Penerapan *Firebase Realtime Database* Untuk Monitoring Cuaca Secara *Realtime*

1st Faris Candra Yoga Pratama
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
fariscyp@student.telkomuniversity.ac.id

2nd Nachwan Mufti Adriansyah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
nachwanma@telkomuniversity.ac.id

3rd Vinsensiun Sigit Widhi Prabowo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
vinsensiusvsw@telkomuniversity.ac.id

***Abstrak* — Perubahan iklim yang semakin nyata berdampak besar pada berbagai aspek kehidupan, termasuk keamanan nasional, geopolitik, dan produksi pangan. Oleh karena itu, pemantauan cuaca secara real-time menjadi sangat penting. Sebagai solusi, telah dikembangkan sistem *Weather Station* yang terintegrasi dengan aplikasi mobile dan situs web menggunakan *firebase realtime database* sebagai tempat penyimpanan berbasis *cloud*. Hasil uji *latency* menunjukkan kestabilan penerimaan data dan pengiriman data dengan rentang 3 – 7 detik untuk penerimaan pada aplikasi *mobile* dan website.**

***Kata kunci*— cuaca, *weather station, firebase realtime database, latency*.**

I. PENDAHULUAN

Iklim dan cuaca memiliki pengaruh signifikan terhadap kehidupan manusia dalam berbagai aspek, seperti keamanan nasional, geopolitik, dan produksi pangan. Perubahan iklim dapat mempengaruhi konflik sumber daya alam, migrasi manusia, serta memaksa pemerintah untuk menyesuaikan kebijakan mereka. Dalam produksi pangan, misalnya, iklim dapat mempengaruhi musim tumbuh tanaman dan hasil pertanian, serta cuaca ekstrem seperti kekeringan atau banjir yang dapat merusak tanaman dan mengancam pasokan pangan. Selain itu, cuaca buruk juga dapat mengganggu transportasi, menyebabkan penundaan perjalanan dan gangguan logistik.

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) menyatakan, terdapat banyak perubahan iklim cuaca yang telah tercatat di Indonesia. Mulai dari perubahan panas hingga curah hujan yang berubah-ubah dari waktu ke waktu. Di Indonesia, rata-rata suhu normal yang tercatat pada tahun 1991 – 2020 tercatat sebesar 26,8^{o}C. Pada tahun 2016 merupakan tahun terpanas diantara tahun lainnya dengan anomali hingga 0,8^{o}C [1].

Sehingga dari permasalahan perubahan cuaca memunculkan kekhawatiran terkait dengan aspek-aspek kehidupan manusia yang dapat terpengaruh. Seperti pada aspek lingkungan yang memunculkan kekhawatiran jika perubahan cuaca terjadi secara tiba-tiba dapat memunculkan bencana alam seperti banjir karena hujan atau kebakaran karena cuaca yang sangat panas. Selain itu, aspek infrastruktur juga dapat terpengaruh akibat adanya perubahan cuaca yang secara langsung.

Oleh karena itu, diperlukan adanya stasiun cuaca yang memiliki kehandalan untuk mengirim informasi data-data cuaca secara cepat dan dapat diakses dimana saja. Selain itu, dibutuhkan juga suatu penyimpanan yang menjadi tempat keseluruhan data-data tersimpan yang memiliki tingkat keamanan yang tinggi serta mampu terintegrasi dengan aplikasi *mobile* dan juga *website* secara *realtime* dengan *latency* yang rendah.

II. KAJIAN TEORI

A. *Firebase Realtime Database*

Firebase merupakan layanan yang dikeluarkan oleh Google yang memberikan kemudahan untuk developer suatu aplikasi atau *website* untuk mengembangkan serta mempercepat proyek yang dikerjakan. Seorang *engineer* aplikasi atau *website* bisa lebih fokus dalam pengerjaan tugasnya tanpa harus memberikan usaha lebih dalam mengatasi *backend*. *Firebase* memiliki banyak fitur-fitur menarik yang dapat digunakan dengan pilihan gratis dan berbayar. Salah satu fiturnya adalah *firebase realtime database*.

Dilansir dari laman dokumentasi resmi *Firebase* [2], *firebase realtime database* merupakan suatu layanan berbasis *cloud* yang di kembangkan oleh Google. Fitur ini memiliki kemampuan untuk menyimpan serta mensinkronisasi data antara server dengan *client* secara *realtime*. Sistem ini memiliki beberapa parameter sistem kerja.

1. Struktur Data

Firebase Realtime Database menggunakan struktur data JSON (*JavaScript Object Notation*) yang memungkinkan data untuk disimpan dalam format *tree* atau pohon dengan *node* dan sub-*node*.

2. Operasi CRUD (*Create, Read, Update, Delete)*

CRUD merupakan operasi dasar yang digunakan dalam sistem pengelolaan data atau tempat penyimpanan data. *Create* digunakan untuk membuat dan menambah data baru. *Read* digunakan untuk membaca atau mengambil

suatu data dari berada pada *node* tertentu. *Update* digunakan untuk memperbaharui data yang telah tersimpan pada penyimpanan atau mengubah nilai yang telah tersimpan. *Delete* digunakan untuk menghapus data atau nilai tertentu yang sudah tersimpan.

3. Sinkronisasi *Realtime*

Penggunaan *Websocket* membantu *firebase realtime database* untuk tetap terhubung dengan *client* dan juga server sehingga memungkinkan adanya komunikasi secara *realtime*.

4. Offline Support

Terdapat mode *offline* yang membuat data-data yang telah tersimpan tidak akan terhapus dan akan memperbaharui kembali ketika perangkat sudah nyala atau *online.*

5. Keamanan dan Autentikasi

Memiliki fitur *rules* yang membantu untuk konfigurasi keamanan pada data yang tersimpan. Terdapat juga integrase *firebase authentication* yang memungkinkan untuk pengambilan data hanya pada *client* yang telah terautentikasi.

6. Skalabilitas

Firebase Realtime Database memiliki kemampuan untuk dapat terintegrasi dengan banyak koneksi atau *client* secara bersamaan dengan tetap menjaga performa yang tinggi.

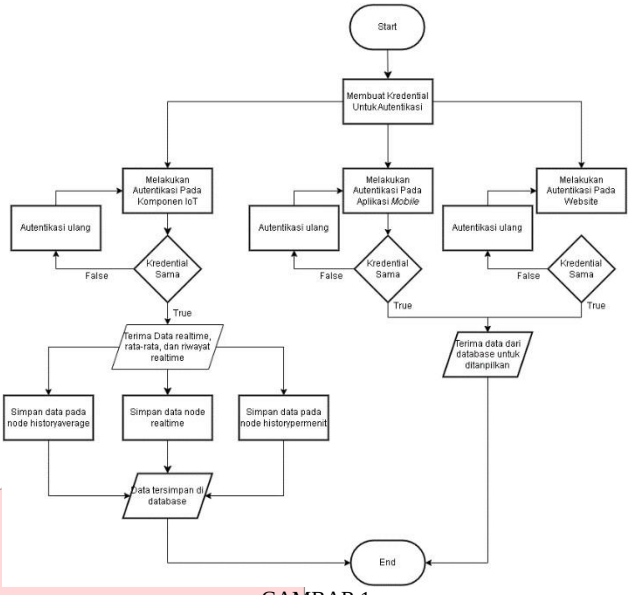
B. Perangkat IoT

Internet of Things (IoT) adalah suatu konsep pemanfaatan komponen digital yang memiliki tujuan untuk memperluas manfaat dari konektifitas internet yang tersambung dalam rentang waktu yang lama [3]. Pemanfaatan IoT dapat di implementasikan pada pembuatan *weather station* yang berguna untuk mendeteksi cuaca. *Weather station* menggunakan mikrokontroler dan sensor untuk mendeteksi parameter-parameter cuaca yang telah ditentukan.

Perangkat IoT yang digunakan pada implementasi ini antara lain mikrokontroler ESP32 sebagai pusat pengolahan data, sensor DHT11 sebagai pendeteksi parameter suhu dan kelembaban cuaca, sensor BMP180 sebagai pendeteksi parameter tekanan udara, dan sensor *raindrop* sebagai pendeteksi parameter hujan. Selain itu, terdapat aplikasi *mobile* yang dibuat menggunakan MIT APP Inventor dan juga *website* yang dibuat menggunakan bahasa pemrograman Golang.

III. METODE

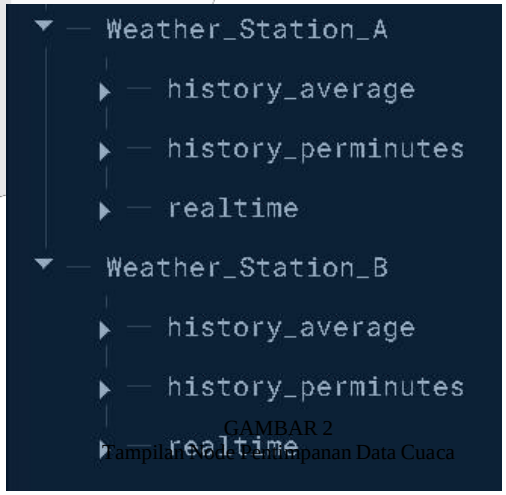
A. Desain Sistem



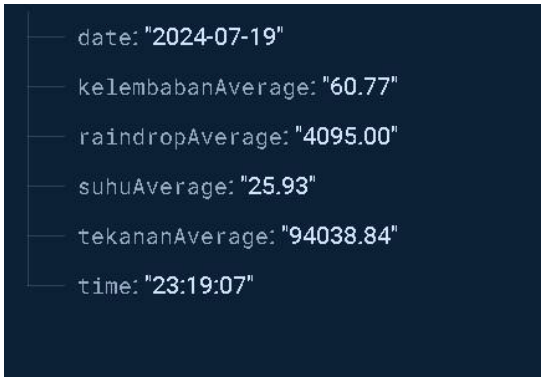
GAMBAR 1
Desain Implementasi Firebase Realtime Database

Gambar 1 merupakan langkah proses desain implementasi pada *firebase realtime database* yang tarsambung dengan komponen perangkat IoT, aplikasi *mobile*, dan *website*. Sistem akan terbagi menjadi tiga cabang untuk setiap komponen-komponen yang tersambung dengan *firebase realtime database*. Sistem dimulai dengan pembuatan kredential khusus yang digunakan untuk autentikasi. Kredential akan diberikan secara langsung setelah pembuatan *firebase realtime database* dibuat dan dapat diakses pada *console firebase.*

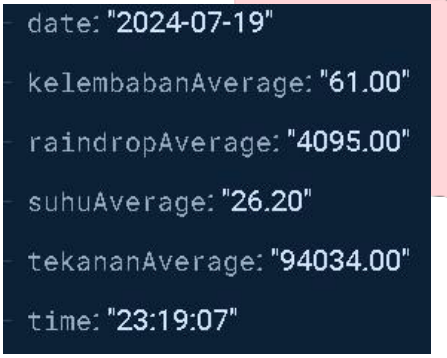
Pada komponen perangkat IoT, mikrokontroler ESP32 akan mengirim kredential yang telah diterima untuk proses verifikasi autentikasi. Setelah dikonfirmasi, maka sistem *firebase realtime database* memperbolehkan mikrokontroler ESP32 untuk mengirim serta menyimpan data-data cuaca yang telah dideteksi oleh setiap sensor. Mikrokontroler ESP32 juga dapat membuat *node* baru yang diperlukan sebagai tempat penyimpanan data parameter cuaca.



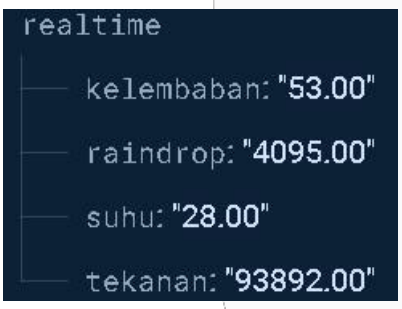
GAMBAR 2
Tampilan Node Penyimpanan Data Cuaca



GAMBAR 3
Data Rata-rata Tersimpan



GAMBAR 4
Data Riwayat Permenit Tersimpan



GAMBAR 5
Data cuaca realtime

Gambar 2 menunjukkan *node* dari 2 perangkat yang berbeda. Data yang diterima oleh *firebase realtime database* dari mikrokontroler ESP32 memiliki 3 *node* yang berbeda di masing-masing perangkat. Terdapat *node* "history average" yang berisikan data rata-rata dari setiap parameter-parameter cuaca yang dideteksi dan telah di olah. Pengolahan data tersebut dilakukan dengan menghitung nilai selama 30 menit seperti pada gambar 3. *Node* yang kedua adalah *node* "history permenit" yang berisikan data hasil monitoring yang dilakukan secara *realtime* seperti pada gambar 4. Dan *node* yang ketiga adalah *node* "*realtime"* yang merupakan *node* untuk monitoring data cuaca *realtime*.

IV. HASIL DAN PEMBAHASAN

Pengujian pengecekan *latency* atau *delay* pengiriman data dilakukan untuk mengetahui jika sistem *weather station* dapat berjalan secara optimal. Sistem *weather station* yang optimal memeliki kemampuan untuk mengirimkan data secara *realtime* dengan cepat. Namun, disetiap pengiriman data pasti selalu terdapat *latency* yang terjadi. Sehingga telah ditetapkan target *latency* diharapkan dapat terkirim dengan keterlambatan 1-5 detik dan tidak lebih dari itu sebagai pengiriman secara *realtime*.

Pengujian dilakukan dengan memlihat data *history* yang telah tersimpan pada *database firebase* dan membandingkan dengan waktu saat data pada aplikasi *mobile* dan *website* terupdate. Koneksi internet antara komponen IoT dan perangkat untuk pengecekan aplikasi *mobile* dan *website* menggunakan koneksi yang sama untuk meningkatkan kualitas pengecekan.

TABEL 1
Pengujian Latency Firebase Realtime Database

No	Waktu Data Terkirim (Firebase)	Waktu Data *Update*		*Latency*	
		Aplikasi *Mobile*	Website	Aplikasi *Mobile*	Website
1	16:53:47	16:53:49	16:53:50	2	3
2	16:54:47	16:54:49	16:54:50	2	3
3	16:55:47	16:55:49	16:55:52	2	5
4	16:56:47	16:56:49	16:56:50	2	3
5	16:57:47	16:57:49	16:57:50	2	3
6	16:58:47	16:58:50	16:58:51	3	4
7	16:59:47	16:59:48	16:59:50	1	3
8	17:00:47	17:00:49	17:00:50	2	3
9	17:01:47	17:01:48	17:01:50	1	3
10	17:02:47	17:02:49	17:02:50	2	3
11	17:03:47	17:03:49	17:03:50	2	3
12	17:04:47	17:04:49	17:04:50	2	3
13	17:05:47	17:05:49	17:05:55	2	7
14	17:06:47	17:06:50	17:06:51	3	4
15	17:07:47	17:07:48	17:07:50	1	3

Dari pengujian *latency* yang telah dilakukan selama 15 menit dengan mengambil sampel data update setiap satu menit seperti pada tabel 1, didapatkan *latency* tertinggi yang didapatkan ketika data terkirim ke aplikasi *mobile* adalah 3 detik. Sedangkan *latency* pengiriman data pada *website* tertinggi didapatkan sebesar 7 detik yang terjadi selama satu kali saja.

V. KESIMPULAN

Pada pengujian *latency* data dari infrastruktur *cloud* menggunakan *firebase realtime database* dengan koneksi internet yang sama antara komponen IoT dengan yang digunakan untuk membuka aplikasi *mobile* dan website. Data yang didapat menunjukkan *latency* data terkirim ke aplikasi *mobile* menunjukkan angka 2 detik sedangkan *latency* data ke website menunjukkan angka 3 detik. Maka dapat disimpulkan bahwa penggunaan *firebase realtime*

database sebagai infrastruktur *cloud* merupakan pilihan yang tepat karena sesuai dengan target yang ditetapkan.

REFERENSI

[1] ”Anomali Suhu Udara Rata-Rata Bulan September 2023”, Badan Meterorologi, Klimatologi, Dan Geofisika (BMKG), 2023. https://www.bmkg.go.id/iklim/?p=ekstrem-perubahan-iklim. [Sep, 15. 2023]

[2] “Dokumentasi,” Firebase, 2022. Available: https://firebase.google.com/docs?hl=id [Apr, 24. 2024]

[3] Y. Efendi, “Internet of Things (IoT) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile,” Jurnal Ilmu Ilmiah Komputer, vol. 4, no. 1, pp 19-26, Apr. 2018.