

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi bertujuan untuk merealisasikan hasil perancangan menjadi sebuah sistem yang dapat berfungsi sesuai dengan kebutuhan penelitian. Implementasi pada penelitian ini terdiri atas tiga tahapan, yaitu implementasi perangkat keras, implementasi perangkat lunak, dan implementasi sistem. Implementasi perangkat keras dilakukan dengan merangkai seluruh komponen sesuai dengan skematik yang telah dirancang. Implementasi perangkat lunak dilakukan dengan mengembangkan program pada ESP32 dan Telegram Bot. Selanjutnya, implementasi sistem dilakukan untuk memastikan bahwa seluruh komponen perangkat keras dan perangkat lunak telah terintegrasi dengan baik sehingga sistem dapat beroperasi sesuai dengan tujuan penelitian.

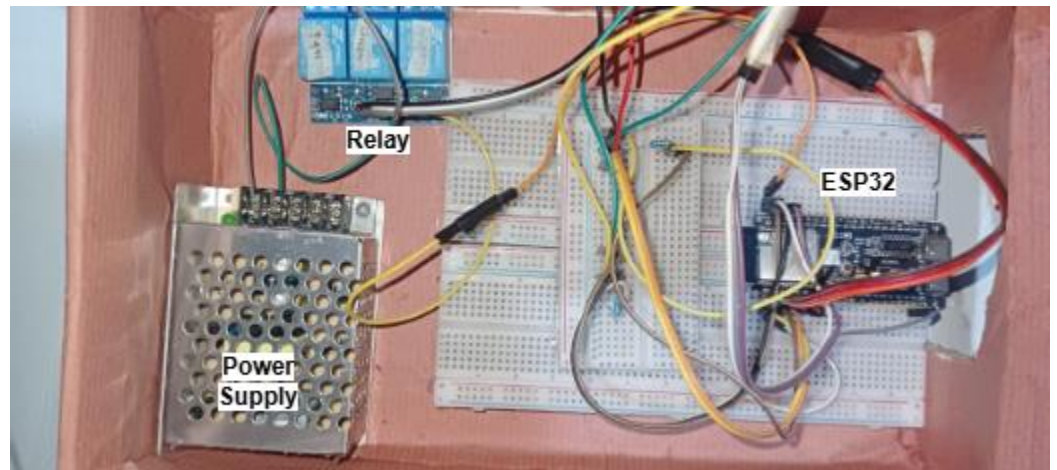
4.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras pada sistem terdiri atas perangkat masukan (input), perangkat pemrosesan, dan perangkat keluaran (output). Perangkat masukan meliputi sensor DS18B20 yang berfungsi untuk mengukur suhu ruangan, Pulse Sensor yang digunakan untuk mendeteksi denyut jantung pengguna dalam satuan *Beats Per Minute (BPM)*, serta sensor LDR yang berfungsi untuk mendeteksi kondisi pencahayaan di sekitar tempat tidur. Data yang diperoleh dari ketiga sensor tersebut kemudian diproses oleh ESP32 sebagai mikrokontroler utama untuk menentukan aksi yang akan dilakukan oleh sistem.

Perangkat keluaran terdiri atas kipas angin, lampu LED *warm light*, dan speaker yang dikendalikan melalui relay sebagai sakelar elektronik. Kipas angin digunakan untuk meningkatkan sirkulasi udara ketika suhu ruangan melebihi batas yang telah ditentukan, lampu LED berwarna

kuning digunakan untuk menciptakan suasana tidur yang lebih nyaman sesuai dengan konsep *sleep hygiene*, sedangkan speaker berfungsi memutar audio relaksasi untuk membantu pengguna mencapai kondisi yang lebih rileks sebelum tidur. Berikut implementasi perangkat keras pada Gambar 4.1.

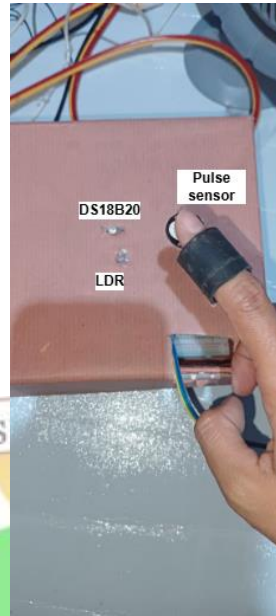




(a) Tampak di Dalam Kotak



(b) Implementasi Sistem



(c) Pulse Sensor ketika Dipakai Di Tangan

Gambar 4. 1 Implementasi Perangkat Keras

Berikut adalah keterangan gambar 4.1:

1. Perangkat masukan (input)

a. Pulse Sensor

Pulse Sensor berfungsi untuk mendeteksi denyut jantung pengguna dalam satuan *Beats Per Minute (BPM)*. Data denyut jantung yang diperoleh digunakan sebagai salah satu parameter dalam menentukan kondisi relaksasi pengguna.

b. Sensor LDR

Sensor *LDR (Light Dependent Resistor)* berfungsi untuk mendeteksi tingkat pencahayaan di sekitar tempat tidur. Perubahan intensitas cahaya akan menghasilkan perubahan nilai ADC yang dibaca oleh ESP32 sebagai dasar pengendalian lampu.

c. Sensor DS18B20

Sensor DS18B20 berfungsi untuk mengukur suhu ruangan di sekitar tempat tidur. Data suhu yang diperoleh digunakan sebagai

acuan dalam mengendalikan kipas angin untuk meningkatkan kenyamanan termal pengguna.

2. Perangkat Pemrosesan (*Processing*)

a. ESP32

ESP32 berfungsi sebagai mikrokontroler utama yang menerima data dari Pulse Sensor, sensor LDR, dan sensor DS18B20. Seluruh data tersebut diproses berdasarkan algoritma yang telah dirancang untuk menentukan kondisi kerja setiap aktuator. Selain itu, ESP32 juga mengirimkan informasi hasil pembacaan sensor dan status sistem ke Telegram Bot melalui jaringan WiFi.

b. Relay

Relay berfungsi sebagai saklar otomatis yang dikendalikan oleh ESP32 untuk menghubungkan atau memutus arus listrik menuju aktuator. Melalui relay, ESP32 dapat mengendalikan kondisi hidup dan mati dari kipas angin, lampu LED, dan speaker secara otomatis maupun berdasarkan perintah pengguna melalui Telegram.

3. Perangkat Keluaran (*Output*)

a. Kipas Angin

Kipas angin berfungsi untuk meningkatkan sirkulasi udara di sekitar tempat tidur ketika suhu ruangan melebihi batas yang telah ditentukan sehingga dapat meningkatkan kenyamanan pengguna.

b. Lampu LED *Warm Light*

Lampu LED berwarna kuning (*warm light*) berfungsi sebagai media pencahayaan yang lebih nyaman untuk mendukung suasana tidur sesuai dengan konsep sleep hygiene.

c. Speaker

Speaker berfungsi sebagai media keluaran untuk memutar audio relaksasi ketika pengguna mengaktifkan fitur Sleep Mode

maupun Relax Mode, atau ketika sistem mendeteksi kondisi tertentu berdasarkan hasil pembacaan sensor.

d. *Power Supply*

Power supply berfungsi sebagai sumber tegangan untuk menyuplai daya pada relay dan aktuator sehingga seluruh komponen dapat bekerja secara stabil sesuai dengan kebutuhan sistem.

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan tahap penerapan hasil perancangan program ke dalam sistem. Pada penelitian ini, implementasi perangkat lunak terdiri atas dua bagian, yaitu implementasi program menggunakan Arduino IDE sebagai media pemrograman ESP32 dan implementasi Telegram Bot sebagai media komunikasi antara pengguna dengan sistem. Kedua perangkat lunak tersebut saling terintegrasi sehingga sistem mampu melakukan pembacaan data sensor, pengendalian aktuator, serta pemantauan kondisi lingkungan tidur secara *real-time*.

1. Arduino IDE

Implementasi Arduino IDE dilakukan sebagai tahap pengembangan program yang akan dijalankan oleh mikrokontroler ESP32. Seluruh logika pengendalian sistem ditulis menggunakan bahasa pemrograman C++ pada Arduino IDE, kemudian dikompilasi dan diunggah ke ESP32. Program yang dibuat meliputi proses inisialisasi perangkat, koneksi WiFi, komunikasi dengan Telegram Bot, pembacaan data dari Pulse Sensor, sensor DS18B20, dan sensor LDR, serta pengendalian relay untuk mengaktifkan atau menonaktifkan kipas angin, lampu LED warm light, dan speaker sesuai dengan algoritma yang telah dirancang.

Selain itu, program juga mengimplementasikan beberapa fungsi utama, yaitu pembacaan data sensor secara berkala, pengiriman data ke Telegram Bot, serta pemrosesan perintah yang dikirimkan oleh pengguna. Implementasi program pada Arduino IDE ditunjukkan pada Gambar 4.2.

```
float tempThreshold = 22.0;

if (temperatureC > tempThreshold) {

    digitalWrite(RELAY_FAN, HIGH);
    fanStatus = "ON";

} else {

    digitalWrite(RELAY_FAN, LOW);
    fanStatus = "OFF";

}
```

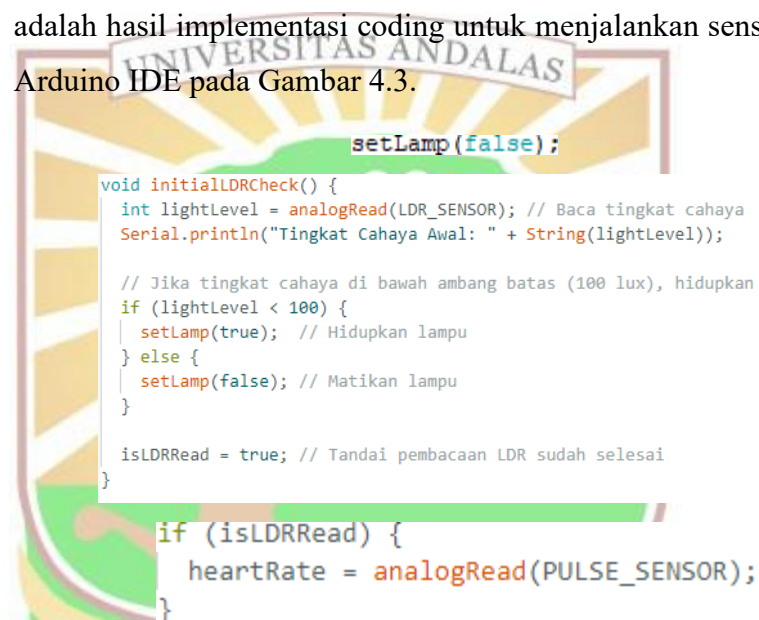
Gambar 4.2 Source Code sensor DS18B20

Gambar 4.2 menunjukkan implementasi source code sensor DS18B20 yang berfungsi untuk mengukur suhu di sekitar tempat tidur. ESP32 membaca data suhu yang diperoleh dari sensor DS18B20, kemudian membandingkannya dengan nilai threshold sebesar 22°C yang telah ditentukan sebagai batas pengambilan keputusan sistem.

Nilai threshold sebesar 22°C dipilih berdasarkan rentang suhu yang direkomendasikan untuk menciptakan lingkungan tidur yang nyaman. Beberapa penelitian mengenai *sleep hygiene* menyebutkan bahwa suhu kamar yang relatif sejuk, yaitu sekitar 18–22°C, dapat membantu meningkatkan kenyamanan saat tidur dan mendukung kualitas tidur yang lebih baik. Oleh karena itu, pada penelitian ini digunakan batas suhu 22°C sebagai acuan untuk mengaktifkan kipas

angin ketika suhu ruangan mulai melebihi kondisi yang dianggap nyaman.

Apabila hasil pembacaan suhu lebih besar dari 22°C, ESP32 akan memberikan perintah kepada relay untuk mengaktifkan kipas angin sehingga sirkulasi udara di sekitar tempat tidur menjadi lebih baik. Sebaliknya, apabila suhu berada pada atau di bawah 22°C, ESP32 akan memerintahkan relay untuk menonaktifkan kipas angin. Berikut adalah hasil implementasi coding untuk menjalankan sensor LDR di Arduino IDE pada Gambar 4.3.



```
setLamp(false);

void initialLDRCheck() {
  int lightLevel = analogRead(LDR_SENSOR); // Baca tingkat cahaya
  Serial.println("Tingkat Cahaya Awal: " + String(lightLevel));

  // Jika tingkat cahaya di bawah ambang batas (100 lux), hidupkan lampu
  if (lightLevel < 100) {
    setLamp(true); // Hidupkan lampu
  } else {
    setLamp(false); // Matikan lampu
  }

  isLDRRead = true; // Tandai pembacaan LDR sudah selesai
}

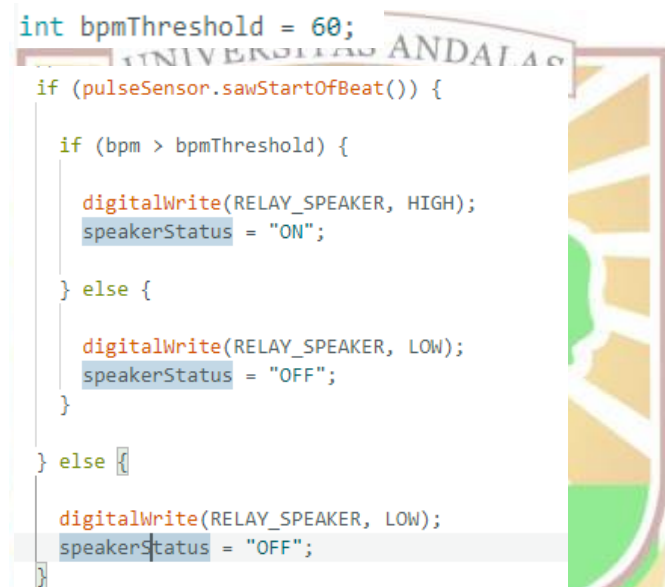
if (isLDRRead) {
  heartRate = analogRead(PULSE_SENSOR);
}
```

Gambar 4.3 Source Code Sensor LDR

Gambar 4.3 merupakan hasil dari coding dari sensor LDR yang berfungsi sebagai pendeteksi intensitas cahaya di sekitar tempat tidur. Pada coding terlihat bahwa sensor LDR hanya akan mendeteksi hanya pada saat awal sistem dihidupkan. Hal ini ditandai dengan kode `initialLDRCheck()` dipanggil hanya di dalam `setup()`. Lampu pada sistem berfungsi sebagai lampu tidur. Pemilihan lampu dilakukan dengan memilih lampu dengan daya yang lebih rendah dengan warna kuning sehingga diharapkan bisa menciptakan suasana tidur yang lebih rileks dan tenang namun tidak terlalu gelap. Lampu akan

otomatis hidup di awal jika sensor LDR mendeteksi cahaya kurang dari 100 lux, namun sebaliknya jika sensor LDR mendeteksi cahaya besar dari 100 lux maka relay tidak akan menghidupkan lampu. Kondisi FALSE merupakan indikasi bahwa lampu dalam keadaan mati pada saat sistem baru diaktifkan.

Berikut adalah implementasi coding untuk menjalankan sensor pulse di Arduino IDE pada gambar 4.4.



```
int bpmThreshold = 60;

while (true) {
  if (pulseSensor.sawStartOfBeat()) {

    if (bpm > bpmThreshold) {

      digitalWrite(RELAY_SPEAKER, HIGH);
      speakerStatus = "ON";

    } else {

      digitalWrite(RELAY_SPEAKER, LOW);
      speakerStatus = "OFF";

    }

  } else {

    digitalWrite(RELAY_SPEAKER, LOW);
    speakerStatus = "OFF";

  }

}
```

Gambar 4.4 Source Code Sensor Pulse

Gambar 4.4 menunjukkan implementasi source code Pulse Sensor yang digunakan untuk mendeteksi denyut jantung pengguna dalam satuan Beats Per Minute (BPM). Data denyut jantung yang diperoleh dari Pulse Sensor dibaca dan diproses oleh ESP32 sebagai salah satu parameter dalam menentukan pengendalian speaker.

Pada program ditetapkan nilai threshold sebesar 60 BPM sebagai batas pengambilan keputusan sistem. Nilai tersebut dipilih berdasarkan rentang denyut jantung normal orang dewasa saat kondisi istirahat, yaitu sekitar 60–100 BPM. Dalam penelitian ini,

nilai 60 BPM digunakan sebagai batas untuk mengidentifikasi apakah pengguna masih memerlukan media relaksasi berupa audio atau tidak. Apabila hasil pembacaan denyut jantung lebih besar dari 60 BPM, ESP32 akan mengirimkan perintah kepada relay untuk mengaktifkan speaker sehingga audio relaksasi dapat diputar. Sebaliknya, apabila nilai denyut jantung berada pada atau di bawah 60 BPM, ESP32 akan memerintahkan relay untuk menonaktifkan speaker. Hal ini didasarkan pada asumsi bahwa ketika denyut jantung telah menurun hingga mencapai kondisi istirahat, pengguna dianggap berada dalam kondisi yang lebih rileks sehingga pemutaran audio relaksasi tidak lagi diperlukan.

2. Bot Telegram

Implementasi Telegram Bot dilakukan sebagai media komunikasi antara pengguna dengan sistem. Telegram Bot digunakan untuk menampilkan informasi hasil pembacaan sensor dan status aktuator, sekaligus memberikan akses kepada pengguna untuk mengendalikan beberapa fungsi sistem melalui aplikasi Telegram. Komunikasi antara ESP32 dan Telegram Bot dilakukan melalui jaringan WiFi sehingga proses pertukaran data dapat berlangsung secara real-time.

Pada implementasinya, Telegram Bot menyediakan beberapa perintah yang dapat digunakan oleh pengguna, yaitu /status, /sleepmode, /relax, dan /report. Perintah /status digunakan untuk menampilkan kondisi sensor dan status aktuator, /sleepmode digunakan untuk mengaktifkan mode tidur dengan konfigurasi aktuator sesuai rancangan sistem, /relax digunakan untuk mengaktifkan audio relaksasi melalui speaker, sedangkan /report digunakan untuk menampilkan ringkasan kondisi lingkungan tidur berdasarkan data yang diperoleh dari sensor.

```

const char* ssid = "Chairunisa";
const char* password = "14112000";

// =====
// TELEGRAM
// =====
#define BOT_TOKEN "8269021560:AAE57nPNmzFTMUXKawThp-dwkIsSee1yCTQ"
#define CHAT_ID "1132931465"

WiFiClientSecure client;
UniversalTelegramBot bot(BOT_TOKEN, client);

void handleNewMessages(int numNewMessages) {

    for (int i = 0; i < numNewMessages; i++) {

        String text = bot.messages[i].text;
        String chat_id = bot.messages[i].chat_id;

        // =====
        // /start
        // =====
        if (text == "/start") {

```



```

String welcome = "";

welcome += "🌙 Smart Sleep Hygiene Assistant\n\n";

welcome += "Selamat datang 🍷\n\n";

welcome += "Sistem ini membantu menciptakan\n";
welcome += "lingkungan tidur yang kondusif\n";
welcome += "secara otomatis.\n\n";

welcome += "Menu:\n";
welcome += "/status\n";
welcome += "/sleepmode\n";
welcome += "/relax\n";
welcome += "/report\n";

bot.sendMessage(chat_id, welcome, "");
}

if (text == "/status") {

    String message = "";

    message += "📊 ROOM MONITORING\n\n";

    message += "🌡 Temperature : ";
    message += String(temperatureC);
    message += " °C\n";

    message += "💡 Lamp : ";
    message += lampStatus;
    message += "\n";

    message += "🌀 Fan : ";
    message += fanStatus;
    message += "\n";
}

```

```

message += "🎵 Speaker : ";
message += speakerStatus;
message += "\n";

message += "❤️ BPM : ";
message += String(bpm);
message += "\n\n";

message += "🟢 Sleep Environment Stable";

bot.sendMessage(chat_id, message, "");
}

if (text == "/sleepmode") {

    digitalWrite(RELAY_LAMP, LOW);
    digitalWrite(RELAY_SPEAKER, HIGH);

    lampStatus = "OFF";
    speakerStatus = "ON";

    String message = "";

    message += "🌙 Sleep Mode Activated\n\n";

    message += "💡 Lamp dimatikan\n";
    message += "🎵 Audio relaksasi diaktifkan\n";
    message += "🛏️ Lingkungan tidur disesuaikan\n\n";

    message += "Selamat beristirahat 😴";

    bot.sendMessage(chat_id, message, "");
}

```

```

if (text == "/relax") {

    digitalWrite(RELAY_SPEAKER, HIGH);

    speakerStatus = "ON";

    String message = "";

    message += "🎵 Relaxation Mode Activated\n\n";
    message += "Audio relaksasi sedang diputar 🎧";

    bot.sendMessage(chat_id, message, "");
}

if (text == "/report") {

    String message = "";

    message += "☑ Sleep Environment Report\n\n";

    message += "🌡 Temperature : ";
    message += String(temperatureC);
    message += " °C\n";

    message += "❤ BPM : ";
    message += String(bpm);
    message += "\n";

    message += "💡 Light Intensity : ";
    message += String(ldrValue);
    message += "\n\n";
}

```

Gambar 4. 5 Source Code Telegram

Gambar 4.5 menunjukkan implementasi source code Telegram Bot yang digunakan sebagai media komunikasi antara pengguna dan sistem *Sleep Hygiene*. Pada tahap implementasi, dilakukan konfigurasi jaringan WiFi dengan memasukkan SSID dan password agar ESP32 dapat terhubung ke internet. Selanjutnya, program juga

dikonfigurasi menggunakan Bot Token yang diperoleh dari BotFather sehingga ESP32 dapat berkomunikasi dengan Telegram Bot.

Setelah ESP32 berhasil terhubung ke jaringan WiFi dan Telegram Bot, sistem akan mengirimkan notifikasi "*Smart Sleep Hygiene Assistant Aktif!*" sebagai tanda bahwa seluruh proses inisialisasi telah berhasil dilakukan dan sistem siap digunakan.

Telegram Bot menyediakan beberapa perintah yang dapat digunakan oleh pengguna untuk memantau maupun mengendalikan sistem. Perintah `/status` digunakan untuk menampilkan informasi kondisi sistem secara real-time, meliputi hasil pembacaan suhu dari sensor DS18B20, nilai denyut jantung dari Pulse Sensor, nilai ADC dari sensor LDR, serta status aktuator berupa kipas angin, lampu LED warm light, dan speaker.

Perintah `/sleepmode` digunakan untuk mengaktifkan mode tidur secara manual. Ketika perintah ini dijalankan, sistem akan mengaktifkan konfigurasi tidur sesuai dengan algoritma yang telah dirancang, yaitu menyalakan lampu LED warm light untuk menciptakan suasana yang nyaman, mengaktifkan speaker untuk memutar audio relaksasi, serta tetap mengendalikan kipas angin berdasarkan kondisi suhu ruangan.

Selanjutnya, perintah `/relax` digunakan untuk mengaktifkan audio relaksasi melalui speaker tanpa mengubah kondisi aktuator lainnya. Fitur ini ditujukan bagi pengguna yang ingin mendengarkan musik relaksasi sebelum tidur atau saat beristirahat.

Selain itu, tersedia pula perintah `/report` yang berfungsi untuk menampilkan ringkasan kondisi sistem berdasarkan hasil pembacaan sensor dan status aktuator. Melalui fitur ini, pengguna dapat memantau kondisi lingkungan tidur secara lebih mudah melalui

aplikasi Telegram. Implementasi antarmuka Telegram Bot dapat dilihat pada Gambar 4.6.



Gambar 4.6 Implementasi Antarmuka Bot Telegram

Gambar 4.6 menunjukkan implementasi antarmuka Telegram Bot sebagai media monitoring dan pengendalian sistem *Sleep Hygiene*. Setelah ESP32 berhasil terhubung ke jaringan WiFi dan Telegram Bot, sistem akan mengirimkan notifikasi "Smart Sleep Hygiene Assistant Aktif!" sebagai tanda bahwa seluruh komponen telah berhasil diinisialisasi dan sistem siap digunakan. Melalui antarmuka ini, pengguna dapat berinteraksi dengan sistem menggunakan beberapa perintah (command), yaitu /status, /sleepmode, /relax, dan /report.

4.1.3 Implementasi Sistem Secara Menyeluruh

Implementasi sistem menyeluruh merupakan tahap integrasi antara perangkat keras dan perangkat lunak yang telah dirancang sehingga membentuk sebuah sistem yang dapat bekerja secara otomatis. Pada tahap ini seluruh komponen, meliputi ESP32, sensor DS18B20, sensor LDR,

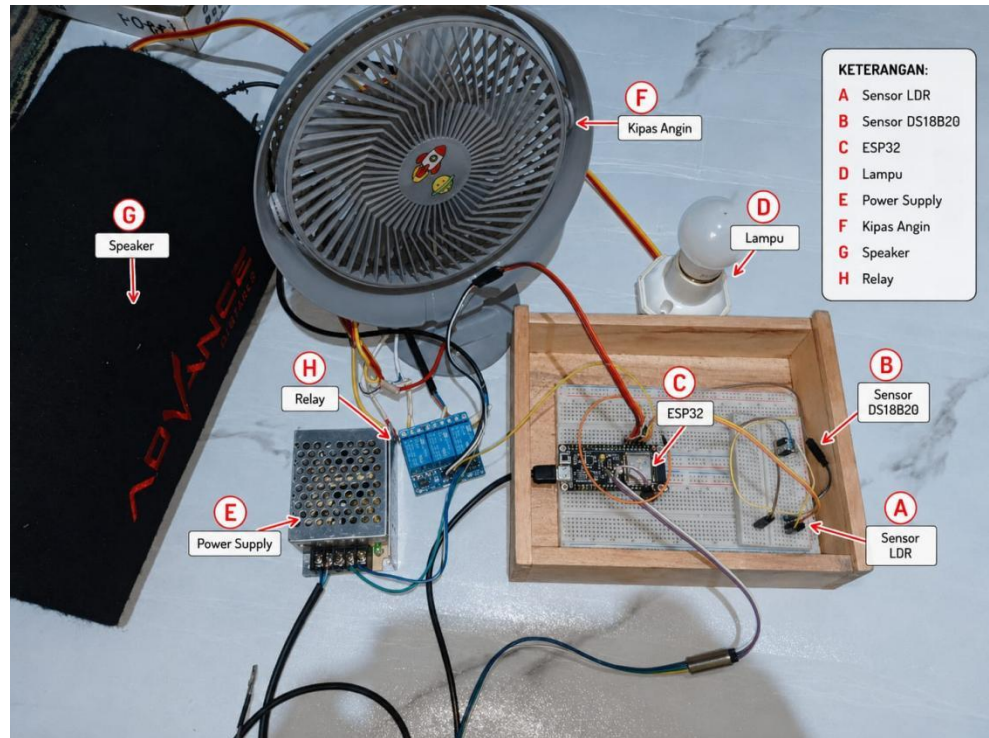
Pulse Sensor, relay, kipas angin, lampu LED warm light, speaker, serta Telegram Bot dihubungkan menjadi satu kesatuan sistem.

ESP32 berperan sebagai pusat pengendali yang menerima data masukan dari ketiga sensor. Sensor DS18B20 digunakan untuk membaca suhu di sekitar tempat tidur, sensor LDR digunakan untuk mendeteksi kondisi pencahayaan ruangan, sedangkan Pulse Sensor digunakan untuk membaca denyut jantung pengguna. Data yang diperoleh dari masing-masing sensor selanjutnya diproses oleh ESP32 dan dibandingkan dengan nilai threshold yang telah ditentukan pada program.

Berdasarkan hasil pengolahan data tersebut, ESP32 akan mengendalikan relay untuk mengaktifkan maupun menonaktifkan aktuator sesuai dengan kondisi yang terdeteksi. Apabila suhu ruangan melebihi 22°C, relay akan mengaktifkan kipas angin untuk meningkatkan sirkulasi udara di sekitar tempat tidur. Selanjutnya, apabila nilai ADC dari sensor LDR melebihi nilai threshold 100 lux, relay akan mengaktifkan lampu LED warm light sebagai pencahayaan yang mendukung suasana tidur yang nyaman. Sementara itu, apabila denyut jantung pengguna berada di atas 60 BPM, relay akan mengaktifkan speaker untuk memutar audio relaksasi sebagai media terapi pendukung tidur.

Selain pengendalian secara otomatis berdasarkan pembacaan sensor, sistem juga dilengkapi dengan fitur monitoring dan pengendalian melalui aplikasi Telegram. Melalui Telegram Bot, pengguna dapat mengetahui kondisi sistem secara real-time menggunakan perintah /status, mengaktifkan Sleep Mode menggunakan perintah /sleepmode, memutar audio relaksasi menggunakan perintah /relax, serta melihat laporan kondisi lingkungan tidur melalui perintah /report. Dengan adanya fitur tersebut, pengguna dapat memantau kondisi lingkungan tidur sekaligus berinteraksi dengan sistem tanpa harus mengoperasikan perangkat secara

langsung. Implementasi sistem secara keseluruhan akan ditunjukkan pada Gambar 4.7.



Gambar 4.7 Implementasi Sistem

Gambar 4.7 menunjukkan hasil implementasi keseluruhan sistem. Seluruh komponen telah dirangkai menjadi satu kesatuan yang terdiri atas ESP32 sebagai mikrokontroler, Pulse Sensor, sensor DS18B20, sensor LDR, tiga buah relay sebagai saklar otomatis, kipas angin, lampu LED *warm light*, speaker, serta catu daya (*power supply*) sebagai sumber tegangan untuk relay dan aktuator. Seluruh komponen tersebut bekerja secara terintegrasi sehingga sistem mampu melakukan monitoring kondisi lingkungan tidur, mengendalikan aktuator secara otomatis, serta mengirimkan informasi kepada pengguna melalui Telegram Bot.

4.2 Pengujian dan Analisa

Pengujian dan analisa sistem dilakukan untuk mengetahui apakah seluruh komponen perangkat keras maupun perangkat lunak telah bekerja sesuai dengan rancangan yang telah dibuat. Selain itu, tahap ini bertujuan untuk mengevaluasi kinerja sistem dalam membaca data dari sensor, mengendalikan aktuator, serta menjalankan fungsi monitoring dan pengendalian melalui Telegram Bot. Hasil pengujian selanjutnya dianalisis untuk mengetahui tingkat keberhasilan sistem dalam mencapai tujuan penelitian, yaitu membantu menciptakan lingkungan tidur yang lebih nyaman melalui penerapan konsep sleep hygiene. Pengujian pada penelitian ini terdiri atas pengujian perangkat keras, pengujian perangkat lunak, dan pengujian fungsional sistem.

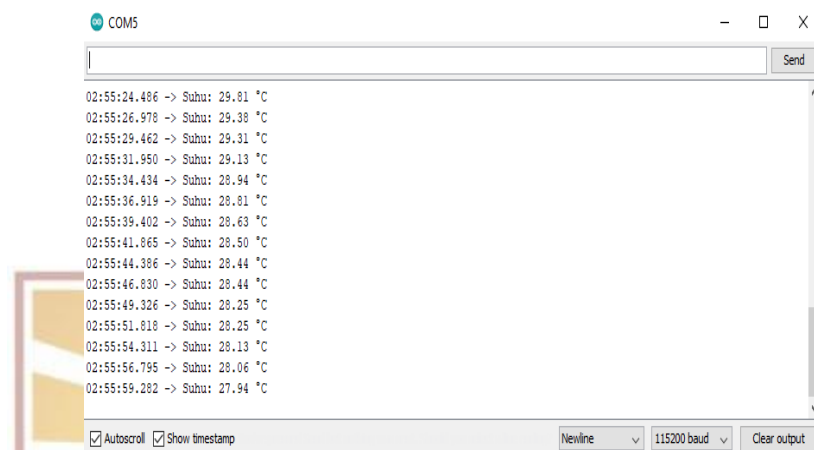
4.2.1 Pengujian dan Analisa Perangkat Keras

Pengujian perangkat keras dilakukan untuk memastikan bahwa setiap komponen yang digunakan pada sistem dapat bekerja sesuai dengan fungsi yang telah dirancang. Pengujian dilakukan secara terpisah pada setiap perangkat, meliputi Pulse Sensor, sensor DS18B20, sensor LDR, ESP32, dan relay. Melalui pengujian ini dapat diketahui kemampuan masing-masing komponen dalam membaca data, memproses sinyal, maupun mengendalikan aktuator sebelum seluruh komponen diintegrasikan menjadi satu sistem.

4.2.1.1 Pengujian dan Analisa Sensor DS18B20

Pengujian dan analisa sensor DS18B20 dilakukan menggunakan termometer digital sebagai alat pembanding. Pengujian ini bertujuan untuk mengetahui tingkat keakuratan pembacaan suhu yang dihasilkan oleh sensor DS18B20 dengan membandingkan hasil pengukurannya terhadap nilai suhu yang ditampilkan oleh termometer digital. Pengujian dilakukan pada beberapa kondisi suhu ruangan, kemudian hasil pembacaan dari kedua perangkat dicatat dan

dibandingkan untuk mengetahui selisih pengukuran serta tingkat akurasi sensor DS18B20. Hasil pembacaan sensor DS18B20 melalui Serial Monitor Arduino IDE dapat dilihat pada Gambar 4.8.



Gambar 4.8 Pembacaan Sensor DS18B20 di Serial Monitor

Selanjutnya, hasil pembacaan sensor DS18B20 dibandingkan dengan hasil pengukuran termometer digital untuk mengetahui nilai selisih (error) pada setiap pengujian. Hasil pengujian sensor DS18B20 ditunjukkan pada Tabel 4.1.

Tabel 4.1 Pengujian dan Analisa Sensor DS18B20

No	Sensor DS18B20 (°C)	Termometer Digital (°C)	Selisih	Persentase Error (%)
1.	25.75	26.00	0.25	0.96
2.	26.81	27.2	0.39	1.43
3.	27.06	27.9	0.84	3.01
4.	27.62	27.9	0.28	1.03
5.	28.00	28.3	0.3	1.06
6.	28.37	28.6	0.23	0.80
7.	28.56	28.8	0.24	0.83
8.	28.94	29.0	0.06	0.2

9.	29.12	30.1	0.98	3.25
10.	29.81	31.1	1.29	4.14
Jumlah			4.86	16.683
Rata-Rata Selisih			0.49	
Rata-Rata Error			1.67	

Berdasarkan hasil pengujian pada Tabel 4.1, sensor DS18B20 mampu mengukur suhu dengan nilai yang mendekati hasil pengukuran termometer digital. Pengujian sensor DS18B20 dilakukan sebanyak 10 kali untuk mengetahui konsistensi hasil pembacaan suhu pada kondisi lingkungan yang sama. Pengulangan pengukuran diperlukan untuk melihat apakah nilai yang dihasilkan sensor relatif stabil dan tidak hanya bergantung pada satu kali pembacaan. Selanjutnya, hasil dari 10 kali pengukuran digunakan untuk memperoleh nilai rata-rata sebagai representasi hasil pembacaan sensor pada kondisi pengujian. Dari 10 percobaan yang dilakukan untuk mengukur suhu, didapat hasil rata-rata selisih pengukuran sebesar 0,49 dan rata-rata erornya adalah 1,67%. Selisih pengukuran yang diperoleh relatif kecil sehingga menunjukkan bahwa sensor DS18B20 memiliki tingkat akurasi yang baik untuk digunakan dalam memantau suhu di sekitar tempat tidur. Perbedaan hasil pengukuran yang muncul pada beberapa pengujian dapat disebabkan oleh toleransi sensor, waktu respons sensor terhadap perubahan suhu, serta perbedaan posisi antara sensor DS18B20 dan termometer digital saat proses pengukuran dilakukan. Berdasarkan hasil tersebut, sensor DS18B20 dinilai mampu memberikan data suhu yang cukup akurat sebagai masukan bagi ESP32 dalam mengendalikan kipas angin secara otomatis. Untuk

mencari hasil rata-rata selisih pengukuran dan rata-rata error dapat diperoleh dengan persamaan berikut :

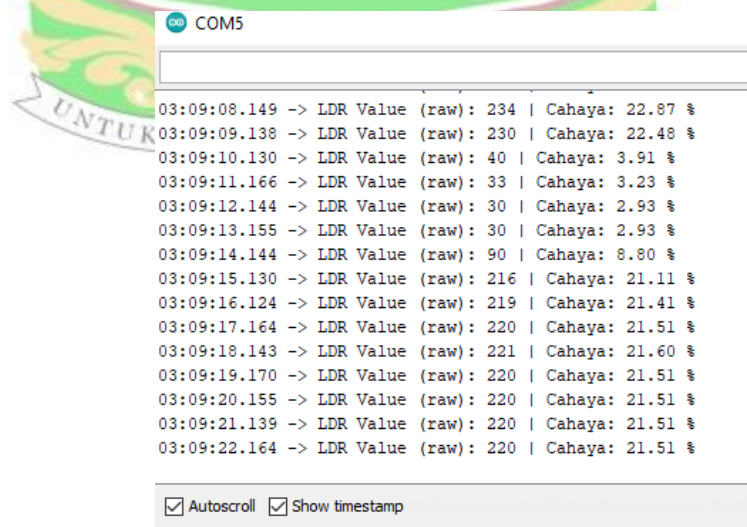
$$\text{Rata - Rata Selisih} = \frac{\text{Zselisih Pengukuran}}{\text{banyak data}} \dots\dots\dots(4.1)$$

$$\%Error = \frac{|\text{nilai sensor} - \text{nilai alat ukur}|}{\text{nilai alat ukur}} \times 100\% \dots\dots\dots(4.2)$$

$$\text{rata - rata error} = \frac{\text{Z\%Error}}{\text{banyak error}} \dots\dots\dots(4.3)$$

4.2.1.2 Pengujian dan Analisa Sensor LDR

Pengujian dan analisa sensor LDR dilakukan menggunakan lux meter sebagai alat pembanding. Pengujian ini bertujuan untuk mengetahui tingkat keakuratan pembacaan suhu yang dihasilkan oleh sensor LDR dengan membandingkan hasil pengukurannya terhadap nilai suhu yang ditampilkan oleh lux meter. Pengujian dilakukan pada beberapa kondisi cahaya ruangan, kemudian hasil pembacaan dari kedua perangkat dicatat dan dibandingkan untuk mengetahui selisih pengukuran serta tingkat akurasi sensor LDR. Berikut adalah hasil dari pembacaan sensor LDR pada serial monitor yang dapat dilihat pada Gambar 4.9:



Gambar 4.9 Pembacaan Sensor LDR pada Serial Monitor

Selanjutnya, hasil pembacaan sensor LDR dibandingkan dengan hasil pengukuran lux meter untuk mengetahui nilai selisih (error) pada setiap pengujian. Hasil dari pengujian sensor LDR dapat dilihat pada tabel 4.2 berikut :

Tabel 4.2 Pengujian dan Analisa Sensor LDR

No.	Sensor LDR (lux)	Sensor Smartphone (lux)	Selisih	Persentase Eror
1.	18	19	1	0.5%
2.	21	23	2	0.8%
3.	27	28	1	0.3%
4.	39	42	3	0.7%
5.	72	73	1	0.1%
6.	82	83	1	0.1%
7.	84	87	3	0.3%
8.	90	91	1	0.1%
9.	119	112	7	0.6%
10.	115	122	3	0.2%
Jumlah			30	3.7%
Rata-Rata Selisih			2.3	
Rata-Rata Eror			0.37%	

Pada pengujian sensor LDR digunakan sensor GL5516 yang dirangkai dengan resistor tetap sebesar 4,7 k Ω sebagai pembagi tegangan. Pembacaan sensor oleh ESP32 menghasilkan nilai ADC yang merepresentasikan perubahan tegangan akibat perubahan intensitas cahaya yang diterima oleh permukaan LDR. Agar hasil pembacaan dapat digunakan untuk menggambarkan intensitas

pencahayaannya dalam satuan Lux, dilakukan proses konversi nilai ADC menjadi tegangan dan resistansi LDR.

Rumus konversi LDR (Light Dependent Resistor) dari nilai bacaan ADC (mikrokontroler) menjadi besar tegangan (V_{out}), hambatan LDR (R_{LDR}), hingga intensitas cahaya dalam satuan lux (Lux) menggunakan rangkaian pembagi tegangan. Untuk mencari nilai tegangan digunakan persamaan pada rumus 4.4 berikut:

$$V_{out} = V_{out} = ADC \times \left(\frac{V_{in}}{Max\ ADC} \right) \dots \dots \dots (4.4)$$

Untuk mencari nilai hambatannya menggunakan rumus pada persamaan 4.5.

$$R_{LDR} = R1 \times \left(\frac{V_{in} - V_{out}}{V_{out}} \right) \dots \dots \dots (4.5)$$

Untuk mengkonversi ke lux menggunakan rumus pada persamaan 4.6 berikut :

$$Lux = \left(\frac{R10 \times 10^9}{R_{LDR}} \right) Lux = \left(\frac{R10 \times 10^9}{R_{LDR}} \right)^{\frac{1}{\gamma}} \dots \dots \dots (4.6)$$

Keterangan:

1. ADC = nilai mentah dari pembacaan pin analog mikrokontroler
2. V_{in} = tegangan masukan sumber (misal: 5v atau 3.3v)
3. R1 = Nilai resistor tetap yang dipasang seri dengan LDR
4. Max ADC = Nilai maksimum bit ADC

Konversi ini dilakukan agar hasil pengujian sensor LDR dapat dinyatakan dalam satuan intensitas cahaya yang lebih mudah diinterpretasikan dibandingkan hanya menggunakan nilai ADC. Nilai Lux tersebut kemudian digunakan untuk mengevaluasi perubahan

kondisi pencahayaan dan respons sistem dalam mengendalikan lampu.

Berdasarkan hasil pengujian pada Tabel 4.2, diujikan sensor LDR untuk mendeteksi cahaya tempat tidur dengan menggunakan sensor smartphone di tempat yang sama. Untuk mencari hasil rata-rata selisih pengukuran dan rata-rata error dapat diperoleh dengan persamaan berikut :

$$\text{Rata - Rata Selisih} = \frac{\text{Zselisih Pengukuran}}{\text{banyak data}} \dots\dots\dots(4.1)$$

$$\%Error = \frac{|\text{nilai sensor} - \text{nilai alat ukur}|}{\text{nilai alat ukur}} \times 100\% \dots\dots\dots(4.2)$$

$$\text{rata - rata error} = \frac{\text{Z\%Error}}{\text{banyak error}} \dots\dots\dots(4.3)$$

Pengujian sensor LDR dilakukan sebanyak 10 kali pada kondisi pencahayaan yang telah ditentukan. Pengulangan sebanyak 10 kali dilakukan untuk mengetahui konsistensi pembacaan sensor terhadap kondisi pencahayaan yang sama serta mengurangi kemungkinan hasil pengujian dipengaruhi oleh variasi pembacaan sensor pada satu kali pengukuran. Hasil dari 10 kali pengujian kemudian digunakan untuk memperoleh nilai rata-rata sebagai gambaran pembacaan sensor pada kondisi tersebut. Dari 10 percobaan yang dilakukan untuk mengukur suhu, didapat hasil rata-rata selisih pengukuran sebesar 2,3 dan rata-rata erornya adalah 0,37%.

4.2.1.3 Pengujian dan Analisa Sensor Pulse

Pengujian dan analisis Pulse Sensor dilakukan untuk mengetahui kemampuan sensor dalam mendeteksi denyut jantung pengguna serta mengevaluasi tingkat keakuratan hasil pengukurannya. Pengujian dilakukan dengan membandingkan nilai denyut jantung (*Beats Per*

Minute/BPM) yang dibaca oleh Pulse Sensor dengan hasil pengukuran menggunakan oximeter sebagai alat pembanding. Oximeter dipilih karena merupakan alat yang umum digunakan untuk mengukur denyut jantung secara non-invasif dan memiliki tingkat akurasi yang baik sehingga dapat dijadikan sebagai acuan dalam proses pengujian. Selama pengujian, Pulse Sensor dan oximeter digunakan secara bersamaan pada kondisi pengguna yang sama, kemudian hasil pembacaan dari kedua perangkat dibandingkan untuk mengetahui selisih pengukuran dan tingkat akurasi *Pulse Sensor*. Hasil pembacaan *Pulse Sensor* pada serial monitor dapat dilihat pada Gambar 4.11.



Gambar 4.10 Pembacaan Sensor Pulse pada Serial Monitor

Untuk mengetahui tingkat akurasi Pulse Sensor, dilakukan pengujian dengan membandingkan hasil pembacaan denyut jantung antara Pulse Sensor dan oximeter sebanyak 10 kali percobaan. Setiap pengujian dilakukan pada waktu yang berbeda untuk memperoleh variasi data pengukuran sehingga dapat diketahui konsistensi pembacaan Pulse Sensor terhadap alat pembanding. Hasil pengujian dan perbandingan pembacaan kedua perangkat dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengujian dan Analisa Sensor Pulse

No.	Sensor Pulse (BPM)	Oximeter (BPM)	Selisih	Persentase Error (%)
1.	86	87	1	1.1
2.	77	79	2	2.5
3.	93	92	1	1.08
4.	95	98	3	3.06
5.	100	99	1	1.01
6.	82	82	0	0
7.	78	79	1	1.2
8.	75	75	0	0
9.	77	78	1	1.2
10.	85	86	1	1.1
Jumlah			11	12.25
Rata-Rata Selisih				1.1
Rata-Rata Error				1.3

Berdasarkan Tabel 4.3, pengujian dilakukan dengan membandingkan hasil pembacaan denyut jantung antara Pulse Sensor dan oximeter yang digunakan secara bersamaan pada jari tangan yang sama. Untuk mencari hasil rata-rata selisih pengukuran dan rata-rata error dapat diperoleh dengan persamaan berikut :

$$\text{Rata - Rata Selisih} = \frac{\text{Zselisih Pengukuran}}{\text{banyak data}} \dots\dots\dots(4.1)$$

$$\%Error = \frac{|\text{nilai sensor}-\text{nilai alat ukur}|}{\text{nilai alat ukur}} \times 100\% \dots\dots\dots(4.2)$$

$$\text{rata - rata error} = \frac{\text{Z\%Error}}{\text{banyak error}} \dots\dots\dots(4.3)$$

Pengujian Pulse Sensor dilakukan sebanyak 10 kali untuk mengetahui konsistensi sistem dalam membaca denyut jantung pengguna dan

menghasilkan nilai BPM. Pengulangan dilakukan karena nilai denyut jantung dapat mengalami perubahan selama proses pengukuran. Dengan melakukan beberapa kali pengukuran, hasil pembacaan dapat diamati secara lebih representatif dan digunakan untuk memperoleh nilai rata-rata BPM pada kondisi pengujian. Dari 10 kali percobaan yang dilakukan, diperoleh rata-rata selisih pengukuran sebesar 1,1 BPM dengan rata-rata persentase kesalahan (*error*) sebesar 1,3%. Hasil tersebut menunjukkan bahwa Pulse Sensor memiliki tingkat akurasi yang baik dalam mendeteksi denyut jantung pengguna, sehingga dapat digunakan sebagai masukan bagi sistem untuk menentukan kondisi relaksasi pengguna dan mengendalikan speaker terapi secara otomatis. Dengan rata-rata error sebesar 1,3%, hasil pembacaan Pulse Sensor masih berada dalam batas yang dapat diterima sehingga tidak memengaruhi proses pengambilan keputusan sistem dalam mengaktifkan maupun menonaktifkan speaker berdasarkan nilai ambang denyut jantung yang telah ditentukan.

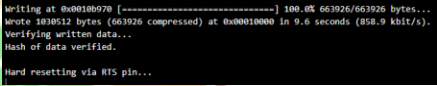
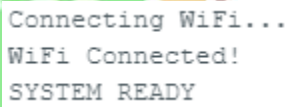
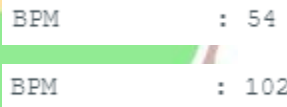
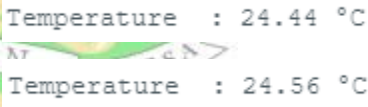
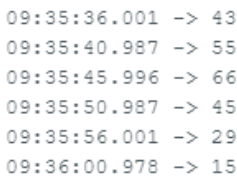
4.2.1.4 Pengujian dan Analisa ESP32

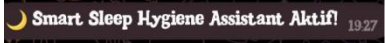
Pengujian dan analisa ESP32 dilakukan untuk mengetahui kemampuan mikrokontroler dalam menjalankan seluruh fungsi sistem, mulai dari membaca data yang diperoleh dari Pulse Sensor, sensor DS18B20, dan sensor LDR, hingga mengendalikan relay sebagai pengontrol aktuator berupa kipas angin, lampu, dan speaker. Selain itu, pengujian juga dilakukan untuk memastikan ESP32 dapat terhubung dengan jaringan Wi-Fi sehingga mampu berkomunikasi dengan Bot Telegram sebagai media monitoring dan pengendalian sistem.

Pada tahap pengujian, program diunggah ke ESP32 menggunakan Arduino IDE. Selanjutnya ESP32 dihubungkan dengan jaringan Wi-

Fi dan diamati melalui Serial Monitor untuk memastikan proses inisialisasi sistem berjalan dengan baik. Pengujian dilakukan dengan mengamati keberhasilan ESP32 dalam membaca data dari masing-masing sensor, mengendalikan relay sesuai logika program, serta mengirim dan menerima data melalui Bot Telegram. Hasil pengujian ESP32 dapat dilihat pada Tabel 4.4

Tabel 4.4 Pengujian dan Analisa ESP32

No.	Parameter Pengujian	Hasil
1.	Program berhasil diunggah ke ESP32	Berhasil 
2.	ESP32 dapat terhubung ke jaringan WiFi	Berhasil 
3.	ESP32 dapat membaca data <i>pulse sensor</i>	Berhasil 
4.	ESP32 dapat membaca data dari sensor DS18B20	Berhasil 
5.	ESP32 dapat membaca data dari sensor LDR	Berhasil 

No.	Parameter Pengujian	Hasil
6.	ESP32 dapat mengendalikan relay	Berhasil Lamp : ON Fan : ON Speaker : OFF
7.	ESP32 dapat mengirim data ke Telegram	Berhasil 
8.	ESP32 dapat menerima perintah Telegram	Berhasil 

Berdasarkan hasil pengujian, ESP32 berhasil membaca data dari seluruh sensor dan mengirimkan sinyal logika HIGH maupun LOW melalui pin GPIO untuk mengendalikan modul relay. Perubahan logika tersebut menyebabkan relay dapat mengaktifkan maupun menonaktifkan kipas angin, lampu, dan speaker sesuai dengan kondisi yang telah diprogram. Selain itu, ESP32 juga berhasil terhubung dengan jaringan Wi-Fi dan Bot Telegram sehingga proses monitoring dan pengendalian sistem dapat dilakukan secara *real-time*.

4.2.1.5 Pengujian dan Analisa Relay

Pengujian dan analisa relay dilakukan untuk mengetahui kemampuan relay dalam menghubungkan dan memutus arus listrik pada aktuator yang digunakan dalam sistem, yaitu kipas angin, lampu, dan speaker. Pengujian dilakukan dengan memberikan sinyal logika HIGH dan LOW dari ESP32 ke masing-masing kanal

relay, kemudian mengamati kondisi aktuator apakah dapat menyala dan mati sesuai dengan perintah yang diberikan oleh mikrokontroler.

Selain itu, pengujian juga bertujuan untuk memastikan bahwa relay mampu bekerja dengan baik sebagai saklar elektronik tanpa terjadi keterlambatan (*delay*) maupun kegagalan saat proses pengaktifan dan penonaktifan aktuator. Hasil pengujian relay dapat dilihat pada Tabel 4.5.

Tabel 4.5 Pengujian dan Analisa Relay

No.	Aktuator	Kondisi Relay	Kondisi Aktuator	Hasil
1.	Kipas Angin	On	Kipas menyala	Berhasil
2.	Kipas Angin	Off	Kipas mati	Berhasil
3.	Lampu	On	Lampu menyala	Berhasil
4.	Lampu	Off	Lampu mati	Berhasil
5.	Speaker	On	Speaker menyala	Berhasil
6.	Speaker	Off	Speaker mati	Berhasil

Berdasarkan hasil pengujian pada Tabel 4.5, seluruh kanal relay mampu bekerja sesuai dengan logika kendali yang diberikan oleh ESP32. Ketika relay menerima sinyal logika HIGH, relay akan menghubungkan arus listrik sehingga aktuator berupa kipas angin, lampu, dan speaker dapat menyala. Sebaliknya, ketika relay

menerima sinyal logika LOW, relay akan memutus arus listrik sehingga seluruh aktuator dapat berhenti bekerja. Hasil tersebut menunjukkan bahwa relay mampu menjalankan fungsinya sebagai saklar otomatis pada sistem *Sleep Hygiene* dengan baik.

4.2.2 Pengujian dan Analisa Perangkat Lunak

Pengujian dan analisis perangkat lunak dilakukan untuk memastikan bahwa seluruh perangkat lunak yang digunakan pada sistem dapat berfungsi sesuai dengan perancangan. Pengujian ini terdiri atas dua bagian, yaitu pengujian program pada Arduino IDE dan pengujian Bot Telegram. Pengujian program Arduino IDE bertujuan untuk mengetahui keberhasilan program dalam menjalankan logika sistem, membaca data sensor, serta mengendalikan aktuator. Sementara itu, pengujian Bot Telegram dilakukan untuk memastikan proses komunikasi antara ESP32 dan Telegram dapat berjalan dengan baik, sehingga pengguna dapat menerima informasi sistem dan memberikan perintah kepada perangkat. Selain itu, pengujian ini juga bertujuan untuk menganalisis waktu respons yang dibutuhkan sistem dalam menjalankan program dan mengirimkan data secara *real-time*.

4.2.2.1 Pengujian dan Analisa Arduino IDE

Pengujian dan analisis program pada Arduino IDE dilakukan untuk mengetahui apakah program yang telah dikembangkan dapat dijalankan pada ESP32 dengan penggunaan sumber daya memori yang masih berada dalam batas kemampuan mikrokontroler. Salah satu parameter yang dianalisis pada pengujian ini adalah penggunaan memori program (*program storage*) dan memori dinamis (*dynamic memory*) yang ditampilkan oleh Arduino IDE

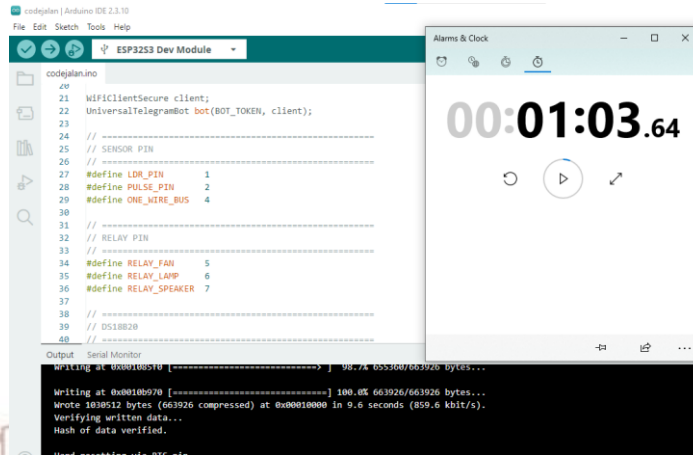
setelah proses kompilasi (*compile*). Pengujian ini bertujuan untuk memastikan bahwa ukuran program tidak melebihi kapasitas memori ESP32 sehingga sistem dapat berjalan secara optimal tanpa mengalami kegagalan saat proses unggah maupun eksekusi program. Hasil pengujian penggunaan memory dapat dilihat pada Gambar 4.11.

```
Sketch uses 1030368 bytes (78%) of program storage space. Maximum is 1310720 bytes.  
Global variables use 46964 bytes (14%) of dynamic memory, leaving 280716 bytes for local variables. Maximum is 327680 bytes.  
esptool v5.2.0  
Serial port COM5:  
Connecting...
```

Gambar 4.11 Pengujian Penggunaan Memory

Berdasarkan hasil kompilasi pada Gambar 4.11, program menggunakan memori penyimpanan (*program storage*) sebesar 1.030.368 byte atau 78% dari kapasitas memori flash ESP32. Sementara itu, penggunaan memori dinamis (*dynamic memory*) untuk variabel global sebesar 46.964 byte atau 14% dari kapasitas RAM yang tersedia. Hasil tersebut menunjukkan bahwa penggunaan memori masih berada di bawah kapasitas maksimum ESP32, sehingga program dapat dijalankan dengan baik tanpa mengalami kekurangan memori (*memory overflow*). Dengan demikian, ESP32 masih memiliki sisa ruang memori yang dapat digunakan apabila di kemudian hari dilakukan penambahan fitur atau pengembangan sistem.

Selain analisis penggunaan memori, pengujian pada Arduino IDE juga dilakukan dengan mengukur waktu yang dibutuhkan untuk mengunggah (*upload*) program ke mikrokontroler ESP32. Pengujian ini bertujuan untuk mengetahui waktu yang diperlukan hingga proses kompilasi selesai dan program berhasil ditanamkan ke dalam memori ESP32. Hasil pengujian waktu unggah program dapat dilihat pada Gambar 4.12.



Gambar 4.12 Waktu Unggah Program

Berdasarkan hasil pengujian pada Gambar 4.12, proses unggah program ke ESP32 memerlukan waktu selama 1 menit 3 detik. Selama proses tersebut, Arduino IDE melakukan kompilasi program, kemudian mengunggah hasil kompilasi ke memori ESP32 hingga program siap dijalankan. Proses unggah berlangsung dengan baik tanpa ditemukan kesalahan (*error*), sehingga menunjukkan bahwa program berhasil ditanamkan ke dalam mikrokontroler. Waktu yang dibutuhkan untuk proses unggah dipengaruhi oleh ukuran program yang digunakan, jumlah pustaka (*library*) yang diintegrasikan, serta spesifikasi perangkat komputer yang digunakan selama proses pemrograman. Dengan demikian, dapat disimpulkan bahwa program berhasil diunggah ke ESP32 dan siap digunakan untuk menjalankan seluruh fungsi pada sistem.

Waktu unggah program selama 1 menit 3 detik dipengaruhi oleh ukuran program yang cukup besar karena menggabungkan beberapa fungsi dalam satu sistem, seperti pembacaan data sensor, pengendalian relay, komunikasi Wi-Fi, serta integrasi dengan Bot Telegram. Meskipun proses unggah memerlukan waktu lebih dari satu menit, program berhasil diunggah tanpa kendala dan dapat

dijalankan dengan baik pada ESP32. Waktu unggah ini hanya terjadi pada saat proses pemrograman atau ketika dilakukan pembaruan program, sehingga tidak memengaruhi kinerja sistem saat digunakan oleh pengguna.

4.2.2.2 Pengujian dan Analisa Aplikasi BOT Telegram

Pengujian dan analisis Bot Telegram dilakukan untuk mengetahui apakah komunikasi antara ESP32 dan Bot Telegram dapat berjalan dengan baik. Pengujian ini bertujuan untuk memastikan bahwa bot mampu menerima perintah yang dikirimkan oleh pengguna serta mengirimkan informasi mengenai kondisi sistem secara *real-time*. Pada tahap ini, beberapa perintah yang telah diprogram, seperti `/status`, `/sleepmode`, `/report`, dan `/relax`, diuji satu per satu untuk melihat apakah sistem dapat memberikan respons sesuai dengan fungsi yang telah dirancang. Hasil pengujian Bot Telegram dapat dilihat pada Tabel 4.6.

Tabel 4.6 Pengujian Waktu Respon Bot Telegram

No.	Fungsi	Banyak Pengujian	Rata-Rata Waktu Respon	Kekuatan Sinyal
1.	<code>/status</code>	5	2 detik	
2.	<code>/sleepmode</code>	5	2 detik	

No.	Fungsi	Banyak Pengujian	Rata-Rata Waktu Respon	Kekuatan Sinyal
				
3.	/relax	5	1 detik	
4.	/report	5	2 detik	
Rata-rata waktu respon keseluruhan				1.75 detik

Berdasarkan Tabel 4.6 menunjukkan bahwa sistem mampu memberikan balasan dalam waktu yang relatif singkat setelah perintah dikirimkan oleh pengguna. Waktu respons yang diperoleh dipengaruhi oleh kualitas koneksi internet pada ESP32 maupun

perangkat pengguna. Semakin baik kualitas jaringan yang digunakan, maka semakin cepat pula respons yang diberikan oleh sistem.

Selain itu, hasil pengujian juga menunjukkan bahwa nilai kekuatan sinyal Wi-Fi (RSSI) selama sistem beroperasi masih berada pada kategori yang baik sehingga komunikasi data antara ESP32 dan Bot Telegram dapat berlangsung secara stabil. Dengan koneksi yang stabil, proses monitoring maupun pengendalian sistem melalui Telegram dapat dilakukan tanpa mengalami gangguan yang berarti.

Pengujian kekuatan sinyal Wi-Fi dilakukan untuk memastikan bahwa pengiriman notifikasi melalui Telegram dapat berjalan dengan lancar tanpa gangguan koneksi. ESP32 sebagai modul Wi-Fi sangat bergantung pada kestabilan sinyal untuk dapat terhubung ke internet. Jika sinyal Wi-Fi terlalu lemah, maka pengiriman data bisa terlambat, gagal, atau bahkan tidak terkirim sama sekali. Oleh karena itu, pengujian ini penting untuk mengetahui jarak efektif dan kondisi optimal agar komunikasi Telegram dapat dilakukan secara real-time dan andal.

Selain kekuatan sinyal Wi-Fi, waktu respon pengiriman notifikasi Telegram juga dipengaruhi oleh perangkat yang digunakan untuk menerima pesan, terutama tipe dan spesifikasi smartphone.

Berdasarkan pengujian, ditemukan bahwa beberapa tipe HP dengan spesifikasi lebih tinggi menerima pesan lebih cepat dibanding HP dengan spesifikasi lebih rendah atau yang memiliki mode hemat daya aktif.

4.2.3 Pengujian dan Analisa Sistem Keseluruhan

Pengujian dan analisa sistem secara keseluruhan dilakukan untuk mengetahui apakah seluruh komponen yang digunakan pada penelitian ini dapat bekerja secara terintegrasi sesuai dengan

perancangan yang telah dibuat. Pengujian dilakukan dengan mengoperasikan sistem secara menyeluruh, mulai dari proses pembacaan data oleh Pulse Sensor, sensor DS18B20, dan sensor LDR, kemudian data diproses oleh ESP32 untuk menentukan tindakan yang harus dilakukan terhadap aktuator berupa kipas angin, lampu, dan speaker. Selain itu, sistem juga diuji untuk memastikan bahwa setiap aktivitas yang terjadi dapat dikirimkan ke Bot Telegram sehingga pengguna dapat melakukan monitoring kondisi sistem secara *real time*.

4.2.3.1 Pengujian Fungsional

Pengujian dan analisis fungsional sistem dilakukan untuk memastikan bahwa seluruh fungsi yang telah dirancang pada sistem dapat berjalan sesuai dengan kebutuhan fungsional penelitian. Pengujian ini bertujuan untuk mengetahui apakah sistem mampu menerima masukan dari setiap sensor, memproses data sesuai dengan logika yang telah diprogram, serta menghasilkan keluaran yang sesuai pada aktuator. Selain itu, pengujian juga dilakukan untuk memastikan bahwa sistem dapat melakukan proses monitoring dan pengendalian melalui Bot Telegram dengan baik.

Selanjutnya dilakukan pengujian dan analisa logika sistem. Pengujian dan analisa logika sistem dilakukan untuk memastikan bahwa sistem mampu menghasilkan keluaran sesuai dengan logika pengendalian yang telah dirancang. Pengujian dilakukan dengan memberikan beberapa kondisi masukan yang berasal dari sensor DS18B20, sensor LDR, dan Pulse Sensor, kemudian mengamati respons sistem terhadap aktuator berupa kipas angin, lampu, dan speaker. Setiap keluaran yang dihasilkan dibandingkan dengan logika yang telah ditentukan pada program untuk mengetahui

apakah sistem telah bekerja sesuai dengan perancangan. Hasil pengujian logika sistem dapat dilihat pada Tabel 4.7.

Tabel 4.7 Pengujian Logika Sistem

No.	Kondisi Cahaya	Kondisi Suhu (°C)	Kondisi Denyut Jantung (BPM)	Kondisi Kipas	Kondisi Lampu	Kondisi Speaker
1.	3629	20.57	40	off	on	off
2.	3479	21.05	100	off	off	on
3.	3675	21.85	82	off	on	on
4.	3569	22.85	55	on	on	off
5.	2555	23.90	61	on	off	on
6.	3705	24.42	77	on	on	on
7.	2987	25.75	68	on	off	on
8.	3026	28.56	58	on	off	off
9.	3708	27.06	59	on	on	off
10.	3679	29.81	86	on	on	on

Berdasarkan hasil pengujian pada Tabel 4.8, sistem berhasil menjalankan logika pengendalian sesuai dengan algoritma yang telah dirancang. Ketika sensor DS18B20 mendeteksi suhu di atas 22°C, ESP32 memberikan perintah kepada relay untuk menghidupkan kipas angin. Sebaliknya, ketika suhu berada pada atau di bawah 22°C, kipas angin akan dimatikan secara otomatis.

Pada pengujian sensor LDR, sistem mampu mengendalikan lampu berdasarkan nilai ADC yang terbaca. Ketika sensor LDR membaca data kurang dari 100 lux, kondisi lingkungan dianggap gelap sehingga lampu dihidupkan. Sebaliknya, apabila nilai yang terbaca berada di

atas 100 lux, sistem menganggap kondisi lingkungan sudah cukup terang sehingga lampu dimatikan.

Sementara itu, Pulse Sensor digunakan untuk memantau denyut jantung pengguna. Ketika nilai denyut jantung berada di atas 60 BPM, sistem mengaktifkan speaker untuk memutar musik relaksasi. Setelah denyut jantung turun hingga berada pada atau di bawah 60 BPM, speaker akan dimatikan secara otomatis sebagai indikator bahwa pengguna telah memasuki kondisi yang lebih rileks.

Hasil pengujian menunjukkan bahwa seluruh kombinasi data masukan dari ketiga sensor menghasilkan keluaran yang sesuai dengan logika yang telah diprogram. Hal ini menunjukkan bahwa algoritma pengendalian pada sistem telah berjalan dengan baik sehingga setiap perubahan kondisi lingkungan maupun kondisi fisiologis pengguna dapat direspons secara otomatis oleh sistem.

Kemudian dilakukan pengujian terhadap konektivitas sistem. yang akan ditampilkan pada Tabel 4.8.

Tabel 4.8 Pengujian Konektivitas Sistem

No	Rencana Pengujian	Indikator Keberhasilan	Hasil Pengujian	Status	Waktu Respon
1.	Sistem dapat terhubung ke jaringan WiFi dan Bot Telegram.	ESP32 berhasil terhubung ke jaringan WiFi dan Bot Telegram.	Sistem berhasil terhubung dan Bot Telegram sehingga notifikasi	Berhasil	2 detik

No	Rencana Pengujian	Indikator Keberhasilan	Hasil Pengujian	Status	Waktu Respon
			awal dikirim.		
3.	Sistem mendeteksi suhu di sekitar tempat tidur.	Sistem dapat mendeteksi suhu di sekitar tempat tidur.	Sistem berhasil mendeteksi suhu.	Berhasil	3 detik
4.	Sistem mendeteksi cahaya sekitar tempat tidur.	Sistem dapat mendeteksi cahaya di sekitar tempat tidur.	Sistem berhasil mendeteksi cahaya.	Berhasil	2 detik
5.	Sistem mematikan dan menghidupkan kipas angin, lampu dan speaker secara otomatis.	Sistem dapat mematikan dan menghidupkan kipas angin, lampu, dan speaker	Sistem dapat mematikan dan mengaktifkan komponen secara otomatis.	Berhasil	1 detik

No .	Rencana Pengujian	Indikator Keberhasilan	Hasil Pengujian	Status	Waktu Respon
		berdasarkan data masukan dari masing-masing sensor.			
6.	Sistem terhubung ke aplikasi telegram.	Sistem dapat menampilkan informasi berupa rekap data selama sistem berjalan saat pengguna berada pada fase tidur.	Sistem dapat menampilkan laporan pada bot telegram.	Berhasil	1 detik

Pengujian konektivitas sistem terhadap bot Telegram dilakukan untuk memastikan bahwa mikrokontroler (ESP32) dapat mengirim dan menerima data secara real-time melalui jaringan internet ke platform

Telegram. Tujuannya adalah untuk memastikan kestabilan komunikasi, memastikan bahwa notifikasi dapat dikirim dengan benar, dan pengguna dapat memberikan perintah atau memantau status sistem dari jarak jauh. Pengujian ini juga berguna untuk melihat respon waktu (latency) dan keandalan sinyal saat digunakan dalam kondisi jaringan yang berbeda.

Kemudian dilakukan pengujian secara keseluruhan pada notifikasi bot telegram. Dilakukan pengujian dengan mengamati dan menganalisa kemampuan aplikasi bot telegram dalam menerima pesan yang dikirim oleh mikrokontroler selama sistem berjalan. Berikut adalah pengujian dan analisa pada bot telegram yang akan ditunjukkan oleh Tabel 4.9.

Tabel 4.9 Pengujian dan Analisa Notifikasi Bot Telegram

No.	Keadaan	Keadaan yang Diharapkan	Hasil	Indikator	Waktu (detik)
1.	Terdapat koneksi internet	Aplikasi telegram dapat mengirimkan data yang diminta.	Notifikasi aplikasi telegram dapat terkirim dan tidak terjadi eror.	Berhasil	1
2.	Tidak ada koneksi internet	Sistem tidak dapat mengirimkan data .	Data tidak terkirim ke bot telegram	Berhasil	-

3.	Sistem menampilkan data sensor dan status perangkat	Sistem dapat menampilkan data sensor dan status perangkat.	Sistem dapat menampilkan data sensor dan status perangkat selama sistem dalam kondisi aktif yang diberikan perintah /status.	Berhasil	2
4.	Sistem menampilkan rekap data harian.	Sistem dapat menampilkan berapa kali perangkat seperti kipas angin dan speaker dihidupkan.	Sistem dapat menampilkan rekap data harian sistem melalui perintah /report.	Berhasil	2
5.	Sistem mengaktifkan mode tidur.	Sistem dapat mengaktifkan mode tidur.	Sistem dapat mengaktifkan mode tidur melalui perintah /sleepmode	Berhasil	2

Pengujian notifikasi bot Telegram dilakukan untuk memastikan bahwa sistem dapat mengirim pesan peringatan atau status secara otomatis dan tepat waktu kepada pengguna melalui bot Telegram. Tujuan dari pengujian ini adalah untuk mengevaluasi apakah sistem benar-benar mampu memberikan notifikasi saat terjadi kondisi tertentu (misalnya suhu tinggi, cahaya terlalu terang, atau denyut jantung tidak normal), serta untuk memastikan bahwa isi pesan dan waktu pengirimannya sesuai dengan logika program. Rata-rata waktu sistem dalam mengirimkan data ke telegram agar bisa dibaca oleh pengguna adalah 2 detik. Berikut merupakan tampilan pesan yang dapat diberikan oleh pengguna pada bot telegram:



Gambar 4.13 Pesan Telegram dengan Perintah Status

Gambar 4.14 merupakan tampilan pesan yang dikirim sistem ke aplikasi bot telegram jika user mengetik `"/status"` yang bertujuan untuk menampilkan data real time sistem.



Gambar 4.14 Pesan Telegram dengan perintah Sleepmode

Gambar 4.15 merupakan tampilan pesan yang dikirim sistem ke aplikasi bot telegram jika user mengetikkan perintah “/sleepmode” yang digunakan jika pengguna ingin mengaktifkan sistem secara manual. Fitur ini dirancang agar pengguna dapat lebih mudah mengaktifkan kondisi relaksasi sebelum tidur, seperti mematikan lampu dan mengaktifkan audio relaksasi secara langsung.



Gambar 4.15 Pesan Telegram dengan perintah Relax

Gambar 4.16 merupakan tampilan pesan yang dikirim sistem ke aplikasi bot telegram jika user mengetikkan perintah “/relax” yang digunakan jika pengguna ingin mengaktifkan audio relaksasi secara manual. Fitur ini difokuskan untuk membantu pengguna mencapai kondisi relaksasi awal melalui pemutaran audio relaksasi tanpa mengubah keseluruhan kondisi sistem.



Gambar 4.16 Pesan Telegram dengan perintah Report

Gambar 4.17 merupakan tampilan pesan yang dikirim sistem ke aplikasi bot telegram jika user mengetikkan perintah “/report” yang digunakan untuk memberikan informasi kondisi lingkungan tidur dan kondisi pengguna berdasarkan hasil pembacaan sensor sehingga pengguna dapat mengetahui apakah kondisi tidur sudah mendukung kualitas tidur atau belum.

4.2.3.2 Pengujian Non-Fungsional

Pengujian non-fungsional adalah proses evaluasi aspek-aspek sistem yang tidak terkait langsung dengan fungsionalitas utama, tetapi lebih kepada kualitas sistem. Pengujian non fungsional dilakukan untuk mengetahui apakah sistem telah memenuhi kebutuhan di luar fungsi utama sistem, seperti keandalan koneksi internet, komunikasi dengan Bot Telegram, serta kestabilan catu daya selama sistem beroperasi. Pengujian ini bertujuan memastikan sistem dapat bekerja secara optimal dan memberikan pengalaman penggunaan yang baik.

a. Pengujian Keandalan Sistem

Pengujian keandalan sistem merupakan pengujian yang dilakukan untuk menunjukkan sejauh mana sistem dapat berfungsi secara konsisten dan stabil tanpa mengalami eror atau gangguan dalam jangka waktu tertentu. Berikut merupakan tabel pengujian sistem yang ditunjukkan oleh Tabel 4.10.

Tabel 4.10 Pengujian Keandalan Sistem

No	Skenario Uji	Durasi Uji	Kriteria Keberhasilan	Hasil Pengamatan	Indikator
1.	Sistem beroperasi secara kontinu.	8 jam	Tidak terjadi <i>hang</i> , <i>restart</i> atau error.	Tidak ditemukan <i>restart</i> atau error.	Berhasil
2.	Sistem kembali bekerja setelah kehilangan daya.	10 kali percobaan	Sistem berhasil booting, reconnect Wi-Fi dan Telegram dalam <10 detik.	10/10 kali sistem kembali terhubung ke Wi-Fi dan Bot Telegram.	Berhasil
3.	Sistem melakukan monitoring secara	8 jam	Sensor tetap terbaca normal dan seluruh	Data sensor tetap terbaca normal	Berhasil

	terus menerus.		output bekerja sesuai logika.	dan output bekerja sesuai logika.	
--	----------------	--	-------------------------------	-----------------------------------	--

Berdasarkan Tabel 4.10, pengujian keandalan sistem dilakukan untuk mengetahui kemampuan sistem dalam beroperasi secara stabil dan konsisten selama digunakan. Selain itu, pengujian ini bertujuan untuk memastikan bahwa sistem tetap dapat menjalankan fungsi monitoring dan pengendalian secara normal dalam jangka waktu tertentu, mampu melakukan pemulihan setelah kehilangan daya listrik, serta tidak mengalami gangguan yang dapat menyebabkan kegagalan sistem. Pengujian ini meliputi tiga skenario, yaitu pengoperasian sistem secara terus-menerus, pengujian setelah kehilangan daya listrik, serta pengujian stabilitas pembacaan sensor selama sistem digunakan. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem tidak hanya mampu bekerja dengan baik pada saat awal dijalankan, tetapi juga tetap mempertahankan kinerjanya selama proses penggunaan.

Pengujian pertama dilakukan dengan mengoperasikan sistem secara terus-menerus selama 8 jam. Selama pengujian berlangsung, seluruh sensor, ESP32, relay, dan perangkat keluaran bekerja secara normal tanpa mengalami hang, restart, maupun kesalahan fungsi. Selain itu, sistem tetap mampu melakukan pembacaan data sensor serta mengendalikan perangkat keluaran sesuai dengan logika yang telah diprogram. Hasil tersebut menunjukkan bahwa sistem memiliki tingkat

kestabilan yang baik untuk digunakan dalam waktu yang relatif lama.

Pengujian kedua dilakukan untuk mengetahui kemampuan sistem dalam melakukan pemulihan setelah kehilangan daya listrik. Pengujian dilakukan sebanyak 10 kali dengan cara memutus dan menghubungkan kembali sumber daya listrik. Hasil pengujian menunjukkan bahwa pada setiap percobaan sistem berhasil melakukan proses booting, terhubung kembali ke jaringan Wi-Fi dan Bot Telegram, serta kembali menjalankan fungsi monitoring dan pengendalian dalam waktu kurang dari 10 detik. Hal ini menunjukkan bahwa sistem memiliki kemampuan pemulihan yang baik setelah terjadi pemadaman listrik.

Pengujian ketiga dilakukan untuk mengevaluasi kestabilan sistem dalam melakukan pembacaan sensor selama sistem beroperasi secara terus-menerus. Selama pengujian 8 jam, seluruh sensor tetap memberikan data yang konsisten dan perangkat keluaran merespons sesuai dengan kondisi yang terdeteksi. Tidak ditemukan kegagalan pembacaan sensor maupun kesalahan dalam proses pengendalian perangkat.

Berdasarkan ketiga skenario pengujian tersebut, dapat disimpulkan bahwa sistem memiliki tingkat keandalan yang baik. Sistem mampu beroperasi secara stabil dalam jangka waktu yang lama, dapat pulih dengan cepat setelah kehilangan daya listrik, serta mempertahankan kinerja pembacaan sensor dan pengendalian perangkat secara konsisten. Hasil ini menunjukkan bahwa sistem layak digunakan sebagai sistem otomasi kamar tidur yang bekerja secara kontinu.

b. Pengujian Kemudahan Penggunaan Sistem

Pengujian kemudahan penggunaan (*usability*) dilakukan untuk mengetahui tingkat kemudahan pengguna dalam memahami dan mengoperasikan sistem yang telah dirancang. Pengujian ini dilakukan sebagai evaluasi awal terhadap pengalaman pengguna (*user experience*) sebelum sistem digunakan secara lebih luas.

Pengujian melibatkan 5 orang responden yang diminta mencoba sistem secara langsung, kemudian memberikan penilaian terhadap beberapa aspek menggunakan skala Likert 1–5, dimana nilai 1 menunjukkan sangat sulit atau sangat tidak setuju, sedangkan nilai 5 menunjukkan sangat mudah atau sangat setuju. Meskipun jumlah responden masih terbatas, hasil pengujian ini dapat memberikan gambaran awal mengenai tingkat kemudahan penggunaan sistem serta mengidentifikasi aspek-aspek yang masih perlu diperbaiki. Hasil pengujian kemudahan penggunaan dapat dilihat pada Tabel 4.11.

Tabel 4.11 Pengujian Kemudahan Penggunaan Sistem

No.	Aspek yang Dinilai	Skala 1-5 (1: sangat sulit, 5: sangat mudah)
1.	Kemudahan memahami cara kerja sistem.	4.2
2.	Kenyamana menggunakan sistem.	4.0
3.	Manfaat sistem dalam membantu mengurangi gangguan tidur.	4.4
Rata-Rata Keseluruhan		4.2

Berdasarkan hasil pengujian pada Tabel 4.11, aspek kemudahan memahami cara kerja sistem memperoleh nilai rata-rata sebesar 4,2. Hal ini menunjukkan bahwa sebagian besar responden dapat memahami cara kerja sistem dengan baik tanpa mengalami kesulitan yang berarti. Pada aspek kenyamanan penggunaan, sistem memperoleh nilai rata-rata sebesar 4,0, yang menunjukkan bahwa sistem cukup nyaman digunakan selama proses pengujian. Sementara itu, aspek manfaat sistem memperoleh nilai rata-rata tertinggi yaitu 4,4, yang menunjukkan bahwa responden memberikan penilaian positif terhadap manfaat sistem dalam membantu menciptakan kondisi tidur yang lebih nyaman. Secara keseluruhan, sistem memperoleh nilai rata-rata sebesar 4,2 dari skala 5. Hasil tersebut menunjukkan bahwa sistem memiliki tingkat kemudahan penggunaan yang baik, sehingga pengguna dapat memahami dan mengoperasikan sistem dengan mudah.

c. Pengujian Efektivitas Sistem

Pengujian efektivitas sistem merupakan pengujian yang dilakukan untuk mengukur apakah sistem benar-benar dapat membantu untuk meningkatkan kualitas tidur pengguna. Pengujian dilakukan dengan memberikan kuesioner ISI (Insomnia Severity Index). Insomnia Severity Index (ISI) adalah instrumen skala penilaian diri (self-report) yang digunakan untuk mengukur tingkat keparahan insomnia seseorang. Kuesioner ini terdiri dari 7 pertanyaan yang mencakup aspek kesulitan tidur, kepuasan tidur, serta dampak insomnia terhadap kehidupan sehari-hari. Skor total ISI berkisar antara 0 hingga 28, dengan kategori sebagai berikut[36]:

- 0 – 7 → Tidak ada insomnia atau sangat ringan
- 8 – 14 → Insomnia ringan
- 15 – 21 → Insomnia sedang
- 22 – 28 → Insomnia parah

Berikut merupakan tabel kuesioner yang berisikan pertanyaan dan skala parameter yang ditunjukkan pada tabel 4.12.

Tabel 4.12 Tabel Pertanyaan Kuesioner ISI

No.	Pertanyaan	Skala Parameter
1.	Seberapa sulit anda mengalami kesulitan untuk tidur?	0 = Tidak Ada 1 = Ringan 2 = Sedang 3 = Parah 4 = Sangat Parah
2.	Seberapa sulit anda mengalami sering terbangun di Tengah malam atau terlalu awal di pagi hari?	0 = Tidak Ada 1 = Ringan 2 = Sedang 3 = Parah 4 = Sangat Parah
3.	Seberapa sulit Anda mengalami kualitas tidur yang buruk secara keseluruhan?	0 = Tidak Ada 1 = Ringan 2 = Sedang 3 = Parah 4 = Sangat Parah
4.	Seberapa puas atau tidak puas Anda dengan pola tidur Anda saat ini?	0 = Sangat Puas 1 = Puas 2 = Netral 3 = Tidak Puas

		4 = Sangat Tidak Puas
5.	Seberapa besar masalah tidur ini mempengaruhi aktivitas sehari-hari Anda (misalnya, pada pekerjaan, kehidupan sosial, konsentrasi, suasana hati)?	0 = Tidak Ada Dampak 1 = Dampak Ringan 2 = Dampak Sedang 3 = Dampak signifikan 4 = Dampak Sangat Parah
6.	Seberapa sering Anda merasa khawatir atau terganggu oleh masalah tidur ini?	0 = Tidak Pernah 1 = Jarang 2 = Kadang-Kadang 3 = Sering 4 = Selalu
7.	Menurut anda, sejauh mana gangguan tidur ini terlihat oleh orang lain (misalnya, teman, keluarga, atau rekan kerja)?	0 = Tidak Terlihat 1 = Sedikit Terlihat 2 = Cukup Terlihat 3 = Jelas Terlihat 4 = Sangat Terlihat

Pada tabel 4.12 berisi pertanyaan yang diajukan kepada responden untuk menilai Tingkat insomnia yang sedang dialami oleh responden. Pengisian dilakukan sebelum dan sesudah pemakaian sistem sebagai tolak ukur pembandingan untuk ditarik kesimpulan apakah sistem efektif atau tidak. Dengan menggunakan kuesioner

ISI sebelum dan sesudah penggunaan sistem, dapat dilakukan analisis mengenai apakah sistem yang dirancang memberikan dampak positif terhadap kualitas tidur pengguna[38]. Berikut merupakan interpretasi skor berdasarkan tabel di atas: Berdasarkan Tabel 4.12 dilakukan pengisian kuesioner oleh responden, berikut merupakan hasil dari data responden yang akan ditunjukkan oleh Tabel 4.13.

Tabel 4.13 Data Responden

No.	Responden	JK	Usia	Skor ISI sebelum	Kategori	Skor ISI sesudah	Kategori	Penurunan (%)
1.	R1	L	20	18	Sedang	16	Sedang	11,0
2.	R2	P	26	21	Sedang	20	Sedang	4,7
3.	R3	P	19	15	Sedang	13	Ringan	15,3
4.	R4	L	21	24	Parah	22	Parah	8,3
5.	R5	P	47	13	Ringan	12	Ringan	7,7
Rata-Rata				18,2		16.6		9,4%

Pengujian efektivitas sistem dalam membantu mengurangi gangguan tidur dilakukan dengan menggunakan Insomnia Severity Index (ISI) sebagai alat ukur. Indeks ini digunakan untuk menilai tingkat keparahan insomnia berdasarkan sejumlah parameter, seperti kesulitan tidur, frekuensi terbangun di malam hari, serta dampaknya terhadap aktivitas sehari-hari. Dalam pengujian ini, 5 orang responden dengan keluhan insomnia ringan hingga berat menggunakan sistem selama 8 hari berturut-turut. Untuk mengevaluasi dampaknya, responden diminta mengisi kuesioner ISI sebelum dan sesudah penggunaan sistem.

Untuk mencari hasil rata-rata selisih perubahan tingkat insomnia responden sebelum dan sesudah pemakaian sistem dapat diperoleh dengan persamaan berikut[36]:

$$\text{persentase Perubahan} = \frac{ISI \text{ sebelum} - ISI \text{ sesudah}}{ISI \text{ sebelum}} \times 100\% \dots \dots \dots (4.4)$$

erdasarkan hasil pengujian pada Tabel 4.12, seluruh responden mengalami penurunan skor *Insomnia Severity Index (ISI)* setelah menggunakan sistem selama delapan malam berturut-turut. Sebelum menggunakan sistem, rata-rata skor ISI responden adalah 18,2, yang termasuk dalam kategori insomnia sedang. Setelah penggunaan sistem, rata-rata skor ISI menurun menjadi 16,6. Hal ini menunjukkan adanya penurunan rata-rata sebesar 1,6 poin atau sekitar 9,4% dibandingkan kondisi sebelum penggunaan sistem.

Jika ditinjau berdasarkan masing-masing responden, seluruh responden menunjukkan penurunan skor ISI meskipun besar penurunannya berbeda-beda. Penurunan terbesar terjadi pada responden ketiga, yaitu dari skor 15 menjadi 13 atau sebesar 15,3%, sehingga kategori insomnia berubah dari sedang menjadi ringan. Sementara itu, penurunan terkecil terjadi pada responden kedua, yaitu dari skor 21 menjadi 20 atau sebesar 4,7%, sehingga masih berada pada kategori insomnia sedang.

Secara keseluruhan, hasil pengujian menunjukkan bahwa penggunaan sistem selama delapan malam memberikan kecenderungan penurunan tingkat keparahan insomnia pada seluruh responden. Meskipun jumlah responden masih terbatas, hasil tersebut menunjukkan bahwa sistem memiliki potensi untuk membantu meningkatkan kualitas tidur pengguna melalui

pengaturan suhu, pencahayaan, dan musik terapi secara otomatis sesuai kondisi lingkungan dan pengguna.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa yang dilakukan terhadap sistem pencegah gejala insomnia ini dapat diperoleh kesimpulan yaitu sebagai berikut:

1. Sistem berhasil dirancang untuk menciptakan lingkungan tidur yang optimal melalui pengaturan suhu, pencahayaan, dan suara secara otomatis berdasarkan data dari sensor DS18B20, LDR, dan Pulse Sensor. Pengujian akurasi menunjukkan sensor DS18B20 memiliki rata-rata eror sebesar 1,67% (akurasi 98,33%), sensor LDR sebesar 0,37% (akurasi 99,63%), dan Pulse Sensor sebesar 1,3% (akurasi 98,7%), sehingga ketiga sensor dinilai layak digunakan sebagai masukan sistem.
2. Sistem berhasil mengendalikan kipas angin, lampu, dan speaker secara otomatis (*on/off*) berdasarkan hasil pembacaan sensor sesuai dengan logika kendali yang telah dirancang, sebagaimana dibuktikan melalui pengujian logika sistem (Tabel 4.7) dan pengujian relay (Tabel 4.5) yang seluruhnya menunjukkan hasil berhasil.
3. Sistem berhasil terhubung dengan Bot Telegram dan mampu menampilkan data riwayat aktivitas sistem kepada pengguna secara *real-time*, dengan rata-rata waktu pengiriman notifikasi selama 2 detik, serta mendukung perintah */status*, */sleepmode*, */relax*, dan */report*.
4. Pengujian kestabilan menunjukkan sistem mampu beroperasi secara kontinu selama 8 jam tanpa kegagalan dan berhasil pulih otomatis ke jaringan WiFi dan Telegram dalam waktu kurang dari 10 detik pada seluruh (10/10) percobaan kehilangan daya.
5. Pengujian awal terhadap lima responden menunjukkan skor kemudahan penggunaan (*usability*) rata-rata sebesar 4,2 dari skala 5, serta penurunan rata-

rata skor *Insomnia Severity Index* (ISI) sebesar 9,4% setelah penggunaan sistem selama delapan malam berturut-turut, yang mengindikasikan bahwa sistem berpotensi mendukung perbaikan kualitas lingkungan tidur pengguna. Namun, mengingat jumlah responden yang terbatas ($n=5$) dan durasi pengujian yang relatif singkat, hasil ini belum dapat digeneralisasi secara luas dan memerlukan pengujian lanjutan dengan cakupan yang lebih besar.

5.2 Saran

Berikut merupakan saran yang dari penelitian ini yang bisa digunakan untuk penelitian selanjutnya agar penelitian ini bisa menjadi referensi dan dapat dikembangkan serta meningkatkan daya kerja sistem sebagai berikut :

1. Pada penelitian selanjutnya diharapkan dapat dikembangkan sebuah sistem yang dapat menentukan tingkat insomnia seseorang.
2. Sistem ini dirancang masih menggunakan kipas angin sebagai media pengatur sirkulasi udara sehingga pengendalian suhu ruangan belum dapat dilakukan secara presisi seperti penggunaan air conditioner (AC). Oleh karena itu, pada pengembangan selanjutnya sistem dapat dikombinasikan dengan perangkat pendingin ruangan yang lebih optimal agar pengaturan suhu dapat dilakukan dengan lebih stabil dan akurat.

