

Sistem *Multihop* Jaringan Sensor Nirkabel pada Media Transmisi Wi-Fi

RATNA SUSANA, FEBRIAN HADIATNA, APRIANTI GUSMANTINI

Teknik Elektro Institut Teknologi Nasional Bandung, Indonesia
Email: ratnassn@yahoo.com

Received 30 November 2020 | *Revised* 12 Desember 2020 | *Accepted* 12 Januari 2021

ABSTRAK

Dengan menerapkan sistem multihop pada jaringan sensor nirkabel, pembacaan kondisi lingkungan dapat dilakukan pada lingkungan yang lebih luas. Pada penelitian ini, sistem multihop jaringan sensor nirkabel menggunakan platform IoT NodeMCU V3 yang memiliki modul Wi-Fi ESP8266. Jumlah sensor node yang digunakan merupakan batas maksimal client yang dapat terhubung kepada Wi-Fi ESP8266, yaitu 1 sink node dan 4 sensor node. Sensor node akan mengirimkan datanya kepada sink node, kemudian data tersebut akan dikirimkan kepada website untuk ditampilkan pada dashboard Adafruit.io. Pengiriman data diuji menggunakan 2 topologi yaitu bus dan tree. Berdasarkan hasil pengujian, jarak maksimal pengiriman data pada topologi bus tanpa penghalang adalah 72 meter dengan delay pengiriman 64 detik dan topologi tree adalah 108 meter dengan delay pengiriman 14 detik. Sistem multihop pada topologi bus dan tree dapat mengirim data dengan 2 penghalang yang memiliki ketebalan 15 cm dengan delay pengiriman 29 detik pada topologi bus dan 14 detik pada topologi tree.

Kata kunci: jaringan sensor nirkabel, multihop, Wi-Fi, NodeMCU V3

ABSTRACT

By applying a multihop system on wireless sensor network, reading environment condition can be done in wider environment. In this study, multihop system in wireless sensor network uses IoT NodeMCU V3 platform which has a Wi-Fi ESP8266 module. The amount of node sensor is the maximum limit of client which can be linked to Wi-Fi access point in Wi-Fi ESP8266 module, i.e 1 sink node and 4 sink node. The node sensor will transfer the data to the sink node, then the data will be transfered to the website to be shown on Adafruit.io dashboard. The transmission data is tested using 2 topologies, i.e bus and tree. Based on the test, the maximum distance of data transmission in bus topology without barrier is 72 meters with delivery delay which takes 64 seconds and in tree topology is 108 seconds with delivery delay which takes 14 seconds. The multihop system in the bus topology and the tree topology can send the data with 2 barriers which has 15 cm width and delivery delay among the nodes which takes 29 seconds in bus topology and 14 seconds in tree topology.

Keywords: wireless sensor network, multihop, Wi-Fi, NodeMCU V3

1. PENDAHULUAN

Jaringan sensor nirkabel merupakan jaringan dengan sekumpulan *sensor node* yang tersebar. *Sensor node* pada jaringan sensor nirkabel memiliki kemampuan untuk mengumpulkan data, berkomunikasi dengan *sensor node* lain serta menyampaikan data menuju *sink node*. Aplikasi *Internet of Things* (IoT) yang saat ini berkembang, salah satunya adalah memanfaatkan jaringan sensor nirkabel. Data-data dari *sensor node* yang terhubung dalam suatu jaringan, dapat dikirimkan ke *cloud* melalui internet yang selanjutnya dapat ditampilkan melalui *smartphone* atau *personal computer* yang terkoneksi dengan internet. Implementasi *sensor node* pada sistem berbasis IoT telah dikembangkan pada sejumlah penelitian yang mengembangkan sistem *monitoring* seperti *monitoring* kualitas air dan tanah pertanian (Syafiqoh, dkk, 2018), sistem *monitoring* cuaca (Babu, dkk, 2018) (Nikhilesh, dkk, 2020) (Islam, 2019), *monitoring* lingkungan (Shirsath & Waghile, 2018) dan *monitoring* tempat penyimpanan benih kedelai (Sasono, dkk, 2017).

Ada beberapa parameter yang dapat diuji untuk mengetahui kinerja dari suatu jaringan sensor nirkabel, seperti *delay* pengiriman data, jarak komunikasi antar *sensor node*, *throughput*, jumlah *sensor node* terpasang ataupun *Received Signal Strength Indicator* (RSSI). Kualitas suatu sistem berbasis jaringan sensor nirkabel salah satunya tergantung dari modul *transceiver* yang digunakan sebagai media transmisi data. Pemilihan modul *transceiver* akan tergantung pada kebutuhan sistem itu sendiri., oleh karena itu hasil pengujian terhadap modul *transceiver* dipandang perlu dilakukan agar dapat dijadikan dasar untuk melakukan desain sistem yang menggunakan jaringan sensor nirkabel. Evaluasi karakteristik XBee Pro dan nRF24L01+ yang dilakukan oleh Fajriansyah dkk adalah salah satu penelitian yang dilakukan dengan pengiriman data antar *sensor node* untuk mendapatkan referensi paket data dan *delay* yang akan menunjukkan performa kedua modul *transceiver* pada pengujian *indoor* (Fajriansyah, dkk, 2016). Pengujian kinerja XBee pada topologi *bus*, *star*, *mesh* dan *hybrid* yang menghasilkan nilai parameter RSSI, *throughput*, *delay* dan *packet loss* (Rofii, dkk, 2018) dapat menjadi acuan pemilihan topologi yang tepat untuk implementasi sistem dengan jaringan sensor nirkabel.

Sensor node yang membentuk jaringan sensor nirkabel mempunyai batas jangkauan pengiriman data, sehingga *sensor node* yang berada jauh dari *sink node* tidak dapat mengirimkan data langsung menuju *sink node*. Namun dengan sistem komunikasi *multihop*, memungkinkan dilakukannya komunikasi antar *sensor node* terdekat terlebih dahulu hingga data yang dikirimkan oleh *sensor node* awal sampai pada *sink node*. Selain itu, *delay* pengiriman data pada komunikasi *multihop* lebih kecil dari pada komunikasi *singlehop* (Kurniawan, dkk, 2016). Dengan sistem komunikasi *multihop*, setiap *sensor node* dapat saling berkomunikasi dengan lebih dari satu *sensor node* lainnya. Konstruksi jaringan sangat mempengaruhi konsumsi daya setiap *node* pada jaringan sistem nirkabel (D. Rajendra Prasad, 2016). Jika dibandingkan dengan sistem komunikasi *singlehop*, komunikasi *multihop* membutuhkan daya yang cukup besar dalam beroperasi (M Pešović, dkk, 2010).

Berdasarkan spesifikasinya *platform* ESP8266 dapat digunakan sebagai *node* pada jaringan sensor nirkabel dengan efisiensi ekonomi dan daya yang lebih baik dibandingkan dengan *Ethernet Shield* untuk Arduino, *Zigbee*, *Wifi Shield* Sparkfun, *Wi-Fi Shield* untuk Arduino, dan *Huzzah Wi-Fi Shield* dari Adafruit (Mehta, 2015). ESP8266 merupakan *platform* IoT yang dilengkapi dengan mikrokontroler dan *chip* Wi-Fi. Sehingga untuk membuat *node* pada jaringan sensor nirkabel ESP8266 tidak memerlukan tambahan *chip* Wi-Fi. Pada saat ini terdapat versi terbaru dari ESP8266 yaitu NodeMCU V3 yang memiliki fungsi dan jumlah pin yang lebih banyak dengan spesifikasi konsumsi daya yang sama. Berdasarkan pengujian

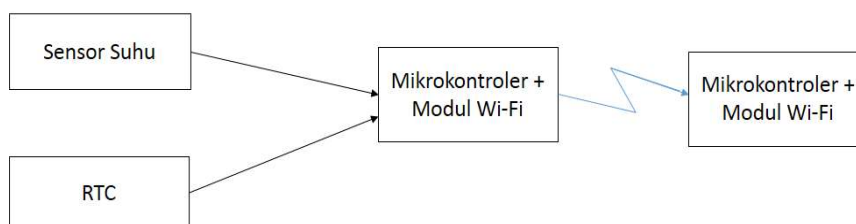
Quality of Service sistem komunikasi yang dilakukan pada penelitian *Smart Home System* dengan sistem *client – server* menggunakan NodeMCU V3, didapatkan hasil bahwa sistem dapat bekerja dengan baik (Purnawan & Rosita, 2019).

Pengembangan *Room Exhaust Ventilation System* (REVS) yang dilakukan oleh Shun, dkk dengan tiga ESP8266 NodeMCU, menunjukkan bahwa dua *sensornode* yang dibangun mampu berkomunikasi dan berinteraksi dengan satu *actuator node* dalam satu jaringan sensor nirkabel. Sistem ini mampu melakukan *monitoring* dan kendali jarak jauh melalui aplikasi web dengan biaya rendah (Shun, dkk, 2020). Demikian pula pada sistem yang dibangun oleh Ouldzira, dkk, dengan empat modul NodeMCU yang difungsikan sebagai *static access point*, *server* dan *client* telah memungkinkan NodeMCU ini saling berkomunikasi melalui protokol HTTP dan melakukan deteksi serta pemantauan objek secara jarak jauh (Ouldzira, dkk, 2019). Sedangkan uji kualitas sinyal pada NodeMCU yang membentuk jaringan yang terdiri dari *coordinator*, *repeater* dan *end device* pada pemantauan suhu, menunjukkan kondisi sinyal yang stabil di luar ruangan dengan jarak di bawah 50 m dan di dalam ruangan dengan jarak di bawah 20 m (Silveira & Bonho, 2016).

Pada penelitian ini dilakukan pengujian pengiriman data menggunakan sistem *multihop* jaringan sensor nirkabel pada media transmisi Wi-Fi untuk topologi *bus* dan *tree*. Platform yang digunakan adalah NodeMCU V3 yang telah memiliki modul Wi-Fi ESP8266 di dalamnya. Dalam membangun sebuah *multihop* jaringan sistem nirkabel, perlu dipastikan bahwa setiap *node* dapat mengetahui data dari *node* mana yang akan diterima dan kepada *node* mana data akan dikirimkan. Selain itu, *node* harus mengetahui apa yang harus dilakukan jika terdapat data yang dikirimkan pada waktu yang bersamaan.

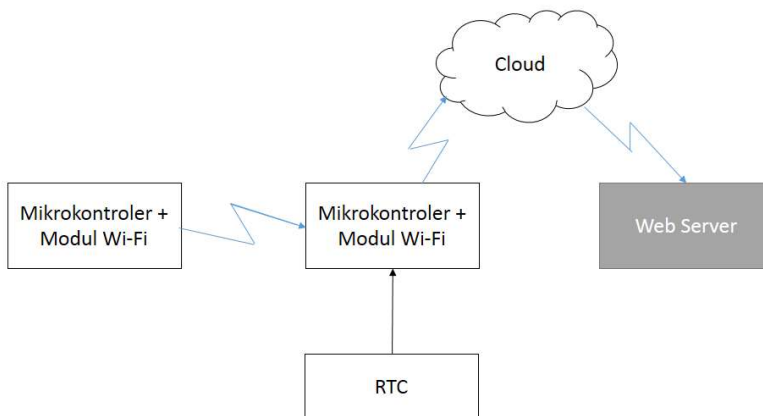
2. PERANCANGAN DAN REALISASI

Penelitian ini akan menguji sistem *multihop* jaringan sensor nirkabel menggunakan 2 topologi jaringan, dengan sampel data suhu dan kelembapan. Ada 4 sensor yang ditempatkan pada 4 titik berbeda, untuk mengetahui kondisi lingkungan pada area yang diamati. Diagram blok *sensor node* pada sistem *multihop* jaringan sensor nirkabel dapat dilihat pada Gambar 1.



Gambar 1. Diagram Blok *Sensor Node* pada Sistem *Multihop* Jaringan Sensor Nirkabel

Pada *sensor node*, sensor suhu dan kelembapan akan membaca data suhu dan kelembapan udara pada lingkungan, keluaran sensor suhu berupa sinyal digital akan dibaca dan diolah oleh mikrokontroler. Pada saat data suhu dan kelembapan dibaca oleh mikrokontroler, RTC (*Real Time Clock*) akan memberikan keterangan waktu pada setiap data. Kemudian data suhu, kelembapan, dan keterangan waktu yang telah diterima oleh mikrokontroler, akan dikirimkan melalui Wi-Fi menuju mikrokontroler pada *sink node* atau *sensor node* lain. Diagram blok *sink node* pada sistem *multihop* jaringan sensor nirkabel dapat dilihat pada Gambar 2.



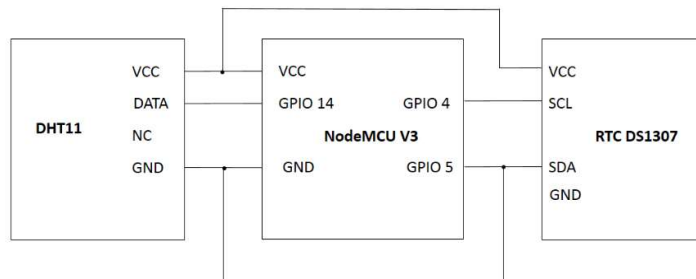
Gambar 2. Diagram Blok *Sink Node* pada Sistem *Multhop* Jaringan Sensor Nirkabel

Sink node akan menerima data dari semua *sensor node* yang dikirimkan melalui Wi-Fi. Sama seperti *sensor node*, *sink node* juga mempunyai kemampuan untuk memberikan keterangan waktu dari RTC. Sehingga, *sensor node* dapat memberikan keterangan waktu penerimaan data dari setiap *sensor node*. Kemudian seluruh data pada *sink node* akan dikirimkan ke *cloud* melalui Wi-Fi yang mempunyai koneksi dengan internet. Data yang terdapat pada *cloud*, dapat diakses oleh *webservice*. Melalui *website* IoT data suhu dan kelembapan dari *webservice* dapat ditampilkan, disimpan, dan diunduh.

Pada penelitian ini *node* yang digunakan berjumlah lima. Empat *node* sebagai *sensor node* dan satu *node* sebagai *sink node*, kelima *node* dapat saling berkomunikasi langsung tanpa *router*. Setiap *node* terdiri dari NodeMCU V3, sensor DHT11 dan RTC DS1307. Sensor DHT11 berfungsi untuk membaca kondisi suhu dan kelembapan udara sebagai sampel data. RTC DS1307 berfungsi untuk memberikan keterangan waktu pada saat data suhu dan kelembapan dibaca oleh mikrokontroler.

2.1 Realisasi Perangkat Keras

Pada realisasinya, pin data pada DHT11 dihubungkan pada pin GPIO 14 NodeMCU V3. Pin SDA dan SCL pada IC RTC DS1307 dihubungkan pada pin GPIO 5 dan GPIO 4 NodeMCU V3. NodeMCU V3 diaktifkan dengan 2 baterai 1,5 volt yang dihubungkan seri menjadi 3 volt. Gambar hubungan pin DHT11 dan RTC DS1307 pada NodeMCU V3 dapat dilihat pada Gambar 4.



Gambar 4. Rangkaian *Node*

Gambar realisasi perangkat keras dari *node* yang dibuat dapat dilihat pada Gambar 5.



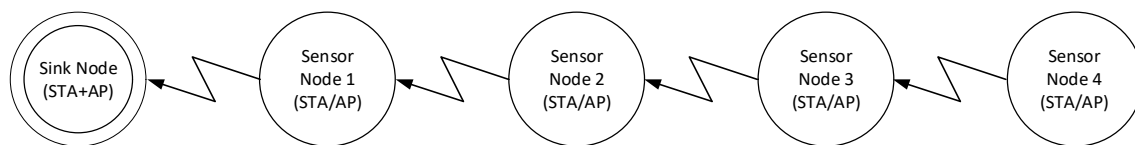
Gambar 5. Hasil Realisasi Perangkat Keras

2.2 Perancangan Perangkat Lunak

Terdapat dua jenis topologi yang akan diuji pada penelitian ini, yaitu topologi *bus* dan topologi *tree*. Sehingga terdapat dua konsep program yang digunakan.

2.2.1 Algoritma Program Topologi *Bus*

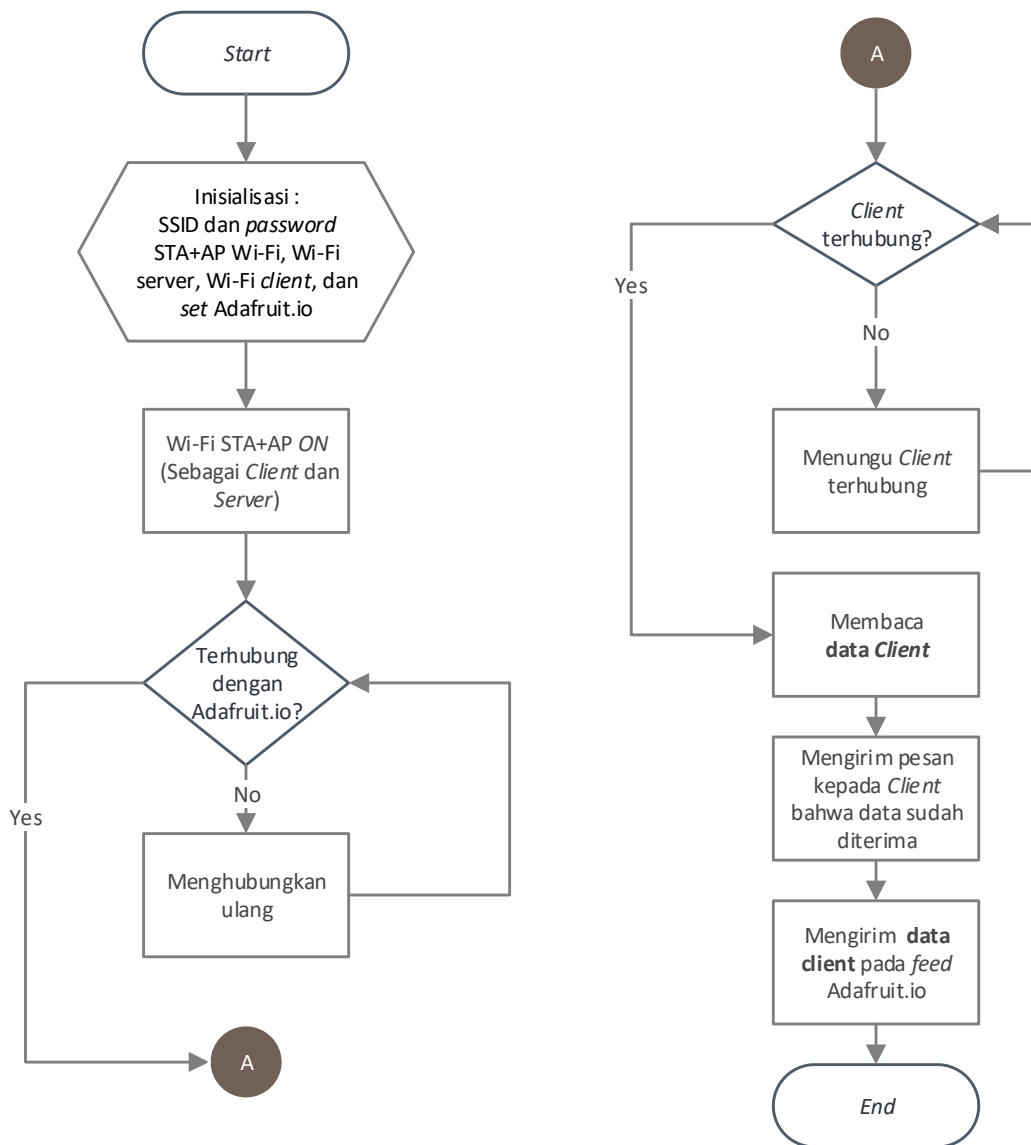
Pada topologi *bus*, terdapat 2 program yang berbeda, yang pertama program untuk *sink node* yang menggunakan mode Wi-Fi *station* (STA) dan *access point* (AP), yang kedua untuk 4 *sensor node* yang menggunakan sistem pergantian mode Wi-Fi antara *station* dan *access point*. Gambar urutan *node* topologi *bus* dapat dilihat pada Gambar 6.



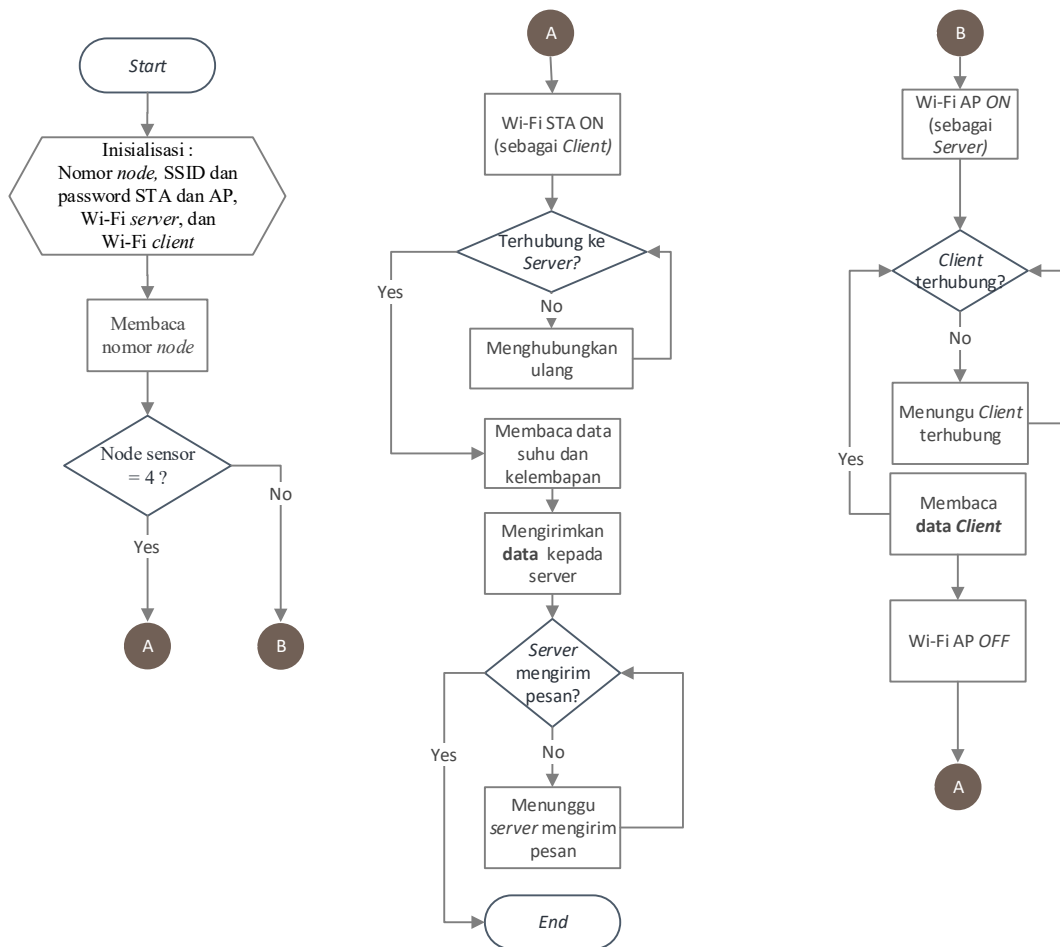
Gambar 6. Urutan *Node* Topologi *Bus*

Program *sink node* menggunakan mode Wi-Fi *station* dan *access point*, sehingga *sink node* menjadi *server* dan *client*. *Sink node* menjadi *server* untuk *sensor node*, sehingga *sink node* dapat mengumpulkan data dari *sensor node*. Pada saat data *client* sudah diterima oleh *server*, *server* akan mengirimkan pesan kepada *client* bahwa data yang dikirimkan sudah diterima. Sebagai *client*, *sink node* akan terhubung kepada Wi-Fi yang mempunyai koneksi internet untuk mengirimkan data yang sudah dikumpulkan kepada *website* IoT Adafruit.io. Selain itu data yang telah dikumpulkan akan ditampilkan pada serial monitor untuk perekaman data menggunakan perangkat lunak PLX-DAQ. *Flowchart* program *sink node* pada topologi *bus* dapat dilihat pada Gambar 7. *Sensor node* 1, 2, 3 dan 4 menerapkan mode Wi-Fi *station* dan *access point* secara bergantian. Sebelumnya, *sensor node* diberikan keterangan nomor *node*, kemudian *node* akan melakukan pembacaan nomor *node*. Apabila nomor *node* merupakan *node* 1, 2, atau 3, program akan menggunakan pergantian mode Wi-Fi. Pada program pergantian mode Wi-Fi, *sensor node* akan menerapkan mode Wi-Fi *access point* atau sebagai *server*, untuk dapat menerima data dari *sensor node* lainnya. Setelah data diterima Wi-Fi *access point* akan dinon-aktifkan, dan mode Wi-Fi akan beralih menjadi *station* atau sebagai *client*. *Sensor node* akan mengirimkan data yang telah diterima dari *sensor node* sebelumnya dan data yang dibaca oleh *sensor node* tersebut kepada *server*. *Server* akan mengirimkan pesan kepada *client* apabila data yang dikirimkan *client* telah sampai di *server*. *Client* hanya akan meneruskan membaca data dari sensor dan mengirimkan datanya kepada *server* apabila *client* telah menerima pesan tersebut. *Flowchart* program *sensor node* 1, 2, 3, dan 4 pada topologi *bus* dapat dilihat pada Gambar 8.

Sensor node tidak dapat menggunakan mode Wi-Fi STA+AP seperti pada *sink node*, sebab alamat IP mode Wi-Fi *acces point* pada ESP8266 merupakan alamat IP yang tetap tidak dapat diubah. Sehingga mode Wi-Fi STA+AP tidak dapat mengirimkan data kepada *node* dengan mode Wi-Fi yang sama, karena mempunyai alamat IP yang sama.



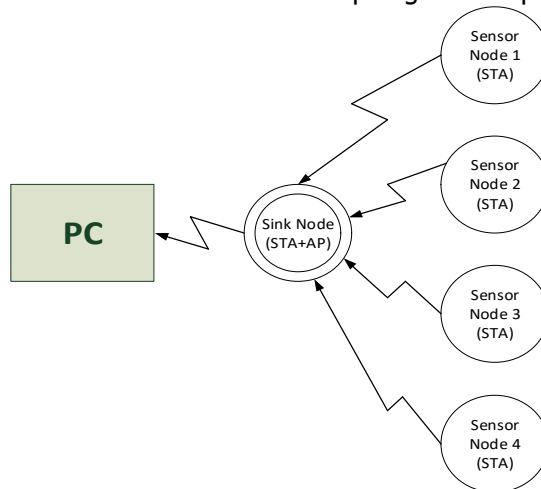
Gambar 7. Flowchart Program Sink Node pada Topologi Bus



Gambar 8. Flowchart Program Sensor node 1, 2, 3, dan 4 pada Topologi Bus

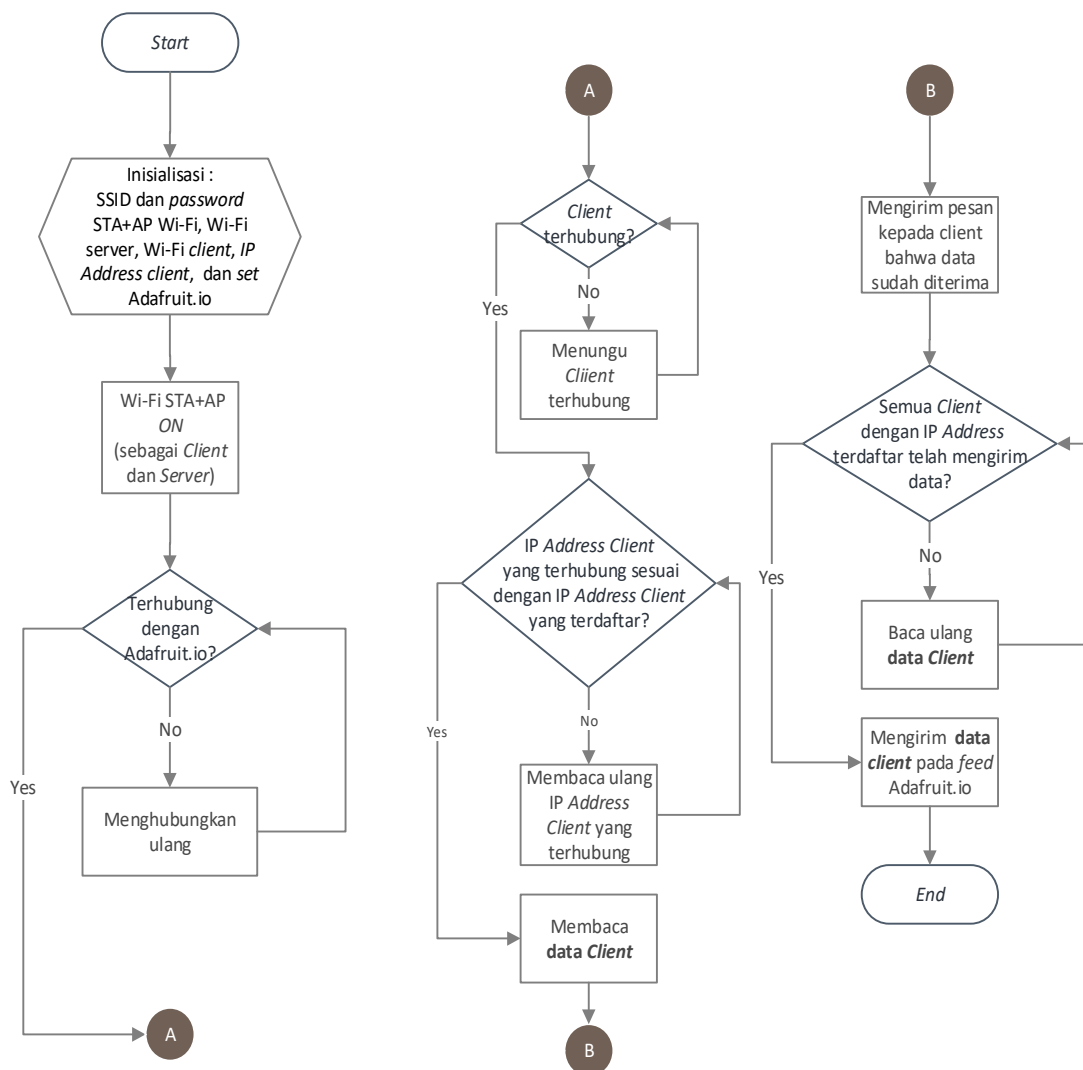
2.2.2 Algoritma Program Topologi Tree

Pada topologi tree, terdapat 2 program yang berbeda, yang pertama program untuk sink node yang menggunakan mode Wi-Fi station dan access point, yang kedua untuk 4 sensor node menggunakan mode Wi-Fi station. Gambar node topologi tree dapat dilihat pada Gambar 9.



Gambar 9. Node Topologi Tree

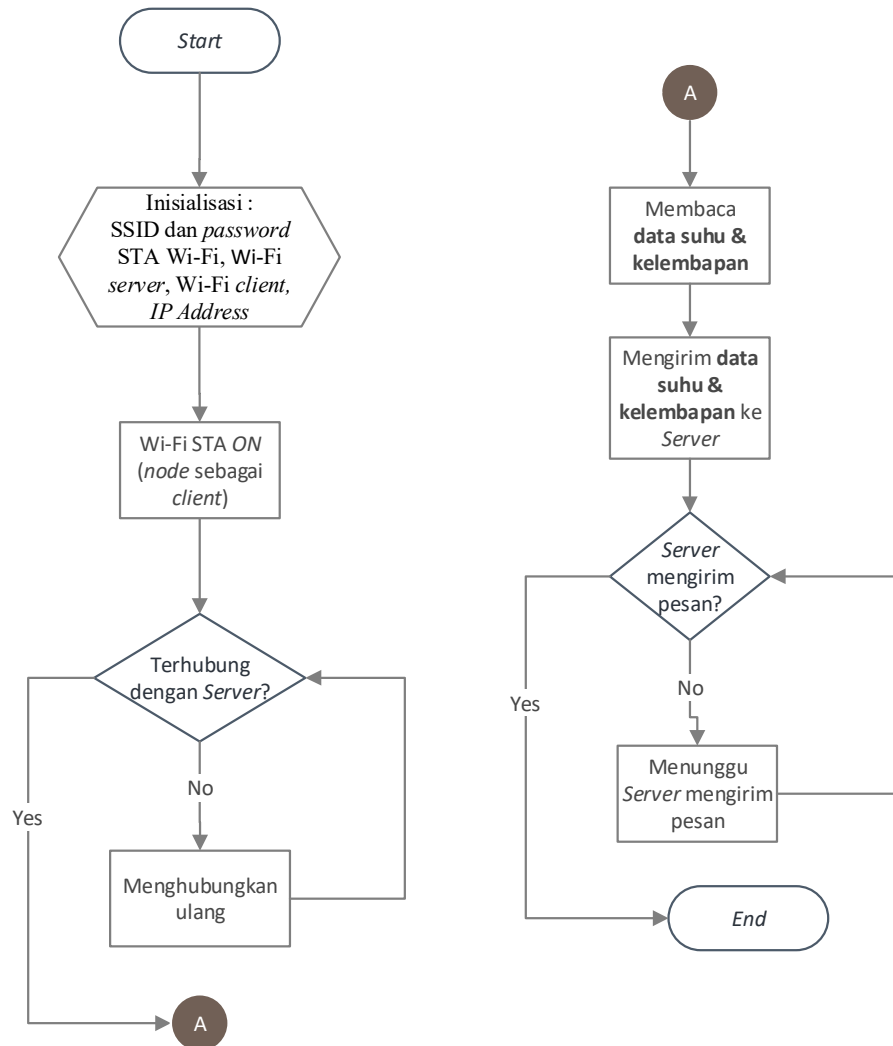
Program *sink node* menggunakan mode Wi-Fi *station* dan *access point*, sehingga *sink node* menjadi *server* dan *client*. *Sink node* menjadi *server* untuk 4 *sensor node*. *Sink node* mempunyai daftar alamat IP *client* yang akan terhubung, sehingga hanya *client* dengan alamat IP terdaftar yang dapat mengirimkan datanya. Sesuai dengan jumlah *client* yang terdaftar, *sink node* akan membaca 4 paket data dari alamat IP yang berbeda, dalam satu *loop* 1 *client* akan mengirimkan 1 data agar penerimaan data tidak bentrok. Pada saat data *client* sudah diterima oleh *server*, *server* akan mengirimkan pesan kepada *client* bahwa data yang dikirimkan sudah diterima. Apabila pada saat pembacaan data, jumlah data yang diterima kurang dari 4, *sink node* akan melakukan pembacaan ulang pada *Alamat IP* yang belum mengirimkan datanya. Sebagai *client*, *sink node* akan terhubung kepada Wi-Fi yang mempunyai koneksi internet untuk mengirimkan data yang sudah dikumpulkan kepada *website* Adafruit.io. Selain itu data yang telah dikumpulkan akan ditampilkan pada serial monitor untuk dapat direkam menggunakan *perangkat lunak* PLX-DAQ. *Flowchart* program *sink node* pada topologi *tree* dapat dilihat pada Gambar 10.



Gambar 10. Flowchart Program Sink Node pada Topologi Tree

Semua *sensor node* mempunyai program yang sama, *sensor node* menerapkan mode Wi-Fi *station* atau sebagai *client*. *Sensor node* akan mengirimkan data yang telah dibaca oleh sensor

kepada *server* dan menunggu pesan balasan dari *server*. *Server* akan mengirimkan pesan kepada *client* apabila data yang dikirimkan *client* telah sampai di *server*. *Client* hanya akan membaca data dari sensor dan mengirimkan datanya kepada *server* apabila *client* telah menerima pesan tersebut. *Flowchart* program *sensor node* pada topologi *tree* dapat dilihat pada Gambar 11.



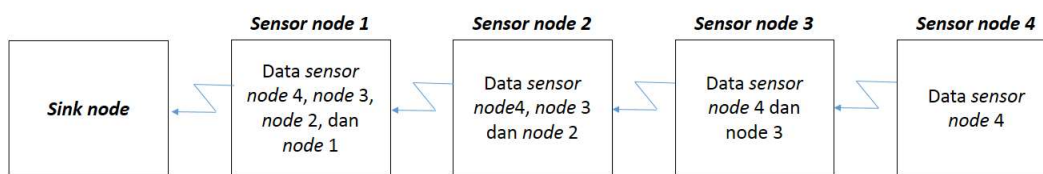
Gambar 11. Flowchart Program Sensor Node pada Topologi Tree

2.2.3 Format paket data

Pada pengiriman paket data dari satu *node* kepada *node* lainnya, diperlukan format paket data agar data yang dikirimkan mudah dimengerti dan jelas dari *node* mana data tersebut dikirimkan. Sistem *multihop* jaringan sensor nirkabel pada topologi *bus* dan topologi *tree* memiliki format data yang berbeda.

a. Format paket data pada topologi bus

Sensor node pada sistem *multihop* jaringan sensor nirkabel dengan topologi *bus*, mengirimkan data dengan cara meneruskan data berurutan dari *sensor node* paling jauh dari *sink node* hingga *sensor node* terdekat dengan *sink node*. Urutan pengiriman data topologi *bus* dapat dilihat pada Gambar 12.



Gambar 12. Urutan Pengiriman Data Topologi Bus

Pada topologi *bus*, semakin dekat *sensor node* dengan *sink node* semakin banyak pula data yang diteruskan. Agar data mudah dibedakan, setiap data harus memiliki identitas dari *node* mana data tersebut berasal. Pada penelitian ini, format paket data topologi *bus* adalah **(nomor *sensor node* /waktu pengiriman/data suhu/data kelembapan)**. Nomor *sensor node* digunakan sebagai identitas *sensor node* pengirim data, tanda garis miring sebagai pemisah jenis data, kemudian semua data ditempatkan dalam tanda kurung agar data terkelompokan. Jumlah karakter yang dikirimkan adalah 18 karakter.

b. Format paket data pada topologi *tree*

Pada sistem *multihop* jaringan sensor nirkabel dengan topologi *tree*, seluruh *sensor node* mengirimkan data secara bersamaan kepada *sink node*. *Sensor node* pada topologi *tree*, memiliki alamat IP yang berbeda-beda. Sehingga *sink node* mampu membedakan data yang diterima data dari alamat IP *sensor node*, tanpa perlu penambahan identitas data. Pada penelitian ini, format paket data topologi *bus* adalah waktu pengiriman/data suhu/data kelembapan. Untuk mempermudah pembacaan data, setelah paket data diterima, *sink node* akan menambahkan alamat IP *sensor node* yang mengirimkan data. Sehingga format data yang ditampilkan pada PLX-DAQ adalah (alamat IP/waktu pengiriman/data suhu/data kelembapan). Alamat IP sebagai identitas *sensor node* pengirim data garis miring sebagai pemisah jenis data, kemudian semua data ditempatkan dalam tanda kurung agar data terkelompokkan. Jumlah karakter yang dikirimkan adalah 14 karakter.

3. PENGUJIAN DAN ANALISIS

Untuk mengetahui kemampuan sistem *multihop* jaringan sensor nirkabel pada media transmisi Wi-Fi dengan menggunakan NodeMCU V3, dilakukan pengujian pada beberapa parameter. Diantaranya adalah kesesuaian data antara yang dikirimkan oleh *sensor node* dengan data yang diterima oleh *sink node*, jarak maksimum pengiriman data, serta *delay* pengiriman data.

3.1 Pengujian Kesesuaian Data

Dalam sistem *multihop* jaringan sensor nirkabel diperlukan pengujian kesesuaian data pada *sensor node* dan *sink node*. Pengujian dilakukan dengan menghubungkan semua *node* pada personal komputer melalui kabel USB. Data yang dikirimkan oleh *sensor node* dan data yang diterima oleh *sink node* direkam menggunakan *perangkat lunak* PLX-DAQ, dan data yang di tampilkan pada *website* Adafruit.io akan diunduh dalam format .csv. Kemudian, seluruh data tersebut dibandingkan dan dilihat kesesuaiannya. Pengujian dilakukan pada topologi *bus* dan topologi *tree*. Jumlah data yang akan direkam dan disesuaikan pada pengujian ini adalah 30 data pada masing-masing *sensor node*. Hasil pengujian kesesuaian data pada topologi *bus* dapat dilihat pada Tabel 1 dan hasil pengujian kesesuaian data pada topologi *tree* dapat dilihat pada Tabel 2.

Tabel 1. Hasil Pengujian Kesesuaian Data pada Topologi *Bus*

Data	Data dikirim oleh <i>Sensor node</i>		Data diterima oleh <i>Sink node</i>		Data yang ditampilkan Adafruit.io	
	Jumlah Data	Persentase	Jumlah Data	Persentase	Jumlah Data	Persentase
Suhu <i>Sensor node 1</i>	30	100%	30	100%	27	90%
Kelembapan <i>Sensor node 1</i>	30	100%	30	100%	27	90%
Suhu <i>Sensor node 2</i>	30	100%	30	100%	25	83%
Kelembapan <i>Sensor node 2</i>	30	100%	30	100%	24	80%
Suhu <i>Sensor node 3</i>	30	100%	30	100%	27	90%
Kelembapan <i>Sensor node 3</i>	30	100%	30	100%	27	90%
Suhu <i>Sensor node 4</i>	30	100%	30	100%	27	90%
Kelembapan <i>Sensor node 4</i>	30	100%	30	100%	27	90%

Tabel 2. Hasil Pengujian Kesesuaian Data pada Topologi *Tree*

Data	Data dikirim oleh <i>Sensor node</i>		Data diterima oleh <i>Sink node</i>		Data yang ditampilkan Adafruit.io	
	Jumlah Data	Persentase	Jumlah Data	Persentase	Jumlah Data	Persentase
Suhu <i>Sensor node 1</i>	30	100%	30	100%	26	87%
Kelembapan <i>Sensor node 1</i>	30	100%	30	100%	27	90%
Suhu <i>Sensor node 2</i>	30	100%	30	100%	24	80%
Kelembapan <i>Sensor node 2</i>	30	100%	30	100%	27	90%
Suhu <i>Sensor node 3</i>	30	100%	30	100%	28	93%
Kelembapan <i>Sensor node 3</i>	30	100%	30	100%	29	97%
Suhu <i>Sensor node 4</i>	30	100%	30	100%	27	90%
Kelembapan <i>Sensor node 4</i>	30	100%	30	100%	27	90%

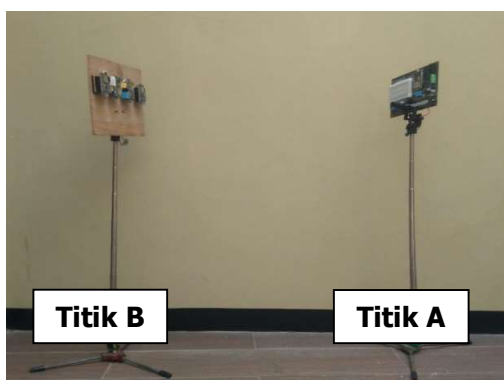
Hasil pengujian menunjukkan bahwa seluruh data pada topologi *bus* dan *tree* 100% dikirim oleh *sensor node* menuju ke *node* lainnya. Kemudian data yang diterima oleh *sink node* 100% sesuai dengan yang dikirim oleh *sensor node*. Namun tidak semua data yang telah diterima oleh *sink node* ditampilkan pada *dashboard* Adafruit.io. Terdapat 1 hingga 6 data yang tidak ditampilkan, hal tersebut disebabkan oleh kecepatan internet yang tidak stabil sehingga menyebabkan beberapa data dari *sink node* tidak terkirim pada *website* Adafruit.io

3.2 Pengujian Jarak dan *Delay* Pengiriman

Media transmisi Wi-Fi memiliki batas jangkauan koneksi, sehingga *node* memiliki jarak maksimal pengiriman data. Selain itu, adanya penghalang di antara *node* dapat mempengaruhi pengiriman data. Untuk mengetahui kemampuan modul Wi-Fi ESP8266 pada NodeMCU V3, dilakukan pengujian jarak maksimal pengiriman data pada daerah terbuka tanpa penghalang dan daerah dengan penghalang.

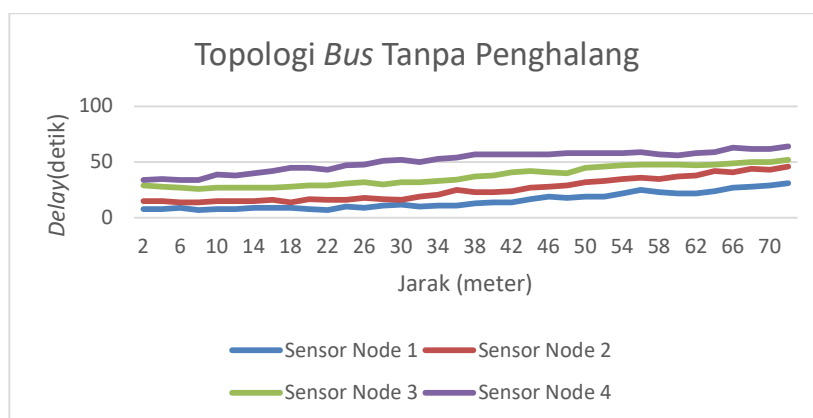
3.2.1 Pengujian tanpa Penghalang

Pengujian tanpa penghalang dilakukan dengan cara menempatkan *node* pengirim dan penerima berhadapan dengan jarak yang bertambah 2 meter setelah pengiriman 30 data. Pada topologi *bus*, *sink node*, *sensor node* 2 dan *sensor node* 4 berada di titik A, kemudian *sensor node* 1 dan *sensor node* 3 pada titik B. Pada topologi *tree*, *sink node* ditempatkan pada titik A dan *sensor node* 1, *sensor node* 2, *sensor node* 3, *sensor node* 4 pada titik B. Titik B akan berpindah 2 meter setelah pengiriman 30 data. Pengujian tanpa penghalang dapat dilihat pada Gambar 13.



Gambar 13. Pengujian tanpa Penghalang

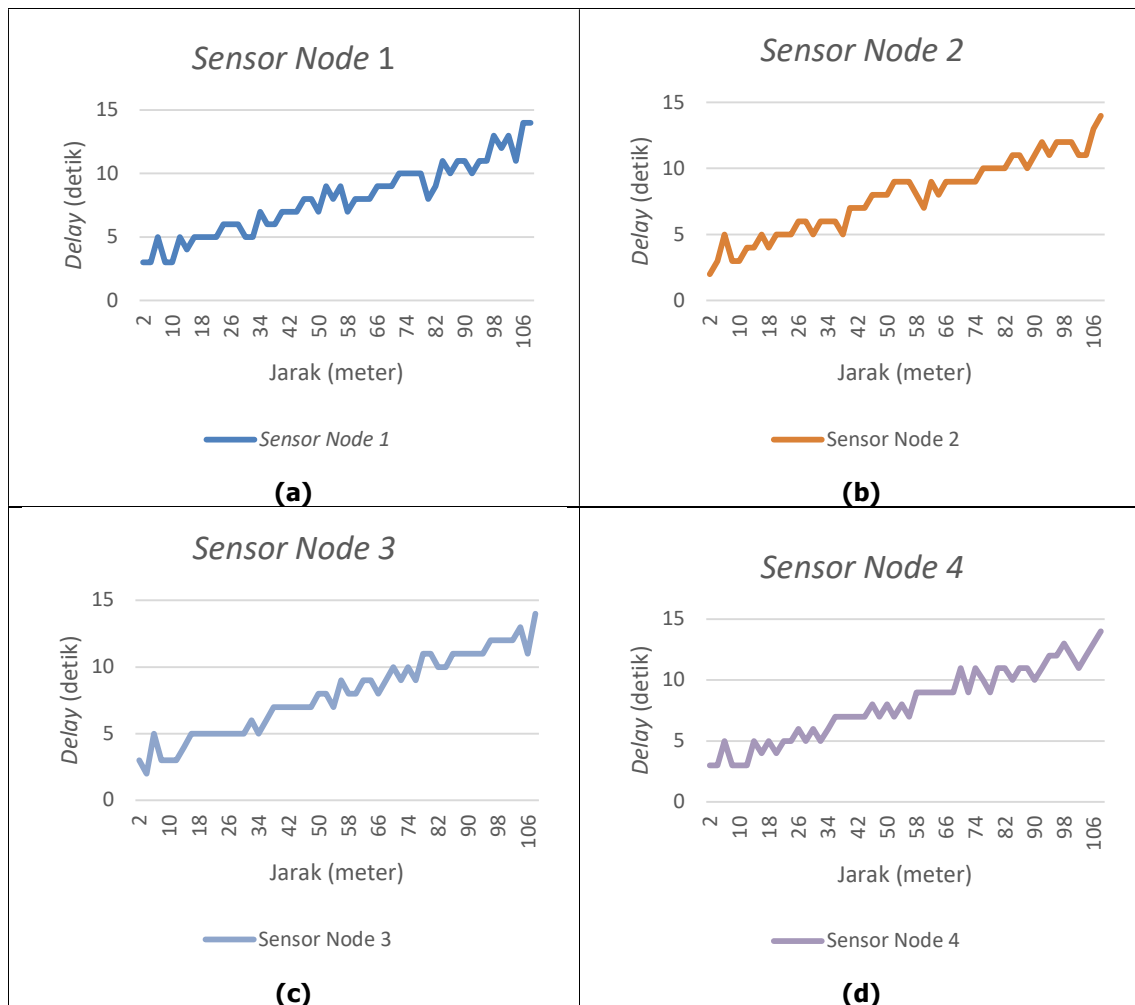
Hasil pengujian jarak dan *delay* waktu pengiriman pada topologi *bus* tanpa penghalang dapat dilihat pada Gambar 14. Sedangkan hasil pengujian jarak dan *delay* waktu pengiriman pada topologi *tree* tanpa penghalang dapat dilihat pada Gambar 15.



Gambar 14. Hasil Pengujian Jarak dan Delay Waktu Pengiriman pada Topologi Bus tanpa Penghalang

Gambar 14 menunjukkan grafik *delay* pengiriman data terhadap jarak pada topologi *bus* tanpa penghalang. Pada topologi *bus*, *sensor node* 1 memiliki *delay* pengiriman tercepat karena posisi *node* berada paling dekat dengan *sink node*, pada posisi 0 meter *sensor node* 1 memerlukan 8 detik untuk menyampaikan datanya pada *sink node*. *Sensor node* 4 memiliki *delay* pengiriman terlama karena posisi *node* berada paling jauh dari *sink node*, sehingga *sensor node* 4 perlu mengirimkan datanya kepada *sensor node* yang lebih dekat dengan *sink node* terlebih dahulu. Pada posisi 0 meter, *sensor node* 4 memerlukan waktu 34 detik untuk menyampaikan datanya pada *sink node*. Jarak maksimum pengiriman data pada topologi *bus*

adalah 72 meter dengan *delay* pengiriman seluruh data 64 detik. Pada saat jarak lebih dari 72 meter *sink node* tidak menerima data.



Gambar 15. Hasil Pengujian Jarak dan *Delay* Waktu Pengiriman pada Topologi *Tree* tanpa Penghalang: (a) *Sensor Node 1* (b) *Sensor Node 2* (c) *Sensor Node 3* (d) *Sensor Node 4*

Gambar 15 menunjukkan grafik *delay* pengiriman data terhadap jarak tanpa penghalang dengan topologi *tree* pada *sensor node 1*, *2*, *3*, dan *4*. Pada topologi *tree* seluruh *sensor node* mengirimkan datanya secara bersamaan kepada *sink node*, sehingga *delay* pengiriman 4 *sensor node* tidak jauh berbeda. Pada jarak 0 meter membutuhkan 2 sampai 3 detik untuk mengirimkan datanya pada *sink node*. Sehingga pada waktu 3 detik *sensor node* dapat menerima 4 data dari 4 *sensor node*. Jarak maksimum pengiriman data pada topologi *tree* adalah 108 meter dengan *delay* pengiriman data 14 detik. Pada saat jarak lebih dari 108 meter *sink node* sudah tidak dapat menerima data. Hasil pengujian menunjukkan bahwa semakin jauh jarak antar *node*, semakin bertambah *delay* pengiriman data. Hal tersebut disebabkan semakin jauh Wi-Fi terhubung semakin lemah sinyal yang dipancarkan oleh Wi-Fi *access point* yang menyebabkan koneksi tidak stabil sehingga membutuhkan *delay* pengiriman lebih lama. Untuk mengetahui perbedaan sistem *multihop* menggunakan topologi *bus* dan *tree* maka dilakukan perbandingan hasil pengujian topologi *bus* dan *tree*.

Perbandingan hasil pengujian topologi *bus* dan *tree* dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan Hasil Pengujian Topologi *Bus* dan *Tree* Tanpa Penghalang

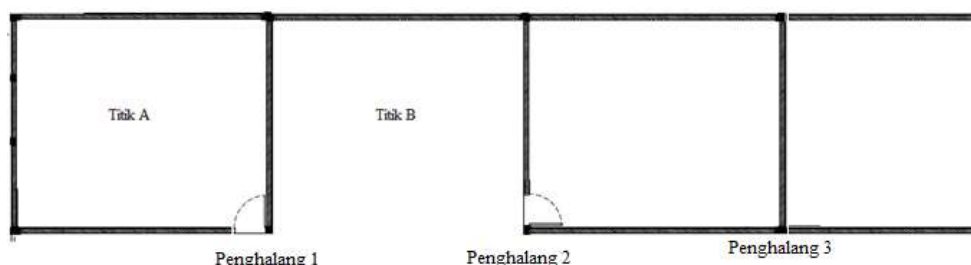
	Jarak Maksimal (meter)	Delay pada <20cm (detik)	Delay pada Jarak Maksimal (detik)
Topologi <i>Bus</i>	72	8	64
Topologi <i>Tree</i>	108	2	14

Perbandingan hasil pengujian menunjukkan, bahwa *delay* pengiriman data antar *node* paling kecil ada pada topologi *tree*, hal tersebut disebabkan karena perbedaan algoritma program yang diterapkan. Pada topologi *bus*, *sensor node* melakukan 4 komunikasi, yaitu pada saat menerima data dari *sensor node* lain, memberikan pesan kepada *sensor node* bahwa data telah diterima, meneruskan data yang diterima dan dibaca kemudian menunggu pesan balasan. Untuk melakukan keempat komunikasi tersebut topologi *bus* menggunakan pergantian mode Wi-Fi dari *access point* ke mode Wi-Fi *station*. Sehingga selain waktu untuk mengirim dan menerima, topologi *bus* membutuhkan waktu untuk melakukan pergantian mode Wi-Fi. Sedangkan pada topologi *tree*, *sensor node* hanya mengirim data dan menunggu balasan pesan dari *sink node* tanpa harus menghubungkan ulang koneksi Wi-Fi sehingga waktu yang diperlukan untuk pengiriman data lebih kecil dibandingkan dengan topologi *bus*.

Jarak maksimum pengiriman data tanpa penghalang menggunakan topologi *bus* berbeda dengan topologi *tree*, topologi *tree* memiliki jarak maksimal lebih jauh. Hal tersebut disebabkan jumlah karakter yang dikirimkan pada topologi *bus* dan topologi *tree* berbeda. Topologi *bus* akan membaca data dari *node* sebelumnya dan meneruskan data tersebut pada *node* berikutnya, sehingga tiap melewati *node* karakter jumlah data akan bertambah. Jumlah karakter pada data yang dikirimkan oleh seluruh *sensor node* pada topologi *bus* adalah 74 karakter. Sedangkan pada topologi *tree*, *sensor node* hanya mengirimkan data yang dibaca oleh sensor tanpa meneruskan data dari *node* lain, jumlah karakter pada data yang dikirimkan topologi *tree* adalah 14 karakter oleh masing-masing *sensor node*.

3.2.2 Pengujian dengan Penghalang

Pengujian dengan penghalang pada topologi *bus* dan topologi *tree* dilakukan dengan menempatkan *node* pada titik A dan *node* pada titik B di antara penghalang tembok dengan ketebalan 15 cm dan jarak antar *node* sejauh 7 m. Jumlah tembok penghalang akan bertambah setelah pengiriman 30 data. Pengujian dengan penghalang dapat dilihat pada Gambar 16.



Gambar 16. Pengujian dengan Penghalang

Hasil pengujian sistem *multihop* dengan penghalang pada topologi *bus* dan *tree* dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian Sistem *Multihop* dengan Penghalang pada Topologi *Bus*

		<i>Delay (detik)</i>		
Topologi	<i>Nomor Node</i>	1 Penghalang	2 Penghalang	3 Penghalang
<i>Bus</i>	<i>Sensor node 1</i>	16	29	-
	<i>Sensor node 2</i>	27	55	-
	<i>Sensor node 3</i>	39	81	-
	<i>Sensor node 4</i>	56	107	-
<i>Tree</i>	<i>Sensor node 1</i>	7	14	-
	<i>Sensor node 2</i>	5	12	-
	<i>Sensor node 3</i>	6	15	-
	<i>Sensor node 4</i>	6	14	-

Hasil pengujian sistem *multihop* dengan penghalang pada topologi *bus* menunjukkan *sensor node 4* (*sensor node* terjauh) membutuhkan waktu 56 detik untuk dapat mengirimkan datanya menuju *sink node* dengan 1 penghalang pada setiap *node* yang dilewati. Sedangkan *sensor node 1* (*sensor node* terdekat) membutuhkan 16 detik untuk mengirimkan datanya kepada *sink node*. Pada saat pengujian dengan 2 penghalang pada topologi *bus*, *delay* pengiriman data menjadi lebih lama, yaitu 107 detik untuk *node* terjauh dan 56 detik untuk *node* terdekat. Pada saat pengujian dilakukan menggunakan 3 penghalang tidak ada data yang terkirim dikarenakan koneksi Wi-Fi terputus karena sinyal Wi-Fi yang sangat lemah.

Pengujian sistem menggunakan 1 penghalang pada topologi *tree*, mengakibatkan setiap *sensor node* membutuhkan waktu 5 hingga 7 detik untuk mengirimkan datanya. Sama seperti pada topologi *bus*, pada saat pengujian menggunakan 2 penghalang pada topologi *tree*, *delay* pengiriman data menjadi lebih lama yaitu 12 hingga 15 detik. Pada saat pengujian dilakukan menggunakan 3 penghalang tidak ada data yang terkirim dikarenakan koneksi Wi-Fi terputus. Berdasarkan hasil pengujian tersebut, jumlah penghalang sangat berpengaruh pada waktu pengiriman data, hal tersebut disebabkan oleh banyaknya penghalang mempengaruhi kekuatan sinyal Wi-Fi *access point* terhadap *client* yang terhubung. Apabila sinyal lemah maka koneksi Wi-Fi tidak stabil sehingga untuk mengirimkan data butuh waktu yang lebih lama.

4. KESIMPULAN

Pengujian dan pengolahan data yang dilakukan terhadap sistem *multihop* jaringan sensor nirkabel pada penelitian ini, menunjukkan bahwa *sink node* dapat menerima seluruh data sesuai dengan data yang dikirimkan oleh *sensor node*. Pada topologi *bus*, jarak maksimal pengiriman data adalah 72 meter dengan *delay* pengiriman seluruh data 64 detik. Sedangkan pada topologi *tree*, jarak maksimal pengiriman data adalah 108 meter dengan *delay* data 14 detik pada masing-masing *sensor node*. Jarak maksimal pengiriman data topologi *tree* lebih jauh dibanding dengan topologi *bus* karena jumlah karakter yang dikirimkan berbeda. Berdasarkan hasil pengujian, *delay* pengiriman data pada topologi *bus* lebih lama dari topologi *tree* karena topologi *bus* melakukan pergantian Wi-Fi untuk dapat mengirim dan menerima data. Kecepatan internet yang stabil dibutuhkan untuk pengiriman data, agar data dapat ditampilkan pada *website* IoT.

DAFTAR RUJUKAN

- Babu, R. S., Palaniappan, T., Anushya, K., Kowsalya, M., & Krishnadevi, M. (2018). IoT Based Weather Monitoring System. *International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)*, 5(13), 105-109.
- D. Rajendra Prasad, P. (2016). Energy Efficient Clustering in MULTI-hop Wireless Sensor. *Brazilian Archives of Biology And Technology*, 1-15.
- Fajriansyah, B., Ichwan, M., & Susana, R. (2016). Evaluasi Karakteristik XBee Pro dan nRF24L01+ sebagai Transceiver Nirkabel. *ELKOMIKA*, 4(1), 83-97.
- Islam, M. M. (2019). Weather Monitoring System Using Internet of Things. *Trends in Technical & Scientific Research*, 65-69.
- Kurniawan, A., Munadi, R., & Mayasari, R. (2016). Implementasi dan Analisa Jaringan Wireless Sensor. *Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri (SENIATI)*, (pp. 20-25).
- M Pešović, U., J. Mahorko, J., Karl, B., & F. Čučej, (2010). Single-hop vs. Multi-hop Energy Efficiency Analysis in Wireless Sensor Networks. *Telekominikacioni Forum TELFOR 2010*, (pp. 471-474).
- Mehta, M. (2015). ESP 8266 : A BREAKTHROUGH IN. *International Journal of Electronics and Communication Engineering & Technology*, 7-11.
- Nikhilesh, K. S., Raaghavendra, Y. H., Soothanan, P. M., & Resmi, R. (2020). Low-Cost IoT Based Weather Monitoring System for Smart Community. *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*. Coimbatore, India: IEEE.
- Ouldzira, H., Mouhsen, A., Lagraini, H., Chhiba, M., Tabyaoui, A., & Amrane, S. (2019). Remote Monitoring of an Object using A Wireless Sensor Network Based on NodeMCU ESP8266. *Indonesian Journal of Electrical Engineering and Computer Science*, 16(3), 1154 - 1162.
- Purnawan, P. W., & Rosita, Y. (2019). Rancang Bangun Smart Home System Menggunakan NodeMCU Esp8266 berbasis Komunikasi Telegram Messenger. *Techno.COM*, 18(4), 348 - 360.
- Rofii, F., Hunaini, F., & Sholawati, S. (2018). Kinerja Jaringan Komunikasi Nirkabel Berbasis XBee pada Topologi Bus, Star dan Mesh. *ELKOMIKA*, 6(3), 393-404.
- Sasono, S. H., Kusumastuti, S., Supriyanto, E., Widodo, S., & Azizcha, D. (2017). QoS Analysis of Wireless Sensor Networks for Temperature and Humidity Monitoring and Control of Soybean Seed storage Based IoT using NodeMCU. *Journal of Applied Information and Communication Technologies (JAICT)*, 2(1), 1-11.

- Shirsath, S., & Waghile, N. (2018). IoT Based Smart Environmental Monitoring Using Wireless Sensor Network. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, 7(6), 3023-3031.
- Shun, W. G., Mariam, W., Hafiza, W., & Anuar, A. Z. (2020). Wireless Sensor Network for Temperature and Humidity Monitoring Systems based on NodeMCU ESP8266. Dalam *Advances in Cyber Security* (pp. 262 - 273). Singapore: Springer.
- Silveira, E., & Bonho, S. (2016). Temperature Monitoring Through Wireless Sensor Network Using An 802.15.4/802.11 Gateway. *International Federation of Automatic Control*, (pp. 120 -125). Elsevier.
- Syafiqoh, U., Sunardi, & Yudhana, A. (2018). Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas Air dan Tanah Pertanian. *Jurnal Informatika : Jurnal Pengembangan IT (JPIT)*. 3(2), 285-289.