

AI Agent (<https://github.com/ArifDwiPurwanto/ai-agent.git>):

- **Alur kerja agent dan arsitektur**

Memiliki fase: observe (observasi), decide (keputusan), act (aksi) dan reflect (refleksi)

- fase observe:
 1. Agent menambahkan pesan user ke STM (Short Term Memory)
 2. Mengumpulkan konteks percakapan dari history
 3. Mencari memori yang relevan dari LTM (Long Term Memory)
 4. Menyiapkan memory tools yang tersedia
 5. Menggabungkan semua informasi menjadi Observation object
- fase decide:
 1. Membuat prompt untuk decision-making
 2. Mengirim ke AI Model
 3. Parsing response dari AI Model untuk menentukan action type:
 - a. "respond": memberikan jawaban langsung
 - b. "use_tool": menggunakan tool tertentu
 - c. "store_memory": simpan informasi penting
 - d. "ask_clarification": meminta klarifikasi
 4. Mengembalikan decision object dengan reasoning dan confidence
- fase act:
 1. Melakukan aksi berdasarkan tipe keputusan:
 - a. "respond": generate response menggunakan AI Model
 - b. "use_tool": eksekusi tool yang sesuai
 - c. "store_memory": menyimpan ke LTM (Long Term Memory)
 - d. "ask_clarification": minta input tambahan dari user
- fase reflect:
 1. Menambahkan response ke STM (Short Term Memory)
 2. Evaluasi kualitas interaksi
 3. Trigger konsolidasi memori jika diperlukan
 4. Log performance metrics

Arsitektur aplikasi memiliki struktur yang modular dan memiliki komponen utama:

- PersonalAssistantAgent : Main agent class yang mengatur semua komponen
- AgentLoop: Implementasi pola "Observe-Decide-Act"
- MemoryManager: Mengelola memori STM dan LTM
- Tools: Mengelola tools yang digunakan
- Models: Mengelola AI Model

Serta aplikasi memiliki fitur untuk logging & monitoring dengan sistem logging komprehensif untuk:

- Chat logs: record semua percakapan
- Performance logs: waktu response, memory usage
- Error logs: debugging dan troubleshooting
- Debug logs: detail internal operations

- **Pendekatan memory (Episodic / Semantic)**

AI Agent yang dibuat ini mengimplementasikan hybrid memory system yang menggabungkan:

- Short-Term-Memory (STM): memori episodik untuk percakapan saat ini, yang berfungsi untuk:
 - Menyimpan percakapan sesi aktif dalam urutan kronologis
 - Context Switching untuk setiap interaksi
 - Temporary workspace untuk reasoning agent
 - Session management agar membersihkan cache saat sesi selesai
- Long-Term-Memory (LTM): memori semantik dan episodik jangka panjang, dengan dual storage system:
 - Vector Database (semantic memory) yang memiliki karakteristik:
 - Semantic Search berdasarkan similarity
 - Context-aware retrieval menggunakan embeddings
 - Conceptual relationships antar informasi
 - Content-based indexing
 - SQLite Database (Structured Episodic Memory)
- **RAG / Prompt strategy yang digunakan**
 - RAG Strategy
 - Hybrid retrieval: Semantic (Vector) + Structured (SQL)
 - Multi-granularity: Recent + Historical content
 - Importance-weighted: Higher importance = higher retrieval priority
 - Context-aware: Retrieval based on current conversation state
 - Prompt Engineering
 - Persona-driven: Different prompts for different agent personalities
 - Structured output: Consistent decision format for parsing
 - Tool-aware: Dynamic tool information integration
 - Context-enriched: User preferences and memory integration
 - Performance Optimizations
 - Lazy loading: Memory loaded only when needed
 - Caching: Frequent queries cached for speed
 - Token management: Smart context truncation
 - Fallback strategies: Graceful degradation when components fail
- **Hasil & refleksi**

Hasil dari pengembangan yang dilakukan adalah dapat mendemonstrasikan pembuatan AI Agent yang komprehensif dengan menggabungkan modern AI techniques (RAG, memory system, tool usage) dalam arsitektur yang scalable dan maintainable. Dan dalam proses pembuatannya mengajarkan pentingnya planning, desain yang modular dan iterative development dalam membuat AI sistem yang kompleks. Beberapa kali mengalami kegagalan akibat error yang terjadi namun dengan prompting yang benar dapat menyelesaikan bug atau error yang terjadi sehingga aplikasi berjalan dengan normal