

Tugas Besar II3160 Teknologi Sistem Terintegrasi

Website Pengelolaan Uang - Finalyze



Disusun Oleh:

Micky Valentino 18222093

Program Studi Sistem dan Teknologi Informasi

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

2024

Daftar Isi

Daftar Isi.....	2
Business Capability Mapping.....	3
Core Capability.....	5
System Modeling.....	7
Software Architecture.....	8
Rencana Pengembangan.....	10
Implementasi Service.....	12
1. OAuth.....	12
2. Pencatatan Pengeluaran.....	15
3. AI Anomaly Detection.....	17
4. Pemanggilan Service Orang Lain.....	19
Implementasi Spesifikasi.....	20
1. Penggunaan certificate HTTPS untuk website.....	20
2. Penggunaan Docker.....	20
3. Penggunaan Service Finalyze.....	21
Lampiran.....	22

Business Capability Mapping

Core Capabilities and Value Chains

Manajemen Pelanggan



Manajemen Transaksi



Enabling Capabilities

Manajemen Pemasaran dan Engagement



Manajemen Teknologi Support

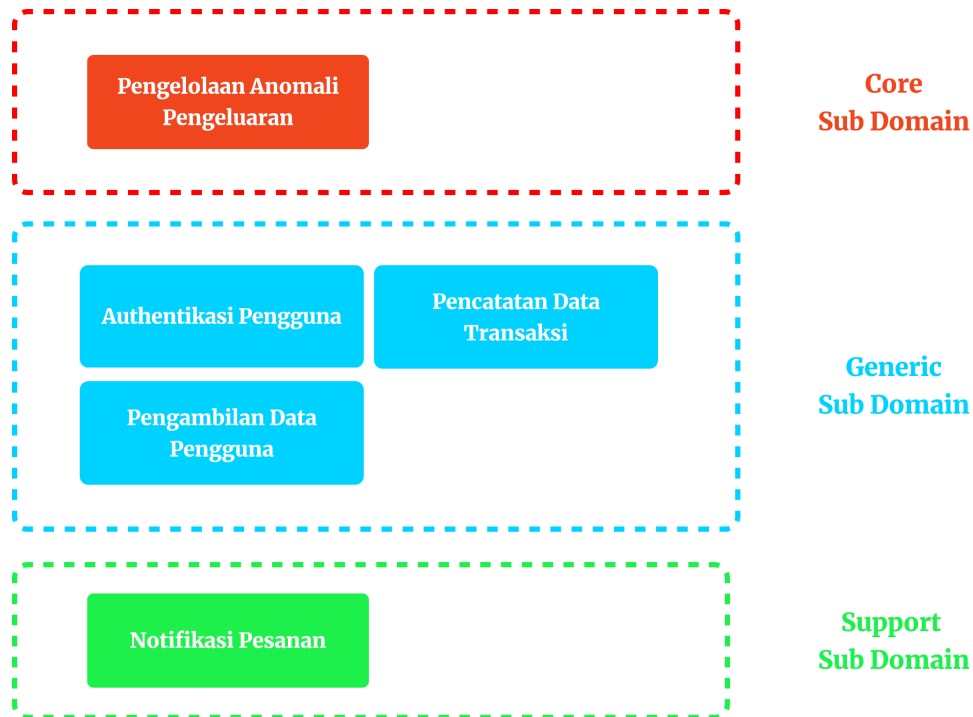


Finalyze dirancang untuk orang yang ingin melakukan manajemen keuangan dengan memberikan layanan terpadu melalui berbagai fitur inti yang mencakup **Manajemen Pelanggan** dan **Manajemen Transaksi**. Selain itu, **Kemampuan Pendukung** hadir untuk memperkuat interaksi pengguna melalui fitur pemasaran seperti manajemen kampanye dan promosi, serta program engagement personal dan loyalitas pengguna. Website ini juga dilengkapi dengan teknologi pendukung untuk pengelolaan data, pembuatan, hingga pemeliharaan situs secara efisien.

Core Capability

Pengelolaan Anomali Pengeluaran

Pengelolaan anomali pengeluaran berbasis machine learning bertujuan untuk **mendeteksi pengeluaran yang tidak wajar** atau mencurigakan dengan membandingkannya terhadap pola pengeluaran pengguna yang telah terbentuk sebelumnya dan harga wajar pengeluaran. Ketika ada transaksi yang signifikan menyimpang dari pola normal, sistem akan mengidentifikasi dan menandainya sebagai anomali. Pendekatan ini dapat digunakan untuk **meningkatkan efisiensi pengelolaan keuangan, mencegah potensi penipuan, atau memberikan wawasan** kepada pengguna agar lebih sadar dalam pengeluarannya.



Penjelasan

Rekomendasi Penetapan Partner

Pengelolaan anomali pengeluaran berbasis machine learning mendeteksi pengeluaran yang tidak wajar dengan membandingkannya terhadap pola pengeluaran pengguna yang telah terbentuk. Sistem ini mengidentifikasi anomali menggunakan algoritma pembelajaran mesin, membantu mencegah penipuan dan memberikan wawasan keuangan.

Authentikasi Pengguna

Melakukan login dengan menggunakan Open . Sub-domain ini memiliki fungsi untuk mengidentifikasi pelanggan agar mengurangi risiko penipuan

Pencatatan Data Transaksi

Melakukan pencatatan data transaksi ke database. Sub-domain ini memiliki fungsi untuk menyimpan histori dalam transaksi dan dapat dijadikan data untuk training machine learning selanjutnya

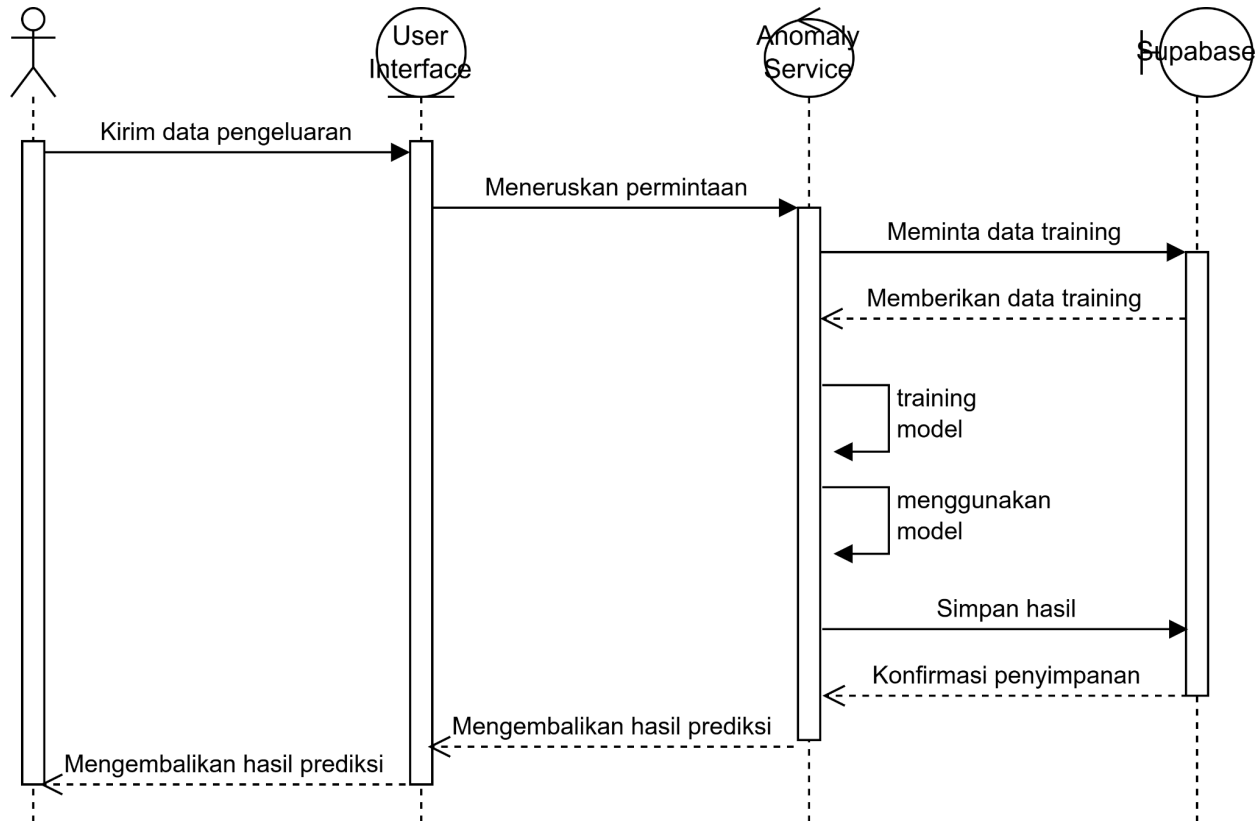
Pengambilan Data Transaksi

Melakukan pengambilan data transaksi dari database. Sub-domain ini memiliki fungsi untuk menampilkan histori itransaksi untuk keperluan user maupun training data untuk membuat machine learning

Notifikasi Pesanan

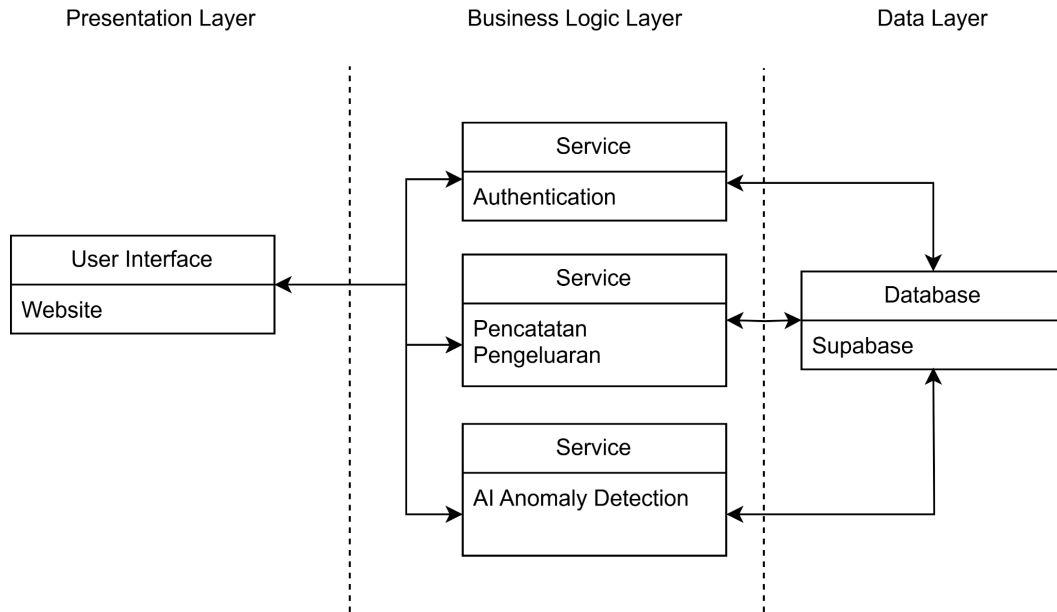
Menampilkan notifikasi bahwa pesanan sudah dibuat. Subdomain ini memiliki fungsi untuk membuat pelanggan menjadi lebih menyadari bahwa transaksi sudah berhasil dibuat

System Modeling



Proses pendeteksian anomali dimulai dari pengiriman data pengeluaran oleh pengguna melalui user interface, dan diteruskan ke **Anomaly Detection Service**. Layanan ini kemudian meminta data training yang relevan dari **Supabase** dan setelah menerima data training, layanan melakukan proses pelatihan model, atau menggunakan model yang telah tersedia, untuk mendeteksi potensi anomali pada data pengeluaran yang diterima. Hasil prediksi anomali akan disimpan di database serta dikembalikan ke **User Interface**, yang kemudian meneruskannya kembali ke pengguna.

Software Architecture



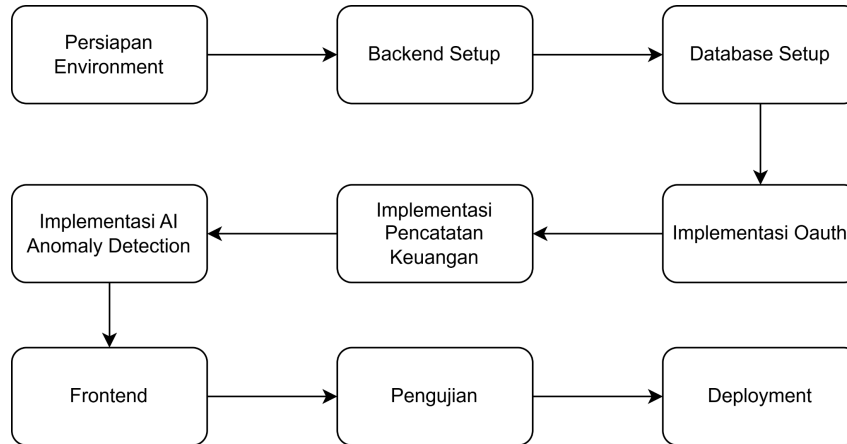
Software dan Development Environment

- **Backend Framework:**
 - **FastAPI (Python web framework)**
Framework web berbasis Python yang digunakan untuk membangun backend aplikasi. Di proyek ini, FastAPI menyediakan rute-rute untuk autentikasi, deteksi anomali, dan rendering halaman dengan HTML melalui template Jinja2
 - **Supabase (Database and authentication)**
Platform backend-as-a-service yang digunakan untuk database dan autentikasi. Proyek ini menggunakan Supabase untuk menyimpan data transaksi, hasil analisis anomali, dan data pengguna
- **Machine Learning:**
 - **scikit-learn (Isolation Forest for anomaly detection)**
- **Authentication:**
 - **Google OAuth (User authentication)**
Layanan autentikasi yang memungkinkan pengguna login dengan akun Google mereka.
- **Frontend:**
 - **HTML/CSS**
Digunakan di sisi frontend untuk merancang halaman dan elemen seperti tombol dan efek animasi dibuat menggunakan CSS untuk tampilan yang interaktif

- Jinja2 (Template engine)
Template engine yang digunakan untuk merender halaman HTML dengan data dinamis dari backend FastAPI
- Development Tools:
 - Python dotenv (Environment management)
Digunakan untuk mengelola variabel lingkungan

Rencana Pengembangan

Alur Pengembangan



1. Persiapan Environment

Pada tahap ini, dilakukan konfigurasi awal seperti setup Supabase dan variabel lingkungan menggunakan dotenv untuk memuat data - data penting, seperti supabase_url dan supabase_key

2. Backend Setup

Backend dikembangkan menggunakan framework FastAPI. Backend berisi pengaturan rute, validasi API key, dan integrasi pada halaman - halaman yang dibuat. Selain itu, pembuatan rute - rute sesuai dengan kebutuhan website

3. Database Setup

Database dikonfigurasi melalui Supabase untuk menghasilkan data - data yang sesuai dengan kebutuhan, seperti data user dan data keuangan. Dilakukan konfigurasi juga terhadap koneksi Supabase agar dapat digunakan.

4. Implementasi OAuth

Melakukan setup untuk OAuth menggunakan Google dan mengkaitkannya dengan supabase.

5. Implementasi Pencatatan Keuangan

Membuat logika untuk mencatat keuangan seperti pengeluaran dan pemasukan dengan memasukkannya ke dalam supabase.

6. Implementasi AI Anomaly Detection

Membuat logika untuk deteksi anomali menggunakan model machine learning. Dengan data training berupa data - data histori dari pengguna.

7. Frontend

Membuat interface di setiap page yang ada

8. Pengujian

Menguji endpoint API, mengetes otentikasi user, input user, dan pengecekan performa model machine learning.

9. Deployment

Sistem dapat di deploy di menggunakan Vercel, Azure, atau Railway

Timeline Pengembangan

Langkah - Langkah	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
	1	8	15	22	29	36	43
Persiapan Environment							
Backend Setup							
Database Setup							
Implementasi AI Anomaly Detection							
Implementasi Pencatatan Keuangan							
Implementasi OAuth							
Frontend							
Pengujian							
Deployment							

Implementasi Service

1. OAuth

Implementasi OAuth di Finalyze menggunakan Google OAuth 2.0 untuk autentikasi pengguna dengan step - step pengerjaan sebagai berikut:

- a. Penyimpanan informasi kredensial Google OAuth (menggunakan .env) agar dapat digunakan dalam login pengguna

```
class Settings:

    GOOGLE_CLIENT_ID = os.getenv("GOOGLE_CLIENT_ID")

    GOOGLE_CLIENT_SECRET = os.getenv("GOOGLE_CLIENT_SECRET")

    CALLBACK_URL = os.getenv("CALLBACK_URL")

settings = Settings()
```

- b. Setup Console Google dengan cara membuat APIAuth baru lalu menambahkan Authorized link yang berlaku serta menambahkan API Supabase. Setup Supabase dengan menambahkan ClientID dan Client Secret agar dapat diakses Supabase.

Name *

Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

1

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins

For use with requests from a browser

URIs 1 *

http://127.0.0.1:8000

URIs 2 *

http://localhost:8000

URIs 3 *

https://finalize.up.railway.app

+ ADD URI

Authorized redirect URIs

For use with requests from a web server

URIs 1 *

https://eqbdcwxzxbndcxvbdsgm.supabase.co/auth/v1/callback

URIs 2 *

https://oauth-theta.vercel.app/auth/callback

URIs 3 *

http://localhost:8000/auth/callback

URIs 4 *

http://localhost:8000/auth/CALLBACK_URL

URIs 5 *

https://finalize.up.railway.app/auth/callback

+ ADD URI

Additional information

Client ID	810904250415-roqje3bd7c8m7i5g92ad8idr34md86hb.apps.googleusercontent.com
Creation date	November 24, 2024 at 2:28:01 PM GMT+7

Client secrets

If you are in the process of changing client secrets, you can manually rotate them without downtime. [Learn more](#)

Client secret	GOCSPX-c-T00AMVYIfdPQuFELj6APxerGr	<div><div></div><div></div></div>
Creation date	November 24, 2024 at 2:28:01 PM GMT+7	
Status	Enabled	

+ ADD SECRET

Google

Enabled

Enable Sign in with Google

Enables Sign in with Google on the web using OAuth or One Tap, or in Android apps or Chrome extensions.

Client IDs

810904250415-roqje3bd7c8m7i5g92ad8idr34md86hb.apps.googleusercontent.com

Comma-separated list of client IDs for Web, OAuth, Android apps, One Tap, and Chrome extensions.

Client Secret (for OAuth)

Client Secret to use with the OAuth flow on the web.

Skip nonce checks

Allows ID tokens with any nonce to be accepted, which is less secure. Useful in situations where you don't have access to the nonce used to issue the ID token, such as with iOS.

Callback URL (for OAuth)

https://eqbdcwxzxbndcxvbdsgm.supabase.co/auth/v1/callback

Copy

- c. Mengambil token akses API Google menggunakan Endpoint /auth/callback untuk mendapatkan informasi pengguna menggunakan endpoint user info Google:

```
@router.get("/callback")

async def callback(code: str):

    try:

        async with httpx.AsyncClient() as client:

            token_response = await client.post(

                GOOGLE_TOKEN_URL,

                data={

                    "client_id": settings.GOOGLE_CLIENT_ID,

                    "client_secret":

settings.GOOGLE_CLIENT_SECRET,

                    "redirect_uri": settings.CALLBACK_URL,


                    "grant_type": "authorization_code",

                    "code": code,

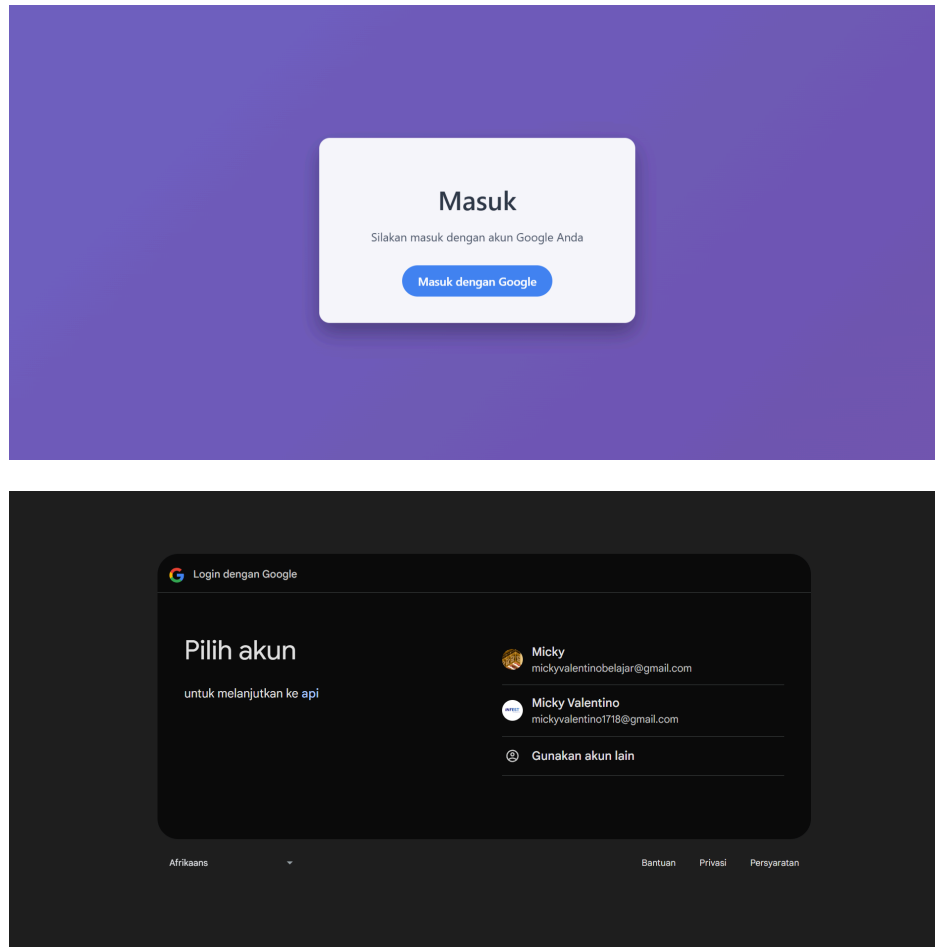
                },

            )
```

- d. Menyimpan data pengguna serta mengarahkan pengguna ke dashboard page

 id	email	full_name	avatar_url	auth_provider
1	mickyvalentino18@gmail.com	Micky Valentino	https://lh3.googleusercontent.com/a/ACg	google
4	18222093@std.stei.itb.ac.id	18222093 Micky Valentino	https://lh3.googleusercontent.com/a/ACg	google

- e. Membuat Frontend untuk mempermudah akses bagi pengguna



2. Pencatatan Pengeluaran

Sistem pencatatan pengeluaran memungkinkan pengguna untuk mencatat, mengkategorikan, dan menyimpan transaksi keuangan di database untuk analisis lebih lanjut. Berikut adalah alur implementasinya:

- a. Membuat Frondend berupa kerangka form untuk memasukkan detail pengeluaran serta memperbaiki tampilan agar mudah digunakan pengguna



- b. Membuat validasi data - data yang dimasukkan dan membuat endpoint untuk mencatat pengeluaran

```
# Save transaction

trans_data = {

    'amount': float(transaction.amount), # Ensure float type

    'date': transaction.date,

    'category': transaction.category,

    'description': transaction.description,

    'user_id': transaction.user_id,

    'created_at': datetime.utcnow().isoformat()

}


# Save to database

result = supabase.table('transactions').insert(trans_data).execute()
```


c. Menyimpan data di Supabase

id	user_id	amount	category	description	date	created_at	
107	1	33000.0	makanan berat	Sate Ayam	2025-01-10	2025-01-10 09:13:11.802964+00	
108	1	35000.0	makanan berat	Roti Tawar	2025-01-10	2025-01-10 09:16:43.359652+00	
109	1	10000.0	makanan berat	Roti Tawar	2025-01-10	2025-01-10 09:17:00.949416+00	
110	1	40000.0	makanan berat	Roti Tawar	2025-01-10	2025-01-10 09:17:16.097504+00	
111	1	40000.0	makanan berat	Sate Ayam	2025-01-10	2025-01-10 09:19:02.581747+00	
112	1	100000.0	makanan berat	Roti Tawar	2025-01-10	2025-01-10 09:19:16.83303+00	
113	1	50000.0	makanan berat	Roti Tawar	2025-01-10	2025-01-10 09:33:05.576594+00	
114	1	200000.0	makanan ringan	Nasi Uduk	2025-01-10	2025-01-10 09:33:26.533666+00	
115	99	250000.0	makanan berat	Nasi Uduk	2025-01-10	2025-01-10 09:34:32.27082+00	
116	99	100000.0	makanan berat	Roti Tawar	2025-01-10	2025-01-10 09:34:47.080727+00	

d. Menampilkan riwayat pengeluaran pengguna dengan cara mengambil data dari Supabase

```
response = supabase.table('transactions')\
    .select('*', anomaly_results('*'))\
    .eq('user_id', user_id)\
    .order('created_at', desc=True)\
    .execute()
```

3. AI Anomaly Detection

Fitur ini mendeteksi anomali dalam transaksi pengguna berdasarkan data historis dan data default menggunakan model machine learning dengan pengerjaan sebagai berikut:

- a. Pengambilan data dari data default dan historis yang berada di Supabase

id	int8	user_id	text	amo...	num...	category	text	description	text	date	date	created_at	timestampz	
107		1		33000.0		makanan berat		Sate Ayam		2025-01-10		2025-01-10 09:13:11.802964+00		
108		1		35000.0		makanan berat		Roti Tawar		2025-01-10		2025-01-10 09:16:43.369652+00		
109		1		10000.0		makanan berat		Roti Tawar		2025-01-10		2025-01-10 09:17:00.949416+00		
110		1		40000.0		makanan berat		Roti Tawar		2025-01-10		2025-01-10 09:17:16.097504+00		

id	uuid	amount	numeric	date	date	category	varchar	description	text	created_at	timestampz	
	0175bd8d-f03e-47	40000.00		2025-01-07		lainnya		Perbaikan Kunci		2025-01-07 02:50:00+00		
	0255fc39-ecbd-43	160000.00		2024-11-05		PDAM		Tagihan PDAM November		2024-11-05 04:15:00+00		
	04b2d2ab-7026-41	70000.00		2024-07-30		kuota		Paket Roaming		2024-07-30 08:10:00+00		
	05f6d72e-804c-41	30000.00		2025-01-02		lainnya		Pasang Stiker		2025-01-02 03:50:00+00		
	07c79ca0-222e-4f	20000.00		2025-01-01		transportasi		Angkot		2024-12-31 23:30:00+00		

- b. Melakukan preprocessing data dengan menghitung statistik data dan melakukan feature engineering
- c. Melakukan pelatihan model (menggunakan model Isolation Forest) terhadap data historis dan data default
- d. Menerapkan model ke data pengeluaran yang berasal dari input pengguna
- e. Melakukan analisis terhadap pengeluaran
- f. Membuat Frontend untuk menampilkan hasil deteksi pengeluaran

Pengeluaran Baru

Jumlah

Kategori

Pilih Kategori

Deskripsi

Tanggal

Kirim Pengeluaran

Hasil Deteksi

Anomali Terdeteksi!

Analisis Jumlah: Pengeluaran sangat tinggi dibanding rata-rata kategori yang sama (normal: Rp 44.000 ± Rp 27.574)

Timing: Transaksi dilakukan pada hari Jumat, tanggal 10

Kategori: makanan berat

* Harga normal berdasarkan rata-rata transaksi dalam kategori yang sama

Histori Pengeluaran

TANGGAL	JUMLAH	KATEGORI	DESKRIPSI	STATUS
10 Januari 2025	Rp 100.000,00	Makanan Berat	Sate Ayam	Anomali

4. Pemanggilan Service Orang Lain

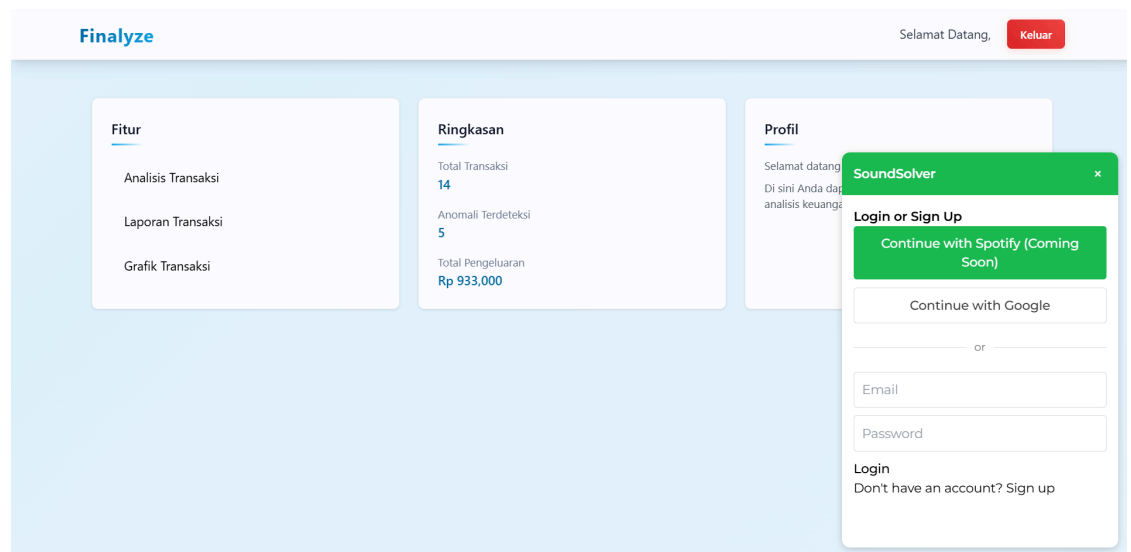
Pada service saya, saya memanggil service berupa chatbot yang dapat menjawab pertanyaan - pertanyaan yang diajukan sehingga dapat memudahkan pengguna jika mengalami kesulitan. Berikut langkah - langkah implementasi pemanggilan service orang:

- Meminta agar service saya di allow untuk memanggil service dengan mengirimkan link service saya
- Menggunakan script injection untuk memanggil service teman saya

```
window.MUSICMATE_API_KEY=
"mk_T4HJ7eDFkynr2vVBWdLtHc6i2WIoc36gghfGB0D0hDc ";

<script
src="https://spotify-bot.azurewebsites.net/static/js/widget-loader.js"></script>
```

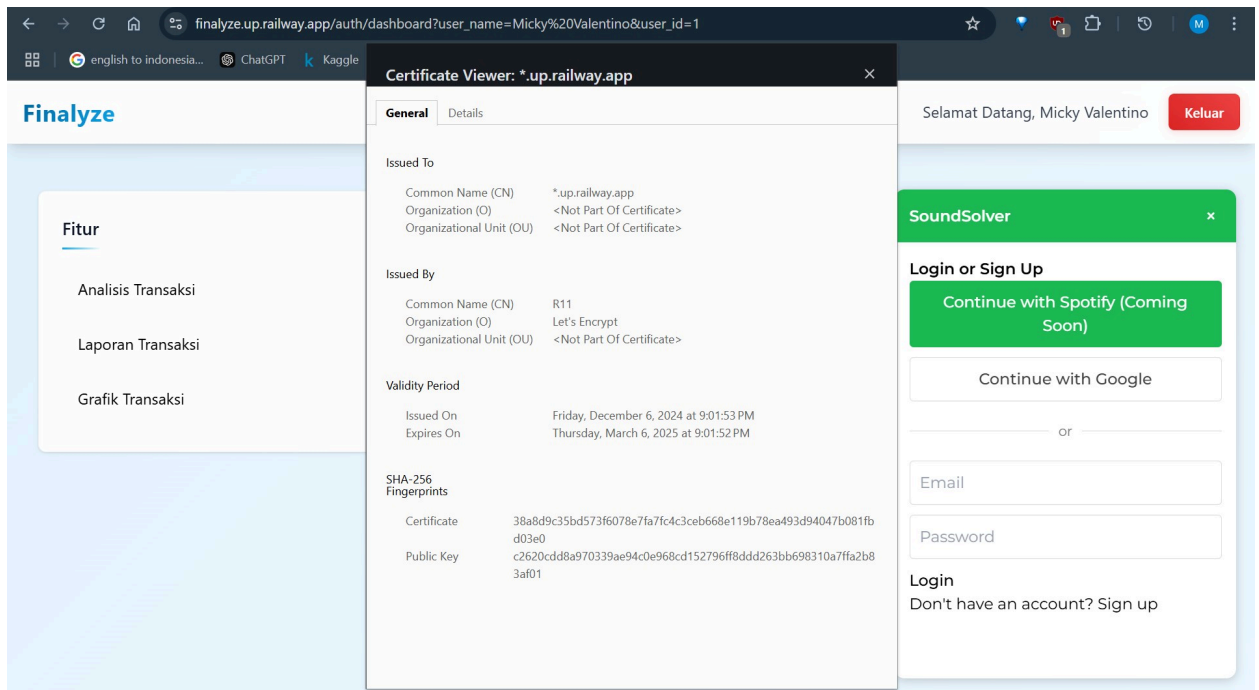
- Hasil Pemanggilan Service:



Implementasi Spesifikasi

1. Penggunaan certificate HTTPS untuk website

Saya menggunakan railway untuk deploy website sehingga sudah disediakan certificate



2. Penggunaan Docker

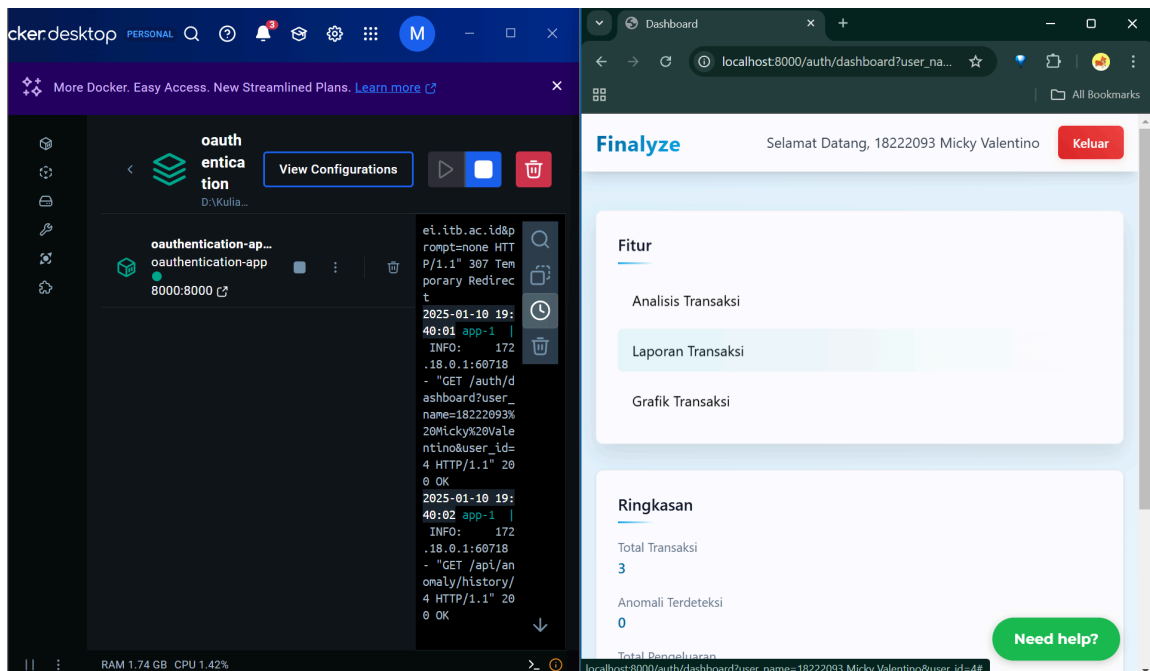
Saya menggunakan docker untuk mengembangkan, mengirimkan, dan menjalankan aplikasi dalam lingkungan yang terisolasi.



```

PS D:\Kuliah\Sem 5\Teknologi Sistem Terintegrasi\FastAPI\OAuthentication> docker-compose up
time="2025-01-10T19:38:57+07:00" level=warning msg="D:\Kuliah\Sem 5\Teknologi Sistem Terintegrasi\FastAPI\OAuthentication\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 1/1
  ✓ container oauthentication-app-1 Recreated                                0.25s
Attaching to app-1
app-1 | INFO: Started server process [1]
app-1 | INFO: Waiting for application startup.
app-1 | INFO: Application startup complete.
app-1 | INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
app-1 | INFO: 172.18.0.1:34384 - "GET /login HTTP/1.1" 404 Not Found
app-1 | INFO: 172.18.0.1:34384 - "GET /favicon.ico HTTP/1.1" 404 Not Found
app-1 | INFO: 172.18.0.1:34384 - "GET / HTTP/1.1" 200 OK
app-1 | INFO: 172.18.0.1:34384 - "GET /auth/login-page HTTP/1.1" 200 OK
app-1 | INFO: 172.18.0.1:34384 - "GET /auth/login HTTP/1.1" 307 Temporary Redirect
app-1 | INFO: 172.18.0.1:60718 - "GET /auth/callback?code=4%2F0AanRRrtMdFFNfcF92dh33a90-3zGjcnwqnNdI9cgmmkQv0IGjbkhNrZ-blICqv1JvOXw&scope=email+profile+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.profile+openid&authuser=0&hd=std.stei.itb.ac.id&prompt=none HTTP/1.1" 307 Temporary Redirect
app-1 | INFO: 172.18.0.1:60718 - "GET /auth/dashboard?user_name=18222093%20Micky%20Valentino&user_id=4 HTTP/1.1" 200 OK
app-1 | INFO: 172.18.0.1:60718 - "GET /api/anomaly/history/4 HTTP/1.1" 200 OK

```



3. Penggunaan Service Finalyze

Saya sudah menambahkan CORS agar service lain dapat menggunakan service saya. Untuk menggunakan service Finalyze, service tersebut dapat mengontak pemilik service Finalyze dan mengirimkan link service untuk diberikan izin. Lalu service tersebut dapat membaca API Documentation yang terdapat pada lampiran untuk melihat endpoint yang ingin dituju dan membuat code yang mengarah ke path endpoint service Finalyze.

Lampiran

Link Github : <https://github.com/MickyV18/Finalyze>

Link Website : <https://finalyze.up.railway.app/>

Link API Documentations : <https://finalyze.up.railway.app/docs>