

Implementation of Failover in a Master-Slave Architecture for Soil Moisture Sensor Data Transmission

Ajeng Fitria^{1*}, Uray Ristian¹ , Syamsul Bahri¹ , Ida Bagus Gede Sarasvananda² 

¹Universitas Tanjungpura, Indonesia.

²Universitas Udayana, Indonesia

* Corresponding Author. E-mail: h1051201031@student.untan.ac.id

Article History

Received:

Dec 12th, 2025

Revised:

Jul 31th, 2026

Accepted:

Aug 01st, 2026

Keywords

IoT,

Failover,

Master-Slave,

Packet loss,

Plant Monitoring

System,

Recovery Time.

ABSTRACT

Development of Internet of Things (IoT) offers significant opportunities in the agricultural sector by enabling automatic and efficient monitoring of crop conditions. A common issue in IoT-based monitoring systems utilizing a master-slave architecture is the loss of data communication when the master node experiences a failure, which reduces system reliability. This research develops a plant monitoring system based on a master-slave architecture with a failover method using ESP32 and HTTP communication. The system consists of three master nodes, each handling two slave nodes, with a total of twelve soil moisture sensors. Data from the sensors are transmitted by the slave nodes to the master nodes, then forwarded to a cloud server. The failover method functions to maintain communication continuity by rerouting data transmission when a master node experiences a failure. The test results show that the system without failover experienced data loss and communication downtime equal to the duration of the master node failure, while the integration of IoT and failover method maintained communication with an average packet loss of 2.22% and a switching delay of approximately 0.78 seconds.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



PENDAHULUAN

Tanaman anggur tumbuh dengan kelembapan tanah ideal 60-75%[1][2]. Pemantauan kelembapan tanah menggunakan teknologi dan internet agar data sensor dapat dipantau secara akurat dan efisien[3][4]. Pemantauan tanaman melibatkan transmisi data melalui jaringan internet menggunakan arsitektur *master-slave*, node *slave* mengirimkan data ke node *master* kemudian diteruskan ke *cloud server*[5]. Selama proses tersebut, *packet loss* dapat terjadi, sehingga metode *failover* diterapkan untuk menjaga sistem tetap beroperasi meski node *master* gagal[6]. *Packet loss* dipengaruhi kualitas transmisi dan kestabilan sinyal[7]. *Failover* memungkinkan pengalihan komunikasi ke node *master* lain yang aktif, dengan pemilihan *master* berdasarkan prioritas atau ketersediaan sumber daya, sehingga sistem tetap berjalan meski terjadi kegagalan[6].

Penelitian sebelumnya berjudul “Pengaplikasian *Internet of Things* Untuk Monitoring Lingkungan Lahan Tanaman Anggur”[8] membahas IoT pada pemantauan lahan tanaman dengan sensor DHT11, LDR, dan YL-69 dengan modul ESP32, menghasilkan akurasi cahaya 72,85%, suhu 93,75%, dan kelembapan tanah 87,25%. Penelitian “*Downtime Prevention Using Failover on IoT-based Contactless Temperature Checkers*”[9] membahas sistem *failover* pada broker MQTT, mencapai tingkat ketersediaan 99,96% dengan downtime rata-rata 19,0972 detik. Selanjutnya, penelitian “Implementasi Pengiriman Data Multi-Node Sensor Menggunakan Metode *Master-Slave*

pada Komunikasi LoRa"[5] menunjukkan metode *master-slave* mengurangi tabrakan data, dengan *packet loss* terkecil 0% pada jarak 50 meter dan tertinggi 8,99% pada 200 meter. Semakin jauh jarak pengiriman, semakin besar *packet loss* yang terjadi [10], [11].

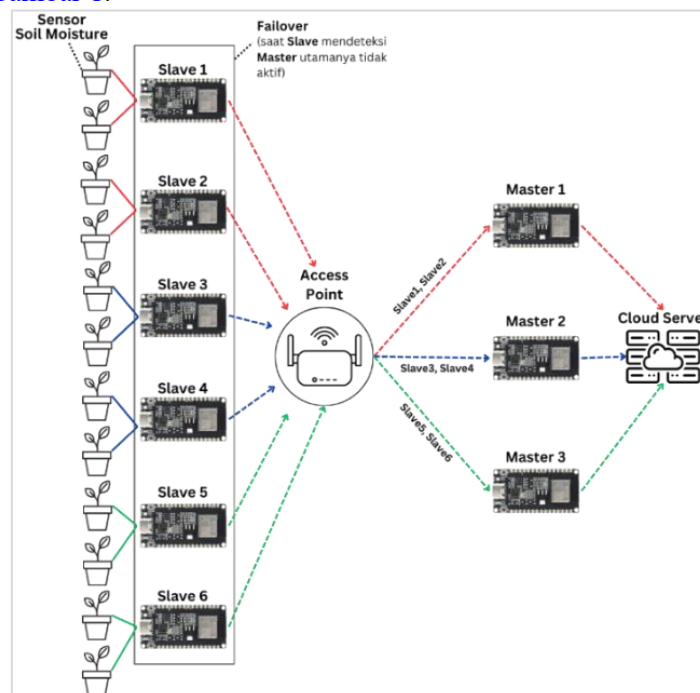
Berdasarkan penelitian sebelumnya, penelitian ini menerapkan metode *failover* pada arsitektur *master-slave* dengan tujuan menjaga pengiriman data dari node *slave* ke node *master*. Sistem dirancang agar ketika node *master* yang bertanggung jawab mengalami kegagalan, data secara otomatis dialihkan ke *master* lain yang masih aktif, sehingga proses pengiriman tetap optimal dan meminimalkan kehilangan data. Fokus penelitian mencakup analisis pengaruh penerapan *failover* terhadap tingkat *packet loss* dan durasi jeda pengiriman, serta evaluasi perbandingan kinerja sistem dalam kondisi dengan *failover* dan tanpa *failover*. Penelitian ini menunjukkan bagaimana metode *failover* dapat meningkatkan keandalan komunikasi dan kestabilan transmisi data sensor kelembapan tanah dalam berbagai skenario operasional.

METODE

Perancangan Sistem

Analisis kebutuhan menentukan perangkat keras dan perangkat lunak yang diperlukan, mencakup perangkat keras berupa ESP32, Expansion Board ESP32, sensor *soil moisture*, dan adaptor 12V 1A, serta perangkat lunak berupa Arduino IDE, Visual Studio Code, dan Laravel.

Penelitian ini mengembangkan sistem pemantauan kelembapan tanah berbasis arsitektur master-slave dengan penerapan metode failover untuk meningkatkan keandalan komunikasi. Sistem dirancang agar proses pemantauan tetap efektif dan berkelanjutan meskipun terjadi kegagalan pada salah satu atau lebih node master. *Node slave* secara periodik mengumpulkan data dari sensor soil moisture dan mengirimkannya ke node master. Apabila node master yang bertanggung jawab tidak aktif, pengiriman data secara otomatis dialihkan ke node master lain yang masih aktif. Data yang diterima kemudian diteruskan ke cloud server untuk penyimpanan dan analisis lebih lanjut sehingga proses pemantauan tanaman tetap konsisten. Perancangan sistem mencakup aspek perangkat keras dan perangkat lunak, dengan ESP32 sebagai pengendali utama dan sensor soil moisture sebagai komponen utama pengukur kelembapan. Node slave bertugas mengumpulkan dan mengirimkan data baik pada kondisi normal maupun saat terjadi *failover*. Skema keseluruhan arsitektur sistem ditunjukkan pada Gambar 1.

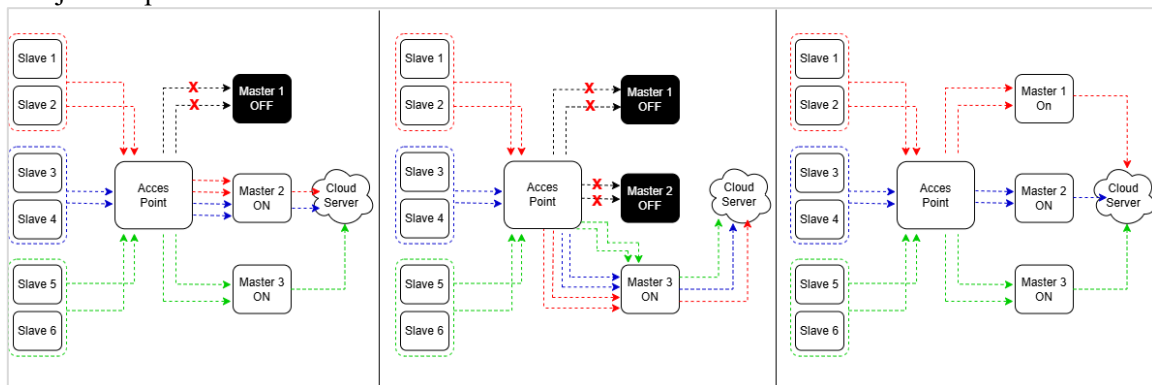


Gambar 1. Arsitektur Sistem Monitoring Tanaman

Perangkat keras terdiri atas enam node *slave* dan tiga node *master* berbasis ESP32. Setiap node *slave* menggunakan dua sensor *soil moisture*, sedangkan node *master* menerima data dari *slave* dan meneruskannya ke *cloud*. Pembagian tugas dilakukan dengan Master 1 mengelola Slave 1 dan Slave 2, Master 2 mengelola Slave 3 dan Slave 4, serta Master 3 mengelola Slave 5 dan Slave 6.

Komunikasi antara node *slave* dan node *master* menggunakan protokol HTTP melalui jaringan Wi-Fi dengan *Static IP* yang unik pada setiap ESP32. Node *slave* mengirimkan data secara langsung ke node *master* kemudian menunggu *response* selama lima detik. Apabila terjadi kegagalan (*time-out*), mekanisme *failover* akan aktif secara otomatis sehingga node *slave* beralih ke node *master* lain hingga data berhasil dikirimkan. Selanjutnya, komunikasi antara node *master* dan *cloud server* juga menggunakan protokol HTTP, di mana node *master* menerima data dari seluruh node *slave* kemudian meneruskannya ke *cloud* untuk penyimpanan dan pemantauan.

Metode *failover* dirancang untuk mengoptimalkan kelancaran transmisi data sensor dari node *slave* ke node *master*. Pada kondisi normal, setiap node *slave* mengirimkan data ke node *master* yang telah ditentukan sehingga mekanisme *failover* tidak aktif. Namun, ketika node *master* mengalami kegagalan atau tidak memberikan *response*, sistem secara otomatis mendeteksi kondisi tersebut dan mengalihkan jalur pengiriman beserta beban kerja ke node *master* lain yang masih aktif. Mekanisme ini memastikan aliran data sensor tetap berlangsung tanpa memerlukan intervensi manual. Proses pengalihan diuji menggunakan tiga skenario, yaitu **Skenario Semua Master Hidup** yang merepresentasikan kondisi normal, **Skenario Satu Master Mati** ketika seluruh node *slave* yang terhubung pada node *master* yang gagal dialihkan ke node *master* aktif lainnya, serta **Skenario Dua Master Mati** ketika seluruh node *slave* mengirimkan data ke satu-satunya node *master* yang masih aktif dan selanjutnya meneruskan data ke *cloud server*. Sistem hanya berhenti mengirimkan data apabila seluruh node *master* tidak aktif atau gagal memberikan *response*. Seluruh skenario *failover* ditunjukkan pada Gambar 2.



Gambar 2. Tiga Skenario Kondisi *Failover*

Basis data menggunakan MySQL untuk menyimpan data hasil pembacaan sensor kelembapan tanah, kinerja transmisi, serta kode unik yang mengidentifikasi sensor, node, dan pot tanaman. Selain data kelembapan tanah, basis data juga menyimpan parameter Quality of Service (QoS) yang meliputi delay, jitter, throughput, dan ukuran paket (*packet size*) untuk menganalisis efektivitas mekanisme *failover* serta performa jaringan. Seluruh peristiwa *failover* juga didokumentasikan pada basis data untuk kebutuhan analisis.

Antarmuka website dirancang secara sederhana agar proses pemantauan dapat dilakukan secara efisien. Website terdiri atas beberapa halaman utama, yaitu **Dashboard**, **Node Master**, **Node Activities**, **Log**, **Failover**, dan **Soil Moisture**, yang masing-masing berfungsi untuk menampilkan informasi kondisi sistem dan aktivitas komunikasi antar node.

Metode Pengujian

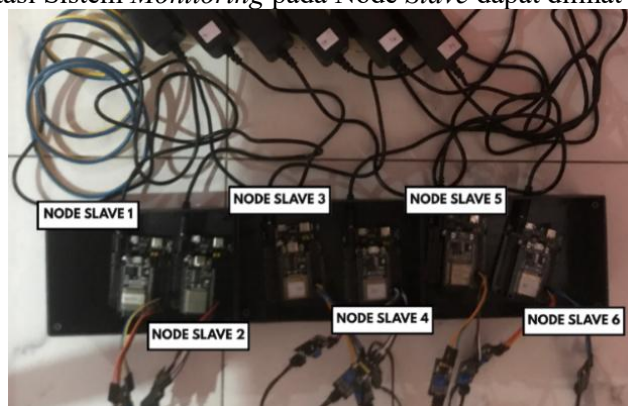
Pengujian sistem dilakukan menggunakan tiga pendekatan, yaitu pengujian tanpa *failover*, pengujian dengan *failover*, dan pengujian *black box*. Pengujian tanpa *failover* maupun dengan *failover* masing-masing dilaksanakan selama satu jam yang dibagi ke dalam lima fase berdurasi 12 menit, meliputi kondisi seluruh node *master* aktif, satu node *master* tidak aktif, dua node *master* tidak

aktif, hingga seluruh node *master* kembali aktif. Selama pengujian, setiap node *slave* mengirimkan data ke *cloud* setiap satu menit. Selain mengukur keberhasilan komunikasi data, pengujian *failover* juga mencatat riwayat terjadinya *failover* beserta parameter *Quality of Service* (QoS). Sementara itu, pengujian *black box* dilakukan untuk memverifikasi bahwa seluruh fitur dan fungsi pada website telah berjalan sesuai dengan spesifikasi tanpa meninjau proses internal sistem.

HASIL DAN PEMBAHASAN

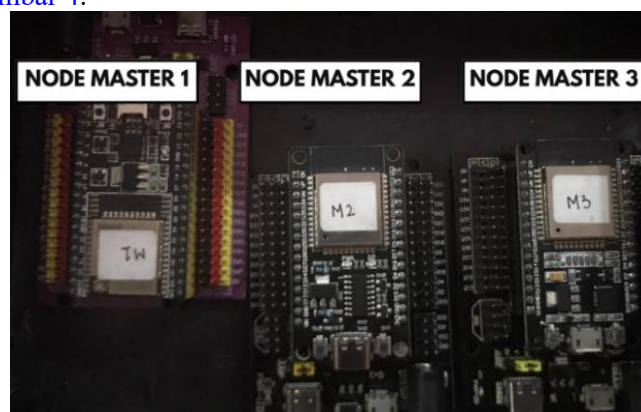
Hasil Implementasi

Pada sistem *monitoring*, implementasi pembacaan data dilakukan melalui enam node *slave* yang masing-masing terhubung dengan dua sensor *soil moisture*, sehingga total terdapat 12 sensor yang aktif memantau kondisi kelembapan tanah. Setiap sensor membaca nilai kelembapan secara langsung melalui probe yang ditancapkan ke media tanah, kemudian data tersebut diproses oleh modul ESP32 pada node *slave* dan dikirimkan ke node *master* sesuai komunikasi yang telah ditetapkan. Implementasi Sistem *Monitoring* pada Node *Slave* dapat dilihat pada [Gambar 3](#).



[Gambar 3](#). Implementasi Sistem *Monitoring* pada Node *Slave*

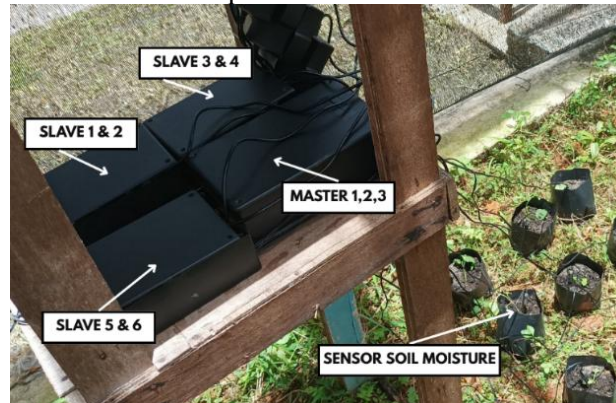
Pada sistem *monitoring* node *master*, node *master* mengoordinasikan aliran data antara node *slave* dan *cloud server* agar komunikasi berjalan teratur. Tiga unit ESP32 sebagai node *master* menerima data kelembapan dari node *slave* dan meneruskannya ke *cloud* melalui Wi-Fi. Pembagian tugas dilakukan secara terstruktur: *master* 1 menangani *slave* 1 dan 2, *master* 2 menangani *slave* 3 dan 4, dan *master* 3 menangani *slave* 5 dan 6, sehingga proses pengiriman data berlangsung rapi dan sesuai alur komunikasi yang telah ditetapkan. Implementasi Sistem *Monitoring* pada Node *Master* dapat dilihat pada [Gambar 4](#).



[Gambar 4](#). Implementasi Sistem *Monitoring* pada Node *Master*

Pada implementasi komunikasi antara node *master* dan node *slave*, transmisi data dilakukan melalui protokol HTTP dengan koneksi internet. Setiap perangkat dikonfigurasi menggunakan IP statis untuk menjaga identitas jaringan tetap dan memastikan data selalu terkirim ke node *master* yang benar. Penggunaan IP statis juga mendukung *failover*, karena menjadi acuan bagi node *slave* saat harus mengalihkan komunikasi ke *master* lain jika *master* utama yang bertanggung jawab atas *slave* tersebut tidak merespons. Apabila dalam 5 detik tidak ada respons dari node *master*, node *slave*

otomatis mengirimkan data ke node *master* lain yang tersedia. Implementasi sistem komunikasi antara node *master* dan node *slave* dilihat pada Gambar 5.

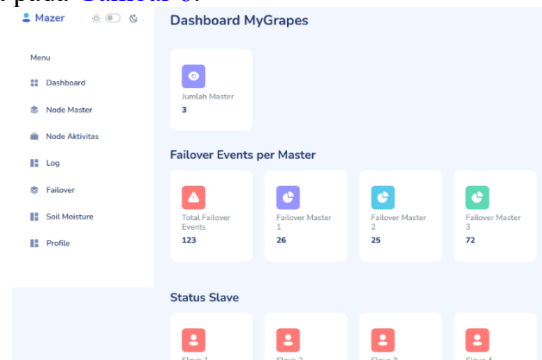


Gambar 5. Implementasi Sistem Komunikasi antara Node Master dan Node Slave

Pada implementasi sistem komunikasi antara node *master* dan *cloud server*, data sensor yang diterima node *master* dikirim ke *cloud* melalui protokol HTTP, dengan node *master* berperan sebagai HTTP client yang mengirimkan data dalam format JSON ke endpoint API.

Pada implementasi metode *failover*, metode *failover* menjaga keberlanjutan pengiriman data sensor dari node *slave* ke node *master* secara otomatis ketika terjadi gangguan pada satu atau lebih node *master*. Dalam kondisi normal, node *slave* mengirim data langsung ke node *master* masing-masing untuk diteruskan ke *cloud* tanpa proses pengalihan. Namun, jika node *master* mengalami kegagalan, *failover* aktif dan mengalihkan pengiriman data ke *master* lain yang masih berfungsi, sehingga gangguan pada aliran data dapat diminimalkan. Apabila seluruh node *master* tidak merespons, proses pengiriman data terhenti hingga salah satu *master* kembali tersedia.

Website pada sistem *monitoring* dibuat informatif, responsif, dan dirancang sebagai media penyajian data yang dikirimkan dari node *master* ke *cloud server*, sehingga pengguna melihat hasil pembacaan sensor, riwayat pengukuran, serta *failover*. Tampilan *website* sistem *monitoring* penelitian ini dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Website Sistem Monitoring Kelembapan Tanah Tanaman

Hasil Pengujian

Pengujian sistem membandingkan dua skenario, yaitu pengujian sistem berbasis arsitektur *master-slave* dan pengujian sistem dengan metode *failover*, untuk menilai pengaruh *failover* terhadap *packet loss* dan jeda pengiriman data. Selain itu, pengujian dilengkapi pengujian *black box* untuk memastikan kinerja sistem sesuai dengan desain yang telah ditetapkan.

Pengujian Sistem Pemantauan Berbasis Arsitektur Master-Slave

Pengujian sistem *monitoring* berbasis arsitektur *master-slave* dilakukan dengan konfigurasi 12 sensor *soil moisture*, 6 node *slave* ESP32 (masing-masing 2 sensor), dan 3 node *master* ESP32 (menangani 2 node *slave* per *master*). Setiap node *slave* membaca kelembapan tanah setiap 60000

ms kemudian mengirimkan ke node *master* melalui Wi-Fi, lalu diteruskan ke *cloud server*. Pengujian dilakukan tiga kali, masing-masing berfokus pada node *master* yang berbeda, tiap pengujian berlangsung satu jam dibagi ke dalam fase-fase 12 menit: seluruh *master* aktif, satu *master* mati, dua *master* mati bersamaan, dan seluruh *master* kembali aktif.

Pengujian 1 berfokus pada kinerja node *master* 1 selama satu jam dengan lima kondisi berurutan. Rincian kondisi dan waktu ditampilkan pada [Tabel 1](#).

Tabel 1. Kondisi Operasional dan Waktu Pelaksanaan Pengujian 1

Kondisi	Waktu Pengujian Pertama	Download (Mbps)	Upload (Mbps)	Keterangan
Sebelum	-	11,12	12,44	-
1	19:14 - 19:26	2,04	11,88	Semua <i>Master</i> Hidup
2	19:26 - 19:38	2,30	12,76	<i>Master</i> 1 Mati
3	19:38 - 19:50	4,01	22,94	<i>Master</i> 1 & <i>Master</i> 2 Mati
4	19:50 - 20:02	4,33	10,55	<i>Master</i> 1 & <i>Master</i> 3 Mati
5	20:02 - 20:14	3,07	11,31	Semua <i>Master</i> Hidup
Sesudah	-	8,09	10,98	-

Berdasarkan Tabel 1, kapasitas jaringan menunjukkan bahwa semakin banyak node *master* yang tidak aktif, kecepatan *download* dan *upload* cenderung meningkat, sedangkan ketika seluruh *master* aktif kecepatan turun signifikan. Penurunan terbesar terjadi pada Kondisi 1, sementara peningkatan tertinggi terlihat pada Kondisi 3 dengan *upload* mencapai 22,94 Mbps, Hasil penerimaan paket data tiap node *slave* selama satu jam ditampilkan pada [Tabel 2](#).

Tabel 2. Hasil Pengujian 1 Pengiriman Paket Data (1 Jam)

ID <i>Slave</i>	Total Paket Dikirim	Total Paket Diterima	Packet loss %
1	24	24	0%
2	24	24	0%
3	48	48	0%
4	48	48	0%
5	48	48	0%
6	48	48	0%
Total	240	240	-
Rata-rata	-	-	0%

Berdasarkan Tabel 2, Pengujian 1 tidak menunjukkan *packet loss* karena seluruh data berhasil dikirim dan diterima, meski terjadi hilangnya komunikasi saat satu atau lebih *master* tidak aktif. Perhentian sementara ini bukan kehilangan paket, melainkan tidak adanya pengiriman selama gangguan. Selain itu, *slave* 1 dan *slave* 2 hanya menghasilkan 24 paket, sedangkan node *slave* lainnya mencapai 48 paket, karena fokus pengujian 1 pada kegagalan *master* 1 sehingga waktu operasional menerima data dari *slave* tersebut lebih singkat dibandingkan *master* lainnya.

Pengujian 2 fokus pada node *master* 2 dengan alur kondisi sama seperti Pengujian 1, namun beda node *master* yang mati. Rincian waktu kondisi pengujian 2 dilihat pada [Tabel 3](#).

Tabel 3. Kondisi Operasional dan Waktu Pelaksanaan Pengujian 2

Kondisi	Waktu Pengujian Kedua	Download (Mbps)	Upload (Mbps)	Keterangan
Sebelum	-	5,48	7,75	-
1	21:01 - 21:13	2,57	10,02	Semua <i>Master</i> Hidup
2	21:13 - 21:25	2,62	5,46	<i>Master</i> 2 Mati
3	21:25 - 21:37	6,82	11,64	<i>Master</i> 2 & <i>Master</i> 1 Mati
4	21:37 - 21:49	4,55	8,21	<i>Master</i> 2 & <i>Master</i> 3 Mati
5	21:49 - 22:01	3,91	3,49	Semua <i>Master</i> Hidup
Sesudah	-	5,05	13,23	-

Pengujian kapasitas jaringan pada Tabel 3 menunjukkan ketidakstabilan dengan kecepatan *download* 2,57-6,82 Mbps dan *upload* 3,49-11,64 Mbps. Jaringan terbebani saat semua *master* aktif, ditandai penurunan *download* dari 5,48 menjadi 2,57 Mbps. Semakin sedikit *master* yang aktif, semakin baik kapasitas jaringan. Hasil paket data Pengujian 2 dilihat pada [Tabel 4](#).

Tabel 4. Hasil Pengujian 2 Pengiriman Paket Data (1 Jam)

ID <i>Slave</i>	Total Paket Dikirim	Total Paket Diterima	<i>Packet Loss</i> %
1	48	48	0%
2	48	48	0%
3	24	24	0%
4	24	24	0%
5	48	48	0%
6	48	48	0%
Total	240	240	-
Rata-rata	-	-	0%

Hasil Pengujian 2, tidak ditemukan *packet loss* dengan 240 paket berhasil dikirim dan diterima, meski terjadi hilang komunikasi saat *master* tidak aktif. Pada tiga kondisi di mana satu atau lebih *master* mati, pengiriman data terhenti sementara tetapi tidak mengakibatkan kehilangan paket karena memang tidak ada pengiriman data. *Slave* 3 dan 4 hanya menghasilkan 24 paket karena fokus pengujian pada kegagalan *master* 2 membuat waktu operasionalnya lebih singkat

Pengujian 3 berfokus pada node *master* 3 selama satu jam dengan alur kondisi yang sama seperti pengujian sebelumnya, namun perbedaan terletak pada *master* yang mengalami gangguan. Rincian waktu tiap kondisi ditampilkan pada Tabel 5.

Tabel 5. Kondisi Operasional dan Waktu Pelaksanaan Pengujian 3

Kondisi	Waktu Pengujian Ketiga	Download (Mbps)	Upload (Mbps)	Keterangan
Sebelum	-	7,19	6,60	-
1	22:07 - 22:19	1,08	14,64	Semua <i>Master</i> Hidup
2	22:19 - 22:31	4,41	5,59	<i>Master</i> 3 Mati
3	22:31 - 22:43	4,50	5,52	<i>Master</i> 3 & <i>Master</i> 1 Mati
4	22:43 - 22:55	5,71	4,11	<i>Master</i> 3 & <i>Master</i> 2 Mati
5	22:55 - 23:07	3,97	15,12	Semua <i>Master</i> Hidup
Sesudah	-	10,19	10,33	-

Kapasitas jaringan pada Pengujian 3 menunjukkan perbedaan kecepatan *download* dan *upload*, *download* berada pada rentang 1,08-5,71 Mbps dan *upload* 4,11-15,12 Mbps. Beban jaringan tertinggi terjadi saat seluruh *master* aktif, sehingga aktivitas node *master* memengaruhi jaringan. Paket yang diterima setiap node *slave* selama satu jam ditampilkan pada Tabel 6.

Tabel 6. Hasil Pengujian 3 Pengiriman Paket Data (1 Jam)

ID <i>Slave</i>	Total Paket Dikirim	Total Paket Diterima	<i>Packet loss</i> %
1	48	48	0%
2	48	48	0%
3	48	48	0%
4	48	48	0%
5	24	24	0%
6	24	24	0%
Total	240	240	-
Rata-rata	-	-	0%

Hasil Pengujian 3, tidak ditemukan *packet loss* dengan 240 paket berhasil dikirim dan diterima, namun komunikasi sempat terhenti ketika *master* tidak aktif meskipun hal ini tidak dikategorikan sebagai kehilangan data. Node *slave* 5 dan 6 menghasilkan 24 paket, lebih sedikit dibanding node lain yaitu 48 paket, karena pengujian fokus pada kegagalan *master* 3 menyebabkan waktu operasionalnya memproses data dari kedua *slave* tersebut lebih singkat.

Berdasarkan ketiga pengujian, sistem dijalankan dengan lima kondisi, hanya kondisi 1 dan 5 yang berjalan ideal karena seluruh node *master* aktif sehingga seluruh node *slave* dapat mengirimkan data. Pada kondisi 2, 3, dan 4, satu atau lebih *master* tidak aktif sehingga pengiriman

data berhenti untuk *slave* yang bergantung pada *master* tersebut, kondisi ini bukan *packet loss*, melainkan kegagalan operasional akibat tidak tersedianya node *master*. Jumlah data yang diterima setiap *master* bergantung pada durasi aktifnya, dengan node *master* yang aktif selama empat kondisi menerima 48 data, sedangkan *master* yang aktif pada dua kondisi menerima 24 data. Total data yang berhasil dikirim dan diterima seluruh pengujian 240 data tanpa ditemukan *packet loss*, karena semua data yang dikirim ke *master* diterima. Hasil menunjukkan keterbatasan sistem tanpa *failover*, keberlangsungan pengiriman bergantung pada ketersediaan node *master* aktif, dengan periode hilangnya komunikasi mencapai 36 menit akibat ketiadaan pengalihan komunikasi.

Parameter QoS ketiga pengujian menunjukkan performa komunikasi yang cukup stabil selama node *master* aktif. Meski terjadi periode hilangnya komunikasi saat *master* tidak aktif, tidak ditemukan *packet loss* karena seluruh paket yang dikirim ke *master* aktif berhasil diterima. Rata-rata tiap parameter ditunjukkan pada Tabel 7.

Tabel 7. Rata-rata QoS Setiap Pengujian Tanpa Failover

Pengujian	Rata-Rata Delay (ms)	Rata-Rata Jitter (ms)	Rata-Rata Throughput (kbps)
1	233,9	549,062	0,03355
2	231,6	404,029	0,03351
3	236.1	581,325	0,03347

Pengujian Sistem Pemantauan Tanaman Berbasis Arsitektur Master-Slave Menggunakan Metode Failover

Pengujian sistem *failover* menggunakan 12 sensor *soil moisture*, 6 node *slave*, dan 3 node *master* ESP32, dengan pembacaan data setiap 60 detik. *Failover* memastikan node *slave* otomatis mengalihkan pengiriman ke *master* lain jika *master* utama tidak merespons. Percobaan komunikasi memiliki batas tunggu 5 detik, jika tidak ada respons, *slave* mencoba *master* berikutnya hingga data diterima atau semua *master* tidak dapat dijangkau. Pengujian dilakukan tiga kali, masing-masing satu jam, dengan lima kondisi 12 menit: semua *master* aktif, satu mati, dua mati, dan kembali aktif, untuk menilai kinerja sistem dalam kondisi normal dan terganggu.

Pengujian 1 fokus pada node *master* 1. Kondisi 1 seluruh *master* aktif, Kondisi 2-4 *master* dimatikan bergantian sehingga node *slave* melakukan *failover* ke *master* yang masih aktif. Kondisi 5 seluruh *master* kembali normal. Rincian waktu tiap kondisi ditunjukkan pada Tabel 8.

Tabel 8. Kondisi Operasional dan Waktu Pelaksanaan Pengujian 1

Kondisi	Waktu Pengujian Pertama	Download (Mbps)	Upload (Mbps)	Keterangan
Sebelum	-	1,39	21,73	-
1	13:47 - 13:59	4,20	18,64	Semua Master Hidup
2	13:59 - 14:11	1,92	15,54	Master 1 Mati
3	14:11 - 14:23	11,47	9,86	Master 1 & Master 2 Mati
4	14:23 - 14:35	7,87	12,68	Master 1 & Master 3 Mati
5	14:35 - 14:47	9,45	15,58	Semua Master Hidup
Sesudah	-	6,47	14,56	-

Berdasarkan Tabel 8, kapasitas jaringan selama Pengujian 1 mengalami fluktuasi, dengan kecepatan *download* 1,39-11,47 Mbps dan *upload* 9,86-18,64 Mbps. Jaringan lebih stabil saat seluruh *master* aktif, namun mengalami penurunan ketika satu atau lebih *master* dimatikan. Hasil penerimaan paket data pada setiap node *slave* dengan *failover* ditunjukkan pada Tabel 9.

Tabel 9. Hasil Pengujian 1 Pengiriman Paket Data (1 Jam)

ID Slave	Total Paket Dikirim	Total Paket Diterima	Packet loss %
1	60	58	3,33%
2	60	59	1,67%
3	60	60	0%
4	60	58	3,33%
5	60	58	3,33%
6	60	58	3,33%
Total	360	351	-

Rata-rata - - 2,5%

Pada Pengujian 1, dari 360 paket dikirim, 351 diterima dan 9 hilang, dengan *packet loss* 2,5%. Node *slave* mengirim data dengan kondisi berbeda, menghasilkan total paket diterima dan jumlah *failover* yang bervariasi pada tiap *master*. *Failover* tertinggi terjadi pada *master* 1 sebanyak 69 kali sesuai fokus scenario. Rincian lengkap *failover* ditampilkan pada Tabel 10.

Tabel 10. Total Paket Diterima dan Jumlah *Failover* Pengujian 1

Node Master	Total Paket Diterima	Failover
1	48	69
2	159	25
3	144	22
Total	351	116

Pengujian 2 fokus pada *master* 2 selama satu jam dengan skenario sama seperti Pengujian 1, hanya berbeda pada *master* yang mengalami gangguan. Waktu dan kondisi dilihat pada Tabel 11.

Tabel 11. Kondisi Operasional dan Waktu Pelaksanaan Pengujian 2

Kondisi	Waktu Pengujian Kedua	Download (Mbps)	Upload (Mbps)	Keterangan
Sebelum	-	6,78	21,57	-
1	15:42 - 15:54	10,20	19,69	Semua Master Hidup
2	15:54 - 16:06	2,65	16,16	Master 2 Mati
3	16:06 - 16:18	2,83	14,64	Master 2 & Master 1 Mati
4	16:18 - 16:30	6,04	13,80	Master 2 & Master 3 Mati
5	16:30 - 16:42	6,57	12,10	Semua Master Hidup
Sesudah	-	1,39	12,06	-

Kapasitas jaringan pada Pengujian 2 menunjukkan kecepatan cenderung tinggi dan stabil saat seluruh *master* aktif, namun menurun ketika satu atau lebih *master* mati karena proses *failover*. Setelah semua *master* kembali aktif, kapasitas jaringan meningkat meski tidak sepenuhnya seperti awal. Hasil penerimaan paket selama satu jam ditampilkan pada Tabel 12.

Tabel 12. Hasil Pengujian 2 Pengiriman Paket Data (1 Jam)

ID Slave	Total Paket Dikirim	Total Paket Diterima	Packet Loss %
1	60	60	0%
2	60	60	0%
3	60	58	3,33%
4	60	56	6,67%
5	60	58	3,33%
6	60	56	6,67%
Total	360	348	-
Rata-rata	-	-	3,33%

Tabel 12 menunjukkan 360 paket dikirim, 348 diterima, terdapat 12 paket hilang dengan *packet loss* 3,33%. Selama Pengujian 2, penerimaan paket bervariasi sesuai kondisi jaringan, dan tercatat jumlah paket yang diterima serta *failover* pada tiap *master*. *Failover* tertinggi terjadi pada *master* 2 sebanyak 69 kali sesuai fokus skenario. Rincian lengkap ditampilkan pada Tabel 13.

Tabel 13. Total Paket Diterima dan Jumlah *Failover* Pengujian 2

Node Master	Total Paket Diterima	Failover
1	161	27
2	48	69
3	139	23
Total	348	119

Pengujian 3 berfokus pada *master 3* dengan skenario yang sama seperti pengujian sebelumnya, namun difokuskan pada kondisi ketika *master 3* tidak aktif. Rincian kondisi dan waktu pengujian ditampilkan pada [Tabel 14](#).

Tabel 14. Kondisi Operasional dan Waktu Pelaksanaan Pengujian 3

Kondisi	Waktu Pengujian Ketiga	Download (Mbps)	Upload (Mbps)	Keterangan
Sebelum	-	4.24	16.27	-
1	17:01 - 17:13	4.02	20.35	Semua <i>Master</i> Hidup
2	17:13 - 17:25	10.26	16.63	<i>Master 3</i> Mati
3	17:25 - 17:37	7.02	21.69	<i>Master 3</i> & <i>Master 1</i> Mati
4	17:37 - 17:49	8.19	15.37	<i>Master 3</i> & <i>Master 2</i> Mati
5	17:49 - 16:01	6.44	16.99	Semua <i>Master</i> Hidup
Sesudah	-	10.37	18.78	-

Kapasitas jaringan Pengujian 3 menunjukkan penurunan kecepatan terjadi saat satu atau lebih *master* mati, menandakan *failover* berlangsung, menunjukkan jaringan tidak stabil sepanjang pengujian. Hasil penerimaan paket ditampilkan pada [Tabel 15](#).

Tabel 15. Hasil Pengujian 3 Pengiriman Paket Data (1 Jam)

ID <i>Slave</i>	Total Paket Dikirim	Total Paket Diterima	Packet Loss %
1	60	59	1.67%
2	60	59	1.67%
3	60	60	0%
4	60	59	1.67%
5	60	60	0%
6	60	60	0%
Total	360	357	-
Rata-rata	-	-	0,83%

Tabel 15 menunjukkan 360 paket dikirim, 357 diterima, 3 paket hilang, *packet loss* 0,83%. Variasi kapasitas jaringan menyebabkan perbedaan jumlah paket diterima. *Failover* terbanyak pada *master 3* yaitu 76 kali karena pengujian fokus pada kondisi *master 3* tidak aktif. Rincian lengkap *failover* tiap node *master* ditampilkan pada [Tabel 16](#).

Tabel 16. Total Paket Diterima dan Jumlah *Failover* Pengujian 3

Node <i>Master</i>	Total Paket Diterima	<i>Failover</i>
1	167	26
2	146	25
3	44	76
Total	357	127

Ketiga pengujian menunjukkan masih terjadi *packet loss*, meski seluruhnya masih dalam kategori baik menurut TIPHON. Pengujian 1 dan 3 “Sangat Baik”, dan Pengujian 2 “Baik”. Hasil seluruh pengujian, 1080 paket dikirim dan 1056 diterima, total paket hilang 24 dan *packet loss* 2,22%. Paket diterima dan *packet loss* tiap *master* selama *failover* ditunjukkan pada [Tabel 17](#).

Tabel 17. Paket Data tiap *Master* dan *Packet loss* Tiga Pengujian *Failover*

Pengujian	<i>Master</i> 1	<i>Master</i> 2	<i>Master</i> 3	Total Paket Dikirim	Total Paket Diterima	Packet loss (%)
1	48	159	44	360	351	2,5%
2	161	48	139	360	348	3,33%
3	167	146	44	360	357	0,83%
Total	376	353	227	1080	1056	-
Rata-rata	-	-	-	-	-	2,22%

Paket diterima dan *packet loss* menunjukkan sistem menangani perpindahan saat *master* mengalami gangguan. Jumlah *failover* tiap node *master* ketiga pengujian dirinci pada [Tabel 18](#).

Tabel 18. Total *Failover* Masing-Masing Node *Master* Setiap Pengujian

Pengujian	<i>Failover Master 1</i>	<i>Failover Master 2</i>	<i>Failover Master 3</i>	Total <i>Failover</i>
1	69	25	22	116
2	27	69	23	119
3	26	25	76	127

Total *failover* menunjukkan sistem memindahkan komunikasi saat *master* gagal. Node *slave* tetap mengirim data melalui *master* lain, menghasilkan jeda perpindahan yang dihitung sebagai total dan rata-rata jeda. Terdapat 362 *failover*, total jeda 285,1 detik, rata-rata 0,78 detik, dan *packet loss* rata-rata 2,22%, menunjukkan sistem memulihkan komunikasi dan kehilangan paket masih dalam batas wajar. Total jeda dan rata-rata dari tiga pengujian ditampilkan pada Tabel 19.

Tabel 19. Total *Failover* Masing-Masing Node *Master* Setiap Pengujian

Pengujian	Total <i>Failover</i>	Total Jeda	Jeda Rata-Rata
1	116	92,44 detik	0,796 detik
2	119	93,09 detik	0,782 detik
3	127	99,57 detik	0,784 detik
Total	362	285,1 detik	0.78 detik

Hasil pengujian dengan *failover* menunjukkan sistem mempertahankan kestabilan pengiriman data dari sisi *throughput* dan ukuran paket serta peningkatan *delay* dan *jitter* selama perpindahan komunikasi. Sistem menjaga kelangsungan transmisi data meski terjadi gangguan pada salah satu atau lebih *master*. Nilai rata-rata parameter QoS dapat dilihat pada Tabel 20.

Tabel 20. Rata-rata QoS Setiap Pengujian dengan *Failover*

Pengujian	Rata-Rata <i>Delay</i> (ms)	Rata-Rata <i>Jitter</i> (ms)	Rata-Rata <i>Throughput</i> (kbps)
1	636,69	648,219	0,097
2	838,95	760,313	0,096
3	705,50	606,624	0,099

Pembahasan

Penelitian ini menerapkan *failover* pada arsitektur *master-slave* untuk transmisi data sensor *soil moisture* dalam sistem monitoring tanaman berbasis mikrokontroler ESP32. Sistem terdiri dari tiga node *master*, enam node *slave*, dan dua belas sensor, dengan setiap node *slave* terhubung ke dua sensor melalui expansion board dan adaptor 12V. Komunikasi antar node menggunakan protokol HTTP melalui Wi-Fi dengan IP statis, sehingga pengiriman data berlangsung tanpa konflik alamat jaringan. Metode *failover* diterapkan untuk memastikan keberlanjutan komunikasi. Jika satu node *master* mengalami gangguan, jalur komunikasi secara otomatis dialihkan ke node *master* lain yang aktif, sehingga data tetap dikirim ke *cloud server*. Sebaliknya, tanpa *failover*, pengiriman data hanya optimal ketika seluruh node *master* aktif, dan gangguan menyebabkan berhentinya komunikasi serta hilangnya sebagian data.

Hasil pengujian menunjukkan sistem *failover* meningkatkan keandalan transmisi. Pada Pengujian 1, 351 dari 360 data berhasil diterima (*packet loss* 2,5%), Pengujian 2 sebanyak 348 data (3,33%), dan Pengujian 3 sebanyak 357 data (0,83%), dengan rata-rata *packet loss* 2,22%. Waktu alih komunikasi antar node *master* rata-rata 0,78 detik, memungkinkan pengiriman data tetap lancar setiap menit. Secara keseluruhan, penerapan *failover* dapat meminimalkan kehilangan data, mempercepat proses alih jalur komunikasi, dan menjaga kelancaran transmisi data meski terjadi gangguan atau perubahan kondisi jaringan.

SIMPULAN

Penerapan metode *failover* secara signifikan meningkatkan keandalan sistem monitoring dibandingkan sistem tanpa metode. Pada sistem tanpa *failover*, saat node *master* mati, pengiriman data dari node *slave* tidak dilakukan sama sekali, sehingga paket data tidak terkirim sejak awal dan kondisi ini tidak termasuk *packet loss*. Sebaliknya dengan *failover*, sistem dapat melanjutkan

komunikasi meskipun terjadi gangguan, ditunjukkan dengan rata-rata *packet loss* yang rendah, yaitu 2,22% (bervariasi antara 0,83% hingga 3,33% antar pengujian), membuktikan optimalisasi keutuhan data. Meskipun sistem tanpa *failover* tidak memiliki jeda perpindahan, namun mengalami periode henti komunikasi total saat *master* mati. Sedangkan sistem dengan *failover* menunjukkan gangguan yang bersifat sementara dengan jeda perpindahan komunikasi singkat, yaitu rata-rata 0,78 detik (0,796 detik, 0,782 detik, dan 0,784 detik pada tiga pengujian individu), sehingga berhasil meminimalkan durasi henti komunikasi dan menjaga kelanjutan pengiriman.

DAFTAR PUSTAKA

- [1] S. Zhu, Y. Liang, and D. Gao, "Study of soil respiration and fruit quality of table grape (*Vitis vinifera* L.) in response to different soil water content in a greenhouse," *Commun. Soil Sci. Plant Anal.*, vol. 49, no. 21, pp. 2689–2699, 2018, doi: 10.1080/00103624.2018.1538369.
- [2] E. Tiara, I. Ruslianto, and Suhardi, "Sistem Pemantauan Dan Kendali Kelembapan Tanah Dan Ph Pada Tanaman Anggur Berbasis Android (Studi Kasus : Greenhouse Fmipa Untan)," *Coding J. Komput. dan Apl.*, vol. 11, no. 03, pp. 437–446, 2023.
- [3] H. Marcos and H. Muzaki, "Monitoring Suhu Udara dan Kelembaban Tanah pada Budidaya Tanaman Pepaya," vol. 03, no. 127, pp. 32–43, 2022.
- [4] H. Y. P. Al-Jufri, O. Novianti, G. Muhammad, R. Adytya, and A. N. Pramudhita, "Otomatisasi Pertanian Dengan Sensor Soil Moisture, Sensor Cahaya, Led Grow Lamps, Dan Pompa Air Untuk Pertumbuhan Tanaman Optimal," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 3, pp. 484–488, 2023, doi: 10.23960/jitet.v11i3.3192.
- [5] A. Yoshua, R. Primananda, and A. S. Budi, "Implementasi Pengiriman Data Multi-Node Sensor Menggunakan Metode Master-slave pada Komunikasi LoRa," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 10, pp. 3445–3454, 2020.
- [6] M. Chouk, *Master-Slave Replication, Failover, and Distributed Recovery*, no. June. 2003.
- [7] H. S. Simarmata and I. R. Widiarsari, "Manajemen Bandwidth Menggunakan Hierarchical Token Bucket (HTB) (Studi Kasus: PT. Orion Cyber Internet)," *Technomedia J.*, vol. 8, no. 2SP, pp. 248–260, 2023, doi: 10.33050/tmj.v8i2sp.2078.
- [8] W. K. Raharja, D. N. Dominicus, and Jalinas, "Pengaplikasian Internet of Things Untuk Monitoring Lingkungan Lahan Tanaman Anggur," *J. Elektro Luceat*, vol. 9, no. 1, 2022.
- [9] M. Abduh, Arini, H. T. Sukmana, and I. M. M. Matin, "Downtime Prevention Using Failover on IoT-based Contactless Temperature Checkers," *2022 10th Int. Conf. Cyber IT Serv. Manag. CITSM 2022*, no. April 2024, 2022, doi: 10.1109/CITSM56380.2022.9935842.
- [10] A. Yanziah, S. Soim, and M. M. Rose, "Analisis jarak jangkauan lora dengan parameter rssi dan packet loss pada area urban," *J. Teknol. Technoscintia*, pp. 59–67, 2020.
- [11] D. S. Hadiningrum, W. W. IGPW, and I. W. A. Arimbawa, "Analisis Pengaruh Jarak Terhadap Delay dan Packet Delivery Ratio (PDR) pada Jaringan LORA untuk Komunikasi Data Irigasi Tetes (Studi Kasus: Dusun Amor-Amor)," *Univ. Mataram Repository*, 2025.