PENGEMBANGAN ALGORITMA KOMUNIKASI ANTAR-UNMANNED AERIAL VEHICLE BERBASIS *PAINLESSMESH* PADA MIKROKONTROLER ESP32

DEVELOPMENT OF AN INTER-UNMANNED AERIAL VEHICLE COMMUNICATIONS ALGORITHM BASED ON PAINLESSMESH USING AN ESP32 MICROCONTROLLER

TUGAS AKHIR

Disusun sebagai syarat mata kuliah Tugas Akhir

Program Studi S1 Teknik Elektro

Disusun oleh:

MUHAMMAD ADEEL MAHDI SUVIYANTO

1102183191

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

BANDUNG

2022

LEMBAR PENGESAHAN

TUGAS AKHIR

PENGEMBANGAN ALGORITMA KOMUNIKASI ANTAR-UNMANNED AERIAL VEHICLE BERBASIS *PAINLESSMESH* PADA MIKROKONTROLER ESP32

DEVELOPMENT OF AN INTER-UNMANNED AERIAL VEHICLE COMMUNICATIONS ALGORITHM BASED ON PAINLESSMESH USING AN ESP32 MICROCONTROLLER

Telah disetujui dan disahkan sebagai Buku Tugas Akhir
Program Studi S1 Teknik Elektro
Fakultas Teknik Elektro
Telkom University

Disusun oleh:
MUHAMMAD ADEEL MAHDI SUVIYANTO
1102183191

Bandung, x Juli 2021

Pembimbing 1	Pembimbing 2
Dr. Eng. Willy Anugrah Cahyadi, S.T, M.T.	Ir. Uke Kurniawan Usman,M.T.

LEMBAR PERNYATAAN ORISINALITAS

Nama	:	Muhammad Adeel Mahdi Suviyanto
NIM	:	1102183191
Alamat	:	Taman Alfa Indah D4/11, Joglo, Kembangan, Jakarta Barat
No Tlp/HP	:	021 73443989 - 087777882699
Email	:	adeelsuviyanto@student.telkomuniversity.ac.id

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri dengan judul:

Pengembangan Algoritma Komunikasi Antar-Unmanned Aerial Vehicle Berbasis *PainlessMesh* Pada Mikrokontroler ESP32
Development of an Inter-Unmanned Aerial Vehicle Communications Algorithm based on PainlessMesh using an ESP32 Microcontroller

Atas pernyataan ini, saya siap menanggung risiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidak aslian karya ini.

Bandung, x Juli 2022

Muhammad Adeel Mahdi Suviyanto
1102183191

ABSTRAK

Placeholder teks Abstrak.

ABSTRACT

Placeholder text for Abstract.

KATA PENGANTAR

Placeholder text for Abstract.

UCAPAN TERIMA KASIH

Placeholder text for Abstract.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Salah satu faktor penentu kesuksesan dalam operasi *Search and Rescue* (SAR) pada bencana alam adalah kecepatan dalam menemukan lokasi dan posisi korban serta pengiriman logistik bantuan penunjang hidup bagi korban tersebut. Namun, kondisi daratan medan bencana alam yang sukar dilewati oleh tim penyelamat dapat menyebabkan lamanya kedua proses tersebut, menurunkan probabilitas keselamatan bagi korban [1]. Pada saat ini, metode yang biasa digunakan untuk pencarian korban adalah menggunakan helikopter dengan pencarian manual dari udara. Akan tetapi, penggunaan helikopter memiliki kelemahan pada sisi biaya operasi serta perubahan cuaca, dimana penerbangan helikopter yang aman hanya dapat dilakukan pada kondisi cuaca cerah tidak berkabut [2]. Oleh karena itu, penggunaan *Unmanned Aerial Vehicles* (UAV) untuk kepentingan SAR dapat meningkat sebagai pendukung terhadap penggunaan helikopter pada operasi SAR.

Menurut (Lakshmi Narayanan, 2015), UAV adalah sebuah tipe pesawat terbang yang dapat mengudara tanpa adanya awak di atas kapal [3]. Terdapat berbagai kegunaan sebuah UAV, salah satunya adalah operasi SAR. UAV memiliki keunggulan yang cocok bagi operasi SAR, yakni kemampuannya untuk melihat suatu area yang luas dengan akses yang cepat tanpa terhalang oleh medan bencana [4]. UAV juga unggul dalam menghadapi cuaca buruk, dimana cuaca berkabut dapat menyebabkan helikopter tidak dapat terbang karena alasan keamanan, sedangkan UAV tetap dapat terbang karena tidak ada personil yang dibahayakan pada kondisi tersebut. Pada lokasi bencana alam, akses medan bencana yang sulit dapat mempersulit operasi SAR, terutama pada pencarian korban dan pengiriman bantuan.

Oleh karena itu, penelitian ini mengusulkan suatu pengembangan pada sistem komunikasi antar-UAV yang kemudian dapat digunakan pada operasi SAR, dimana data yang dikirimkan berupa koordinat lokasi salah satu unit UAV dalam jaringan. Untuk mewujudkan koordinasi antar masing-masing unit UAV, maka diperlukan sistem komunikasi antar-UAV yang memenuhi kriteria kinerja minimum: *throughput* (laju pengiriman data) minimum 16 KBps, *packet loss* (persentase paket data yang hilang dalam transmisi) dibawah 25%, jarak minimum antar unit UAV minimum 25 meter, dan *round-trip delay* (waktu tempuh pengiriman data bolak-balik) dibawah 4000 milisekon.

Terdapat beberapa arsitektur komunikasi yang layak untuk penggunaan pada komunikasi antar-UAV, seperti *Flying Ad-Hoc Network* (FANET) [5] dan *Centralized* berbasis teknologi seluler (LTE, 5G) [6]. Namun, ketergantungan teknologi seluler terhadap infrastruktur yang telah ada di darat membuat komunikasi berbasis seluler

kurang sesuai jika digunakan untuk kondisi bencana, karena rusaknya infrastruktur fisik (menara *Base Transceiver Station* (BTS)), disrupsi pada infrastruktur penunjang (listrik), dan juga *overload* oleh melonjaknya jumlah pengguna jaringan di waktu yang bersamaan [7]. Oleh karena itu, penulis akan menggunakan teknologi FANET berbasis IEEE 802.11 WiFi dengan menggunakan mikrokontroler ESP32 dalam sebuah jaringan WiFi Mesh.

Mikrokontroler ESP32 digunakan karena harganya yang ekonomis dan telah memiliki kapabilitas WiFi IEEE 802.11 secara *built-in*, dapat diprogram menggunakan bahasa pemrograman Arduino yang berbasiskan C dan C++, serta memiliki dokumentasi dan *plug-in* yang lengkap. ESP32 juga mendukung beragam mode operasi WiFi IEEE 802.11 dari 802.11b/g/n dan mode khusus Espressif yakni 802.11 *Long Range*. Setiap mikrokontroler ESP32 beroperasi pada pita frekuensi 2.4 GHz. Pada sistem yang dirancang, masing-masing unit UAV akan dipasangkan satu unit board ESP32 yang kemudian akan berkomunikasi satu sama lain pada mode WiFi ad-hoc.

Dalam tugas akhir ini, dirancang sebuah algoritma komunikasi antar-UAV berbasis WiFi Mesh pada mikrokontroler ESP32, serta akan menganalisis sistem yang dihasilkan dengan parameter kinerja jaringan (*throughput, packet loss, round-trip delay, signal strength* (RSSI) terhadap jarak antar unit UAV, dampak sistem penerbangan dan kendali UAV terhadap kinerja sistem, serta mode IEEE 802.11 yang digunakan ESP32 terhadap kinerja sistem. Diharapkan hasil sistem yang diperoleh dapat dijadikan salah satu metode komunikasi antar-UAV pada kegunaan operasi SAR dalam menemukan posisi korban.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaiamana korelasi antara jarak antar-UAV terhadap kinerja jaringan (*throughput, packet loss, range, round-trip delay*) yang telah diimplementasikan?
2. Apakah sistem penerbangan dan kendali UAV mempengaruhi kinerja sistem komunikasi antar-UAV?
3. Apa mode WiFi 802.11 yang cocok digunakan untuk kegunaan sistem komunikasi antar-UAV?
4. Apakah algoritma komunikasi yang dihasilkan dapat diimplementasikan di lokasi medan bencana alam?

1.3 Tujuan dan Manfaat

Tujuan dari perancangan algoritma komunikasi antar-UAV ini adalah:

1. Mengetahui korelasi jarak antar-node dan perbedaan ketinggian antar-UAV terhadap kinerja jaringan yang diimplementasikan, dengan parameter kinerja *throughput, packet loss, range, round-trip delay*.
2. Mengetahui dampak sistem penerbangan dan kendali UAV terhadap kinerja jaringan yang diimplementasikan.
3. Mengetahui korelasi mode WiFi 802.11 yang digunakan pada sistem terhadap kinerja jaringan yang diimplementasikan
4. Mengetahui apakah algoritma yang dihasilkan berguna untuk operasi SAR pada bencana alam.

Adapun manfaat dari penelitian ini adalah:

1. Mengembangkan sebuah algoritma komunikasi antar-UAV yang cukup handal dengan *link quality* tinggi sehingga UAV dapat berkomunikasi satu sama lain untuk mengirimkan data.
2. Sebagai tahap pertama dari pengembangan sistem UAV SAR otonom.
3. Sebagai sumber pustaka bagi penelitian di masa depan mengenai permasalahan terkait.

1.4 Batasan Masalah

Agar pembahasan dalam penelitian dapat difokuskan, maka terdapat pembatasan masalah sebagai berikut:

1. Menggunakan 2 unit drone dan satu *Base Station* (BS) untuk menyederhanakan sistem rancangan.
2. Data komunikasi antar-UAV yang dikirimkan berupa data koordinat (Lintang dan Bujur) dengan 5 angka desimal untuk tingkat kepresisian 1 meter [8].
3. Setiap node ESP32 menggunakan kanal 1 WiFi 2.4 GHz dengan rentang frekuensi 2401-2423 MHz.
4. Kendali masing-masing drone dilakukan secara terpisah dari sistem komunikasi yang diuji dan dilakukan secara manual oleh operator menggunakan *remote control* (R/C).
5. Parameter yang digunakan pada analisis algoritma jaringan yang diimplementasikan adalah *throughput* jaringan, *packet loss*, *round-trip delay*, dan *signal strength* dalam RSSI (dBm).

6. Pengujian dilakukan di area Gedung N Fakultas Teknik Elektro Telkom University, dengan jarak antar UAV 20 meter, 50 meter, dan 100 meter, dengan ketinggian setingkat lantai 1, 2, dan 3 pada Gedung N.

7. Pengujian dilakukan dengan kondisi drone OFF dan ON.

1.5 Metode Penelitian

Metode penelitian yang digunakan pada tugas akhir ini antara lain:

1. Studi Literatur

   Studi literatur dilakukan dengan mempelajari beberapa materi yang berkaitan dengan penelitian ini, dengan sumber yang digunakan berupa jurnal, artikel, buku, dan situs web yang terpercaya.

2. Perancangan Sistem

   Pada tahap ini, dilakukan perancangan sistem sesuai dengan target yang telah ditentukan. Melalui perancangan sistem, dihasilkan suatu gambaran jelas mengenai struktur penyusunan sistem dan dapat dilakukan analisis secara matematis.

3. Implementasi

   Sistem yang telah dirancang kemudian diimplementasikan melalui perangkaian komponen-komponen yang telah ditentukan, serta melakukan pemrograman sistem tersebut.

4. Pengukuran Empiris

   Pada tahap ini, sistem yang telah diimplementasikan diuji melalui beberapa tes yang menguji sistemnya secara kuantitatif untuk menghasilkan data empiris yang dapat diolah dalam bentuk grafik.

5. Analisis Statistik

   Hasil pengukuran kemudian dianalisis berdasarkan teori yang telah dikemukakan, dan menghitung faktor-faktor lainnya seperti keakuratan alat pengukur dan faktor-faktor eksternal yang mempengaruhi kinerja sistem.

1.6 Jadwal Pelaksanaan

Berikut adalah jadwal pelaksanaan penelitian ini, rincian waktu dan *milestone* dirangkum dalam tabel di bawah ini:

Tabel 1.1: Jadwal pelaksanaan penelitian.

No.	**Deskripsi Tahapan**	**Durasi**	**Tanggal Selesai**	***Milestone***
1	Rumusan masalah dan studi literatur	2 Minggu	21 Oktober 2021	Mengidentifikasi permasalahan dan studi literatur.
2	Desain sistem	2 Minggu	29 Oktober 2021	Diagram blok sistem, sketsa dasar sistem, diagram alur sistem, dan spesifikasi alat.
3	Pemilihan komponen	1 Minggu	5 November 2021	Pendataan komponen sistem yang akan digunakan.
4	Perancangan dan pembuatan sistem	1 Bulan	3 Desember 2021	Implementasi sistem secara fisik.
5	Pengujian sistem	2 Minggu	8 Juli 2022	*Test flight* dan pengujian jaringan sistem.
6	Penyusunan Laporan/Buku TA	2 Minggu	22 Juli 2022	Laporan/Buku TA selesai.

1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan pada tugas akhir ini adalah:

BAB I: PENDAHULUAN

Bab ini berisi uraian singkat mengenai latar belakang permasalahan, rumusan masalah, tujuan dan manfaat, pembatasan masalah, serta jadwal pelaksanaan penelitian.

BAB II: TINJAUAN PUSTAKA DAN KONSEP DASAR SISTEM

Bab ini berisi uraian mengenai landasan teori serta membahas konsep dasar sistem yang dibahas dalam tugas akhir ini.

BAB III: PERANCANGAN SISTEM

Bab ini berisi uraian mengenai rancangan sistem dari sisi desain perangkat keras maupun perangkat lunak, fungsi dan fitur, serta spesifikasi sistem.

BAB IV: HASIL DAN ANALISIS

Bab ini berisi uraian mengenai hasil pengujian sistem, serta analisis dari hasil pengujian tersebut secara rinci terhadap parameter yang sudah ditentukan.

BAB V: SIMPULAN DAN SARAN

Bab ini berisi rincian kesimpulan dari penelitian yang telah dikerjakan, serta saran untuk penelitian berikutnya.

BAB II
TINJAUAN PUSTAKA DAN KONSEP DASAR SISTEM

2.1 Prinsip Kerja Sistem

Berdasarkan rumusan masalah yang telah dipaparkan pada bab 1, berikut adalah konsep prinsip kerja sistem yang dicanangkan:

1. Terdapat 2 drone dan 1 *Base Station* (BS), masing-masing menggunakan mikrokontroler ESP32 yang berkomunikasi satu sama lain menggunakan JSON *messages* pada jaringan PainlessMesh. Drone 1 disebut sebagai *"sender node"*, dan drone 2 disebut sebagai *"receiver node"*.

2. ESP32 Drone 1 mengaktifkan modul GPS dan mendata koordinat lokasi, ketinggian terbang drone, dan jumlah satelit yang mendapat *lock*, kemudian menghidupkan LED hijau pada *board*.

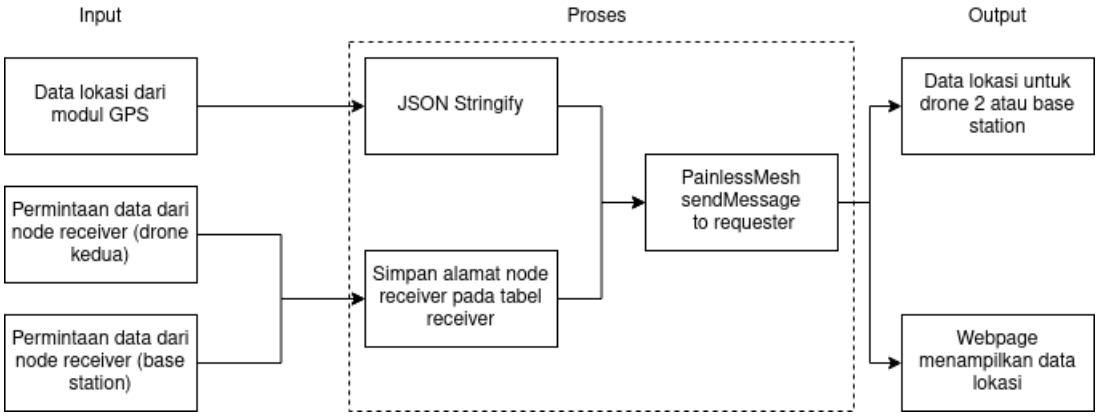
3. Drone 2 mengirimkan permintaan data lokasi kepada drone 1.

4. Drone 1 mengirimkan data melalui jaringan mesh kepada drone 2. Drone 2 menghidupkan LED hijau setelah menerima data lokasi yang valid.

5. BS mengirimkan permintaan data pada drone 1 berupa *string* dalam *message*.

6. Drone 1 mengirimkan data melalui jaringan mesh kepada BS.

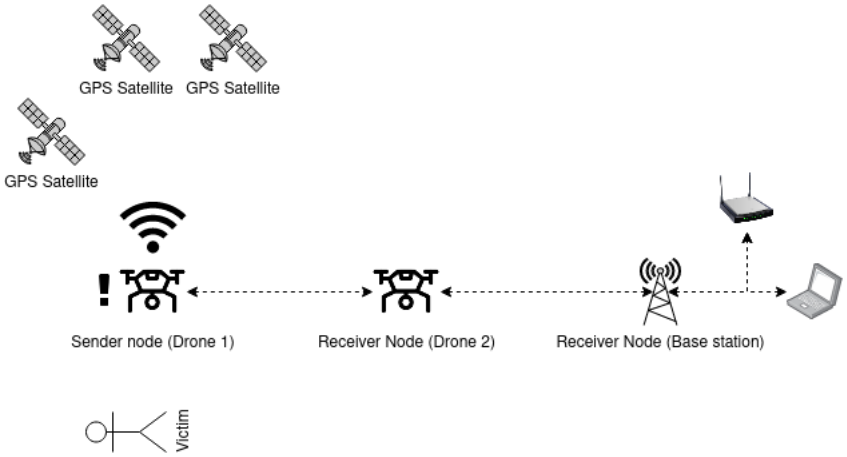
7. BS menampilkan data kepada pengguna melalui *WiFi Access Point* yang dapat diakses dari sebuah *website*. WiFi AP tersebut dibangun menggunakan sebuah board ESP8266 yang menerima data dari board ESP32 yang terhubung dengan jaringan mesh melalui sambungan serial.



Gambar 2.1: Desain konsep sistem

PainlessMesh bekerja menggunakan mode AP-STA pada mikrokontroler ESP32, dimana *board* bekerja sebagai *Wi-Fi Access Point* sekaligus *Wi-Fi Station Mode*

untuk membentuk sebuah jaringan mesh. Masing-masing node memiliki sebuah node ID, dan pada aplikasi penelitian ini memiliki penanda sebagai *sender* (node yang mendata lokasi GPS dan mengirimkan data tersebut) dan *receiver* (node yang menerima data lokasi tersebut).



Gambar 2.2: Diagram sistem

Node *sender* memiliki modul GPS yang berkomunikasi dengan satelit untuk mendapatkan data lokasi yang terlampir dalam *NMEA sentences*. Data tersebut kemudian diolah menggunakan *library* TinyGPS++ untuk mengekstraksi data lokasi dari *NMEA sentences* tersebut.

Sender node memiliki sebuah *flag* ”locationReady” dimana jika nilainya *”true”* maka dapat menerima permintaan data lokasi dari *receiver node*. Kriteria untuk *flag* tersebut memiliki nilai *”true”* adalah telah mendapatkan *lock* satelit minimal 3 dan telah mendapat nilai lintang dan bujur bukan nol.

Akses ke *Base Station* menggunakan sebuah board ESP8266 yang terhubung secara serial dengan board ESP32 yang terhubung dengan jaringan mesh. Konfigurasi ini diperlukan karena limitasi dari board ESP32 yang memiliki koneksi bersamaan maksimum sebanyak 2 (1 dari klien *Access Point*, 1 menjadi klien AP sebagai *Station Mode*), sedangkan PainlessMesh menggunakan kedua mode tersebut untuk mempertahankan jaringan mesh yang dibentuk.

2.2 Riset Terkait

Terdapat beberapa penelitian yang telah dilaksanakan sebelumnya yang terkait dengan tugas akhir ini, yang akan digunakan sebagai dasar atau referensi dalam pengerjaan. Beberapa penelitian terkait dapat dilihat pada tabel 2.1.

Tabel 2.1: Penelitian terkait

	Judul	**Metode**	**Kesimpulan**	**Kelebihan**	**Kekurangan**
[9]	Use of High Mobility Nodes to Improve Connectivity in Wireless Sensor Networks	PainlessMesh-based Opportunistic Mobile Ad Hoc Networks	Menggunakan high mobility nodes berupa drone sebagai messenger data antara cluster sensor dengan server	Penelitian mengimplementasikan metode security berbasis secret-key cryptography. Packet loss antara sensor dan server rendah, hanya 4,48% pada kondisi terburuk walaupun packet melewati 2 hop.	Tidak menguji round-trip delay packet
[10]	Performance Evaluation of ESP8266 Mesh Networks	Pengujian one-way delay dan data rate pada jaringan PainlessMesh ESP8266	Jumlah node berkorelasi dengan meningkatnya single hop delay dan menurunnya stabilitas jaringan, serta besar payload menentukan data rate dan korupsi data.	Penelitian menguji dari 2 hingga 16 node sehingga terlihat gambaran kasar stabilitas jaringan PainlessMesh. Kinerja jaringan 2 node memiliki delay 2.49 ms sehingga cukup untuk aplikasi yang tidak terlalu kompleks.	ESP8266 tidak memiliki performa yang cukup untuk mengirimkan dan menerima payload data yang besar. Pengujian data rate dibatasi pada 2 node.

[11]	Implementing Wireless Mesh Network Topology Between Multiple Wi-Fi Powered Nodes for IoT Systems	Penggunaan 3 node PainlessMesh berbasis ESP32 untuk komunikasi multidirectional output sensor dan input push button	Mesh bersifat self-healing, pada saat terjadi disrupsi maka mesh secara otomatis mengatur diri.	Implementasi 3 node dan arah data bolak balik dapat dilakukan	Tidak ada pengujian jarak jauh
[12]	A dust sensor monitoring system using Wi-Fi mesh network	Implementasi 9 node berbasis ESP32 dan ESP-Mesh untuk monitoring tingkat debu pada suatu ruangan.	Sistem efektif dengan measurement error dibawah 5%	Mesh dapat berfungsi tanpa intervensi manusia, memiliki sifat self-healing dan auto-configuration.	Jaringan memiliki topologi tree, bukan mesh murni, sehingga jika terjadi gangguan pada node akar maka proses self-heal berjalan lebih lama.

Berdasarkan penelitian yang telah dilakukan sebelumnya, maka pada penelitian tugas akhir ini dibatasi menjadi 3 node, serta menguji *throughput, packet loss, round-trip delay,* dan *signal strength (dBm)*; serta pengujian jarak antar node terhadap kinerja jaringan. Khusus untuk node base station akan disambungkan dengan sebuah board ESP8266 melalui koneksi serial UART sebagai HTTP Server untuk koneksi web server yang menampilkan data lokasi dari node *sender*.

2.3 *Unmanned Aerial Vehicles*

UAV adalah sebuah pesawat terbang yang dapat mengudara tanpa awak [3], dikendalikan secara *remote* atau secara otonom. Salah satu tipe UAV adalah *quadcopter drone*. Pada sistem ini, UAV digunakan sebagai pengirim data lokasi GPS sebagai node *sender*, dan penerima data lokasi GPS sebagai node *receiver*.

2.4 *Quadcopter*

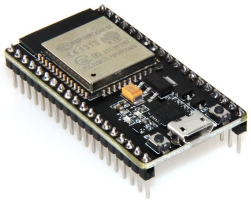
Drone *quadcopter* adalah suatu jenis UAV yang memiliki 4 rotor pada masing-masing sudut. Sama seperti helikopter, *quadcopter* memiliki kemampuan untuk *hover*. Terdapat sepasang rotor yang berputar searah jarum jam dan sepasang rotor

yang berputar berlawanan arah jarum jam, sehingga pada kondisi *steady state* total torsi pada *drone* adalah nol. Hal tersebut juga menyebabkan konfigurasi *quadcopter* tidak membutuhkan *tail rotor*. Keempat rotor tersebut juga menghasilkan daya angkat yang besar, sehingga cocok digunakan untuk membawa *payload*.

2.5 ESP32

Pada penelitian ini, ESP32 digunakan sebagai modul komunikasi antar node baik drone maupun base station. Board-board ESP32 dapat dikonfigurasikan menjadi jaringan mesh menggunakan mode APSTA/*Coexistence Mode*, dimana secara bersamaan ESP32 mem-*broadcast* SSID sambil terhubung dengan jaringan lain [13].

ESP32 juga mendukung protokol *proprietary* buatan Espressif bernama 802.11 LR *(Long Range)*, dengan jarak hingga 1 km selama ada *line-of-sight* antara *node*. [13].



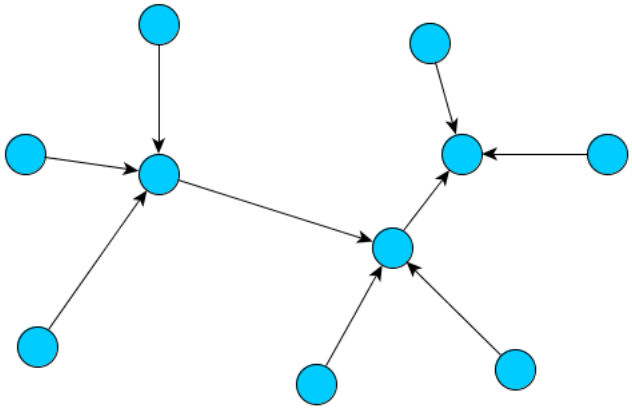
Gambar 2.3: ESP32

2.6 PainlessMesh

PainlessMesh merupakan sebuah library yang memudahkan pengguna *board* ESP32 dalam membuat sebuah jaringan mesh ad-hoc berbasis WiFi. Karena batasan dari *board* ESP32, maka jaringan yang dihasilkan merupakan *"pseudo-mesh"* atau mesh yang tidak sempurna, sehingga pada kondisi standar, semua node memiliki kedudukan yang sama dan diatur dalam topologi *tree*.

Jaringan PainlessMesh menggunakan mode APSTA pada board ESP32, dimana setiap board ESP32, disebut dengan "node", berfungsi sebagai WiFi *Access Point* (AP) sekaligus sebagai klien WiFi *Station Mode*. Setiap node berbasis ESP32 dapat memiliki sebanyak 10 klien yang terhubung pada 1 AP, dan dapat terhubung pada AP lainnya sebagai sebuah klien.

Keistimewaan dari library PainlessMesh adalah kemampuannya untuk *autoconfigure*, dimana setiap node dapat memutus sambungan dan menyambung ulang setiap saat, dan jaringan mesh dapat berjalan melalui proses konfigurasi secara otomatis.



Gambar 2.4: Topologi jaringan PainlessMesh [14], panah menunjukkan arah koneksi dari klien ke AP

2.7 JavaScript Object Notation (JSON)

PainlessMesh menggunakan JavaScript Object Notation untuk setiap pengiriman pesan antar-node dalam jaringan. Penggunaan JSON memudahkan penafsiran *(parsing)* setiap pesan yang diterima untuk kemudian diolah menjadi data yang ditampilkan [15].

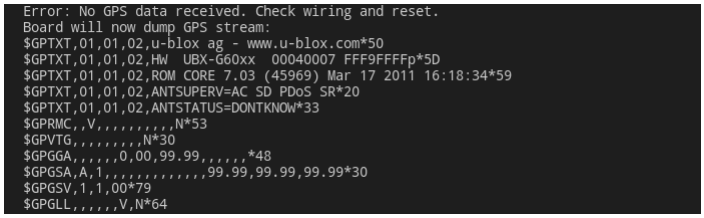
```
{
    "dest": 887034362,
    "from": 37418,
    "type":9,
    "msg": "The message I want to send"
}
```

Gambar 2.5: Setiap pesan pada jaringan PainlessMesh menggunakan JSON.

Setiap paket data PainlessMesh menggunakan skema JSON untuk menentukan tujuan (dari *key "dest"*), dari mana paket data tersebut berasal (*key "from"*), jenis paket berdasarkan nilai integer pada *key "type"*, serta pesan yang akan dikirim pada *key "msg"*.

2.8 GPS NMEA Data

Modul GPS NEO-6M menghasilkan output serial mentah berupa data NMEA 0183, sebuah standar komunikasi piranti GPS berupa karakter-karakter ASCII yang membentuk kalimat-kalimat NMEA [16].



Gambar 2.6: Contoh data mentah dari modul GPS NEO-6M berupa data GPS NMEA.

2.9 Pengukuran Kinerja Jaringan

2.9.1 Round-trip delay

Round-trip delay adalah waktu yang diperlukan untuk sebuah paket data dikirimkan dari node pengirim ke node penerima, ditambah waktu yang diperlukan untuk paket *acknowledge* diterima oleh node pengirim dari node penerima. Pengukuran *round-trip delay* dapat menggunakan fungsi yang sudah diimplementasikan dalam *library* PainlessMesh yakni `painlessMesh::startDelayMeas()`.

Setiap node pada jaringan PainlessMesh memiliki waktu internal yang tersinkronisasi satu sama lain melalui proses *"Time sync"* [15] yang berlangsung secara periodis setiap 10 menit. PainlessMesh menjaga waktu jaringan pada tingkat kepresisian mikrosekon menggunakan fungsi `micros()` pada mikrokontroler ESP32. Kemudian setelah fungsi `startDelayMeas()` dipanggil, maka *round-trip delay* dapat dihitung menggunakan rumus berikut:

$$t_{roundtrip} = (t_3 - t_0) - (t_2 - t_1)$$

t_0 = waktu internal pada pengirim *delay measurement*
t_1 = *timestamp* pada saat paket *delay measurement* diterima
t_2 = *timestamp* pada saat respon terhadap *delay measurement* dikirimkan
t_3 = *timestamp* pada saat respon diterima oleh pengirim *delay measurement*

2.9.2 Throughput

Throughput adalah banyaknya data yang dapat ditransmisikan dalam suatu satuan waktu. Pada penelitian ini, *throughput* dihitung setiap pengiriman paket data, dibandingkan dengan perbedaan *timestamp* antara pengiriman dan penerimaan paket data tersebut.

$$Throughput = \frac{sizeof(packet)}{t_1 - t_0}$$

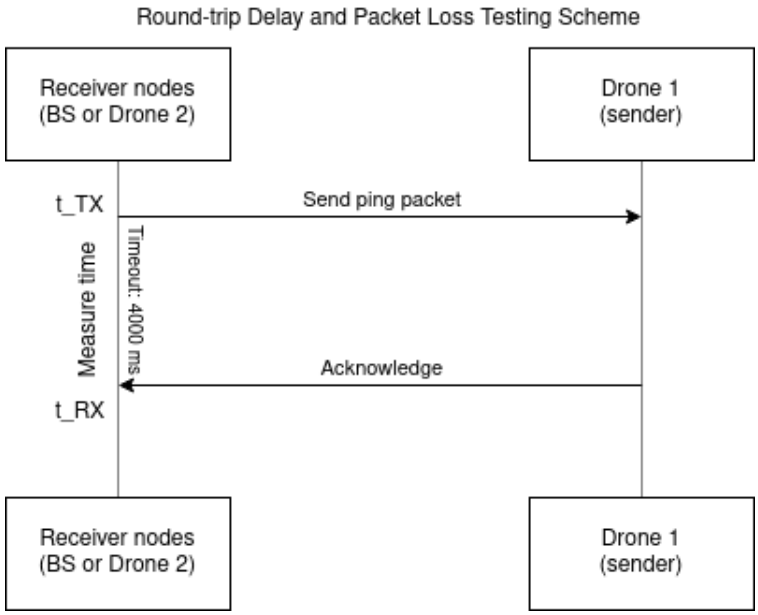
t_0 = waktu internal pada saat pengirim mengirim paket data
t_1 = *timestamp* pada saat penerima menerima paket data

2.9.3 Packet loss

Packet loss adalah jumlah packet data yang hilang dibandingkan packet data yang berhasil dikirimkan.

$$PacketLoss = (1 - \frac{p_{TX}}{ack_{RX}}) \times 100\%$$

p_{TX} = jumlah packet data yang dikirimkan
ack_{RX} = jumlah packet *acknowledge* yang diterima



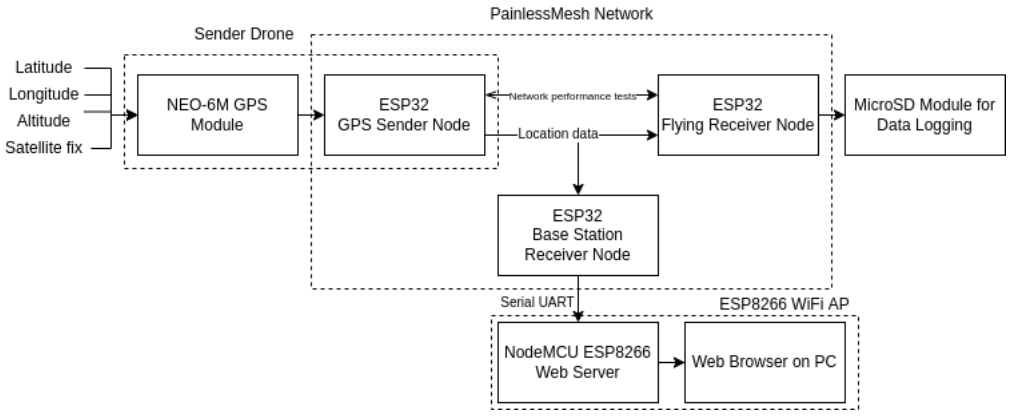
Gambar 2.7: Skema pengujian *round-trip delay* dan *packet loss*

BAB III
PERANCANGAN SISTEM

3.1 Desain Sistem

Desain dan implementasi tugas akhir ini bertujuan untuk merancang sebuah sistem algoritma komunikasi antar-UAV menggunakan jaringan mesh berbasis ESP32. Perancangan sistem terdiri dari proses perancangan blok diagram sistem, penjelasan sistem secara umum, perancangan perangkat keras sistem, serta perancangan dan penjelasan perangkat lunak sistem yang meliputi algoritma komunikasi tersebut.

3.1.1 Diagram Blok



Gambar 3.1: Diagram blok sistem.

Gambar 3.1 merupakan diagram blok dari sistem yang dirancang. Secara keseluruhan, sistem dapat dibagi menjadi 3 bagian, yakni ***Sender***, ***Flying Receiver***, dan ***Base Station Receiver***.

3.1.1.1 Sender Node

Sender node merupakan board ESP32 yang diterbangkan menggunakan drone *Victim Finder* dan bertugas menerima data lokasi dari modul GPS dan kemudian mengirimkan data lokasi tersebut berdasarkan permintaan dari *receiver*. Board ESP32 mendapatkan data lokasi dari modul GPS dengan menggunakan komunikasi serial UART pada baud rate 9600, yang kemudian modul GPS mengirimkan data lokasi berupa kalimat-kalimat NMEA. Data tersebut kemudian diolah menggunakan library TinyGPS++ untuk memudahkan penafsiran kalimat-kalimat NMEA tersebut.

Karena sistem *messaging* di PainlessMesh menggunakan JSON, maka data lokasi yang telah didapatkan (lintang, bujur, ketinggian, dan jumlah satelit yang sudah *lock*) dikirimkan berupa JSON *values* dengan *key* berupa *latitude, longitude,*

altitude, dan *satellite*. Data JSON tersebut dihitung besarnya dalam byte untuk kepentingan pengukuran *throughput*, kemudian ditambahkan *timestamp* pengiriman.

3.1.1.2 Flying Receiver Node

Receiver node ini merupakan board ESP32 yang dilengkapi board MicroSD untuk kepentingan *data logging*. Board ini akan diterbangkan oleh sebuah drone dan bertugas melakukan permintaan data lokasi kepada node Sender, sekaligus melakukan pengujian kinerja jaringan, yakni pengujian *throughput*, pengujian *packet loss*, dan pengujian *round-trip delay*. Hasil dari pengujian tersebut kemudian disimpan dalam sebuah kartu microSD untuk diolah dan dianalisis.

3.1.1.3 Base Station Receiver Node

Receiver node ini merupakan board ESP32 yang disambungkan secara serial dengan sebuah board ESP8266 yang bertugas menjadi web server untuk menampilkan data lokasi pada sebuah situs web. Konfigurasi ini diperlukan karena keterbatasan pengujian, dimana mode 802.11LR pada ESP32 merupakan mode *proprietary* yang tidak didukung oleh piranti WiFi lainnya, serta jaringan PainlessMesh yang membutuhkan mode AP dari ESP32, sedangkan board ESP32 hanya mendukung maksimal 1 AP per board. Oleh karena itu, dibutuhkan satu board lain untuk menjadi web server.

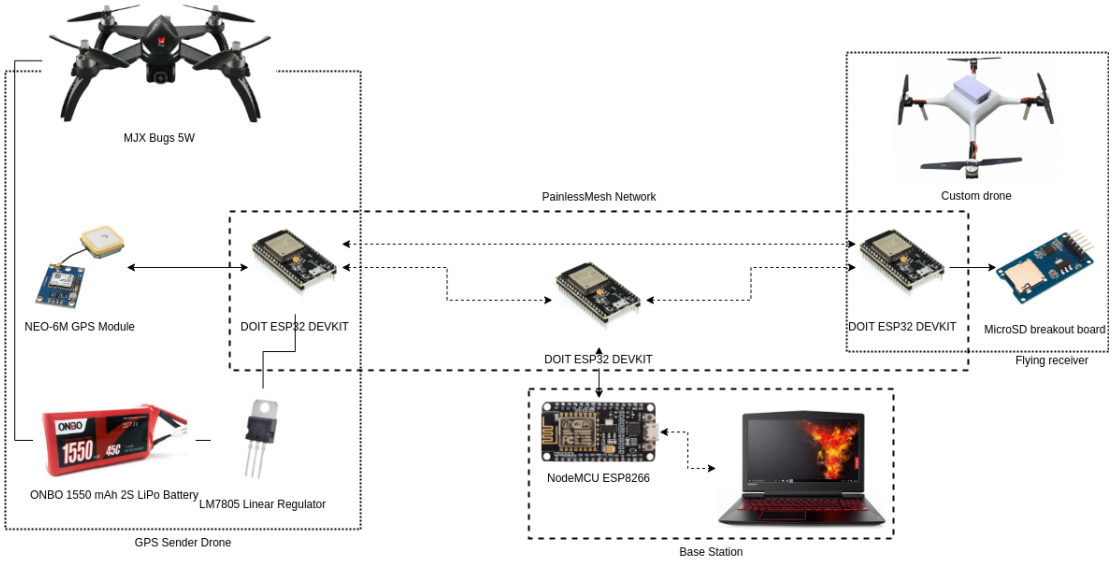
3.1.2 Fungsi dan Fitur

Sistem yang dirancang untuk tugas akhir ini memiliki fungsi komunikasi data lokasi dari satu drone kepada drone lain dan *base station*. Karena posisi antar drone dan *base station* yang bervariasi serta target penggunaan di daerah tanpa infrastruktur, maka dibutuhkan metode komunikasi yang fleksibel terhadap disrupsi dan tidak bergantung pada infrastruktur yang sudah ada, oleh karena itu digunakan sebuah jaringan mesh menggunakan PainlessMesh sehingga komunikasi dapat terjalin pada jaringan yang mampu melakukan *self-healing* dan *autoconfiguration* ketika terjadi disrupsi.

3.2 Desain Perangkat Keras

Perangkat yang digunakan pada sistem ini adalah tiga buah board DOIT ESP32 Development Kit, sebuah board NodeMCU ESP8266, sebuah modul GPS NEO-6M, sebuah board modul MicroSD, baterai Lithium-Polymer (LiPo) 2S 1500 mAh, sebuah linear regulator LM7805, dan dua buah drone. Sistem dapat menerima daya

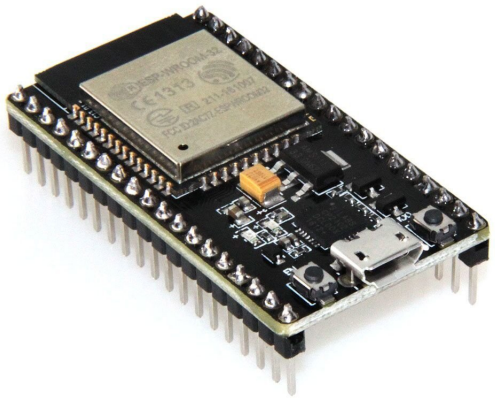
dari baterai LiPo melalui konektor *balance* JST XH, yang kemudian tegangannya diturunkan menggunakan linear regulator LM7805 ke tegangan 5V.



Gambar 3.2: Diagram perangkat keras sistem.

Perancangan sistem membutuhkan komponen-komponen pendukung untuk dapat merealisasikan sistem, diantaranya adalah board DOIT-ESP32-DEVKIT, modul GPS NEO-6M, drone MJX Bugs 5W, dan *power supply*, dengan spesifikasi sebagai berikut:

1. DOIT-ESP32-DEVKIT
   Board DOIT-ESP32-DEVKIT adalah sebuah *development kit* yang menggunakan mikrokontroler ESP32, sehingga memiliki kapabilitas Wi-Fi dan Bluetooth. Pada perancangan sistem ini, ESP32 digunakan sebagai modul komunikasi pada drone dan *base station*.



Gambar 3.3: Board DOIT-ESP32-DEVKIT

Tabel 3.1: Spesifikasi DOIT-ESP32-DEVKIT

SOC	2x Xtensa 32-bit LX6 microprocessor up to 240MHz, 448KB ROM, 520KB SRAM
Konektivitas	WiFi 802.11 (B/G/N/LR), Bluetooth
IO	18 channel ADC, 10 capacitive sensing GPIO, 3 UART interfaces, 3 SPI interfaces, 2 I^2C interfaces, 16 PWM output channels, 2 channel DAC, 2 I^2S interfaces
Masukan tegangan	5V regulated, 6V - 20V unregulated
Tegangan operasional	3.3V
Temperatur operasi	-40°C - 85°C

2. NEO-6M

   NEO-6M adalah modul GPS *receiver standalone* yang menggunakan u-blox 6 positioning engine. Pada perancangan sistem ini, NEO-6M digunakan sebagai modul GPS untuk mendapatkan data lokasi untuk dikirimkan.

Gambar 3.4: Modul GPS NEO-6M

Tabel 3.3: Spesifikasi DOIT-ESP32-DEVKIT

Time to first fix	27 detik *cold start*, 27 detik *warm start*, 1 detik *hot start*
Sensitivitas	-161dBm *(tracking and navigation)*, -160dBm *(reacquisition)*, -147dBm *(cold start)*, -156dBm *(hot start)*.
Akurasi posisi horizontal	2.5m
Masukan tegangan	2.7V - 3.6V
Maksimum arus	67mA
Temperatur operasi	-40°C - 85°C

3. *Power supply* yang digunakan pada *Sender node* dan *Flying receiver* node menggunakan sebuah linear regulator LM7805 yang dihubungkan pada sebuah baterai LiPo 2S 7.4V, sehingga menghasilkan tegangan keluaran 5V.
4. MJX Bugs 5W
   Pada perancangan sistem, drone ini digunakan untuk membawa board ESP32 dan modul GPS sebagai node *sender*.

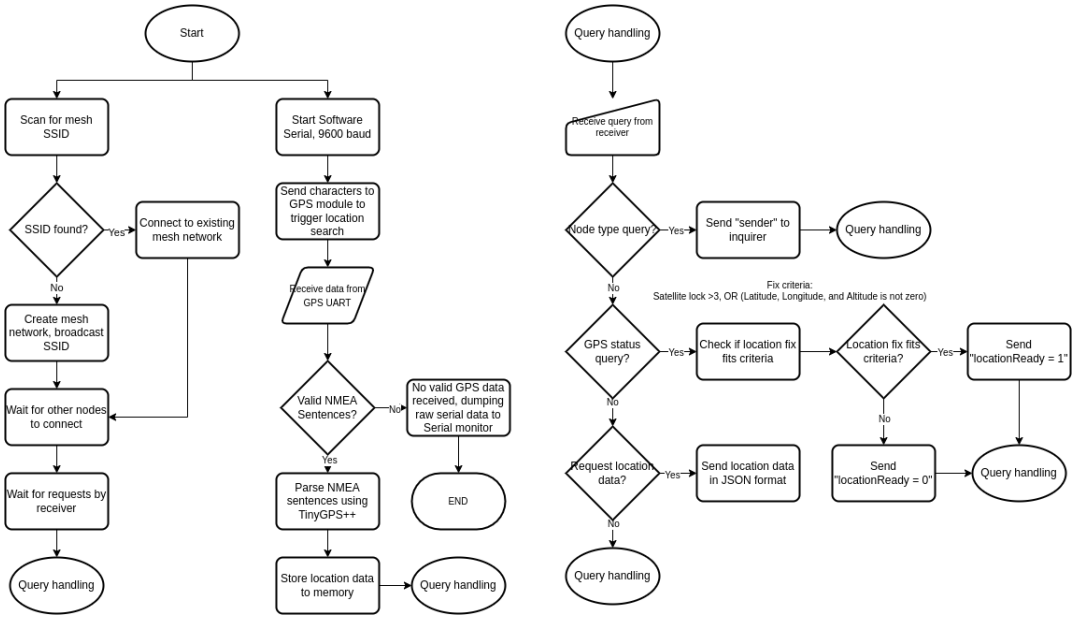
Gambar 3.5: Drone MJX Bugs 5W

Tabel 3.5: Spesifikasi drone MJX Bugs 5W

Frekuensi Remote Control	2.4GHz
Baterai drone	7.4V 2S 1800mAh
Jarak Remote Control	300-500m
Motor	1800KV BLDC
ESC	6A Brushless

3.3 Desain Perangkat Lunak

3.3.1 Sender Node



Gambar 3.6: Diagram alur sender node.

Gambar 3.6 menunjukkan diagram alur sistem pada node sender. Pada bagian kiri adalah proses inisialisasi sistem, yakni inisialisasi jaringan mesh dan inisialisasi sistem GPS. Pada proses inisialisasi jaringan mesh, board ESP32 melakukan *scanning* terhadap SSID jaringan mesh yang sudah ditetapkan, dan jika ditemukan akan mencoba bergabung dalam jaringan mesh tersebut. Jika SSID tidak ditemukan, maka board ESP32 akan membuat jaringan mesh tersebut dengan sendirinya. Setelah inisialisasi jaringan sudah selesai, maka board ESP32 akan menunggu koneksi dari node lain serta menunggu permintaan dari sebuah node receiver.

Untuk proses inisialisasi GPS, board ESP32 memulai sebuah koneksi serial UART dengan modul GPS NEO-6M menggunakan *library* Software Serial dengan baud rate 9600 bps. Modul GPS NEO-6M membutuhkan sebuah input konstan dari koneksi UART, oleh karena itu board ESP32 secara terus-menerus mengirimkan karakter kepada modul GPS tersebut. Terdapat waktu tunggu lima detik untuk menunggu respon dari modul GPS, jika tidak ada input dari modul GPS setelah lima detik sejak board pertama dinyalakan, maka board akan menghentikan eksekusi program dan menampilkan data mentah dari pin RX di Serial Monitor. Sedangkan jika modul GPS berhasil mengirimkan data melalui UART, maka eksekusi program diteruskan, dan data dari modul GPS tersebut yang berupa kalimat-kalimat NMEA akan ditafsirkan menggunakan *library* TinyGPS++ untuk mendapatkan data lokasi

yang kemudian disimpan dalam memori.

Setelah proses inisialisasi sistem selesai, maka board ESP32 akan menunggu permintaan dari node receiver. Terdapat tiga jenis permintaan/*query* yang diterima oleh node sender, yaitu:

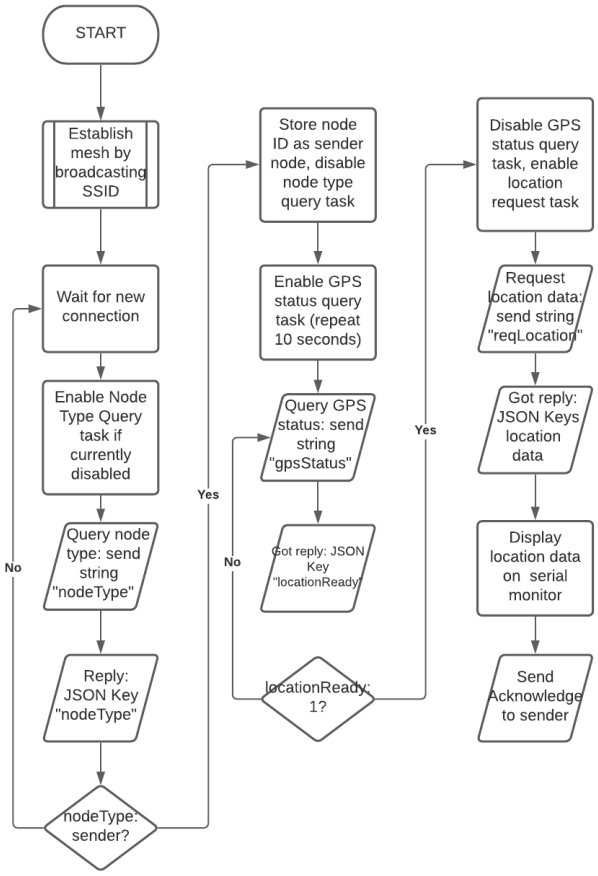
1. *Node Type Query*
   Query ini merupakan sebuah permintaan dari node receiver yang menanyakan tipe node apa yang telah terhubung dalam jaringan mesh. Respon dari *query* ini untuk node sender adalah mengirimkan sebuah pesan JSON dengan *key* ”nodeType” dan *value* ”sender”.

2. *GPS Status Query*
   Query ini adalah permintaan dari node receiver untuk mengecek status GPS node sender dan kesiapan node sender untuk mengirimkan data lokasi. Kriteria kesiapan tersebut adalah jumlah satelit yang sudah *fix* ≥ 3; atau nilai lintang, bujur, dan ketinggian tidak nol. Node sender akan mengirimkan sebuah pesan JSON dengan *key* ”locationReady” dan *value* 0 jika kriteria kesiapan GPS belum terpenuhi, dan 1 jika kriteria sudah terpenuhi.

3. *Request Location Query*
   Query ini adalah permintaan data lokasi dari node receiver. Node sender akan mengirimkan sebuah pesan JSON dengan key *latitude, longitude, altitude*, dan *satellite*, berdasarkan data yang sudah didapatkan dari modul GPS.

Node sender juga menerima permintaan pengecekan *round-trip delay* dari node receiver yang menggunakan fungsi `startDelayMeas()`. Fungsi tersebut juga digunakan untuk pengujian *packet loss*.

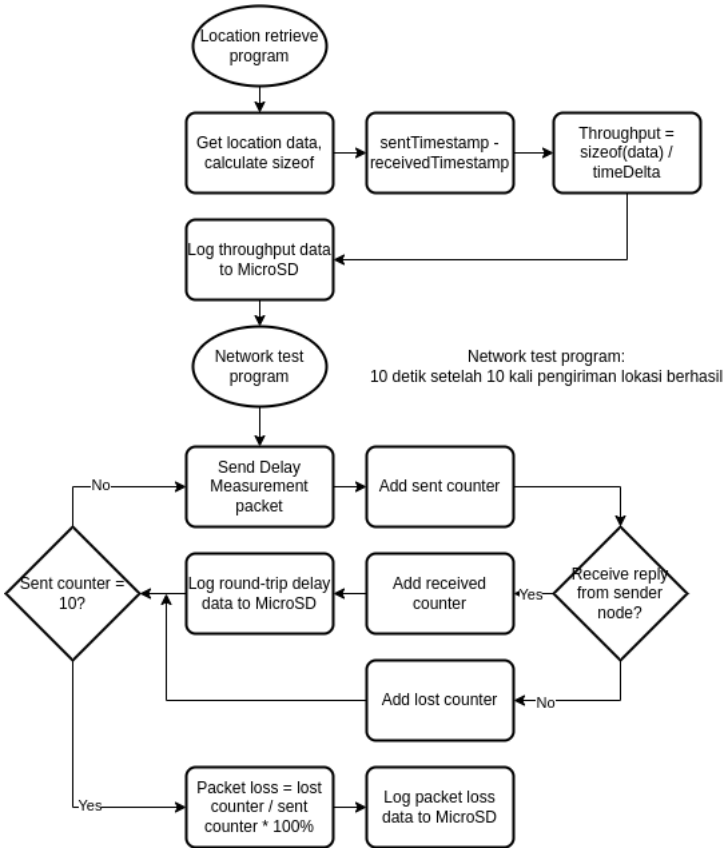
3.3.2 Flying Receiver Node



Gambar 3.7: Diagram alur node flying receiver, program penarikan data lokasi dari node sender.

Flying receiver node adalah node ESP32 yang diterbangkan menggunakan sebuah drone yang bertugas menerima data lokasi dari sender node, menguji kinerja jaringan terbang, sekaligus mencatat ke sebuah kartu memori MicroSD. Tugas-tugas tersebut dibagi menjadi dua program pada satu node, yaitu program pertama yang menarik data lokasi dari sender node, dan program kedua yang menguji kinerja jaringan setelah selesainya pelaksanaan program pertama.

Pelaksanaan masing-masing task pada node ini terjadwal menggunakan fitur *Task Scheduler*, dan dilaksanakan secara berurutan, yakni melakukan koneksi pada jaringan mesh, melakukan *query* tipe node-node lainnya pada jaringan, melakukan *query* kondisi GPS pada node sender, dan melakukan *query* data lokasi dari sender node.



Gambar 3.8: Diagram alur node flying receiver, program pengujian jaringan.

Berdasarkan yang dipaparkan pada Bab 2, pengujian kinerja jaringan yang dilakukan adalah pengujian *throughput*, *round-trip delay*, dan *packet loss*.

1. *Throughput*

   Pengujian *throughput* dilakukan bersamaan dengan program penarikan data lokasi, yakni dengan menghitung besar paket data lokasi yang dikirim, membandingkan waktu pengiriman dan waktu penerimaan paket data, lalu membagi besar paket data tersebut dengan selisih waktu pengiriman dan penerimaan paket data.

2. *Round-trip delay*

   Pengujian *round-trip delay* menggunakan fungsi `startdelayMeas()` pada library PainlessMesh yang menghasilkan *return value* sebesar nilai *delay* dalam mikrosekon.

3. *Packet loss*

   Pengujian *packet loss* menumpang pada pengujian *round-trip delay* yang dilakukan sebanyak 10 kali, lalu membandingkan berapa kali fungsi `startdelayMeas()`

dipanggil terhadap jumlah output yang didapatkan, yakni respon dari node sender.

4. *Signal strength*
   Pendataan *signal strength* dilakukan setiap kali ada permintaan data dari receiver, menggunakan fungsi `WiFi.RSSI()` dari ESP32.

DAFTAR PUSTAKA

[1] N. SYAFITRI, R. SUSANA, I. AMMARPRAWIRA, M. FAUZI, and A. JABBAAR, **?**The Autonomous Disaster Victim Search Robot using the Waypoint Method**?**, *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, p. 347, May 2020. DOI: 10.26760/elkomika.v8i2.347.

[2] C. Shimanski, *Risks in Mountain Rescue*. Mountain Rescue Association, 2008.

[3] R. G. Lakshmi Narayanan and O. C. Ibe, **?**6 - Joint Network for Disaster Relief and Search and Rescue Network Operations**?**, in *Wireless Public Safety Networks 1*, D. Câmara and N. Nikaein, Eds., Elsevier, Jan. 2015, pp. 163–193, ISBN: 978-1-78548-022-5. DOI: 10.1016/B978-1-78548-022-5.50006-6.

[4] *Drones for Search and Rescue — Learn How and Why They are Used*, https://flytnow.com/drone for-search-rescue/.

[5] M. A. Khan, A. Safi, I. M. Qureshi, and I. U. Khan, **?**Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols**?**, in *2017 First International Conference on Latest Trends in Electrical Engineering and Computing Technologies (INTELLECT)*, Karachi: IEEE, Nov. 2017, pp. 1–9, ISBN: 978-1-5386-2969-7. DOI: 10.1109/INTELLECT.2017.8277614.

[6] X. Lin, V. Yajnanarayana, S. D. Muruganathan, *et al.*, **?**The Sky Is Not the Limit: LTE for Unmanned Aerial Vehicles**?**, *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, Apr. 2018, ISSN: 0163-6804, 1558-1896. DOI: 10.1109/MCOM.2018.1700643.

[7] A. M. Townsend and M. L. Moss, *Telecommunications Infrastructure in Disasters: Preparing Cities for Crisis Communications*. New York University, Apr. 2005.

[8] *Precision of coordinates - OpenStreetMap Wiki*, https://wiki.openstreetmap.org/wiki/Precision_

[9] L. Santos, P. Nascimento, L. Bento, R. Machado, P. Ferrari, and C. Amorim, **?**Use of High Mobility Nodes to Improve Connectivity in Wireless Sensor Networks**?**, in Jan. 2021, pp. 528–545, ISBN: 978-3-030-63091-1. DOI: 10.1007/978-3-030-63092-8_36.

[10] Y. Chia, R. Arjadi, E. Setyaningsih, P. Wibowo, and M. Sudrajat, **?**Performance Evaluation of ESP8266 Mesh Networks**?**, *Journal of Physics: Conference Series*, vol. 1230, p. 012 023, Jul. 2019. DOI: 10.1088/1742-6596/1230/1/012023.

[11] M. Manvi and S. Maakar, **?**Implementing Wireless Mesh Network Topology between Multiple Wi-Fi Powered Nodes for IoT Systems**?**, vol. 7, pp. 2395–0056, Oct. 2020.

[12] Z. Guo, X. Ma, P. Zhang, and Z. Liu, **?**A dust sensor monitoring system using Wi-Fi mesh network**?**, *Journal of Physics: Conference Series*, vol. 1754, no. 1, p. 012 129, Feb. 2021, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1754/1/012129.

[13] *Wi-Fi Driver - ESP32 - — ESP-IDF Programming Guide latest documentation*, https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html.

[14] *Home · Wiki · painlessMesh / painlessMesh*, https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home.

[15] *Mesh protocol · Wiki · painlessMesh / painlessMesh*, https://gitlab.com/painlessMesh/painlessM/wikis/mesh-protocol.

[16] *GPS - NMEA sentence information*, http://aprs.gids.nl/nmea/.