

**Paicode: Agentic AI berbasis CLI untuk membantu
proses coding secara interaktif ditenagai LLM
eksternal via API**

**I PUTU GEDE GILANG TEJA KRISHNA
225410001**

**INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA
Angkatan 2022**

2025

Lembar Pengesahan

Halaman ini berisi pengesahan skripsi oleh dosen pembimbing dan penguji sesuai format kampus. Silakan sesuaikan isi, penandatanganan, tanggal, dan stempel sesuai ketentuan.

Pernyataan Keaslian

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi ini adalah hasil karya sendiri dan tidak memuat karya orang lain yang pernah diajukan untuk memperoleh gelar akademik di perguruan tinggi manapun, kecuali bagian-bagian tertentu yang dirujuk sebagaimana tercantum dalam daftar pustaka.

Yogyakarta,

I PUTU GEDE GILANG TEJA KRISHNA

NIM: 225410001

Kata Pengantar

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Ucapan terima kasih disampaikan kepada semua pihak yang telah membantu dalam penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

Yogyakarta,

I PUTU GEDE GILANG TEJA KRISHNA

Ucapan Terima Kasih

Penulis menyampaikan terima kasih kepada orang tua, keluarga, dosen pembimbing, penguji, rekan-rekan, dan semua pihak yang telah memberikan dukungan moral maupun material sehingga skripsi ini dapat diselesaikan.

Yogyakarta,

I PUTU GEDE GILANG TEJA KRISHNA

Abstrak

Tuliskan abstrak dalam bahasa Indonesia di sini.

Abstract

Write the abstract in English here.

Daftar Isi

Lembar Pengesahan	i
Pernyataan Keaslian	ii
Kata Pengantar	iii
Ucapan Terima Kasih	iv
Abstrak	v
Abstract	vi
1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
2 Tinjauan Pustaka	4
2.1 Teori Dasar	4
2.2 Penelitian Terkait	4
2.3 Posisi Penelitian	4
3 Metodologi Penelitian	5
3.1 Metode Pengembangan	5
3.2 Arsitektur Sistem	5
3.3 Alat dan Lingkungan	5
3.4 Prosedur Penelitian	5
4 Implementasi dan Hasil	6
4.1 Implementasi Paicode	6
4.2 Contoh Sesi Interaktif	6
4.3 Evaluasi	6

5	Kesimpulan dan Saran	7
5.1	Kesimpulan	7
5.2	Saran	7

Daftar Gambar

Daftar Tabel

BAB 1

Pendahuluan

1.1 Latar Belakang

Perkembangan *Large Language Model* (LLM) telah mendorong lahirnya beragam asisten pemrograman yang mampu membantu pengembang perangkat lunak dalam menulis, meninjau, dan memodifikasi kode. Meskipun demikian, sebagian besar asisten tersebut beroperasi sebagai ekstensi editor atau layanan berbasis *cloud* yang menyimpan, memproses, atau melatih dari data pengguna. Kondisi ini menimbulkan kekhawatiran terkait privasi, kendali atas data, serta ketergantungan pada antarmuka tertentu.

Di sisi lain, *Command Line Interface* (CLI) tetap menjadi lingkungan kerja yang penting bagi banyak pengembang karena sifatnya yang ringan, dapat diotomasi, dan mudah diintegrasikan dengan beragam alat. Integrasi kemampuan agen cerdas yang *stateful* dan *proactive* ke dalam CLI berpotensi mempercepat proses pengembangan perangkat lunak tanpa mengorbankan prinsip *local-first*. Pendekatan *local-first* memusatkan kendali pada mesin pengguna sehingga interaksi, konteks, dan perubahan berkas terjadi secara lokal, sementara panggilan LLM eksternal hanya dilakukan sebatas kebutuhan inferensi [2, 1].

Penelitian ini menghadirkan **Paicode**, sebuah agen AI berbasis CLI yang dirancang untuk membantu proses pengembangan perangkat lunak secara interaktif. Paicode mampu: (i) mengobservasi struktur proyek (`TREE`, `LIST_PATH`); (ii) membaca dan menulis berkas (`READ`, `WRITE`); (iii) memodifikasi kode secara terarah dengan batasan perubahan berbasis diff (`MODIFY`); serta (iv) menegakkan kebijakan keamanan sistem berkas (memblokir akses ke direktori sensitif seperti `.git`, `venv`, dan `.env`). Sistem diimplementasikan pada lingkungan Ubuntu dengan bahasa pemrograman Python, pengelolaan dependensi melalui Poetry, dan menggunakan API Gemini sebagai LLM.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah yang diajukan adalah sebagai berikut:

1. Bagaimana merancang arsitektur agen AI berbasis CLI yang *stateful*, *proactive*, dan menjunjung prinsip *local-first* untuk mendukung proses pengembangan perangkat lunak?
2. Bagaimana mengintegrasikan kemampuan observasi proyek, manipulasi berkas, serta modifikasi kode terarah berbasis deskripsi pengguna dengan pengamanan terhadap jalur dan direktori sensitif?
3. Bagaimana mengevaluasi efektivitas Paicode dalam membantu tugas-tugas pemrograman dibandingkan proses manual atau alat pembanding yang relevan?

1.3 Batasan Masalah

Agar fokus penelitian terjaga dan implementasi dapat dilakukan secara terukur, batasan-batasan berikut ditetapkan:

- Lingkungan target adalah sistem operasi Ubuntu (Linux) dengan antarmuka CLI.
- Bahasa pemrograman utama adalah Python; contoh dan skenario uji berfokus pada ekosistem Python/Unix.
- Layanan LLM eksternal menggunakan API Gemini; kualitas respons bergantung pada model dan tidak menjadi ruang lingkup untuk dioptimasi ulang.
- Dukungan multi-pengguna, kolaborasi real-time, dan integrasi langsung dengan editor tidak dibahas pada versi ini.
- Aspek visual seperti diagram dan ilustrasi antarmuka ditunda pada tahap akhir; fokus laporan adalah narasi dan hasil teknis.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah membangun dan mengevaluasi sebuah agen AI berbasis CLI yang dapat membantu pengembang dalam proses pemrograman secara interaktif. Secara khusus, penelitian menargetkan:

1. Merancang arsitektur Paicode yang mencakup modul agen, jembatan LLM, antarmuka CLI, lapisan keamanan sistem berkas, serta komponen tampilan terminal.
2. Mengimplementasikan kemampuan observasi proyek, manipulasi berkas, dan modifikasi kode terarah dengan mekanisme *patch/diff* untuk membatasi ruang perubahan.

3. Menyusun prosedur evaluasi dengan skenario tugas pemrograman yang representatif dan mengukur peningkatan produktivitas atau kualitas hasil.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini meliputi:

- **Akademis:** menyediakan studi kasus dan arsitektur rujukan untuk pengembangan agen AI *local-first* di lingkungan CLI, serta memperkaya literatur mengenai integrasi LLM dalam alur kerja rekayasa perangkat lunak.
- **Praktis:** menghadirkan alat bantu yang *privacy-aware* dan mudah diintegrasikan dengan berbagai IDE karena beroperasi langsung pada sistem berkas; memfasilitasi pembuatan struktur proyek, pembacaan, dan modifikasi kode secara cepat dan terarah.

BAB 2

Tinjauan Pustaka

2.1 Teori Dasar

Bahas konsep-konsep: Command Line Interface (CLI), AI Agent, Large Language Model (LLM), local-first software, manajemen dependensi (Poetry), dan Rich TUI.

2.2 Penelitian Terkait

Ringkas penelitian/alat terkait AI coding assistant atau agentic AI. Bandingkan pendekatan dan hasil.

2.3 Posisi Penelitian

Posisikan kontribusi paicode dibandingkan karya-karya sebelumnya.

BAB 3

Metodologi Penelitian

3.1 Metode Pengembangan

Jelaskan model pengembangan (prototyping, waterfall, agile, R&D) yang digunakan dan alasan pemilihan.

3.2 Arsitektur Sistem

Jelaskan arsitektur paicode: modul `agent.py`, `llm.py`, `fs.py`, `cli.py`, dan `ui.py`. Ser-takan diagram alir atau class diagram bila perlu.

3.3 Alat dan Lingkungan

Sebutkan Python, Poetry, Git, Google Gemini API, serta lingkungan Ubuntu.

3.4 Prosedur Penelitian

Jabarkan langkah-langkah eksperimen, pengumpulan data, dan kriteria evaluasi.

BAB 4

Implementasi dan Hasil

4.1 Implementasi Paicode

Uraikan proses instalasi (`poetry install`), konfigurasi API key (`poetry run pai config -set <KEY>`), dan cara menjalankan (`poetry run pai`).

4.2 Contoh Sesi Interaktif

Berikan contoh sesi terminal, output TREE/LIST, dan pembacaan/penulisan/MODIFY file.

4.3 Evaluasi

Jelaskan skenario uji, metrik evaluasi, dan bandingkan dengan proses manual atau alat lain.

BAB 5

Kesimpulan dan Saran

5.1 Kesimpulan

Rangkum temuan utama dan pencapaian tujuan penelitian.

5.2 Saran

Berikan saran pengembangan (multi-LLM support, integrasi editor, perbaikan keamanan, pengujian lanjutan, dsb.).

Bibliografi

- [1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [2] Ashish Vaswani et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.