

Format Laporan Proyek Akhir

Judul Proyek

- **Nama proyek:** Figure Mortis
- **Mahasiswa:** Muhamad Fadhil Mauladhani - 221232008
- **Deskripsi proyek:** Sebuah aplikasi Android yang menampilkan daftar tokoh sejarah yang meninggal pada hari ini atau tanggal tertentu. Aplikasi ini mengambil data awal dari Muffinlabs History API (<https://history.muffinlabs.com>) dan kemudian menggunakan AI Generatif (Google Gemini) dengan "Grounding Search" untuk memberikan biografi yang terverifikasi dan mendetail untuk setiap tokoh. Kemudian pengguna bisa menyimpan biografi tokoh sejarah tersebut untuk dilihat kembali di lain waktu.

Tujuan Proyek

- Mengembangkan aplikasi mobile Android menggunakan bahasa pemrograman Java yang dapat mengambil dan menampilkan data dari API eksternal.
- Mengintegrasikan model AI generatif untuk memperkaya konten dan memberikan informasi mendalam kepada pengguna.
- Menerapkan antarmuka pengguna yang bersih, intuitif, dan menarik.
- Integrasi API, penanganan data JSON, dan pemanfaatan teknologi AI modern.

Fitur Proyek

- **Daftar Tokoh Harian:**
 - Aplikasi secara otomatis mendeteksi tanggal saat ini dan mengambil daftar tokoh yang meninggal pada tanggal tersebut dari API history.muffinlabs.com.
 - Daftar nama dan tahun kematian ditampilkan di halaman utama.
- **Tampilan Detail Biografi:**
 - Ketika pengguna memilih satu nama dari daftar, aplikasi akan menampilkan halaman detail.
 - Halaman ini akan berisi biografi mendetail tentang tokoh tersebut, yang dihasilkan oleh Google Gemini. Informasi ini mencakup nama lengkap, tanggal

lahir (jika ada), biografi tiga paragraf, dan daftar sumber yang dapat diverifikasi.

- **Menyimpan Detail Biografi:**
 - Pengguna bisa menyimpan detail biografi tokoh sejarah yang disimpan pada database lokal sehingga bisa diakses kembali.
- **Indikator Pemuatan (Loading):**
 - Menampilkan animasi atau indikator pemuatan saat aplikasi sedang mengambil data dari History API dan saat menunggu respons dari Gemini API untuk memastikan pengguna tahu proses sedang berjalan.

Teknologi yang Digunakan

- **Bahasa pemrograman:** Java
- **Framework:** Android SDK
- **Database:** SQLite.
- **Lain-lain:**
 - **Muffinlabs History API:** Untuk mendapatkan daftar awal tokoh sejarah.
 - **Google AI (Gemini API):** Untuk menghasilkan biografi mendetail dengan sumber.
 - **Large Language Model:** Menggunakan Gemini 2.0 Flash yang mampu melakukan grounding search.
 - **Retrofit/Volley:** Library untuk menangani permintaan jaringan (API calls).
 - **Gson:** Library untuk mem-parsing data JSON.
 - **Google AI Java Client Library:** Untuk berinteraksi dengan Gemini API.

Desain Aplikasi

- **Desain antarmuka pengguna (UI):**
 - **Halaman Utama:** Menggunakan RecyclerView untuk menampilkan daftar nama tokoh. Setiap item akan menggunakan CardView untuk tampilan yang modern dan rapi. Judul halaman akan menampilkan tanggal saat ini.
 - **Halaman Detail:** Menggunakan ScrollView yang berisi beberapa TextView

untuk menampilkan nama, tanggal lahir, biografi, dan daftar sumber. Desain akan fokus pada keterbacaan.

- **Halaman Saved Figures:** Menggunakan RecyclerView untuk menampilkan daftar nama tokoh yang disimpan, tampilannya kurang lebih akan seperti halaman utama.

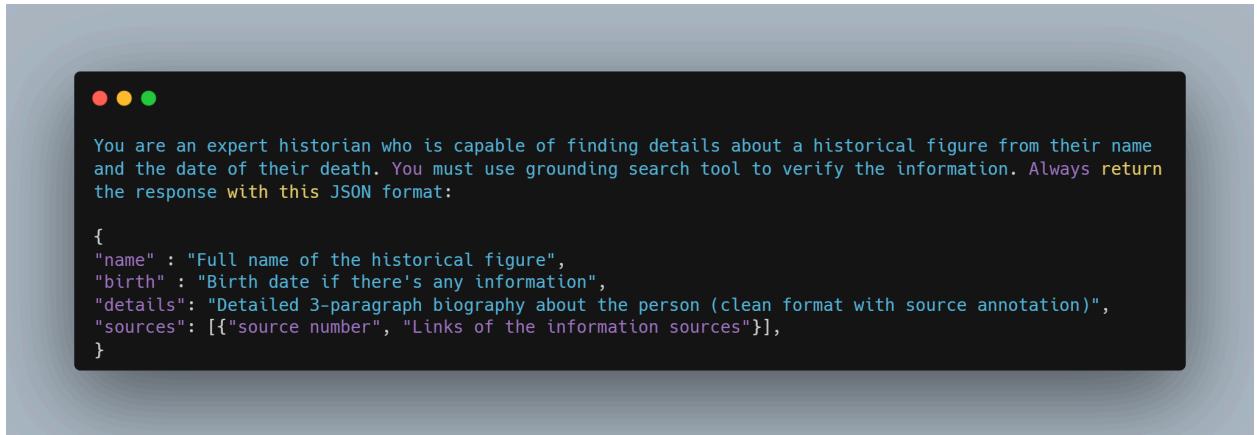
Implementasi (Rencana Langkah-langkah)

1. **Setup Proyek:** Membuat proyek baru di Android Studio.
2. **Konfigurasi Dependensi:** Menambahkan *library* Retrofit/Volley, Gson, dan Google AI ke dalam file build.gradle.
3. **Integrasi History API:**
 - Membuat kelas mode di Java untuk merepresentasikan struktur data JSON dari Muffinlabs History API.
 - Request API akan melakukan hit ke <https://history.muffinlabs.com/date/{m}/{d}>
 - Data yang didapatkan akan berbentuk seperti ini:



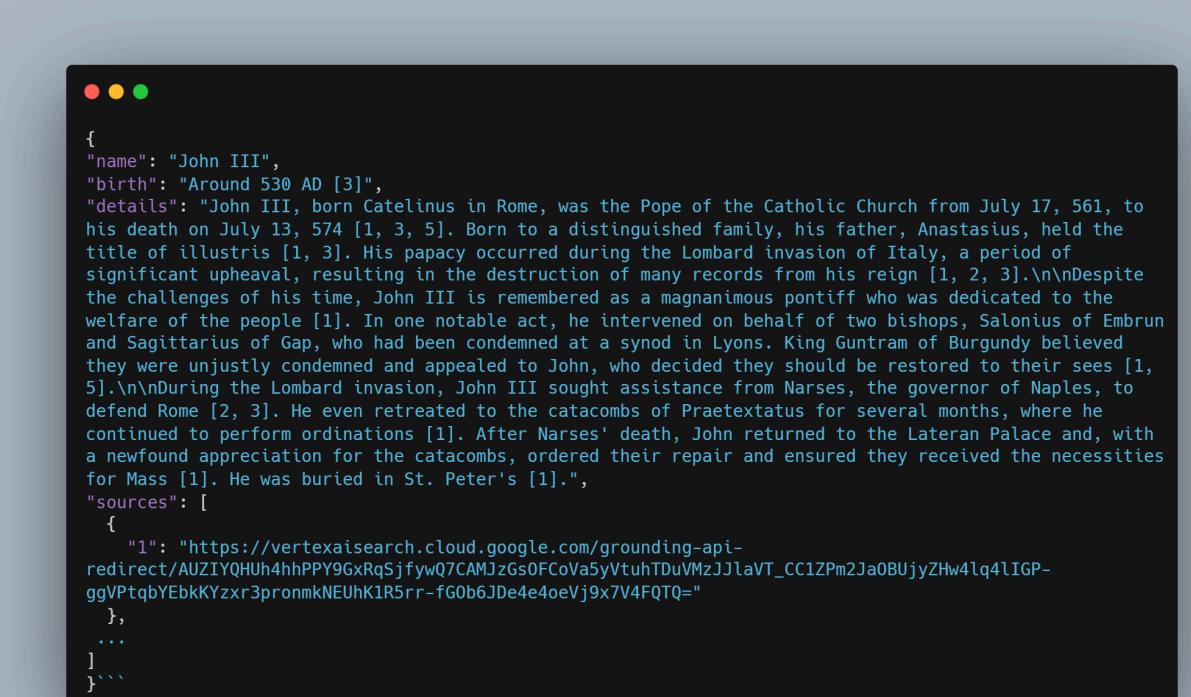
```
{  
    "date": "July 13",  
    "url": "https:// wikipedia.org/wiki/July_13",  
    "data": {  
        "Events": [  
            {}  
        ],  
        "Births": [  
            {}  
        ],  
        "Deaths": [  
            {  
                "year": "574",  
                "text": "John III, pope of the Catholic Church",  
                "html": "574 - <a href=\"https:// wikipedia.org/wiki/Pope_John_III\" title=\"Pope John III\">John III</a>, pope of the Catholic Church",  
                "no_year_html": "<a href=\"https:// wikipedia.org/wiki/Pope_John_III\" title=\"Pope John III\">John III</a>, pope of the Catholic Church",  
                "links": [  
                    {  
                        "title": "Pope John III",  
                        "link": "https:// wikipedia.org/wiki/Pope_John_III"  
                    }  
                ]  
            },  
        ]  
    }  
}
```

4. **UI Halaman Utama:** Mendesain layout XML dan membuat Adapter untuk RecyclerView.
5. **Integrasi Gemini API:**
 - Membuat prompt Generative AI yang sesuai dengan tugasnya, berikut adalah rancangan prompt nya:



Model AI diinstruksikan untuk selalu menggunakan “grounding search” dengan Google Search untuk mencari informasi terkait tokoh yang ditanyakan sehingga bisa menghindari halusinasi dan mampu memberikan sumber yang dikutipnya. Format data yang dikirimkan diperintahkan untuk berbentuk JSON.

- Membuat fungsi untuk mengirim *prompt* (nama tokoh dan tahun kematian) ke Gemini dan mem-parsing respons JSON yang diterima. Berikut contoh prompt yang diberikan : “Who was John III, pope of the Catholic Church that was deceased in 574?”
- Hasil response JSON akan dirender sebagus mungkin dengan mempertimbangkan annotasi atau label sumber jawaban sehingga pengguna bisa diarahkan untuk memverifikasi sendiri biografi yang dibacanya. Berikut contoh respons Generatif AI yang akan di parsing:



```
{  
  "name": "John III",  
  "birth": "Around 530 AD [3]",  
  "details": "John III, born Catelinus in Rome, was the Pope of the Catholic Church from July 17, 561, to his death on July 13, 574 [1, 3, 5]. Born to a distinguished family, his father, Anastasius, held the title of illustris [1, 3]. His papacy occurred during the Lombard invasion of Italy, a period of significant upheaval, resulting in the destruction of many records from his reign [1, 2, 3].\n\nDespite the challenges of his time, John III is remembered as a magnanimous pontiff who was dedicated to the welfare of the people [1]. In one notable act, he intervened on behalf of two bishops, Salonius of Embrun and Sagittarius of Gap, who had been condemned at a synod in Lyons. King Guntram of Burgundy believed they were unjustly condemned and appealed to John, who decided they should be restored to their sees [1, 5].\n\nDuring the Lombard invasion, John III sought assistance from Narses, the governor of Naples, to defend Rome [2, 3]. He even retreated to the catacombs of Praetextatus for several months, where he continued to perform ordinations [1]. After Narses' death, John returned to the Lateran Palace and, with a newfound appreciation for the catacombs, ordered their repair and ensured they received the necessities for Mass [1]. He was buried in St. Peter's [1].",  
  "sources": [  
    {  
      "1": "https://vertexaisearch.cloud.google.com/grounding-api-  
            redirect/AUZIYQHUh4hhPPY9GxRqSjfywQ7CAMJzGs0FCoVa5yVtuhTduVmJJlaVT_CC1ZPm2Ja0BUjyZHw4lq4lIGP-  
            ggVPtqbYEbkKYzxr3pronmkNEUhK1R5rr-fG0b6JDe4e4oeVj9x7V4FQTQ="  
    },  
    ...  
  ]  
}
```

Pada properti “details” sebisa mungkin setiap pernyataan yang dikeluarkan diberi label sumber [1, 3, 5] untuk memberi tahu pengguna darimana informasinya berasal sehingga meminimalkan halusinasi.

6. **UI Halaman Detail:** Membuat activity dan layout XML baru untuk menampilkan biografi.
 7. **Navigasi:** Mengimplementasikan Intent untuk berpindah dari halaman utama ke halaman detail sambil mengirimkan data tokoh yang dipilih.
- **Desain Database:** Database menggunakan SQLite untuk menyimpan informasi sederhana di perangkat pengguna. Berikut skema yang dirancang:
 - **Nama Tabel:** `saved_figures`
 - **Nama Database:** `mortis.db`
 - **Kolom Tabel:**
 - `id`: INTEGER PRIMARY KEY AUTOINCREMENT
 - `name`: TEXT NOT NULL
 - `birth_date`: TEXT

- `death_year`: TEXT
- `details`: TEXT NOT NULL
- `sources`: TEXT (Akan menyimpan daftar sumber sebagai string JSON)

Pengujian

- **Pengujian Fungsional:** Klik setiap item di daftar untuk memastikan halaman detail terbuka dengan data yang benar. Pastikan indikator pemuatan muncul dan hilang dengan benar. Model Generatif AI mengembalikan respons dalam waktu yang cepat (dibawah 60 detik).
- **Pengujian UI:** Uji pada berbagai ukuran layar (emulator) untuk memastikan tidak ada elemen UI yang tumpang tindih atau terpotong.
- **Pengujian Jaringan:** Uji dalam mode pesawat untuk memastikan penanganan error koneksi berfungsi dan halaman "Saved Figures" bisa diakses.

Kesimpulan (yang Diharapkan)

Proyek ini akan menghasilkan aplikasi "Figure Mortis" yang fungsional dan inovatif. Proyek ini akan menjadi bukti kemampuan dalam mengembangkan aplikasi Android dengan Java, mengintegrasikan layanan eksternal, dan menerapkan teknologi AI canggih untuk menciptakan pengalaman pengguna yang kaya informasi dan menarik.

Saran:

- Menggunakan model AI dengan konteks lebih besar dan kemampuan reasoning, sehingga lebih banyak sumber informasi yang bisa dibuat sintesis.
- Membuat layanan API model Generatif AI sendiri untuk memangkas biaya.

Dokumentasi

1. Draft kode Gemini Service:



```
package com.example;

import com.google.common.collect.ImmutableList;
import com.google.common.collect.ImmutableMap;
import com.google.genai.Client;
import com.google.genai.ResponseStream;
import com.google.genai.types.*;
import com.google.gson.Gson;

import java.util.ArrayList;
import java.util.Base64;
import java.util.List;
import java.util.Map;

public class App {
    public static void main(String[] args) {
        String apiKey = System.getenv("GEMINI_API_KEY");
        Client client = Client.builder().apiKey(apiKey).build();
        Gson gson = new Gson();

        String model = "gemini-2.0-flash";
        List<Content> contents = ImmutableList.of(
            Content.builder()
                .role("user")
                .parts(ImmutableList.of(
                    Part.fromText("Who was John III, pope of the Catholic Church that was deceased in 574")
                ))
                .build(),
        );
        GenerateContentConfig config =
            GenerateContentConfig
                .builder()
                .temperature(.0f)
                .responseMimeType("text/plain")
                .systemInstruction(
                    Content
                        .fromParts(
                            Part.fromText("You are an expert historian who is capable of finding details about a historical figure from their name and the date of their death. You must use grounding search tool to verify the information. Always return the response with this JSON format:\n\n\"name\" : \"Full name of the historical figure\", \"birth\" : \"Birth date if there's any information\", \"details\" : \"Detailed 3-paragraph biography about the person (clean format with source annotation)\", \"sources\" : [ {\"source number\", \"Links of the information sources\"} ], \"links\" : [ {\"url\", \"description\"} ]\n")
                        )
                )
                .build();
        ResponseStream<GenerateContentResponse> responseStream = client.models.generateContentStream(model,
            contents,
            config);

        for (GenerateContentResponse res : responseStream) {
            if (res.candidates().isEmpty() || res.candidates().get(0).content().get().parts().isEmpty() || res.candidates().get(0).content().get().parts().isEmpty()) {
                continue;
            }

            List<Part> parts = res.candidates().get(0).content().get().parts().get();
            for (Part part : parts) {
                System.out.println(part.text());
            }
        }

        responseStream.close();
    }
}
```

2. Draft pallet warna aplikasi

Menggunakan <https://material-foundation.github.io/material-theme-builder/> untuk menentukan pallet yang cocok. Aplikasi mengambil tema gelap / vampir / gothic yang cocok diasosiasikan dengan "Mortis" atau kematian. Font yang digunakan adalah Cinzel dan Crimson Text.



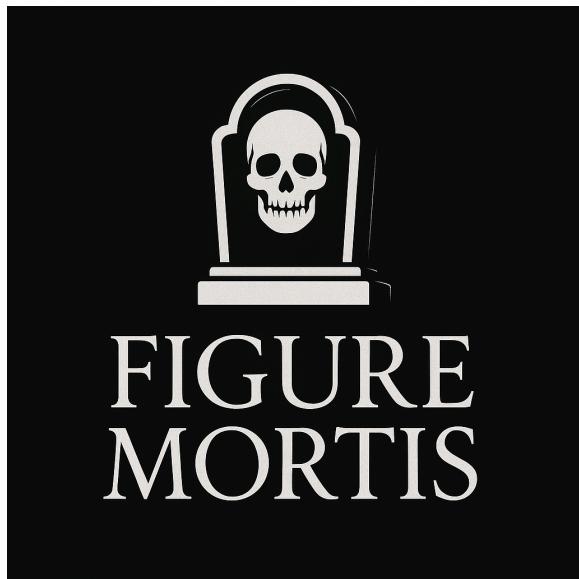
Dark Scheme			
Primary	Secondary	Tertiary	Error
On Primary	On Secondary	On Tertiary	On Error
Primary Container	Secondary Container	Tertiary Container	Error Container
On Primary Container	On Secondary Container	On Tertiary Container	On Error Container
Surface Dim	Surface	Surface Bright	Inverse Surface
Surf. Container Lowest	Surf. Container Low	Surf. Container	Surf. Container High
Surf. Container Highest			
On Surface	On Surface Var.	Outline	Outline Variant
			Scrim
			Shadow

Warna dasar diambil dari artwork game Warhammer oleh Markus Lenz:



3. Draft logo, maskot dan ikon

Logo mengikuti tema yang sama yaitu seputar “Mortis”, logo utama adalah “Gravestone” yang merepresentasikan kematian atau tempat peristirahatan seseorang.



Maskot mengambil tema kerangka hidup untuk mengisi desain halaman loading dan halaman kosong. Mengambil referensi dari artwork yang sama dengan color palette.

