

IMPLEMENTASI *CHANGE DETECTION* UNTUK EFISIENSI PENGIRIMAN PAKET DATA PADA IOT MASTER-SLAVE DI *GREENHOUSE*

Uray Ristian¹, Hirzen Hasfani², Uray Syaziman Kesuma Wijaya³
1,2,3) Sistem Komputer, Fakultas MIPA, Universitas Tanjungpura, Indonesia

Article Info

Article history:

Received: 04 September 2024

Revised: 30 Oktober 2024

Accepted: 31 Oktober 2024

ABSTRACT

Abstrak

Greenhouse adalah bangunan yang secara khusus dirancang untuk melindungi tanaman dari kondisi lingkungan eksternal yang tidak menguntungkan, serta mengatur lingkungan internal guna menciptakan kondisi ideal yang mendukung pertumbuhan tanaman secara optimal. Namun, sistem *monitoring* konvensional di *Greenhouse* mengalami masalah tingginya penggunaan kuota internet dan *packet loss* akibat ketergantungan pada WiFi *access point* (AP). Dalam penelitian ini, dikembangkan sebuah sistem monitoring tanaman di *Greenhouse* menggunakan arsitektur *master-slave* menggunakan protokol ESP-NOW untuk mengatasi masalah *packet loss*. Selain itu, algoritma *change detection* diterapkan pada node *master* guna mengurangi pengiriman data yang redundan serta memperkecil ukuran paket data yang dikirimkan. Hasil pengujian menunjukkan bahwa tingkat *packet loss* mencapai 1,43% dari total 222.793 pengiriman data selama tiga hari, yang termasuk dalam kategori "sangat bagus" menurut standar TIPHON. Algoritma *change detection* juga berhasil mengurangi estimasi penggunaan kuota internet bulanan hingga 82,4 MB. Sistem yang dikembangkan ini terbukti efektif dan efisien dalam memantau kondisi lingkungan tanaman dengan tingkat keandalan yang tinggi, sekaligus mampu menghemat penggunaan kuota internet.

Kata Kunci: *Greenhouse, Master-Slave, Change Detection, Packet Loss, Penggunaan Kuota Internet.*

Abstract

Greenhouse is a building specifically designed to protect plants from unfavorable external environmental conditions and regulate the internal environment to create ideal conditions that optimally support plant growth. However, conventional monitoring systems in *Greenhouse* face issues such as high internet data usage and packet loss due to reliance on WiFi access points (AP). This research developed a *Greenhouse* plant monitoring system using a master-slave architecture with the ESP-NOW protocol to address packet loss. Additionally, a change detection algorithm was applied on the master node to reduce redundant data transmission and minimize the size of transmitted data packets. The test results showed that the packet loss rate reached 1.43% of the total 222,793 data transmissions over three days, which falls into the "very good" category according to the TIPHON standard. The change detection algorithm also successfully reduced the estimated monthly internet data usage by up to 82.4 MB. The developed system proved to be effective and efficient in monitoring plant environmental conditions with high reliability while significantly saving internet data usage.

Keywords: *Greenhouse, Master-Slave, Change Detection, Packet Loss, Internet Data Usage.*

Djtechno: Jurnal Teknologi Informasi oleh Universitas Dharmawangsa Artikel ini bersifat open access yang didistribusikan di bawah syarat dan ketentuan dengan Lisensi Internasional Creative Commons Attribution NonCommercial ShareAlike 4.0 ([CC-BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/)).



Corresponding Author:

E-mail : eristian@siskom.untan.ac.id

1. PENDAHULUAN

Greenhouse adalah struktur bangunan yang dirancang secara khusus untuk melindungi tanaman dari kondisi lingkungan eksternal yang tidak menguntungkan dan untuk mengatur lingkungan internal, sehingga menciptakan keadaan yang optimal dan mendekati kondisi ideal bagi pertumbuhan dan pemeliharaan tanaman (Jonet dkk., 2024) (Ramadhan dkk., 2021). *Greenhouse* dapat menciptakan kondisi yang terkontrol, memungkinkan berbagai jenis tanaman tumbuh dengan baik meskipun kondisi di luar *Greenhouse* tidak mendukung (Zain dkk., 2024). Ini termasuk pengaturan suhu udara, kelembapan udara, dan kelembapan tanah untuk memenuhi kebutuhan spesifik tanaman, yang pada akhirnya dapat meningkatkan hasil dan kualitas tanaman (Ardiansyah dkk., 2024).

Dalam perkembangan teknologi pertanian, konsep *Internet of Things* (IoT) telah menjadi elemen kunci dalam mendukung fungsi *Greenhouse*. Dengan integrasi IoT, sistem pemantauan dan pengendalian lingkungan di dalam *Greenhouse* dapat dilakukan secara otomatis dan *real-time* (Kasliono dkk., 2024). Sensor-sensor yang terhubung melalui IoT berperan dalam mengumpulkan data terkait suhu udara, kelembapan udara, dan kelembapan tanah secara kontinu. Data yang diperoleh kemudian dianalisis untuk menilai kondisi tanaman dan lingkungan di dalam *Greenhouse* secara menyeluruh, sehingga memungkinkan petani untuk mengambil keputusan yang berdasarkan pada informasi yang akurat dan terkini (Nawangsari, 2024).

Meskipun integrasi IoT dalam sistem *monitoring* tanaman di *Greenhouse* menawarkan berbagai keuntungan, terdapat tantangan yang perlu diatasi untuk

memastikan efisiensi dan keandalan sistem. Salah satu tantangan utama adalah tingginya penggunaan kuota internet yang diperlukan untuk mentransmisikan data secara terus-menerus dari sensor-sensor di lapangan ke *cloud server* jika menggunakan banyak sensor dan node. Dalam sistem *monitoring* tanaman di *Greenhouse*, sensor *soil moisture* dan DHT11 digunakan untuk mengukur parameter penting seperti kelembapan tanah, suhu udara, dan kelembapan udara (Arafat dkk., 2021). Data dari sensor-sensor ini dikirimkan secara berkala dari setiap node ke *cloud server* melalui internet. Ketika jumlah node yang beroperasi dalam sistem meningkat, jumlah data yang dikirimkan juga meningkat secara signifikan, yang berpotensi menyebabkan penggunaan kuota internet yang sangat besar. Selain itu, sistem ini sering kali bergantung pada perangkat *access point* (AP) untuk mengirimkan data sensor dari banyak node ke *cloud server*. Ketergantungan pada perangkat *access point* untuk pengiriman data ke *cloud server* menimbulkan risiko terjadinya *packet loss*, terutama ketika banyak node aktif mengirimkan data secara bersamaan. *Packet loss* ini dapat menyebabkan hilangnya sebagian data yang dikirimkan, yang pada gilirannya dapat mengurangi keakuratan data yang diterima oleh *cloud server* (Liyaka, 2022). Ketidakakuratan data ini dapat berdampak negatif pada analisis kondisi tanaman dan lingkungan *Greenhouse* sehingga berpotensi mengarah pada pengambilan keputusan yang tidak optimal oleh petani.

Untuk mengatasi tantangan dalam penggunaan kuota internet dan risiko *packet loss* pada sistem *monitoring* tanaman di *Greenhouse* berbasis IoT, penelitian ini menerapkan algoritma *change detection*. Algoritma ini dirancang khusus untuk mengidentifikasi perubahan pada data sensor dan memberikan tanggapan terhadap adanya perubahan tersebut (Chughtai dkk., 2021). Dengan pendekatan ini, volume dan ukuran paket data yang dikirimkan dapat dikurangi sehingga menghemat penggunaan kuota internet tanpa mengurangi keakuratan informasi yang diperlukan. Selain itu, implementasi arsitektur *master-slave* menggunakan protokol ESP-NOW menawarkan solusi tambahan untuk efisiensi pengiriman data. Dalam konfigurasi ini, data sensor dari node *slave* dikirimkan ke node *master* menggunakan protokol ESP-NOW, yang

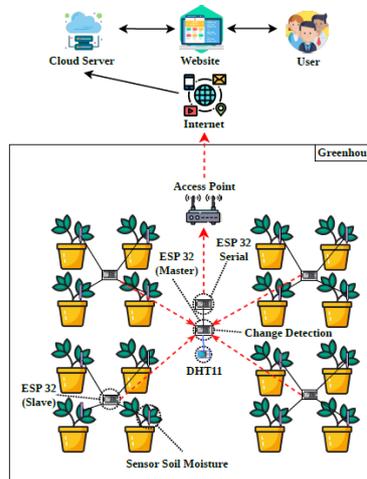
tidak memerlukan koneksi Wi-Fi melalui *access point* (Silva dkk., 2021). Hal ini tidak hanya mengurangi ketergantungan pada *access point*, tetapi juga mengurangi risiko *packet loss*. Dengan demikian, seluruh data yang terkumpul di node *master* dapat dikirimkan ke *cloud server* melalui satu titik saja, yakni node *master*. Pendekatan ini memastikan bahwa komunikasi antar node berlangsung secara efisien dan reliabel, sementara penggunaan sumber daya jaringan diminimalkan.

Beberapa penelitian terdahulu telah membahas arsitektur *master-slave*, algoritma *change detection*, dan sistem *monitoring* tanaman. Penelitian mengenai arsitektur *master-slave* untuk pengendalian pencahayaan LED menunjukkan penghematan energi hingga 90% (Lee dkk., 2022). Penelitian tentang algoritma *change detection* mengintegrasikan *machine learning* dan *change detection* 3D untuk mendeteksi perubahan tanah dan mencari sumber daya air dengan akurasi 88% (Almalki & Angelides, 2022). Selain itu, penelitian mengenai sistem *monitoring* tanaman yang mengembangkan sistem IoT untuk memantau suhu, kelembapan tanah, dan pH pada tanaman anggur, dengan waktu rata-rata pompa air alkali 28 detik dan respon pompa penyiraman 3,54 detik (Ruslianto dkk., 2022). Penelitian lain tentang pertanian cerdas berbasis IoT untuk memantau lahan pertanian secara *real-time* melalui antarmuka web dan melakukan penyiraman otomatis saat tanah menunjukkan kekeringan (Dwiyatno dkk., 2022).

2. METODE PENELITIAN

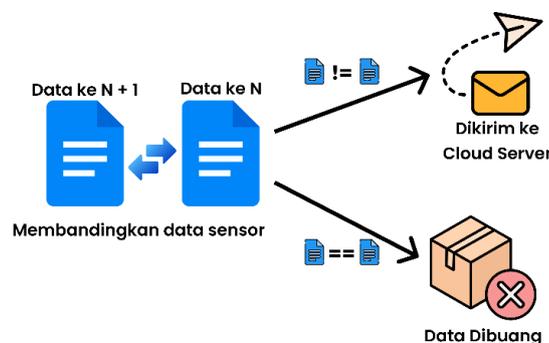
Penelitian ini dilakukan di Lab IoT Smart Farming Greenhouse FMIPA Untan, di mana tanaman anggur ditanam di dalam pot. Sistem *monitoring* yang dikembangkan menggunakan arsitektur *master-slave* untuk memungkinkan pengumpulan data secara efisien dari beberapa node, serta menerapkan algoritma *change detection* guna mengurangi pengiriman data yang tidak perlu. Gambar 1 memberikan gambaran sistem keseluruhan *monitoring* tanaman berbasis arsitektur *master-slave* yang menggunakan algoritma *change detection*. Dalam konfigurasi ini, node *master* bertanggung jawab membaca suhu dan kelembapan udara di *Greenhouse*, serta mengumpulkan data kelembapan tanah dari node-node *slave*, yang kemudian difilter

menggunakan algoritma *change detection*. Setiap node *slave* dilengkapi dengan empat sensor kelembapan tanah (*soil moisture*), di mana setiap sensor ditempatkan pada pot tanaman yang berbeda.



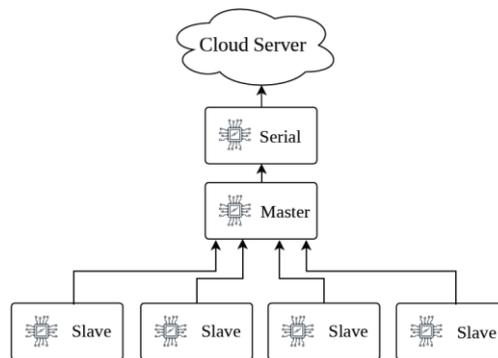
Gambar 1. Deskripsi Sistem Monitoring Tanaman

Setelah data kelembapan tanah terkumpul di node *master* dan difilter menggunakan algoritma *change detection*, hanya data yang mengalami perubahan yang akan dikirimkan ke *cloud server*. Gambar 2 menunjukkan perancangan algoritma *change detection* yang diterapkan pada node *master*. Algoritma ini bekerja dengan membandingkan dua set data dalam rentang waktu tertentu. Jika terdeteksi perubahan pada data sensor dibandingkan dengan data sebelumnya, maka data tersebut akan dikirimkan ke *cloud server*. Sebaliknya, jika tidak ada perubahan yang terdeteksi, data tidak akan dikirimkan, dan sensor akan terus melakukan pembacaan ulang hingga perubahan terdeteksi.

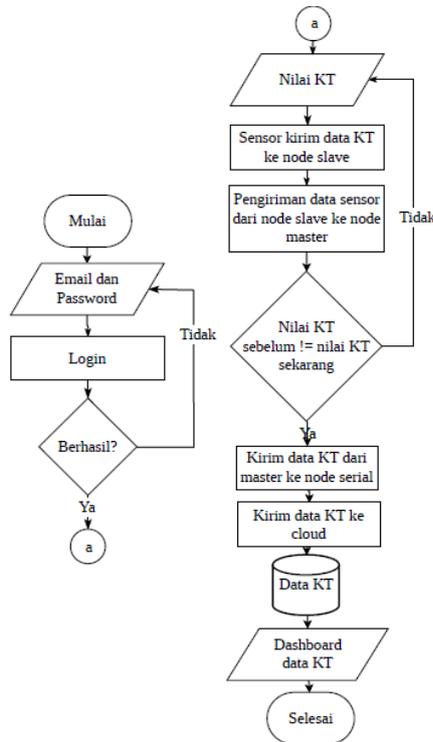


Gambar 2. Perancangan Algoritma *Change Detection*

Setelah data difilter, data tersebut diteruskan ke node serial melalui komunikasi kabel RX/TX. Node serial ini berfungsi sebagai penghubung antara node *master* dan *cloud server*, karena modul Wi-Fi pada node *master* sudah sepenuhnya digunakan untuk komunikasi dengan node-node *slave* melalui protokol ESP-NOW. Gambar 3 memperlihatkan pola arsitektur *master-slave* serta peran node serial dalam pengiriman data ke *cloud server*. Sedangkan untuk diagram alir sistem pembacaan sensor kelembapan tanah pada node *slave* dapat dilihat pada Gambar 4.

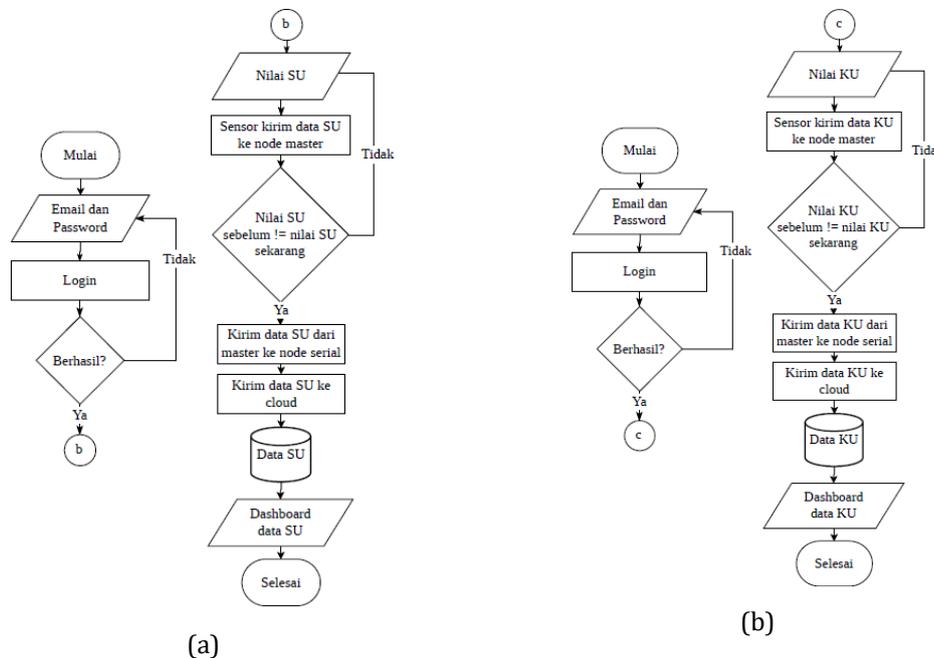


Gambar 3. Pola Arsitektur *Master Slave*



Gambar 4. Diagram Alir Pembacaan Sensor Kelembapan Tanah (KT) pada Node Slave

Pada Gambar 4, proses dimulai dengan pengguna melakukan *login* ke *website*. Setelah *login*, sensor kelembapan tanah (KT) membaca dan data sensor dikirim ke node *slave*, selanjutnya data dari node *slave* dikirim ke node *master*. Di node *master*, data difilter menggunakan algoritma *change detection* untuk mendeteksi perubahan data sensor dibandingkan dengan data yang sudah dikirim sebelumnya. Data yang berubah kemudian diteruskan ke node serial untuk dikirim ke *cloud server*, di mana data disimpan dalam database dan ditampilkan di *website* untuk pemantauan *real-time*. Adapun diagram alir pembacaan sensor DHT11 untuk mengukur suhu (SU) dan kelembapan udara (KU) pada node *master* dapat dilihat pada Gambar 5.



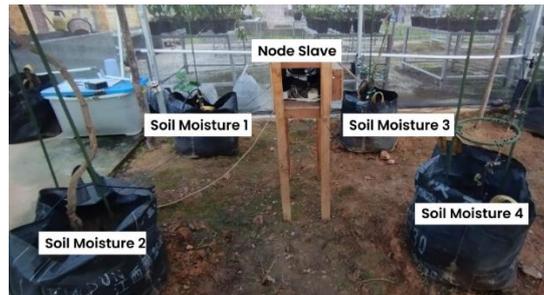
Gambar 5. Diagram Alir Sistem Pembacaan Sensor Suhu Udara (SU) (a) dan Kelembapan Udara (KU) (b) pada Node Master

Pada Gambar 5, setelah pengguna *login*, sensor DHT11 pada node *master* mengukur suhu udara (SU) dan kelembapan udara (KU). Data yang diperoleh kemudian difilter menggunakan algoritma *change detection* untuk mendeteksi perubahan dibandingkan dengan data sebelumnya. Data yang berubah diteruskan ke node serial untuk dikirim ke *cloud server* dan ditampilkan pada *website* untuk pemantauan secara *real-time*.

3. HASIL DAN PEMBAHASAN

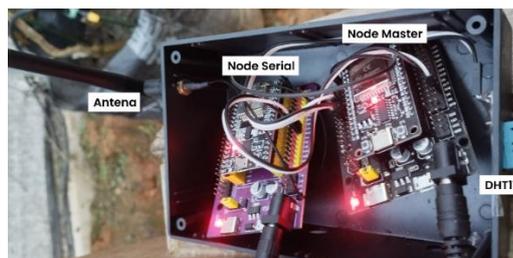
A. Hasil Implementasi

Pada sistem *monitoring* ini, implementasi dilakukan pada node *slave* dan node *master*. Node *slave* dilengkapi dengan empat sensor *soil moisture* yang dihubungkan ke ESP32 untuk mengukur tingkat kelembapan tanah. Setelah sensor-sensor tersebut ditancapkan ke dalam tanah, sistem akan secara otomatis mulai membaca data kelembapan tanah. Implementasi sistem pada node *slave* dapat dilihat pada Gambar 6.



Gambar 6. Implementasi Sistem pada Node *Slave*

Pada node *master*, sensor DHT11 digunakan untuk mengukur suhu udara dan kelembapan udara di dalam *Greenhouse*. Selain itu, node *master* dilengkapi dengan node serial yang berfungsi sebagai penghubung dalam pengiriman data sensor yang telah terkumpul. Hal ini diperlukan karena, ketika node *master* menerima data sensor dari node *slave* melalui protokol ESP-NOW, data tidak dapat dikirim langsung ke *cloud server*. Oleh karena itu, node serial digunakan untuk mengatasi keterbatasan ini dengan menghubungkan node *master* melalui kabel pada pin RX dan TX. Implementasi sistem pada node *master* ditampilkan pada Gambar 7.



Gambar 7. Implementasi Sistem pada Node *Master*

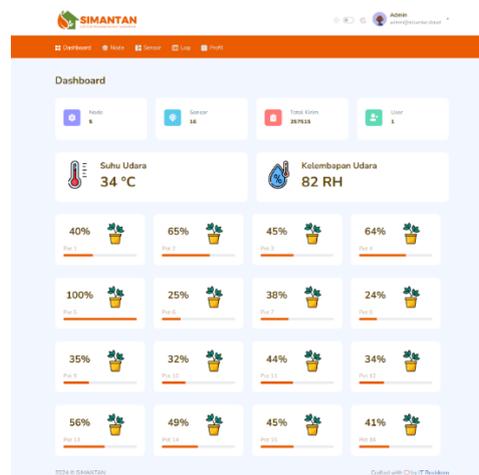
Pada implementasi sistem komunikasi antara node *master* dan node *slave*, data dikirim dari node *slave* dan diterima oleh node *master* menggunakan protokol ESP-NOW. Protokol ini memungkinkan komunikasi langsung antar perangkat ESP tanpa memerlukan jaringan Wi-Fi atau koneksi internet. Setiap node *slave* harus mendaftarkan alamat MAC dari node *master* untuk menjamin komunikasi yang tepat. Setelah pendaftaran, node *slave* dapat mengirimkan data sensor secara bergiliran

dengan node *slave* lainnya ke node master secara nirkabel. Setelah semua data terkumpul, data akan difilter menggunakan algoritma *change detection*. Jika algoritma ini mendeteksi perubahan data sensor dibandingkan dengan data sebelumnya, data data tersebut akan diteruskan ke node serial yang terhubung ke Wi-Fi melalui access point, sehingga node serial dapat mengirimkan data ke *cloud server*, jika salah satu data sensor tidak mengalami perubahan maka data sensor tersebut akan dibuang. Implementasi sistem komunikasi ini antara node *master* dan node *slave* menggunakan algoritma *change detection* ditunjukkan pada Gambar 8.



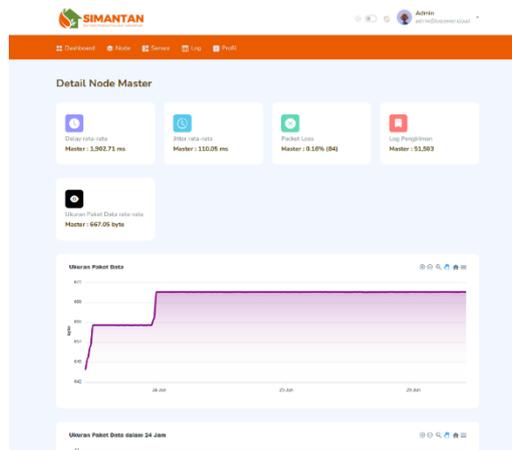
Gambar 8. Implementasi Sistem Komunikasi Antara Node *Slave* dan Node *Master* Menggunakan Algoritma *Change Detection*

Aplikasi berbasis web dirancang untuk mendukung pemantauan *Greenhouse* tanaman, dengan tujuan menciptakan lingkungan optimal bagi pertumbuhan tanaman. Aplikasi ini menyajikan data kelembapan tanah secara *real-time* untuk setiap pot tanaman anggur, serta data suhu dan kelembapan udara dalam *Greenhouse*. Tampilan *dashboard* web sistem *monitoring* tanaman ditunjukkan pada Gambar 9.



Gambar 9. Tampilan *Dashboard* Web Sistem *Monitoring* Tanaman

Selain itu, aplikasi juga memantau parameter sistem seperti *packet loss* dan ukuran paket data yang dikirim dalam sistem *monitoring* tanaman. Pemantauan ini memastikan bahwa data yang dikirimkan oleh sensor diterima dengan akurat dan tepat waktu, sehingga mendukung pengambilan keputusan yang lebih baik dalam pengelolaan *Greenhouse*. Tampilan halaman *log* setiap node untuk memeriksa *packet loss* dan ukuran paket data dapat dilihat pada Gambar 10.



Gambar 10. Tampilan Halaman *Log* Setiap Node

B. Pengujian Sistem

Dalam skema pengujian ini, node *slave* mengirimkan data sensor kelembapan tanah ke node *master* dengan interval 250 milisekon. Node *master* mengambil data dari node *slave* setiap 2000 milisekon. Pengujian berlangsung selama 3 hari. Pengukuran ukuran paket data dilakukan melalui sistem *monitoring* di node *master*. Ukuran paket data yang diukur meliputi dua skenario: tanpa algoritma *change detection* dan menggunakan algoritma *change detection*. Data yang dihasilkan diubah ke dalam bentuk JSON dalam satu *batch* untuk diukur ukuran paketnya, lalu dikirimkan beserta informasi ukuran paket dan urutan pengiriman untuk parameter pengukuran *packet loss*. Grafik hasil pengujian ukuran paket data di node *master* tanpa algoritma *change detection* setiap 24 jam (1 hari) selama 3 hari dapat dilihat pada Gambar 11.



Gambar 11. Grafik Pengujian Ukuran Paket Data Tanpa Algoritma *Change Detection*

Pada Gambar 11, ukuran paket data pada hari pertama tercatat sebesar 11,23 MB, sedangkan pada hari kedua sebesar 10,91 MB, dan pada hari ketiga sebesar 10,51 MB. Total ukuran paket data selama 3 hari mencapai 32,65 MB, dengan jumlah data pengiriman oleh node *slave* dan node *master* mencapai 256.610 data yang diterima. Jika diasumsikan pengumpulan data berlangsung selama 30 hari (1 bulan), total ukuran paket data diperkirakan mencapai 326,5 MB, yang menunjukkan penggunaan kuota internet sebesar 326,5 MB per bulan. Penggunaan ini tergolong cukup besar dan masih dapat dioptimalkan menggunakan algoritma *change detection*. Grafik hasil pengujian ukuran paket data di node *master* menggunakan algoritma *change detection* setiap 24 jam selama 3 hari dapat dilihat pada Gambar 12.



Gambar 12. Grafik Pengujian Ukuran Paket Data Menggunakan Algoritma *Change Detection*

Pada Gambar 12, ukuran paket data pada hari pertama tercatat sebesar 7,65 MB, sedangkan pada hari kedua sebesar 8,25 MB, dan pada hari ketiga sebesar 8,51 MB. Total ukuran paket data selama 3 hari mencapai 24,41 MB, dengan jumlah pengiriman data oleh node *slave* dan node *master* mencapai 222.793 data yang diterima. Jika diasumsikan pengumpulan data berlangsung selama 30 hari (1 bulan), total ukuran paket data diperkirakan mencapai 244,1 MB, yang menunjukkan penggunaan kuota

internet sebesar 244,1 MB per bulan. Penggunaan algoritma *change detection* ini dibandingkan dengan tanpa algoritma *change detection* mampu menghemat kuota internet sebesar 82,4 MB.

Pada pengujian *packet loss* menggunakan arsitektur *master-slave* dengan algoritma *change detection* selama 3 hari pada node *master*, dari total 55.561 data yang diterima, terdapat 802 data yang mengalami *packet loss*, atau sekitar 1,42%. Grafik hasil pengujian jumlah data yang dikirim dan *packet loss* pada node *master* setiap setengah jam dapat dilihat pada Gambar 13.



Gambar 13. Grafik Pengujian Jumlah Pengiriman dan *Packet Loss* pada Node *Master*

Pada pengujian *packet loss* pada node *slave* pertama, dari total 38.866 data yang diterima, terdapat 585 data yang mengalami *packet loss*, atau sekitar 1,48%. Grafik hasil pengujian jumlah data yang dikirim dan *packet loss* pada node *slave* pertama setiap setengah jam dapat dilihat pada Gambar 14.



Gambar 14. Grafik Pengujian Jumlah Pengiriman dan *Packet Loss* pada Node *Slave* Pertama

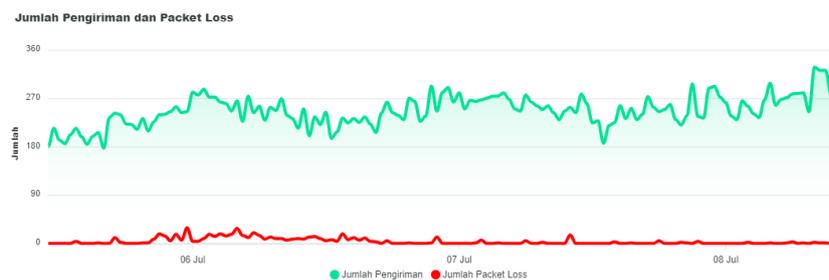
Pada pengujian *packet loss* pada node *slave* kedua, dari total 52.201 data yang diterima, terdapat 664 data yang mengalami *packet loss*, atau sekitar 1,26%. Grafik hasil pengujian jumlah data yang dikirim dan *packet loss* pada node *slave* kedua setiap setengah jam dapat dilihat pada Gambar 15.



Gambar 15. Grafik Pengujian Jumlah Pengiriman dan *Packet Loss* pada Node *Slave* Kedua
Pada pengujian *packet loss* pada node *slave* ketiga, dari total 40.935 data yang diterima, terdapat 589 data yang mengalami *packet loss*, atau sekitar 1,42%. Grafik hasil pengujian jumlah data yang dikirim dan *packet loss* pada node *slave* ketiga setiap setengah jam dapat dilihat pada Gambar 16.



Gambar 16. Grafik Pengujian Jumlah Pengiriman dan *Packet Loss* pada Node *Slave* Ketiga
Pada pengujian *packet loss* pada node *slave* keempat, dari total 35.230 data yang diterima, terdapat 560 data yang mengalami *packet loss*, atau sekitar 1,56%. Grafik hasil pengujian jumlah data yang dikirim dan *packet loss* pada node *slave* keempat setiap setengah jam dapat dilihat pada Gambar 17.



Pada pengujian keseluruhan node, total rata-rata *packet loss* adalah 1,43%, yang berarti sebanyak 3.200 data dari keseluruhan 222.793 data yang diterima mengalami *packet loss*. Analisis hasil pengujian *packet loss* dilakukan berdasarkan kategori

performansi jaringan sesuai dengan standar TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) untuk parameter *packet loss*, sebagaimana yang diuraikan dalam Tabel 1 (Rachman dkk., 2023).

Tabel 1. Performansi *Packet Loss* Jaringan Standar TIPHON

<i>Packet Loss</i> (%)	Indeks	Kategori
$0 \leq p < 3$	4	Sangat Bagus
$3 \leq p < 15$	3	Bagus
$15 \leq p \leq 25$	2	Sedang
> 25	1	Jelek

Berdasarkan Tabel 1, kategori performansi *packet loss* berdasarkan standar TIPHON menunjukkan bahwa *packet loss* yang terjadi pada node *master*, node *slave* pertama, node *slave* kedua, node *slave* ketiga, node *slave* keempat, serta *packet loss* keseluruhan termasuk dalam kategori "sangat bagus" karena semuanya berada di bawah 3%. Tingkat *packet loss* yang rendah ini menandakan bahwa sistem *monitoring* berbasis *master-slave* menggunakan algoritma *change detection* memiliki keandalan yang tinggi dalam pengiriman data dari node *slave* ke node *master* karena tidak bergantung pada WiFi *access point* untuk pengiriman data sensor dari node *slave* ke node *master*.

4. SIMPULAN

Penelitian ini menunjukkan bahwa sistem *monitoring* tanaman di *Greenhouse* efektif dalam memantau kondisi lingkungan dengan menggunakan arsitektur *master-slave* dan protokol ESP-NOW. Sistem ini berhasil meminimalkan risiko *packet loss* dan menghemat penggunaan kuota internet melalui penerapan algoritma *change detection*. Hasil pengujian menunjukkan bahwa *packet loss* keseluruhan mencapai 1,43% dari total 222.793 pengiriman data selama 3 hari, yang termasuk kategori "sangat bagus" menurut standar TIPHON. Tingkat *packet loss* yang rendah ini mengindikasikan keandalan sistem dalam transmisi data untuk pemantauan *real-time*. Selain itu, penerapan algoritma *change detection* berkontribusi pada pengurangan estimasi kuota internet bulanan sebesar 82,4 MB.

PUSTAKA

- Almalki, F. A., & Angelides, M. C. (2022). Autonomous flying IoT: A synergy of machine learning, digital elevation, and 3D structure change detection. *Computer Communications*, 190, 154–165. <https://doi.org/10.1016/j.comcom.2022.03.022>
- Arafat, Ratna, S., Wagino, & Ibrahim. (2021). Perancangan dan Pengujian Alat untuk Monitoring KelembabanTanah dan Pemberian Pupuk Cair pada Tanaman Cabai Berbasis Internet of Things. *Technologia*, 12(4), 286–291. <https://doi.org/10.31602/tji.v12i4.5639>
- Ardiansyah, Pratama, A. H., Novita, I., Nurlian, S., Sam, S. R. A., Amalia, K. P., Iftitah, A., Hafid, H. A., & Mubarak, H. (2024). Pengembangan Greenhouse dalam Mendukung Pertanian Berkelanjutan. *Jurnal Pengabdian Masyarakat Hasanuddin*, 5(1), 13–20. <https://doi.org/10.31947/jpmh.v5i1.34136>
- Chughtai, A. H., Abbasi, H., & Karas, I. R. (2021). A review on change detection method and accuracy assessment for land use land cover. *Remote Sensing Applications: Society and Environment*, 22, 1–13. <https://doi.org/10.1016/j.rsase.2021.100482>
- Dwiyatno, S., Krisnaningsih, E., Hidayat, D. R., & Sulistiyono. (2022). Smart Agriculture Monitoring Penyiraman Tanaman Berbasis Internet of Things. *Jurnal PROSISKO*, 9, 38–43.
- Jonet, R. V., Fevria, R., Violita, Handayani, D., & Arjulis, W. (2024). Perbandingan Pertumbuhan Tanaman Selada Hijau (*Lactuca Sativa* L.) di Dalam Dan Di Luar Greenhouse Yang Dibudidayakan Secara Hidroponik (Studi Kasus We Farm Hidroponik). *Jurnal Pendidikan Tambusai*, 8(4), 17941–17950. <https://jptam.org/index.php/jptam/article/view/14897/11402>
- Kasliono, Ruslianto, I., & Erniajan, Y. (2024). Penerapan Model Waterfall dalam Pengembangan Perangkat Lunak Pemantauan Tanaman Anggur Berbasis Mobile Menggunakan IoT. *Journal of Computer System and Informatics (JoSYC)*, 5(3), 526–534. <https://doi.org/10.47065/josyc.v5i3.5099>
- Lee, C.-T., Chen, L.-B., Chu, H.-M., Hsieh, C.-J., & Liang, W.-C. (2022). An Internet of Things (IoT)-Based Master-Slave Regionalized Intelligent LED-Light-Controlling System. *Applied Sciences (Switzerland)*, 12(1), 1–23. <https://doi.org/10.3390/app12010420>
- Liyaka, K. K. S. (2022). Detection of Malicious Nodes in IoT Networks based on Packet Loss using ML. *STM Journals*, 9(3), 9–17. <https://doi.org/10.37591/JoMCCMN>
- Nawang Sari, Y. R. (2024). Penerapan Teknologi Cerdas dalam Pengelolaan Tanaman untuk Meningkatkan Efisiensi Sumber Daya dan Hasil Pertanian. *Jurnal Literasi Indonesia (JLI)*, 1(2), 69–76. <https://jli.staiku.ac.id/index.php/st/index>
- Rachman, D. A., Muhyidin, Y., & Sunandar, M. A. (2023). Analysis Quality of Service of Internet Network Fiber to The Home Service PT. XYZ Using Wireshark. *Jurnal Informatika dan Teknik Elektro Terapan*, 11(3), 997–1006. <https://doi.org/10.23960/jitet.v11i3%20s1.3436>
- Ramadhan, A. D., Astufik, R. P., & Surya, Y. A. (2021). Sistem Kontrol dan Monitoring Greenhouse Hidroponik pada Tanaman Sawi Berbasis Aplikasi App Inventor. *SinarFe7*, 4(1), 23–28. <https://journal.fortei7.org/index.php/sinarFe7/article/view/7>
- Ruslianto, I., Ristian, U., & Hasfani, H. (2022). Sistem Pintar Untuk Anggur (Sipunggur) pada Kawasan Tropis Berbasis Internet of Things (IoT). *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 8(1), 121–127.
- Silva, S. A. V. e, Garcia, A. T., Kelbouscas, A. L. da S., Coch, V. B., & Oliveira, V. M. de. (2021). Network of Embedded Systems for Motor Rehabilitation Post-Stroke of Upper Limbs using ESP8266 and ESPNOW protocol. *Proceedings do XV Simpósio Brasileiro de Automação Inteligente*. <https://doi.org/10.20906/sbai.v1i1.2813>
- Zain, A. M., Hidayatul W, M. A., & Fathoni, K. (2024). Pemanfaatan Greenhouse Sebagai Potensi Desa Meningkatkan Produktivitas Pertanian Dan Kesejahteraan Masyarakat(Studi Kasus Desa Kedung Banteng Sukorejo Ponorogo). *Social Science Academic*, 591–600. <https://doi.org/10.37680/ssa.v0i0.5945>