

# Cloud Computing Menggunakan Firebase Pada Pernacangan Smart Collar Untuk Monitoring Kesehatan Sapi

1<sup>st</sup> Rivanka Abyan Yusuf  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
rivankaay@student.telkomuniversity.ac.id

2<sup>nd</sup> Dharu Arseno  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
darseno@telkomuniversity.ac.id

3<sup>rd</sup> Dhoni Putra  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia  
setiawandhoni@telkomuniversity.ac.id

**Abstrak** — Umumnya kondisi kesehatan sapi adalah bagian hal yang terpenting bagi para peternak. Dan beberapa tahun terakhir khususnya tahun 2022 banyak sekali kasus sapi yang terkena wabah PMK (Penyakit Mulut dan Kuku). Wabah PMK (Penyakit Mulut dan Kuku) di Kabupaten Bandung Barat semakin meluas, nyaris 5.000 ekor ternak terpapar [1]. Dan untuk mencegah terjadinya masalah tersebut maka terdapat sebuah solusi yaitu dengan cara memantau sapi dengan alat yang bisa mendeteksi kondisi kesehatan sapi. *Smart Collar* merupakan salah satu rancangan yang dibuat untuk mengatasi masalah tersebut. Namun, pada pembuatan jurnal kali ini adalah lebih berfokus terhadap sistem cloud yang dimana saling berhubungan dengan Arduino IDE maupun Android Studio yang digunakan untuk melaukan pemantauan berdasarkan data yang didapat dari smart collar. Sehingga dengan parameter-parameter tersebut maka para para peternak dapat mengindikasi ada atau tidaknya gejala pada hewan ternak mereka.

**Kata kunci**— *Smart Collar, firebase, sistem cloud, ternak, sapi*

## I. PENDAHULUAN

Maraknya penyakit pada hewan ternak terutama sapi yang terjadi pada tahun ,dampak ekonomi akibat *outbreak* PMK diperkirakan sebesar Rp 38,67 triliun [2]. Agar tidak terus merugikan perekonomian peternak, maka salah satu solusi yang tergagaskan adalah dengan membuat *Smart Collar* sebagai alat untuk mendeteksi kesehatan pada hewan ternak khususnya sapi [3]. Dan untuk mendapatkan pendeteksian kondisi hewan dengan menggunakan beberapa sensor suhu, detak jantung, jarak, dan sensor gerak [4].

Dan *Smart Collar* ini dirancang dengan tujuan untuk mendeteksi menggunakan beberapa sensor yaitu sensor suhu, sensor jarak, sensor detak jantung, dan sensor gerakan. Semua sensor tersebut tentunya telah dipilih dengan penuh pertimbangan yaitu dengan memantau seta melihat kondisi perilaku dari hewan ternak itu sendiri. Dengan *Smart Collar* inilah maka akan mempermudah pendeteksian gejala penyakit pada hewan ternak tersebut.

Penyimpanan dan pengiriman data hasil menggunakan internet. Dan data hasil itu di simpan dalam sistem *cloud* yaitu *firebase*. Dalam menerima ataupun mengirimkan data, alat bantu lain juga berperan dalam pengambilan data, alat bantu tersebut adalah website dan aplikasi [5]. *Firebase* itu sendiri merupakan database yang tidak bisa digunakan/diakses oleh umum.

Berdasarkan beberapa kebutuhan untuk menyimpan hasil data yang telah didapatkan dari parameter-parameter sensor, maka dibuatlah sistem *cloud* berupa *Firebase* yang berada pada website. Didalam *firebase* ini juga terdapat Autentikasi yang berguna untuk melihat kapan tanggal *user* melakukan registrasi dan kapan tanggal terakhir *user* melakukan *login* pada aplikasi. Dan terdapat pula *Realtime Database* sebagai tempat menyimpan data sementara yang dikirim dari *Smart Collar*.

## II. KAJIAN TEORI

### A. Smart Collar

#### Smart Collar



GAMBAR 1.  
*Smart Collar*

*Smart Collar* merupakan alat untuk mendeteksi kondisi kesehatan dari hewan ternak terutama sapi sebagai objek penelitian kali ini. *Smart collar* dapat mendeteksi dengan cara menempelkannya pada leher dari sapi tersebut dan kemudian sensor-sensor dari *Smart Collar* ini akan membaca misal berapa suhu yang didapat atau berapa detakan jantung permenit yang didapat terhadap sapi. Agar data tersebut dapat dipantau dengan mudah maka diperlukan aplikasi.

*Smart Collar* perangkat yang dapat dikenakan seperti jam tangan pintar itu menjaga kesehatan data manusia. *Smart Collar* mempunyai ukuran besar lebih dari jam

tangan pintar sehingga perangkat keras dapat dibuat lebih mudah dengan perangkat IoT umum [3].

## B. Aplikasi



GAMBAR 2.  
Aplikasi Android Studio

Android Studio merupakan lingkungan pengembangan terintegrasi (IDE) untuk pengembangan platform Android. Dulu diumumkan pada 16 Mei 2013 di konferensi Google I/O oleh Manajer Produk Google, Katherine Chou [7]. Android Studio adalah alat yang sangat penting bagi para pengembang Android, karena akan sangat membantu mengembangkan aplikasi dengan cepat, efisien, dan sesuai standar terbaru dari platform Android.

Secara keseluruhan, tentunya memilih Android Studio sebagai platform pengembangan aplikasi sudah terencana dan melihat dari berbagai aspek seperti menawarkan alat yang kuat, dukungan resmi, dan fitur-fitur khusus untuk pengembangan aplikasi Android, menjadikannya pilihan utama bagi banyak pengembang.

## C. Sistem Cloud



GAMBAR 3.  
Firebase

Firebase adalah platform pengembangan aplikasi berbasis cloud yang disediakan oleh Google. Platform ini dirancang untuk memungkinkan pengembang dengan mudah membangun aplikasi berkualitas tinggi dengan berbagai fitur termasuk penyimpanan data, autentikasi pengguna, analitik, notifikasi, *hosting* web, dan banyak lagi. Firebase mengurangi kerumitan proses pengembangan dengan menyediakan berbagai alat dan layanan siap pakai.

Beberapa fitur yang ada pada Firebase digunakan, seperti fitur *Authentication* dan *Realtime Database*. Fitur *Authentication* merupakan layanan yang disediakan oleh Firebase untuk mengelola dan menyediakan solusi autentikasi pengguna dalam aplikasi. Hal ini memungkinkan pengembang dengan mudah mengintegrasikan sistem autentikasi ke dalam aplikasi sehingga pengguna merasa aman.



GAMBAR 4.  
Firebase Authentication

Fitur *Authentication* pada *Firebase* memiliki beberapa keunggulan dibanding kompetitornya, seperti mudahnya mengintegrasikan, sistem keamanan yang sangat kokoh, skalabilitas yang baik, metode enkripsi yang kuat, dapat dengan mudah mengustomisasi tampilan antarmuka.



GAMBAR 5.  
Firebase Realtime Database

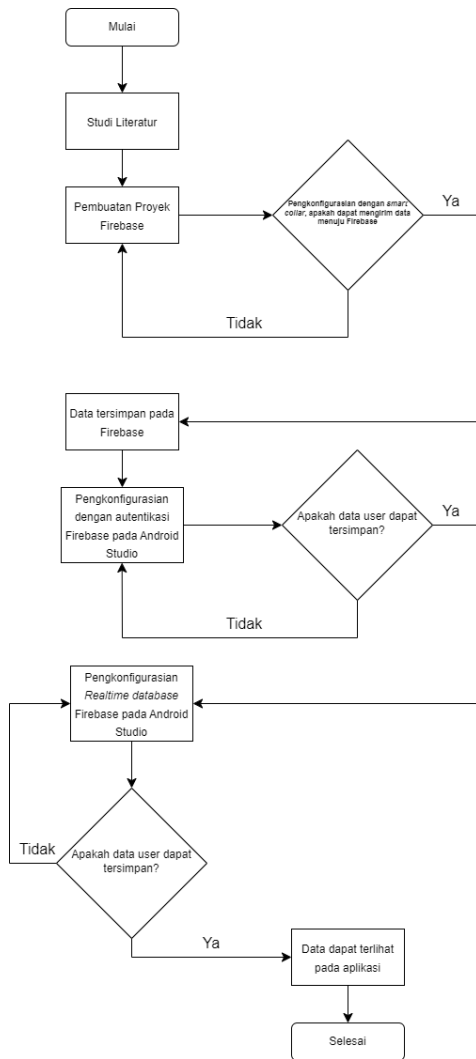
Fitur yang digunakan berikutnya merupakan fitur yang cukup penting yaitu *Firebase Realtime Database*. *Firebase Realtime Database* merupakan salah satu solusi yang diberikan oleh platform *Firebase* untuk menyimpan dan mengelola data dengan kemampuan sinkronisasi secara real-time. Platform ini menggunakan database NoSQL yang difokuskan pada kebutuhan aplikasi yang menginginkan data terkini dan terus-menerus diperbarui antar berbagai perangkat pengguna. Ini memberikan kecepatan dalam mengakses dan menyelaraskan data di berbagai perangkat, menjadikannya pilihan yang efektif untuk aplikasi yang berorientasi pada respons cepat dan sinkronisasi data yang efisien.

*Firebase Realtime Database* memiliki beberapa keunggulan seperti, data dapat disinkronkan secara otomatis, API yang mudah digunakan dan diimplementasikan, struktur data hierarkis, menangani kondisi jaringan koneksi yang buruk, mudah diintegrasikan dengan *Firebase Authentication*, mendeteksi perubahan data secara waktu sebenarnya, menangani banyak pengguna tanpa mengalami penurunan kinerja, keamanan tingkat tinggi.

Perlu diingat bahwa *Firebase Realtime Database* juga memiliki beberapa kekurangan dalam hal kueri kompleks dan indeks, dan jika memiliki kebutuhan aplikasi yang lebih kompleks maka lebih disarankan untuk memakai *Firebase Cloud Firestore*. Karena *Firebase Cloud Firestore* menawarkan kelebihan dibidang struktur data dan kueri. Dan karena pada penelitian ini tidak menggunakan aplikasi yang terlalu kompleks maka pada penelitian ini lebih memilih kearah *Firebase Realtime Database* dibanding menggunakan *Firebase Cloud Firestore*.

III. METODE

Pengerjaan sistem cloud yang dilakukan pada pemantauan kondisi kesehatan pada sapi menggunakan alat bantu berupa Firebase. Dan pada sistem *cloud* ini digunakan untuk membuat suatu sistem penyimpanan data. *Firestore Realtime Database* memungkinkan untuk mem-build aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data dipertahankan secara lokal, dan meskipun sedang *offline*, peristiwa realtime terus dipicu, sehingga pengguna akhir akan merasakan pengalaman yang responsif [4]. Berikut ini merupakan diagram alir dari metode pengerjaan sistem *cloud* menggunakan *Firestore*.



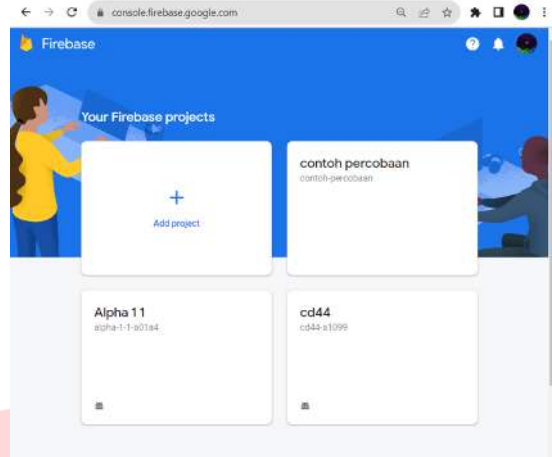
GAMBAR 6. Flowchart *Firestore*

Gambar.1 merupakan *Flowchart Firestore* yang berfungsi sebagai jalannya alur pengerjaan pada sistem penyimpanan yang dibuat. Selain berfungsi sebagai jalannya alur pengerjaan, *flowchart* dibuat agar perencanaan terskema dengan baik.

A. Pembuatan Proyek *Firestore*

Pembuatan Proyek *Firestore* dilakukan untuk menambahkan proyek apa yang ingin dibuat. Hal tersebut bisa dilakukan dengan cara pergi ke laman

“<https://console.firebase.google.com/>” kemudian klik “*add project*”



GAMBAR 7. Menambahkan proyek pada *Firestore*

B. Pengkonfigurasiian dengan *Smart Collar*

Pengkondisian dengan *Smart Collar* dilakukan agar *Firestore* base dapat terhubung kepada *Smart Collar* sehingga data yang dikirimkan dari *Smart Collar* akan dapat tersimpan pada *Firestore*. Penkonfigurasiian dilakukan dengan cara memprogram Arduino IDE agar *Firestore* terintegrasi.

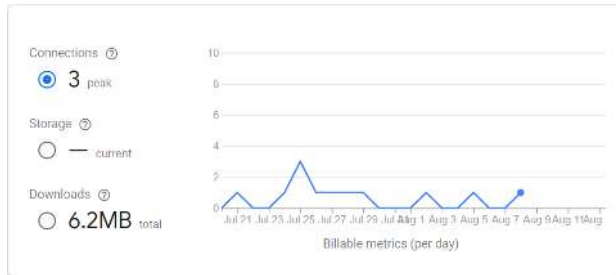
TABEL 1. Sketch source code *firebase*

```
#include <FirebaseESP8266.h>
#define FIREBASE_HOST "alpha-1-1-a01a4-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "3waBTdU8gstD0nHarwwQyg2VWe8IxzK8Kxmjqhvs"
FirebaseData firebaseData;
.....
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);
}
.....
Firebase.setInt(firebaseData, "/Suhu2", Smacowsuhu);
Firebase.setInt(firebaseData, "/Jarak2", distance);
Firebase.setInt(firebaseData, "/Jantung2", heartRate);
Firebase.setInt(firebaseData, "/accX2", accX);
Firebase.setInt(firebaseData, "/accY2", accY);
Firebase.setInt(firebaseData, "/accZ2", accZ);
.....
```

Pada Tabel 1, merupakan *source code* untuk mengintegrasikan *Firestore* dengan *Smart Collar*. Dan berikut ini adalah tahapan cara kerja *firebase* terhadap *Smart Collar* dan aplikasi. Pada Arduino IDE memiliki fungsi antara *smart collar* dengan *Firestore*.



aplikasi. Dalam penggunaan aplikasi yang telah ter-program terdapat 2 pengguna yang telah terdeteksi terhadap *Firestore*. yang didapatkan memiliki beberapa variasi yang diberikan oleh sensor suhu tersebut.



GAMBAR 14.  
Grafik *Billable Metrics* Pengguna

Grafik *Billable Metrics* Pengguna pada Gambar 14. merupakan berapa total biaya yang digunakan oleh para pengguna untuk mengirimkan data terhadap *Firestore* pada bagian *Realtime Database usage*.

Pada *Realtime Database* struktur hirarkis adalah struktur penyimpanan data yang digunakan pada *Firestore Realtime Database*. Struktur hirarkis ini berbentuk pasangan "key" dan "value". Key merupakan string unik yang digunakan sebagai alamat atau identifikasi dari suatu data berstruktur hirarkis. Database memiliki key yang unik untuk setiap node atau path. Key ini bisa berupa angka atau nilai lainnya namun biasanya berupa string. Contoh key dalam *Firestore Realtime Database* bisa menjadi seperti "users", "messages", atau "settings". Sedangkan value merupakan data aktual yang disimpan dalam suatu node atau path dalam database. Value ini bisa berupa tipe data apapun, termasuk string, angka, boolean, objek JSON, dan sebagainya. Setiap key memiliki value yang terkait, yang berisi informasi yang ingin anda simpan dalam database.

## V. KESIMPULAN

Berdasarkan hasil yang telah diperoleh dari pengerjaan sistem penyimpanan data menggunakan *Firestore*, *Firestore* berhasil terintegrasi baik dengan aplikasi maupun dengan *Smart Collar* itu sendiri. Sistem keamanan penyimpanan hasil data pun dapat terbilang berhasil karena fitur *Authentication* pada *Firestore* memberikan jaminan keamanan bagi data penggunaannya. Aktivitas login pengguna juga berhasil tersinkronisasi dengan baik pada *Firestore Authentication* karena dapat terdeteksi dengan baik pada fitur tersebut. Serta pada *Billable Metrics* dapat dilihat berapa besar data hasil dalam Megabyte total yang telah dikirim dan diterima melalui *Firestore* tersebut dapat

dikatakan bahwa sensor tersebut mengalami tingkat keakuratan cukup baik.

## VI. REFERENSI

- [1] F. R. Aldeyano, A. Sudrajat, and A. M. Susiati, "Level of Understanding of Dairy Cattle Farmers Against Foot and Mouth Disease Cases in West Bandung Lembang," vol. 11, pp. 115–124, 2023.
- [2] A. Firman, I. Trisman, and R. H. Puradireja, "Dampak Ekonomi Akibat Outbreak Penyakit Mulut Dan Kuku Pada Ternak Sapi Dan Kerbau Di Indonesia," *Mimb. Agribisnis J. Pemikir. Masy. Ilm. Berwawasan Agribisnis*, vol. 8, no. 2, p. 1123, 2022, doi: 10.25157/ma.v8i2.7749.
- [3] S. Nootyaskool and P. Ounsrimung, "Smart collar design to predict cow behavior," *JCSSE 2020 - 17th Int. Jt. Conf. Comput. Sci. Softw. Eng.*, pp. 92–97, 2020, doi: 10.1109/JCSSE49651.2020.9268342.
- [4] Y. P. Pratama *et al.*, "Designing of a Smart Collar for Dairy Cow Behavior Monitoring with Application Monitoring in Microservices and Internet of Things-Based Systems," *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 527–533, 2019, doi: 10.1109/ELECSYM.2019.8901676.
- [5] R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail, and S. S. S. Ranjit, "Smart home system using android application," *2013 Int. Conf. Inf. Commun. Technol. ICoICT 2013*, pp. 277–280, 2013, doi: 10.1109/ICoICT.2013.6574587.
- [6] U. Pakijan, "Diagnosis SeOagai Satafi Satu Syarat Vntu^Meraifi ^elarSaijana 'Efipnomi di TafiuCtas Efipnomi Jurusan vianajemen," 2004.
- [7] R. Thamizharasi, "Android Mobile Application Build on Android studio," *Int. J. Mod. Comput. Sci.*, vol. 4, no. 1, pp. 1–4, 20