

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Pada tahap ini dilakukan implementasi sistem *monitoring* daya listrik berbasis *Internet of Things* (IoT) menggunakan mikrokontroler ESP32, sensor PZEM-004T, LCD 20x4, relay serta sistem notifikasi *Whatsapp*. Sistem ini dirancang untuk membaca parameter listrik secara *real-time* dan memberikan notifikasi otomatis apabila terjadi penggunaan daya berlebih.

Implementasi sistem terdiri dari dua bagian utama, yaitu implementasi perangkat keras (*hardware*) dan implementasi perangkat lunak (*software*).

4.1.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini terdiri dari beberapa komponen utama, yaitu:

Tabel 4. 1 Perangkat Keras

No	Komponen	Fungsi
1	ESP32	Sebagai mikrokontroler utama
2	Sensor PZEM-004T	Sebagai sensor pengukur daya listrik
3	LCD 20x4	Sebagai media tampilan lokal
4	Relay	Sebagai pengendali beban listrik

Sensor PZEM-004T dihubungkan ke jalur listrik untuk membaca nilai tegangan, arus, daya dan energi. Data dari sensor dikirim ke ESP32 melalui komunikasi serial. Selanjutnya, ESP32 memproses data tersebut dan menampilkannya pada LCD serta mengirimkan notifikasi melalui *Whatsapp* apabila nilai daya melebihi batas yang telah ditentukan.

Selain itu, relay digunakan sebagai saklar otomatis yang dapat memutus aliran listrik apabila terjadi kondisi *over-power*. Semua komponen dirakit sesuai skema rangkaian, dan dilakukan pengujian awal menggunakan multimeter guna memastikan sensor dan aktuator bekerja dengan baik.

4.1.2 Perangkat Lunak (*Software*)

Perangkat lunak dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman C/C++. Program pada ESP32 berfungsi untuk:

1. Membaca data dari sensor PZEM-004T
2. Menampilkan data pada LCD 20x4
3. Menghubungkan ESP32 ke jaringan WiFi
4. Mengirim notifikasi *Whatsapp* melalui API
5. Mengontrol relay berdasarkan kondisi daya
6. Data yang dibaca meliputi:
 - a. Tegangan (*Volt*)
 - b. Arus (*Ampere*)
 - c. Daya (*Watt*)
 - d. Energi (*kWh*)

Program berjalan secara terus-menerus dan memperbarui data secara *real-time*.

4.2 Rangkaian Keseluruhan Sistem

Berikut adalah rangkaian keseluruhan dari sistem *monitoring* daya listrik berbasis *Internet of Things* (IoT) yang telah dirancang, beserta penjelasan fungsional masing-masing komponen yang digunakan.

Sistem ini bekerja dengan mengukur parameter listrik berupa tegangan, arus dan daya menggunakan sensor PZEM-004T, kemudian memproses data tersebut pada mikrokontroler ESP32. Hasil pengukuran ditampilkan secara lokal pada LCD I2C 20x4 dan dikirimkan ke *Whatsapp* ketika ada perintah status. Apabila daya terukur melebihi batas yang ditetapkan, sistem secara otomatis memutus aliran listrik melalui relay dan mengirimkan notifikasi peringatan kepada pengelola sekolah melalui *Whatsapp*.



Gambar 4. 1 Rangkaian Keseluruhan Sistem

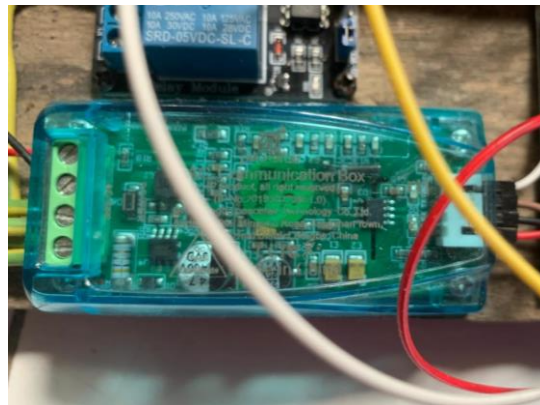
Penjelasan Fungsi Masing-Masing Modul:

1. Sensor PZEM-004T Digunakan untuk mengukur parameter listrik secara presisi dan *real-time*, meliputi tegangan (V), arus (A) dan daya aktif (W). Sensor ini berkomunikasi dengan ESP32 melalui protokol *UART Serial*.
2. ESP32 Bertugas sebagai mikrokontroler utama sekaligus modul komunikasi WiFi. ESP32 membaca data dari sensor PZEM-004T, menampilkan data pada LCD, menjalankan logika proteksi *trip overcurrent*, mengontrol relay serta mengirimkan notifikasi ke *Whatsapp*.
3. LCD I2C 20x4 Berfungsi sebagai antarmuka tampilan lokal yang menampilkan nilai tegangan, arus, daya, batas daya yang ditetapkan, status koneksi *Whatsapp*, dan status relay secara *real-time*.

4. Relay Module 5V Berfungsi sebagai saklar yang memutus atau menyambungkan aliran listrik ke beban. Relay dikontrol oleh ESP32 melalui pin GPIO 26 dan akan otomatis memutus aliran listrik ketika sistem mendeteksi kondisi daya berlebih (*overcurrent*).
5. *Power Supply 5V* Menyuplai daya ke seluruh sistem, termasuk ESP32, LCD dan relay module.

4.2.1 Rangkaian PZEM-004T

Sensor PZEM-004T digunakan untuk mengukur parameter listrik secara langsung pada instalasi listrik sekolah. Sensor ini terhubung ke ESP32 melalui komunikasi serial UART, sehingga data tegangan, arus dan daya dapat dibaca secara berkala oleh mikrokontroler.



Gambar 4. 2 Rangkaian PZEM-004T

Keterangan koneksi:

- a. VCC PZEM-004T → 5V
- b. GND PZEM-004T → GND ESP32
- c. TX PZEM-004T → RXD2 ESP32 (GPIO 16)
- d. RX PZEM-004T → TXD2 ESP32 (GPIO 17)

4.2.2 Rangkaian LCD I2C 20x4

LCD I2C 20x4 diposisikan sebagai tampilan utama pada panel *monitoring*. Antarmuka I2C digunakan agar koneksi ke ESP32 hanya memerlukan dua jalur data (SDA dan SCL), sehingga penggunaan pin GPIO lebih efisien. Alamat I2C yang digunakan adalah 0x27.



Gambar 4. 3 Rangkaian LCD I2C 20x4

Keterangan koneksi:

- a. VCC LCD → 5V
- b. GND LCD → GND ESP32
- c. SDA LCD → SDA ESP32 (GPIO 21)
- d. SCL LCD → SCL ESP32 (GPIO 22)

4.2.3 Rangkaian Relay Module

Relay module berfungsi sebagai saklar elektronik yang dikontrol oleh sinyal digital dari ESP32. Relay digunakan untuk memutus aliran listrik ke beban secara otomatis ketika sistem mendeteksi kondisi *overcurrent*.



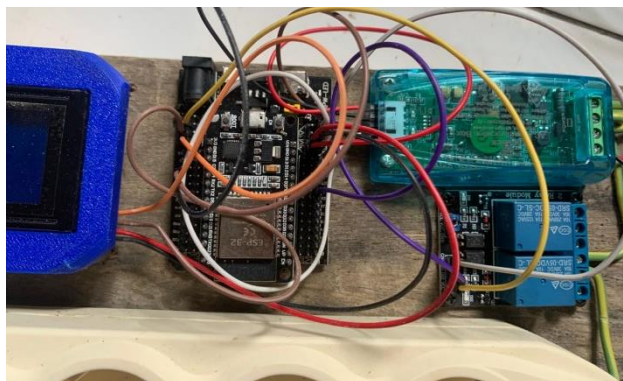
Gambar 4. 4 Rangkaian Relay Module

Keterangan koneksi:

1. VCC Relay → 5V
2. GND Relay → GND ESP32
3. IN Relay → GPIO 26 ESP32

4.2.4 Rangkaian ESP32

ESP32 berperan sebagai pusat kendali sistem sekaligus modul komunikasi IoT. Mikrokontroler ini menerima data dari sensor PZEM-004T melalui Serial, menampilkan data pada LCD melalui I2C, mengontrol relay melalui GPIO 26 serta mengelola komunikasi *Whatsapp* melalui koneksi WiFi bawaan untuk keperluan *monitoring* jarak jauh dan notifikasi WhatsApp.



Gambar 4. 5 Rangkaian ESP32

ESP32 menjalankan dua *task* secara paralel menggunakan arsitektur *FreeRTOS*. *Task* pertama pada *Core 0* menangani pembacaan sensor, tampilan LCD dan logika proteksi trip. *Task* kedua pada *Core 1* menangani koneksi Wifi dan komunikasi *Whatsapp* secara terus-menerus, sehingga kedua proses dapat berjalan bersamaan tanpa saling mengganggu.

4.3 Integrasi Notifikasi WhatsApp

Dalam sistem *monitoring* daya listrik berbasis IoT ini, WhatsApp digunakan sebagai media komunikasi yang memungkinkan pengelola sekolah memantau kondisi daya listrik dan mengontrol sistem secara jarak jauh secara *real-time*. Berbeda dengan sistem yang hanya merespons perintah pengguna, sistem ini juga mengirimkan notifikasi otomatis kepada pengguna ketika terjadi kondisi daya berlebih (*overcurrent*), sehingga pengelola sekolah dapat segera mengambil tindakan tanpa harus memantau sistem secara terus-menerus.

Dengan cara ini, pengelola memiliki dua kemudahan sekaligus, yaitu menerima peringatan otomatis saat terjadi gangguan, sekaligus memiliki kendali penuh untuk memeriksa status sistem melalui WhatsApp.

4.3.1 Pembuatan Konfigurasi WhatsApp

Integrasi notifikasi *WhatsApp* dilakukan menggunakan layanan *gateway WhatsApp* yang terhubung ke broker MQTT HiveMQ. *Gateway* ini bertugas menjembatani komunikasi antara broker MQTT dengan akun WhatsApp pengelola sekolah, sehingga pesan yang dipublikasikan ESP32 ke topik MQTT dapat diteruskan sebagai pesan *WhatsApp* secara otomatis.

```

11 // ===== MQTT =====
12 const char* mqttServer = "broker.hivemq.com";
13 const int mqttPort = 1883;
14 const char* topicSensor = "fani/sensor";
15 const char* topicStatus = "esp32/relay/status";
16 const char* topicCommand = "esp32/relay/command";
17 const char* topicLimit = "fani/limit";
18 const char* topicNotif = "fani/notifikasi";
19

```

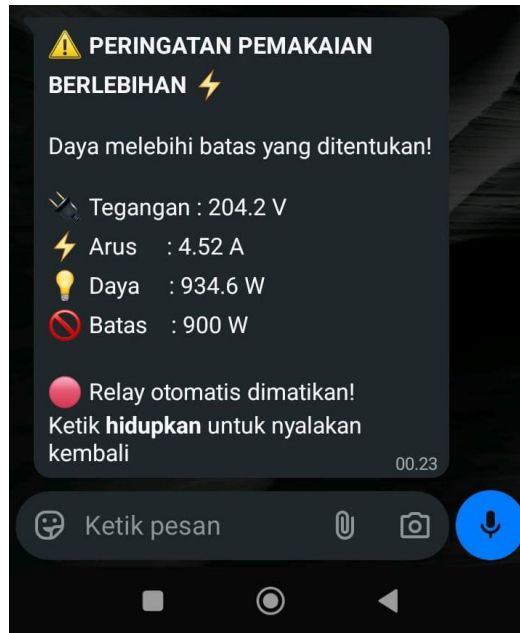
Gambar 4. 6 Konfigurasi Gateway WhatsApp

Berikut langkah-langkah konfigurasi yang dilakukan:

1. Menyiapkan layanan gateway WhatsApp yang mendukung komunikasi menggunakan protokol MQTT sebagai penghubung antara sistem *monitoring* dan aplikasi *WhatsApp*.
2. Mengonfigurasi *gateway WhatsApp* untuk berlangganan (*subscribe*) pada topik *fani/notifikasi*, yang digunakan untuk menerima pesan notifikasi peringatan yang dikirimkan oleh ESP32 berdasarkan hasil pembacaan sensor.
3. Mengonfigurasi *gateway WhatsApp* untuk berlangganan pada topik *esp32/relay/status*, yang digunakan untuk menerima informasi status relay sebagai bentuk konfirmasi terhadap kondisi sistem.

4.3.2 Notifikasi Otomatis Saat *Overcurrent*

Sistem akan mengirimkan notifikasi *WhatsApp* secara otomatis ketika daya terukur melebihi batas yang ditetapkan. Notifikasi dikirim setelah relay trip dan menunggu jeda 5 detik sebagai buffer untuk memastikan kondisi *overcurrent* benar-benar terjadi, bukan sekadar lonjakan sesaat.



Gambar 4. 7 Notifikasi *Overcurrent* di *WhatsApp*

Pesan notifikasi yang dikirim memuat informasi lengkap dalam format JSON yang kemudian ditampilkan sebagai pesan WhatsApp, meliputi:

1. *Voltage*: nilai tegangan saat trip (*Volt*)
2. *Current*: nilai arus saat trip (*Ampere*)
3. *Power*: nilai daya aktual saat trip (*Watt*)
4. *Limit*: nilai batas daya yang ditetapkan (*Watt*)

Notifikasi ini diterima oleh pengelola sekolah secara otomatis tanpa perlu meminta data terlebih dahulu, sehingga penanganan gangguan dapat dilakukan dengan cepat meskipun pengelola sedang tidak berada di lokasi sekolah.

4.4 Implementasi Program

Program pada sistem *monitoring* ini dijalankan pada mikrokontroler ESP32 yang berfungsi sebagai pusat pengolahan data dan komunikasi dengan jaringan

internet. ESP32 bertugas membaca data dari sensor, mengirimkan data ke broker MQTT serta mengirimkan notifikasi melalui gateway WhatsApp.

Komunikasi data antara perangkat dan sistem *monitoring* menggunakan protokol MQTT dengan broker publik HiveMQ. Setiap data sensor, perintah kontrol serta notifikasi dikirim melalui topik MQTT yang telah ditentukan.

4.4.1 Program ESP32

Program pada ESP32 dikembangkan menggunakan Arduino IDE dengan beberapa *library* pendukung untuk menghubungkan perangkat ke jaringan WiFi serta komunikasi MQTT. Beberapa *library* yang digunakan dalam program antara lain:

1. *WiFi.h* untuk menghubungkan ESP32 ke jaringan WiFi
2. *PubSubClient.h* untuk komunikasi dengan broker MQTT
3. *ArduinoJson.h* untuk kirim data sensor

4.4.2 Fungsi Program ESP32

Program yang dijalankan pada ESP32 memiliki beberapa fungsi utama sebagai berikut:

1. Menghubungkan ESP32 ke WiFi agar dapat berkomunikasi melalui internet.
2. Menghubungkan perangkat ke broker MQTT sebagai media pertukaran data.
3. Membaca data dari sensor yang terpasang pada sistem *monitoring*.
4. Mengirimkan data sensor ke broker MQTT melalui topik *fani/sensor*.
5. Menerima perintah pengendalian relay dari topik *esp32/relay/command*.
6. Mengirimkan informasi status relay ke topik *esp32/relay/status*.
7. Mengirimkan pesan peringatan ke topik *fani/notifikasi* yang kemudian diteruskan ke pengguna melalui *gateway WhatsApp*.

4.4.3 Pengujian Sistem

Setelah program berhasil diimplementasikan pada mikrokontroler ESP32, tahap selanjutnya adalah melakukan pengujian sistem secara keseluruhan. Pengujian ini bertujuan untuk memastikan bahwa setiap komponen pada sistem *monitoring* dapat bekerja dengan baik serta mampu berkomunikasi melalui protokol MQTT.

Pengujian dilakukan terhadap beberapa bagian utama sistem, yaitu koneksi WiFi pada ESP32, komunikasi dengan broker MQTT, pengiriman data sensor, pengendalian relay serta pengiriman notifikasi melalui *gateway WhatsApp*.

Proses pengujian dilakukan dengan menjalankan sistem secara langsung dan mengamati respon perangkat ketika data sensor dikirim ke broker MQTT serta ketika perintah pengendalian relay diterima dari sistem *monitoring*. Berikut merupakan contoh hasil pengujian komunikasi data pada sistem.

Tabel 4. 2 Hasil Pengujian Komunikasi MQTT

No	Kondisi Pengujian	Topik MQTT	Hasil
1	ESP32 terhubung ke WiFi	-	Berhasil
2	Pengiriman data sensor	fani/sensor	Berhasil
3	Pengiriman status relay	esp32/relay/status	Berhasil
4	Penerimaan perintah relay	esp32/relay/command	Berhasil
5	Pengiriman notifikasi	fani/notifikasi	Berhasil

Berdasarkan hasil pengujian pada Tabel 4.2, dapat disimpulkan sistem mampu:

1. Menghubungkan ESP32 ke jaringan internet melalui WiFi.
2. Mengirimkan data sensor ke broker MQTT secara *real-time*.
3. Menerima perintah pengendalian relay dari sistem *monitoring*.
4. Mengirimkan status relay sebagai konfirmasi kondisi perangkat.
5. Mengirimkan pesan notifikasi yang diteruskan ke pengguna melalui WhatsApp.

Dengan demikian, sistem *monitoring* berbasis *Internet of Things* yang dirancang dapat berfungsi dengan baik sesuai dengan tujuan penelitian.

4.4.4 Hasil Implementasi

Setelah seluruh proses perancangan perangkat keras dan perangkat lunak selesai dilakukan, sistem kemudian diuji untuk memastikan bahwa seluruh komponen dapat bekerja secara terintegrasi. Pengujian dilakukan terhadap komunikasi MQTT, pengendalian relay serta sistem notifikasi WhatsApp.

1. Komunikasi MQTT merupakan bagian utama dalam sistem ini karena berfungsi sebagai media pertukaran data antara ESP32 dan sistem monitoring. ESP32 dikonfigurasi untuk terhubung dengan broker MQTT publik **HiveMQ** melalui jaringan internet. Pada sistem ini terdapat beberapa topik MQTT yang digunakan, yaitu:

- a. **fani/sensor** untuk mengirimkan data sensor dari ESP32
- b. **esp32/relay/status** untuk mengirimkan status relay
- c. **fani/limit** untuk menampilkan nilai batas yang digunakan pada sistem
- d. **fani/notifikasi** untuk mengirimkan pesan peringatan

Hasil pengujian menunjukkan bahwa komunikasi antara ESP32 dan broker MQTT dapat berjalan dengan baik dan stabil.

2. Mekanisme Pemutusan Relay Otomatis, pada sistem ini berfungsi sebagai pengaman yang akan memutus aliran listrik secara otomatis apabila nilai parameter yang dipantau melebihi batas yang telah ditentukan. Pada kondisi normal, relay akan tetap dalam keadaan aktif sehingga perangkat listrik dapat bekerja seperti biasa. Namun ketika nilai sensor mencapai atau melebihi nilai

batas (*limit*), sistem akan secara otomatis mematikan relay untuk mencegah terjadinya kondisi yang tidak diinginkan.

Hasil pengujian menunjukkan bahwa:

- a. Relay tetap aktif ketika nilai sensor berada di bawah batas yang ditentukan.
- b. Relay akan mati secara otomatis ketika nilai sensor mencapai batas sistem.
- c. Status perubahan relay berhasil dikirimkan ke broker MQTT melalui topik `esp32/relay/status`.

Hal ini menunjukkan bahwa mekanisme pengamanan sistem bekerja sesuai dengan logika program yang telah dirancang.

3. Sistem notifikasi WhatsApp digunakan untuk memberikan informasi kepada pengguna ketika terjadi kondisi tertentu pada sistem *monitoring*. ESP32 akan mengirimkan pesan notifikasi ke broker MQTT melalui topik **fani/notifikasi**. Pesan tersebut kemudian diterima oleh *gateway WhatsApp* yang telah berlangganan (*subscribe*) pada topik tersebut, dan selanjutnya diteruskan ke pengguna melalui aplikasi WhatsApp.

Contoh pesan notifikasi yang dikirimkan oleh sistem adalah sebagai berikut:

“⚠ PERINGATAN PEMAKAIAN BERLEBIHAN ⚡

Daya melebihi batas yang ditentukan!”

Hasil pengujian menunjukkan bahwa:

- a. Pesan notifikasi berhasil dikirim dari ESP32 ke broker MQTT
- b. Gateway WhatsApp berhasil menerima pesan dari topik MQTT
- c. Pesan notifikasi berhasil diteruskan kepada pengguna melalui WhatsApp

Dengan demikian, sistem notifikasi yang dirancang mampu memberikan informasi secara cepat kepada pengguna mengenai kondisi sistem.