Laporan Membuat AI Agent di Terminal yang Dapat Melakukan Aktifitas Sesuai yang Diperintahkan

Dosen Pengampu : ***Liptia Venica, S.T., M.T.***

Di Ajukan Untuk Memenuhi Tugas Mata Kuliah Machine Learning

Disusun Oleh :

Abdul Rafi (2300062)

Program Studi Mekatronika & Kecerdasan Buatan

Universitas Pendidikan Indonesia

Kampus Purwakarta

2025

1. Latar Belakang & Konsep

Laporan ini menjelaskan implementasi **AI Agent berbasis Terminal (CLI)** yang mampu menginterpretasikan perintah bahasa natural dari pengguna dan secara cerdas menjalankan fungsi Python kustom (disebut *Custom Tools* atau *Function Calling*) untuk berinteraksi dengan sistem operasi dan data.

- **Masalah:** Developer, System Administrator, atau pengguna biasa sering membutuhkan alat yang cepat untuk melakukan tugas-tugas administratif rutin—seperti merapikan folder, mengecek status sistem, atau mencari data pengguna—tanpa harus menulis skrip atau menjalankan *query* yang rumit.
- **Solusi:** Membuat AI Agent berbasis Terminal yang menghubungkan Model Bahasa Besar (LLM, yaitu Grok-4-Fast melalui OpenRouter API) dengan *Custom Tools* yang ditulis dalam Python. Hal ini memungkinkan pengguna untuk memerintah sistem menggunakan bahasa sehari-hari.
- **Target Pengguna:** Developer, System Administrator, dan pengguna yang ingin meningkatkan efisiensi kerja dengan otomatisasi berbasis bahasa alami.

2. Deskripsi "Custom Tool" Unik

A. Tool 1: smart_file_organizer (File Organizer)

Detail	Deskripsi
Nama Tool	smart_file_organizer
Fungsi Utama	Memindai folder target, memfilter file berdasarkan ekstensi, dan memindahkan file-file yang cocok ke subfolder baru yang namanya sesuai dengan ekstensi file tersebut
Parameter (Input)	path (string): Lokasi folder target (e.g., 'D:/Download/project'). extension (string): Jenis file yang dicari (e.g., 'pdf', 'jpg', 'py').
Logika Singkat	Script menggunakan modul os dan shutil untuk memindai direktori, memfilter ekstensi, menentukan nama folder tujuan (extension.upper()), membuat direktori tersebut jika belum ada, dan memindahkan file.

B. Tool 2: system_check (System Check)

Detail	Deskripsi
Nama Tool	system_check
Fungsi Utama	Mengambil dan mengembalikan status kinerja sistem komputer secara real-time, termasuk penggunaan CPU dan ketersediaan RAM.
Parameter (Input)	Tidak ada parameter.
Logika Singkat	Menggunakan library psutil untuk mendapatkan persentase penggunaan CPU, total RAM, dan RAM yang tersedia/digunakan.

C. Tool 3: database_query (Database Query)

Detail	Deskripsi
Nama Tool	database_query
Fungsi Utama	Mensimulasikan pencarian data profil pengguna dari database berdasarkan ID unik (user_id), dan mengembalikan data pengguna jika ditemukan.
Parameter (Input)	user_id (string): ID unik pengguna yang dicari (e.g., 'USR100').
Logika Singkat	Mencari ID yang diberikan di dalam dictionary Python yang bertindak sebagai database simulasi, kemudian mengembalikan data yang cocok dalam format string JSON.

3. Implementasi Interface (Terminal)eskripsi

Interface Terminal (CLI) dibuat interaktif dan mudah dibaca menggunakan *library* **colorama** untuk *styling* output, sesuai persyaratan tugas.

1. **Input User:** Hijau (Fore.GREEN)
2. **Proses Thinking:** Kuning (Fore.YELLOW)
3. **Output Agent (Final Answer):** Cyan (Fore.CYAN)
4. **Tool Call/Output Log:** Biru (Fore.BLUE)

Snippet Kode Utama (Interface & Loop)

Berikut adalah bagian kode yang bertanggung jawab untuk *styling* dan menjalankan loop interaktif Agent, termasuk eksekusi *Tool Call* secara dinamis:

```
# Warna Sesuai Tugas (Hijau untuk User, Cyan untuk Output Agent)
COLOR_USER = Fore.GREEN
COLOR_AGENT = Fore.CYAN
COLOR_THINKING = Fore.YELLOW
COLOR_TOOL = Fore.BLUE
```

```
while True:
    try:
        user_input = input(f"{COLOR_USER}You: {Style.RESET_ALL}")

        if user_input.lower() in ['q', 'quit', 'exit']:
            print(f"{COLOR_AGENT}Goodbye!")
            break
        agent.add_message("user", user_input)
        print(f"{COLOR_THINKING}[AI Thinking]: Memproses permintaan...{Style.RESET_ALL}")
        while True:
            response, tool_calls = agent.get_response()
            if len(tool_calls) > 0:
                tool = tool_calls[0]
                tool_name = tool['name']
                tool_args = tool['arguments']
                tool_call_id = tool['tool_call_id']
                print(f"\n{COLOR_TOOL}[Tool Call]: Memanggil {tool_name} dengan argumen: {tool_args}{Style.RESET_ALL}")
                try:
                    result = tool_dictionary[tool_name](**tool_args)
                    print(f"{COLOR_TOOL}[Tool Output]: {result}{Style.RESET_ALL}")
                    agent.add_tool_call(tool_call_id, result)
                except Exception as e:
                    error_message = f"Tool execution failed: {str(e)}"
                    print(f"{COLOR_TOOL}[Tool Output - Error]: {error_message}{Style.RESET_ALL}")
                    agent.add_tool_call(tool_call_id, error_message)
                print(f"{COLOR_THINKING}[AI Thinking]: Menganalisis hasil tool untuk jawaban akhir...{Style.RESET_ALL}")
            else:
                print("\n" + "="*50 + "\n")
                break
```

```
PS D:\ai_agent_project> py main.py
=== AI Admin Agent Terminal Interface (3 Tools Aktif) ===
--------------------------------------------------
Instruksi Uji Coba:
1. System Check: 'cek sisa RAM di laptop saya'
2. File Organizer: 'rapikan file PDF di D:/ai_agent_project/UJI_COBA_AGENT'
3. Database Query: 'cari data untuk user USR100'
--------------------------------------------------
Ketik 'q' untuk keluar.
==================================================
You: cek sia RAM dan CPU saya
[AI Thinking]: Memproses permintaan...
```

4. Bukti Uji Coba (Demonstrasi)

1. Skenario Uji Coba (File Organizer - Setelah Modifikasi)

```
You: rapihkan semua file pdf di folder D:/ai_agent_project/UJI_COBA_AGENT
[AI Thinking]: Memproses permintaan...

[Tool Call]: Memanggil smart_file_organizer dengan argumen: {'path': 'D:/ai_agent_project/UJI_CO
BA_AGENT', 'extension': 'pdf'}
[Tool Output]: Berhasil memindahkan 7 file PDF ke subfolder 'PDF' di 'D:/ai_agent_project/UJI_CO
BA_AGENT'.
[AI Thinking]: Menganalisis hasil tool untuk jawaban akhir...
File PDF di folder `D:/ai_agent_project/UJI_COBA_AGENT` berhasil dirapikan. Terdapat 7 file PDF
yang telah dipindahkan ke subfolder baru bernama `PDF`. Jika ada file tambahan atau instruksi la
in, beri tahu saya!
==================================================
```

```
You: rapihkan semua file jpd di folder D:/ai_agent_project/UJI_COBA_AGENT
[AI Thinking]: Memproses permintaan...

[Tool Call]: Memanggil smart_file_organizer dengan argumen: {'path': 'D:/ai_agent_project/UJI_CO
[Tool Output]: Berhasil memindahkan 6 file JPG ke subfolder 'JPG' di 'D:/ai_agent_project/UJI_CO
BA_AGENT'.
[AI Thinking]: Menganalisis hasil tool untuk jawaban akhir...
Saya telah merapikan file JPG (asumsi dari "jpd" sebagai typo) di folder D:/ai_agent_project/UJI
_COBA_AGENT. Berhasil memindahkan 6 file JPG ke subfolder 'JPG'.

Jika ada file lain yang perlu dirapikan atau ini bukan yang dimaksud, beri tahu saya!
==================================================
```

2. Skenario Uji Coba (Database Query)

```
You: cari data  untuk user USR330
[AI Thinking]: Memproses permintaan...

[Tool Call]: Memanggil database_query dengan argumen: {'user_id': 'USR330'}
[Tool Output]: Data ditemukan untuk User ID USR330: {"name": "Joko Susilo", "status": "Active",
"last_login": "2025-11-21", "role": "Tester"}
[AI Thinking]: Menganalisis hasil tool untuk jawaban akhir...
Data ditemukan untuk User ID USR330:
- **Name**: Joko Susilo
- **Status**: Active
- **Last Login**: 2025-11-21
- **Role**: Tester
==================================================
```

3. scenario uji coba (System Check)

```
You: cek sia RAM dan CPU saya
[AI Thinking]: Memproses permintaan...

[Tool Call]: Memanggil system_check dengan argumen: {}
[Tool Output]: System Status:
- CPU Usage: 7.2%
- RAM Total: 23.89 GB
- RAM Available: 12.1 GB
- RAM Used: 49.3%
```

```
[AI Thinking]: Menganalisis hasil tool untuk jawaban akhir...
Berikut adalah status terbaru sistem Anda berdasarkan pengecekan terkini:

- **Penggunaan CPU**: 4.9%
- **Total RAM**: 23.89 GB
- **RAM Tersedia**: 12.14 GB
- **RAM Digunakan**: 49.2%

Jika Anda ingin pengecekan ulang atau detail lebih lanjut, beri tahu saya!
```

5. Kesimpulan

Implementasi AI Agent ini telah berhasil memenuhi semua persyaratan tugas. Agent berhasil menginterpretasikan permintaan bahasa natural pengguna dan memilih salah satu dari tiga *Custom Tools* unik yang tersedia (File Organizer, System Check, atau Database Query). Mekanisme *Tool-Calling* memungkinkan Agent untuk berinteraksi secara dinamis dengan fungsi Python yang memodifikasi sistem file (memindahkan file) atau mengambil data real-time (status sistem). Hal ini membuktikan bahwa LLM dapat digunakan sebagai *otak* untuk mengendalikan skrip Python untuk otomatisasi tugas-tugas sistem yang kompleks.

Link Git Hub:

https://github.com/Danis1223/AI-Agent-di-Terminal-yang-Dapat-Melakukan-Aktifitas-Sesuai-yang-Diperintahkan.git