

## Rancang Bangun Sistem Keamanan pada Rak Server Berbasis Internet of Things

Syifa Aditya Setiawan<sup>1</sup>, Yan Sofyan Andhana Saputra<sup>2\*</sup>

<sup>1,2</sup>Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Darma Persada  
1,2Jl. Taman Malaka Selatan No.22, Pondok Kelapa, Duren Sawit, DKI Jakarta, Indonesia 13450  
\*yansofyan@gmail.com

**Abstrak** — Server memiliki peranan yang krusial dalam pengelolaan data dan jaringan. Server harus beroperasi setiap hari selama 24 jam. Perancangan alat untuk menjaga kondisi rak server agar tetap stabil dan tidak terjadi overheat merupakan upaya untuk menjaga kinerja server dan mencegah munculnya masalah yang lebih besar pada server. Manfaat dari alat ini adalah monitoring tidak perlu langsung datang ke ruang server untuk memeriksa suhu dan kelembaban yang terjadi pada rak server. Selain itu alat ini terintegrasi dengan Telegram, sehingga proses pengecekan suhu dan kelembaban menjadi lebih mudah dilakukan. Hasil pengamatan pada penelitian ini diantaranya alat memberikan respon realtime kondisi rak server yaitu suhu dan kelembaban serta memberikan pemberitahuan ke Telegram jika terjadi situasi suhu di luar batas yang telah ditetapkan.

Kata kunci – Rak Server, Monitoring, Suhu, Telegram

Copyright © 2024 JURNAL TIFDA  
All rights reserved.

### I. PENDAHULUAN

Dalam era teknologi kontemporer yang berbasis komputerisasi, server menjadi komponen vital dalam mendukung berbagai aktivitas digital. Server adalah sistem komputer yang dirancang untuk menyediakan layanan, menyimpan data, mengelola informasi, dan menjalankan aplikasi berbasis jaringan bagi pengguna atau perangkat lain yang terhubung ke jaringan. Sebagai inti dari infrastruktur teknologi, server harus memiliki kemampuan yang andal untuk memenuhi banyak permintaan didukung oleh sumber daya seperti kapasitas memori yang besar dan kemampuan pemrosesan yang optimal.

Keberlangsungan operasi server juga dipengaruhi oleh faktor eksternal seperti suhu dan kelembaban lingkungan di sekitar rak server. Pengelolaan kondisi ini menjadi sangat penting untuk menjaga stabilitas dan mencegah gangguan teknis. Oleh karena itu, pengembangan sistem keamanan berbasis teknologi untuk rak server termasuk pemantauan dan pengendalian kondisi lingkungan menjadi kebutuhan mendesak guna memastikan kelangsungan fungsi server secara optimal. Server tetap dalam kondisi optimal dan terhindar dari kerusakan merupakan hal yang wajib dilakukan, pengelolaan suhu dan

kelembaban di rak server harus dilakukan sesuai dengan standar yang telah ditetapkan. Suhu di dalam ruang server idealnya berada dalam rentang 20-25°C memiliki pengaruh signifikan terhadap kinerja server. Pemantauan suhu dan kelembaban secara manual tidak hanya memakan waktu dan tenaga tetapi juga kurang efektif dalam memberikan respons cepat terhadap perubahan kondisi lingkungan.

Dengan memanfaatkan teknologi Internet of Things (IoT), sistem pemantauan suhu dan kelembaban dapat dilakukan secara real-time dan otomatis. IoT memungkinkan pengawasan yang lebih efisien, sekaligus memberikan kemampuan untuk mengontrol kondisi lingkungan di rak server. Ketika suhu terlalu tinggi atau terlalu rendah, sistem berbasis IoT dapat memberikan peringatan dini atau mengambil tindakan untuk menjaga stabilitas server, sehingga risiko gangguan kinerja atau kegagalan fungsi server dapat diminimalkan.

Rak server sering kali diisi dengan komponen-komponen elektronik yang mudah panas yang jika tidak dikelola dengan baik dapat menyebabkan korsleting, api, atau asap. Oleh karena itu, penambahan sensor pendeksi api dan asap menjadi komponen penting dalam sistem keamanan rak server untuk

mengidentifikasi potensi kebakaran sejak dini. Dengan rancang bangun sistem berbasis IoT, pemantauan dan pengelolaan suhu, kelembapan, serta deteksi bahaya dapat diintegrasikan untuk menciptakan lingkungan operasional yang aman dan efisien bagi server. Konsep IoT tidak hanya mendukung monitoring tetapi juga menghubungkan berbagai perangkat elektronik dalam jaringan, sehingga memberikan solusi menyeluruh dalam menjaga stabilitas kinerja server.

Konsep Internet of Things (IoT) memungkinkan monitoring jarak jauh dengan memanfaatkan perangkat yang mampu mengirimkan informasi secara real-time melalui jaringan internet tanpa memerlukan interaksi langsung manusia. IoT merupakan teknologi yang dirancang untuk mengoptimalkan fungsi perangkat elektronik, seperti sensor dan peralatan lain yang terhubung ke jaringan, guna meningkatkan efisiensi dan responsivitas.

Saat ini, rak server belum dilengkapi dengan sistem monitoring yang terintegrasi, sehingga menyulitkan admin dalam memantau kondisi rak server secara real-time. Ketidakhadiran sistem ini dapat menghambat deteksi dini terhadap potensi masalah, seperti perubahan suhu, kelembapan, atau ancaman lainnya. Oleh karena itu, penerapan sistem monitoring berbasis IoT diharapkan mampu menyediakan solusi keamanan yang andal, memungkinkan pengawasan kondisi rak server kapan saja dan di mana saja, sekaligus memberikan peringatan dini yang dapat mengurangi risiko gangguan pada server.

## II. TINJAUAN PUSTAKA

### NodeMCU ESP8266

NodeMCU ESP8266 adalah platform dan kit pengembangan IoT *open-source* yang menggunakan bahasa pemrograman Lua untuk membuat prototipe produk IoT atau membuat sketch dengan IDE. Logika dan susunan pemrograman bahasa Lua sama dengan bahasa C, hanya syntaxnya yang berbeda. Saat anda berbicara bahasa Lua, anda dapat menggunakan alat Lua loader dan Lua uploader. Kit ini didasarkan pada modul ESP8266, yang menggabungkan GPIO, PWM (*Pulse Width Modulation*), IIC, 1-Wire, dan ADC ke dalam satu board.

### Internet of Things

*Internet of Things* (IoT) adalah sistem yang memiliki identitas unik untuk objek dan orang dan memungkinkan pengiriman data melalui jaringan tanpa menggunakan dua arah antara manusia ke manusia yaitu sumber ke tujuan atau interaksi manusia ke komputer.

### Sensor API IR Flame Detection

Flame sensor atau sensor api merupakan alat pendekripsi kebakaran melalui adanya nyala api yang muncul secara tiba-tiba. Sensor ini umumnya

digunakan sebagai bagian dari sistem keamanan atau pengendalian otomatis untuk mendeteksi kebakaran atau bahaya api. Flame sensor bekerja dengan mengukur radiasi elektromagnetik yang dihasilkan oleh nyala api, dan ketika radiasi tersebut mencapai ambang batas tertentu, sensor akan memberikan sinyal atau memicu tindakan tertentu, seperti mengaktifkan sistem pemadam api atau memberi peringatan.

### Sensor Gas MQ2

Sensor MQ-2 adalah salah satu sensor yang peka terhadap asap dan digunakan untuk membedakan pemasukan gas yang dapat menyala. Alat deteksi gas ini adalah alat ukur atau instrumentasi keberadaan sensor sangat penting untuk alat instrumentasi. Sensor adalah alat yang dapat mengubah signal dalam besar fisis (non-listrik) menjadi signal listrik.

### Sensor DHT22

Sensor DHT22 adalah sensor suhu dan kelembaban, juga disebut sensor AM2302. Sensor DHT22 merupakan sensor terbaik berdasarkan respons, kecepatan pembacaan data, ukurannya kecil, dan dapat mengirimkan sinyal hingga 20 meter. Sensor DHT22 mendekripsi tingkat suhu udara yang tinggi secara otomatis, dan mikrokontroler memproses data suhu dan menampilkan output yang telah diprogram pada layar LCD.

### LED

*Light-Emitting Diode* (LED) adalah suatu divais semikonduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju. Pada umumnya, diode dan chip LED membutuhkan tegangan tertentu untuk beroperasi. Namun jika diberikan tegangan yang terlalu tinggi, LED dapat rusak atau terbakar, bahkan jika tegangan itu maju (Vf).

### Bot Telegram

Bot adalah aplikasi pihak ketiga yang memungkinkan pengguna mengirim pesan, perintah, dan permintaan secara bersamaan dengan Telegram. Memanfaatkan HTTPS untuk API Telegram, kami dapat mengelola bot.

### Android

Android adalah sistem operasi untuk telepon seluler berbasis Linux yang berisi sistem operasi, middleware, dan aplikasi. Android memberikan platform terbuka bagi pengembang untuk membuat aplikasi mereka sendiri. Logo Android warna hijau dirancang oleh desainer grafis Irina Blok untuk Google pada tahun 2007.

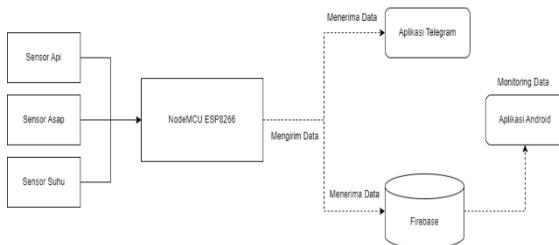
### Firebase

Firebase adalah suatu layanan dari Google yang membantu pengembang aplikasi membuat aplikasi yang lebih mudah dibuat. Dengan Firebase, pengembang aplikasi dapat berkonsentrasi pada membuat aplikasi tanpa melakukan banyak pekerjaan.

### III. METODOLOGI

Bagian ini menggambarkan perancangan perangkat keras (*Hardware*) dan perancangan perangkat lunak (*Software*) Sistem keamanan pada rak server berbasis *Internet of Things*.

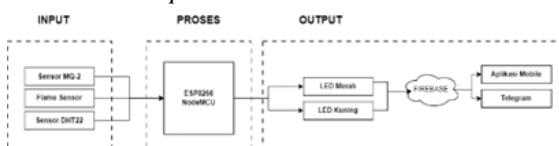
#### Arsitektur Sistem



Gambar 1. Diagram Arsitektur Sistem

Berdasarkan Gambar 1 NodeMCU berfungsi sebagai kepala untuk mengendalikan data yang dikirimkan oleh sensor api, sensor suhu, dan sensor asap. NodeMCU akan mengirimkan data kepada firebase dan aplikasi telegram. Setelah itu firebase akan mengirimkan kembali ke aplikasi android yang digunakan untuk *monitoring* keamanan CPU.

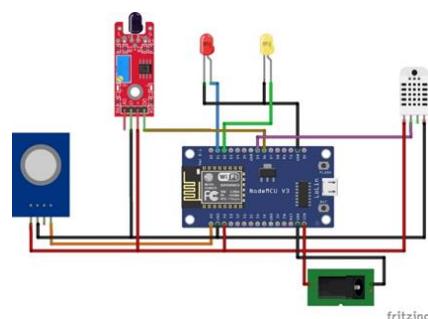
#### Arsitektur Komponen



Gambar 2. Diagram Arsitektur Komponen Sistem

Pada Gambar 2 adalah rancang bangun sistem keamanan pada rak server berbasis *internet of things* terbagi menjadi tiga bagian, yaitu: Input, Proses, dan Output.

- Blok Input, menjelaskan mengenai masukan pada sensor yang digunakan, yaitu Sensor Gas MQ-2, Sensor Api (*Flame Detection*), dan Sensor DHT22.
- Blok Proses, berfungsi sebagai pengontrol kerja alat dari inputan yang tersedia, proses ini menggunakan Mikrokontroler ESP8266.
- Blok Output, berfungsi sebagai keluaran yang akan ditampilkan melalui aplikasi mobile dan Telegram. Data keluarannya pun akan terindikasi melalui lampu LED yang tersedia.



Gambar 3. Rangkaian Komponen Alat IoT

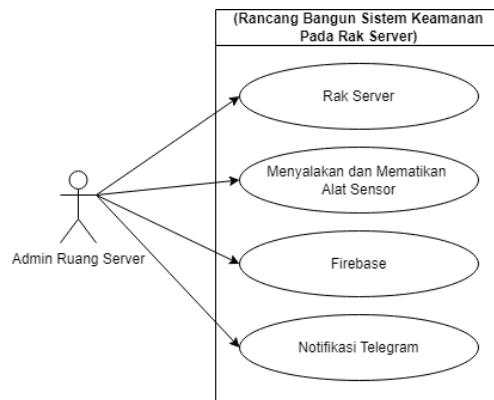
Pada Gambar 3 merupakan gambaran rangkaian perangkat IoT yang dibuat untuk sistem keamanan rak server.

#### Perancangan Perangkat Lunak

Untuk rancangan perangkat lunak, penulis melakukan pendekatan dengan menggunakan *Unified Modeling Language* (UML). UML adalah pendekatan yang sangat efektif untuk merancang dan mendokumentasikan sistem perangkat lunak. Dalam perancangan alat sistem keamanan rak server berbasis IoT, penulis menggunakan jenis diagram UML untuk merinci aspek-aspek pada rancang bangun sistem yaitu: *Use Case Diagram*, *Activity Diagram*, dan *Flowchart Diagram*.

#### Diagram Use Case

*Use Case Diagram* membantu mengidentifikasi dan memodelkan interaksi antara aktor (pengguna atau sistem lain) dengan sistem. Gambar 4 adalah *use case diagram* Sistem keamanan pada rak server berbasis IoT.

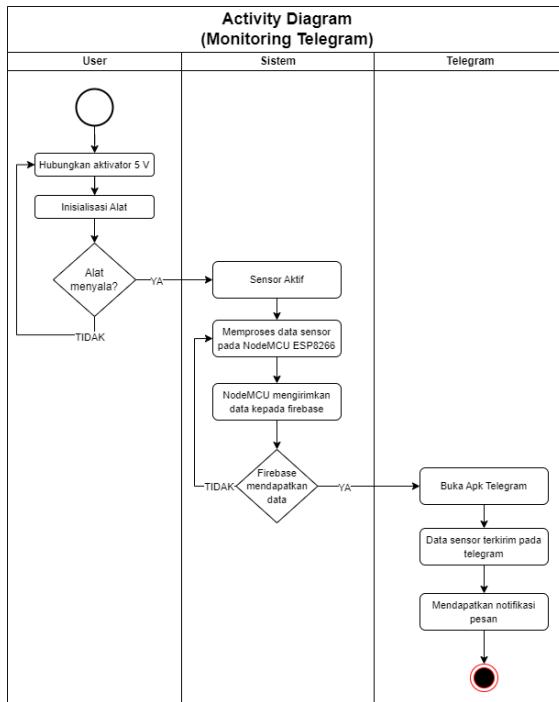


Gambar 4. Use Case Diagram Admin Server

Pada *use case* keamanan rak server pengguna dapat memonitor keamanan rak server dengan melihat informasi dari sensor suhu, sensor api, dan sensor asap dengan aksi yang diberikan yaitu Menerima data dari NodeMCU ESP8266, memproses data, dan menampilkan informasi pada aplikasi Telegram. Apabila terdapat pendekatan berbahaya sistem dapat memberikan pemberitahuan darurat melalui aplikasi Telegram ketika mendekati kondisi berbahaya seperti suhu tinggi atau deteksi api/asap. Setelah itu pengguna dapat mengendalikan operasi alat sensor seperti menyalakan dan mematikan sensor suhu, sensor api maupun sensor asap. Pengendalian ini memanfaatkan proses pengiriman perintah kepada NodeMCU untuk mengatur status operasional sensor. Terakhir pengguna dapat melihat data monitoring actual pada deteksi suhu, deteksi api, dan juga deteksi asap pada rak server, data yang diperoleh akan dipantau dan akan dikirimkan dalam aplikasi Telegram.

### Diagram Activity

Activity Diagram Merupakan langkah-langkah atau tindakan dalam proses yang dilakukan dalam suatu aktivitas. Activity diagram membantu dalam pemodelan dan mendokumentasikan aktivitas-aktivitas yang terjadi dalam sistem. Gambar 5 adalah *activity diagram* pada penelitian yang dilakukan.

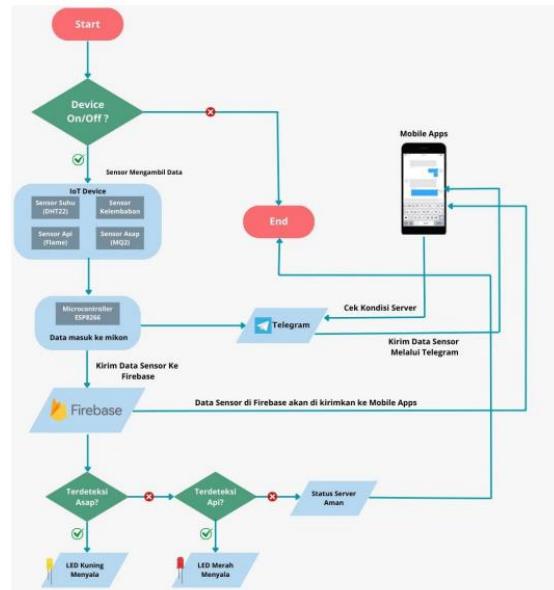


Gambar 5. *Activity Diagram* Monitoring Aplikasi

### Diagram Flowchart

Diagram *flowchart* berfungsi untuk menjelaskan alur proses dalam prosedur penggunaan alat dari sistem keamanan rak server berbasis *internet of things* secara detail, berikut ini adalah penjelasan dari *flowchart* alat yang sudah dibuat antara lain proses pertama pengguna harus memastikan bahwa alat sudah menyala dan terinisialisasi dengan baik, setelah itu sensor-sensor yang digunakan dalam perancangan ini akan aktif dan mulai untuk mengambil data sesuai dengan kondisi rak server. Data yang didapatkan akan masuk dan diproses pada mikrokontroler lalu dikirimkan ke Telegram untuk mendapatkan notifikasi pesan.

Data juga akan disimpan di *Firebase* dan melakukan monitoring data menggunakan *mobile apps*. Data sensor yang sudah didapatkan akan menghasilkan keluaran berupa LED merah menyala jika terdeteksi api dan LED kuning menyala jika terdeteksi asap pada rak. Gambar 6 adalah aliran grafis *flowchart* diagram untuk sistem monitoring keamanan rak server berbasis *internet of things*.



Gambar 6. Diagram *Flowchart*

### IV. HASIL DAN PEMBAHASAN

Sistem pemantauan pada penelitian ini dibangun di atas mikrokontroler, yang di tempatkan pada papan tunggal NodeMCU ESP8266 Lolin. Mikrokontroler ini dapat dioperasikan dari jarak jauh dan memiliki koneksi internet. NodeMCU ESP8266 berfungsi sebagai central processing unit (CPU) untuk sensor suhu DHT22, sensor Flame Api, dan LED yang digunakan sebagai indikator. Data yang dikumpulkan akan dikirim ke aplikasi Flutter melalui jaringan Wi-Fi serta mampu untuk memberikan notifikasi melalui Telegram.

Pada Implementasi monitoring sensor dengan aplikasi Flutter menggunakan spesifikasi Handphone dijabarkan pada Tabel 1.

Tabel 1. Spesifikasi Handphone

No.	Nama	Type
1.	Perangkat	Oppo A15
2.	Sistem Operasi	Android
3.	Chipset	Delapan-Inti
4.	Versi Perangkat Lunak	10
5.	Memori RAM	3GB
6.	Memori Internal ROM	32GB

### Integrasi Notifikasi Telegram

Untuk mengetahui bagaimana data sensor tersebut diolah dan dikirimkan sebagai respons terhadap pesan baru dari telegram. Ketika pesan baru diterima, data dari sensor suhu, api, dan asap akan diambil, diolah, dan dikirimkan sebagai respons melalui telegram. Pengiriman respons ini memicu notifikasi ke pengguna melalui telegram, menyampaikan informasi aktual mengenai kondisi rak server berbasis pada data yang diterima dari sensor.

Mikrokontroler akan menggunakan *HTTP Request* untuk mengirim pesan ke API Telegram dengan Token API Bot sebagai parameter untuk autentifikasi. Sehingga data dari mikrokontroler akan terkirim ke Telegram melalui jaringan internet seperti pada Gambar 7.

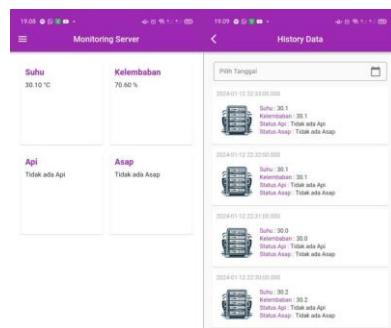


Gambar 7. Notifikasi Data Sensor melalui Telegram

Pada program *integrasi notifikasi Telegram* terdapat alur untuk menangani pesan yang diterima dari Telegram. Pertama, program memanggil fungsi `getUpdates` untuk mendapatkan jumlah pesan baru dari Telegram dengan memberikan parameter `bot.last_message_received + 1`. Jumlah pesan baru ini kemudian dicetak di Serial Monitor melalui `Serial.println(numNewMessages)` untuk keperluan debugging. Selanjutnya, terdapat kondisi `if (numNewMessages > 0)` yang mengecek apakah ada pesan baru. Jika jumlah pesan baru lebih dari 0, program akan memanggil fungsi `handleNewMessages` untuk menangani pesan-pesan tersebut. Selain itu, nilai variabel `numNewMessages` diatur kembali menjadi 0, dan pesan "Send to Telegram" dicetak di Serial Monitor. Proses ini mengindikasikan bahwa notifikasi Telegram muncul sebagai respons terhadap adanya pesan baru yang telah diterima dan ditangani oleh fungsi `handleNewMessages`.

#### Konektivitas Aplikasi Mobile Flutter

Aplikasi mobile ini digunakan hanya untuk memonitoring data sensor yang disimpan pada *firebase* untuk melakukan koneksi dan integrasi pada aplikasi mobile seperti terlihat pada Gambar 8.



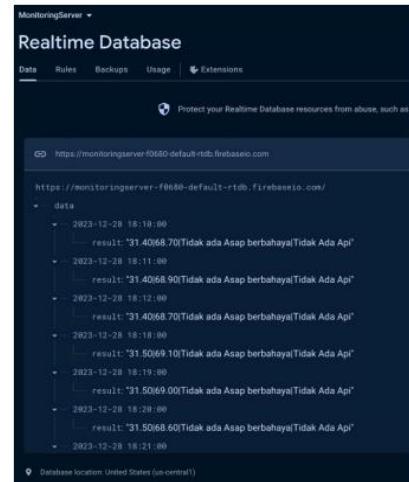
Gambar 8. Tampilan Monitoring Mobile Apps

Untuk melakukan koneksi dan integrasi pada aplikasi *mobile* dengan kode algoritma 1.

#### Algorithma 1

```
if (millis() - lastFirebaseUpdate > 10000) {
    String parentPath = "/data/" +
        getCurrentDate() + "/result";
    String result = String(t) + "/" + String(h) + "/" +
        String(api) + "/" + String(asap);
    Firebase.setString(fbdo, parentPath, result);
    Serial.println(result);
    lastFirebaseUpdate = millis();
```

Pada blok ini, terdapat pengecekan waktu terakhir pembaruan *Firebase* menggunakan kondisi `millis() - lastFirebaseUpdate > 10000` yang menandakan bahwa data akan dikirim ke *Firebase* setiap 10 detik sekali.



Gambar 9. Data pada Firebase

Kemudian dibentuklah sebuah *path* untuk *Firebase* yang mencakup tanggal saat ini. Data suhu, kelembaban, tingkat gas API, dan tingkat asap yang telah diukur oleh sensor disusun menjadi satu string dalam variabel `result`. Data ini kemudian dikirimkan ke *Firebase* menggunakan fungsi `setString`. Selanjutnya, data tersebut dicetak di Serial Monitor seperti yang terlihat pada Gambar 9.

#### Hasil Pengamatan Sensor

Tabel 2 adalah hasil pengamatan pada sensor api, sensor asap, dan juga sensor suhu :

Tabel 2. Pengamatan pada Sensor

Sensor	Waktu Pengamatan	Kondisi	Tindakan	Nilai Suhu	Nilai Kelembapan	Status Server
Sensor Api	08.00	Terdeteksi Api	Lampu LED Merah Menyal	34.2° C	62.80 %	Bahaya
	08.30	Tidak ada api	Lampu LED Mati	28° C	53.36 %	Aman
Sensor Asap	09.30	Terdeteksi Asap	Lampu LED Kuning Menyal	37.4° C	59.16 %	Bahaya
	10.00	Tidak ada asap	Lampu LED Mati	26° C	42.79 %	Aman

Pengamatan terhadap sensor dilakukan setiap interval 30 menit untuk memastikan responsivitas terhadap perubahan kondisi lingkungan. Sensor mendeteksi dua skenario utama diantaranya pertama kondisi di mana terdapat indikasi "Terdeteksi Api" atau "Terdeteksi Asap" dan kedua kondisi tanpa adanya indikasi api maupun asap. Saat sensor mendeteksi keberadaan api, sistem secara otomatis mengaktifkan lampu LED merah sebagai tanda peringatan visual sementara status server berubah menjadi "bahaya" menandakan potensi ancaman yang memerlukan respons cepat. Jika sensor mendeteksi keberadaan asap, lampu LED kuning menyala menunjukkan situasi waspada yang membutuhkan perhatian khusus. Sebaliknya, ketika sensor tidak mendeteksi api maupun asap, status server tetap berada dalam kondisi "Aman" mencerminkan lingkungan yang stabil dan terkendali. Selain itu, selama proses pengamatan, sistem juga mencatat data suhu dan kelembaban secara *real-time* memberikan informasi tambahan yang relevan untuk analisis lebih lanjut terutama ketika terjadi deteksi api atau asap. Hal ini memungkinkan sistem untuk tidak hanya memberikan peringatan tetapi juga menyediakan data lingkungan yang dapat digunakan untuk penilaian risiko yang lebih mendalam.

#### Hasil Pengamatan Server

Hasil pengamatan server menunjukkan adanya dua kondisi operasional utama, yaitu Normal dan Darurat, yang masing-masing merepresentasikan status lingkungan berdasarkan respons sensor. Kondisi Normal terjadi ketika sensor tidak mendeteksi adanya ancaman seperti api atau asap, lampu LED Mati. Pada kondisi ini, server mencatat nilai suhu dan kelembaban dalam kisaran yang stabil dan dianggap sebagai parameter lingkungan yang ideal. Situasi ini mencerminkan bahwa tidak ada perubahan signifikan di lingkungan yang memerlukan intervensi.

Kondisi Darurat diidentifikasi ketika sensor mendeteksi keberadaan api, dalam skenario yang dilakukan setiap nilai suhu dan kelembaban yang dicatat oleh server meningkat secara signifikan, menjadi indikator utama dari perubahan lingkungan yang drastis dan berpotensi berbahaya. Status server otomatis berubah menjadi "Bahaya" dan sistem secara otomatis menyalakan lampu LED merah yang bertujuan untuk memberikan peringatan dini agar tindakan mitigasi dapat segera dilakukan. Tabel 3 adalah hasil pengamatan pada kondisi rak server:

Tabel 3. Pengamatan pada Server

Kondisi	Kondisi	Tindakan	Nilai Suhu	Nilai Kelembaban	Status Server
Normal	Tidak ada api	Lampu LED Mati	24.5° C	42.79 %	Aman
	Tidak ada asap	Lampu LED Mati	25.2° C	53.36 %	Aman
Darurat	Terdeteksi Api	Lampu LED Merah Menyala	37.4° C	59.16 %	Bahaya
	Terdeteksi Asap	Lampu LED Kuning Menyala	26° C	61.02 %	Bahaya

#### V. KESIMPULAN

Dalam pengembangan sistem keamanan pada rak server berbasis Internet of Things (IoT) dengan menggunakan rangkaian alat rancang bangun yang melibatkan NodeMCU ESP8266, sensor DHT22, sensor MQ-2, sensor flame API, dan integrasi dengan Firebase dan Telegram diperoleh beberapa kesimpulan diantaranya sistem sensor berhasil membedakan kondisi Normal dan Darurat secara akurat, dengan respons yang sesuai seperti perubahan status server menjadi "Bahaya" saat mendeteksi api atau asap, serta pencatatan parameter suhu dan kelembaban yang relevan. Integrasi NodeMCU ESP8266 sebagai mikrokontroler menjadi inti dari sistem rangkaian alat ini. NodeMCU ESP8266 digunakan untuk mengumpulkan data dari sensor DHT22 (suhu dan kelembaban), sensor MQ-2 (gas), dan sensor flame API (api). Dengan demikian, sistem mampu memberikan pemantauan realtime terhadap kondisi lingkungan yang kritis bagi keamanan rak server dan Sensor-sensor dapat memberikan data yang diperlukan untuk tiga aspek yaitu deteksi kebakaran melalui sensor flame API, identifikasi asap melalui sensor MQ-2, dan pemantauan suhu dan kelembaban melalui sensor DHT22. Setiap sensor memberikan informasi spesifik yang diperlukan untuk mengidentifikasi potensi risiko dan mengambil tindakan preventif atau respons cepat.

Adapun saran dalam penelitian selanjutnya dapat mempertimbangkan untuk mengintegrasikan sensor tambahan yang dapat memberikan informasi lebih detail tentang kondisi lingkungan, misalnya sensor deteksi gerakan untuk memonitor aktivitas fisik di sekitar rak server atau sensor kebisingan untuk mendeteksi suara aneh yang mungkin mengindikasikan potensi masalah, kedua menerapkan teknologi *machine learning* untuk meningkatkan kemampuan prediktif sistem, ketiga dapat menambahkan sensor untuk memantau kualitas udara di dalam server. Kualitas udara yang buruk dapat memengaruhi kinerja perangkat keras, dan pemantauan ini dapat membantu dalam menjaga lingkungan yang optimal.

#### DAFTAR PUSTAKA

- [1] Angkasah, A., Studi Teknik Informatika Politeknik Aceh Jln Politeknik Aceh Desa Pango Raya, P., & Aceh, B. (2019). Sistem pengukur suhu server menggunakan arduino dan raspberry pi (temperature server measurement system using arduino and raspberry pi). *Jurnal J-Innovation*, 8(2).
- [2] Fatra, D., & Syazili, A. (n.d.). Sistem monitoring suhu jarak jauh pada ruang server berbasis internet of things. *Bina Darma Conference on Computer Science*.
- [3] Hadi Rantelinggi, P. (2020). Pemantau Suhu Menggunakan NodeMcu, IoT dan Cayenne pada Rack Server. *Telematika*, 13(2), 80–90. <https://doi.org/10.35671/telematika.v13i2.1001>

- 
- [4] Pratama, R., Siambaton, Mhd. Z., & Haramaini, T. (2022). Implementasi Internet of Things pada Sistem Kendali Suhu dan Kelembaban Rak Server Berbasis Mikrokontroler. *Sudo Jurnal Teknik Informatika*, 1(4), 145–153. <https://doi.org/10.56211/sudo.v1i4.129>
- [5] Sistem Komputer dan Kecerdasan Buatan, J., Maulana Yusuf, E., Pratama, F., Surya Kencana No, J., & Selatan, T. (2023). Rancang Bangun Sistem Monitoring Suhu Dan Kelembaban Ruang Server Berbasis Iot Menggunakan Arduino Pada Pt Bintaro Serpong Damai.
- [6] Yuhanan, A., & Suhada, K. (n.d.). LPPM STMIK ROSMA / Prosiding Seminar Nasional : Inovasi & Adopsi Teknologi Rancang Bangun Monitoring Suhu danan Kelembaban Ruang Server Berbasis Web Menggunakan Node MCU ESP8266.