

Analisis *Traffic* Pada Implementasi *Wireless Sensor Network* Polusi Udara

Ikrimah Muiz¹, Dodi Wisaksono Sudiharto², Aji Gautama Putrada³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹ikrimah@student.telkomuniversity.com,

²dodiwisaksono@telkomuniversity.ac.id, ³ajigps@telkomuniversity.ac.id

Abstrak

Polusi udara merupakan salah satu faktor signifikan yang banyak mempengaruhi kondisi kesehatan manusia. Polusi itu datang dari gas pembuangan kendaraan, asap industri, ataupun asap rokok. Untuk memperbaiki keadaan tersebut, akan dilakukan pemantauan polusi udara dengan memanfaatkan teknologi *Wireless Sensor Network* serta menerapkan jaringan *broadcast*. Pada penerapan *WSN* polusi udara ini menggunakan sensor MQ-5, datanya akan dikirim dan berpusat pada satu *receiver* yang akan dihubungkan ke jaringan internet lalu ditampilkan ke Thingspeak Server. Oleh karena itu pada penelitian ini akan menganalisis *traffic* berdasarkan implementasi *WSN* polusi udara menggunakan teknologi ESP8266 untuk transmisi datanya. Dalam penelitian akan diterapkan jaringan *broadcast* sebanyak 4 node dan topologi *star*, hasil penelitian didapatkan performa jaringan *broadcast* lebih kecil menghasilkan *packet loss* sebesar 0% dibandingkan dengan performa topologi *star* tanpa jaringan *broadcast* menghasilkan *packet loss* sebesar 6%.

Kata Kunci: *Traffic, Wireless Sensor Network, Jaringan broadcast.*

Abstract

Air pollution is one of the significant factors that affect many human health conditions. The pollution comes from gas disposal vehicles, industrial smoke, or cigarette smoke. To improve the situation, air pollution monitoring will be carried out by utilizing Wireless Sensor Network technology and apply broadcast network. In the application of WSN this air pollution uses MQ-5 sensors, the data will be sent and centered on one receiver that will be connected to the internet network and then shown to Thingspeak Server. Therefore, in this research will analyze the traffic based on the implementation of WSN air pollution using ESP8266 technology for data transmission. In the research will be applied broadcast network as much as 4 node and star topology, result of research got performance of broadcast network resulted less packet loss 0% compared with star topology performance without broadcast mechanism resulting in 6% packet loss.

Keywords: *Traffic, Wireless Sensor Network, broadcast network.*

1. Pendahuluan

1.1 Latar Belakang

Polusi udara adalah salah satu faktor signifikan yang banyak mempengaruhi kondisi kesehatan manusia seperti infeksi pada kulit dan mata, iritasi pada hidung. Polusi itu datang dari gas pembuangan kendaraan, asap industri ataupun asap rokok. Untuk memperbaiki keadaan tersebut, akan dilakukan pemantauan polusi udara dengan memanfaatkan teknologi *Wireless Sensor Network*.

Masalah yang terdapat pada tugas akhir ini ialah penerapan *WSN* dengan menggunakan banyak sensor untuk mendeteksi polusi udara. Semua data dari *node* akan berpusat pada satu *receiver* dan dapat menimbulkan masalah apabila *node* yang ada jumlahnya sangat banyak. Masalah yang timbul adalah *packet loss* yang disebabkan setiap data dari *node* akan dikirim ke satu receiver, maka dari itu akan diterapkan jaringan *broadcast* untuk mengatasi masalah tersebut dan dibandingkan dengan topologi *star* untuk melihat kinerja jaringannya.

Berdasarkan paper sebelumnya [1] menjelaskan bahwa meningkatnya penggunaan dan implementasi dalam segi ruang lingkup dan skala pada jaringan sensor nirkabel dapat menyebabkan terjadinya penumpukan (*congestion*), terjadinya *congestion* pada jaringan dapat menyebabkan *packet loss* pada pengiriman data. Dalam *WSN*, *congestion* terjadi ketika beban *traffic* melebihi kapasitas yang tersedia pada setiap titik dalam jaringan dan akibat terjadinya *congestion* kinerja jaringan akan menurun [2].

Jaringan *broadcast* merupakan sebuah metode pengiriman data dimana setiap node saling terhubung sehingga pada saat node mengirimkan datanya maka semua node yang terhubung akan menerima data tersebut begitu pula node yang lain ketika mengirimkan datanya. Oleh karena itu, node bertindak sebagai pengirim dan penerima data secara bersamaan[3].

1.2 Topik dan Batasannya

Berdasarkan latar belakang di atas, maka perumusan masalah dalam pembuatan tugas akhir ini adalah bagaimana cara membandingkan kinerja jaringan *broadcast* dan *star* pada implementasi *Wireless Sensor Network* polusi udara, dan bagaimana cara menampilkan hasil data *Wireless Sensor Network* polusi udara pada Thingspeak Server.

Dalam pengerjaan tugas akhir ini terdapat beberapa batasan masalah yaitu implementasi polusi udara menggunakan *Wireless Sensor Network*, implementasi *Wireless Sensor Network* menerapkan jaringan *broadcast* dan topologi *star*, menampilkan hasil data *Wireless Sensor Network* polusi udara menggunakan Thingspeak Server, serta menampilkan *delay* dan *packet loss*.

1.3 Tujuan

Berdasarkan dari perumusan masalah di atas, maka tujuan pembuatan Tugas Akhir ini implementasi jaringan *broadcast* dan *star* pada *Wireless Sensor Network* polusi udara dengan menampilkan hasil data *Wireless Sensor Network* polusi udara pada Thingspeak Server, membandingkan kinerja jaringan *broadcast* dan topologi *star*.

1.4 Organisasi Tulisan

1. Pendahuluan

Pada bab ini dijabarkan mengenai latar belakang, perumusan masalah, tujuan, batasan masalah, metodologi penyelesaian masalah dan sistematika penulisan.

2. Studi Terkait

Pada bab ini terdapat ringkasan hasil kajian pustaka yang terkait dengan masalah yang diajukan penulis pada bab pendahuluan dan menjelaskan teori-teori pendukung yang digunakan untuk menyelesaikan masalah yang diajukan oleh penulis.

3. Sistem yang Dibangun

Pada bab ini berisi analisis dan rancangan sistem yang dibangun, pengumpulan dan ilustrasi pengolahan data yang dilakukan dalam membangun sistem serta skenario pengujian yang dilakukan.

4. Evaluasi

Pada bab ini membahas hasil pengujian dari sistem yang dibangun serta menjelaskan hasil analisis terhadap hasil pengujian yang telah dilakukan.

5. Kesimpulan

Pada bab ini menjelaskan kesimpulan dari keseluruhan hasil pengerjaan tugas akhir yang mengacu pada tujuan penelitian, skenario pengujian dan analisis hasil pengujian pada bab-bab sebelumnya serta memberikan saran untuk penelitian selanjutnya.

2. Studi Terkait

Berdasarkan [4] sistem yang dibangun memanfaatkan teknologi *Wireless Sensor Network* dengan menggunakan 11 node pada network simulator dan membandingkan topologi *star*, *mesh* dan *tree* untuk membangun sistem monitoring jembatan agar dapat memantau jika terjadi kerusakan pada jembatan. Pengujian ini menggunakan ZigBee sebagai transmisi datanya. Hasil pengujian dengan mengirimkan 156 paket, topologi *star* mendapatkan *packet loss* sebanyak 140 paket atau 90,32% paket yang hilang. Topologi *mesh* mendapatkan *packet loss* sebanyak 1 paket atau 0,641% paket yang hilang, sedangkan topologi *tree* mendapatkan *packet loss* 0%. Sehingga didapatkan topologi yang terbaik ialah *tree*, *mesh*, dan *star*.

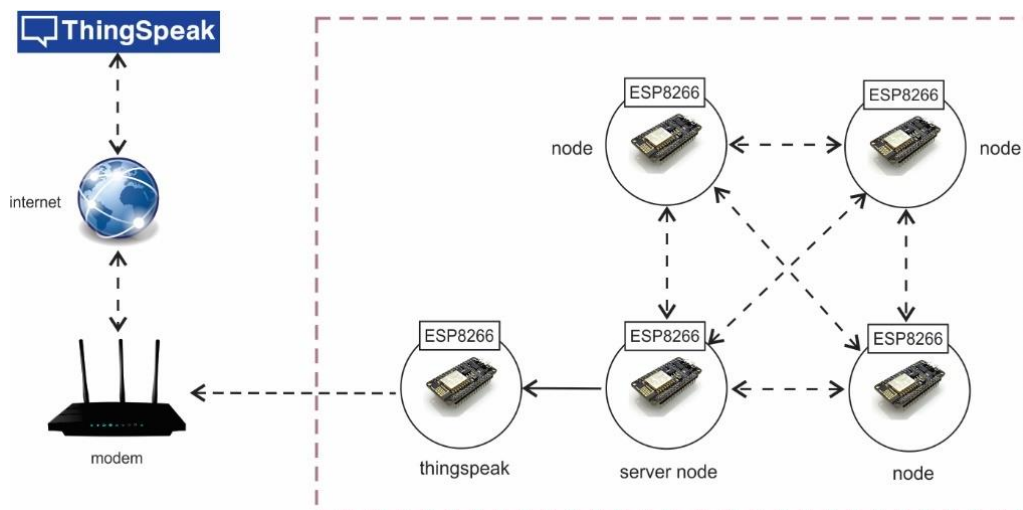
Berdasarkan [5] sistem yang dibangun ialah pemantauan polusi udara berbasis *Wireless Sensor Network* dengan menggunakan perangkat ZigBee dan *General Packet Radio Service* (GPRS) untuk protokol transfer datanya serta menerapkan topologi *mesh* seperti pohon yang terdiri dari node koordinator, router, dan end device. Sistem ini dibangun menjadi beberapa *cluster* yang masing-masing *cluster* terdiri dari sekumpulan node stasiun. Node stasiun terdiri dari perangkat polusi udara. Setiap node memiliki router dan pada masing-masing *cluster* memiliki satu node koordinator yang berfungsi membangun dan mengendalikan komunikasi data dengan sejumlah node stasiun dan meneruskan datanya. Kelemahan dari sistem ini membutuhkan banyak router untuk proses transmisi datanya karena setiap *cluster* memiliki satu buah router agar dapat saling berkomunikasi dengan *cluster* lainnya.

Wireless Sensor Network, Internet of Things, dan data cloud saat ini banyak digunakan oleh masyarakat kita dimana membutuhkan topologi yang cerdas untuk berkomunikasi antar perangkat. Sistem yang dibangun ialah sistem komunikasi berbasis *IoT* untuk mendeteksi hama pada tumbuhan, beberapa perangkat yang terhubung secara *real time* akan digunakan untuk memantau hama tikus terutama dalam industri makanan dan kegiatan pertanian. Sistem ini menggunakan perangkat lunak Arduino, Nordic 2.4GHz, transduser dan sensor detektor gerak, yang bertujuan agar bisa memonitoring hama tikus perperiodik mingguan, bulanan ataupun tahunan yang diakses dari jarak jauh menggunakan smartphone ataupun komputer. Topologi yang digunakan pada pengujian ini ialah *star* dengan menggunakan banyak node, hasil penelitian setiap mengirim 128 paket terdapat 4 hingga 6 paket hilang yang dipengaruhi oleh jarak node. Pada topologi mesh didapatkan *packet loss* yang lebih kecil akan tetapi ketika sejumlah besar node mengirimkan data secara bersamaan maka terjadi tabrakan yang mengakibatkan hilangnya paket tersebut. Jadi pada sistem pemantauan hama tikus ini membatasi penggunaan node sebanyak 36 node untuk agar paket yang dikirim sukses [6].

3. Sistem yang Dibangun

3.1 Gambaran Umum Sistem

Gambaran umum pada sistem yang akan dibangun ialah akan menggunakan 5 buah ESP8266, 4 buah sensor *MQ-5*, aplikasi *Wireshark*, dan Thingspeak Server. Berikut gambaran umum dari sistemnya.



Gambar 4 Gambaran Umum Sistem

Berdasarkan gambaran umum diatas menggambarkan bahwa penerapan jaringan *broadcast* pada sistem ini *WSN* diaplikasikan untuk melihat kinerja jaringannya. Penerapan jaringan *broadcast* ini menggunakan library *easyMesh* dimana penentuan jalur *node* telah ditentukan secara otomatis dari library *easyMesh* tersebut. Empat *node* pada gambaran umum diatas merupakan sensor untuk mendeteksi polusi udara, satu *node* sebagai *server*, dan satunya lagi sebagai *thingspeak server* yang berfungsi untuk mengupload data dari *server* ke Thingspeak Server. Pada penelitian ini diterapkan jaringan *broadcast* dan membandingkan kinerja topologi *star* dengan mengacu pada parameter *packet loss*[7], untuk mengukur *delay* jaringan *broadcast* dilakukan hanya pada saat esp8266 mengirimkan datanya ke Thingspeak Server menggunakan *wireshark*. Sensor yang digunakan untuk mendeteksi polusi udara ialah sensor *MQ-5* [8]. Cara mengukur polusi udara ialah meletakkan 4 *node* beserta sensornya pada jarak yang berbeda dan menggunakan mikrokontroler ESP8266 [10] sebagai media transmisi datanya. Setelah itu hasil dari deteksi sensor akan ditampilkan oleh Thingspeak Server [11] dan dilakukan *monitoring* menggunakan *wireshark* [12] untuk mengetahui kinerja jaringannya. *Background traffic* pada penelitian ini yaitu semakin banyak *node* yang digunakan maka semakin banyak pula *background traffic* / gangguan pada jaringan *broadcast* ini saat melakukan pengiriman karena datanya berpusat pada satu *receiver* dan juga penerapan *udp flood* pada server yang dapat mempengaruhi pengiriman data dan menghasilkan *packet loss* yang sangat besar. Penambahan sensor pada penerapan jaringan *broadcast* ini tidak dapat diterapkan *virtual sensor* disebabkan oleh penelitian ini menggunakan esp8266 sedangkan *virtual sensor* hanya dapat diterapkan pada Arduino saja.

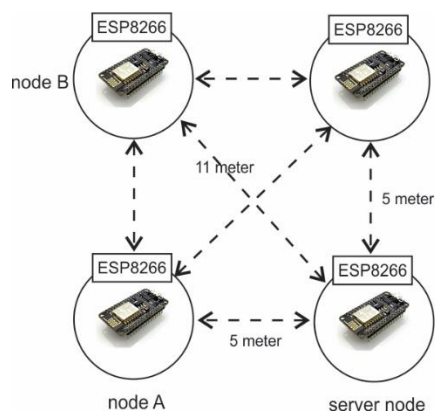
3.2 Skenario Pengujian

Pada penelitian ini untuk mengetahui performa dari sistem yang dibangun akan dilakukan pertama pengujian jarak node, meliputi pemberian jarak yang berbeda antar *server* dan masing-masing *node*. Pengujian ini dilakukan untuk mengetahui performa kinerja jaringan serta reaksi data pada saat melakukan pengiriman dengan jarak yang berbeda-beda. Kedua pengujian *wireshark*, akan dilakukan capture data yang akan dikirimkan ke Thingspeak server menggunakan *wireshark*. Pengujian ini dilakukan untuk mengetahui kinerja jaringan pada saat mengirim data ke Thingspeak server, parameter yang diambil untuk mengetahui kinerja jaringannya yaitu *packet loss* dan *delay* yang didapatkan. Ketiga pengujian topologi, akan dilakukan pengujian antara jaringan *broadcast* dan *star* dengan cara mengirimkan 50 data ke server. Pengujian ini dilakukan agar mengetahui seberapa besar *packet loss* pada jaringan lokal *broadcast* dan topologi *star* untuk mengetahui model jaringan yang mana mempunyai performa lebih bagus. Keempat pengujian udp flood, meliputi memberikan DoS pada server sehingga mempengaruhi pengiriman datanya. Pengujian ini dilakukan untuk mengetahui performa dari jaringan *broadcast* dan topologi *star* serta pengaruh dari udp flood ketika diterapkan pada *server*. Kelima pengujian *thingspeak server*, meliputi monitoring hasil data yang telah ditampilkan oleh Thingspeak server. Pengujian ini dilakukan untuk mengetahui performa sistem pada saat menampilkan datanya ke antarmuka Thingspeak server dalam bentuk grafik.

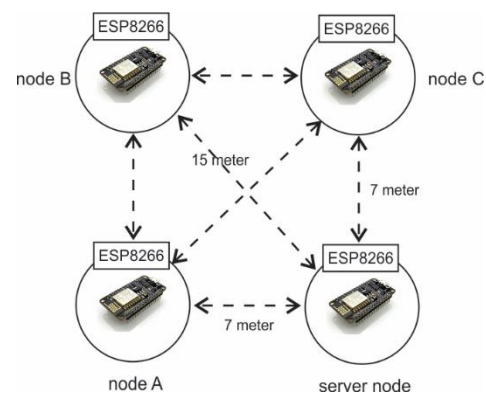
4. Evaluasi

4.1 Analisis Pengujian Jarak Node

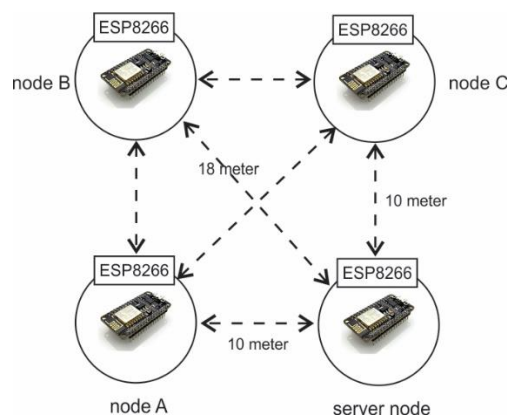
Pengujian ini dilakukan untuk mengetahui kinerja jaringan *broadcast* serta hasil data yang ditampilkan masing-masing *node* diberi jarak yang berbeda-beda. Berikut gambaran pengujiannya.



Gambar 6 Pengujian Jarak I



Gambar 7 Pengujian Jarak II



Gambar 8 Pengujian Jarak III

Berdasarkan gambar 6,7, dan 8 akan dilakukan pengujian jarak antara *node* ke server. *Node* A dan C akan diberikan jarak yang sama sedangkan *node* B diberikan jarak yang berbeda dengan server. Dari ketiga pengujian masing-masing memiliki varian jarak yang berbeda-beda. Pada penelitian ini menggunakan sensor MQ-5 dengan satuan ppm semakin tinggi nilai ppmnya maka semakin tercemar pula udara yang dideteksi, serta ditampilkan panjangnya data yang dihasilkan oleh sensor MQ-5 dengan istilah length.

Hasil pengujiannya (lihat pada lampiran tabel A,B, dan C). Rata-rata pengujian sekitar 50 data yang sampai ke server terlihat bahwa semakin jauh jarak *node* maka semakin jarang pula server menerima data dari *node* dan sebaliknya semakin dekat jarak *node* dan server semakin cepat pula pengiriman datanya. Oleh karena itu analisa pada pengujian kali ini ialah jarak antara *node* dan server mempengaruhi proses pengiriman data.

4.2 Analisis Pengujian Wireshark

Pengujian ini dilakukan untuk mengetahui kinerja jaringan pada saat ESP8266 sebagai thingspeak mengirim datanya ke Thingspeak Server dan diberikan jarak 10 meter antara *node* dan *server*. Pada sistem data upload ini menerapkan sistem *interrupt* dimana data yang terlebih dahulu masuk ke ESP8266 sebagai thingspeak data itu pula yang akan di upload ke Thingspeak Server. Cara *capture* data dari *wireshark* ialah menghubungkan ESP8266 pada jaringan internet lalu *capture* transmisi datanya menggunakan *wireshark*. Pada pengujian kali ini ip ESP8266 ialah 192.168.137.78 sedangkan ip tujuan ialah 52.0.246.0. Cara filter pengiriman data ke Thingspeak Server di *wireshark* ialah `ip.dst==52.0.256.0&&tcp`. Pada hasil *capture wireshark* dapat dilihat informasi [FIN,ACK] menandakan bahwa paket berhasil dikirim sedangkan informasi *packet loss* dapat dilihat di RETRANSMISSION menandakan paket tidak berhasil terkirim. Pada penelitian ini untuk melihat kinerja jaringannya dengan memperhatikan *packet loss* yang didapatkan dan *delay*, berikut perhitungan *packet loss* dan *delay*.

- Jumlah paket yang dikirim = 157
- Jumlah paket yang diterima = 151

$$\text{Presentase packet loss} = \frac{\text{jumlah paket dikirim} - \text{jumlah paket sampai}}{\text{jumlah paket dikirim}} \times 100\%$$

$$\begin{aligned} \text{Presentase packet loss} &= (157 - 151) / 157 \\ &= 0.03821 \times 100\% = 3.82\% \end{aligned}$$

Berdasarkan perhitungan *packet loss* diatas didapatkan besar nilainya yaitu, 3.82% dengan melihat tabel standarisasi. Berikut tabel standarisasinya.

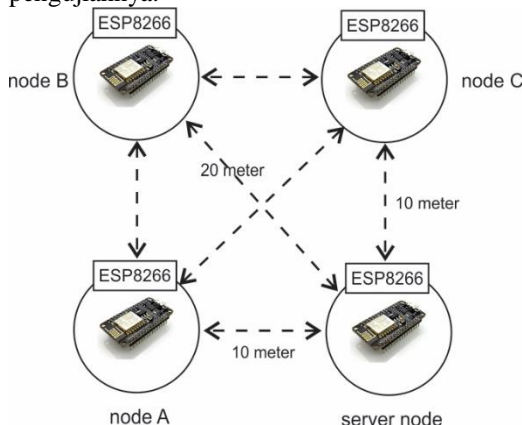
Tabel 5 Standarisasi *packet loss* [9]

| Kategori Degradasi | Packet Loss (%) | Indeks |
|--------------------|-----------------|--------|
| Sangat Bagus | 0 | 4 |
| Bagus | 3 | 3 |
| Sedang | 15 | 2 |
| Jelek | 25 | 1 |

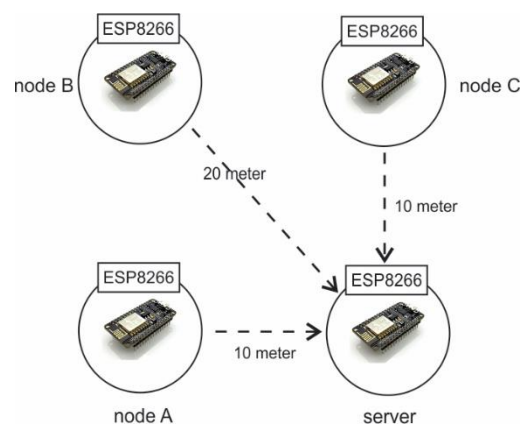
Berdasarkan tabel diatas terlihat bahwa *packet loss* yang didapatkan sebesar 3% termasuk dalam kategori “Bagus“. Rata-rata dari perhitungan *delay* dari pengujian *wireshark* ialah 0.308969794 *second*. Oleh karena itu analisa pada pengujian ini didapatkan kinerja jaringan yang bagus karena menghasilkan *packet loss* dan *delay* yang kecil.

4.3 Analisis Pengujian Topologi

Pengujian ini dilakukan untuk mengetahui kinerja jaringan pada saat melakukan pengiriman data ke server dengan membandingkan dua jenis jaringan yaitu jaringan *broadcast* dan *star*. Berikut gambaran pengujiannya.



Gambar 9 Block Diagram Jaringan *broadcast*



Gambar 10 Block Diagram Topologi *star*

Berdasarkan gambar 9 dan 10 akan dilakukan pengujian jaringan *broadcast* dan topologi *star*. Kedua model jaringan ini diberikan masing-masing jarak, *node A* diberi jarak 10 meter sedangkan *node B* diberi jarak 20 meter. Pada pengujian kali ini akan dikirim 50 data dari *node B* dan akan dilihat kinerja masing-masing jaringan dengan memperhatikan parameter *packet loss*. Berikut hasil pengujiannya.

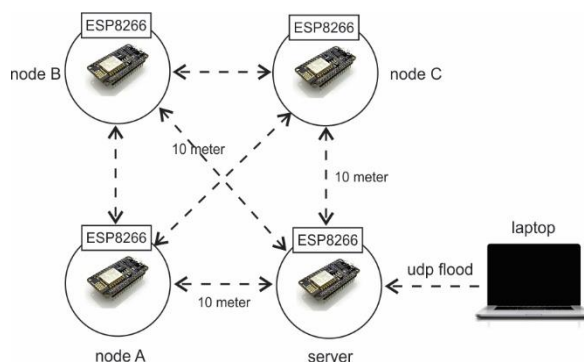
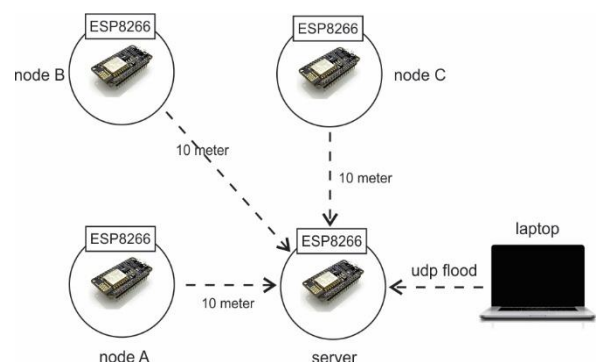
Tabel 6 Hasil pengujian *packet loss*

| Jaringan | Data dikirim | Data diterima | Packet loss |
|-----------|--------------|---------------|-------------|
| Broadcast | 50 | 50 | 0% |
| Star | 50 | 47 | 6% |

Berdasarkan tabel 6 terlihat bahwa dari hasil pengujian *packet loss* masing-masing jaringan memiliki perbedaan. Jaringan *broadcast* menghasilkan *packet loss* 0% sedangkan topologi *star* 6%. Oleh karena itu pada pengujian kali ini terlihat bahwa jaringan *broadcast* lebih unggul dibandingkan topologi *star* dengan menghasilkan *packet loss* yang lebih kecil, disebabkan karena jaringan *broadcast* setiap *node*-nya saling terhubung jadi memiliki banyak pilihan jalur pengiriman dan meminimalisir terjadinya kehilangan data sedangkan topologi *star* hanya memiliki satu jalur pengiriman yang langsung terhubung dengan server.

4.4 Analisis Pengujian Udp Flood

Pengujian ini dilakukan untuk mengetahui kinerja jaringan pada saat server diserang oleh *Udp Flood* dengan membandingkan dua jenis jaringan yaitu jaringan *broadcast* dan *star*. *Udp flood* adalah serangan *DoS* (*denial of service*) menggunakan *UDP* (*User Datagram Protokol*) dimana paket dalam jumlah yang sangat banyak akan dikirimkan ke server mengakibatkan kesulitan untuk menampung sehingga mempengaruhi pengiriman data melalui server. Berikut gambaran pengujiannya.

Gambar 11 Block Diagram Jaringan *broadcast*Gambar 12 Block Diagram topologi *star*

Berdasarkan gambar 11 dan 12 akan dilakukan pengujian jaringan *broadcast* dan topologi *star*. Kedua model jaringan ini diberikan masing-masing jarak *node* diberi jarak 10 meter ke server sedangkan server akan diserang oleh *udp flood*. Berikut hasil pengujiannya

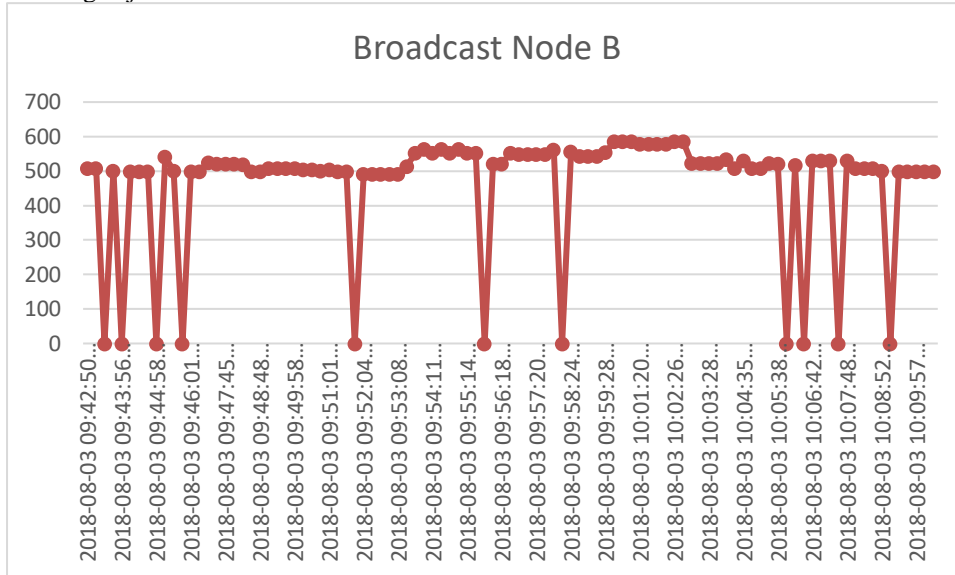
Tabel 7 Hasil Pengujian *packet loss* dari Udp Flood

| Jaringan | Data dikirim | Data diterima | Packet loss |
|-----------|--------------|---------------|-------------|
| Broadcast | 100 | 17 | 83% |
| Star | 100 | 12 | 88% |

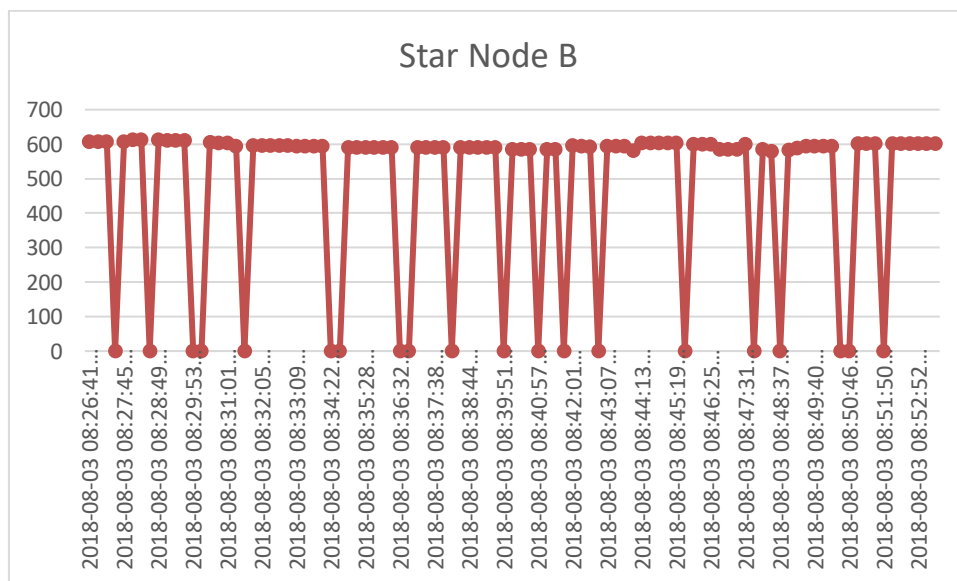
Berdasarkan tabel diatas terlihat bahwa data dari Thingspeak server mendapatkan *packet loss* dari jaringan *broadcast* sebesar 83% sedangkan topologi *star* sebanyak 88%. Oleh karena itu jaringan *broadcast* lebih unggul dibandingkan topologi *star* meskipun mendapatkan *packet loss* yang sangat besar Udp flood ini menyerang server dengan mengirimkan paket yang jumlahnya sangat banyak sehingga server mengalami peningkatan traffic yang mempengaruhi pengiriman paket dari *node* ke server Penerapan *udp flood* pada server ini sangat mempengaruhi pengiriman paket dari *node* ke server karena menghasilkan *packet loss* yang sangat besar.

4.5 Analisis Pengujian Thingspeak Server

Pengujian kali ini dilakukan untuk mengetahui data sensor yang dikirim oleh *node* dapat ditampilkan oleh antarmuka Thingspeak Server dan melihat perbandingan data yang dihasilkan antara jaringan *broadcast* dan *star*, pada sistem ini menerapkan konsep *IoT*. Berikut hasil pengujian yg dihasilkan dari *node* B dengan jarak 15 meter ke server.



Gambar 10 Grafik data jaringan *broadcast*



Gambar 11 Grafik data topologi *star*

Berdasarkan gambar 10 dan 11 terlihat bahwa terdapat perbedaan grafik data pada penerapan jaringan *broadcast* dan topologi *star*. Pengujian ini dilakukan agar hasil data dapat dilihat tanpa menggunakan serial dan melihat perbedaan ketika menggunakan jaringan *broadcast* dan *star*. Oleh karena itu terlihat dari grafik data bahwa jaringan *broadcast* lebih sering menerima data karena menghasilkan *packet loss* yang lebih kecil dibandingkan topologi *star*.

5. Kesimpulan

Berdasarkan hasil rancangan sistem dan pengujian yang telah dilakukan, didapatkan kesimpulan telah berhasil merancang dan mengimplementasikan *Wireless Sensor Network* menggunakan mikrokontroler ESP8266 dengan menghubungkan sensor MQ-5 untuk mendeteksi polusi udara serta menerapkan jaringan *broadcast*. Telah berhasil melakukan analisis *traffic* pada implementasi *Wireless Sensor Network* polusi udara, diantaranya sebagai berikut :

- Pada analisis *wireshark*, hasil penelitian dari pengiriman 157 paket, jaringan *broadcast* mendapatkan kinerja jaringan yang bagus dengan *packet loss* 3.82% dan *delay* 0.308969794 *second*.
- Pada analisis *packet loss* di jaringan lokal, hasil yang didapatkan penelitian jaringan *broadcast* jauh lebih unggul dengan *packet loss* yaitu 0% dibandingkan topologi *star* mendapatkan *packet loss* yaitu 6%.
- Pada analisis Thingspeak server diterapkan sistem *interrupt* dimana data yang lebih dahulu sampai ke server maka data itu pula yang akan diupload ke Thingspeak server dan dapat dilihat pada Thingspeak server grafik data jaringan *broadcast* lebih kecil menghasilkan *packet loss* dibandingkan topologi *star*.

Adapun saran untuk penelitian selanjutnya yaitu:

- a. Menambahkan lebih banyak *node* pada penerapan jaringan *broadcast*.
- b. Membandingkan jaringan *broadcast* dengan topologi lainnya seperti ring, bus, dan tree.
- c. Menggunakan ip address untuk identitas setiap *node*, dan mengatur protokol routing secara manual pada penerapan jaringan *broadcast*.

Daftar Pustaka

- [1] M. Aafiuddin, Setyorini, dan M. A. Nugroho, "Analisis komparasi Performa Algoritma CoDel dan RED sebagai Active Queue Management pada Jaringan Sensor Nirkabel", Skripsi, Universitas Telkom, 2016.
- [2] M. Shah, M. Y. Khan, H. Khan, M. N. Hayat, S. R. Jan, "Congestion Control in Wireless Sensor Networks An overview of Current Trends", *International Journal of Science and Engineering Applications*, vol. 5, no.5, 2016.
- [3] F. Wang, J.Liu, "Dust-Cycle-Aware Broadcast in Wireless Sensor Network", in *Prosiding IEEE Infocom*, pp. 468-476, 2009.
- [4] E. N. Amalina, E. Setijadi, and Suwadi, "Perbandingan Topologi WSN (Wireless Sensor Network) Untuk Sistem Pemantauan Jembatan", in *Prosiding Conference on Smart-Green Technology in Electrical and Information Systems*, hal. 85-89, Bali, 2013.
- [5] M. Iqbal, M. Fuad, H. Sukoco, and H. Alatas, "Hybrid Tree-Like Mesh Topology as New Wireless Sensor Network Platform", *Telkonnika*, vol. 13, no. 3, 2016.
- [6] C.Cambra, S. Sendra, L.Garcia, and J. Lloret, "Low Cost Wireless Sensor Network for Rodents Detection", Skripsi, de Granada University, 2017.
- [7] J. Pamungkas, Wirawan, "Desain Real-Time Monitoring Berbasis Wireless Sensor Network Upaya Mitigasi Bencana Erupsi Gunungapi", *JNTETI*, Vol 4, No 3, 2015.
- [8] M. F. Putra, A. H. Kridalaksana, and Z. Arifin, "Rancang Bangun Alat Pendeteksi Kebocoran Gas LPG Dengan Sensor MQ-6 Berbasis Mikrokontroler Melalui Smartphone Android Sebagai Media Informasi", *Jurnal Informatika Mulawarman*, Vol 12, No 1, 2017.
- [9] TIPHON, "Telecommunication and Internet Protokol Harmonization Over Networks; General aspect of Quality of Service(QoS)". Sophia Antipolis Cedex, France:TIPHON, 1999.
- [10] M. Mehta, "ESP8266: A Breakthrough in Wireless Sensor Networks And Internet of Things", *International Journal Electronics and Communications Engineering & Technology*, Vol 6, No 8, 2015.
- [11] N. Ome, G. S. Rao, "Internet of Things (IoT) based Sensors to Cloud system using ESP8266 and Arduino Due", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 5, No. 10, 2016.
- [12] R. Wulandari, "Analisis QoS (Quality Of Service) Pada Jaringan Internet (Studi Kasus: Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi)", *Jurnal Teknik Informatika dan Sistem Informasi*, Vol. 2, No. 2, 2016.

