

Penerapan Push Notification OneSignal Pada System Monitoring Suhu Menggunakan Kodular dan Firebase Berbasis ESP32

¹Harlan Kurnia AR

¹Universitas Putra Indonesia "YPTK" Padang, Indonesia

INFORMASI ARTIKEL

Diterima 26 Januari 2026
Direvisi 21 Februari 2026
Diterbitkan 23 Februari 2026

Kata kunci:

Push Notification;
OneSignal;
Kodular;
Firebase;
ESP32;

ABSTRAK

Sistem monitoring suhu berbasis IoT yang ada sebelumnya masih memiliki kelemahan, di mana pengguna harus terus membuka aplikasi untuk memantau kondisi suhu. Tanpa fitur peringatan aktif, pengguna berisiko terlambat merespons kondisi kritis jika aplikasi tidak sedang dijalankan. Penelitian ini bertujuan untuk mengembangkan fitur notifikasi aktif pada aplikasi Android menggunakan Push Notification OneSignal. Tujuannya agar sistem dapat memberikan peringatan dini (*early warning*) kepada pengguna secara otomatis tanpa harus membuka aplikasi terlebih dahulu saat kondisi suhu atau kelembaban melewati batas normal. Sistem dibangun menggunakan mikrokontroler ESP32, sensor DHT11, dan Firebase Realtime Database. Pengembangan aplikasi menggunakan platform *low-code* Kodular yang diintegrasikan dengan API OneSignal dan Firebase Cloud Messaging (FCM) untuk manajemen distribusi notifikasi. Hasil penelitian ini menunjukkan bahwa aplikasi monitoring suhu yang dikembangkan mampu memberikan notifikasi otomatis kepada pengguna tanpa harus menjalankan aplikasi. Sistem peringatan ini akan aktif jika suhu melampaui 35°C atau kelembaban mencapai lebih dari 70%. Berdasarkan data pengujian, fitur Push Notification terbukti sangat efektif dengan rata-rata jeda waktu pengiriman hanya 3,11 detik, serta tetap dapat diandalkan untuk memberikan peringatan dini meskipun aplikasi sedang tidak berjalan. Sehingga dapat ditarik kesimpulan bahwa fitur push notification onesignal pada kodular dapat digunakan untuk melengkapi system monitoring suhu berbasis IOT, sebagai pemberitahuan pada pengguna jika suhu dan kelembaban melewati batas.

Implementation of OneSignal Push Notification on Temperature Monitoring System Using Kodular and Firebase Based on ESP32

ARTICLE INFO

Received January 26, 2026
Revised February 21, 2026
Published February 23, 2026

Keyword:

Push Notification;
OneSignal;
Kodular;
Firebase;
ESP32;

ABSTRACT

Existing IoT-based temperature monitoring systems still have a weakness, where users must continuously open the application to monitor temperature conditions. Without an active alert feature, users risk delays in responding to critical conditions if the application is not running. This research aims to develop an active notification feature in an Android application using OneSignal push notifications. The goal is for the system to automatically provide early warnings to users without having to first open the application when temperature or humidity conditions exceed normal limits. The system was built using an ESP32 microcontroller, a DHT11 sensor, and the Firebase Realtime Database. The application development used the Kodular low-code platform integrated with the OneSignal API and Firebase Cloud Messaging (FCM) for notification distribution management. The results of this study show that the developed temperature monitoring application is capable of providing automatic notifications to users without having to run the application. This warning system will be activated if the temperature exceeds 35°C or humidity reaches more than 70%. Based on test data, the push notification feature proved very effective with an average delivery time of only 3.11 seconds, and remains reliable in providing early warnings even when the application is not running. So it can be concluded that the onesignal push notification feature on Kodular can be used to complement the IoT-based temperature monitoring system, as a notification to users if the temperature and humidity exceed the limit.

This work is licensed under a [Creative Commons Attribution 4.0](#)



Corresponding Author:

Harlan Kurnia AR, Universitas Putra Indonesia YPTK, Sumatera Barat Indonesia
Email: harlankurnia@upiypk.ac.id

1. PENDAHULUAN

Efisiensi dalam pengawasan suhu lingkungan merupakan aspek krusial di berbagai bidang, mulai dari industri farmasi, penyimpanan logistik pangan, hingga pemantauan perangkat elektronik yang sensitif. Terkadang pada beberapa bidang, perubahan suhu yang fluktuatif tanpa penanganan yang cepat seringkali menjadi penyebab utama kerusakan material atau malfungsi sistem. Selama ini, metode pemantauan yang diterapkan menggunakan metode konvensional yang mengandalkan pengecekan manual[1]. Hal ini dinilai kurang efektif karena keterbatasan ruang dan waktu, serta tingginya risiko keterlambatan respon jika terjadi anomali suhu.

Perkembangan teknologi Internet of Things (IoT) menawarkan solusi inovatif melalui integrasi sensor cerdas dan konektivitas nirkabel. IoT mengalami perkembangan pesat dalam waktu yang singkat, didorong oleh kemajuan teknologi informasi yang signifikan[2]. Salah satu platform yang handal dalam implementasi ini adalah ESP32, sebuah mikrokontroler yang memiliki modul Wi-Fi terintegrasi untuk transmisi data secara *real-time*[3]. Pemrograman ESP32 menggunakan *software* Arduino IDE yang merupakan perangkat lunak *open-source* yang mempermudah pengguna dalam

menulis dan mengunggah kode ke perangkat Arduino[4]. Dalam ekosistem pemantauan ini, Firebase berperan sebagai basis data berbasis awan (*cloud database*) yang menjamin sinkronisasi data suhu secara instan antara perangkat keras dan aplikasi pengguna. Firebase merupakan platform besutan Google yang dirancang untuk mengoptimalkan proses pengembangan aplikasi[5]. platform yang menyediakan ekosistem fitur lengkap guna mengoptimalkan proses pengembangan aplikasi bagi para pengembang secara efisien[6].

Namun, pengawasan yang hanya bersifat pasif melalui *dashboard* aplikasi seringkali membuat pengguna melewati kondisi kritis. Oleh karena itu, diperlukan suatu mekanisme peringatan aktif. Penggunaan layanan OneSignal sebagai penyedia layanan Push Notification merupakan solusi untuk mengirimkan peringatan langsung ke perangkat Android pengguna saat suhu mencapai ambang batas tertentu. Fitur Push Notification memberikan kemampuan bagi sistem untuk mendistribusikan informasi peringatan kepada pengguna secara real-time , meskipun aplikasi pada perangkat seluler tersebut sedang tidak dijalankan atau dalam kondisi tidak aktif[7]. Aplikasi antarmuka pengguna dalam penelitian ini dikembangkan menggunakan Kodular. Sebagai alat pengembang berbasis web yang memiliki kemiripan dengan MIT App Inventor. Merupakan sebuah platform berbasis visual yang dirancang untuk memudahkan siapa saja, termasuk pemula tanpa latar belakang pemrograman[8]. Dengan menggunakan kodular yang berbasis blok pada pembuatan aplikasi, dapat memudahkan integrasi antara API OneSignal dan database Firebase tanpa memerlukan pengkodean manual yang kompleks.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pemantauan suhu yang tidak hanya akurat dalam penyajian data, tetapi juga responsif dalam memberikan peringatan dini. Dengan mengintegrasikan ESP32, Firebase, dan fitur notifikasi OneSignal melalui aplikasi Kodular, diharapkan tercipta sebuah sistem pemantauan yang mampu meminimalkan risiko kerusakan akibat perubahan suhu yang tidak terkendali secara efektif dan efisien.

2. METODE

Metode penelitian yang digunakan dalam pengembangan sistem monitoring suhu dan kelembaban ini mengikuti kerangka kerja sistematis yang terdiri dari beberapa tahapan utama, yaitu: identifikasi kebutuhan, perancangan sistem (*system design*) , implementasi perangkat keras dan lunak, serta pengujian fungsionalitas. Lebih jelasnya mengenai kerangka kerja dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Kerja Penelitian

2.1. Identifikasi dan Analisis Kebutuhan

Tahap awal dimulai dengan menentukan spesifikasi kebutuhan sistem. Spesifikasi kebutuhan sistem sangat penting untuk menyelaraskan antara kebutuhan pengguna dalam menyelesaikan masalahnya, sehingga perangkat lunak atau sistem yang dibangun sesuai dengan tujuan dan fungsinya [9]. Penelitian ini merupakan pengembangan lebih lanjut dari penelitian sebelumnya yang hanya berfokus pada pembuatan sistem monitoring suhu dan kelembaban saja. Pada tahap ini, sistem dikembangkan dan dibangun khusus untuk pengujian fitur push notification guna meningkatkan efektivitas sistem monitoring.

Perangkat keras utama yang digunakan meliputi sensor suhu dan kelembaban DHT11 yang dikoneksikan dengan mikrokontroler ESP32. Dari sisi perangkat lunak, diperlukan akun Firebase sebagai repositori data *real-time* dan platform OneSignal untuk manajemen distribusi notifikasi ke perangkat Android. Dalam sistem ini, ditentukan ambang batas parameter lingkungan sebagai pemicu notifikasi, yaitu suhu di atas 35°C dan kelembaban di atas 70%. Penetapan angka tersebut didasarkan pada asumsi bahwa nilai tersebut merupakan batas optimal kenyamanan suhu ruangan dan indikasi kelembaban tinggi yang memerlukan penanganan lebih lanjut agar tidak merusak perangkat atau mengganggu aktivitas di area monitoring.

Untuk memastikan fungsionalitas tersebut, skenario pengujian dirancang sebagai berikut:

1. Pengujian Kondisi Normal: Mengamati apakah data sensor terkirim ke Firebase secara *real-time* tanpa memicu notifikasi saat parameter berada di bawah ambang batas.
2. Pengujian Ambang Batas (Trigger): Memberikan perlakuan panas atau uap pada sensor untuk mencapai suhu $>35^{\circ}\text{C}$ atau kelembaban $>70\%$ guna menguji kecepatan dan keberhasilan OneSignal dalam mengirimkan Push Notification ke aplikasi Android.

2.2. Perancangan Sistem

Tahap ini bertujuan untuk menyusun kerangka teknis yang komprehensif agar sistem yang dibangun tepat guna, efisien, dan menjawab seluruh kebutuhan pengguna. Tahapan ini meliputi pembuatan context diagram, data flow diagram, flowchart dan blok diagram.

2.3. Implementasi dan Pengkodean

Tahap ini meliputi pembuatan hardware dan software. Dimana pada sisi hardware melibatkan perangkaan sensor dan esp32, serta penulisan kode program pada Arduino IDE untuk proses kerja ESP32. Pada sisi software melibatkan penyusunan blok program pada Kodular. Logika pemrograman terfokus pada efisiensi pengiriman data agar tidak terjadi penumpukan permintaan pada Firebase yang dapat menyebabkan latensi. Pembuatan *real-time database* untuk setiap perubahan data suhu dan kelembaban serta Firebase Cloud Messaging untuk keperluan Push Notification. Melakukan penyetingan oneSignal Push Notification agar dapat terkoneksi dengan Firebase dan aplikasi yang dibuat dengan Kodular.

2.4. Pengujian

Tahap akhir adalah pengujian sistem secara menyeluruh yang mencakup:

- a. Pengujian sensor DHT11 dalam melakukan pembacaan suhu dan kelembaban
- b. Pengujian database Firebase saat menerima data suhu dan kelembaban
- c. Pengujian aplikasi Android saat update data suhu dan kelembaban
- d. Pengujian notifikasi saat nilai suhu atau kelembaban melewati batas

3. HASIL DAN PEMBAHASAN

Prosedur penelitian dilakukan melalui empat tahapan sistematis, yaitu identifikasi kebutuhan, perancangan sistem, penerapan (implementasi), serta pengujian (evaluasi).

3.1. Identifikasi Kebutuhan

Pada fase ini, dilakukan pengaturan terhadap spesifikasi perangkat dan mekanisme operasional sistem untuk memastikan semua fitur optimal. Alat dan bahan yang digunakan pada penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Alat dan Bahan

No	Kebutuhan
1	NodeMCU ESP32
2	DHT11
3	Kabel Jumper
4	Power Supply 9V 1A
5	SmartPhone Android
6	Laptop
7	Internet

Mekanisme Kerja Sistem: Secara operasional, sensor akan menangkap data suhu dan kelembaban yang kemudian diproses oleh ESP32 untuk dikirimkan ke Firebase. Setiap perubahan data suhu dan kelembaban akan langsung di kirimkan ke aplikasi android yang dibuat dengan kodular, sehingga pengguna dapat memonitor setiap perubahan tersebut. Apabila hasil pembacaan melampaui batas parameter yang ditentukan yaitu pada suhu 35°C atau nilai kelembaban pada 70%,

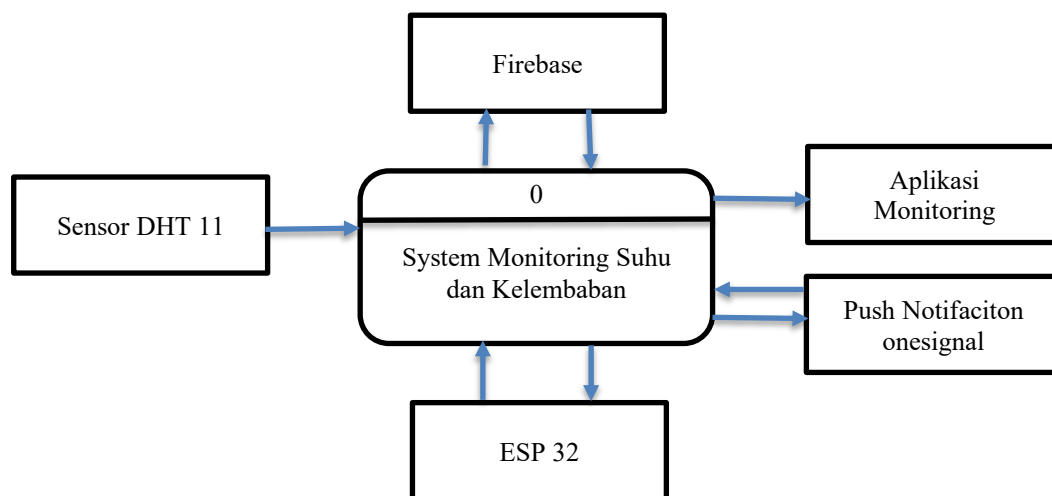
maka sistem akan mendeteksi bahwa keadaan sedang bahaya. Keadaan tersebut akan memicu OneSignal untuk mengirimkan peringatan (Push Notification) secara instan ke ponsel pengguna sebagai bentuk sistem peringatan dini (*early warning system*). Hal ini berguna jika pengguna tidak sedang menjalankan aplikasi pada smartphonanya.

3.2. Perancangan Sistem

Secara umum, Sistem yang dibangun pada sistem pengontrolan pengeringan ikan ini dapat digambarkan secara umum pada Context Diagram, Data Flow Diagram, Flowchart dan Blok Diagram dibawah ini.

1. Context Diagram

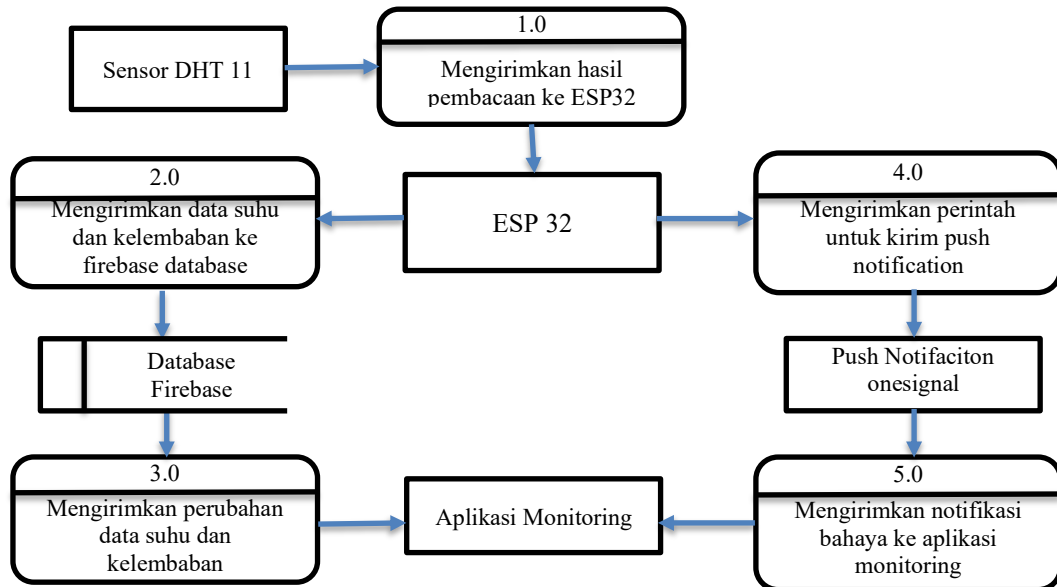
Context Diagram adalah diagram sederhana yang menggambarkan hubungan antara *entity* luar, masukan dan keluaran dari sistem[10]. Context Diagram pada sistem monitoring suhu dan kelembaban ruangan terdapat proses dan 5 entitas eksternal. Gambar dari Context Diagram sistem ini dapat dilihat pada Gambar 2.



Gambar 2. Context Diagram

2. Data Flow Diagram

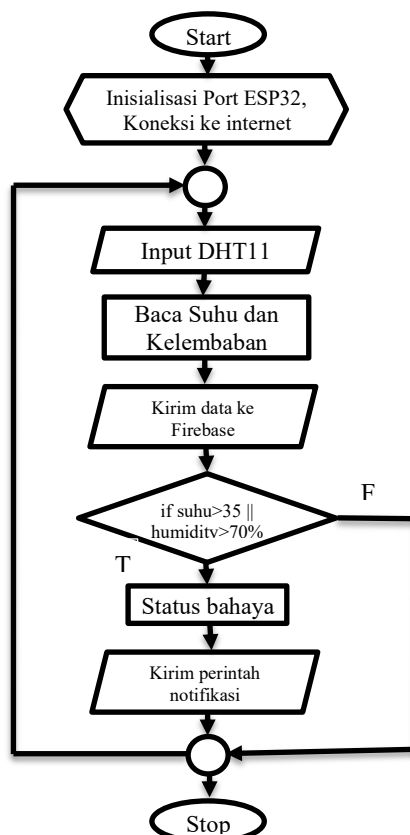
Dataflow Diagram membantu memahami bagaimana data diproses dan dipindahkan di seluruh sistem, memastikan efisiensi dan keakuratan dalam pengolahan informasi[11]. Diagram aliran data sistem monitoring suhu dan kelembaban memiliki 5 subproses. Dimana pada diagram ini ditampilkan aliran data dimulai dari subproses pertama hingga akhir. Lebih jelasnya dapat dilihat pada Gambar 3.



Gambar 3. Data Flow Diagram

3. Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program[12]. Dibandingkan meninjau kode sumber secara mendalam, penggunaan Flowchart jauh lebih efektif bagi pengembang lain untuk menangkap alur program. Bentuk Flowchart programnya adalah seperti Gambar 4.



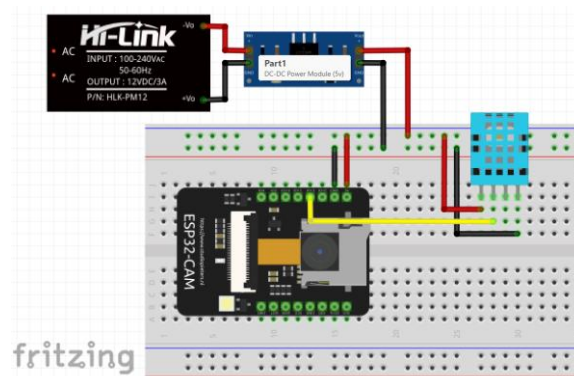
Gambar 4. Flowchart Sistem

3.3. Implementasi

Selanjutnya memasuki tahap implementasi pada sistem sistem monitoring suhu dan kelembaban yang terdiri dari dua tahap. Tahap pertama adalah pembuatan hardware. Tahap kedua pembuatan software, baik untuk pembuatan database Firebase, pembuatan aplikasi android dengan kodular serta pembuatan push notification dengan onesignal.

1. Perakitan Perangkat Keras

Meliputi proses instalasi komponen elektronik pada papan sirkuit, termasuk memastikan keakuratan sensor pembacaan melalui pengaturan catu daya yang stabil pada ESP32. Gambar rangkaian sensor DHT11 dan ESP32 dapat dilihat pada Gambar 5.



Gambar 5. Gambar Rangkaian Sistem

2. Pengembangan Perangkat Lunak

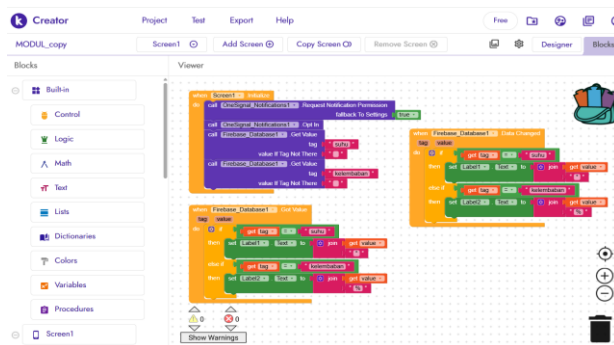
Pada tahap pengembangan perangkat lunak terdapat beberapa tahap, yaitu pembuatan database Firebase, pembuatan aplikasi Android menggunakan Kodular, Push Notification pada Onesignal serta pembuatan program ESP32 pada IDE Arduino. Penjelasan untuk setiap tahapnya akan dijabarkan seperti berikut:

- **Integrasi Firebase Real-time Database**

Firebase difungsikan sebagai infrastruktur pusat (hub) yang menjamin sinkronisasi data secara kontinyu antara ESP32 dan aplikasi Android. Integrasi ini dibangun menggunakan alamat URL dan token otentikasi unik yang ditanamkan pada firmware ESP32, sehingga setiap perubahan nilai suhu dan kelembaban pada sisi sensor langsung memicu pembaruan status pada database tanpa memerlukan proses request manual.

- **Kodular**

Pengembangan antarmuka Android untuk sistem monitoring suhu dan kelembaban ini menggunakan platform Kodular. Pemilihan platform *low code* Kodular didasarkan terutama pada efisiensi pengembangan dan kemudahan penggunaan, bukan pada performa aplikasi yang dihasilkan secara inheren. Proses pengembangan aplikasi pada Kodular menggunakan sistem pemrograman berbasis blok[13]. Penggunaan sistem ini secara signifikan mempercepat penyusunan logika program. Bentuk pemrograman block dapat dilihat pada Gambar 6.



- **Mekanisme Push Notification OneSignal**

OneSignal adalah layanan pihak ketiga (platform SaaS) yang digunakan untuk mengirimkan pesan singkat atau notifikasi dari server secara langsung ke *smartphone* pengguna yang telah terinstal aplikasi monitoring secara *real-time*[14]. OneSignal diimplementasikan sebagai jembatan komunikasi aktif yang bekerja secara independen di latar belakang sistem operasi Android. Kontribusi utamanya adalah menjamin distribusi notifikasi peringatan dini tetap terkirim ke *smartphone* pengguna meskipun aplikasi sedang tidak dijalankan, sehingga mengatasi kelemahan pada sistem monitoring pasif konvensional.

- **Mekanisme Transmini Data ESP32**

Proses pemrograman pada board ESP32 merupakan langkah krusial guna menjamin fungsionalitas system, baik dalam akurasi pembacaan data sensor suhu dan kelembapan serta efisiensi transmisi data ke cloud database Firebase. Data suhu dan kelembapan yang telah diakuisisi oleh sensor DHT11 dikirimkan oleh ESP32 ke Firebase Realtime Database melalui protokol *HTTPS Post*. Integrasi ini memanfaatkan pustaka *Firebase-ESP32* yang mengamankan transmisi menggunakan *database URL* dan *API Key* (token otentikasi) unik. Proses ini memastikan setiap perubahan nilai sensor di sisi *hardware* langsung tersinkronisasi ke *cloud database* secara instan.

3.4. Pengujian

Pengujian merupakan tahap terakhir dalam penelitian ini. Pada tahap ini semua komponen hardware dan software akan diuji. Mulai dari pengujian sensor DHT11, pengujian Database Firebase, pengujian Aplikasi Android saat *update* data, dan pengujian latensi pengiriman notifikasi dari OneSignal Push Notification. Semua tahap tersebut dapat dilihat pada penjabaran berikut :

1. Pengujian Sensor DHT11

Pengujian sensor DHT11 dalam membaca suhu dan kelembaban dilakukan dengan memanfaatkan fitur Serial Monitor pada Arduino IDE. Fitur seperti serial monitor pada Arduino IDE memungkinkan peneliti untuk melakukan debugging secara real-time, memantau keluaran data, dan memastikan sistem bekerja dengan benar[15]. Dalam prosesnya, pengujian dilakukan dengan memberikan stimulus berupa sumber panas dan uap air untuk mengamati respons serta perubahan data yang ditangkap oleh sensor. Hasil dari pengujian tersebut kemudian dicatat dan disajikan dalam Tabel 2.

Tabel 2. Hasil Pengujian Sensor DHT11

No	Waktu Pengujian	Input Kondisi Lingkungan	Pembacaan Suhu (°C)	Pembacaan Kelembaban (%)	Status Sensor
1	09:15:20	Suhu Ruangan Normal	29.00	62	Berhasil
2	09:18:45	Paparan Uap Air (Humid)	29.00	72	Berhasil
3	09:25:12	Paparan Panas (Korek Api)	35.00	58	Berhasil
4	09:26:55	Paparan Panas (Korek Api)	36.00	55	Berhasil
5	09:34:30	Lingkungan Lembab	30.00	74	Berhasil
6	09:36:10	Paparan Panas (Korek Api)	34.00	50	Berhasil
7	09:38:05	Lingkungan Lembab	31.00	71	Berhasil

2. Pengujian Database Firebase (Sinkronisasi Cloud)

Proses pengujian ini bertujuan untuk memvalidasi keberhasilan ESP32 dalam mengirimkan data ke Firebase Realtime Database. Pengujian dilakukan dengan memantau secara saksama setiap perubahan data sensor yang muncul pada Serial Monitor dan mencocokkannya dengan perubahan data yang terjadi secara langsung pada database. Dengan membandingkan kedua output tersebut secara *real-time*, peneliti dapat memastikan bahwa integrasi sistem komunikasi data antara perangkat keras dan platform cloud telah berjalan dengan akurat dan stabil. Hasil pengujian database firebase dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian Database Firebase

No	Data Sensor (S/K)	Data Terupdate di Firebase	Latensi Firebase (dtk)	Status Update
1	29.00 / 62	29.00 / 62	0.45	Sukses
2	29.00 / 72	29.00 / 72	0.52	Sukses
3	35.00 / 58	35.00 / 58	0.85	Sukses
4	36.00 / 55	36.00 / 55	0.60	Sukses
5	30.00 / 74	30.00 / 74	0.72	Sukses
6	34.00 / 50	34.00 / 50	0.48	Sukses
7	31.00 / 71	31.00 / 71	0.65	Sukses
Rata-rata			0.61 detik	

3. Pengujian Aplikasi Android Saat Update Data

Pengujian ini bertujuan untuk memvalidasi fungsionalitas fitur *listener* pada aplikasi Kodular dalam menampilkan data terbaru dari Firebase Realtime Database secara otomatis. Proses pengujian dilakukan dengan mengukur durasi waktu atau latensi, mulai dari saat terjadinya perubahan data pada Firebase hingga data tersebut berhasil diperbarui pada tampilan aplikasi. Melalui metode ini, efektivitas sinkronisasi data secara real-time dapat dipastikan berjalan dengan responsif sesuai dengan kebutuhan pengguna. Hasil pengujian update data pada aplikasi dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian Aplikasi Saat Update Data

No	Nilai pada Firebase	Nilai Muncul di Aplikasi	latensi (dtk)	Status Update
1	30.00 / 62	30.00 / 62	0.25	Sukses
2	34.00 / 55	34.00 / 55	0.30	Sukses
3	29.00 / 71	29.00 / 71	0.45	Sukses
4	28.00 / 67	28.00 / 67	0.28	Sukses
5	27.00 / 55	27.00 / 55	0.35	Sukses
6	30.00 / 50	30.00 / 50	0.22	Sukses
7	31.00 / 51	31.00 / 51	0.40	Sukses
Rata-rata			0.32 detik	

4. Pengujian Notifikasi

Pada tahap terakhir yaitu tahap pengujian notifikasi menggunakan Push Notification OneSignal, dilakukan dengan menaikkan nilai suhu dan kelembaban pada sensor DHT11. Pengujian dilakukan untuk menghitung total waktu (jeda) sejak sensor DHT11 mendeteksi nilai di atas ambang batas (suhu > 35° atau kelembaban > 70) hingga push notification muncul pada smartphone. Hasil pengujian latensi pengiriman notifikasi dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian Latensi Pengiriman Notifikasi Bahaya

No	Kondisi	Waktu Deteksi Sensor	Waktu Notifikasi Diterima	Total Latensi (detik)
1	Suhu > 35°	10:00:01.20	10:00:03.45	2.25
2	Kelembaban > 70%	10:05:15.40	10:05:18.10	2.70
3	Suhu > 35°	10:10:22.15	10:10:26.50	4.35
4	Suhu > 35°	10:13:40.30	10:13:43.05	2.75
5	Kelembaban > 70%	10:19:10.55	10:19:14.20	3.65
6	Suhu > 35°	10:25:05.10	10:25:08.00	2.90
7	Kelembaban > 70%	10:28:12.45	10:28:15.65	3.20
Rata-rata				3.11 e tik

Analisis Data Pengujian :

- Akurasi & Responsivitas
Sensor menunjukkan kemampuan deteksi yang sangat baik terhadap berbagai stimulus (suhu ruangan, uap air, hingga panas ekstrem). Meskipun DHT11 memiliki batasan jeda pembacaan (2 detik), transisi data pada Serial Monitor berjalan stabil tanpa adanya galat pembacaan (*error reading*).
- Efisiensi Sinkronisasi Cloud
Rata-rata latensi sebesar 0,61 detik membuktikan jalur komunikasi antara ESP32 dan Firebase sangat efisien. Nilai latensi tertinggi (0,85 detik) masih berada di bawah ambang batas 1 detik, yang berarti protokol HTTPS Post yang digunakan berhasil mengirimkan paket data sensor secara instan tanpa hambatan trafik yang berarti.
- Kecepatan Update Interface
Rata-rata latensi 0,32 detik menunjukkan fitur Real-time Listener pada Kodular bekerja dengan performa tinggi. Pembaruan data yang terjadi hampir seketika (instan) saat aplikasi terbuka memastikan pengguna selalu mendapatkan informasi kondisi lingkungan terbaru tanpa perlu melakukan *refresh* manual.
- Kecepatan Respon
Rata-rata waktu transmisi notifikasi sebesar 3.11 detik menunjukkan bahwa sistem ini sangat layak disebut sebagai *early warning system*. Jeda waktu yang singkat memungkinkan pengguna mengambil tindakan preventif dengan cepat sebelum kerusakan material terjadi.

- Stabilitas Sistem
Berdasarkan pengujian, tingkat keberhasilan pengiriman adalah 100%, yang menandakan integrasi antara ESP32, Firebase, dan OneSignal memiliki stabilitas yang baik untuk pemantauan jangka panjang.

Terdapat beberapa faktor kunci yang memengaruhi kecepatan notifikasi dalam sistem ini meliputi:

- Kualitas Sinyal Wi-Fi
Kekuatan sinyal pada mikrokontroler ESP32 memengaruhi kecepatan transmisi data awal. Gangguan frekuensi atau kepadatan lalu lintas internet pada Internet Service Provider (ISP) dapat menyebabkan packet loss, sehingga terjadi proses pengiriman ulang data yang menambah durasi latensi.
- Interval Reading Sensor
Kecepatan sensor DHT11 dalam melakukan pembacaan ulang (biasanya memiliki jeda minimal 2 detik antar pembacaan) juga berkontribusi pada total waktu deteksi hingga eksekusi program.
- Server-Side Processing (Firebase & OneSignal)
Karena sistem menggunakan layanan pihak ketiga, data harus menempuh perjalanan melalui *gateway* internasional menuju server Firebase dan OneSignal (biasanya di Singapura atau Amerika Serikat). Beban trafik pada server pusat atau proses antrean (*queueing*) pada layanan *Firebase Cloud Messaging* (FCM) sering kali menjadi penyebab utama fluktuasi latensi.
- Manajemen Daya Smartphone
Fitur optimasi baterai pada Android terkadang menunda munculnya notifikasi jika perangkat dalam kondisi *deep sleep*, yang dapat menambah sedikit jeda waktu pada sisi penerima.
- Beban Kerja Smartphone
Jika smartphone penerima sedang menjalankan aplikasi berat di latar belakang, terjadi antrean pemrosesan pada CPU ponsel yang mengakibatkan notifikasi tidak muncul seketika meskipun data sudah diterima oleh sistem.

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa:

1. Penelitian ini telah berhasil merancang dan membangun sistem monitoring suhu dan kelembapan berbasis IoT dengan mengintegrasikan mikrokontroler ESP32, sensor DHT11, Firebase Realtime Database dan dilengkapi dengan OneSignal Push Notification yang memberikan peringatan saat aplikasi monitoring tidak sedang berjalan.
2. Pengujian sensor DHT11 melalui Serial Monitor Arduino IDE membuktikan bahwa alat mampu mendeteksi perubahan suhu dan kelembapan secara akurat saat diberi stimulus panas maupun uap air.
3. Pengujian menunjukkan bahwa ESP32 mampu melakukan transmisi data ke Firebase menggunakan token otentikasi dengan tingkat keberhasilan 100% dan latensi rata-rata yang rendah (sekitar 0,61 detik), memastikan keamanan dan integritas data secara realtime.
4. Fitur real-time listener pada aplikasi Kodular memiliki performa tinggi dengan rata-rata waktu pembaruan data sebesar 0,32 detik, sehingga informasi pada ponsel pengguna selalu update tanpa perlu refresh manual.
5. Pengujian notifikasi melalui OneSignal berhasil mengirimkan peringatan bahaya ke smartphone dengan rata-rata jeda waktu 3,11 detik saat parameter melewati ambang batas (suhu > 35°C atau kelembapan > 70%), sehingga sistem ini layak digunakan sebagai perangkat peringatan dini (*early warning system*) yang efektif.

5. KETERBATASAN SISTEM

Meskipun sistem monitoring suhu dan kelembaban berbasis IoT ini telah berhasil diimplementasikan dan menunjukkan performa yang baik dalam pengiriman notifikasi bahaya, terdapat beberapa keterbatasan fundamental yang perlu diidentifikasi dan dibahas secara terbuka:

1. Ketergantungan pada Koneksi Internet
Sistem ini dirancang sepenuhnya dengan mengadopsi konsep *Internet of Things* (IoT), di mana komunikasi antara perangkat keras (ESP32) dengan *cloud database* (Firebase) dan layanan notifikasi (OneSignal) bergantung penuh pada koneksi internet yang stabil. Apabila terjadi pemadaman jaringan Wi-Fi lokal atau gangguan pada ISP, sistem akan kehilangan kemampuan untuk mengirim data dan notifikasi secara *real-time*. Data sensor tidak akan terbaru di aplikasi Android, dan peringatan bahaya tidak akan sampai ke pengguna.
2. Menggunakan Layanan Pihak Ketiga
Seluruh arsitektur perangkat lunak dibangun di atas layanan SaaS (Software as a Service) gratis, yaitu Firebase dan OneSignal. Performa sistem sangat dipengaruhi oleh kebijakan, beban server, dan stabilitas layanan dari Google (Firebase) dan OneSignal.
3. Batasan Akurasi dan Kecepatan Sensor DHT11
Pemilihan sensor DHT11 didasarkan pada pertimbangan biaya dan kemudahan integrasi. Namun, sensor ini memiliki batasan akurasi dan kecepatan *sampling* yang lebih rendah dibandingkan sensor kelas *industry*. Terdapat jeda minimal 2 detik untuk setiap pembacaan sensor, yang menjadi batasan paling dasar dalam kecepatan respons sistem secara keseluruhan pada sisi *hardware*.
4. Keterbatasan Platform Pengembangan Aplikasi (Kodular)
Aplikasi Android dikembangkan menggunakan *platform low-code* Kodular, yang dipilih karena efisiensi pengembangan dan kemudahan penggunaan. Meskipun fungsionalitas utama tercapai, *platform low-code* memiliki keterbatasan dalam kustomisasi tingkat lanjut, optimasi performa *native* aplikasi, dan skalabilitas jika dibandingkan dengan pengembangan aplikasi secara manual menggunakan kode penuh (misalnya, Android Studio)

DAFTAR PUSTAKA

- [1] S. Viola Widyasari, M. Ihsan Muttaqin, T. Putri Ananda, and A. Stefanie, "Implementasi Internet of Things Pada Sistem Monitoring Kematangan Buah Pepaya California Dengan Metode Deep Learning," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 7, no. 3, pp. 1946–1952, 2023, doi: 10.36040/jati.v7i3.6953.
- [2] M. F. Azhar and L. Nurpulaela, "Implementasi Penggunaan ESP32 Sebagai IOT Pada Project Smart Charger di PT. Pasifik Satelit Nusantara Bekasi," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 8, no. 4, pp. 7248–7253, 2024, doi: 10.36040/jati.v8i4.10201.
- [3] F. Nuryadi, N. Wayan, P. Septiani, M. Lestari, and P. Tanah, "Implementasi Esp32 Untuk Sistem Pemantauan Kesuburan Tanah Berbasis IoT," SEMNAS RISTEK, vol. 9, no. 1, pp. 79–86, 2025, doi: 10.30998/semnasristek.v9i1.7550.
- [4] A. Anisa, S. Paembonan, and R. Suppa, "Rancang Bangun Alat Pengukur Tinggi Badan Otomatis Menggunakan Arduino Pada Posyandu Kelurahan Sampoddo," Jurnal Informatika Dan Teknik Elektro Terapan, vol. 13, no. 1, 2025, doi: 10.23960/jitet.v13i1.5792.
- [5] A. Sutanti, M. K. MZ, Mustika, and P. Damayanti, "Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur," KOMPUTA, vol. 9, no. 1, pp. 1-8, 2020, doi: 10.34010/komputa.v9i1.3718.
- [6] H. Kurnia AR, "Implementasi Iot Pada Sistem Monitoring Suhu Dan Kelembaban Menggunakan Esp32, Firebase Dan Kodular," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 9, no. 1, pp. 1781–1787, 2025, doi: 10.36040/jati.v9i1.12874.

-
- [7] M. Muharir and N. Alamsyah, "Pemanfaatan Mobile Push Notification Dalam Penyampaian Informasi Perkuliahan Mahasiswa Pada Fakultas Teknologi Informasi Berbasis Android," *Technologia: Jurnal Ilmiah*, vol. 11, no. 2, p. 111, 2020, doi: 10.31602/tji.v11i2.2860.
- [8] L. Adytia Rahmadan, S. Zakir, L. Efriyanti, and S. Supriadi, "Perancangan Media Pembelajaran Kewirausahaan Berbasis Android Menggunakan Kodular Pada Kelas Xii Man 1 Agam," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1932-1938, 2023, doi: 10.36040/jati.v7i3.7728.
- [9] P. A. Wijaya, N. David, M. Veronika, and Toyib, "Pengembangan Aplikasi Toko Baju Dengan Sistem Kodular Berbasis Android," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 6, pp. 10055-10063, 2025, doi: 10.36040/jati.v9i6.15791.
- [10] G. A. D.S.S, "Analisa Kebutuhan Kebutuhan Sistem Informasi Manajemen Perusahaan Dagang," *Kurawal - Jurnal Teknologi, Informasi Dan Industri*, vol. 4, no. 1, pp. 17-30, 2021, doi: 10.33479/kurawal.v4i1.400.
- [11] N. Hasanah and A. Purnomo, "Implementasi Data Mining Untuk Pengelompokan Buku Menggunakan Algoritma K-Means Clustering (Studi Kasus : Perpustakaan Politeknik LPP Yogyakarta)," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 4, no. 2, pp. 300-311, 2022, doi: 10.47233/jteksis.v4i2.499.
- [12] M. A. H. Mubarak and A. Hamdani, "Perancangan Sistem Informasi Data dan Kepegawaian di UPTD Dinas Pendidikan Kecamatan Asembagus," *JUSIFOR : Jurnal Sistem Informasi Dan Informatika*, vol. 3, no. 2, pp. 116-124, 2024, doi: 10.70609/jusifor.v3i2.5354.
- [13] R. A. Setyawan, "Penerapan Firebase Realtime Database Pada Aplikasi Catatan Harian Diabetes Melitus," *Jurnal Informatika Komputer, Bisnis Dan Manajemen*, vol. 22, no. 1, pp. 1-9, 2024, doi: 10.61805/fahma.v22i1.102.
- [14] P. Danito, A. A. Ketut, A. Cahyawan Wiranatha, M. Agus, and D. Suarjaya, "Aplikasi Radio Online Universitas Udayana Berbasis Android," *JITTER-Jurnal Ilmiah Teknologi Dan Komputer*, vol. 1, no. 2, 2020, doi: 10.24843/jitter.v1i2.69031.
- [15] A. Mahardika and M. Ihsan, "Rancang Bangun Smart Door Menggunakan ESP32 CAM Berbasis Internet of Things (IoT)," *Jurnal Digital Teknologi Informasi*, vol. 7, no. 2, pp. 6-13, 2024, doi: 10.32502/digital.v7i2.9299.