakan sebuah gambar kotak berwarna biru

dengan titik pusat berasal dari nilai konversi koordinat bumi menjadi koordinat *canvas* dan ukurannya yang disesuaikan dengan luasan cakupan kamera dalam 1 foto berdasarkan perhitungan GSD sebelumnya. Jika dilihat pada Gambar 3.15 dibawah ini, apabila diibaratkan titik-titik pada *canvas* merupakan titik koordinat posisi UAV yang berhasil terekam oleh GCS dan kotak biru merupakan luasan cakupan kamera yang telah dihitung menggunakan rumus GSD.



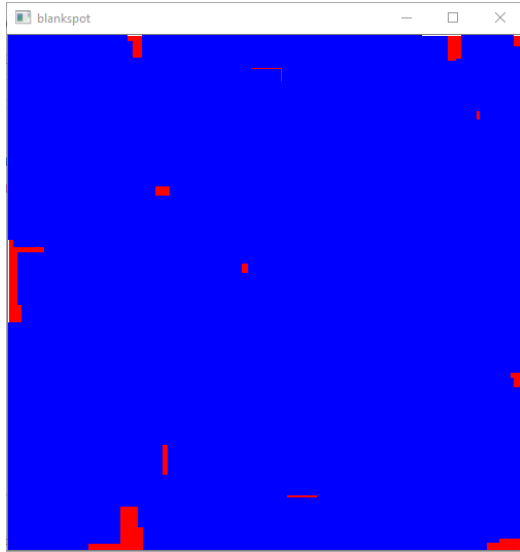
Gambar 3.15. Ilustrasi Metode *Plotting*

Kemudian kotak biru tersebut dilakukan proses *plotting* pada *canvas* yang mana titik pusat kotak tersebut disesuaikan dengan titik koordinat posisi UAV yang berhasil terekam, sehingga hasilnya dapat dilihat seperti pada Gambar 3.16 dibawah ini.

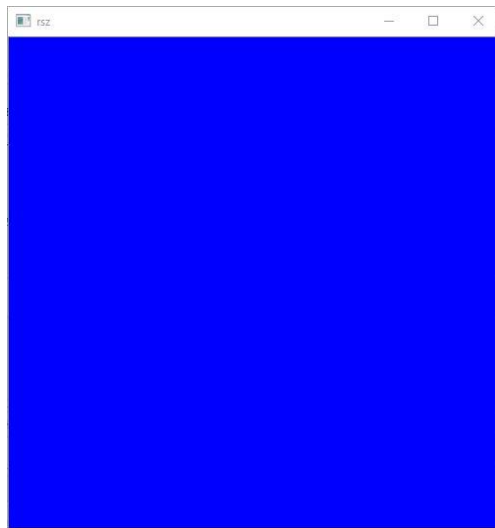


Gambar 3.16. Ilustrasi Hasil Metode *Plotting*

Tahap ini terus dijalankan hingga UAV menyelesaikan misi penerbangan. Sehingga pada akhirnya dihasilkan gambar *canvas* yang tertutupi kotak-kotak berwarna biru. Apabila gambar *canvas* keseluruhan menjadi berwarna biru seperti Gambar 3.18 maka diindikasikan UAV berhasil melakukan pemotretan seluruh area lahan yang diinginkan. Namun apabila pada *canvas* terdapat sisa gambar berwarna merah seperti pada Gambar 3.17 maka dapat diindikasikan wilayah/area tersebut tidak terfoto atau tidak terlewati sempurna oleh UAV saat misi penerbangan. Berikut program dan beberapa ilustrasi penjelasan:



Gambar 3.17. Ilustrasi *Canvas* dengan *Zona Blanspot*



Gambar 3.18. Ilustrasi *Canvas* Tanpa *Zona Blank spot*

Berikut program yang digunakan dalam menghasilkan keluaran seperti yang dijelaskan sebelumnya:

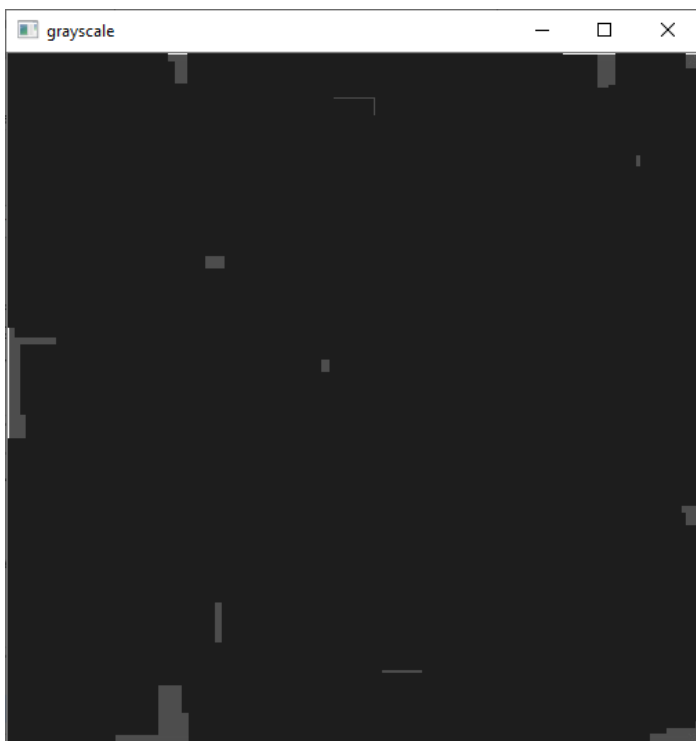
```
def plot(koorlat, koorlon):
    poinz=bumiTOcanvas(koorlat, koorlon)
    x=poinz[0]
    y=poinz[1]
    pF=167
    lF=222
    x1=int (x-(pF/2))
    y1=int (y-(lF/2))
    x2=int (x+(pF/2))
    y2=int (y+(lF/2))
    cv2.rectangle(rsz, (x1, y1), (x2, y2),
(255,0,0), -1)
```

- Tahap 4

Tahap selanjutnya adalah melakukan analisa terhadap hasil penggambaran pergerakan UAV sebelumnya. Pada tahap ini akan dilakukan pencarian titik koordinat *blank spot* yang terjadi pada *canvas*. Untuk melakukan analisa tersebut digunakan metode *blob analysis* dengan memanfaatkan *library cv2.findContours* yang terdapat pada OpenCV. Mekanismenya adalah sebagai berikut:

1. Mengubah gambar *canvas* menjadi gambar dengan tipe *GRAYSCALE* untuk memudahkan dalam proses *thresholding*. Berikut program yang digunakan dalam menghasilkan keluaran seperti yang dijelaskan:

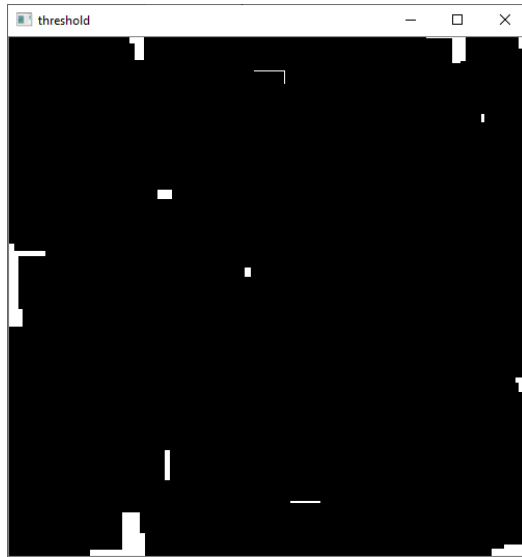
```
gray = cv2.cvtColor(rsz, cv2.COLOR_BGR2GRAY)
```



Gambar 3.19. Hasil *Grayscale Canvas*

2. Melakukan *thresholding* pada gambar *grayscale* sebelumnya. Tahap ini dilakukan untuk membuat seluruh gambar berwarna merah pada *canvas* akan berubah menjadi putih dan seluruh gambar berwarna biru pada *canvas* akan berubah menjadi hitam. Berikut program yang digunakan:

```
retval, fix = cv2.threshold(gray, 50, 255,  
cv2.THRESH_BINARY)
```

Gambar 3.20. Hasil *Thresholding Canvas*

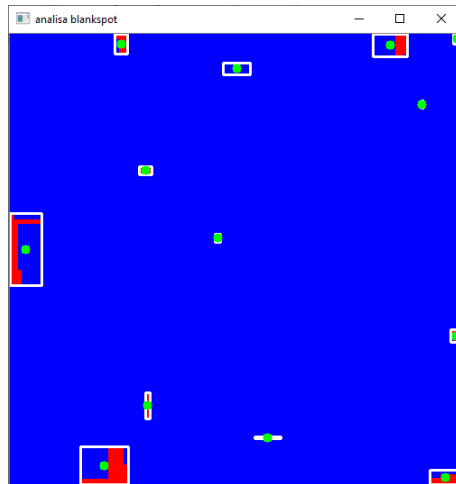
3. Kemudian dilakukan pengimplementasian metode *blob analysis* untuk mendapatkan jumlah total *blank spot* dan mendapatkan titik pusat masing-masing area *blank spot*. Dimana nantinya data titik pusat area *blank spot* pada *canvas* akan digunakan untuk tahap implementasi algoritma genetika selanjutnya. Berikut program yang digunakan:

```
try: hierarchy = hierarchy[0]
except: hierarchy = []
for contour, hier in zip(contours, hierarchy):
    (x,y,w,h) = cv2.boundingRect(contour)
    cv2.rectangle(rsz, (x,y), (x+w,y+h), (255,
    255, 255), 2)
    cv2.circle(rsz,(int(x+w/2),int(y+h/2)),
    5,(0, 255, 0) ,-1)
    print(int(x+w/2),"",int(y+h/2))
    cenx.append(int(x+w/2))
```

```

ceny.append(int (y+h/2))
cenxy.append((int (x+w/2),int(y+h/2)))
print("total blob=",len(contours))

```



Gambar 3.21. Implementasi *FindContour* Pada *Canvas*

Pada hasil analisa didapatkan koordinat *blank spot* seperti berikut:

Tabel 3.2. Data Koordinat *Zona Blank spot*

Koordinat(pixel,pixel)
(482, 492)
(104, 479)
(285, 448)
152 , 412
(494, 335)
(230, 226)
(17, 239)
(150, 151)
(456, 78)

(251, 38)
(495, 5)
(421, 12)
(123, 11)

3.2.2. Implementasi Algoritma Genetika

Pada tahap ini akan dilakukan pengimplementasian algoritma genetika dalam menyelesaikan permasalahan *Travel Sales Planner*. Dimana dalam melakukan suatu misi penerbangan, lama waktu UAV dapat mengudara sangat bergantung dari kapasitas baterai yang digunakan. Banyak atau jauhnya suatu rute penerbangan akan berdampak pada kebutuhan kapasitas baterai yang besar pula. Untuk itu penggunaan algoritma genetika dimaksudkan untuk melakukan penyusunan rute penerbangan baru untuk mengeksekusi seluruh titik koordinat *blank spot* yang telah dianalisa pada tahap sebelumnya. Susunan rute tersebut diharapkan merupakan susunan rute yang memiliki jarak tempuh paling minimum sehingga menghabiskan waktu dan daya baterai yang minim pula. Berikut mekanisme pengimplementasian algoritma genetika:

3.2.2.1. Pengkodean

Dari data analisa sebelumnya kita dapatkan kumpulan titik koordinat *canvas* yang diindikasikan merupakan zona *blank spot*. Data koordinat tersebut dalam tahapan ini disebut sebagai sebuah gen. Sebuah rute perjalanan diibaratkan sebagai suatu solusi yang ingin dicapai. Kumpulan gen yang merupakan urutan susunan data koordinat diartikan sebagai suatu kromosom atau individu. Satu kromosom atau individu mewakili satu vektor solusi. Kemudian kumpulan dari banyaknya solusi disebut dengan populasi.

Pada tahap ini data koordinat yang didapat sebelumnya akan mengalami proses pengkodean. Pengkodean dilakukan untuk mewakili suatu nilai data koordinat dengan menggunakan suatu angka desimal. Pengkodean dilakukan berdasarkan urutan data yang diterima dari hasil analisa sebelumnya. Berikut program yang digunakan:

```
for i in range(kota):
    order.append((i))
```

```
pot=order[1: ]
```

Tabel 3.3. Tabel Hasil Pengkodean Koordinat Zona *Blank spot*

Koordinat(pixel,pixel)	Pengkodean
(482, 492)	0
(104, 479)	1
(285, 448)	2
152 , 412	3
(494, 335)	4
(230, 226)	5
(17, 239)	6
(150, 151)	7
(456, 78)	8
(251, 38)	9
(495, 5)	10
(421, 12)	11
(123, 11)	12

3.2.2.2. *Membangkitkan Populasi*

Dalam tahap ini dilakukan pembangkitan suatu populasi dengan metode acak. Ukuran populasi atau banyaknya individu yang ingin dibangkitkan disesuaikan dengan kebutuhan. Dengan dibangkitkannya populasi ini, maka akan tersedia banyaknya vektor solusi. Berikut program yang digunakan:

```
popu=0  
popu=[]  
for i in range(kota):  
    order.append((i))
```

```

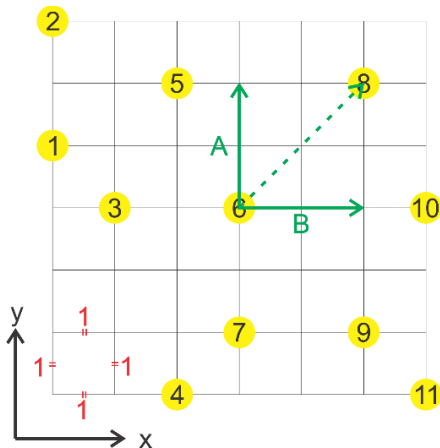
    pot=order[1:]
for i in range(popsiz):
    shuffle(pot)
    t=pot.copy()
    t.insert(0,0)
    popu.append(t)

```

Tabel 3.4. Tabel Hasil Pembangkitan Populasi

Populasi	
[4, 7, 2, 11, 1, 0, 9, 12, 10, 5, 8, 6, 3]	<< Solusi 1
[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	<< Solusi 2
[1, 6, 11, 0, 8, 4, 3, 9, 10, 5, 12, 2, 7]	<< Solusi 3
[1, 12, 0, 7, 3, 9, 10, 11, 2, 4, 5, 8, 6]	<< Solusi 4
[12, 9, 2, 6, 8, 11, 3, 0, 5, 1, 4, 10, 7]	<< Solusi 5

3.2.2.3. Evaluasi

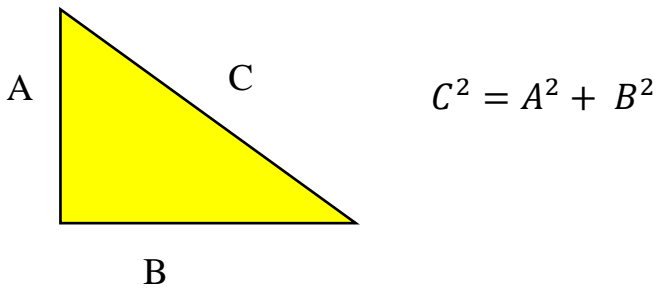


Gambar 3.22. Ilustrasi Metode Equidistance

Kemudian tahap selanjutnya solusi-solusi yang dibangkitkan dalam populasi akan dilakukan evaluasi. Evaluasi bertujuan untuk

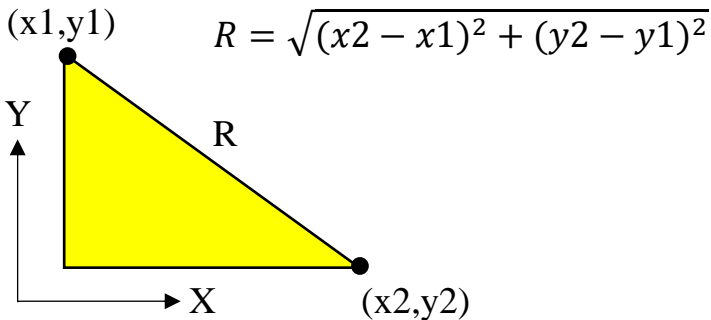
mengukur tingkat kesesuaian suatu vektor solusi dengan solusi yang dicari atau diinginkan. Evaluasi dilakukan dengan menghitung jarak masing-masing antar koordinat dan memasukkannya kedalam fungsi fitness.

Dalam melakukan perhitungan jarak antar koordinat dilakukan dengan metode perhitungan *equidistance* atau yang umum kita kenal dengan istilah metode *pythagoras*. Ilustrasi dalam perhitungan jarak dapat dilihat pada Gambar 3.22. Dimana rumus *pythagoras* pada umumnya adalah sebagai berikut:



Gambar 3.23. Metode Pythagoras

Dengan rumus dasar yang sama yaitu *pythagoras*, terdapat suatu rumus lain yang dapat digunakan dalam melakukan perhitungan jarak antar koordinat, seperti dijelaskan dibawah ini:



Gambar 3.24. Metode Perhitungan Jarak Dua Titik Koordinat

Kemudian rumus tersebut diimplementasikan dalam bentuk program yang mana seperti dibawah berikut:

```
def ukurjarak(pop):
    tanda=pop
    jartot=0
    for i in range(len(tanda)-1):
        jarak=math.sqrt(((ye[tanda[i+1]]-
            ye[tanda[i]])**2)+((ex[tanda[i+1]]-
            ex[tanda[i]])**2))
        jartot=jartot+jarak
    return jartot
```

Tabel 3.5. Tabel Hasil Evaluasi

Populasi	Jarak
[4, 7, 2, 11, 1, 0, 9, 12, 10, 5, 8, 6, 3]	4430.22
[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	3982.09
[1, 6, 11, 0, 8, 4, 3, 9, 10, 5, 12, 2, 7]	4238.33
[1, 12, 0, 7, 3, 9, 10, 11, 2, 4, 5, 8, 6]	4230.62
[12, 9, 2, 6, 8, 11, 3, 0, 5, 1, 4, 10, 7]	4015.39

Langkah kedua adalah dengan melakukan perhitungan nilai fitness dari tiap masing-masing solusi yang dibangkitkan. Fungsi fitness yang digunakan pada tugas akhir ini adalah sebagai berikut:

$$f(x) = \frac{1}{Jarak^8+1} \tag{3.12}$$

Tabel 3.6. Hasil Perhitungan Nilai Fitness

Populasi	Jarak	Fitness
[4, 7, 2, 11, 1, 0, 9, 12, 10, 5, 8, 6, 3]	4430.22	6.73e-30

[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	3982.09	1.58e-29
[1, 6, 11, 0, 8, 4, 3, 9, 10, 5, 12, 2, 7]	4238.33	9.60e-30
[1, 12, 0, 7, 3, 9, 10, 11, 2, 4, 5, 8, 6]	4230.62	9.74e-30
[12, 9, 2, 6, 8, 11, 3, 0, 5, 1, 4, 10, 7]	4015.39	1.47e-29

Berikut program yang digunakan dalam menghasilkan keluaran seperti yang dijelaskan sebelumnya.

```
def fitness(pop2):
    global bestr
    global bestpop
    fitness=0
    save=0
    fitness=[]
    for i in range(len(pop2)):
        d=ukurjarak(pop2[i])
        if d<bestr:
            bestr=d
            bestpop=pop2[i]
            save=i
            print(bestr)
            print(bestpop)
            f=1/(pow(d,8)+1)
            fitness.append((f))
    return fitness,save
```

Pada tahap diatas akan dihasilkan nilai fitness yang sebanding dengan nilai jarak yang dihasilkan setiap solusi atau dapat dikatakan nilai jarak yang besar akan menghasilkan nilai fitness yang besar pula. Namun hal ini berkebalikan dengan fungsi tujuan yang diinginkan pada penyelesaian masalah yang ada. Dimana hasil fungsi tujuan yang dicari

merupakan hasil yang mempunyai nilai paling rendah. Oleh karenanya untuk melakukan penyesuaian terhadap fungsi tujuan yang diinginkan dilakukan tahap normalisasi nilai fitness. Pada tahap ini vektor solusi yang menghasilkan jarak terpendek akan mempunyai nilai fitness yang paling besar begitu pula sebaliknya.

Tabel 3.7. Tabel Hasil Normalisasi

Populasi	Jarak	Fitness	Normalisasi
[4, 7, 2, 11, 1, 0, 9, 12, 10, 5, 8, 6, 3]	4430.22	6.73e-30	0.119
[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	3982.09	1.58e-29	0.279
[1, 6, 11, 0, 8, 4, 3, 9, 10, 5, 12, 2, 7]	4238.33	9.60e-30	0.169
[1, 12, 0, 7, 3, 9, 10, 11, 2, 4, 5, 8, 6]	4230.62	9.74e-30	0.172
[12, 9, 2, 6, 8, 11, 3, 0, 5, 1, 4, 10, 7]	4015.39	1.47e-29	0.26

Berikut program yang digunakan dalam menghasilkan keluaran seperti yang dijelaskan sebelumnya.

```
def normfit(fitfit):
    #normalize fitness
    sum=0
    fitfix=0
    fitfix=[]
    for i in range(len(fitfit)):
        sum=sum+fitfit[i]
    for i in range(len(fitfit)):
        f=fitfit[i]/sum
        fitfix.append((f))
    return fitfix
```

3.2.2.4. *Elitisme*

Dari tahapan diatas akan didapatkan satu buah vektor solusi yang dianggap terbaik atau dikatakan vektor solusi yang mempunyai nilai fitness tertinggi dan atau menghasilkan jarak total terendah diantara vektor-vektor solusi lainnya didalam populasi tersebut. Kemudian selanjutnya merupakan tahapan elitisme. Tahap ini merupakan suatu proses yang bertujuan untuk mempertahankan individu-individu terbaik disuatu generasi kedalam generasi selanjutnya. Sehingga individu yang dianggap terbaik pada generasi sebelumnya akan tetap eksis atau muncul di populasi pada generasi selanjutnya. Tahap elitisme dilakukan dengan melalui penyalinan individu terbaik sebanyak jumlah yang telah ditentukan. Individu terbaik akan menggantikan posisi individu-individu yang memiliki nilai fitness paling rendah dalam populasi tersebut.

Tabel 3.8. Tabel Populasi Sebelum Tahap Elitisme

Populasi	Normalisasi
[4, 7, 2, 11, 1, 0, 9, 12, 10, 5, 8, 6, 3]	0.119
[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	0.279
[1, 6, 11, 0, 8, 4, 3, 9, 10, 5, 12, 2, 7]	0.169
[1, 12, 0, 7, 3, 9, 10, 11, 2, 4, 5, 8, 6]	0.172
[12, 9, 2, 6, 8, 11, 3, 0, 5, 1, 4, 10, 7]	0.26

<< Solusi
Terburuk
<< Solusi
Terbaik

Tabel 3.9. Tabel Populasi Setelah Tahap Elitisme

Populasi	Normalisasi
[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	0.279
[11, 6, 10, 8, 12, 0, 5, 3, 4, 9, 7, 2, 1]	0.279
[1, 6, 11, 0, 8, 4, 3, 9, 10, 5, 12, 2, 7]	0.169

<< Penyalinan
<< Solusi
Terbaik

[1, 12, 0, 7, 3, 9, 10, 11, 2, 4, 5, 8, 6]	0.172
[12, 9, 2, 6, 8, 11, 3, 0, 5, 1, 4, 10, 7]	0.26

Berikut program yang digunakan dalam menghasilkan keluaran seperti yang dijelaskan sebelumnya.

```
def elitism(pop, fit, many, fitbes):
    fitsort=fit.copy()
    fitsort.sort()
    kopifit=[]
    for t in range(many):
        for i in range(len(fit)):
            if fit[i]==fitsort[t]:
                pop[i]=bestpop
                fit[i]=math.inf
                kopifit.append((i))
    for i in range(len(kopifit)):
        fit[kopifit[i]]=fitbes
    return pop
```

3.2.2.5. *Kawin Silang*

Tahapan kawin silang adalah proses memunculkannya individu-individu baru dari hasil perkawinan silang antara dua buah individu. Tahap ini bertujuan untuk memunculkan vektor-vektor solusi baru pada proses pembaruan populasi. Proses kawin silang diawali dengan pemilihan 2 buah individu yang akan dijadikan sebagai induk. Pemilihan dilakukan berdasarkan kalkulasi yang mempertimbangkan nilai fitness masing-masing individu. Individu yang mempunyai nilai fitness besar akan jauh lebih berpeluang untuk terpilih sebagai induk dibandingkan individu yang mempunyai nilai fitness yang rendah. Begitupula sebaliknya. Berikut program yang digunakan:

```

def pickone(list,prob):
    index=0
    r=random.random()
    while(r>0):
        if index>len(list)-1:
            index=0
            r=r-prob[index]
            index=index+1
        index=index-1
    #print(index)
    return list[index]

```

Proses kawin silang dilakukan dengan mengambil potongan gen pada salah satu induk dan mengambil potongan gen yang berbeda pada induk lainnya. Hasil pengambilan potongan gen dari kedua induk kemudian digabungkan untuk menjadi satu buah individu atau vektor solusi baru. Tahap ini akan dilakukan berulang sebanyak jumlah individu dalam populasi. Berikut program yang digunakan:

```

def crossover(orderAkk,orderBee):
    orderAk=orderAkk[1:]
    orderBe=orderBee[1:]
    test=0
    neworder=0
    neworder=[]
    start=random.randint(0,len(orderAk)-2)
    end=random.randint(start+1,len(orderAk)-1)
    for i in range(start,end+1):
        neworder.append((orderAk[i]))
    test=np.array(neworder+orderBe)
    _, idx =np.unique(test,return_index=True)
    test=test[np.sort(idx)]
    tested=test.tolist()
    return tested

```

3.2.2.6. *Mutasi*

Lalu tahap selanjutnya merupakan proses mutasi. Proses ini bertujuan untuk menghadirkan kemungkinan adanya kemunculan individu-individu baru yang bukan berasal dari hasil kawin silang. Proses mutasi akan menghasilkan individu baru dengan melakukan perubahan urutan atau penggantian elemen dari vektor solusi. Dalam proses mutasi terdapat variabel probabilitas mutasi. Nilai ini akan berpengaruh pada seberapa banyak sebuah individu mengalami mutasi. Semakin besar nilai probabilitas mutasi maka akan semakin banyak proses mutasi yang dilakukan pada individu tersebut begitu pula sebaliknya. Berikut program yang digunakan:

```
def mutate(order,rate):
    #print(order)
    baru=order
    for i in range(len(order)):
        x=random.random()
        if x<rate:
            temp1=random.randint(1,len(order)-1)
            temp2=(temp1+1)%len(order)
            baru=swap(order,temp1,temp2)
    baru.insert(0,0)
    return baru

def swap(arr,i,j):
    temp=arr[i]
    arr[i]=arr[j]
    arr[j]=temp
    nu=arr
    return nu
```

3.2.2.7. *Repetisi*

Pada tahap mutasi dihasilkan individu-individu baru yang membentuk suatu populasi baru. Populasi baru tersebut akan menggantikan posisi populasi yang lama. Tahap selanjutnya populasi

baru tersebut akan mengalami proses pengulangan dimana populasi tersebut akan kembali menjadi masukan data pada tahap evaluasi. Proses pengulangan tersebut akan terus terjadi sampai fungsi tujuan menghasilkan nilai konvergensi yang sesuai dengan yang diharapkan. Namun, pada Tugas Akhir ini proses pengulangan tidak berdasarkan tingkat kesesuaian nilai konvergensi yang didapat tetapi berdasarkan jumlah iterasi yang telah dicapai. Nilai iterasi yang diinginkan telah ditetapkan sebelumnya. Oleh karena itu proses pengulangan tidak menghiraukan nilai kesesuaian konvergensi yang telah didapat tetapi dibatasi dari jumlah iterasi yang telah dicapai. Berapapun tingkat kesesuaian konvergensi yang didapat, proses pengulangan akan berakhir ketika nilai iterasi telah mencapai jumlah yang telah ditentukan. Hasil keluaran yang digunakan nantinya merupakan hasil vektor solusi terbaik yang bertahan sampai proses pengulangan berakhir. Vektor solusi terbaik yang didapatkan merupakan jawaban dari susunan koordinat yang memiliki jarak tempuh terpendek yang dihasilkan selama proses optimasi.

3.2.3. Konversi koordinat

Hasil yang didapatkan pada akhir proses implementasi algoritma genetika merupakan suatu susunan koordinat jalur terbang yang menghasilkan jarak tempuh paling pendek. Nilai koordinat yang didapat merupakan nilai koordinat kartesian. Oleh karena itu, data koordinat yang didapat perlu dilakukan konversi menjadi data koordinat bumi agar data tersebut dapat dieksekusi atau diterima oleh *flight controller* pada UAV. Berikut program yang digunakan:

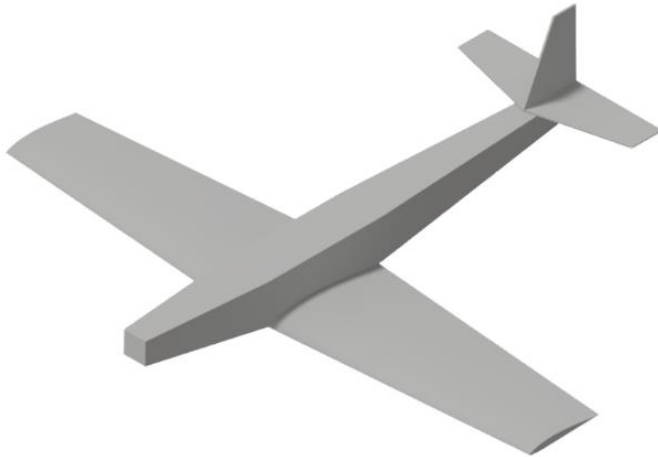
```
def canvastObumi(pointx,pointy):  
    long=(pointx*(deltalong)/500)+long1  
    lat=(-(pointy*deltalat)/500)+lat1  
    return lat,long
```

3.3. Perancangan Wahana

Pada tahap ini dilakukan perancangan wahana yang akan diterapkan pada pengembangan sistem autopilot pada Tugas Akhir ini, Perancangan dibagi dalam dua bagian yaitu perancangan mekanik dan perancangan elektronik.

3.3.1. Perancangan Mekanik

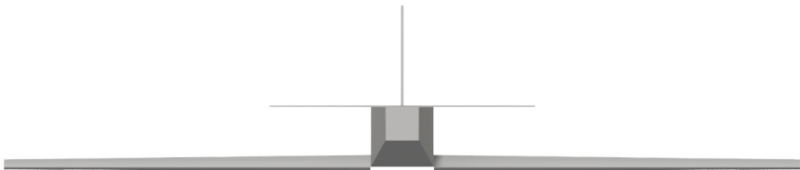
Perancangan mekanik merupakan perancangan yang dilakukan untuk membuat wahana UAV itu sendiri. Dalam melakukan perancangan ini terdapat beberapa faktor yang perlu diperhatikan. Pertama, UAV mempunyai prinsip yang sama seperti halnya pesawat terbang oleh karena itu pemodelan wahana harus memperhatikan sisi aerodinamis dari UAV. Kedua, jenis UAV yang dibuat harus disesuaikan dengan kebutuhan pada misi penerbangan yang akan dijalani. Dalam misi penerbangan pada Tugas Akhir ini, UAV yang dibutuhkan adalah UAV yang mempunyai karakteristik yang dapat terbang dengan kecepatan cukup rendah, Dengan kemampuan terbang dengan kecepatan rendah membuat UAV meghabiskan daya baterai yang tidak terlalu banyak. Kemudian UAV dapat melakukan manuver cukup tajam. Dengan kemampuan manuver yang cukup tajam membuat kemungkinan terjadinya *miss waypoint* menjadi semakin berkurang. Lalu UAV memiliki *payload* yang cukup besar untuk memuat berbagai komponen elektronik dan keperluan pemetaan lainnya. Oleh karena itu penulis memilih tipe UAV berjenis *trainer* dengan jenis sayap *low wing* atau dapat dikatakan posisi sayap bawah. Dengan pilihan tersebut dirasa dapat menjawab seluruh kebutuhan yang ada pada misi penerbangan yang akan dijalankan.



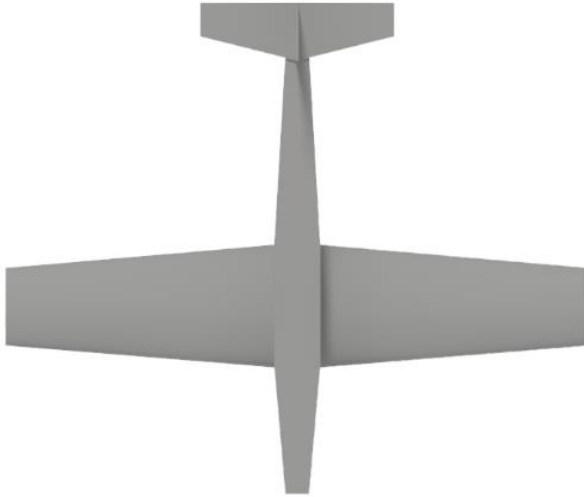
Gambar 3.25. *Design 3D UAV*



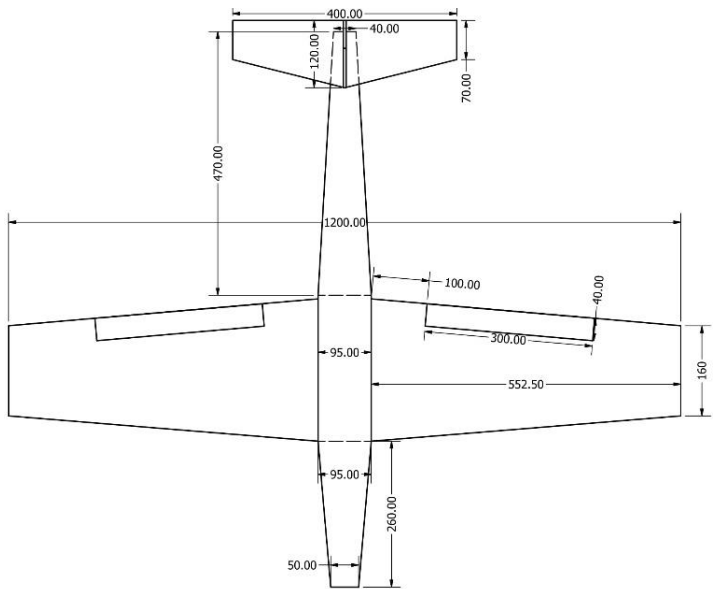
Gambar 3.26. *Design 3D UAV Tampak Samping*



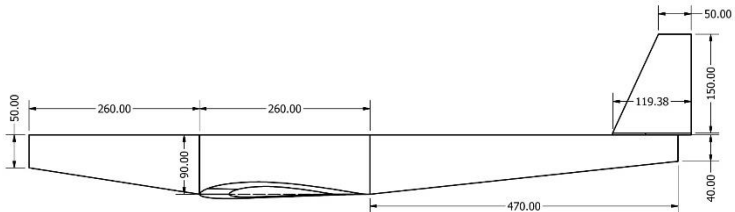
Gambar 3.27. *Design 3D UAV Tampak Depan*



Gambar 3.28. *Design 3D UAV Tampak Atas*

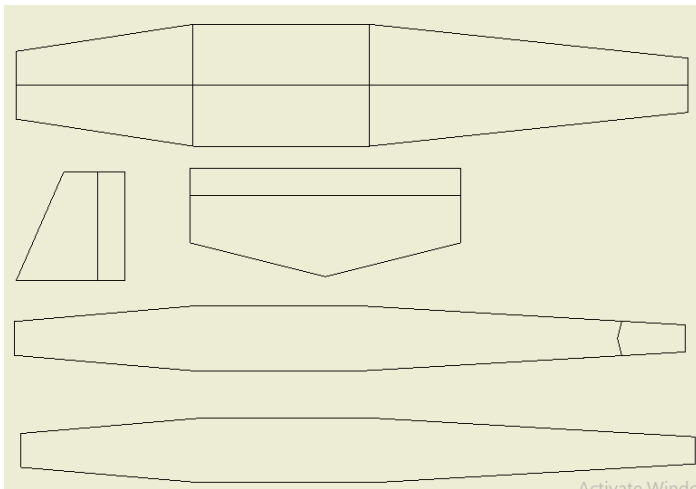


Gambar 3.29. *Design 2D UAV Tampak Atas*



Gambar 3.30. *Design 2D UAV Tampak Samping*

Dalam melakukan proses produksi, berat UAV menjadi satu hal yang perlu diperhatikan. Beban yang terlalu berat pada UAV akan membuat kerja komponen motor pada UAV menjadi berlebih. Dengan demikian, baterai yang digunakan akan cepat habis ataupun diperlukannya kapasitas baterai yang besar. Dengan memperhatikan hal tersebut, penulis melakukan beberapa pertimbangan dalam melakukan pemilihan material yang akan digunakan dalam proses produksi UAV tersebut.



Gambar 3.31. *Design Cetak UAV 2D*

Pada bagian *body* UAV, penulis memilih material *polyfoam* sebagai solusi tersebut. Dimana material tersebut mempunyai karakteristik bahan yang ringan dan lentur. Namun terdapat kekurangan dimana material tersebut mudah patah. Untuk mensiasati permasalahan tersebut penulis melakukan pelapisan menggunakan perekat isolasi. Dengan melakukan pelapisan tersebut material *polyfoam* mempunyai tingkat kekuatan yang lebih tinggi dibandingkan tanpa dilakukan pelapisan. Untuk melakukan proses produksi tersebut, penulis melakukan konversi *Design* 3D menjadi 2D untuk setiap bagian sisi UAV seperti pada Gambar 3.31. Kemudian dilakukan pemotongan pada material *polyfoam*.

Pada bagian sayap penulis memilih jenis material foam sebagai solusi. Material foam memiliki karakteristik yang sama dengan material *polyfoam*. Karakteristiknya yang mudah patah dan terkelupas membuat pelapisan menggunakan perekat isolasi menjadi pilihan. Perbedaan dengan *polyfoam*, material foam mempunyai ketebalan yang lebih tinggi. Dengan kelebihan tersebut membuat produksi sayap dapat dilakukan secara utuh. Produksi sayap dilakukan dengan menggunakan kawat panas untuk membentuk *airfoil* UAV pada foam yang ada.

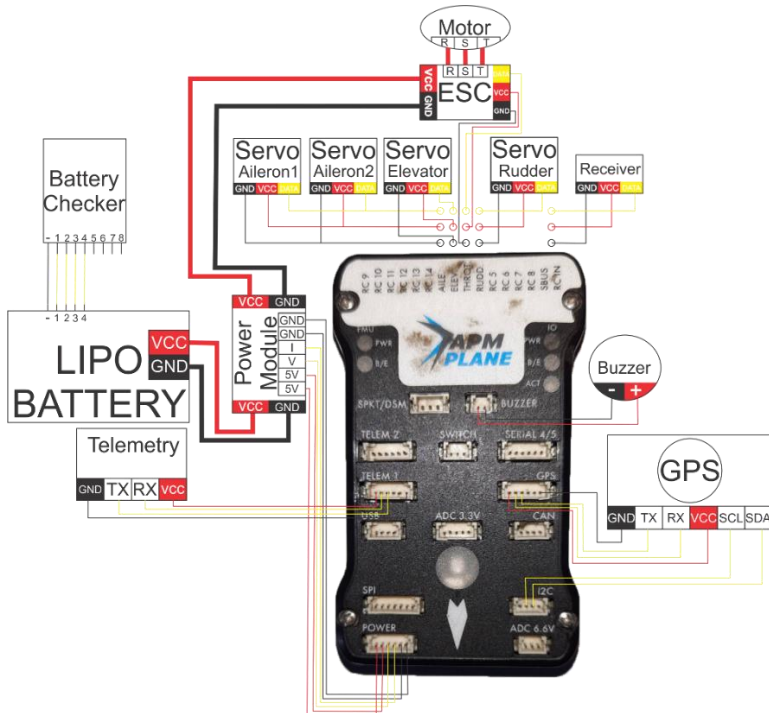


Gambar 3.32. Tahap Pembuatan Sayap UAV

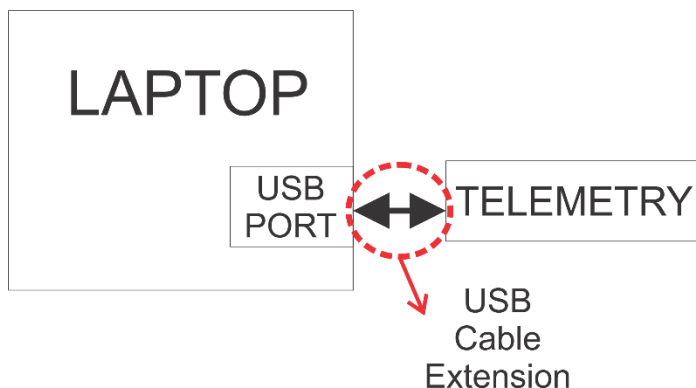
3.3.2. Perancangan Elektronik

Pada bagian ini akan membahas mengenai komponen-komponen apa saja yang menunjang dalam melakukan realisasi sistem pada tugas akhir ini. Komponen elektronik yang akan digunakan terbagi dalam dua bagian, yaitu elektronik pada UAV dan pada *Ground Control System*(GCS). Bagian elektronik pada UAV yang pertama terdiri dari *flight controller*. *Flight controller* memuat banyak fitur sensor yang mendukung dalam melakukan otomasi pada UAV. Dengan demikian UAV dapat terbang secara otomatis mengikuti perintah yang diberikan oleh GCS atau melakukan misi yang telah *diupload* pada *flight controller*. Kedua, terdapat komponen baterai lipo yang digunakan sebagai suplai daya utama pada UAV. Digunakan baterai lipo karena baterai tersebut mempunyai nilai C yang tinggi sehingga dapat memberikan suplai arus yang cukup pada motor tanpa mengakibatkan penurunan tegangan yang berlebih terhadap baterai. Ketiga, terdapat komponen motor yang digunakan sebagai penggerak utama pada UAV. Motor akan memberikan daya dorong atau tarik pada UAV. Keempat, terdapat komponen servo yang digunakan sebagai penggerak utama dalam mengontrol fungsi *aileron*, *elevator* dan *rudder* pada UAV. Dengan komponen tersebut membuat UAV dapat bergerak secara *pitch*, *roll* dan *yaw*. Kelima, terdapat komponen GPS yang berguna memberikan data GPS maupun kompas yang dibutuhkan oleh *flight controller*. Keenam, terdapat komponen *receiver* yang berguna untuk menerima data dari *transmitter* yaitu *remote control*. Dengan demikian UAV dapat dikontrol secara mode manual oleh pilot menggunakan sebuah *remote control*, pergantian mode penerbangan dan lainnya. Ketujuh, terdapat komponen *power module* yang berguna untuk memberikan suplai daya pada *flight controller* sekaligus berperan sebagai sensor tegangan dan arus. Peran *power module* sebagai sensor tersebut memungkinkan dilakukannya pemantauan tegangan dan arus baterai yang masih tersedia maupun yang digunakan selama UAV melakukan misi penerbangan. Kedelapan, terdapat komponen telemetry yang digunakan sebagai alat perantara komunikasi antara UAV dan GCS. Dengan demikian GCS dapat melakukan pemantauan pergerakan UAV maupun data-data lain yang dibutuhkan pada UAV. Kesembilan, terdapat komponen *buzzer* yang berguna memberikan informasi berbagai status *flight controller* yang

dikemas dalam bentuk suara. Kesepuluh, terdapat komponen *lipo battery checker* yang berguna untuk mengetahui nilai tegangan pada baterai. Komponen ini juga dapat berguna untuk memberikan peringatan berupa suara apabila tegangan setiap sel pada baterai berada dibawah nilai tegangan yang kita inginkan. Dengan demikian komponen tersebut dapat mencegah terjadinya *drop* tegangan pada baterai lipo sehingga dapat memperpanjang usia penggunaan baterai. Dan yang terakhir terdapat komponen kamera yang digunakan untuk melakukan pemotretan selama misi penerbangan. Komponen kamera yang digunakan tidak memiliki keterkaitan terhadap sistem elektronik pada UAV.



Gambar 3.33. Wiring Komponen Elektronik UAV



Gambar 3.34. *Wiring* Komponen Elektronik GCS

Pada bagian GCS, pertama terdapat *hardware* berupa laptop sebagai perangkat utama dalam melakukan pemrosesan dalam optimasi yang dikerjakan pada tugas akhir ini dan sebagai *interface* dalam melakukan *monitoring* pada UAV. Kedua, terdapat komponen telemetry yang tersambung pada laptop. Komponen ini akan tersambung secara *wireless* dengan komponen telemetry yang terdapat pada UAV. Dengan demikian dapat terjadinya komunikasi dua arah antara GCS dan UAV. Ketiga, terdapat komponen transmitter berupa radio *remote control* yang terhubung secara *wireless* dengan komponen *receiver* pada UAV. Dengan menggunakan *remote control*, pilot dapat melakukan pergerakan pada UAV. Terlihat pada Gambar 3.33 dan 3.34 keseluruhan *wiring* komponen elektronik yang terpasang pada UAV dan GCS.

Tabel 3.10. *Wiring* Komponen Elektronik

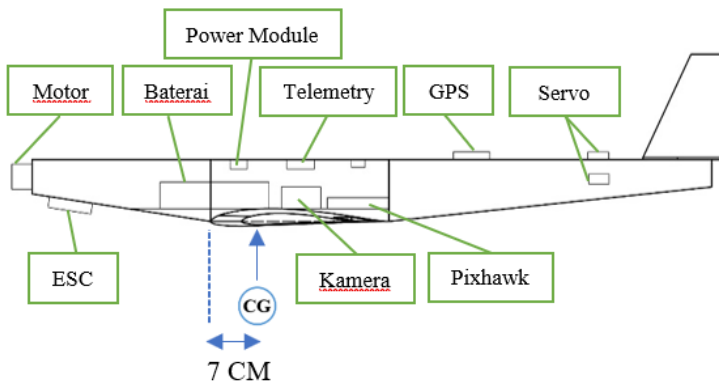
Komponen	Bagian	Koneksi
Fixed Wing		
ESC	VCC	VCC Power Module
	GND	GND Power Module
	BEC	Pin 3 Main Out FC
	Fasa 1	Fasa Motor
	Fasa 2	Fasa Motor
	Fasa 3	Fasa Motor

GPS	VCC	Port GPS FC
	GND	
	TX	
	RX	
	SCL	Port I2C FC
	SDA	
Battery	VCC	VCC Power Module
	GND	GND Power Module
	Balancer	Battery Checker
Telemetry	VCC	Port Telem 1 FC
	GND	
	TX	
	RX	
Receiver	VCC	Pin RC IN FC
	GND	
	DATA	
Servo Aileron	VCC	Pin 1 Main Out FC
	GND	
	DATA	
Servo Elevator	VCC	Pin 2 Main Out FC
	GND	
	DATA	
Servo Rudder	VCC	Pin 4 Main Out FC
	GND	
	DATA	
Buzzer	VCC	Port Buzzer FC
	GND	
Ground Control System		
Telemetry	VCC	Port USB Laptop
	GND	
	TX	
	RX	

Tabel 3.11. Spesifikasi Komponen Elektronik

Komponen	Spesifikasi
Fixed wing	
Flight Controller	Pixhawk PX4
ESC	Hobbywing 60A
GPS	UBLOX M8N GPS module with compass
Battery	ONBO 4200 mAh 4s 25C
Telemetry	500mW 433MHz 3DR Radio Telemetry
Receiver	Frsky fx400r
Servo	SG90 Tower Pro Micro 9g mini
Motor	Motodoro 1350 kv, Propeller 9x6
Buzzer	Pixhawk Buzzer
Battery Checker	-
Camera	Xiaomi YI Action Cam
Ground Control System	
Laptop	HP Pavilion g4 Notebook PC
Telemetry	500mW 433MHz 3DR Radio Telemetry
Transmitter	Frsky Taranis QX7

3.3.3. Perakitan



Gambar 3.35. Peletakan Komponen Pada UAV

Pada tahap ini akan dilakukan pemasangan komponen-komponen elektronik pada wahana UAV. Dalam hal pemasangan elektronik tersebut, terdapat hal yang perlu diperhatikan yaitu titik *central of gravity*. Titik *central of gravity* atau bisa dikatakan sebagai titik keseimbangan merupakan salah satu kunci kesempurnaan UAV untuk dapat mengudara. Peletakan komponen elektronik pada UAV haruslah menghasilkan nilai torsi=0 pada titik *central of gravity* tersebut. Adanya perbedaan tata letak titik *central of gravity* pada UAV akan menghasilkan karakteristik terbang UAV yang berbeda pula. Peletakan *flight controller* perlu diperhatikan untuk tidak berada dekat dengan komponen-komponen yang menghasilkan medan magnet seperti motor dan esc. Hal ini dikarenakan dapat mengganggu kerja dari sensor-sensor yang berada pada *flight controller*.

3.4. Komunikasi GCS dan UAV

Pada tugas akhir ini dilakukan optimasi jalur penerbangan pada sebuah UAV dalam melakukan misi pemotretan lahan dari udara. Dalam melakukan optimasi penulis melakukan analisa terhadap pergerakan UAV selama misi penerbangan untuk mengindikasikan keberadaan suatu zona *blank spot* yang dihasilkan selama misi penerbangan. Untuk melakukan analisa pergerakan UAV tersebut penulis mengolah dan mencatat setiap pergerakan UAV menggunakan data GPS yang diambil pada UAV selama misi penerbangan. Data GPS didapatkan karena adanya komunikasi *wireless* antara GCS dan UAV menggunakan komponen telemetry. Untuk melakukan komunikasi tersebut penulis menggunakan protokol MAVLINK yang telah dikemas dalam suatu *library* bernama dronekit. Pada *library* tersebut terdapat fungsi-fungsi yang dapat digunakan untuk mengirim atau meminta suatu data yang dibutuhkan dari GCS terhadap UAV. Dengan demikian UAV dapat dikontrol menggunakan pemrograman yang telah dibuat penulis tanpa menggunakan *software* Mission Planner. Berikut beberapa fungsi pada *library drone-kit* yang digunakan pada tugas akhir ini dalam menunjang optimasi.

- Langkah awal dalam memulai komunikasi antara GCS dan UAV adalah dengan melakukan koneksi antara GCS dan UAV. Berikut program yang digunakan untuk melakukan koneksi tersebut. Untuk nomor COM disesuaikan dengan COM pada telemetry yang tersambung pada laptop. Berikut program yang digunakan:

```

from dronekit import connect, VehicleMode,
LocationGlobalRelative, Command, Battery,
LocationGlobal, Attitude

connection_string = "COM6"

print("Connection to the vehicle on %s" %
connection_string)
vehicle = connect(connection_string, baud=57600,
wait_ready=False)
print("vehicle is connected")

```

- Dalam melakukan misi *Auto Take-Off* menggunakan *library* drone kit pada UAV, terdapat perbedaan mekanisme yang harus dilakukan dalam pengaplikasian terhadap UAV jenis *rotor* dan *fixed wing*. Dimana pada UAV jenis *rotor* dapat menggunakan fungsi *takeoff* yang telah disediakan pada *library* drone kit dan dapat terbang dalam mode *Guided*, sedangkan pada UAV jenis *fixed wing* tidak demikian. Terdapat mekanisme yang harus disusun dalam melakukan misi *auto take-off* tersebut, yaitu menambahkan parameter *waypoint auto take-off* pada rencana penerbangan dan terbang dalam keadaan mode *Auto*. Berikut pemrograman yang digunakan dalam membuat misi *auto take-off* pada UAV yang digunakan pada tugas akhir ini. Berikut program yang digunakan:

```

def cek_takeoff():
    vehicle.commands.download()

```

```

vehicle.commands.wait_ready()
tmp_mission = list(vehicle.commands)
is_mission = False
if len(tmp_mission) >= 1:
    is_mission = True
if is_mission and tmp_mission[0].command ==
mavutil.mavlink.MAV_CMD_NAV_TAKEOFF:
    print ("Takeoff already in the mission")
else:
    print("Takeoff not in the mission: adding")
    cmds=vehicle.commands
    vehicle.commands.clear()
    vehicle.flush()
    vehicle.commands.download()
    vehicle.commands.wait_ready()
    takeoff_item = Command( 0, 0, 0, 3,
mavutil.mavlink.MAV_CMD_NAV_TAKEOFF, 0, 0,
takeoff_pitch, 0, 0, takeoff_heading, 0, 0,
takeoff_altitude)
    vehicle.commands.add(takeoff_item)
    for item in tmp_mission:
        vehicle.commands.add(item)
    vehicle.flush()
    print(">>>>>Done")

```

- Berikut fungsi-fungsi *library drone-kit* yang digunakan pada tugas akhir ini beserta penjelasannya.

Tabel 3.12. Protokol Komunikasi

Fungsi	Penjelasan
vehicle.armed = True	Melakukan <i>arming</i> pada UAV
vehicle.armed = False	Melakukan <i>disarming</i> pada UAV

<code>vehicle.mode = VehicleMode("AUTO")</code>	Membuat UAV berada dalam mode Auto
<code>vehicle.mode = VehicleMode("GUIDED")</code>	Membuat UAV berada dalam mode Guided
<code>vehicle.mode = VehicleMode("LOITER")</code>	Membuat UAV berada dalam mode Loiter
<code>vehicle.mode = VehicleMode("RTL")</code>	Membuat UAV berada dalam mode RTL
<code>print(" Mode: %s" % vehicle.mode.name)</code>	Mengetahui <i>current mode</i> pada UAV
<code>wp=LocationGlobalRelative(lat,long,alt)</code>	Mendapatkan data GPS pada UAV
<code>vehicle.simple_goto(wp)</code>	Memberikan perintah UAV menuju <i>waypoint</i> tertentu

BAB 4

PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan mengenai hasil pengujian yang dilakukan terhadap rancangan sistem yang telah dibuat pada bab sebelumnya. Selain itu pada bab ini akan memperlihatkan hasil produk UAV yang berhasil dirancang pada Tugas Akhir ini.

4.1. Realisasi Alat



Gambar 4.1. UAV Tampak Atas



Gambar 4.2. UAV Tampak Samping



Gambar 4.3. UAV Tampak Belakang



Gambar 4.4. *Ground Control Systems*

4.2. Pengujian Karakteristik UAV

Pada tahap ini dilakukan pengujian tahap pertama pada UAV setelah dilakukan perakitan. Pengujian dilakukan dengan bertujuan untuk mengetahui karakteristik dari pada UAV dan nilai-nilai parameter yang dibutuhkan dalam melakukan pengaturan autopilot pada *flight controller*.

Hasil analisa pada pengujian ini didapatkan beberapa hal yaitu UAV dapat mengudara dalam mode *stabilize* dengan persentase *throttle* sebesar 30% selama 20 menit dan mencapai tegangan baterai dari keadaan penuh 16.8 v sampai 14.4 v. Berikut beberapa parameter yang didapatkan selama pengujian ini terlihat pada Gambar 4.5. Dalam hal keselamatan UAV dilakukan pengaturan *failsafe* pada *flight controller*. *Failsafe* yang digunakan adalah *battery failsafe*. Dimana dilakukan pengaturan batas limit tegangan baterai sebesar 14.4 v. Apabila tegangan baterai aktual berada dibawah nilai limit tegangan baterai yang ditetapkan, maka *flight controller* akan melakukan mode *Return To Launch* (RTL) sebagai aksi dari *failsafe*.

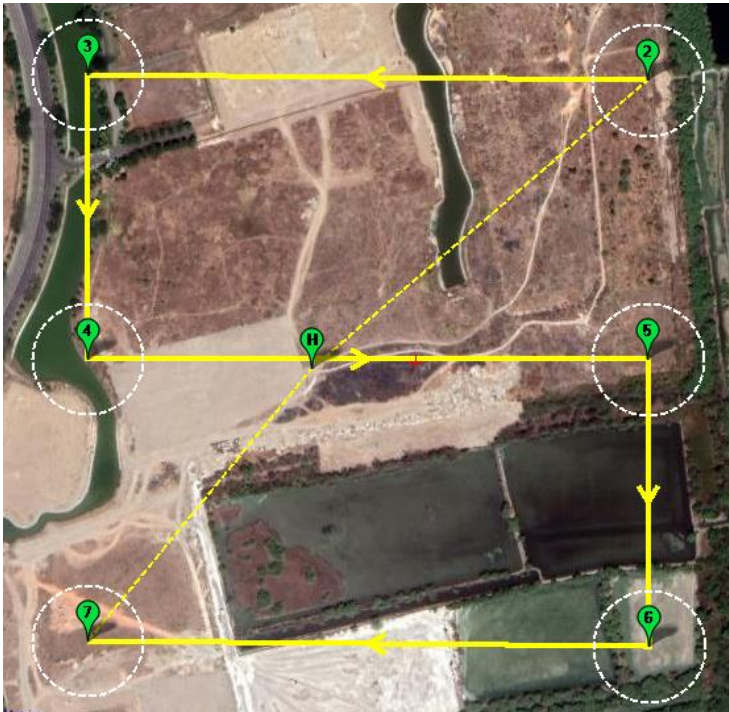
Servo Roll Pid P: 1.0 I: 0.30 D: 0.080 INT_MAX: 30	Servo Pitch Pid P: 1.3 I: 0.30 D: 0.040 INT_MAX: 30	Servo Yaw Yaw 2 roll: 1.00 Integral: 0.00 Dampening: 0.00 Intergrator Max: 15
L1 Control - Turn Control Period: 17 Damping: 0.75	Other Mix's P to T: [] Rudder Mi: 0.500	Throttle 0-100% Cruise: 30 Min: 0 Max: 50 SlewRate: 100
TECS Climb Max (m/s): 5.0 Sink Min (m/s): 2.0 Sink Max (m/s): 5.0 Pitch Dampening: 0.0 Time Const: 5.0	Navigation Angles Bank Max: 45 Pitch Max: 20 Pitch Min: -25	Airspeed m/s Cruise: 12.0 FBW mi: 9 FBW ma: 22 Ratio: 1.9936

Gambar 4.5. Nilai Parameter Kontrol UAV

4.3. Pengujian Sistem

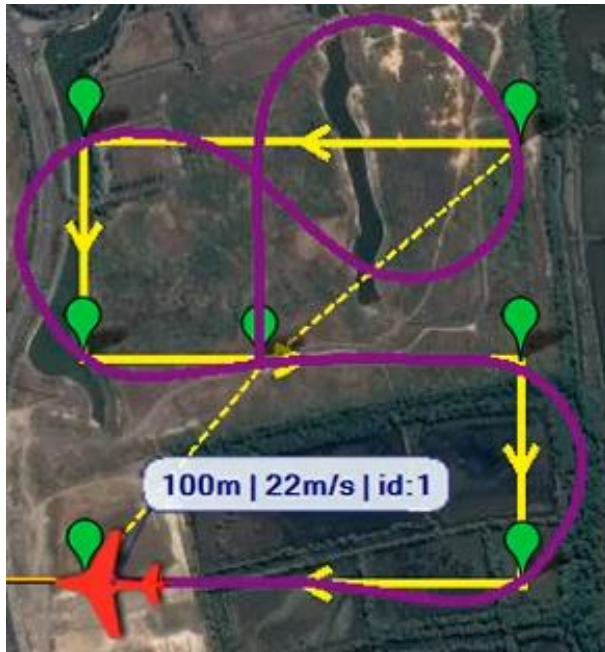
4.3.1. Simulasi

Pada tahap ini dilakukan pengujian dalam pengimplementasian keseluruhan algoritma optimasi terhadap UAV. Dimana dilakukan pemeriksaan kesesuaian protokol komunikasi yang disusun dalam algoritma optimasi terhadap tingkah laku UAV yang diinginkan. Pengujian dilakukan pada *software* Mission Planner dengan memanfaatkan fitur *simulation* pada *software* tersebut. Dengan menjalankan program optimasi pada terminal python maka pergerakan UAV akan tersimulasi melalui fitur tersebut. Dalam simulasi tersebut, dilakukan pengaturan rute rencana terbang yang dapat dilihat pada Gambar 4.6 dibawah ini.



Gambar 4.6. Pola Rute Penerbangan

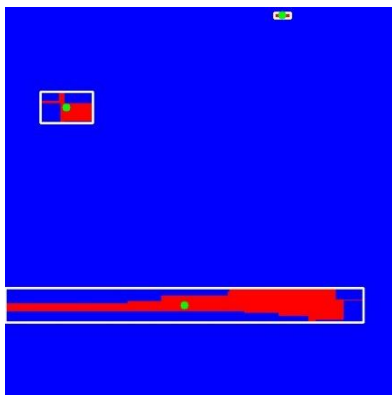
Kemudian dilanjutkan dengan menjalankan program optimasi yang telah terlampir pada bagian akhir buku ini. Dengan menjalankan program tersebut terlihat UAV bergerak sesuai dengan rute penerbangan yang direncanakan sebelumnya. Jalur pergerakan UAV dapat dilihat pada Gambar 4.7 dibawah ini.



Gambar 4.7. Simulasi Jalur Pergerakan UAV Saat Mengudara

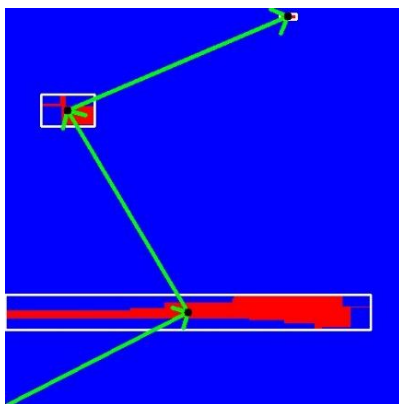
Setelah UAV berhasil mencapai titik *waypoint* terakhir, sesuai dengan program optimasi yang dirancang maka UAV akan melakukan analisa terhadap pergerakannya selama mengudara. Analisa ini bertujuan untuk mencari adanya indikasi keberadaan zona *blank spot* yang dihasilkan selama misi penerbangan berlangsung. Sehingga terlihat pada simulasi ketika keadaan mencapai titik *waypoint* terakhir maka dari

program yang dijalankan akan menghasilkan data seperti Gambar 4.8 dibawah ini.



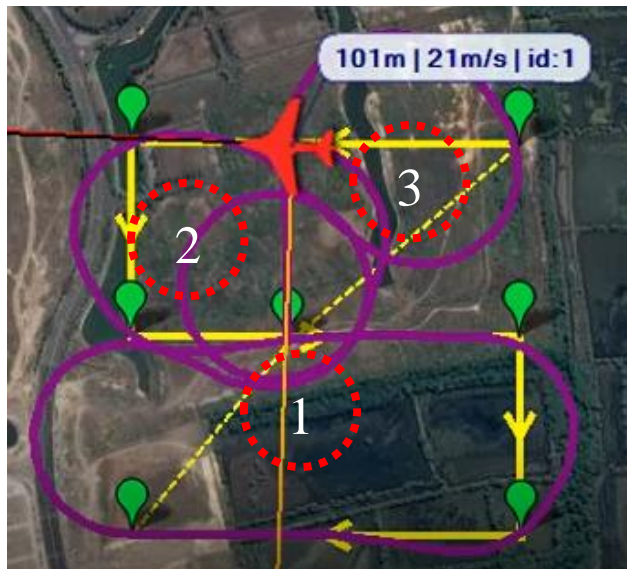
Gambar 4.8. Hasil Analisa Zona *Blank spot*

Pada tahap ini didapatkannya data berupa titik koordinat yang diindikasikan sebagai zona *blank spot*. Data yang didapatkan kemudian dilanjutkan dalam proses penyusunan rute penerbangan baru menggunakan algoritma genetika. Penerapan menggunakan algoritma genetika akan menghasilkan data rute penerbangan seperti yang diilustrasikan pada Gambar 4.9 dibawah ini.



Gambar 4.9. Hasil Implementasi Algoritma Genetika

Setelah didapatkannya data rute penerbangan baru tersebut maka data tersebut dikirimkan kepada UAV untuk dilakukan eksekusi. UAV akan terbang menuju seluruh *waypoint* pada rute penerbangan baru tersebut. Terlihat pada simulasi UAV berhasil menerima data dari hasil pemrosesan yang dilakukan pada tahap optimasi. UAV terlihat berhasil menuju titik *waypoint* yang terdapat pada rute penerbangan baru. Pergerakan UAV tersebut dapat dilihat pada Gambar 4.10 dibawah ini. Garis berwarna ungu menggambarkan jalur yang telah dilewati oleh UAV. Terlihat pada Gambar 4.10 dibawah, garis ungu yang tergambar pada simulasi telah berpotongan atau melewati radius wilayah titik *waypoint* pada rute penerbangan baru. Dengan berhasilnya UAV dalam terbang sesuai dengan yang direncanakan dan sukses dalam mengeksekusi titik rute penerbangan baru menandakan protokol komunikasi yang disusun pada algoritma optimasi yang dibuat pada tugas akhir ini dapat direalisasikan.



Gambar 4.10. Keseluruhan Jalur Pergerakan UAV

4.3.2. Realisasi

Pada tahap ini dilakukan pengujian terhadap keseluruhan sistem yang telah dibuat. Pengujian dilakukan pada suatu area lahan kosong di Perumahan Pakuwon City, Surabaya. Dalam pengujian kali ini dilakukan misi pemotretan untuk lahan seluas 2500 m^2 atau berukuran $500\text{ m} \times 500\text{ m}$. Pengujian dilakukan dengan memperhatikan dan mengimplementasikan nilai-nilai parameter yang terdapat pada pengujian sub bab 4.2. dan terdapat tambahan hal lainnya seperti dijelaskan berikut.

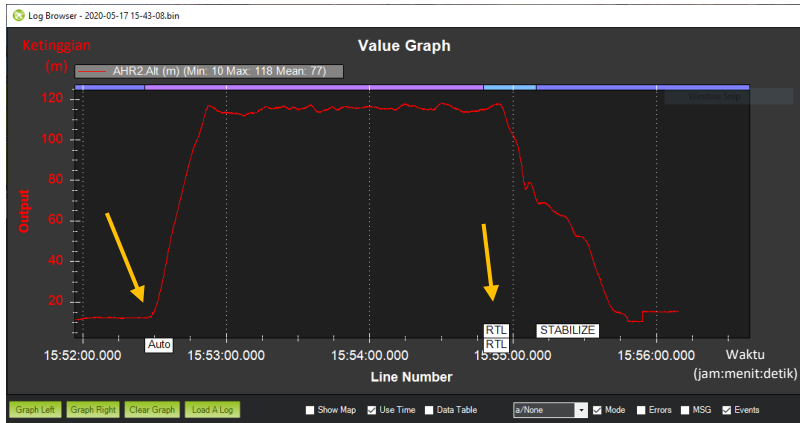
Dalam hal perencanaan jalur terbang dilakukan pengaturan nilai *sidelap* sebesar 0% dimana menghasilkan jarak antar jalur terbang horizontal sebesar 225 m . Sedangkan nilai *overlap* bergantung pada lama waktu *time lapse* yang diatur pada kamera. Pada misi penerbangan ini waktu *time lapse* pada kamera diatur selama 2 detik. Kedua, UAV akan mengudara pada ketinggian 120 m dan radius *waypoint* sebesar 50 m . Dengan parameter-parameter diatas didapatkan jalur penerbangan seperti Gambar 4.6.

Dalam misi penerbangan tersebut diawali dengan misi *auto Take-Off* hingga UAV mencapai ketinggian 100 meter dan misi penerbangan diakhiri dengan mode *Return To Launch* pada ketinggian 50 meter. Berikut hasil pengujian yang didapatkan.

4.3.2.1. Tanpa Penerapan Algoritma Pengembangan Sistem Autopilot

Pada pengujian ini dilakukan misi penerbangan dengan menggunakan sistem autopilot konvensional pada umumnya. Dimana UAV akan terbang mengeksekusi seluruh *waypoint* yang ada sebanyak satu kali dan kembali ke titik koordinat awal setelah melewati titik *waypoint* terakhir, kemudian melakukan pendaratan. Pengaturan misi penerbangan dan komunikasi antara GCS dilakukan melalui *interface software* Mission Planner. Setelah dilakukannya proses misi penerbangan tersebut dilakukan proses analisa terhadap seluruh data-data yang didapatkan terhadap UAV. Berikut data-data analisa yang didapatkan.

Pertama, UAV berhasil menyelesaikan misi penerbangan selama 2 menit 21 detik. Waktu tersebut terhitung dari UAV berada dalam mode AUTO pada waktu $15:52:26$ hingga UAV mencapai mode RTL yang menandakan telah menyelesaikan misi penerbangan pada waktu $15:55:47$



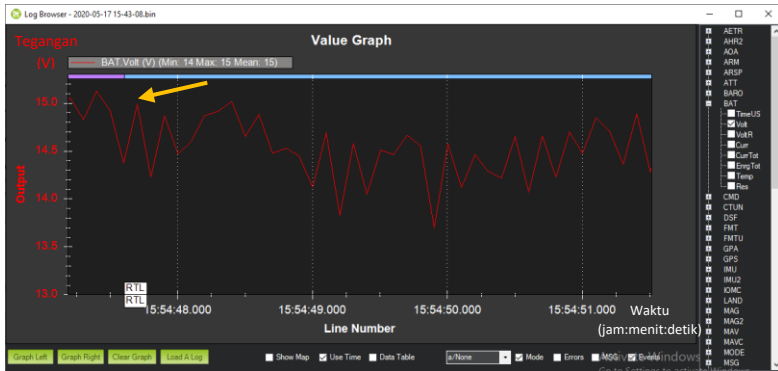
Gambar 4.11. Grafik Ketinggian Autopilot Konvensional Misi 1

Kedua, UAV dalam melakukan misi penerbangan tersebut menurunkan tegangan baterai sebesar 1.2 V. Terhitung dari UAV memasuki mode AUTO pada waktu 15:52:26 dimana kondisi tegangan baterai bernilai 16.2 V hingga UAV mencapai mode RTL dimana kondisi tegangan baterai bernilai 15.00 V pada waktu 15:55:47.



Gambar 4.12. Grafik 1 Tegangan Baterai Autopilot Konvensional Misi 1

1



Gambar 4.13. Grafik 2 Tegangan Baterai Autopilot Konvensional Misi 1

Ketiga, dalam misi penerbangan ini UAV berhasil melakukan pemotretan lahan dalam mode lensa kamera *wide lense* sebanyak 70 foto dengan waktu *time lapse* selama 2 detik dan resolusi kamera sebesar 16 MP. Kumpulan foto tersebut kemudian diolah dengan memasuki tahap proses *stitching* untuk mendapatkan gambar keseluruhan peta secara utuh.

Dalam melakukan proses *stitching* penulis menggunakan *software* Agisoft Photoscan Professional. Untuk menghasilkan gambar peta secara utuh dilakukan beberapa tahap pemrosesan yang dilakukan pada *software* tersebut. Hasil peta yang didapatkan akan dipengaruhi oleh setiap tahap pemrosesan didalamnya. Semakin tinggi kualitas yang diatur dalam setiap tahap pemrosesan akan menghasilkan gambar peta yang semakin baik namun waktu yang dibutuhkan dalam pemrosesan akan menjadi lebih lama. Berikut tahapan-tahapan yang dilakukan dalam proses *stitching* beserta waktu dan kualitas keluaran yang dibutuhkan atau dihasilkan.

Tabel 4.1. Tabel Kalkulasi Proses *Stitching* Penggunaan Autopilot Konvensional Misi 1

Urutan	Tahapan	Kualitas	Waktu(menit)
1	Align Photos	Low	05:21

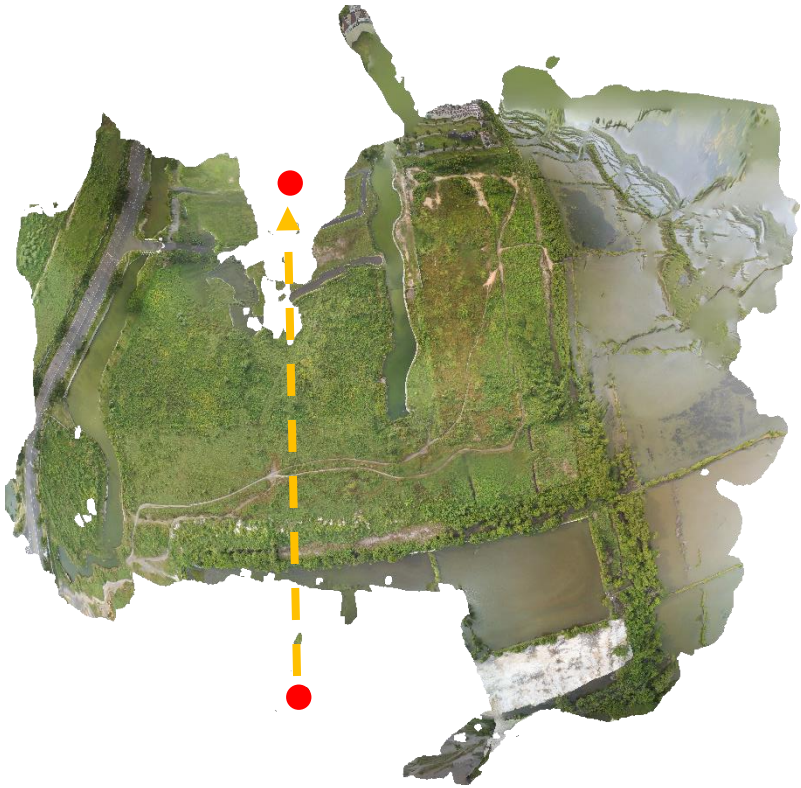
2	Build Dense Cloud	Low	03:54
3	Build Mesh	Low	04:00
4	Build Texture	Low	01:20
5	Build Tiled Model	Low	26:00
6	Build Orthomosaic	Low	05:05
7	Export Orthomosaic		00:37
Total Waktu			48 menit

Hasil *stitching* dapat dilihat pada Gambar 4.14 di bawah ini. Dapat dilihat dari foto peta dibawah, adanya area yang tidak tergambar sempurna. Hal ini dapat diindikasikan sebagai akibat adanya area yang tidak terfoto oleh UAV saat misi penerbangan berlangsung.



Gambar 4.14. Hasil *Stitching* Misi 1

Untuk menghasilkan hasil cakupan peta yang sesuai direncanakan sebelumnya diperlukan misi penerbangan lanjutan untuk melakukan pemotretan lahan pada bagian lahan yang diperkirakan tidak terfoto. Rencana penerbangan kedua didapatkan dari analisa hasil *stitching* pada misi penerbangan pertama. Berikut perencanaan titik *waypoint* rute penerbangan berdasarkan analisa hasil proses *stitching*.



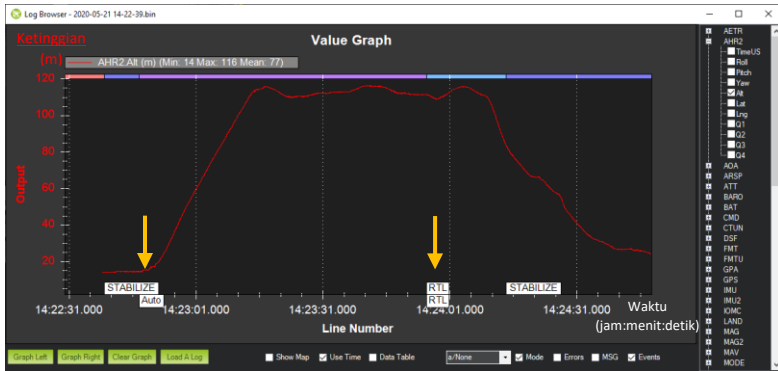
Gambar 4.15. Ilustrasi Rute Penerbangan Hasil Analisa Manual

Dari hasil analisa diatas didapatkan 2 titik *waypoint* dimana UAV diharuskan melakukan pemotretan ulang. Kemudian titik *waypoint* tersebut disusun pada *software Mission Planner* untuk menghasilkan rute penerbangan baru untuk UAV. Berikut Gambar 4.16 merupakan ilustrasi rencana penerbangan baru.



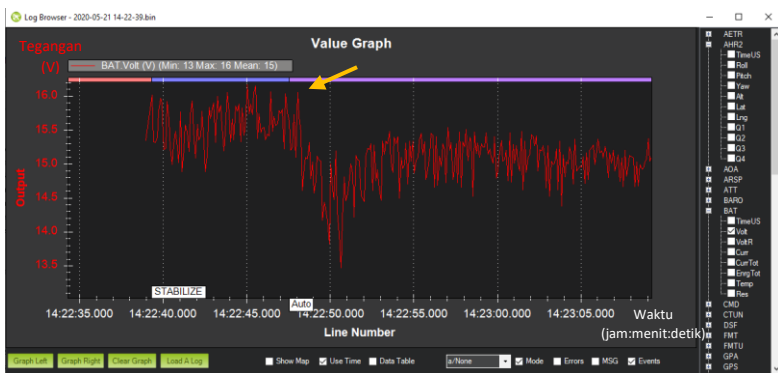
Gambar 4.16. Rute Penerbangan Misi 2

Dalam misi penerbangan tersebut. Pertama, UAV berhasil menyelesaikan misi penerbangan selama 1 menit 8 detik. Waktu tersebut terhitung dari UAV berada dalam mode AUTO pada waktu 14:22:47 hingga UAV mencapai mode RTL yang menandakan telah menyelesaikan misi penerbangan pada waktu 14:23:55.



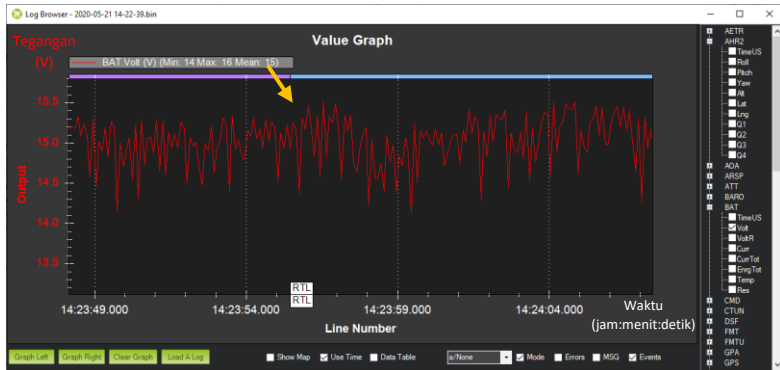
Gambar 4.17. Grafik Ketinggian Autopilot Konvensional Misi 2

Kedua, UAV dalam melakukan misi penerbangan tersebut menurunkan tegangan baterai sebesar 0.5 V. Terhitung dari UAV memasuki mode AUTO pada waktu 14:22:47 dimana kondisi tegangan baterai bernilai 16.0 V hingga UAV mencapai mode RTL dimana kondisi tegangan baterai bernilai 15.50 V pada waktu 14:23:55.



Gambar 4.18. Grafik 1 Tegangan Baterai Autopilot Konvensional Misi 2

2



Gambar 4.19. Grafik 2 Tegangan Baterai Autopilot Konvensional Misi 2

Ketiga, dalam misi penerbangan kedua ini UAV berhasil melakukan pemotretan lahan dalam mode kamera *wide lense* sebanyak 34 foto dengan waktu *time lapse* selama 2 detik dan resolusi kamera sebesar 16 MP. Kumpulan foto tersebut kemudian diolah dengan memasuki tahap proses *stitching* untuk mendapatkan gambar keseluruhan peta secara utuh.

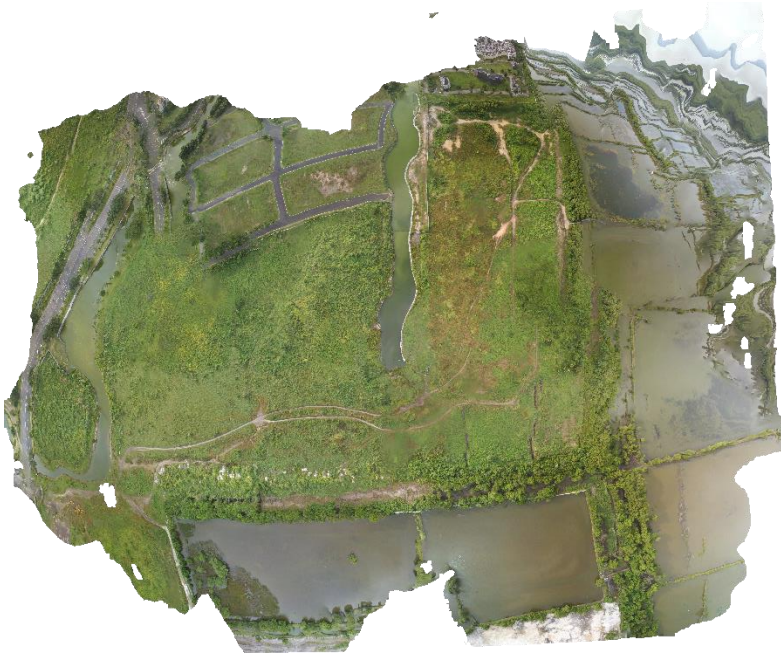
Tahap selanjutnya merupakan tahap *stitching*. Dimana pada tahap ini dilakukan penggabungan data foto pada misi penerbangan pertama dengan data foto pada misi penerbangan kedua. Berikut tahapan-tahapan yang dilakukan dalam proses *stitching* beserta waktu dan kualitas keluaran yang dibutuhkan atau dihasilkan.

Tabel 4.2. Tabel Kalkulasi Proses *Stitching* Penggunaan Autopilot Konvensional Misi 2

Urutan	Tahapan	Kualitas	Waktu(menit)
1	Align Photos	Low	07:16
2	Build Dense Cloud	Low	06:24
3	Build Mesh	Low	04:47

4	Build Texture	Low	01:57
5	Build Tiled Model	Low	25:00
6	Build Orthomosaic	Low	06:30
7	Export Orthomosaic		00:33
Total Waktu			55 menit

Hasil *stitching* dapat dilihat pada Gambar 4.20 dibawah ini. Terlihat zona *blank spot* yang terindikasi pada misi penerbangan pertama dapat diperbaiki pada misi penerbangan kedua. Sehingga peta yang dihasilkan sesuai dengan cakupan wilayah yang diinginkan.



Gambar 4.20. Hasil *Stitching* Penggabungan Foto Misi 1 dan Misi 2

Dari pembahasan diatas merupakan sistem autopilot konvensional yang pada umumnya dipakai dalam melakukan misi pemotretan lahan dari udara. Dimana pada umumnya untuk menghasilkan hasil peta yang sempurna dibutuhkan minimal 2 kali penerbangan. Penerbangan pertama dilakukan sesuai misi yang direncanakan dan pada misi penerbangan kedua dilakukan untuk melakukan koreksi terhadap data yang didapatkan pada misi penerbangan pertama. Berikut Tabel 4.3 hasil kalkulasi total waktu dan kapasitas baterai yang dibutuhkan.

Dalam perhitungan kapasitas baterai, diasumsikan baterai lipo 4200 mAh 4s 25C dengan tegangan fungsi ideal pada *range* 14.8 V hingga 16.8 V dan kurva *discharge* dianggap linier. Sehingga perhitungan dilakukan dengan rumus perbandingan seperti berikut.

$$\text{Kapasitas} = \frac{4200 \times \text{Penurunan Tegangan}}{2} \quad (4.1)$$

Tabel 4. 3. Tabel Kalkulasi Proses Pemetaan Dengan Sistem Autopilot Konvensional

Variabel	Keterangan		Total
	Misi 1	Misi 2	
Waktu Terbang	2 menit 21 detik	1 menit 8 detik	3 menit 29 detik
Waktu Stitching	48 menit	55 menit	1 jam 43 menit
		TOTAL >>	1 jam 46 menit
Penggunaan Kapasitas Baterai	$= \frac{4200 \times 1.2}{2}$ $= 2520 \text{ mAh}$	$= \frac{4200 \times 0.5}{2}$ $= 1050 \text{ mAh}$	$= 2520$ $+ 1050$ $= 3570 \text{ mAh}$

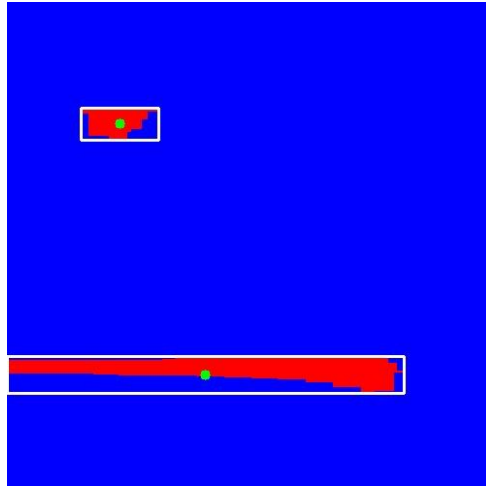
4.3.2.2. Penerapan Algoritma Pengembangan Sistem Autopilot

Pada tahap ini dilakukan penerapan algoritma pemrograman yang telah dirancang pada Tugas Akhir ini. Penerapan dilakukan bersamaan saat pelaksanaan misi penerbangan tanpa penerapan algoritma pengembangan sistem autopilot seperti diatas. Sehingga parameter ataupun variabel yang digunakan seperti jalur penerbangan dan lainnya mempunyai kesamaan. Algoritma yang diterapkan dapat dilihat pada

bagian lampiran pada buku ini. Pada pengimplementasian algoritma ini, misi penerbangan UAV dan komunikasi dengan GCS tidak dilakukan melalui *interface software* Mission Planner. Tetapi akan dilakukan secara keseluruhan melalui program optimasi yang dijalankan melalui terminal python. Melalui program optimasi tersebut seluruh pergerakan UAV dilakukan analisa. Dengan demikian dapat diketahui ada atau tidaknya keberadaan zona *blank spot*. Berikut Gambar 4.21 dan 4.22 hasil analisa pergerakan UAV selama misi penerbangan tersebut.



Gambar 4.21. Realisasi Hasil Pencatatan Pergerakan UAV



Gambar 4.22. Realisasi Hasil Analisa Zona *Blank Spot*

Dari hasil analisa tersebut didapatkan bahwa terdapat 2 buah area yang diindikasikan sebagai titik zona *blank spot*. Titik tersebut berada pada koordinat *canvas* (204, 384) dan (116, 125). Data tersebut jika dikonversikan dalam koordinat bumi menjadi (-7.282783096, 112.8180628608) dan (-7.280427749999999, 112.8172594032).

Kemudian data tersebut dilakukan pemrosesan menggunakan algoritma genetika untuk mendapatkan susunan rute penerbangan baru dalam melakukan eksekusi pemotretan ulang yang menghasilkan jarak tempuh paling pendek. Dalam pemrosesan menggunakan algoritma genetika terdapat beberapa nilai parameter yang ditetapkan. Nilai-nilai tersebut dapat dilihat pada Tabel 4.4 dibawah ini.

Tabel 4.4. Nilai Variabel Dalam Implementasi Algoritma Genetika

Variabel	Keterangan
Jumlah Populasi	30 kromosom
Elitism	2 kromosom
Persentasi Mutasi	30%
Jumlah Repetisi	100

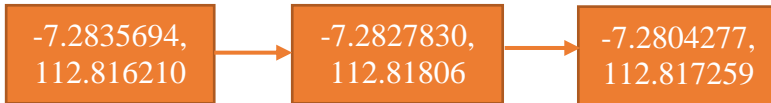
Hasil dari penerapan algoritma tersebut didapatkan suatu susunan rute penerbangan baru. Pada kasus ini didapatkan susunan rute sebagai berikut.

Berikut susunan rute penerbangan baru dalam koordinat *canvas*:

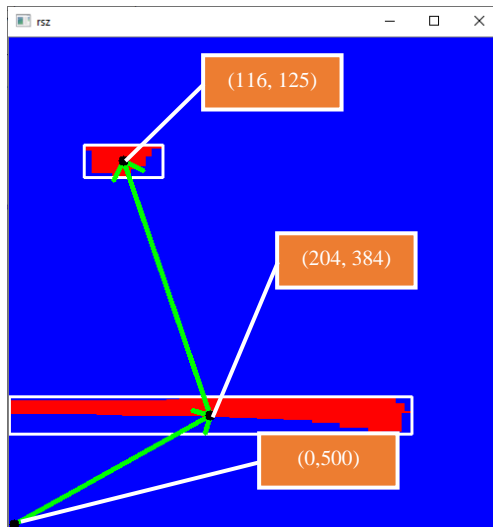


Gambar 4.23. Rute Penerbangan Dalam Koordinat *Canvas*

Berikut susunan rute penerbangan baru dalam koordinat bumi.:

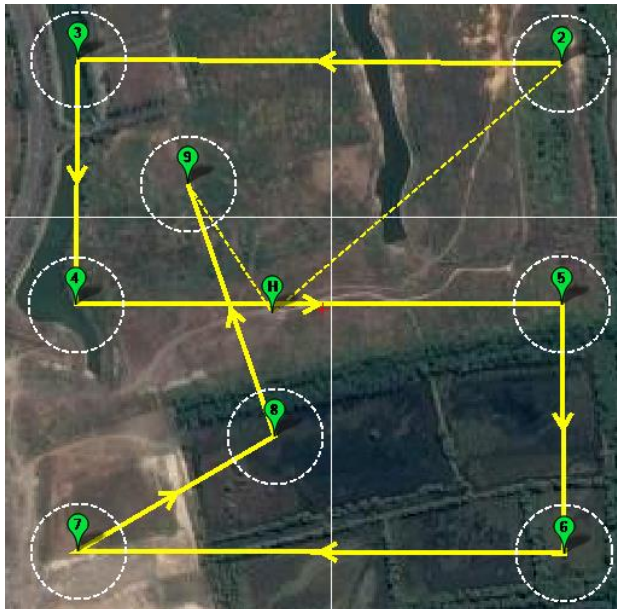


Gambar 4.24. Rute Penerbangan Dalam Koordinat Bumi



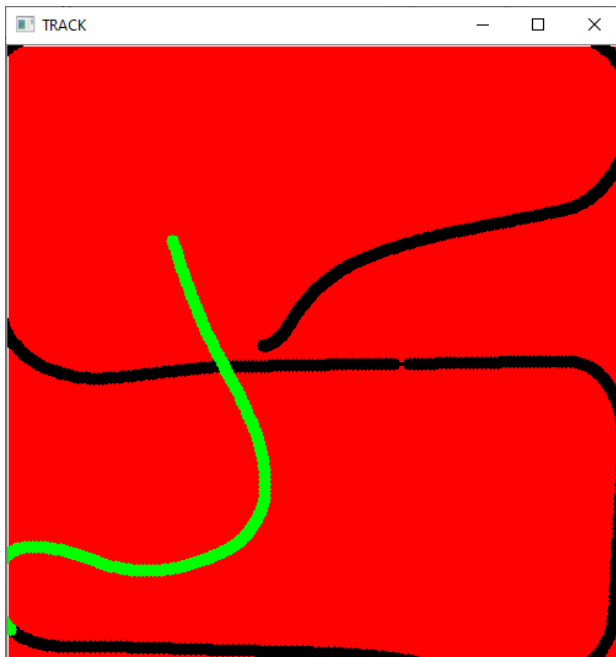
Gambar 4.25. Rute Penerbangan Hasil Implementasi Algoritma Genetika

Karena adanya keterbatasan komponen dan waktu pengujian yang tersedia dikarenakan wabah covid-19 yang sedang melanda, belum dilakukannya pengujian dengan menerapkan keseluruhan program pengembangan dan optimasi sesuai yang diusulkan pada tugas akhir ini. Oleh karenanya untuk mendapatkan data konsumsi baterai dan waktu yang dibutuhkan dalam mengeksekusi seluruh *waypoint* pada misi utama yang disertai penambahan *waypoint blank spot* hasil analisa, maka setelah UAV berhasil mendapatkan data *waypoint blank spot*, pilot akan melakukan pendaratan terhadap UAV. Kemudian dilakukan pengaturan misi penerbangan baru seperti yang dapat dilihat pada Gambar 4.26. Dimana misi penerbangan baru memiliki data *waypoint* yang sama dengan misi penerbangan utama tetapi diberi penambahan data *waypoint blank spot* yang didapatkan dari hasil analisa pada penerbangan sebelumnya. Kemudian dilakukan penerbangan UAV kembali dengan misi tersebut dan dilakukan *takeoff* dengan arah yang sama dengan penerbangan sebelumnya.



Gambar 4.26. Rute Penerbangan Hasil Analisa *Blank Spot*

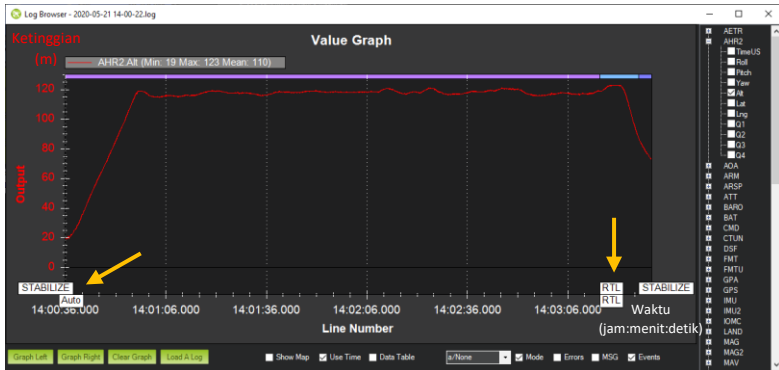
Dengan dilakukannya penerapan sistem autopilot tersebut, maka pergerakan UAV akan menjadi seperti yang diilustrasikan pada Gambar 4.27 dibawah ini. Jalur berwarna hijau merupakan alur pergerakan UAV ketika mengeksekusi rute penerbangan baru koordinat *blank spot*.



Gambar 4.27. Jalur Pergerakan UAV Dalam Penggunaan Sistem Autopilot Hasil Pengembangan

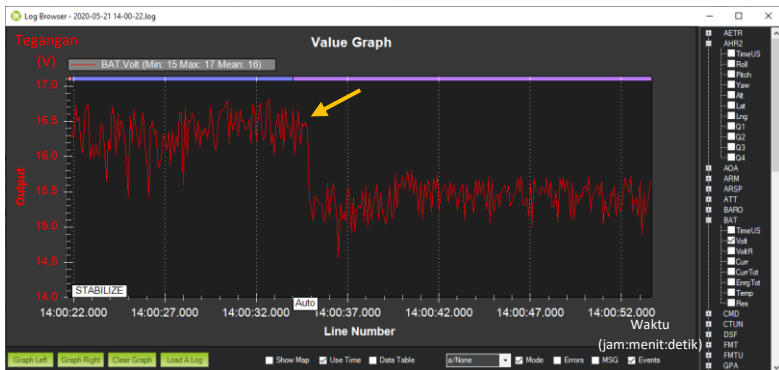
Setelah rute penerbangan baru selesai dieksekusi, UAV akan kembali ke titik awal dan pilot akan melakukan ambil alih kontrol UAV untuk melakukan proses pendaratan. Kemudian dilakukannya proses analisa terhadap seluruh data-data yang didapatkan terhadap UAV. Berikut data-data analisa yang didapatkan.

Pertama, UAV berhasil menyelesaikan misi penerbangan selama 2 menit 41 detik. Waktu tersebut terhitung dari UAV berada dalam mode AUTO pada waktu 14:00:34 hingga UAV mencapai mode RTL yang menandakan telah menyelesaikan misi penerbangan pada waktu 14:03:15



Gambar 4.28. Grafik Ketinggian Sistem Autopilot Hasil Pengembangan

Kedua, UAV dalam melakukan misi penerbangan tersebut menurunkan tegangan baterai sebesar 1 V. Terhitung dari UAV memasuki mode AUTO pada waktu 14:00:34 dimana kondisi tegangan baterai bernilai 16.5 V hingga UAV mencapai mode RTL dimana kondisi tegangan baterai bernilai 15.50 V pada waktu 14:03:15



Gambar 4.29. Grafik 1 Tegangan Baterai Sistem Autopilot Hasil Pengembangan



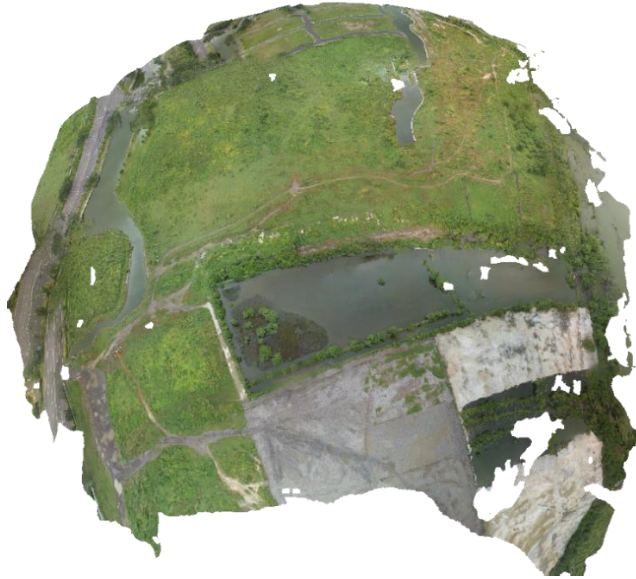
Gambar 4.30. Grafik 2 Tegangan Baterai Sistem Autopilot Hasil Pengembangan

Tahap selanjutnya merupakan tahap *stitching*. Dimana pada tahap ini dilakukan penggabungan data foto yang didapatkan selama misi penerbangan. Berikut tahapan-tahapan yang dilakukan dalam proses *stitching* beserta waktu dan kualitas keluaran yang dibutuhkan atau dihasilkan dan hasil peta yang didapatkan.

Tabel 4.5. Tabel Kalkulasi Proses *Stitching* Dengan Sistem Autopilot Hasil Pengembangan

Urutan	Tahapan	Kualitas	Waktu(menit)
1	Align Photos	Low	04:22
2	Build Dense Cloud	Low	03:50
3	Build Mesh	Low	06:13
4	Build Texture	Low	01:30
5	Build Tiled Model	Low	30:00
6	Build Orthomosaic	Low	05:37

7	Export Orthomosaic		00:33
Total Waktu			54 menit



Gambar 4.31. Hasil *Stitching* Penggunaan Sistem Autopilot Hasil Pengembangan

Dari analisa dan pembahasan diatas berikut Tabel 4.6 hasil kalkulasi total waktu dan kapasitas baterai yang dibutuhkan.

Tabel 4.6. Tabel Kalkulasi Proses Pemetaan Dengan Sistem Autopilot Hasil Optimasi

Variabel	Keterangan
	Misi 1
Waktu Terbang	2 menit 41 detik

Waktu Stitching	54 menit
TOTAL >>	56 menit 41 detik
Penggunaan Kapasitas Baterai	$= \frac{4200 \times 1}{2}$ $= 2100 \text{ mAh}$

4.3.2.3. Analisa Hasil Pengujian

Dapat dilihat dari kedua Gambar 4.32 hasil pemetaan dibawah, gambar sebelah kiri merupakan gambar peta yang diambil menggunakan sistem autopilot konvensional pada umumnya dan sebelah kanan merupakan gambar peta yang diambil menggunakan sistem autopilot hasil optimasi. Dimana terlihat pada gambar sebelah kiri peta yang dihasilkan memiliki dua buah area yang tidak tergambar atau disebut sebagai zona *blank spot*. Ketidakefektifan pada peta yang dihasilkan, dapat dianalisa dengan beberapa cara. Pertama, dilakukan perbandingan dengan hasil peta yang didapat dari hasil pemetaan menggunakan UAV dengan parameter nilai *sidelap* yang besar sehingga peta yang dihasilkan berpeluang sempurna dan hasil peta dijadikan sebagai acuan. Kedua, dapat dilakukan perbandingan menggunakan hasil peta yang didapat dari *platform google maps*. Namun, pada tugas akhir ini analisa hasil peta hanya dilakukan dengan melihat tingkat kesempurnaan pada hasil *stitching* yang didapatkan tanpa melakukan perbandingan dengan gambar peta yang dijadikan sebagai acuan referensi.

Permasalahan fenomena *blank spot* dapat terselesaikan dalam penggunaan sistem autopilot hasil optimasi dimana terlihat pada gambar peta sebelah kanan menghasilkan gambar peta yang lebih sempurna yang mana terhindar dari fenomena zona *blank spot* seperti yang terjadi dalam penggunaan sistem autopilot konvensional. Kemudian terlihat pada tabel 4.7 total waktu dan kapasitas baterai yang dibutuhkan dalam proses pemetaan menggunakan sistem autopilot konvensional menghasilkan waktu dan kapasitas baterai yang jauh lebih besar apabila dibandingkan dengan sistem autopilot optimasi. Sehingga didapatkannya nilai

persentase penghematan konsumsi baterai dan durasi waktu seperti pada rumus (4.1) dan (4.2) dibawah ini.



Gambar 4.32. Perbandingan Hasil *Stitching* Dalam Perbedaan Penggunaan Sistem Autopilot

Tabel 4.7. Hasil Perbandingan Autopilot Konvensional dan Autopilot Optimasi

Variabel	Keterangan	
	Autopilot Konvensional	Autopilot Optimasi
Waktu Terbang	3 menit 29 detik	2 menit 41 detik
Waktu Stitching	1 jam 43 menit	54 menit
TOTAL	1 jam 46 menit	56 menit
Konsumsi Baterai	3570 mAh	2100 mAh

$$\% \text{ Penghematan Waktu} = \frac{(106 \text{ menit} - 56 \text{ menit})}{106 \text{ menit}} \times 100 = 47.16 \% \quad (4.1)$$

$$\% \text{ Penghematan Baterai} = \frac{(3570 \text{ mAh} - 2100 \text{ mAh})}{3560 \text{ mAh}} \times 100 = 41.17\% \quad (4.2.)$$

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1. KESIMPULAN

Pada hasil pengujian yang telah dilakukan dapat disimpulkan bahwa penggunaan sistem autopilot hasil pengembangan dan optimasi pada Tugas Akhir ini relatif dapat meminimalisir terjadinya zona *blank spot* pada hasil peta yang didapatkan. Kemudian implementasi penggunaan algoritma genetika memberikan dampak terhadap nilai konsumsi baterai dan durasi penerbangan. Dimana jika dibandingkan hasil analisa yang didapatkan dari dua metode yang dibahas, durasi waktu dan besar kapasitas baterai yang dibutuhkan dalam penggunaan algoritma pengembangan sistem autopilot yang dikembangkan pada Tugas Akhir ini menghasilkan waktu dan besar kapasitas baterai yang lebih sedikit dan kecil dibandingkan dengan penggunaan sistem autopilot konvensional pada umumnya. Dalam satu kali pengujian, UAV terbang menggunakan baterai 4S 4200 mAh 25C pada ketinggian 120 m dengan nilai sudut kontrol *roll* sebesar 45 derajat dan persentase *cruise throttle* sebesar 30% apabila dilihat dari variabel konsumsi daya baterai dan durasi waktu, maka penggunaan algoritma pengembangan sistem autopilot yang dirancang pada Tugas Akhir ini akan menghemat 41% dari kapasitas baterai dan 47 % dari durasi waktu yang dibutuhkan apabila dibandingkan dengan penggunaan sistem autopilot konvensional pada umumnya.

5.2. SARAN

Pada penelitian selanjutnya dapat dilakukan pengembangan dalam penambahan penggunaan komponen mikrokontroler pada UAV sebagai perangkat utama pengganti laptop dalam seluruh proses penjalanan program optimasi pada UAV. Dengan demikian proses berjalannya optimasi tidak terpengaruhi oleh kuat sinyal komponen telemetry antara GCS dan UAV, sehingga proses pemetaan dapat dilakukan untuk cakupan wilayah yang lebih luas.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] M. Z. Spasial, “Perbedaan UAV dan Drone,” *Zona Spasial*, Nov. 13, 2018. <https://zonaspasial.com/2018/11/perbedaan-uav-dan-drone/> (accessed Nov. 01, 2019).
- [2] T. H. Purwanto, “Pemanfaatan Foto Udara Format Kecil untuk Ekstraksi Digital Elevation Model dengan Metode Stereoplotting,” *Maj. Geogr. Indones.*, vol. 31, p. 17, 2017.
- [3] H. Gularso, I. S. Subiyanto, M. Si, and L. M. Sabri, “Jurnal Geodesi Undip April 2013,” vol. 2, p. 17, 2013.
- [4] R. Mardiyanto, R. Hidayat, E. Aprilian, and H. Suryoatmojo, “Development of Autopilot system of Unmanned Aerial Vehicle for Aerial Mapping Application,” in *2018 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Bali, Indonesia, Aug. 2018, pp. 357–361, doi: 10.1109/ISITIA.2018.8710966.
- [5] I. Suroso, “PERAN DRONE/UNMANNED AERIAL VEHICLE (UAV) BUATAN STTKD DALAM DUNIA PENERBANGAN,” p. 9.
- [6] S. Wibowo, R. Sumiharto, and R. Hujja, “Desain Pengembangan Autopilot Pesawat Udara Tanpa Awak Menggunakan AVR-Xmega Sebagai Perangkat OBDH,” *J. Teknol.*, vol. 8, pp. 11–19, Jan. 2015.
- [7] A. Ahmad and A. M. Samad, “Aerial mapping using high resolution digital camera and unmanned aerial vehicle for Geographical Information System,” in *2010 6th International Colloquium on Signal Processing & its Applications*, Mallaca City, May 2010, pp. 1–6, doi: 10.1109/CSPA.2010.5545303.
- [8] K. N. Tahar, A. Ahmad, W. A. A. W. M. Akib, and W. M. N. W. Mohd, “Aerial mapping using autonomous fixed-wing unmanned aerial vehicle,” in *2012 IEEE 8th International Colloquium on Signal Processing and its Applications*, Malacca, Malaysia, Mar. 2012, pp. 164–168, doi: 10.1109/CSPA.2012.6194711.
- [9] “Pixhawk Overview — Copter documentation.” <http://ardupilot.org/copter/docs/common-pixhawk-overview.html> (accessed Nov. 20, 2019).
- [10] “Mission Planner Overview — Mission Planner documentation.” <https://ardupilot.org/planner/docs/mission-planner-overview.html> (accessed Mar. 20, 2020).

- [11] S. Tjhin, *SISTEM KEAMANAN SEPEDA MOTOR MELALUI SHORT MESSAGE SERVICE MENGGUNAKAN AVR MIKROKONTROLER ATMEGA8*. 2014.
- [12] G. R. Bradski and A. Kaehler, *Learning OpenCV: [computer vision with the OpenCV library*. Beijing: O'Reilly, 2011.
- [13] A. Dhawan, "Implementation of Hand Detection based Techniques for Human Computer Interaction," *Int. J. Comput. Appl.*, vol. 72, p. 8.
- [14] D. Krisrenanto, M. Rivai, and F. Budiman, "Identifikasi Jumlah dan Tingkat Aktivitas Orang Berbasis Pengolahan Citra Menggunakan Raspberry Pi," *J. Tek. ITS*, vol. 6, no. 1, pp. 105–109, Mar. 2017, doi: 10.12962/j23373539.v6i1.21397.
- [15] "Contours : Getting Started — OpenCV-Python Tutorials 1 documentation." https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html (accessed Mar. 20, 2020).
- [16] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*. Hoboken, NJ: Wiley, 2005.
- [17] Kevin Krisnandi and Halim Agung, "Implementasi Algoritma Genetika untuk Memprediksi Waktu dan Biaya Pengerjaan Proyek Konstruksi," *J. Ilm. Fifo*, vol. 9, no. 2, pp. 90–97, 2017, doi: 10.22441/fifo.v9i2.2565.
- [18] D. Sundarningsih, W. Mahmudy, and Sutrisno, "Penerapan algoritma genetika untuk optimasi vehicle routing problem with time window (VRPTW) : Studi kasus air minum kemasan," *DORO Repos. J. Mhs. PTHK Univ. Brawijaya*, vol. 5, Jan. 2015.
- [19] M. Muliadi, "PEMODELAN ALGORITMA GENETIKA PADA SISTEM PENJADWALAN PERKULIAHAN PRODI ILMU KOMPUTER UNIVERSITAS LAMBUNG MANGKURAT," vol. 01, p. <http://klik.ulm.ac.id/index.php/klik/article/view/8>, Sep. 2014, doi: 10.20527/klik.v1i1.8.
- [20] B. Santosa and P. Willy, *Metoda Metaheuristik Konsep dan Implementasi*. Guna Widya, 2011.
- [21] S. Kuswara, *Paradigma Sistem Cerdas*. Surabaya: Bayumedia, 2003.
- [22] A. W. wati, "Penerapan Algoritma Genetika Dalam Optimasi Model Dan Simulasi Dari Suatu Sistem," *J. Keilmuan Teh. Ind. 1 4*, 2011.

- [23] Kusri Kusri and Jazi Eko Istiyanto, “Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data,” *J. Inform. Univ. Petra Kristian*, vol. 8, no. 2, Nov. 2007.
- [24] S. Lukas, T. Anwar, and W. Yuliani, “Penerapan Algoritma Genetika untuk Traveling Salesman Problem dengan Menggunakan Metode Order Crossover dan Insertion Mutation,” vol. 0, Jan. 2005.
- [25] D. M. Pierre, N. Zakaria, and A. J. Pal, “Master-Slave Parallel Vector-Evaluated Genetic Algorithm for Unmanned Aerial Vehicle’s path planning,” in *2011 11th International Conference on Hybrid Intelligent Systems (HIS)*, Melacca, Malaysia, Dec. 2011, pp. 517–521, doi: 10.1109/HIS.2011.6122158.
- [26] Z. Cheng and D. Li, “Improved GASA Algorithm for Mutation Strategy UAV Path Planning,” in *2018 10th International Conference on Communication Software and Networks (ICCSN)*, Chengdu, Jul. 2018, pp. 506–510, doi: 10.1109/ICCSN.2018.8488319.
- [27] Ze Cheng, Ying Sun, and Yanli Liu, “Path planning based on immune genetic algorithm for UAV,” in *2011 International Conference on Electric Information and Control Engineering*, Wuhan, China, Apr. 2011, pp. 590–593, doi: 10.1109/ICEICE.2011.5777407.
- [28] Y. V. Pehlivanoglu and A. Hacioglu, “Vibrational Genetic Algorithm Based Path Planner for Autonomous UAV in Spatial Data Based Environments,” in *2007 3rd International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, Jun. 2007, pp. 573–578, doi: 10.1109/RAST.2007.4284058.
- [29] M. Cakir, “2D path planning of UAVs with genetic algorithm in a constrained environment,” in *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, Istanbul, Turkey, May 2015, pp. 1–5, doi: 10.1109/ICMSAO.2015.7152235.
- [30] Z. He and L. Zhao, “The Comparison of Four UAV Path Planning Algorithms Based on Geometry Search Algorithm,” in *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, Aug. 2017, pp. 33–36, doi: 10.1109/IHMSC.2017.123.
- [31] J. Chen, F. Ye, and T. Jiang, “Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm,”

- in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Chengdu, Oct. 2017, pp. 1434–1438, doi: 10.1109/ICCT.2017.8359869.
- [32] B. Meng and X. Gao, “UAV Path Planning Based on Bidirectional Sparse A* Search Algorithm,” in *2010 International Conference on Intelligent Computation Technology and Automation*, Changsha, China, May 2010, pp. 1106–1109, doi: 10.1109/ICICTA.2010.235.
- [33] A. M. Moradi Sizkouhi, S. Majid Esmailifar, M. Aghaei, A. K. Vidal de Oliveira, and R. Ruther, “Autonomous Path Planning by Unmanned Aerial Vehicle (UAV) for Precise Monitoring of Large-Scale PV plants,” in *2019 IEEE 46th Photovoltaic Specialists Conference (PVSC)*, Chicago, IL, USA, Jun. 2019, pp. 1398–1402, doi: 10.1109/PVSC40753.2019.8980862.
- [34] H. Li, S. Wu, P. Xie, Z. Qin, and B. Zhang, “A Path Planning for One UAV Based on Geometric Algorithm,” in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, Xiamen, China, Aug. 2018, pp. 1–5, doi: 10.1109/GNCC42960.2018.9019122.
- [35] N. Lu, Y. Gong, and J. Pan, “Path planning of mobile robot with path rule mining based on GA,” in *2016 Chinese Control and Decision Conference (CCDC)*, Yinchuan, China, May 2016, pp. 1600–1604, doi: 10.1109/CCDC.2016.7531239.
- [36] L. Yujie, P. Yu, S. Yixin, Z. Huajun, Z. Danhong, and S. Yong, “Ship Path Planning Based on Improved Particle Swarm Optimization,” in *2018 Chinese Automation Congress (CAC)*, Xi’an, China, Nov. 2018, pp. 226–230, doi: 10.1109/CAC.2018.8623037.
- [37] D. Zhang, Y. Xian, J. Li, G. Lei, and Y. Chang, “UAV Path Planning Based on Chaos Ant Colony Algorithm,” in *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, Hangzhou, China, Oct. 2015, pp. 81–85, doi: 10.1109/CSMA.2015.23.
- [38] Z. Lv, L. Yang, Y. He, Z. Liu, and Z. Han, “3D environment modeling with height dimension reduction and path planning for UAV,” in *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, Kunming, Jul. 2017, pp. 734–739, doi: 10.1109/ICMIC.2017.8321551.
- [39] S. Dawnee, M. M. S. Kumar, S. Jayanth, and V. K. Singh, “Experimental performance evaluation of various path planning

- algorithms for obstacle avoidance in UAVs,” in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, Jun. 2019, pp. 1029–1034, doi: 10.1109/ICECA.2019.8821841.
- [40] J. A. Gonçalves and R. Henriques, “UAV photogrammetry for topographic monitoring of coastal areas,” *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 101–111, Jun. 2015, doi: 10.1016/j.isprsjprs.2015.02.009.
- [41] “SkyWalker Series_Aircraft Systems_HOBBYWING TECHNOLOGY CO., LTD. -.”
http://www.hobbywing.com/goods.php?id=407&filter_attr=
(accessed Jul. 06, 2020).
- [42] FrSky, “Taranis Q X7,” *FrSky - Lets you set the limits*, Jul. 18, 2017. <https://www.frsky-rc.com/taranis-q-x7-3/> (accessed Jul. 06, 2020).
- [43] “3DR Pixhawk 1 · PX4 v1.9.0 User Guide.”
https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk.html
(accessed Jul. 06, 2020).
- [44] “Common Power Module — Copter documentation.”
<https://ardupilot.org/copter/docs/common-3dr-power-module.html>
(accessed Jul. 06, 2020).

Halaman ini sengaja dikosongkan

LAMPIRAN A

➤ Program Sistem Autopilot Hasil Pengembangan dan Optimasi

```
from dronekit import connect, VehicleMode,
LocationGlobalRelative, Command, Battery,
LocationGlobal, Attitude
from pymavlink import mavutil

import time
import math
import numpy as np
import psutil
import argparse
import copy
import cv2
import random
from random import shuffle
import pandas as pd
import xlwt
from xlwt import Workbook

#Set up option parsing to get connection string
connection_string = "COM6"

print("Connection to the vehicle on %s" %
connection_string)
vehicle = connect(connection_string, baud=57600,
wait_ready=False)
print("vehicle is connected")

#####
#####
```

```

lat1=-7.279291
long1=112.8162003
lat2=-7.279291
long2=112.8207654
lat3=-7.283838
long3=112.8207654
deltalong=long2-long1
deltalat=lat2-lat3

im = cv2.imread("merah.jpg")
rsz= cv2.resize(im,(500, 500),interpolation =
cv2.INTER_NEAREST)

global r
wb=Workbook()
wbGA=Workbook()

sheet1 = wb.add_sheet('koordinat')
sheet2 = wbGA.add_sheet('track GA')

#####
#####
def set_ap_mode(mode):                #--- Set
Autopilot mode
    time_0 = time.time()
    try:
        tgt_mode    = VehicleMode(mode)
    except:
        return(False)

    while (vehicle.mode != tgt_mode):
        vehicle.mode = tgt_mode
        time.sleep(0.2)
        if time.time() < time_0 + 5:

```

```

        return (False)

    return (True)

def go_wp(lat,long,altt,radius,currentway,sumway):
    jrk=math.inf
    wp=LocationGlobalRelative(lat,long,altt)
    vehicle.simple_goto(wp)
    global r
    while jrk>radius:
        #print(" Global Location: %s" %
plane.vehicle.location.global_frame)
        live_gps = vehicle.location.global_frame

jrk=distanceBetween(live_gps.lat,live_gps.lon, lat,
long)
        print('menuju waypoint GA
%d/%d:%d'%(currentway+1,sumway,jrk))
        #print(jrk)
        plot(live_gps.lat,live_gps.lon)

#print(vehicle.location.global_relative_frame.alt)
        sheet2.write(r, 0,live_gps.lat)
        sheet2.write(r, 1,live_gps.lon)
        r=r+1

def distanceBetween(lat1, long1, lat2, long2):
    delta = math.radians(long1-long2)
    sdlong = math.sin(delta)
    cdlong = math.cos(delta)

    lat1 = math.radians(lat1)
    lat2 = math.radians(lat2)

    slat1 = math.sin(lat1)

```

```

    clat1 = math.cos(lat1)
    slat2 = math.sin(lat2)
    clat2 = math.cos(lat2)

    delta = (clat1 * slat2) - (slat1 * clat2 *
cdlong)
    delta = delta*delta
    delta += ((clat2 * sdlong)*(clat2 * sdlong))
    delta = math.sqrt(delta)
    denom = (slat1 * slat2) + (clat1 * clat2 *
cdlong)
    delta = math.atan2(delta, denom)

    return delta * 6372795

def bumiTOcanvas(latlive,lonlive):
    xx=((500*(longlive-long1))/deltalong)
    yy=((500*(lat1-latlive))/deltalat)
    return xx,yy

def canvasTObumi(pointx,pointy):
    long=((pointx*(deltalong)/500)+long1)
    lat=((-(pointy*deltalat)/500)+lat1 )
    return lat,long

def plot(koorlat,koorlon):
    poinz=bumiTOcanvas(koorlat,koorlon)
    x=poinz[0]
    y=poinz[1]
    pF=167
    lF=222
    #pF=100
    #lF=100
    x1=int (x-(pF/2))
    y1=int (y-(lF/2))

```

```

x2=int (x+(pF/2))
y2=int (y+(lF/2))
cv2.rectangle(rsz, (x1, y1), (x2, y2),
(255,0,0), -1)

def analisaBS():
    cenx=[]
    ceny=[]
    cenxy=[]
    gray = cv2.cvtColor(rsz, cv2.COLOR_BGR2GRAY)
    retval, fix = cv2.threshold(gray, 50, 255,
cv2.THRESH_BINARY)

    contours, hierarchy =
cv2.findContours(fix,cv2.RETR_EXTERNAL,cv2.CHAIN_AP
PROX_SIMPLE)
    try: hierarchy = hierarchy[0]
    except: hierarchy = []
    for contour, hier in zip(contours, hierarchy):
        (x,y,w,h) = cv2.boundingRect(contour)
        cv2.rectangle(rsz, (x,y), (x+w,y+h), (255,
255, 255), 2)
        cv2.circle(rsz,(int(x+w/2),int(y+h/2)),
5,(0, 255, 0) ,-1)
        print(int(x+w/2),"",int(y+h/2))
        cenx.append(int(x+w/2))
        ceny.append(int (y+h/2))
        cenxy.append((int (x+w/2),int(y+h/2)))
    print("total blob=",len(contours))
    live_gps = vehicle.location.global_frame
    tambah=bumiTOcanvas(live_gps.lat,live_gps.lon)
    loix=round (tambah[0])
    loiy=round (tambah[1])
    cenx.insert(0,loix)
    ceny.insert(0,loiy)

```

```

cenxy.insert(0,(loix,loiy))
return cenx,ceny,cenxy,len(contours)+1

#####
#####
def ukurjarak(pop):
    tanda=pop
    jartot=0
    for i in range(len(tanda)-1):
        jarak=math.sqrt(((ye[tanda[i+1]]-
ye[tanda[i]])**2) + ((ex[tanda[i+1]]-
ex[tanda[i]])**2))
        jartot=jartot+jarak
    return jartot

def fitnes(pop2):
    global bestr
    global bestpop
    fitness=0
    save=0
    fitness=[]
    for i in range(len(pop2)):
        d=ukurjarak(pop2[i])
        if d<bestr:
            bestr=d
            bestpop=pop2[i]
            save=i
            print(bestr)
            print(bestpop)
            f=1/(pow(d,8)+1)
            fitness.append((f))
    return fitness,save

def normfit(fitfit):
    #normalize fitness

```

```

sum=0
fitfix=0
fitfix=[]
for i in range(len(fitfit)):
    sum=sum+fitfit[i]
for i in range(len(fitfit)):
    f=fitfit[i]/sum
    fitfix.append((f))
return fitfix

def elitism(pop,fit,many,fitbes):
    fitsort=fit.copy()
    fitsort.sort()
    kopifit=[]
    for t in range(many):
        for i in range(len(fit)):
            if fit[i]==fitsort[t]:
                pop[i]=bestpop
                fit[i]=math.inf
                kopifit.append((i))
    for i in range(len(kopifit)):
        fit[kopifit[i]]=fitbes
    return pop

def pickone(list,prob):
    index=0
    r=random.random()
    while(r>0):
        if index>len(list)-1:
            index=0
            r=r-prob[index]
            index=index+1
    index=index-1
    #print(index)
    return list[index]

```

```

def crossover(orderAkk,orderBee):
    orderAk=orderAkk[1:]
    orderBe=orderBee[1:]
    test=0
    neworder=0
    neworder=[]
    start=random.randint(0,len(orderAk)-2)
    end=random.randint(start+1,len(orderAk)-1)
    for i in range(start,end+1):
        neworder.append((orderAk[i]))
    test=np.array(neworder+orderBe)
    _, idx =np.unique(test,return_index=True)
    test=test[np.sort(idx)]
    tested=test.tolist()
    return tested

def mutate(order,rate):
    #print(order)
    baru=order
    for i in range(len(order)):
        x=random.random()
        if x<rate:
            temp1=random.randint(1,len(order)-1)
            temp2=(temp1+1)%len(order)
            baru=swap(order,temp1,temp2)
    baru.insert(0,0)
    return baru

def swap(arr,i,j):
    temp=arr[i]
    arr[i]=arr[j]
    arr[j]=temp
    nu=arr
    return nu

```



```

#####
#####

while not vehicle.mode.name=="AUTO":
    print(" Mode: %s" % vehicle.mode.name)

r=0
while vehicle.mode.name=="AUTO":
    live_gps = vehicle.location.global_frame
    plot(live_gps.lat, live_gps.lon)
    sheet1.write(r, 0, live_gps.lat)
    sheet1.write(r, 1, live_gps.lon)
    print("plotting")
    print(r)
    r=r+1
    time.sleep(1)

hasilBS=analisaBS()
clone1 = copy.copy(rsz)
#cv2.imshow('ZONA BS', clone1)
cv2.imwrite(r"D:\SALIK\TUGAS AKHIR\latihan drone
kit\how_do_drones_work-master\PLANC\ZONA BLANK
SPOT.jpg", clone1)

if hasilBS[3]>2:
    ex=hasilBS[0]
    ye=hasilBS[1]
    v=hasilBS[2]
    kota=hasilBS[3]
    t=[]
    order=[]

```

```

fitness=[]
bestr=math.inf
popsize=30

popu=0
popu=[]
for i in range(kota):
    order.append((i))
    pot=order[1:]
for i in range(popsize):
    shuffle(pot)
    t=pot.copy()
    t.insert(0,0)
    popu.append(t)

print("VV GA MODE ON VV")
for i in range(100):
    a=0
    b=0
    a=fitnes(popu)
    b=normfit(a[0])
    popu=elitism(popu,b,2,b[a[1]])
    nupop=0
    nupop=[]
    for i in range (len(popu)):
        orderA=pickone(popu,b)
        orderB=pickone(popu,b)
        orderC=crossover(orderA,orderB)
        orderD=mutate(orderC,0.3)
        nupop.append((orderD))
    popu=nupop
#print(v)
#print(bestpop)

```

```

    for i in range(len(bestpop)-1):
        cv2.arrowedLine(rsz, v[bestpop[i]],
v[bestpop[i+1]], (0,255,0), 3, 5, 0, 0.08)
        #cv2.line(rsz,
v[bestpop[i]],v[bestpop[i+1]], (0,0,0), 2)
    for i in range(len(bestpop)):
        cv2.circle(rsz, v[bestpop[i]], 5, (0,0,0),
-1)

    print("VV KOORDINAT BLANK SPOT VV")
    nulat=[]
    nulong=[]
    nuord=[]
    for i in range(len(bestpop)-1):

nukoor=canvasTObumi(ex[bestpop[i+1]],ye[bestpop[i+1]
])
        nulat.append(nukoor[0])
        nulong.append(nukoor[1])
        nuord.append((nukoor[0],nukoor[1]))
    print(nuord)
    clone2 = copy.copy(rsz)
    #cv2.imshow('PATTERN WITH GA',clone2)
    cv2.imwrite(r"D:\SALIK\TUGAS AKHIR\latihan
drone kit\how_do_drones_work-master\PLANC\PATTERN
WITH GA.jpg",clone2)

    print("VV EKSEKUSI KOORDINAT BLANK SPOT VV")

    while not vehicle.mode.name=="GUIDED":
        print(" Mode: %s" % vehicle.mode.name)
        set_ap_mode("GUIDED")
        time.sleep(2)
    print(" Mode: %s" % vehicle.mode.name)

```

```

r=0
for i in range(len(nuord)):

go_wp(nulat[i],nulong[i],100,80,i,len(nuord))

    print("VV SELESIAI VV")
    #cv2.imshow('SETELAH GA',rsz)
    cv2.imwrite(r"D:\SALIK\TUGAS AKHIR\latihan
drone kit\how_do_drones_work-master\PLANC\AFTER
GA.jpg",rsz)

while not vehicle.mode.name=="RTL":
    print(" Mode: %s" % vehicle.mode.name)
    set_ap_mode("RTL")
    time.sleep(2)

print(" Mode: %s" % vehicle.mode.name)
wb.save('log GPS.xls')
wbGA.save('log GPS GA.xls')

cv2.waitKey(0)
cv2.destroyAllWindows()

```

LAMPIRAN B

➤ Program Penggambaran Jalur Penerbangan UAV

```
import pandas as pd
import cv2

datway = pd.read_excel('log GPS.xls',usecols =
"A,B",skiprows=0,header=None)
datGA = pd.read_excel('log GPS GA.xls',usecols =
"A,B",skiprows=0,header=None)

im = cv2.imread("merah.jpg")
rsz= cv2.resize(im,(500, 500),interpolation =
cv2.INTER_NEAREST)

waylat=datway[0].tolist()
waylong=datway[1].tolist()

latGA=datGA[0].tolist()
longGA=datGA[1].tolist()

lat1=-7.279291
long1=112.8162003
lat2=-7.279291
long2=112.8207654
lat3=-7.283838
long3=112.8207654
deltalong=long2-long1
deltalat=lat2-lat3

#####
def bumiTOcanvas(latlive,longlive):
    xx=(500*(longlive-long1))/deltalong
    yy=(500*(lat1-latlive))/deltalat
```

```

    return xx,yy
#####

nuord=[]
for i in range(len(waylat)):
    nukoor=bumiTOcanvas(waylat[i],waylong[i])
    nuord.append((round
(nukoor[0]),round(nukoor[1])))

for i in range(len(waylat)-1):
    cv2.line(rsz, nuord[i],nuord[i+1], (0,0,0), 2)
for i in range(len(waylat)):
    cv2.circle(rsz, nuord[i], 5, (0,0,0), -1)

trackGA=[]
for i in range(len(latGA)):
    koor=bumiTOcanvas(latGA[i],longGA[i])
    trackGA.append((round
(koor[0]),round(koor[1])))

for i in range(len(latGA)-1):
    cv2.line(rsz, trackGA[i],trackGA[i+1],
(0,255,0), 2)
for i in range(len(latGA)):
    cv2.circle(rsz, trackGA[i], 5, (0,255,0), -1)

cv2.imshow('TRACK',rsz)

```

LAMPIRAN C

➤ Dokumentasi





LAMPIRAN D

➤ Datasheet Komponen

1. Hobbywing Skywalker ESC 60A[41]

Specifications

Model	Cont. Current	Burst Current (50%)	BEC Mode	BEC Output	BEC Output Capability				Battery Cell		Weight	Size L*W*H
					2S Lipo	3S Lipo	4S Lipo	5S Lipo	Lipo	NiMH		
Skywalker-6A	6A	8A	Linear	5V0.8A	3 servos				2S	5-6 cells	5.5g	32*12*4.5
Skywalker-12A	12A	15A	Linear	5V1A	3 servos	2 servos			2-3S	5-9 cells	8g	38*18*6
Skywalker-12AE	12A	15A	Linear	5V2A	5 servos	4 servos			2-3S	5-9 cells	10g	38*18*7
Skywalker-15A	15A	20A	Linear	5V2A	5 servos	4 servos			2-3S	5-9 cells	16.5g	48*22*5.6
Skywalker-20A	20A	25A	Linear	5V2A	5 servos	4 servos			2-3S	5-9 cells	19g	42*25*6
Skywalker-30A	30A	40A	Linear	5V2A	5 servos	4 servos			2-3S	5-9 cells	37g	68*25*6
Skywalker-40A	40A	55A	Linear	5V3A	5 servos	4 servos			2-3S	5-9 cells	39g	68*25*6
Skywalker-40A-UBEC	40A	55A	Switch	5V3A	5 servos	5 servos	5 servos		2-4S	5-12 cells	43g	65*25*12
Skywalker-50A-UBEC	50A	65A	Switch	5V5A	8 servos	8 servos	8 servos		2-4S	5-12 cells	41g	65*29*10
Skywalker-60A-UBEC	60A	80A	Switch	5V5A	8 servos	8 servos	8 servos	8 servos	2-6S	5-18 cells	63g	77*35*14
Skywalker-60A-OPTO	60A	80A	N/A	N/A					2-6S	5-18 cells	60g	80*38*12
Skywalker-80A-UBEC	80A	100A	Switch	5V5A	8 servos	8 servos	8 servos	8 servos	2-6S	5-18 cells	62g	80*38*12
Skywalker-80A-OPTO	80A	100A	N/A	N/A					2-6S	5-18 cells	79g	80*38*12

Programmable Items (The option written in bold font is the default setting)

- Brake Setting: Enabled / Disabled
- Battery Type: **Lipo** / NiMH
- Low Voltage Protection Mode(Cut-Off Mode): **Soft Cut-Off (Gradually reduce the output power)** /Cut-Off (Immediately stop the output power)
- Low Voltage Protection Threshold(Cut-Off Threshold): Low / **Medium** / High
 - For lithium battery, the battery cell number is calculated automatically. Low / medium / high cutoff voltage for each cell is: 2.85V/3.15V/3.3V. For example: For a 3S Lipo, when "Medium" cutoff threshold is set, the cut-off voltage will be: 3.15*3=9.45V
 - For NiMH battery, low / medium / high cutoff voltages are 0%/50%/65% of the startup voltage (i.e. the initial voltage of battery pack), and 0% means the low voltage cut-off function is disabled. For example: For a 6 cells NiMH battery, fully charged voltage is 1.44*6=8.64V, when "Medium" cut-off threshold is set, the cut-off voltage will be: 8.64*50%=4.32V.
- Startup Mode: **Normal** / Soft / Super-Soft (300ms / 1.5s / 3s)
 - Normal mode is suitable for fixed-wing aircraft. Soft or Super-soft modes are suitable for helicopters. The initial acceleration of the Soft and Super-Soft modes are slower, it takes 1.5 second for Soft startup or 3 seconds for Super-Soft startup from initial throttle advance to full throttle. If the throttle is completely closed (throttle stick moved to bottom position) and opened again (throttle stick moved to top position) within 3 seconds after the first startup, the re-startup will be temporarily changed to normal mode to get rid of the chance of a crash caused by slow throttle response. This special design is suitable for aerobatic flight when quick throttle response is needed.
- Timing: **Low** / Medium / High, (3.75° / 15° / 26.25°)
Usually, low timing is suitable for most motors. To get higher speed, High timing value can be chosen.

Begin To Use Your New ESC

IMPORTANT! Because different transmitter has different throttle range, please calibrate throttle range before flying.

Throttle range setting (Throttle range should be reset whenever a new transmitter is being used)

Switch on the transmitter, move throttle stick to the top position	Connect battery pack to the ESC, and wait for about 2 seconds	The "Beep-Beep" tone should be emitted, means the top point of throttle range has been confirmed	Move throttle stick to the bottom position, several "beep-" tones should be emitted to present the amount of battery cells	A long "Beep-" tone should be emitted, means the lowest point of throttle range has been correctly confirmed
--	---	--	--	--

Normal startup procedure

Move throttle stick to bottom position and then switch on transmitter.	Connect battery pack to ESC, special tone like "♪ 123" means power supply is OK	Several "beep-" tones should be emitted to present the amount of lithium battery cells	When self-test is finished, a long "beep-----" tone should be emitted	Move throttle stick upwards to go flying
--	---	--	---	--

Protection Function

- Start up failure protection: If the motor fails to start within 2 seconds of throttle application, the ESC will cut-off the output power. In this case, the throttle stick **MUST** be moved to the bottom again to restart the motor. (Such a situation happens in the following cases: The connection between ESC and motor is not reliable, the propeller or the motor is blocked, the gearbox is damaged, etc.)
- Over-heat protection: When the temperature of the ESC is over about 110 Celsius degrees, the ESC will reduce the output power.

3. Throttle signal loss protection: The ESC will reduce the output power if throttle signal is lost for 1 second, further loss for 2 seconds will cause the output to be cut-off completely.

Trouble Shooting

Trouble	Possible Reason	Action
After power on, motor does not work, no sound is emitted	The connection between battery pack and ESC is not correct	Check the power connection. Replace the connector.
After power on, motor does not work, such an alert tone is emitted: "beep-beep, beep-beep-beep-beep." (Every "beep-beep" has a time interval of about 1 second)	Input voltage is abnormal, too high or too low.	Check the voltage of battery pack
After power on, motor does not work, such an alert tone is emitted: "beep-, beep-, beep-" (Every "beep-" has a time interval of about 2 seconds)	Throttle signal is irregular	Check the receiver and transmitter Check the cable of throttle channel
After power on, motor does not work, such an alert tone is emitted: "beep-, beep-, beep-" (Every "beep-" has a time interval of about 0.25 second)	The throttle stick is not in the bottom (lowest) position	Move the throttle stick to bottom position
After power on, motor does not work, a special tone "♪♯6712" is emitted after 2 beep tone (beep-beep-)	Direction of the throttle channel is reversed, so the ESC has entered the program mode	Set the direction of throttle channel correctly
The motor runs in the opposite direction	The connection between ESC and the motor need to be changed.	Swap any two wire connections between ESC and motor

Program the ESC with your transmitter (4 Steps)

Note: Please make sure the throttle curve is set to 0 when the throttle stick is at bottom position and 100% for the top position.

1. Enter program mode
2. Select programmable items
3. Set item's value (Programmable value)
4. Exit program mode

1. Enter program mode

- 1) Switch on transmitter, move throttle stick to top position, connect the battery pack to ESC
- 2) Wait for 2 seconds, the motor should emit special tone like "beep-beep-"
- 3) Wait for another 5 seconds, special tone like "♪♯6712" should be emitted, which means program mode is entered



2. Select programmable items

After entering program mode, you will hear 8 tones in a loop with the following sequence. If you move the throttle stick to bottom within 3 seconds after one kind of tones, this item will be selected.

1. "beep" brake (1 short tone)
2. "beep-beep-" battery type (2 short tone)
3. "beep-beep-beep-" cutoff mode (3 short tone)
4. "beep-beep-beep-beep-" cutoff threshold (4 short tone)
5. "beep----" startup mode (1 long tone)
6. "beep----beep-" timing (1 long 1 short)
7. "beep----beep-beep-" set all to default (1 long 2 short)
8. "beep----beep----" exit (2 long tone)

Note: 1 long "beep----" = 5 short "beep-"



3. Set item value (Programmable value)

You will hear several tones in loop. Set the value matching to a tone by moving throttle stick to top when you hear the tone, then a special tone "♪♯1515" emits, means the value is set and saved. (Keeping the throttle stick at top, you will go back to Step 2 and you can select other items; or moving the stick to bottom within 2 seconds will exit program mode directly)

Items	Tones	"beep-" 1 short tone	"beep-beep-" 2 short tones	"beep-beep-beep-" 3 short tones
Brake		Off	On	
Battery type		Lipo	NiMH	
Cutoff mode		Soft-Cut	Cut-Off	
Cutoff threshold		Low	Medium	High
Start mode		Normal	Soft	Super soft
Timing		Low	Medium	High



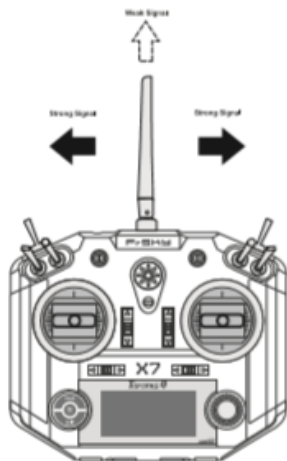
4. Exit program mode

There are 2 ways to exit program mode:

1. In step 3, after special tone "♪♯1515", please move throttle stick to the bottom position within 2 seconds.
2. In step 2, after tone "beep----beep----" (th at is: The item #8), move throttle stick to bottom within 3 seconds.

⚠ Cautions on handling antenna

- ⊗ Do not touch the antenna during operation. Doing so could interfere with transmission, causing a crash.
- ⊗ Do not carry the transmitter by the antenna. The antenna wire could break and prevent transmission.
- ⊗ Do not pull the antenna forcefully. The antenna wire could break and prevent transmission.

**Rotating Antenna**

The antenna can be rotated 180 degrees and angled 90 degrees. Forcing the antenna further than this can cause damage to the antenna. The antenna is not removable.

Angle adjustment of the antenna

The antenna rotation and angle can be adjusted. The antenna features weak radio signal in the forward direction and strong radio signal in the sideways directions. Adjust the antenna angle to match your flying style.

Specifications

Model Name: Taranis Q X7

Number of channels: up to 16 channels

Operating Voltage Range: 6-15V (2S, 3S Lipos are acceptable)

Operating Current: 210mA maximum (both RF module and backlight are on)

Operating Temperature: -10-45 C

Backlight LCD Screen: 128*64 outdoor readable LCD

Model Memories: 60 (extendable by MicroSD (TF) card)

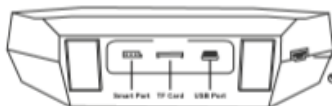
Compatibility: FrSky X series, D series and V8-II series receivers (plus other receivers if an external module is used)

Taranis Q X7-EU version is only compatible with X/LR series receivers EU version.

Features

- Quad Ball Bearing Gimbals
- Receiver Match
- Audio Speech Outputs (values, alarms, settings, etc.)
- Real-time Flight Data Logging
- Receiver Signal Strength Indicator (RSSI) Alerts
- Super Low Latency
- Vibration Alerts
- Model files are compatible with TARANIS X9D/X9D Plus/X9E.
- Open source firmware OpenTx installed.

FrSky Electronic Co., Ltd www.frsky-cs.com Contact us : frsky@frsky-cs.com
 Add:F-4,Building C, Zhongguo Technology Park, No.3 Yuansi Road, Wuxi, 214125, Jiangsu, China Technical Support: sales@frsky.com



Interface definition

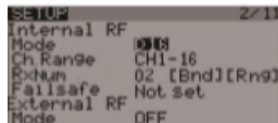
1. TF card is not provided with shipment.
2. USB port is for upgrading and reading / writing MicroSD cards and internal memory of radio contents.

⚠ Notes and Warnings for Battery (Not provided)

- ⚠ Please connect a battery in the battery compartment before use.
- ⚠ The voltage range should be DC 6-15V.
- ⚠ Be careful not to drop the battery.
- ⚠ Don't pull the battery wires as this could produce short-circuits and may cause the battery to fire even explode.
- ⚠ Do not remove the battery from the TARANIS Q X7 transmitter while the voltage warning is blinking as this could cause internal settings and memories to be destroyed.
- ⚠ Do not use the transmitter if a "Backup Error" warning occurs.

Model Setup for Taranis Q X7 internal RF Module

Enter the MODEL SETUP menu.



Step 1: Set the Mode for Taranis Q X7 Internal RF

Refer to the table below and set the Taranis Q X7 to the mode corresponding to your receiver (D8, D16 or LR12).

Mode of Taranis Q X7	Compatible Receivers	Number of Output Channels
D8	V8-II series in D mode (V8FR-II, V8R7-II, V8R4-II, VD5M, etc.) D series (D8R-II plus, D8R-XP, D6FR, D4R-II, etc.)	8 channels
D16	X series (X8R, etc.)	Up to 16 channels
LR12	L series (L9R, etc.)	12 channels

Mode of Taranis Q X7-EU version	Compatible Receivers	Number of Output Channels
D16-EU	X series-EU version receivers	Up to 16 channels
LR12	L series (L9R, etc.)	12 channels

3. Pixhawk[43]

❖ Specifications

- *Processor*
 - *32bit STM32F427 Cortex-M4F core with FPU*
 - *168 MHz*
 - *256 KB RAM*
 - *2 MB Flash*
 - *32 bit STM32F103 failsafe co-processor*
- *Sensors*
 - *ST Micro L3GD20H 16 bit gyroscope*
 - *ST Micro LSM303D 14 bit accelerometer / magnetometer*
 - *Invensense MPU 6000 3-axis accelerometer/gyroscope*
 - *MEAS MS5611 barometer*
- *Interfaces*
 - *5x UART (serial ports), one high-power capable, 2x with HW flow control*
 - *2x CAN (one with internal 3.3V transceiver, one on expansion connector)*
 - *Spektrum DSM / DSM2 / DSM-X® Satellite compatible input*
 - *Futaba S.BUS® compatible input and output*
 - *PPM sum signal input*
 - *RSSI (PWM or voltage) input*
 - *I2C*
 - *SPI*
 - *3.3 and 6.6V ADC inputs*
 - *Internal microUSB port and external microUSB port extension*
- *Normal Operation Maximum Ratings*
 - *Power module input (4.8V to 5.4V)*
 - *Servo rail input (4.8V to 5.4V) UP TO 10V FOR MANUAL OVERRIDE, BUT AUTOPILOT PART*

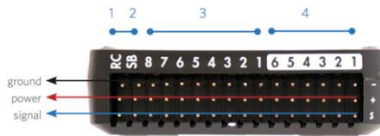
**WILL BE UNPOWERED ABOVE 5.7V IF POWER
MODULE INPUT IS NOT PRESENT**

- USB power input (4.8V to 5.4V)

❖ *Connectors*



- 1 Input/output reset button
- 2 SD card
- 3 Flight management reset button
- 4 Micro-USB port



- 1 Radio control receiver input
- 2 S.Bus output
- 3 Main outputs
- 4 Auxillary outputs

Gambar Lampiran 1. Pixhawk²

² Dronecode Project, "3DR Pixhawk 1", diakses dari https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk.html pada tanggal 06 juli 2020, pukul 19.03 WIB.

❖ Pinouts

TELEM1, TELEM2 ports

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	TX (OUT)	+3.3V
3 (blk)	RX (IN)	+3.3V
4 (blk)	CTS (IN)	+3.3V
5 (blk)	RTS (OUT)	+3.3V
6 (blk)	GND	GND

GPS port

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	TX (OUT)	+3.3V
3 (blk)	RX (IN)	+3.3V
4 (blk)	CAN2 TX	+3.3V
5 (blk)	CAN2 RX	+3.3V
6 (blk)	GND	GND

SERIAL 4/5 port - due to space constraints two ports are on one connector.

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	TX (#4)	+3.3V
3 (blk)	RX (#4)	+3.3V
4 (blk)	TX (#5)	+3.3V
5 (blk)	RX (#5)	+3.3V
6 (blk)	GND	GND

ADC 6.6V

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	ADC IN	up to +6.6V
3 (blk)	GND	GND

ADC 3.3V

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	ADC IN	up to +3.3V
3 (blk)	GND	GND
4 (blk)	ADC IN	up to +3.3V
5 (blk)	GND	GND

I2C

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	SCL	+3.3 (pullups)
3 (blk)	SDA	+3.3 (pullups)
4 (blk)	GND	GND

CAN

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	CAN_H	+12V
3 (blk)	CAN_L	+12V
4 (blk)	GND	GND

SPI

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	SPI_EXT_SCK	+3.3
3 (blk)	SPI_EXT_MISO	+3.3
4 (blk)	SPI_EXT_MOSI	+3.3
5 (blk)	ISPI_EXT_NSS	+3.3
6 (blk)	IGPIO_EXT	+3.3
7 (blk)	GND	GND

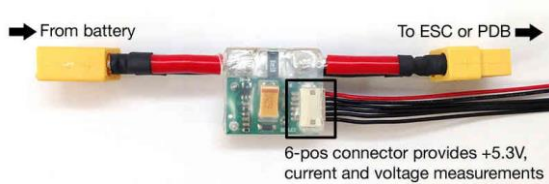
POWER

Pin	Signal	Volt
1 (red)	VCC	+5V
2 (blk)	VCC	+5V
3 (blk)	CURRENT	+3.3V
4 (blk)	VOLTAGE	+3.3V
5 (blk)	GND	GND
6 (blk)	GND	GND

SWITCH

Pin	Signal	Volt
1 (red)	VCC	+3.3V
2 (blk)	IIO_LED_SAFETY	GND
3 (blk)	SAFETY	GND

4. Power Module[44]



Gambar Lampiran 2. Power Module³

❖ Specifications

- Maximum input voltage of 18V (4S lipo).
- Maximum of 90 Amps (but only capable of measuring up to 60 Amps).
- Provides 5.37V and 2.25Amp power supply to the autopilot.

³ ArdupilotDevTeam, "Common Power Module", diakses dari <https://ardupilot.org/copter/docs/common-3dr-power-module.html>, pada tanggal 06 juli 2020, pukul 19.32 WIB.

5. Frsky FX400R

FX400R V1.0 Frsky compatible receiver Manual

Specifications

Dimension: 29mm*15mm*6mm
Weight:3g
SBUS output only
Work Mode : D16
Channels : SBUS CH1-CH16
Smart port ready
Operating Range: About 1000 meter
Operating voltage: 5V

Features:

Smart Port enabled
SBUS output
Lighter weight and physically smaller than X4R/X4RSB
Compatible With F3 / F4 series flight controller

Overview:

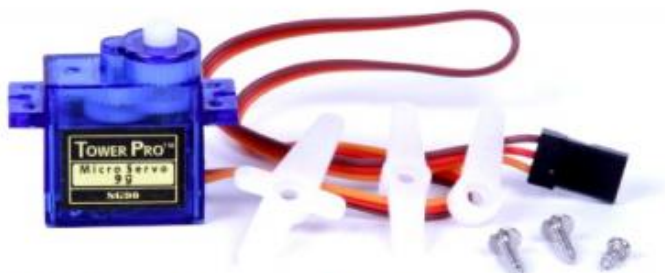


Green LED	Red LED	Status
Solid	Solid	Binding mode
ON	Flashing	Binding Successful
Solid	OFF	Normal
OFF	Flashing	Signal Lost
Flashing Twice	OFF	Failsafe Set

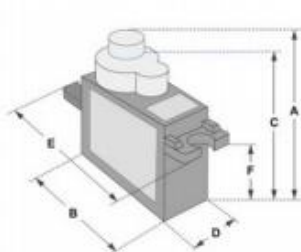
6. Servo Tower Pro SG-90

SERVO MOTOR SG90

DATA SHEET



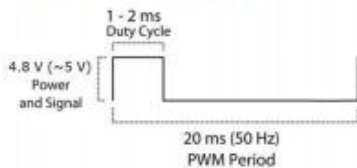
Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "180" (~1ms pulse) is all the way to the left.

Dimensions & Specifications	
A (mm) :	32
B (mm) :	23
C (mm) :	28.5
D (mm) :	12
E (mm) :	32
F (mm) :	19.5
Speed (sec) :	0.1
Torque (kg-cm) :	2.5
Weight (g) :	14.7
Voltage :	4.8 - 6

PWM=Orange (⏏)
Vcc = Red (+)
Ground=Brown (-)



7. 3DR Radio Telemetry



Gambar Lampiran 3. 3DR Radio Telemetry 500mW⁴

DESCRIPTION	SPECIFICATIONS	WARRANTY	REVIEWS (3)	ATTACHMENT (1)	DISCUSSION FORUM (1)
	Range(m)				2500
	Frequency				433MHz
	Rated Power (W)				0.5
	Receive Sensitivity(dBm)				-124
	Current(A)				0.5
	Voltage(V)				3.6 to 7
	Operating Temperature (°C)				-10 to 85
	Module Size (LxWxH) mm				54x260x12
	Antenna Diameter(mm)				9
	Antenna Length(mm)				105
	Cable Length(mm)				200
	Weight(gm)				67
	Shipment Weight				0.165 kg
	Shipment Dimensions				10 × 6 × 4 cm

Activate

⁴ ROBU.IN,” 3DR Radio Telemetry 433MHz 500mW for PIXHAWK and APM”, diakses dari <https://robu.in/product/3dr-radio-telemetry-433mhz-500mw-for-pixhawk-and-apm/>, pada tanggal 06 juli 2020, pukul 22.22 WIB.

8. Ublox NEO M8N GPS Module with Compass For APM 2.6 2.8 Pixhawk PX4

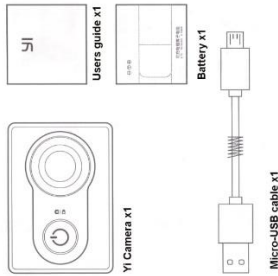
DESCRIPTION	SPECIFICATIONS	WARRANTY	REVIEWS (5)	ATTACHMENT (1)	VIDEOS (1)
DISCUSSION FORUM					

Model	Ublox NEO-M8N
Receiver Type	72-channel Ublox M8 engine.
Main Chip	Ublox NEO-M8N
Supply Voltage (V)	0.5~3.6
Sensitivity	Tracking & Navigation: -161 dBm
Position Accuracy	Autonomous: 2.5 m SBAS: 2.0 m
Acceleration	<4g
Navigation Update Rate	up to 18 HZ.
Operating Temperature Range	-45°C to 105°C
Tracking Sensitivity	-161 dBm
Co-Ordinate System	WGS-84
Capture Time	0.1s Average
Cold Start Time	30s
Warm Start Time	1s
Boot Time	1s Average
Maximum Altitude	18000
Dimensions (mm) LxWxH	54 x 22 (Dia x W)
Weight (gm)	23g
Cable Length	20 cm
Shipment Weight	0.135 kg
Shipment Dimensions	7 x 6 x 6 cm

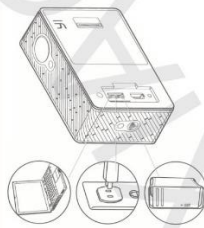
9. Xiaomi yi cam

Getting started with Yi Action Camera

Included:

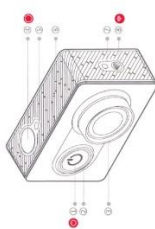


For additional charge it is possible to connect USB cable to the power adapter or to the PC. It is recommended to use adapters 5V/1.5A and more portable power units.



Charging

Operating of a camera



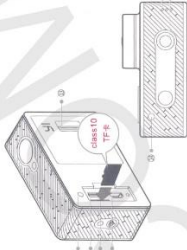
- 1) power-ON button/switching of the mode
- 2) ring-shaped LED indicator
- 3) lens
- 4) videophoto button
- 5) status indicator
- 6) microphone

- 7) WiFi indicator
- 8) power-ON/WIFI button for Bluetooth

Main Operations

- 1) **Power on/off/Power On:** press the button
Power Off: press the button for longer (if unloaded, the red indicator lights up)
- 2) **Connections/shut down of WiFi:**
Connecting to WiFi: press the button and you are connected to the network. **Disconnecting from WiFi:** After successful connection to WiFi, the dark blue indicator will light up.
Switching-off WiFi: press the button, and you at once are connectable disconnected from WiFi. The indicator blinks 5 times and once extinguished.

Operating of a camera



- 1) status indicator
- 2) micro-USB Connector
- 3) socket for TF Card
- 4) socket for HDMI
- 5) battery cover
- 6) status indicator
- 7) holes for dynamics
- 8) slot for monopod

If you do not use high-quality TF card (max. Supports a card on 64 Gb), it can effect quality of filming or cause its interruption.

Main Operations

- 3) **Switching of modes:**
After switch on of the camera, push a button power On to change a mode provided.
Photo: factory adjustments of a mode for operating of camera after loading by default. (they can be changed through the mobile app). press the button of a photovideo - from is possible to take a picture.
Video: push a button power on for switching in a mode of video (red indicator lights up). After successful connection to WiFi, the dark blue indicator will light up. Once again press the button of a photovideo and you will film in 3D.
- 4) **Color of the LED indicator indicates charging level:**
blue: 50%-100%
red: 10%-49%
red: 0-1%



The LED indicator

Yi Sport Camera App

- 5.) **Installation on mobile Yi Sport Camera Application**

System requirements of phone Android 4.1 and above or IOS 7 for installation on mobile phone. The QR code which is shown as a QR code which is from the right side. Follow the further instructions for completion of installation and successful interface of the camera with mobile.



- 6.) **Firmware Update**
Updating of the camera through the mobile application: first make sure that the camera is powered on and the application, after interfaces to the camera if there is a new version, phone will automatically request updating.

- 7.) **Return to factory installations**
If you have any problem, please press the button of power on and on button WIFI at the same time and keep them in current 2 seconds. After you will hear a sound signal, release these two buttons ---you have returned to factory installations, and your camera will automatically reboot.

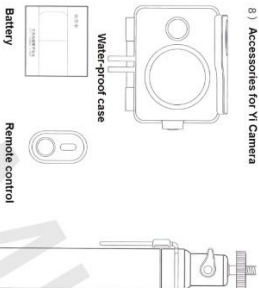
Presence of toxic and harmful substances

Name	Toxic and harmful substances or elements					
	Pb	Hg	Cd	Cr ⁶⁺	PBB	PFOS
Base of the camera	0	0	0	0	0	0
BATTERY	0	0	0	0	0	0
Cable	0	0	0	0	0	0
Circuit board	X	0	0	0	0	0

0 "toxic and harmful substances in all accessories of the given arrangement to occur in quantity below limiting value according to S1711463;2006 "Requirements of restriction availability of poisonous and harmful substances in the goods" -
 X "Maximum availability of toxic and harmful substances at least in one of the accessories of the given arrangement in the quantity exceeding limiting value, according to S1711463;2006 "Requirements of restriction availability of poisonous and harmful substances in the goods" -

Accessories for Yi Camera

- 8.) **Accessories for Yi Camera**



Warranty Conditions

- Into guarantee service enters:
- In current of 7 days from the moment of purchase if service center M will detect occurrence of a problem of an arrangement, you can return the goods or change it;
 - In current of 15 days from the moment of purchase if service center M will detect occurrence of a problem of an arrangement, you can change the goods or take advantage of free corrective maintenance;
 - In current of 12 months from the moment of purchase if service center M will detect occurrence of a problem of an arrangement, you can take advantage of free corrective maintenance.

The main specifications of Yi Camera

Lens: F2.8 DFOV: 155°
 Focal length: 2.73 ~55mm
 Video standard: The highest supported standard 1080P-60fps
 Sensor: Sony CMOS 16 - megapixel camera
 Weight: 72g (Camera + battery)
 Standards of storage: MICRO SDHC and Micro SDXC card
 Product Name: Yi Camera
 In: 1A 5V
 Standard consumption of energy: 2 W
 The operating standard: GB/T 28258-2012

XIAOMI.MI.COM

Warranty Conditions

- Conditions of guarantee repair:
- The list of circumstances (including, but not being limited) at which warranty service is impossible:
- Completion of maintenance service, failing the negligent attitude, abuse, hit of a liquid, accident if have torn or have covered a label and marks of an arrangement;
 - Validity of a guarantee has ended;
 - The damages which have arisen at force-major circumstances such as: impact of a lightning, a fire or flooding;

BIOGRAFI PENULIS



Penulis memiliki nama lengkap Muhammad Ichlasul Salik. Lahir di kota Bogor pada tanggal 28 Desember 1998. Selama proses perkuliahan di ITS penulis aktif berkontribusi dalam Tim ROBOTIKA ITS. Penulis merupakan salah satu anggota Tim ROBOTIKA ITS yaitu BAYUCARAKA ITS. Selama periode 2018-2019 penulis menjabat menjadi Ketua Umum Tim BAYUCARAKA ITS. Beberapa prestasi yang penulis dapatkan selama menjadi anggota Tim BAYUCARAKA ITS antara lain Juara 2 divisi *racing plane* pada KRTI 2017 dan Peringkat 5 dalam *Tubitak International UAV Turkey Competition* 2018.

Penulis dapat dihubungi melalui email: muhammadichlasulsalik@gmail.com