

# Pengembangan Sistem Backend Aplikasi Identifikasi Kepribadian Anak Sebagai Rekomendasi Pendampingan Untuk Guru Dan Orang Tua

1<sup>st</sup> Rizky Ramadhani Syam

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

rizkysyam@ student.telkomuniversity  
.ac.id

2<sup>nd</sup> Casi Setianingsih

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3<sup>rd</sup> Marisa W. Paryasto

Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

**Abstrak** — Kepribadian seseorang mencerminkan identitas individu mereka, yang tidak selalu terlihat secara fisik. Kepribadian dapat dipengaruhi oleh lingkungan, keluarga, dan sifat bawaan sejak lahir. Kepribadian anak yang terbentuk melalui gaya pengasuhan di rumah secara signifikan berdampak pada lingkungan sekolahnya, sehingga penting bagi orang tua dan guru untuk memahami kepribadian anak agar dapat memberikan dukungan yang tepat. Namun, tes psikologi yang ada saat ini yang digunakan untuk menilai kepribadian anak tidak efektif karena durasinya yang panjang. Sehingga diperlukan aplikasi mobile yang mampu mengidentifikasi kepribadian anak melalui analisis garis telapak tangan. Dalam pengembangannya diperlukan sistem backend untuk memproses data. Sistem dibuat menggunakan FastAPI dan PostgreSQL dan di-deploy pada layanan google cloud platform. Pengujian beban pada sistem backend yang dibuat menunjukkan bahwa sistem dapat berjalan efisien pada beban ringan dan sedang, namun pada beban berat, sistem mengalami penurunan performa.

**Kata kunci**— kepribadian anak, aplikasi mobile, sistem backend

## I. PENDAHULUAN

Kepribadian menggambarkan ciri dari sifat seseorang [1]. Walaupun kepribadian merupakan bawaan lahir, tapi kepribadian dapat dibentuk dan diarahkan pada usia *golden ages* atau usia dini. Tahap *golden ages* anak tidak dapat diulang di masa mendatang. Oleh karena itu, penting untuk mengetahui kepribadian anak. Lingkungan keluarga dan sekolah adalah salah satu factor yang mempengaruhi kepribadian anak [2]. Beberapa pihak yang berperan aktif didalamnya adalah orang tua dan guru. Sehingga penting bagi orang tua dan guru untuk mengetahui jenis kepribadian anak. Sehingga kepribadian anak dapat diarahkan ke kepribadian yang positif.

Kepribadian seseorang dapat dikenali dengan melakukan beberapa tes kepribadian. Beberapa cara yang biasa dilakukan untuk mengetahui kepribadian seseorang antara lain, tes big five personalities yang dilakukan dengan cara menjawab pertanyaan dan menemui psikolog [3], [4]. Namun, cara tersebut masih memerlukan waktu lebih

banyak. Biaya yang dikeluarkan juga cukup besar yaitu mulai dari Rp 200.000 hingga Rp 400.000 [5]. Untuk mengatasi beberapa masalah tersebut, diperlukan cara yang lebih optimal agar tidak menghabiskan waktu dan biaya yang besar. Saat ini kepribadian dapat diidentifikasi berdasarkan biometrik yang dimiliki masing-masing orang. Biometrik yang umum digunakan yaitu sidik jari dan garis telapak tangan [6], [7]. Namun metode telapak tangan masih terbatas dalam penggunaannya.

Untuk memanfaatkan potensi ini, penelitian ini bertujuan untuk membuat aplikasi untuk identifikasi kepribadian. Untuk mencapai tujuan ini, maka dibuat sistem *backend* dengan *Application Programming interface* (API) menggunakan *framework* FastAPI yang digunakan untuk menghubungkan aplikasi identifikasi kepribadian dengan basis data. Dan juga dibuat basis data menggunakan PostgreSQL sebagai tempat penyimpanan data aplikasi identifikasi kepribadian. Dengan pembuatan sistem ini diharapkan aplikasi identifikasi kepribadian dapat berjalan efisien.

## II. KAJIAN TEORI

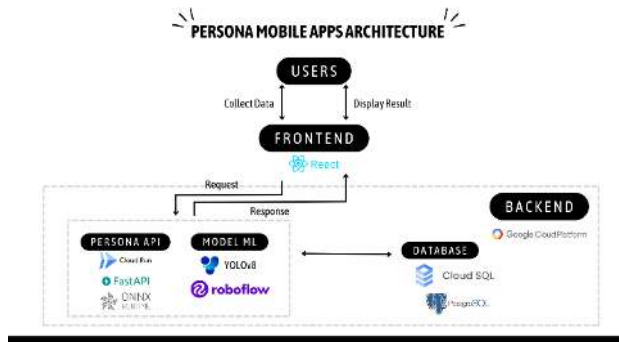
FastAPI adalah framework web modern berbasis Python yang digunakan untuk membangun API dengan cepat dan mudah. Salah satu keunggulan utama FastAPI adalah kemampuannya untuk menangani permintaan asinkron secara efisien, yang memungkinkan penanganan lebih banyak permintaan dengan latensi yang lebih rendah. Selain itu, FastAPI secara otomatis menghasilkan dokumentasi API menggunakan Swagger UI dan ReDoc, yang sangat memudahkan pengembangan dan pengujian API [8].

*Database management system* (DMBS) adalah Sistem pengorganisasian dan pengolahan database pada komputer. Sistem ini dirancang untuk mampu melakukan berbagai data dengan beberapa referensi data yang sama. DBMS ini mampu diakses oleh berbagai aplikasi. Salah satu DMBS yang populer digunakan adalah PostgreSQL. PostgreSQL adalah sistem manajemen basis data relasional yang dikenal karena keandalannya, skalabilitasnya, dan dukungannya terhadap

standar SQL. PostgreSQL menawarkan banyak fitur canggih seperti dukungan untuk JSON, replikasi data, dan kemampuan untuk menangani transaksi ACID (Atomicity, Consistency, Isolation, Durability), yang memastikan integritas dan konsistensi data. [10]. [9].

### III. METODE

Rancangan dari aplikasi identifikasi kepribadian dapat dilihat pada gambar berikut:



GAMBAR 1  
Arsitektur aplikasi identifikasi kepribadian

Backend dari Arsitektur Aplikasi Identifikasi Kepribadian akan dibangun di Google Cloud Platform (GCP) dan memanfaatkan Cloud SQL dengan PostgreSQL sebagai sistem manajemen basis data. Infrastruktur backend mencakup Persona API, yang dikembangkan menggunakan FastAPI dan berjalan di Google Cloud Run, menyediakan layanan API yang dapat diskalakan dan efisien. Model pembelajaran mesin yang digunakan untuk deteksi objek, diintegrasikan dengan ONNX Runtime untuk kinerja yang dioptimalkan untuk pengembangan dan penerapan model.

#### A. Basis Data

Membuat arsitektur basis data menggunakan PostgreSQL melibatkan beberapa tahap. Pertama, analisis kebutuhan data aplikasi untuk memahami jenis data, hubungan, dan volume. Berdasarkan analisis ini, mendesain skema basis data, mendefinisikan tabel, kolom, dan hubungan. Selanjutnya, mengimplementasikan skema dengan membuat tabel dan mendefinisikan strukturnya pada aplikasi postgresQL.

#### B. Sistem API

Tahapan pertama dalam pembuatan sistem API aplikasi identifikasi kepribadian adalah melakukan pengaturan database. Tahapan ini melibatkan koneksi database dan inisialisasi database yang akan digunakan. Serta konfigurasi alembic untuk migrasi database. Tahapan selanjutnya adalah mendefinisikan Schemas. schemas merupakan representasi data yang digunakan untuk validasi data yang masuk dan keluar dari aplikasi. Pembuatan schemas menggunakan library pydantic.

Selanjutnya adalah implementasi service, tahapan ini merupakan tahap penulisan logika bisnis yang akan dijalankan oleh API, mencakup pembuatan fungsi operasi Create, Read, Update dan Delete (CRUD) serta fungsi lain

yang diperlukan. setelah mengimplementasikan servis, Tahapan berikutnya adalah mendefinisikan endpoint. Endpoint merupakan jalur atau URL yang dapat diakses klien untuk berinteraksi dengan API.

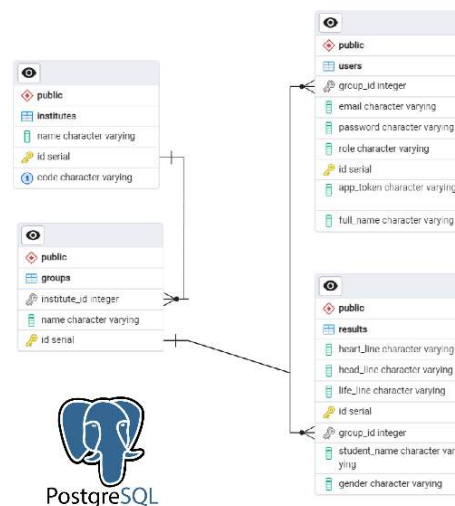
#### C. Deploy API

Agar sistem backend dapat digunakan pada aplikasi identifikasi kepribadian diperlukan untuk men-deploy sistem backend di cloud. pada aplikasi identifikasi kepribadian ini sistem backend di-deploy pada layanan google cloud platform. Secara spesifik, Cloud Run digunakan untuk sistem API, dan Cloud SQL digunakan untuk database.

### IV. HASIL DAN PEMBAHASAN

#### A. Basis Data

Sistem manajemen basis data yang digunakan dalam pengembangan basis data aplikasi persona adalah PostgreSQL. Terdapat 4 entitas dalam database ini, yaitu 'users', 'groups', 'institutes', dan 'results'. 'users' untuk menyimpan data pengguna, 'groups' untuk menyimpan data kelas siswa, 'institutes' untuk menyimpan data sekolah, dan 'results' untuk menyimpan data hasil identifikasi kepribadian. Berikut adalah Entity relationship diagram untuk database aplikasi persona:



GAMBAR 2.  
Database Entity Relationship Diagram.

Gambar 2 menunjukkan bahwa Tabel 'institutes' menyimpan data institut dengan kolom 'id', 'nama', dan 'kode'. Tabel 'groups' berhubungan dengan institusi melalui kolom 'institute\_id' dan memiliki kolom 'id' dan 'nama'. Tabel 'users' menghubungkan pengguna dengan grup melalui kolom 'group\_id' dan memiliki kolom 'email', 'password', 'role', 'app\_token', dan 'nama\_lengkap'. Tabel 'hasil' mencatat hasil yang terkait dengan grup dan pengguna melalui kolom 'group\_id', dan memiliki kolom 'nama\_siswa', 'jenis\_kelamin', 'garis\_hati', 'garis\_kepala', dan 'garis\_hidup'.

#### B. Sistem API

Ada 4 kategori utama endpoint yang dibuat yaitu institut, pengguna, grup, dan hasil. Endpoint adalah sebuah URL yang menjadi titik akses dari sebuah API yang akan dipanggil ketika Anda ingin menggunakan sistem API aplikasi persona.

Setiap endpoint diasosiasikan dengan satu metode HTTP seperti GET, POST, PUT, DELETE, dll. Setiap metode ini memiliki kegunaannya masing-masing. Setiap metode ini memiliki kegunaannya masing-masing. Berikut ini adalah daftar API yang ada di aplikasi Persona:

Institute	
DELETE	/api/v1/institute/ Delete
GET	/api/v1/institute/code/{code} Get
GET	/api/v1/institute/id/{id} Get
GET	/api/v1/institute/ GetAll
POST	/api/v1/institute/ Create
PUT	/api/v1/institute/ Update

GAMBAR 3  
Institute Endpoint.

Pada daftar endpoint “Institute” mencakup operasi CRUD (Create, Read, Update, Delete). Endpoint DELETE di ‘/api/v1/institute/’ digunakan untuk menghapus data pada tabel institute. Endpoint GET di ‘/api/v1/institute/code/{code}’ digunakan untuk mengambil data Institute berdasarkan kode tertentu. Endpoint GET di ‘/api/v1/institute/id/{id}’ digunakan untuk mengambil data Institute berdasarkan ID tertentu. Endpoint GET di ‘/api/v1/institute/’ digunakan untuk mengambil semua data institute yang tersedia. Endpoint POST di ‘/api/v1/institute/’ digunakan untuk membuat atau menambahkan data institusi baru. Terakhir, endpoint PUT di ‘/api/v1/institute/’ digunakan untuk memperbarui/ menyunting data institusi yang sudah ada.

User	
DELETE	/api/v1/user/ Delete
GET	/api/v1/user/email/{email} Get By Email
GET	/api/v1/user/{id} Get
GET	/api/v1/user/ GetAll
POST	/api/v1/user/ Create
POST	/api/v1/user/login Login
POST	/api/v1/user/bk Create Bk
PUT	/api/v1/user/ Update
PUT	/api/v1/user/group Update Group

GAMBAR 4  
User Endpoint.

Pada daftar endpoint “User”, terdapat beberapa endpoint yaitu, endpoint DELETE ‘/api/v1/user/{id}’ digunakan untuk menghapus data pengguna berdasarkan ID. Endpoint GET ‘/api/v1/user/email/{email}’ digunakan untuk mengambil data pengguna berdasarkan email dari data pengguna yang ada. Endpoint GET ‘/api/v1/user/{id}’ digunakan untuk mengambil data pengguna berdasarkan ID yang dimasukkan. Endpoint GET ‘/api/v1/user/’ digunakan untuk mengambil semua data pengguna yang ada pada database. Endpoint POST ‘/api/v1/user/’ digunakan untuk membuat atau memasukkan data pengguna baru. Endpoint POST ‘/api/v1/user/login’ digunakan untuk proses login pengguna. Endpoint POST ‘/api/v1/user/BK’ digunakan untuk membuat user dengan role BK. Endpoint PUT ‘/api/v1/user/{id}’

digunakan untuk memperbarui informasi data pengguna yang telah ada sebelumnya.

Group	
DELETE	/api/v1/group/ Delete
GET	/api/v1/group/institute/{institute_id} Get By Institute Id
GET	/api/v1/group/{id} Get
GET	/api/v1/group/ GetAll
POST	/api/v1/group/ Create
PUT	/api/v1/group/ Update

GAMBAR 5  
Group Endpoint.

Pada daftar endpoint “Group”. Terdapat beberapa endpoint yaitu, endpoint DELETE ‘/api/v1/group’ digunakan untuk menghapus data group berdasarkan id. Endpoint GET ‘/api/v1/group/institute/{institute\_id}’ digunakan untuk mengambil data grup berdasarkan “institute\_id” dari data grup yang ada. Endpoint GET ‘/api/v1/group/{id}’ digunakan untuk mengambil data grup berdasarkan ID yang dimasukkan. Endpoint GET ‘/api/v1/group/’ digunakan untuk mengambil semua data grup yang ada pada database. Endpoint POST ‘/api/v1/group/’ digunakan untuk membuat atau memasukkan data grup baru. Endpoint PUT ‘/api/v1/group/’ digunakan untuk memperbarui informasi data grup yang telah ada sebelumnya.

Result	
GET	/api/v1/result/ GetAll
GET	/api/v1/result/group/{group_id} Get By Group Id
GET	/api/v1/result/institute/{institute_id} Get By Institute Id
POST	/api/v1/result/remove-bg Remove Bg
POST	/api/v1/result/predict Predict
POST	/api/v1/result/ Create

GAMBAR 6  
Result Endpoint.

Pada daftar endpoint “Result”, terdapat beberapa endpoint yaitu, endpoint GET ‘/api/v1/result/’ digunakan untuk mengambil semua data hasil yang ada pada database. Endpoint GET ‘/api/v1/result/group/{group\_id}’ digunakan untuk mengambil data hasil berdasarkan “group\_id” dari data hasil yang ada. Endpoint GET ‘/api/v1/result/institute/{institute\_id}’ digunakan untuk mengambil data hasil berdasarkan “institute\_id” dari data hasil yang ada. Endpoint POST ‘/api/v1/result/predict’ digunakan untuk melakukan prediksi kepribadian melalui model machine learning yang telah dibuat. Endpoint POST ‘/api/v1/result/’ digunakan untuk melakukan prediksi kepribadian serta memasukkannya ke dalam database.

### C. Deploy API

Google Cloud Run dan Google Cloud SQL merupakan dua layanan penting yang digunakan dalam pengembangan sistem backend untuk aplikasi identifikasi kepribadian anak. Berikut ini penjelasan mengenai pengembangan dan spesifikasi keduanya:

### 1. Google Cloud Run

Setelah membuat API, deploy API tersebut ke google cloud platform. Untuk menghubungkan aplikasi persona dengan database yang telah dibuat adapun konfigurasi cloud service dijalankan dari sistem API aplikasi persona.

TABLE I.  
SPESIFIKASI CLOUD RUN

Specifications	Specification Information
Region	Asia-southeast2(Jakarta)
Container Port	8080
CPU Allocation	CPU is only allocated during request processing
Memory	2 Gb
CPU Size	1 CPU
Request Timeout	300 Seconds
Maximum Request	80 Concurrent in 1 Instance
Environment Execution	Second Generation
Instance Standby	Min:0 Max:10
Link Service	https://Persona-api-2nryopyb2q-et.a.run.app

Database berada di wilayah Asia-tenggara2 (Jakarta) dengan container port 8080 dan alokasi CPU yang hanya digunakan pada saat proses request. Sistem ini memiliki memori sebesar 2 GB dan ukuran CPU sebesar 1 unit. Waktu tunggu permintaan diatur ke 300 detik, dengan maksimum 80 permintaan bersamaan dalam 1 instance. Lingkungan eksekusi yang digunakan adalah second generation, dengan jumlah standby instance berkisar antara 0 hingga 10.

### 2. Google Cloud SQL

Setelah membuat database, deploy database ke google cloud platform agar dapat terhubung ke aplikasi persona melalui API Berikut adalah spesifikasi konfigurasi Cloud SQL:

TABLE II.  
SPESIFIKASI CLOUD SQL

Specifications	Specification Information
Region	Asia-southeast2(Jakarta)
DB Version	PostgreSQL 15.5
vCPUs	1 vCPUs
Memory	628.74 MB
Storage	10GB

Spesifikasi database yang digunakan adalah di wilayah Asia-tenggara2 (Jakarta) dengan PostgreSQL versi 15.5. Basis data ini didukung oleh 1 vCPU dengan memori 628,74 MB dan penyimpanan 10 GB.

### D. Pengujian Beban

Pengujian beban adalah jenis pengujian yang mengevaluasi kinerja, keandalan, dan stabilitas sistem dalam kondisi ekstrem atau pada kapasitas maksimum. Tujuan dari pengujian beban adalah untuk menentukan perilaku sistem dan kemampuannya untuk berfungsi di bawah beban yang berat, di luar batas operasional normalnya. Hasil dari skenario 3 menunjukkan tidak ada kegagalan permintaan dan memiliki waktu respons rata-rata 20 detik. Waktu permintaan tercepat adalah 7 detik, dan yang terlama adalah 36 detik. Waktu respons rata-rata adalah 20 detik, dengan deviasi standar 6 detik dan persentil ke-95 29 detik.

TABEL III.  
HASIL PENGUJIAN BEBAN

Scenari o	Avg. Response Time	Error Rate	Min. Response Time	Max. Response Time	Standar Deviation	P95 Response Time
1	4 s	0%	2 s	10 s	1 s	6 s
2	6 s	0%	3 s	12 s	1 s	11 s
3	20 s	0%	7 s	36 s	6 s	34 s

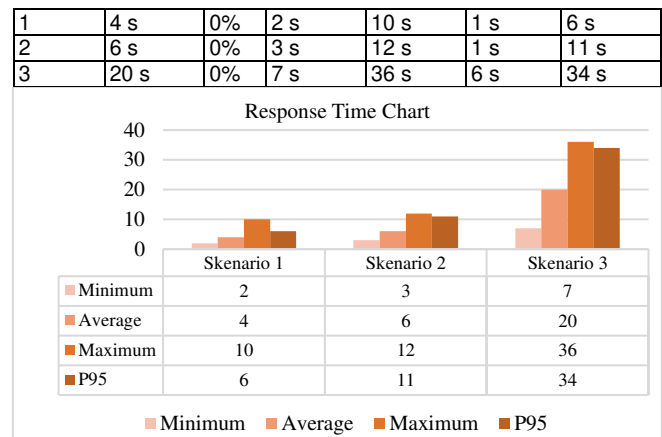


FIG. 1.

Respon Time Chart

Dari hasil pengujian beban API, terlihat bahwa sistem cenderung stabil ketika menangani hingga 200 request tanpa mengalami kegagalan. Waktu respon rata-rata meningkat secara signifikan; pada skenario 1 dan 2, waktu respon rata-rata relatif cepat, namun pada skenario 3 dengan 15 pengguna virtual, waktu respon sistem API meningkat secara signifikan menjadi 20 detik. Hal ini mengindikasikan bahwa server mulai kesulitan untuk mengelola jumlah permintaan yang lebih besar.

## V. KESIMPULAN

Berdasarkan penelitian dan pengembangan yang telah dilakukan, dapat ditarik Kesimpulan. sistem backend yang dikembangkan berhasil menyediakan solusi yang efisien dan biaya rendah untuk memahami dan mendukung kepribadian anak. *Deployment* di Google Cloud Platform (GCP) menjamin skalabilitas dan keandalan sistem. Ini menunjukkan potensi aplikasi ini untuk digunakan secara luas oleh guru dan orang tua dalam mendukung pembangunan kepribadian anak.

## REFERENSI

- [1] I. Ali, "Personality traits, individual innovativeness and satisfaction with life," *Journal of Innovation & Knowledge*, vol. 4, no. 1, pp. 38-46, 2019.
- [2] I. Imroatun, F. Widat, M. Fauziddin, S. Farida, S. Maryam and Zulaiha, "Youtube as a Media For Strengthening Character Education in Early Childhood," *2nd Bukittinggi International Conference on Education (BICED)*, vol. 1779, no. 1, pp. 1-6, 2021.
- [3] C. Barbaranelli, G. V. Caprara, A. Rabasca and C. Pastorelli, "A questionnaire for measuring the Big Five in late childhood," *Personality and Individual Differences*, vol. 34, no. 4, pp. 645-664, 2003.
- [4] F. B. Evans and S. E. Finn, "Training and Consultation in Psychological Assessment With Professional Psychologists: Suggestions for Enhancing the Profession and Individual Practices," *Journal of Personality Assessment*, vol. 99, no. 2, pp. 175-185, 2016.

- [5] B. T. Yates and J. Taub, "Assessing the Costs, Benefits, Cost-Effectiveness, and Cost-Benefit of Psychological Assessment: We Should, We Can, and Here's How," *Psychological Assessment*, vol. 15, no. 4, pp. 478-495, 2003.
- [6] E. A. Kastama, Variasi perilaku manusia menurut sidik jarinya, Jakarta: Lembaga Penerbit Fakultas Ekonomi Universitas Indonesia, 2000.
- [7] H. H. K. Tin, "An Effective Method of a Person's Character or Future using the Palm Print Images," *3rd International Conference on Multidisciplinary Research & Practice*, vol. IV, no. I, pp. 9-12, 2016.
- [8] "FastAPI," FastAPI, [Online]. Available: <https://fastapi.tiangolo.com/>. [Diakses 25 Desember 2023].
- [9] "PostgreSQL," PostgreSQL, [Online]. Available: <https://www.postgresql.org>. [Diakses 8 July 2024].
- [10] M. Fikry, Basis Data, Lhokseumawe: Unimal Press, 2019.

