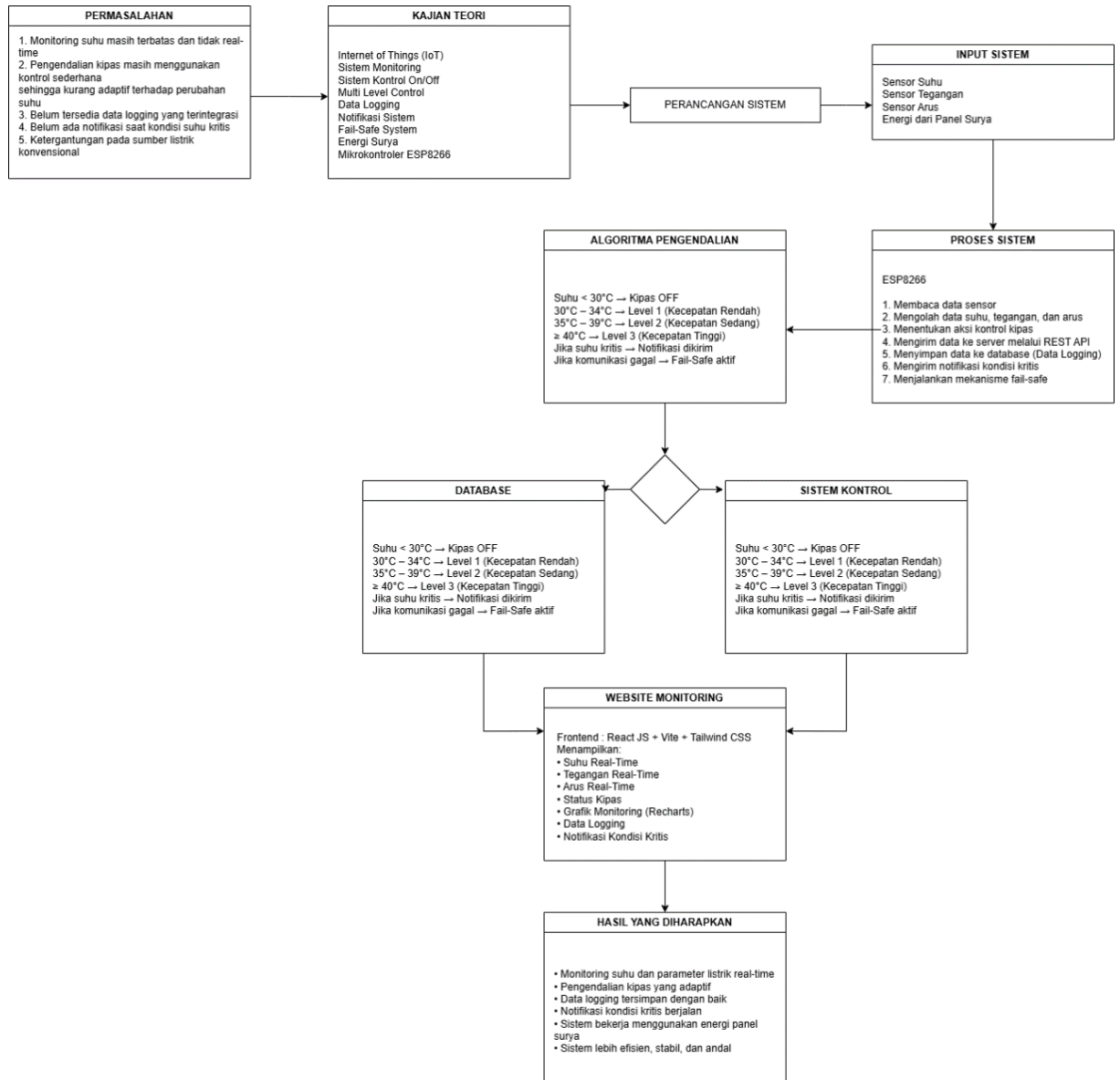


BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Berfikir



Gambar 3. 1 Kerangka berfikir

3.2 Metode Penelitian

Penelitian ini menggunakan metode *Research and Development* (R&D) atau metode rancang bangun, yaitu metode penelitian yang bertujuan untuk merancang,

mengembangkan, mengimplementasikan, dan menguji suatu produk atau sistem sehingga dapat berfungsi sesuai dengan kebutuhan yang telah ditentukan. Metode ini dipilih karena penelitian berfokus pada pengembangan sistem *monitoring* suhu dan pengendalian kipas berbasis *Internet of Things* (IoT) yang terintegrasi dengan *Data logging*, notifikasi, *Multi level control*, dan panel surya sebagai sumber energi.

Menurut (Shindika Sari et al., 2023), Metode penelitian rancang bangun digunakan untuk membuat produk tertentu dan menguji tingkat efektivitas produk tersebut. Produk yang dikembangkan dalam penelitian ini adalah sistem pemantauan suhu dan pengendalian kipas berbasis IoT yang dapat memantau parameter listrik dan suhu secara real-time melalui web, mengendalikan kipas secara otomatis berdasarkan rentang suhu tertentu, menyimpan hasil pengukuran ke dalam database, dan menggunakan panel surya sebagai sumber energi utama.

Analisis kebutuhan, perancangan sistem, implementasi perangkat keras dan perangkat lunak, integrasi sistem, pengujian, dan evaluasi kinerja adalah langkah-langkah yang dilakukan secara sistematis untuk memastikan bahwa sistem yang dikembangkan dapat berfungsi sesuai dengan tujuan penelitian.

3.2.1 Tahapan Penelitian

Tahapan penelitian yang digunakan dalam pengembangan sistem ini terdiri dari beberapa Langkah sebagai berikut:

a. Analisis kebutuhan

Pada tahap ini dilakukan identifikasi kebutuhan sistem yang meliputi kebutuhan perangkat keras (hardware) dan perangkat lunak (software). Analisis dilakukan untuk menentukan spesifikasi komponen yang digunakan, seperti mikrokontroler ESP8266, sensor suhu, sensor tegangan, sensor arus, panel surya, baterai, serta teknologi *website* yang digunakan dalam sistem *monitoring*.

b. Perancangan Sistem

Tahap perancangan dilakukan dengan membuat desain arsitektur sistem, diagram alir sistem, perancangan basis data, perancangan antarmuka *website*, serta

perancangan algoritma pengendalian kipas menggunakan metode *ON/OFF* dan *Multi level control*.

Perancangan sistem meliputi:

1. Perancangan perangkat keras (Hardware)
2. Perancangan perangkat lunak (Software)
3. Perancangan komunikasi data IoT menggunakan REST API
4. Perancangan *Database Data logging*
5. Perancangan *Dashboard monitoring* berbasis *website*
6. Perancangan mekanisme notifikasi dan *fail safe system*

c. Implementasi Sistem

Tahap implementasi dilakukan dengan merealisasikan seluruh rancangan yang telah dibuat menjadi sistem yang dapat beroperasi. Implementasi perangkat keras dilakukan dengan merakit dan menghubungkan seluruh komponen seperti panel surya, TP4056, baterai 18650, *buck converter*, ESP8266 NodeMCU, sensor DHT22, sensor INA219, relay, MOSFET, transistor, dioda, dan kipas DC sesuai dengan rancangan yang telah dibuat.

Pada implementasi perangkat lunak dilakukan pengembangan program ESP8266 menggunakan Arduino IDE untuk membaca data sensor, mengirim data ke server, serta menjalankan algoritma pengendalian kipas. Selain itu, dibangun *website monitoring* menggunakan React JS dengan dukungan Tailwind CSS dan Recharts untuk menampilkan data *monitoring* secara *realtime* dalam bentuk tabel dan grafik. Basis data MySQL digunakan untuk menyimpan data hasil *monitoring* dan *Data logging* yang dikirimkan melalui REST API.

d. Integrasi Sistem

Tahap integrasi dilakukan dengan menghubungkan seluruh komponen perangkat keras dan perangkat lunak sehingga dapat bekerja sebagai satu sistem yang utuh. Sensor DHT22 dan INA219 diintegrasikan dengan ESP8266 untuk

memperoleh data suhu, tegangan, dan arus secara *realtime*. Data yang diperoleh kemudian dikirimkan ke server melalui REST API dan disimpan pada *Database*.

Website monitoring diintegrasikan dengan *Database* sehingga dapat menampilkan data secara *realtime*, riwayat data *monitoring*, grafik perubahan parameter, serta status sistem. Pada tahap ini juga dilakukan integrasi antara sistem *monitoring* dengan sistem pengendalian kipas menggunakan metode *Multi level control*, serta integrasi panel surya dan baterai sebagai sumber energi utama yang menyuplai kebutuhan daya seluruh sistem.

e. Pengujian Sistem

Setelah sistem berhasil diintegrasikan, dilakukan pengujian untuk mengetahui tingkat keberhasilan dan kinerja sistem yang telah dikembangkan. Pengujian dilakukan terhadap seluruh fungsi utama sistem, meliputi akurasi sensor suhu DHT22, akurasi sensor INA219 dalam mengukur tegangan dan arus, keberhasilan pengiriman data dari ESP8266 ke server, serta kemampuan *website* dalam menampilkan data *monitoring* secara *realtime*.

Pengujian juga dilakukan terhadap fitur *Data logging* untuk memastikan data tersimpan dengan baik pada *Database*. Selain itu, dilakukan pengujian sistem pengendalian kipas menggunakan metode *ON/OFF* dan *Multi level control* untuk mengetahui respons sistem terhadap perubahan suhu. Pengujian panel surya dilakukan untuk mengevaluasi kemampuan panel dalam menyuplai kebutuhan energi sistem, sedangkan pengujian *fail safe system* dilakukan melalui simulasi kondisi gangguan, seperti kegagalan sensor, gangguan komunikasi jaringan, dan kondisi suhu kritis.

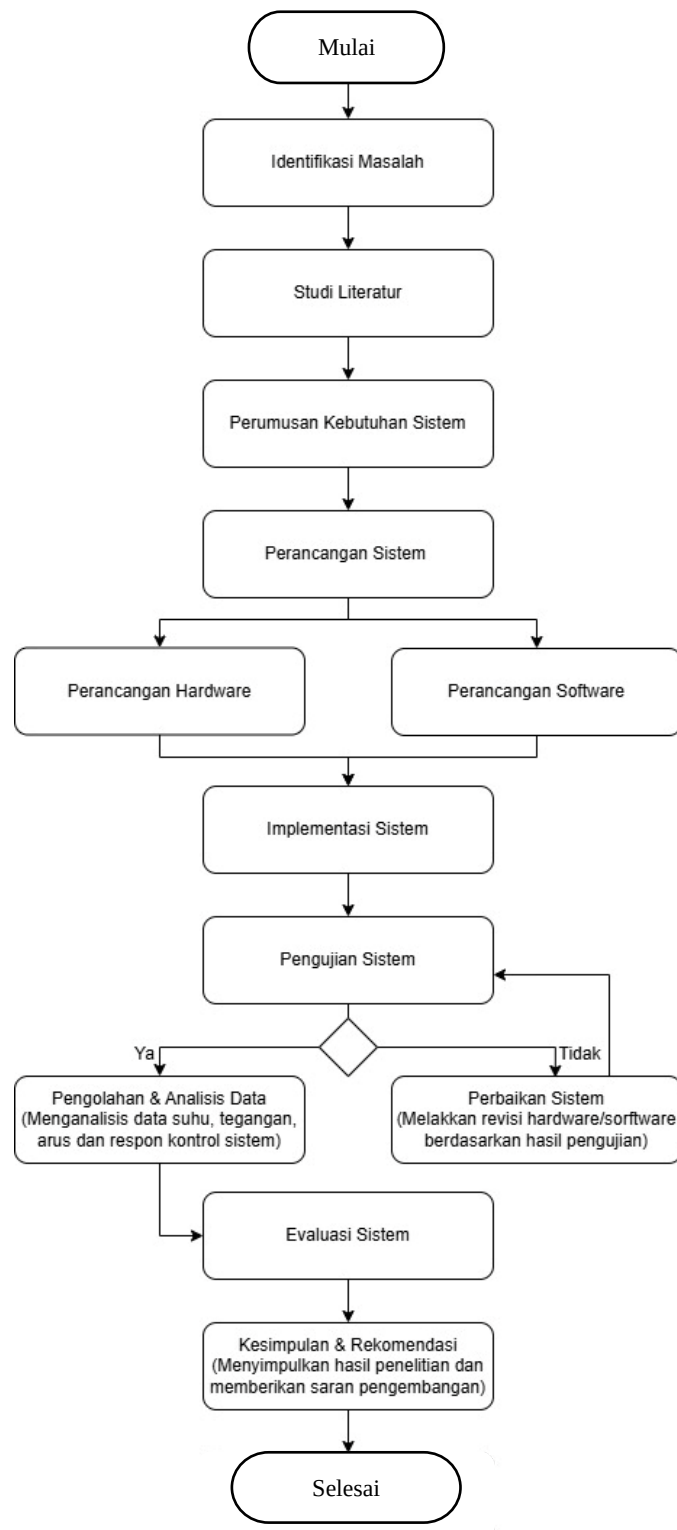
f. Evaluasi dan Analisis Hasil

Tahap terakhir adalah evaluasi dan analisis hasil pengujian yang telah dilakukan. Evaluasi bertujuan untuk mengetahui tingkat keberhasilan sistem berdasarkan parameter yang telah ditetapkan pada tahap perancangan. Analisis

dilakukan terhadap akurasi pembacaan sensor, kecepatan pengiriman data, performa *Data logging*, respons pengendalian kipas, efektivitas notifikasi, serta kemampuan panel surya dalam menyuplai energi bagi sistem.

Hasil analisis digunakan untuk menilai keandalan sistem secara keseluruhan serta mengetahui apakah sistem *monitoring* suhu dan pengendalian kipas berbasis IoT yang dikembangkan telah mampu memenuhi tujuan penelitian. Selanjutnya, hasil evaluasi tersebut dijadikan dasar dalam penyusunan kesimpulan dan rekomendasi untuk pengembangan sistem pada penelitian selanjutnya.

3.3 Prosedur Penelitian



Gambar 3. 2 prosedur Penelitian

Prosedur penelitian merupakan tahapan yang dilakukan secara sistematis untuk mencapai tujuan penelitian. Tahapan penelitian ini disusun secara berurutan mulai dari identifikasi permasalahan hingga penarikan kesimpulan berdasarkan hasil pengujian dan analisis sistem yang dikembangkan.

3.3.1 Identifikasi Masalah

Pada tahap ini dilakukan pengamatan dan analisis terhadap sistem *monitoring* suhu dan pengendalian kipas yang telah ada. Hasil pengamatan menunjukkan bahwa sebagian besar sistem masih menggunakan metode pengendalian sederhana yang kurang adaptif terhadap perubahan suhu lingkungan. Selain itu, sistem yang tersedia umumnya belum dilengkapi dengan fitur *Data logging*, notifikasi kondisi kritis, serta pemanfaatan energi terbarukan sebagai sumber daya utama. Permasalahan tersebut menjadi dasar dalam pengembangan sistem yang diusulkan pada penelitian ini.

3.3.2 Studi Literatur

Tahap studi literatur dilakukan dengan mengumpulkan berbagai referensi yang berkaitan dengan penelitian, baik berupa jurnal ilmiah, buku, prosiding, maupun sumber akademik lainnya. Kajian literatur difokuskan pada konsep *Internet of Things (IoT)*, sistem *monitoring*, sistem kontrol *ON/OFF* dan *Multi level control*, sensor DHT22, sensor INA219, mikrokontroler ESP8266, *Data logging*, notifikasi sistem, *fail safe system*, serta pemanfaatan panel surya sebagai sumber energi. Studi literatur bertujuan untuk memperoleh landasan teoritis yang mendukung proses perancangan dan pengembangan sistem.

3.3.3 Analisis Kebutuhan Sistem

Pada tahap ini dilakukan identifikasi kebutuhan sistem berdasarkan hasil analisis masalah dan studi literatur. Kebutuhan sistem meliputi fungsi *monitoring* suhu, pengukuran parameter listrik berupa tegangan dan arus, pengendalian kipas otomatis menggunakan metode *ON/OFF* dan *Multi level control*, penyimpanan data *monitoring* secara otomatis melalui *Data logging*, penyampaian notifikasi kondisi kritis, serta penerapan mekanisme *fail safe* untuk menjaga keamanan sistem. Tahap

ini juga mencakup penentuan kebutuhan perangkat keras dan perangkat lunak yang akan digunakan dalam penelitian.

3.3.4 Perancangan Sistem

Tahap perancangan sistem dilakukan dengan menyusun desain perangkat keras dan perangkat lunak yang akan digunakan. Perancangan perangkat keras meliputi penyusunan arsitektur sistem, diagram blok, dan rangkaian elektronik yang terdiri atas panel surya, TP4056, baterai 18650, *buck converter*, ESP8266 NodeMCU, sensor DHT22, sensor INA219, relay, MOSFET IRLZ44N, transistor, dioda, dan kipas DC. Perancangan perangkat lunak meliputi penyusunan flowchart sistem, perancangan basis data, perancangan REST API, serta perancangan antarmuka *website monitoring* menggunakan ReactJS.

3.3.5 Implementasi Sistem

Pada tahap ini dilakukan proses realisasi seluruh rancangan menjadi sebuah prototipe yang dapat beroperasi. Implementasi perangkat keras dilakukan dengan merakit seluruh komponen sesuai rancangan yang telah dibuat. Selanjutnya dilakukan pemrograman ESP8266 untuk membaca data sensor, mengendalikan kipas, dan mengirimkan data ke server. Pada sisi perangkat lunak dilakukan pembangunan *website monitoring*, basis data, serta layanan REST API yang digunakan untuk komunikasi data antara perangkat IoT dan server.

3.3.6 Pengujian Sistem

Tahap pengujian dilakukan untuk mengetahui tingkat keberhasilan dan kinerja sistem yang telah dikembangkan. Pengujian mencakup pengujian akurasi sensor DHT22 dalam membaca suhu, pengujian sensor INA219 dalam mengukur tegangan dan arus, pengujian komunikasi data antara ESP8266 dan server, pengujian *Data logging*, pengujian *website monitoring*, serta pengujian mekanisme notifikasi. Selain itu, dilakukan pengujian metode *ON/OFF* dan *Multi level control* untuk mengetahui kemampuan sistem dalam mengendalikan kecepatan kipas berdasarkan perubahan suhu. Pengujian panel surya dan mekanisme *fail safe system* juga dilakukan untuk memastikan sistem dapat beroperasi secara stabil dan aman.

3.3.7 Pengumpulan Data

Pada tahap ini dilakukan pengambilan data hasil pengujian yang meliputi data suhu, tegangan, arus, status kipas, waktu respons sistem, serta data operasional panel surya. Data dikumpulkan secara otomatis melalui mekanisme *Data logging* dan disimpan ke dalam basis data untuk keperluan analisis lebih lanjut.

3.3.8 Analisis Data

Data yang diperoleh dari hasil pengujian kemudian dianalisis menggunakan pendekatan kuantitatif. Analisis dilakukan untuk mengetahui tingkat akurasi sensor, kecepatan respons sistem kontrol, efektivitas metode *Multi level control*, keberhasilan proses *Data logging*, performa *website monitoring*, serta kemampuan panel surya dalam menyuplai energi bagi sistem. Hasil analisis digunakan untuk mengevaluasi kinerja sistem secara keseluruhan berdasarkan parameter yang telah ditentukan.

3.3.9 Penarikan Kesimpulan dan Rekomendasi

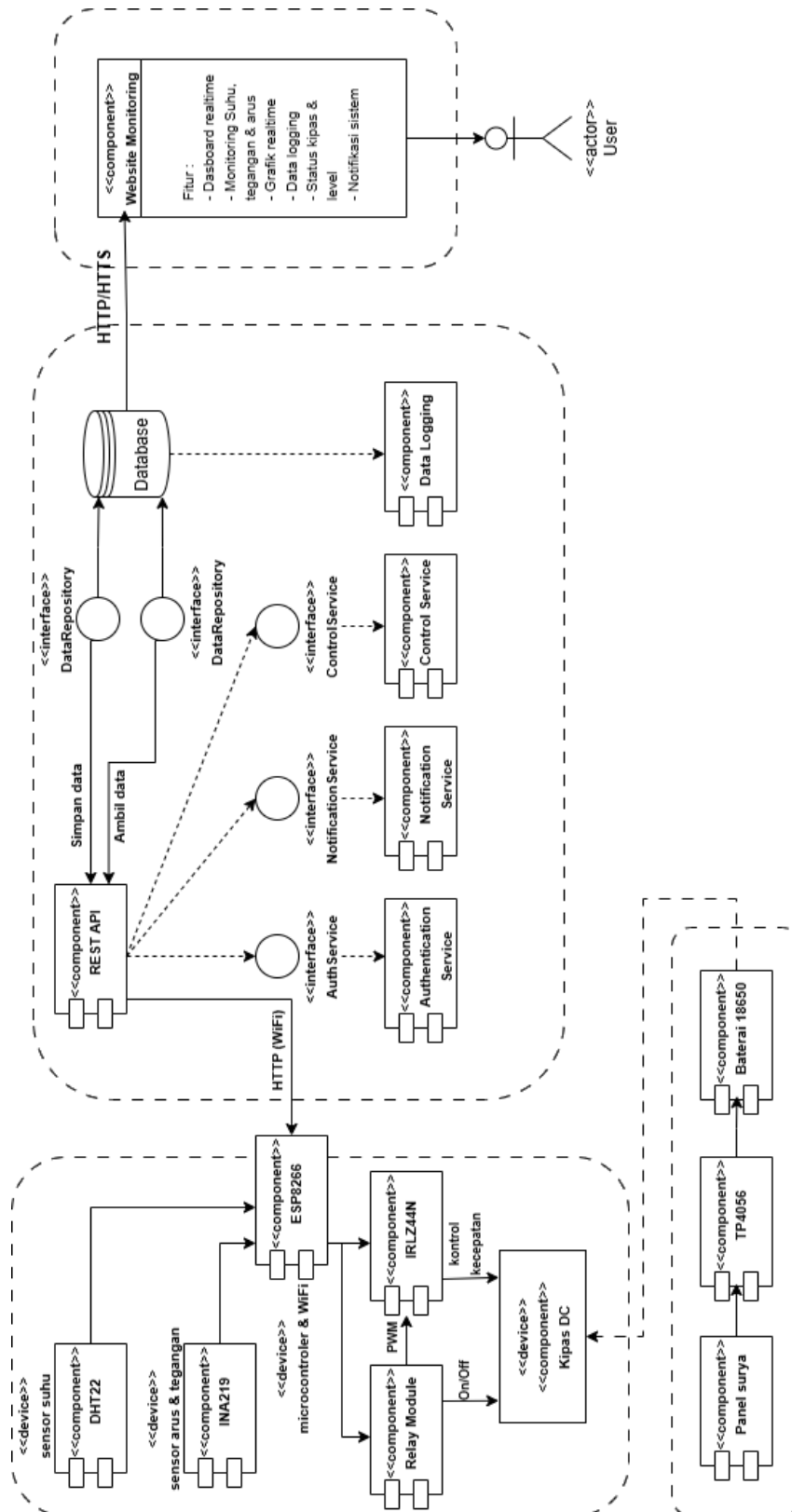
Tahap terakhir adalah penarikan kesimpulan berdasarkan hasil analisis yang telah dilakukan. Kesimpulan disusun untuk menjawab rumusan masalah dan tujuan penelitian yang telah ditetapkan. Selain itu, diberikan rekomendasi sebagai bahan pengembangan pada penelitian selanjutnya, baik dari aspek perangkat keras, perangkat lunak, metode pengendalian, maupun peningkatan fitur sistem agar diperoleh performa yang lebih optimal.

3.4 Rancangan Sistem

Bagian ini menjelaskan rancangan sistem *monitoring* suhu dan pengendalian kipas otomatis berbasis IoT yang akan dibangun. Penjelasan meliputi analisis kebutuhan sistem, flowchart sistem, perancangan UML (*Use case diagram*, *activity diagram*, dan *sequence diagram*), serta perancangan antarmuka pengguna (interface) pada *website monitoring*. Rancangan ini menjadi acuan dalam tahap implementasi sistem, baik pada sisi perangkat keras (ESP8266, sensor DHT22, dan sensor INA219) maupun pada sisi perangkat lunak (server, *Database*, dan *website monitoring*).

3.4.1 Arsitektur Sistem

Arsitektur sistem memperlihatkan gambaran menyeluruh mengenai struktur sistem *monitoring* suhu dan pengendalian kipas otomatis yang berbasis *Internet of Things* (IoT). Diagram ini menunjukkan keterkaitan antar komponen, mulai dari perangkat (edge device), sumber daya, komunikasi data, backend server, *Database*, sampai dengan *website monitoring*. Melalui arsitektur ini, dapat diketahui bagaimana alur perpindahan data serta proses pengendalian sistem berlangsung secara keseluruhan. Arsitektur sistem pada penelitian ini dapat dilihat pada gambar berikut.



Gambar 3. 3 Arsitektur Sistem

1. Layer Perangkat (*Edge Device*)

Layer ini menjadi bagian inti yang menangani proses akuisisi data sekaligus pengendalian perangkat. Sensor DHT22 berfungsi membaca suhu lingkungan, sementara sensor INA219 digunakan untuk mengukur tegangan, arus, serta daya pada sistem. Keseluruhan data sensor tersebut diolah oleh NodeMCU ESP8266 selaku mikrokontroler utama. Berdasarkan hasil pembacaan suhu, NodeMCU akan mengatur Relay Module dan Driver MOSFET IRLZ44N untuk mengendalikan kondisi *ON/OFF* serta mengatur kecepatan putaran kipas DC melalui metode *Multi level control*.

2. Layer Sumber Daya

Layer ini berperan sebagai penyedia energi bagi keseluruhan sistem. Panel surya berfungsi mengonversi energi cahaya matahari menjadi energi listrik, yang kemudian disalurkan ke modul TP4056 untuk proses pengisian baterai *rechargeable* 18650. Energi yang tersimpan pada baterai selanjutnya dimanfaatkan sebagai catu daya, sehingga sistem tetap dapat beroperasi meskipun intensitas cahaya matahari berkurang.

3. Layer Komunikasi dan Aplikasi (*Backend*)

Layer ini berperan menghubungkan perangkat IoT dengan *website monitoring*. NodeMCU ESP8266 mengirimkan data sensor melalui jaringan *WiFi* dengan protokol HTTP menuju REST API. REST API kemudian mengolah data tersebut, menyimpannya ke dalam *Database MySQL*, serta menyediakan layanan untuk pengambilan data *monitoring*, pengelolaan *Data logging*, pengaturan set point, dan pengiriman notifikasi sistem apabila terdeteksi kondisi tertentu.

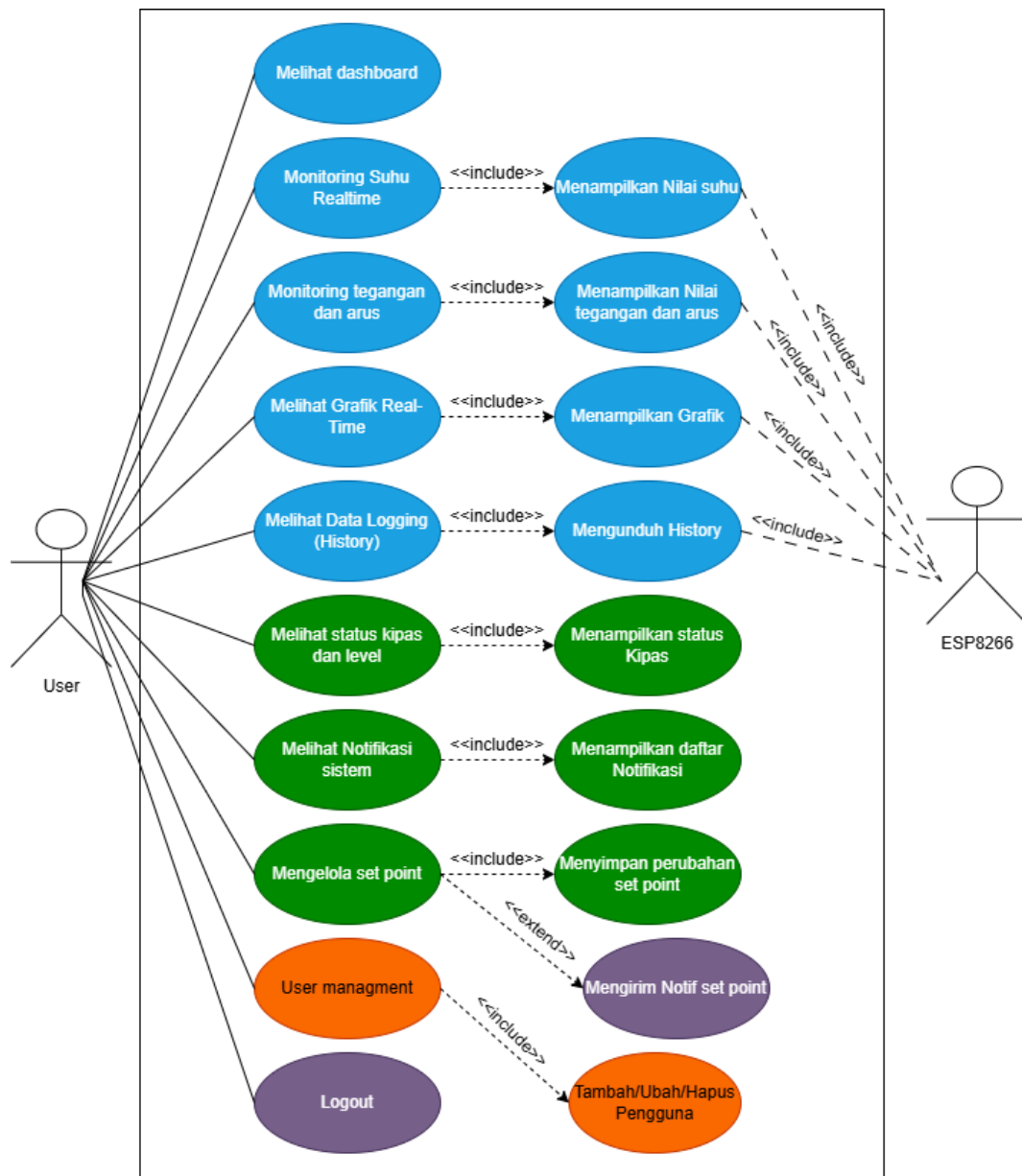
4. Layer Presentasi (*Frontend*)

Layer ini merupakan antarmuka yang dipakai pengguna dalam melakukan *monitoring* sistem melalui *website*. *Website* menyajikan berbagai informasi, seperti *Dashboard realtime*, data suhu, tegangan, arus, grafik *monitoring*, *Data logging (history)*, status dan level kipas, notifikasi sistem, pengaturan set point, hingga manajemen pengguna. Seluruh data tersebut diperoleh dari

Database melalui REST API, sehingga informasi yang ditampilkan senantiasa diperbarui secara *realtime*.

3.4.2 Use case Diagram

Use case diagram berfungsi untuk menggambarkan bentuk interaksi antara pengguna (user) dengan sistem pemantauan suhu serta pengendalian kipas otomatis yang berbasis teknologi IoT. Melalui diagram ini, dapat terlihat berbagai fungsi utama yang tersedia bagi user beserta keterkaitan antar proses yang berlangsung di dalam sistem. Gambaran *Use case* diagram pada penelitian ini disajikan pada gambar berikut.



Gambar 3. 4 Use case diagram sistem

1. Use case Melihat Dashboard

Menggambarkan kegiatan user saat membuka halaman utama *website monitoring* guna memperoleh gambaran umum mengenai kondisi sistem.

2. Use case Monitoring Suhu Realtime

Menggambarkan proses pemantauan suhu secara langsung (*realtime*) berdasarkan data yang dihasilkan oleh sensor DHT22. Proses ini memiliki hubungan include dengan proses penampilan nilai suhu.

3. *Use case Monitoring Tegangan dan Arus*

Menggambarkan proses pemantauan besaran tegangan dan arus listrik melalui sensor INA219, yang mencakup hubungan include terhadap proses penampilan nilai tegangan dan arus tersebut.

4. *Use case Melihat Grafik Realtime*

Menggambarkan aktivitas user dalam mengamati grafik hasil pemantauan suhu, arus, dan tegangan yang ditampilkan secara *realtime* pada *website*.

5. *Use case Melihat Data logging (History)*

Menggambarkan proses penelusuran data riwayat pemantauan yang sebelumnya telah disimpan dalam *Database* sistem.

6. *Use case Melihat Status Kipas dan Level*

Menggambarkan proses pengecekan kondisi kipas beserta tingkat kecepatan yang sedang berjalan pada sistem *monitoring*.

7. *Use case Melihat Notifikasi Sistem*

Menggambarkan aktivitas user dalam memeriksa notifikasi atau peringatan yang muncul ketika sistem mendeteksi kondisi tertentu.

8. *Use case Mengelola Set Point*

Menggambarkan proses penentuan batas nilai suhu yang dijadikan patokan dalam mekanisme kontrol otomatis kipas.

9. *Use case User Management*

Menggambarkan proses pengelolaan akun pengguna, mencakup penambahan, perubahan, maupun penghapusan data user dalam sistem.

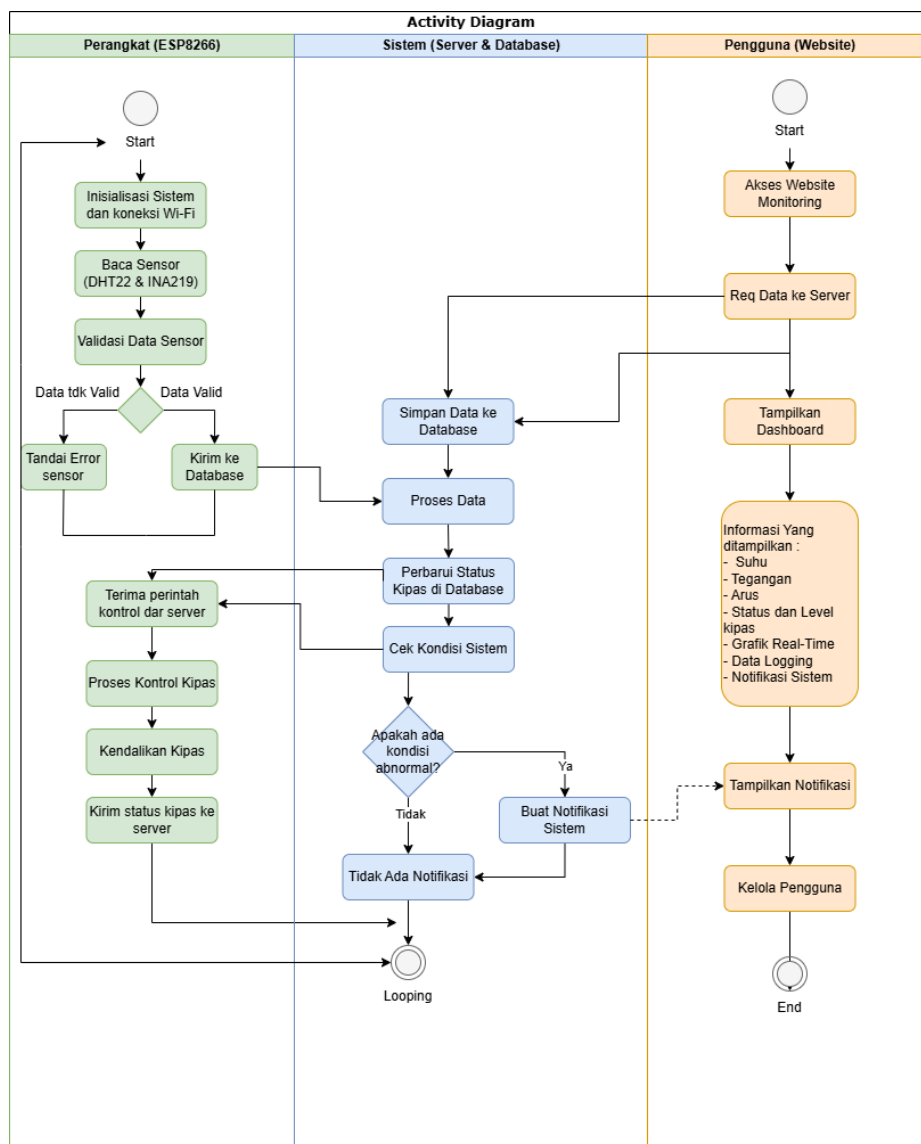
10. *Use case Logout*

Menggambarkan proses keluarnya user dari sistem *monitoring* berbasis *website*.

Dalam *Use case* diagram ini, terdapat dua aktor utama, yaitu User dan ESP8266. User berperan langsung dalam berinteraksi dengan *website monitoring* untuk mengamati data maupun mengatur konfigurasi sistem, sementara ESP8266 bertindak sebagai perangkat inti yang bertugas mengirimkan data *monitoring* ke sistem secara *realtime*.

3.4.3 Activity Diagram

Activity diagram menampilkan tahapan alur kerja sistem pemantauan suhu dan pengendalian kipas otomatis berbasis IoT, dimulai dari proses pembacaan sensor, pengiriman data, pengolahan data, sampai dengan penyajian informasi pada *website monitoring*. Diagram ini memperlihatkan bentuk interaksi antara perangkat ESP8266, server beserta *Database*, dan pengguna *website* dalam menjalankan proses *monitoring* secara *realtime*. Gambaran *activity diagram* pada penelitian ini dapat dilihat pada gambar berikut.



Gambar 3. 5 Activity diagram sistem

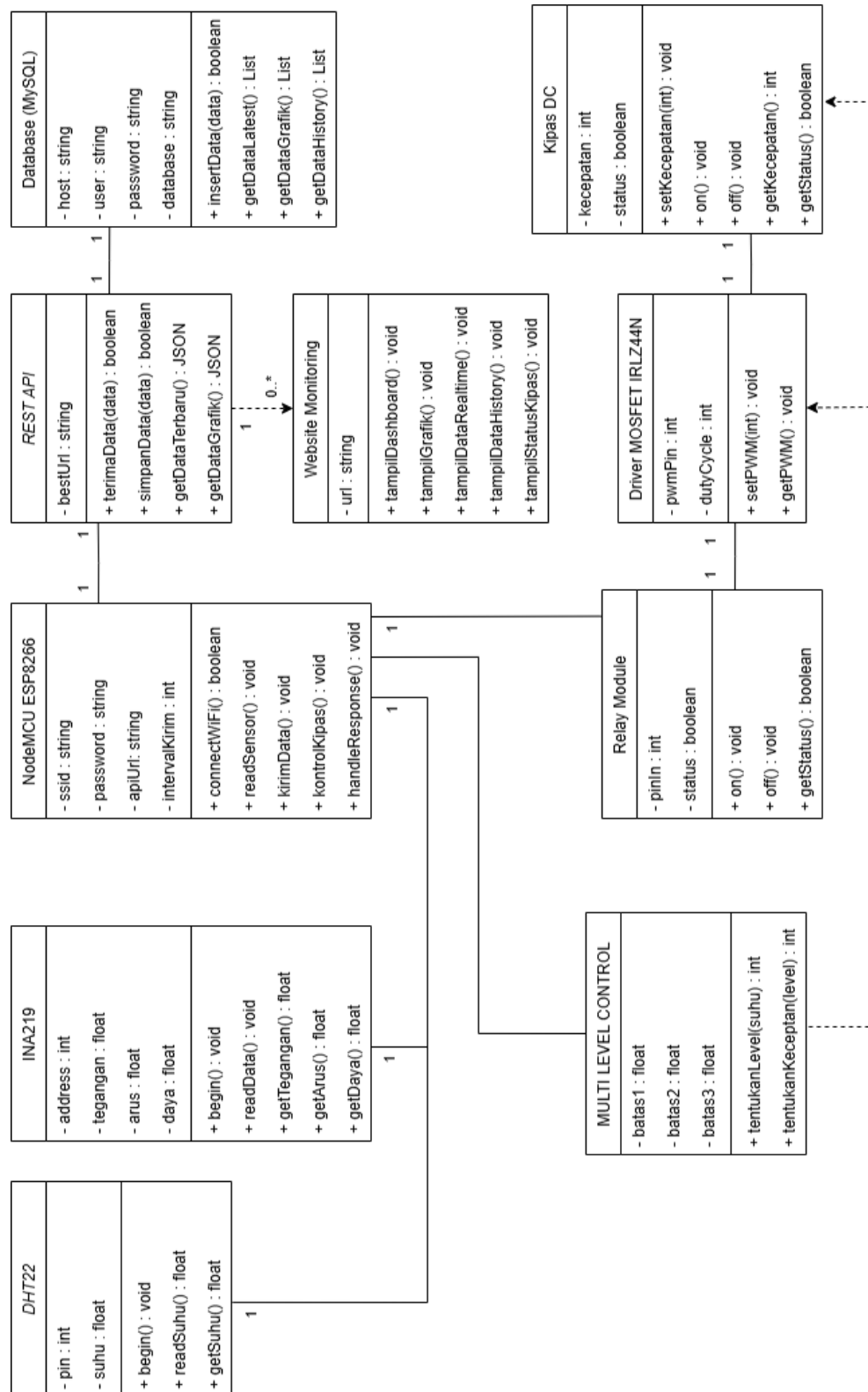
1. Aktivitas Inisialisasi Sistem dan Koneksi *WiFi*
Menggambarkan tahap awal saat ESP8266 melakukan proses inisialisasi sistem sekaligus menghubungkan perangkat tersebut ke jaringan internet.
2. Aktivitas Membaca Sensor
Menggambarkan proses pengambilan data suhu melalui sensor DHT22, serta pengambilan data arus dan tegangan melalui sensor INA219.
3. Aktivitas Validasi Data Sensor
Menggambarkan proses pemeriksaan keabsahan data sensor sebelum data tersebut diteruskan ke *Database*.
4. Aktivitas Mengirim Data ke *Database*
Menggambarkan proses pengiriman data hasil *monitoring* dari ESP8266 menuju server dan *Database* secara *realtime*.
5. Aktivitas Proses Data
Menggambarkan tahapan pengolahan data *monitoring* yang dilakukan pada sisi server sebelum data ditampilkan di *website monitoring*.
6. Aktivitas Pembaruan Status Kipas
Menggambarkan proses penyesuaian status serta level kecepatan kipas berdasarkan hasil pembacaan suhu terkini.
7. Aktivitas Cek Kondisi Sistem
Menggambarkan proses pemeriksaan terhadap kondisi sistem untuk mendeteksi ada tidaknya kondisi abnormal.
8. Aktivitas Pembuatan Notifikasi Sistem
Menggambarkan proses pembentukan notifikasi ketika terdeteksi kondisi abnormal, seperti suhu yang terlalu tinggi atau adanya gangguan pada sensor.
9. Aktivitas Menampilkan *Dashboard*
Menggambarkan proses penyajian data *monitoring* pada *website*, mencakup nilai suhu, arus, tegangan, grafik *realtime*, status kipas, *Data logging*, serta notifikasi sistem.
10. Aktivitas Kelola Pengguna

Menggambarkan proses pengaturan akun pengguna pada *website monitoring*.

Pada activity diagram ini, terdapat tiga swimlane utama, yakni perangkat ESP8266, sistem server dan *Database*, serta pengguna *website*. Ketiga elemen tersebut saling berkaitan satu sama lain dalam menjalankan proses *monitoring* serta pengendalian kipas otomatis secara *realtime* dan berkesinambungan.

3.4.4 Class Diagram

Class diagram menggambarkan struktur kelas-kelas yang menyusun sistem *monitoring* suhu dan pengendalian kipas otomatis berbasis IoT beserta atribut, metode, dan relasi antar kelas. Diagram ini digunakan untuk memperlihatkan hubungan antar komponen sistem dalam proses *monitoring*, pengiriman data, penyimpanan data, hingga pengendalian kipas otomatis menggunakan metode *Multi level control*. Pada penelitian ini, class diagram sistem dapat dilihat pada gambar berikut :



Gambar 3. 6 Class diagram sistem

Class utama pada sistem terdiri dari DHT22 sebagai sensor suhu, INA219 sebagai sensor arus dan tegangan, NodeMCU ESP8266 sebagai pengendali utama, REST API sebagai penghubung sistem, *Database MySQL* sebagai penyimpanan data, *Website monitoring* sebagai tampilan *monitoring realtime*, serta Relay dan MOSFET IRLZ44N sebagai pengendali kipas DC otomatis.

1. DHT22

Merepresentasikan sensor suhu yang digunakan untuk membaca temperatur lingkungan secara *realtime*. Kelas ini memiliki atribut pin dan suhu serta metode `readTemperature()` untuk membaca data suhu.

2. Kelas INA219

Merepresentasikan sensor arus dan tegangan pada sistem *monitoring*. Kelas ini memiliki atribut tegangan, arus, dan daya serta metode `readVoltage()` dan `readCurrent()` untuk membaca data listrik.

3. Kelas NodeMCU ESP8266

Merepresentasikan mikrokontroler utama pada sistem. Kelas ini memiliki metode `connectWiFi()`, `sendData()`, dan `controlFan()` untuk menghubungkan sistem ke internet, mengirim data *monitoring*, dan mengontrol kipas otomatis.

4. Kelas REST API

Merepresentasikan layanan penghubung antara NodeMCU, *Database*, dan *website monitoring*. Kelas ini memiliki metode `receiveData()` dan `sendJSON()` untuk menerima serta mengirim data *monitoring*.

5. Kelas *Database MySQL*

Merepresentasikan penyimpanan data *monitoring* dan *Data logging* sistem. Kelas ini memiliki metode `saveData()` dan `getData()` untuk menyimpan serta mengambil data *monitoring*.

6. Kelas *Website monitoring*

Merepresentasikan antarmuka *monitoring* berbasis web yang digunakan untuk menampilkan data *realtime*, grafik *monitoring*, dan histori data. Kelas ini memiliki metode `displayData()` dan `showGraph()`.

7. Kelas Relay Module

Merepresentasikan saklar elektronik yang digunakan untuk menghidupkan dan mematikan kipas DC secara otomatis.

8. Kelas MOSFET IRLZ44N

Merepresentasikan driver PWM yang digunakan untuk mengatur kecepatan kipas DC berdasarkan level suhu.

9. Kelas Kipas DC

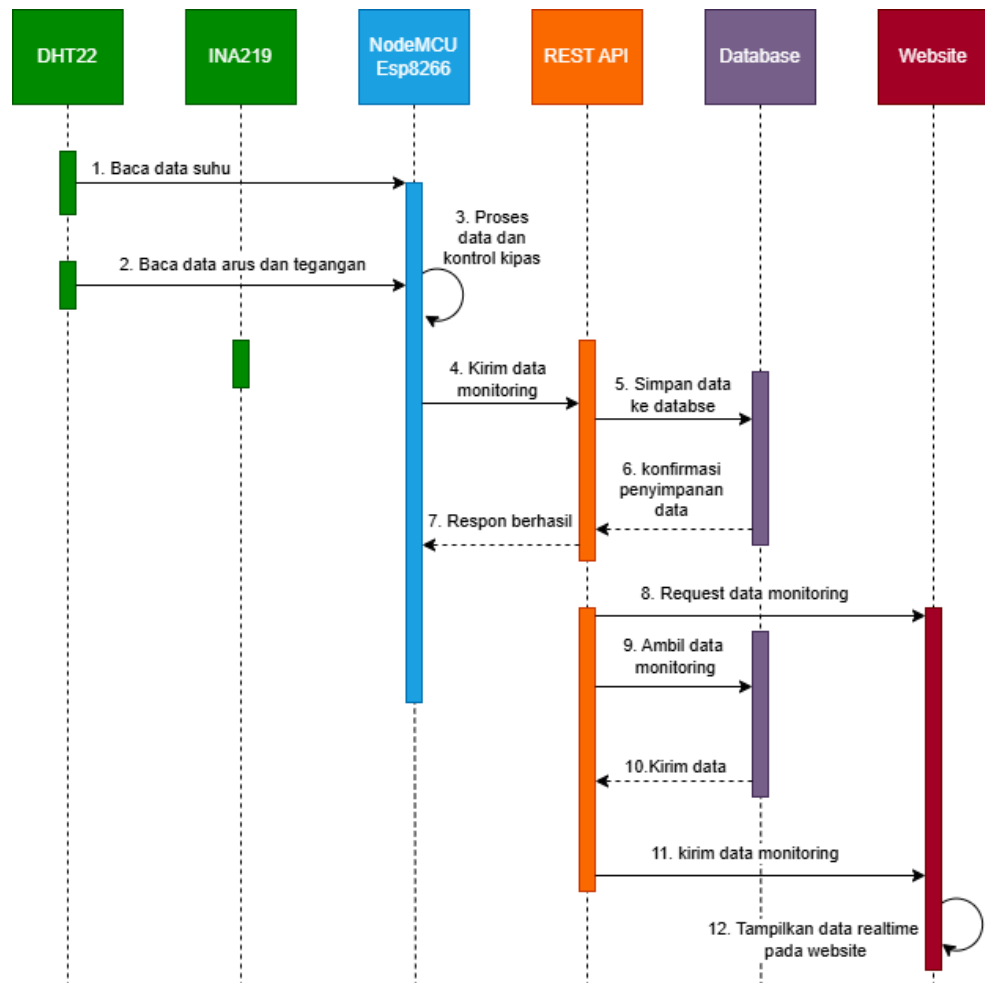
Merepresentasikan perangkat output yang digunakan untuk menurunkan suhu sistem secara otomatis.

10. Kelas *Multi level control*

Merepresentasikan sistem kontrol otomatis yang digunakan untuk menentukan level kecepatan kipas berdasarkan suhu yang terbaca oleh sensor.

3.4.5 **Sequence Diagram**

Sequence diagram digunakan untuk menggambarkan urutan interaksi yang terjadi antar komponen pada sistem pemantauan suhu dan pengendalian kipas otomatis berbasis IoT secara *realtime*. Diagram ini memperlihatkan alur komunikasi data yang dimulai dari sensor, NodeMCU ESP8266, REST API, *Database*, hingga *website monitoring*. Gambaran sequence diagram pada penelitian ini dapat dilihat pada gambar berikut.



Gambar 3. 7 *sequence* diagram sistem Monitoring

1. Objek DHT22

Merupakan sensor suhu yang berfungsi membaca kondisi temperatur lingkungan, kemudian meneruskan data tersebut ke NodeMCU ESP8266.

2. Objek INA219

Merupakan sensor arus dan tegangan yang berperan dalam pembacaan parameter kelistrikan pada sistem *monitoring*.

3. Objek NodeMCU ESP8266

Merupakan mikrokontroler utama yang bertugas menerima data dari sensor, mengolah data *monitoring*, serta mengendalikan kipas otomatis berdasarkan hasil pembacaan suhu.

4. Objek REST API

Merupakan layanan perantara yang menerima data *monitoring* dari ESP8266 dan meneruskannya ke *Database* maupun *website monitoring*.

5. Objek *Database*

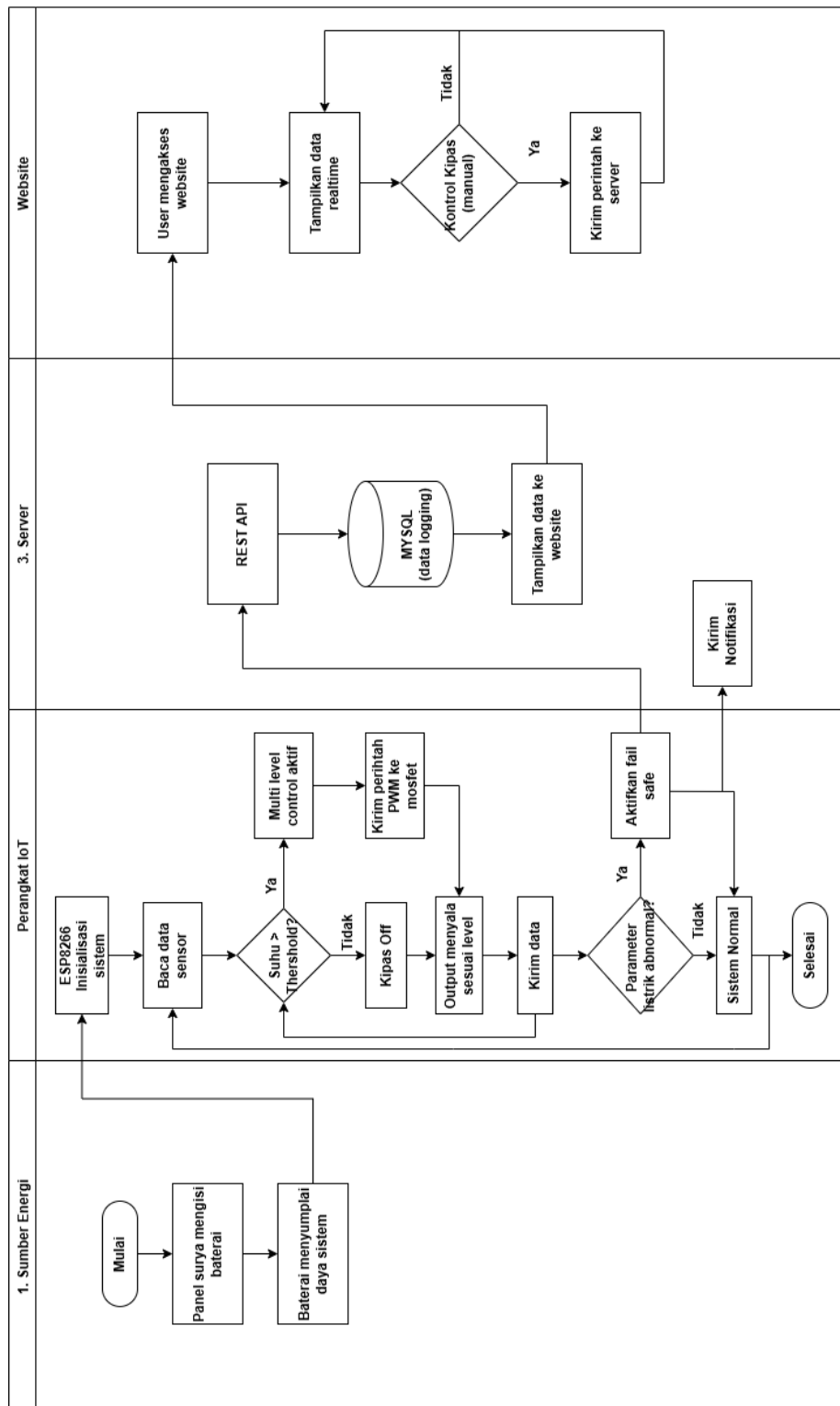
Merupakan tempat penyimpanan data *monitoring*, meliputi suhu, arus, tegangan, dan status sistem secara *realtime*.

6. Objek *Website*

Merupakan antarmuka berbasis web yang berfungsi menyajikan data *monitoring* secara *realtime* kepada pengguna.

Alur proses pada sequence diagram ini diawali dengan sensor DHT22 dan INA219 yang melakukan pembacaan data suhu, arus, dan tegangan. Data hasil pembacaan tersebut selanjutnya diolah oleh NodeMCU ESP8266 untuk keperluan *monitoring* sekaligus pengendalian kipas secara otomatis. Setelah itu, data dikirimkan ke REST API untuk disimpan ke dalam *Database*. Setelah proses penyimpanan data berhasil dilakukan, *website* akan mengirimkan permintaan (request) data *monitoring* ke server, kemudian menampilkan data tersebut secara *realtime* kepada pengguna.

3.4.6 Flowchart Sistem



Gambar 3. 8 Flowchart sistem monitoring suhu dan parameter listrik

Flowchart sistem pada Gambar 3.9 *Flowchart* sistem monitoring suhu dan parameter listrik, menggambarkan alur kerja sistem pemantauan parameter suhu dan listrik berbasis *Internet of Things* (IoT), dimulai dengan inisialisasi ESP8266, sensor DHT22, sensor INA219, relay, driver MOSFET, dan koneksi Wi-Fi. Setelah berhasil terhubung ke jaringan, ESP8266 membaca data suhu, tegangan, arus, dan daya, kemudian memvalidasi data tersebut sebelum diproses lebih lanjut. Data yang valid digunakan sebagai dasar untuk pengendalian kipas menggunakan metode On/Off dan *Multi-level* Control, di mana kecepatan kipas disesuaikan secara bertahap berdasarkan perubahan suhu yang terdeteksi oleh sensor DHT22.

Selain mengontrol kipas, sistem ini juga menerapkan mekanisme fail-safe dengan memantau parameter tegangan dan arus melalui sensor INA219. Jika kondisi listrik di luar batas aman terdeteksi, ESP8266 akan menonaktifkan relay, secara otomatis memutus pasokan daya ke kipas untuk melindungi perangkat. Selain itu, semua data pemantauan dikirim ke server melalui REST API, disimpan dalam database MySQL untuk pencatatan, dan ditampilkan secara real-time di situs web pemantauan. Selama sistem beroperasi, seluruh proses pembacaan sensor, kontrol, pemeriksaan fail-safe, dan transmisi data dilakukan secara terus-menerus sehingga status sistem dapat dipantau tanpa henti.

1. *Power System* (Sistem Catu Daya)

Tahap awal diawali oleh panel surya yang menghasilkan energi listrik saat menerima paparan cahaya matahari. Energi tersebut kemudian disalurkan ke modul TP4056 untuk melakukan pengisian pada dua buah baterai rechargeable 18650 yang dirangkai secara paralel. Selanjutnya, arus keluaran dari baterai melewati saklar *ON/OFF* yang berfungsi sebagai pengendali daya utama, sebelum akhirnya masuk ke modul LM2596. Modul LM2596 bertugas menurunkan sekaligus menstabilkan tegangan menjadi 5 V DC, sehingga dapat dimanfaatkan sebagai sumber catu daya bagi keseluruhan komponen sistem.

2. Inisialisasi Sistem

Setelah catu daya siap tersedia, ESP8266 menjalankan proses inisialisasi yang mencakup pengaturan pin I/O, inisialisasi sensor DHT22 dan INA219, konfigurasi koneksi *WiFi*, serta persiapan komunikasi dengan REST API. Tahap ini dimaksudkan agar seluruh perangkat berada dalam kondisi siap untuk menjalankan proses *monitoring* dan pengendalian.

3. Koneksi *WiFi*

Selanjutnya, ESP8266 berupaya melakukan koneksi ke jaringan *WiFi*. Jika koneksi belum berhasil terjalin, sistem akan melakukan percobaan ulang secara berulang hingga koneksi berhasil dilakukan. Setelah terhubung, sistem akan melanjutkan ke tahap pembacaan data sensor.

4. Pembacaan Data Sensor

ESP8266 membaca nilai suhu dari sensor DHT22, serta membaca nilai tegangan dan arus melalui sensor INA219. Data yang diperoleh dari kedua sensor tersebut kemudian divalidasi untuk memastikan bahwa data berada dalam kondisi normal dan layak diproses lebih lanjut.

5. Validasi Data Sensor

Data hasil pembacaan sensor diperiksa guna memastikan tidak terdapat kesalahan pembacaan maupun kegagalan fungsi sensor. Jika data dinyatakan tidak valid, sistem akan menampilkan informasi kesalahan (*error sensor*). Sebaliknya, apabila data dinyatakan valid, sistem akan melanjutkan ke tahap penentuan level kontrol.

6. *Multi level control*

Nilai suhu yang terbaca dari sensor DHT22 dijadikan dasar dalam menentukan level pengendalian motor DC melalui metode *Multi level control*. Sistem membagi kondisi suhu ke dalam beberapa level pengendalian, yakni Level 0 (motor tidak berputar), Level 1 (kecepatan rendah), Level 2 (kecepatan sedang), dan Level 3 (kecepatan tinggi). Masing-masing level menghasilkan nilai PWM

yang berbeda, sehingga kecepatan putaran motor dapat berubah secara bertahap mengikuti kondisi suhu yang terjadi.

7. Pengendalian Motor DC

Setelah level kontrol ditetapkan, ESP8266 mengirimkan sinyal kendali menuju rangkaian relay dan MOSFET IRLZ44N. Relay berperan sebagai saklar penghubung daya motor, sementara MOSFET menerima sinyal PWM dari ESP8266 untuk mengatur kecepatan putaran motor DC sesuai dengan level kontrol yang telah ditentukan.

8. Pengiriman Data ke Server

Di samping mengendalikan motor, ESP8266 juga mengirimkan data *monitoring* meliputi suhu, tegangan, arus, status motor, dan level kontrol ke REST API melalui koneksi internet.

9. Penyimpanan Data

REST API menerima data yang dikirim oleh ESP8266, kemudian melakukan validasi sebelum data tersebut disimpan ke dalam *Database MySQL*. Proses penyimpanan ini berfungsi sebagai *data logging*, sehingga seluruh riwayat pengukuran dapat tersimpan dan dimanfaatkan untuk keperluan analisis selanjutnya.

10. Monitoring Melalui Website

Website monitoring secara berkala mengirimkan permintaan (*request*) data ke REST API. Data yang diterima kemudian ditampilkan dalam bentuk dashboard *monitoring* yang memuat informasi suhu, tegangan, arus, daya, status motor, level kontrol, grafik *realtime*, serta riwayat data (*history*).

11. Notifikasi Sistem

Server melakukan pemeriksaan terhadap data yang masuk guna mendeteksi kemungkinan kondisi abnormal, seperti suhu yang melampaui batas yang ditentukan, arus berlebih, tegangan rendah, atau kegagalan pembacaan sensor.

Apabila kondisi tersebut ditemukan, sistem akan membangkitkan notifikasi dan mengirimkannya ke *website*, sehingga pengguna dapat segera mengetahui adanya gangguan pada sistem.

12. Proses *Looping*

Keseluruhan rangkaian proses mulai dari pembacaan sensor, pengendalian motor, pengiriman data, penyimpanan data, hingga *monitoring* pada *website* berlangsung secara terus-menerus (*looping*) selama sistem masih memperoleh catu daya dan koneksi internet tetap tersedia. Dengan mekanisme tersebut, sistem dapat menjalankan proses *monitoring* dan pengendalian secara *realtime* sekaligus menyimpan seluruh data hasil pengukuran secara otomatis.

3.4.7 Rancangan Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini berfungsi sebagai media utama dalam proses *monitoring* suhu, pengukuran parameter listrik, pengendalian kipas, komunikasi data, serta penyediaan sumber energi berbasis panel surya. Adapun perangkat keras yang digunakan dapat dilihat pada Tabel 3.

Tabel 3. 1 Komponen *hardware*

No	Nama Komponen	Fungsi
1.	Panel Surya	Sumber energi utama sistem yang mengubah energi cahaya matahari menjadi energi listrik
2.	TP4056	Modul charger dan proteksi baterai lithium 18650
3.	Baterai 18650	Media penyimpanan energi listrik dari panel surya
4.	Buck Converter	Menurunkan dan menstabilkan tegangan sesuai kebutuhan sistem
5.	ESP8266 NodeMCU	Mikrokontroler utama dan modul komunikasi <i>WiFi</i>
6.	DHT22	Sensor suhu untuk mengukur temperatur lingkungan
7.	INA219	Sensor arus dan tegangan untuk <i>monitoring</i> parameter listrik
8.	Relay 2 Channel	Saklar elektronik untuk pengendalian beban
9.	MOSFET IRLZ44N	Pengendali kecepatan kipas pada metode <i>Multi level control</i>
10.	Kipas DC	Aktuator pendingin yang dikendalikan oleh sistem
11.	Transistor	Komponen pendukung rangkaian pengendali kipas

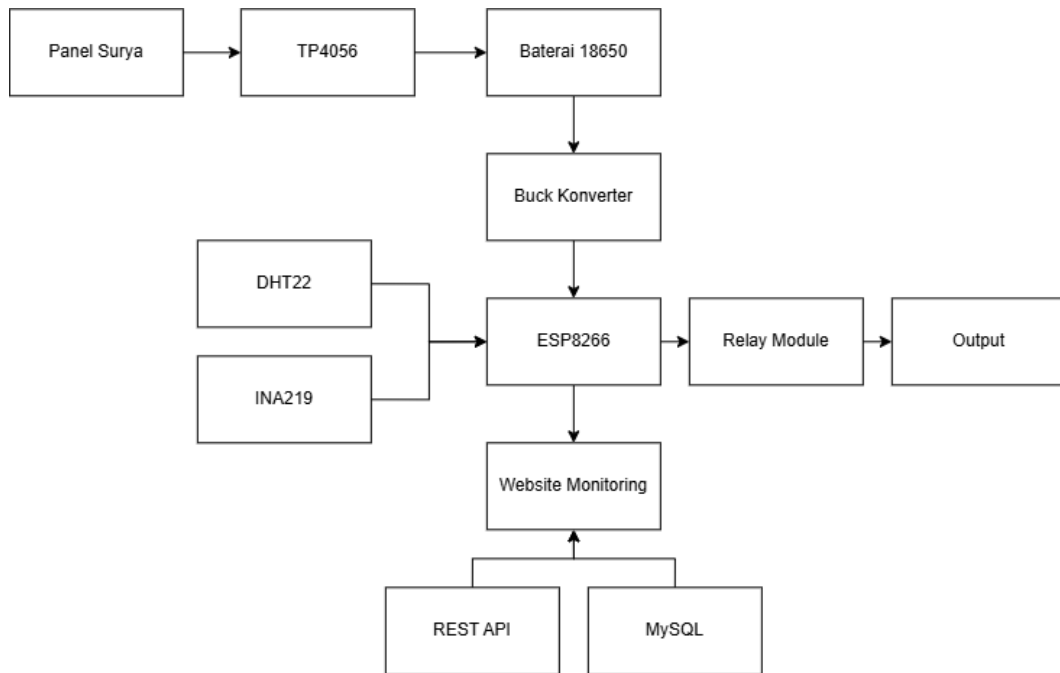
No	Nama Komponen	Fungsi
12.	Dioda	Proteksi terhadap tegangan balik (back EMF) pada motor kipas
13.	Kabel Jumper	Penghubung antar komponen elektronik
14.	PCB/Breadboard	Media perakitan dan pengujian rangkaian elektronik

Panel surya digunakan sebagai sumber energi utama yang menyuplai kebutuhan daya sistem. Energi listrik yang dihasilkan panel surya disimpan pada baterai 18650 melalui modul TP4056 yang berfungsi sebagai pengatur proses pengisian sekaligus proteksi baterai. Tegangan keluaran kemudian disesuaikan menggunakan buck converter agar sesuai dengan kebutuhan operasional perangkat elektronik yang digunakan.

ESP8266 NodeMCU berfungsi sebagai pusat pengendali sistem yang bertugas membaca data dari sensor DHT22 dan INA219, mengolah data, mengendalikan kipas, serta mengirimkan data ke server melalui jaringan internet. Sensor DHT22 digunakan untuk mengukur suhu lingkungan, sedangkan sensor INA219 digunakan untuk mengukur tegangan dan arus pada sistem panel surya.

Pada bagian aktuator digunakan kipas DC yang dikendalikan menggunakan kombinasi *relay* dan MOSFET IRLZ44N. Penggunaan MOSFET memungkinkan penerapan metode *Multi level control* sehingga kecepatan kipas dapat disesuaikan berdasarkan rentang suhu yang telah ditentukan. Selain itu, transistor dan dioda digunakan sebagai komponen pendukung untuk meningkatkan stabilitas dan keamanan rangkaian.

A. Diagram Blok System

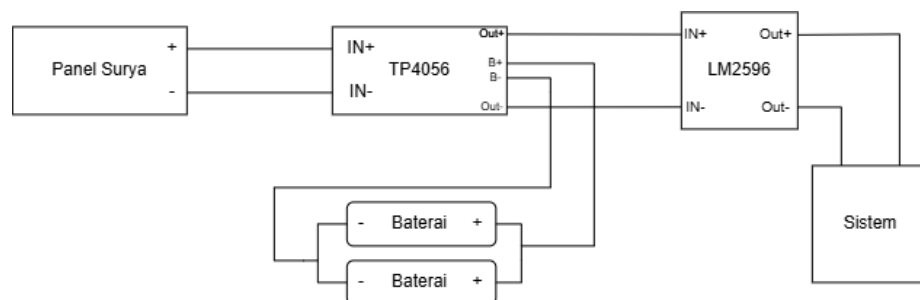


Gambar 3. 9 Diagram Blok Sistem

Pada Gambar 3.9, diagram blok sistem menggambarkan hubungan antara komponen utama dari sistem pemantauan suhu dan kontrol kipas berbasis *Internet of Things* (IoT). Diagram ini menunjukkan aliran distribusi daya dari panel surya ke semua perangkat, proses akuisisi data oleh sensor DHT22 dan INA219, pemrosesan data oleh ESP8266, transmisi data ke situs web *monitoring*, dan proses kontrol kipas berdasarkan perintah pengguna dan pembacaan suhu.

B. Wiring Diagram

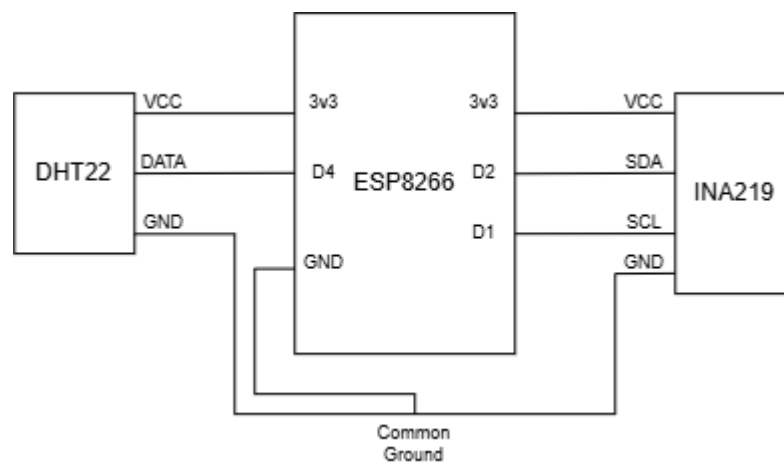
1. Wiring Diagram catu daya



Gambar 3. 10 Wiring Diagram catu daya

Wiring diagram catu daya menunjukkan sirkuit catu daya utama sistem. Panel surya berfungsi sebagai sumber energi, mengubah energi cahaya menjadi energi listrik. Tegangan keluaran dari panel surya diarahkan ke modul TP4056 untuk mengisi baterai 18650, yang diatur secara paralel untuk meningkatkan kapasitas penyimpanan energi tanpa mengubah tegangan keluaran. Energi yang disimpan dalam baterai kemudian dikirim ke modul LM2596 untuk menurunkan dan menstabilkan tegangan menjadi 5 V, yang berfungsi sebagai sumber daya untuk semua komponen sistem.

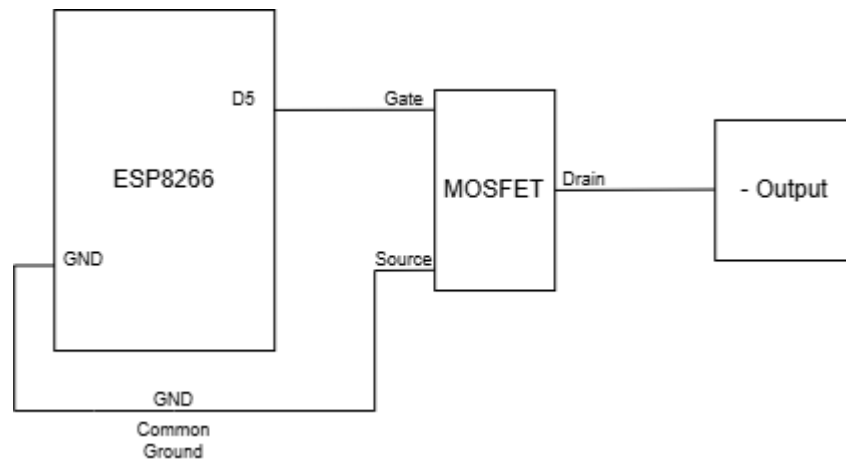
2. *Wiring Diagram ESP8266 dan sensor*



Gambar 3. 11 *Wiring Diagram Sensor*

Wiring diagram sensor ini menunjukkan koneksi antara mikrokontroler ESP8266 dan sensor DHT22 serta INA219. Sensor DHT22 digunakan untuk mengukur suhu lingkungan dan terhubung ke pin digital pada ESP8266 sebagai input suhu. Sensor INA219 terhubung melalui komunikasi I2C melalui pin SDA dan SCL untuk membaca parameter listrik seperti tegangan, arus, dan daya. Semua data sensor diproses oleh ESP8266 sebelum dikirim ke server melalui jaringan WiFi.

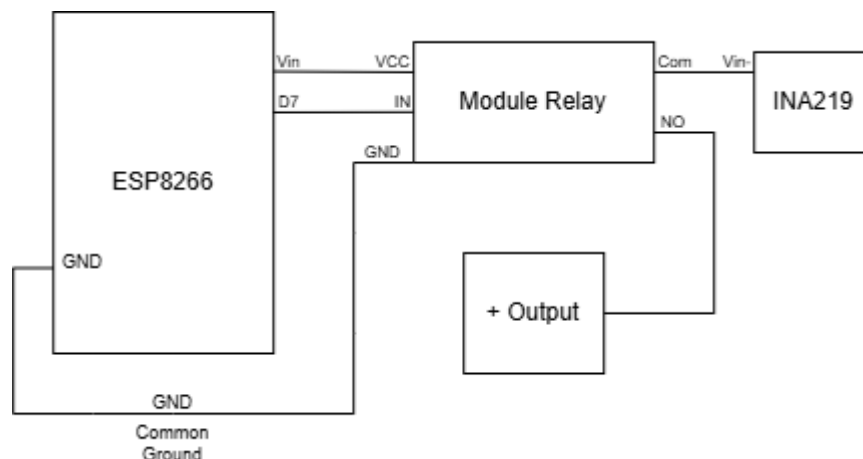
3. *Wiring Diagram driver kipas (MOSFET)*



Gambar 3. 12 *Wiring Diagram driver kipas*

Wiring diagram driver kipas menunjukkan sirkuit kontrol kecepatan kipas menggunakan MOSFET IRLZ44N. ESP8266 menghasilkan sinyal PWM yang diteruskan ke gerbang MOSFET melalui resistor 220 Ω , sementara resistor pull-down 10k Ω digunakan untuk menjaga MOSFET dalam keadaan OFF saat sistem pertama kali dinyalakan. Dioda 1N4007 dihubungkan secara paralel dengan kipas sebagai dioda *flyback* untuk melindungi MOSFET dari tegangan induksi balik yang dihasilkan oleh motor DC ketika kipas berhenti berputar. Kipas DC 5 V adalah perangkat output dalam sistem.

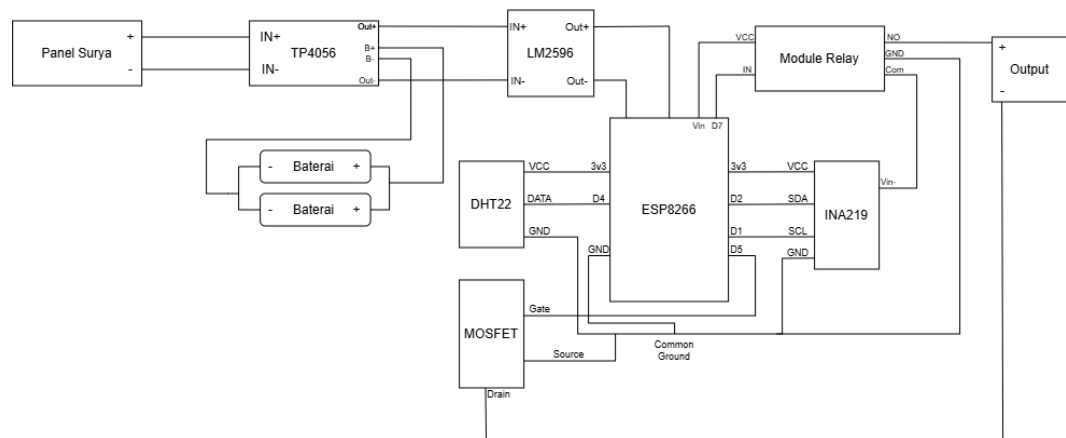
4. *Wiring Diagram fail safe relay*



Gambar 3. 13 *Wiring Diagram fail safe sistem*

Wiring diagram fail safe menggambarkan mekanisme keamanan sistem, yang menggunakan relay yang dikendalikan oleh ESP8266. Data tegangan dan arus yang diperoleh dari sensor INA219 digunakan sebagai dasar pengambilan keputusan. Jika parameter listrik berada dalam batas normal, relay tetap aktif, memungkinkan aliran daya ke penggerak kipas. Sebaliknya, jika tegangan atau arus melebihi batas aman yang telah ditentukan, ESP8266 akan menonaktifkan relay, secara otomatis memutus pasokan daya ke kipas. Mekanisme ini dirancang untuk melindungi perangkat dari kerusakan yang disebabkan oleh kondisi listrik yang tidak normal.

5. *Wiring Diagram* sistem keseluruhan

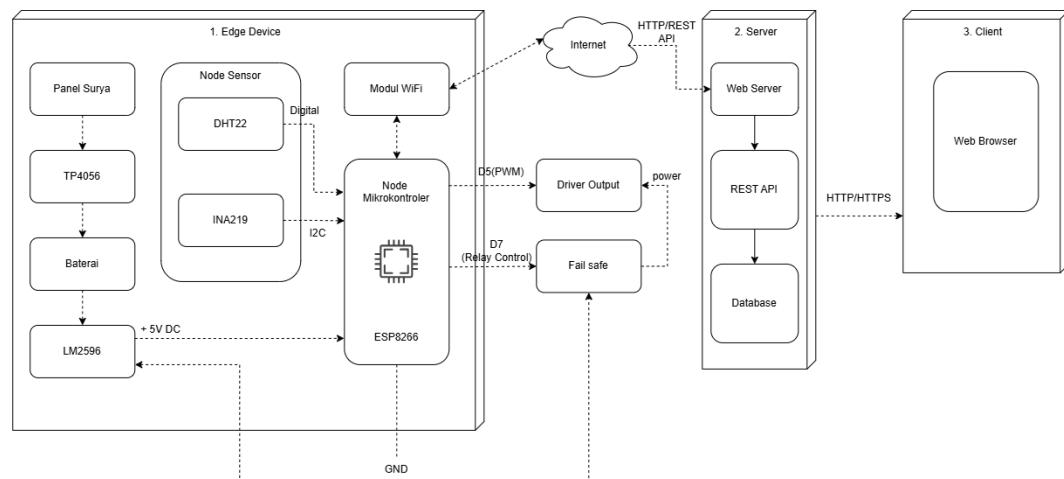


Gambar 3. 14 *Wiring Diagram* Sistem Keseluruhan

Wiring diagram sistem keseluruhan mengintegrasikan semua subsistem, termasuk catu daya, sensor, mikrokontroler, penggerak kipas, sistem *fail safe*, dan komunikasi *Internet of Things* (IoT). Panel surya berfungsi sebagai sumber energi utama, mengisi baterai melalui TP4056. Energi dari baterai distabilkan oleh LM2596 sebelum digunakan untuk memberi daya pada ESP8266, sensor DHT22, sensor INA219, relay, dan driver MOSFET. ESP8266 memproses semua data sensor, mentransmisikan informasi ke situs web secara *real-time*, menerima perintah kontrol dari pengguna, dan mengatur kecepatan satu kipas DC 5 V berdasarkan suhu lingkungan. Selain itu, sistem ini menerapkan mekanisme *fail safe* yang memutus daya ke kipas jika sensor INA219 mendeteksi tingkat tegangan

atau arus yang melebihi batas aman, sehingga memastikan operasi berkelanjutan dan keamanan perangkat.

C. Deployment Diagram



Gambar 3. 15 Deployment diagram Keseluruhan Sistem

Gambar 3.15 Deployment diagram keseluruhan sistem menggambarkan arsitektur fisik dari sistem pemantauan suhu dan kontrol kipas berbasis *Internet of Things* (IoT) yang terdiri dari tiga node utama: perangkat IoT (*edge device*), server, dan klien.

Node perangkat IoT (*edge device*) menyimpan semua komponen perangkat keras yang dipasang dalam sistem pemantauan. Node ini mengandung subsistem catu daya yang terdiri dari panel surya sebagai sumber energi, modul TP4056 untuk mengontrol pengisian baterai, baterai 18650 yang terhubung secara paralel untuk menyimpan energi, dan modul LM2596 yang menurunkan dan menstabilkan tegangan menjadi 5 V. Selain itu, ada sensor DHT22 untuk membaca suhu lingkungan dan sensor INA219 untuk mengukur parameter listrik seperti tegangan, arus, dan daya. Semua data sensor diproses oleh mikrokontroler ESP8266, yang berfungsi sebagai pusat kendali sistem.

ESP8266 terhubung ke driver MOSFET IRLZ44N untuk mengatur kecepatan putaran kipas DC menggunakan Kontrol *Multi-level* berdasarkan suhu yang terdeteksi oleh sensor DHT22. Sementara itu, ESP8266 juga mengontrol modul

relay sebagai mekanisme *fail-safe*, yang akan memutuskan daya ke kipas jika sensor INA219 mendeteksi bahwa tegangan atau arus berada di luar batas aman yang telah ditentukan.

Node Server berfungsi sebagai pusat manajemen data, yang terdiri dari Web Server, REST API, dan basis data MySQL. ESP8266 mengirimkan data pemantauan waktu nyata melalui jaringan *WiFi* menggunakan protokol HTTP/REST API ke server. REST API memproses permintaan dari perangkat IoT dan kemudian menyimpan data tersebut dalam basis data MySQL sebagai repositori untuk pemantauan dan *data logging*.

Node Klien adalah perangkat pengguna yang mengakses sistem melalui peramban web. Situs web pemantauan menampilkan informasi tentang suhu, tegangan, arus, grafik pemantauan, riwayat data (*data logging*), status sistem, dan menyediakan fitur kontrol *ON/OFF* kipas. Perintah yang dikeluarkan oleh pengguna melalui situs web dikirim kembali ke ESP8266 melalui REST API, memungkinkan pemantauan dan kontrol dua arah.

Secara keseluruhan, diagram penyebaran ini menunjukkan bahwa sistem menerapkan konsep *edge computing*, di mana akuisisi data dan kontrol dilakukan pada perangkat IoT, sementara server mengelola penyimpanan data dan menyediakan layanan *monitoring* berbasis web. Arsitektur ini memungkinkan sistem untuk melakukan pemantauan waktu nyata, kontrol kipas adaptif, pencatatan data berkelanjutan, dan penerapan mekanisme *fail-safe* untuk meningkatkan keamanan dan keandalan sistem.

3.4.8 Rancangan Perangkat Lunak (Software)

Perangkat lunak digunakan untuk mendukung proses pemrograman mikrokontroler, pengembangan *website monitoring*, pengelolaan basis data, serta komunikasi data antara perangkat IoT dan server. Perangkat lunak yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.2 *component software*.

Tabel 3. 2 *Component software*

No	Nama Komponen	Fungsi
1.	Arduino IDE	Pemrograman dan pengunggahan program ke ESP8266
2.	Visual Studio Code	Pengembangan source code aplikasi dan <i>website</i>
3.	React JS	<i>Framework</i> frontend untuk membangun <i>website monitoring</i>
4.	Vite	Build tool dan development server React
5.	Tailwind CSS	<i>Framework</i> CSS untuk desain antarmuka <i>website</i>
6.	Recharts	Library visualisasi data dalam bentuk grafik
7.	JavaScript	Bahasa pemrograman untuk logika aplikasi web
8.	REST API	Media komunikasi data antara ESP8266 dan server
9.	MySQL	Sistem manajemen basis data untuk <i>Data logging</i>
10.	Web Browser	Media akses dan pengujian <i>website monitoring</i>

Arduino IDE digunakan untuk mengembangkan program yang berjalan pada ESP8266 NodeMCU. Program tersebut bertugas membaca data sensor, menjalankan algoritma pengendalian kipas, serta mengirimkan data *monitoring* ke server melalui jaringan internet.

Pada sisi server dan antarmuka pengguna, *website monitoring* dikembangkan menggunakan React JS dengan dukungan Vite sebagai *build tool*. Desain antarmuka dibuat menggunakan Tailwind CSS, sedangkan visualisasi data *monitoring* ditampilkan menggunakan Recharts dalam bentuk grafik *realtime*. Data yang diterima dari perangkat IoT disimpan ke dalam basis data MySQL melalui REST API sehingga dapat ditampilkan kembali sebagai *Data logging* dan informasi *monitoring* bagi pengguna.

A. Perancangan Sistem Kontrol

Perancangan sistem kontrol merupakan tahap perancangan mekanisme pengendalian kipas secara otomatis berdasarkan perubahan suhu yang terdeteksi oleh sensor DHT22. Pada penelitian ini, sistem kontrol menggunakan metode *Multi level control*, yaitu metode pengendalian yang membagi rentang suhu ke dalam beberapa level sehingga kecepatan kipas dapat menyesuaikan kondisi suhu secara bertahap. Tujuan penerapan metode ini adalah untuk menjaga suhu tetap berada

pada rentang yang diinginkan serta mengurangi konsumsi daya akibat penggunaan kipas yang beroperasi pada kecepatan maksimum secara terus-menerus.

Proses pengendalian diawali dengan pembacaan suhu oleh sensor DHT22. Data suhu kemudian dikirim ke NodeMCU ESP8266 untuk diproses sesuai dengan aturan kontrol yang telah ditentukan. Selanjutnya, NodeMCU menentukan level kecepatan kipas berdasarkan nilai suhu yang terbaca, kemudian mengirimkan sinyal PWM (*Pulse Width Modulation*) ke *driver* MOSFET IRLZ44N untuk mengatur kecepatan putaran kipas DC. Selain itu, relay digunakan sebagai saklar elektronik untuk mengaktifkan atau menonaktifkan kipas sesuai kebutuhan sistem.

Pada penelitian ini, sistem kontrol dibagi menjadi beberapa *level* kecepatan kipas berdasarkan rentang suhu yang telah ditentukan. Semakin tinggi suhu yang terbaca, maka semakin tinggi pula level kecepatan kipas yang diberikan. Sebaliknya, apabila suhu berada pada kondisi normal, kipas akan berputar pada kecepatan rendah atau dalam kondisi mati sehingga penggunaan energi menjadi lebih efisien.

Sebagai acuan dalam proses pengendalian, pembagian level suhu dan kecepatan kipas ditunjukkan pada Tabel berikut:

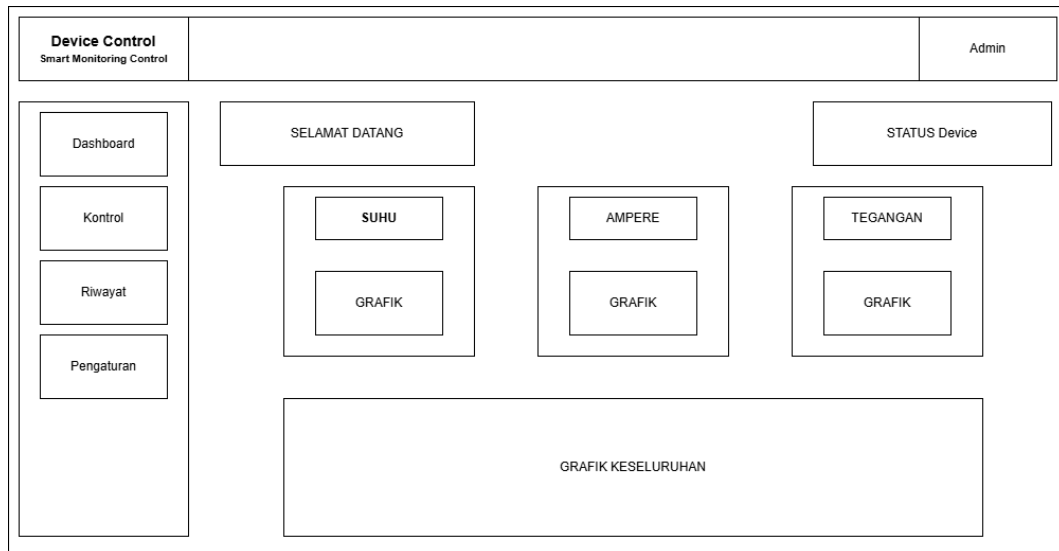
Tabel 3. 3 Perancangan Level Sistem Kontrol

NO	Level	Rentang Suhu	Kecepatan Kipas	Kondisi
1.	Level 0	$< 30^{\circ}\text{C}$	OFF (0%)	Kipas mati
2.	Level 1	$30\text{--}34^{\circ}\text{C}$	Rendah (30%)	Pendinginan ringan
3.	Level 2	$35\text{--}39^{\circ}\text{C}$	Sedang (60%)	Pendinginan normal
4.	Level 3	$\geq 40^{\circ}\text{C}$	Tinggi (100%)	Pendinginan maksimal

Berdasarkan perancangan tersebut, sistem kontrol diharapkan mampu merespons perubahan suhu secara otomatis, menjaga kestabilan suhu, serta meningkatkan efisiensi penggunaan energi pada sistem *monitoring* berbasis panel surya. Seluruh proses pengendalian berlangsung secara *realtime* sehingga kipas dapat menyesuaikan kondisi suhu tanpa memerlukan intervensi pengguna.

B. Perancangan Antarmuka Sistem

1. Halaman *Dashboard*



Gambar 3. 16 *Wireframe dashboard*

Pada halaman ini ditampilkan informasi suhu, arus, dan tegangan hasil pembacaan sensor DHT22 dan INA219 yang dikirimkan oleh ESP8266 melalui REST API.

Pada bagian atas *Dashboard* terdapat indikator status perangkat yang menunjukkan kondisi koneksi sistem apakah dalam keadaan *online* atau *offline*. Selain itu, tersedia tiga card *monitoring* utama yang menampilkan nilai suhu, arus, dan tegangan secara *realtime* lengkap dengan indikator status kondisi sistem.

Halaman *Dashboard* juga dilengkapi dengan grafik *monitoring realtime* menggunakan library Recharts. Grafik digunakan untuk menampilkan perubahan data suhu, arus, dan tegangan secara periodik sehingga pengguna dapat melakukan analisis kondisi sistem secara visual dan *realtime*.

2. Halaman Grafik *Monitoring*



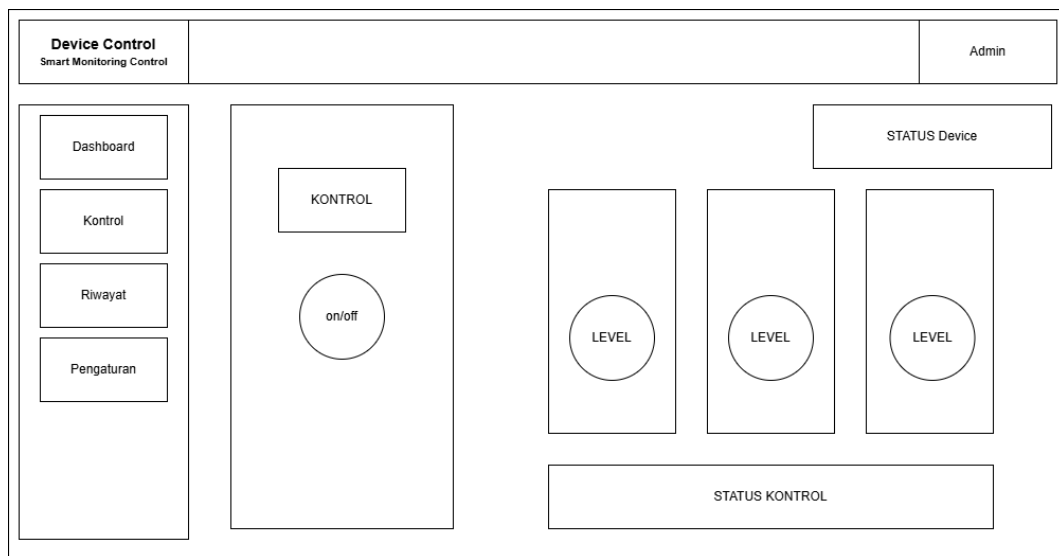
Gambar 3. 17 wireframe grafik monitoring

Grafik *monitoring realtime* dirancang untuk memvisualisasikan perubahan data sensor dalam bentuk grafik garis (*line chart*). Grafik menampilkan data suhu, arus, dan tegangan yang diperoleh dari *Database MySQL* secara *realtime*.

Setiap parameter memiliki warna yang berbeda untuk memudahkan identifikasi data, yaitu warna biru untuk suhu, hijau untuk arus, dan kuning untuk tegangan. Grafik juga dilengkapi fitur tooltip sehingga pengguna dapat melihat detail nilai sensor pada waktu tertentu.

Penggunaan grafik *realtime* bertujuan untuk mempermudah proses *monitoring* sistem serta membantu pengguna dalam menganalisis kestabilan suhu dan parameter listrik selama sistem beroperasi.

3. Halaman Kontrol Sistem



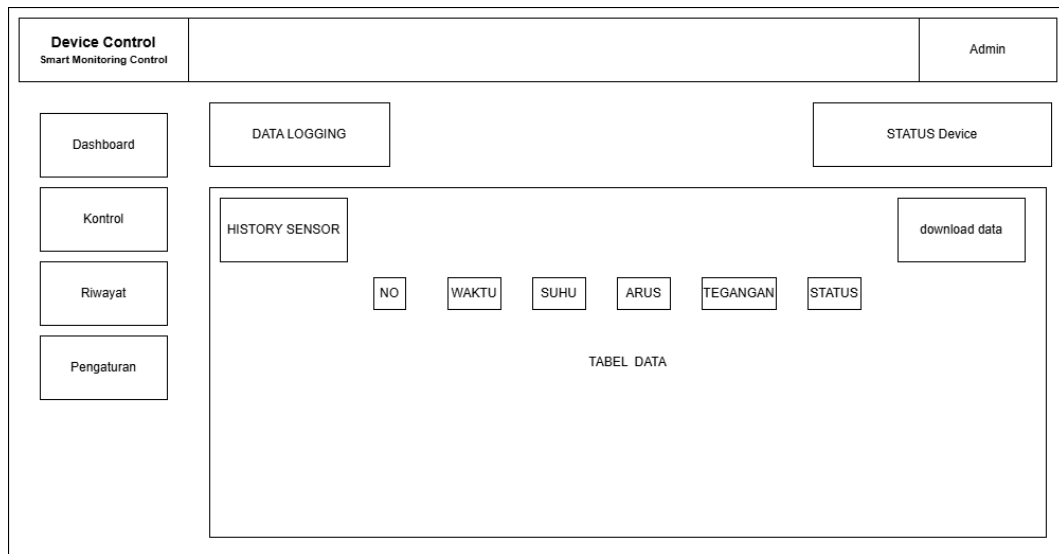
Gambar 3. 18 Wireframe Sistem kontrol

Halaman kontrol sistem digunakan untuk melakukan pengendalian kipas secara manual maupun menggunakan metode *Multi level control*. Pada halaman ini terdapat tombol kontrol *ON/OFF* yang digunakan untuk mengaktifkan dan menonaktifkan kipas secara *realtime*.

Selain kontrol *ON/OFF*, halaman ini juga menyediakan fitur *Multi level control* yang terdiri dari tiga level kecepatan kipas, yaitu level rendah dengan PWM 30%, level sedang dengan PWM 60%, dan level tinggi dengan PWM 100%.

Setiap level kontrol ditampilkan dalam bentuk card interaktif sehingga pengguna dapat memilih tingkat kecepatan kipas sesuai kebutuhan sistem. Halaman kontrol juga menampilkan status level aktif yang sedang digunakan oleh sistem.

4. Halaman *Data logging*



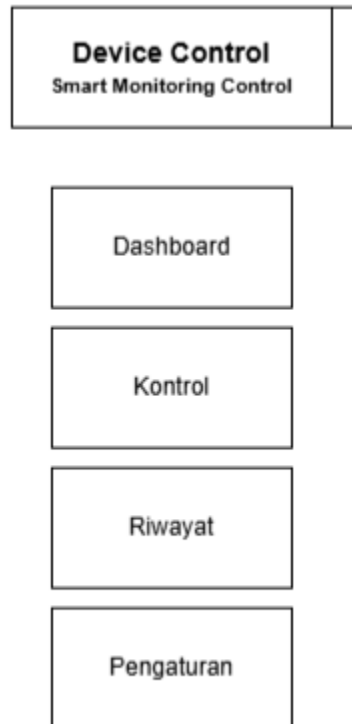
Gambar 3. 19 Wireframe data logging

Halaman *Data logging* digunakan untuk menampilkan histori data *monitoring* yang tersimpan pada *Database MySQL*. Data yang ditampilkan meliputi waktu *monitoring*, suhu, arus, tegangan, serta status kondisi sistem.

Data logging disajikan dalam bentuk tabel untuk mempermudah proses pembacaan dan analisis data hasil *monitoring*. Selain itu, sistem juga menyediakan fitur *export* data yang digunakan untuk menyimpan data *monitoring* dalam bentuk file tertentu sebagai dokumentasi atau bahan analisis penelitian.

Pada bagian bawah halaman ditampilkan informasi tambahan berupa total data *monitoring*, status perangkat, dan level kontrol aktif sistem. Fitur ini membantu pengguna dalam mengetahui kondisi sistem secara keseluruhan.

5. Navigasi Sidebar



Gambar 3. 20 Wireframe sidebar

Sistem navigasi *website* dirancang menggunakan sidebar menu yang berada pada sisi kiri halaman. Sidebar berfungsi untuk mempermudah perpindahan halaman antar fitur sistem.

Menu navigasi terdiri dari *Dashboard*, *Kontrol*, *Riwayat*, dan *Pengaturan*. Desain sidebar menggunakan tampilan modern berbasis dark mode dengan indikator menu aktif agar pengguna lebih mudah mengenali halaman yang sedang digunakan.

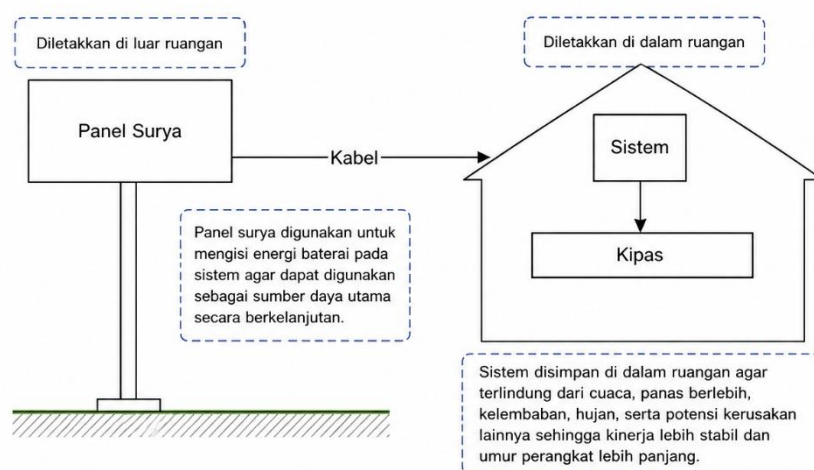
Navigasi sistem dirancang responsif sehingga tetap dapat digunakan dengan baik pada berbagai ukuran layar perangkat.

3.5 Rencana Implementasi Sistem

Rencana implementasi merupakan tahapan penerapan rancangan sistem ke dalam bentuk prototype yang dapat digunakan untuk melakukan monitoring suhu,

pengukuran parameter listrik, pengendalian kipas, dan penyimpanan data secara terintegrasi. Implementasi dilakukan dengan menggabungkan perangkat keras, perangkat lunak, sistem komunikasi, serta sumber energi panel surya sesuai dengan rancangan yang telah ditentukan. Tahapan implementasi disusun secara bertahap agar setiap bagian sistem dapat diuji sebelum diintegrasikan secara keseluruhan.

Penempatan perangkat dalam penelitian ini dibagi menjadi dua lokasi, yaitu panel surya ditempatkan di luar ruangan, sedangkan perangkat sistem ditempatkan di dalam ruangan. Panel surya ditempatkan di luar ruangan agar dapat menerima paparan sinar matahari yang digunakan untuk menghasilkan dan mengisi energi pada baterai sebagai sumber daya sistem. Sementara itu, perangkat sistem yang terdiri dari mikrokontroler, sensor, rangkaian pengendali, dan kipas ditempatkan di dalam ruangan untuk mengurangi paparan langsung terhadap hujan, kelembapan, panas berlebih, debu, dan kondisi lingkungan lainnya yang berpotensi memengaruhi kinerja atau menyebabkan kerusakan perangkat.



Gambar 3. 21 Rencana Implementasi Sistem

Gambar 3.22 Rencana implementasi Sistem, menunjukkan rancangan penempatan panel surya dan sistem pada lingkungan penelitian. Panel surya ditempatkan di luar ruangan dan dihubungkan menggunakan kabel menuju sistem yang berada di dalam ruangan. Energi yang dihasilkan panel surya digunakan untuk mengisi baterai melalui rangkaian pengisian dan pengaturan tegangan, sedangkan

sistem di dalam ruangan digunakan untuk menjalankan proses monitoring dan pengendalian kipas.

Rencana implementasi sistem dilakukan melalui beberapa tahapan sebagai berikut.

1. Implementasi Perangkat Keras

Tahap pertama dilakukan dengan merakit dan mengintegrasikan komponen perangkat keras berdasarkan rancangan sistem. ESP8266 digunakan sebagai mikrokontroler utama yang mengolah data dari sensor DHT22 dan INA219. Sensor DHT22 digunakan untuk memperoleh data suhu, sedangkan INA219 digunakan untuk mengukur tegangan dan arus listrik pada sistem.

Pengendalian kipas dilakukan menggunakan MOSFET IRLZ44N yang dikendalikan oleh ESP8266 melalui sinyal PWM. Kipas DC digunakan sebagai aktuator untuk menghasilkan aliran udara berdasarkan kondisi suhu yang terdeteksi. Seluruh komponen sistem ditempatkan di dalam ruangan, sedangkan panel surya ditempatkan di luar ruangan dan dihubungkan dengan rangkaian sumber energi melalui kabel.

2. Implementasi Sistem Sumber Energi

Tahap berikutnya adalah mengimplementasikan panel surya sebagai sumber energi pada prototype. Panel surya digunakan untuk menghasilkan energi listrik dari sinar matahari yang kemudian digunakan untuk mengisi baterai melalui modul TP4056. Baterai 18650 yang digunakan sebagai penyimpan energi disusun secara paralel sesuai dengan rancangan sistem.

Energi dari baterai selanjutnya melewati buck converter LM2596S untuk menyesuaikan tegangan dengan kebutuhan perangkat. Rangkaian sumber energi tersebut dihubungkan dengan sistem sehingga dapat menyediakan daya untuk ESP8266, sensor, dan komponen pendukung lainnya.

3. Implementasi Program ESP8266

Pada tahap ini dilakukan pemrograman ESP8266 menggunakan Arduino IDE. Program dirancang untuk melakukan inisialisasi sistem, membaca data suhu dari DHT22, membaca tegangan dan arus dari INA219, serta mengendalikan kipas berdasarkan nilai suhu yang diperoleh.

Pengendalian kipas diterapkan menggunakan tiga tingkat kontrol. Level 1 menggunakan PWM sebesar 30%, Level 2 menggunakan PWM sebesar 60%, dan Level 3 menggunakan PWM sebesar 100%. Dengan pengaturan tersebut, kecepatan kipas dapat disesuaikan berdasarkan kondisi suhu yang terdeteksi oleh sensor.

ESP8266 juga diprogram untuk melakukan komunikasi melalui jaringan Wi-Fi dan mengirimkan data hasil pengukuran ke REST API. Data tersebut selanjutnya digunakan dalam proses penyimpanan dan penampilan informasi pada website monitoring.

4. Implementasi REST API dan Database

REST API diimplementasikan sebagai penghubung antara ESP8266 dengan sistem backend dan database. Data hasil pembacaan sensor yang dikirimkan oleh ESP8266 diterima oleh REST API untuk diproses dan disimpan ke dalam database MySQL.

Database MySQL digunakan sebagai media penyimpanan data hasil monitoring. Data yang disimpan meliputi waktu pengukuran, suhu, tegangan, dan arus. Penyimpanan tersebut digunakan untuk menerapkan fungsi data logging sehingga data hasil pengukuran dapat digunakan kembali sebagai riwayat monitoring.

5. Implementasi Website Monitoring

Website monitoring diimplementasikan menggunakan React JS sebagai antarmuka sistem. Website dirancang untuk menampilkan informasi hasil pengukuran suhu, tegangan, dan arus yang diperoleh dari database melalui REST API.

Selain menampilkan kondisi sistem, website juga menyediakan tampilan data logging atau riwayat hasil pengukuran. Data tersebut digunakan untuk memberikan informasi mengenai perubahan parameter sistem dari waktu ke waktu sehingga kondisi sistem dapat dipantau secara lebih mudah.

6. Implementasi Integrasi Sistem

Setelah perangkat keras, program ESP8266, REST API, database, dan website selesai dikembangkan, seluruh bagian sistem diintegrasikan. ESP8266 membaca data dari DHT22 dan INA219 kemudian mengolah data tersebut untuk menentukan kondisi pengendalian kipas.

Data hasil pengukuran dikirim melalui jaringan Wi-Fi menuju REST API dan disimpan ke dalam database MySQL. Data yang telah tersimpan kemudian diakses oleh website monitoring untuk ditampilkan kepada pengguna. Dengan demikian, proses monitoring, pengendalian, komunikasi data, dan data logging dapat berjalan sebagai satu kesatuan sistem.

7. Implementasi Pengujian Sistem

Tahap terakhir dalam rencana implementasi adalah melakukan pengujian terhadap sistem yang telah terintegrasi. Pengujian dilakukan untuk mengetahui apakah setiap fungsi sistem telah berjalan sesuai dengan rancangan.

Pengujian meliputi pembacaan suhu menggunakan DHT22, pengukuran tegangan dan arus menggunakan INA219, pengujian pengendalian kipas pada Level 1, Level 2, dan Level 3, pengujian komunikasi antara ESP8266 dengan REST API, pengujian penyimpanan data pada MySQL, serta pengujian tampilan data pada website monitoring.

Pengujian sensor dilakukan dengan membandingkan hasil pengukuran sensor dengan alat ukur pembanding untuk memperoleh nilai error dan akurasi. Pengujian pengendalian kipas dilakukan untuk mengetahui respons kipas pada setiap tingkat PWM, sedangkan pengujian komunikasi dan data logging dilakukan untuk memastikan data dapat dikirim, disimpan, dan ditampilkan dengan baik.

Berdasarkan tahapan tersebut, rencana implementasi sistem dilakukan mulai dari pemasangan perangkat keras, penerapan sumber energi panel surya, pemrograman ESP8266, implementasi REST API dan database, pengembangan website monitoring, hingga integrasi dan pengujian sistem. Tahapan tersebut menjadi dasar dalam penerapan prototype sistem monitoring suhu dan pengendalian kipas IoT dengan multi level control, data logging, dan integrasi panel surya.

3.6 Teknik Pengumpulan Data

Dalam penelitian ini, data yang diperoleh digunakan sebagai dasar dalam proses perancangan, implementasi, pengujian, serta evaluasi sistem *monitoring* suhu dan pengendalian kipas berbasis *Internet of Things* (IoT) dengan *Multi level control* dan *Data logging* berbasis panel surya. Teknik pengumpulan data yang digunakan dalam penelitian ini meliputi studi literatur, studi kasus, dan eksperimen.

3.6.1 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan dan mempelajari berbagai sumber referensi yang relevan dengan topik penelitian. Referensi diperoleh dari jurnal ilmiah nasional dan internasional, buku akademik, prosiding seminar, artikel penelitian, serta sumber ilmiah lainnya yang berkaitan dengan *Internet of Things* (IoT), sistem *monitoring*, sistem kontrol, *data logging*, *fail safe system*, energi surya, sensor DHT22, sensor INA219, dan mikrokontroler ESP8266.

Kegiatan studi literatur bertujuan untuk memperoleh landasan teori yang kuat sebagai dasar dalam perancangan dan pengembangan sistem. Selain itu, studi literatur digunakan untuk memahami perkembangan teknologi terkini, membandingkan hasil penelitian terdahulu, serta mengidentifikasi kesenjangan penelitian (*research gap*) yang menjadi dasar pengembangan sistem yang diusulkan.

Hasil studi literatur kemudian digunakan sebagai acuan dalam menentukan metode pengendalian, pemilihan komponen perangkat keras, perancangan perangkat lunak, serta penyusunan parameter pengujian yang digunakan dalam penelitian.

3.6.2 Studi Kasus

Studi kasus dilakukan pada sistem *monitoring* suhu dan pengendalian kipas berbasis IoT yang dikembangkan dalam penelitian ini. Fokus studi kasus adalah menganalisis kebutuhan sistem yang mampu melakukan *monitoring* suhu, pengukuran parameter listrik, pengendalian kipas secara otomatis, penyimpanan data *monitoring*, serta pemanfaatan panel surya sebagai sumber energi utama.

Melalui studi kasus ini dilakukan identifikasi terhadap permasalahan yang sering ditemukan pada sistem *monitoring* dan pengendalian suhu konvensional, seperti pengendalian kipas yang masih bersifat sederhana, keterbatasan pencatatan data secara otomatis, serta belum optimalnya pemanfaatan energi terbarukan dalam operasional sistem. Hasil identifikasi tersebut digunakan sebagai dasar dalam menentukan kebutuhan dan spesifikasi sistem yang akan dikembangkan.

Studi kasus juga membantu peneliti dalam memahami kondisi operasional sistem secara menyeluruh sehingga rancangan yang dihasilkan dapat sesuai dengan kebutuhan pengguna dan tujuan penelitian yang telah ditetapkan.

3.6.3 Eksperimen

Dalam penelitian ini. Eksperimen dilakukan dengan mengoperasikan sistem yang telah dirancang dan diimplementasikan untuk mengetahui kinerja sistem dalam berbagai kondisi pengujian. Pengujian dilakukan secara terstruktur dengan memberikan variasi kondisi suhu untuk mengamati respons sistem terhadap perubahan suhu yang terjadi.

Data yang dikumpulkan selama eksperimen meliputi data suhu yang diperoleh dari sensor DHT22, data tegangan dan arus yang diperoleh dari sensor INA219, status pengendalian kipas, waktu respons sistem, serta data operasional panel surya. Selain itu, dilakukan pengamatan terhadap keberhasilan komunikasi data antara ESP8266 dan server, kinerja *website monitoring*, serta proses penyimpanan data melalui mekanisme *Data logging*.

Eksperimen juga dilakukan untuk menguji efektivitas metode *ON/OFF* dan *Multi level control* dalam mengendalikan kecepatan kipas berdasarkan perubahan

suhu yang terdeteksi. Pengujian dilakukan pada beberapa rentang suhu yang telah ditentukan untuk mengetahui tingkat respons dan stabilitas sistem dalam menjaga kondisi suhu sesuai dengan parameter yang ditetapkan.

Seluruh data hasil eksperimen direkam secara otomatis melalui sistem *Data logging* yang terintegrasi dengan basis data. Data tersebut selanjutnya digunakan sebagai bahan analisis untuk mengevaluasi akurasi sensor, performa sistem kontrol, efektivitas *monitoring realtime*, serta kemampuan panel surya dalam mendukung kebutuhan energi sistem secara keseluruhan. Dengan demikian, hasil eksperimen dapat digunakan untuk menilai tingkat keberhasilan sistem yang dikembangkan sesuai dengan tujuan penelitian.

3.7 Pengolahan dan Analisis Data

Data yang diperoleh dalam penelitian ini terdiri atas data suhu yang dibaca oleh sensor DHT22, data tegangan dan arus yang diperoleh dari sensor INA219, data status pengendalian kipas, waktu respons sistem, serta data operasional panel surya. Seluruh data tersebut direkam secara otomatis melalui mekanisme *Data logging* dan disimpan ke dalam basis data untuk selanjutnya dilakukan proses pengolahan dan analisis.

3.7.1 Pengolahan Data

Tahap pengolahan data dilakukan dengan mengelompokkan dan menyusun data hasil pengujian berdasarkan jenis parameter yang diukur. Data yang telah terkumpul kemudian diperiksa untuk memastikan kelengkapan dan konsistensinya sebelum dilakukan analisis lebih lanjut. Selanjutnya, data disajikan dalam bentuk tabel, grafik, dan diagram agar lebih mudah dipahami dan dianalisis.

Data suhu, tegangan, dan arus yang diperoleh dari sensor akan dibandingkan dengan hasil pengukuran menggunakan alat ukur referensi untuk mengetahui tingkat akurasi sensor yang digunakan. Selain itu, data hasil pengujian sistem kontrol akan dikelompokkan berdasarkan rentang suhu dan level kecepatan kipas yang dihasilkan oleh metode *Multi level control*. Proses pengolahan data juga

mencakup perhitungan waktu respons sistem serta rekapitulasi data yang tersimpan pada fitur *Data logging*.

3.7.2 Analisis Akurasi Sensor

Analisis akurasi sensor dilakukan untuk mengetahui tingkat ketelitian sensor DHT22 dan INA219 dalam mengukur parameter yang digunakan pada penelitian ini. Pengujian dilakukan dengan cara membandingkan hasil pembacaan sensor dengan alat ukur standar yang memiliki tingkat akurasi lebih baik. Sensor DHT22 dibandingkan dengan termometer digital sebagai acuan dalam pengukuran suhu, sementara sensor INA219 dibandingkan dengan multimeter digital untuk pengukuran tegangan dan arus.

Analisis dilakukan dengan menghitung selisih antara hasil pembacaan sensor dan nilai referensi, kemudian menentukan besarnya persentase kesalahan (*error*) serta tingkat akurasi sensor. Persentase kesalahan dihitung dengan menggunakan rumus berikut:

$$Error(\%) = \frac{|X_{(ref)} - X_{(sensor)}|}{X_{(ref)}} \times 100\% \quad (1)$$

Sementara itu, nilai akurasi sensor dihitung menggunakan persamaan:

$$akurasi (\%) = 100\% - Error (\%) \quad (2)$$

Hasil perhitungan tersebut selanjutnya dianalisis guna mengetahui tingkat ketelitian sensor dalam membaca suhu, tegangan, dan arus. Semakin kecil nilai error yang dihasilkan, semakin tinggi pula tingkat akurasi sensor, sehingga sensor tersebut dapat dinyatakan layak digunakan sebagai perangkat *monitoring* pada sistem yang dikembangkan. Hasil analisis akurasi sensor ini selanjutnya akan disajikan dalam bentuk tabel disertai pembahasan pada Bab IV.

3.7.3 Analisis Kinerja Sistem Monitoring

Analisis kinerja sistem *monitoring* dilakukan untuk mengetahui kemampuan sistem dalam menampilkan data suhu, tegangan, dan arus secara *realtime* pada *website monitoring*. Analisis dilakukan dengan mengamati keberhasilan proses

pengiriman data dari ESP8266 ke server, kecepatan pembaruan data pada *website*, serta keberhasilan penyimpanan data ke dalam basis data.

Selain itu, dilakukan analisis terhadap fitur *Data logging* untuk memastikan seluruh data hasil *monitoring* dapat tersimpan dengan baik dan dapat ditampilkan kembali sesuai dengan waktu pencatatan. Hasil analisis ini digunakan untuk mengevaluasi efektivitas sistem *monitoring* yang telah dikembangkan.

3.7.4 Analisis *Multi level control*

Analisis *Multi level control* dilaksanakan guna mengevaluasi kinerja sistem dalam melakukan pengendalian kecepatan kipas secara otomatis berdasarkan perubahan suhu yang terbaca oleh sensor DHT22. Pada penelitian ini, metode *Multi level control* diterapkan dengan cara membagi rentang suhu ke dalam beberapa tingkatan (*level*) pengendalian, sehingga kecepatan kipas dapat menyesuaikan diri terhadap kondisi suhu secara bertahap. Analisis ini bertujuan untuk mengetahui kemampuan sistem dalam memberikan respons yang sesuai terhadap setiap perubahan suhu, sekaligus menjaga agar suhu tetap berada pada kondisi yang diharapkan.

Proses analisis dilakukan dengan membandingkan nilai suhu hasil pembacaan sensor terhadap *level* kecepatan kipas yang telah ditetapkan sebelumnya. Adapun parameter yang menjadi fokus analisis meliputi kesesuaian antara nilai suhu dengan *level* kontrol yang aktif, waktu respons sistem terhadap perubahan suhu yang terjadi, serta kestabilan kinerja kipas pada masing-masing *level* pengendalian. Hasil dari analisis tersebut selanjutnya disajikan dalam bentuk tabel disertai pembahasan, guna mengetahui sejauh mana keberhasilan metode *Multi level control* dalam mengendalikan kipas secara otomatis.

3.7.5 Analisis Kinerja Sistem Kontrol

Analisis kinerja sistem kontrol dilakukan untuk mengetahui kemampuan metode *ON/OFF* dan *Multi level control* dalam mengendalikan kipas berdasarkan perubahan suhu yang terdeteksi oleh sensor DHT22. Analisis dilakukan dengan

membandingkan kondisi suhu terhadap level kecepatan kipas yang dihasilkan oleh sistem.

Parameter yang diamati meliputi ketepatan aktivasi kipas pada setiap rentang suhu, kestabilan perubahan level kecepatan kipas, serta waktu respons sistem dalam memberikan tindakan pengendalian. Hasil analisis digunakan untuk menilai efektivitas metode kontrol yang diterapkan dalam menjaga kondisi suhu sesuai dengan parameter yang telah ditentukan.

3.7.6 Analisis Kinerja Panel Surya

Analisis kinerja panel surya dilakukan untuk mengetahui kemampuan sumber energi dalam mendukung operasional sistem. Data yang dianalisis meliputi tegangan dan arus yang dihasilkan panel surya, kondisi pengisian baterai, serta kemampuan sistem dalam beroperasi menggunakan energi yang dihasilkan.

Hasil pengukuran dari sensor INA219 digunakan untuk mengevaluasi kestabilan suplai daya yang diberikan oleh panel surya dan baterai. Analisis ini bertujuan untuk mengetahui apakah sumber energi yang digunakan mampu memenuhi kebutuhan daya sistem selama proses pengujian berlangsung.

3.7.7 Analisis Deskriptif Kuantitatif

Metode analisis yang digunakan dalam penelitian ini adalah analisis deskriptif kuantitatif. Data hasil pengujian disajikan dalam bentuk tabel, grafik, dan persentase untuk menggambarkan kinerja sistem secara objektif. Analisis dilakukan tanpa melakukan pengujian hipotesis statistik karena penelitian ini berfokus pada perancangan, implementasi, dan evaluasi kinerja sistem.

Hasil analisis deskriptif digunakan untuk menjelaskan hubungan antara perubahan suhu dengan respons pengendalian kipas, performa *monitoring realtime*, efektivitas *Data logging*, serta kemampuan panel surya dalam menyuplai energi bagi sistem. Berdasarkan hasil tersebut, dapat ditentukan tingkat keberhasilan sistem yang telah dikembangkan sesuai dengan tujuan penelitian yang telah ditetapkan.