**Richard Rafer Guy – 222117056**

**Plan And Solve (PS) Prompting**

1. **Pendahuluan**

**Plan-and-Solve (PS) Prompting** adalah metode yang memanfaatkan zero-shot Chain-of-Thought (CoT) yang memungkinkan LLM menyusun rencana secara eksplisit untuk menyelesaikan suatu masalah, lalu menghasilkan proses penalaran sebelum menentukan jawaban akhir. Berbeda dengan pendekatan few-shot CoT yang membutuhkan contoh langkah-langkah dalam prompt, PS prompting hanya memerlukan pernyataan masalah dan kalimat pemicu sederhana. Metode ini terdiri dari dua langkah: (1) LLM menyusun rencana dan menyelesaikan masalah melalui penalaran bertahap; (2) LLM mengekstrak jawaban akhir dengan instruksi khusus, seperti “Jadi, jawabannya (angka) adalah”.

1. **Penjelasan Dataset yang Digunakan**

Pada percobaan yang dilakukan, terdapat 3 dataset yang digunakan. Berikut adalah penjelasan mengenai 3 dataset (dari paper utama teknik ToT, untuk dataset yang telah menggunakan format JSON terdapat pada github paper utama) tersebut.

1. **GSM8K** : merupakan kumpulan data (dataset) yang berisi soal cerita matematika tingkat dasar. Dataset ini dirancang khusus untuk mengukur atau pelatihan kemampuan LLM dalam menyelesaikan masalah matematika yang memerlukan penalaran multi-langkah. Jumlah data yang digunakan dalam percobaan ini sebanyak 45 data. Dataset digunakan pada eksekusi program adalah dataset GSM8K yang telah disusun dengan format JSON.
2. **AQuA** : adalah dataset yang berfokus pada soal-soal aljabar. Berbeda dengan GSM8K yang fokus pada aritmatika dasar, AQuA menyajikan tantangan yang lebih kompleks karena melibatkan konsep aljabar. Jumlah data yang digunakan pada percobaan adalah 45 data, dengan format JSON yang akan digunakan pada proses eksekusi program.
3. **Symbolic Reasoning (Last Letter)** : adalah dataset tentang manipulasi teks berdasarkan aturan spesifik. Soal atau input dari dataset ini akan berupa daftar kalimat atau kata. Sedangkan jawaban dari soal tersebut adalah gabungan dari posisi terakhir kata tersebut. Data yang digunakan pada percobaan ini adalah 45 data, yang telah disusun dengan format JSON untuk dieksekusi oleh program.
4. **Penjelasan Library yang Digunakan**

Berikut adalah daftar library yang digunakan dalam kode program, beserta penjelasan fungsi/kegunaannya masing-masing:

1. **os** :digunakan untuk berinteraksi dengan sistem operasi (misalnya mendapatkan value dari env variable).
2. **google.generativeai** :digunakan untuk mengatur API key, inisialisasi model, dan mengirim prompt serta menerima hasil dari model.
3. **python-dotenv** :merupakan library untuk membaca dan memuat isi dari file .env.
4. **json** :digunakan untuk membaca dan menyimpan data dalam format JSON.
5. **argparse** :digunakan untuk membaca argumen yang diberikan melalui command-line saat mengeksekusi program.
6. **re** :library ini digunakan untuk pemrosesan regex, untuk ekstraksi atau pembersihan teks.
7. **time** :library ini membantu memberikan delay proses, untuk menghindari rate limit API.
8. **pandas :** digunakan untuk membaca dataset dengan format CSV.
9. **Penjelasan Kode Program**

Bagian ini akan membahas program yang digunakan, cara kerja, hingga peran masing-masing komponen yang mereferensi ke dokumentasi kode yang telah dilakukan :

1. File **utils/gemini\_client.py,** digunakan untuk menyiapkan konfigurasi gemini client dengan API Key yang terdapat pada .env, dan juga menyiapkan function untuk mengirim promp dan mendapatkan hasil response dari LLM. Selain itu **utils/io\_utils.py**, sebagai penyedia function untuk load & save data dalam format JSON.
2. File **main.py** berfungsi sebagai titik awal program. Program ini menerima beberapa argument seperti, --technique (untuk memilih teknik yang ingin digunakan), --output (untuk memilih file penyimpanan hasil output pemrosesan), --dataset (untuk memilih dataset yang akan digunakan).
3. Function **main** **(main.py – Line 267-277),** merupakan function yang pertama kali dipanggil ketika main.py dieksekusi. Function ini akan mendapatkan argument pada command-line, dan memproses tugas tertentu berdasarkan dataset yang dipilih.
4. **Mekanisme Perhitungan Performansi**