

Pengintegrasian *Mobile Application* dan Server dalam Pengiriman Data *Real-Time* dari *Database* ke Aplikasi *Mobile*: Studi Implementasi dengan Flutter, Dart, dan Python

1st Alldo Faiz Ramadhani
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

alldofaiz@student.telkomuniversity.ac.id

2nd Rita Purnamasari
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ritapurnamasari@telkomuniversity.ac.id

3rd Rustam
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

rustamtelu@telkomuniversity.ac.id

Abstrak — Penelitian ini membahas implementasi dan cara kerja subsistem *Mobile Application* dan Server dalam pengiriman data dari database ke *Mobile Application*. Implementasi *Mobile Application* menggunakan Visual Studio Code, Flutter, dan bahasa pemrograman Dart, sedangkan Server menggunakan Visual Studio Code dan bahasa pemrograman Python. Data dari database diambil oleh Server dan dikirim ke *Mobile Application* melalui Firebase. *Mobile Application* menampilkan data dan menerima notifikasi, sementara Server berfungsi sebagai penghubung antara database dan *Mobile Application*. Metode yang digunakan meliputi pengembangan antarmuka pengguna, integrasi Firebase, permintaan HTTP, dan penjadwalan tugas latar belakang. Hasil dan pembahasan menunjukkan bahwa *Mobile Application* berhasil menampilkan data dan menerima notifikasi, sedangkan Server berhasil mengambil data dari database dan mengirimkannya ke *Mobile Application*. Kesimpulan dari penelitian ini adalah subsistem *Mobile Application* dan Server memiliki peran penting dalam pengiriman data dari database ke *Mobile Application*.

Kata kunci – *Mobile Application*, Server, *Real-Time*, Python, Flutter

I. PENDAHULUAN

Subsistem *Mobile Application* dan Server memainkan peran penting dalam menghubungkan database dengan *Mobile Application*. *Mobile Application* berfungsi sebagai antarmuka pengguna yang menampilkan data dan menerima notifikasi, sementara Server berperan sebagai jembatan antara database dan *Mobile Application*. Implementasi *Mobile Application* menggunakan alat pengembangan seperti Visual Studio Code, Flutter, dan bahasa pemrograman Dart, sedangkan Server menggunakan Visual Studio Code dan bahasa pemrograman Python. Penelitian ini bertujuan untuk menjelaskan secara rinci implementasi dan mekanisme kerja dari kedua subsistem ini.

Mobile Application menjadi sarana utama untuk menampilkan data yang berasal dari database kepada pengguna. Dalam implementasinya, penggunaan Visual Studio Code sebagai *Integrated Development Environment* (IDE) memungkinkan pengembangan aplikasi yang efisien. Flutter, sebagai framework yang digunakan, memberikan kemudahan dalam membuat antarmuka pengguna yang

responsif dan menarik. Penggunaan bahasa pemrograman Dart memberikan keleluasaan dalam menulis kode aplikasi yang kompatibel dengan platform Flutter.

Pengembangan antarmuka pengguna (UI), integrasi Firebase, permintaan HTTP, dan penjadwalan tugas latar belakang menjadi faktor kunci dalam implementasi *Mobile Application*. Desain UI yang baik akan meningkatkan pengalaman pengguna dengan memastikan tampilan yang menarik dan navigasi yang intuitif. Integrasi Firebase memungkinkan pertukaran data yang aman antara *Mobile Application* dan Server. Permintaan HTTP digunakan untuk berkomunikasi dengan *backend* atau API dari *Mobile Application*, memungkinkan pengambilan dan pengiriman data secara efisien. Sementara itu, penjadwalan tugas latar belakang pada Server memungkinkan pengambilan data secara berkala dari database dan pengiriman data ke *Mobile Application* sesuai dengan kebutuhan.

II. KAJIAN TEORI

A. *Mobile Application*:

Subsistem *Mobile Application* memiliki peran utama dalam menampilkan data yang berasal dari database kepada pengguna. Implementasi *Mobile Application* melibatkan penggunaan Visual Studio Code sebagai lingkungan pengembangan terintegrasi (*Integrated Development Environment*) yang membantu dalam pengembangan aplikasi. Flutter, sebuah framework yang digunakan, memungkinkan pembuatan antarmuka pengguna yang responsif dan menarik. Selain itu, bahasa pemrograman Dart digunakan untuk menulis kode aplikasi yang berjalan di atas platform Flutter.

Pengembangan antarmuka pengguna (UI), integrasi Firebase, permintaan HTTP, dan penjadwalan tugas latar belakang menjadi aspek penting dalam implementasi *Mobile Application*. Pengembangan UI yang baik memastikan pengguna dapat berinteraksi dengan aplikasi secara intuitif. Integrasi Firebase memungkinkan pengiriman dan penerimaan data antara *Mobile Application* dan Server. Penggunaan permintaan HTTP memfasilitasi komunikasi dengan *backend* atau API dari *Mobile Application*. Penjadwalan tugas latar belakang digunakan pada Server

untuk mengambil dan mengirim data dari *database* ke *Mobile Application*.

B. Server:

Subsistem Server berperan sebagai penghubung antara *database* dan *Mobile Application*. Implementasi Server melibatkan penggunaan Visual Studio Code sebagai IDE dan bahasa pemrograman Python sebagai bahasa utama. Server bertugas mengambil data dari *database* dan mengirimkannya ke *Mobile Application* melalui Firebase. Dengan menggunakan Firebase sebagai perantara, proses pengiriman data dapat dilakukan secara aman dan efisien.

III. METODE

Metode yang digunakan dalam penelitian ini meliputi beberapa tahapan yang terdiri dari pengembangan antarmuka pengguna pada *Mobile Application* menggunakan Visual Studio Code, Flutter, dan bahasa pemrograman Dart. Integrasi Firebase digunakan untuk mengirim dan menerima data antara *Mobile Application* dan Server, memastikan pertukaran data yang andal dan aman. Permintaan HTTP digunakan untuk berkomunikasi dengan *backend* atau API dari *Mobile Application*, memfasilitasi pengambilan dan pengiriman data dari dan ke sumber eksternal.

Selain itu, penjadwalan tugas latar belakang digunakan pada Server untuk secara otomatis mengambil dan mengirim data dari *database* ke *Mobile Application*. Dengan adanya penjadwalan tugas latar belakang, proses ini dapat dilakukan secara teratur dan dapat disesuaikan sesuai kebutuhan.

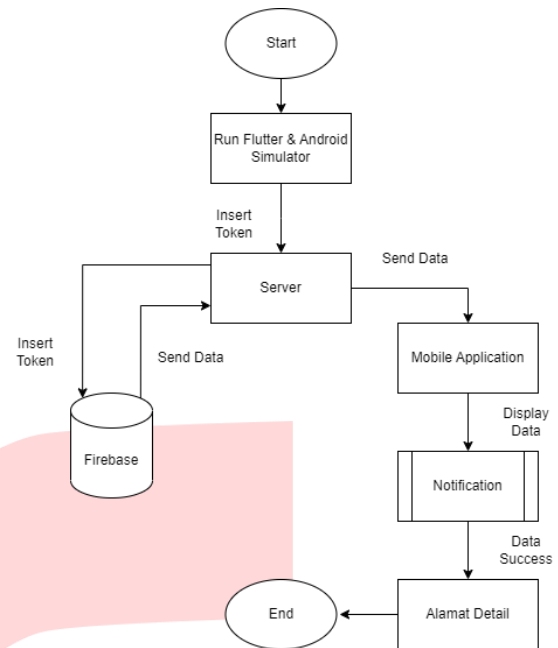
Dengan menggunakan metode ini, diharapkan bahwa implementasi kedua subsistem (*Mobile Application* dan Server) dapat berhasil dilakukan dengan baik. Selain itu, penjelasan yang mendalam mengenai cara kerja masing-masing subsistem diharapkan dapat disampaikan dengan jelas dalam penelitian ini.

IV. HASIL DAN PEMBAHASAN

Implementasi *Mobile Application* berhasil menampilkan data dari *database* dan menerima notifikasi. Pengguna dapat melihat informasi bencana dan menggunakan tombol Google Maps untuk navigasi lokasi. Implementasi Server berhasil mengambil data dari *database* dan mengirimkannya ke *Mobile Application* melalui Firebase. Data dikirim ulang setiap 5 detik untuk memastikan notifikasi tetap diperbarui.

Gambar 1 adalah gambar flowchart bagaimana cara *Mobile Application* dan server bekerja, dari pengiriman token ke firebase sampai pengiriman data ke *Mobile Application*. Pada *Mobile Application* diakhiri dengan Halaman Alamat Detail sebagai akhir dari pengiriman data.

Langkah pertama jalankan Flutter dan Android Simulator terlebih dahulu, setelah itu baru jalankan server untuk mengambil token dari Android Simulatornya. Selanjutnya server akan mengirim token kedalam *database* untuk mendaftarkan *device* kedalam *database*. Jika *device* sudah terdaftar kedalam *database*, baru server bisa mengambil data dari *database* dan mengirimkannya ke *Mobile Application*. Jika data berhasil diterima *Mobile Application*, *Mobile Application* akan menerima notifikasi berupa “Kebakaran”. Tahap terakhir jika kita klik notifikasi tersebut akan menyajikan “Alamat Detail” dari notifikasi tersebut.



GAMBAR 1
(Gambar Flowchart Bagaimana Cara *Mobile Application* dan Server Bekerja)

V. KESIMPULAN

Subsistem *Mobile Application* dan Server memiliki peran penting dalam pengiriman data dari *database* ke *Mobile Application*. Implementasi *Mobile Application* menggunakan Visual Studio Code, Flutter, dan bahasa pemrograman Dart, sementara Server menggunakan Visual Studio Code dan bahasa pemrograman Python. *Mobile Application* berhasil menampilkan data dan menerima notifikasi, sedangkan Server berhasil mengambil data dari *database* dan mengirimkannya ke *Mobile Application*. Proses ini memungkinkan pengguna untuk melihat informasi bencana dan menggunakan navigasi lokasi melalui *Mobile Application*. Kesimpulan dari penelitian ini adalah implementasi *Mobile Application* dan Server secara efektif mendukung pengiriman data dari *database* ke *Mobile Application*.

REFERENSI

Electronic References

●Journal

- [1] Wang, C., Li, Y., Liu, Q., & Zhang, Z. (2019). "Development of Mobile Application for Data Visualization and Monitoring System in Smart Grid." [Online]. IEEE Access, 7, 18315-18326.
- [2] Chen, Y., Zhou, L., Liu, H., & Li, B. (2019). "Implementation of a Mobile Application for Real-Time Monitoring and Control of IoT Devices." IEEE Access, 7, 53301-53310.
- [3] Almehmadi, F. M., Alsaleh, N. A., Alzahrani, H. S., & Kamal, T. S. (2020). "Development and Evaluation of a Mobile Application for Emergency Management in Saudi Arabia." Journal of Sensors, 2020.
- [4] Chen, Y., Li, B., & Zhou, L. (2018). "A Real-Time Data Monitoring System for Internet of Things Based on Mobile Application." 2018 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 459-464.

