

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama pelaksanaan magang di Kuasar, posisi yang dijalankan adalah sebagai *AI Engineer Intern* dengan fokus utama pada pengembangan *backend* untuk sistem Alfred AI. *Platform* ini merupakan sistem percakapan AI berbasis Open WebUI yang dirancang untuk mengintegrasikan dan mengelola berbagai *Large Language Model* (LLM) secara bersamaan dalam satu antarmuka *backend* yang modular.

Seluruh aktivitas dijalankan dalam cakupan tim *backend engineering*, dengan fokus pada pengembangan komponen inti seperti perancangan API, *deployment* sistem, serta konfigurasi infrastruktur *cloud*. Setiap tahapan pekerjaan dilaksanakan di bawah supervisi langsung dari mentor teknis, disertai koordinasi intensif lintas divisi untuk memastikan integrasi sistem berjalan optimal. Kolaborasi utama yang terjalin bersama yakni:

1. Tim *Frontend*

Berfungsi untuk memastikan integrasi API *backend* dengan antarmuka pengguna berjalan optimal. Aktivitas ini mencakup penyesuaian struktur JSON, penambahan *field respons*, dan sinkronisasi *endpoint* agar sesuai dengan spesifikasi *frontend*.

2. Tim Model AI

Bertanggung jawab dalam mendukung proses integrasi dan registrasi model AI seperti Gemini, DeepSeek, dan MedGemma ke dalam sistem. Kegiatan meliputi pengujian model serta konfigurasi *endpoint backend* agar siap digunakan secara dinamis.

3. *Supervisor* dan *Project Lead*

Mengikuti pertemuan rutin mingguan untuk membahas perkembangan tugas, mengidentifikasi hambatan teknis, serta menetapkan prioritas kerja yang relevan dengan *roadmap* proyek.

Dalam pelaksanaan teknis harian, digunakan beberapa *tools* dan *platform* utama, antara lain:

1. Bitbucket, digunakan dalam sistem untuk manajemen versi kode dan kolaborasi pengembangan.

2. Google Cloud Platform (GCP), digunakan untuk deployment sistem, pengaturan *instance* VM, serta konfigurasi database menggunakan Cloud SQL.
3. Discord dan media komunikasi internal, sebagai media utama untuk diskusi teknis harian, *troubleshooting*, dan koordinasi lintas tim.

3.2 Tugas Kerja Magang

Selama masa pelaksanaan magang di Kuasar, tanggung jawab teknis yang dijalankan berfokus pada perancangan dan pengembangan backend untuk sistem Alfred AI, dengan penekanan pada integrasi *Large Language Model* (LLM) dan pembangunan layanan API yang handal dan aman. Aktivitas yang dilakukan tidak hanya mencakup aspek teknis, tetapi juga berkaitan dengan kolaborasi lintas tim serta penyusunan dokumentasi untuk mendukung keberlangsungan sistem. Adapun rincian tugas yang dijalankan adalah sebagai berikut:

1. Perancangan dan Pengembangan API *Backend*
Merancang serta membangun API *backend* yang menjadi penghubung utama antara *platform* Alfred AI dan berbagai model LLM. API yang dikembangkan dirancang agar modular, efisien, serta mampu menangani komunikasi dua arah dengan tingkat keamanan tinggi.
2. Implementasi dan Integrasi Model LLM
Mengintegrasikan beberapa model LLM ke dalam sistem Alfred AI, termasuk model-model seperti GPT, DeepSeek, Gemini, LLaMA, MedGemma, dan lainnya. Integrasi dilakukan melalui endpoint dinamis yang memungkinkan pengguna memilih model yang ingin digunakan dalam sesi mereka.
3. Optimalisasi Arsitektur *Backend* untuk LLM *Inference*
Mengoptimalkan performa *backend* agar mampu menjalankan proses *inference* dari model-model LLM secara real-time dan terukur. Infrastruktur *backend* juga dikembangkan untuk mendukung fitur unik seperti diskusi multi-pengguna (*multi-user discussion*), penyimpanan *knowledge artifact* sebagai memori *prompt* yang dapat digunakan kembali, serta sistem autentikasi pengguna berbasis pembayaran.
4. Kolaborasi Tim dan Sinkronisasi Sistem
Berkoordinasi secara aktif dengan tim *frontend* dan tim AI *engineer* untuk

memastikan keselarasan antara fungsionalitas antarmuka pengguna dan proses inferensi model. Kolaborasi ini dilakukan melalui diskusi teknis rutin, *code review*, dan pemanfaatan alat bantu manajemen proyek.

5. Penyusunan Dokumentasi Teknis

Menyusun dokumentasi teknis yang mencakup arsitektur sistem, desain *endpoint* API, proses integrasi model, serta panduan penggunaan sistem. Dokumentasi ini disusun sebagai bagian dari proses standarisasi internal dan untuk mendukung pengembangan berkelanjutan oleh tim.

3.3 Uraian Pelaksanaan Magang

Proses kerja mengikuti siklus pengembangan mingguan yang terstruktur berdasarkan prioritas proyek dan kebutuhan sistem. Tabel berikut merangkum ringkasan pekerjaan yang diselesaikan setiap minggunya selama periode magang berlangsung.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Melakukan <i>setup environment</i> GCP dan memasang <i>dependencies</i> , memulai riset untuk <i>deployment</i> model DeepSeek dan menyelesaikan konfigurasi Vertex AI dan integrasi model.
2	Melakukan <i>deployment</i> DeepSeek melalui Hugging Face, menangani konfigurasi firewall, <i>fine-tuning prompt model</i> , dan optimasi akurasi prediksi.
3	Menginstal Open WebUI melalui SSH, melakukan <i>deployment</i> model AI dan implementasi <i>middleware</i> FastAPI, serta mengamankan <i>deployment</i> dengan HTTPS dan <i>benchmarking</i> .
4	Melakukan <i>deployment</i> Open WebUI ke URL publik, verifikasi konfigurasi <i>backend</i> dan perbaikan akses SSH, dan optimasi performa model DeepSeek.
5	Migrasi database PostgreSQL dan konfigurasi <i>proxy</i> untuk Cloud SQL di GCP dan pengembangan <i>endpoint</i> API dan pengamanan sistem backend.

Bersambung ke halaman berikutnya

Minggu Ke -	Pekerjaan yang dilakukan
6	Deployment model DeepSeek R1:1.5B dan R1 7B, integrasi OCR dan Google OAuth, dan aktivasi Web Search API dan perbaikan autentikasi.
7	Perbaikan error pada Open WebUI, migrasi <i>backend</i> ke VM baru, dan <i>deployment</i> model LLaMA dan peningkatan performa sistem.
8	Menyelesaikan setup <i>backend</i> , validasi database dan fitur API dan menyusun dokumentasi sistem serta verifikasi fitur <i>chat</i> pada Cloud SQL.
9	Finalisasi dokumentasi teknis (README.md dan deployment.md), pengembangan dan pengujian <i>endpoint</i> CRUD untuk <i>chat</i> dan proyek, dan integrasi fitur <i>upload file</i> .
10	Verifikasi <i>deployment</i> pada GCP, modifikasi skema <i>database</i> dan integrasi sistem <i>artifact</i> dan perbaikan <i>error</i> saat pembuatan proyek.
11	Modifikasi <i>endpoint</i> (seperti <i>artifact</i> dan <i>project</i>), implementasi sistem undangan pengguna menggunakan Gmail, dan <i>refactor backend</i> dan peningkatan logika sistem.
12	Perbaikan pada <i>endpoint</i> API (<i>search</i> , <i>meta field</i>), penyempurnaan alur undangan pengguna, dan penambahan dukungan <i>item/chat</i> lintas <i>endpoint</i> .
13	Refactor logika <i>backend</i> (contoh: <i>knowledge.py</i>), penyempurnaan alur pendaftaran pengguna, dan perbaikan <i>race condition</i> , dan pembatasan Swagger UI di produksi.
14	Finalisasi integrasi model (Gemini, DeepSeek, MedGemma), pembuatan <i>endpoint</i> pencarian untuk <i>admin</i> , dan riset tentang metode enkripsi data.
15	Penyempurnaan sistem <i>backend</i> Alfred AI, mulai dari pengamanan login, pengembangan <i>endpoint</i> baru, hingga reintegrasi model Gemini. Beberapa perbaikan juga dilakukan pada validasi <i>file</i> dan penyesuaian API agar sesuai dengan kebutuhan <i>frontend</i> .

Bersambung ke halaman berikutnya

Minggu Ke -	Pekerjaan yang dilakukan
16	Perbaiki sistem autentikasi, pemulihan <i>server backend</i> , serta restrukturisasi kode agar lebih terkelola. Selain itu, dilakukan migrasi model AI ke lingkungan pengembangan dan investigasi masalah performa sistem, yang didiskusikan dalam rapat teknis mingguan.

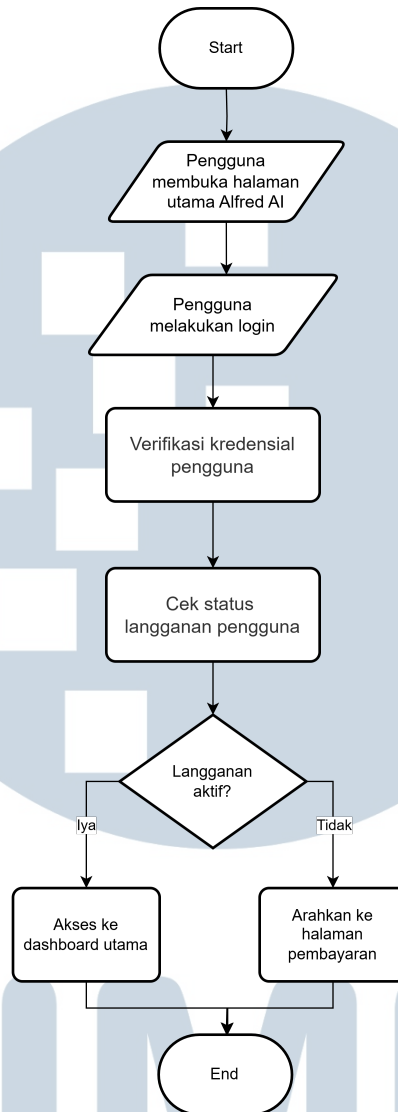
Pembagian tugas yang dilakukan dari minggu ke minggu menunjukkan peningkatan kompleksitas pekerjaan serta keterlibatan aktif dalam pengembangan sistem backend. Penguasaan teknologi seperti Python, FastAPI, dan GCP turut berkembang seiring dengan praktik langsung di proyek berskala nyata. Selain itu, pengalaman bekerja sama dalam tim rekayasa perangkat lunak juga memperdalam pemahaman terhadap manajemen sistem AI yang kompleks dan skalabel.

3.3.1 Perancangan Alfred AI

A *Flowchart* (alur sistem Alfred AI)

Proses dimulai dengan mengakses halaman utama *platform* Alfred AI melalui *website*. Setelah halaman berhasil dimuat, sistem menyediakan fitur autentikasi untuk memastikan hanya pengguna yang terverifikasi yang dapat melanjutkan ke tahap berikutnya dalam *website* ini.



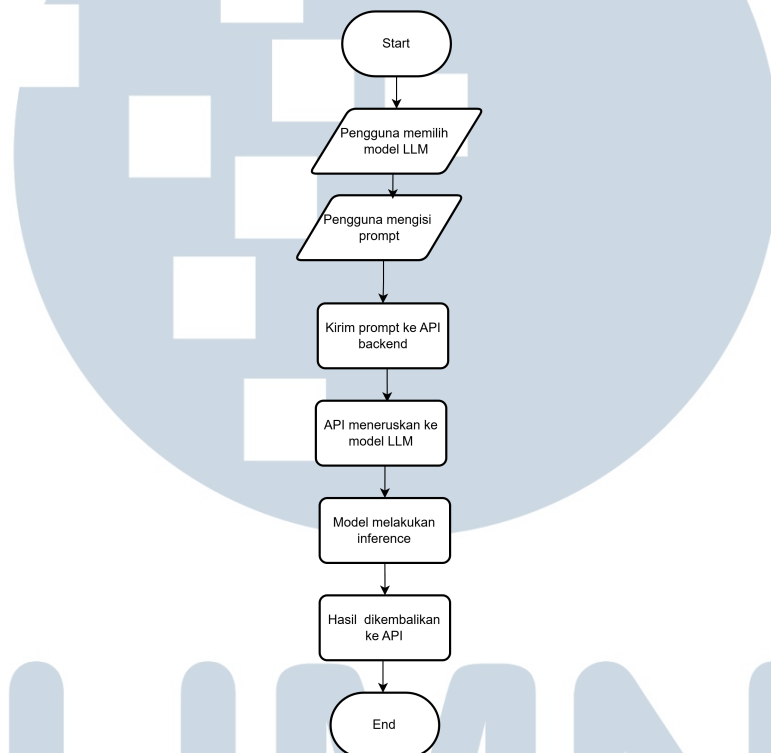


Gambar 3.1. Flowchart verifikasi user

Tahapan *login* dilakukan dengan memasukkan kredensial yang sesuai, yang kemudian diverifikasi oleh sistem *backend*. Setelah verifikasi berhasil, sistem secara otomatis memeriksa status langganan untuk memastikan akses terhadap fitur utama hanya diberikan kepada akun yang memiliki langganan aktif. Apabila status langganan aktif terkonfirmasi, sistem akan mengarahkan ke halaman *dashboard* utama. Sebaliknya, jika langganan tidak aktif, pengguna akan diarahkan ke halaman pembayaran untuk melanjutkan atau memperbarui langganan.

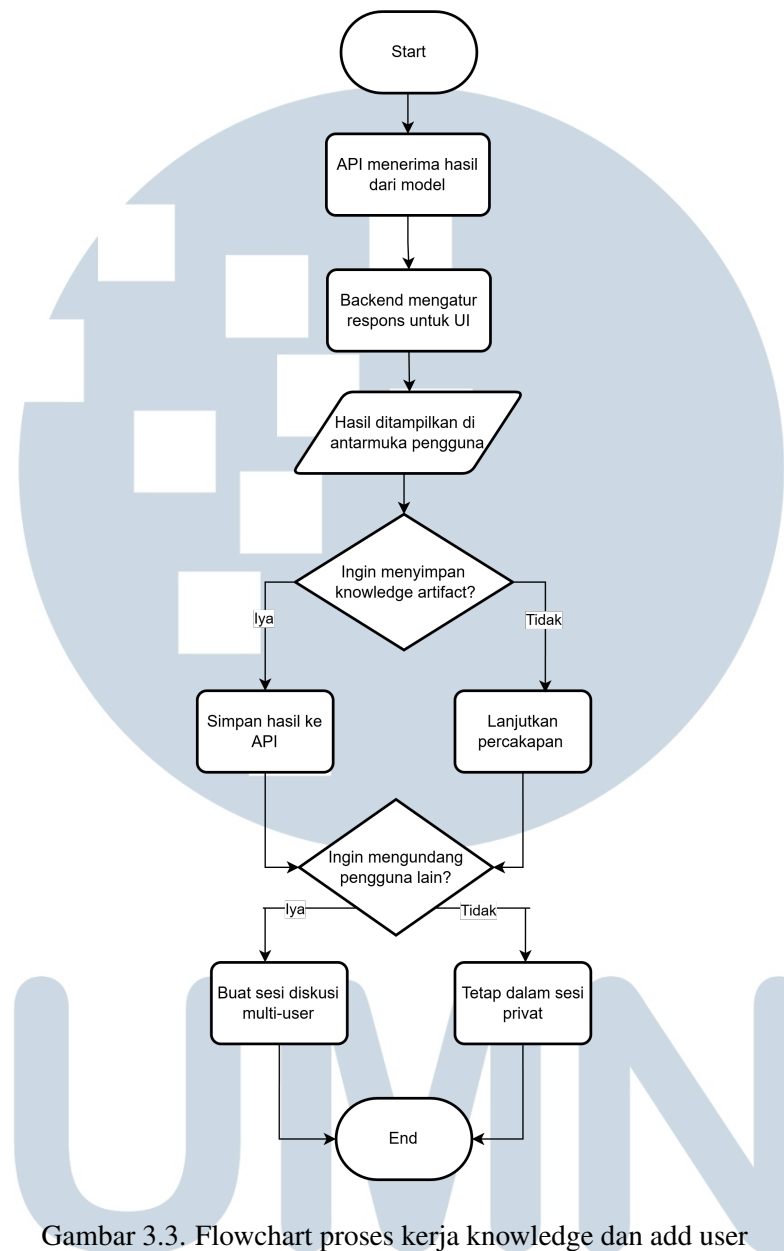
Setelah berhasil mengakses *dashboard*, langkah selanjutnya adalah memilih model LLM (*Large Language Model*) yang tersedia sesuai kebutuhan. Setelah model dipilih, sistem memungkinkan pengisian *prompt* sebagai masukan untuk

model. *Prompt* yang telah diisi akan diteruskan melalui *API backend*, yang kemudian memproses dan mengarahkan prompt tersebut ke model LLM yang telah ditentukan. Model akan melakukan proses inferensi berdasarkan *input* yang diberikan dan menghasilkan *output* dalam bentuk respons. Respons ini dikembalikan ke sistem *backend* melalui API untuk kemudian disiapkan dan ditampilkan pada antarmuka pengguna.



Gambar 3.2. Flowchart pemilihan model

Setelah hasil inferensi ditampilkan, sistem menyediakan opsi lanjutan yakni untuk menyimpan hasil tersebut sebagai *knowledge artifact* atau tidak. Jika disetujui, hasil akan disimpan ke dalam basis data sebagai referensi yang dapat diakses kembali di masa mendatang. Selain itu, terdapat opsi untuk memulai kolaborasi dengan pengguna lain melalui sesi diskusi *multi-user*. Jika kolaborasi diperlukan, sistem akan membuat ruang diskusi bersama dengan mengundang *user* lain menggunakan email, jika tidak, percakapan tetap berlangsung dalam sesi privat. Proses ini berakhir ketika seluruh kebutuhan interaksi terpenuhi, baik dalam bentuk hasil respons model maupun penyimpanan data untuk keperluan lanjutan.



Gambar 3.3. Flowchart proses kerja knowledge dan add user

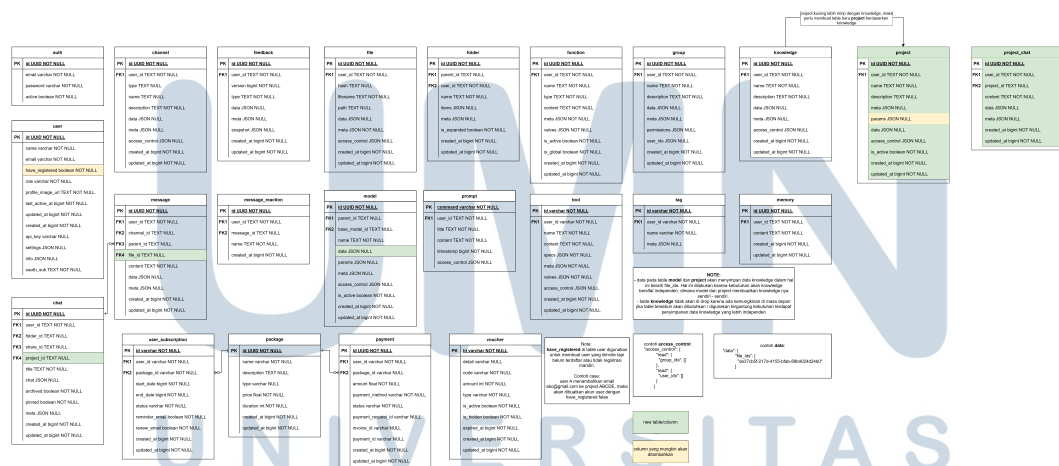
Dengan alur ini, sistem memastikan bahwa seluruh proses mulai dari autentikasi, pemilihan model, hingga tampilan hasil berjalan secara efisien dan terintegrasi.

B Perancangan Database

Struktur *database* Alfred AI dirancang modular dan *scalable*. Tabel-tabel inti seperti *user*, *chat*, *project*, dan *knowledge* saling terhubung untuk mendukung skenario kolaborasi, pencatatan riwayat, dan penyimpanan data model secara efisien. Berikut penjelasan yang lebih rinci mengenai kegunaan tiap tabel pada

database dibawah:

- Tabel *user* menyimpan info pengguna, termasuk kolom *have registered* untuk menandai apakah pengguna sudah aktif terdaftar atau tidak.
- Tabel *project* dan *project chat* menjadi fondasi utama pengelolaan proyek. Masing-masing mempunyai kolom seperti *meta*, *params*, *data*, dan *access control* dalam bentuk JSON, yang memudahkan fleksibilitas struktur data dan pengaturan hak akses.
- Tabel *knowledge* menyimpan data yang bisa digunakan lintas proyek dan dapat berdiri secara independen. Ini memungkinkan pembuatan *knowledge base* yang *reusable*.
- Tabel *chat* terhubung ke proyek lewat kolom *project id*, dan juga mengacu ke tabel *message*, yang mendukung lampiran file (*file id*) dan reaksi pengguna (*message reaction*).
- Tabel tambahan seperti *model*, *tool*, *prompt*, *tag*, *feedback*, dan *memory* menyediakan *metadata* dan alat bantu untuk sistem berjalan optimal.

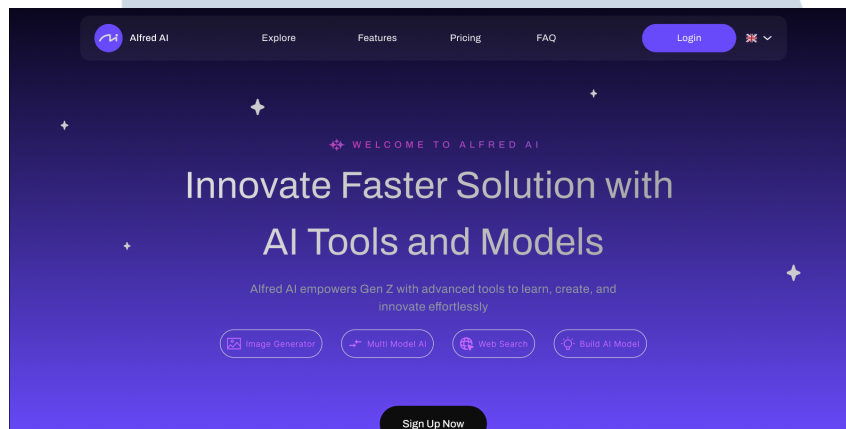


Gambar 3.4. Database ERD

Desain ini mendukung pengembangan fitur lanjutan, seperti pencatatan granular, *multi-user collaboration*, dan *knowledge sharing* antar proyek.

C Perancangan Antarmuka (*Mockup*)

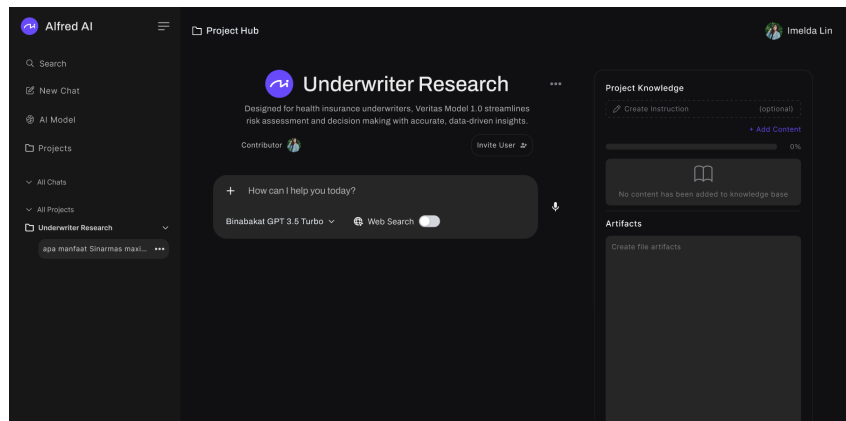
Tampilan antarmuka Alfred AI dirancang dengan fitur-fitur yang mendukung kolaborasi dalam proyek berbasis kecerdasan buatan secara efektif. Pada halaman beranda, pengguna dapat melihat penekanan pada kemudahan dalam mengakses berbagai fitur seperti *multi-model AI*, *image generator*, *web search*, hingga fitur untuk membangun model AI sendiri.



Gambar 3.5. Mockup halaman beranda

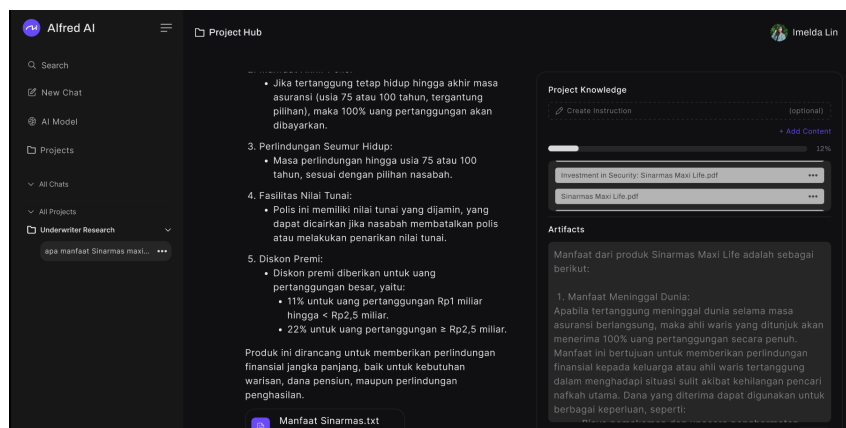
Fitur-fitur yang ditawarkan memberikan fleksibilitas dalam menggunakan berbagai alat berbasis AI dalam satu *platform* terpadu. Desain halaman ini juga menyoroti kemudahan akses untuk mendaftar dan masuk ke dalam sistem, menjadikan *platform* ini cocok untuk pengguna dengan kebutuhan eksplorasi teknologi AI yang efisien dan terpusat.

Berikut salah satu contoh *dashboard* utama dalam tampilan proyek bernama “*Underwriter Research*”, dimana sistem menyediakan kemampuan untuk mengunggah dokumen penting seperti PDF yang berkaitan dengan topik riset, sebagai contoh dokumen Sinarmas Maxi Life.pdf. Dokumen yang diunggah akan masuk ke dalam bagian “*Project Knowledge*” dan dapat diproses lebih lanjut menjadi *structured content* yang informatif.



Gambar 3.6. Mockup dashboard project

Selain unggahan *file*, sistem juga menampilkan hasil ekstraksi pengetahuan dalam bentuk *Artifacts*, di mana informasi dari dokumen ditransformasikan menjadi teks terstruktur yang dapat dijadikan referensi atau jawaban atas pertanyaan. Sebagai contoh, ditampilkan penjelasan manfaat produk asuransi Sinarmas Maxi Life yang telah terurai dengan jelas dalam bentuk poin-poin. Integrasi antara unggahan dokumen dan pembangkitan pengetahuan ini memperkuat nilai guna *platform* dalam mendukung riset dan pengambilan keputusan berbasis data.

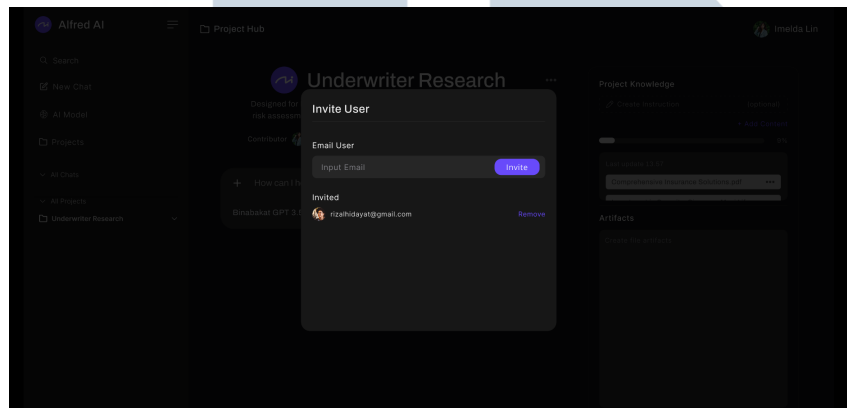


Gambar 3.7. Mockup user menggunakan fitur artifact

Setelah berhasil masuk ke dalam *platform* dan memilih proyek yang sedang dikerjakan, tersedia fitur yang memungkinkan pengguna untuk mengundang kolaborator lain melalui input alamat email. Proses ini dilakukan melalui panel “*Invite User*” yang dapat diakses langsung dalam tampilan proyek aktif.

Setelah undangan berhasil dikirim, alamat email yang telah diundang akan ditampilkan dalam daftar “*Invited*” bersama opsi untuk menghapus undangan

tersebut jika diperlukan. Mekanisme ini memungkinkan kolaborasi antar pengguna dalam satu proyek berjalan dengan lebih terstruktur dan aman, mengingat hanya pengguna yang diundang secara langsung yang memiliki akses terhadap proyek tersebut.



Gambar 3.8. Mockup invite user

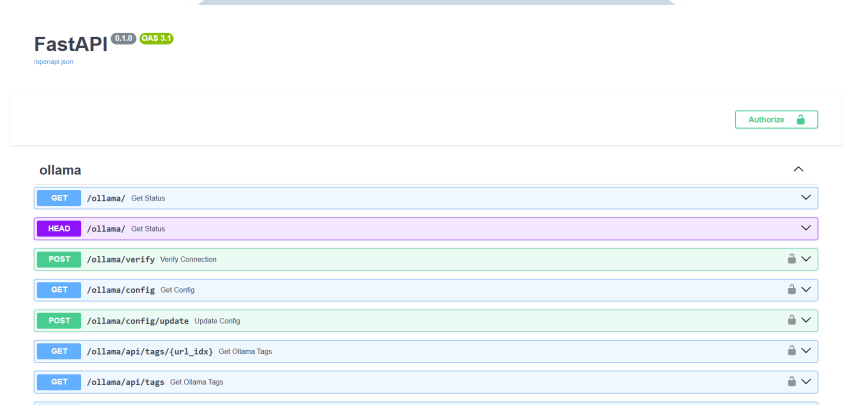
Seluruh alur ini menunjukkan integrasi menyeluruh antara kolaborasi tim, pemanfaatan dokumen sebagai pengetahuan, dan interaksi dengan model AI yang dapat mendukung riset dan pengambilan keputusan secara lebih efisien.

3.3.2 Hasil Alfred AI

A *Endpoint*

Setelah melalui proses perancangan dan implementasi *backend*, sistem Alfred AI telah berhasil dioperasikan secara fungsional melalui layanan FastAPI yang dibangun selama periode magang. Pengujian dilakukan untuk memastikan komunikasi antara API dan model-model LLM berjalan secara optimal, termasuk dalam mengirim dan menerima *prompt*, serta menangani sesi percakapan pengguna. Beberapa hasil yang ditampilkan pada bagian ini mencakup tangkapan layar dari *endpoint* FastAPI yang menunjukkan respons sistem terhadap permintaan pengguna, termasuk validasi token, pemilihan model, dan hasil inferensi. Selain itu, antarmuka situs Alfred AI yang telah terintegrasi penuh dengan layanan *backend* juga diperlihatkan sebagai bukti keberhasilan implementasi sistem secara menyeluruh. Gambar berikut memperlihatkan dokumentasi endpoint FastAPI yang telah dikembangkan yakni menunjukkan bagian dari kategori */ollama*, yaitu *endpoint* untuk memverifikasi koneksi dan konfigurasi dari salah satu model LLM

bernama Ollama. *Endpoint* ini penting untuk memastikan bahwa *backend* dapat berkomunikasi secara aktif dengan model-model yang disediakan, serta memantau status integrasi.



Gambar 3.9. Endpoint Ollama

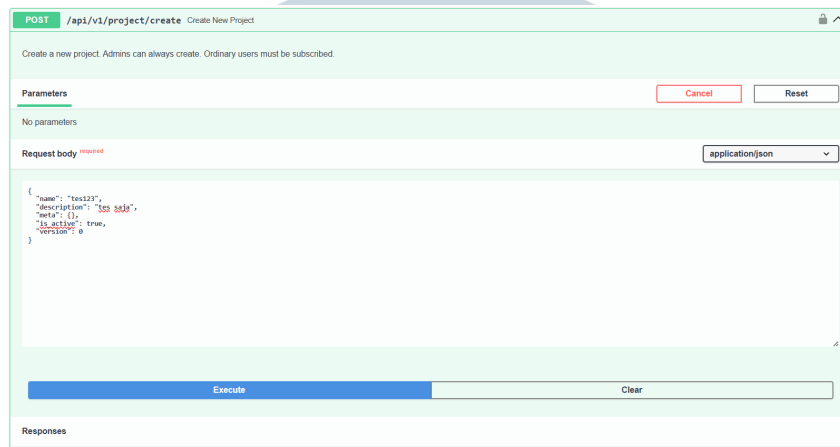
Selanjutnya, gambar dibawah menunjukkan endpoint pada kategori */project*, di mana tersedia fungsi-fungsi utama seperti pembuatan proyek (*/create*), pembaruan (*/update*), penghapusan (*/delete*), serta penambahan dan pengelolaan file proyek. *Endpoint* ini memungkinkan pengguna untuk menyusun dan mengelola proyek yang terhubung langsung dengan sesi chat dan *knowledge artifact* dalam Alfred AI.



Gambar 3.10. Endpoint projects

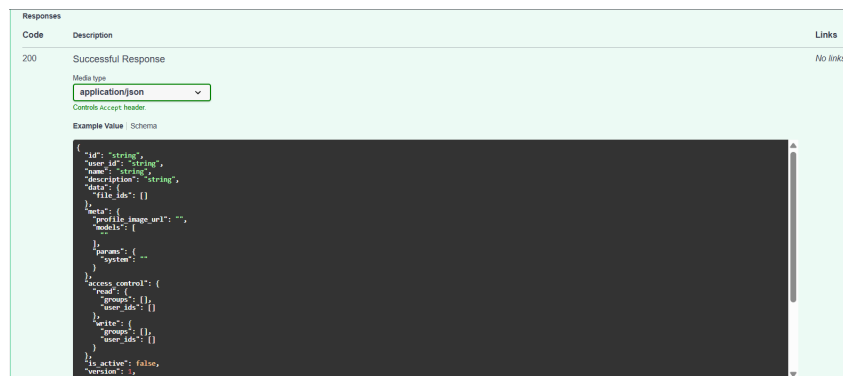
Sebagai bagian dari proses pengujian dan validasi layanan backend Alfred AI, dilakukan percobaan pada salah satu *endpoint* utama, yaitu POST */api/v1/project/create*. *Endpoint* ini berfungsi untuk membuat entitas proyek baru di dalam sistem, yang kemudian dapat diisi dengan berbagai *file*, konfigurasi model,

serta informasi lainnya yang berkaitan dengan kegiatan percakapan dan *knowledge management*.



Gambar 3.11. Percobaan endpoint

Setelah tombol *Execute* dijalankan, sistem memproses permintaan pembuatan proyek dan memberikan respons berhasil (kode 200) yang berisi objek proyek lengkap dalam format JSON. Respons tersebut mencakup informasi penting seperti ID proyek, ID pengguna, deskripsi, *metadata*, status proyek, serta kontrol akses terhadap proyek tersebut.



Gambar 3.12. Percobaan endpoint success

Hasil uji coba ini membuktikan bahwa layanan API *backend* dapat menerima input dengan baik, memprosesnya sesuai struktur data yang ditentukan, dan mengembalikan *output* yang konsisten dengan skema yang telah dirancang. *Endpoint* ini juga menjadi fondasi utama dalam proses manajemen proyek dan integrasi *file* yang berkaitan dengan penggunaan model LLM dalam sistem Alfred AI.

Selain pengujian terhadap modul proyek dan integrasi model, sistem *backend* Alfred AI juga mencakup berbagai kategori *endpoint* lainnya yang menyusun arsitektur layanan secara menyeluruh. Beberapa *endpoint* ini dikelompokkan berdasarkan fungsi masing-masing dan membentuk fondasi interaksi antara pengguna, model AI, serta berbagai komponen sistem pendukung.

Kategori *audio* memungkinkan konversi teks ke suara atau sebaliknya melalui teknologi *speech-to-text* dan *text-to-speech*. Fitur ini relevan untuk memperluas aksesibilitas sistem serta memperkaya pengalaman pengguna melalui antarmuka suara.

Endpoint dalam *retrieval* dan *knowledge* digunakan untuk mengelola dan mengakses sumber pengetahuan atau dokumen *internal* yang digunakan sebagai referensi dalam proses inferensi. Salah satu fitur unggulan Alfred AI, yaitu *knowledge artifact*, sangat bergantung pada kemampuan sistem dalam menyimpan, menghubungkan, dan memanggil kembali informasi ini saat dibutuhkan. Kategori seperti *configs*, *auths*, dan *users* bertugas mengelola konfigurasi sistem, proses autentikasi, dan manajemen pengguna. *Endpoint* dalam kategori ini berperan penting dalam menjaga keamanan dan personalisasi layanan, termasuk mengatur peran, hak akses, serta status langganan pengguna.

Pada sisi komunikasi, tersedia endpoint dalam kategori *channels* dan *chats* yang digunakan untuk mengelola jalur percakapan dan sesi interaksi pengguna dengan model AI. Kategori *models*, *prompts*, dan *tools* berfungsi untuk menangani pemilihan model, pembuatan prompt dinamis, serta integrasi berbagai utilitas tambahan ke dalam alur interaksi pengguna.

Sistem ini juga menyediakan fitur *memories* dan *folders* yang memungkinkan pengguna menyimpan riwayat interaksi, menyusun informasi dalam *folder* tertentu, dan mengelola struktur data secara lebih tertata. Adapun kategori *groups*, *files*, dan *functions* difungsikan untuk mendukung kolaborasi multi-user, pengunggahan file, serta integrasi fungsi-fungsi tambahan berbasis *programmatic logic*.

Selain itu, Alfred AI memiliki modul *evaluations* yang ditujukan untuk menguji performa model atau *pipeline* berdasarkan kriteria tertentu. Endpoint pada kategori *utils* dan *default* disediakan sebagai dukungan umum dan fallback logic untuk menjaga fleksibilitas sistem dalam menangani berbagai jenis permintaan.

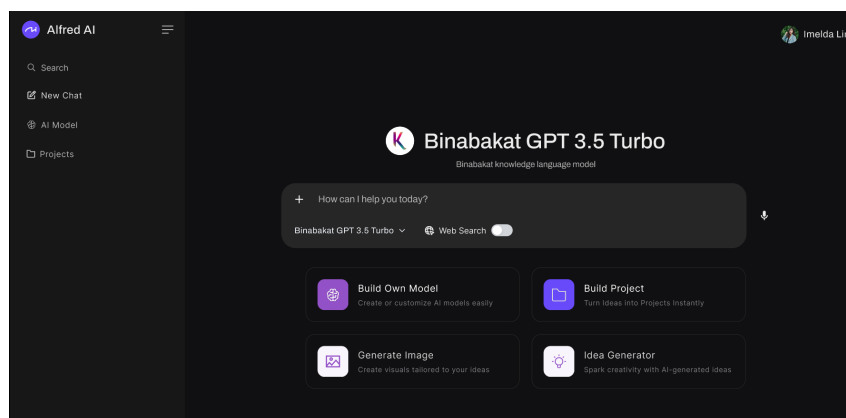
Dengan cakupan *endpoint* yang sangat luas dan modular ini, Alfred AI dikembangkan sebagai *platform* percakapan berbasis LLM yang tangguh dan dapat disesuaikan dengan kebutuhan *enterprise* maupun pengguna individu. Seluruh *endpoint* yang telah dirancang, diuji, dan diimplementasikan ini memberikan

fleksibilitas tinggi untuk mengelola data, *model*, konfigurasi, serta interaksi pengguna dalam satu ekosistem yang terpadu.

Keberhasilan integrasi masing-masing *endpoint* selama masa magang menunjukkan bahwa sistem *backend* Alfred AI telah dibangun dengan pendekatan arsitektur yang skalabel dan siap digunakan dalam skala produksi. Proses pengembangan ini tidak hanya memperkuat kemampuan teknis dalam mengelola layanan *backend* berbasis AI, tetapi juga memberikan pemahaman mendalam terkait desain sistem yang mendukung kolaborasi, efisiensi, dan keberlanjutan.

B Implementasi

Hasil implementasi Alfred AI juga ditampilkan dalam bentuk tampilan antarmuka *web* Alfred AI, gambar dibawah menunjukkan halaman utama Alfred AI setelah pengguna berhasil masuk. Dalam tampilan ini terlihat opsi pemilihan model LLM (seperti "Binabakat GPT 3.5 Turbo"), fitur pencarian, pembuatan proyek, dan pembuatan model. Antarmuka ini menjadi bukti bahwa API yang dikembangkan terintegrasi dengan lancar ke dalam sistem frontend dan mampu melayani pengguna dalam berbagai skenario interaksi.

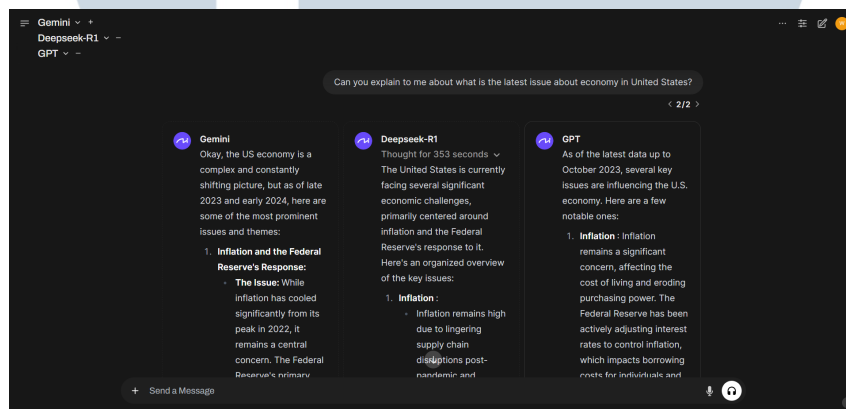


Gambar 3.13. Dashboard awal

Sebagai bagian dari proses validasi akhir terhadap integrasi model dalam sistem Alfred AI, dilakukan uji coba langsung dengan menggunakan *prompt* kepada beberapa model LLM yang telah berhasil diimplementasikan, yaitu Gemini, Deepseek-R1, dan GPT. Uji coba dilakukan dengan memberikan pertanyaan yang sama kepada ketiga model, yakni mengenai isu ekonomi terkini di Amerika Serikat. Tujuannya adalah untuk mengevaluasi kemampuan model dalam memahami

konteks, menyampaikan informasi faktual, serta membandingkan perbedaan sudut pandang dan tingkat detail dari masing-masing model.

Hasil dari uji coba menunjukkan bahwa ketiga model secara umum memberikan respons yang konsisten, dengan inflasi sebagai isu utama yang disoroti. Model Gemini menekankan bahwa meskipun inflasi telah menurun sejak puncaknya pada tahun 2022, isu ini masih menjadi perhatian utama dan erat kaitannya dengan kebijakan suku bunga oleh *Federal Reserve*. Deepseek-R1 menambahkan dimensi penjelasan lebih lanjut dengan mengaitkan inflasi terhadap gangguan rantai pasokan serta dampak berkepanjangan dari pandemi. Sementara itu, GPT menyampaikan bahwa inflasi berdampak langsung terhadap daya beli masyarakat dan biaya hidup, dan menyoroti peran kebijakan moneter dalam menentukan tingkat suku bunga serta biaya pinjaman.



Gambar 3.14. Prompt output

Dengan keberhasilan implementasi ini, sistem Alfred AI kini mampu menjalankan fungsionalitas backend secara penuh, mengelola permintaan pengguna, memproses komunikasi dengan model LLM, serta menyajikan hasil dalam bentuk antarmuka pengguna yang intuitif. Pencapaian ini tidak hanya memperlihatkan kesiapan sistem untuk digunakan dalam skala produksi, tetapi juga menunjukkan kapabilitas rancangan *backend* yang stabil, modular, dan adaptif terhadap pengembangan lebih lanjut.

3.4 Kendala yang Ditemukan

Selama pelaksanaan magang, terdapat sejumlah tantangan yang muncul baik dari sisi teknis maupun proses kerja. Beberapa kendala utama yang dihadapi meliputi:

1. Ketidaksesuaian API dengan *frontend*

Beberapa *endpoint* belum memenuhi ekspektasi *frontend* dalam hal struktur data dan *field* yang dikembalikan, sehingga menyebabkan *error* tampilan dan perlu perbaikan mendalam.

2. Konfigurasi dan *deployment* GCP

Permasalahan muncul pada saat pengaturan VM, autentikasi dengan Cloud SQL, serta konfigurasi *firewall* yang membutuhkan *debugging* lanjutan untuk memastikan sistem dapat berjalan secara stabil di lingkungan *cloud*.

3. *Race condition* dan *bug* asinkron

Terjadi gangguan pada *endpoint* yang digunakan secara paralel, seperti aktivasi akun dan pengambilan data daftar *chat*, yang menyebabkan perilaku tak terduga dalam sistem.

Tantangan-tantangan tersebut menjadi ruang pembelajaran yang signifikan dalam memahami pentingnya desain sistem *backend* yang kokoh dan siap beroperasi dalam skala produksi.

3.5 Solusi yang Diterapkan

Sebagai bentuk penanganan terhadap kendala yang muncul, diterapkan kombinasi strategi teknis dan kolaboratif untuk menjaga stabilitas sistem dan mempercepat proses *debugging*. Langkah-langkah perbaikan yang dilakukan antara lain:

1. Kolaborasi intensif dengan tim *frontend*

Melakukan diskusi rutin untuk menyamakan struktur respons API, menyesuaikan format data, dan melakukan pengujian bersama, serta menambahkan lapisan validasi dan sistem *fallback* untuk menangani masukan tidak valid dan memastikan kestabilan sistem.

2. *Monitoring* dan *logging* di GCP

Memanfaatkan fitur *Logs Explorer* untuk memantau log runtime serta melakukan penelusuran *error* secara *real-time*. Langkah ini membantu dalam mengoptimalkan konfigurasi lingkungan dan pengaturan sistem produksi.

3. *Refactoring* kode

Membersihkan dan menyederhanakan kode pada modul seperti *project.py*, *knowledge.py*, dan *artifact.py* guna meningkatkan *maintainability* dan efisiensi proses pengembangan selanjutnya.

Pendekatan terhadap pemecahan masalah tersebut membentuk dasar keterampilan berpikir sistematis, meningkatkan akurasi dalam debugging, serta mendorong praktik kolaboratif yang efektif dalam tim *engineering*. Seluruh proses tersebut menjadi bekal penting untuk menghadapi tantangan profesional di dunia kerja sebagai pengembang sistem berbasis AI.

