Jurnal Riset Rekayasa Elektro
Volume 8 No 1, Juni 2026, Hal. 50~63
DOI : 10.30595/jrre.v8i1.30103

P-ISSN: 2685 - 4341
E-ISSN: 2685 - 5313

Sistem Monitoring dan Kontrol Generator Set LPG Berbasis *Internet of Things* dengan Validasi *Hash Blockchain*

IoT-Based LPG-fueled Generator Set Monitoring and Control System with Blockchain Hash Validation

Rey Gasta Isrofiasrory[1], Arya Kusumawardana[2], Agil Ziddan Achmad[3], Muhammad Ihsanul Rizqi[4]
[1,2,3,4] Program Studi D4 Teknologi Rekayasa Pembangkit Energi, Universitas Negeri Malang
[1,2,3,4] Jl. Semarang No.5, Sumbersari, Kec. Lowokwaru, Kota Malang, Provinsi Jawa Timur, Indonesia
email: [*1]rey.gasta.2205317@students.um.ac.id, [2]arya.kusumawardana.ft@um.ac.id,
[3]agil.ziddan.2205317@students.um.ac.id, [4]muhammad.ihsanul.2205317@students.um.ac.id

Informasi Artikel
Diajukan, 16 Juni 2025
Diterima, 31 Maret 2026
Diterbitkan, 5 Juni 2026

Kata Kunci :
Monitoring, Generator set LPG, *Internet of Things*, *blockchain*, ESP32, *Firebase*

ABSTRAK

Sistem monitoring genset konvensional yang mengandalkan indikator panel fisik tidak memiliki kemampuan akses jarak jauh dan mekanisme verifikasi integritas data historis. Penelitian ini merancang sistem monitoring dan kontrol generator set berbahan bakar LPG berbasis *Internet of Things* dengan validasi integritas data menggunakan teknologi *blockchain*. Arsitektur sistem terdiri dari modul PZEM-004T yang terhubung ke mikrokontroler ESP32-S3 untuk akuisisi parameter kelistrikan, *Firebase Realtime Database* untuk penyimpanan data *real-time*, Raspberry Pi sebagai *gateway* yang menghitung *hash* SHA-256 dan menyimpannya pada *smart contract* Ethereum Sepolia, serta dashboard berbasis React.js untuk visualisasi dan kontrol jarak jauh melalui mekanisme relay. Pengujian terhadap 10 sampel menunjukkan rata-rata error sensor 1,76% untuk tegangan, 0,27% untuk arus, dan 0,71% untuk frekuensi. *Response time* kontrol jarak jauh tercatat rata-rata 1,5 detik dengan tingkat keberhasilan 100%. *Latency* pengiriman data *Firebase* rata-rata 733,6 ms dan *REST API* rata-rata 206,3 ms. Mekanisme validasi *hash blockchain* mendeteksi seluruh skenario manipulasi data dengan waktu validasi rata-rata 1,38 detik. Keterbatasan penelitian ini mencakup pengujian yang dilakukan pada jaringan testnet dengan kondisi trafik relatif stabil.

Keyword :
LPG-fueled generator set, Monitoring, Internet of Things, blockchain, ESP32, *Firebase*

ABSTRACT

Conventional generator monitoring systems that rely on physical panel indicators lack remote access capabilities and mechanisms for verifying historical data integrity. This study designed an LPG-fueled generator monitoring and control system based on the Internet of Things with data integrity validation using blockchain technology. The system architecture consists of a PZEM-004T module connected to an ESP32-S3 microcontroller for electrical parameter acquisition, *Firebase Realtime* Database for real-time data storage, Raspberry Pi as a gateway that calculates SHA-256 hash and stores it on Ethereum Sepolia smart contract, and a React.js-based dashboard for visualization and remote control through relay mechanisms. Testing on 10 samples showed average sensor errors of 1.76% for voltage, 0.27% for current, and 0.71% for frequency. Remote control response time averaged 1.5 seconds with a 100% success rate. *Firebase* data transmission latency averaged 733.6 ms and REST API averaged 206.3 ms. The blockchain hash validation mechanism detected all data manipulation scenarios with an average validation time of 1.38 seconds. The limitations of this study include testing conducted on a testnet network with relatively stable traffic conditions.

1. PENDAHULUAN

Ketersediaan energi listrik yang stabil merupakan kebutuhan operasional bagi berbagai sektor. Data pemadaman di Penyulang SRN 02 Surakarta menunjukkan rata-rata 7,26 kali gangguan per pelanggan per tahun [1], kondisi yang mendorong penggunaan generator set (genset) sebagai sumber daya cadangan. Genset berbahan bakar *Liquefied Petroleum Gas* (LPG) memiliki karakteristik efisiensi termal 80,8% dan *Specific Fuel Consumption* (SFC) 0,9481 kg/kW.jam, berbeda dengan diesel Dexlite yang mencapai efisiensi 64,03% dengan SFC 1,3089 kg/kW.jam [2]. Dari aspek emisi, genset LPG menghasilkan CO sebesar 0,073% dan CO_2 sebesar 6,66%, sedangkan genset pertalite menghasilkan CO 0,81% dan CO_2 8,56% [3]. Konsumsi bahan bakar LPG rata-rata 0,12 kg per pengujian dibandingkan pertalite sebesar 0,15 kg, dengan biaya perawatan yang lebih rendah karena karakteristik pembakaran LPG yang tidak mengangkat minyak dari dinding silinder [4].

Meskipun genset LPG memiliki karakteristik tertentu dari sisi efisiensi dan emisi, sistem monitoring genset konvensional masih mengandalkan indikator panel fisik tanpa kemampuan akses jarak jauh dan tanpa manajemen data historis. Ketiadaan pemantauan *real-time* menyebabkan parameter kelistrikan seperti tegangan, arus, dan frekuensi tidak terpantau secara kontinu, yang berpotensi menimbulkan *downtime* tak terduga [5]. Di sisi lain, data monitoring yang tersimpan secara konvensional rentan terhadap manipulasi dan tidak memiliki mekanisme verifikasi integritas, sehingga keandalan data historis untuk keperluan analisis pemeliharaan tidak dapat dijamin.

Implementasi sistem monitoring berbasis IoT dengan mikrokontroler ESP32 dan modul PZEM-004T yang mengakuisisi parameter tegangan, arus, daya, dan frekuensi telah diuji dalam berbagai penelitian. Akurasi pembacaan sensor dilaporkan mencapai hampir 99% pada jarak 150 meter kondisi *Non-Line of Sight* dengan RSSI -107 dBm [6]. Penggunaan *Firebase* sebagai *database real-time* menghasilkan error 1,71% dengan akurasi 98,29% [7]. Protokol ESP-NOW menunjukkan kemampuan transmisi hingga 600 meter dalam kondisi *Line of Sight* [8], sedangkan delay komunikasi Modbus TCP rata-rata 225 ms dengan tingkat keberhasilan 100% [9]. *Firebase Realtime* Database terbukti dapat menyinkronkan data dengan delay transmisi rata-rata 0,62 detik pada sistem pendeteksi kebakaran [10]. Sistem monitoring berbasis ESP32 dan *Firebase* terbukti dapat mengirimkan data sensor secara stabil, dengan dashboard React.js yang memanfaatkan virtual DOM untuk *respons real-time* [11], [12].

Permasalahan integritas dan keamanan data merupakan tantangan tersendiri dalam sistem IoT berbasis *cloud*. Teknologi *blockchain* dapat berfungsi sebagai *immutable ledger* untuk mengatasi masalah *single point of failure* dengan fitur *tamper resistance* [13]. Survei terhadap *blockchain-based Digital Twins* menyimpulkan bahwa *blockchain* dapat memastikan data *immutability*, *confidentiality*, dan *integrity* pada sistem industri berbasis IoT [14]. Integrasi *blockchain* dengan deep learning untuk keamanan data sensor IoT menunjukkan tingkat akurasi deteksi mencapai 97,39% [15]. Namun demikian, penelitian-penelitian tersebut belum secara spesifik mengintegrasikan *blockchain* pada sistem monitoring genset yang membutuhkan waktu respon cepat dan kontrol jarak jauh.

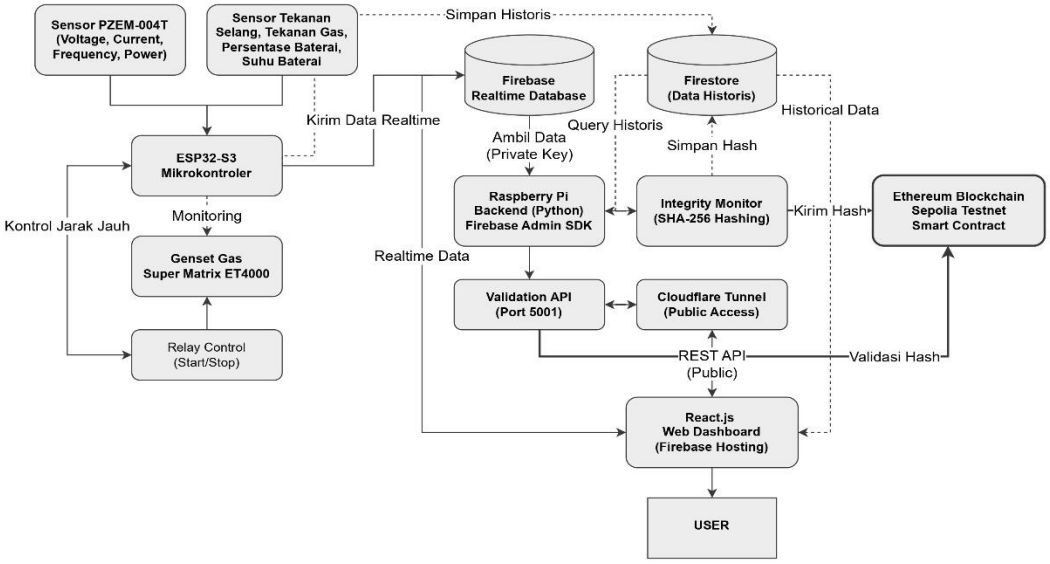
Kajian-kajian tersebut belum secara spesifik mengkaji integrasi mekanisme verifikasi integritas data berbasis kriptografi pada sistem monitoring *generator set*. Oleh karena itu, penelitian ini merancang dan memvalidasi sistem monitoring genset berbahan bakar gas LPG dengan arsitektur ESP32-S3, modul pengukuran PZEM-004T, Raspberry Pi, dan *Firebase* guna memantau parameter kelistrikan meliputi tegangan, arus, frekuensi, dan daya secara jarak jauh melalui *dashboard* berbasis web, yang dilengkapi mekanisme pencatatan *hash* SHA-256 *pada smart contract* Ethereum sebagai bukti otentikasi data pemantauan. Sistem ini juga menyediakan kendali ON/OFF genset secara *remote* melalui mekanisme relay yang terhubung ke *Firebase*. Validasi sistem dilakukan melalui *black box testing*, pengujian akurasi modul pengukuran, serta evaluasi kinerja *real-time* yang mencakup *response time*, *latency*, dan delay notifikasi.

2. METODE PENELITIAN

Pengembangan sistem menggunakan pendekatan *Research and Development* (R&D) melalui empat tahapan meliputi *Define*, *Design*, *Develop*, dan *Validate*. Pada tahap *Validate*, dilakukan pengujian terhadap fungsionalitas aplikasi, akurasi pembacaan modul PZEM-400T, performa sistem secara *real-time*, serta verifikasi integritas data berbasis *blockchain*.

2.1. Arsitektur Sistem

Sistem monitoring mengintegrasikan delapan komponen utama sebagaimana ditunjukkan pada Gambar 1. modul PZEM-004T membaca parameter listrik berupa tegangan, arus, frekuensi, dan daya. Seluruh data dikirimkan ke mikrokontroler ESP32-S3 melalui komunikasi serial UART. ESP32-S3 berfungsi sebagai *gateway* yang mentransmisikan data ke *Firebase Realtime* Database untuk monitoring dan ke Firestore untuk penyimpanan data historis [11]. Meskipun sistem mendukung pemantauan berbagai parameter, penelitian ini membatasi fokus pengujian dan validasi pada parameter kelistrikan yang diakuisisi oleh modul PZEM-004T meliputi tegangan, arus, frekuensi, dan daya.

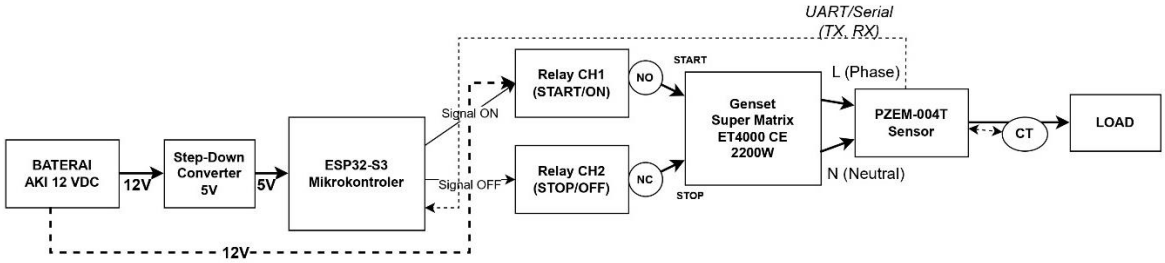


Gambar 1. Arsitektur Sistem Monitoring Genset dengan Integrasi *Blockchain*.

Raspberry Pi backend menjalankan Python script yang terhubung ke *Firebase* melalui Admin SDK dengan dua fungsi yaitu mengelola logika kontrol relay untuk menerima perintah START/STOP, dan menjalankan modul *Integrity Monitor* yang menghasilkan *hash* SHA-256 dari setiap data parameter. *Hash* tersebut dikirimkan ke *smart contract* pada jaringan Ethereum Sepolia Testnet untuk disimpan sebagai bukti integritas data. *Dashboard* web berbasis React.js di-deploy pada *Firebase* Hosting, mengambil data *real-time* dari *Firebase* dan melakukan validasi *hash* melalui *Validation API* yang diekspos ke publik menggunakan Cloudflare Tunnel [12].

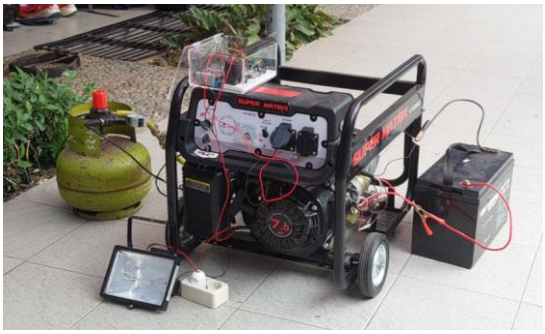
2.2. Perancangan *Hardware*

Sistem kontrol genset dibagi menjadi tiga bagian yaitu pengendali hardware, server penghubung, dan *interface dashboard*. Blok diagram hardware ditunjukkan pada Gambar 2. Catu daya ESP32-S3 berasal dari aki 12 VDC yang diturunkan menjadi 5 VDC menggunakan DC-DC *converter*, sedangkan relay menggunakan 12 VDC langsung dari baterai. Pemisahan sumber daya ini dilakukan karena relay membutuhkan arus lebih besar untuk menggerakkan kontak mekanis.



Gambar 2. Blok Diagram Hardware Sistem Monitoring dan Kontrol Genset.

ESP32-S3 dikonfigurasikan dengan dua pin GPIO *output* yaitu GPIO14 (*RELAY_START_PIN*) mengontrol Relay *Channel* 1 yang terhubung ke terminal *starter* genset dengan konfigurasi *Normally Open* (NO), dan GPIO15 (*RELAY_OFF_PIN*) mengontrol Relay Channel 2 dengan konfigurasi *Normally Closed* (NC) yang berfungsi untuk mematikan mesin dan membuka kontak agar starter dapat berfungsi. Konfigurasi NC pada relay *shutdown* memastikan genset tetap dapat dimatikan meskipun ESP32-S3 mengalami kegagalan daya. Gambar 3 menunjukkan implementasi sistem yang terdiri dari genset LPG Super Matrix ET4000 CE 2200W, tabung LPG, baterai, dan perangkat IoT yang terintegrasi pada satu kesatuan. Genset yang digunakan memiliki kapasitas daya maksimum 2200W dengan tegangan output 220V AC.



Gambar 3. Implementasi Sistem Monitoring dan Kontrol Genset LPG.

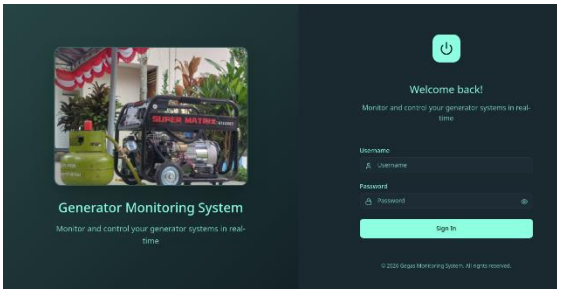
Sistem menggunakan konsep perintah tunggal melalui *Firebase* node /starter_engine/master_control yang menerima perintah "*ON*" atau "*OFF*". ESP32-S3 memeriksa *Firebase* setiap 1 detik untuk mendeteksi perubahan perintah. Interval polling 1 detik dipilih sebagai keseimbangan antara kecepatan respons dan efisiensi penggunaan bandwidth jaringan. Relay Channel 1 pada GPIO14 terhubung ke terminal *starter* mesin dengan konfigurasi *Normally Open*, sedangkan Relay Channel 2 pada GPIO15 terhubung ke sirkuit mematikan mesin dengan konfigurasi *Normally Closed*. Ketika perintah berubah dari "*OFF*" menjadi "*ON*", ESP32-S3 menjalankan urutan yaitu mengaktifkan GPIO15 dengan sinyal *LOW* untuk melepas Relay Channel 2, kemudian mengaktifkan GPIO14 dengan sinyal *LOW* untuk menghidupkan starter motor selama maksimal 3 detik yang dikontrol oleh *timer* pengaman. Setelah 3 detik, GPIO14 dimatikan dengan sinyal *HIGH* untuk mencegah kerusakan starter. Mekanisme *timer* pengaman ini mencegah starter bekerja terlalu lama yang dapat menyebabkan *overheat* pada motor *starter*.

Modul PZEM-004T dipasang pada output genset untuk mengukur parameter listrik. Modul terhubung ke ESP32-S3 melalui pin GPIO16 (RX) dan GPIO17 (TX) dengan kecepatan komunikasi serial 9600 baud sesuai spesifikasi modul [16]. Data pembacaan dikirim setiap detik ke *Firebase Realtime Database* dan ke REST API server untuk penyimpanan data historis. Pengiriman data secara paralel ke dua endpoint memastikan redundansi penyimpanan sehingga data tetap tersedia meskipun salah satu layanan mengalami gangguan.

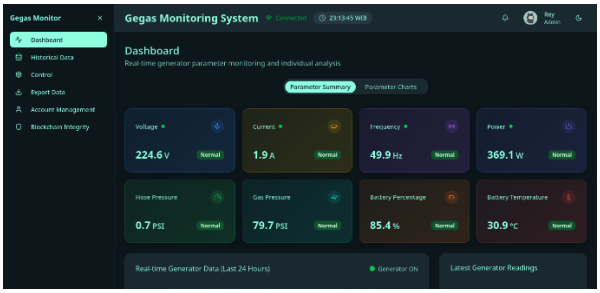
2.3. Perancangan Software

2.3.1 Dashboard Web

Aplikasi web dirancang menggunakan React.js dengan TypeScript dan di-deploy pada *Firebase Hosting*. Halaman *login* sebagaimana ditampilkan pada Gambar 4 menggunakan autentikasi *Firebase* yang memvalidasi *username* dan *password* pengguna. *Dashboard* utama mendukung dua mode tampilan yaitu *dark mode* sebagaimana ditunjukkan pada Gambar 5 yang menampilkan delapan parameter monitoring *real-time* meliputi tegangan, arus, frekuensi, daya, tekanan selang, tekanan gas, persentase baterai, dan temperatur baterai.

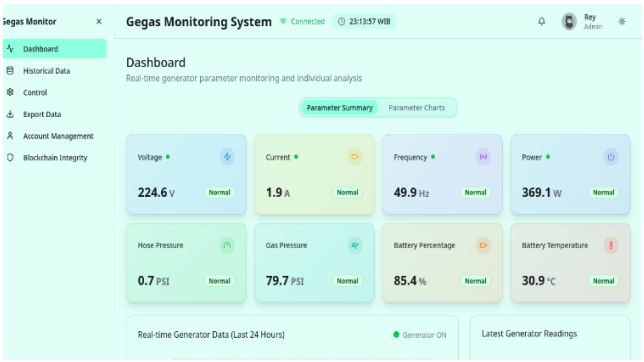


Gambar 4. Halaman *Login* Sistem Monitoring

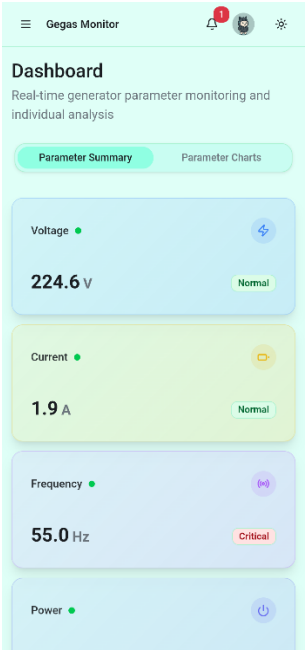


Gambar 5. *Dashboard Dark Mode*

Dashboard juga tersedia dalam *light mode* sebagaimana ditunjukkan pada Gambar 6 yang memberikan opsi tampilan dengan latar belakang terang untuk kondisi pencahayaan yang berbeda. Indikator status visual menggunakan *color coding* yang konsisten pada kedua mode yaitu hijau untuk kondisi Normal yang mengindikasikan parameter berada dalam rentang aman, merah untuk kondisi *Critical* yang memerlukan perhatian segera, dan kuning untuk kondisi Warning sebagai peringatan dini sebelum mencapai batas kritis. *Interface* dirancang responsif untuk perangkat *smartphone* sebagaimana ditunjukkan pada Gambar 7 sehingga pengguna dapat memantau kondisi genset dari berbagai perangkat dengan seluruh fitur yang tetap dapat diakses.

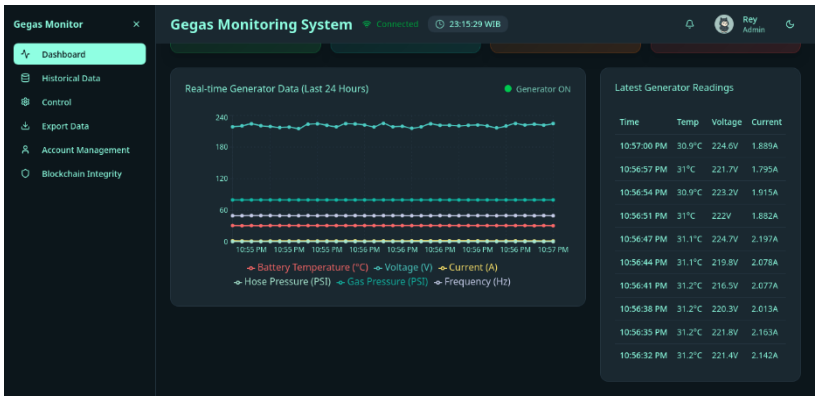


Gambar 6. *Dashboard Light Mode*

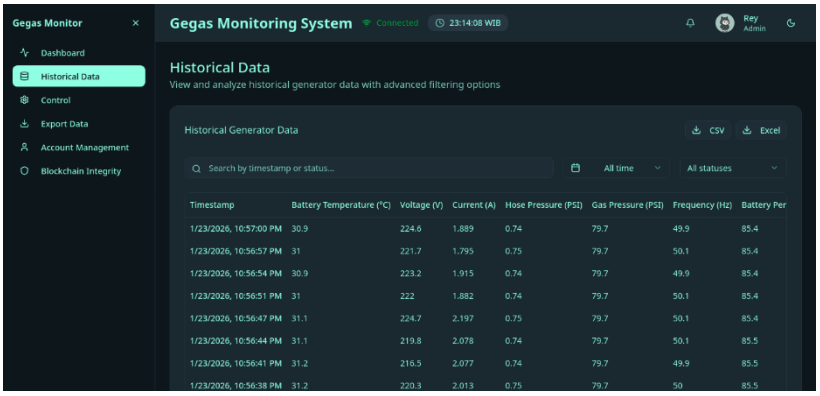


Gambar 7. *Dashboard* Tampilan *Smartphone*

Fitur *toggle Parameter Summary* dan *Parameter Charts* sebagaimana ditunjukkan pada Gambar 8 digunakan untuk analisis data dari perspektif berbeda. Fitur *Historical Data* pada Gambar 9 menampilkan tabel dengan *search bar* dan *dropdown* filter waktu meliputi *All time*, *Last 7 Days*, dan *Last 30 Days*.



Gambar 8. Tampilan *Parameter Summary* dan *Parameter Charts*

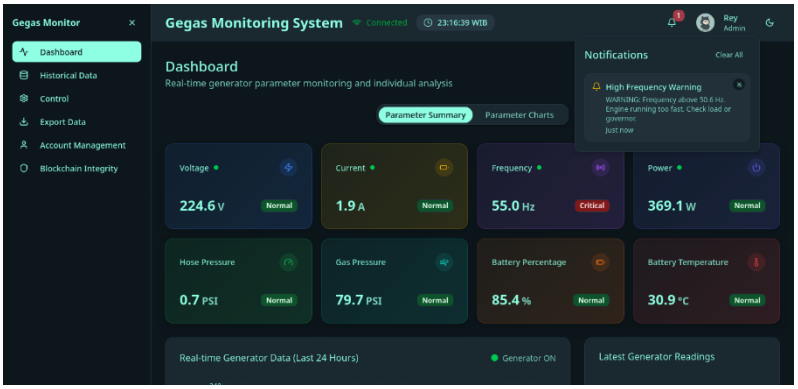


Gambar 9. Halaman *Historical Data*

Setiap parameter divisualisasikan melalui grafik *real-time* dengan *threshold lines* berwarna sebagaimana ditunjukkan pada Gambar 10 untuk memudahkan identifikasi visual. Panel notifikasi *real-time* pada

Gambar 11 menggunakan *Firebase Cloud Messaging* untuk *alert* seperti *Overload Alert* dan *Low Frequency Warning*.

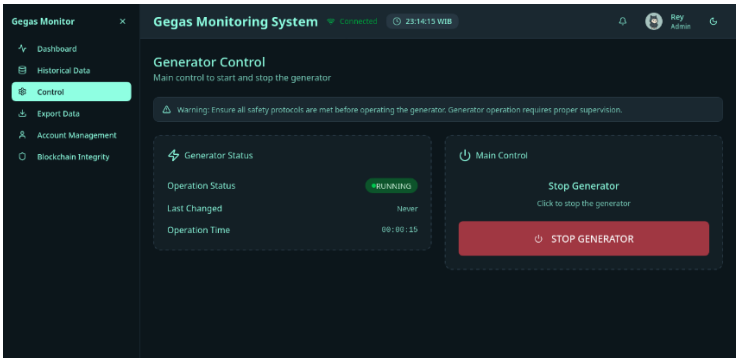
Gambar 10. Visualisasi Grafik *Real-Time* dengan *Threshold Lines*



Gambar 11. Tampilan Notifikasi *Real-Time*

Halaman *Export Data* sebagaimana ditunjukkan pada Gambar 12 menyediakan fitur pengunduhan data historis dengan tiga opsi rentang waktu yaitu *Complete Dataset* untuk mengambil seluruh data yang tersimpan, *Last 7 Days* untuk data satu minggu terakhir, dan *Last 30 Days* untuk data satu bulan terakhir. Format file yang didukung meliputi CSV untuk kompatibilitas dengan berbagai aplikasi spreadsheet dan Excel untuk pengguna yang memerlukan format *native* Microsoft Office. Halaman *Generator Control* sebagaimana ditunjukkan pada Gambar 13 menampilkan *interface* minimalis yang terdiri dari tombol *START* untuk menghidupkan genset dan tombol *STOP* untuk mematikan genset. Kedua tombol tersebut terhubung dengan *Firebase* node */starter_engine/master_control* yang menerima perintah dan meneruskannya ke ESP32-S3 untuk dieksekusi oleh relay pada sisi *hardware*.

Gambar 12. Halaman *Export Data*

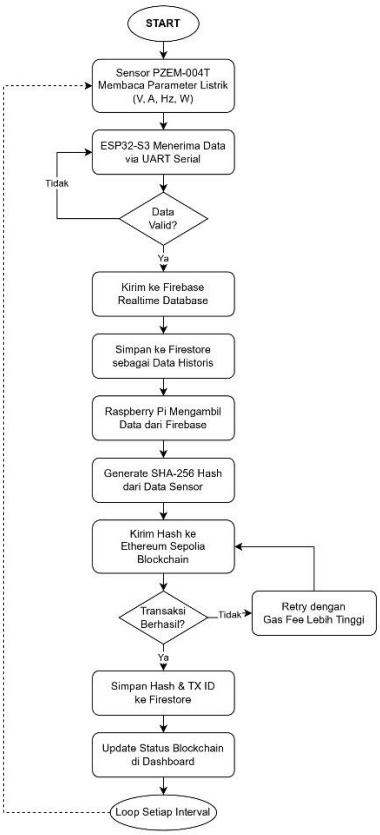


Gambar 13. Halaman *Generator Control*

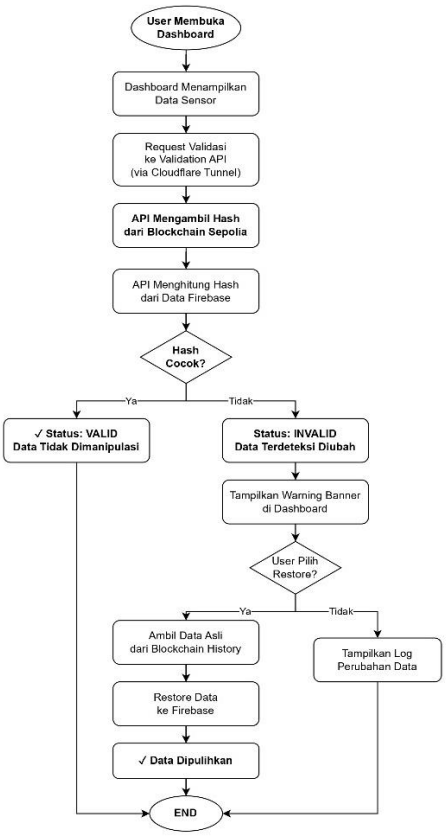
2.3.2 Mekanisme Integritas Data *Blockchain*

Alur akuisisi data dan penyimpanan ke *blockchain* ditunjukkan pada Gambar 14. Proses dimulai ketika modul PZEM-004T membaca parameter listrik dan mengirimkannya ke ESP32-S3 melalui UART serial. ESP32-S3 memvalidasi data dan mengirimkannya ke *Firebase Realtime Database* serta menyimpan ke Firestore sebagai data historis. Raspberry Pi menjalankan modul *Integrity Monitor* yang secara periodik mengambil data parameter dari *Firebase*, kemudian menghasilkan *hash* SHA-256 dari setiap record data. *Hash* tersebut dikirimkan ke *smart contract* pada jaringan Ethereum Sepolia Testnet melalui transaksi *blockchain*. Jika transaksi gagal karena *gas fee* tidak mencukupi, sistem melakukan *retry* dengan *gas fee* lebih tinggi. Setelah transaksi berhasil, *hash* beserta *Transaction ID* disimpan ke Firestore dan status *blockchain* diperbarui pada dashboard.

Alur validasi integritas data ditunjukkan pada Gambar 15. Ketika pengguna membuka dashboard, sistem menampilkan data parameter sekaligus melakukan *request* validasi ke Validation API melalui Cloudflare Tunnel. API mengambil *hash* yang tersimpan di *blockchain* Sepolia, kemudian menghitung ulang *hash* dari data di *Firebase*. Jika *hash* cocok, status ditampilkan sebagai Valid. Jika hash tidak cocok, status ditampilkan sebagai *Invalid* disertai *warning banner*. Pengguna dapat memilih opsi *Restore* untuk mengambil data asli dari *blockchain history* dan mengembalikannya ke *Firebase*.

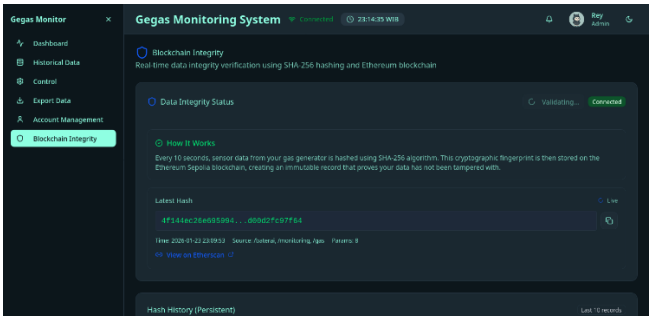


Gambar 14. *Flowchart* Alur Akuisisi Data ke *Blockchain*

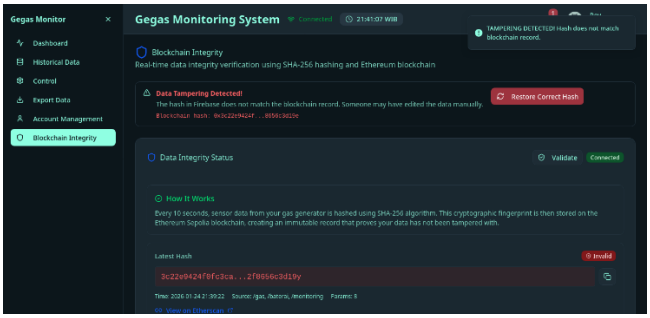


Gambar 15. *Flowchart* Alur Validasi Integritas Data

Halaman *Blockchain Integrity* sebagaimana ditunjukkan pada Gambar 16 menampilkan status validasi *hash* data modul PZEM-400T terhadap *blockchain* Ethereum Sepolia. Ketika *hash* valid, sistem menampilkan *badge* hijau *Valid* beserta informasi *Transaction ID* yang dapat diverifikasi melalui Etherscan. Jika terdeteksi manipulasi data sebagaimana ditunjukkan pada Gambar 17, sistem menampilkan *badge* merah Invalid disertai *warning banner* yang menginformasikan adanya ketidaksesuaian antara *hash* di *Firebase* dengan *hash* yang tersimpan di *blockchain*. Pengguna dapat menekan tombol *Restore from Blockchain* untuk mengembalikan data asli.

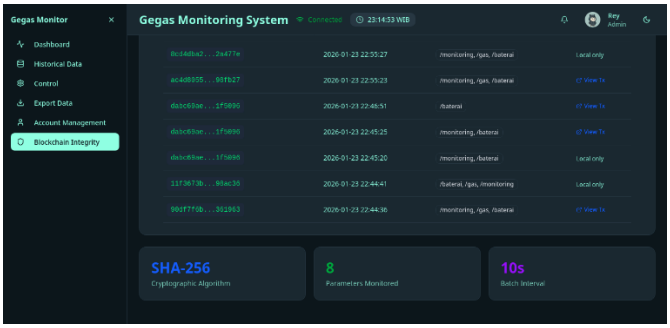


Gambar 16. Halaman Validasi Integritas Data Status *Valid*

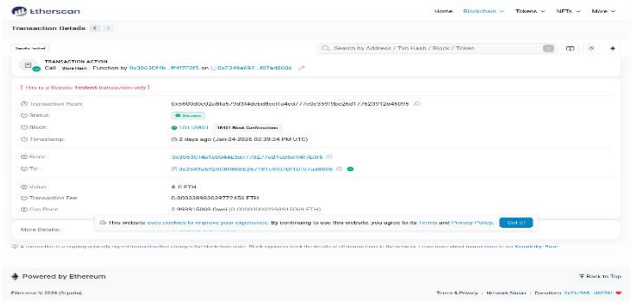


Gambar 17. Status *Invalid* dengan *Warning Banner*

Riwayat *hash* yang tersimpan di *blockchain* dapat dilihat pada bagian *Hash History* sebagaimana ditunjukkan pada Gambar 18 yang menampilkan daftar hash beserta *timestamp* penyimpanan. Setiap transaksi dapat diverifikasi secara publik melalui Etherscan sebagaimana ditunjukkan pada Gambar 19 yang memuat informasi *Transaction Hash*, *Block Number*, dan *Input Data* berisi *hash* SHA-256 dari data parameter modul PZEM-400T.



Gambar 18. Riwayat *Hash* yang Tersimpan di *Blockchain*



Gambar 19. Tampilan Transaksi *Hash* pada Etherscan Sepolia

2.4. Metode Pengujian

Validasi sistem dilakukan melalui lima kategori pengujian untuk memverifikasi kesesuaian antara spesifikasi desain dan hasil implementasi

2.4.1 *Black Box Testing*

Pengujian aplikasi web dilakukan menggunakan metode *black box testing* terhadap 20 *test case* yang mencakup modul *Login*, Dashboard, *Historical Data*, *Control*, *Export Data, Account Management*, dan Notifikasi. Pendekatan ini fokus pada validasi *input* dan *output* sistem tanpa memperhatikan implementasi internal kode program, dengan target tingkat keberhasilan 100%.

2.4.2 Pengujian Akurasi Modul PZEM-400T

Akurasi pembacaan modul PZEM-004T divalidasi dengan membandingkan hasilnya terhadap multimeter Sanwa RD700 sebagai alat ukur referensi. Pengujian dilakukan melalui 10 percobaan untuk setiap parameter dengan batas toleransi error ≤5% untuk tegangan dan arus, serta ≤1% untuk frekuensi. Toleransi frekuensi ditetapkan lebih ketat karena *deviasi* frekuensi berkorelasi langsung dengan kondisi pembebanan genset. Perhitungan persentase error mengacu pada persamaan (1).

$$Error(\%) = \frac{X_{sensor} - X_{referensi}}{X_{referensi}} \times 100 \qquad (1)$$

2.4.3 Pengujian *Response Real-time*

Evaluasi kinerja *real-time* mencakup empat parameter. *Response time* kontrol jarak jauh diukur dari saat perintah *START/STOP* dikirim hingga eksekusi hardware terjadi, dengan target ≤5 detik. Validasi status dilakukan melalui pembacaan tegangan *output* genset. *Latency* pengiriman data ke *Firebase* diukur dengan target ≤1000 ms menggunakan pencatatan waktu pada ESP32-S3. *Latency* REST API untuk komunikasi dengan *backend* Raspberry Pi diukur dengan target serupa. Delay notifikasi alarm diuji dengan memasukkan parameter anomali ke *Firebase* dan mengukur selisih waktu hingga notifikasi muncul pada antarmuka web, dengan target ≤2 detik.

2.4.4 Pengujian Sinkronisasi Data

Konsistensi data diverifikasi dengan membandingkan nilai tegangan dan frekuensi antara *Serial Monitor* ESP32-S3, *Firebase Realtime Database*, dan tampilan *dashboard*. Ketiga sumber data harus menampilkan nilai identik karena dashboard berfungsi sebagai *layer* visualisasi langsung dari *Firebase* tanpa pengolahan tambahan.

2.4.5 Pengujian Verifikasi *Hash Blockchain*

Mekanisme integritas data berbasis *blockchain* diuji melalui dua skenario. Pengujian validasi integritas dilakukan sebanyak 10 percobaan, terdiri dari 5 skenario data normal dan 5 skenario data manipulasi. Pada skenario normal, *hash* yang tersimpan di *Firebase* diverifikasi kesesuaiannya dengan *hash* pada *blockchain* Sepolia. Pada skenario manipulasi, parameter modul PZEM-400T diubah secara lokal untuk menghasilkan *hash* berbeda, kemudian diverifikasi terhadap *blockchain* untuk memastikan sistem mengembalikan status *Invalid*.

Pengujian transaksi *blockchain* mengambil 10 transaksi terakhir dari *smart contract* yang di-deploy pada jaringan Ethereum Sepolia. Parameter yang dicatat meliputi *hash* data, *Transaction ID, gas used*, dan waktu konfirmasi. Seluruh transaksi dapat diverifikasi secara publik melalui Etherscan.

3. HASIL DAN PEMBAHASAN

Pengujian meliputi akurasi pembacaan modul PZEM-004T pada parameter tegangan, arus, dan frekuensi, fungsionalitas aplikasi web, kinerja transmisi data ke *Firebase* dan *blockchain* sinkronisasi data antarmuka, serta verifikasi integritas data berbasis *blockchain*.

3.1. Pengujian Black Box Testing

Pengujian fungsionalitas dilakukan terhadap 20 *test case* yang mencakup modul *Login, Dashboard, Historical Data, Control, Export Data, Account Management,* dan Notifikasi dengan target keberhasilan 100%. Hasil pengujian ditampilkan pada Tabel 1.

Tabel 1. Hasil Pengujian *Black Box Testing*

No	Deskripsi Kasus Uji	Hasil yang Diharapkan	Status
1	Login Valid	Dashboard tampil	Berhasil
2	Login Gagal	Pesan error	Berhasil
3	*Logout*	Kembali ke login	Berhasil
4	Akses Dashboard	Parameter tampil	Berhasil
5	Status threshold	Warna & teks sesuai	Berhasil

No	Deskripsi Kasus Uji	Hasil yang Diharapkan	Status
6	Update grafik	Data *real-time*	Berhasil
7	Update tabel	Data terbaru	Berhasil
8	Pagination	20 Data/halaman	Berhasil
9	Filter 24 jam	Data sesuai	Berhasil
10	Filter kombinasi	Data sesuai	Berhasil
11	Start sistem	Status Running	Berhasil
12	Stop sistem	Status Stopped	Berhasil
13	Sinkron Status	Update Otomatis	Berhasil
14	Ekspor semua data	CSV berisi semua data	Berhasil
15	Ekspor data 30 hari	CSV berisi data 30 hari	Berhasil
16	Ekspor tanpa data	Error data tidak tersedia	Berhasil
17	Info akun	Data profile tampil	Berhasil
18	Pengaturan notifikasi	Tersimpan	Berhasil
19	Alert kritis	Badge merah	Berhasil
20	Hapus notifikasi	Notifikasi kosong	Berhasil

Hasil *black box testing* mencatat tingkat keberhasilan 100% dari 20 test case, mengindikasikan bahwa integrasi React.js-*Firebase* berfungsi normal dan tidak ditemukan *bug* kritis.

3.2. Pengujian Akurasi Tegangan

Pengujian akurasi modul pengukuran tegangan dilakukan dengan membandingkan hasilnya terhadap multimeter melalui 10 percobaan dengan batas toleransi error ≤5% sebagaimana ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Akurasi Tegangan

No	PZEM-400T (V)	Data Multimeter (V)	Error (%)	Status
1	228,1	224	1,83	Berhasil
2	228,2	224	1,88	Berhasil
3	228,1	224	1,83	Berhasil
4	228,0	224	1,79	Berhasil
5	228,6	225	1,60	Berhasil
6	228,6	225	1,60	Berhasil
7	228,6	225	1,60	Berhasil
8	228,6	225	1,60	Berhasil
9	228,7	225	1,64	Berhasil
10	228,9	224	2,19	Berhasil

Hasil pengujian akurasi tegangan menunjukkan rata-rata error 1,76% dengan keberhasilan 100% dari 10 percobaan sesuai batas error ≤5%. Hasil ini sejalan dengan penelitian yang mencapai akurasi 99% dengan error 1% dan akurasi tegangan 99,25% pada kondisi beban dinamis [6], [17].

3.3. Pengujian Akurasi Arus

Pengujian akurasi modul pengukuran arus menggunakan *clamp meter* sebagai alat ukur referensi dengan batas error ≤5%. Rincian hasil pengujian ditampilkan pada Tabel 3.

Tabel 3. Hasil Pengujian Akurasi Arus

No	PZEM-400T (A)	*Clamp Meter* (A)	Error (%)	Status
1	1,9	1,9	0,00	Berhasil
2	3,5	3,5	0,00	Berhasil
3	1,6	1,6	0,00	Berhasil
4	2,0	2,0	0,00	Berhasil
5	0,1	0,1	0,00	Berhasil
6	2,1	2,1	0,00	Berhasil
7	4,0	4,0	0,00	Berhasil
8	2,1	2,1	0,00	Berhasil
9	3,6	3,7	2,70	Berhasil
10	0,1	0,1	0,00	Berhasil

Hasil pengujian menunjukkan rata-rata error sebesar 0,27% dengan status berhasil pada 10 dari 10 percobaan (100%), sejalan dengan penelitian yang mencatat akurasi modul pengukuran arus 99,82% [17].

3.4. Pengujian Akurasi Frekuensi

Pengujian frekuensi menggunakan multimeter dengan batas keberhasilan error ≤1%, lebih ketat karena frekuensi merupakan parameter kunci stabilitas genset. Data hasil pengujian disajikan pada Tabel 4.

Tabel 4. Hasil Pengujian Akurasi Frekuensi

No	PZEM-400T (Hz)	Multimeter (Hz)	Error (%)	Status
1	50,80	50,28	1,03	Gagal
2	50,7	50,50	0,40	Berhasil
3	50,6	50,88	0,55	Berhasil
4	50,9	50,46	0,87	Berhasil
5	50,7	51,17	0,92	Berhasil
6	50,6	50,13	0,94	Berhasil
7	50,6	51,09	0,96	Berhasil
8	50,8	50,93	0,26	Berhasil
9	50,9	51,01	0,22	Berhasil
10	50,7	51,19	0,96	Berhasil

Hasil pengujian menunjukkan rata-rata error sebesar 0,71% dengan status berhasil pada 9 dari 10 percobaan (90%). Satu percobaan yang gagal memiliki error 1,03% yang sedikit melampaui batas toleransi 1%, kemungkinan disebabkan oleh fluktuasi beban pada saat pengukuran.

3.5. Pengujian *Response Time*

Pengujian *response time* dilakukan menggunakan program pada ESP32-S3 yang mencatat waktu pengiriman perintah dan waktu eksekusi hardware secara otomatis. Status genset divalidasi melalui pembacaan tegangan, dengan nilai terdeteksi saat *ON* dan 0V saat *OFF*, sebagaimana data lengkap pengujian disajikan pada Tabel 5.

Tabel 5. Hasil Pengujian *Response Time*

No	Perintah	Waktu Kirim	Waktu Eksekusi	*Response* (detik)	Status
1	ON Genset	22:52:25	22:52:26	1	Berhasil
2	OFF Genset	22:52:45	22:52:47	2	Berhasil
3	ON Genset	22:58:00	22:58:01	1	Berhasil
4	OFF Genset	22:58:16	22:58:18	2	Berhasil
5	ON Genset	23:06:10	23:06:11	1	Berhasil
6	OFF Genset	23:06:20	23:06:22	2	Berhasil
7	ON Genset	23:09:01	23:09:02	1	Berhasil
8	OFF Genset	23:09:50	23:09:52	2	Berhasil
9	ON Genset	23:14:06	23:14:07	1	Berhasil
10	OFF Genset	23:14:25	23:14:27	2	Berhasil

Pengujian *response time* dari aplikasi web hingga eksekusi hardware mencapai rata-rata 1,5 detik dengan keberhasilan 100% (10/10 percobaan berhasil, target ≤5 detik). Capaian ini sebanding dengan studi yang mencatat delay 225 ms dan delay 0,62 detik, di mana variasi dapat disebabkan oleh kompleksitas sistem dan *bandwidth* jaringan [9], [10].

3.6. Pengujian *Latency*

Pengujian *latency* dilakukan menggunakan program pada ESP32-S3 yang mencatat waktu pengiriman dan penerimaan data, kemudian menampilkan hasilnya melalui *Serial Monitor*. Nilai *latency* yang diperoleh untuk setiap pengiriman data dirangkum pada Tabel 6.

Tabel 6. Hasil Pengujian *Latency* Pengiriman Data

No	*Firebase* (ms)	*REST API* (ms)	Status
1	872	425	Berhasil
2	715	103	Berhasil
3	648	107	Berhasil
4	754	591	Berhasil
5	757	346	Berhasil
6	917	101	Berhasil
7	651	118	Berhasil
8	806	152	Berhasil
9	650	105	Berhasil
10	566	115	Berhasil

Hasil pengujian *latency Firebase* mencapai rata-rata 733,6 ms dengan keberhasilan 100% dari 10 percobaan, sedangkan *REST API* mencapai rata-rata 206,3 ms dengan keberhasilan 90%. Satu kegagalan pada *REST API* kemungkinan disebabkan oleh *network congestion* pada saat pengujian berlangsung. Perbedaan *latency* antara *Firebase* dan *REST API* disebabkan oleh *overhead* protokol *Firebase Realtime* Database yang melakukan *handshake WebSocket* dan sinkronisasi *state*, sedangkan *REST API* menggunakan protokol HTTP sederhana tanpa mekanisme sinkronisasi tambahan. *Latency* di bawah 1000 ms masih dapat diterima untuk monitoring IoT dan delay pengiriman data di bawah 1 detik memenuhi kriteria operasi *real-time* [8], [10]. Berdasarkan hasil tersebut, kedua metode pengiriman data memenuhi batas toleransi yang ditetapkan.

3.7. Pengujian *Delay* Notifikasi

Pengujian *delay* notifikasi dilakukan secara manual dengan memasukkan parameter anomali ke *Firebase*, kemudian mencatat selisih waktu antara deteksi pada *database* dan munculnya notifikasi pada antarmuka web menggunakan *timer*. Skenario pengujian mencakup berbagai jenis *alert* meliputi tegangan, arus, frekuensi, daya, tekanan gas, tekanan selang, temperatur baterai, dan kapasitas baterai dengan target delay ≤2 detik. Hasil pengujian selengkapnya ditampilkan pada Tabel 7.

Tabel 7. Hasil Pengujian *Delay* Notifikasi

No	Skenario Alarm	Waktu Deteksi	Waktu Notifikasi	Delay (detik)	Status
1	Tegangan > 240 V	22:28:24	22:28:25	1	Berhasil
2	Arus > 12,3 A	22:30:12	22:30:13	1,28	Berhasil
3	Suhu Baterai > 60°C	22:31:45	22:31:46	1,27	Berhasil
4	Gas Habis < 10 psi	22:31:46	22:31:46	1,46	Berhasil
5	Tegangan < 215 V	22:34:22	22:34:23	1,25	Berhasil
6	Frekuensi > 51,5Hz	22:35:58	22:35:59	0,87	Berhasil
7	Tekanan Selang > 2,2 psi	22:37:11	22:37:13	1,40	Berhasil
8	Daya > 2200 W	22:38:27	22:38:27	1,32	Berhasil
9	Baterai < 20 %	22:39:42	22:39:43	1,27	Berhasil
10	Tegangan > 240 V	22:41:10	22:41:11	1	Berhasil

Hasil pengujian *delay* notifikasi mencapai rata-rata 1,21 detik dengan nilai minimum 0,87 detik dan maksimum 1,46 detik. Seluruh 10 percobaan berhasil dengan tingkat keberhasilan 100%, memenuhi target delay ≤2 detik yang ditetapkan. Variasi *delay* disebabkan oleh perbedaan waktu pemrosesan *Firebase Cloud Messaging* dan kondisi jaringan pada saat pengujian berlangsung. Hasil ini konsisten dengan penelitian yang melaporkan *delay* notifikasi 2 detik pada arsitektur NodeMCU-*Firebase*-Blynk untuk *Home Automation*, menunjukkan bahwa notifikasi *real-time* pada *platform* ESP32 dapat mencapai waktu *respons* di bawah 2 detik [18].

3.8. Pengujian Sinkronisasi Data

Pengujian sinkronisasi dilakukan dengan mencocokkan nilai tegangan dan frekuensi dari modul PZEM-004T pada *Serial Monitor*, *Firebase*, dan tampilan web yang digabung dalam satu kolom karena dashboard berfungsi sebagai *layer* visualisasi yang menampilkan data secara langsung dari *Firebase* tanpa melakukan proses pengukuran atau pengolahan data tambahan. Hasil uji ditampilkan pada Tabel 8.

Tabel 8. Hasil Pengujian Sinkronisasi Data

No	PZEM-004T (V)	*Firebase* / *Dashboard*	Selisih (V)	Frekuensi (Hz)	Sinkron
1	227,85	227,81	0,04	50	Ya
2	221,30	221,26	0,04	50	Ya
3	229,60	229,54	0,06	50	Ya
4	224,10	224,07	0,03	50	Ya
5	230,95	230,88	0,07	50	Ya
6	222,75	222,70	0,05	50	Ya
7	228,40	228,34	0,06	50	Ya
8	220,90	220,86	0,04	50	Ya
9	231,20	231,17	0,03	50	Ya
10	225,65	225,60	0,05	50	Ya

Berdasarkan hasil pengujian sinkronisasi data antara mikrokontroler ESP32-S3 dan *Firebase Realtime* Database yang merepresentasikan tampilan dashboard menunjukkan tingkat keberhasilan sebesar 100%. Dari 10 kali pengujian, sebagian data menunjukkan nilai identik antara pembacaan modul PZEM-400T dan tampilan *dashboard*, sementara sebagian lainnya mengalami selisih kecil 0,03-0,07V. Selisih tersebut disebabkan oleh pembulatan *floating point* pada proses transmisi data dari ESP32-S3 ke *Firebase*. Meskipun terdapat selisih, nilai tersebut tidak mempengaruhi akurasi monitoring karena berada jauh di bawah batas toleransi error yang ditetapkan sebesar 5%. Untuk parameter frekuensi, seluruh pengujian menunjukkan nilai yang tersinkronisasi pada 50 Hz tanpa selisih.

3.9. Pengujian Verifikasi Hash *Blockchain*

Pengujian verifikasi hash *blockchain* dilakukan sebanyak 10 percobaan dengan 5 skenario data normal dan 5 skenario manipulasi. Pada skenario normal, *hash* yang tersimpan di *Firebase* diverifikasi terhadap *blockchain* Sepolia. Pada skenario manipulasi, parameter kelistrikan diubah secara lokal kemudian diverifikasi terhadap *blockchain* untuk memastikan sistem mendeteksi ketidaksesuaian. Hasil pengujian ditampilkan pada Tabel 9.

Tabel 9. Hasil Pengujian Verifikasi *Hash Blockchain*

No	Skenario	Hash Tersimpan	Validasi	Waktu (s)
1	Data Normal	57f19d3d...b0d5723	*Valid*	1,14
2	Data Normal	8e989322...5b4a734	*Valid*	1,03
3	Data Normal	63aa7777...2802936	*Valid*	1,11
4	Data Normal	8cd63cd1...4b60d84	*Valid*	1,11
5	Data Normal	8cd63cd1...4b60d84	*Valid*	1,26
6	Manipulasi Tegangan	330b5b99...1ad39f	*Invalid*	2,42
7	Manipulasi Arus	8939d3b0...1300530	*Invalid*	1,61
8	Manipulasi Frekuensi	ebe6cc4e...259debc	*Invalid*	1,4
9	Manipulasi Daya	cc61088b...68eeb72	*Invalid*	1,4
10	Manipulasi Kapasitas Baterai	69a02b74...7065e9	*Invalid*	1,26

Hasil pengujian validasi integritas data menunjukkan tingkat akurasi deteksi 100% dengan rata-rata waktu validasi 1,38 detik. Seluruh data normal menghasilkan status *Valid* yang berarti *hash* di *Firebase* cocok dengan *hash* di *blockchain*, sedangkan seluruh data manipulasi menghasilkan status *Invalid* yang mengindikasikan sistem berhasil mendeteksi perubahan data. Hasil ini sejalan dengan penelitian yang menyatakan bahwa mekanisme *hash* SHA-256 pada *blockchain* mampu mendeteksi perubahan data sekecil apapun karena sifat *one-way function* dan *collision resistance* dari algoritma tersebut [19].

3.10. Pengujian Transaksi *Blockchain*

Pengujian transaksi *blockchain* dilakukan dengan mengambil 10 transaksi terakhir yang tercatat pada *smart contract* di jaringan Ethereum Sepolia. Parameter yang diukur meliputi *hash* data, *Transaction ID*, *gas used*, dan waktu konfirmasi. Hasil pengujian ditampilkan pada Tabel 10.

Tabel 10. Hasil Pengujian Transaksi *Blockchain*

No	Waktu	TX Hash	Gas Used	Waktu Konfirmasi (s)	Status
1	08:17:36 PM	0xfcd118...4c059	52000	14,1	Berhasil
2	08:17:48 PM	0x73c7f4...71a64	52000	17,3	Berhasil
3	08:18:00 PM	0x856dc4...0c94	52000	15,2	Berhasil
4	08:18:12 PM	0x57f19d...4745f	52000	13,2	Berhasil
5	10:03:24 PM	0x8e9893...01d20	52000	12,4	Berhasil
6	10:03:24 PM	0x63aa77...fcc4a	52000	13,1	Berhasil
7	10:04:00 PM	0x8cd63c...3e475	52000	14,6	Berhasil
8	10:07:12 PM	0x8cd63c...3e475	52000	15,9	Berhasil
9	10:10:12 PM	0xfb67eb...f2d5	52000	13,2	Berhasil
10	10:10:24 PM	0xc941ff...5bb1	52000	13	Berhasil

Hasil pengujian transaksi *blockchain* menunjukkan tingkat keberhasilan 100% dengan rata-rata waktu konfirmasi 14,2 detik dan *gas used* sebesar 52.000 per transaksi. Waktu konfirmasi ini sesuai dengan karakteristik jaringan Ethereum Sepolia yang memiliki block time sekitar 12.5-24.7 detik [20]. Penggunaan *gas* 52.000 per transaksi tergolong efisien untuk operasi penyimpanan *hash* karena hanya menyimpan *string* 64 karakter tanpa data *payload* tambahan. Seluruh transaksi dapat diverifikasi secara publik melalui Etherscan pada alamat kontrak yang telah di-deploy.

4. KESIMPULAN

Sistem monitoring dan kontrol genset LPG berbasis IoT dengan validasi integritas data *blockchain* telah berhasil diimplementasikan dan diuji. Pengujian akurasi modul PZEM-004T menunjukkan rata-rata error 1,76% untuk tegangan, 0,27% untuk arus, dan 0,71% untuk frekuensi, seluruhnya memenuhi batas toleransi yang ditetapkan. *Response time* kontrol jarak jauh mencapai rata-rata 1,5 detik dengan tingkat keberhasilan 100%. *Latency Firebase* rata-rata 733,6 ms dan REST API 206,3 ms berada di bawah batas 1000 ms, sedangkan delay notifikasi rata-rata 1,21 detik memenuhi target ≤2 detik. Mekanisme validasi hash *blockchain* mendeteksi seluruh skenario manipulasi data dengan akurasi 100% dan waktu validasi rata-rata 1,38 detik.

Keterbatasan penelitian ini meliputi pengujian *blockchain* yang dilakukan pada jaringan testnet Sepolia dengan kondisi trafik yang relatif stabil. Penelitian selanjutnya dapat mengevaluasi implementasi pada jaringan *mainnet* dan mengintegrasikan algoritma prediktif untuk perawatan preventif berdasarkan data historis yang telah tervalidasi.

DAFTAR PUSAKA

[1] M. Rifa'i, S. Kanata, dan L. Muntasiroh, "Reliability Analysis of 20kV Distribution System SRN 02 Transmission Line of PT PLN (Persero) UP3 Surakarta," *Jambura J. Electr. Electron. Eng.*, vol. 7, no. 1, hlm. 11–17, Jan 2025, doi: 10.37905/jjeee.v7i1.26291.

[2] A. Rahmandhika, "Pengaruh Variasi Pembebanan Terhadap Performansi Mesin Diesel Single-Fuel Berbahan Bakar Dexlite dan Liquified Petroleum Gas," *J. Mesin Nusant.*, vol. 7, no. 1, hlm. 89–101, Jun 2024, doi: 10.29407/jmn.v7i1.21874.

[3] E. Tonadi, A. F. Silaen, E. Rusliono, dan N. Niharman, "Perbandingan Konsumsi Bahan Bakar dan Emisi Gas Buang antara Bahan Bakar Pertalite dengan Gas LPG pada Mesin Genset Honda GX 160," *J. Rekayasa Energi Dan Mek.*, vol. 4, no. 2, hlm. 96, Feb 2025, doi: 10.26760/JREM.v4i2.96.

[4] G. Talib Hashem dan M. F Al-Dawody, "Use of LPG in SI engine- A review study," *Al-Qadisiyah J. Eng. Sci.*, Mar 2021, doi: 10.30772/qjes.v14i1.751.

[5] M. H. KHOWARIZMI, "Real-Time Monitoring System for Solar Power Plants Using PLC and IoT-Based Architecture," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, 2025.

[6] Y. Rafsyam *dkk.*, "Design and Development of a Transceiver Device with a Power Control and Monitoring Application for Boarding House Tenants based on LoRa," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 13, no. 2, hlm. 115, Apr 2025, doi: 10.26760/elkomika.v13i2.115.

[7] A. Suryaningrat, D. Kurnianto, dan R. A. Rochmanto, "Sistem Monitoring Kelembaban Tanaman Cabai Rawit menggunakan Irigasi Tetes Gravitasi berbasis Internet Of Things (IoT)," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, hlm. 568, Jul 2022, doi: 10.26760/elkomika.v10i3.568.

[8] L. Hartawan *dkk.*, "Design of IoT-based Greenhouse Temperature and Humidity Monitoring System," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 12, no. 4, hlm. 1051, Des 2024, doi: 10.26760/elkomika.v12i4.1051.

[9] S. C. Abadi, N. W. Nugraha, I. A. Dhimyati, dan A. H. Sumarso, "Penerapan Human Machine Interface Berbasis Augmented Reality pada Sistem SCADA Modular Production System," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 11, no. 2, hlm. 285, Apr 2023, doi: 10.26760/elkomika.v11i2.285.

[10] E. A. Hw, R. Tulloh, S. Hadiyoso, dan D. N. Ramadan, "Sistem Pemantauan dan Pendeteksi Kebakaran berbasis Logika Fuzzy dan Real-time Database," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 3, hlm. 577, Jul 2021, doi: 10.26760/elkomika.v9i3.577.

[11] R. Asokan, D. P. Ruiz, dan V.-D. Păvăloaia, Ed., *Smart Data Intelligence: Proceedings of ICSMDI 2025*. dalam

[12] E. P. Chopra, "Creating Live Dashboards for Data Visualization: Flask vs. React," vol. 8, no. 9, 2021.

[13] S. Soni dan A. Singh, "A hybrid *blockchain* and smart contract framework for resilient IoT security in smart homes," *Front. Blockchain*, vol. 8, hlm. 1707911, Des 2025, doi: 10.3389/fbloc.2025.1707911.

[14] S. Suhail *dkk.*, "*Blockchain*-Based Digital Twins: Research Trends, Issues, and Future Challenges," *ACM Comput. Surv.*, vol. 54, no. 11s, hlm. 1–34, Jan 2022, doi: 10.1145/3517189.

[15] E. Şahin, N. N. Arslan, dan F. Aydemir, "Interactive Use of Deep Learning and Ethereum *Blockchain* for the Security of IIoT Sensor Data," *Bilecik Şeyh Edebali Üniversitesi Fen Bilim. Derg.*, vol. 11, no. 2, hlm. 369–384, Nov 2024, doi: 10.35193/bseufbd.1381786.

[16] M. F. Mata-Rivera, R. Zagal-Flores, dan C. Barria-Huidobro, Ed., *Telematics and Computing: 14th International Congress of Telematics and Computing, WITCOM 2025, Huatulco, Mexico, November 3–7, 2025, Proceedings, Part II*, vol. 2705. dalam Communications in Computer and Information Science, vol. 2705. Cham: Springer Nature Switzerland, 2025. doi: 10.1007/978-3-032-09738-5.

[17] K. A. Yasa, I. M. Purbhawa, I. M. Sumerta Yasa, I. W. Teresna, A. Nugroho, dan S. Winardi, "Prototype Pemantauan Konsumsi Energi Listrik pada *Firebase* Menggunakan PZEM-004T," *J. Eksplora Inform.*, vol. 12, no. 2, hlm. 104–112, Des 2023, doi: 10.30864/eksplora.v12i2.993.

[18] P. K. A. Windesi, M. R. Sampebua, dan R. M. Kmurawak, "IOT-BASED HOME AUTOMATION USING NODEMCU ESP8266," *J. Ris. Inform.*, vol. 4, no. 4, hlm. 391–396, Sep 2022, doi: 10.34288/jri.v4i4.431.

[19] C. Gonzalez-Amarillo, C. Cardenas-Garcia, M. Mendoza-Moreno, G. Ramirez-Gonzalez, dan J. C. Corrales, "*Blockchain*-IoT Sensor (BIoTS): A Solution to IoT-Ecosystems Security Issues," *Sensors*, vol. 21, no. 13, hlm. 4388, Jun 2021, doi: 10.3390/s21134388.

[20] F. Javed dan J. Mangues-Bafalluy, "An Empirical Smart Contracts Latency Analysis on Ethereum *Blockchain* for Trustworthy Inter-Provider Agreements," 3 Maret 2025, *arXiv*: arXiv:2503.01397. doi: 10.48550/arXiv.2503.01397.