

# Dokumentasi Teknis

## 1. Pendahuluan

Skrip python `generate_insight` bertujuan untuk menghasilkan narasi wawasan otomatis dari data dengan berbagai jenis grafik menggunakan Azure OpenAI API. Sistem ini membantu dalam menganalisis data dari grafik dan menyajikan informasi tersebut dalam bentuk naratif yang mudah dipahami.

## 2. Diagram Alur (Flowchart)

Diagram alur berikut menggambarkan proses kerja dari skrip ini:



### 3. Mock-up

Berikut adalah tampilan antarmuka pengguna yang diharapkan untuk skrip ini:

#### Tampilan Konsol

Ketika skrip dijalankan, pengguna akan melihat output insight di terminal seperti ini:

```
PS C:\Users\IYOM\tetris> python generate_insight_bar chart.py
njualan berdasarkan kategori produk.
```

Dari grafik bar chart yang dibuat, kita dapat melihat bahwa kategori produk "Software Development Tools" memiliki penjualan tertinggi dengan jumlah 3283 unit, sementara kategori produk "Data Analysis & Management" memiliki penjualan terendah dengan hanya 526 unit. Perbedaan yang signifikan ini menunjukkan adanya kesenjangan yang cukup besar antara kategori produk tersebut.

Dalam perbandingan antar kategori produk, kita dapat melihat bahwa "Software Development Tools" mendominasi penjualan dengan jumlah yang jauh lebih tinggi dibandingkan dengan kategori produk lainnya. Hal ini menunjukkan bahwa permintaan pasar untuk alat pengembangan perangkat lunak sangat tinggi, sementara kategori "Data Analysis & Management" memiliki permintaan yang lebih rendah.

Adanya variasi dalam penjualan antara kategori produk dapat disebabkan oleh beberapa faktor. Salah satunya adalah adopsi teknologi di berbagai industri. Kategori produk "Software Development Tools" mungkin mengalami peningkatan penjualan karena permintaan yang tinggi untuk pengembangan aplikasi dan perangkat lunak di berbagai sektor industri. Di sisi lain, kategori "Data Analysis & Management" mungkin mengalami penurunan penjualan karena kurangnya fokus pada analisis data di beberapa pasar.

Berdasarkan temuan ini, rekomendasi yang dapat diberikan adalah untuk lebih memperkuat pemasaran dan penawaran produk dalam kategori "Data Analysis & Management" dengan mengidentifikasi pasar potensial yang mungkin belum terjangkau. Selain itu, untuk kategori "Software Development Tools", perlu untuk terus mempertahankan kualitas produk dan berinovasi agar tetap memenuhi kebutuhan pengembang perangkat lunak yang terus berkembang.

Dengan memahami dinamika penjualan berdasarkan kategori produk, perusahaan dapat mengarahkan strategi pemasaran dan pengembangan produk secara lebih efektif untuk memaksimalkan potensi penjualan dan memenuhi kebutuhan pasar dengan lebih baik.

### 4. Penjelasan Kode/Skrip

#### Impor Pustaka

```
import pandas as pd
import json
import os
from dotenv import load_dotenv
from openai import AzureOpenAI
import matplotlib.pyplot as plt
```

Pustaka-pustaka ini digunakan untuk manipulasi data, interaksi dengan API, dan visualisasi data.

## Fungsi untuk Menghasilkan Narasi dari AI

```
def menghasilkan_narasi(prompt_sistem, prompt_pengguna):  
    pesan = [  
        {"role": "system", "content": prompt_sistem},  
        {"role": "user", "content": prompt_pengguna}  
    ]  
    respons = client.chat.completions.create(  
        model="gpt-35-turbo",  
        messages=pesan,  
        temperature=0.7,  
        max_tokens=800,  
        top_p=0.95,  
        frequency_penalty=0,  
        presence_penalty=0,  
        stop=None  
    )  
    return respons.choices[0].message.content
```

Fungsi ini mengambil prompt dari sistem dan pengguna, kemudian mengirimnya ke API Azure OpenAI untuk mendapatkan narasi yang dihasilkan

## Memuat Variabel Lingkungan

```
load_dotenv()  
  
# Konfigurasi untuk Azure OpenAI API dari file .env  
azure_endpoint = os.getenv('AZURE_ENDPOINT')  
api_key = os.getenv('API_KEY')  
api_version = os.getenv('API_VERSION')
```

Bagian ini memuat variabel lingkungan dari file `.env` yang berisi konfigurasi untuk API Azure OpenAI.

## Menginisialisasi Klien Azure OpenAI

```
client = AzureOpenAI(  
    azure_endpoint=azure_endpoint,  
    api_key=api_key,  
    api_version=api_version  
)
```

Inisialisasi klien Azure OpenAI menggunakan konfigurasi yang telah dimuat.

## Fungsi Utama untuk Membaca Data, Memanggil API, dan Menghasilkan Wawasan

```
def generate_insight(chart_path):  
    data = read_chart_data(chart_path)  
    prompt_sistem = "Berikan wawasan berdasarkan data berikut:"  
    prompt_pengguna = json.dumps(data.to_dict())  
    insights = menghasilkan_narasi(prompt_sistem, prompt_pengguna)  
    return insights
```

Fungsi ini membaca data dari diagram, memanggil API untuk menghasilkan wawasan, dan mengembalikan wawasan tersebut. `prompt_sistem` memberikan konteks kepada model AI, sedangkan `prompt_pengguna` mengandung data spesifik yang akan dianalisis.

## Fungsi Membaca Data dari Diagram atau Chart

```
def read_chart_data(chart_path):  
    # Implementasi membaca data dari file diagram batang  
    pass
```

Placeholder fungsi untuk membaca data dari file diagram batang.

## Menampilkan atau Menyimpan Hasil

```
if __name__ == '__main__':  
    chart_path = 'path/to/chart.png'  
    insights = generate_insight(chart_path)  
    print(insights)
```

## 5. Petunjuk Penggunaan dan Instalasi

Untuk menggunakan skrip ini, ikuti langkah-langkah berikut:

1. Pastikan Anda telah menginstal Python di sistem Anda.
2. Instal pustaka yang diperlukan menggunakan pip

```
pip install pandas python-dotenv openai plotly
```

3. Buat file `.env` dan tambahkan konfigurasi berikut:

```
AZURE_ENDPOINT=your_azure_endpoint  
API_KEY=your_api_key  
API_VERSION=your_api_version
```

4. Jalankan skrip dengan path ke diagram atau chart yang ingin dibuat:

```
python generate_insight_barchart.py
```

5. Wawasan yang dihasilkan akan ditampilkan di terminal atau disimpan ke file seperti yang ditentukan dalam skrip.