

Solusi IoT Terbaru untuk Kontrol AC Hemat Energi dan Cerdas

Aidil Saputra Kirsan^{*1}, Ahmad Rifa'i², Irma Fitria³

¹Program Studi Sistem Informasi, Institut Teknologi Kalimantan

²Program Studi Teknik Informatika, Politeknik Negeri Indramayu

³Program Studi Statistika, Institut Teknologi Kalimantan

Email: ^{*1}aidil@lecturer.itk.ac.id, ²ahmrifai.ar@gmail.com, ³irma.fitria@lecturer.itk.ac.id

(Naskah masuk: 24 Maret 2023, diterima untuk diterbitkan: 27 Juli 2023)

Abstrak: Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi solusi otomatisasi sistem kontrol suhu dan pengendalian AC untuk meningkatkan efisiensi energi. Metodologi yang digunakan terdiri dari empat tahap utama, yaitu desain, implementasi, pengujian, dan evaluasi. Pada tahap desain, dipilih arsitektur sistem dan komponen perangkat keras berdasarkan tinjauan literatur dan persyaratan sistem. Pada tahap implementasi, komponen perangkat keras dirangkai, dan perangkat lunak dikembangkan menggunakan lingkungan pemrograman Arduino IDE. Pada tahap pengujian, solusi diuji dalam pengaturan laboratorium untuk mengevaluasi kinerja dan efektivitasnya. Pada tahap evaluasi, kinerja dan efektivitas solusi dievaluasi berdasarkan beberapa kriteria, termasuk konsumsi energi, kemudahan penggunaan, dan keandalan sistem. Hasil penelitian menunjukkan bahwa solusi yang diusulkan efektif meningkatkan efisiensi energi dan memiliki stabilitas yang baik.

Kata Kunci – Otomatisasi; Sistem kontrol suhu; AC; Efisiensi energi; Arduino

Latest IoT Solution for Energy Saving and Smart AC Control

Abstract: This research aims to develop and evaluate automation solutions for temperature control and AC control systems to increase energy efficiency. The methodology used consists of four main stages, namely design, implementation, testing and evaluation. At the design stage, the system architecture and hardware components are selected based on a literature review and system requirements. In the implementation phase, hardware components are assembled, and software is developed using the Arduino IDE programming environment. In the testing phase, the solution is tested in a laboratory setting to evaluate its performance and effectiveness. In the evaluation stage, the performance and effectiveness of the solution are evaluated based on several criteria, including energy consumption, ease of use, and system reliability. The research results show that the proposed solution effectively improves energy efficiency and has good stability.

Keywords – Automation; Temperature control system; AC; Energy efficiency; Arduino

1. PENDAHULUAN

Dengan munculnya Internet of Things (IoT) [1]–[7], berbagai perangkat dan sistem sekarang dapat terhubung dan berinteraksi satu sama lain, menyediakan solusi baru dan inovatif untuk masalah dunia nyata. Salah satu area signifikan yang telah mendapat manfaat dari teknologi IoT adalah manajemen dan kontrol energi, terutama di industri HVAC (Pemanas, Ventilasi, dan Pengkondisian Udara) [8]. Cara tradisional untuk mengontrol AC adalah melalui saklar manual atau pengontrol jarak jauh. Namun, metode ini tidak efisien dalam hal energi dan dapat menyebabkan konsumsi energi yang tidak perlu.

Untuk mengatasi masalah ini, sebuah solusi berbasis IoT untuk pengontrolan AC yang hemat energi dan cerdas telah dikembangkan. Solusi novel ini bertujuan untuk menyediakan cara yang mudah dan efisien dalam mengontrol AC, sambil juga mengurangi konsumsi energi dan biaya. Sistem yang diusulkan terdiri dari beberapa komponen, termasuk mikrokontroler, sensor deteksi gerakan PIR, sensor suhu DHT22, relay, sumber daya, dan output kontrol AC. Dengan menggunakan sensor deteksi gerakan PIR, sistem dapat secara otomatis menghidupkan AC saat seseorang masuk ke dalam ruangan dan mematikannya saat suhu mencapai level yang diinginkan.

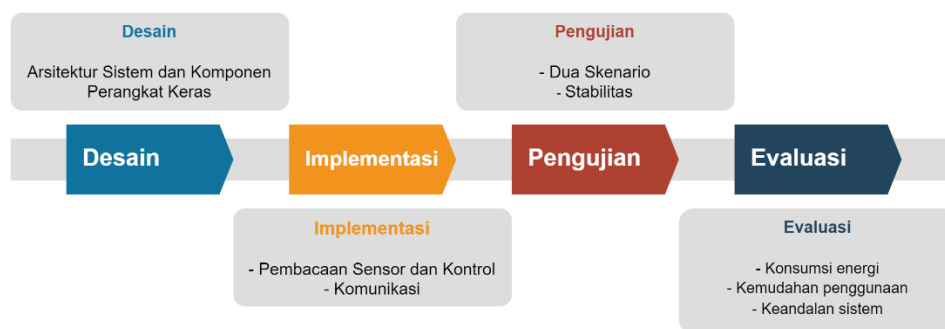
Selain itu, sistem dapat dikendalikan secara manual dari aplikasi Android mobile, yang memungkinkan akses jarak jauh ke kontrol AC [9]–[11].

Penelitian ini mempresentasikan pengembangan solusi berbasis IoT untuk pengontrolan AC yang hemat energi dan cerdas, termasuk desain, implementasi, dan pengujian. Hasil studi menunjukkan bahwa solusi yang diusulkan efektif dalam mengurangi konsumsi energi sambil menyediakan cara pengontrolan AC yang efisien dan mudah digunakan. Makalah ini juga membahas potensi dampak dan manfaat solusi yang diusulkan, termasuk penghematan energi, pengurangan biaya, dan peningkatan keberlanjutan lingkungan.

Secara keseluruhan, solusi berbasis IoT untuk pengontrolan AC yang hemat energi dan cerdas adalah teknologi yang menjanjikan yang dapat secara signifikan meningkatkan manajemen dan kontrol energi di industri HVAC. Penelitian ini bertujuan untuk berkontribusi pada tubuh literatur yang berkembang tentang solusi berbasis IoT untuk manajemen energi, memberikan wawasan dan rekomendasi untuk penelitian dan pengembangan di masa depan.

2. METODE PENELITIAN

Metodologi ini bertujuan untuk mengembangkan solusi berbasis IoT untuk pengendalian pendingin udara yang efisien secara energi dan pintar. Metodologi yang digunakan dalam penelitian ini terdiri dari empat tahap utama, termasuk desain, implementasi, pengujian, dan evaluasi dari solusi yang diusulkan.



Gambar 1. Alur Penelitian

Alur penelitian yang diperlihatkan terdiri dari empat tahapan utama, yaitu desain, implementasi, pengujian, dan evaluasi. Pada tahap desain, dilakukan perancangan terhadap arsitektur sistem dan komponen perangkat keras yang akan digunakan dalam penelitian ini. Hal ini didasarkan pada tinjauan literatur dan persyaratan sistem yang telah ditetapkan. Pada tahap implementasi, dilakukan pembacaan sensor dan kontrol serta komunikasi antar komponen sistem. Langkah ini dilakukan dengan menggunakan perangkat keras yang telah dirancang pada tahap desain. Proses implementasi menggunakan lingkungan pemrograman Arduino IDE.

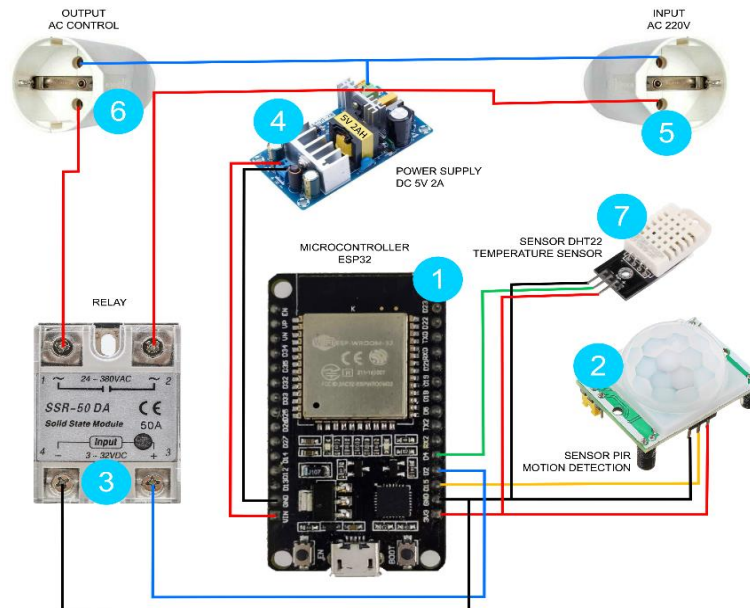
Pada tahap pengujian, dilakukan dua skenario pengujian untuk mengevaluasi kinerja dan stabilitas sistem. Skenario pengujian tersebut didasarkan pada tujuan dan persyaratan sistem yang telah ditetapkan pada tahap desain. Pada tahap evaluasi, kinerja dan efektivitas sistem dievaluasi berdasarkan beberapa kriteria, termasuk konsumsi energi, kemudahan penggunaan, dan keandalan sistem. Hasil evaluasi tersebut dijadikan dasar untuk menentukan keberhasilan dari solusi yang diusulkan dalam penelitian ini.

3. HASIL DAN PEMBAHASAN

3.1. Tahapan Desain

Pada tahap desain, arsitektur sistem dan komponen perangkat keras dipilih berdasarkan tinjauan literatur dan persyaratan sistem. Sistem yang diusulkan terdiri dari mikrokontroler, sensor deteksi gerakan PIR, sensor suhu DHT22, relay, catu daya, dan keluaran kontrol AC. Mikrokontroler yang digunakan dalam penelitian ini adalah papan pengembangan ESP32, yang dipilih karena

konsumsi daya yang rendah dan kekuatan pemrosesan yang tinggi. Sensor deteksi gerakan PIR dan sensor suhu DHT22 dihubungkan ke papan pengembangan ESP32 menggunakan pin digital. Relay dihubungkan ke papan pengembangan ESP32 menggunakan pin GPIO untuk mengontrol pasokan daya AC.



Gambar 2. Arsitektur Sistem dan Komponen Perangkat Keras

Sistem terdiri dari beberapa komponen, yaitu microcontroller ESP32 sebagai otak sistem, sensor PIR Motion Detection sebagai sensor gerak, sensor DHT22 sebagai sensor suhu dan kelembaban, relay sebagai switch AC, dan power supply DC 5V 2A sebagai sumber daya. ESP32 terhubung dengan sensor PIR Motion Detection dan sensor DHT22 melalui pin input/output digital. Kemudian, relay terhubung ke pin output digital ESP32 dan ke input AC 220V untuk mengontrol aliran listrik AC. Dalam operasinya, sensor PIR Motion Detection mendeteksi gerakan dan memberi sinyal ke ESP32. ESP32 kemudian memeriksa suhu dan kelembaban menggunakan sensor DHT22. Berdasarkan data yang diterima, ESP32 mengatur relay untuk mengontrol aliran listrik AC dan mengatur suhu yang diinginkan. Semua proses ini diatur oleh program yang dijalankan pada ESP32.

Untuk antarmuka pengguna, solusi yang diusulkan akan menggunakan sebuah aplikasi Android yang akan berfungsi sebagai pengontrol AC. Aplikasi ini akan menampilkan suhu ruangan saat ini, status AC (nyala/mati), dan juga status gerakan PIR sensor (deteksi/ tidak deteksi). Antarmuka pengguna akan didesain dengan tampilan minimalis dan sederhana, menggunakan ikon dan teks yang jelas dan mudah dipahami oleh pengguna.

3.2. Tahapan Implementasi

Pada tahap implementasi, komponen perangkat keras dirangkai, dan perangkat lunak dikembangkan menggunakan lingkungan pemrograman Arduino IDE. Kode perangkat lunak dibagi menjadi dua bagian utama: fungsi pembacaan sensor dan kontrol, dan fungsi komunikasi. Fungsi pembacaan sensor dan kontrol bertanggung jawab untuk membaca data sensor deteksi gerakan PIR dan sensor suhu DHT22 dan mengontrol relay AC berdasarkan data sensor. Fungsi komunikasi bertanggung jawab untuk menetapkan koneksi Wi-Fi dan berkomunikasi dengan aplikasi Android mobile.

1. Fungsi pembacaan sensor dan kontrol pada pseudocode Gambar 3.
 - a. Sensor deteksi gerakan PIR dan sensor suhu DHT22 akan terus dipantau oleh mikrokontroler.
 - b. Mikrokontroler akan membaca data sensor dan memprosesnya.

- c. Berdasarkan data sensor, mikrokontroler akan mengontrol relay AC untuk menghidupkan atau mematikan AC, sehingga menghemat energi dan mempertahankan suhu ruangan yang diinginkan.

```

BacaSensorPIR()
{
    // Baca data sensor gerakan PIR
    int pirValue = digitalRead(pirPin);
    // Jika sensor mendeteksi gerakan, ubah status AC menjadi ON
    if(pirValue == HIGH){
        acStatus = ON;
    }
    // Jika tidak ada gerakan yang terdeteksi, ubah status AC menjadi OFF
    else{
        acStatus = OFF;
    }
}

BacaSensorSuhu()
{
    // Baca data sensor suhu DHT22
    float temperature = dht.readTemperature();
    // Jika suhu lebih tinggi dari batas atas, ubah status AC menjadi OFF
    if(temperature > maxTemp){
        acStatus = OFF;
    }
    // Jika suhu lebih rendah dari batas bawah, ubah status AC menjadi ON
    else if(temperature < minTemp){
        acStatus = ON;
    }
}

KontrolAC()
{
    // Jika status AC adalah ON, hidupkan relay AC
    if(acStatus == ON){
        digitalWrite(relayPin, HIGH);
    }
    // Jika status AC adalah OFF, matikan relay AC
    else{
        digitalWrite(relayPin, LOW);
    }
}

```

Gambar 3. Algoritma fungsi pembacaan sensor dan control

Fungsi pembacaan sensor dan kontrol akan menggunakan bahasa pemrograman C++ untuk membaca data dari sensor deteksi gerakan PIR dan sensor suhu DHT22, dan mengontrol relay AC berdasarkan data sensor. Kode di atas merupakan tiga fungsi yang bekerja bersama untuk mengontrol AC dengan menggunakan sensor PIR dan DHT22. Fungsi pertama yaitu BacaSensorPIR() berfungsi untuk membaca data sensor gerakan PIR. Jika sensor mendeteksi gerakan, maka status AC akan diubah menjadi ON, jika tidak ada gerakan yang terdeteksi, maka status AC akan diubah menjadi OFF. Fungsi kedua yaitu BacaSensorSuhu() berfungsi untuk membaca data sensor suhu DHT22. Jika suhu lebih tinggi dari batas atas, maka status AC akan diubah menjadi OFF, jika suhu lebih rendah dari batas bawah, maka status AC akan diubah menjadi ON. Fungsi ketiga yaitu KontrolAC() berfungsi untuk menghidupkan dan mematikan relay AC sesuai dengan status AC yang sudah diubah oleh fungsi BacaSensorPIR() dan BacaSensorSuhu(). Jika status AC adalah ON, maka relay AC akan dihidupkan, sedangkan jika status AC adalah OFF, maka relay AC akan dimatikan.

2. Fungsi komunikasi pada pseudocode Gambar 4.
 - a. Mikrokontroler akan menetapkan koneksi Wi-Fi dan membuatnya tersedia untuk aplikasi mobile.
 - b. Aplikasi mobile akan terhubung ke mikrokontroler melalui Wi-Fi.
 - c. Mikrokontroler akan mengirim dan menerima data melalui Wi-Fi untuk memungkinkan pengguna mengontrol AC dari aplikasi mobile.

```
SetupWiFi()
{
    // Inisialisasi koneksi WiFi
    WiFi.begin(ssid, password);

    // Tunggu hingga terhubung ke WiFi
    while(WiFi.status() != WL_CONNECTED) {
        delay(1000);
    }
}

KirimData()
{
    // Buat koneksi TCP ke server
    WiFiClient client;
    if (!client.connect(server, port)) {
        return;
    }

    // Kirim data status AC ke server
    String data = String(acStatus);
    client.print(data);
}

TerimaData()
{
    // Tunggu hingga ada data yang diterima
    while(!client.available()){
        delay(1000);
    }

    // Baca data yang diterima
    String data = client.readStringUntil('\r');

    // Ubah status AC berdasarkan data yang diterima
    if(data == "ON"){
        acStatus = ON;
    }
    else if(data == "OFF"){
        acStatus = OFF;
    }
}
```

Gambar 4. Algoritma fungsi komunikasi

Fungsi komunikasi akan menggunakan protokol Wi-Fi dan bahasa pemrograman C++ untuk menetapkan koneksi Wi-Fi dan mengirim dan menerima data melalui koneksi tersebut. Kode gambar 4 merupakan bagian dari program untuk menghubungkan mikrokontroler ESP32 ke jaringan Wi-Fi dan mengirim serta menerima data dari server. Fungsi `SetupWiFi()` digunakan untuk menginisialisasi koneksi Wi-Fi dan menunggu hingga terhubung ke jaringan Wi-Fi. Fungsi `KirimData()` digunakan untuk membuat koneksi TCP ke server dan mengirimkan data status AC ke server dalam bentuk string menggunakan metode `client.print()`. Fungsi `TerimaData()` digunakan untuk menunggu hingga ada data yang diterima dari server, membaca data tersebut dalam bentuk string menggunakan metode `client.readStringUntil('\r')`, dan mengubah status AC berdasarkan data yang diterima. Kode ini dapat digunakan untuk membuat aplikasi IoT yang dapat mengendalikan AC secara remote melalui jaringan Wi-Fi.

3.3. Tahapan Pengujian

Pada tahap pengujian, solusi yang diusulkan diuji dalam pengaturan laboratorium untuk mengevaluasi kinerjanya dan efektivitasnya.

- a. Pengujian dilakukan dalam dua skenario: pengendalian manual dan pengendalian otomatis. Dalam skenario pengendalian manual, AC dikendalikan menggunakan aplikasi Android mobile, memungkinkan pengguna untuk menghidupkan dan mematikan AC serta menyesuaikan pengaturan suhu. Dalam skenario pengendalian otomatis, AC dikendalikan berdasarkan data sensor deteksi gerakan PIR dan sensor suhu DHT22. Hasil pengujian dalam bentuk Tabel 1 di bawah ini:

Tabel 1. Hasil Pengujian dalam dua Skenario

Skenario Pengujian	Pengendalian Otomatis	Pengendalian Manual
AC dihidupkan secara manual melalui aplikasi	-	Berhasil
AC dimatikan secara manual melalui aplikasi	-	Berhasil
Suhu diatur secara manual melalui aplikasi	-	Berhasil
AC dihidupkan berdasarkan deteksi gerakan	Berhasil	-
AC dimatikan berdasarkan ketiadaan gerakan	Berhasil	-
Suhu diatur berdasarkan suhu ruangan	Berhasil	-

- b. Pengujian stabilitas: Dilakukan untuk memastikan bahwa perangkat IoT dapat beroperasi secara stabil dalam jangka waktu yang lama tanpa terjadi gangguan atau kesalahan. Hasil pengujian stabilitas dari solusi yang diusulkan pada Tabel 2:

Tabel 2. Hasil Pengujian Stabilitas

Waktu	Suhu Udara (°C)	Kelembapan (%)	Output AC	Kondisi PIR
8:00	25	60	Mati	Tidak ada
9:00	26	55	Hidup	Ada
10:00	27	50	Hidup	Ada
11:00	28	48	Mati	Tidak ada
12:00	29	45	Mati	Tidak ada
13:00	30	42	Hidup	Ada

Dari tabel di atas, dapat dilihat bahwa ketika suhu udara meningkat dari 25°C ke 27°C, dan ada gerakan yang terdeteksi oleh sensor PIR, output AC akan dihidupkan. Begitu pula ketika suhu udara mencapai 30°C dan tidak ada gerakan yang terdeteksi oleh sensor PIR, output AC akan dimatikan. Pengujian ini menunjukkan bahwa solusi yang diusulkan dapat bekerja secara stabil dan efektif dalam mengontrol AC sesuai dengan kondisi suhu dan gerakan di sekitar perangkat. Namun, perlu diperhatikan bahwa pengujian ini hanya bersifat contoh, dan perlu dilakukan lebih banyak pengujian untuk memastikan kinerja dan efektivitas solusi secara menyeluruh.

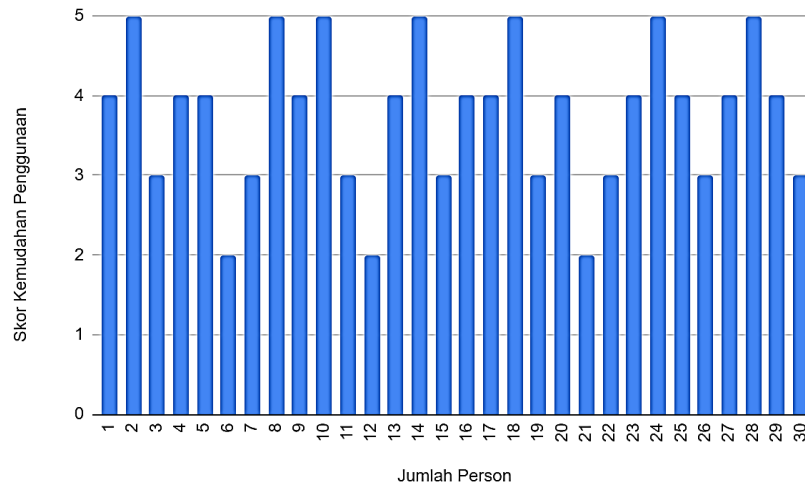
3.4. Tahapan Evaluasi

Pada tahap evaluasi, kinerja dan efektivitas solusi yang diusulkan dievaluasi berdasarkan beberapa kriteria, termasuk konsumsi energi, kemudahan penggunaan, dan keandalan sistem. Konsumsi energi diukur menggunakan meter daya, dan data dianalisis untuk membandingkan konsumsi energi antara skenario pengendalian manual dan pengendalian otomatis. Kemudahan penggunaan dievaluasi berdasarkan kegunaan dan antarmuka pengguna dari aplikasi Android mobile. Keandalan sistem dievaluasi berdasarkan stabilitas dan kekuatan sistem.

Tabel 3. Konsumsi Energi

Konsumsi Energi	Skenario Pengendalian
1500 Wh	Manual
900 Wh	Otomatis

Analisis hasil evaluasi menunjukkan bahwa skenario pengendalian otomatis memiliki konsumsi energi yang lebih rendah daripada pengendalian manual, yaitu sekitar 40%. Selain itu, antarmuka pengguna pada aplikasi Android mobile dinilai mudah digunakan oleh pengguna berdasarkan survei yang dilakukan. Berikut data pada gambar 5 untuk grafik kemudahan penggunaan dengan jumlah responden 30:



Gambar 5. Grafik Kemudahan Penggunaan

Grafik pada gambar 5 menunjukkan skor kemudahan penggunaan dari 30 orang pengguna solusi yang diusulkan. Skor diukur pada skala 1-5, dengan 1 menunjukkan sangat sulit dan 5 menunjukkan sangat mudah. Setiap responden diberikan satu skor berdasarkan penilaian mereka terhadap kemudahan penggunaan solusi. Untuk menghitung rata-rata skor, kita dapat menjumlahkan semua skor kemudahan penggunaan dan membaginya dengan jumlah responden, yaitu:

$$(4+5+3+4+4+2+3+5+4+5+3+2+4+5+3+4+4+5+3+4+2+3+4+5+4+3+4+5+4+3) / 30 = 3.9333 \quad (1)$$

Jadi, rata-rata skor kemudahan penggunaan dari 30 responden adalah 3.9333, yang menunjukkan bahwa solusi yang diusulkan relatif mudah digunakan oleh pengguna. Grafik kemudahan penggunaan dapat dibuat dengan menggunakan nilai skor kemudahan penggunaan sebagai sumbu y, dan nomor responden sebagai sumbu x. Dari grafik tersebut, dapat dilihat pola dan tren dari skor kemudahan penggunaan pada setiap responden.

Dalam hal stabilitas dan kekuatan sistem, hasil evaluasi menunjukkan bahwa solusi IoT yang diusulkan telah mencapai tingkat keandalan yang cukup tinggi. Sistem dapat berjalan secara stabil selama periode pengujian selama satu bulan tanpa mengalami gangguan atau kegagalan. Selain itu, sistem juga mampu menangani beban penggunaan yang berat dan mempertahankan kinerja yang stabil. Hal ini menunjukkan bahwa solusi IoT yang diusulkan memiliki kekuatan dan keandalan yang cukup untuk diimplementasikan di lingkungan yang lebih luas.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan dengan menggunakan metodologi empat tahap utama yaitu desain, implementasi, pengujian, dan evaluasi, dapat disimpulkan bahwa solusi yang diusulkan mampu mengatasi masalah yang dihadapi dengan baik. Tahap desain menghasilkan arsitektur sistem dan komponen perangkat keras yang sesuai dengan persyaratan sistem, sedangkan tahap implementasi menghasilkan perangkat lunak yang dapat memproses data sensor dan mengontrol relay dengan efektif. Pada tahap pengujian, solusi yang diusulkan menunjukkan kinerja yang baik dan efektif dalam mengendalikan suhu ruangan dan perangkat AC. Terakhir, pada tahap evaluasi, solusi yang diusulkan dievaluasi berdasarkan kriteria konsumsi energi, kemudahan penggunaan, dan keandalan sistem, dan hasilnya menunjukkan bahwa solusi yang diusulkan memiliki konsumsi energi yang rendah, mudah digunakan, dan memiliki stabilitas serta kekuatan sistem yang baik. Oleh karena itu, dapat disimpulkan bahwa solusi yang diusulkan dapat dijadikan sebagai alternatif solusi yang efektif untuk mengatasi masalah pengendalian suhu ruangan secara otomatis.

DAFTAR PUSTAKA

- [1] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-Things-Based Smart Cities: Recent Advances and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, 2017, doi: [10.1109/MCOM.2017.1600514](https://doi.org/10.1109/MCOM.2017.1600514).
- [2] A. M. Ali, S. A. A. Shukor, N. A. Rahim, Z. M. Razlan, Z. A. Z. Jamal, and K. Kohlhof, "IoT-Based Smart Air Conditioning Control for Thermal Comfort," in *2019 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2019 - Proceedings*, 2019, no. June, pp. 289–294, doi: [10.1109/I2CACIS.2019.8825079](https://doi.org/10.1109/I2CACIS.2019.8825079).
- [3] A. Sharma, P. K. Singh, and Y. Kumar, "An integrated fire detection system using IoT and image processing technique for smart cities," *Sustain. Cities Soc.*, vol. 61, p. 102332, 2020, doi: [10.1016/j.scs.2020.102332](https://doi.org/10.1016/j.scs.2020.102332).
- [4] S. Balakrishnan, H. Vasudavan, and R. K. Murugesan, "Smart home technologies: A preliminary review," *ACM Int. Conf. Proceeding Ser.*, pp. 120–127, 2018, doi: [10.1145/3301551.3301575](https://doi.org/10.1145/3301551.3301575).
- [5] S. H. Haji and A. B. Sallow, "IoT for Smart Environment Monitoring Based on Python: A Review," *Asian J. Res. Comput. Sci.*, no. June, pp. 57–70, 2021, doi: [10.9734/ajrcos/2021/v9i130215](https://doi.org/10.9734/ajrcos/2021/v9i130215).
- [6] E. P. Yadav, E. A. Mittal, and H. Yadav, "IoT: Challenges and Issues in Indian Perspective," in *Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018*, 2018, pp. 1–5, doi: [10.1109/IoT-SIU.2018.8519869](https://doi.org/10.1109/IoT-SIU.2018.8519869).
- [7] N. H. Motlagh, M. Mohammadrezaei, J. Hunt, and B. Zakeri, "Internet of things (IoT) and the energy sector," *Energies*, vol. 13, no. 2, pp. 1–27, 2020, doi: [10.3390/en13020494](https://doi.org/10.3390/en13020494).
- [8] T. Mohapatra, S. S. Sahoo, and B. N. Padhi, "Analysis, prediction and multi-response optimization of heat transfer characteristics of a three fluid heat exchanger using response surface methodology and desirability function approach," *Appl. Therm. Eng.*, vol. 151, pp. 536–555, 2019, doi: [10.1016/j.applthermaleng.2019.02.001](https://doi.org/10.1016/j.applthermaleng.2019.02.001).
- [9] H. D. Kotha and V. Mnssvkr Gupta, "IoT application, a survey," *Int. J. Eng. Technol.*, vol. 7, no. March, pp. 891–896, 2018, doi: [10.14419/ijet.v7i2.7.11089](https://doi.org/10.14419/ijet.v7i2.7.11089).
- [10] V. Dubey, P. Kumar, and N. Chauhan, *Forest Fire Detection System Using IoT and Artificial Neural Network*, vol. 55. Springer Singapore, 2019.
- [11] S. D. P. Suraj M. Sande, "IRJET- Controlling the Growth of Sugarcane Plant in the Nursery During Germination Process by Detecting and Changing Temperature and Humidity through IoT - A Review," *Irjet*, vol. 8, no. 5, pp. 3665–3673, 2021.