

RANCANG BANGUN PROTOTIPE SISTEM MONITORING ATS-AMF DAN BATTERY CHARGER MENGGUNAKAN TELEGRAM DI PERUMDA AIR MINUM TIRTA SANJIWANI UNIT PRODUKSI BLANGSINGA

Ni Nyoman Triandani Ayuningsih¹, Cokorde Gede Indra Partha², I Wayan Arta Wijaya²

¹Mahasiswa Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana

²Dosen Program Studi Teknik Elektro, Fakultas, Universitas Udayana

Jl. Raya Kampus Unud Jimbaran, Kec. Kuta Sel, Kabupaten Badung, Bali 80361

Email: ntriandani3@gmail.com, cokindra@unud.ac.id, artawijaya@unud.ac.id

ABSTRAK

Sistem monitoring *Automatic Transfer Switch - Automatic Mains Failure* (ATS-AMF) dan *battery charger* diperlukan untuk memudahkan operator dalam melakukan pemantauan sistem dari jarak jauh sehingga dapat menjaga kelangsungan suplai air ke konsumen. Sistem monitoring berbasis *Internet of Things* (IoT) tidak mengharuskan operator selalu *standby* berada di lokasi, operator dapat memonitoring sistem dari jarak jauh di luar waktu jaga operator. Perancangan prototipe sistem menggunakan mikrokontroler NodeMCU ESP32 DEVKIT V1 dan aplikasi Telegram untuk mengirim dan menerima pesan dari jarak jauh, serta memberikan notifikasi spam saat terjadi gangguan pada sistem ATS-AMF dan *battery charger*. Data yang dikirim dari ATS-AMF yaitu data tegangan, frekuensi, arus, daya listrik, sumber listrik yang sedang digunakan, dan data gangguan kedua sumber listrik PLN dan genset. Data yang dikirim dari *battery charger* yaitu data tegangan, arus, tahapan pengisian baterai, *state of charge* baterai, dan data gangguan. Hasil pengujian menunjukkan keberhasilan pengiriman data 100% pada *chatbot* Telegram, dengan *delay* rata-rata antara 4,8 detik hingga 5,7 detik. Sistem monitoring ini memastikan pemantauan yang efektif dari jarak jauh untuk menjaga kelancaran aliran listrik dan kelangsungan suplai air kepada konsumen.

Kata kunci : Monitoring, *Internet of Things*, NodeMCU ESP32, Telegram

ABSTRACT

The monitoring system of the Automatic Transfer Switch - Automatic Mains Failure (ATS-AMF) and battery charger is necessary to facilitate operators in remotely monitoring the system, thus ensuring the continuous supply of water to consumers. An Internet of Things (IoT)-based monitoring system does not require operators to always be on standby at the location, operators can monitor the system remotely outside their working hours. The prototype system is designed using the NodeMCU ESP32 DEVKIT V1 microcontroller and Telegram application for sending and receiving messages remotely, as well as providing spam notifications when disruptions occur in the ATS-AMF and battery charger systems. Data sent from the ATS-AMF includes voltage, frequency, current, power, the power source being used, and disruption data from both the PLN electricity source and the generator. Data sent from the battery charger includes voltage, current, battery charging stages, battery state of charge, and disruption data. Test results show a 100% success rate in data transmission through the Telegram chatbot, with an average delay ranging from 4.8 seconds to 5.7 seconds. This monitoring system ensures effective remote monitoring to maintain the smooth flow of electricity and the continuous supply of water to consumers.

Key Words: Monitoring, *Internet of Things*, NodeMCU ESP32, Telegram.

1. PENDAHULUAN

Sistem penyediaan air di Unit Produksi Blangsinga Perumda Air Minum Tirta Sanjiwani menggunakan sumber listrik utama yaitu listrik PLN dan sumber listrik cadangan yaitu genset 25 kW. Proses peralihan listrik antara PLN dengan genset saat terjadi pemadaman listrik menggunakan sistem *Automatic Transfer Switch - Automatic Mains Failure* (ATS-AMF). Operator saat ini memonitoring sistem ATS-AMF secara langsung atau pengamatan langsung, yang mengharuskan operator selalu *standby* di lokasi untuk memastikan ATS-AMF berfungsi dengan baik atau tidak. Kondisi tersebut menjadi masalah serius jika terjadi pemadaman di luar waktu jaga operator, selain melakukan monitoring ATS-AMF kondisi genset sebagai sumber listrik cadangan juga harus diperhatikan, jika kondisi aki genset tidak mencukupi maka sistem ATS-AMF akan terhenti dan terjadi gangguan di kedua sumber listrik PLN dan genset. Sistem monitoring *battery charger* diperlukan untuk memonitoring pengisian aki genset dari jarak jauh untuk menjaga kondisi aki agar selalu dalam kondisi siap digunakan saat menghidupkan genset.

Internet of Things (IoT) digunakan untuk melakukan monitoring, berbagi, mengirim, dan menerima data dari jarak jauh secara efektif dengan menghubungkan perangkat fisik dan server melalui jaringan internet [1].

Prototipe sistem monitoring ATS-AMF dan *battery charger* berbasis IoT diperlukan untuk memberikan informasi dan melakukan pemantauan dari jarak jauh guna menjaga

kelangsungan suplai air ke konsumen. Operator dapat memonitoring kondisi sistem ATS-AMF dan *battery charger* secara langsung melalui *chatbot* Telegram dengan mengirimkan pesan perintah dan dapat menampilkan notifikasi spam saat terjadi gangguan yang harus diatasi pada sistem ATS-AMF dan *battery charger*. Data yang dikirim dari sistem monitoring ATS-AMF yaitu data tegangan, frekuensi, arus, daya listrik, sumber listrik yang sedang digunakan, dan data gangguan kedua sumber PLN dan genset. Data yang dikirim dari sistem monitoring *battery charger* yaitu data tegangan, arus, tahapan pengisian baterai, *state of charge* baterai, dan data gangguan.

2. KAJIAN PUSTAKA

2.1 Mikrokontroler NodeMCU ESP32

Mikrokontroler NodeMcu ESP32 adalah salah satu jenis mikrokontroler *open source* yang diperkenalkan oleh *Espressif System* dan merupakan pengembangan dari mikrokontroler ESP8266. Spesifikasi ESP32 sudah tersedia modul Wi-Fi dalam chip, sehingga mendukung dalam pembuatan sistem aplikasi *internet of things* [2].

Pada pin *out* terdiri dari:

- 15 pin ADC (*Analog Digital Converter*, berfungsi untuk merubah sinyal analog ke digital),
- 2 pin DAC (*Digital Analog Converter*, berfungsi untuk merubah sinyal digital ke analog),
- 21 pin PWM (*Pulse Width Modulation*),
- 9 *Touch Sensor*,
- 3 pin jalur antarmuka UART,
- Pin antarmuka I2C dan 3 pin SPI



Gambar 1. Spesifikasi ESP32 [3].

2.2 Arduino IDE

Arduino *Integrated Development Enviroment* (IDE) adalah perangkat lunak yang digunakan untuk menulis, menguji, dan mengirimkan program kontrol sebuah mikrokontroler. Bahasa pemrograman yang digunakan yaitu bahasa C dan C++ yang disimplifikasi dengan *libraries* Arduino. Arduino IDE memiliki antarmuka yang terdiri dari *Editor Program, Verify, Uploader, New, Save, Open*, dan *Serial Monitor*. Selain itu, terdapat lima menu yaitu ada *File, Edit, Sketch, Tool*, dan *Help* [4].

2.3 Internet of Things (IoT)

Internet of things (IoT) merupakan jaringan infrastruktur global dinamis yang memiliki kemampuan konfigurasi berdasarkan standar protokol komunikasi antara mesin, peralatan, antarmuka cerdas, dan benda fisik lainnya yang terhubung, serta terintegrasi ke dalam jaringan informasi. Sederhananya IoT dapat menggabungkan berbagai *device* yang saling terhubung dan bertukar data melalui internet sebagai infrastruktur utama dalam menghubungkan objek-objek tertentu. IoT bekerja dengan memproses data dari hasil pembacaan sensor pada benda fisik yang nyata kemudian dikirimkan melalui jaringan internet ke server [5].

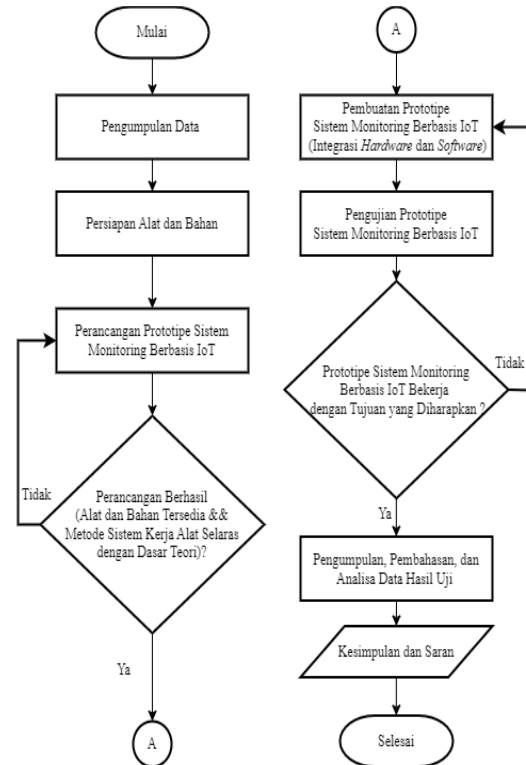
2.4 Telegram

Telegram adalah aplikasi pengiriman pesan dengan fokus pada kecepatan dan keamanan. Pesan Telegram dapat tersinkronisasi di sejumlah ponsel, tablet, ataupun *computer* (Windows, Mac, Linux). Telegram mendukung adanya *bot*, untuk memudahkan pengguna dalam membuat semacam aplikasi chattingan khusus dan juga menggantikan tugas moderasi di dalam sebuah grup. Bot bisa dijalankan atas dasar perintah atau melalui bahasa pemrograman seperti Python, Java, PHP, dan lainnya [6].

3. METODOLOGI PENELITIAN

Penelitian ini dilaksanakan di Perumda Tirta Sanjiwani Kabupaten Gianyar Unit Produksi Blangsinga sebagai tempat acuan

yang belum tersedia sistem monitoring ATS-AMF dan *battery charger*. Waktu pelaksanaan dimulai bulan Juli sampai Desember 2023.



Gambar 2. Diagram Alir Penelitian

Penjelasan Gambar 2 yaitu sebagai berikut: Langkah 1. Pengumpulan Data

Pengumpulan data diperoleh dari observasi lapangan di Perumda Tirta Sanjiwani Kabupaten Gianyar Unit Produksi Blangsinga. Data yang dikumpulkan yaitu data spesifikasi alat di *datasheet*. Data tersebut diperlukan untuk memilih alat dan bahan perancangan prototipe sistem yang sesuai.

Langkah 2. Persiapan Alat dan Bahan

Tahap persiapan alat dan bahan dilakukan dengan mengidentifikasi dan mempersiapkan alat serta bahan yang diperlukan agar dapat memenuhi spesifikasi alat untuk membangun prototipe sistem.

Langkah 3. Perancangan dan Pembuatan Prototipe Sistem

Tahap perancangan dan pembuatan prototipe sistem yang terdiri dari *hardware* dan *software* (program monitoring) dengan terlebih dahulu menyesuaikan komponen yang digunakan dengan dasar teori dan

membuat desain rangkaian pada fritzing. Baris program dibuat dalam Arduino IDE. *Integrasi software* dan *hardware* dilakukan ketika setiap perangkat keras sudah terhubung dan tidak terdapat *error* di baris program.

Langkah 4. Pengujian Prototipe Sistem

Pengujian dilakukan sebagai tolak ukur keberhasilan sistem. Pendekatan eksperimental yang terstruktur digunakan hingga sistem bekerja sesuai dengan tujuan. Langkah 5. Pengumpulan dan Analisa Data Hasil Uji Prototipe Sistem

Pengambilan data dari proses pengujian dilakukan agar dapat dianalisis untuk mengetahui hasil dari rancang bangun prototipe sistem yang di rancang.

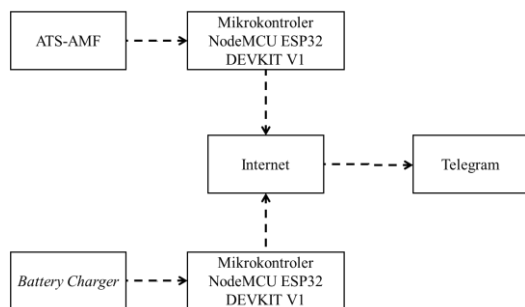
Langkah 6. Kesimpulan

Penarikan kesimpulan berdasarkan hasil pengujian dan analisis data.

4. HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem Monitoring ATS-AMF dan Battery Charger

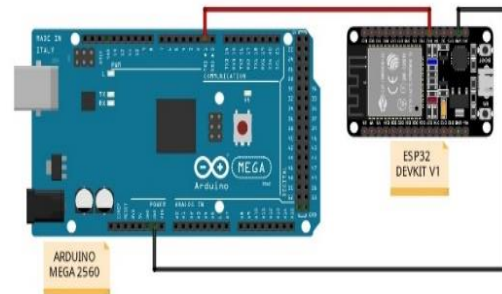
Perancangan sistem monitoring berbasis IoT pada sistem ATS-AMF dan *battery charger* ini menggunakan mikrokontroler NodeMCU ESP32 DEVKIT V1 yang mengambil data hasil pembacaan sensor dari Arduino Mega 2560 pada sistem ATS-AMF dan data hasil pembacaan sensor dari Arduino Uno R3 pada sistem *battery charger*. Pengiriman data ke Telegram menggunakan modul ESP32 sebagai modul Wi-Fi dan komunikasi. NodeMCU ESP32 DEVKIT V1 harus terhubung dengan jaringan internet, sehingga dapat mengirimkan pesan dan notifikasi kepada operator saat terjadi gangguan pada sumber listrik melalui Aplikasi Telegram.



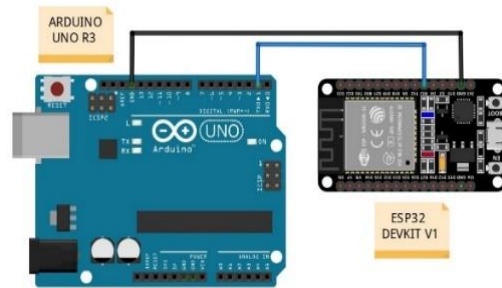
Gambar 3. Blok Diagram Sistem Monitoring

4.2 Perancangan Perangkat Keras Prototipe Sistem Monitoring ATS-AMF dan Battery Charger

Perancangan perangkat keras sistem monitoring ATS-AMF dan *battery charger* ditunjukkan pada Gambar 4 dan Gambar 5 berikut.

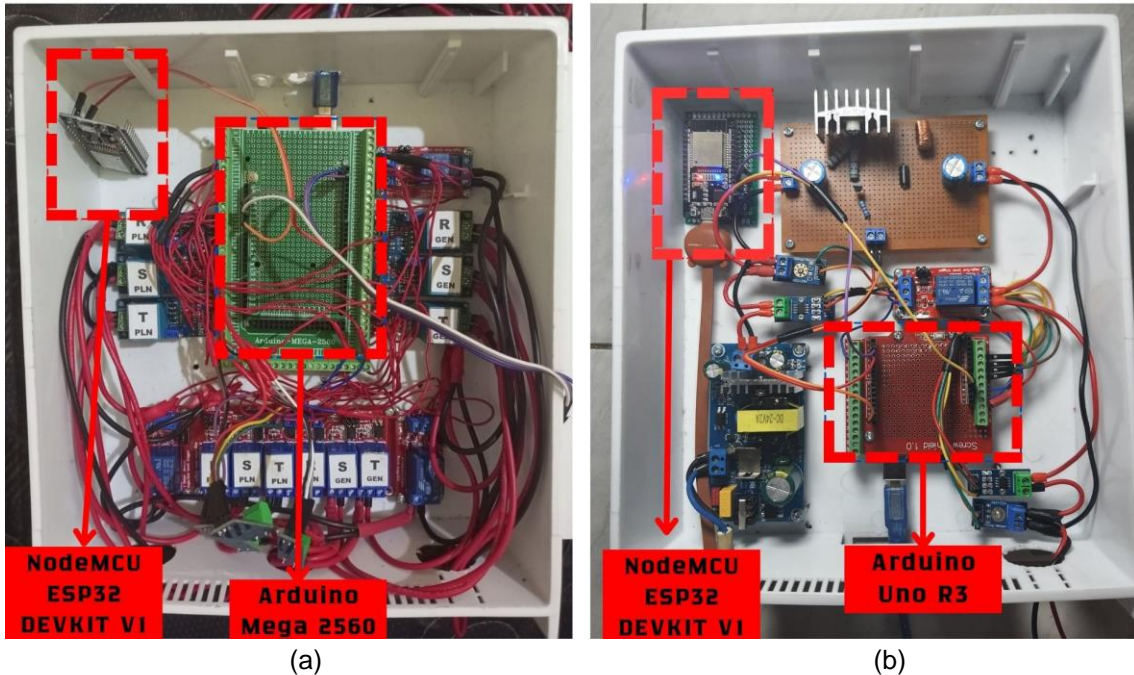


Gambar 4. Rangkaian ESP32 dengan Arduino Mega pada Sistem Monitoring ATS-AMF



Gambar 5. Rangkaian ESP32 dengan Arduino Uno pada Sistem Monitoring Battery Charger

Sistem monitoring ATS-AMF dan *battery charger* melakukan komunikasi serial dengan menghubungkan pin TX0 pada Arduino Mega 2560 dan Arduino Uno R3 ke pin RX2 pada ESP32 DEVKIT V1. Pin TX pada Arduino digunakan untuk mengirim data ke ESP32 DEVKIT V1 dan pin RX pada ESP32 digunakan untuk menerima data. Selain itu, dihubungkan pin GND (ground) pada Arduino ke pin GND (ground) pada ESP32 DEVKIT V1. Hasil perancangan perangkat keras prototipe sistem monitoring ditunjukkan pada Gambar 6 berikut.



Gambar 6 Hasil Perancangan *Hardware* Sistem Monitoring (a)ATS-AMF; (b)*Battery Charger*

Penjelasan bagian - bagian dari perancangan prototipe sistem monitoring berbasis IoT adalah sebagai berikut:

- a) Mikrokontroler Arduino Mega 2560
Sistem ATS-AMF pada Gambar 10(a) menggunakan Arduino Mega 2560 sebagai pusat pemrosesan data pembacaan sensor arus ACS712-30A, sensor tegangan, dan frekuensi ZMPT101B. Data tersebut akan dikirim oleh Arduino Mega 2560 ke ESP32.
- b) Mikrokontroler Arduino Uno R3
Sistem *battery charger* pada Gambar 10(b) menggunakan Arduino Uno R3 sebagai pusat pemrosesan data pembacaan sensor arus ACS712-20A, sensor tegangan DC, dan *buck converter*. Data tersebut akan dikirim oleh Arduino Uno R3 ke ESP32.
- c) Mikrokontroler NodeMCU ESP32
NodeMCU ESP32 digunakan berjumlah 2 buah yang berfungsi sebagai pusat pemrosesan data hasil kirim dari Arduino Mega 2560 dan Arduino Uno R3 untuk dikirim ke aplikasi Telegram untuk menampilkan data monitoring sistem.

4.3 Perancangan Perangkat Lunak

Perangkat lunak pada sistem monitoring ini menggunakan aplikasi Arduino IDE untuk pemrograman ESP32 agar dapat membaca data dari Arduino melalui komunikasi serial dan mengirim data ke *bot* Telegram. Pemrograman dapat berjalan menggunakan perintah user pada *chatbot* Telegram.



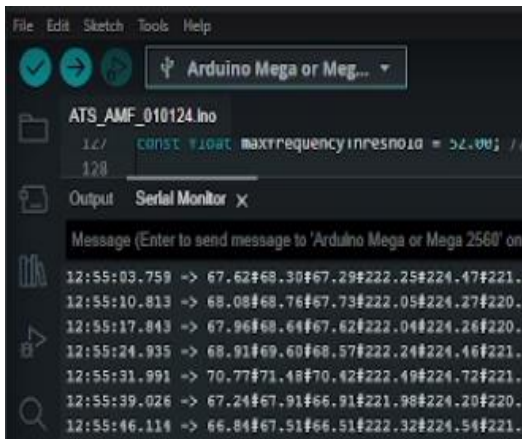
Gambar 7 *User Interface Bot* Telegram

Data yang dikirim dari sistem monitoring ATS-AMF yaitu data tegangan, frekuensi, arus, daya listrik, sumber listrik yang sedang digunakan, dan data gangguan

kedua sumber PLN dan genset. Data yang dikirim dari sistem monitoring *battery charger* yaitu data tegangan, arus, tahapan pengisian baterai, *state of charge* baterai, dan data gangguan.

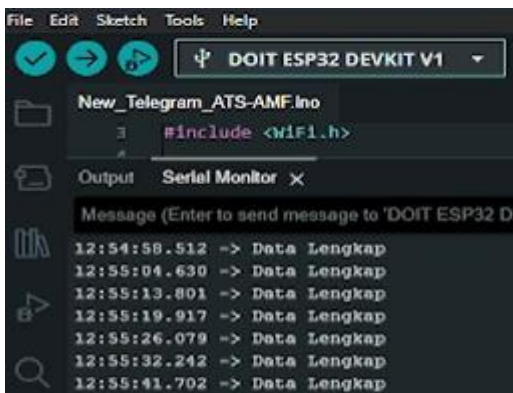
4.4 Pengujian Komunikasi Serial

Hasil pengujian komunikasi serial pengiriman data serial antara Arduino dengan ESP32, data berhasil dikirim ke ESP32 oleh Arduino ditandai dengan tulisan parsing data pada serial monitor.



Gambar 8 Arduino Berhasil Mengirim Data

Hasil pengujian komunikasi serial pembacaan data serial antara Arduino dengan ESP32 yaitu data berhasil diterima oleh ESP32 ditandai dengan tulisan "Data Lengkap" pada tampilan serial monitor



Gambar 9 ESP32 Berhasil Menerima Data

4.5 Pengujian Koneksi ESP32 dengan Jaringan Internet

Pengujian koneksi ESP32 dengan jaringan internet dilakukan untuk

mengetahui waktu yang dibutuhkan ESP32 terhubung dengan koneksi jaringan internet Wi-Fi dan untuk mengetahui jarak terjauh ESP32 untuk terkoneksi ke jaringan internet Wi-Fi. Pengujian ini menggunakan Wi-Fi TP-LINK router Wi-Fi TL-WR840N. ESP32 berhasil terkoneksi jaringan internet ditandai dengan LED menyala berwarna biru.

Tabel 1. Hasil Pengujian Jarak Koneksi ESP32 dengan Wi-Fi

Jarak	Waktu	Gambar
0 m	03.61 s	00:03.61
10 m	03.74 s	00:03.74
20 m	03.88 s	00:03.88
30 m	04.47 s	00:04.47
40 m	05.57 s	00:05.57

Hasil pengujian Tabel 1 menunjukkan bahwa jarak dapat mempengaruhi koneksi antara ESP32 dengan Wi-Fi, semakin jauh ESP32 dari router semakin lama *delay* ESP32 terkoneksi dengan Wi-Fi. Penempatan antara router Wi-Fi dengan ESP32 diberikan jarak sedekat mungkin yaitu dengan maksimal 40 meter dan menggunakan jaringan internet yang baik agar koneksi internet tetap dalam jangkauan ESP32.

4.6 Pengujian Unjuk Kerja Monitoring Online melalui Telegram

Telegram digunakan untuk menyampaikan informasi jarak jauh dengan cepat dan akurat. Operator dapat membalas pesan dan menerima notifikasi pada bot Telegram. Fungsi perintah pada bot Telegram digunakan untuk memudahkan operator dalam memonitoring sistem. Pengujian fungsi perintah pada sistem monitoring ini berhasil beroperasi sesuai rancangan.

Sistem monitoring ATS-AMF menggunakan fungsi perintah yaitu perintah /START, /STATUS, /PLN, /GENSET, /INFO, /ON_NOTIF, /OFF_NOTIF, dan notifikasi spam saat terjadi gangguan kedua sumber genset dan PLN. Fungsi perintah pada sistem monitoring *battery charger* yaitu perintah /START, /STATUS, /BUCK, /BATERAI, /INFO, /ON_NOTIF,

/OFF_NOTIF, dan notifikasi spam saat terjadi gangguan input dan output. Notifikasi spam terus dikirim sampai pesan dibalas "OK" dan gangguan diperbaiki, jika gangguan belum diperbaiki, maka notifikasi spam akan muncul kembali dalam 5 menit setelah pesan "OK" dikirim. Notifikasi spam dapat dinonaktifkan dengan perintah /OFF_NOTIF jika terjadi gangguan permanen dan dapat di aktifkan kembali dengan perintah /ON_NOTIF. Notifikasi spam ini berfungsi untuk memberikan informasi secara cepat kepada operator agar gangguan cepat diatasi sehingga dapat menjaga kelangsungan suplai air ke konsumen.

Pengujian waktu pengiriman pesan dan notifikasi spam menggunakan chatbot Telegram dilakukan untuk mengetahui respon waktu (durasi) yang dibutuhkan pada saat pengiriman data ke Telegram.

Tabel 2. Hasil Pengujian Waktu Pengiriman Pesan pada Sistem Monitoring ATS-AMF

Perintah	Waktu Perintah Dikirim	Waktu Pesan Diterima	Delay (Detik)	Keterangan (Sukses/Gagal)
/START	13:39:18	13:39:23	5	Sukses
/INFO	13:39:28	13:39:35	7	Sukses
/STATUS	13:39:42	13:39:47	5	Sukses
/PLN	13:39:54	13:40:00	6	Sukses
/OFF_NOTIF	13:46:26	13:46:34	8	Sukses
/ON_NOTIF	13:48:01	13:48:06	5	Sukses
/GENSET	13:52:36	13:52:40	4	Sukses

Tabel 3. Hasil Pengujian Waktu Pengiriman Notifikasi Spam pada Sistem Monitoring ATS-AMF

Notifikasi Spam	Gangguan Sistem PLN dan Genset	Delay (Detik)	Notifikasi Spam (Diterima/Tidak)
1	Undervoltage Sumber PLN, Undervoltage Genset	5	Diterima
2	Overvoltage Sumber PLN, Overvoltage Sumber Genset	4	Diterima
3	Under Frequency Sumber PLN, Under Frequency Sumber Genset	6	Diterima
4	Over Frequency Sumber PLN, Over Frequency Sumber Genset	7	Diterima

Tabel 4. Hasil Pengujian Waktu Pengiriman Pesan Sistem Monitoring Battery Charger

Perintah	Waktu Perintah Dikirim	Waktu Pesan Diterima	Delay (Detik)	Keterangan (Sukses/Gagal)
/START	16:10:53	16:10:58	5	Sukses
/INFO	16:11:06	16:11:11	5	Sukses
/STATUS	16:11:17	16:11:24	7	Sukses
/BUCK	16:11:27	16:11:34	7	Sukses
/BATERAI	16:11:39	16:11:45	6	Sukses
/OFF_NOTIF	16:17:33	16:17:38	5	Sukses
/ON_NOTIF	16:17:47	16:17:51	4	Sukses

Tabel 5. Hasil Pengujian Waktu Pengiriman Notifikasi Spam Monitoring Battery Charger

Notifikasi Spam	Gangguan Sistem	Delay (Detik)	Notifikasi Spam (Diterima/Tidak)
1	Gangguan Output Tegangan Lebih	5	Diterima
2	Gangguan Output Tegangan Kurang	4	Diterima
3	Gangguan Input Tegangan Kurang dan Gangguan Output Tegangan Kurang	5	Diterima
4	Gangguan Input Tegangan Kurang dan Gangguan Output Tegangan Lebih	4	Diterima
5	Gangguan Input Tegangan Kurang	6	Diterima

Hasil pengujian waktu pengiriman pesan dan notifikasi spam menunjukkan bahwa data 100% sukses diterima pada Telegram, dengan delay rata-rata yaitu pada Tabel 2 delay sebesar 5,7 detik, Tabel 3 delay sebesar 5,5 detik, Tabel 4 delay sebesar 5,6 detik, dan Tabel 5 delay sebesar 4,8 detik.

Hasil pengujian keseluruhan waktu pengiriman pesan dan notifikasi spam pada sistem ATS-AMF dan sistem battery charger menunjukkan bahwa adanya delay rata-rata 4,8 detik hingga 5,7 detik karena dipengaruhi koneksi internet yang kurang stabil dan jarak antara ESP32 dengan Wi-Fi.

5. KESIMPULAN

Berdasarkan prototipe yang telah dirancang, dapat disimpulkan:

1. Prototipe sistem monitoring berbasis IoT berhasil dibuat menggunakan

NodeMCU ESP32 DEVKIT V1 dan *chatbot* Telegram sebagai pengirim perintah dan menerima balasan dari perintah yang dikirim, serta dapat menerima notifikasi secara spam saat terjadi gangguan pada sistem ATS-AMF dan *battery charger*.

2. Sistem monitoring yang dirancang berhasil dibangun dan bekerja dengan baik, dengan tingkat kesesuaian sebesar 100% sesuai dengan fungsi perintah yang dirancang. Pengiriman pesan dari NodeMCU ESP32 ke Telegram terdapat *delay* yang dipengaruhi oleh koneksi jaringan Wi-Fi yang kurang stabil dan juga dipengaruhi oleh jarak antara NodeMCU ESP32 dengan jaringan internet (Wi-Fi). Penempatan jaringan Wi-Fi dengan ESP32 diberikan jarak sedekat mungkin kurang dari 40 meter.

6. DAFTAR PUSTAKA

- [1] Ardiansyah, "Monitoring Daya Listrik Berbasis IoT (Internet of Things)," *Skripsi*, Universitas Islam Indonesia, 2020.
- [2] N. Y. Sapriyanto, "Sistem Kontrol Dan Monitoring Daya Listrik Rumah Berbasis Internet of Things," *Skripsi*, Universitas Dinamika Surabaya, 2020.
- [3] A. P. W. Hendrawan and N. P. Agustini, "Simulasi Kendali Dan Monitoring Daya Listrik Peralatan Rumah Tangga Berbasis ESP32," *Alinier Jurnal*, vol. 3, pp. 54-67, 2022.
- [4] J. Nussey, *Arduino*, 2nd Edition, 2020.
- [5] Y. Fauzan, "Kotak Penerima Paket Berbasis IoT Menggunakan Modul ESP32-CAM," *Skripsi*, Universitas Islam Negeri Syarif Hidayatullah, 2020.
- [6] S. Sawidin, Y. R. Putung, A. P. Waroh, T. Marsela, Y. H. Sorongan and C. P. Asa, "Kontrol dan Monitoring Sistem Smart Home Menggunakan Web Thinger.io Berbasis IoT," *Prosiding The 12th Industrial Research Workshop and National Seminar*, pp. 464-471, 2021.