

Rancang Bangun Sistem Pemantauan Real-Time Ruang Server dengan Integrasi Telegram Bot Dan Esp32

Muhammad Afrizal Firdaus¹, Nugroho Budhisantosa²

^{1,2} Universitas Esa Unggul

e-mail: afrizalfirdaus2823@gmail.com¹, nugroho.budhisantosa@esaunggul.ac.id²

Abstrak

Ruang Ruang server merupakan lingkungan yang sangat sensitif terhadap perubahan suhu dan kelembapan. Pemantauan kondisi ruangan secara manual sering kali tidak efisien dan memakan waktu. Oleh karena itu, penelitian ini bertujuan untuk merancang sistem pemantauan real-time berbasis Internet of Things (IoT) menggunakan ESP32 dan Telegram Bot untuk meningkatkan efisiensi dan keamanan ruang server. Sistem ini menggunakan sensor DHT11 untuk mengukur suhu dan kelembapan, sensor MQ-2 untuk mendeteksi asap atau gas, serta sensor api untuk mendeteksi keberadaan nyala api. Data dari sensor diproses oleh ESP32 dan dikirimkan sebagai notifikasi real-time melalui Telegram, memungkinkan pengguna memantau kondisi ruangan dari mana saja dan kapan saja. Pengujian menunjukkan bahwa sistem ini dapat memberikan notifikasi secara real-time saat parameter lingkungan melebihi ambang batas yang telah ditentukan. Selain itu, sistem dapat mengaktifkan perangkat tambahan secara otomatis, seperti kipas (prototipe dari AC) ketika suhu terlalu tinggi, lampu hijau (prototipe pemanas ruangan) ketika kelembapan melebihi batas, dan lampu merah (prototipe pemadam api) ketika terdeteksi api. Dengan fitur otomatisasi ini, sistem tidak hanya memberikan perlindungan tambahan melalui deteksi dini kebakaran, tetapi juga meningkatkan efisiensi operasional dengan mengurangi kebutuhan pemantauan manual.

Kata Kunci: *Internet of Things, ESP32, Telegram Bot, Ruang Server, Pemantauan Real-time, Deteksi Kebakaran*

Abstract

A server room is a highly sensitive environment affected by changes in temperature and humidity. Manual monitoring of room conditions is often inefficient and time-consuming. Therefore, this study aims to design a real-time monitoring system based on the Internet of Things (IoT) using ESP32 and Telegram Bot to enhance the efficiency and security of the server room. The system utilizes a DHT11 sensor to measure temperature and humidity, an MQ-2 sensor to detect smoke or harmful gases, and a flame sensor to detect the presence of fire. Data from these sensors is processed by the ESP32 and sent as real-time notifications via Telegram, allowing users to monitor room conditions anytime and anywhere. Testing has shown that the system provides real-time notifications when environmental parameters exceed predefined thresholds. Additionally, the system can automatically activate supplementary devices, such as a fan (prototype of an AC) when the temperature is too high, a green light (prototype of a room heater) when humidity exceeds the limit, and a red light (prototype of a fire extinguisher) when fire is detected. With these automation features, the system not only provides additional protection through early fire detection but also improves operational efficiency by reducing the need for manual monitoring.

Keywords: *Internet of Things, ESP32, Telegram Bot, Server Room, Real-time Monitoring, Fire Detection*

PENDAHULUAN

Internet of Things (IoT) adalah teknologi inovatif yang mengintegrasikan perangkat fisik dengan sistem digital melalui jaringan internet. Hal ini memungkinkan perangkat untuk saling berkomunikasi, berbagi data, dan bekerja secara otomatis untuk memenuhi berbagai kebutuhan

manusia secara efisien (Al-Fuqaha et al., 2015). IoT telah memberikan dampak signifikan dalam banyak sektor, salah satunya dalam pemantauan kondisi lingkungan seperti suhu dan kelembapan. Menurut Ahmed (2020), pemantauan suhu dan kelembapan sangat penting untuk menjaga kestabilan kondisi lingkungan, terutama di ruang server yang menyimpan perangkat sensitif terhadap perubahan suhu dan kelembapan. Dalam konteks ini, sistem pemantauan berbasis IoT memungkinkan pengawasan secara real-time yang sangat berguna untuk menjaga agar ruang server tetap dalam kondisi yang optimal.

Ruang server di Indonesia memiliki standar suhu ruang server antara 21–27°C dan kelembapan antara 45% hingga 70% (Ichsan, 2023). Penurunan atau peningkatan suhu yang terlalu ekstrim dapat menyebabkan kerusakan pada perangkat keras, seperti overheating yang dapat merusak sistem komputer dan jaringan. Menurut Zhang et al. (2018), suhu yang tidak stabil dapat memengaruhi kinerja perangkat keras dalam ruang server, dan pada akhirnya dapat mengakibatkan kegagalan sistem yang berdampak pada operasional bisnis. Oleh karena itu, penting untuk menggunakan sistem pemantauan yang dapat mendeteksi suhu dan kelembapan secara real-time.

Selain itu, ruang server juga rentan terhadap ancaman lain seperti kebakaran. Oleh karena itu, sistem pemantauan suhu dan kelembapan harus dilengkapi dengan sensor tambahan seperti sensor gas atau asap dan sensor api. Sebagaimana disarankan oleh Zhang dan Xiao (2020), penggunaan sensor gas seperti MQ-2 untuk mendeteksi asap atau kebocoran gas sangat penting untuk mendeteksi kebakaran lebih dini. Sensor api yang mendeteksi adanya nyala api melalui cahaya inframerah juga sangat diperlukan untuk memberikan peringatan dini pada potensi kebakaran.

Mikrokontroler ESP32 dipilih sebagai pengendali utama dalam sistem ini. ESP32 memiliki banyak keunggulan seperti fitur Wi-Fi dan Bluetooth, serta konsumsi daya yang rendah. Menurut Chen et al. (2019), ESP32 memiliki kemampuan komputasi yang cukup tinggi dan mendukung berbagai protokol komunikasi, termasuk I2C, SPI, dan UART. Kelebihan lainnya adalah kemampuannya untuk terhubung dengan berbagai sensor dan perangkat IoT lainnya dengan efisien. Dengan menggunakan ESP32, sistem ini dapat mengintegrasikan sensor DHT11 untuk suhu dan kelembapan, sensor MQ2 untuk gas/asap, dan sensor api untuk mendeteksi api dengan efektif.

Sistem ini dirancang untuk memberikan notifikasi secara real-time kepada pengguna menggunakan Telegram Bot. Menurut Li et al. (2020), Telegram Bot merupakan pilihan yang tepat untuk aplikasi IoT karena kemudahannya dalam pengirimannya, serta dukungannya terhadap berbagai platform. Selain itu, Telegram juga menawarkan kelebihan dalam hal keamanan, karena menggunakan enkripsi end-to-end untuk komunikasi antar perangkat. Pengguna dapat menerima laporan mengenai kondisi ruang server kapan saja dan di mana saja, sehingga memungkinkan mereka untuk mengambil tindakan yang cepat jika ada perubahan kondisi yang kritis.

Menurut pendapat dari IoT expert, Kumar et al. (2021), dengan integrasi berbagai sensor ke dalam satu sistem berbasis IoT, ruang server dapat dimonitor secara menyeluruh dan proaktif. Dalam hal ini, penggunaan sistem berbasis IoT untuk pemantauan suhu, kelembapan, gas, dan api sangat membantu dalam mencegah terjadinya kerusakan pada perangkat yang ada di ruang server. Teknologi ini juga mempermudah manajemen ruang server dengan cara yang lebih efisien, dengan memungkinkan monitoring dari jarak jauh melalui platform seperti Telegram.

Dengan sistem pemantauan berbasis IoT ini, ruang server akan terjaga dalam kondisi ideal, mengurangi risiko overheat, dan memberikan peringatan dini terhadap potensi kebakaran. Sistem ini tidak hanya menjaga kestabilan perangkat keras, tetapi juga memastikan keamanan ruang server dan mengurangi risiko kerusakan yang dapat berdampak pada operasional perusahaan. Selain itu, pengelolaan ruang server juga akan lebih efisien karena data dapat dipantau secara real-time. Dengan adanya sistem ini, manajer IT atau teknisi dapat melakukan tindakan preventif lebih cepat, yang pada gilirannya meningkatkan efisiensi operasional dan mengurangi biaya perawatan.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pemantauan real-time pada ruang server Universitas Esa Unggul Bekasi. Sistem ini dirancang agar tetap dapat memantau kondisi ruangan meskipun penjaga tidak berada di lokasi. Menggunakan ESP32,

sensor DHT11, sensor MQ2, sensor api, dan Telegram Bot, sistem ini diharapkan mampu meningkatkan efisiensi pemantauan sekaligus memastikan keamanan operasional perangkat di ruang server. Oleh karena itu, penelitian ini diusung dengan judul “Rancang Bangun Sistem Pemantauan Real-time Ruang Server dengan Integrasi Telegram Bot dan ESP32.”

METODE

Metode yang digunakan dalam penelitian ini mengadopsi Sistem Development Life Cycle (SDLC), yang terdiri dari beberapa tahap penting: analisa, perancangan, implementasi, dan pengujian. SDLC adalah pendekatan sistematis untuk pengembangan perangkat lunak yang telah diakui oleh para ahli sebagai metode yang efisien dan terstruktur (Pressman, 2014). Pada tahap analisa, pengumpulan data dilakukan melalui studi literatur yang melibatkan buku, jurnal ilmiah, dan referensi lainnya. Langkah ini berfungsi untuk menyediakan dasar teori dan pemahaman yang diperlukan dalam merancang sistem, serta meninjau solusi yang sudah ada. Seperti yang dijelaskan oleh Sommerville (2011), tahap analisa adalah langkah awal yang penting karena memastikan pemahaman yang mendalam terhadap masalah yang akan diselesaikan, serta membantu dalam merumuskan persyaratan sistem.

Pada perancangan, alat bantu seperti Figma dan Visual Paradigm digunakan untuk membuat flowchart dan rangkaian komponen, yang bertujuan untuk menerjemahkan kebutuhan sistem menjadi desain perangkat yang siap diimplementasikan. Pendekatan ini sejalan dengan pendapat McConnell (2004) yang menyatakan bahwa perancangan sistem yang baik akan mempermudah implementasi dan pengujian. Implementasi mencakup pengintegrasian sensor dengan modul ESP32 serta penerapan kode program, yang memungkinkan sistem untuk bekerja secara otomatis, mengirimkan data ke platform Telegram, dan memberi notifikasi secara real-time. Menurut Pressman (2014), tahap implementasi adalah saat di mana sistem diuji coba dengan komponen-komponen yang terintegrasi untuk memastikan bahwa setiap bagian dapat berfungsi sesuai rencana.

Pada pengujian, dilakukan pemeriksaan menyeluruh untuk memastikan setiap perangkat berfungsi dengan baik, termasuk sensor, konektivitas internet, dan aplikasi Telegram. Pengujian sistem ini sangat penting untuk memastikan bahwa data yang dikirimkan akurat dan tepat waktu. Sebagaimana dijelaskan oleh Beizer (1990), pengujian harus dilakukan pada setiap tahap pengembangan untuk memastikan kualitas sistem dan mengurangi potensi kesalahan saat sistem diterapkan.

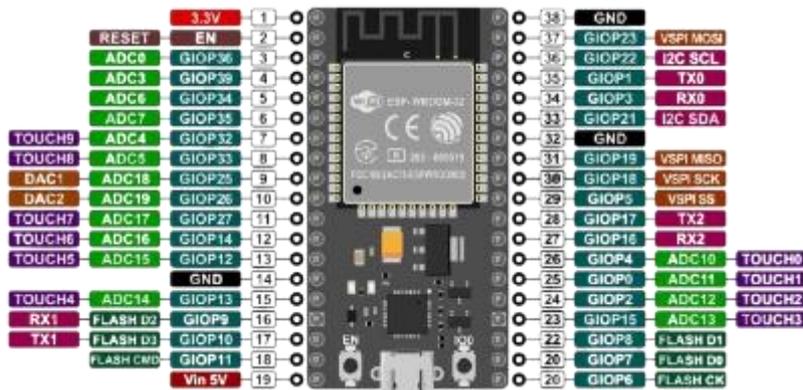
Sistem yang dirancang bertujuan untuk memantau suhu, kelembapan, keberadaan gas atau asap, dan api di ruang server secara real-time. Sistem ini menggunakan ESP32 untuk mengolah data dari sensor DHT11 (suhu dan kelembapan), sensor MQ-2 (gas/asap), dan sensor api. Ketika parameter lingkungan mencapai batas kritis, sistem akan memberikan notifikasi melalui Telegram dan dapat mengaktifkan perangkat seperti kipas atau lampu untuk menjaga kestabilan lingkungan. Ini sesuai dengan pendapat Hennessey (2006) bahwa sistem yang menggunakan perangkat terhubung (IoT) dapat membantu pemantauan secara real-time dan memberikan tindakan otomatis.

Pada perancangan sistem, sensor-sensor utama dihubungkan dengan ESP32 yang akan memproses data dari sensor dan mengirimkan hasilnya ke Telegram. Sistem ini mengandalkan komunikasi Wi-Fi yang memungkinkan data untuk dipantau secara real-time. Sensor DHT11 dipilih karena akurasinya yang baik dalam mengukur suhu dan kelembapan. Sensor MQ-2 digunakan untuk mendeteksi gas berbahaya, sedangkan sensor api mampu mendeteksi keberadaan api dengan pancaran cahaya inframerah. Seperti yang disarankan oleh Sauter (2011), penggunaan sensor-sensor yang tepat merupakan kunci dalam menciptakan sistem pemantauan yang efektif dan andal.

Pada prinsip kerja ESP32, data yang diterima dari sensor diproses oleh ESP32 sesuai dengan program yang telah ditulis. Hasil pemrosesan ini akan menentukan apakah tindakan tertentu perlu dilakukan, seperti mengaktifkan relay untuk mengontrol perangkat atau mengirimkan notifikasi kepada pengguna melalui Telegram. ESP32 memiliki lebih dari 30 pin I/O yang dapat disesuaikan dengan kebutuhan, serta mendukung konektivitas Wi-Fi dan Bluetooth untuk pengiriman data secara real-time. Menurut Rojas (2017), penggunaan ESP32 sangat

menguntungkan dalam proyek-proyek yang membutuhkan konektivitas yang stabil dan kemampuan pengolahan data yang cepat.

Konfigurasi pin ESP32 mencakup penggunaan pin GPIO yang fleksibel, memungkinkan integrasi sensor dan aktuator. ESP32 beroperasi dengan tegangan 3,3V dan mendukung berbagai protokol komunikasi seperti UART, SPI, dan I2C, menjadikannya pilihan ideal untuk sistem pemantauan lingkungan ruang server. Pemilihan ESP32 sebagai pusat kendali dan pengolahan data ini didukung oleh pendapat Yick et al. (2010) yang menjelaskan bahwa ESP32 menawarkan fleksibilitas tinggi dengan biaya yang relatif rendah, menjadikannya sangat efisien untuk aplikasi Internet of Things (IoT).



Gambar 1. Konfigurasi Pin ESP32

HASIL DAN PEMBAHASAN

Implementasi Dan Pengujian Membuat Telegram Bot

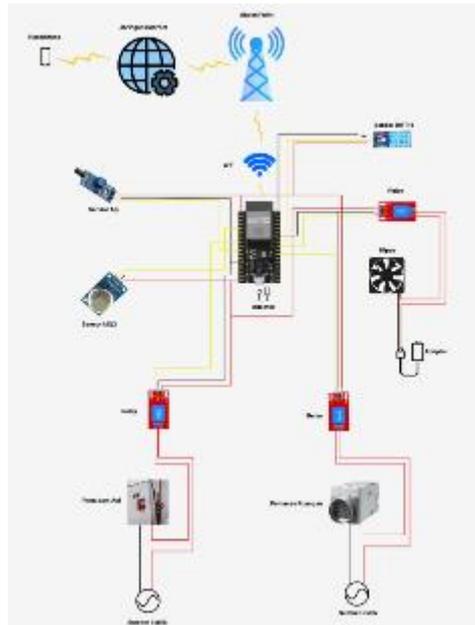
1. Buka aplikasi Telegram
2. Klik tanda search
3. Masukkan kalimat BotFather pada pencarian
4. Klik Yang paling atas dan bercentang biru
5. Klik mulai
6. Klik /newbot
7. Buat nama bot
8. Copy Bot Api (7793071850:AAH1tdCWvAl_-4OX2FcjnPiBdY4KSe1Tg_c)
9. Untuk melihat bot nya klik link yang paling atas
10. Bot telegram berhasil dibuat

Mengetahui IDBot

1. Buka aplikasi Telegram
2. Klik tanda search
3. Masukkan kalimat IDbot pada pencarian
4. Klik yang paling atas
5. Klik mulai
6. Maka id bot akan muncul (6939800600)

Rangkaian Keseluruhan Sistem

Rangkaian keseluruhan dalam sistem pemantauan ruangan server terdiri dari beberapa komponen utama, yaitu ESP32, sensor DHT11, sensor MQ2, sensor Api, Relay, Kipas dan Adaptor. ESP32 bertindak sebagai pusat kendali utama pada sistem ini. Untuk lebih jelasnya keseluruhan rangkaian terlihat pada gambar 2



Gambar 2. Rangkaian Keseluruhan Sistem

Download dan Install Aplikasi Arduino IDE

1. Buka Google Chrome (Bisa menggunakan platform lain)
2. Cari "Arduino.cc" pada search bar, lalu enter
3. Klik pada navbar "Software"
4. Lalu pilih download sesuai sistem operasi yang kita miliki dan tunggu hingga selesai
5. Buka file explorer ke bagian download, maka file yang baru didownload akan muncul, klik 2 kali pada file untuk melakukan penginstalan
6. Lakukan penginstalan dan Tunggu hingga proses penginstalan selesai

Penginstalan library dan proses upload ke esp 32

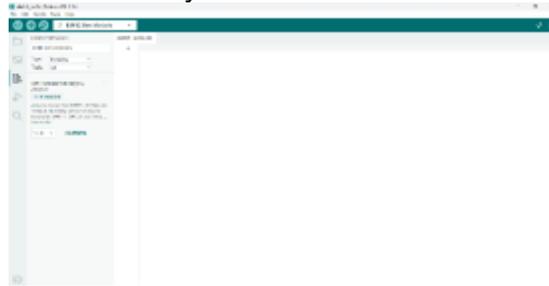
1. Buka Aplikasi Arduino IDE yang telah diinstal tadi



2. Ke navbar "sketch" → "include library" → "manage library"



3. Lakukan pencarian “DHT sensor library” lalu install



4. Lakukan pencarian “UniversalTelegramBot” lalu install

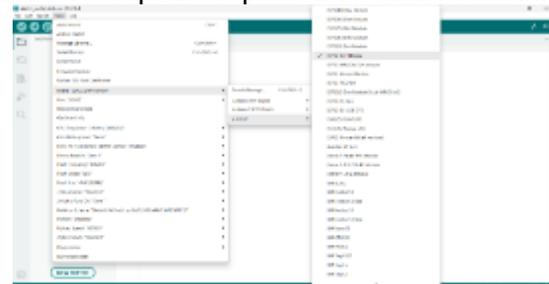


5. Hubungkan Esp 32 yang sudah terhubung dengan sensor sensor (seperti gambar 2) dengan Laptop

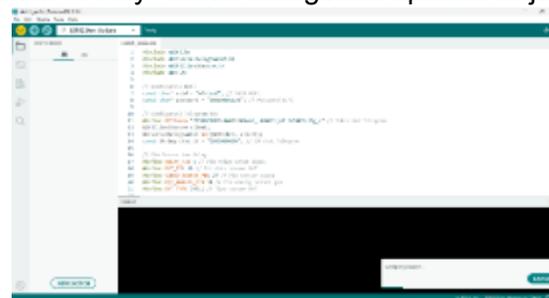
6. Ke navbar “Tools” → “Port” → pilih port yang tersedia (biasanya com2 / com3)



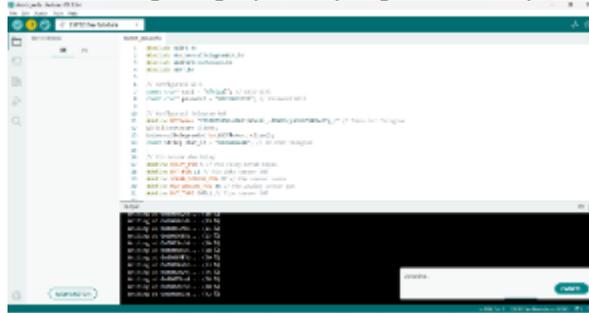
7. Ke navbar “Tools” → “Board” → “esp32” → pilih “ESP32 Dev Module”



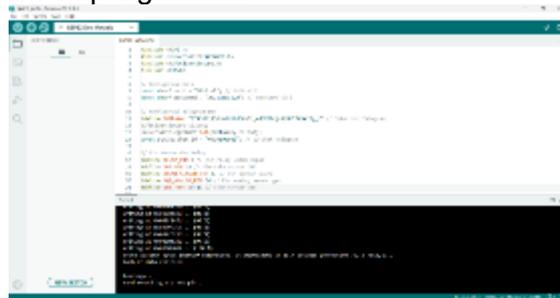
8. Lakukan Coding dan lakukan “Verify” untuk mengecek apakah terjadi error atau tidak



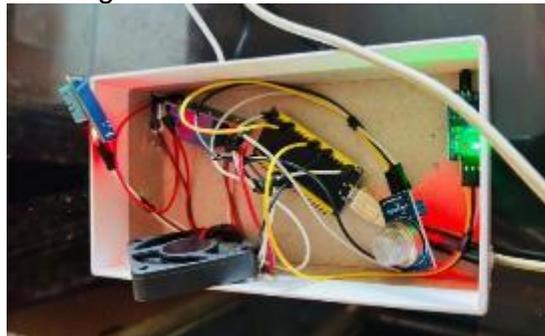
9. Jika tidak terjadi error, kita bisa langsung upload program ke Esp32



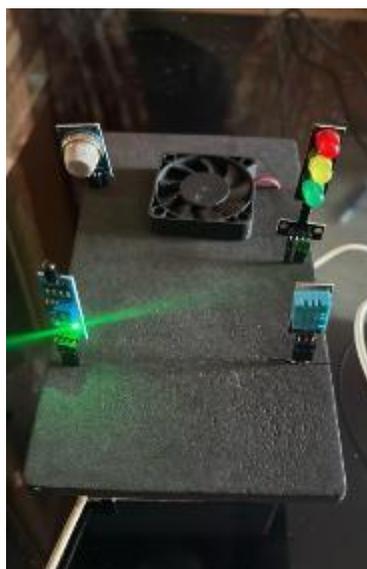
10. Upload selesai dan sistem siap digunakan



Berikut merupakan gambar dari rangkaian asli :



Gambar 3. Rangkaian Sistem yang Belum Dirapikkan

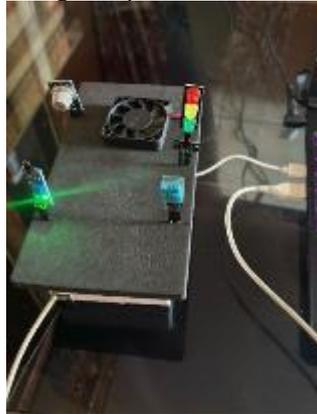


Gambar 4. Rangkaian Sistem yang Sudah Dirapikkan

Pengujian Pengontrolan Suhu dan Kelembapan

Tujuan pengujian sistem pengontrolan suhu adalah untuk memastikan bahwa konfigurasi antara ESP32, sensor DHT11, dan Telegram Bot berfungsi dengan baik. Pengujian ini juga bertujuan untuk memverifikasi apakah kipas atau pendingin tambahan dapat menyala secara otomatis ketika suhu melebihi ambang batas yang ditentukan, serta memastikan pemanas ruangan aktif jika kelembapan terlalu tinggi. Selain itu, sistem harus dapat mengirimkan notifikasi ke Telegram Bot secara real time. Dalam prototipe ini, lampu hijau digunakan sebagai representasi pemanas ruangan, sementara kipas mewakili AC. Langkah Pengujian :

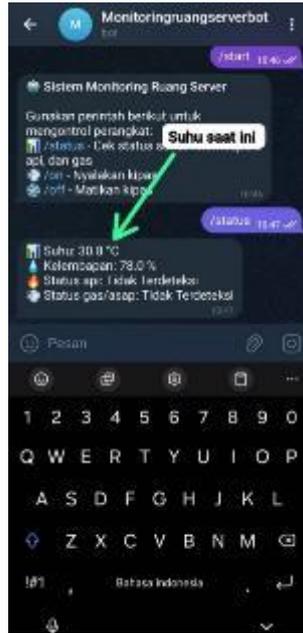
1. Nyalakan sistem dengan menghubungkan power ESP 32 dan kipas pada sumber



2. Ketikkan /start pada Bot Telegram



3. Pilih /status untuk mengetahui status suhu saat ini



4. Nyalakan korek api di sekitar sensor DHT 11 untuk menaikkan suhu



5. Pilih /status lagi pada bot untuk melihat apakah suhunya naik



6. Nyalakan terus korek api didekat sensor DHT 11 sampai melebihi batas suhu dan kita lihat apakah akan mengirimkan notifikasi dan apakah akan menyalakan kipas secara otomatis



- 7. Tunggu beberapa saat hingga suhu menurun untuk melihat apakah akan mengirimkan notifikasi dan apakah akan mematikan kipas secara otomatis



- 8. Pilih /status lagi untuk melihat status kelembapan



- 9. Naikkan kelembapan pada sensor DHT11
- 10. Kita lihat jika kelembapan melebihi batas apakah akan mengirimkan notifikasi dan akan menyalakan lampu hijau (prototipe dari pemanas ruangan)



- 11. Kita tunggu sampai normal Kembali apakah akan mematikan lampu hijau dan mengirimkan notifikasi



- 12. Pengujian selesai.
Berdasarkan hasil pengujian, sistem berfungsi dengan baik sesuai dengan yang diharapkan. Ketika suhu melebihi ambang batas yang telah ditentukan, kipas (prototipe dari pendingin tambahan) secara otomatis menyala dan mengirimkan notifikasi ke Telegram. Begitu suhu kembali normal, sistem akan mematikan kipas secara otomatis dan mengirimkan notifikasi kembali. Demikian pula, jika kelembapan melebihi ambang batas, lampu hijau (prototipe dari pemanas ruangan) akan menyala, dan ketika kelembapan kembali normal, lampu hijau akan mati serta notifikasi dikirimkan ke Telegram. Selain itu, sistem dilengkapi dengan fitur kontrol manual melalui Telegram, memungkinkan pengguna untuk menyalakan kipas secara manual dengan perintah /on dan memamatkannya dengan perintah /off, memberikan fleksibilitas tambahan dalam pengoperasian perangkat pendingin.



Pengujian Sensor MQ-2

Tujuan dari pengujian ini adalah untuk memastikan bahwa ESP32, sensor MQ2, dan Telegram Bot berfungsi dengan baik sesuai rancangan. Pengujian ini juga bertujuan untuk memverifikasi apakah sistem dapat mendeteksi keberadaan gas atau asap melalui sensor MQ2 dan secara otomatis mengirimkan notifikasi ke Telegram jika kondisi tersebut terdeteksi.

Langkah Pengujian :

1. Nyalakan sistem
2. Arahkan gas / asap didekat sensor MQ2



3. Kita lihat pada Telegram Bot apakah akan mengirimkan notifikasi



4. Selesai

Berdasarkan hasil pengujian, sensor MQ2, ESP32, dan Telegram Bot berhasil berfungsi sesuai dengan yang diharapkan. Sistem mampu mendeteksi keberadaan gas atau asap dan mengirimkan notifikasi ke Telegram secara real-time, membuktikan bahwa integrasi antar komponen berjalan dengan baik.

Pengujian Sensor API

Tujuan pengujian sensor Api adalah untuk memastikan bahwa sensor api, ESP32, dan Telegram Bot telah terintegrasi dengan baik dan berfungsi sebagaimana mestinya. Pengujian ini juga bertujuan untuk memastikan apakah sistem dapat mendeteksi api dan mengirimkan notifikasi secara real-time melalui Telegram. Langkah Pengujian :

1. Nyalakan sistem
2. Nyalakan korek api disekitar sensor



3. Kita lihat pada Telegram Bot apakah akan mengirimkan notifikasi dan menyalakan lampu merah (prototipe dari pemadam api)



4. Kita lihat setelah api sudah tidak terdeteksi apakah akan mengirimkan notifikasi dan mematikan lampu merah



5. Selesai

Berdasarkan hasil pengujian, sistem yang terdiri dari ESP32, sensor api, lampu merah (prototipe pemadam api), dan Telegram Bot telah berfungsi sesuai dengan yang diharapkan. Ketika sensor api mendeteksi keberadaan api, sistem secara otomatis mengirimkan notifikasi real-time melalui Telegram dan menyalakan lampu merah sebagai indikator aktivasi pemadam api. Begitupun Ketika sudah tidak terdeteksi api maka akan mematikan lampu merah dan mengirimkan notifikasi.

SIMPULAN

Berdasarkan hasil penelitian dan implementasi serta pengujian yang telah dilakukan, sistem pemantauan real-time ruang server berbasis ESP32 dan Telegram Bot telah berhasil dirancang dan diimplementasikan. Sistem ini mampu memantau suhu, kelembapan, keberadaan asap atau gas, serta keberadaan api, dan mengirimkan notifikasi secara real-time kepada pengguna. Selain itu, sistem juga dapat mengontrol perangkat secara otomatis, seperti menyalakan kipas (prototipe dari AC) ketika suhu melebihi ambang batas, menyalakan lampu merah (prototipe pemadam kebakaran) saat terdeteksi api, dan menyalakan lampu hijau (prototipe pemanas ruangan) ketika kelembapan melebihi batas yang ditentukan.

DAFTAR PUSTAKA

- Beizer, B. (1990). *Software testing techniques* (2nd ed.). Van Nostrand Reinhold.
- Hennessey, J. (2006). *Real-time systems and embedded systems*. Pearson Education.
- Ichsan, R. N. (2023). Sistem pemantauan dan pengendali suhu ruangan server berbasis internet of things dengan notifikasi alarm dan telegram. *Jurnal Elektro Kontrol (ELKON)*, 3(2), 11–24. <https://doi.org/10.24176/elkon.v3i2.10864>
- McConnell, S. (2004). *Code complete* (2nd ed.). Microsoft Press.
- Pressman, R. S. (2014). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill Education.
- Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner's approach* (8th ed.). McGraw-Hill Education.
- Pratama, I. P. A. E., & Suakanto, S. (2015). *Wireless sensor network*. Informatika.
- Rizal Rinaldi, M., Hamzah, A., & Lestari, U. (2018). Sistem pemantauan lingkungan ruang server berbasis internet of things menggunakan protokol message queue telemetry transport. *Jurnal Jarkom*, 6(2), 107–117.
- Rojas, R. (2017). *ESP32 programming for beginners: A hands-on guide to building IoT projects*. CreateSpace Independent Publishing Platform.
- Sauter, M. (2011). *From GSM to LTE: An introduction to mobile networks and mobile broadband*. Wiley.

- Sommerville, I. (2011). *Software engineering* (9th ed.). Addison-Wesley.
- Stroustrup, B. (2013). *The C++ programming language* (4th ed.). Addison-Wesley.
- Subkhi Mahmasani. (2018). Implementasi internet of things pada sistem pemantauan dan kendali suhu ruang server. 274–282.
- Sulaeman, B., Informatika, T., & Selatan, S. (2024). Rancang bangun sistem pengering sepatu otomatis menggunakan sensor kelembaban suhu berbasis Arduino. 12(3).
- Utami, F. H., & A. (2015). *Rekayasa perangkat lunak* (1st ed.). Deepublish.
- Waworundeng, J., Dumanaw, O., & Rumawouw, T. (2021). Temperature and humidity prototype detector based IoT in Universitas Klabat information system server room. *Cogito Smart Journal*, 7(1), 193–203.
- Yick, J., Mukherjee, B., & Ghosal, D. (2010). Wireless sensor network survey. *Computer Networks*, 52(12), 2292–2330.
- Yudaningtyas, E. (2017). *Belajar sistem kontrol: Soal dan pembahasan* (1st ed.). UB Press.