Implementasi Miniatur *Mesh Network* Pada *Monitoring* Angin Kencang

1st Afrizal Bagus Rizkia
Fakultas Teknik Elektro
Universitas Telkom
Bandung,Indonesia
afrizalbagus@student.telkomuniversity.ac.id

2nd Ida Wahidah Hamzah
Fakultas Teknik Elektro
Universitas Telkom
Bandung,Indonesia
wahidah@telkomuniversity.ac.id

3rd Fardan
Fakultas Teknik Elektro
Universitas Telkom
Bandung,Indonesia
Fardanfnn@telkomuniversity.ac.id

Abstrak — Saat ini, akses mengenai informasi merupakan kebutuhan manusia yang sangat penting. Namun karena infrastruktur komunikasi yang terbatas, tidak semua daerah memiliki akses informasi yang memadai, terutama di daerah terpencil. Salah satu pemanfaatan teknologi *wireless mesh network* (WMN) dapat menjadi solusi untuk mengatasi permasalahan keterbatasan infrastruktur jaringan di daerah. Untuk meningkatkan performa WMN, maka perlu adanya rancangan mekanisme pengiriman informasi yang efisien. Pada penelitian ini akan dilakukan berbagai macam jenis pengujian seperti, melakukan uji fungsionalitas dari node. Pada penelitian ini dilakukan pengujian terhadap QoS (*Quality of Service*) seperti *throughput, delay*, dan *packet loss*. Pada pengujian QoS arsitektur WMN menunjukan bahwa kualitas layanan dipengaruhi oleh jarak,jumlah *node* yang dilewati dan material yang menghalangi. Pada pengujian QoS yang telah dilakukan menunjukkan hasil *throughput* sebesar 625,1 bps, *delay* 1,356 ms dan paket loss sebesar 0%.

Kata kunci— WMN, QoS, Jaringan

I. PENDAHULUAN

Mitigasi bencana merupakan serangkaian upaya untuk mengurangi dampak bencana sebelum bencana terjadi, termasuk kesiapan dan tindakan-tindakan untuk mengurangi risiko pada jangka waktu yang lama. Usaha dari tindakan mitigasi bisa dilakukan pada saat prabencana, saat bencana dan pasca bencana. Prabencana merupakan kesiapan atau pemahaman masyarakat mengenai informasi mitigasi mengenai upaya untuk mengantisipasi terjadinya bencana [1]

Bencana angin kencang merupakan salah satu bencana yang sering terjadi di Indonesia yang dapat mengakibatkan kerusakan ringan hingga berat atau bahkan timbulnya korban jiwa. Pembuatan monitoring angin kencang sederhana ini dibuat untuk mengatasi berbagai permasalahan yang ada seperti minimnya mitigasi mengenai angin kencang, terdapat titik yang tidak tercakup alat milik BMKG, dan mahalnya alat *automatic weather station* BMKG.

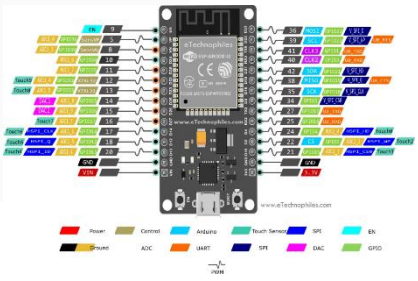
Pemanfaatan dari pembuatan monitoring angin kencang memiliki manfaat yang sangat penting di berbagai bidang seperti: pertanian, militer, tempat wisata, dan lain sebagainya. Berdasarkan kegiatan-kegiatan tersebut, maka perlu dilakukan pemantauan angin kencang sebagai langkah antisipasi awal untuk memperkecil dampak seperti kecelakaan kerja, atau bahkan timbulnya korban jiwa.

II. KAJIAN TEORI

Pada topik ini, implementasi *wireless mesh network* digunakan untuk proses pengiriman data antar *node* sebelum dikirimkan ke *database*. Terdapat beberapa spesifikasi produk yang digunakan yaitu ESP32, pada implementasi *wireless mesh network* penulis menggunakan *visual studio code* dengan librari *painlessmesh* untuk mengelola program *mesh network* dan pengujian *quality of services* menggunakan *software wireshark*.

A. ESP32

ESP32 merupakan sebuah mikrokontroler SoC yang tangguh dengan Wi-Fi 802.11 b/g/n yang terintegrasi, Bluetooth dengan mode ganda versi 4.2, dan berbagai periferal lainnya. Mikrokontroler ini merupakan pengembangan dari chip 8266 terutama dalam hal implementasi dua core clock dengan versi yang berbeda, yaitu hingga 240 MHz. Selain fitur-fitur tersebut, ESP32 juga terdapat peningkatan jumlah GPIO dari 17 pin menjadi 36 pin, jumlah saluran PWM sebesar 16, dan terdapat memori flash dengan ukuran 4MB.



GAMBAR 1
ESP32

Pada gambar 1 merupakan perangkat ESP32 yang digunakan sebagai mikrokontroller untuk memproses data yang didapatkan dari sensor, dan mengirimkan ke *database*.

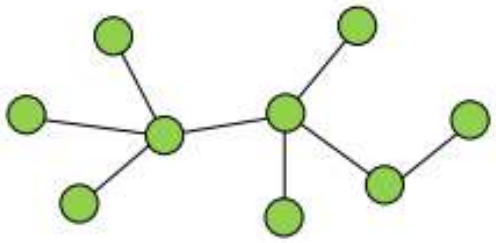
B. *Wireless Mesh Network*

Wireless Mesh Network (WMN) merupakan salah satu jaringan nirkabel atau tanpa kabel dengan menggunakan topologi *mesh*. Setiap *node* dalam jaringan akan saling

berkomunikasi untuk saling mengatur *node* yang digunakan. Pada saat proses kirim ataupun penerimaan data melalui sebuah *node*, setiap *node* berfungsi untuk membuat jalur pengiriman sendiri, setiap *node* bekerja sama untuk menyelesaikan tugasnya, meskipun satu *node* tidak dapat bekerja, *node* lain dapat mengambil alih proses komunikasi dengan mengatur dirinya sendiri

C. *Painlessmesh*

Painlessmesh merupakan suatu librari jaringan *mesh WiFi* yang dikhususkan penggunaanya untuk ESP8266 dan ESP32. Librari *painlessmesh* memungkinkan pada perangkat untuk menjadi konfigurasi otomatis dan mudah diatur dalam algoritma dari *routing*, *self-organizing network*, toleransi kesalahan dan terdiri dari beberapa gateway. *Painlessmesh* adalah jaringan ad-hoc yang tidak memerlukan rencana perutean maupun pusat pengontrol, dan semua *node* sama [2], [3]

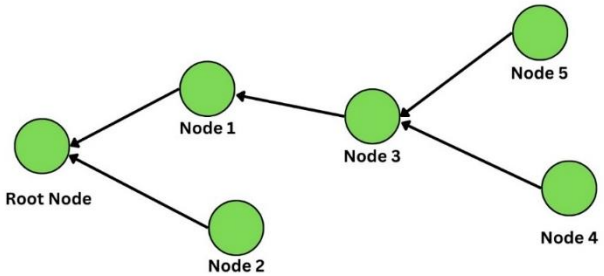


GAMBAR 2
Diagram Jaringan *Painlessmesh*

Gambar 2 merupakan jaringan *painlessmesh* yang terdiri dari beberapa *node*.

D. *Multihop Mesh*

Jaringan sensor nirkabel mempunyai batas jangkauan *node* bisa berkomunikasi, sehingga *node* yang berada diluar jangkauan *root node* tidak bisa mengirimkan data pada *root node* secara langsung, namun dengan Sistem komunikasi *multihop* memungkinkan *node* yang berada diluar jangkauan *root node* bisa berkomunikasi antar *node* terdekat terlebih dahulu, setelah mengirimkan data pada *node* terdekat, data pada *node* akan dikirim lagi melawati *node* lain sampai data yang dikirim oleh *node* awal bisa sampai pada *root node*.



GAMBAR 3
Multihop Mesh

Pada gambar 3 yaitu pada saat data dari *node* 5 dan *node* 4 dikirim ke *root node* akan melalui *node* 3 dan *node* 1.

E. *Visual Studio Code*

Visual studio merupakan salah satu *software* yang digunakan untuk melakukan pengembangan seperti menulis, mengedit, melakukan debug dan membuat kode tertentu. Pada visual studio juga terdapat beberapa fitur seperti pengeditan, pengecekan kesalahan kode, kompilator kode, ekstensi dan masih banyak lagi fitur yang dapat digunakan.

GAMBAR 4
Logo Visual Studio Code

F. *Wireshark*

Wireshark merupakan sebuah aplikasi yang dapat melakukan capture paket data untuk memantau, menganalisis, dan mencatat lalu lintas trafik pada jaringan internet. Penggunaan *wireshark* diimplementasikan pada pengujian parameter QoS untuk mengamati pengiriman data antar *node* pada jaringan *mesh network*, parameter yang diamati yaitu *throughput*, *delay* dan *packet loss*.

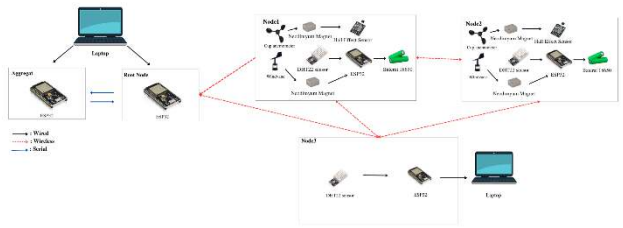
GAMBAR 5
Logo *Wireshark*

III. METODE

Penelitian yang dilakukan yaitu mengenai pengamatan kekuatan sinyal, lama waktu sinkronisasi antar *node* dan *quality of services* pada jaringan *mesh*. Pengujian kekuatan sinyal dilakukan dengan cara menguji kekuatan sinyal antar *node* pada area line of sight (LOS) berada di area terbuka dengan jarak pengujian 10 meter – 50 meter. Pada pengujian lamanya waktu sinkronisasi dilakukan dengan cara menghubungkan *node* ke dalam jaringan *mesh* dan pengujian *quality of services* dilakukan dengan menguji seluruh *node* untuk melakukan proses pengiriman data antar *node* yang diimplementasikan pada jaringan *mesh* menggunakan *software wireshark*. Pengujian *multihop* dilakukan dengan cara menempatkan *node* dititik terjauh dari *root node* dan akan dilakukan proses pengiriman melalui *node* terdekat.

A. Topologi Jaringan

Topologi jaringan yang digunakan yaitu menggunakan implementasi dari *painlessmesh network*. Dengan *painlessmesh*, *node* tidak perlu dihubungkan ke *node* pusat. *Node* bertanggung jawab untuk membuat jalur yang dapat dilalui satu sama lain. Ini memungkinkan beberapa *node* dapat tersebar di area yang luas. *Node* dapat dapat berfungsi secara dinamis dan dapat mengirimkan paket satu sama lain untuk memastikan bahwa paket mencapai tujuan akhir *node*.



GAMBAR 6
Implementasi Jaringan *Painlessmesh*

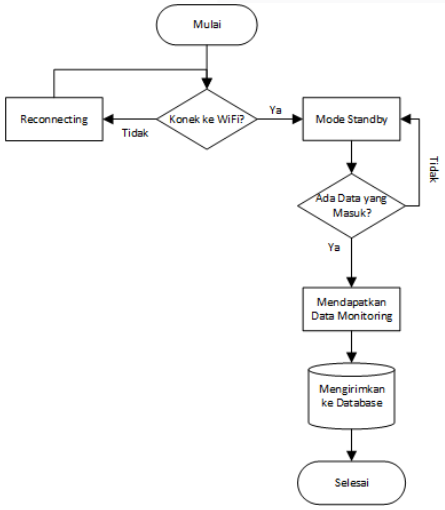
Pada gambar 6 merupakan hasil implementasi *painlessmesh*, pada terdiri dari satu buah laptop yang berfungsi untuk sumber daya dari aggregat dan *root node*. Pada pengujian terdiri dari 3 buah *node*, dimana pada *node* 1 dan *node* 2 menggunakan mikrokontroller ESP32, sensor *hall effect* untuk mendeteksi kecepatan angin dan arah angin serta menggunakan DHT22 untuk mendeteksi suhu dan kelembaban udara. Pada *node* 3 terdiri dari mikrokontroller ESP32 dan DHT22, seluruh *node* dihubungkan dengan menggunakan jaringan *WiFi mesh*.

B. Desain Sistem Jaringan *Mesh*

Implementasi jaringan *mesh* dibedakan menjadi 3 sistem yaitu aggregat. *root node* dan *node*.

1. Aggregat

Aggregat merupakan hasil implementasi dari ESP32 untuk melakukan proses pengiriman data hasil *monitoring* dari *node* untuk disimpan ke dalam *database*.

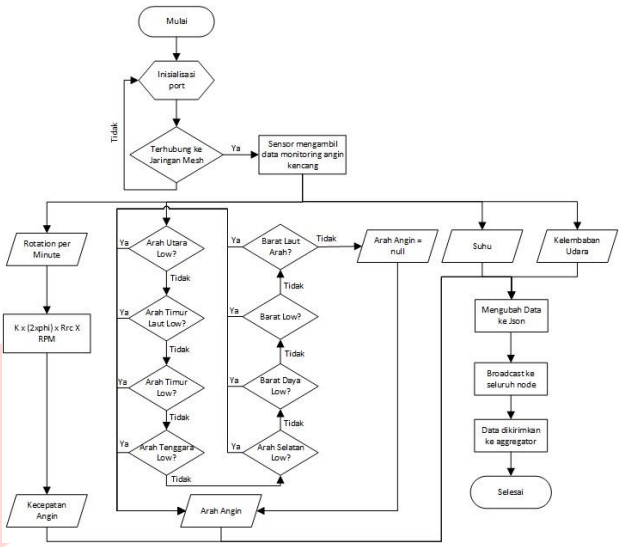


GAMBAR 7
Sistem Kerja Aggregat

Berdasarkan gambar 7 aggregat akan bekerja dengan cara melakukan penerimaan data yang dikirim oleh *node* 1-3 dan melakukan proses pengiriman data kepada *cloud database* untuk dilakukan penyimpanan. Aggregat akan menunggu data masuk pada saat mode menunggu data diterima, pada saat data *monitoring* diterima akan diteruskan untuk dilakukan proses penyimpanan pada cloud *database.*

2. *Node*

Node pada jaringan *mesh* juga memiliki fungsi yang sama seperti *root node* namun berfungsi sebagai *client* untuk memperluas coverage jaringan *mesh*.



GAMBAR 8
Sistem Kerja *Node*

Pada gambar 8 menjelaskan mengenai cara kerja dari *node* 1-3 yang digunakan, diawali dengan melakukan inisialisasi port dan menyalakan alat, setelah alat menyala menghubungkan ke dalam jaringan *mesh WiFi*, jika tidak berhasil terhubung ke dalam *WiFi* maka mencoba untuk menghidupkan dan mematikan tombol switch dan jika berhasil *node* akan melakukan proses pengambilan data sensor seperti data kecepatan angin, arah angin, suhu dan kelembaban udara. Pada seluruh *node* akan dilakukan dengan metode broadcast, di mana data akan dikirimkan ke seluruh *node* untuk mendapatkan informasi data dari masing-masing *node* dan akan dikirimkan ke aggregat untuk dilakukan proses pengiriman data menuju cloud *database*.

3. *Root Node*

Root node pada jaringan *mesh* difungsikan untuk membentuk sebuah *gateway* sekaligus *router* untuk menjalankan proses *routing*.

C. *Quality of services* (QoS)

Quality of services merupakan metode pengukuran untuk mengetahui seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu jaringan.

1. *Throughput*

Throughput adalah kecepatan dari jumlah transfer data yang dihitung dalam satuan bps. *Throughput* yaitu jumlah total paket yang diamati selama interval waktu. Untuk menghitung nilai *throughput* menggunakan rumus.

$$Throughput = \frac{Paket\ data\ diterima}{Lama\ pengamatan} \quad (1)$$

Throughput dihitung dengan menggunakan rumus total jumlah paket diterima dibagi dengan lama waktu pengamatan. Indeks kategori *throughput* berdasarkan standar TIPHON sebagai berikut:

TABEL 1
Kategori Indeks *Throughput*

Kategori	*Throughput*	Indeks
Sangat Bagus	100 bps	4
Bagus	75 bps	3
Sedang	50 bps	2
Jelek	< 25 bps	1

Sumber : TIPHON

Berdasarkan tabel 1 menunjukkan bahwa semakin besar *throughput* maka menandakan jaringan yang digunakan semakin bagus. Jika besar *throughput* lebih dari 100 bps maka jaringan tersebut dapat dikategorikan sangat bagus dengan nilai indeks bernilai 4, sedangkan jika nilai *throughput* kurang dari 25 bps dikategorikan jaringan sangat jelek dengan nilai indeks bernilai 1.

2. *Delay*

Delay merupakan waktu yang dibutuhkan oleh sebuah paket yang berasal dari pengirim ke penerima melalui jaringan internet. Hal tersebut dapat mengakibatkan masalah sinkronisasi pada alat. *Delay* dapat dipengaruhi oleh jarak antar *node*, media yang dilalui serta gangguan frekeunsi.kongesti atau juga waktu proses yang lama. Untuk menghitung *delay* dapat menggunakan rumus sebagai berikut:

$$Delay = \frac{Total\ variasi\ delay}{Total\ paket\ yang\ diterima} \quad (2)$$

Delay dihitung dengan menggunakan rumus total variasi *delay* dibagi dengan total paket yang diterima. Indeks kategori *delay* berdasarkan standar TIPHON sebagai berikut:

TABEL 2
Kategori Indeks *Delay*

Kategori	Besar *Delay*	Indeks
Sangat Bagus	<150 ms	4
Bagus	150 ms s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Jelek	>450 ms	1

Sumber : TIPHON

Berdasarkan tabel 2 dapat dilihat bahwa semakin kecil *delay* menandakan jaringan semakin bagus ditandai dengan nilai indeks yang besar, semakin besar nilai indeks maka semakin bagus. Apabila nilai *delay* <150 ms menandakan jaringan sangat bagus dengan ditandai nilai indeks 4, sedangkan untuk yang paling buruk yaitu apabila nilai *delay* >450 ms ditandai dengan nilai indeks yang menandakan kualitas jaringan buruk.

3. *Packet loss*

Packet loss merupakan persentase paket yang hilang pada saat proses mentransmisikan data. *Packet loss* merupakan nilai parameter yang menggambarkan tentang kondisi dari jumlah total paket yang hilang, paket hilang dapat terjadi karena tabrakan antar paket dan congestion pada jaringan. Untuk menghitung nilai *packet loss*, menggunakan rumus berikut :

$$Packet\ loss = \left[\left(\frac{(Paket\ dikirim - Paket\ diterima)}{Paket\ diterima}\right) x\ 100\right] \quad (3)$$

Packet loss dapat dihitung menggunakan rumus dari jumlah paket dikirim dikurangi dengan jumlah paket diterima dan dibagi dengan paket diterima dikalikan dengan 100 untuk menghasilkan persentase paket loss. Indeks kategori *packet loss* berdasarkan standar TIPHON sebagai berikut:

TABEL 3
Kategori Indeks *Packet loss*

Kategori	Besar *Packet loss*	Indeks
Sangat Bagus	0%	4
Bagus	1-3%	3
Sedang	4-15%	2
Jelek	16-25%	1

Sumber : TIPHON

Berdasarkan tabel 3 menunjukkan bahwa semakin kecil persentase *packet loss* dari hasil pengiriman menandakan jaringan sangat baik, sedangkan ketika persentase *packet loss* semakin besar maka menandakan jaringan yang digunakan buruk. Jika nilai persentase *packet loss* berkisar diantar 0-2,99 % menandakan jaringan yang digunakan sangat bagus ditandai dengan nilai indeks 4, sedangkan jika *packet loss* lebih dari 25% menandakan jaringan yang digunakan sangat buruk ditandai dengan nilai indeks 1.

D. Sinkronisasi Jaringan *Mesh*

Sinkronisasi waktu merupakan salah satu hal yang paling penting pada sistem pengiriman data. Namun terdapat keterbatasan yang berada pada jaringan *wireless* sehingga membutuhkan pengembangan dibandingkan dengan jaringan kabel. Berdasarkan pengaplikasian yang yang telah banyak digunakan pada jaringan *wireless*, peningkatan waktu seperkian milidetik dapat mempengaruhi performa aplikasi secara signifikan

E. RSSI (*Received Signal Strength Indicator*)

RSSI merupakan suatu pengukuran terhadap daya pancar dari sebuah perangkat *wireless*. Pengukuran dilakukan berdasarkan kekuatan sinyal yang diterima, berdasarkan hal tersebut RSSI memiliki tujuan untuk mengetahui tingkat akurasi dari percobaan pengukuran kekuatan sinyal dari perangkat *wireless* [4]. RSSI digunakan untuk menunjukkan kekuatan sinyal yang diterima oleh *receiver* dari pemancar, kekuatan sinyal *wireless* menggunakan satuan dBm sebagai satuan pengukuran. Rentang kekuatan sinyal dimulai dari -10 dBm sampai dengan kurang lebih -100 dBm. Jika kekuatan sinyal semakin mendekati positif maka kualitas sinyal yang dipancarkan semakin bagus.

F. *Multihop* Data

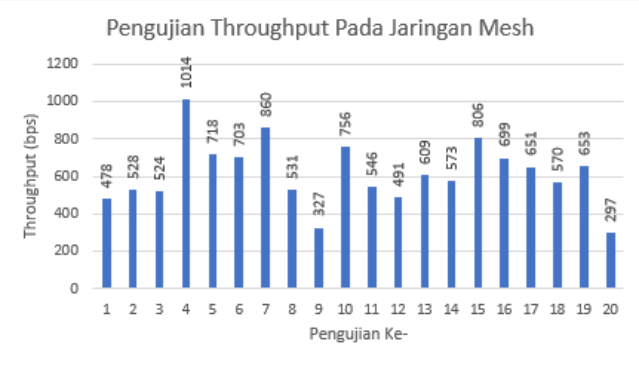
Perancangan sistem didasarkan pada perencanaan area simulasi, jumlah *node* sensor, dan sistem komunikasi antar *node* dengan *root node*. Sebagai salah satu solusi maka komponen yang berada pada jaringan wireless sensor dirancang supaya dapat melakukan komunikasi secara *multihop*. Dengan menggunakan metode mutihop ini maka *node-node* yang terletak jauh dari *root node* dapat melakukan komunikasi melalui *node-node* yang terdekat [5].

IV. HASIL DAN PEMBAHASAN

Pada pengujian implementasi miniatur jaringan *mesh* untuk monitoring angin kencang dilakukan dengan cara melakukan pengujian *quality of services* seperti *throughput*, *delay* dan *packet loss*, menguji lama waktu sinkronisasi pembuatan jaringan *mesh* pada tiap *node*, melakukan uji kekuatan sinyal setiap *node* dan pengujian pengiriman data melalui *multihop mesh*.

A. *Quality of services*

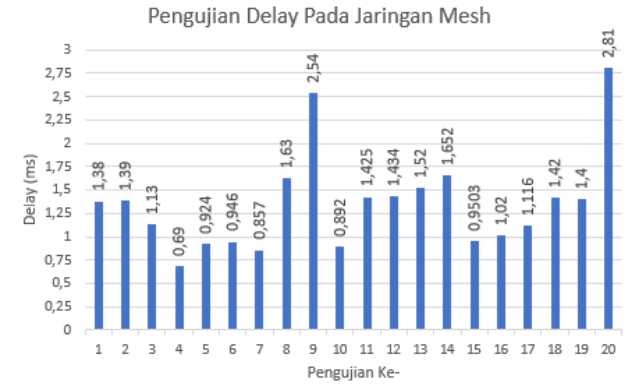
1. *Throughput*



GAMBAR 9
Hasil Pengujian *Throughput*

Pada gambar 9 menjelaskan mengenai pengujian *throughput* pada jaringan *mesh*, pengujian dilakukan sebanyak 20 kali dengan masing-masing waktu pengujian selama 10 menit. Pengujian *throughput* tertinggi terjadi pada pengujian ke 4 dengan total 1014 bps dan terendah pada pengujian ke 20 dengan total *throughput* 297 bps dan rata-rata hasil yang didapatkan yaitu 625,1 bps. Perbedaan paling jauh terjadi pada saat pengujian data ke-4 dan data ke-20 karena dipengaruhi oleh lamanya waktu sinkronisasi seluruh *node*, dan interferensi jaringan *mesh*.

2. *Delay*



GAMBAR 10
Hasil Pengujian *Delay*

Berdasarkan pada gambar 10 didapatkan hasil pengujian *delay mesh network* menggunakan *software wireshark*, pengujian dilakukan sebanyak 20 kali dan dilakukan masing-masing selama 10 menit. Hasil yang didapatkan yaitu *delay* tertinggi terjadi pada pengujian ke-20 dengan *delay* sebesar 2,81 ms sedangkan yang terendah pada pengujian ke-4 dengan rata-rata *delay* dari 20 kali pengujian yaitu 1,356 ms. Pada pengujian *delay* terdapat perbedaan *delay* pada pengujian ke 20 dan pengujian ke-4, dimana *delay* dipengaruhi oleh berbagai macam gangguan seperti halangan yang menutupi antena pada ESP32, gangguan frekeunsi radio, dan konfigurasi jaringan yang digunakan.

3. *Packet loss*

Hasil pengujian *packet loss* dari 20 kali percobaan menggunakan *software wireshark* dengan waktu pengamatan setiap 10 menit didapatkan rata-rata *packet loss* sebesar 0% karena pada saat pengujian menggunakan *wireshark* dilakukan dengan menguji privat jaringan *mesh*, sehingga minim terjadinya gangguan pengiriman paket data.

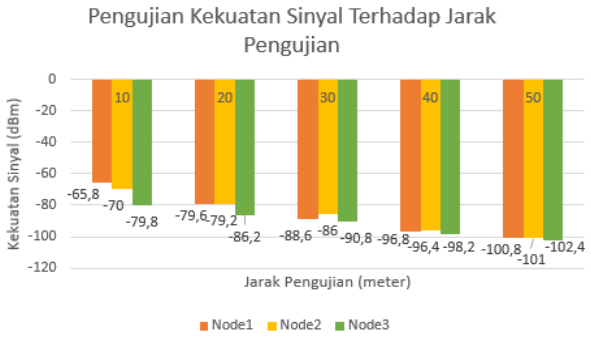
B. Sinkronisasi Jaringan *Mesh*

TABEL 4
Hasil Pengujian Sinkronisasi Jaringan

No	Pengujian	1	2	3	4	Rata-rata
1.	2 *node*	6,54	3,4	1,7	1,9	3,38
2.	3 *node*	53,3	17,4	6,5	55,3	33,12
3.	4 *node*	74,1	13,3	16,3	12,8	29,12

Pada tabel 4 dilakukan pengujian waktu sinkronisasi dari jaringan *mesh* yang terdiri dari 2 *nodes*, 3 *nodes* dan 4 *nodes*. Pengujian dilakukan dengan cara mengambil waktu proses yang dibutuhkan untuk melakukan pembuatan jaringan *mesh*. Pada saat terdiri dari 2 *nodes* waktu yang dibutuhkan untuk melakukan sinkronisasi rata-rata sebesar 3423,75ms, pada saat 3 *nodes* rata-rata waktu yang dibutuhkan yaitu 33183,25ms dan saat jaringan *mesh* terdiri dari 4 *nodes* waktu rata-rata yang dibutuhkan untuk melakukan sinkronisasi *mesh* yaitu 29182,52ms. Waktu sinkronisasi antar *node* dipengaruhi oleh komputer yang digunakan, jaringan sensor yang digunakan, kondisi suhu alat, phase *noise*, frekeunsi *noise* dan gangguan jam.

C. RSSI (*Received Signal Strength Indicator*)

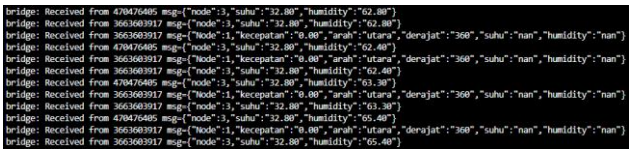


GAMBAR 11
Pengujian Kekuatan Sinyal

Pada gambar 11 menggambarkan pengujian kekuatan sinyal pada *node* yang diuji. Pada proses pengujian kekuatan pancaran sinyal yaitu *node* 1 dan 2 yang terdiri dari sensor *hall effect* dan DHT22 sedangkan *node* 3 terdiri dari sensor DHT22. Pengujian dilakukan berada pada area *line of sight* dan berada di luar ruangan. Berdasarkan grafik pada gambar 10 dapat disimpulkan bahwa semakin jauh jarak *node* dari *root node* maka sinyal yang dipancarkan akan semakin kecil dan buruk.

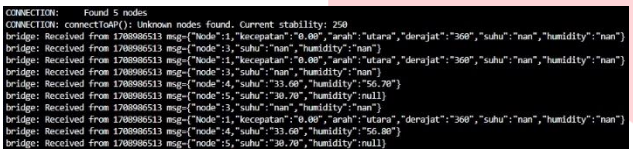
D. *Multihop* Data

Pada pengujian pengiriman data *multihop* pada jaringan *mesh* menggunakan sebuah *node* yang terdiri dari DHT22 yang ditempatkan jauh dari *root node*. Pada pengujian yang pertama *node* terjauh melalui satu kali hop untuk sampai ke *root node* dan pada pengujian ke dua melalui dua kali hop sebelum data sampai ke *root node*.



GAMBAR 12
Hasil Pengujian Melewati 1 Hop

Pada gambar 12 adalah hasil *screenshoot* dari *serial monitor* pada *root node*, data *node* 3 yang dikirim ke *root node* melalui *node* 1 dan data yang dikirim dari *node* 3 dan *node* 1 berhasil diterima pada *root node*.



GAMBAR 13
Hasil Pengujian Melewati 2 Hop

Pada gambar 13 adalah hasil *screenshoot* dari *serial monitor* pada *root node*, data *node* 5 dan *node* 4 yang dikirim ke *root node* melalui *node* 3 kemudian dari *node* 3 data dikirim ke *root node* melalui *node* 1, dan data yang dikirim dari *node* 5 dan *node* 4 berhasil diterima pada *root node* dengan skema pengujian seperti pada gambar 3.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan pada jaringan *mesh*, mendapatkan hasil pengujian *throughput* didapatkan rata-rata nilai 625,1 bps, pada pengujian *delay* didapatkan waktu *delay* dengan rata-rata 1,356 ms dengan rata-rata *packet loss* sebesar 0%. Pada pengujian waktu sinkronisasi pembuatan jaringan mesh pada 2 *node*, 3 *node* dan 4 *node* lama pembuatan jaringan mesh berbeda-beda setiap jumlah *node* yang ditambahkan berbeda. Sedangkan Sedangkan pada pengujian kekuatan sinyal (RSSI) setiap *node* di area *line of sight* dapat disimpulkan bahwa semakin jauh jarak *node* dari *root node* maka sinyal yang dipancarkan semakin kecil atau melemah. Pada pengujian *multihop mesh* dapat dikatakan berhasil dikarenakan data yang dikirimkan *node* terjauh dari *root node* dapat diterima dengan baik.

REFERENSI

[1] I. S. Wekke, *Mitigasi Bencana*. Penerbit Adab, 2021.

[2] R. H. Arjadi, E. Setyaningsih, P. Wibowo, and M. I. Sudrajat, “Performance evaluation of ESP8266 *mesh network*s,” in *journal of physics: Conference series*, IOP Publishing, 2019, p. 012023.

[3] M. V Ushakova, Y. A. Ushakov, P. N. Polezhaev, and V. N. Tarasov, “Model of the distributed self-organizing *network* of IoT sensors,” in *2018 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC)*, IEEE, 2018, pp. 1–6.

[4] N. F. Puspitasari, “Analisis Rssi (Receive Signal Strength Indicator) Terhadap Ketinggian Perangkat Wi-Fi Di Lingkungan Indoor,” *Data Manajemen Dan Teknologi Informasi (DASI)*, vol. 15, no. 4, p. 32, 2014.

[5] R. SUSANA, F. HADIATNA, and A. GUSMANTINI, “Sistem *Multihop* Jaringan Sensor Nirkabel pada Media Transmisi Wi-Fi,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 9, no. 1, p. 232, 2021.