

kategori tidak sehat seperti pada tabel (II.1). Hal ini mendakan bahwa kondisi partikel material tidak sehat untuk manusia.

4.3 Pengujian Sistem Komunikasi

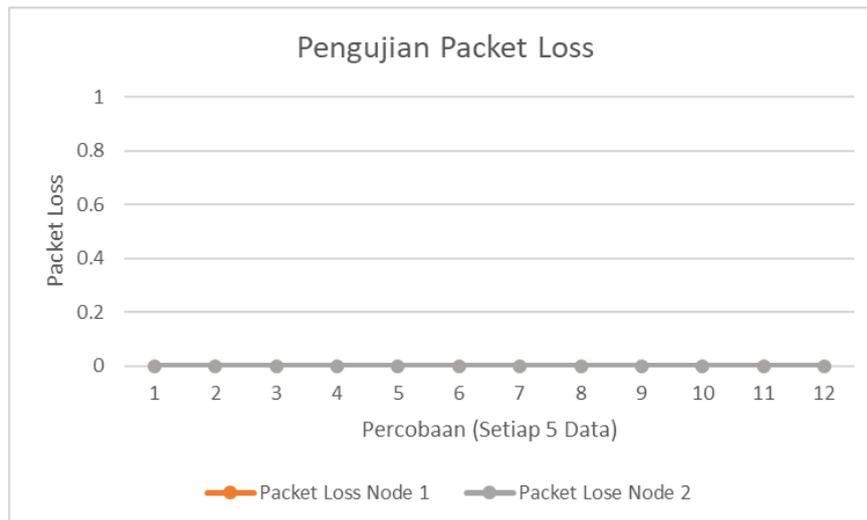
Pengujian sistem komunikasi atau *Quality of Service* (QoS) dilakukan menggunakan perhitungan manual dan *software* WireShark untuk mendapatkan parameter nilai *packet loss*, *delay time*, dan *Jitter* untuk mengetahui kualitas dari kinerja perangkat *Internet of Things* (IoT) yang digunakan. Berikut adalah hasil pengujian parameter pada *Quality of Service*.

3.3.1 Pengujian *Packet Loss* Pengiriman Data

Pengujian *Packet Loss* memiliki tujuan untuk mengetahui jumlah perbandingan paket data yang dikirim dan yang tidak di terima. Pengujian ini dilakukan dengan cara mengirim data ke *platform* IoT sebanyak 60 kali dan di lakukan perbandingan berapa banyak data yang di terima oleh plaform IoT. Menghitung *packet loss* dapa dilakukan dengan persamaan sebagai berikut:

$$Packet Loss = \frac{(\text{Paket Data Dikirim} - \text{Data Diterima}) \times 100\%}{\text{Paket Data Dikirim}} \quad (4.2)$$

Dari Persamaan 4.2 diatas akan diperoleh *packet loss* yang ingin dicari. Berikut adalah hasil pengujian *packet loss* pada sensor *node* 1 dalam 60 kali pengiriman data.



Gambar 4. 7 Grafik Pengujian *Packet Loss* Sensor *Node 1* dan Sensor *Node 2*

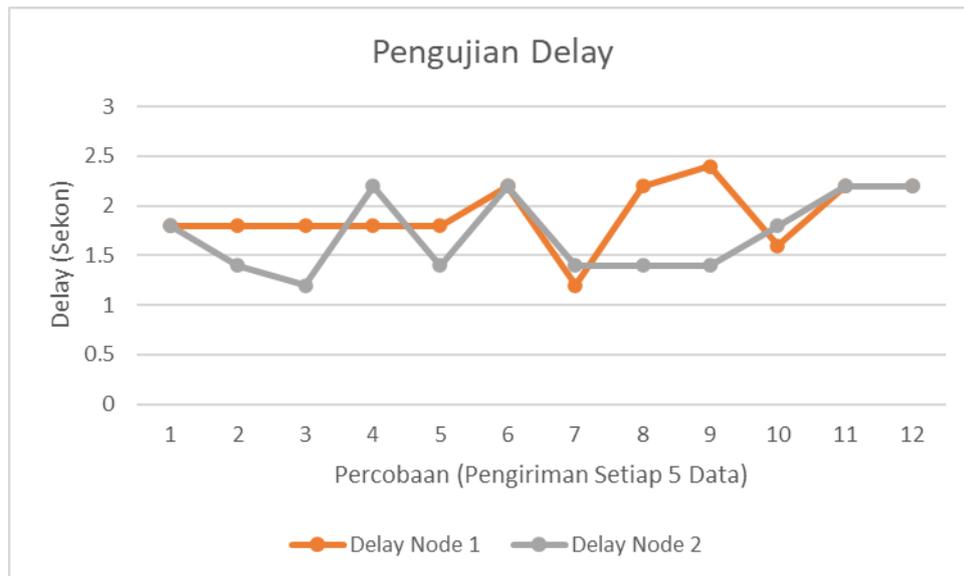
Hasil dari pengujian *packet loss* pada sensor *node1* dan sensor *node 2* didapatkan tidak ada paket data yang hilang karena semua data berhasil diterima oleh Antares sehingga komunikasi antara Antares dengan alat berjalan dengan baik.

3.3.2 Pengujian *Delay* Pengiriman Data

Pengujian ini dilakukan untuk mengukur nilai *delay* pada sensor *node 1* dan sensor *node 2* yang merupakan akumulasi berbagai waktu tunda pada suatu jaringan. Pengujian nilai *delay* dilakukan dengan menghitung selisih waktu antara pembacaan di Antares dengan serial monitor pada Arduino IDE. Pengujian dilakukan dengan pengiriman sebanyak 60 kali dari mikrokontroler ESP32 sensor *node 1* dan mikrokontroler ESP32 sensor *node2* menuju Antares. Perhitungan dilakukan dengan membandingkan waktu yang muncul pada Serial monitor di Arduino IDE dan waktu yang muncul pada Antares. Menghitung rata-rata *delay* dapat dilakukan dengan menggunakan persamaan berikut:

$$\text{Rata - rata Delay} = \frac{\text{Total Delay}}{\text{Total Data Diterima}} \quad (4.3)$$

Dari persamaan 4.3 diatas akan di peroleh nilai *delay* yang ingin dicari, berikut adalah hasil pengujian ddelay dalam 60 kali pengiriman data.



Gambar 4. 8 Grafik Penguujian *Delay* Sensor *Node 1* dan Sensor *Node 2*

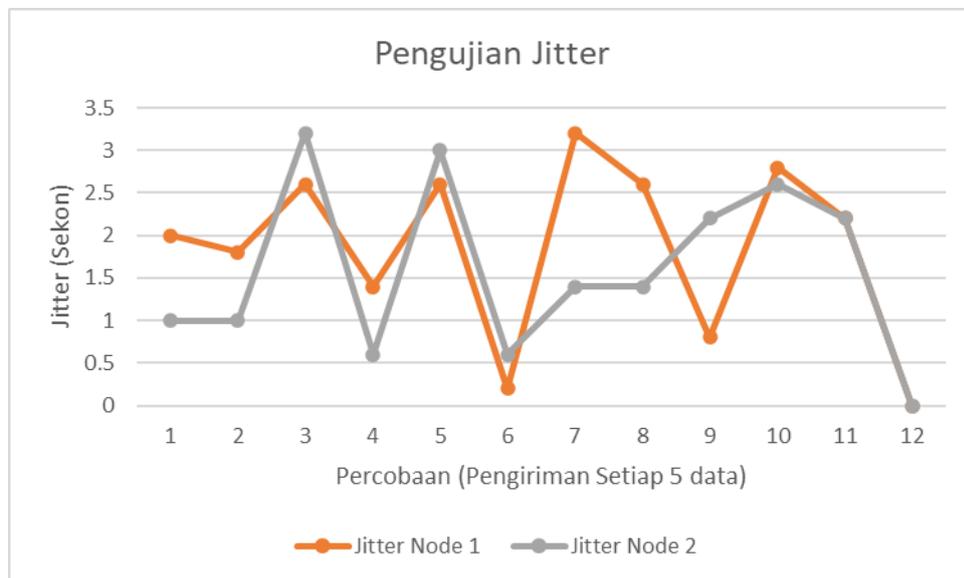
Pada Gambar 4.10 diatas menunjukkan *delay* pengiriman setiap 5 data. Pada grafik delay node 1 dan node 2 diatas dapat dilihat adanya kenaikan dan penurunan grafik, hal itu terjadi karna jaringan internet pada node 1 dan node 2 menggunakan jaringan internet yang sama, sehingga hasil dari delay node 1 dan node 2 tidak stabil. Dari hasil pengujian di atas dan dengan menggunakan persamaan 4.3 maka dapat diambil bahwa *delay* pengiriman data dari mikrokontroler ESP32 sensor *node 1* ke Antares adalah 23 detik dengan rata- rata 1,916 detik, sedangkan untuk hasil pengujian *delay* pengiriman data dari mikrokontroler ESP32 sensor *node 2* ke Antares adalah 20,6 detik dengan rata-rata *delay* 1,716 detik. Pengujian ini tergolong dalam kategori jelek dan tergolong pada indeks 1 seperti tabel diatas.

3.3.3 Penguujian Jitter Pengiriman Data

Penguujian ini dilakukan untuk mengukur variasi delay atau perbedaan selang waktu paket datang pada lokasi tujuan. Penguujian jitter dilakukan dengan cara mengambil data yang sudah didapat dari pengujian delay sebelumnya. Jitter memiliki pergitungan yang dapat dilakukan dengan menggunakan persamaan sebagai berikut:

$$Jitter = \frac{\text{Total Variasi Delay}}{\text{Total Paket yang Diterima}} \quad (4.4)$$

Dari persamaan jitter pada 4.4 data yang diperlukan adalah total variasi delay dan banyak data yang dikirim. Berikut adalah hasil pengujian jitter dari 60 kali pengiriman data.



Gambar 4. 9 Grafik Pengujian *Jitter* Sensor *Node 1* dan Sensor *Node 2*

Pada Gambar 4.11 menunjukkan nilai jitter pengiriman setiap 5 data. Dari hasil pengujian jitter node 1 dan node 2 terlihat grafik mengalami kenaikan dan penurunan, hal ini di karena pengolahan nilai jitter yang dilakukan menggunakan nilai data delay sebelumnya, sehingga nilai jitter yang didapatkan mengalami hal yang sama seperti nilai data delay , hal ini terjadi dikarenakan jaringan internet yang tidak stabil. memaka dapat diambil bahwa jitter pengiriman data dari mikrokontroler sensore *node 1* ke Antares sebesar 22,2 detik dengan rata rata 1,85 detik dan Jitter pengiriman data dari mikrokontroler sensor node 2 ke Antares sebesar 19,2 detik dengan rata-rata 1,6 detik.

4.4 Tampilan data terkirim pada Antares sebagai IoT Platform



Gambar 4. 10 Tampilan Pada Antares

Pada Gambar diatas dapat dilihat tampilan Antares sebagai IoT Platform yang digunakan penulis. Setelah membuka website dan berhasil login dengan email terdaftar, selanjutnya penulis menentukan *application* dan *device* yang digunakan seperti gambar pada nomor 1. Selanjutnya, *device* yang telah di-set sebelumnya dihubungkan pada program di Arduino agar Antares dapat menerima data yang dikirimkan dari mikrokontroler ESP32 sensor node 1 dan sensor node 2. Pada pengujian yang dilakukan, nilai data yang dikirim berupa nilai gas Partikel Material (PM 10), gas Karbon Monoksida (CO), gas Nitrogen Dioksida (NO₂), gas Ozon (O₃), dan gas Sulfur Dioksida (SO₂), seperti pada nomor 3 dan nomor 2. Data yang dikirim akan ditampilkan secara real-time yang dapat terlihat seperti gambar nomor 4. Data yang tersimpan pada Antares dapat di *unduh* dengan format json dengan menekan tombol yang terlihat pada nomor 5.

4.5 Tampilan User Interface Aplikasi Android



Gambar 4. 11 Tampilan *Layer 1 Mobile App*

Pada gambar 4.13 menunjukkan tampilan awal dari aplikasi Android. *User Interface* pada aplikasi *Android* dirancang dan di *desain* menggunakan *website* MIT App Inventor. Setelah selesai dan berhasil membuat keseluruhan sistem dan tampilan aplikasi, unduh dan *install* dahulu aplikasi tersebut pada *smartphone*. Pada layar utama dari aplikasi *Android* ini akan di tampilkan data dari sensor *node 1* yang terlihat nilai gas ozon, nilai gas Sulfur Dioksida, nilai gas Karbon Monoksida, nilai gas Nitrogen Dioksida, dan Partikulat Material. Tombol bertuliskan *Next* akan menghubungkan ke halaman seperti pada gambar berikut.



Gambar 4. 12 Tampilan *Layer 2 Mobile App*

Gambar 4.14 merupakan tampilan *Layer 2* dari *mobile App* yang menampilkan data sensor *node 2*, pada halaman ini akan di tampilkan nilai gas ozon, nilai gas Sulfur Dioksida, nilai gas Karbon Monoksida, nilai gas Nitrogen Dioksida, dan Partikulat Material. Tombol *Back* berfungsi untuk menghubungkan pengguna ke halaman utama.

BAB V

SIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa rancang bangun alat sistem monitoring polusi udara menggunakan sensor multi titik berbasis Internet of Things, didapatkan beberapa kesimpulan sebagai berikut:

1. Sistem ini sudah berhasil dibuat, yaitu merancang alat *monitoring* polusi udara menggunakan sensor multi titik berbasis Internet of Things dalam memantau kualitas udara. Setiap node sensor memiliki 5 buah sensor, yaitu sensor MQ-131, sensor MQ-136, Sensor MQ-7, sensor MICS-6814, dan sensor SHARP GP2Y1010AU0F.
2. Sistem dapat menerima data yang telah dikirim ke Antares, lalu menampilkan hasil *monitoring* dengan baik pada aplikasi Android. Delay pengiriman rata-rata pada sensor node 1 dan sensor node2 sebesar 1,916 detik dan 1,716 detik. Dalam pengiriman paket data pada sensor *node* 1 dan sensor *node* 2, tidak ada paket data yang hilang dalam pengiriman kemudian data tersebut dapat dilihat atau di akses melalui *mobile app*.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan pada Tugas Akhir ini, saran yang dapat diberikan penulis sebagai berikut.

1. Pengembangan penelitian selanjutnya, melakukan perbandingan alat monitoring yang di rancang dengan alat ukur kualitas udara yang aktual, agar alat yang di rancang memiliki nilai pembanding untuk melakukan kalibrasi.
2. Untuk pengembangan pada aplikasi *mobile app* di tambahkan *fitur public download* agar memudahkan pengguna melihat riwayat nilai polusi udara.

DAFTAR PUSTAKA

- [1] Setiawan, Verda Nando. 2020. Kendaraan Bermotor Sumbang 60% Polusi, Menhub Dorong Mobil Listrik. [Diakses 9 Juni 2022]
- [2] StateofGlobalAir, “Health Impact, Burden on Your Health” Jun 10, 2022 Available: StateofGlobalAir,. [Diakses 10 Juni 2022].
- [3] IQAir, “Air quality in Indonesia” Jun 10, 2022 Available: IQAir,. [Diakses 10 Juni 2022].
- [4] Sianipar, Afrida Br, and Febri Yuliani. Optimalisasi Fungsi Papan Indeks Standar Pencemar Udara (Ispu) oleh Dinas Lingkungan Hidup dan Kebersihan Kota Pekanbaru. Diss. Riau University, 2017.
- [5] Haryono Setiyo Huboyo , Endro Sutrisno, Analisis Konsentrasi Particulate Matter (PM10) Pada Udara Diluar Ruang (Studi Kasus : Stasiun Tawang – Semarang)
- [6] Wicaksono, Yonny., Suismono, Andi. 2010. Deteksi Gas Berbahaya CO, CO₂, dan NO_x Dengan Penampil Dot Matrix dan Level Bahaya serta Besarnya, <http://digilib.its.ac.id/public/ITSNonDegree-14525-paperpdf.pdf> diakses tanggal 01 april 2016.
- [7] Muhammad Zikri, Rizaldy Khair, *Rancang Bangun Monitoring Polusi Udara Berbasis Arduino*, Jurnal Teknovasi Volume 05, Nomor 01, 2018, 27 – 38.
- [8] Seaton, A. (2015). Particulate air pollution and acute health effects
- [9] P. M. Mannucci and M. Franchini, “Health effects of ambient air pollution in developing countries,” *Int. J. Environ. Res. Public Health*, vol. 14, no. 9, pp. 1–8, 2017, doi: 10.3390/ijerph14091048.
- [10] Elvin Dwi Handoko, “Analisis Dampak Nirtrogen Dioksida (N₂O) Di Kota Yogyakarta.” [online]. Available <https://dspace.uui.ac.id/handle/123456789/30728>
- [11] Aljuaid, H., & Alwabel , N(2019). Air pollution prediction using machine learning algorithms. *International Journal of Engineering and Advanced Technology*, 8(6 Special Issue 3), 160-164.

- [12] D. Setiadi , M. Nurdin, and A. Muhaemin, “PENERAPAN INTERNET OF THINGS (IoT) PADA SISTEM MONITORING IRIGASI (SMART IRIGRASI),”*Jurnal Infotronik*, vol. 3, no 2, 2018.
- [13] Gagan Parmar, Sagar Lakhani, dan Manju K. Chattopadhyay, “An IoT Based Low-Cost Air Pollution Monitoring System”, *Proceeding International conference on Recent Innovations in Signal Processing and Embedded Systems*, 2017
- [14] “Antares | Reliable IoT Platform.” <https://antares.id/id/docs.html>(acssed Desember 27, 2022) .
- [15] J. Schiller et al., “Live Programming of mobile apps in App Inventor,” *Promot. 2014 – Proc . 2nd work. Program. Mob. Touch, Part SPLASH 2014*, pp. 1-8, 2014, doi:10.1145/2688471.268842.
- [16] M. Purwahid and J Trilok, “Analisis Quality of Service (QoS) Jaringan Internet Untuk Mendukung Rencana Strategis Infrastruktur Jaringan Komputer Di SMK N 1 Sukadana,” *JTSKI*, vol 2, no 3, pp. 100-109, 2019.
- [17] I.B.A.E.M. Putra, M.S.I Adnyana and L. Jasa, “Analisis Quality of Service pada Jaringan Komputer,” *Majalah Ilmiah Teknologi Elektro*, vol. 20, no 1, pp. 95-101, 2021
- [18] Aini, N., Ruktiari, R, Pratama, M.R., & Buana , A. F. (2019). *Sistem Prediksi Tingkat Pencemaran Polusi Udara dengan Algoritma Naïve Bayes di Kota Makassar.*
- [19] Armin Ernst, M.D., and Joseph D. Zibrak, M.D. “Carbon Monoxide Poisoning” 1998. *N Engl J Med* 1998; 339:1603-1608 DOI: 10.1056/NEJM199811263392206.
- [20] MQ131_Datasheet.(2019). MQ-131 Gas Sensor. Technical Data
- [21] HANWEI ELECTRONICS CO.,LTD. <http://www.hwsensor.com>
- [22] Datasheet, Data MQ-7 gas sensor - Hanwai, [Online].Available
- [23] GP2Y1010AU0F. (2017). GP2Y1010AU0F Sensor. Technical Data
- [24] Datasheet, Data MICS-6814 gas sensor - Hanwai, [Online].Available