Nama : Raffli Islami Fashya

NIM : 24360003

Matkul : Pemrograman Berorientasi Objek untuk sistem AI Agenik

Judul:
Implementasi Sistem AI Agenik Sederhana Berbasis OOP (Python)

Prompt:

buatkan Program sederhana berbasis Pemrograman Berorientasi Objek (OOP) yang menggambarkan sistem AI Agenik,dan tambahkan UML diagram

Deskripsi:
Program ini menggambarkan konsep dasar Sistem AI Agenik menggunakan pendekatan Pemrograman Berorientasi Objek (OOP) dalam Python. Sistem terdiri dari dua kelas utama, yaitu **Lingkungan (Environment)** dan **Agen (Agent)**, di mana Agen berinteraksi dengan Lingkungan untuk mencapai tujuan tertentu.

Kode Program

```
class Lingkungan:
    """Mewakili lingkungan tempat Agen beroperasi."""
    def __init__(self, status_awal="Belum Selesai"):
        self.status = status_awal
        print(f"Lingkungan diinisialisasi dengan status: {self.status}")

    def get_status(self):
        return self.status

    def ubah_status(self, aksi):
        if aksi == "Ubah ke Selesai":
            self.status = "Selesai"
            print("Lingkungan: Status diubah menjadi 'Selesai'.")
        else:
            print(f"Lingkungan: Aksi '{aksi}' tidak mengubah status.")

class Agen:
    """Mewakili agen AI yang berinteraksi dengan Lingkungan."""
    def __init__(self, nama, tujuan):
        self.nama = nama
        self.tujuan = tujuan
        print(f"Agen '{self.nama}' memiliki tujuan: {self.tujuan}")
```

```
            def persepsi(self, lingkungan):
                status_lingkungan = lingkungan.get_status()
                print(f"Agen {self.nama}: Mempersepsikan status lingkungan: {status_lingkungan}")
                return status_lingkungan

            def berfikir(self, status_lingkungan):
                if status_lingkungan != self.tujuan:
                    aksi = "Ubah ke Selesai"
                    print(f"Agen {self.nama}: Berfikir untuk melakukan aksi '{aksi}'.")
                    return aksi
                else:
                    aksi = "Tidak melakukan apa-apa"
                    print(f"Agen {self.nama}: Tujuan sudah tercapai. Berfikir untuk melakukan aksi
        '{aksi}'.")
                    return aksi

            def aksi(self, lingkungan):
                status = self.persepsi(lingkungan)
                aksi_yang_dipilih = self.berfikir(status)
                if aksi_yang_dipilih != "Tidak melakukan apa-apa":
                    lingkungan.ubah_status(aksi_yang_dipilih)
                else:
                    print(f"Agen {self.nama}: Tidak ada aksi yang dilakukan.")

        # --- Simulasi Sistem ---
        lingkungan_utama = Lingkungan()
        agen_cerdas = Agen(nama="Bot Cepat", tujuan="Selesai")

        print("\n--- Siklus 1 ---")
        agen_cerdas.aksi(lingkungan_utama)

        print("\n--- Siklus 2 ---")
agen_cerdas.aksi(lingkungan_utama)
```

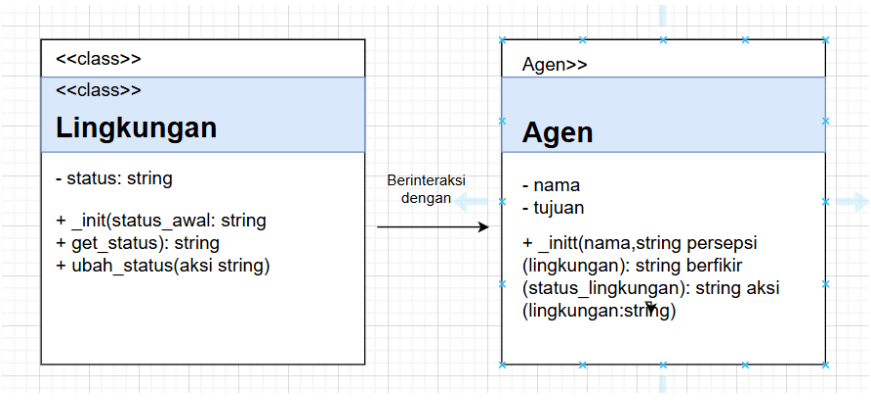
Output Program

```
Lingkungan diinisialisasi dengan status: Belum Selesai
Agen 'Bot Cepat' memiliki tujuan: Selesai

--- Siklus 1 ---
Agen Bot Cepat: Mempersepsikan status lingkungan: Belum Selesai
Agen Bot Cepat: Berfikir untuk melakukan aksi 'Ubah ke Selesai'.
Lingkungan: Status diubah menjadi 'Selesai'.

--- Siklus 2 ---
Agen Bot Cepat: Mempersepsikan status lingkungan: Selesai
Agen Bot Cepat: Tujuan sudah tercapai. Berfikir untuk melakukan aksi 'Tidak melakukan apa-apa'.
Agen Bot Cepat: Tidak ada aksi yang dilakukan.
```

Diagram UML



Penjelasan Tambahan

Sistem ini menerapkan konsep dasar OOP:

- **Encapsulation:** atribut status, nama, dan tujuan bersifat privat dan diakses lewat method.
- **Association:** kelas *Agen* berinteraksi langsung dengan objek *Lingkungan*.
- **Behavioral Loop:** Agen menjalankan siklus *Persepsi* → *Berfikir* → *Aksi* untuk mencapai tujuan.