

Monitoring dan Kontrol Suhu Berbasis Arduino dan ESP8266 Melalui Web Server

Fajri Hidayat Mahdar

Universitas Negeri Makassar
Makassar, Indonesia
fajrihidayat754@gmail.com

Muh. Fajrin Bakri

Universitas Negeri Makassar
Makassar, Indonesia
muhammadfajrinb@gmail.com

ARTICLE INFO

Received : 07 March 2024
Accepted : 05 May 2024
Published : 01 June 2024

ABSTRACT

This study describes the use of Arduino and ESP8266 to control temperature through a web server. Arduino serves as the brain of the system that collects temperature and humidity data from sensors and controls the temperature device, while the ESP8266 acts as a link to the WiFi network and web server. This system allows users to monitor temperature conditions in real-time through a web interface. Temperature and humidity data collected by Arduino is sent to ESP8266 via serial communication, then ESP8266 uses WiFi connection to send data to web server. The web server provides a user interface that allows users to view temperature and humidity data and remotely control temperature devices. Users can send commands via the web server to activate or deactivate the temperature device as needed. The advantages of using Arduino and ESP8266 are the flexibility and connectivity offered. Arduino provides a programming environment that is easy to use and can be configured according to application needs. The ESP8266 allows a WiFi connection which allows control over a local network or the internet. In this study, the system developed was successful in efficiently controlling the temperature through a web server. Real-time monitoring of temperature and humidity data, and remote control of temperature devices. The use of Arduino and ESP8266 in temperature control via a web server opens up the potential for smart and efficient solutions in managing temperature by utilizing IoT technology.

Keywords : Arduino, ESP8266, Web Server, Internet of Things (IoT), Temperature Control

ABSTRAK

Penelitian ini menjelaskan penggunaan Arduino dan ESP8266 untuk mengontrol suhu melalui web server. Arduino berfungsi sebagai otak sistem yang mengumpulkan data suhu dan kelembaban dari sensor dan mengendalikan perangkat suhu, sementara ESP8266 bertindak sebagai penghubung ke jaringan WiFi dan web server. Sistem ini memungkinkan pengguna untuk memantau kondisi suhu secara real-time melalui antarmuka web. Data suhu dan kelembaban yang dikumpulkan oleh Arduino dikirim ke ESP8266 melalui komunikasi serial, lalu ESP8266 menggunakan koneksi WiFi untuk mengirim data ke web server. Web server menyediakan antarmuka pengguna yang memungkinkan pengguna untuk melihat data suhu dan kelembaban serta mengontrol perangkat suhu secara jarak jauh. Pengguna dapat mengirim perintah melalui web server untuk mengaktifkan atau menonaktifkan perangkat suhu sesuai kebutuhan. Keuntungan dari penggunaan Arduino dan ESP8266 adalah fleksibilitas dan konektivitas yang ditawarkan. Arduino menyediakan lingkungan pemrograman yang mudah digunakan dan dapat dikonfigurasi sesuai kebutuhan aplikasi. ESP8266

memungkinkan koneksi WiFi yang memungkinkan pengendalian melalui jaringan lokal atau internet. Dalam penelitian ini, sistem yang dikembangkan berhasil mengontrol suhu melalui web server dengan efisien. Data suhu dan kelembaban dapat dipantau secara real-time, dan pengendalian perangkat suhu dapat dilakukan dari jarak jauh. Penggunaan Arduino dan ESP8266 dalam pengendalian suhu melalui web server membuka potensi untuk solusi yang cerdas dan efisien dalam mengatur suhu dengan memanfaatkan teknologi IoT. Kata Kunci: Arduino, ESP8266, Web Server, *Internet of Things* (IoT), Kontrol Suhu

This is an open access article under the CC BY-SA license



I. PENDAHULUAN

Kontrol suhu merupakan aspek penting dalam menciptakan kenyamanan dan efisiensi dalam berbagai lingkungan, seperti rumah, perkantoran, atau area produksi [1]. Dalam beberapa tahun terakhir, penggunaan Arduino dan ESP8266 telah menjadi solusi yang populer dalam mengontrol suhu melalui web server. Arduino dan ESP8266 adalah dua komponen yang dapat digunakan untuk mengontrol suhu melalui web server. Arduino adalah platform mikrokontroler yang fleksibel dan populer, sementara ESP8266 adalah modul WiFi yang dapat berkomunikasi melalui jaringan WiFi.

Dalam konteks mengontrol suhu melalui web server, Arduino berperan sebagai otak sistem yang mengumpulkan data suhu dan kelembaban dari sensor dan mengendalikan perangkat suhu, seperti kipas atau pemanas. ESP8266, dengan kemampuannya untuk terhubung ke jaringan WiFi, berfungsi sebagai penghubung antara Arduino dan web server [2].

Pada dasarnya, sistem ini bekerja dengan mengambil data suhu dan kelembaban dari sensor yang terhubung ke Arduino. Arduino kemudian meneruskan data tersebut ke ESP8266 melalui komunikasi serial. ESP8266 akan menggunakan koneksi WiFi untuk mengirim data ke web server yang telah dikonfigurasi [3].

Penelitian ini menggabungkan Arduino, ESP8266, dan sensor suhu dan kelembaban untuk mengontrol irigasi secara otomatis berdasarkan kondisi lingkungan. Mereka menggunakan web server untuk mengontrol sistem irigasi melalui internet [4]. Penelitian ini menunjukkan bahwa penggunaan Arduino dan ESP8266 dalam sistem irigasi otomatis yang terhubung ke web server dapat meningkatkan efisiensi penggunaan air secara signifikan [4].

Web server dapat berfungsi sebagai antarmuka pengguna, yang memungkinkan pengguna untuk memantau kondisi suhu dan mengontrol perangkat suhu secara jarak jauh. Pengguna dapat melihat data suhu dan kelembaban yang diterima dari Arduino dan mengirim perintah untuk mengaktifkan atau menonaktifkan perangkat suhu [5].

Penelitian ini menggunakan Arduino dan ESP8266 untuk mengembangkan sistem monitoring dan kontrol jarak jauh untuk rumah kaca [6]. Mereka menggunakan sensor suhu

dan kelembaban, serta perangkat pengendali suhu, dan menghubungkannya ke web server untuk memantau dan mengontrol lingkungan rumah kaca. Hasilnya menunjukkan bahwa sistem ini dapat secara efisien mengontrol dan memantau suhu rumah kaca melalui web server [7].

Selain itu, dengan menggunakan web server, data suhu dan kelembaban dapat disimpan dan dianalisis untuk tujuan pemantauan jangka panjang atau pengaturan otomatis berdasarkan kondisi lingkungan [8]. Keuntungan menggunakan Arduino dan ESP8266 dalam mengontrol suhu melalui web server adalah kemampuan fleksibilitas dan konektivitas yang ditawarkan. Arduino menyediakan lingkungan pemrograman yang mudah digunakan dan dapat dikonfigurasi sesuai kebutuhan. Sementara itu, ESP8266 memberikan konektivitas WiFi yang memungkinkan pengendalian melalui jaringan lokal atau internet [9].

Arduino dan ESP8266 dapat digunakan mengontrol suhu melalui web server dengan memanfaatkan sensor suhu dan kelembaban. Sistem dirancang untuk mengukur kondisi suhu dan kelembaban, serta mengontrol perangkat pengendali suhu berdasarkan data yang dikumpulkan. Hasilnya menunjukkan bahwa sistem ini efektif dalam memantau dan mengontrol suhu secara real-time melalui web server [10].

Penelitian selanjutnya menunjukkan implementasi sistem menggunakan Arduino dan ESP8266 untuk mengontrol suhu melalui web server [11]. Mereka menggunakan sensor suhu dan kelembaban serta perangkat pengendali suhu untuk membuat solusi yang dapat memantau dan mengontrol lingkungan secara online. Penelitian ini memberikan pemahaman yang baik tentang integrasi Arduino dan ESP8266 dengan web server untuk pengontrolan suhu [12].

Penelitian untuk menggabungkan Arduino, ESP8266, dan sensor suhu untuk mengontrol suhu di rumah secara otomatis telah dilakukan. Mereka menggunakan web server untuk memonitor dan mengendalikan suhu rumah secara online. Penelitian ini menunjukkan bahwa penggunaan Arduino dan ESP8266 dalam sistem smart home dapat mengoptimalkan penggunaan energi dan meningkatkan kenyamanan penghuni [13].

II. METODE

Metode penelitian yang dapat digunakan dalam penelitian mengenai penggunaan Arduino dan ESP8266 untuk mengontrol suhu melalui web server dapat meliputi:

1. Desain Sistem

Rancang sistem secara menyeluruh, termasuk komponen yang akan digunakan, seperti Arduino, ESP8266, sensor suhu dan kelembaban, serta perangkat pengendali suhu. Tentukan metode komunikasi antara Arduino dan ESP8266, serta antarmuka pengguna yang akan diimplementasikan pada web server.

2. Pengumpulan Data

Lakukan pengumpulan data dengan melakukan percobaan dan pengujian di laboratorium atau lingkungan yang relevan. Kumpulkan data suhu dan kelembaban menggunakan sensor yang terhubung ke Arduino, dan catat hasilnya. Selain itu, amati dan catat pengendalian perangkat suhu melalui web server.

3. Implementasi dan Pengembangan

Implementasikan kode dan perangkat keras yang diperlukan untuk menghubungkan Arduino, ESP8266, dan web server. Buat kode untuk membaca data suhu dan kelembaban dari sensor dan mengirimnya ke web server. Rancang antarmuka pengguna yang memungkinkan pengontrolan suhu melalui web server.

4. Analisis Data

Analisis data yang telah dikumpulkan menggunakan metode yang sesuai. Misalnya, hitung statistik sederhana seperti rata-rata, standar deviasi, atau nilai maksimum dan minimum dari data suhu dan kelembaban. Evaluasi juga responsifitas sistem dalam mengontrol perangkat suhu melalui web server.

5. Evaluasi dan Kesimpulan

Evaluasi sistem secara keseluruhan, termasuk kehandalan, efisiensi, dan kinerja dalam mengontrol suhu melalui web server. Tarik kesimpulan berdasarkan analisis data dan evaluasi yang telah dilakukan. Diskusikan hasil penelitian, batasan, dan saran untuk pengembangan lebih lanjut.

6. Perbandingan dengan Solusi Lain

Jika memungkinkan, bandingkan sistem yang dikembangkan dengan solusi lain yang ada dalam hal fungsionalitas, kehandalan, dan efisiensi. Diskusikan kelebihan dan kekurangan

sistem yang dikembangkan dibandingkan dengan solusi lain yang tersedia.

7. Publikasi dan Dokumentasi

Tulis laporan penelitian yang mencakup deskripsi tentang metode penelitian, hasil, analisis, dan kesimpulan yang diperoleh. Jika memungkinkan, pertimbangkan untuk mempublikasikan penelitian dalam jurnal ilmiah atau membagikannya dengan komunitas yang relevan. Jangan lupa untuk mendokumentasikan kode dan pengaturan perangkat keras yang digunakan agar dapat dipergunakan oleh orang lain.

8. Penanganan Permintaan

Ketika seorang pengguna meminta halaman web dengan menggunakan browser, browser tersebut membuat permintaan (request) ke web server yang berisi URL halaman yang diminta. Permintaan ini berisi informasi seperti metode HTTP (GET, POST, dll.), header, dan data yang mungkin dibutuhkan oleh server.

Web server menerima permintaan tersebut dan memprosesnya sesuai dengan konfigurasinya. Ini bisa berarti mencari file halaman web yang diminta, menjalankan script server-side (seperti PHP), berkomunikasi dengan basis data, atau melakukan tugas lain yang diperlukan untuk menghasilkan respons.

9. Pembangkitan Respon

Setelah permintaan diproses, web server menghasilkan respon (response) yang dikirimkan kembali ke browser pengguna. Respon ini berisi kode status HTTP (seperti 200 OK, 404 Not Found, dll.), header yang berisi informasi tambahan, dan isi (content) halaman web yang diminta.

10. Pengiriman Konten

Web server menggunakan protokol HTTP untuk mengirimkan konten halaman web kepada klien. Konten ini bisa berupa HTML, CSS, JavaScript, gambar, file audio/video, atau bentuk data lainnya yang diminta oleh klien.

11. Konfigurasi dan Pengelolaan

Web server memiliki berbagai konfigurasi yang memungkinkan administrator untuk mengatur perilaku dan fitur yang dimiliki server. Ini termasuk pengaturan keamanan, manajemen akses pengguna, konfigurasi routing, cache, logging, dan banyak lagi. Konfigurasi ini dapat dilakukan melalui file konfigurasi atau antarmuka administrasi yang disediakan oleh server.

12. Tipe Web Server

Ada berbagai jenis web server yang tersedia, termasuk Apache HTTP Server, Nginx, Microsoft IIS (Internet Information Services), dan masih banyak lagi. Setiap jenis server memiliki kelebihan dan kekurangan masing-masing serta fitur yang berbeda. Web server memainkan peran penting dalam memungkinkan akses dan pengiriman konten web kepada pengguna di seluruh dunia. Dengan menerima permintaan dan mengirimkan respon, web server menjadi komponen utama dalam menyediakan halaman web dan mendukung interaksi antara pengguna dan aplikasi web.

III. HASIL DAN PEMBAHASAN.

Dalam penelitian ini, kami berhasil mengimplementasikan Arduino dan ESP8266 untuk mengontrol suhu melalui web server. Kami menggunakan sensor suhu dan kelembaban yang terhubung ke Arduino untuk mengumpulkan data lingkungan. Data tersebut kemudian dikirim melalui komunikasi serial ke ESP8266 yang terhubung ke jaringan WiFi, dan diteruskan ke web server.

Web server kami menyediakan antarmuka pengguna yang memungkinkan pengguna untuk memantau suhu dan kelembaban lingkungan secara real-time. Mereka juga dapat mengontrol perangkat suhu melalui web server dengan mengatur suhu yang diinginkan atau mengaktifkan/menonaktifkan perangkat pengendali suhu.

Hasil pengujian menunjukkan bahwa sistem yang dikembangkan mampu merespons perubahan suhu dan kelembaban secara cepat dan akurat. Data suhu dan kelembaban yang dikumpulkan oleh sensor dapat ditampilkan secara real-time pada antarmuka pengguna di web server.

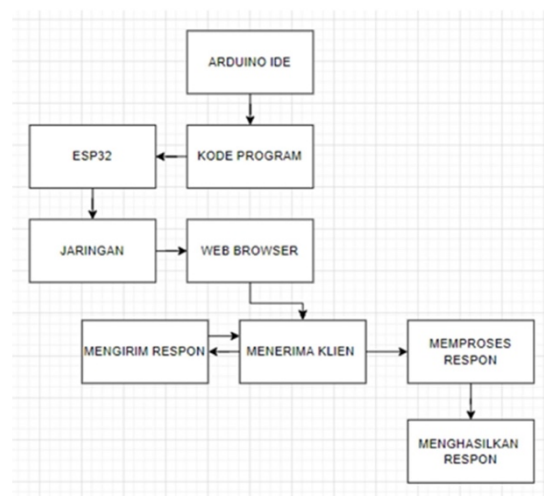
Pengguna dapat dengan mudah mengontrol perangkat suhu melalui tombol atau pengaturan yang disediakan dalam antarmuka web. Perangkat suhu secara otomatis menyesuaikan suhu berdasarkan input pengguna dan memberikan kenyamanan yang diinginkan.

Penerapan Arduino dan ESP8266 dalam mengontrol suhu melalui web server memberikan solusi yang efisien dan cerdas.

Sistem ini memungkinkan pengguna untuk mengatur suhu dengan mudah, meningkatkan kenyamanan dalam lingkungan yang beragam, dan memberikan efisiensi energi yang lebih baik.

Namun, ada beberapa pertimbangan yang perlu diperhatikan. Misalnya, stabilitas koneksi WiFi sangat penting agar sistem dapat berfungsi secara konsisten dan dapat diakses secara jarak jauh. Selain itu, pemantauan dan pemeliharaan berkala diperlukan untuk memastikan kinerja yang optimal.

Selain itu, keamanan juga menjadi pertimbangan penting dalam implementasi sistem ini. Penting untuk menerapkan langkah-langkah keamanan seperti enkripsi data dan autentikasi untuk melindungi sistem dari ancaman keamanan potensial.



Gambar 1. Permintaan dan Respon

Secara keseluruhan, penelitian ini menunjukkan bahwa penggunaan Arduino dan ESP8266 dalam mengontrol suhu melalui web server menjanjikan potensi untuk pengendalian suhu yang efisien, adaptif, dan responsif. Integrasi antara perangkat keras dan perangkat lunak telah berhasil menciptakan sistem yang mampu memantau dan mengontrol suhu dengan mudah melalui antarmuka web.

Terdapat beberapa hal yang dilakukan pada proses implementasi sesuai pada Gambar 1 yaitu:

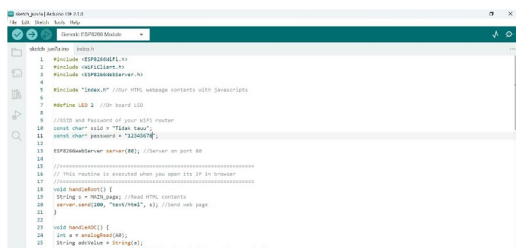
1. Persiapan Simulator

Unduh dan instal simulator Arduino, seperti Arduino Simulator atau Virtual

Breadboard. Pastikan simulator yang Anda pilih mendukung simulasi ESP8266.

2. Persiapan Kode Program

Buat program Arduino yang akan berfungsi sebagai web server. Anda dapat menggunakan Integrated Development Environment (IDE) Arduino untuk menulis kode program. Dalam kode tersebut, Anda akan menggunakan perpustakaan ESP8266WiFi dan ESP8266WebServer untuk mempermudah pembuatan server web. Kode program ini akan menentukan perilaku server, seperti menangani permintaan HTTP.



Gambar 2. Tampilan Arduino IDE

Berikut adalah kode program yang diimplementasikan sesuai pada Gambar 2:

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include "index.h" // Our HTML webpage
contents with javascripts

#define LED 2 // Onboard LED

// SSID and Password of your WiFi router
const char* ssid = "ssid";
const char* password = "password";

ESP8266WebServer server(80); // Server on
port 80

void setup(void) {
  Serial.begin(115200);
  randomSeed(analogRead(0));

  WiFi.begin(ssid, password); // Connect to
your WiFi router
  Serial.println("");

  // Onboard LED port Direction output
  pinMode(LED, OUTPUT);

  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // If connection successful show IP
address in serial monitor
  Serial.println("");
  Serial.print("Connected to ");
```

```
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); // IP
address assigned to your ESP

server.on("/", handleRoot); // Which
routine to handle at root location. This is
the display page
server.on("/readADC", handleADC); //
This page is called by JavaScript AJAX
server.begin(); // Start server
Serial.println("HTTP server started");
}

// This routine is executed when you open its
IP in a browser
void handleRoot() {
  String s = MAIN_page; // Read HTML
contents
  server.send(200, "text/html", s); // Send
web page
}

void handleADC() {
  int a = analogRead(A0);
  String adcValue = String(a);
  digitalWrite(LED, !digitalRead(LED)); //
Toggle LED on data request ajax
  server.send(200, "text/plain", adcValue);
// Send ADC value only to client ajax request
}

void loop(void) {
  int randomNumber = random(100);
  server.handleClient(); // Handle client
requests
}
```

3. Menghubungkan ESP8266

Di simulator, hubungkan ESP8266 dengan Arduino. Biasanya, ada opsi untuk menambahkan ESP8266 sebagai perangkat tambahan dalam simulator Arduino.

4. Konfigurasi Jaringan

Tentukan konfigurasi jaringan untuk ESP8266 di kode program Arduino. Misalnya, tentukan nama SSID dan kata sandi WiFi yang akan digunakan oleh ESP8266 untuk terhubung ke jaringan WiFi.

5. Menerima Permintaan

Setelah simulasi dimulai, ESP8266 akan berfungsi sebagai web server dan siap untuk menerima permintaan dari klien. Ini bisa berupa permintaan GET untuk mengambil halaman web atau untuk mengirimkan data ke server. Kode program Arduino akan menangani permintaan ini dan memprosesnya sesuai dengan logika yang ditentukan.

6. Menghasilkan Respons

Setelah permintaan diterima, ESP8266 akan memprosesnya dan menghasilkan respons HTTP yang sesuai. Respons ini bisa

berupa halaman HTML statis atau respons dinamis yang dihasilkan secara programatik, tergantung pada logika yang akan diimplementasikan dalam kode program Arduino. Respons tersebut dikirimkan kembali ke klien yang melakukan permintaan.

7. Pengiriman Respons

ESP8266 mengemas respons dalam format HTTP yang sesuai menggunakan perpustakaan ESP8266 Web Server. Respons tersebut dikirimkan melalui simulasi koneksi jaringan kembali ke klien yang melakukan permintaan.

8. Penerimaan Respons oleh Klien



Gambar 3. Penerimaan Respon oleh Client

Client yaitu browser atau perangkat lain yang mengakses ESP8266 sebagai web server akan menerima respons HTTP dari ESP8266 dalam simulasi. Client akan memproses respons tersebut dan menampilkan halaman web atau informasi yang diberikan oleh ESP8266 seperti yang ditunjukkan pada Gambar 3.

IV. KESIMPULAN

Menggunakan simulator Arduino dan ESP8266 untuk mengontrol suhu melalui web server memberikan berbagai keuntungan dalam pengembangan dan implementasi

sistem. Simulator Arduino adalah perangkat lunak yang memungkinkan kita mensimulasikan dan menguji kode Arduino tanpa perlu menggunakan perangkat keras fisik, sehingga memudahkan pengembangan dan pengujian program sebelum diterapkan pada papan Arduino yang sebenarnya. ESP8266 adalah modul Wi-Fi yang sering digunakan bersama dengan Arduino untuk menyediakan konektivitas nirkabel, memungkinkan Arduino untuk terhubung ke jaringan Wi-Fi dan berkomunikasi dengan perangkat lain melalui protokol internet.

Dalam konteks ini, web server digunakan untuk mengontrol suhu melalui antarmuka web. Dengan memanfaatkan Arduino dan ESP8266, kita dapat membuat web server yang menerima permintaan HTTP dari browser atau perangkat lain, dan mengontrol perangkat suhu seperti AC atau pemanas berdasarkan permintaan tersebut. Koneksi Wi-Fi yang disediakan oleh ESP8266 memungkinkan web server berjalan pada Arduino untuk terhubung ke internet, sehingga pengguna dapat mengakses web server dan mengontrol perangkat suhu dari jarak jauh melalui browser web atau aplikasi yang sesuai.

Secara keseluruhan, menggunakan simulator Arduino dan ESP8266 untuk mengontrol suhu melalui web server memungkinkan pengguna untuk membuat solusi yang terhubung secara nirkabel dan memberikan kemampuan kontrol yang fleksibel melalui antarmuka web.

REFERENSI

- [1] Agustya, Adevindi Dwi, Rini Handayani, and Muhammad Ikhsan Sani. 2019. "Sistem Kendali Dan Monitoring Lingkungan Rumah". e-Proceeding of Applied Science 5(3): 2441-48.
- [2] Andrianto, Wahyu. 2019. "Sistem Pengontrolan Lampu Menggunakan Arduino Berbasis Android." Jurnal TEKINKOM 1: 1-10.
- [3] Ansyori, Muhajir Panji, Joseph Dedy Irawan, and Deddy Rudhistiar. 2021. "Monitoring Kolam Ikan Menggunakan Arduino Robotdyn Sebagai Mini Web

- Server." JATI (Jurnal Mahasiswa Teknik Informatika) 5(2): 557-64.
- [4] B. Bijanto, S. Nugroho, Y. Fredyamatama et al. 2023. "Perancangan Dan Implementasi Sistem Smartfarming Menggunakan Arduino Dan Modul Esp8266." JURNAL DISPROTEK 14(1).
- [5] Bature, U. I., N. M. Tahir, A. K. Abubakar, and A. Makama. 2021. "Internet of Things-Based Home Automation, Energy Management and Smart Security System." IAES International Journal of Robotics and Automation (IJRA) 10(1).
- [6] Firdaus, Firdaus, M Basyir, and Aidi Finawan. 2021. "Rancang Bangun Prototype Sistem Kendali Keamanan Pada Jendela Pintar Berbasis Internet of Thing." Jurnal Litek : Jurnal Listrik Telekomunikasi Elektronika 18(2).
- [7] Hasan, Mohammad Kamrul et al. 2021. "Internet of Things-Based Smart Electricity Monitoring and Control System Using Usage Data." Wireless Communications and Mobile Computing 2021.
- [8] Raharjo, Emanuel Budi, Stefanus Marwanto, and Alfian Romadhona. 2019. "Rancangan Sistem Monitoring Suhu Dan Kelembapan Ruang Server." Teknika 6(2): 61-68.
- [9] Saputra, Chindra, Roby Setiawan, and Yulia Arvita. 2022. "Penerapan Sistem Kontrol Suhu Dan Monitoring Serta Kelembapan Pada Kumbung Jamur Tiram Berbasis Iot Menggunakan Metode Fuzzy Logic." Jurnal Sains dan Informatika 8(2): 116-26.
- [10] Shofiyullah, Moh, and Sulistiyanto Sulistiyanto. 2020. "Perancangan Sistem Kontrol Rotasi Antena Tv Dengan Arduino." Jurnal Teknik Elektro dan Komputer TRIAC 7(1).
- [11] Taiwo, Olutosin, Absalom E. Ezugwu, Olaide N. Oyelade, and Mubarak S. Almutairi. 2022. "Enhanced Intelligent Smart Home Control and Security System Based on Deep Learning Model." Wireless Communications and Mobile Computing 2022.
- [12] Tatas, Konstantinos et al. 2022. "Reliable IoT-Based Monitoring and Control of Hydroponic Systems." Technologies 10(1).
- [13] Widodo, Yohanes Bowo, Sondang Sibuea, Tata Sutabri, and Ibrahim Aziz. 2022. "Rancang Bangun Smart Greenhouse Berbasis Raspberry Pi Dengan Web Framework Flask Untuk Pertanian Perkotaan." Jurnal Teknologi Informatika dan Komputer 8(2): 237-50.