

Perancangan Dan Implementasi Aplikasi Mobile *Stress Detection App* Menggunakan *Google Cloud*

Rio Aji Pangestu

Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur

21081010029@student.upnjatim.ac.id

Abstrak—*Stress Detection App* adalah aplikasi yang menggunakan pembelajaran mesin untuk mengenali wajah dan menganalisis tingkat stres melalui survei singkat. Aplikasi ini memungkinkan pengguna untuk mencatat suasana hati harian, mengisi survei, dan menerima tip serta trik untuk mengelola tingkat stres. Dengan fokus pada tiga tingkat stres: rendah, menengah, dan tinggi, aplikasi ini bertujuan untuk membantu pengguna yang mengalami stres dan mencegah dampak negatif yang mungkin timbul. Metodologi penelitian yang digunakan melibatkan penggunaan *Google App Engine* untuk *backend*, *Google Cloud Datastore* untuk manajemen database, serta *TensorFlow* untuk pelatihan model pembelajaran mesin. Pengembangan *frontend* aplikasi dilakukan menggunakan Figma untuk desain antarmuka pengguna yang responsif dan menarik. Hasil dari penelitian ini mencakup pengembangan *frontend* dan *backend* aplikasi, termasuk pembuatan desain awal menggunakan Figma, training data untuk analisis wajah, pembuatan server dengan Node.js dan Express.js, serta pengujian menggunakan Expresso. Dengan adanya *Stress Detection App*, diharapkan pengguna dapat lebih memahami tingkat stres mereka dan mengelolanya dengan lebih efektif, sehingga dapat menghindari kemungkinan terburuk yang dapat terjadi akibat stres berlebih.

Kata Kunci— *Stress Detection App*, *Google Cloud*, Tingkat Stres, *frontend*, *backend*

I. PENDAHULUAN

Stress Detection App merupakan aplikasi yang memanfaatkan pembelajaran mesin untuk memberikan pengenalan wajah dan menganalisis tingkat stres dari survei singkat. Aplikasi ini berfokus pada 3 tingkat stres : tingkat rendah, menengah, dan tinggi [1]. Tingkat stres dianalisis berdasarkan suasana hati kita dan ringkasan survei [2]. Dengan *Stress Detection App*, pengguna dapat mencatat suasana hati mereka setiap hari, mengisi survei, dan melihat tip dan trik tentang tingkat stres mereka. Berharap aplikasi ini akan membantu situasi mereka yang tidak mengerti dan menyelamatkan segala kemungkinan terburuk yang bisa terjadi.

II. LANDASAN TEORI

A. Google Cloud Platform (GCP)

Google Cloud Platform (GCP) adalah suite layanan cloud yang ditawarkan oleh Google, yang mencakup berbagai layanan seperti komputasi awan, penyimpanan data, basis data, kecerdasan buatan, dan banyak lagi. GCP memungkinkan organisasi untuk membangun, mengelola, dan mendeploy

aplikasi mereka di infrastruktur global Google yang canggih [3].

B. Node.js

Node.js adalah lingkungan runtime JavaScript yang memungkinkan pengembang untuk menjalankan kode JavaScript di sisi server. Ini dibangun di atas mesin JavaScript V8 dari Google Chrome dan menggunakan pendekatan non-blocking I/O dan event-driven untuk menangani banyak koneksi secara bersamaan dengan kinerja tinggi [4]. Node.js juga dilengkapi dengan NPM (*Node Package Manager*) yang memungkinkan pengguna untuk mengelola dan menggunakan paket-paket JavaScript dari repositori publik. Dengan fungsionalitas ini, Node.js menjadi pilihan populer untuk pengembangan aplikasi web real-time dan berkinerja tinggi.

C. Express.js

Express.js adalah kerangka kerja web yang minimalis dan fleksibel untuk Node.js. Ini menyediakan serangkaian fitur yang memungkinkan pengembang untuk dengan mudah membangun aplikasi web, seperti penanganan rute, pengaturan middleware, dan manajemen permintaan dan respons HTTP [5]. Express.js memudahkan pengembangan aplikasi web dengan menyediakan antarmuka yang sederhana dan intuitif, tetapi juga memungkinkan fleksibilitas yang luas untuk menyesuaikan aplikasi sesuai kebutuhan. Dengan ekosistem yang kuat dan dukungan yang luas dari komunitas, Express.js telah menjadi salah satu kerangka kerja web Node.js yang paling populer dan banyak digunakan.

D. Python

Python adalah bahasa pemrograman tingkat tinggi yang mudah dipahami dan sangat serbaguna. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python menekankan sintaksis yang jelas dan mudah dibaca, yang membuatnya menjadi pilihan yang populer untuk pemula sekaligus para profesional [6]. Python mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, pemrograman fungsional, dan pemrograman prosedural.

Keunggulan Python termasuk kesederhanaan, kemudahan dalam prototyping, serta ekosistem yang kaya dengan berbagai pustaka dan kerangka kerja untuk berbagai keperluan, mulai dari pengembangan web hingga kecerdasan buatan [6]. Python digunakan secara luas dalam berbagai bidang, termasuk

pengembangan perangkat lunak, analisis data, keamanan jaringan, dan penelitian ilmiah.

E. Kotlin

Kotlin adalah bahasa pemrograman modern yang berjalan di atas platform *Java Virtual Machine (JVM)*, serta dapat dikompilasi ke JavaScript atau kode sumber Native. Dikembangkan oleh JetBrains pada tahun 2011, Kotlin dirancang untuk menjadi bahasa yang ekspresif, ringkas, dan interoperabel dengan kode Java [7].

Kotlin menyediakan fitur-fitur yang canggih seperti penanganan null secara aman, ekstensi fungsi, dan pemrograman berorientasi objek yang kuat, membuatnya menjadi pilihan yang populer untuk pengembangan aplikasi Android. Kotlin juga memiliki dukungan yang kuat dari industri, dan digunakan secara luas dalam pengembangan perangkat lunak di berbagai platform, termasuk web, server, dan aplikasi seluler. Keunggulan Kotlin termasuk kesederhanaan sintaksisnya, keamanan tipe statis, serta dukungan yang kuat dari IntelliJ IDEA dan ekosistem perkakas pengembangan JetBrains lainnya [7].

III. METODOLOGI PENELITIAN

A. App Engine

Google App Engine adalah salah satu layanan unggulan dari *Google Cloud Platform (GCP)*. Ini adalah platform yang sepenuhnya dikelola yang memungkinkan pengembang untuk membangun dan menjalankan aplikasi web dan layanan tanpa harus khawatir tentang infrastruktur yang mendasarinya [8]. Berikut adalah beberapa poin penting tentang *Google App Engine*:

- 1) Skalabilitas Otomatis : *App Engine* secara otomatis menangani peningkatan lalu lintas dengan menyesuaikan kapasitas sumber daya sesuai kebutuhan aplikasi. Ini memungkinkan aplikasi untuk menangani jumlah pengguna yang bervariasi tanpa mengorbankan kinerja.
- 2) Berbagai Lingkungan Bahasa: *App Engine* mendukung berbagai bahasa pemrograman populer termasuk Python, Java, Go, Node.js, PHP, dan .NET [8]. Ini memberikan fleksibilitas kepada pengembang untuk memilih bahasa yang paling sesuai dengan kebutuhan mereka.
- 3) Lingkungan Pembangunan yang Sederhana: Pengembang dapat dengan mudah mendeploy aplikasi mereka menggunakan antarmuka baris perintah atau konsol web yang sederhana. Ini membuat proses pengembangan dan penyebaran menjadi lebih cepat dan lebih efisien.
- 4) Integrasi dengan Layanan *Google Cloud*: *App Engine* terintegrasi dengan baik dengan layanan *Google Cloud* lainnya seperti *Google Cloud Storage*, *Google Cloud SQL*, *BigQuery*, dan *Cloud Firestore*. Ini memungkinkan pengembang untuk dengan mudah

memanfaatkan berbagai fitur yang disediakan oleh platform *Google Cloud* [9].

B. Datastore

Google Cloud Datastore adalah layanan manajemen database NoSQL yang sepenuhnya dikelola yang disediakan oleh *Google Cloud Platform (GCP)*. Ini adalah layanan penyimpanan awan yang fleksibel, skalabel, dan sangat tersedia yang dirancang untuk menyimpan dan mengelola data aplikasi yang bersifat semi-struktural. Berikut adalah beberapa poin penting tentang *Google Cloud Datastore*:

- 1) Model Data NoSQL: *Datastore* menggunakan model data NoSQL yang tidak memerlukan skema yang ketat. Ini berarti bahwa pengembang dapat menyimpan data semi-struktur atau tak terstruktur tanpa perlu mendefinisikan skema terlebih dahulu. Hal ini memungkinkan untuk fleksibilitas dalam mengubah struktur data tanpa perlu memperbarui skema database.
- 2) Skalabilitas Otomatis: *Datastore* secara otomatis mengelola peningkatan lalu lintas dan volume data dengan menyesuaikan kapasitas penyimpanan dan kinerja secara otomatis. Ini memastikan bahwa aplikasi tetap responsif dan tersedia, bahkan ketika ada peningkatan tiba-tiba dalam lalu lintas pengguna.
- 3) Konsistensi Data: *Datastore* menawarkan model konsistensi yang kuat, termasuk *Strong Consistency* dan *Eventual Consistency*, yang memungkinkan pengembang untuk memilih tingkat konsistensi yang sesuai dengan kebutuhan aplikasi mereka
- 4) *Datastore* terintegrasi dengan baik dengan ekosistem *Google Cloud* lainnya, seperti *Google Cloud Functions*, *Google App Engine*, dan *Google Cloud Storage*. Ini memungkinkan pengembang untuk dengan mudah mengakses dan memanfaatkan data dari berbagai layanan cloud Google.

C. Tensorflow

TensorFlow adalah platform sumber terbuka yang dikembangkan oleh Google untuk pembelajaran mesin dan kecerdasan buatan [10]. Ini memungkinkan pengembang untuk membuat dan melatih model pembelajaran mesin, mulai dari jaringan saraf tiruan hingga model statistik yang kompleks. Salah satu fitur utama TensorFlow adalah representasi data yang disebut "tensor", yang memungkinkan operasi matematika yang efisien dan paralel di seluruh kumpulan data besar.

TensorFlow menyediakan berbagai alat dan pustaka yang kuat untuk pemodelan, pelatihan, dan evaluasi model pembelajaran mesin, serta integrasi dengan infrastruktur komputasi yang berbeda, seperti CPU, GPU, dan TPU (*Tensor Processing Unit*) [10]. TensorFlow telah digunakan secara luas dalam berbagai

aplikasi, termasuk pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, dan banyak lagi.

D. Retrofit

Retrofit adalah sebuah library yang dikembangkan oleh Square untuk Android yang menyediakan cara sederhana dan kuat untuk mengakses *API RESTful* [11]. Ini memungkinkan pengembang untuk dengan mudah mengintegrasikan layanan web ke dalam aplikasi Android mereka dengan cara yang bersih, efisien, dan mudah dipahami. Berikut adalah beberapa poin penting tentang Retrofit:

- 1) Kemudahan Penggunaan: Retrofit menyediakan antarmuka yang intuitif dan mudah dipahami untuk mendefinisikan permintaan HTTP ke *API RESTful* [9]. Ini memungkinkan pengembang untuk dengan cepat membuat permintaan GET, POST, PUT, DELETE, dan lainnya dengan hanya beberapa baris kode.
- 2) Serialisasi dan Deserialisasi Otomatis: Retrofit secara otomatis menangani proses serialisasi dan deserialisasi data JSON ke dan dari objek Java. Ini memudahkan pengembang untuk mengirim permintaan dan menerima respons dalam format JSON tanpa perlu melakukan proses ini secara manual.
- 3) Manajemen Permintaan: Retrofit menyediakan berbagai fitur untuk mengelola permintaan HTTP, termasuk penjadwalan, pembatalan, dan penanganan kesalahan [11]. Ini memastikan bahwa aplikasi tetap responsif dan dapat mengelola koneksi jaringan dengan baik.
- 4) Integrasi dengan OkHttp: Retrofit dibangun di atas perpustakaan HTTP OkHttp, yang merupakan salah satu perpustakaan HTTP terkemuka untuk Android. Ini memberikan kinerja yang cepat dan andal, serta berbagai fitur lanjutan seperti caching, logging, dan pemantauan lalu lintas.

E. Figma

Figma merupakan alat desain berbasis web yang populer yang digunakan oleh desainer, pengembang, dan tim untuk membuat prototipe, desain antarmuka pengguna (UI), dan kolaborasi secara real-time. Berbeda dengan alat desain tradisional yang memerlukan instalasi perangkat lunak di komputer pengguna, Figma berjalan sepenuhnya di browser web, memungkinkan kolaborasi yang mudah antara tim yang bekerja secara remote. Berikut adalah beberapa poin penting tentang Figma:

- 1) Antarmuka Pengguna Intuitif: Figma memiliki antarmuka pengguna yang ramah pengguna dan mudah dipahami, yang memungkinkan pengguna untuk dengan cepat mulai membuat desain dan prototipe tanpa harus mempelajari banyak hal.

- 2) Prototyping Interaktif: Figma memungkinkan pengguna untuk membuat prototipe interaktif yang mencakup animasi, interaksi klik, dan transisi antar halaman. Ini memungkinkan pengguna untuk mengevaluasi dan menguji pengalaman pengguna mereka sebelum mengimplementasikannya dalam pengembangan aplikasi.
- 3) Kolaborasi Real-time: Figma memungkinkan beberapa pengguna untuk bekerja secara bersamaan pada desain yang sama dalam waktu nyata. Ini memungkinkan kolaborasi tim yang efisien, bahkan jika anggota tim berada di lokasi yang berbeda.
- 4) Komponen dan Variasi: Figma memungkinkan pengguna untuk membuat dan menggunakan komponen desain yang dapat digunakan kembali dalam berbagai proyek. Ini memungkinkan konsistensi desain dan pengembangan yang lebih cepat.
- 5) Dukungan untuk Berbagai Perangkat: Figma dapat digunakan untuk merancang antarmuka pengguna untuk berbagai platform, termasuk web, mobile, dan desktop. Ini membuatnya menjadi pilihan yang fleksibel untuk desain aplikasi dan situs web.

F. Expresso

Expresso adalah sebuah kerangka kerja pengujian untuk Node.js, khususnya dirancang untuk menguji aplikasi berbasis Express.js. Ini menyediakan berbagai fitur dan utilitas yang memudahkan pengembang untuk menulis dan menjalankan tes otomatis pada aplikasi Express.js mereka.

Dengan Expresso, pengembang dapat membuat dan menjalankan serangkaian tes yang mencakup berbagai aspek dari aplikasi mereka, seperti pengujian rute, pengujian middleware, dan pengujian unit untuk fungsi-fungsi khusus. Ekosistem pengujian yang kuat ini membantu meningkatkan keandalan dan kualitas aplikasi secara keseluruhan, serta memfasilitasi pengembangan berkelanjutan dan penyebaran aplikasi. Meskipun Expresso telah menjadi populer di kalangan pengembang Node.js, perlu dicatat bahwa pengembangan pada kerangka kerja ini telah berhenti sejak 2013, dan penggunaan alternatif seperti Jest atau Mocha lebih umum digunakan saat ini.

IV. HASIL DAN PEMBAHASAN

A. Frontend

Dalam proses pembuatan *frontend* dari aplikasi *Stress Detection App* terdapat beberapa software – software yang digunakan agar bisa menghasilkan antarmuka pengguna (UI) yang responsif dan menarik, serta pengalaman pengguna (UX) yang memuaskan.

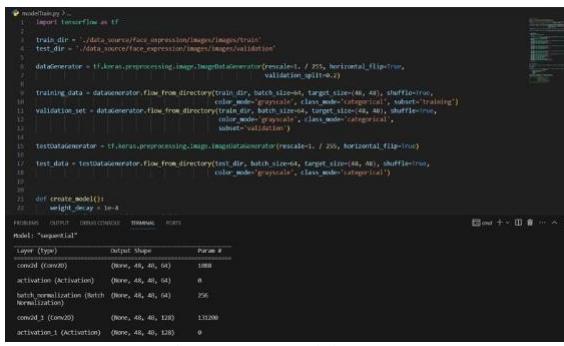
- 1) Figma



Gbr. 1 Pembuatan Prototype Dasar

Pada gambar 1 merupakan tahap perancangan awal dan pembuatan prototype dasar di Figma. Hal ini dilakukan agar mempermudah dalam memahami desain yang akan di aplikasikan.

2) Training Data



Gbr. 2 Proses Training Data

Pada gambar 2 dilakukan training data pada fitur camera untuk melakukan analisis wajah berdasarkan dataset yang diberikan. Selain itu, training data ini juga digunakan untuk data validation dari data gambar yang akan diperoleh menggunakan kamera.

B. Backend

Backend dari aplikasi *Stress Detection App* merupakan bagian dari sistem yang bertanggung jawab atas pemrosesan data, analisis, dan menyediakan layanan kepada *frontend*. Dalam konteks ini, *backend* bertujuan untuk mendukung fungsionalitas aplikasi, seperti pengenalan wajah dan analisis tingkat stres dari survei singkat.

1) Google Cloud Platform (GCP)

Backend aplikasi ini menggunakan layanan *Google Cloud Platform (GCP)* untuk infrastruktur awannya. GCP menyediakan fleksibilitas dalam membangun, mengelola, dan mendeploy aplikasi, serta layanan-layanan seperti *App Engine* dan *Datastore* yang digunakan untuk menyimpan data aplikasi [3].

2) Pembuatan Server

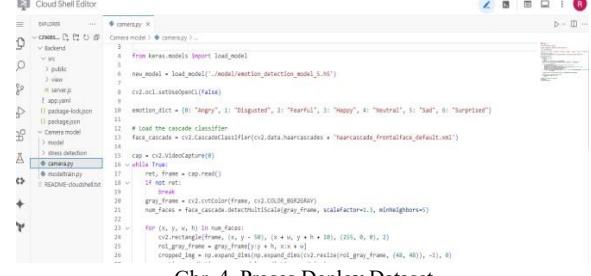
Pada gambar 3 merupakan gambar pembuatan server pada *App Engine*. Source code ini menggunakan Node.js dan Express.js sebagai teknologi utama dalam pengembangan *backend*. Node.js memungkinkan

penggunaan JavaScript di sisi server, sedangkan Express.js menyediakan kerangka kerja web yang minimalis dan fleksibel untuk Node.js [4][5]. Dengan kombinasi ini, pengembang dapat membangun *backend* yang responsif dan berkinerja tinggi.



Gbr. 3 Proses Pembuatan Server

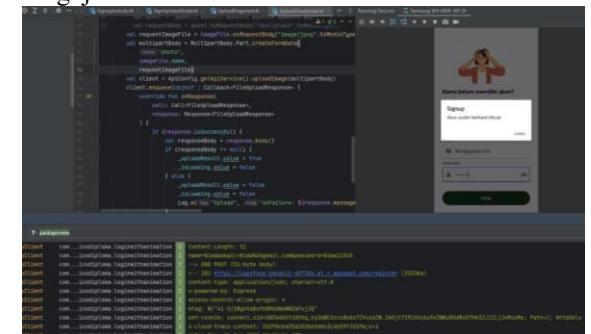
3) Deploy Dataset



Gbr. 4 Proses Deploy Dataset

Pada gambar 4 merupakan proses deploy dataset ke dalam *App Engine* dan *Datastore* untuk disimpan sebagai data baru. Pada proses ini, Python digunakan dalam konteks pembangunan aplikasi web, sementara Kotlin digunakan untuk pengembangan aplikasi Android. Kedua bahasa pemrograman ini memberikan fleksibilitas dan keamanan yang diperlukan dalam pengembangan aplikasi mobile [6].

4) Pengujian



Gbr. 5 Proses Pengujian

Gambar 5 merupakan pengujian terhadap aplikasi yang akan digunakan. Pada proses pengujian ini, Expresso digunakan sebagai kerangka kerja pengujian untuk Node.js, khususnya dalam menguji aplikasi berbasis Express.js. Ini memungkinkan

pengembang untuk menulis dan menjalankan tes otomatis pada *backend* mereka untuk meningkatkan keandalan dan kualitas aplikasi secara keseluruhan.

C. Tampilan Aplikasi

1) Halaman *Welcome*



Gbr. 6 Halaman *Welcome*

Halaman ini muncul saat pengguna baru pertama kali menggunakan aplikasi ini sebelum masuk ke dalam aplikasi.

2) Halaman Login Atau Daftar

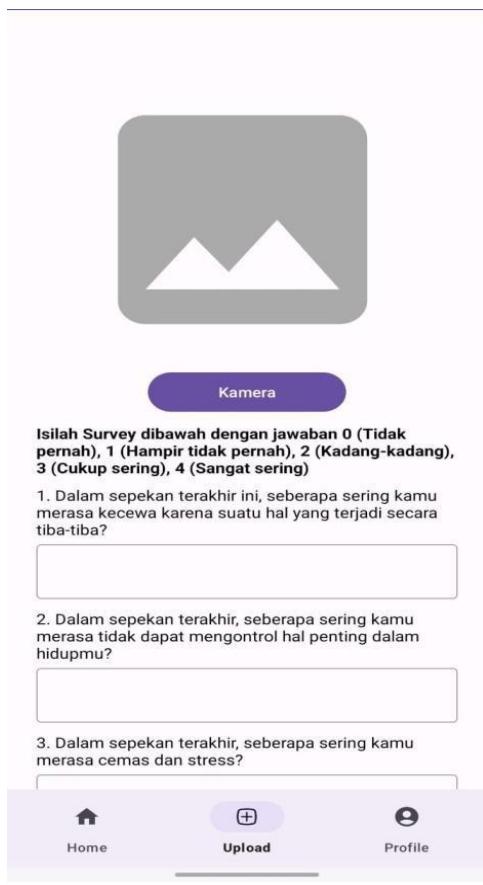
Pada Gambar 7 dan Gambar 8, pengguna diminta untuk mendaftarkan email apabila pengguna baru pertama kali menggunakan aplikasi ini dan belum pernah mendaftarkan emailnya.

Gbr. 7 Halaman Daftar

Gbr. 8 Halaman Login

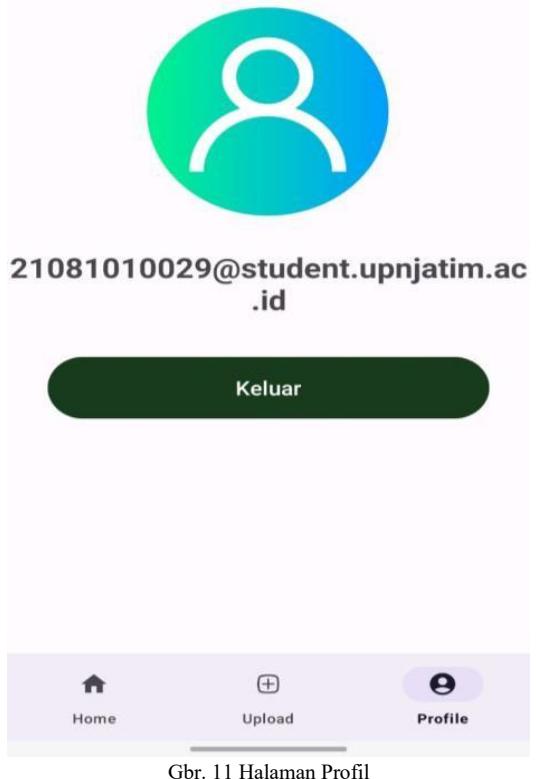


Gbr. 9 Halaman Beranda

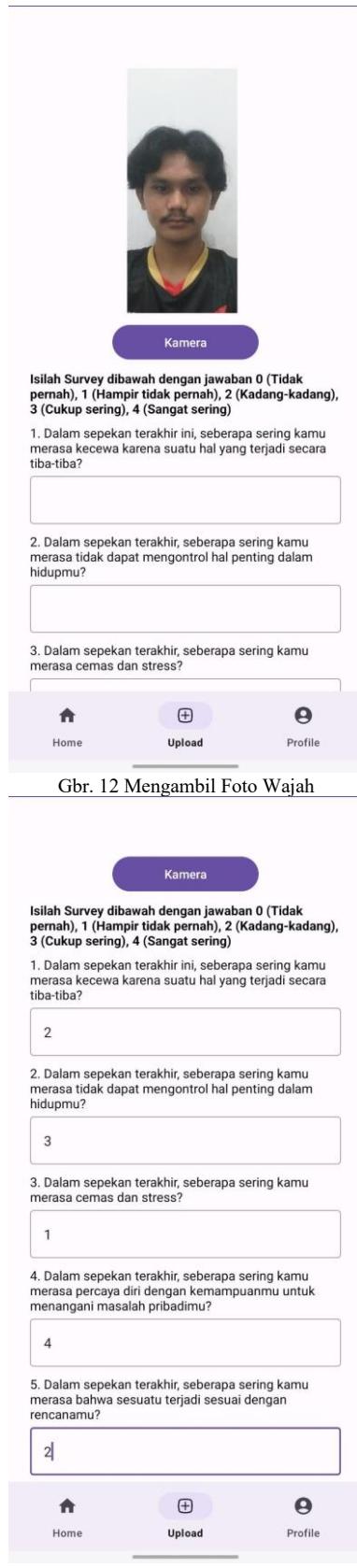


Gbr. 10 Halaman Utama

- 3) Halaman Beranda
Halaman Beranda pada Gambar 9 menampilkan beberapa fitur aplikasi seperti tips & trick, layanan konseling dan history. Selain itu pada history juga terdapat informasi – informasi tentang analisis tingkat stress dari pengguna.
- 4) Halaman Utama
Berdasarkan Gambar 10, halaman utama terdapat fitur kamera yang dapat menangkap gambar secara langsung dari pengguna untuk dilakukan analisis wajah pada pengguna.
- 5) Halaman Profil
Halaman pada Gambar 11 ini bertujuan apabila pengguna ingin keluar dari aplikasi ini pengguna tinggal klik tombol “logout/keluar” kemudian pengguna akan otomatis keluar dari aplikasi ini.



Gbr. 11 Halaman Profil



D. Hasil Pengujian Aplikasi

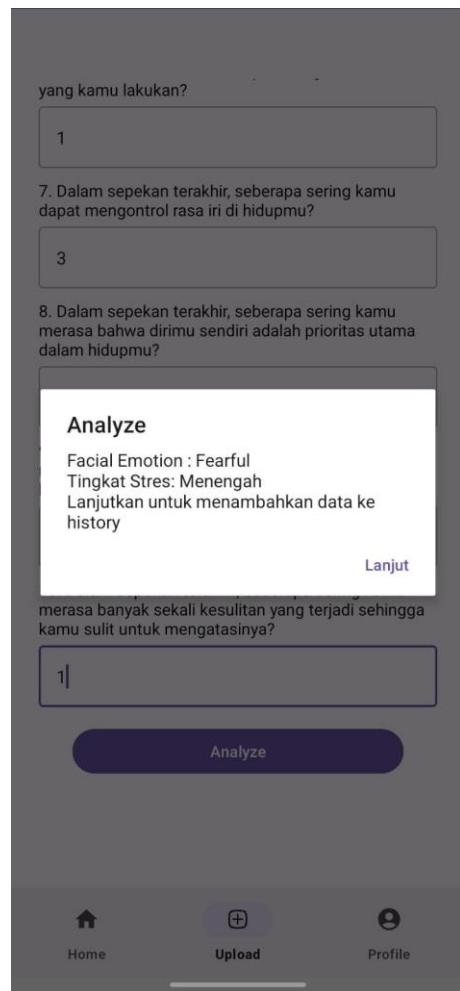
1) Mengambil Foto Wajah

Langkah pertama setelah login dan masuk ke aplikasi adalah mengambil foto wajah menggunakan fitur kamera yang telah disediakan di aplikasi. Foto harus memperlihatkan ekspresi wajah agar sistem mudah dalam melakukan analisis terhadap tingkat stress.

2) Mengisi Survey

Langkah selanjutnya yaitu mengisi survey melalui pertanyaan – pertanyaan yang telah disediakan di aplikasi. Hal ini dilakukan untuk lebih mendalam bagaimana kondisi pengguna secara *real time*. Survey diisi dengan 4 tingkatan pernyataan menggunakan angka yaitu 0 (Tidak Pernah), 1 (Hampir Tidak Pernah), 2 (Kadang-kadang), 3 (Cukup sering), dan 4 (Sangat Sering).

3) Hasil Analisis



Gbr. 14 Hasil Analisis

Setelah mengisi survey, langkah terakhir adalah melihat hasil analisis. Hasil analisis akan

mengelompokkan tingkat stress pengguna menjadi tiga tingkatan yaitu rendah, menengah, dan tinggi.

V. KESIMPULAN

Stress Detection App merupakan solusi yang inovatif dalam membantu pengguna untuk mengelola tingkat stres mereka melalui pengenalan wajah dan analisis survei singkat. Dalam pengembangannya, berbagai teknologi dan platform digunakan untuk memastikan kinerja, keamanan, dan pengalaman pengguna yang optimal.

Secara keseluruhan, aplikasi ini menggabungkan berbagai elemen teknologi, seperti *Google Cloud Platform* (GCP) untuk infrastruktur awan, Node.js dan Express.js untuk *backend*, serta Python dan Kotlin untuk pengembangan aplikasi web dan mobile. Integrasi dengan TensorFlow juga memungkinkan analisis wajah yang canggih untuk menentukan tingkat stres.

Selain itu, dalam pengembangan *frontend*, Figma digunakan untuk merancang antarmuka pengguna yang menarik dan responsif, sementara dalam pengembangan *backend*, Expresso digunakan untuk memastikan keandalan aplikasi melalui pengujian otomatis.

Stress Detection App dibuat untuk memberikan kontribusi positif dalam membantu pengguna mengatasi stres dan meningkatkan kesejahteraan mental mereka. Dengan penggunaan teknologi yang canggih dan pendekatan yang holistik, aplikasi ini memiliki potensi untuk memberikan dampak yang signifikan dalam meningkatkan kesehatan mental masyarakat.

UCAPAN TERIMA KASIH

Penulis menyampaikan rasa syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan jurnal berjudul “Perancangan dan Implementasi Aplikasi Mobile *Stress Detection App* Menggunakan *Google Cloud*” dengan baik. Jurnal ini sepenuhnya disusun dan dikerjakan sendiri oleh penulis sebagai bagian dari pengembangan pengetahuan dan penerapan teknologi di bidang informatika.

Ucapan terima kasih disampaikan pula kepada Universitas Pembangunan Nasional “Veteran” Jawa Timur, yang telah memberikan fasilitas serta dukungan akademik selama proses penelitian dan penulisan. Akhir kata, penulis berharap jurnal ini dapat memberikan manfaat dan inspirasi bagi pembaca, khususnya dalam bidang pengembangan aplikasi berbasis machine learning dan cloud computing.

REFERENSI

- [1] Al-Atawi, Dkk. (2023) *Stress Monitoring Using Machine Learning, IoT And Cloud Computing*. Sensors, 23(21): 8875..
- [2] Giannakakis, DKK. (2025). *Stress Detection Identifying Relevant Facial Action Units*. ScienceDirect.
- [3] P. Borra (2024). “A Survey of Google Cloud Platform (GCP): Features, Services, and Applications,” International Journal of Advanced Research in Science, Communication and Technology (IJARSCT).
- [4] Hemant Kumar & Dr. Karuna Sharma (2024). “Comparative Analysis Node.js (JavaScript Implementation in Server Side)”, International Journal of Research Publication and Reviews.
- [5] S. Verma (2024) “Express.Js and its Usage in Web Development,” International Journal of Research Publication and Reviews, vol. 5, no. 4, pp. 3247-3249.
- [6] V. Cutting and N. Stephen, (2021). “A Review on using Python as a Preferred Programming Language for Beginners,” International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 8.
- [7] M. S. Talib, A.-M. Al-Najar, A. H. Hassan and Z. S. Talib, (2024). “Kotlin Programming Language: A Comprehensive Overview of Definition, Applications, Advantages, and Limitations,” International Journal of Computer Applications Technology & Research.
- [8] Z. Maulidia and L. Venica, (2024). “Serverless Computing: Analisis Cloud Run dan App Engine dalam Profile Website Deployment,” SISTEMASI: Jurnal Sistem Informasi, vol. 13, no. 2.
- [9] K. Azkiya, M. Irsan, & M. F. Fathoni, (2024). “Implementation of App Engine and Cloud Storage as REST API on Smart Farm Application”, Sinkron: Jurnal dan Penelitian Teknik Informatika.
- [10] N. Wiranda, H. S. Purba, & R. A. Sukmawati (2023), “Survei Penggunaan Tensorflow pada Machine Learning untuk Identifikasi Ikan Kawasan Lahan Basah,” Indonesian Journal of Electronics and Instrumentation Systems.
- [11] M. H. B. Pratama , B. T. Hanggara , W. H. N. Putra (2025). "ANALISIS PERBANDINGAN PERFORMA NETWORKING LIBRARY RETROFIT DAN VOLLEY DALAM PENGAMBILAN DATA MENGGUNAKAN REST API", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.