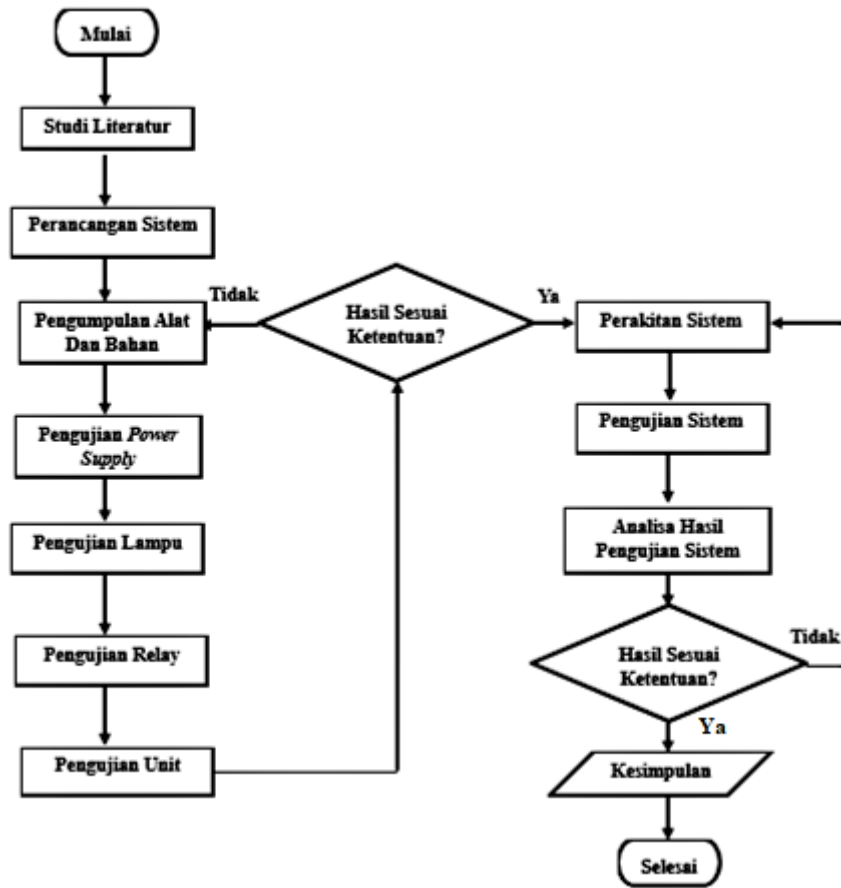


BAB III

METODE PENELITIAN

2.1.Flowchart Penelitian



Gambar 3. 1. Flowchart Alur Penelitian

Pada penelitian ini terbagi menjadi beberapa tahapan diantaranya tahap studi literatur, identifikasi masalah, perancangan sistem, uji coba sistem, dan pengambilan kesimpulan

3.2.Rincian Tahapan Penelitian

3.2.1. Kajian Pustaka

Pada penelitian ini, Kajian pustaka digunakan untuk mendapatkan referensi yang relevan mengenai perancangan prototipe sistem kontrol penerangan otomatis yang menggunakan mikrokontroler. Sumber referensi ini dapat dicari dan ditemukan dalam bentuk buku, jurnal, internet, dan sumber-sumber lainnya. Beberapa materi yang akan dijadikan acuan adalah tentang pengertian dan konsep sistem kontrol, serta mikrokontroler *ESP32* yang akan digunakan untuk melakukan pengolahan data.

Studi Literasi merupakan langkah yang dijalankan untuk mengumpulkan dan memperoleh pemahaman tentang materi yang relevan dengan penelitian yang akan dilakukan, sebagaimana yang tercantum dalam Bab II penelitian. Proses literasi ini melibatkan dua metode yang digunakan, yaitu:

1. Pendampingan

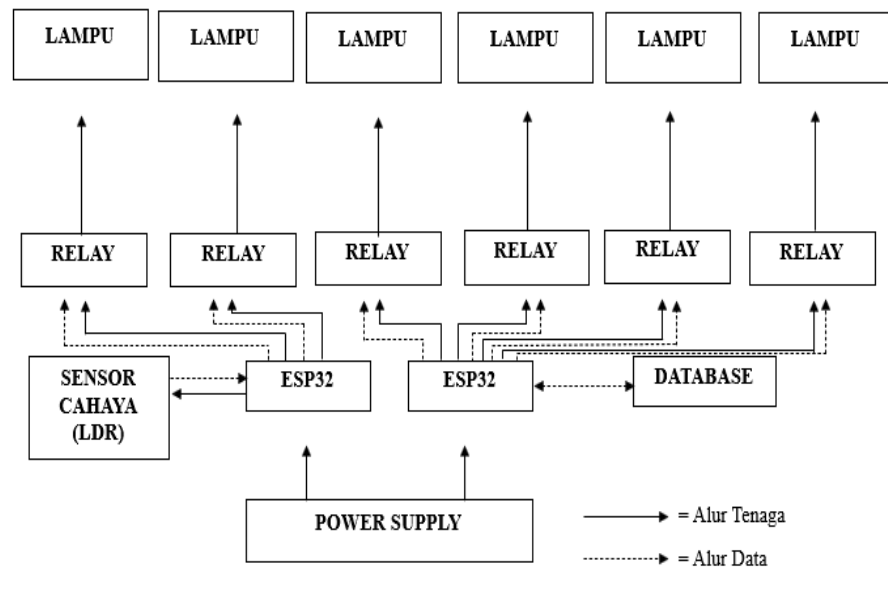
Pendampingan yang dilakukan dalam penelitian ini bertujuan untuk mendapatkan bimbingan dari dosen pembimbing mengenai pelaksanaan penulisan laporan tugas akhir dan pembuatan perangkat.

2. Literatur

Literatur penunjang merupakan kegiatan pengumpulan dan membaca beberapa jurnal, artikel, buku, skripsi, dll yang memiliki keterkaitan dengan judul penelitian.

3.2.2. Perancangan Sistem

Perancangan sistem adalah proses perencanaan dan pembuatan desain sistem yang melibatkan berbagai alat dan komponen pendukung yang bertujuan untuk menciptakan suatu sistem dengan cara kerja sesuai dengan konsep yang diinginkan.



Gambar 3. 2. Arsitektur Sistem

Perancangan sistem kontrol penerangan otomatis menggunakan teknologi *ESP32* berbasis *Internet of Things* dilakukan dengan beberapa tahapan berikut:

1. Identifikasi kebutuhan sistem: Langkah pertama adalah mengidentifikasi kebutuhan sistem. Hal ini meliputi tujuan sistem kontrol penerangan, jenis pengguna yang akan menggunakan penerangan, jenis lampu yang akan digunakan, dan alat apa saja yang diperlukan untuk mengimplementasikan sistem kontrol penerangan.

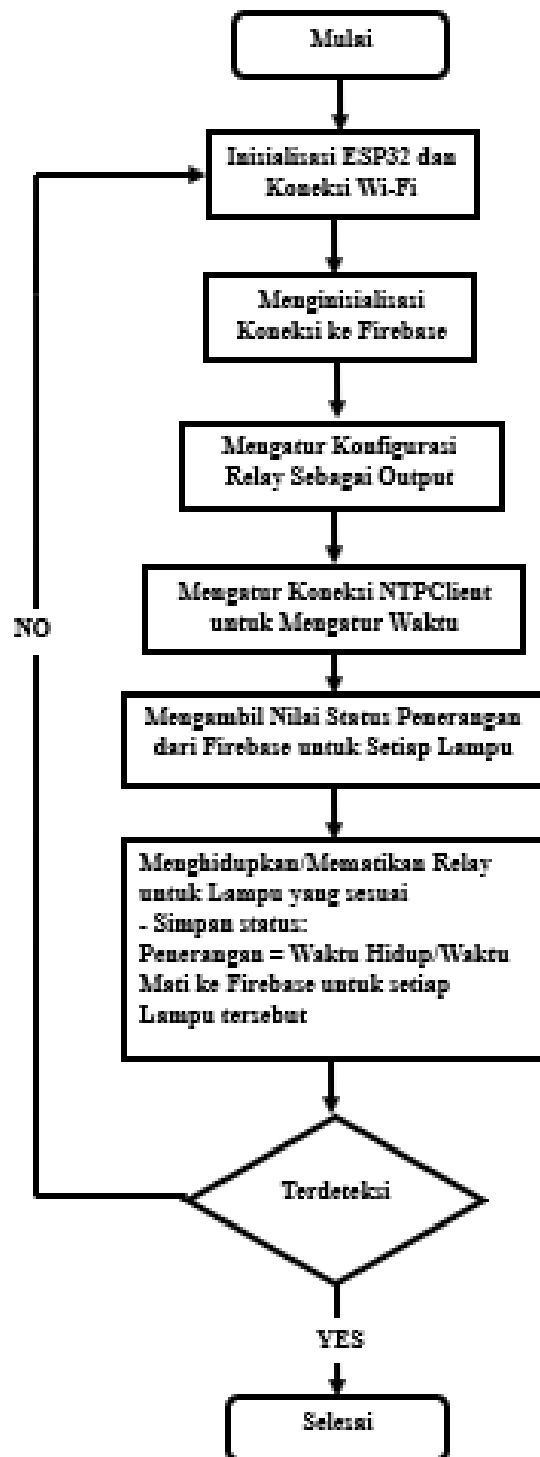
2. Perancangan skema sistem: Melibatkan beberapa komponen penting dalam pengembangan sistem kontrol penerangan otomatis. Perancangan dimulai dengan merancang skema yang mencakup semua komponen yang dibutuhkan dan bagaimana komponen-komponen tersebut akan saling terhubung. Mikrokontroler ESP32 akan berperan sebagai otak sistem, mengendalikan dan memproses data dari sensor cahaya dan komponen lainnya. Sensor cahaya, seperti LDR, akan digunakan untuk mengukur intensitas cahaya di sekitar penerangan, yang akan menjadi masukan penting untuk pengendalian penerangan otomatis. Selain itu, NTPClient akan digunakan untuk mendapatkan waktu yang akurat dari server NTP, memastikan waktu yang tepat dalam pengaturan penerangan berdasarkan waktu. Modul *relay* akan bertindak sebagai saklar elektronik yang dikendalikan oleh mikrokontroler, mengatur nyalanya lampu penerangan sesuai dengan kebutuhan sistem. Sumber daya listrik yang stabil dan koneksi internet yang baik juga menjadi faktor penting dalam perancangan skema sistem. Perangkat lunak atau firmware akan dikembangkan untuk mengatur logika dan algoritma pengendalian penerangan otomatis. Perancangan skema sistem ini harus memperhatikan koneksi dan interaksi antara komponen-komponen tersebut, serta memastikan ketersediaan daya listrik yang memadai, konektivitas internet yang handal, dan komunikasi yang efektif antara sistem kontrol dan perangkat penerangan.
3. Pembuatan perangkat keras: Sistem kontrol penerangan otomatis, perlu dirancang mikrokontroler ESP32 sebagai pusat pengendalian. Sensor

cahaya seperti LDR dipasang untuk mengukur intensitas cahaya, sedangkan NTPClient digunakan untuk mendapatkan waktu yang akurat. Modul relai mengendalikan daya listrik yang mengalir ke lampu penerangan. Ketersediaan sumber daya listrik yang stabil dan konektivitas internet yang handal juga penting. Semua komponen ini harus terhubung dengan baik untuk menciptakan sistem kontrol penerangan otomatis yang efektif.

4. Konfigurasi *Firebase*: Selanjutnya, *Firebase* harus dikonfigurasi untuk sistem akses gerbang. *Firebase* akan digunakan untuk menyimpan daftar lampu yang digunakan.

3.2.2.1. Flowchart Sistem Kontrol

1. Flowchart Sistem Kontrol ESP1



Gambar 3. 3. Flowchart Sistem Kontrol Pada ESP1

Berikut adalah penjelasan *Flowchart* Sistem pada Gambar 3.3:

1. Mulai:
Menandakan awal dari program
2. Inisialisasi ESP32 dan koneksi Wi-Fi:
Melakukan inisialisasi ESP32 dan menghubungkannya ke jaringan Wi-Fi.
3. Menginisialisasi koneksi ke Firebase:
Melakukan inisialisasi koneksi ke Firebase menggunakan informasi host dan token autentikasi yang telah ditentukan.
4. Mengatur konfigurasi relay sebagai output:
Mengatur pin relay sebagai output pada ESP32.
5. Mengatur koneksi NTPClient untuk mengatur waktu:
Mengatur koneksi NTPClient untuk mendapatkan waktu dari server NTP dan mengatur waktu pada *ESP32*.
6. Mengambil nilai waktu penyalaan relay dari Firebase:
Mengambil nilai waktu penyalaan relay dari Firebase untuk menentukan waktu kapan relay harus dinyalakan atau dimatikan.
7. Mengambil nilai status penerangan dari Firebase untuk setiap lampu:
Mengambil status penerangan dari Firebase untuk setiap lampu yang terhubung dengan relay.
8. Jika waktu saat ini = waktu penyalaan relay, lanjutkan ke langkah selanjutnya: Memeriksa apakah waktu saat ini sama dengan waktu penyalaan relay yang ditentukan.
9. Tunggu beberapa detik:

Memberikan jeda waktu hingga lampu menyala sebelum melanjutkan ke Langkah selanjutnya.

10. Jika waktu saat ini = waktu pemadaman relay, lanjutkan ke langkah selanjutnya: Memeriksa apakah waktu saat ini sama dengan waktu pemadaman relay yang ditentukan. relay untuk lampu yang sesuai:

11. Tunggu beberapa detik:

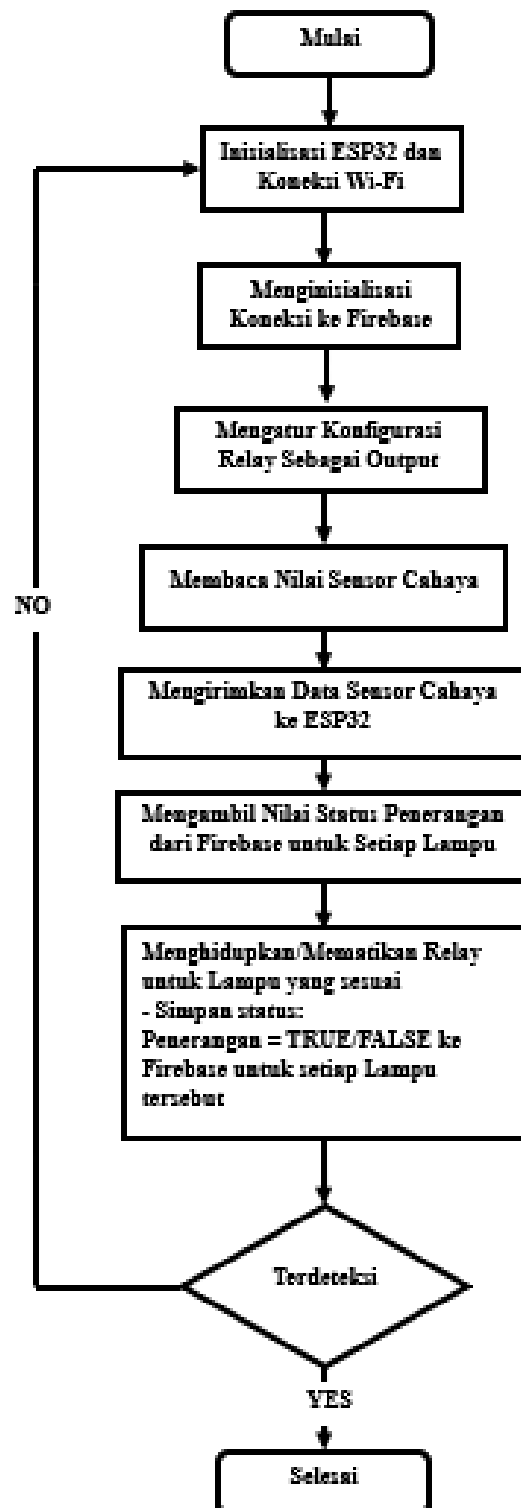
Memberikan jeda waktu hingga lampu padam sebelum melanjutkan ke Langkah selanjutnya.

12. Tunggu beberapa detik:

Memberikan jeda waktu sebelum melanjutkan kembali ke langkah 5.

13. Selesai:

Menandakan akhir program.

2. *Flowchart* Sistem Kontrol ESP2Gambar 3. 4. *Flowchart* Sistem Kontrol Pada ESP2

Berikut adalah penjelasan *Flowchart* Sistem pada Gambar 3.4:

1. Mulai:
Menandakan awal dari program
2. Inisialisasi ESP32 dan koneksi Wi-Fi:
Melakukan inisialisasi ESP32 dan menghubungkannya ke jaringan Wi-Fi.
3. Menginisialisasi koneksi ke Firebase:
Melakukan inisialisasi koneksi ke Firebase menggunakan informasi host dan token autentikasi yang telah ditentukan.
4. Mengatur konfigurasi relay sebagai output:
Mengatur pin relay sebagai output pada ESP32.
5. Membaca nilai sensor cahaya:
Membaca nilai dari sensor cahaya untuk mendapatkan informasi tentang intensitas cahaya di sekitar.
6. Mengirimkan data sensor cahaya ke *ESP32*:
Mengirimkan nilai sensor cahaya ke *ESP32* untuk penyimpanan dan pemantauan.
7. Mengambil nilai status penerangan dari Firebase untuk setiap lampu:
Mengambil status penerangan dari Firebase untuk setiap lampu yang terhubung dengan relay.
8. Jika nilai sensor cahaya rendah, lanjutkan ke langkah selanjutnya:
Memeriksa nilai sensor cahaya untuk menentukan apakah lampu harus dinyalakan berdasarkan ambang batas rendah yang ditentukan.
9. Jika status penerangan = TRUE, lanjutkan ke langkah selanjutnya:

Memeriksa status penerangan untuk menentukan apakah lampu harus dinyalakan atau dimatikan.

10. Menghidupkan relay untuk lampu yang sesuai:

Menghidupkan relay yang terhubung dengan lampu yang sesuai berdasarkan keputusan sebelumnya.

11. Tunggu beberapa detik:

Memberikan jeda waktu sebelum melanjutkan kembali ke langkah 5.

12. Jika status penerangan = FALSE, lanjutkan ke langkah selanjutnya:

Mematikan relay yang terhubung dengan lampu yang sesuai berdasarkan keputusan sebelumnya.

13. Memadamkan relay untuk lampu yang sesuai:

Memadamkan relay yang terhubung dengan lampu yang sesuai berdasarkan keputusan sebelumnya.

14. Tunggu beberapa detik:

Memberikan jeda waktu sebelum melanjutkan kembali ke langkah 5.

15. Selesai:

Menandakan akhir program.

3.2.3. Alat Dan Bahan

Komponen dan peralatan yang digunakan untuk penelitian sistem kontrol penerangan menggunakan sensor cahaya LDR (*Light Dependent Resistor*) dan ESP32 berbasis *Internet of Things* diantaranya sebagai berikut :

Tabel 3. 1. Alat dan Bahan

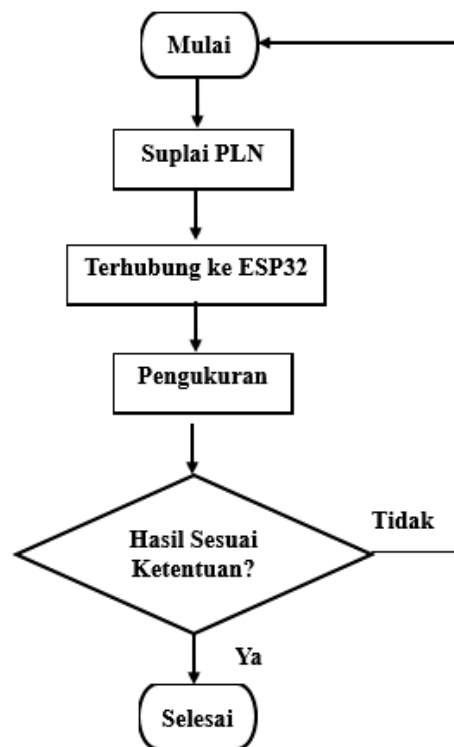
NO	Nama	Spesifikasi	Jumlah
1	Mikrokontroler <i>ESP32</i>	-	2 Unit
2	Kabel <i>Jumper</i>	-	-
3	<i>Breadboard</i>	-	-
4	<i>Power Supply</i>	5VDC	1 buah
5	<i>Relay</i>	5V	2 Buah
6	Kabel <i>USB</i>	-	1 Buah
7	PC	-	1 Unit
8	<i>Software Arduino IDE</i>	-	-
9	<i>Akun Firebase</i>	-	-
10	Lampu	5 watt	6 buah
11	Fitting Lampu	-	6 buah
12	Jaringan Internet	-	-
13	<i>LDR Module</i>		1 buah

3.2.4. Pengujian *Power Supply*

Power Supply yang digunakan dalam penelitian ini adalah sebuah perangkat yang berfungsi sebagai pengubah tegangan AC dari sumber daya listrik PLN dengan tegangan 220V menjadi tegangan DC dengan output sebesar 5V. Nilai tegangan ini disesuaikan dengan kebutuhan suplai listrik untuk ESP32.



Gambar 3. 5. Adapter 5 V

Gambar 3. 6. *Flowchart* Pengujian *Power Supply*

Gambar 3.6 menggambarkan flowchart dari pengujian power supply. Pengujian ini bertujuan untuk memverifikasi apakah power supply mampu menyediakan

suplai listrik dari sumber daya PLN dan untuk mengukur tegangan dan arus yang mengalir ketika suplai dari PLN disalurkan ke ESP32. Tegangan suplai PLN sebesar 220 V dikonversi oleh adaptor sehingga menghasilkan tegangan keluaran sesuai dengan kebutuhan. Tegangan keluaran ini akan diukur menggunakan multimeter. Jika tegangan tidak terbaca, maka dilakukan pemeriksaan, perbaikan, atau penggantian adaptor yang digunakan, atau penyesuaian pengukuran hingga hasil pengukuran yang akurat diperoleh, dan pengujian dianggap selesai.



Gambar 3. 7. Pengujian *power supply*

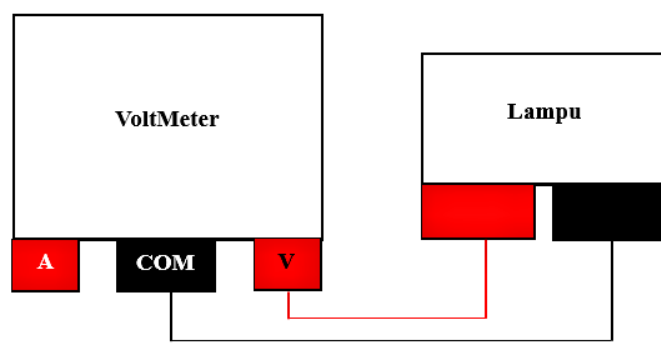
Gambar 3.7 menampilkan ilustrasi kegiatan pengujian suplai tegangan terhadap ESP32 menggunakan voltmeter. Hasil dari pengujian tersebut kemudian dicatat dan ditampilkan dalam Tabel 3.2.

Tabel 3. 2.Pengujian *power supply*

No	Vin	Vout	Waktu (t)
1	220 VAC	4,68 VDC	1
2	220 VAC	4.68 VDC	5
3	220 VAC	4,68 VDC	10
4	220 VAC	4,68 VDC	15
5	220 VAC	4.68 VDC	20

Tabel 3.2 adalah catatan hasil pengujian tegangan power supply terhadap ESP32 yang dilakukan sebanyak lima kali. Hasil pengujian menunjukkan bahwa tegangan yang diperoleh stabil sebesar 4,99 V, sesuai dengan spesifikasi yang dibutuhkan.

3.2.5. Pengujian Lampu

Gambar 3. 8. *Wiring* Pengujian Lampu

Gambar 3.8. menunjukkan *wiring* pengujian lampu dimana pengujian dilakukan dengan mengukur tegangan baterai menggunakan voltmeter. Kutub positif lampu dihubungkan dengan *port* tegangan pada voltmeter dan kutub

negatif lampu dihubungkan dengan *port* COM voltmeter.



Gambar 3. 9. Rangkaian Pengujian Lampu yang Terhubung ke Sistem Kontrol

Gambar 3.9 menunjukkan rangkaian lampu yang terhubung ke sistem kontrol untuk pengujian keandalan dan kapasitas daya listrik. Rangkaian ini dirancang untuk menguji performa sistem dalam menghidupkan dan mengendalikan lampu.

Pada rangkaian tersebut, terdapat beberapa komponen utama yang terhubung dengan baik. Pertama, terdapat ESP32 sebagai mikrokontroler yang bertindak sebagai otak sistem. ESP32 terhubung dengan relay yang berfungsi sebagai saklar untuk mengontrol daya yang diteruskan ke lampu.

Selain itu, terdapat beberapa lampu yang dihubungkan ke relay melalui fitting lampu. Lampu-lampu ini akan dinyalakan dan dimatikan secara oleh sistem kontrol berdasarkan penginputan nilai pada *Realtime Database*

Rangkaian juga dilengkapi dengan sumber daya listrik, seperti power supply atau baterai, yang memberikan daya yang cukup untuk menghidupkan semua lampu yang terhubung.

Pada pengujian, sistem akan dihidupkan dan dipastikan bahwa lampu dapat menyala dengan baik dalam kondisi normal. Dengan melakukan pengujian ini dapat memastikan bahwa rangkaian lampu yang terhubung ke sistem kontrol memiliki keandalan yang baik dan mampu mengendalikan daya listrik sesuai dengan kapasitas yang dibutuhkan.



Gambar 3. 10. Pengukuran nilai tegangan pada saat lampu dinyalakan

Gambar 3.10 menunjukkan proses pengujian nilai tegangan pada saat lampu dinyalakan dalam rangkaian sistem kontrol lampu. Pengujian ini bertujuan untuk memastikan bahwa tegangan yang diberikan kepada lampu berada dalam kisaran yang tepat dan aman.

Pada gambar tersebut, terdapat voltmeter yang terhubung antara sumber daya listrik dan lampu yang akan dinyalakan. Voltmeter digunakan untuk mengukur tegangan yang diterima oleh lampu saat sistem kontrol diaktifkan.

Proses pengujian dimulai dengan menghidupkan sistem kontrol lampu dan memastikan bahwa lampu dinyalakan dengan baik. Setelah itu, voltmeter dihubungkan ke titik pengukuran yang tepat pada rangkaian, yang biasanya terhubung langsung ke lampu.

Kemudian, tegangan pada voltmeter akan terbaca, dan pengujian dilakukan untuk memverifikasi bahwa nilai tegangan sesuai dengan yang diharapkan dan sesuai dengan spesifikasi lampu yang digunakan. Hal ini penting untuk memastikan bahwa tegangan yang diberikan tidak terlalu rendah atau terlalu tinggi, karena dapat berdampak pada kinerja dan umur lampu.

Dengan melakukan pengujian nilai tegangan pada saat lampu dinyalakan, kita dapat memastikan bahwa sistem kontrol lampu memberikan tegangan yang sesuai dan aman untuk pengoperasian lampu.

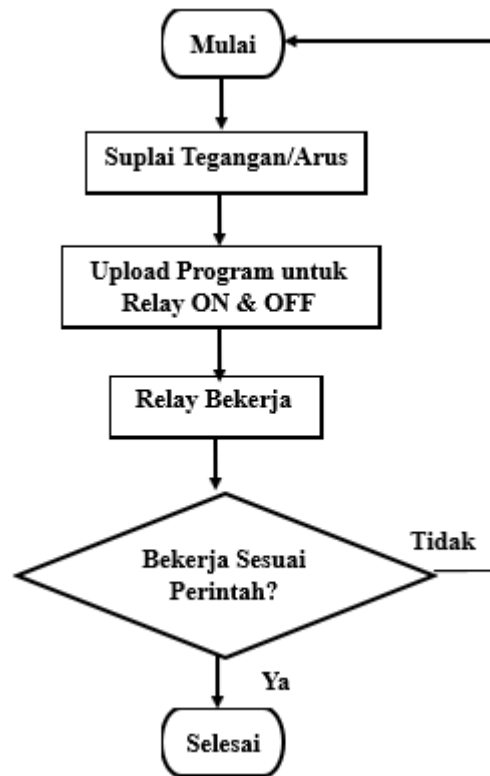
Tabel 3. 3.Hasil Pengukuran Kapasitas

PERCOBAAN LAMPU KE	NILAI TEGANGAN (V)	DAYA LAMPU (Watt)	ARUS PADA LAMPU (mA)
1	227,9	10	43,87
2	226,7	9,8	43,64
3	226,8	9,9	43,66
4	227,5	9,9	43,80
5	227,0	9,9	43,70
6	226,4	9,8	43,59

Berdasarkan tabel 3.3 yang diberikan, percobaan lampu menunjukkan hasil yang cukup memuaskan. Nilai tegangan yang terukur berkisar antara 226,4 V hingga 227,9 V, menunjukkan bahwa tegangan yang diberikan pada lampu relatif stabil dan berada dalam kisaran yang diharapkan. Selain itu, daya lampu yang digunakan berkisar antara 9,8 Watt hingga 10 Watt, sesuai dengan spesifikasi yang diberikan. Begitu pula dengan arus pada lampu, yang berada dalam rentang 43,59 mA hingga 43,87 mA, menunjukkan kestabilan dalam pengaliran arus. Dengan demikian, kesimpulan yang dapat diambil adalah bahwa sistem kontrol lampu dalam percobaan ini berfungsi dengan baik dan mampu memberikan tegangan yang stabil serta sesuai untuk pengoperasian lampu.

Pada tahap ini dilaksanakan pengujian integrasi perangkat keras dengan *Firestore*: Setelah perangkat keras dan *Firestore* dikonfigurasi, langkah selanjutnya adalah mengintegrasikan perangkat keras dengan *Firestore*. Ini melibatkan pengaturan koneksi antara *ESP32* dan *Firestore* dan pengaturan aturan *Firestore* untuk memproses data dari perangkat keras.

3.2.6. Pengujian Relay



Gambar 3. 11. Flowchart pengujian relay

Gambar 3.11 menampilkan flowchart pengujian relay yang bertujuan untuk menguji respons relay terhadap perintah yang diberikan oleh mikrokontroler. Pengujian ini bertujuan untuk memverifikasi apakah relay dapat menghubungkan atau memutuskan aliran arus OUTPUT sesuai dengan perintah yang diterima dari mikrokontroler.

```

#define RELAY_PIN 21

void setup() {

  pinMode(RELAY_PIN, OUTPUT);

}

void loop() {

  digitalWrite (21 ,LOW);
  delay(3000);
  digitalWrite (21,HIGH);
}

```

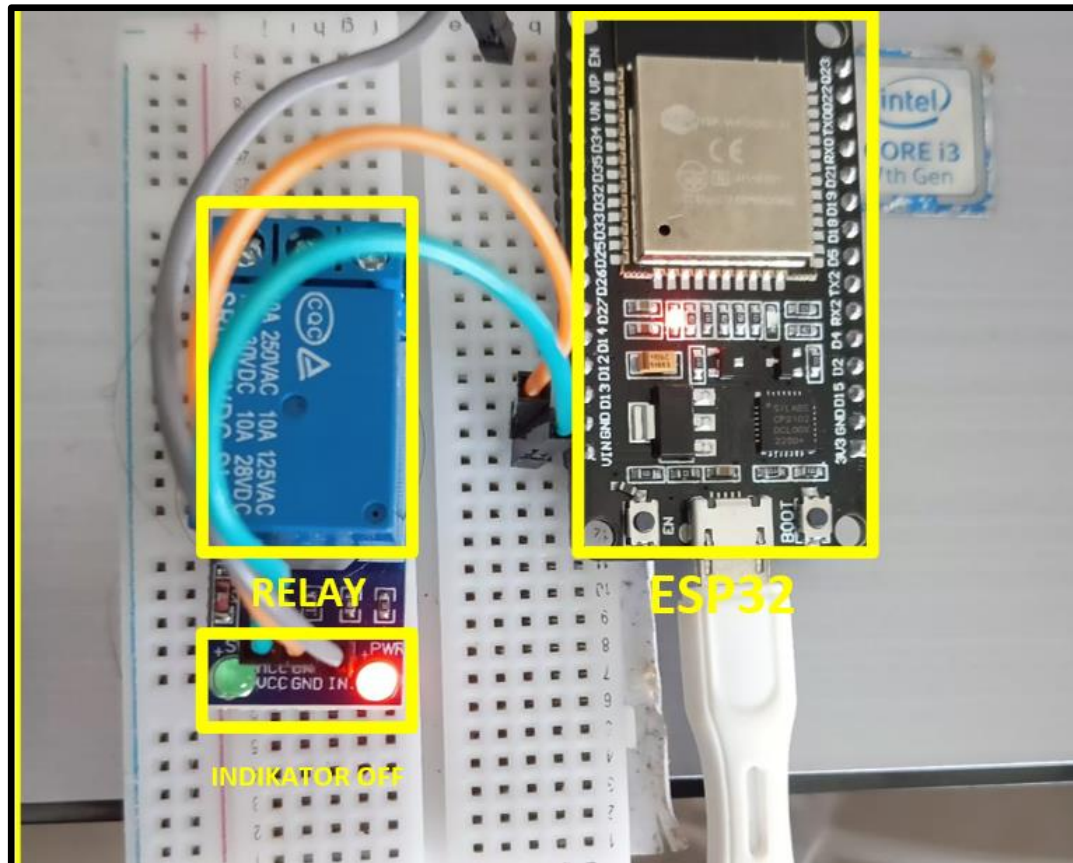
Gambar 3. 12. Program Pengujian Relay

Gambar 3.12 menampilkan program pengujian relay menggunakan ESP32 untuk mengendalikan relay. Program ini berfungsi untuk mengaktifkan relay, menunda selama 3 detik, dan kemudian mematikan relay..

Tabel 3. 4.Konektivitas Pin *Relay* Dan *ESP32*

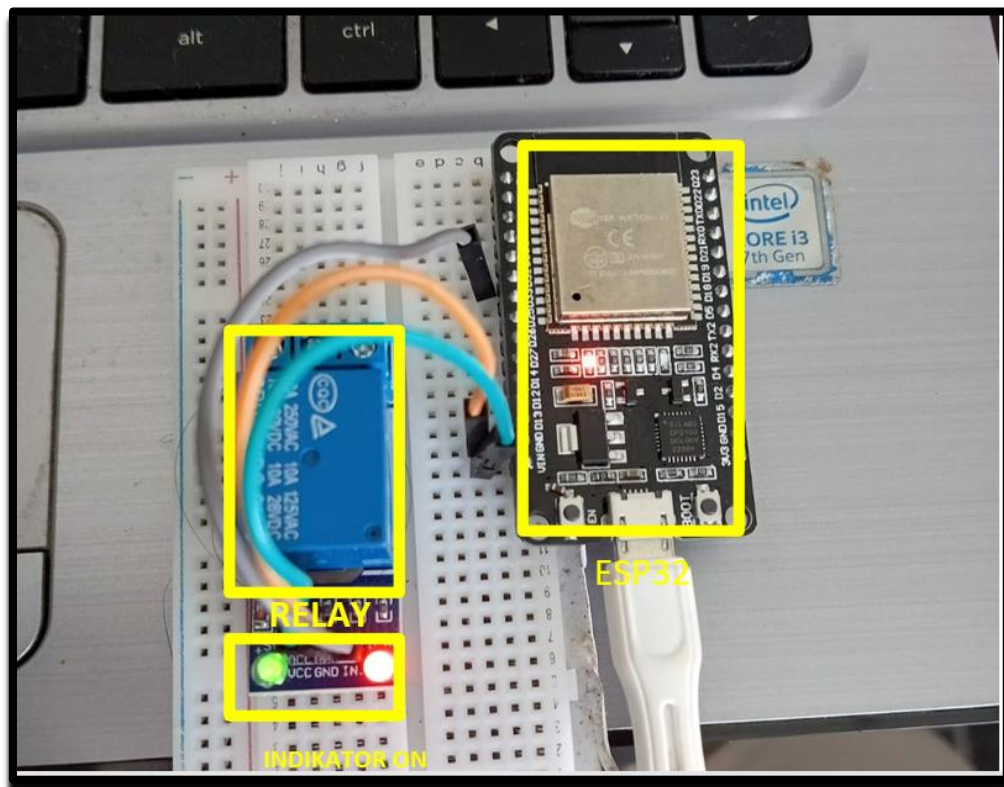
RELAY	ESP32
VCC	5V
GND	GND
OUT	12

Tabel 3.4 adalah *wiring* antara relay dan *ESP32* dengan pin 12 sebagai *output* untuk mengatur relay.



Gambar 3. 13. Hasil Pengujian Ketika Relay Dalam Kondisi *OFF*

Gambar 3.13 menampilkan hasil pengujian saat relay dalam keadaan mati (*OFF*). Hasil yang terlihat pada indikator LED hijau yang tidak menyala menunjukkan bahwa pin NO dan COM tidak terhubung.



Gambar 3. 14. Pengujian Relay ON

Gambar 3.14 menampilkan hasil pengujian saat relay dalam keadaan menyala (ON). Hasil yang terlihat pada indikator LED hijau yang menyala menunjukkan bahwa pin NO dan COM terhubung..

Tabel 3. 5.Hasil Pengujian *Relay*

No	Input	Output	Waktu (t)
1	LOW	Tersambung	1
2.	HIGH	Terputus	4
4	LOW	Tersambung	7
5	HIGH	Terputus	10

Tabel 3.5 adalah catatan data yang diambil selama lima kali pengujian dengan interval waktu 3 detik. Hasil pengujian menunjukkan bahwa relay berfungsi dengan baik dan merespons perintah yang diberikan oleh ESP32 sesuai yang diharapkan.

3.2.7. Pengujian Unit

Tujuan dari pengujian unit adalah untuk mengetahui tingkat akurasi dari sensor maupun komponen pendukung yang digunakan pada penelitian ini. Nilai akurasi sensor dihitung menggunakan persamaan:

$$\%Akurasi = 100\% - \%Error \quad (3.1)$$

$\%Error$ merupakan persentase kesalahan sensor dalam melakukan pembacaan nilai yang terdeteksi dengan pembacaan oleh alat ukur utama. Persentase *error* pengukuran pada sensor-sensor yang digunakan pada sistem dihitung menggunakan persamaan (Leny and Haryudo, 2019):

$$\%Error = \frac{\text{Nilai Sebenarnya Terbaca}}{\text{Nilai Sebenarnya}} \times 100 \quad (3.2)$$

Nilai sebenarnya merupakan nilai hasil pengukuran oleh alat ukur pembanding menggunakan multimeter. Nilai terbaca merupakan nilai yang hasil pengukuran sensor cahaya

Untuk menghitung standar deviasi, digunakan persamaan di bawah ini:

$$\sigma = \sqrt{\frac{\sum (X_i - \bar{X})^2}{n}} \quad (3.3)$$

Simbol yang digunakan pada persamaan (3.3) terdiri dari:

σ = Standar deviasi

X_i = Nilai data X ke-i

\bar{X} = Nilai rata-rata dari data X

n = Banyaknya data yang diambil

Setelah mendapatkan nilai standar deviasi, maka dapat dilakukan perhitungan standar *error* (Su'ud, M.TI, 2018). Penggunaan persamaan standar deviasi bertujuan untuk mengetahui standar *error* (SE) dari data pengujian dengan persamaan (3.4):

$$SE = \frac{\sigma}{\sqrt{n}} \quad (3.4)$$

Pengujian unit ini terdiri dari pengujian-pengujian yang tercantum di bawah ini:

Pada tahap ini merakit perangkat keras sesuai dengan rancangan sistem.

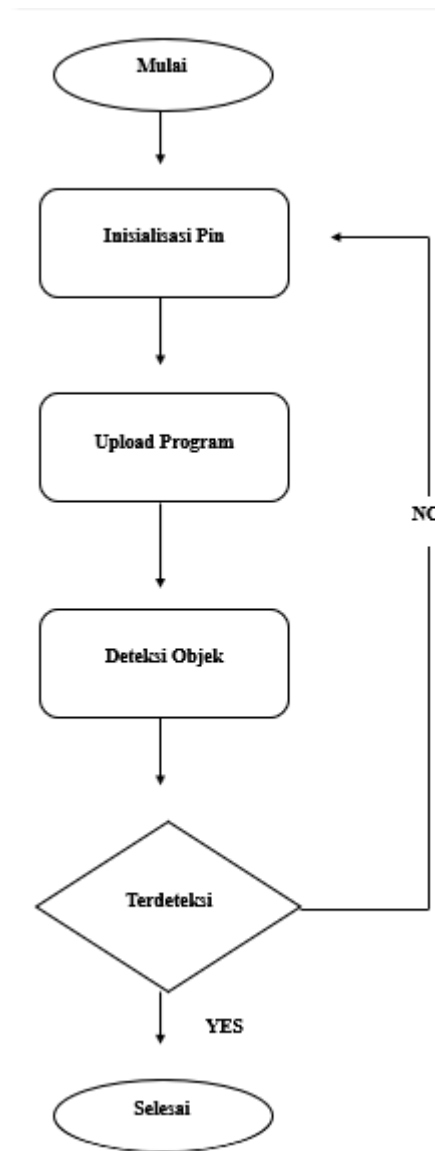
Ini melibatkan langkah-langkah berikut:

1. Memeriksa dan mempersiapkan semua komponen yang diperlukan dan komponen pendukung lainnya, termasuk ESP32, Relay, LDR, dan komponen lain yang diperlukan.
2. Mengupload program ke ESP32 yang telah disesuaikan untuk mengatur waktu penyalaan relay dan membaca nilai sensor cahaya.
3. Mengikuti skema yang telah dirancang sebelumnya untuk menentukan posisi dan koneksi setiap komponen, termasuk ESP32, Relay, dan LDR.
4. Memasang ESP32 pada breadboard dengan hati-hati, memperhatikan penempatan pin yang benar.

5. Memasang dan menghubungkan lampu dengan Relay menggunakan kabel jumper dan memperhatikan penempatan pin yang benar.
6. Memasang *Relay* yang akan digunakan, memastikan kabelnya terhubung dengan benar ke *ESP32*.
7. Memasang sensor cahaya (LDR) pada breadboard dengan hati-hati, memperhatikan penempatan pin yang benar.
8. Menghubungkan salah satu pin sensor cahaya (LDR) ke pin analog *ESP32* menggunakan kabel jumper.
9. Memastikan koneksi antara sensor cahaya (LDR) dan *ESP32* terpasang dengan baik.
10. Melakukan pengaturan pada program *ESP32* untuk membaca nilai sensor cahaya menggunakan pin yang telah ditentukan.
11. Mengatur koneksi *NTPClient* pada program *ESP32* untuk mendapatkan waktu dari server *NTP*.
12. Mengambil nilai waktu penyalan relay dari *Firestore* untuk menentukan waktu kapan relay harus dinyalakan atau dimatikan.
13. Mengambil nilai status penerangan dari *Firestore* untuk setiap lampu yang terhubung dengan relay.
14. Jika waktu saat ini = waktu penyalan relay, lanjutkan ke langkah 15. Jika tidak, lanjutkan ke langkah 18.
15. Jika status penerangan = ON, lanjutkan ke langkah 16. Jika status penerangan = OFF, lanjutkan ke langkah 19.
16. Membaca nilai sensor cahaya (LDR).

17. Jika nilai sensor cahaya rendah, lanjutkan ke langkah 18. Jika nilai sensor cahaya tinggi, lanjutkan ke langkah 20.
18. Menghidupkan relay untuk lampu yang sesuai. Simpan status penerangan = ON ke Firebase untuk lampu tersebut. Kembali ke langkah 12.
19. Matikan relay untuk lampu yang sesuai. Simpan status penerangan = OFF ke Firebase untuk lampu tersebut. Kembali ke langkah 12.
20. Tunggu beberapa detik. Kembali ke langkah 12.
21. Selesai: Menandakan akhir dari program.

3.2.7.1. Pengujian Sensor Cahaya



Gambar 3. 15. *Flowchart* Pengujian Sensor *LDR*

Pada Gambar 3.15 adalah *flowchart* pengujian sensor *LDR* (cahaya) melibatkan verifikasi kemampuan sensor untuk mendeteksi cahaya atau intensitas cahaya yang berada disekitar sensor tersebut. Langkah pengujian intensitas cahaya yaitu

membaca nilai intensitas cahaya yang terdeteksi oleh sensor dan periksa hasilnya melalui *serial monitor*.

Pengujian ini membantu memverifikasi kinerja sensor *LDR* dan memastikan bahwa sensor dapat diandalkan dalam sistem yang membutuhkan pembacaan perubahan nilai intensitas cahaya..

```
1  const int ldrPin = A0; // Pin analog untuk sensor LDR
2
3  void setup() {
4      Serial.begin(115200);
5      pinMode(ldrPin, INPUT);
6  }
7
8  void loop() {
9      int ldrValue = analogRead(ldrPin); // Membaca nilai analog dari sensor LDR
10     Serial.print("Nilai LDR: ");
11     Serial.println(ldrValue);
12
13     delay(1000); // Jeda 1 detik antara setiap pembacaan nilai LDR
14 }
15
16
17
```

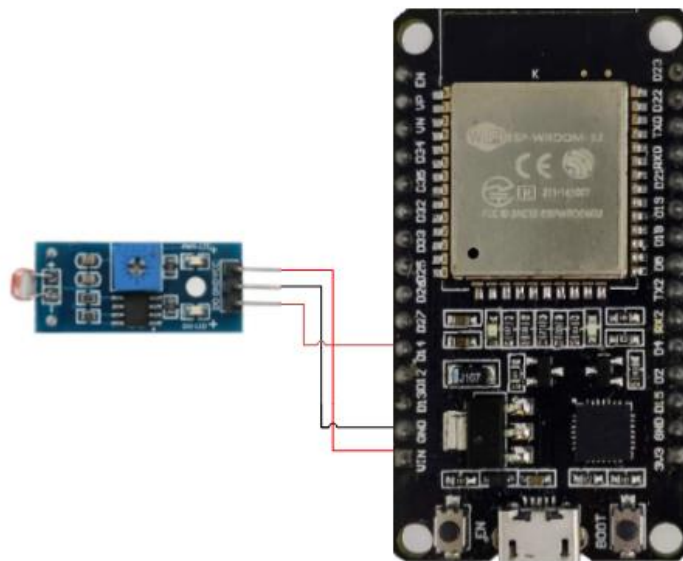
Gambar 3. 16. Program Pengujian Sensor LDR

Pada Gambar 3.16 dengan menggunakan pin analog (pin A0) pada ESP32 untuk terhubung dengan sensor cahaya (LDR). Program ini akan membaca nilai intensitas cahaya dari sensor LDR dan membandingkannya dengan nilai ambang (500). Jika nilai intensitas cahaya yang terbaca lebih rendah dari ambang batas, maka serial monitor akan menampilkan "Objek Terdeteksi". Jika nilai intensitas cahaya lebih tinggi dari ambang batas, maka serial monitor akan menampilkan "Tidak ada objek".

Tabel 3. 6.Konektivitas Pin *ESP32* dan *LDR Sensor*

LDR Sensor	ESP32
VCC	5V
GND	GND
OUT	34

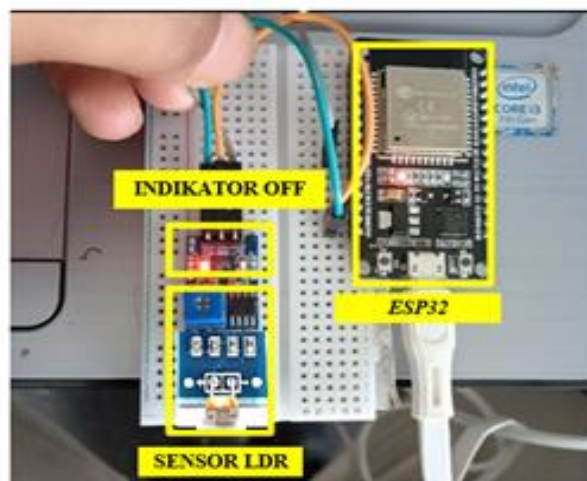
Pada Tabel 3.6 adalah konfigurasi pin yang digunakan antara *ESP32* dan *Sensor LDR* dengan pin 34 sebagai input nilai intensitas cahaya.



Gambar 3. 17. Wiring Sensor LDR

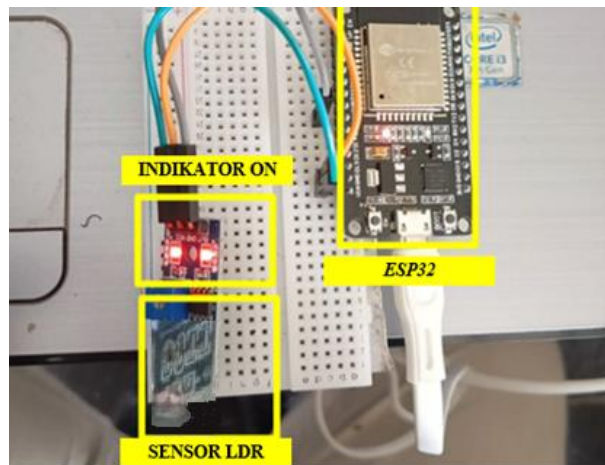
Pada Gambar 3.17, terlihat rangkaian sensor LDR yang menarik perhatian karena meskipun tidak ada input yang secara eksplisit diberikan pada sensor, sensor ini tetap memberikan input ke mikrokontroler ESP32. Rangkaian ini

menghubungkan pin 34 dari sensor ke pin A0 pada mikrokontroler ESP32, sedangkan pin positif (+) sensor dihubungkan ke pin 5V pada papan Arduino, dan pin negatif (-) sensor dihubungkan ke pin GND pada papan Arduino. Perlu diperhatikan bahwa VCC dan GND pada sensor tidak dihubungkan dengan komponen apapun.



Gambar 3. 18. Rangkaian Pengujian Sensor LDR Ketika Tidak Mendeteksi Objek

Pada Gambar 3.18 dilakukan pengujian sensor dengan membaca nilai intensitas cahaya lebih tinggi. Maka indikator *led* sebelah kanan mati dan *serial monitor* akan menampilkan hasil seperti Gambar 3.17.



Gambar 3. 19. Pengujian Pengujian Sensor *LDR*

Pada Gambar 3.19 dilakukan pengujian dengan membaca nilai intensitas cahaya. Lalu *led menyala* indikator *led sebelah kanan menyala* dan *serial monitor* akan menampilkan nilai cahaya lebih rendah.

3.2.7.2. Pengujian Mikrokontroler ESP32

Tujuan dari pengujian ESP32 sebagai mikrokontroler utama dalam penelitian ini adalah sebagai berikut:

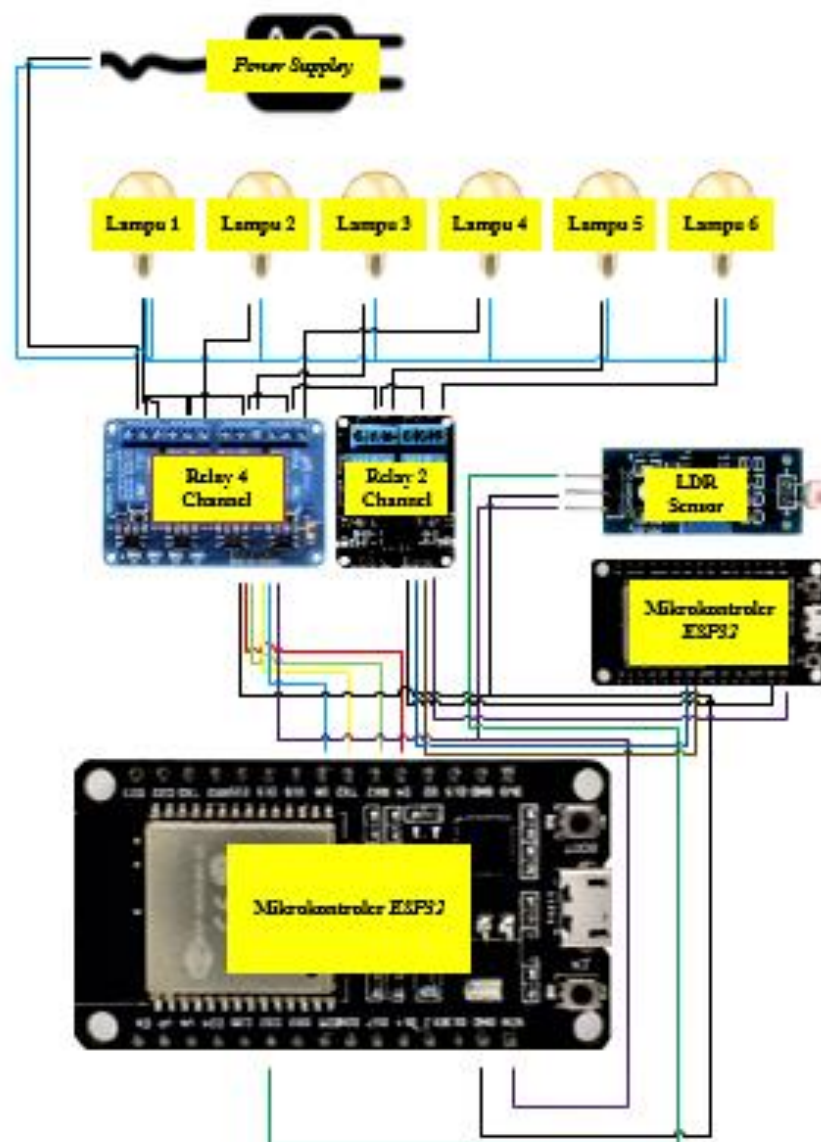
1. Untuk mengevaluasi kesesuaian pin I/O pada ESP32 dengan pin I/O dari komponen-komponen yang digunakan dalam sistem kontrol penerangan.
2. Untuk mengidentifikasi kondisi tegangan ESP32 saat tidak ada komponen tambahan terhubung dan saat memberikan pasokan tegangan kepada komponen-komponen yang digunakan dalam sistem kontrol penerangan.
3. Untuk menentukan apakah diperlukan sumber daya eksternal untuk mengoperasikan komponen-komponen utama dan pendukung dalam sistem kontrol penerangan.

Berikut adalah hasil pengujian dari penggunaan ESP32 sebagai mikrokontroler utama dalam penelitian ini::

1. Hasil pengujian menunjukkan bahwa penelitian ini melibatkan komponen-komponen yang membutuhkan 5 pin I/O (4 pin untuk relay dan 1 pin untuk sensor cahaya LDR).
2. Tegangan Dalam pengujian, tegangan keluaran ESP32 diukur menggunakan multimeter dengan mengukur tegangan antara VCC dan GND pada ESP32. Hasilnya menunjukkan bahwa tegangan keluaran ESP32 saat tidak ada komponen tambahan terhubung berkisar antara 4,9V hingga 5,02V
3. Saat semua komponen yang diperlukan dalam penelitian terhubung dengan ESP32, semua komponen dapat diberikan pasokan daya dan berfungsi dengan baik sesuai tugasnya.

3.2.8. Perakitan Sistem

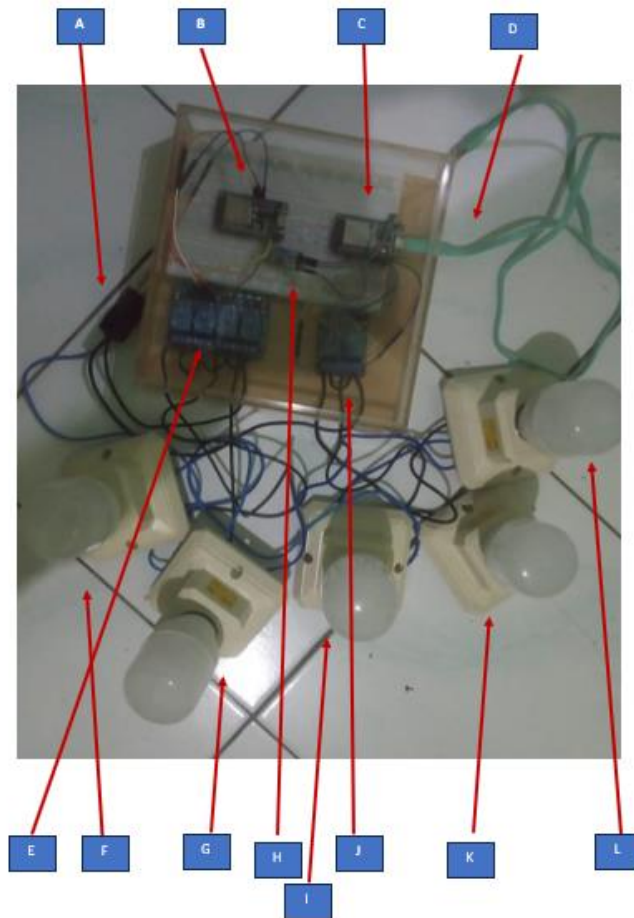
Perakitan sistem adalah proses menggabungkan seluruh alat dan komponen pendukung sesuai dengan desain yang telah direncanakan. Tujuannya adalah menciptakan sistem yang berfungsi sesuai dengan konsep kerja yang telah ditentukan sebelumnya.



Gambar 3. 20. *Wiring* Sistem Kontrol Penerangan Otomatis Menggunakan Mikrokontroler *ESP32* Berbasis *Internet of Things*

Gambar 3.20 menunjukkan susunan kabel dan konektor dalam sistem kontrol penerangan otomatis menggunakan mikrokontroler ESP32 berbasis Internet of Things (IoT). Pada gambar tersebut, terlihat beberapa komponen utama

yang saling terhubung dengan ESP32 sebagai inti sistem. Salah satunya adalah sensor cahaya yang berfungsi untuk mendeteksi tingkat pencahayaan di sekitarnya. Data dari sensor cahaya ini akan digunakan oleh ESP32 untuk mengambil keputusan terkait penerangan otomatis sesuai kondisi pencahayaan yang ada. Selain itu, beberapa lampu juga terlihat terhubung dengan ESP32. Lampu-lampu ini akan dikendalikan oleh sistem berdasarkan informasi dari sensor cahaya dan perintah yang diterima dari platform IoT. Sistem ini juga terhubung dengan platform IoT melalui koneksi internet, yang memungkinkan pengguna untuk mengontrol dan memantau sistem secara jarak jauh. Dengan perakitan ini, sistem kontrol penerangan otomatis dapat berfungsi secara efisien dan terintegrasi dengan internet, sehingga memungkinkan operasi otomatis dan akses kendali dari mana saja melalui platform IoT.



Gambar 3. 21 Sistem Kontrol Penerangan Otomatis Menggunakan Mikrokontroler ESP32 Berbasis Internet of Things

Gambar 3.21. menunjukkan rangkaian sistem kontrol penerangan otomatis dari *wiring* Gambar 3.19. yang terdiri dari:

- a. Steker/Staker, untuk menghubungkan lampu dengan aliran listrik
- b. Modul Mikrokontroler ESP32 mikrokontroler yang menghubungkan dengan sistem *internet of things* yang mengendalikan 3 lampu sebagai output
- c. Modul Mikrokontroler ESP32 mikrokontroler yang

menghubungkan dengan sistem *internet of things* yang mengendalikan 2 lampu sebagai output

- d. Kabel Micro USB, sebagai penghubung mikrokontroler ESP32 dan juga sebagai penghubung pada saat penguploadan program
- e. Modul relay 4 *channel*, relay kontrol penerangan otomatis pada 3 buah lampu dan untuk menyambungkan dan memutuskan aliran listrik lampu.
- f. Lampu 1, objek yang dikendalikan
- g. Lampu 2, objek yang dikendalikan
- h. Sensor LDR, sensor pengukur cahaya pada ruangan
- i. Lampu 3, objek yang dikendalikan
- j. Modul relay 2 *channel*, relay kontrol penerangan otomatis pada 2 buah lampu sekaligus relay untuk menyambungkan dan memutuskan aliran listrik lampu.
- k. Lampu 4, objek yang dikendalikan
- l. Lampu 5, objek yang dikendalikan

3.2.9. Pengujian Sistem

Pengujian sistem adalah proses pengujian secara menyeluruh terhadap seluruh komponen yang membentuk alat dalam penelitian ini.

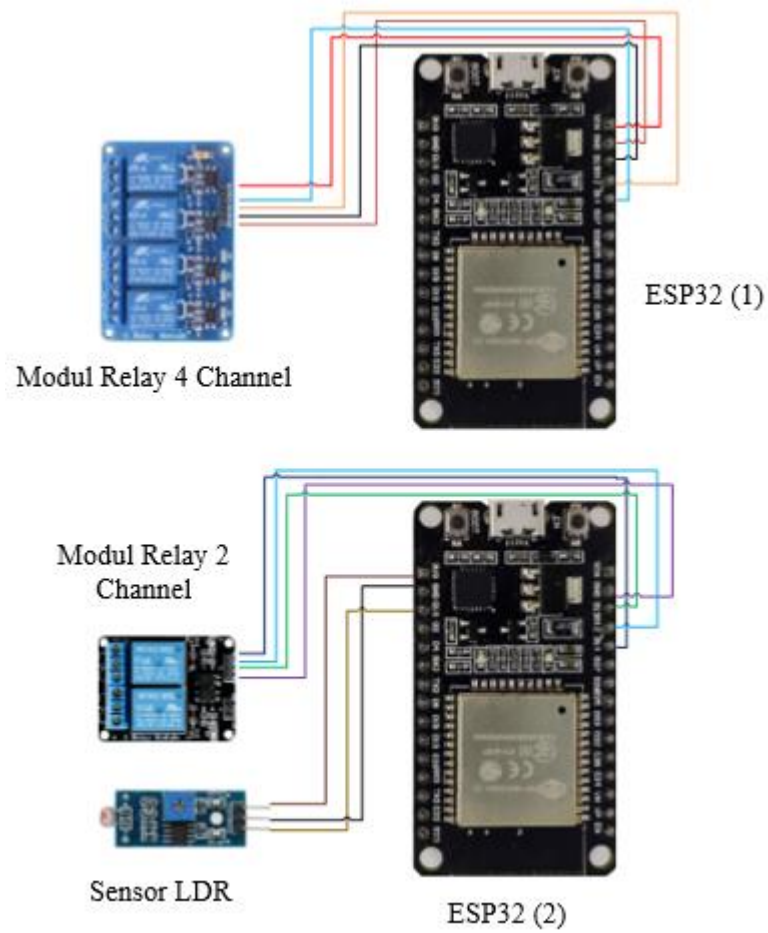
3.2.9.1. Pengujian Sistem Kontrol

Pengujian sistem kontrol penerangan otomatis menggunakan mikrokontroler ESP32 berbasis Internet of Things (IoT) adalah tahap krusial dalam pengembangan alat ini. Dalam proses pengujian ini, berbagai langkah dilakukan untuk memastikan keseluruhan sistem berfungsi dengan baik dan dapat menghadapi situasi nyata dengan akurat. Pertama, dilakukan pengujian fungsionalitas mikrokontroler ESP32 untuk memverifikasi apakah mikrokontroler berfungsi sesuai harapan dan dapat berkomunikasi dengan komponen lainnya. Selanjutnya, sensor cahaya diuji untuk memastikan sensitivitas dan akurasi dalam mendeteksi tingkat pencahayaan di sekitarnya. Pengujian juga mencakup verifikasi bahwa sensor memberikan respons yang tepat saat kondisi pencahayaan berubah.

Setelah itu, dilakukan pengujian koneksi dengan platform Firebase untuk memastikan sistem dapat terhubung dengan baik ke internet dan platform Firebase dengan stabil. Data dari sensor cahaya dikirimkan dan diunggah ke platform Firebase untuk memantau dan mengontrol sistem secara jarak jauh. Pengujian ini penting untuk memastikan sistem memiliki konektivitas yang handal dan responsif terhadap perintah dari platform Firebase. Selanjutnya, sistem kendali penerangan otomatis diuji dalam berbagai situasi untuk memastikan kemampuannya dalam mengenali perubahan pencahayaan dan menyesuaikan

tingkat penerangan lampu secara otomatis. Responsivitas dan keandalan sistem diperiksa untuk memastikan bahwa sistem dapat beroperasi secara konsisten dan dapat diandalkan dalam jangka waktu yang lebih lama.

Selain itu, pengujian integrasi dan keselarasan antara komponen sistem dilakukan untuk memastikan bahwa semua bagian sistem terhubung dengan benar dan berfungsi bersama secara harmonis sesuai dengan desain yang telah direncanakan. Sistem juga diuji dalam kondisi ekstrim dan mengalami gangguan untuk menguji daya tahannya dalam situasi yang tidak diharapkan. Seluruh aspek keselamatan sistem juga diuji untuk memastikan penggunaan yang aman dan menghindari risiko yang tidak diinginkan. Setelah semua tahap pengujian berhasil dilalui dan sistem terbukti beroperasi dengan baik, alat kendali penerangan otomatis menggunakan mikrokontroler ESP32 berbasis Internet of Things siap untuk diimplementasikan secara praktis dan memberikan manfaat yang signifikan dalam penggunaan sehari-hari.



Gambar 3. 22 Wiring Sistem Kendali

Gambar 3.22 menggambarkan konfigurasi koneksi mikrokontroler yang berfungsi sebagai otak pengendali dari seluruh sistem dalam penelitian ini. Pada sistem ini, mikrokontroler bertanggung jawab untuk mengendalikan objek yang ditampilkan dalam Gambar 3.19.

Tabel 3. 7 Detail Hubungan Pin Kendali Antara Sensor Relay Dengan Mikrokontroler ESP32

Jenis Komponen	Pin Mikrokontroler ESP32
Sensor LDR	D15 ESP (2)
Modul Relay 4 Channel Pertama	
- IN4 (<i>Charging</i> Baterai 1)	D4 (ESP32(1))
- IN3 (<i>Discharging</i> Baterai 1)	D5 (ESP32(1))
- IN2 (<i>Charging</i> Baterai 2)	D18 (ESP32(1))
- IN1 (<i>Discharging</i> Baterai 2)	D19 (ESP32(1))
Modul Relay 2 Channel Kedua	
- IN1 (<i>Charging</i> Baterai 3)	D33 ESP (2)
- IN2 (<i>Discharging</i> Baterai 1)	D32 ESP (2)

Berdasarkan Tabel 3.7. dapat diketahui posisi pin dari kedua buah modul relay 4 *channel* dan 2 *channel* yang dihubungkan dengan Mikrokontroler ESP32. Tahap-tahap pengujian sistem kontrol penerangan otomatis lampu adalah seperti di bawah ini:

- a. Ketika memasukan nilai input lampu 1 pada realtime database, maka relay kontrol penerangan otomatis ESP32 (1) pada lampu 1 terhubung ke lampu, proses tetap sama untuk lampu 2 dan lampu 3.
- b. Saat sensor menerima input cahaya yang sesuai dengan batas nilai pada program ESP32 (2), maka relay kontrol penerangan otomatis ESP32 (2) pada lampu 5 terhubung ke lampu
- c. Ketika memasukan nilai input lampu 4 pada realtime database, maka relay kontrol penerangan otomatis ESP32 (2) pada lampu 4 terhubung ke lampu,

3.2.9.2. Pengujian Sistem Internet of Things

Pengujian Sistem Internet of Things merupakan tahap yang krusial dalam memastikan kinerja dan kehandalan sistem. Dalam pengujian ini, berbagai aspek yang terkait dengan fungsionalitas, konektivitas, dan integrasi sistem diuji secara menyeluruh.

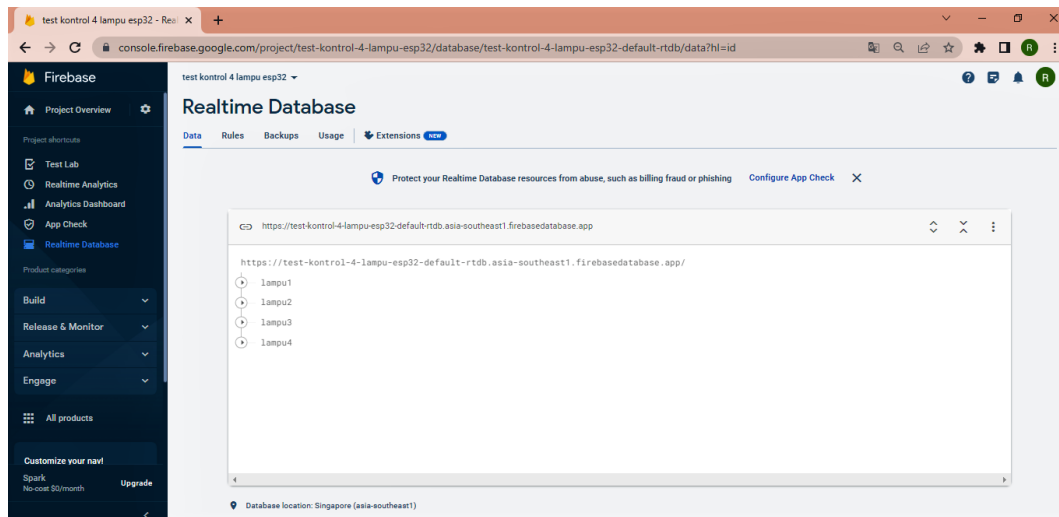
3.2.9.2.1. Pengujian Komunikasi ESP32



Gambar 3. 23 Alur Pengujian Komunikasi ESP32

Gambar 3.23 dilakukan pengujian untuk memastikan mikrokontroler ESP32 berfungsi dengan baik dan dapat berkomunikasi dengan sensor cahaya serta lampu. Respons dan performa sensor cahaya juga diuji untuk memastikan keakuratannya dalam mengukur tingkat pencahayaan. Selanjutnya, sistem diuji untuk memverifikasi koneksi dengan platform Firebase agar dapat mengirim dan menerima data dari jarak jauh.

3.2.9.2.2. Pengujian Penampilan Data Pada Firebase



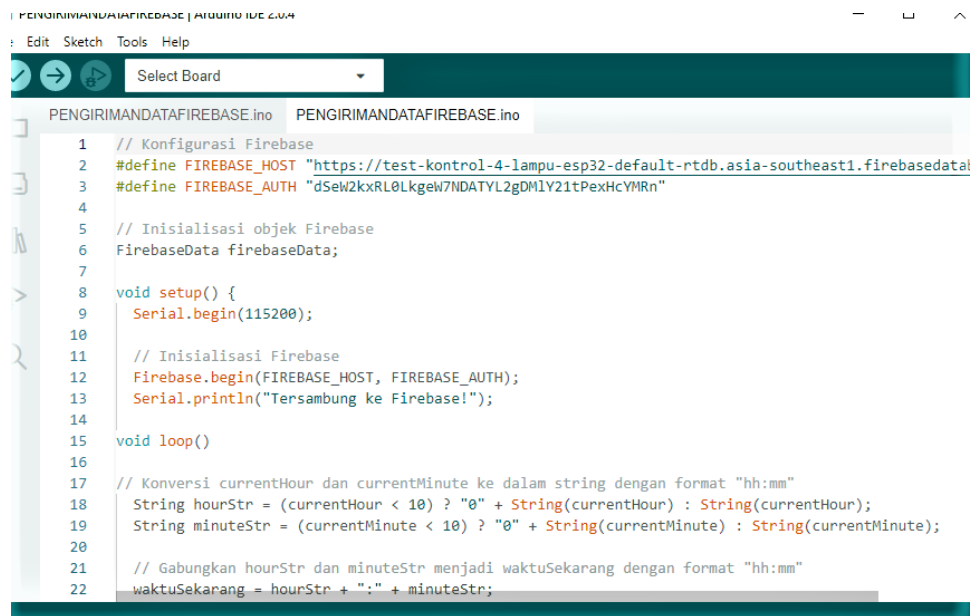
Gambar 3. 24 Struktur *Database*

Dalam konfigurasi *Database Firebase* berikut langkah-langkahnya :

1. Membuat akun proyek *Firebase*
2. Membuat *Database*
3. Mengkonfigurasi autentikasi dan perizinan akses
4. Menyusun sturuktur data.

Gambar 3.24 Pada pengujian juga melibatkan verifikasi respons sistem terhadap perintah dari platform *Firebase* dalam mengendalikan penerangan otomatis secara akurat dan tepat waktu. Seluruh komponen sistem juga diintegrasikan dan diuji untuk memastikan bahwa sistem dapat berfungsi harmonis sesuai dengan rencana desain. Aspek keamanan dan keselamatan sistem juga menjadi perhatian dalam pengujian untuk memastikan sistem dapat digunakan dengan aman oleh pengguna. Setelah pengujian berhasil dan semua

komponen sistem terbukti berfungsi sesuai harapan, sistem kontrol penerangan otomatis menggunakan mikrokontroler ESP32 berbasis Firebase siap untuk diimplementasikan secara praktis dan diharapkan dapat memberikan solusi yang efisien dan cerdas dalam mengatur penerangan secara otomatis.



```

1 // Konfigurasi Firebase
2 #define FIREBASE_HOST "https://test-kontrol-4-lampu-esp32-default-rtdb.asia-southeast1.firebaseio.com"
3 #define FIREBASE_AUTH "dSeW2kxRL0LkgeW7NDATYL2gDM1Y21tPexHcYMRn"
4
5 // Inisialisasi objek Firebase
6 FirebaseData firebaseData;
7
8 void setup() {
9   Serial.begin(115200);
10
11   // Inisialisasi Firebase
12   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
13   Serial.println("Tersambung ke Firebase!");
14
15 void loop()
16
17 // Konversi currentHour dan currentMinute ke dalam string dengan format "hh:mm"
18 String hourStr = (currentHour < 10) ? "0" + String(currentHour) : String(currentHour);
19 String minuteStr = (currentMinute < 10) ? "0" + String(currentMinute) : String(currentMinute);
20
21 // Gabungkan hourStr dan minuteStr menjadi waktuSekarang dengan format "hh:mm"
22 waktuSekarang = hourStr + ":" + minuteStr;

```

Gambar 3. 25 Pengiriman Data Oleh Mikrokontroler ESP32

Gambar 3.25 menggambarkan proses pengiriman data oleh mikrokontroler ESP32 menggunakan NTPClient dalam sistem. NTPClient berperan sebagai alat untuk mengakses dan mendapatkan data waktu yang akurat dari server waktu (NTP server) yang terhubung dengan internet. Pada awal proses, ESP32 menggunakan NTPClient untuk mengambil data waktu yang tepat dari NTP server. Setelah memperoleh data waktu yang akurat, ESP32 menggunakan koneksi jaringan, seperti WiFi, untuk mengirimkan data waktu tersebut ke tujuan yang ditentukan.

Proses pengiriman data dimulai dengan ESP32 mengakses NTPClient dan mengirimkan permintaan untuk mendapatkan informasi waktu dari server waktu yang terhubung. Setelah menerima respons dari server, data waktu yang diperoleh dikirim melalui koneksi jaringan ke perangkat lain atau platform yang membutuhkan data waktu yang tepat. Penggunaan NTPClient pada ESP32 memastikan bahwa sistem memiliki referensi waktu yang tepat dan sinkron dengan waktu yang sebenarnya di luar sistem, yang sangat penting dalam berbagai aplikasi IoT yang memerlukan waktu yang akurat, seperti sistem kontrol penerangan otomatis berbasis waktu. Dengan adanya NTPClient, ESP32 dapat memberikan data waktu yang andal dan akurat, sehingga sistem dapat berfungsi dengan tepat dan diandalkan dalam mengatur dan mengendalikan penerangan secara otomatis berdasarkan waktu.

3.3.Subjek dan Objek Penelitian

Pada penelitian ini subjek yang digunakan adalah berupa sensor cahaya, *setpoint* nilai cahaya sebagai batas untuk pengalihan sistem pengontrolan pada lampu. Sedangkan untuk objek penelitian yang digunakan adalah enam buah lampu dengan kapasitas daya masing-masing sebesar 5 watt sebagai media penerangan.

3.4.Analisis Data

Analisis data dilakukan untuk menyesuaikan teori yang dikaji dengan sistem kerja alat yang dirancang. Sehingga didapatkan hasil implementasi yang sesuai

rancangan dengan pembahasan secara ilmiah. Juga dapat mengetahui kelebihan maupun kelemahan sistem.

Analisis data dilakukan dengan menganalisis hasil implementasi alat dan hasil pengujian dari sistem yang dibangun dimana analisis sistem yang diuji terdiri dari analisis sistem kendali, dan sistem Internet of Things dengan tahapan seperti di bawah ini:.

1. Dilaksanakan pengujian kendali untuk memutuskan dan menyambungkan sistem kontrol penerangan terhadap enam buah lampu. Hasil dari pengendalian tersebut diuraikan secara lugas dan jelas.
2. Dilakukan pengujian Internet of Things untuk menampilkan data pengendalian masing-masing lampu, data yang ditampilkan pada sistem Internet of Things akan didokumentasikan dan diuraikan secara lugas dan jelas.
3. Seluruh hasil pengujian beserta uraiannya dituangkan pada BAB IV.

3.5.Kesimpulan

Setelah data diambil kemudian dilakukan analisa terhadap hasil pengujian, maka selanjutnya dapat menentukan kesimpulan dari hasil pengujian