PEMANFAATAN *LITTLEFS* PADA SISTEM STOPKONTAK OTOMATIS BERBASIS *WEBSOCKET*

IMMANUEL
H1051211035

SKRIPSI

PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA
PONTIANAK
2025

PEMANFAATAN *LITTLEFS* PADA SISTEM STOPKONTAK OTOMATIS BERBASIS *WEBSOCKET*

IMMANUEL
H1051211035

Skripsi
Sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains pada Program Studi Rekayasa Sistem Komputer



PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS TANJUNGPURA
PONTIANAK
2025

LEMBAR PENGESAHAN

Judul Tugas Akhir	: Pemanfaatan *LittleFS* Pada Sistem Stopkontak Otomatis Berbasis *WebSocket*
Nama Mahasiswa	: Immanuel
NIM	: H1051211035
Jurusan/Program Studi	: Rekayasa Sistem Komputer
Tanggal Lulus	: 16 Juli 2025
SK Pembimbing	: No. 266/UN22.8/TD.06/2025 /03 Februari 2025
SK Penguji	: No.2207/UN22.8/TD.06/2025 /15 Juli 2025

Dosen Pembimbing

Pembimbing I

Syamsul Bahri, S.Kom., M.Cs.
NIP. 198802272015041001

Pembimbing II

Hirzen Hasfani, S.Kom., M.Cs.
NIP. 199305032022031003

Dosen Penguji

Ketua Penguji

Dwi Marisa Midyanti, S.T., M.Cs.
NIP. 198003192015042001

Anggota Penguji

Kasliono, S.Mat., M.Cs.
NIP. 199306202022031005

Pimpinan Sidang
(merangkap anggota penguji)

Syamsul Bahri, S.Kom., M.Cs.
NIP. 198802272015041001

Sekretaris Sidang
(merangkap anggota penguji)

Hirzen Hasfani, S.Kom., M.Cs.
NIP. 199305032022031003

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Tanjungpura

Prof. Dr. Gusrizal, S.Si., M.Si.
NIP. 197108022000031001

PERNYATAAN INTEGRITAS AKADEMIK

Saya yang bertanda tangan di bawah ini:

Nama	: Immanuel
NIM	: H1051211035
Program Studi/ Jurusan	: Rekayasa Sistem Komputer
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

dengan ini menyatakan bahwa dokumen ilmiah Tugas Akhir yang disajikan ini tidak mengandung unsur pelanggaran integritas akademik sesuai Peraturan Menteri Pendidikan, Kebudayaan, Riset, Dan Teknologi Republik Indonesia Nomor 39 Tahun 2021. Apabila di kemudian hari dokumen ilmiah Tugas Akhir ini mengandung unsur pelanggaran integritas akademik sesuai ketentuan perundangan tersebut, maka saya bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Demikian pernyataan ini untuk digunakan sebagaimana mestinya.

Pontianak, 11 Agustus 2025



Immanuel
H1051211035

Pemanfaatan *LittleFS* Pada Sistem Stopkontak Otomatis Berbasis *Websocket*

Abstrak

ESP32 memiliki keterbatasan dalam memori dan penyimpanan data, terutama dalam menjaga data tetap tersimpan saat terjadi pemadaman daya. Untuk itu dirancang sistem stopkontak otomatis berbasis *WebSocket* yang mengintegrasikan *LittleFS* sebagai sistem *file* penyimpanan data. *LittleFS* bertujuan untuk mengimplementasikan sistem penyimpanan pada kondisi pemadaman, serta mengukur performa pemulihan data dan waktu respon sistem. Sistem ini dapat memantau serta mengendalikan perangkat listrik secara *real-time* melalui jaringan lokal. *WebSocket* digunakan untuk komunikasi dua arah antara *web* dan perangkat, sementara *LittleFS* pada sistem stopkontak otomatis berfungsi menyimpan data konfigurasi, status perangkat, pengaturan *timer*, data sensor, dan *log* aktivitas meskipun terjadi pemadaman daya. Pengujian dilakukan dalam dua aspek, yaitu data *loss* dan waktu respon pemulihan sistem. Pengujian data *loss* dilakukan sebanyak 30 siklus pemadaman, menunjukkan keberhasilan pemulihan data tanpa kehilangan informasi. Sementara itu, pengujian waktu respon dilakukan sebanyak 30 siklus pemadaman, menunjukkan bahwa waktu respon tercatat rata-rata antara 283,2 ms. Secara keseluruhan, *LittleFS* berhasil mempertahankan data saat pemadaman daya dan memulihkan data secara otomatis, serta memberikan waktu respon sistem yang responsif meskipun ukuran data meningkat.

Kata Kunci: *LittleFS*, Stopkontak Otomatis, *WebSocket*, ESP32, Penyimpanan Data

Utilization of LittleFS in Websocket-based Automatic Power Outlet System

Abstract

ESP32 has limitations in memory and data storage, especially in keeping data stored during a power outage. For this reason, a WebSocket-based automatic power outlet system is designed that integrates LittleFS as a data storage file system. LittleFS aims to implement a storage system under outage conditions, as well as measure data recovery performance and system response time. The system can monitor and control electrical devices in real-time through a local network. WebSocket is used for two-way communication between the web and the device, while LittleFS on the automatic power outlet system serves to store configuration data, device status, timer settings, sensor data, and activity logs despite power outages. Testing was conducted in two aspects, namely data loss and system recovery response time. Data loss testing was conducted for 30 outage cycles, showing successful data recovery without loss of information. Meanwhile, the response time test was conducted for 30 outage cycles, showing that the response time was recorded at an average of 283.2 ms. Overall, LittleFS successfully retains data during power outages and recovers data automatically, as well as provides decent system response time despite increasing data size.

Keywords: LittleFS, Automatic Power Outlet, WebSocket, ESP32, Data Storage

PRAKATA

Segala syukur dan puji hanya bagi Tuhan Yesus Kristus, oleh karena anugerah-Nya yang melimpah, kemurahan dan kasih setia yang besar, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **"Pemanfaatan *LittleFS* Pada Sistem Stopkontak Otomatis Berbasis *WebSocket*"**. Tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom.) pada Jurusan Rekayasa Sistem Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Tanjungpura Pontianak.

Semua hasil yang diperoleh ini tidak lepas dari segala dukungan dan bantuan dari berbagai pihak yang terus mendorong penulis untuk menyelesaikan tugas akhir ini dengan baik. Sebagai penghargaan, izinkanlah penulis mengucapkan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Gusrizal, S.Si., M.Si., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Tanjungpura.
2. Bapak Tedy Rismawan, S.Kom., M.Cs., selaku Ketua Jurusan Rekayasa Sistem Komputer FMIPA Universitas Tanjungpura Pontianak.
3. Bapak Syamsul Bahri, S.Kom., M.Cs., sebagai dosen Pembimbing Pertama yang telah memberikan masukan serta meluangkan waktu dan pikiran yang memberikan dorongan dalam bimbingan penulisan skripsi.
4. Bapak Hirzen Hasfani, S.Kom., M.Cs., sebagai dosen Pembimbing Kedua yang telah memberikan masukan serta meluangkan waktu dan pikiran yang memberikan dorongan dalam bimbingan penulisan skripsi.
5. Ibu Dwi Marisa Midyanti, S.T., M.Cs., selaku dosen Penguji Pertama, yang telah memberikan kontribusi dengan memberikan masukan dan pertanyaan yang memperkaya isi dari tugas akhir ini.
6. Bapak Kasliono, S.Mat., M.Cs., selaku dosen Penguji Pertama, yang telah memberikan kontribusi dengan memberikan masukan dan pertanyaan yang memperkaya isi dari tugas akhir ini
7. Bapak Ikhwan Ruslianto, S.Kom, M.Cs., sebagai dosen Pembimbing Akademik (PA) yang telah memberi masukan dan motivasi yang membangun selama masa perkuliahan.

8. Kepada orang tua yang selalu memberikan dukungan, doa, motivasi, dan kasih sayang tanpa henti sepanjang perjalanan penulisan tugas akhir ini.
9. Kepada kakak-kakak saya Bella Septiana Nababan, Helprida Nababan, dan Janiawati Nababan yang selalu memberi doa, semangat, dan dukungan tanpa henti sepanjang perjalanan penulisan tugas akhir ini.
10. Teman-teman saya Andy, Alif, Fioni, Fauzi, Jero, Novianti, Yessa, Yunita, dan Zevanya yang memberi semangat dan saran dalam tugas akhir ini sehingga dapat terselesaikan.
11. Para rekan-rekan seperjuangan mahasiswa khususnya Sistem Komputer 2021 (Nemotic) yang telah memberikan semangat dan dukungan dalam menjalani proses perkuliahan dan penulisan tugas akhir ini.
12. Semua pihak yang terlibat langsung maupun yang tidak langsung dalam penulisan tugas akhir ini.

Pontianak, 16 Juli 2025

Immanuel

DAFTAR ISI

LEMBAR PENGESAHAN ii
PERNYATAAN INTEGRITAS AKADEMIK iii
Abstrak iv
Abstract v
PRAKATA vi
DAFTAR ISI viii
DAFTAR GAMBAR xi
DAFTAR TABEL xiii
DAFTAR KODE PROGRAM xiv
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 5
1.6 Sistematika Penulisan 5
BAB 2 LANDASAN TEORI 7
2.1 Perbandingan dengan penelitian terdahulu 7
2.2 *LittleFS* 9
2.3 *Wear leveling* 10
2.4 *Copy-On-Write* 10
2.5 *WebSocket* 10
2.6 SPI *Flash Memory* 11
2.7 *Wifi Manager* 12
2.8 ESP32 12
2.9 Arduino IDE 13
2.10 Modul PZEM-004T 14
2.11 *Real-Time Clock* 14
2.12 *Relay* 15
2.13 Data *Loss* 15

2.14 Waktu Respon 16
2.15 *Hypertext Markup Language* 16
2.16 *CSS* 17
2.17 *JavaScript* 17
2.18 Balsamiq Mockup 17
2.19 Penanganan Keterbatasan Memori 18
2.20 Diagram Alir 18
BAB 3 METODE PENELITIAN 19
3.1 Studi Literatur 19
3.2 Analisis Kebutuhan 20
3.3 Perancangan Sistem 20
3.4 Implementasi 21
3.5 Pengujian 21
3.6 Kesimpulan dan Saran 22
BAB 4 PERANCANGAN 23
4.1 Deskripsi Sistem 23
4.2 Perancangan Arsitektur Sistem 23
4.3 Perancangan Perangkat Keras 26
4.4 Perancangan Perangkat Lunak 29
4.5 Perancangan *LittleFS* Pada Sistem Stopkontak Otomatis 32
4.6 Perancangan Antar Muka 34
4.7 Perancangan Pengujian Data *Loss* Menggunakan *LittleFS* 37
4.8 Perancangan Pengujian Waktu respon Pemulihan Data 37
BAB 5 IMPLEMENTASI, PENGUJIAN DAN PEMBAHASAN 39
5.1 Implementasi Perangkat Keras 39
5.2 Implementasi Perangkat Lunak 40
5.3 Implementasi *LittleFS* Pada Sistem Kontak Otomatis 44
5.4 Implementasi Antarmuka 50
5.5 Pengujian Sistem 53
5.6 Pembahasan 60
BAB 6 PENUTUP 62
6.1 Kesimpulan 62

6.2 Saran 62
DAFTAR PUSTAKA 64
LAMPIRAN 68

DAFTAR GAMBAR

Gambar 2.1 Pola Arsitektur *LittleFS* 9
Gambar 2.2 Mekanisme Kerja Protokol *Websocket* 11
Gambar 2.3 *Wifi Manager* 12
Gambar 2.4 ESP32 13
Gambar 2.5 Arduino IDE 13
Gambar 2.6 Modul PZEM-004T 14
Gambar 2.7 *Real-Time Clock* 14
Gambar 2.8 *Relay* 15
Gambar 3.1 Diagram Alir Penelitian 19
Gambar 4.1 Arsitektur Sistem Pemanfaatan *LittleFS* pada Sistem Stopkontak Otomatis Berbasis *WebSocket* 24
Gambar 4.2 Perancangan Sistem Kontrol Stopkontak 27
Gambar 4.3 Perancangan Sistem Pemantauan Konsumsi Daya 28
Gambar 4.4 Perancangan Keseluruhan Perangkat Keras 29
Gambar 4.5 Diagram Alir Sistem Penyimpanan Data Pada Sistem *File LittleFS*30
Gambar 4.6 Perancangan Penyimpanan Data dengan *LittleFS* 31
Gambar 4.7 Perancangan Mekanisme Pemulihan Data dan Kendali Perangkat . 32
Gambar 4.8 Tampilan Pengaturan Wi-Fi 34
Gambar 4.9 Tampilan Halaman *Login* 35
Gambar 4.10 Tampilan Halaman *Dashboard* 35
Gambar 4.11 Tampilan Halaman *Log* Aktivitas 36
Gambar 4.12 Tampilan Halaman Pengaturan Perangkat 36
Gambar 5.1 Implementasi Sistem Stopkontak Otomatis 39
Gambar 5.2 Proses Partisi *LittleFS* 45
Gambar 5.3 *File* Tabel Partisi *LittleFS* 45
Gambar 5.4 Info Blok *LittleFS* 48
Gambar 5.5 Tampilan Status *LittleFS* Pada Antarmuka *Web* 49
Gambar 5.6 Tampilan Halaman Konfigurasi WiFi 50
Gambar 5.7 Tampilan Halaman *Login* 51
Gambar 5.8 Tampilan Halaman *Dashboard* 51

Gambar 5.9 Tampilan Halaman Pengaturan *Timer* 52
Gambar 5.10 Tampilan Halaman *Log* Aktivitas 52
Gambar 5.11 Proses Koneksi Wi-Fi 53
Gambar 5.12 Penyimpanan data 53
Gambar 5.13 Pengiriman Kembali Data ke Antarmuka *Web* 54
Gambar 5.14 Grafik Hasil Pengujian Waktu respon 59

DAFTAR TABEL

Tabel 2.1 Perbandingan dengan Penelitian Terdahulu 7
Tabel 4.1 Penggunaan Pin ESP32 dengan *Relay* dan RTC 27
Tabel 4.2 Penggunaan Pin ESP32 dengan PZEM-004T 28
Tabel 4.3 Struktur Direktori dan *File* dalam *LittleFS* 33
Tabel 4.4 Perancangan Pengujian Data *Loss* 37
Tabel 4.5 Perancangan Pengujian Waktu respon 38
Tabel 5.1 Hasil Pengujian Data *Loss* 55
Tabel 5.2 Hasil Pengujian Waktu respon 58

DAFTAR KODE PROGRAM

Kode Program 5.1. Sistem Stopkontak Otomatis dengan Protokol *Websocket* .. 41
Kode Program 5.2 Penyimpanan Data dengan *LittleFS*..... 42
Kode Program 5.3 Mekanisme Pemulihan Data dan Kendali Perangkat..... 43
Kode Program 5.4 *Wear Leveling* 46
Kode Program 5.5 *Copy-On-Write* 47
Kode Program 5.6 Mode *Copy-On-Write* 47
Kode Program 5.7 Penanganan Keterbatasan Kapasitas *LittleFS*..... 49

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Kebutuhan energi listrik terus meningkat seiring dengan pertumbuhan populasi dan penggunaan perangkat elektronik dalam kehidupan sehari-hari. Peningkatan penggunaan perangkat elektronik secara luas menyebabkan konsumsi listrik meningkat, yang berpotensi menimbulkan pemborosan energi dan kenaikan tagihan akibat keterbatasan pasokan listrik (Nurihsan & Paputungan, 2022). Sekitar 80% pemborosan listrik terjadi karena faktor manusia, seperti membiarkan perangkat *standby* atau tetap terhubung ke stopkontak yang menyebabkan penggunaan listrik tidak efisien (Gumilang et al., 2022). Sehingga diperlukannya sistem pada stopkontak yang dapat memantau dan mengatur penggunaan listrik.

Dengan semakin berkembangnya teknologi, pengelolaan penggunaan listrik dapat dilakukan dengan lebih efisien melalui sistem berbasis *Internet of Things* (IoT). IoT memungkinkan perangkat seperti stopkontak untuk terhubung dengan jaringan lokal, sehingga pengguna dapat memantau dan mengendalikan perangkat tersebut secara *remote*. Hal ini tidak hanya membantu mengurangi pemborosan energi, tetapi juga meningkatkan kenyamanan dan keamanan pengguna dalam mengatur penggunaan listrik di rumah (Gumilang et al., 2022).

Perangkat IoT pada mikrokontroler ESP32 biasanya memiliki keterbatasan daya proses, memori, dan kapasitas penyimpanan (Fauzi & Bhawiyuga, 2019). Maka pengelolaan data pada perangkat IoT memerlukan sistem penyimpanan dengan menggunakan *file system*. *File system* merupakan metode untuk memberi nama pada berkas dan meletakkannya pada media penyimpanan, atau suatu sistem untuk mengetahui bagaimana cara menyimpan data dari *file* tertentu dan organisasi file yang digunakan (Farizy & Harianja, 2020). *LittleFS*, sebagai *file system* yang dirancang untuk perangkat dengan sumber daya terbatas, merupakan pilihan yang memadai (Grokhotkov, 2023). *LittleFS* menawarkan keunggulan dalam hal keandalan penggunaan memori, yang sangat penting untuk pengoperasian perangkat IoT seperti stopkontak otomatis meskipun terjadinya pemadaman daya

listrik. Dengan menggunakan *LittleFS*, data yang disimpan meliputi data konfigurasi, status perangkat, dan *log* aktivitas secara optimal.

Protokol komunikasi juga memiliki peranan penting dalam pengoperasian stopkontak otomatis berbasis IoT. *WebSocket* memiliki kemampuan untuk mendukung komunikasi dua arah secara *real-time* antara perangkat dan *server* dengan latensi rendah (Mitrović et al., 2023). Dalam protokol *Websocket*, pengiriman data dari *server* ke *client* tidak perlu mengenali ulang *client*, sehingga ukuran paket yang dikirimkan lebih kecil dan hemat energi (Putra et al., 2024). Ukuran paket yang kecil membantu mengoptimalkan penggunaan ruang penyimpanan pada sistem berkas seperti *LittleFS*, sehingga meminimalkan risiko kehabisan kapasitas secara cepat. *LittleFS* diintegrasikan untuk menyimpan data di memori *flash*, termasuk *file* konfigurasi, data, dan *log* aktivitas yang dapat diakses kembali setelah sistem melakukan *reboot* atau pemulihan (Grokhotkov, 2023). Hal ini mempengaruhi sinkronisasi data dapat berjalan lebih cepat atau lebih lambat saat *WebSocket* melakukan pengiriman data yang disimpan *LittleFS* pada kondisi perangkat mulai kembali aktif. Pada sistem stopkontak otomatis ini, data penggunaan listrik dapat dikirim secara *real-time*, sementara perintah pengendalian diterima dengan cepat, sehingga meningkatkan responsivitas sistem dan kenyamanan pengguna dalam mengelola konsumsi energi.

Penelitian sebelumnya yang berjudul "*Methods of increasing the reliability of external memory of digital substation bay controllers*" (Lantsev & Frolov, 2019), mengevaluasi sistem *file* populer untuk mikrokontroler dengan sumber daya terbatas, seperti FATFS, SPIFFS, dan *LittleFS*, dalam pengendalian gardu induk digital. *LittleFS* menggunakan ROM sebesar 10 kB dan mendukung struktur direktori untuk pengelolaan *file*. Sistem ini dilengkapi dengan fitur *wear leveling* yang mendistribusikan operasi tulis secara merata di seluruh memori *flash*, serta mekanisme *copy-on-write* yang memastikan data terlindungi saat terjadi kegagalan daya. Selain itu, *LittleFS* mampu menjaga stabilitas sistem dalam penggunaan jangka panjang. Meski demikian, penelitian ini mencatat bahwa *LittleFS* memiliki keterbatasan dalam menangani *file* berukuran besar.

Penelitian lain yang berjudul "Analisis Komparasi Protokol *WebSocket* dan MQTT dalam Proses *Push Notification*" (Putra et al., 2024), membandingkan

kinerja kedua protokol dalam pengiriman notifikasi secara *real-time* pada aplikasi modern. Penelitian ini menunjukkan bahwa *WebSocket* unggul dalam skenario aplikasi *real-time* karena mendukung koneksi dua arah, sehingga menghasilkan waktu *delay* yang lebih rendah dibandingkan MQTT. Dalam pengujian pada satu klien, *WebSocket* memiliki rata-rata *delay* sebesar 0,547 detik, lebih cepat dibandingkan dengan MQTT yang mencapai 0,653 detik. Selain itu, WebSocket juga terbukti lebih efisien dalam penggunaan sumber daya CPU, dengan rata-rata penggunaan CPU sebesar 6,602% dibandingkan dengan 8,056% pada MQTT. Penelitian ini menegaskan bahwa *WebSocket* sangat cocok untuk sistem yang membutuhkan respons cepat dan efisiensi tinggi, khususnya dalam aplikasi *push notification* berbasis jaringan lokal.

Penelitian lain yang berjudul "Pengujian Stopkontak Pintar Menggunakan ESP32" (Sudaryanto et al., 2020), merancang sistem stopkontak pintar menggunakan ESP32 yang berfungsi sebagai kontrol utama. Sistem ini mengendalikan *relay* untuk menghubungkan atau memutus aliran listrik berdasarkan perintah yang diterima dari pengguna melalui aplikasi berbasis web. ESP32 dilengkapi dengan fitur Wi-Fi yang memungkinkan perangkat terhubung ke jaringan internet untuk menerima perintah secara jarak jauh. Pengujian dilakukan sebanyak 10 kali pada setiap lubang stopkontak, dan hasilnya menunjukkan tingkat keberhasilan sistem pada semua percobaan. Penelitian ini menunjukkan bahwa penggunaan ESP32 dengan *relay* efektif dalam mendukung penghematan konsumsi listrik melalui kontrol yang mudah dan responsif.

Berdasarkan penelitian yang telah dipaparkan di atas, maka diangkatlah penelitian dengan judul “Pemanfaatan *LittleFS* Pada Sistem Stop Kontak Otomatis Berbasis *WebSocket*”. Penelitian ini bertujuan untuk mengimplementasikan sistem stopkontak otomatis yang dapat dikendalikan secara *real-time* melalui jaringan lokal tanpa memerlukan akses internet dan pemanfaatan *LittleFS* dalam menyimpan data bahkan saat terjadi pemadaman daya. Sistem ini memanfaatkan protokol *WebSocket* untuk memungkinkan komunikasi dua arah antara perangkat dan antarmuka *web*, serta *LittleFS* sebagai *file system* untuk menyimpan data konfigurasi, status perangkat, dan *log* aktivitas yang kemudian akan ditampilkan dalam bentuk *website*. Dengan demikian, sistem ini dapat memberikan pengalaman

kontrol perangkat yang responsif dan handal, bahkan ketika perangkat mengalami *restart* atau kegagalan daya.

1.2 Rumusan Masalah

Berdasarkan latar belakang, permasalahan yang terdapat pada penelitian ini adalah:

1. Bagaimana metode *LittleFS* dalam menyimpan data dan mengatasi data *loss* saat terjadi pemadaman daya pada sistem stopkontak otomatis?
2. Bagaimana pengaruh *LittleFS* dalam waktu respon pemulihan data melalui protokol *WebSocket* pada sistem stopkontak otomatis?

1.3 Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan pada *prototype* sistem stopkontak otomatis menggunakan satu slot stopkontak saja.
2. Stopkontak dilengkapi *relay*, RTC dan modul PZEM-004T.
3. Penelitian hanya mencakup konfigurasi Wi-Fi, data tegangan listrik, fungsi pengaturan status perangkat (*on/off*), pengaturan *timer* otomatis, dan log aktivitas yang disimpan *LittleFS*.
4. Sistem menggunakan protokol *WebSocket* sebagai komunikasi melalui jaringan lokal menggunakan *web server*.
5. Penelitian dilakukan pada jaringan lokal tanpa integrasi dengan layanan *cloud* atau IoT lainnya.
6. Sistem dapat mengatur *on/off* dan *timer* stopkontak dan memantau tegangan, arus, dan daya listrik.
7. Data yang dikumpulkan meliputi status perangkat, *log* aktivitas, dan konsumsi listrik.
8. Perhitungan waktu respon tidak memperhatikan kualitas perangkat/*device* dan jaringan.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui metode *LittleFS* dalam menyimpan data dan mengatasi data *loss* saat terjadi pemadaman daya pada sistem stopkontak otomatis.
2. Mengetahui pengaruh *LittleFS* dalam waktu respon pemulihan data melalui protokol *WebSocket* pada sistem stopkontak otomatis.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah membantu dalam pengendalian dan pemantauan perangkat listrik secara otomatis dengan protocol *WebSocket*. Sistem memastikan penyimpanan data menggunakan *LittleFS*, sehingga konfigurasi tetap tersimpan meskipun terjadi pemadaman daya. Selain itu, *LittleFS* meningkatkan ketahanan *memory flash* pada mikrokontroler dalam memperpanjang umur memori.

1.6 Sistematika Penulisan

Penulisan skripsi ini dibagi menjadi 6 bab agar permasalahan dapat dijelaskan dan diuraikan untuk menyelesaikan penelitian ini. Adapun sistematika penulisan skripsi ini terdiri dari:

BAB 1 PENDAHULUAN

Bab pendahuluan berisi latar belakang untuk menjelaskan permasalahan dalam penyimpanan data pada sistem stopkontak otomatis berbasis IoT serta pentingnya penggunaan *LittleFS* dan *WebSocket*. Bab ini juga mencakup rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bab landasan teori berisi dasar teori yang mendukung penelitian ini. Landasan teori yang dibahas mencakup *LittleFS*, protokol komunikasi *WebSocket*, *Wifi Manager*, ESP32, Arduino IDE, Modul PZEM-004T, RTC (*Real-Time Clock*), Relay, Data *loss,* Waktu Respon, HTML, CSS, *JavaScript,* Balsamiq Mockup, dan Diagram Alir (*Flowchart*).

BAB 3 METODE PENELITIAN

Bab metode penelitian menjelaskan prosedur yang digunakan dalam penelitian ini. Bab ini mencakup diagram alir penelitian, studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi, pengujian, serta teknik analisis data hasil pengujian yang digunakan untuk mengevaluasi kinerja sistem.

BAB 4 PERANCANGAN SISTEM

Bab ini menjelaskan tahapan perancangan sistem, baik dari sisi perangkat keras maupun perangkat lunak. Perancangan perangkat keras meliputi penggunaan ESP32 sebagai pengendali utama, *relay* sebagai aktuator, *Real-Time Clock* (RTC) sebagai pengatur detik, menit, dan hari, serta modul ZEM-004T untuk pengukuran

tegangan dan arus listrik. Sedangkan perancangan perangkat lunak mencakup penerapan *LittleFS* untuk penyimpanan data, komunikasi *real-time* menggunakan *WebSocket*, serta antarmuka *web* untuk monitoring dan kontrol perangkat.

BAB 5 IMPLEMENTASI, PENGUJIAN, DAN PEMBAHASAN

Bab ini membahas implementasi sistem yang telah dirancang pada tahap sebelumnya. Tahapan ini mencakup implementasi perangkat keras, perangkat lunak, serta pengujian sistem, termasuk pengujian data *loss* terhadap fungsi *LittleFS* dan waktu respon pada sistem komunikasi *WebSocket*. Hasil pengujian kemudian dianalisis dan dibandingkan untuk melihat efektivitas sistem dalam menyimpan dan memulihkan data setelah pemadaman daya.

BAB 6 PENUTUP

Bab penutup berisi kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan lebih lanjut. Kesimpulan didasarkan pada hasil analisis pengujian, sedangkan saran diberikan untuk peningkatan sistem dalam penelitian mendatang.