



TUGAS AKHIR - EE 184801

RANCANG BANGUN SWARM DRONE BERBASIS LEADER-FOLLOWER

Michael Jacob Ephariem Jehuda
NRP 0711174500072

Dosen Pembimbing
Ronny Mardiyanto S.T., M.T., Ph.D.
Astria Nur Irfansyah S.T., M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - EE 184801

RANCANG BANGUN SWARM DRONE BERBASIS LEADER-FOLLOWER

Michael Jacob Ephariem Jehuda
NRP 0711174500072

Dosen Pembimbing
Ronny Mardiyanto S.T.,M.T.,Ph.D.
Astria Nur Irfansyah S.T.,M.Eng.,Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - EE 184801

DESIGN SWARM DRONES BASED ON LEADER-FOLLOWER.

Michael Jacob Ephariem Jehuda
NRP 0711174500072

Supervisor

Ronny Mardiyanto S.T.,M.T.,Ph.D.
Astria Nur Irfansyah S.T.,M.Eng.,Ph.D.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

[Halaman ini sengaja dikosongkan]

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul

“Rancang Bangun Swarm Drone Berbasis Leader-Follower”

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, 3 Juni 2019



Michael Jacob Ephariem Jehuda
Nrp 07111745000072

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SWARM DRONE BERBASIS LEADER-FOLLOWER

TUGAS AKHIR

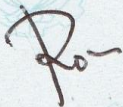
Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I



Ronny Mardiyanto, S.T., M.T., Ph.D.
NIP. 198101182003121003

Dosen Pembimbing II



Astria Nur Irfansyah, S.T., M.Eng., PH.D.
NIP: 198103252010121002



[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SWARM DRONE BERBASIS LEADER-FOLLOWER

Michael Jacob Ephariem Jehuda
07111745000072

Dosen Pembimbing I : Ronny Mardiyanto S.T.,M.T.,Ph.D.

Dosen Pembimbing II:Astria Nur Irfansyah S.T.,M.Eng.,Ph.D.

ABSTRAK

Swarm Drone adalah suatu teknologi kendaraan udara tak berawak yang tidak hanya terdiri dari satu kendaraan udara tak berawak tetapi terdapat lebih dari satu kendaraan udara tak berawak yang dapat bergerak dan berkomunikasi secara bersamaan yang pada umumnya modelnya sama dengan spesies hewan saat membentuk suatu informasi termasuk dimiliter. Perkembangan teknologi swarm drone telah mengalami kemajuan pesat karena teknologi banyak digunakan untuk kepentingan bermacam-macam contohnya teknologi swarm drone telah digunakan saat ajang asean game yang bertugas untuk menghibur para peserta asean game guna selain menghibur swarm drone juga telah dikembangkan untuk kepentingan pertahanan diberbagai dunia salah satunya di amerika. Teknologi ini digunakan untuk pengintaian lewat udara. Oleh karena itu dengan perkembangan teknologi yang semakin pesat saat ini maka dibuatlah suatu prototipe swarm drone pada tugas akhir ini yang dapat berkomunikasi antar drone dan dapat melakukan suatu pergerakan informasi sesuai yang diinginkan oleh pengguna. Pada tugas akhir ini quadcopter terdiri dari 3 pesawat yang satu pesawat memiliki peranan sebagai leader dan 2 follower.Hasil dari tugas akhir ini adalah drone leader dan foloower dapat mencapai tujuan yang diinginkan dengan kecepatan rata-rata untuk naik adalah 0.1483588 m/s dan kecepatan rata-rata untuk mundur adalah 0.1557605 m/s sedangkan untuk kecepatan rata-rata maju untuk drone dapat berkoordinasi mencapai suatu titik adalah 0.1406205 m/s dan kecepatan rata-rata mundur untuk drone dapat berkoordinasi mencapai suatu titik adalah 0.1431069 m/s

Kata Kunci: swarm drone, Komunikasi, quadcopter dan UAV

[Halaman ini sengaja dikosongkan]

DESIGN SWARM DRONES BASED ON LEADER FOLLOWER

Michael Jacob Ephariem Jehuda
0711154000104

Supervisor I: Ronny Mardiyanto S.T.,M.T.,Ph.D.

Supervisor II: Astria Nur Irfansyah S.T.,M.Eng.,Ph.D.

ABSTRACT

Swarm Drone is a technology of unmanned aerial vehicles that not only consists of one unmanned aerial vehicle but there are more than one unmanned aerial vehicle that can move and communicate simultaneously which generally the model is the same as animal species when forming an information including military. The development of swarm drone technology has undergone rapid progress because technology is widely used for various purposes, for example swarm drone technology has been used during the asean game which is tasked with entertaining asean game participants in addition to entertaining swarm drones that have also been developed for defense purposes in various worlds. in America. This technology is used for air surveillance. Therefore, with the rapid development of technology, a prototype swarm drone was created in this final project that can communicate between drones and can carry out information movements as desired by the user. In this final assignment, the quadcopter consists of 3 aircraft, one aircraft has a role as leader and 2 followers. The results of this final project are drone leaders and foloower can reach the desired destination with an average speed to rise is 0.1483588 m / s and average speed the data for backward is 0.1557605 m / s while for the advanced average speed for the drone it can coordinate reaching a point is 0.1406205 m / s and the average reverse speed for the drone can coordinate reaching a point is 0.1431069 m / s

Keywords: drone swarm, communication, quadcopter and UAV

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala syukur dan puji hanya bagi Tuhan Yesus Kristus, oleh karena anugerahnya yang melimpah, kemurahan dan kasih setianya akhirnya penulis dapat menyusun dan menyelesaikan buku tugas akhir dengan judul: **“RANCANG BANGUN SWARM DRONE BERBASIS LEADER-FOLLOWER”** maksud dan tujuan penulis menyelesaikan buku tugas akhir ini adalah untuk memenuhi persyaratan kelulusan Strata I bidang studi Elektronika, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Surabaya.

Penulis ingin mengucapkan terima kasih kepada kedua orang tua yang selalu memberikan dukungan dan doa yang tulus tiada henti maka penulis dapat menyelesaikan buku tugas akhir. Terima kasih kepada bapak Ronny Mardiyanto S.T.,M.T.,Ph.D. dan Astria Nur Irfansyah S.T.,M.Eng.,Ph.D. yang telah bersedia meluangkan waktunya untuk memberikan masukan, arahan dan membimbing penulis dalam pembuatan tugas akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada buku tugas akhir ini. buku ini memang masih jauh dari sempurna. Oleh karena itu penulis mengharapkan saran dan kritik yang membangun dari pembaca. Akhir kata, semoga buku tugas akhir ini dapat memberikan manfaat dalam pengembangan keilmuan dikemudian hari.

Surabaya, 3 Juni 2019

Penulis

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	ERROR!
BOOKMARK NOT DEFINED.	
LEMBAR PENGESAHAN ..	ERROR! BOOKMARK NOT DEFINED.
ABSTRAK	V
KATA PENGANTAR.....	IX
DAFTAR ISI.....	XI
DAFTAR GAMBAR.....	XV
DAFTAR TABEL	XIX
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	2
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Metodologi.....	2
1.6. Sistematika.....	4
1.7. Relevansi.....	4
BAB 2 DASAR TEORI & TINJAUAN PUSTAKA	5
2.1. Teori Dasar Quadcopter	5
2.2. Metode Swarm Drone	7
2.2.1. Centralized Coordination	7
2.2.2. Hierarchical Coordination	7
2.2.3. Conensus Coordination	8
2.2.4. Emergent Coordination	8
2.3. Flight Controller	8
2.3.1. Internal Sensor	9
2.3.2. Eksternal Sensor.....	10
2.3.2.1. <i>Global Positioning System</i>	10
2.4. Kontrol dan Komunikasi	11
2.4.1. Radio Kontrol.....	11

2.4.2.	Ground Control Station	13
2.4.3.	Mavlink Protocol	14
2.5.	Brushless DC Motor	15
2.6.	Electronic speed controller (ESC)	18
2.7.	Haversine Formula	19
2.8.	Kontroller PID	21
2.9.	Arduino.....	21
2.10.	ESP8266	22
2.11.	Lipo Battery	22
2.12.	Tinjauan Pustaka.....	24
2.12.1.	<i>Wireless Communications and Control for Swarms of Cellular-Connected UAVs</i>	24
2.12.2.	<i>Massive MIMO for Communications with Drone Swarms</i>	24
2.12.3.	<i>On the impact of inter-UAV communications interference in the 2.4 GHz band</i>	25
BAB 3	PERANCANGAN SISTEM	27
3.1.	Swarm Drone.....	27
3.2.	Drone Leader Follower.....	28
3.2.1.	Konfigurasi Posisi Swarm Leader Follower	28
3.3.	Rancang Bangun Mekanik Drone Leader Follower	30
3.4.	Rancang Bangun Elektronik	30
3.4.1.	ESC, Motor, dengan Flight Controller	31
3.4.2.	Receiver, Arduino dan Flight Controller	32
3.4.3.	Arduino, ESP8266 dan Flight Controller	33
3.4.4.	HMCL58831 dengan arduino	34
3.4.5.	Diagram Blok Elektronik Drone Leader	34
3.4.6.	Diagram Blok Elektronik Drone Follower	35
3.5.	Rancang Bangun Software	36
3.5.1.	Perancangan Konfigurasi Mission Planner	36
3.5.2.	Konfigurasi Firmware	37
3.5.3.	Kalibrasi Kompas	38
3.5.4.	Kalibrasi Acel dan Level	39
3.5.5.	Kalibrasi Radio Control	39
3.5.6.	Kalibrasi ESC	40

3.5.7.	Konfigurasi Failsafe	40
3.5.8.	Konfigurasi PID	41
3.5.9.	Konfigurasi Flight Modes	41
3.5.10.	Perancangan Segmen Program pada Arduino IDE.....	42
3.5.11.	Diagram Alur Pengambilan Data Gps dan Pergerakan Drone Leader.....	42
3.5.12.	Diagram Alur Pengambilan data GPS dan Pergerakan Drone Follower	43
3.5.13.	Pendefinisian Library Arduino pada Drone Leader & Follower.	44
3.5.14.	Pendefinisian Library NodeMcu pada Drone Leader & Follower.	44
3.5.15.	Segmen Program Pengaturan Awal Arduino Drone Leader dan Drone Follower.	45
3.5.16.	Segmen Program Pengaturan Awal NodeMCU Drone Leader dan Drone Follower.	46
3.5.17.	Segmen Program Pengambilan data GPS melalui Mavlink	47
3.5.18.	Segmen Program Penerimaan data GPS dari Drone Leader.	47
3.5.19.	Segmen Program Pengiriman data GPS dari Drone Leader ke Drone Follower	48
3.5.20.	Segmen Program Pergerakan Drone Leader dan Drone Follower menggunakan rumus haversine.	49
BAB 4	PENGUJIAN DAN ANALISA	53
4.1.	Pengujian Konsumsi Daya.....	53
4.2.	Pengujian Kecepatan Naik Turun berdasarkan Waktu.....	54
4.3.	Pengujian Kecepatan Gerak Maju drone leader	57
4.4.	Pengujian Kecepatan Gerak Mundur drone leader	58
4.5.	Pengujian Koneksi Router	60
4.6.	Pengujian Kecepatan Drone Leader dan Follower Naik	61
4.7.	Pengujian Kecepatan Drone Leader dan Follower Turun	63
4.8.	Pengujian Kecepatan Drone Leader dan Follower Maju	65
4.9.	Pengujian Kecepatan Drone Leader dan Follower Mundur ..	67

4.10. Pengujian Kecepatan Drone Leader dan Follower menuju titik point	69
BAB 5 KESIMPULAN DAN SARAN	73
5.1. Kesimpulan.....	73
5.2. Saran.....	73
DAFTAR PUSTAKA.....	75
LAMPIRAN	77
RIWAYAT HIDUP.....	109

DAFTAR GAMBAR

Gambar 1-1 Diagram Blok Sistem Leader dan Follower	3
Gambar 2-1 Quadcopter.....	5
Gambar 2-2 Thrust Gambar 2-3 Roll.....	6
Gambar 2-4 Pitch Gambar 2-5 Yaw	7
Gambar 2-6 Model Kontrol Swarm Metode Central.....	7
Gambar 2-7 Model Kontrol Swarm Metode Hierarchi	8
Gambar 2-8 Model Kontrol Swarm Metode Conseus	8
Gambar 2-9 Model Kontrol Swarm Metode Emergent.....	8
Gambar 2-10 Prinsip Kerja <i>Inertia Measurement Unit</i>	9
Gambar 2-11 GPS	11
Gambar 2-12 Perbedaan Sinyal dengan PWM.....	12
Gambar 2-13 <i>Radio Control</i>	13
Gambar 2-14 Tampilan Utama Mission Planner.....	14
Gambar 2-15 Struktur Pesan MAVLINK (“ <i>Micro Air Vehicle Communication Protocol</i> ”).....	15
Gambar 2-16 Prinsip BLDC Motor.....	15
Gambar 2-17 BLDC Motor	16
Gambar 2-18 Efek Grafitasi dan Pergerakan pada Quadcopter	17
Gambar 2-19 ESC	18
Gambar 2-20 Kontruksi pada ESC.....	19
Gambar 2-21 Sudut Tujuan Yang Dituju	20
Gambar 2-22 Pin Arduino Nano	22
Gambar 2-23 ESP8266.....	22
Gambar 2-24 Lipo Baterai	24
Gambar 3-1 Swarm Drone Leader-Follower.....	27
Gambar 3-2 Sistem Drone Leader Follower	28
Gambar 3-3 Konfigurasi Posisi Drone Leader Follower.....	29
Gambar 3-4 Frame Drone Leader-Follower.....	30
Gambar 3-5 Rancang Bangun Mekanik Drone Leader dan Follower ...	30
Gambar 3-6 Pengkabelan ESC, Motor dengan Flight Controller.....	31
Gambar 3-7 Wiring Diagram Receiver, Arduino Nano dan Minipix	32
Gambar 3-8 Wiring Diagram ESP, Arduino dan Minipix.....	33
Gambar 3-9 Wiring Diagram HMC5833L dan Arduino.....	34
Gambar 3-10 Diagram Blok Elektronik Drone Leader	35

Gambar 3-11 Diagram Blok Elektronik Drone Follower	36
Gambar 3-12 Menu Utama Mission Planner	37
Gambar 3-13 Firmware pada Mission Planner.	37
Gambar 3-14 Pemilihan Frame pada Mission Planner	38
Gambar 3-15 Kalibrasi Kompas pada Mission Planner	38
Gambar 3-16 Kalibrasi Accel dan Level pada Mission Planner	39
Gambar 3-17 Kalibrasi Radio Control pada Mission Planner	39
Gambar 3-18 kalibrasi ESC	40
Gambar 3-19 Konfigurasi Failsafe pada Mission Planner	40
Gambar 3-20 Konfigurasi PID menggunakan Mission Planner.....	41
Gambar 3-21 Konfigurasi Flight Mode pada mission planner.....	41
Gambar 3-22 Flowchart Diagram Alur Drone Leader	42
Gambar 3-23 Flowchart Diagram Alur Drone Follower.....	43
Gambar 3-24 Pendefinisian Library pada Arduino	44
Gambar 3-25 Pendefinisian Library Pada NodeMCU	45
Gambar 3-26 Segmen Program Pengaturan Awal pada Arduino	45
Gambar 3-27 Segmen Program Pengaturan Awal NodeMCU pada Drone Follower.....	46
Gambar 3-28 Segmen Program Pengambilan Data GPS.	47
Gambar 3-29 Segmen Program Request Data GPS	47
Gambar 3-30 Segmen Program Penerimaan Data pada Drone Follower	48
Gambar 3-31 Segmen Program Pengiriman Data pada Leader ke Follower.....	49
Gambar 3-32 Program Penentuan Jarak Posisi menggunakan Haversine Formula.....	50
Gambar 3-33 Program Penentuan Sudut menggunakan Haversine Formula.....	50
Gambar 3-34 Program Penentuan Titik Posisi x dan Titik Posisi y yang Terdapat pada Drone Leader dan Drone Follower.....	51
Gambar 3-35 Program Pergerakan Drone Leader dan Drone Follower.	52
Gambar 4-1 Pengujian Konsumsi Daya Baterai Lipo.....	53
Gambar 4-2 Pengujian Kecepatan Naik dan Turun Quadcopter.....	55
Gambar 4-3 Grafik Kecepatan naik drone leader	56
Gambar 4-4 Grafik Kecepatan Turun Drone Leader	57
Gambar 4-5 Pengujian Kecepatan Maju Quadcopter.....	57

Gambar 4-6 Grafik Kecepatan Maju Drone Leader	58
Gambar 4-7 Pengujian Kecepatan Mundur quadcopter	59
Gambar 4-8 Grafik Kecepatan Mundur Drone Leader.....	60
Gambar 4-9 Pengujian Koneksi Router	61
Gambar 4-10 Pengujian Drone Leader dan Follower Saat Naik	62
Gambar 4-11 Grafik Kecepatan Naik Drone Leader dan Drone follower	63
Gambar 4-12 Pengujian Drone Leader dan Follower Saat Turun	64
Gambar 4-13 Grafik Kecepatan Turun Drone Leader dan Drone follower	65
Gambar 4-14 Pengujian Pengiriman Saat Drone Leader dan Follower maju	66
Gambar 4-15 Grafik Kecepatan Maju Drone Leader dan Drone follower	67
Gambar 4-16 Pengujian drone leader dan drone follower saat mundur	68
Gambar 4-17 Grafik Kecepatan Mundur Drone Leader dan Drone Follower	69
Gambar 4-18 Pengujian drone leader dan follower menuju titik point .	70
Gambar 4-19 Grafik Kecepatan Menuju Titik Waypoint Drone Leader dan Drone Follower	71

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4-1 Konsumsi Daya.....	54
Tabel 4-2 Pengujian Kecepatan Naik Berdasarkan Waktu	55
Tabel 4-3 Pengujian Kecepatan Turun Berdasarkan Waktu	56
Tabel 4-4 Pengujian Kecepatan Maju Berdasarkan Waktu.....	58
Tabel 4-5 Pengujian Kecepatan Mundur Berdasarkan Waktu	59
Tabel 4-6 Pengujian Koneksi Router	61
Tabel 4-7 Pengujian Pengiriman antara Drone Leader dan Follower Saat Naik Berdasarkan Waktu	62
Tabel 4-8 Pengujian Pengiriman antara Drone Leader dan Follower Saat Turun Berdasarkan Waktu	64
Tabel 4-9 Pengujian Pengiriman antara Drone Leader dan Follower Saat Maju Berdasarkan Waktu.....	66
Tabel 4-10 Pengujian Pengiriman antara Drone Leader dan Follower Saat Mundur Berdasarkan Waktu	68
Tabel 4-11 Pengujian Pengiriman antara Drone Leader dan Follower Saat Menuju Titik Point Berdasarkan Waktu	70

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi pada kendaraan udara tak berawak saat ini telah menjadi suatu bagian menarik pada banyaknya peneliti yang berkeinginan untuk mengembangkannya. Kendaraan udara tak berawak ini memiliki banyak kegunaannya dan dapat digunakan untuk kepentingan penduduk maupun kepentingan militer dan pertahanan. Peranan para peneliti dari sistem ini adalah berperan dari desain efisiensi pesawat hingga pengembangan kontrol elektronik yang digunakan untuk sistem otomatisnya[1].

Akhir-akhir ini ditemukannya sebuah pengembangan penelitian pada teknologi kendaraan udara tak berawak yang digunakan untuk berkoordinasi dengan kendaraan udara tak berawak lainnya sehingga pada bidang ini penerapannya masih sangatlah belum luas dan masih memerlukan pendekatan baru untuk dijadikan berbagai macam aplikasi lainnya. Kendaraan udara tak berawak yang dapat berkoordinasi dengan kendaraan udara lainnya disebut juga dengan swarm drone.

Swarm drone saat ini telah digunakan untuk kemajuan teknologi kendaraan tak berawak yang menganut sistem teknologi industri 4.0. Maka dari itu swarm drone juga dituntut untuk menggunakan suatu teknologi yang canggih agar mampu berkoordinasi antar satu drone dengan drone lainnya. Teknologi industri 4.0 yang digunakan untuk koordinasi swarm drone ini ialah menggunakan media internet. Swarm drone dapat digunakan untuk menyelesaikan sebuah misi yang tidak dapat dilakukan oleh satu kendaraan udara tak berawak karena membutuhkan waktu yang lebih lama untuk menyelesaikan sebuah misi[2]. Selain menyelesaikan misi yang lebih cepat swarm drone juga dapat digunakan untuk aplikasi yang bermacam-macam salah satu contohnya ialah digunakan untuk media entertainment yang telah digunakan saat ajang *ASEAN-games* 2018. Dari uraian tersebut maka penulis merancang sebuah prototipe swarm drone berbasis *leader-follower* yang digunakan untuk tugas akhir ini.

Perancangan pada swarm drone terdiri dari 3 pesawat berjenis quadcopter dimana 1 pesawat quadcopter ditugaskan sebagai *leader* dan

2 quadcopter lainnya ditugaskan sebagai *follower*. tugas dari pesawat *leader* ialah memberikan perintah kepada 2 *follower* lainnya berupa perintah posisi koordinat dan perintah untuk menjalankan sistem misi quadcopternya. data posisi yang didapat dari *follower* nantinya akan dihitung dengan jarak posisi quadcopter *follower* yang didapatkan melalui GPS (*Global Positioning System*). Dengan demikian quadcopter pada *follower* dapat mengeksekusi perintah pergerakan dari drone yang berperan sebagai *leader*.

1.2. Perumusan Masalah

Masalah yang akan dibahas pada tugas akhir ini adalah sebagai berikut:

1. Bagaimana *leader* mengirim data posisi GPS ke *follower*.
2. Bagaimana *leader* menentukan data posisi GPS yang ditentukan sesuai pengguna.
3. Bagaimana *follower* melaksanakan instruksi perintah yang diberikan oleh *leader*.

1.3. Tujuan

Tujuan yang ingin dicapai pada tugas akhir ini adalah sebagai berikut:

1. Drone sebagai *leader* dapat berkoordinasi dengan *follower*.
2. Drone sebagai *leader* dapat berada diposisi yang ditentukan oleh pengguna.
3. *Follower* dapat melaksanakan perintah yang telah diberikan oleh *leader*.

1.4. Batasan Masalah

Batasan Masalah dari tugas akhir ini adalah sebagai berikut:

1. Cakupan luasan wilayah yang terbatas hingga 10 meter.
2. Pelaksanaan pergerakan drone *leader* dan *follower* hanya dapat dilakukan pada luar ruangan dan tidak ada gedung
3. Terbatasnya kemampuan baterai sehingga drone *leader* dan *follower* hanya mampu terbang selama 10 menit.

1.5. Metodologi

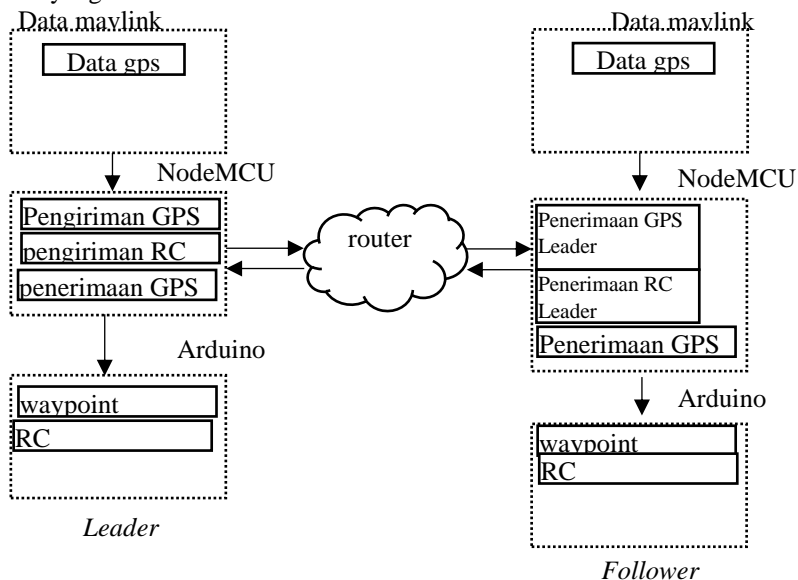
Metodologi dalam penelitian ini meliputi beberapa bagian diantaranya desain, tipe, pendekatan, teknik, dan proses penelitian.

1) Studi Literatur

Studi literatur didapat dari pengumpulan berbagai macam-macam informasi melalui dari jurnal,buku mengenai referensi tentang komunikasi antar swarm drone dan didapatkan referensi dari sumber internet web asal untuk referensi tentang nodemcu dan arduino.

2) Perancangan dan Pembuatan Alat

Setelah didapatkan sumber literatur yang sesuai lalu dilakukanlah perancangan alat pada sistem drone *leader* maupun *follower*.perancangan dan pembuatan alat dilakukan dimana masing-masing drone *leader* maupun drone *follower* terdapat node mcu dan arduino guna untuk pengiriman dan penerimaan data GPS agar sesuai dengan pergerakan yang diinginkan. Dibawah ini merupakan perancangan dan pembuatan alat yang dilakukan.



Gambar 1-1 Diagram Blok Sistem Leader dan Follower

3) Pengambilan Data

Setelah perancangan dan pembuatan selesai dilakukan hal yang dilakukan selanjutnya ialah pengambilan data pada drone *leader* dan drone *follower*. data yang akan diambil diantaranya adalah data GPS, dan data kemampuan jarak koneksi wifi.

4) Kesimpulan dan Saran

1.6. Sistematika

Penulisan laporan tugas akhir ini mengikuti sistematika sebagai berikut.

BAB 1 Pendahuluan

Pada Bab ini akan dijelaskan mengenai latar belakang, perumusan masalah, tujuan, batasan masalah, metodologi, sistematika, dan relevansi guna untuk penyelesaian tugas akhir ini.

BAB 2 Dasar Teori

Pada bab ini akan dijelaskan mengenai teori dasar dan referensi-referensi acuan dalam penyelesaian tugas akhir ini.

BAB 3 Perancangan Sistem

Dalam bab ini akan dilakukan perancangan sistem pada drone leader maupu drone follower yang terdiri dari perancangan hardware, perancangan elektronik dan perancangan software.

BAB 4 Pengujian dan Analisa

Dalam bab ini akan dilakukan pengujian dan analisa yang didapat sesuai dengan hasil yang telah dilakukan.

BAB 5 Kesimpulan dan Saran

Dalam ini bab ini berisi kesimpulan yang telah didapat melalui Analisa serta akan diperoleh saran untuk penelitian selanjutnya.

1.7. Relevansi

Hasil dari tugas akhir ini diharapkan dapat memberikan manfaat terhadap khalayak umum terkait,

1. Metode pengiriman data secara *wireless* dan *real-time*.
2. Metode estimasi panjang jarak yang ditempuh menggunakan GPS.

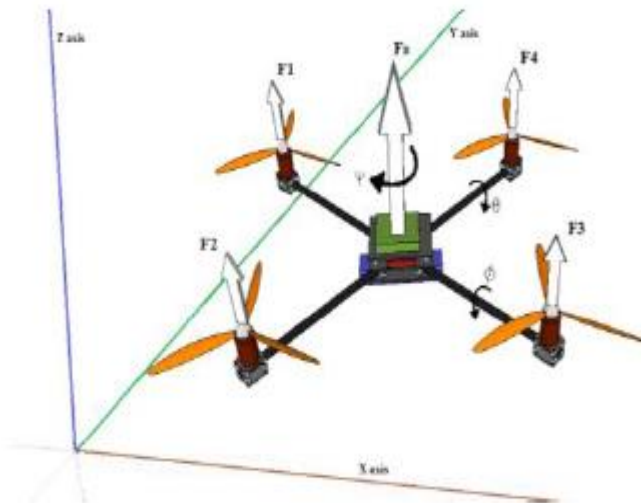
BAB 2

Dasar Teori & Tinjauan Pustaka

Pada bab ini akan dibahas mengenai dasar teori dan tinjauan pustaka yang diperoleh dari beberapa paper penelitian yang pernah dilakukan sebelumnya. Bab ini berisi tentang teori dasar quadcopter, metode swarm drone, flight controller, kontrol, komunikasi, penggunaan rumus yang digunakan untuk menemukan jarak sudut dan tinjauan pustaka yang pernah dilakukan oleh beberapa penelitian sebelumnya

2.1. Teori Dasar Quadcopter

Quadcopter merupakan suatu UAV dengan empat baling-baling yang masing-masingnya dikendalikan oleh motor dan menggunakan pengendali kecepatan elektronik (*ESC*). Sensor yang digunakan untuk mengatur pengukuran sudut UAV pada sumbu X, Y dan Z ialah *Accelerometers*, Selain itu Fungsi *Accelerometers* ini ialah untuk penstabilan pada *quadcopter* dengan cara penyesuaian RPM[3].

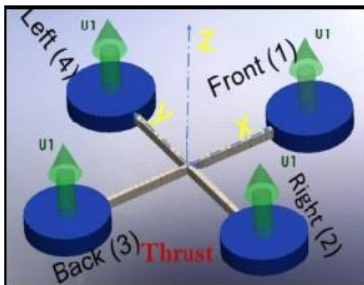


Gambar 2-1 Quadcopter

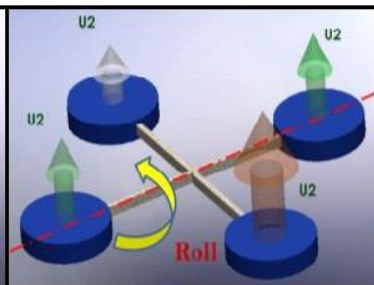
Dengan menggunakan prinsip ini maka kecepatannya-pun dapat disesuaikan (RPM sebagai fungsi dari tegangan yang disediakan untuk motor) masing-masing motor pada quadcopter digunakan untuk

memanipulasi yaw, tilt, dan roll Quadcopter dengan benar. Kemiringan dan putaran dapat dikontrol dengan cara mengubah kecepatan motor yang sesuai, sementara kontrol yaw membutuhkan keseimbangan yang sama dan putaran yang halus sehingga keempat fungsi motor dapat mengubah gaya momen yang diterapkan pada quad. Dibawah ini merupakan cara pergerakan suatu Quadrotor dalam empat kondisi yang berbeda:

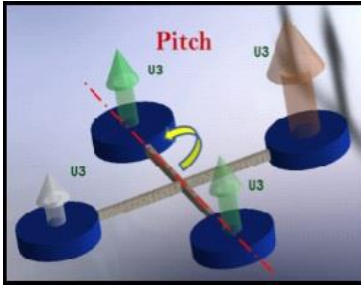
1. Gerakan ke atas z-arah: (gambar 2.2) sehubungan dengan tubuh. Gerakan ini disediakan oleh gaya dorong yang dihasilkan oleh keempat motor yang berputar dengan kecepatan sudut yang sama.
2. Gerakan roll Φ : (gambar 2.3) gerakan ini diperoleh dari perbedaan gaya antara gaya dorong yang dihasilkan oleh baling-baling kanan dan kiri.
3. Gerakan pitch θ : (gambar 2.4) gerakan ini diperoleh dari perbedaan gaya antara gaya dorong yang dihasilkan oleh baling-baling depan dan belakang.
4. Gerakan Yaw ψ : (gambar 2.5) gerakan ini diperoleh dengan mengendalikan dua set flap yang ditempatkan di bawah baling-baling samping untuk mengarahkan kembali komponen kecil dari gaya dorong normal-ke-tubuh pada bidang horizontal. Komponen gaya bekerja dalam arah yang berlawanan yang menciptakan pasangan-momen tentang sumbu vertikal tubuh



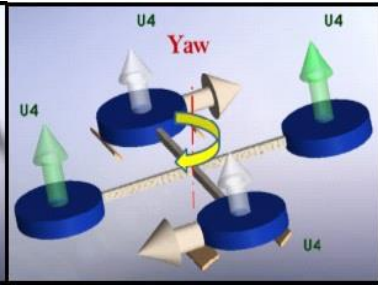
Gambar 2-2 Thrust



Gambar 2-3 Roll



Gambar 2-4 Pitch



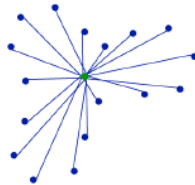
Gambar 2-5 Yaw

2.2. Metode Swarm Drone

dengan perkembangan teknologi robot saat ini, Pada sistem robot terdapat suatu prospek aplikasi yang luas dalam penelitian teoritis dan aplikasi rekayasa. Kontrol koordinasi formasi merupakan salah satu area penelitian paling penting yang digunakan pada sistem swarm robot untuk berbagai aplikasi (mis., Produksi industri, aerospace, dan urusan militer). ada beberapa penelitian strategi koodinasi yang dapat dimanfaatkan untuk pada robot[4].

2.2.1. Centralized Coordination

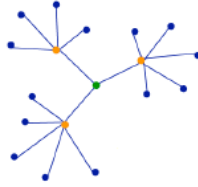
Centralized Coordination merupakan Suatu elemen swarm yang dapat berkomunikasi dengan perencana terpusat yang bertugas untuk mengkoordinasikan semua tugas.



Gambar 2-6 Model Kontrol Swarm Metode Central

2.2.2. Hierarchical Coordination

Hierarchical Coordination merupakan suatu elemen swarm yang dikendalikan oleh agen level "squad", yang pada gilirannya dikendalikan oleh pengendali level yang lebih tinggi.



Gambar 2-7 Model Kontrol Swarm Metode Hierarchi

2.2.3. Conensus Coordination

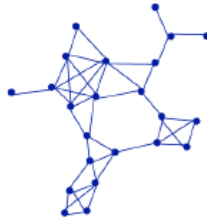
Conensus Coordination merupakan suatu elemen swarm yang dapat berkomunikasi satu sama lain dan menggunakan "voting" atau metode berbasis lelang untuk bertemu pada solusi.



Gambar 2-8 Model Kontrol Swarm Metode Conesus

2.2.4. Emergent Coordination

Emergent Coordination merupakan suatu elemen swarm yang dapat berkoordinasi muncul secara alami oleh elemen-elemen kawanannya individu yang bereaksi satu sama lain, seperti pada kawanannya hewan.



Gambar 2-9 Model Kontrol Swarm Metode Emergent

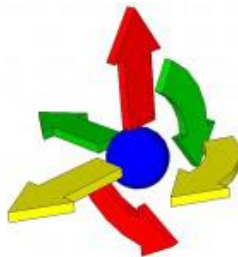
2.3. Flight Controller

Flight Controller merupakan suatu *controller* yang bertanggung jawab atas sebagian besar semua tindakan dan perilaku UAV. pada tugas akhir ini *flight controller* yang digunakan adalah Mini Pixhawk. Mini Pixhawk autopilot adalah perangkat keras yang bersifat *open source* dan

independen yang bertujuan untuk menyediakan suatu perangkat keras *controller* yang canggih bagi akademisi, penggemar dan industri dengan biaya rendah dan mempunyai berbagai macam fungsi. Pada pixhawk terdapat beberapa sensor yang terletak pada internal maupun eksternal[5].

2.3.1. Internal Sensor

Pada pixhawk terdapat sensor internal yang disebut *Inertia Measurement Unit* (IMU). Papan IMU Pixhawk memiliki giroskop 3-sumbu dan akselerometer 3-sumbu. Fungsi giroskop adalah untuk mengukur sudut rotasi UAV, sedangkan fungsi akselerometer adalah untuk mengukur akselerasi linear. Papan IMU bertanggung jawab untuk menjaga kestabilan UAV di udara agar tetap stabil. Cara kerjanya adalah dengan cara membaca data sensor dan mengirimkan sinyal pada ESC agar dapat mengubah kecepatan rotor sehingga mencapai keseimbangan horizontal pada sumbu longitudinal dan lateral. Untuk mengatur parameter IMU default, maka sensor-sensor ini perlu adanya kalibrasi awal. Kalibrasi dilakukan ketika UAV berdiri diam di tanah dan dilakukan pada saat sebelum terbang. Selain itu UAV perlu adanya sensor ketinggian sehingga dapat mempertahankan ketinggiannya. Sensor ketinggian yang terdapat pada internal pixhawk adalah barometer. Tugas barometer adalah untuk menganalisis perubahan tekanan atmosfer. Kombinasi dari pembacaan IMU dan barometer dapat menghasilkan penerbangan hover yang stabil pada lingkungan yang tidak berangin. Perhitungan heading uav dilakukan dengan menggunakan kompas atau magnetometer. Dengan menggunakan medan magnet bumi, maka kompas dapat menentukan arah mana yang ingin dituju oleh UAV[6].



Gambar 2-10 Prinsip Kerja *Inertia Measurement Unit*

2.3.2. Eksternal Sensor

Fungsi adanya Sensor eksternal adalah sebagai penambahan fitur yang bermanfaat pada Pixhawk sehingga dapat meningkatkan kualitas penerbangan UAV dan dapat bergerak mencapai tujuan tertentu. Sensor-sensor ini tidak tertanam di papan utama Pixhawk dan oleh karena itu maka terdapat tambahan koneksi yang berada diluar.

2.3.2.1. *Global Positioning System*

GPS atau *Global Positioning System* adalah instrumen yang digunakan untuk menentukan letak posisi suatu sistem. Ini didasarkan pada konstelasi satelit, yang mengorbit Bumi pada jarak 20; 000 km. Untuk memiliki pengukuran posisi pesawat yang relatif akurat, modul penerima harus melihat beberapa satelit. Ini beroperasi berdasarkan dua potong informasi. Penting untuk mengetahui posisi satelit, tetapi juga jarak antara masing-masing satelit dan penerima. Posisi satelit diperoleh dari almanak, yang terdiri dari data konstelasi satelit generik. Dari tabel waktu ini, penerima dapat menentukan satelit mana yang terlihat dan dapat digunakan. Almanak disiarkan secara permanen oleh semua satelit. Selain itu, penerima juga menerima informasi ephemeris, yang berkaitan dengan orbit satelit yang diamati. Kisaran antara penerima dan satelit dihitung dengan mengukur waktu yang dibutuhkan oleh sinyal radio untuk melakukan perjalanan dari satelit ke penerima. Jarak adalah produk dari kecepatan, yang merupakan kecepatan cahaya, dan waktu itu. Secara teoritis, untuk menerapkan proses triangulasi, tiga satelit harus cukup. Namun, kesalahannya akan signifikan karena penerima tidak memiliki kristal yang cukup akurat untuk mengukur waktu. Akibatnya, ketika semua pengukuran benar, keempatnya bola pada Gambar 2.11 merupakan contoh modul gps yang akan dipakai pada tugas akhir. namun demikian, penggunaan satelit tambahan sering diperlukan untuk meningkatkan pengukuran karena berbagai sumber kesalahan dapat mengubah hasil yang diharapkan. Sumber-sumber ini termasuk yang berikut ini:

- Jam satelit
- Jam penerima
- Gangguan GPS
- Kesalahan atmosfer
- Kesalahan multipath



Gambar 2-11 GPS

Dua yang pertama memiliki efek paling signifikan. Kesalahan multipath sebagian besar terjadi di dekat bangunan atau pegunungan di mana ombaknya dapat dipantulkan. Namun, klub aeromodelisme yang digunakan untuk menguji terbang pesawat terletak di lahan pertanian dan fenomena ini diabaikan. Hal yang sama berlaku untuk kesalahan atmosfer, yang mungkin disebabkan oleh cuaca buruk seperti awan atau hujan. Namun, tidak mungkin menerbangkan pesawat saat ini ketika ada terlalu banyak angin atau saat hujan. Gangguan GPS disebabkan oleh gangguan pada perangkat lain, yang seringkali dilarang oleh hukum. Mempertimbangkan kesalahan ini, dalam praktiknya, lebih dari enam satelit diperlukan untuk mendapatkan yang benar estimasi posisi. Selama penerbangan, ini tidak pernah menjadi masalah karena lebih dari sembilan satelit selalu tersedia (antara sembilan dan lima belas satelit) [7].

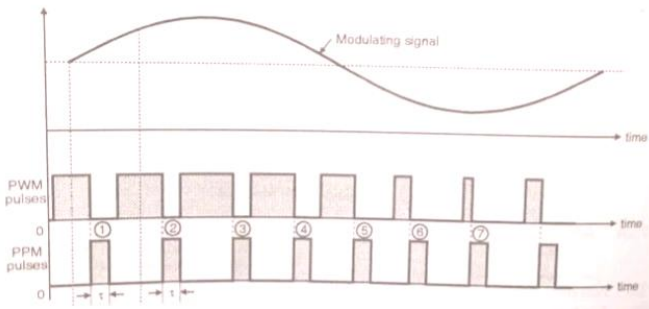
2.4. Kontrol dan Komunikasi

Untuk mengontrol UAV, perlu adanya komponen perantara antara UAV dan pengguna percobaan. Komponen tersebut sangat penting untuk uji coba karena mereka dapat menentukan perilaku UAV.

2.4.1. Radio Kontrol

Mengemudikan UAV bisa menjadi tantangan besar bagi pilot pemula, karena UAV dapat terbang dalam berbagai arah ruang 3D: kiri dan kanan, depan dan belakang, atas dan bawah. Cara termudah dan

tercepat untuk mulai mengendalikan UAV adalah menggunakan pengontrol radio. Penerima radio pada UAV diperlukan untuk menerima data frekuensi radio (RF). Data ini ditransmisikan oleh pemancar radio yang dikendalikan oleh pengguna sehingga dapat mengontrol sikap UAV. Pengguna memiliki kontrol manual terhadap perilaku Pitch, Roll, Yaw, dan Throttle UAV. Masing-masing perilaku ini dikaitkan dengan saluran frekuensi radio. Untuk mulai mengendalikan UAV, diperlukan pemancar dan penerima radio minimal 4 channel. Saluran lain digunakan untuk memulai misi, mengubah mode penerbangan atau untuk mengontrol servos. Kontroler RC yang digunakan dalam tugas akhir ini bekerja pada pita frekuensi 2.4GHz dan menggunakan *Pulse-Position Modulation* (PPM) untuk mengirim sinyalnya. *Pulse-Position Modulation* dilakukan sesuai dengan sinyal modulasi lebar pulsa. Setiap jejak sinyal yang termodulasi lebar pulsa menjadi titik awal untuk pulsa dalam sinyal PPM. Oleh karena itu, posisi pulsa ini sebanding dengan lebar pulsa PWM. Tetapi kelemahan utama dari teknik modulasi PPM adalah, sinkronisasi antara pemancar dan penerima harus diperlukan.



Gambar 2-12 Perbedaan Sinyal dengan PWM

Ada dua mode konfigurasi RC: Mode 1 dan Mode 2. Kontroler Mode 1 RC memiliki kontrol Roll dan Pitch di tongkat kiri controller, dan kontrol Throttle dan Yaw di stick kanan controller. Pada pengontrol Mode 2 RC, stik kiri mengontrol Yaw dan Throttle, dan stik kiri, Pitch and Roll. Penggunaan pengontrol RC hanya memungkinkan kontrol UAV di Line of Sight (LoS) pengguna. Gambar 2.13 menunjukkan contoh pengendali Mode 2 RC, dan arah mana yang harus dipindahkan pengguna

untuk menambah atau mengurangi perilaku UAV yang dijelaskan yang diketahui[8].



Gambar 2-13 Radio Control

2.4.2. Ground Control Station

GCS adalah komponen yang paling berguna untuk mengendalikan, berkomunikasi, dan memantau UAV. Ini memungkinkan pengguna untuk mengontrol dan berkomunikasi dengan UAV. GCS dapat dipakai untuk memperbarui firmware pengontrol penerbangan, mengubah parameter yang telah ditentukan sebelumnya, merencanakan misi yang dipandu titik jalan, menganalisis data telemetri, mengkalibrasi sensor internal dan eksternal, mengkalibrasi saluran pengontrol radio untuk batas tongkat minimum dan maksimum, memeriksa sensor untuk kegagalan atau perilaku buruk, memeriksa status baterai dan untuk mengirim perintah MAVLink, yang ditafsirkan oleh Pixhawk. selain itu dapat digunakan untuk status UAV dalam penerbangan, seperti pemantauan dan kontrol posisi waktu nyata. dalam tugas akhir, perangkat lunak yang digunakan adalah Mission planner. Mission Planner adalah stasiun kontrol tanah sumber terbuka berfitur lengkap dengan dukungan komunitas besar. Mission planner dibuat oleh Michael Osborne untuk autopilot open-source ArduPilot dan Hanya tersedia untuk sistem operasi Windows. Mission Planner dilisensikan berdasarkan ketentuan GNU General Public License versi 3[9].



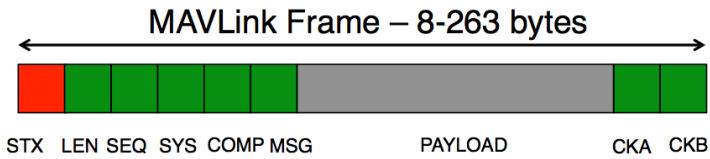
Gambar 2-14 Tampilan Utama Mission Planner

Dalam tampilan utama Gambar 2.14 pada angka 1 (sisi kiri), berfungsi untuk membaca informasi sensor IMU, seperti kecenderungan dan heading UAV, kecepatan dan ketinggian vertikal; Perbaikan GPS, tingkat baterai, kekuatan sinyal komunikasi, mode penerbangan saat ini dan keadaan rotor UAV.

Dalam tampilan utama Gambar 2.14 angka 2 (sisi kanan) ada peta dua dimensi (disediakan oleh Google Maps) di mana berfungsi untuk melihat posisi GPS dan jumlah satelit UAV saat ini, titik arah yang diprogram sebelumnya dan memantau secara real time penerbangan UAV.

2.4.3.Mavlink Protocol

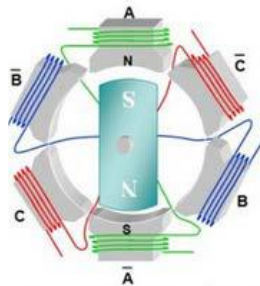
MAVLink adalah protokol pesan khusus header yang menggunakan grup pesan untuk mengirimkan data antara UAV dan GCS. Ini dirancang agar dapat diandalkan, cepat dan aman terhadap kesalahan transmisi. Mavlink dirilis pada 2009 oleh Lorenz Meier di bawah lisensi LGPL (Murilhas, 2015; Scherer, 2015). Setiap pesan dienkripsi-byte dengan konten terkait sensor, yang dapat diinterpretasikan oleh Ground Station Control atau oleh Raspberry Pi, yang mana berfungsi sebagai perantara pesan antara Pixhawk dan GCS. Perintah seperti tinggal landas, naikkan atau turunkan ketinggian (kenaikan atau penurunan throttle, masing-masing) dikirim oleh GCS dan ditafsirkan oleh Pixhawk untuk melakukan tindakan yang diinginkan[10].



Gambar 2-15 Struktur Pesan MAVLINK (“Micro Air Vehicle Communication Protocol”)

2.5. Brushless DC Motor

Motor Brushless DC motor (BLDC) adalah jenis motor yang paling banyak digunakan di quadcopter. BLDC adalah motor dengan magnet permanen, yang digerakkan oleh arus searah (DC) listrik. sistem pergantian mekanis diatur oleh rotor magnet permanen dan stator dengan urutan kumparan. Magnet permanen berputar dan konduktor pengangkut arus diperbaiki. Kumparan armature diaktifkan oleh transistor pada posisi rotor yang benar sehingga medan dinamo berada di ruang quadrature dengan kutub medan rotor. Rotary encoders (sensor Hall) digunakan untuk mengatur posisi rotor dan diposisikan di sekitar stator. Umpan balik posisi rotor dari sensor membantu untuk menentukan apakah beralih arus jangkar. Pergantian elektronik dilakukan dengan cara menghilangkan pengaturan komutator dan menyikat motor DC klasik. Dengan pergantian ini maka dapat tercapai operasi yang lebih handal dan lebih tenang. Ada berbagai konfigurasi fisik, yang tergantung pada gulungan stator. Gulungan dapat dikonfigurasi pada motor tunggal-fase, dua-fase, tiga-fase. Di sisi lain, motor BLDC yang paling banyak digunakan adalah tiga fase

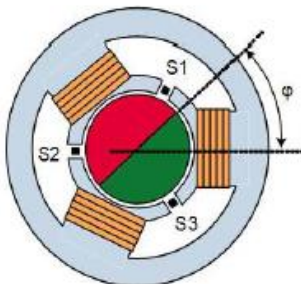


Gambar 2-16 Prinsip BLDC Motor

Pemilihan motor tergantung pada dorongan yang dibutuhkan untuk membuat quadcopter ke udara. Kebutuhan ini dipengaruhi oleh berat total dan ukuran bingkai quadcopter. Nilai penting yang digunakan untuk motor adalah *Trust to Weight Ratio* (TWR), yang dapat dinyatakan dengan persamaan:

$$TWR = T / W \quad (1)$$

Di mana T adalah Trust dan W mewakili Weight. Untuk melepas quadcopter, TWR harus lebih tinggi dari 1. Namun, untuk mempertahankan penerbangan stabil adalah nilai sama dengan 1 cukup. Kemiringan quadcopter menyebabkan vektor trust memiliki komponen vertikal dan horizontal. Kecenderungan ini membutuhkan TWR setidaknya 1,3 untuk sudut maksimum (sekitar 40 derajat dalam sumbu x, y) . Secara umum, quadcopter harus memberikan dorongan dua kali lebih banyak daripada berat drone. Jika daya dorong yang diberikan oleh motor kecil, performa terbang akan buruk. Jika nilai TWR yang diperlukan adalah 2 dan ada 4 motor, maka daya dorong yang harus dimiliki masing-masing motor harus sama dengan setengah dari berat quadcopter . Ukuran motor BLDC ditandai dengan 4 digit angka (pola XXYY). Di mana XX mewakili lebar stator dan YY mewakili berat stator dalam milimeter. Stator yang lebih tinggi dapat memberikan daya lebih besar pada RPM yang lebih tinggi. Di sisi lain, stator yang lebih luas dapat memberikan torsi lebih banyak pada RPM yang lebih.

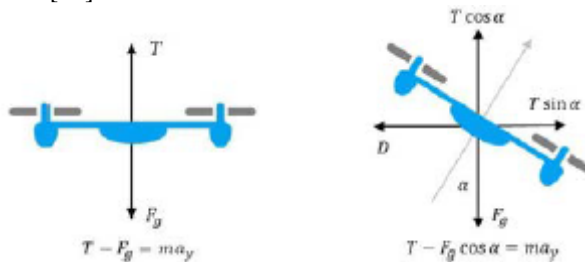


Gambar 2-17 BLDC Motor

Nilai kedua yang digunakan dalam motor BLDC adalah spesifikasi motor kV. Angka ini adalah jumlah putaran per menit yang akan dihidupkan motor ketika satu volt diterapkan tanpa beban pada motor. Jika baling-baling terpasang ke motor, akan ada pengurangan dalam jumlah RPM karena inersia tambahan dari baling-baling dan hambatan udara. Motor dengan baling-baling putaran kV lebih tinggi lebih cepat; Namun, motor dengan kV lebih rendah menghasilkan torsi yang lebih tinggi. Yang mengarah pada kombinasi motor kV rendah dan baling-baling yang lebih besar atau motor kV tinggi dan baling-baling yang lebih kecil.

Efisiensi motor listrik adalah rasio antara output daya mekanik dan input daya listrik. Output daya mekanik ditentukan oleh torsi dan kecepatan yang diperlukan dan daya listrik ditentukan oleh tegangan dan pasokan arus. Efisiensi motor tidak konstan di atas profil dorong. Untuk mencapai daya dorong yang lebih tinggi, motor harus menarik lebih arus. Kombinasi yang baik dari motor drone adalah kepercayaan yang tinggi dengan hasil arus yang rendah. Motor yang tidak efisien menghasilkan kepercayaan kecil atau membutuhkan terlalu banyak arus. Efisiensi drone motor tipikal memiliki nilai 7.

Jika motor mencapai RPM maksimum, dorongan maksimum torsi motor dan kombinasi baling-baling tercapai. Torsi motor menentukan seberapa cepat motor dapat memvariasikan kecepatan baling-baling. Torsi motor juga memengaruhi kinerja terbang. Jika torsi motor rendah, quadcopter dapat memiliki respons yang lambat terhadap perintah dan penerbangannya bisa kaku. Penting untuk menemukan keseimbangan yang baik antara motor RPM dan torsi, dengan anggapan bahwa motor drone cocok[11].



Gambar 2-18 Efek Grafitasi dan Pergerakan pada Quadcopter

2.6. Electronic speed controller (ESC)

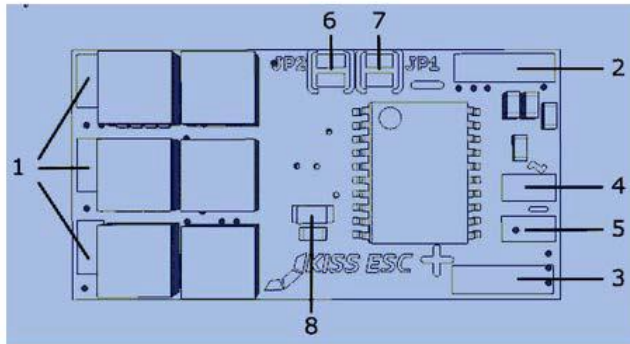
ESC dapat bertugas untuk mengubah kecepatan dan arah motor dengan memanipulasi tegangan yang diterapkan padanya. Selain itu, dapat memberikan start-up yang terkendali untuk motor, membalikkan putaran motor atau membalikkan perlindungan polaritas, dalam hal koneksi positif ke kontak negatif, perlindungan terhadap kesalahan sirkuit kesalahan, dan menyediakan berbagai fitur lainnya, seperti kurva akselerasi yang halus, mengatur kecepatan tertinggi atau membatasi arus.

Kontroler yang mengalihkan koneksi ke motor yang terhubung dan terputus sekitar 2000 kali per detik oleh Pulse Width Modulation (PWM). Transistor MOSFET digunakan sebagai pengalih alih-alih sakelar mekanis. Kecepatan switching cepat dan motor tidak dapat mendeteksinya. Ini mengarah ke motor rata-rata pulsa ini keluar. Jika ada baterai 24V yang terhubung, hanya setengah dari masa pakai, motor melihat baterai seperti 12 V dan berjalan dengan kecepatan setengah. Kecepatan switching juga mempengaruhi induktansi motor, yang menjaga arus motor mengalir konstan. Namun, arus itu hanya mengalir separuh waktu dari baterai, sehingga arus baterai akan menjadi setengah dari arus.



Gambar 2-19 ESC

ESC sering digunakan dalam model Radio-controlled, khususnya dengan motor BLDC, di mana ia menyediakan sumber listrik tegangan rendah daya 3-fase yang diproduksi secara elektronik. Pengontrol biasanya memiliki 3 set kabel. Kabel pertama dihubungkan ke baterai utama drone. Kabel kedua adalah kawat servo tipikal yang dicolokkan ke throttle penerima, dan kabel ketiga digunakan untuk daya motor[12].



Gambar 2-20 Kontruksi pada ESC

1. Solder pad untuk fase motor 3-BLDC
2. Negatif (-) koneksi LIPO
3. Positif (+) Koneksi LIPO
4. Sinyal servo atau input dari sinyal PWM
5. Referensi GND dari Sinyal PWM
6. Solder jumper, untuk mengubah arah Rotation (CW / CCW)
7. Solder jumper, untuk memvariasikan jenis sinyal input PWM
8. Status LED

2.7. Haversine Formula

Haversine formula adalah sebuah persamaan yang digunakan untuk menghitung lingkaran besar jarak antara titik awal dan akhir menggunakan koordinat latitude dan longitude. Rumus ini digunakan untuk mengetahui jarak antara titik posisi awal ke titik yang diinginkan. Tanpa rumus ini pembuatan robot otonom sangat sulit karena ada banyak variasi dalam koordinat GPS[13].

$$X = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi \cos\varphi \sin^2\left(\frac{\Delta\gamma}{2}\right) \quad (2)$$

$$Y = 2 \sqrt{X} \operatorname{atan} 2(\sqrt{x}, \sqrt{1-x}) \quad (3)$$

$$\text{Distance} = Y \times R \quad (4)$$

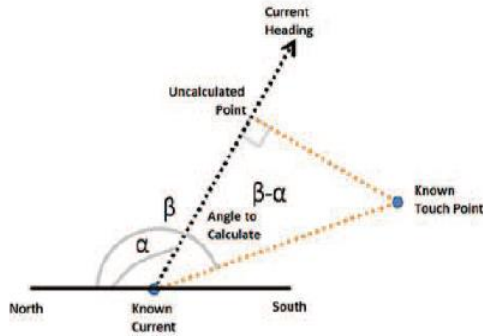
Keterangan:

$\Delta\varphi$ = Latitude tujuan- latitude awal

$\Delta\gamma$ = Longitude tujuan- longitude awal

φ = Latitude
 γ = Longitude

Setelah mendapatkan jarak posisi dari haversine formula maka dapat diketahui sudut heading dari posisi awal ke titik tujuan yang ingin dituju. Gambar 2.21 merupakan sudut tujuan yang ingin dituju



Gambar 2-21 Sudut Tujuan Yang Dituju

Sudut tujuan dapat dihitung menggunakan rumus sebagai berikut.

$$X = \sin(\Delta\gamma) \cos(\varphi_1) \quad (5)$$

$$Y = \cos(\varphi_1) \sin(\varphi_2) - \sin(\varphi_1) \cos(\varphi_2) \cos(\Delta\gamma) \quad (6)$$

$$\text{Angle} = \tan^{-1} \left(\frac{Y}{X} \right) \quad (7)$$

Keterangan:

$\Delta\varphi$ = Lattitude tujuan- latitude awal

$\Delta\gamma$ = Longitude tujuan- longitude awal

φ_1 = Lattitude awal

φ_2 = Lattitude Tujuan

$K = 180/\pi$

Hasil dari bearing dan jarak pada tugas akhir ini akan diolah menjadi referensi antar UAV ke UAV lainnya sehingga robotpun dapat berjalan sesuai yang diinginkan

2.8. **Kontroller PID**

Kontroller PID merupakan suatu kontroller yang digunakan pada kalangan industri saat ini karena memiliki kestrukturan yang simple tetapi memiliki kinerja yang baik dalam beberapa proses sesuai yang diinginkan. Dalam penggunaan control PID harus melakukan 3 prosedur diantaranya: identifikasi pemodelan proses, structure control design dan pid parameter tuning. Pid controller mempunyai 3 parameter yaitu propotal gain K, integral time derivative time. Prosedur yang digunakan untuk menemukan nilai pada parameter controller disebut tuning[14]. output dari controller PID.dapat dihitung sebagai berikut

$$u(t) = Kp e(t) + Ki \int_0^t e(T) dT + Kd \frac{de(t)}{dt} \quad (8)$$

Keterangan:

- nilai Kp digunakan untuk mengurangi nilai rise time tapi tidak dapat menghilangkan nilai dari error steady state.
- Nilai Ki digunakan untuk menghilangkan error steady state tetapi membuat response transient menjadi tidak sesuai.
- Nilai Kd digunakan untuk meningkatkan kestabilan pada system, mengurangi overshoot dan mengubah response transient menjadi sesuai yang diinginkan.

2.9. **Arduino**

Arduino Nano adalah suatu board microcontroller yang dikembangkan oleh Arduino.cc dan berbasis pada atmega 328p/atmega 168. Board arduino banyak digunakan dalam robotika, sistem tertanam dan proyek elektronika dimana otomasi merupakan bagian penting dari otomasi.arduino nano memiliki fungsi yang kurang lebih sama dari arduino duemilanove tetapi tidak memiliki dc power jack dan dapat beroperasi menggunakan kabel mini -B usb.



Gambar 2-22 Pin Arduino Nano

2.10. ESP8266

ESP8266 adalah nama pengontrol mikro yang dirancang oleh Espressif Systems. ESP8266 sendiri adalah solusi jaringan WiFi mandiri sebagai jembatan dari pengontrol mikro yang ada ke WiFi dan juga mampu menjalankan aplikasi mandiri.

Modul ini dilengkapi dengan konektor USB bawaan dan beragam pin-out. Dengan kabel micro USB, Anda dapat menghubungkan NodeMCU devkit ke laptop Anda dan mem-flash-nya tanpa masalah, sama seperti Arduino. Ini juga segera papan tempat memotong roti ramah.



Gambar 2-23 ESP8266

2.11. Lipo Battery

Baterai lithium polimer, atau lebih tepatnya baterai polimer lithium-ion (disingkat dengan berbagai macam LiPo, LIP, Li-poly, dan lainnya), adalah baterai isi ulang teknologi lithium-ion dalam format kantong. Tidak seperti sel silindris dan prismatic, LiPos datang dalam paket lunak atau kantong, yang membuatnya lebih ringan tetapi juga kurang kaku.

Quadcopters biasanya menggunakan baterai LiPo yang datang dalam berbagai ukuran dan konfigurasi. Kami biasanya menggunakan baterai 3S1P, yang menunjukkan 3 sel secara paralel. Setiap sel 3,7 volt, sehingga baterai ini memiliki peringkat 11,1 volt. Baterai LiPo juga memiliki peringkat C dan peringkat daya dalam mAh (yang berarti milliamps per jam). Peringkat C menggambarkan tingkat di mana daya dapat diambil dari baterai, dan peringkat daya menggambarkan berapa banyak daya baterai dapat memasok. Baterai yang lebih besar lebih berat sehingga selalu ada trade off antara durasi penerbangan dan total berat. Aturan umum adalah menggandakan daya baterai akan memberi Anda 50% lebih banyak waktu penerbangan, dengan anggapan quadcopter Anda dapat mengangkat bobot tambahan.

Baterai Li Po memiliki tiga hal utama yang menjadikannya pilihan baterai yang sempurna untuk pesawat RC dan terlebih lagi untuk helikopter RC dibandingkan jenis baterai isi ulang konvensional seperti NiCad, atau NiMH.

- Baterai Li Po ringan dan dapat dibuat dalam hampir semua bentuk dan ukuran.
- Baterai Li Po memiliki tingkat debit tinggi untuk memberi daya pada motor listrik yang paling banyak diminati.
- Baterai Li Po menyimpan banyak daya dalam satu paket kecil.

Sama seperti sel-sel lithium-ion lainnya, LiPos bekerja berdasarkan prinsip interkalasi dan deintercalasi ion litium dari bahan elektroda positif dan bahan elektroda negatif, dengan elektrolit cair menyediakan media konduktif. Untuk mencegah elektroda saling menyentuh secara langsung, pemisah mikroporous di antaranya memungkinkan hanya ion dan bukan partikel elektroda untuk bermigrasi dari satu sisi ke sisi lain.

memiliki case yang fleksibel, tipe foil (laminasi polimer), sehingga relatif tidak dibatasi. Sendiri sel-sel lebih dari 20% lebih ringan dari sel-sel silinder setara dengan kapasitas yang sama.

Menjadi ringan adalah keuntungan ketika aplikasi membutuhkan berat minimum, seperti dalam kasus model yang dikendalikan radio. Namun, telah diselidiki bahwa tekanan sedang pada tumpukan lapisan yang menyusun sel menghasilkan peningkatan kapasitas retensi, karena kontak antara komponen dimaksimalkan dan delaminasi dan deformasi dicegah, yang terkait dengan peningkatan impedansi sel dan degradasi sel



Gambar 2-24 Lipo Baterai

2.12. Tinjauan Pustaka

Pada tinjauan pustaka diperoleh dari beberapa paper yang pernah dilakukan sebelumnya. Dari paper tersebut akan dilakukan perbandingan antara proyek yang dilakukan pada paper dengan pengerjaan tugas akhir ini.

2.12.1. Wireless Communications and Control for Swarms of Cellular-Connected UAVs

Pada paper menggunakan metode komunikasi menggunakan triangle formation dimana metode ini swarm drone akan mengirimkan informasi satu sama lain sehingga drone dapat menyelesaikan informasi sesuai yang diinginkan pengguna. Berbeda halnya dengan tugas akhir ini pada tugas akhir akan dirancang swarm drone berbasis leader follower dimana komunikasi swarm drone terpusat dari leader. Pengiriman informasi akan dikirim pada follower-follower lain dan follower tidak bertugas untuk mengirim informasi kepada follower lain [2].

2.12.2. Massive MIMO for Communications with Drone Swarms

Pada paper ini komunikasi antara swarm drone yang digunakan ialah menggunakan zigbee dimana zigbee bertugas untuk mengirim data informasi antara swarm drone lainnya. Berbeda halnya yang dilakukan pada tugas akhir ini, pengiriman pada tugas akhir ini menggunakan modul node mcu yang bertugas untuk mengirim data remote dan data gps yang terletak pada drone leader dan drone follower terdapat juga node mcu

yang bertugas untuk menerima data yang diberikan kepada drone leader[15].

2.12.3. On the impact of inter-UAV communications interference in the 2.4 GHz band

Pada paper ini swarm drone masih dilengkapi peralatan yang lengkap dan tidak ada perbedaan komponen pada drone leader dan drone yang masih dilengkapi dengan receiver dan remote controller yang berbeda.berbeda halnya dengan perancangan tugas akhir ini akan dibuat drone leader dan drone follower dimana drone leader terdapat receiver sedangkan drone follower tidak terdapat receiver. Drone leader bertugas untuk mengirim data remote kepada drone follower sehingga drone leader dan drone follower dapat bergerak keposisi yang diinginkan oleh pengguna[16].

[Halaman ini sengaja dikosongkan]

BAB 3 PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem yang terdapat pada tugas akhir ini antara lain: swarm drone, sistem leader-follower, penggunaan rumus haversine formula, perancangan hardware, perancangan elektronik dan perancangan software.

3.1. Swarm Drone

Swarm Drone yang digunakan pada tugas akhir ini terdiri dari 3 quadcopter yaitu 1 leader dan 2 follower. Gambar 3.1 merupakan swarm drone leader dan follower yang dibuat pada tugas akhir ini.

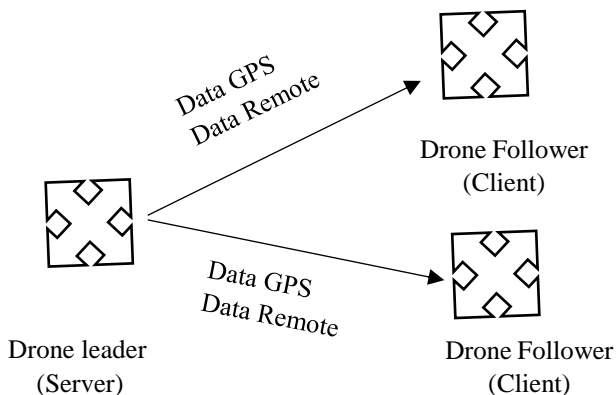


Gambar 3-1 Swarm Drone Leader-Follower

Pada gambar 3.1 drone yang bernomer 1 merupakan drone sebagai leader yang bertugas untuk memberikan instruksi kepada pesawat yang bernomer 2 dan 3 yang bertugas sebagai follower. drone leader dapat mengirimkan intruksi kepada follower apabila sistem pada drone leader dijadikan sebagai server dan sistem pada drone follower dijadikan sebagai client. dari kedua hal tersebut maka terdapat perbedaan antara sistem leader maupun sistem follower.

3.2. Drone Leader Follower

Seperti yang dijelaskan pada sebelumnya drone leader dapat berkoordinasi dengan sistem follower yaitu dengan cara server dan client. Penggunaan server client pada tugas akhir ini selain bertujuan untuk pengiriman intruksi yang diberikan oleh follower sistem server client juga berfungsi agar pengiriman intruksi dapat berkoordinasi dengan jarak yang jauh .hal ini bertujuan agar penerbangan pada drone leader dan drone follower tetap aman dan tidak mengalami masalah.pengiriman antara server dan client menggunakan router.jika pengiriman antara server dan client gagal maka drone leader follower akan melakukan pendaraan secara otomatis.Gambar 3.2 merupakan cara pengkoordinasian yang terdapat pada drone leader dan drone follower.



Gambar 3-2 Sistem Drone Leader Follower

3.2.1. Konfigurasi Posisi Swarm Leader Follower

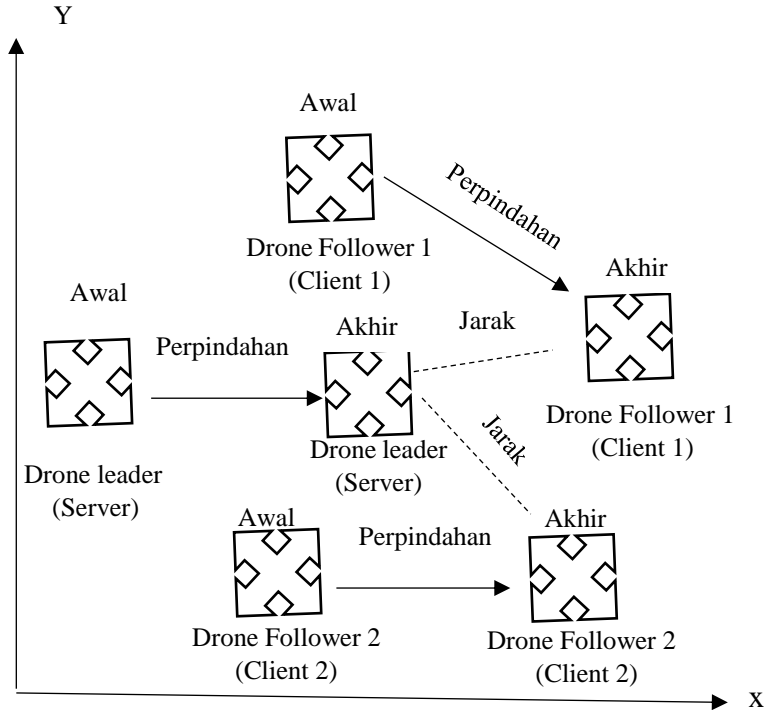
Dari posisi yang didapat menggunakan haversine formula maka selanjutnya adalah mencari titik x dan y pada drone leader maupun drone follower.estimasi jarak antara drone leader dan follower didapat dari persamaan jarak dan sudut yang didapatkan dari haversine formula

$$X = \text{jarak} \cos \varphi \quad (9)$$

$$Y = \text{jarak} \sin \varphi \quad (10)$$

Dari persamaan diatas maka didapat posisi pada x dan y pada drone leader dan follower.dari jarak tersebut maka terdapat selisih antara drone

leader maupun drone follower gambar 3.3 merupakan ilustrasi gambar konfigurasi drone leader dan drone follower.



Gambar 3-3 Konfigurasi Posisi Drone Leader Follower

Untuk setting posisi drone leader dan drone follower menuju posisi drone leader dan drone follower selanjutnya menggunakan suatu persamaan sebagai berikut:

$$X \text{ perpindahan} = X \text{ set point} - X \text{ awal} \quad (11)$$

$$Y \text{ perpindahan} = Y \text{ set point} - Y \text{ awal} \quad (12)$$

Sedangkan untuk menghitung jarak posisi antara drone leader dan drone follower dibutuhkan suatu persamaan sebagai berikut

$$X \text{ jarak drone leader-follower1} = X \text{ drone leader} - X \text{ drone follower1} \quad (13)$$

$$Y \text{ jarak drone leader-follower1} = Y \text{ drone leader} - Y \text{ drone follower1} \quad (14)$$

$$X \text{ jarak drone leader-follower2} = X \text{ drone leader} - X \text{ drone follower2} \quad (15)$$

$Y \text{ jarak drone leadefollower} = Y \text{ drone leader} - Y \text{ drone follower}$ (16)

3.3. Rancang Bangun Mekanik Drone Leader Follower

Perancangan hardware merupakan perancangan perangkat keras berupa perancangan mekanik yang terdapat pada drone leader dan drone follower. Pada kedua drone ini menggunakan perancangan mekanik berupa ukuran dan frame yang sama. Gambar 3.3 merupakan frame tipe zmr yang dipakai pada drone leader maupun drone follower.



Gambar 3-4 Frame Drone Leader-Follower

ukuran pada frame yang digunakan pada drone leader dan drone follower berukuran 250mm. dan gambar 3.4 merupakan hasil rancang bangun mekanik yang telah dibuat pada drone leader maupun drone follower.



Gambar 3-5 Rancang Bangun Mekanik Drone Leader dan Follower

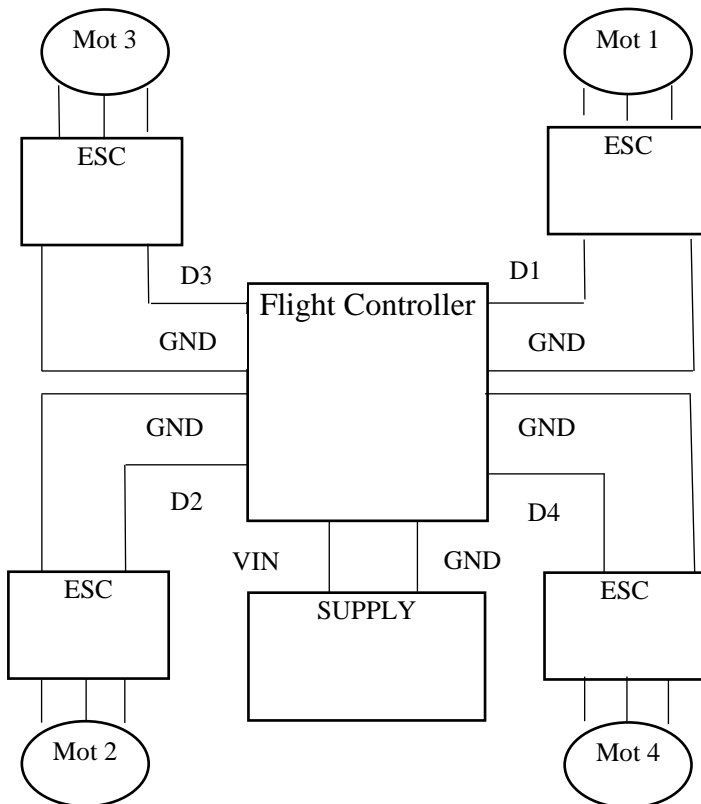
3.4. Rancang Bangun Elektronik

Bagian ini membahas tentang perancangan elektronik yang terdapat pada drone leader maupun drone follower. Pada drone terdapat beberapa komponen penting yang berfungsi dari pengambilan data hingga untuk pengintegrasian antara flight controller agar drone berjalan dengan

sempurna. Dibawah ini merupakan bagian-bagian dari komponen sebagai berikut:

3.4.1. ESC, Motor, dengan Flight Controller

Pada bagian ini akan dibahas mengenai pengintegrasian antara esc motor, PDB dan flight controller. pentingnya bagian ini adalah untuk mengatur arah putaran motor dan mengatur urutan pergerakan motor yang terdapat pada quadcopter. dibawah ini merupakan diagram integrasi kabel antara ESC, Motor dan Flight Controller

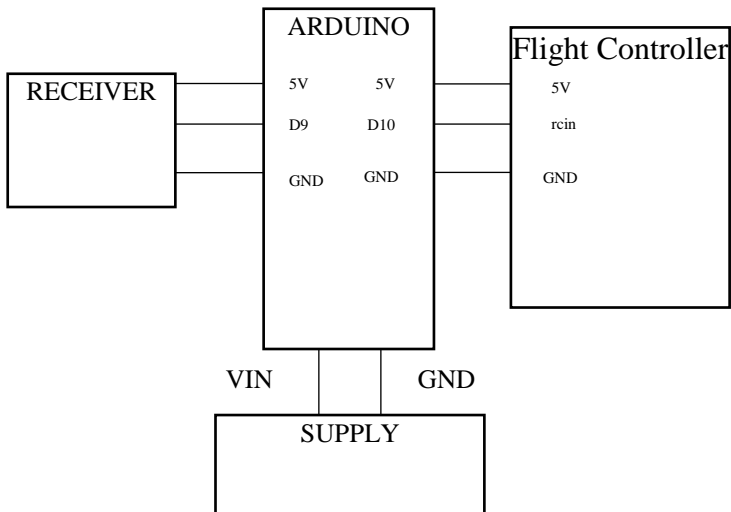


Gambar 3-6 Pengkabelan ESC, Motor dengan Flight Controller

Pengkabelan pada ESC dan flight controller dihubungkan sesuai dengan urutan yang terdapat pada gambar diatas. ESC 1 dihubungkan pada flight controller ESC1,ESC2 dihubungkan pada flight controller ESC2 dan seterusnya. Hal ini berfungsi agar pada saat terbang quadcopter dapat terbang sesuai dengan konfigurasi firmware quadcopter yang tersedia pada flight controller agar tidak mengalami kesalahan. Pada motor cw maupun ccw dihubungkan sesuai dengan urutan esc tetapi untuk masalah pengkabelan pada motor cw salah satu kabelnya harus dibalik agar putaranya sesuai dengan konfigurasi yang ada pada flight controller. Supply pada esc menggunakan battery lipo yang telah diintegrasikan menggunakan PDB begitu juga flight controller disupply menggunakan lipo yang telah diconvert menggunakan power modul yang tersedia pada PDB.

3.4.2. Receiver,Arduino dan Flight Controller

Berikut ini merupakan perancangan wiring antara arduino dan flight controller.perancangan ini bertujuan untuk mengatur sinyal ppm yang diberikan oleh receiver.



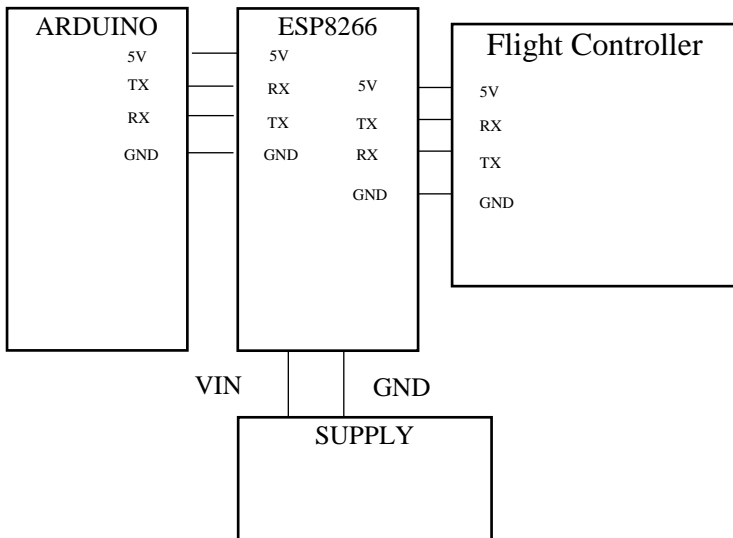
Gambar 3-7 Wiring Diagram Receiver,Arduino Nano dan Minipix

Data receiver yang dikirim akan diterima dan diproses oleh arduino menggunakan interupt sehingga data ouput yang akan dikeluarkan oleh

arduino sama dengan data input yang dikirim oleh receiver. Perbedaan dari leader dan follower pada sistem ini adalah pada drone sebagai leader dilengkapi dengan receiver sedangkan drone sebagai follower tidak. Fungsi dari adanya receiver pada drone leader adalah untuk memberi perintah dan intruksi kepada drone leader maupun drone follower agar tidak terjadi kesalahan saat terbang.

3.4.3. Arduino,ESP8266 dan Flight Controller

Berikut ini merupakan perancangan wiring antara arduino dengan esp8266. Perancangan ini bertujuan untuk memberikan data dari hasil mavlink yang telah di proses esp8266.

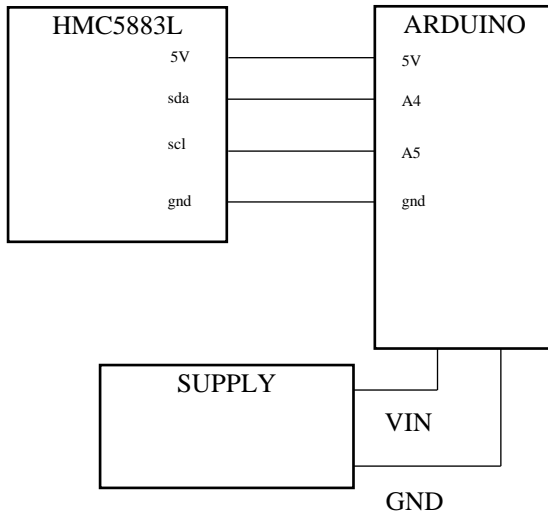


Gambar 3-8 Wiring Diagram ESP,Arduino dan Minipix

Data yang proses oleh nodemcu melalui mavlink merupakan data GPS dan data altitude. Data ini terdapat dalam satu protocol mavlink sehingga akses waktu pada gps dan data altitude tidak mengalami delay yang terlalu lama. Tugas dari arduino ini adalah memberikan referensi data yang diterima sehingga drone yang berperan sebagai leader maupun follower dapat diatur pergerakan posisinya.

3.4.4. HMCL5883I dengan arduino

Berikut ini merupakan perancangan wiring antara arduino dengan HMCL5883I. Perancangan ini bertujuan untuk memberikan referensi pada drone leader maupun follower agar dapat mengetahui rotasi sesuai dengan posisi yang dituju.

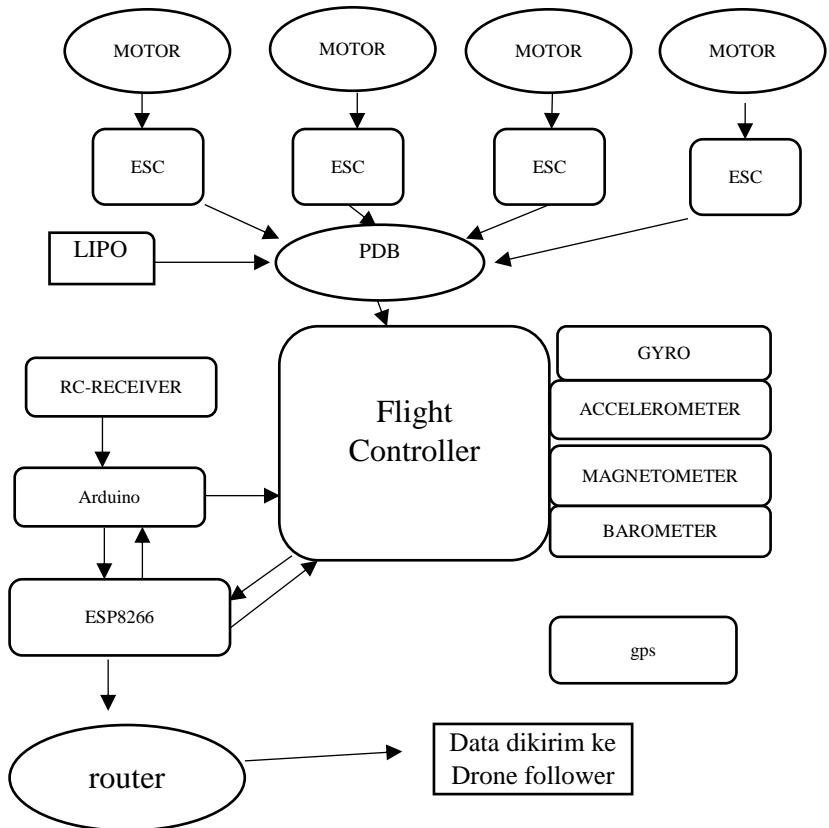


Gambar 3-9 Wiring Diagram HMC5833L dan Arduino

Sensor ini diakses menggunakan I2C dimana pin A4 pada arduino nano dapat berfungsi sebagai pin SCL dan pin A5 pada arduino nano berfungsi sebagai pin SDA.

3.4.5. Diagram Blok Elektronik Drone Leader

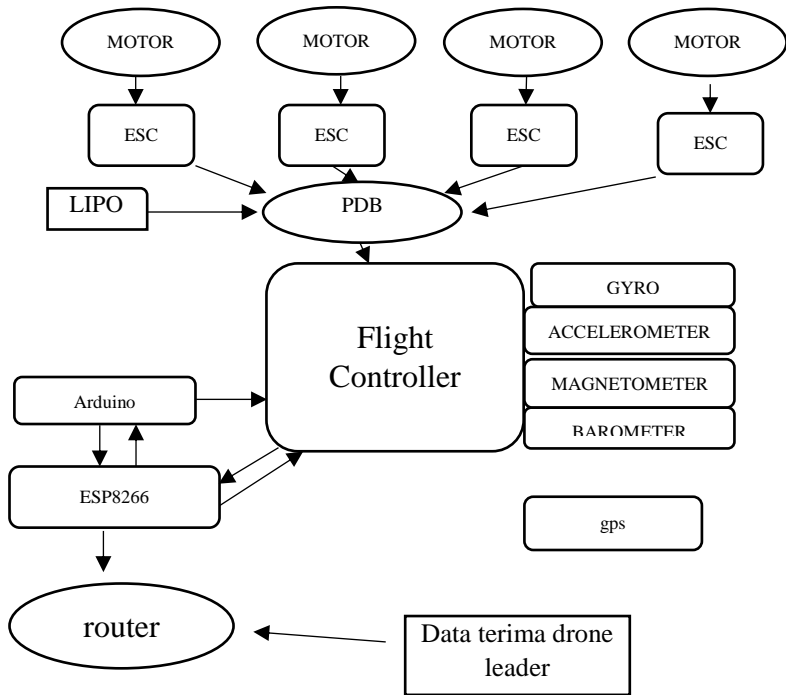
Dari beberapa komponen diatas pada drone *leader* dan drone *follower* terdapat komponen yang berbeda yaitu terdapat receiver pada drone *leader* sedangkan drone *follower* tidak terdapat receiver. gambar 3.9 merupakan diagram blok elektronik yang terdapat pada drone leader.



Gambar 3-10 Diagram Blok Elektronik Drone Leader

3.4.6. Diagram Blok Elektronik Drone Follower

Diagram blok ini menjelaskan ada perbedaan komponen yang tidak terdapat pada drone. selain menjelaskan perbedaan diagram blok ini menjelaskan cara pengintegrasian komponen pada drone follower. gambar 3.10 merupakan diagram blok elektronik yang terdapat pada drone follower.



Gambar 3-11 Diagram Blok Elektronik Drone Follower

3.5. Rancang Bangun Software

Pada bagian ini membahas tentang perancangan software yang akan dipakai pada drone leader maupun drone follower. Perancangan ini terdiri dari proses kalibrasi, pengaturan perhitungan pergerakan hingga perancangan program. Terdapat 2 bagian pada perancangan software yaitu perancangan konfigurasi mission planner dan perancangan program pada software arduino.

3.5.1. Perancangan Konfigurasi Mission Planner

Berikut ini merupakan perancangan konfigurasi pada mission planner. Tujuan perancangan ini dilakukan agar drone tidak terjadi kesalahan pada saat terbang. Apabila konfigurasi ini selesai maka drone

akan diterbangkan secara manual tanpa menggunakan program. Berikut ini merupakan macam- macam konfigurasinya.

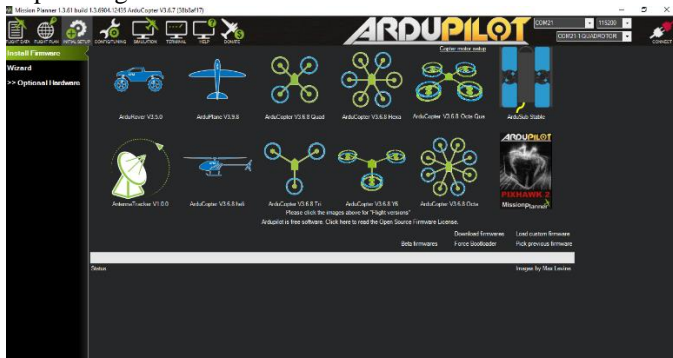
3.5.2. Konfigurasi Firmware

Berikut ini merupakan langkah awal persiapan pada drone sebelum diuji maupun dikalibrasi agar tidak terjadi kesalahan. Software yang digunakan tugas akhir ini adalah Mission planner. Dibawah ini merupakan menu utama dari mission planner.



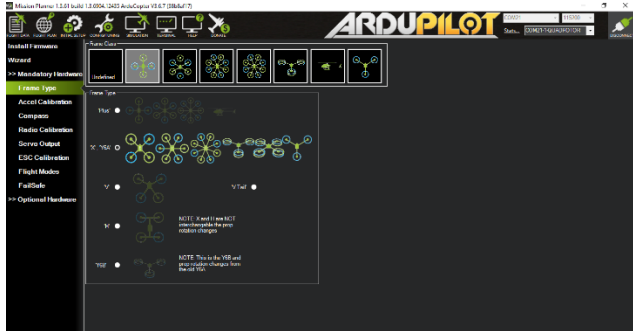
Gambar 3-12 Menu Utama Mission Planner

Sesudah masuk dari menu utama langkah selanjutnya adalah pemilihan firmware pada mission planner. pemilihan firmware dilakukan dengan menggunakan kabel usb dan tanpa menkoneksikannya terlebih dahulu ke mission planner agar firmware pada mission planner dapat diupload pada flight controller



Gambar 3-13 Firmware pada Mission Planner.

Firmware yang digunakan pada tugas akhir ini ialah firmware quadcopter. Sesudah melakukan pemilihan firmware pada quad yang harus dilakukan selanjutnya ialah pemilihan frame pada mission planner.



Gambar 3-14 Pemilihan Frame pada Mission Planner

Frame yang digunakan oleh quadcopter drone maupun follower ialah quadcopter dengan tipe frame x.

3.5.3. Kalibrasi Kompas

Kalibrasi kompass dilakukan saat sebelum terbang agar tidak terjadi kesalahan terbang saat dilakukan. Kalibrasi kompass dilakukan dengan cara memutar rotasi pada saat miring kanan, miring kiri. Gerakan kebalik, hadap atas dan hadap bawah. Kalibrasi dilakukan menggunakan software mission planner.



Gambar 3-15 Kalibrasi Kompas pada Mission Planner

3.5.4. Kalibrasi Acel dan Level

Selain Kalibrasi kompas hal yang selanjutnya perlu dilakukan adalah kalibrasi accel dan level. Kalibrasi ini dilakukan dengan level ketinggian quadcopter yang sama pada saat miring kanan, miring kiri, hadap atas, hadap bawah dan pada saat copter dibalik.



Gambar 3-16 Kalibrasi Acel dan Level pada Mission Planner

3.5.5. Kalibrasi Radio Control

Kalibrasi Radio Control merupakan kalibrasi data yang dikirim oleh remot pada flight controller. Hal ini berfungsi agar pengiriman data remot sesuai pada flight controller. Radio control pada tiga akhir ini menggunakan sinyal ppm sehingga kalibrasi hanya dilakukan pada 8 channel saja.



Gambar 3-17 Kalibrasi Radio Control pada Mission Planner

3.5.6. Kalibrasi ESC

Kalibrasi dilakukan dengan menggunakan mission planner. Berikut ini merupakan cara kalibrasi ESC menggunakan software mission planner.



Gambar 3-18 kalibrasi ESC

Kalibrasi dilakukan dengan cara menekan tombol esc kalibrasi lalu esc akan mengkalibrasi sendiri melalui sistem ini. Kalibrasi ini dilakukan bertujuan agar kecepatan antar motor yang terdapat pada quadcopter bisa sama.

3.5.7. Konfigurasi Failsafe

Setelah mengkalibrasi ESC hal selanjutnya adalah mengkonfigurasi failsafe yang terdapat pada pixhawk.hal ini bertujuan untuk mengamankan sistem pada drone apabila terjadi hal-hal yang tidak diinginkan terutama pada sinyal remot yang dikirim. Berikut ini adalah cara mengkonfigurasi failsafe yang terdapat pada mission planner



Gambar 3-19 Konfigurasi Failsafe pada Mission Planner

3.5.8. Konfigurasi PID

Setelah melakukan kalibrasi-kalibrasi diatas langkah selanjutnya adalah mengkonfigurasi nilai pid pada quadcopter. Konfigurasi ini dilakukan dengan metode tuning. Dibawah ini merupakan cara konfigurasi PID stabilize pada saat terbang.



Gambar 3-20 Konfigurasi PID menggunakan Mission Planner

Konfigurasi ini dilakukan agar saat terbang drone mengalami stabil dan tidak terjadi kesalahan pada saat terbang.

3.5.9. Konfigurasi Flight Modes

Setelah drone dapat terbang dalam mode stabilize maka selanjutnya memberikan setting flight mode pada channel 5. Berikut ini merupakan cara setting flight modes pada mission planner.



Gambar 3-21 Konfigurasi Flight Mode pada mission planner

Pada tugas akhir ini drone menggunakan 3 flight mode antara lain: stabilize, loiter dan land. Mode stabilize bertujuan untuk mengecek apakah data remot masuk atau tidak. Mode loiter bertujuan untuk

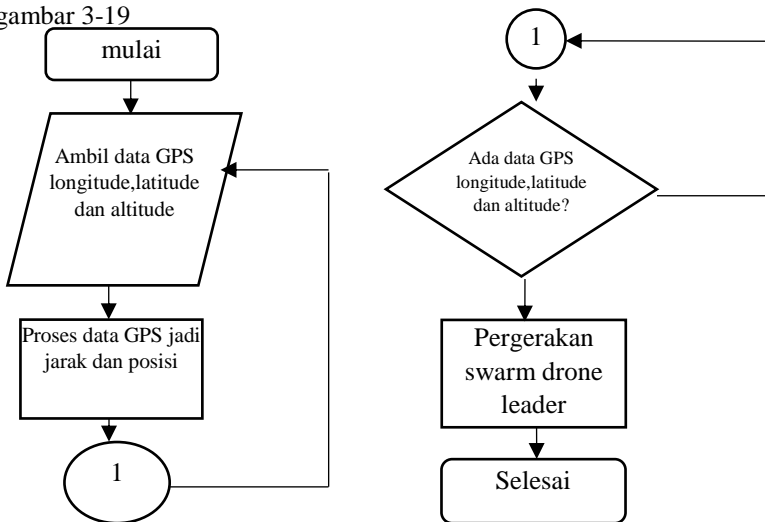
mempertahankan posisi pada quadcopter sehingga quadcopter dapat dikontrol ke posisi lain menggunakan arduino. Sedangkan mode land bertujuan untuk mengamankan drone apabila sulit pengguna tugas akhir sulit mengendalikannya.

3.5.10. Perancangan Segmen Program pada Arduino IDE

Setelah melakukan perancangan-perancangan software pada mission planner. Hal selanjutnya yang dilakukan adalah perancangan segmen-segmen program pada arduino.hal ini bertujuan agar drone dapat dikendalikan secara autonomous. Berikut ini merupakan langkah-langkah pembuatan segmen program pada drone sebagai leader maupun drone sebagai follower:

3.5.11. Diagram Alur Pengambilan Data Gps dan Pergerakan Drone Leader

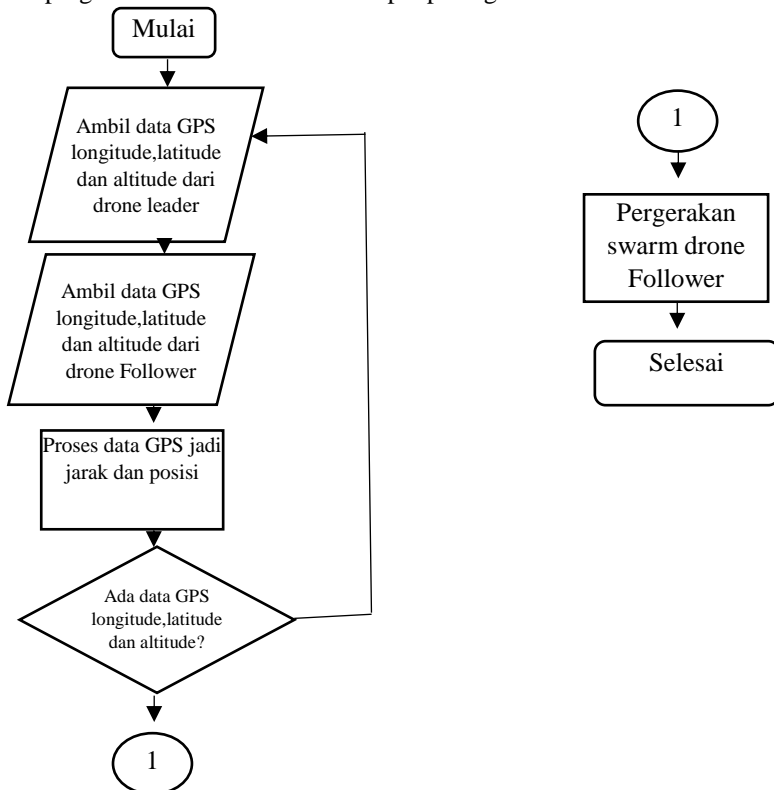
Sebelum melakukan perancangan pada segmen program hal yang dilakukan terlebih dahulu ialah perancangan diagram alur pengambilan data GPS dan pergerakannya.drone pada leader menentukan data posisi yang dituju melalui pengambilan data longitude dan latitude. setelah itu maka drone tersebut dapat menentukan jarak posisi yang dituju.diagram alur pengambilan data GPS dan pergerakan drone leader terdapat pada gambar 3-19



Gambar 3-22 Flowchart Diagram Alur Drone Leader

3.5.12. Diagram Alur Pengambilan data GPS dan Pergerakan Drone Follower

Sesudah melakukan perancangan diagram alur pada drone leader hal yang dilakukan selanjutnya ialah pembuatan diagram alur pengambilan data GPS dan pergerakan drone follower. pergerakan drone follower diatur dari data hasil kirim yang terdapat pada drone leader sehingga data tersebut akan dibandingkan dengan data gps yang terdapat pada drone follower. dengan ini maka drone follower dapat mengetahui posisi yang ingin ditujunya. diagram alur pengambilan data GPS dan pergerakan drone follower terdapat pada gambar 3-20.



Gambar 3-23 Flowchart Diagram Alur Drone Follower

3.5.13. Pendefinisian Library Arduino pada Drone Leader & Follower.

Pendefinisian library arduino digunakan untuk pendefinisian awal program guna untuk pengambilan fungsi-fungsi pada arduino. Pada arduino leader dan follower terdapat beberapa library yang digunakan dibawah ini.

```
#include <stdio.h>
#include <string.h>
#include <CPPM.h>
#include <Arduino.h>
#include <math.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#define RADIUS 6371000
#define address 0x1E //0011110b, I2C 7bit address of HMC5883
#define Declination -0.00669
#define pi 3.1415926535897932384626433832795028841

SoftwareSerial fromnodemcu(2, 3);//RXTX
```

Gambar 3-24 Pendefinisian Library pada Arduino

Salah satu Pendefinisian penting yang digunakan ialah CPPM.h dimana pendefinisian tersebut berfungsi untuk pengintegrasian arduino agar dapat dilakukan dengan PPM pada remot. Sedangkan pada follower CPPM.h berfungsi untuk menggantikan kerja sinyal ppm yang ada pada receiver sehingga data remote dapat diakses pula hanya dengan menggunakan arduino saja tanpa menggunakan receiver.

3.5.14. Pendefinisian Library NodeMcu pada Drone Leader & Follower.

Sama halnya dengan arduino pendefinisian library juga dilakukan pada nodemcu guna untuk pengambilan fungsi-fungsi yang terdapat pada library.pada nodemcu leader dan follower terdapat beberapa library yang digunakan dibawah ini.

```
#include <mavlink.h>
#include <SoftwareSerial.h>
```

```

SoftwareSerial _MavLinkSerial(4, 14); // RX, TX
SoftwareSerial toarduino(0, 5); // RX, TX
#include <ESP8266WiFi.h>
#include <SPI.h>

```

Gambar 3-25 Pendefinisian Library Pada NodeMCU

Pendefinisian library dilakukan guna untuk mendapatkan data mavlink dan koneksi ke router. pada mavlink ada beberapa data yang diambil diantaranya adalah data longitude, data latitude dan data altitude. setelah itu data tersebut akan diolah pada masing-masing drone leader maupun drone follower.

3.5.15. Segmen Program Pengaturan Awal Arduino Drone Leader dan Drone Follower.

Segmen program pengaturan awal arduino terdapat pada drone leader maupun drone follower. dibawah ini merupakan segmen pengaturan awal yang ada pada arduino drone sebagai leader maupun arduino drone sebagai follower.

```

void setup() {
  Serial.begin(9600);
  frommodemcu.begin(9600);
  //frommcu.begin(9600);
  CPFM.begin();
  Wire.begin();
  Wire.beginTransmission(address); //open communication with HMC5883
  Wire.write(0x02); //select mode register
  Wire.write(0x00);
  Wire.endTransmission();
}

```

Gambar 3-26 Segmen Program Pengaturan Awal pada Arduino

Program diatas merupakan program pemanggilan untuk fungsi-fungsi diantaranya ialah fungsi ppm, fungsi i2c dan fungsi untuk serial. Fungsi i2c dipakai untuk pemanggilan address pada HMCL5883L.

3.5.16. Segmen Program Pengaturan Awal NodeMCU Drone Leader dan Drone Follower.

Sama Halnya dengan Arduino pada NodeMCU dibutuhkan pengaturan awal. Dibawah ini merupakan Segmen Program Pengaturan Awal yang terdapat pada NodeMcu didrone leader maupun drone follower.

```
char ssid[] = "a206";           // SSID of your home WiFi
char pass[] = "a206jaya";       // password of your home WiFi

IPAddress server(192,168,0,2);    // the fix IP address of the server
// IPAddress server(192,168,43,232); // the fix IP address of the server

//
  IPAddress ip(192, 168, 0, 3);
  IPAddress gateway(192, 168, 0, 1);
  IPAddress subnet(255, 255, 255, 0);
//IPAddress follower(192, 168, 0, 103);
  WiFiClient client;
int secondd=0;
void setup() {

  WiFi.config(ip, gateway, subnet); // forces to use the fix IP
  WiFi.begin(ssid, pass);
  WiFi.mode(WIFI_STA);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(10);
  }
  Serial.println("Connected to wifi");
  Serial.println("MAVLink starting.");
  _MavLinkSerial.begin(9600);
  toarduino.begin(9600);
  toarduino1.begin(9600);
  Serial.begin(9600);
}
```

Gambar 3-27 Segmen Program Pengaturan Awal NodeMCU pada Drone Follower

Program diatas merupakan pengaturan awal pada nodemcu yaitu konfigurasi IP dan konfigurasi nodeMCU sebagai Server dan Client. Pada drone yang berperan sebagai leader akan dikonfigurasi sebagai server guna untuk mengirim data yang diinginkan oleh pengguna. Sedangkan

pada drone yang berperan sebagai follower akan dikonfigurasi sebagai client guna untuk menerima data hasil kiriman yang telah dikirim oleh drone sebagai leader.

3.5.17. Segmen Program Pengambilan data GPS melalui Mavlink

Segmen program pengambilan data gps dilakukan dengan pemanggilan protocol Mavlink.dibawah ini merupakan program segmen program pengambilan data GPS melalui protocol mavlink.

```
case MAVLINK_MSG_ID_GLOBAL_POSITION_INT: // #33
mavlink_msg_global_position_int_decode(msg, &global_position);
latitude=global_position.lat;
longitude=global_position.lon;
altitude=global_position.relative_alt;
```

Gambar 3-28 Segmen Program Pengambilan Data GPS.

Data yang diambil melalui protocol Mavlink ialah data longitude, data latitude dan data altitude yang terdapat pada minipix. Cara pengambilan data pada minipix ialah dengan cara komunikasi serial antara telem1 minipix dengan NodeMCU. Sebelum melakukan pengambilan data GPS pada gambar 3.25 hal yang dilakukan terlebih dahulu ialah request data pada mavlink agar data pengambilan GPS dapat dilakukan secara berulang-berulang. Dibawah ini merupakan program request data mavlink yang dilakukan nodemcu pada drone leader maupun drone Follower.

```
mavlink_message_t msg;
uint8_t buf[MAVLINK_MAX_PACKET_LEN];

// To be setup according to the needed information to be requested from the Pixhawk
const int maxStreams = 1;
const uint8_t MAVStreams[maxStreams] = {MAV_DATA_STREAM_POSITION};
const uint16_t MAVRates[maxStreams] = {0x05};
for (int i=0; i < maxStreams; i++) {
mavlink_msg_request_data_stream_pack(2, 200, msg, 1, 0, MAVStreams[i], MAVRates[i], 1);
uint16_t len = mavlink_msg_to_send_buffer(buf, msg);
_MavLinkSerial.write(buf, len);
}
}
```

Gambar 3-29 Segmen Program Request Data GPS

3.5.18. Segmen Program Penerimaan data GPS dari Drone Leader.

Setelah drone Leader mengirim data pada follower yang dilakukan ialah penerimaan data hasil kirim. Dibawah ini merupakan

segmen program penerimaan data hasil kirim pada drone Leader.

```
if (client.connect(server, 80))
{

    String request = client.readStringUntil('\n');

    if(request=="\0"){
        client.stop();
        return;
    }
    else{
        Serial.println(request);

        toarduino.print(request);
        secondd=0;
    }
    else{
        client.stop();
        return;
    }
}
```

Gambar 3-30 Segmen Program Penerimaan Data pada Drone Follower

3.5.19. Segmen Program Pengiriman data GPS dari Drone Leader ke Drone Follower

Setelah berhasil melakukan pengambilan data GPS pada drone Leader dan Drone Follower.hal yang dilakukan selanjutnya ialah pengiriman data longitude dan latitude yang terdapat pada drone leader. dibawah ini merupakan segmen program pengiriman data GPS dari drone leader ke drone Follower.


```

float calc_dist(float flat1, float flon1, float flat2, float flon2)
{
float dist_calc=0;
float dist_calc2=0;
float diflat=0;
float diflon=0;

diflat=radians(flat2-flat1);
flat1=radians(flat1);
flat2=radians(flat2);
diflon=radians((flon2)-(flon1));

dist_calc = (sin(diflat/2.0)*sin(diflat/2.0));
dist_calc2= cos(flat1);
dist_calc2*=cos(flat2);
dist_calc2*=sin(diflon/2.0);
dist_calc2*=sin(diflon/2.0);
dist_calc +=dist_calc2;

dist_calc=(2*atan2(sqrt(dist_calc),sqrt(1.0-dist_calc)));

dist_calc*=6371000.0; //Converting to meters
//Serial.println(dist_calc);
return dist_calc;
}

```

Gambar 3-32 Program Penentuan Jarak Posisi menggunakan Haversine Formula

```

float bearing(float lat1, float long1, float lat2, float long2) {
float dlon = radians(long2-long1);
lat1 = radians(lat1);
lat2 = radians(lat2);
float a1 = sin(dlon) * cos(lat2);
float a2 = sin(lat1) * cos(lat2) * cos(dlon);
a2 = cos(lat1) * sin(lat2) - a2;
a2 = atan2(a1, a2);

return degrees(a2);
}

```

Gambar 3-33 Program Penentuan Sudut menggunakan Haversine Formula

Dari penggunaan rumus pada gambar 3.32 dan gambar 3.33 maka drone leader dan drone follower dapat menentukan jarak posisi yang diinginkan. Setelah menentukan jarak posisi dan sudut yang diinginkan maka drone *leader* maupun drone *follower* dapat mengetahui titik x dan y sehingga drone *leader* dan drone *follower* dapat bergerak sesuai yang diinginkan. Gambar 3.34 merupakan segmen program untuk menentukan titik x dan titik y yang terdapat pada drone *leader* dan drone *follower*.

```
double positionx;
double positiony;
double convertpositiony(double distance, double angle) {
    positionx=distance*cos(todegree(angle));
    return positionx;
}

double convertpositionx(double distance, double angle) {
    positiony=distance*sin(todegree(angle));
    return positiony;
}
```

Gambar 3-34 Program Penentuan Titik Posisi x dan Titik Posisi y yang Terdapat pada Drone Leader dan Drone Follower

Jika titik posisi x dan titik posisi y sudah terdapat pada drone *leader* dan drone *follower* maka titik posisi x dan titik posisi y akan dijadikan referensi pergerakan drone *leader* dan drone *follower* agar dapat mengetahui drone harus maju mundur maupun bergerak kesamping melalui pemberian nilai pada roll dan pitch pada arduino. Gambar 3.35 merupakan segmen program pergerakan pemberian nilai roll dan pitch pada drone *leader* maupun *follower* dan dibawah ini merupakan persamaan perhitungan untuk melakukan pergerakan pada drone leader maupun drone follower.

Error X=Set point X- Posisi X (17)

Control roll=1500+Kp*Error X+Kd*(Error X- Error Sebelum X) (18)

If(Control roll>1800) Control roll=1800 (19)

Else if(Control roll<1200) Control roll=1200 (20)

Error Y=Set point Y- Posisi Y (21)

Control pitch=1500+Kp*Error Y+Kd*(Error Y- Error Sebelum Y) (22)

If(Control pitch >1800) Control pitch =1800 (23)

Else if(Control pitch <1200) Control pitch =1200 (24)

```
void controlwaypoint(double x,double y,double kp,double ki,double kd,double distance,double headingg,int statelanjut)
{
    errorx=0-x;
    integralk=integralk+errorx;
    derivativex=errorx-errorsebelumx;
    errory=0-y;
    integraly=integraly+error;
    derivatvey=error-errorsebelumy;
    controlrcwaypoint.pitch=(kp*error)+(ki*integralk)+(kd*derivativex);
    controlrcwaypoint.roll=(kp*errorx)+(ki*integralk)+(kd*derivativex);
    if(controlrcwaypoint.roll>800){controlrcwaypoint.roll=800;}
    else if (controlrcwaypoint.roll<(-800)){controlrcwaypoint.roll=-800;}
    if (controlrcwaypoint.pitch>800){controlrcwaypoint.pitch=800;}
    else if (controlrcwaypoint.pitch<(-800)){controlrcwaypoint.pitch=-800;}
    errorsebelumx=errorx;
    errorsebelumy=error;
    if(abs(distance)<2)
    {
        controlrcwaypoint.pitch=0;
        controlrcwaypoint.roll=0;
        switchauto=statelanjut;
    }
}
```

Gambar 3-35 Program Pergerakan Drone Leader dan Drone Follower

BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai hasil simulasi dan analisa yang terdiri dari beberapa bagian yaitu Pada bagian pertama membahas tentang konsumsi daya yang terdapat pada drone, bagian kedua membahas tentang kecepatan maksimal naik turun berdasarkan waktu, bagian ketiga membahas tentang kecepatan gerak maju dan mundur drone berdasarkan waktu, bagian keempat ialah test range koneksi router, bagian kelima ialah kecepatan naik dan turun antara drone leader dan drone follower, bagian keenam ialah kecepatan maju dan mundur antara drone leader dan follower dan yang terakhir ialah kecepatan menuju titik point antara drone leader dan drone follower.

4.1. Pengujian Konsumsi Daya

Pengujian Konsumsi Daya dilakukan guna untuk mengetahui ketahanan daya uji baterai pada saat drone terbang. peralatan yang digunakan untuk pengujian ini antara lain: Quadcopter, Baterai checker, remote dan Stopwatch HP. Pengujian dilakukan saat drone terbang dan dilakukan uji tahan baterai menggunakan stopwatch HP. Gambar 4.1 merupakan pengujian Konsumsi Daya Baterai dan Tabel 4.1 merupakan hasil pengujian dari Konsumsi Daya Baterai.



Gambar 4-1 Pengujian Konsumsi Daya Baterai Lipo

Tabel 4-1 Konsumsi Daya

NO	Pengujian	Tegangan (V)		Selisih tegangan (V)	Lama pemakaian (menit)
		Awal	Akhir		
1	Pengujian 1	12,50	11,80	0,70	1,5
2	Pengujian 2	11,80	11,20	0,60	1,3
3	Pengujian 3	12,60	11,74	0,86	2,3
4	Pengujian 4	11,74	11,17	0,57	1,1
5	Pengujian 5	12,34	11,39	0,95	2,4
6	Pengujian 6	12,60	11,94	0,56	1,4

Dari hasil data yang didapatkan maka semakin lama pemakaian baterai yang digunakan untuk terbang maka tegangan pada batteraipun akan habis sesuai dengan kebutuhan terbang yang dipakai. Baterai yang digunakan untuk quadcopter leader maupun follower menggunakan baterai dengan daya sebesar 1100 mah.

4.2. Pengujian Kecepatan Naik Turun berdasarkan Waktu

Pengujian Kecepatan Naik Turun dilakukan guna untuk mengetahui seberapa cepat drone terbang naik ke ketinggian tertentu. Pengujian membutuhkan beberapa alat diantaranya yaitu quadcopter, Stopwatch HP dan remote. Pengujian dilakukan dengan cara melihat waktu drone saat terbang ke ketinggian yang diinginkan. Gambar 4.2 merupakan pengujian kecepatan naik turun drone dan gambar 4.3 merupakan hasil grafik pengujian kecepatan naik dan untuk hasil grafik pengujian kecepatan turun drone terdapat pada gambar 4.4



Gambar 4-2 Pengujian Kecepatan Naik dan Turun Quadcopter

Tabel 4-2 Pengujian Kecepatan Naik Berdasarkan Waktu

No	Pengujian	Ketinggian (milimeter)	Waktu (detik)
1	Pengujian 1	500	6
2	Pengujian 2	1000	10
3	Pengujian 3	1500	12
4	Pengujian 4	2000	14
5	Pengujian 5	2500	15
6	Pengujian 6	3000	17
7	Pengujian 7	3500	19
8	Pengujian 8	4000	20
9	Pengujian 9	4500	22
10	Pengujian 10	5000	24

Dari hasil yang didapat tabel 4.2 waktu yang dibutuhkan pengujian kecepatan naik jika ketinggian yang dicapai quadcopter semakin jauh maka waktu yang dibutuhkan akan semakin lama.



Gambar 4-3 Grafik Kecepatan naik drone leader

Tabel 4-3 Pengujian Kecepatan Turun Berdasarkan Waktu

No	Pengujian	Ketinggian (milimeter)	Waktu (detik)
1	Pengujian 1	500	4
2	Pengujian 2	1000	8
3	Pengujian 3	1500	13
4	Pengujian 4	2000	15
5	Pengujian 5	2500	17
6	Pengujian 6	3000	19
7	Pengujian 7	3500	22
8	Pengujian 8	4000	25
9	Pengujian 9	4500	26
10	Pengujian 10	5000	28

Dari hasil yang didapat tabel 4.3 waktu yang dibutuhkan pengujian kecepatan Turun jika ketinggian yang dicapai quadcopter semakin jauh maka waktu yang dibutuhkan akan semakin lama.



Gambar 4-4 Grafik Kecepatan Turun Drone Leader

4.3. Pengujian Kecepatan Gerak Maju drone leader

Pengujian Kecepatan Gerak Maju Drone Leader dilakukan guna untuk mengetahui seberapa cepat drone leader maju untuk mencapai jarak yang diinginkan. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: stopwatch, remote dan stopwatch HP. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader maju mencapai jarak yang berbeda. gambar 4.5 merupakan pengujian kecepatan maju drone leader dan gambar 4.6 merupakan hasil grafik pengujian kecepatan maju drone leader.



Gambar 4-5 Pengujian Kecepatan Maju Quadcopter

Tabel 4-4 Pengujian Kecepatan Maju Berdasarkan Waktu

No	Pengujian	Jarak (meter)	Waktu (detik)
1	Pengujian 1	0.5	3
2	Pengujian 2	1	5
3	Pengujian 3	1.5	6
4	Pengujian 4	2	9
5	Pengujian 5	2.5	12
6	Pengujian 6	3	15
7	Pengujian 7	3.5	16
8	Pengujian 8	4	18
9	Pengujian 9	4.5	23
10	Pengujian 10	5	25

Dari hasil tabel 4.4 adalah jika jarak yang ditempuh semakin jauh maka waktu yang dibutuhkan quadcopter semakin membutuhkan waktu yang lama untuk mencapai ketujuan jarak yang diinginkan.



Gambar 4-6 Grafik Kecepatan Maju Drone Leader

4.4. Pengujian Kecepatan Gerak Mundur drone leader

Pengujian Kecepatan Gerak Mundur Drone Leader dilakukan guna untuk mengetahui seberapa cepat drone leader mundur untuk mencapai jarak yang diinginkan. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: stopwatch, remote dan stopwatch HP. Pengujian

dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader mundur mencapai jarak yang berbeda. gambar 4.7 merupakan pengujian kecepatan mundur drone leader dan gambar 4.8 merupakan hasil grafik pengujian kecepatan maju drone leader.



Gambar 4-7 Pengujian Kecepatan Mundur quadcopter

Tabel 4-5 Pengujian Kecepatan Mundur Berdasarkan Waktu

No	Pengujian	Jarak (meter)	Waktu (detik)
1	Pengujian 1	0.5	3
2	Pengujian 2	1	5
3	Pengujian 3	1.5	6
4	Pengujian 4	2	9
5	Pengujian 5	2.5	12
6	Pengujian 6	3	15
7	Pengujian 7	3.5	16

No	Pengujian	Jarak (meter)	Waktu (detik)
8	Pengujian 8	4	18
9	Pengujian 9	4.5	23
10	Pengujian 10	5	25

Dari hasil tabel 4.5 pengujian kecepatan mundur pada quadcopter leader akan semakin cepat jika jarak yang ditempuh pada quadcopter semakin pendek sehingga membutuhkan waktu yang cepat



Gambar 4-8 Grafik Kecepatan Mundur Drone Leader

4.5. Pengujian Koneksi Router

Pengujian Koneksi Router dilakukan guna untuk mengetahui seberapa jauh router dapat memberikan koneksi sinyal wifi ke drone leader dan drone follower. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: Router dan HP. Pengujian dilakukan dengan menguji koneksi router menggunakan hp guna untuk mengetahui seberapa jauh koneksi antara drone leader dan drone follower. gambar 4.9 merupakan pengujian koneksi router dan tabel 4.6 merupakan hasil pengujian koneksi router.



Gambar 4-9 Pengujian Koneksi Router

Tabel 4-6 Pengujian Koneksi Router

No	Pengujian	Jarak (meter)	Hasil
1	Pengujian 1	1	Konek
2	Pengujian 2	2	konek
3	Pengujian 3	3	konek
4	Pengujian 4	4	konek
5	Pengujian 5	5	konek
6	Pengujian 6	6	konek
7	Pengujian 7	7	konek
8	Pengujian 8	8	konek
9	Pengujian 9	9	Konek
10	Pengujian 10	10	Tidak konek

Dari hasil tabel 4.6 pengujian akan mengalami masalah pada jarak lebih dari 10 meter sehingga untuk penerbangan antara quadcopter leader dan follower tidak bisa lebih dari 10 meter.

4.6. Pengujian Kecepatan Drone Leader dan Follower Naik

Pengujian Kecepatan Gerak Naik Drone Leader dan Drone Follower dilakukan guna untuk mengetahui seberapa cepat drone leader dan drone follower naik ke ketinggian yang diinginkan saat naik. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: drone leader, 2 drone follower, remote, router dan HP. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader dan follower naik mencapai ke ketinggian yang diinginkan .gambar 4.10 merupakan

pengujian drone leader dan drone follower saat naik gambar 4.11 merupakan hasil grafik pengujian drone leader dan drone follower saat naik.



Gambar 4-10 Pengujian Drone Leader dan Follower Saat Naik

Tabel 4-7 Pengujian Pengiriman antara Drone Leader dan Follower Saat Naik Berdasarkan Waktu

No	Pengujian	Ketinggian (milimeter)	Waktu (detik)
1	Pengujian 1	500	6
2	Pengujian 2	1000	12
3	Pengujian 3	1500	14
4	Pengujian 4	2000	15
5	Pengujian 5	2500	16
6	Pengujian 6	3000	18
7	Pengujian 7	3500	19
8	Pengujian 8	4000	22
9	Pengujian 9	4500	24
10	Pengujian 10	5000	25

Dari hasil tabel 4.7 dapat disimpulkan bahwa semakin jauh ketinggian yang ditempuh maka waktu kecepatan drone leader dan drone

follower untuk ke tujuan ketinggian tersebut akan membutuhkan waktu yang semakin lama.



Gambar 4-11 Grafik Kecepatan Naik Drone Leader dan Drone follower

4.7. Pengujian Kecepatan Drone Leader dan Follower Turun

Pengujian Kecepatan Gerak Turun Drone Leader dan Drone Follower dilakukan guna untuk mengetahui seberapa cepat drone leader dan drone follower Turun dari ketinggian yang diinginkan. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: drone leader, 2 drone follower, remote, router dan HP. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader dan follower turun dari ketinggian yang diinginkan. Gambar 4.12 merupakan pengujian kecepatan drone leader dan drone follower turun dan gambar 4.13 merupakan hasil grafik pengujian kecepatan drone leader dan drone follower saat turun.



Gambar 4-12 Pengujian Drone Leader dan Follower Saat Turun

Tabel 4-8 Pengujian Pengiriman antara Drone Leader dan Follower Saat Turun Berdasarkan Waktu

No	Pengujian	Ketinggian (milimeter)	Waktu (detik)
1	Pengujian 1	500	4
2	Pengujian 2	1000	8
3	Pengujian 3	1500	12
4	Pengujian 4	2000	15
5	Pengujian 5	2500	17
6	Pengujian 6	3000	19
7	Pengujian 7	3500	20
8	Pengujian 8	4000	22
9	Pengujian 9	4500	24
10	Pengujian 10	5000	25

Dari hasil tabel 4.8 dapat disimpulkan bahwa pengujian kecepatan drone leader dan follower akan semakin lama jika jarak ketinggian yang ditempuh semakin tinggi.



Gambar 4-13 Grafik Kecepatan Turun Drone Leader dan Drone follower

4.8. Pengujian Kecepatan Drone Leader dan Follower Maju

Pengujian Kecepatan Gerak Maju Drone Leader dan Drone Follower dilakukan guna untuk mengetahui seberapa cepat drone leader dan drone follower maju mencapai jarak yang diinginkan. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: drone leader, 2 drone follower, remote, router dan HP. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader dan follower saat maju mencapai jarak yang diinginkan. gambar 4.14 merupakan pengujian kecepatan drone leader dan drone follower saat maju mencapai jarak yang diinginkan dan gambar 4.15 merupakan hasil grafik pengujian kecepatan drone leader dan drone follower saat maju mencapai jarak yang diinginkan.

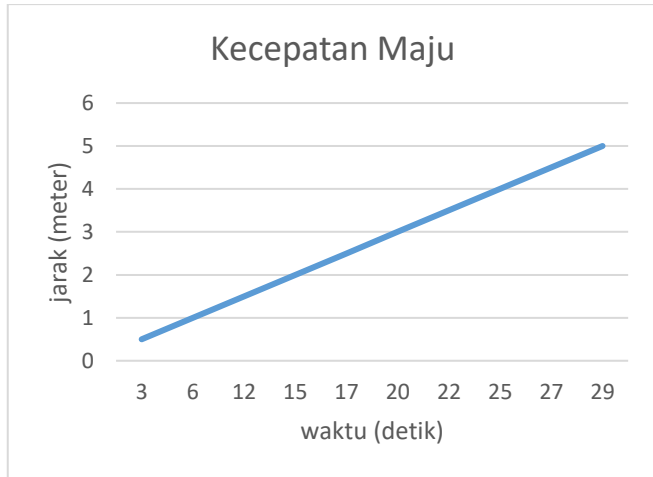


Gambar 4-14 Pengujian Pengiriman Saat Drone Leader dan Follower maju

Tabel 4-9 Pengujian Pengiriman antara Drone Leader dan Follower Saat Maju Berdasarkan Waktu

No	Pengujian	Jarak (meter)	Waktu (detik)
1	Pengujian 1	0,5	7
2	Pengujian 2	1	10
3	Pengujian 3	1,5	13
4	Pengujian 4	2	16
5	Pengujian 5	2,5	19
6	Pengujian 6	3	20
7	Pengujian 7	3,5	22
8	Pengujian 8	4	23
9	Pengujian 9	4,5	24
10	Pengujian 10	5	26

Dari hasil tabel 4.9 pengujian drone leader dan follower akan cepat jika jarak maju yang ditempuh semakin pendek dan jika jarak tempuh semakin panjang maka membutuhkan waktu yang semakin lama



Gambar 4-15 Grafik Kecepatan Maju Drone Leader dan Drone follower

4.9. Pengujian Kecepatan Drone Leader dan Follower Mundur

Pengujian Kecepatan Gerak Mundur Drone Leader dan Drone Follower dilakukan guna untuk mengetahui seberapa cepat drone leader dan drone follower mundur mencapai jarak yang diinginkan. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: drone leader, 2 drone follower, remote, router dan HP. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader dan follower saat mundur mencapai jarak yang diinginkan. gambar 4.16 merupakan pengujian kecepatan drone leader dan drone follower saat mundur mencapai jarak yang diinginkan dan gambar 4.17 merupakan hasil grafik pengujian kecepatan drone leader dan drone follower saat mundur mencapai jarak yang diinginkan.



Gambar 4-16 Pengujian drone leader dan drone follower saat mundur

Tabel 4-10 Pengujian Pengiriman antara Drone Leader dan Follower Saat Mundur Berdasarkan Waktu

No	Pengujian	Jarak (meter)	Waktu (detik)
1	Pengujian 1	0.5	9
2	Pengujian 2	1	11
3	Pengujian 3	1.5	12
4	Pengujian 4	2	14
5	Pengujian 5	2.5	16
6	Pengujian 6	3	17
7	Pengujian 7	3.5	23
8	Pengujian 8	4	24
9	Pengujian 9	4.5	25
10	Pengujian 10	5	27

Dari hasil tabel 4.10 pengujian drone leader dan follower akan cepat jika jarak mundur yang ditempuh semakin pendek dan jika jarak tempuh semakin panjang maka membutuhkan waktu yang semakin lama



Gambar 4-17 Grafik Kecepatan Mundur Drone Leader dan Drone Follower

4.10. Pengujian Kecepatan Drone Leader dan Follower menuju titik point

Pengujian Kecepatan Gerak menuju titik point drone leader dan drone Follower dilakukan guna untuk mengetahui seberapa cepat drone leader dan drone follower mencapai tujuan yang diinginkan. Pengujian ini dilakukan dengan beberapa alat yang dibutuhkan antara lain: drone leader, 2 drone follower, remote, router dan HP. Pengujian dilakukan dengan menghitung waktu yang dibutuhkan saat drone leader dan follower mencapai tujuan yang diinginkan .gambar 4.18 merupakan pengujian kecepatan drone leader dan drone follower saat mencapai tujuan yang diinginkan dan gambar 4.19 merupakan hasil grafik pengujian kecepatan drone leader dan drone follower saat mencapai tujuan yang diinginkan.



Gambar 4-18 Pengujian drone leader dan follower menuju titik point

Tabel 4-11 Pengujian Pengiriman antara Drone Leader dan Follower Saat Menuju Titik Point Berdasarkan Waktu

No	Pengujian	Jarak (meter)	Waktu
1	Pengujian 1	0.5	3
2	Pengujian 2	1	6
3	Pengujian 3	1.5	12
4	Pengujian 4	2	15
5	Pengujian 5	2.5	17
6	Pengujian 6	3	20
7	Pengujian 7	3.5	22
8	Pengujian 8	4	25
9	Pengujian 9	4.5	27
10	Pengujian 10	5	29

Dari hasil tabel 4.11 pengujian drone leader dan follower akan cepat jika jarak yang ditempuh semakin pendek dan jika jarak tempuh semakin panjang maka membutuhkan waktu yang semakin lama



Gambar 4-19 Grafik Kecepatan Menuju Titik Waypoint Drone Leader dan Drone Follower

[Halaman ini sengaja dikosongkan]

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil Pengujian yang dilakukan bab 4 maka dapat disimpulkan bahwa:

1. Waktu yang dibutuhkan drone leader untuk drone leader naik,turun,maju dan mundur akan sangat cepat apabila jarak yang ditempuh sangat dekat
2. Dibandingkan dengan drone leader saja drone follower membutuhkan waktu sedikit lebih lama untuk komunikasi tetapi waktu yang dibutuhkan drone leader atau drone follower akan sangat cepat bila jarak yang ditempuh tidak jauh
3. Komunikasi antara drone leader dan drone follower akan mengalami gangguan jika jarak tempuh antara drone leader dan drone follower dengan wifi router lebih dari 10 meter.
4. Semakin lama waktu pemakaian battery maka semakin cepat pula daya habis pada batterai pada drone leader maupun drone follower dan waktu maksimal penggunaan drone leader maupun drone follower ialah 10 menit.
5. Hasil kecepatan didapat dari pembagian antara jarak dan waktu sehingga didapatkan suatu hasil antara lain:drone leader dan follower dapat mencapai tujuan yang diinginkan dengan kecepatan rata-rata untuk naik adalah 0.1483588 m/s dan kecepatan rata-rata untuk mundur adalah 0.1557605 m/s sedangkan untuk kecepatan rata-rata maju untuk drone dapat berkoordinasi mencapai suatu titik adalah 0.1406205 m/s dan kecepatan rata-rata mundur untuk drone dapat berkoordinasi mencapai suatu titik adalah 0.1431069 m/s.

5.2. Saran

Saran dari penulis untuk pengembangan tugas akhir ini ialah dapat dikembangkan untuk sarana pengintaian maupun dapat dijadikan untuk bahan entertainment menggunakan pembentukan informasi/path planning antar drone leader dan drone follower.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] G. Varela, P. Caamamo, F. Orjales, A. Deibe, F. Lopez-Pena, and R. J. Duro, "Swarm intelligence based approach for real time UAV team coordination in search operations," in *2011 Third World Congress on Nature and Biologically Inspired Computing*, Salamanca, Spain, 2011, pp. 365–370.
- [2] T. Zeng, M. Mozaffari, O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Wireless Communications and Control for Swarms of Cellular-Connected UAVs," *ArXiv181200743 Eess*, Dec. 2018.
- [3] I. Gaponov and A. Razinkova, "Quadcopter design and implementation as a multidisciplinary engineering course," in *Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012*, Hong Kong, China, 2012, pp. H2B-16-H2B-19.
- [4] R. L. McClanahan, "Improving Unmanned Aerial Vehicle Formation Flight and Swarm Cohesion by Using Commercial Off the Shelf Sonar Sensors," p. 98.
- [5] B. Henrivaux, "Master thesis : Integration of an Open Source Flight Controller into a Fixed Wing Remotely Piloted Aircraft," p. 113.
- [6] C. A. Amadi, "Design and Implementation of Model Predictive Control on Pixhawk Flight Controller," p. 135.
- [7] M. H. A. Hamid, A. H. Adom, N. A. Rahim, and M. H. F. Rahiman, "Navigation of mobile robot using Global Positioning System (GPS) and obstacle avoidance system with commanded loop daisy chaining application method," in *2009 5th International Colloquium on Signal Processing & Its Applications*, Kuala Lumpur, Malaysia, 2009, pp. 176–181.
- [8] T. Buchegger, G. Ossberger, A. Reiszahn, A. Stelzer, and A. Springer, "Pulse delay techniques for PPM impulse radio transmitters," in *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, Reston, Virginia, USA, 2003, pp. 37–41.
- [9] N. M. Zimmerman, "Flight Control and Hardware Design of Multi-Rotor Systems," p. 123.
- [10] S. Atoev, K.-R. Kwon, S.-H. Lee, and K.-S. Moon, "Data analysis of the MAVLink communication protocol," in *2017 International Conference on Information Science and Communications Technologies (ICISCT)*, Tashkent, 2017, pp. 1–3.
- [11] M. Bangura, "Aerodynamics and Control of Quadrotors," p. 215.

- [12] B. Tefay, B. Eizad, P. Crosthwaite, S. Singh, and A. Postula, "Design of an Integrated Electronic Speed Controller for Compact Robotic Vehicles," p. 9, 2011.
- [13] I. Ghazi and M. R. Maqbool, "GPS Based Autonomous Vehicle Navigation and Control System," p. 7.
- [14] R. K. Chauhan, B. S. Rajpurohit, R. E. Hebner, S. N. Singh, and F. M. G. Longatt, "Design and analysis of PID and fuzzy-PID controller for voltage control of DC microgrid," in *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, Bangkok, Thailand, 2015, pp. 1–6.
- [15] P. Chandhar, D. Danev, and E. G. Larsson, "Massive MIMO for Communications with Drone Swarms," *ArXiv170701039 Cs Math*, Jul. 2017.
- [16] F. Fabra, C. T. Calafate, J.-C. Cano, and P. Manzoni, "On the impact of inter-UAV communications interference in the 2.4 GHz band," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, Spain, 2017, pp. 945–950.

LAMPIRAN

PROGRAM:

```
#include <stdio.h>
#include <string.h>
#include <CPPM.h>
#include <Arduino.h>
#include <math.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#define RADIUS 6371000
#define address 0x1E //0011110b, I2C 7bit address of HMC5883
#define Declination -0.00669
#define pi 3.1415926535897932384626433832795028841
```

```
SoftwareSerial tonodemcu(3, 2);//RXTX
```

```
unsigned long timer1 = 0;
void tahanthrottle();
//ch1=roll
//ch2=pitch
//ch3=throttle
//ch4=yaw
```

```
typedef struct GPSmavlink
{
    char datagps[50];
}GPS;
```

```
typedef struct dataremotearduinooppm
{
    int aile;
    int elev;
    int thro;
    int rudd;
    int gear;
    int aux1;
```

```
int aux2;  
int aux3;  
}dataremoteppm;
```

```
typedef struct controlremotecompass  
{  
int pitch;  
int roll;  
int throt;  
int rudd;  
}controlremotecompass;
```

```
typedef struct controlremotewithoutcompass  
{  
int pitch;  
int roll;  
int throt;  
int rudd;  
}controlremotewithoutcompass;
```

```
typedef struct controlremotewaypoint  
{  
int pitch;  
int roll;  
int throt;  
int rudd;  
}controlremotewaypoint;
```

```
typedef struct convertgps  
{  
float longitude;  
float latitude;  
}convertgps;
```



```

void setoutputpwm(int ch1,int ch2,int ch3,int ch4,int ch5,int ch6,int
ch7,int ch8)
{
  CPPM.oservos[0]=ch1+3000;
  CPPM.oservos[1]=ch2+3000;
  CPPM.oservos[2]=ch3+3000;
  CPPM.oservos[3]=ch4+3000;
  CPPM.oservos[4]=ch5;
  CPPM.oservos[5]=ch6;
  CPPM.oservos[6]=ch7;
  CPPM.oservos[7]=ch8;
  // Serial.flush();
}

```

```

void setoutputpwpid(int ch1,int ch2,int ch3,int ch4,int ch5,int ch6,int
ch7,int ch8)
{
  CPPM.oservos[0]=ch1+3000;
  CPPM.oservos[1]=ch2+3000;
  // CPPM.oservos[2]=ch3+3000;
  CPPM.oservos[3]=ch4+3000;
  CPPM.oservos[4]=ch5;
  CPPM.oservos[5]=ch6;
  CPPM.oservos[6]=ch7;
  CPPM.oservos[7]=ch8;
  // tahanthrottle();
  // Serial.flush();
}

```

```

double costodegree(double x)
{
  double ret, val;
  val = PI / 180.0;
  ret = cos( x*val );

  return ret;
}

```

```

}

double sintodegree(double x)
{
    double ret, val;
    val = PI / 180.0;
    ret = sin( x*val );

    return ret;
}
dataremoteppm ppm;
void controlremotenormally()
{
    ppm.aile=CPPM.read(CPPM_AILE)-3000;
    ppm.elev = CPPM.read(CPPM_ELEV)-3000;
    ppm.thro = CPPM.read(CPPM_THRO)-3000;
    ppm.rudd = CPPM.read(CPPM_RUDD)-3000;
    ppm.gear=CPPM.read(CPPM_GEAR) ;
    ppm.aux1=CPPM.read(CPPM_AUX1) ;
    ppm.aux2 = CPPM.read(CPPM_AUX2);
    ppm.aux3 = CPPM.read(CPPM_AUX3) ;

}

char inputangka[50];
controlremotewaypoint controlrcwaypoint;
double errorx,integralx,derivativex,errorsebelumx;
double errory,integraly,derivatvey,errorsebelumy;
double erroralt,integralalt,derivativealt,errorsebelumalt;
double
errorheading,integralheading,derivativeheading,errorsebelumheading;
int statemisi;
int gantichannel;
int switchauto=0;

```

```

void controlwaypoint(double x,double y,double kp,double ki,double
kd,double distance,double headingg,int statelanjut)
{
    errorx=0-x;
    integralx=integralx+errorx;
    derivativex=errorx-errorsebelumx;
    errory=0-y;
    integraly=integraly+error;
    derivatvey=error-errorsebelumy;

controlrcwaypoint.pitch=(kp*error)+(ki*integraly)+(kd*derivatvey);
    controlrcwaypoint.roll=(kp*errorx)+(ki*integralx)+(kd*derivativex);
    if(controlrcwaypoint.roll>800){controlrcwaypoint.roll=800;}
    else if (controlrcwaypoint.roll<(-800)){controlrcwaypoint.roll=-800;}
    if (controlrcwaypoint.pitch>800){controlrcwaypoint.pitch=800;}
    else if (controlrcwaypoint.pitch<(-800)){controlrcwaypoint.pitch=-
800;}
    errorsebelumx=errorx;
    errorsebelumy=error;
    if(abs(distance)<2)
    {
        controlrcwaypoint.pitch=0;
        controlrcwaypoint.roll=0;
        switchauto=statelanjut;
    }
}

```

```

int trig= 6;          // membuat variabel trig yang di set ke-pin 3
int echo= 7;         // membuat variabel echo yang di set ke-pin 2
long durasi, jarak; // membuat variabel durasi dan jarak

```

```

int jarak(){
digitalWrite(trig, LOW);

```

```
delayMicroseconds(8);
digitalWrite(trig, HIGH);
delayMicroseconds(8);
digitalWrite(trig, LOW);
delayMicroseconds(8);
```

```
durasi= pulseIn(echo, HIGH);
jarakk= (durasi/2) / 29.1;
```

```
return jarakk;
```

```
}
```

```
void controlalt(double kp,double ki,double kd,double alt,double
setpoint){
    erroralt=alt-setpoint;
    integralalt=integralalt+erroralt;
    derivativealt=erroralt-errorsebelumalt;
```

```
controlrcwaypoint.throt=(kp*erroralt)+(ki*integralalt)+(kd*derivativealt
);
```

```
    if(controlrcwaypoint.throt>800){controlrcwaypoint.throt=800;}
    else if (controlrcwaypoint.throt<(-800)){controlrcwaypoint.throt=-
800;}
```

```
    errorsebelumalt=erroralt;
```

```
}
```

```
controlremotecompass remotecompass;
void controlremotewithcompasspid(double ch1,double ch2, double
ch3,double ch4,double sudut,double suduttujuan)
{
```

```

remotecompass.roll=ch1*costodegree(sudut-suduttujuan)-
ch2*sintodegree(sudut-suduttujuan);
remotecompass.pitch=ch2*costodegree(sudut-
suduttujuan)+ch1*sintodegree(sudut-suduttujuan);
remotecompass.throt=ch3;
remotecompass.rudd =ch4;

```

```

setoutputpwmpid(remotecompass.roll,remotecompass.pitch,remotecompass.throt,
remotecompass.rudd ,CPPM.read(CPPM_GEAR),CPPM.read(CPPM_AUX1),CPPM.read(CPPM_AUX2),CPPM.read(CPPM_AUX3));
}

```

```

controlremotewithoutcompass rcnocompass;
void rcnocompass(int ch1,int ch2, int ch3,int ch4,int ch5,int ch6 ,int ch7,int ch8)

```

```

{
rcnocompass.roll=ch1;
rcnocompass.pitch=ch2;
rcnocompass.throt=ch3;
rcnocompass.rudd=ch4;

```

```

setoutputpwm(rcnocompass.roll,rcnocompass.pitch,rcnocompass.throt,rcnocompass.rudd,CPPM.read(CPPM_GEAR),CPPM.read(CPPM_AUX1),CPPM.read(CPPM_AUX2),CPPM.read(CPPM_AUX3));
}

```

```

char dataterimamavlink;
int penandags=0;
int gpsmasuk=0;
int datamin=-1;
int dataminalt=-1;
GPS gps;

```

```

float headingHMCL(){
  int x,y,z;
  int xmin,xmax,ymin,ymax,zmin,zmax;
  xmin=0; xmax=0; ymax=0; ymin = 0; zmin=0;zmax=0;
  Wire.beginTransmission(address);
  Wire.write(0x03);
  Wire.endTransmission();
  Wire.requestFrom(address, 6);
  if(Wire.available()<=6){
    x = Wire.read()<<8| Wire.read(); //X msb
    z = Wire.read()<<8| Wire.read(); //Z msb
    y = Wire.read()<<8| Wire.read(); //Y msb
  }
  float heading=atan2(y, x)+ Declination;
  return (heading* 180 / PI);
}

```

```
int misi=0;
```

```

void serialEvent() {
while (Serial.available()>0)
{
  dataterimamavlink=Serial.read();
  if(penandagps==0)
  {
    if(dataterimamavlink=='A')penandagps=1;
    else penandagps=0;
  }

  else if(penandagps==1)
  {
    switch(gpsmasuk)
    {
      case 0:if(dataterimamavlink=='-'){ gps.datagps[0]=-1;datamin=-1;gpsmasuk=1;}

```

```

else
if(dataterimamavlink=='+'){ gps.datagps[0]=1;datamin=1;gpsmasuk=1;}
    else { penandagps=0;gpsmasuk=0;}
    break;
    case 1:gps.datagps[1]=dataterimamavlink;gpsmasuk=2;break;
    case 2:gps.datagps[2]=dataterimamavlink;gpsmasuk=3;break;
    case 3:gps.datagps[3]=dataterimamavlink;gpsmasuk=4; break;
    case 4:gps.datagps[4]=dataterimamavlink;gpsmasuk=5; break;
    case 5:gps.datagps[5]=dataterimamavlink;gpsmasuk=6; break;
    case 6:gps.datagps[6]=dataterimamavlink;gpsmasuk=7; break;
    case 7:gps.datagps[7]=dataterimamavlink;gpsmasuk=8; break;
    case 8:gps.datagps[8]=dataterimamavlink;gpsmasuk=9; break;
    case 9:if(dataterimamavlink=='B')gpsmasuk=10;
        else {gpsmasuk=0;penandagps=0;}
    break;
    case 10:gps.datagps[9]=dataterimamavlink;gpsmasuk=11; break;
    case 11:gps.datagps[10]=dataterimamavlink;gpsmasuk=12; break;
    case 12:gps.datagps[11]=dataterimamavlink;gpsmasuk=13; break;
    case 13:gps.datagps[12]=dataterimamavlink;gpsmasuk=14; break;
    case 14:gps.datagps[13]=dataterimamavlink;gpsmasuk=15; break;
    case 15:gps.datagps[14]=dataterimamavlink;gpsmasuk=16; break;
    case 16:gps.datagps[15]=dataterimamavlink;gpsmasuk=17; break;
    case 17:gps.datagps[16]=dataterimamavlink;gpsmasuk=18; break;
    case 18:gps.datagps[17]=dataterimamavlink;gpsmasuk=19; break;
    case 19:gps.datagps[18]=dataterimamavlink;gpsmasuk=20; break;
    case 20:
    if(dataterimamavlink=='C'){gpsmasuk=21;}
        else {gpsmasuk=0;penandagps=0;}
    break;
    //////////////////////////////////////
    case 21:if(dataterimamavlink=='m')
    {
        gpsmasuk=22;
    }
    else if(dataterimamavlink=='p')
    {

```

```

    gpsmasuk=45;
}
break;
/////angka[0]=tanda positif/negatif,angka[1]=ratusan          ribu,
angka[2]=puluhan          ribu,          angka[3]=ribuan,
angka[4]=ratusan,angka[5]=puluhan,angka[6]=satuan/////
case 22:
if(dataterimamavlink=='f')
{
    gpsmasuk=41;
}
else if(dataterimamavlink=='g')
{
    gpsmasuk=23;
}
else if(dataterimamavlink=='h')
{
    gpsmasuk=28;
}
else if(dataterimamavlink=='i')
{
    gpsmasuk=34;
}
break;
/////////minusribuan//////////
case          23:inputangka[0]=dataterimamavlink;dataminalt=-
1;gpsmasuk=24;break;
case
24:inputangka[1]='0';inputangka[2]='0';inputangka[3]=dataterimamavlin
k;gpsmasuk=25;break;
case 25:inputangka[4]=dataterimamavlink;gpsmasuk=26;break;
case 26:inputangka[5]=dataterimamavlink;gpsmasuk=27;break;
case
27:inputangka[6]=dataterimamavlink;gpsmasuk=0;penandagps=0;break;
/////////minuspuluhanribu//////////

```



```

    case                28:inputangka[0]=dataterimavlink;dataminalt=-
1;gpsmasuk=29;break;
    case
29:inputangka[1]='0';inputangka[2]=dataterimavlink;gpsmasuk=30;b
reak;
    case 30:inputangka[3]=dataterimavlink;gpsmasuk=31;break;
    case 31:inputangka[4]=dataterimavlink;gpsmasuk=32;break;
    case 32:inputangka[5]=dataterimavlink;gpsmasuk=33;break;
    case
33:inputangka[6]=dataterimavlink;gpsmasuk=0;penandagps=0;break;
    ///////minusratusanribu/////
    case                34:inputangka[0]=dataterimavlink;dataminalt=-
1;gpsmasuk=35;break;
    case 35:inputangka[1]=dataterimavlink;gpsmasuk=36;break;
    case 36:inputangka[2]=dataterimavlink;gpsmasuk=37;break;
    case 37:inputangka[3]=dataterimavlink;gpsmasuk=38;break;
    case 38:inputangka[4]=dataterimavlink;gpsmasuk=39;break;
    case 39:inputangka[5]=dataterimavlink;gpsmasuk=40;break;
    case
40:inputangka[6]=dataterimavlink;gpsmasuk=0;penandagps=0;break;
    ///////ratusan/////
    case                41:inputangka[0]=dataterimavlink;dataminalt=-
1;gpsmasuk=42;break;
    case
42:inputangka[1]='0';inputangka[2]='0';inputangka[3]='0';inputangka[4]
=dataterimavlink;gpsmasuk=43;break;
    case 43:inputangka[5]=dataterimavlink;gpsmasuk=44;break;
    case
44:inputangka[6]=dataterimavlink;gpsmasuk=0;penandagps=0;break;
    ///////angka[0]=tandapositifnegatif,angka[1]=ratusan        ribu,
angka[2]=puluhan        ribu,        angka[3]=ribuan,
angka[4]=ratusan,angka[5]=puluhan,angka[6]=satuan/////
    case 45:if(dataterimavlink=='f')
    {
        gpsmasuk=61;
    }

```

```

else if(dataterimamavlink=='g')
{
    gpsmasuk=46;
}
else if(dataterimamavlink=='h')
{
    gpsmasuk=50;
}
else if(dataterimamavlink=='i')
{
    gpsmasuk=55;
}
break;
////////ribuan////////
case
46:inputangka[0]='+';inputangka[1]='0';inputangka[2]='0';inputangka[3]
=dataterimamavlink;dataminalt=1;gpsmasuk=47;break;
    case 47:inputangka[4]=dataterimamavlink;gpsmasuk=48;break;
    case 48:inputangka[5]=dataterimamavlink;gpsmasuk=49;break;
    case
49:inputangka[6]=dataterimamavlink;gpsmasuk=0;penandagps=0;break;
    //////////puluhanribu////////
    case
50:inputangka[0]='+';inputangka[1]='0';inputangka[2]=dataterimamavlin
k;dataminalt=1;gpsmasuk=51;break;
    case 51:inputangka[3]=dataterimamavlink;gpsmasuk=52;break;
    case 52:inputangka[4]=dataterimamavlink;gpsmasuk=53;break;
    case 53:inputangka[5]=dataterimamavlink;gpsmasuk=54;break;
    case
54:inputangka[6]=dataterimamavlink;gpsmasuk=0;penandagps=0;break;
    //////////43//ratusanribu////////
    case
55:inputangka[0]='+';inputangka[1]=dataterimamavlink;dataminalt=1;gp
smasuk=56;break;
    case 56:inputangka[2]=dataterimamavlink;gpsmasuk=57;break;
    case 57:inputangka[3]=dataterimamavlink;gpsmasuk=58;break;

```

```

    case 58:inputangka[4]=dataterimamavlink;gpsmasuk=59;break;
    case 59:inputangka[5]=dataterimamavlink;gpsmasuk=60;break;
    case
60:inputangka[6]=dataterimamavlink;gpsmasuk=0;penandagps=0;break;
    //ratusan////////
    case
61:inputangka[0]='+';inputangka[1]='0';inputangka[2]='0';inputangka[3]
='0';inputangka[4]=dataterimamavlink;dataminalt=1;gpsmasuk=62;brea
k;
    case 62:inputangka[5]=dataterimamavlink;gpsmasuk=63;break;
    case
63:inputangka[6]=dataterimamavlink;gpsmasuk=0;penandagps=0;break;
    ///////////////////////////////////////////////////
    }
    }
    }
    }
}

```

```

double cobaangkaa;
double convertgpstoalt()
{
    cobaangkaa=(((inputangka[1]-'0')*100000)+((inputangka[2]-
'0')*10000)+((inputangka[3]-'0')*1000)+((inputangka[4]-
'0')*100)+((inputangka[5]-'0')*10)+((inputangka[6]-'0')*1)))
    *dataminalt;
    return cobaangkaa;
}

```

```

double convertgpstolat(char *datagps)
{
    convertgps gps;
    gps.latitude=((datagps[1]-'0')*1+(datagps[2]-'0')*0.1+(datagps[3]-
'0')*0.01+(datagps[4]-'0')*0.001+(datagps[5]-'0')*0.0001+(datagps[6]-
'0')*0.00001+(datagps[7]-'0')*0.000001+(datagps[8]-
'0')*0.0000001)*datamin;
    return gps.latitude;
}

```

```
}
```

```
double convertgpstolongitu(char *datagps)
```

```
{  
    convertgps gps;  
    gps.longitude=(datagps[9]-'0')*100+(datagps[10]-'0')*10+(datagps[11]-  
'0')*1+(datagps[12]-'0')*0.1+(datagps[13]-'0')*0.01+(datagps[14]-  
'0')*0.001+(datagps[15]-'0')*0.0001+(datagps[16]-  
'0')*0.00001+(datagps[17]-'0')*0.000001+(datagps[18]-'0')*0.0000001;  
    return gps.longitude;  
}
```

```
float bearing(float lat1, float long1, float lat2, float long2) {
```

```
    float dlon = radians(long2-long1);  
    lat1 = radians(lat1);  
    lat2 = radians(lat2);  
    float a1 = sin(dlon) * cos(lat2);  
    float a2 = sin(lat1) * cos(lat2) * cos(dlon);  
    a2 = cos(lat1) * sin(lat2) - a2;  
    a2 = atan2(a1, a2);
```

```
    return degrees(a2);
```

```
}
```

```
int gantithrottle=0;
```

```
int a=0;
```

```
/////cobakoneksiheartbetmavlink/////
```

```
void ulang(unsigned long rate,int switchganti,int detik){
```

```
    if ((millis() - timer1) > rate) {
```

```
        timer1 = millis();
```

```
        a++;
```

```
        if(a>detik){
```

```
            a=0;
```

```
            gantithrottle=switchganti;
```

```
        }
```

```
    } else
```

```

    {
        CPPM.oservos[2]=3812;
//      Serial.println("naik");
    }

}

}

void tahanthrottle(){
    switch(gantithrottle){
        case 0:ulang(1000,1,2);    break;
        case 1:CPPM.oservos[2]=3040; break;
    }
}

float calc_dist(float flat1, float flon1, float flat2, float flon2)
{
    float dist_calc=0;
    float dist_calc2=0;
    float diflat=0;
    float diflon=0;

    diflat=radians(flat2-flat1);
    flat1=radians(flat1);
    flat2=radians(flat2);
    diflon=radians((flon2)-(flon1));

    dist_calc = (sin(diflat/2.0)*sin(diflat/2.0));
    dist_calc2= cos(flat1);
    dist_calc2*=cos(flat2);
    dist_calc2*=sin(diflon/2.0);
    dist_calc2*=sin(diflon/2.0);
    dist_calc +=dist_calc2;
}

```



```

        case
        controlwaypoint(posisix1,(posisiy1),30,0,5,distance1,sudutpesawat,0);
        controlremotewithcompasspid(controlrcwaypoint.roll,controlrcwaypoint
        .pitch, controlrcwaypoint.throt , ppm.rudd,sudutpesawat,sudutTujuan1);
        break;
    }
}

```

```

void missi1(double posisix,double posisiy,double distance,double
sudutTujuan){
    CPPM.oservos[6]=ppm.aux1;
    controlremotenormally();
    controlwaypoint(posisix,(posisiy),30,0,5,distance,sudutpesawat,2);
    tahanthrottle();
}

```

```

controlremotewithcompasspid(controlrcwaypoint.roll,controlrcwaypoint
.pitch, controlrcwaypoint.throt , ppm.rudd,sudutpesawat,sudutTujuan) ;
if(ppm.gear>3800)
{
    CPPM.oservos[0]=3000;
    CPPM.oservos[1]=3000;
    CPPM.oservos[2]=3000;
    CPPM.oservos[3]=3000;
}
Serial.print("ch1="); Serial.print(CPPM.oservos[0]);Serial.print("--");
Serial.print("ch2=");Serial.print(CPPM.oservos[1]);Serial.print("--");
Serial.print("ch3="); Serial.print(CPPM.oservos[2]);Serial.print("--");
Serial.print("ch4="); Serial.print(CPPM.oservos[3]);Serial.print("--");
Serial.print("ch5="); Serial.print(CPPM.oservos[4]);Serial.print("--");
Serial.print("ch6="); Serial.print(CPPM.oservos[5]);Serial.print("--");
Serial.print("distance=");Serial.print(distance);Serial.print("--");

```

```

Serial.print("lati=");Serial.print(convertgpstolat(gps.datagps)*1000000
);Serial.print("--");
    Serial.print("longi=");
Serial.println(convertgpstolongitu(gps.datagps)*1000000);
    Serial.print("x=");Serial.print( posisix);Serial.print("--");
    Serial.print("y="); Serial.println( posisiy);

}

void missi2(double posisix1,double posisiy1,double distance1,double
sudutTujuan1){
    CPPM.oservos[6]=ppm.aux1;
    controlremotenormally();

    controlwaypoint(posisix1,posisiy1,30,0,5,distance1,sudutpesawat,3);
//    tahanthrottle();
    CPPM.oservos[3]=3000;

controlremotewithcompasspid(controlrcwaypoint.roll,controlrcwaypoint
.pitch, controlrcwaypoint.throt , ppm.rudd,sudutpesawat,sudutTujuan1) ;
    if(ppm.gear>3800)
    {
        CPPM.oservos[0]=3000;
        CPPM.oservos[1]=3000;
        CPPM.oservos[2]=3000;
        CPPM.oservos[3]=3000;
    }
//CPPM.oservos[4]=3800;
//    tonodemcu.println("a");

Serial.print("ch1="); Serial.print(CPPM.oservos[0]);Serial.print("--");
Serial.print("ch2=");Serial.print(CPPM.oservos[1]);Serial.print("--");
Serial.print("ch3="); Serial.print(CPPM.oservos[2]);Serial.print("--");
Serial.print("ch4="); Serial.print(CPPM.oservos[3]);Serial.print("--");

```



```

    Serial.print("ch5="); Serial.print(CPPM.oservos[4]);Serial.print("--
");
    Serial.print("ch6="); Serial.print(CPPM.oservos[5]);Serial.print("--
");
//   Serial.print("altgps=");Serial.print(alt);Serial.print("--");
    Serial.print("distance=");Serial.print(distance1);Serial.print("--");

Serial.print("lati=");Serial.print(convertgpstolat(gps.datagps)*10000000
);Serial.print("--");
    Serial.print("longi=");
Serial.println(convertgpstolongitu(gps.datagps)*10000000);
    Serial.print("x=");Serial.print( posisix1);Serial.print("--");
    Serial.print("y="); Serial.println( posisiy1);

//   Serial.print("rem");
//   if(ppm.gear<=2600)
//   {
//       Serial.print("a");
//   }
//
//   else if(ppm.gear>=2600&&ppm.gear<3300)
//   {
//       Serial.print("b");
//   }
//   else if(ppm.gear>=3300)
//   {
//       Serial.print("c");
//   }
/////remote aktifin arming
//   if( ppm.aux2<2800)
//   {
//       Serial.print("d");
//   }
//   else if( ppm.aux2>=3000)

```

```

// {
//   Serial.print("e");
// }
// // remote aktifin misi follower
// if( ppm.aux3<=2800)
// {
//   Serial.print("f");
// }
// else if( ppm.aux3>=3000&&ppm.aux3<=3300)
// {
//   Serial.print("g");
// }
// else if( ppm.aux3>=3500)
// {
//   Serial.print("h");
// }
// Serial.print("\n");
}

```

```

void missi3(double posisix2,double posisiy2,double distance2,double
sudutTujuan2){

```

```

    CPPM.oservos[6]=ppm.aux1;
    controlremotenormally();

```

```

    controlwaypoint(posisix2,posisiy2,30,0,5,distance2,sudutpesawat,4);
//   tahanthrottle();
    CPPM.oservos[3]=3000;

```

```

controlremotewithcompasspid(controlrcwaypoint.roll,controlrcwaypoint
.pitch, controlrcwaypoint.throt , ppm.rudd,sudutpesawat,sudutTujuan2) ;
if(ppm.gear>3800)

```

```

{
    CPPM.oservos[0]=3000;
    CPPM.oservos[1]=3000;
    CPPM.oservos[2]=3000;
    CPPM.oservos[3]=3000;

```

```

    }
    //CPPM.oservos[4]=3800;
    // tonodemcu.println("a");

    Serial.print("ch1="); Serial.print(CPPM.oservos[0]);Serial.print("--");
    Serial.print("ch2");Serial.print(CPPM.oservos[1]);Serial.print("--");
    Serial.print("ch3="); Serial.print(CPPM.oservos[2]);Serial.print("--");
    Serial.print("ch4="); Serial.print(CPPM.oservos[3]);Serial.print("--");
    Serial.print("ch5="); Serial.print(CPPM.oservos[4]);Serial.print("--");
    Serial.print("ch6="); Serial.print(CPPM.oservos[5]);Serial.print("--");
    // Serial.print("altgps=");Serial.print(alt);Serial.print("--");
    Serial.print("distance=");Serial.print(distance2);Serial.print("--");

    Serial.print("lati=");Serial.print(convertgpstolat(gps.datagps)*10000000);Serial.print("--");
    Serial.print("longi=");
    Serial.println(convertgpstolongitu(gps.datagps)*10000000);
    Serial.print("x=");Serial.print( posisix2);Serial.print("--");
    Serial.print("y="); Serial.println( posisiy2);

    // Serial.print("rem");
    // if(ppm.gear<=2600)
    // {
    // Serial.print("a");
    // }
    //
    // else if(ppm.gear>=2600&&ppm.gear<3300)
    // {
    // Serial.print("b");
    // }
    // else if(ppm.gear>=3300)
    // {

```

```

//   Serial.print("c");
//   }
/////remote aktifin arming
//   if( ppm.aux2<2800)
//   {
//   Serial.print("d");
//   }
//   else if( ppm.aux2>=3000)
//   {
//   Serial.print("e");
//   }
//   // remote aktifin misi follower
//   if( ppm.aux3<=2800)
//   {
//   Serial.print("f");
//   }
//   else if( ppm.aux3>=3000&&ppm.aux3<=3300)
//   {
//   Serial.print("g");
//   }
//   else if( ppm.aux3>=3500)
//   {
//   Serial.print("h");
//   }
//   Serial.print("\n");
//   }

void missi4(){
  CPPM.oservos[0]=3000;
  CPPM.oservos[1]=3000;
  CPPM.oservos[2]=2900;
  CPPM.oservos[3]=3000;
  CPPM.oservos[4]=3800;
  CPPM.oservos[5]=2000;
  CPPM.oservos[6]=3800;
// CPPM.oservos[7]=;

```

```

    Serial.print("ch1="); Serial.print(CPPM.oservos[0]);Serial.print("--
");
    Serial.print("ch2");Serial.print(CPPM.oservos[1]);Serial.print("--");
    Serial.print("ch3="); Serial.print(CPPM.oservos[2]);Serial.print("--");
    Serial.print("ch4="); Serial.print(CPPM.oservos[3]);Serial.print("--");
    Serial.print("ch5="); Serial.print(CPPM.oservos[4]);Serial.print("--
");
    Serial.print("ch6="); Serial.print(CPPM.oservos[5]);Serial.print("--
");
// Serial.print("altgps=");Serial.print(alt);Serial.print("--");
    Serial.print("distance=");Serial.print(distance);Serial.print("--");

Serial.print("lati=");Serial.print(convertgpstolat(gps.datagps)*10000000
);Serial.print("--");
    Serial.print("longi=");
Serial.println(convertgpstolongitu(gps.datagps)*10000000);
    Serial.print("x=");Serial.print( posisix);Serial.print("--");
    Serial.print("y="); Serial.println( posisiy);

// Serial.print("rem");
// if(ppm.gear<=2600)
// {
// Serial.print("a");
// }
//
// else if(ppm.gear>=2600&&ppm.gear<3300)
// {
// Serial.print("b");
// }
// else if(ppm.gear>=3300)
// {
// Serial.print("c");
// }
/////remote aktifin arming
// if( ppm.aux2<2800)

```

```

// {
// Serial.print("d");
// }
// else if( ppm.aux2>=3000)
// {
// Serial.print("e");
// }
// // remote aktifin misi follower
// if( ppm.aux3<=2800)
// {
// Serial.print("f");
// }
// else if( ppm.aux3>=3000&&ppm.aux3<=3300)
// {
// Serial.print("g");
// }
// else if( ppm.aux3>=3500)
// {
// Serial.print("h");
// }
// Serial.print("\n");
}

```

```

void setup() {
  Serial.begin(9600);
  tonodemcu.begin(9600);
  pinMode(trig, OUTPUT); // set pin trig menjadi OUTPUT
  pinMode(echo, INPUT); // set pin echo menjadi INPUT
  //frommcu.begin(9600);
  CPPM.begin();
  Wire.begin();
  Wire.beginTransmission(address); //open communication with
  HMC5883
  Wire.write(0x02); //select mode register
  Wire.write(0x00);
}

```

```

Wire.endTransmission();

}

int kirimauto;
int kirimflightmode;
int kirimarming;

void loop() {
//-72779088.00--latitude1127907328.00
//-72780136.00--latitude1127908224.00
//1127960192.00
//-72855144.00

if(CPPM.synchronized())
{
// -72845576.00
// 1127973376.00
// -72849384.00
// 1127968000.00

//-7.2845672 tengah
//112.7973376

//-72845656.00 sebelahkiri
//1127973504.00

//-7.2845424 sebelahkanan
//112.7972736

//-72844936.00 tengah belakang

```

```
//1127973376.00
```

```
// ppm.aile=CPPM.read(CPPM_AILE)-3000;  
// ppm.elev = CPPM.read(CPPM_ELEV)-3000;  
// ppm.thro = CPPM.read(CPPM_THRO)-3000;  
// ppm.rudd = CPPM.read(CPPM_RUDD)-3000;  
// ppm.gear=CPPM.read(CPPM_GEAR) ;  
// ppm.aux1=CPPM.read(CPPM_AUX1) ;  
// ppm.aux2 = CPPM.read(CPPM_AUX2);  
// ppm.aux3 = CPPM.read(CPPM_AUX3) ;
```

```
    sudutpesawat=headingHMCL();
```

```
sudutTujuan=bearing(convertgpstolat(gps.datagps),convertgpstolongitu(  
gps.datagps),-7.2845672, 112.7973376);  
    distance=calc_dist(convertgpstolat(gps.datagps),  
convertgpstolongitu(gps.datagps) ,-7.2845672, 112.7973376);  
    posisiy=convertpositiony(distance,sudutTujuan);//y  
    posisix=convertpositionx(distance,sudutTujuan);//x  
    ////////////////////////////////////
```

```
sudutTujuan1=bearing(convertgpstolat(gps.datagps),convertgpstolongitu  
(gps.datagps),-7.2844936, 112.7973376);  
    distance1=calc_dist(convertgpstolat(gps.datagps),  
convertgpstolongitu(gps.datagps) ,-7.2844936, 112.7973376);  
    posisiy1=convertpositiony(distance,sudutTujuan1);//y  
    posisix1=convertpositionx(distance,sudutTujuan1);//x  
    ////////////////////////////////////
```

```
sudutTujuan2=bearing(convertgpstolat(gps.datagps),convertgpstolongitu  
(gps.datagps),-7.2847744, 112.7967488);  
    distance2=calc_dist(convertgpstolat(gps.datagps),  
convertgpstolongitu(gps.datagps) ,-7.2847744, 112.7967488);  
    posisiy2=convertpositiony(distance,sudutTujuan2);//y
```



```

posisix2=convertpositionx(distance,sudutTujuan);//x
////////////////////////////////////
double alt=convertgpstoalt();
ppm.aux1 = CPPM.read(CPPM_AUX1) ;
if(ppm.aux1<=2800){gantichannel=0;
}
else if(ppm.aux1>=3000){gantichannel=1;
}

switch(gantichannel)
{
case 0:
CPPM.oservos[6]=ppm.aux1;
controlremotenormally();
// controlwaypoint(posisix,(posisiy),30,0,5,distance,sudutpesawat,0);
controlwaypoint(posisix,posisiy,30,0,5,distance,sudutpesawat,0);

controlremotewithcompasspid(controlrcwaypoint.roll,controlrcwaypoint
.pitch, ppm.thro , ppm.rudd,sudutpesawat,sudutTujuan);
tahanthrottle();
// CPPM.oservos[2]=3000;
// autowaypoint(posisix,posisiy,distance,sudutTujuan ,posisix1,
posisiy1,distance1,sudutTujuan1);
if(ppm.gear>3800)
{
CPPM.oservos[0]=3000;
CPPM.oservos[1]=3000;
CPPM.oservos[2]=2600;
CPPM.oservos[3]=3000;
}
// Serial.print("ch1="); Serial.print(CPPM.oservos[0]);Serial.print("--");
// Serial.print("ch2=");Serial.print(CPPM.oservos[1]);Serial.print("--");
// Serial.print("ch3="); Serial.print(CPPM.oservos[2]);Serial.print("--");
// Serial.print("ch4="); Serial.print(CPPM.oservos[3]);Serial.print("-
-");

```

```

// Serial.print("ch5="); Serial.print(CPPM.oservos[4]);Serial.print("-
-");
// Serial.print("ch6="); Serial.print(CPPM.oservos[5]);Serial.print("-
-");
// Serial.print(" distance=");Serial.print(distance);Serial.print("--");
//
Serial.print("lati=");Serial.print(convertgpstolat(gps.datagps)*10000000
);Serial.print("--");
//
// Serial.print("longi=");
Serial.println(convertgpstolongitu(gps.datagps)*10000000);
// Serial.print("x=");Serial.print( posisix);Serial.print("--");
// Serial.print("y="); Serial.println( posisiy);
Serial.print("rem");
if(ppm.gear<=2600)
{
Serial.print("a");
}

else if(ppm.gear>=2600&&ppm.gear<3300)
{
Serial.print("b");
}
else if(ppm.gear>=3300)
{
Serial.print("c");
}
////remote aktifin arming
if( ppm.aux2<2800)
{
Serial.print("d");
}
else if( ppm.aux2>=3000)
{
Serial.print("e");
}
// remote aktifin misi follower

```

```

if( ppm.aux3<=2800)
{
Serial.print("f");
}
else if( ppm.aux3>=3000&&ppm.aux3<=3300)
{
Serial.print("g");
}
else if( ppm.aux3>=3500)
{
Serial.print("h");
}
Serial.print("\n");
break;

// case 2:
// missi2(posisix1,posisiy1,distance1,sudutTujuan1);
// Serial.println("missi2");
// break;
//
// case 3:
// missi3(posisix2,posisiy2,distance2,sudutTujuan2);
// Serial.println("missi3");
// break;
// case 4:
// missi4();
// Serial.println("selesai");
// break;

case 1:
CPPM.oservos[6]=ppm.aux1;
controlremotenormally();
rcnocompas(ppm.aile,ppm.elev,
ppm.thro,ppm.rudd,ppm.gear,ppm.aux1,ppm.aux2,ppm.aux3);

```

```

gantithrottle=0;
Serial.print("rem");
if(ppm.gear<=2600)
{
  Serial.print("a");
}

else if(ppm.gear>=2600&&ppm.gear<3300)
{
  Serial.print("b");
}
else if(ppm.gear>=3300)
{
  Serial.print("c");
}
///remote aktifin arming
if( ppm.aux2<2800)
{
  Serial.print("d");
}
else if( ppm.aux2>=3000)
{
  Serial.print("e");
}
///remote aktifin misi follower
if( ppm.aux3<=2800)
{
  Serial.print("f");
}
else if( ppm.aux3>=3000&&ppm.aux3<=3300)
{
  Serial.print("g");
}
else if( ppm.aux3>=3500)
{
  Serial.print("h");
}

```

```

    }
    Serial.print("\n");
    //  tonodemcu.println("b");

    //  Serial.print("ch1="); Serial.print( CPPM.oservos[0]);Serial.print("--");
    //  Serial.print("ch2="); Serial.print(CPPM.oservos[1]);Serial.print("--");
    //  Serial.print("ch3="); Serial.print(CPPM.oservos[2]);Serial.print("--");
    //  Serial.print("ch4=");Serial.print(CPPM.oservos[3]);Serial.print("--");
    //  Serial.print("ch5="); Serial.print( CPPM.oservos[4]);Serial.print("--");
    //  Serial.print("ch6="); Serial.print(CPPM.oservos[5]);Serial.print("--");
    //  Serial.print("ch7="); Serial.print(CPPM.oservos[6]);Serial.print("--");
    //  Serial.print("ch8=");Serial.println(CPPM.oservos[7]);
    //  Serial.flush();
    //  tonodemcu.flush();
    break;
}

}
tonodemcu.flush();
}

```

[Halaman ini sengaja dikosongkan]

RIWAYAT HIDUP



Penulis, Michael Jacob Ephariem Jehuda, lahir di Surabaya, 21 Februari 1996. Merupakan anak pertama dari dua bersaudara dari pasangan Bapak Winston Leonard dan Ibu Sunarni. Bertempat tinggal bersama orang tua di Jalan Cipta Menanggal 1 blok 18 AA1, Kecamatan Gayungan, Kota Surabaya, Jawa Timur, Indonesia. Penulis memulai Studinya di TK Elizabeth lulus tahun 2002, SD Negeri Dukuh Menanggal 2 lulus tahun 2008, dilanjutkan di SMP Negeri 36 Surabaya lulus tahun 2011, dan SMA 17 Agustus 1945 tahun 2014. Kemudian pada tahun 2014 melanjutkan pendidikan tinggi di Diploma III Teknik Elektro ITS Bidang Studi Komputer Kontrol lulus tahun 2017. Kemudian penulis melanjutkan S1 teknik elektro ITS dan pada bulan Juni 2019 penulis mengikuti ujian Tugas Akhir di Bidang Studi Elektronika Program Studi S1 Teknik Elektro ITS Surabaya sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik Elektro. Penulis dapat dihubungi pada alamat email sebagai berikut: michaeljacob6789@gmail.com