

Analisis Performa Aplikasi Pengenalan Wajah Fast : Latensi Dan Penggunaan Sumber Daya

1st Mochamad Ilman Yassir Rizqi
Telkom University
Fakultas Teknik Elektro
Bandung, Indonesia
ilmanrizqi@student.telkomuniversity.a
c.id

2nd Uke Kurniawan Usman
Telkom University
Fakultas Teknik Elektro
Bandung, Indonesia
ukeusman@telkomuniversity.ac.id

3rd Heru Syah Putra
Telkom University
Fakultas Teknik Elektro
Bandung, Indonesia
herusyahputra@telkomuniversity.ac.id

Abstrak — Sistem presensi konvensional masih mengalami masalah efisiensi dalam pengelolaan kehadiran. Studi ini merancang aplikasi *Face Attendance for Students (FAST)* menggunakan teknologi pengenalan wajah berbasis *Machine Learning* yang terhubung dengan *Firestore Realtime Database*. FAST melakukan pencatatan kehadiran mahasiswa secara otomatis dengan memanfaatkan Google ML Kit dan FaceNet, sementara Firestore berfungsi sebagai tempat penyimpanan data utama. Hasil analisis mengindikasikan bahwa sistem menunjukkan latensi pengenalan wajah antara 36–182 ms (median 162 ms) dan pencatatan presensi antara 2–39 ms (median 4 ms), memastikan respon yang cepat dan efisien. Aplikasi ini juga menunjukkan efisiensi dalam penggunaan sumber daya, dengan pemakaian memori hanya 28 MB dari total 384 MB yang ada. Dengan kinerja tersebut, FAST menjadi jawaban inovatif dan efisien untuk presensi perkuliahan.

Kata kunci — presensi, pengenalan wajah, Firestore, *Machine Learning*, efisiensi sistem.

I. PENDAHULUAN

Dalam dunia pendidikan, sistem presensi menjadi salah satu aspek penting dalam pengelolaan akademik untuk mengikuti ujian akhir semester khususnya di Universitas Telkom adalah memiliki kehadiran minimal 75%, yang dihitung berdasarkan jumlah kehadiran setiap semester [1]. Metode manual seperti tanda tangan di daftar hadir atau penggunaan kartu identitas masih sering digunakan, tetapi rentan terhadap kecurangan dan kurang efisien dalam pencatatan. Untuk mengatasi masalah ini, pengenalan wajah berbasis *Machine Learning* menjadi solusi yang lebih modern dan efektif.

Aplikasi *FAST* dirancang untuk mencatat kehadiran mahasiswa secara otomatis dan real-time menggunakan teknologi pengenalan wajah. Penelitian ini berfokus pada penerapan backend firebase yang bertanggung jawab dalam pengolahan sumber daya, latensi dan data presensi secara real-time pada *FAST*. Firestore Realtime Database digunakan sebagai penyimpanan data utama, sementara FaceNet dan Google ML Kit diterapkan untuk proses pengenalan wajah.

II. KAJIAN TEORI

A. Face Recognition

Face Recognition merupakan perkembangan dari teknologi face detection, dimana teknologi ini dapat menghasilkan atau men-generate wajah dari hasil tangkapan kamera dengan data wajah yang diketahui komputer, sehingga komputer dapat mengenali keberadaan seseorang [2].

B. Firestore Realtime Database

Firestore Realtime Database merupakan fitur layanan yang berfungsi untuk menyimpan data pengguna seperti username, nama lengkap, NIM (untuk mahasiswa), kode dosen (untuk dosen), serta data lainnya yang berkaitan dengan presensi dan jadwal [3].

C. Google ML Kit

Google ML Kit menyediakan Face Detection API yang digunakan untuk mendeteksi wajah pada gambar yang diambil melalui kamera perangkat. API ini dapat mengidentifikasi posisi wajah dan titik-titik penting (seperti mata, hidung, dan mulut) dalam gambar [4].

D. FaceNet

FaceNet adalah sistem yang mengubah gambar wajah menjadi ruang Euclidean, di mana jarak antara titik menunjukkan tingkat kesamaan wajah. Teknologi ini menggunakan arsitektur Deep Convolutional Neural Network (DCNN) untuk mengambil fitur wajah yang sangat representative [5].

E. Back-End

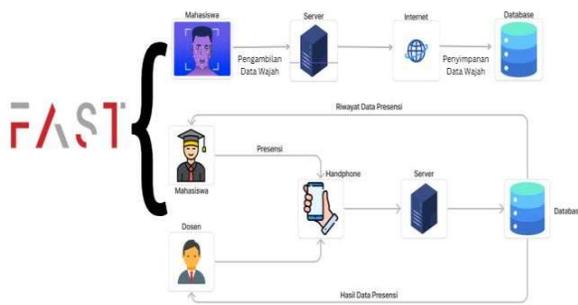
Back-end merupakan pengembangan situs atau aplikasi yang berperan pada bagian belakang pengguna, atau bisa dimaksudkan dengan sistem agar situs atau aplikasi dapat berjalan dengan lancar.

III. METODE

A. Perancangan Sistem

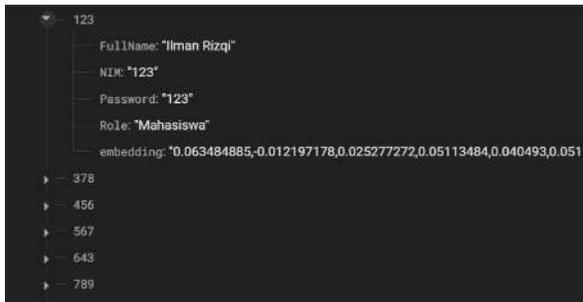
Sistem back-end *FAST* pada Gambar 1 dirancang dengan mempertimbangkan efisiensi dalam pengolahan data dan presensi. Arsitektur yang digunakan melibatkan Firestore

sebagai basis data utama, serta integrasi dengan Google ML Kit & FaceNet untuk pengenalan wajah. Data yang dikirimkan oleh aplikasi dikirimkan melalui API untuk kemudian diproses di Firebase.



GAMBAR 1
Arsitektur Backend Fast

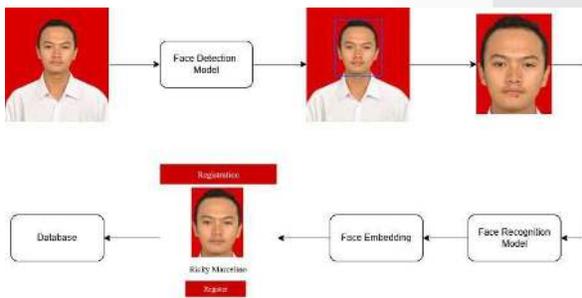
Sistem absensi yang menggunakan pengenalan wajah ini memanfaatkan firebase realtime database sebagai penyimpanan utamanya. Pada Gambar 2 , terlihat data mahasiswa yang disimpan dalam firebase , meliputi nama , nim , peran (role) , kata kunci dan vektor embedding yang dihasilkan dari pemrosesan wajah menggunakan FaceNet. Vektor embbending ini berperan sebagai representasi angka dari ciri khas masing-masing wajah pengguna.



GAMBAR 2
Penyimpanan Data di Firebase

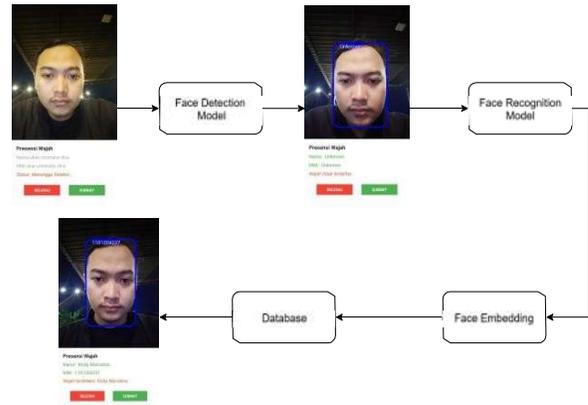
B. Implementasi Sistem

Perancangan implementasi sistem pada aplikasi pengenalan wajah untuk presensi perkuliahan di kelas pada prodi S1 Teknik Telekomunikasi bertujuan untuk mempermudah mahasiswa dalam melakukan presensi perkuliahan di kelas.



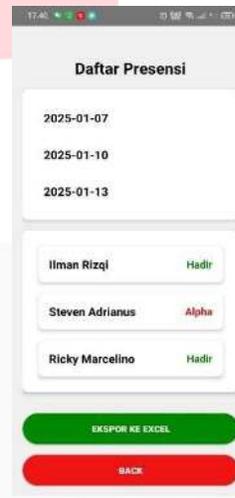
GAMBAR 3
Registrasi Wajah

Pada Gambar 3 Mahasiswa mendaftarkan wajah mereka ke dalam sistem yang kemudian disimpan dalam bentuk embedding pada firebase.



GAMBAR 4
Proses Presensi

Gambar 4 menunjukkan saat melakukan presensi, sistem akan membandingkan wajah mahasiswa dengan data wajah yang sudah tersimpan.



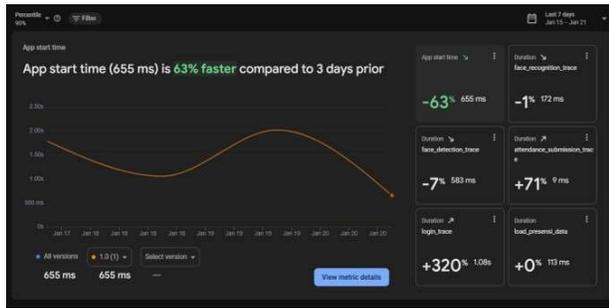
GAMBAR 5
Daftar Kehadiran

Setelah identifikasi berhasil, data kehadiran secara otomatis tercatat di firebase dan menampilkan output di aplikasi pada gambar 5.

IV. HASIL DAN PEMBAHASAN

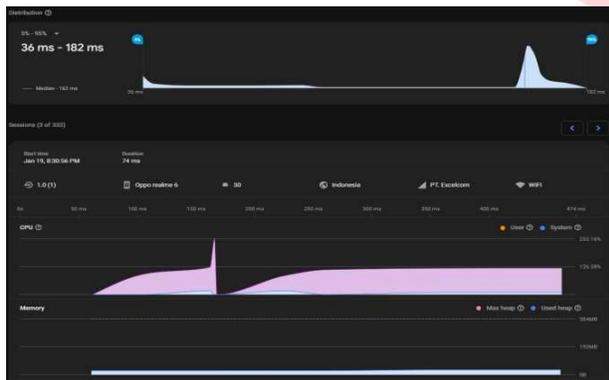
Hasil penerapan firebase dalam aplikasi Face Attendance For Student memainkan peran penting dalam pengelolaan data presensi mahasiswa secara real-time. Firebase Realtime Database dipilih karena kemampuannya dalam menangani data dalam jumlah besar dengan latensi rendah, sehingga memastikan bahwa setiap data kehadiran dapat dicatat dan performa tiap tampilan dapat dimonitoring melalui performance firebase.

Dapat dilihat pada gambar 6 menunjukkan berbagai metrik performa aplikasi, termasuk waktu start aplikasi (app start time) yang tercatat berada di 655ms yang menunjukkan peningkatan performa aplikasi dan latensi berbagai operasi backend.



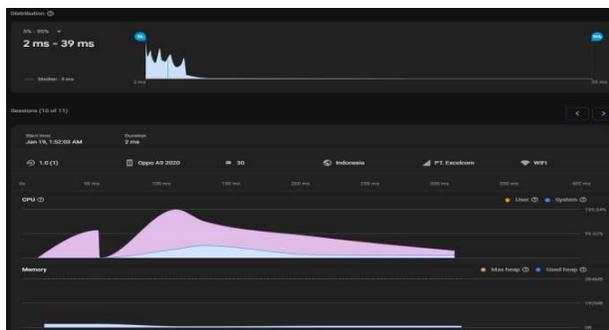
GAMBAR 6
Dashboard Firebase Performance Monitoring

Metrik pada Gambar 7 menunjukkan waktu yang dibutuhkan sistem untuk mengenali wajah mahasiswa dan mencocokkannya dengan data yang ada di Firebase. terlihat sesi `trace_recognition_face` pada gambar 6 memiliki latensi antara 36 ms dan 182 ms, dengan median 162 ms, menunjukkan waktu respons yang cepat untuk proses pengenalan wajah. Durasi sesi yang tercatat adalah 74 ms, dan penggunaan CPU menunjukkan tingkat yang tinggi pada ruang pengguna (253,16%) dibandingkan dengan ruang sistem (17,69%).



GAMBAR 7
Tampilan metrik pada sesi pengenalan wajah

Gambar 8 menunjukkan `attendance_submission_trace` dengan latensi rendah, yang berkisar antara 2 ms hingga 39 ms, dengan median 4 ms, yang menunjukkan efisiensi tinggi dalam pengiriman data presensi. Durasi sesi secara keseluruhan hanya 2 ms, yang menunjukkan proses yang sangat cepat. Jumlah CPU yang digunakan lebih banyak di area user (199.04%) daripada di area system (99.52%), menunjukkan bahwa aplikasi melakukan sebagian besar pemrosesan. Meskipun demikian, penggunaan memori tetap efisien, dengan hanya 28 MB yang digunakan dari 384 MB, yang menunjukkan pengelolaan sumber daya yang optimal.



GAMBAR 8
Tampilan metrik pada sesi pencatatan presensi

V. KESIMPULAN

Penelitian ini berhasil merancang sistem presensi berbasis pengenalan wajah dengan dukungan teknologi *Machine Learning* yang terhubung ke *Firestore Database*. Pengembangan back-end yang diterapkan memastikan pemrosesan data kehadiran berlangsung secara real-time dengan tingkat latensi yang rendah serta efisiensi yang tinggi. Berdasarkan analisis performa, sistem menunjukkan respons cepat dalam mengenali wajah mahasiswa. Latensi dalam proses ini berkisar antara 36 ms hingga 182 ms, dengan median 162 ms, yang mengindikasikan efektivitas dan kecepatan yang tinggi. Selain itu, pencatatan presensi berjalan dengan latensi rendah, yakni antara 2 ms hingga 39 ms, dengan median 4 ms, sehingga memungkinkan pengelolaan data kehadiran secara optimal.

Efisiensi penggunaan sumber daya sistem juga terbukti memuaskan. Pemrosesan lebih dominan terjadi di ruang pengguna, menunjukkan bahwa mayoritas tugas diproses langsung oleh aplikasi. Selain itu, penggunaan memori tetap efisien, hanya menggunakan 28 MB dari total kapasitas 384 MB yang tersedia.

REFERENSI

- [1] Universitas Telkom, "Pedoman Akademik Universitas Telkom," Bandung, 2020
- [2] N. Rao K Mahalakshmi, "A Novel Face Detection and Recognition System Using Machine Learning Approaches," *International Journal of Science and Research (IJSR)*, vol. 12, no. 6, pp. 2730–2738, Jun. 2023, doi: 10.21275/SR23626104114.
- [3] Dicoding, "Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya." Accessed: Jun. 17, 2024. [Online]. Available: Dicoding, "Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya."
- [4] Google Developers, "Object Detection and Tracking with ML Kit on Android." Accessed: Oct. 09, 2024. [Online]. Available: <https://codelabs.developers.google.com/mlkit-android-odt?hl=id#0>
- [5] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 815–823. doi: 10.1109/CVPR.2015.72