

**IMPLEMENTASI *MULTI-STAGE QUALITY GATE* DAN *MULTI-LEVEL
CHUNKING* DALAM PEMBANGUNAN *KNOWLEDGE GRAPH*
LITERATUR ILMIAH UNTUK SISTEM *RESEARCH AGENTIC AI*
MYSRE**

Disusun untuk memenuhi salah satu syarat pelaksanaan PKL/Magang
Program Studi Informatika



Disusun oleh:

Nama : Muhammad Zaidan Rafat

NIM : 231730016

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI
SULTAN MAULANA HASANUDDIN BANTEN
2026**

HALAMAN PENGESAHAN

Laporan hasil pelaksanaan PKL/Magang ini telah disusun oleh:

Nama : Muhammad Zaidan Rafat
NIM : 231730016
Program Studi : Informatika
Judul PKL/Magang : Implementasi *Multi-Stage Quality Gate* dan *Multi-Layered Chunking* dalam Pembangunan *Knowledge Graph* Literatur Ilmiah untuk Sistem *Research Agentic AI* MySRE
Tempat PKL/Magang : Badan Riset dan Inovasi Nasional
Waktu Pelaksanaan : 18 Februari - 17 Juli 2026

Laporan ini telah diperiksa dan disahkan sebagai laporan hasil pelaksanaan PKL/Magang.

Serang, 13 Juni 2026

Dosen Pembimbing PKL/Magang,



Rio Nurtantyana, S.Pd., M.Pd., M.Sc., Ph.D
NIP.99312152024031001

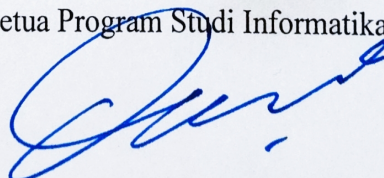
Pembimbing Internal,



Ahmad Tabrani, M.T.I.
NIP.198509172018011002

Mengetahui,

Ketua Program Studi Informatika



Ahmad Tabrani, M.T.I.
NIP.198509172018011002

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan hasil pelaksanaan PKL/Magang yang berjudul “Implementasi *Multi-Stage Quality Gate* dan *Multi-Level Chunking* dalam Pembangunan *Knowledge Graph* Literatur Ilmiah untuk Sistem *Research Agentic AI MySRE*” dengan baik.

Laporan ini disusun sebagai salah satu syarat akademik untuk memenuhi kurikulum Program Studi Informatika pada Fakultas Sains dan Teknologi UIN Sultan Maulana Hasanuddin Banten.

Selama pelaksanaan magang dan penyusunan laporan ini, penulis memperoleh bimbingan, arahan, serta dukungan dari berbagai pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ahmad Tabrani, M.T.I., selaku Ketua Program Studi Informatika sekaligus Dosen Pembimbing Universitas yang telah memberikan arahan, saran, dan bimbingan akademik kepada penulis selama masa perkuliahan dan magang.
2. Kepala Pusat Riset Sains Data dan Informasi (PR-SDI) BRIN beserta seluruh jajaran staf administrasi dan riset atas kesempatan yang diberikan untuk belajar di lingkungan riset nasional.
3. Rio Nurtantyana, S.Pd., M.Pd., M.Sc., Ph.D., selaku Pembimbing Lapangan di PR SDI BRIN yang telah memberikan bimbingan teknis, transfer pengetahuan, dan dukungan penuh selama proses magang riset.
4. Seluruh peneliti dan anggota Kelompok Riset *Human Computer Interaction - Visualization* (HCI-V) PR SDI BRIN atas kolaborasi riset dan diskusi ilmiah yang berharga bagi penulis.
5. Orang tua dan keluarga tercinta atas doa, motivasi, serta dukungan moril yang senantiasa diberikan selama pelaksanaan magang.
6. Rekan-rekan mahasiswa Program Studi Informatika Universitas Islam Negeri Sultan Maulana Hasanuddin Banten dan seluruh pihak yang telah membantu penyelesaian laporan ini.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan dalam aspek teknis maupun penyajian. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Semoga laporan ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan di bidang teknologi informasi dan rekayasa data.

Serang, 12 Juni 2026

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	vi
DAFTAR GAMBAR.....	vii
RINGKASAN.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan PKL/Magang.....	3
1.3.1 Tujuan Umum.....	3
1.3.2 Tujuan Khusus.....	4
1.4 Manfaat PKL/Magang.....	5
1.4.1 Manfaat Teoritis.....	5
1.4.2 Manfaat Praktis.....	5
BAB II KAJIAN PUSTAKA.....	7
2.1 Kajian Pustaka.....	7
2.1.1 Agentic AI & Retrieval-Augmented Generation (RAG).....	7
2.1.2 Knowledge Graph & Graph Database (Neo4j).....	8
2.1.3 Arsitektur Backend & API (FastAPI / Node.js).....	8
2.1.4 Database Management & ORM (Supabase & Prisma).....	9
2.1.5 Multi-Level Chunking.....	10
2.1.6 Multi-Stage Quality Gating (Semantic Gating).....	10
2.2 Kerangka Pikir Teoritis.....	12
BAB III METODE PKL/MAGANG.....	14
3.1 Lokasi PKL/Magang.....	14
3.2 Desain Pelaksanaan PKL/Magang.....	17
3.3 Penentuan Objek PKL/Magang.....	17

3.4 Metode Pengumpulan Data.....	18
3.5 Instrumen PKL/Magang.....	19
3.6 Teknik Analisis Data.....	21
BAB IV HASIL PKL/MAGANG DAN PEMBAHASAN.....	23
4.1 Deskripsi Kegiatan PKL/Magang.....	23
4.2 Permasalahan yang Ditemukan.....	24
4.3 Metode Penyelesaian Masalah.....	25
4.4 Hasil Pelaksanaan PKL/Magang.....	27
4.4.1 Analisis Kebutuhan Sistem Aplikasi Brain.....	27
4.4.2 Perancangan Arsitektur dan Fondasi Teknologi Backend.....	29
4.4.3 Spesifikasi Kontrak Komunikasi dan Antarmuka API.....	32
4.4.4 Perancangan Skema Database.....	33
4.4.5 Implementasi Multi Level Chunking & Stuffed Context.....	38
4.4.6 Strategi Prompt Engineering dan Ekstraksi Berbasis Skema.....	42
4.4.7 Rekayasa Multi-Stage Semantic Gating.....	44
4.5 Pembahasan.....	49
4.6 Pelajaran yang Diperoleh.....	52
BAB V KESIMPULAN DAN SARAN.....	54
5.1 Kesimpulan.....	54
5.2 Keterbatasan.....	54
5.3 Implikasi.....	55
5.4 Saran.....	55
DAFTAR PUSTAKA.....	57
LAMPIRAN.....	59
L.1 Biodata Peserta Magang (Muhammad Zaidan Rafat).....	59
L.2 Biodata Pembimbing Lapangan BRIN (Dr. Rio Nurtantyana).....	60
L.3 Biodata Pembimbing Universitas (Ahmad Tabrani, M.T.I.).....	64
L.4 Dokumentasi Pertemuan Progres (Progress Meetings) MBKM.....	67

DAFTAR TABEL

Tabel 4.1 Kegiatan PKL/Magang.....	13
Tabel 4.2 Kebutuhan Fungsional Sistem Backend.....	15
Tabel 4.3 Kebutuhan Non-Fungsional Sistem Backend.....	15
Tabel 4.4 Spesifikasi Teknologi dan Library	15
Tabel 4.5 Skema API Infrastruktur.....	15
Tabel 4.7 Skema Model User (Manajemen Identitas).....	16
Tabel 4.8 Skema Model Project (Kontainer Kerja).....	16
Tabel 4.9 Skema Model DocumentChunk (Gudang Vektor).....	34
Tabel 4.10 Skema Model ChatSession dan ChatMessage.....	39
Tabel 4.11 Skema Properti Identitas dan Metadata (Node :Literature).....	40
Tabel 4.12 Skema Properti 5 Pilar Ilmiah (Node :Literature).....	41
Tabel 4.13 Skema Properti Relasi Semantik (Edge)	42
Tabel 4.14 Parameter Kalibrasi Threshold Dinamis per Pilar.....	43
Tabel 4.16 Hasil Analisis Metrik Topologi Graf dengan Gephi	44

DAFTAR GAMBAR

Gambar 2.1 Bagan Alur Kerangka Pikir Pengembangan Sistem MySRE.	13
Gambar 3.1 Struktur Organisasi Pusat Riset Sains Data dan Informasi.....	15
Gambar 3.2 Alur Kerja Pusat Riset Sains Data dan Informasi.....	15
Gambar 3.3 Profil Kelompok Riset Interaksi Manusia-Komputer	16
Gambar 4.2 : ERD Postgresql (Supabase).....	34
Gambar 4.3 : Pipeline Multi-Level Chunking.....	39
Gambar 4.4 : Snippet Code Purifikasi Data.....	40
Gambar 4.5 : Anchor Density Query.....	41
Gambar 4.6 : Arsitektur Ekstraksi LLM.....	42
Gambar 4.7 : Snippet Code Prompt Ekstraksi Pilar	43
Gambar 4.8 : Snippet Code Purification Output LLM.....	44
Gambar 4.9 : Multi Stage Semantic Gating.....	45
Gambar 4.10 : Snippet Code Domain Filtering.....	48
Gambar 4.11 : Snippet Code Edge Reasoning Quality Check.....	49

RINGKASAN

Ledakan volume publikasi ilmiah global menyulitkan sintesis korelasi antar-penelitian secara manual akibat fragmentasi informasi riset dalam teks tidak terstruktur. Untuk mengatasi masalah tersebut, dikembangkan proyek *Service Agentic AI* (SAAI) melalui produk MySRE dengan fitur “*Brain*” pada Pusat Riset Sains Data dan Informasi (PR-SDI) BRIN. Sistem ini dirancang untuk mengekstraksi 5 pilar ilmiah (Latar Belakang, Metodologi, Tujuan, Gap, dan Arahana Masa Depan) dari dokumen PDF mentah dan merekonstruksinya menjadi *Knowledge Graph* terstruktur pada Neo4j.

Pelaksanaan magang menempatkan penulis sebagai *Backend & Knowledge Data Engineer*. Metode pelaksanaan didasarkan pada siklus prototyping untuk merancang arsitektur *decoupled asinkronus* berbasis FastAPI. Sistem ini mengintegrasikan basis data hibrida antara Supabase (relasional dan vektor) dengan Neo4j (graf). Pemrosesan data dioptimalkan menggunakan teknik Multi-Level Chunking (*Small-to-Big Retrieval*) dan empat lapis penjamin mutu Multi-stage Semantic Gating. Evaluasi kualitas relasi semantik graf diukur secara kuantitatif menggunakan perangkat lunak Gephi untuk membandingkan metrik topologi antar-versi sistem.

Hasil pelaksanaan magang berupa infrastruktur backend asinkronus yang stabil memproses dokumen beraliran berat tanpa menghambat antarmuka pengguna (dilacak via SSE), serta terbentuknya struktur graf yang lebih terkoneksi dengan peningkatan kerapatan graf (*graph density*) dari 0,305 ke 0,514 pada versi SRE-v.2. Selain peningkatan keterampilan teknis dalam pengelolaan basis data graf dan orkestrasi kecerdasan buatan, penulis memperoleh wawasan mengenai iklim kerja riset profesional serta penentuan spesialisasi arah karir sebagai Backend Engineer dan Data Engineer.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Volume publikasi ilmiah global telah mencapai titik jenuh yang melampaui kapasitas kognitif manusia dalam mengelola informasi. Data *National Science Board* (2024) mencatat produksi artikel sains dan teknik dunia menyentuh angka 3,3 juta dokumen pada tahun 2022, dengan akselerasi pertumbuhan sebesar 59% dalam satu dekade terakhir [1]. Ledakan data ini bukan sekadar statistik, melainkan sebuah hambatan struktural yang menghalangi efisiensi riset. Peneliti menghadapi tantangan informasi berupa melimpahnya literatur yang justru mempersulit identifikasi kebaruan (*novelty*) dan pemetaan kaitan logis antar-studi. Proses tinjauan literatur konvensional yang mengandalkan pembacaan manual terhadap teks tidak terstruktur menjadi metode yang kurang efisien dan tidak skalabel bagi percepatan inovasi nasional.

Menghadapi krisis skalabilitas tersebut, arsitektur kecerdasan artifisial perlu diarahkan dari model bahasa generatif pasif menuju sistem otonom yang lebih deterministik. Keterbatasan model konvensional berakar pada alur kerja yang statis dan linear, sehingga memiliki risiko kegagalan penalaran multi-langkah yang kompleks dan rentan menghasilkan halusinasi pada dokumen teknis [2]. Tantangan ini mendesak penerapan *Agentic AI*, yaitu paradigma sistem yang memperkenalkan lapisan kontrol eksplisit untuk melakukan penalaran aktif, perencanaan tugas, dan penggunaan alat (*tool use*) secara mandiri. Dalam konteks rekayasa data, pendekatan agen digunakan untuk mengonversi narasi ilmiah yang tersebar menjadi *Knowledge Graph* yang terstandarisasi. Melalui representasi entitas dan relasi semantik yang presisi, sistem ini dirancang untuk memastikan bahwa setiap temuan riset tidak lagi berdiri sebagai fragmen data yang terisolasi, melainkan sebagai bagian dari jaringan pengetahuan yang dapat diverifikasi.

Implementasi teknologi *Agentic AI* and *Knowledge Graph* tersebut dilaksanakan di Pusat Riset Sains Data dan Informasi (PR SDI), Badan Riset dan Inovasi Nasional (BRIN). Sebagai salah satu unit kerja riset nasional, PR-SDI melalui kelompok riset *Human Computer Interaction - Visualization* memiliki fokus untuk mentransformasi cara peneliti berinteraksi dengan informasi ilmiah yang kompleks. Melalui integrasi program Merdeka Belajar Kampus Merdeka (MBKM), visi tersebut diwujudkan menjadi aksi rekayasa yang konkret. Program ini berfungsi sebagai medium untuk menyelaraskan perkembangan teori akademik dengan kebutuhan infrastruktur riset nasional. Kolaborasi ini menjadi fondasi bagi pembangunan *Smart Research Environment* (SRE) di Indonesia.

Namun, hambatan teknis yang dihadapi dalam mewujudkan hal tersebut terletak pada fragmentasi informasi di dalam artikel ilmiah. Pengetahuan riset saat ini tersimpan di dalam narasi teks tidak terstruktur, yang diidentifikasi sebagai pelebaran jarak (*widening gap*) dan menjadi hambatan bagi pembentukan sistem pengetahuan [3]. Kondisi ini menyulitkan identifikasi kaitan logis antara Latar Belakang, Metodologi, Tujuan, *Gap* (Kesenjangan), hingga Arahana Masa Depan antar-dokumen secara otomatis. Ketidadaan mekanisme sintesis semantik menyebabkan proses pencarian korelasi penelitian tetap menjadi proses manual yang repetitif dan rentan terhadap ketidakakuratan data. Tantangan tersebut berfokus pada hambatan dalam memetakan posisi sebuah penelitian di dalam peta pengetahuan (*knowledge map*) yang lebih luas secara konsisten.

Guna mengatasi kendala tersebut, dikembangkan proyek *Service Agentic AI* (SAAI) melalui produk *My Smart Research Environment* (MySRE) dengan fokus utama pada fitur "*Brain*". Sistem ini dirancang untuk melakukan dekomposisi terhadap narasi artikel ilmiah dan merekonstruksinya menjadi *Knowledge Graph* yang merepresentasikan keterkaitan antar-pilar riset secara terstruktur. Dalam kerangka kerja ini, peran *Backend & Knowledge Data Engineer* diperlukan untuk membangun fondasi arsitektur yang menangani alur pemrosesan data mulai dari *ingestion*, *smart chunking*, hingga pemodelan

relasi semantik yang stabil. Dengan infrastruktur *backend* tersebut, diharapkan aplikasi “*Brain*” dapat mendukung proses asimilasi pengetahuan riset secara terstruktur pada instansi terkait.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka rumusan masalah dalam pelaksanaan kegiatan magang ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun arsitektur *backend* terdistribusi yang asinkron menggunakan FastAPI untuk memproses ekstraksi teks ilmiah beraliran berat (*heavy-duty I/O*) tanpa memblokir antarmuka pengguna (*UI blocking*)?
2. Bagaimana memodelkan skema *Knowledge Graph* pada basis data Neo4j untuk merepresentasikan keterkaitan semantik dari lima pilar artikel ilmiah (Latar Belakang, Metodologi, Tujuan, *Gap*, dan Arahana Masa Depan) secara deterministik?
3. Bagaimana mengimplementasikan metode *Multi-Level Chunking* pada *pipeline* pemrosesan teks untuk mempertahankan hierarki informasi struktural (*metadata header*) saat dokumen dipecah?
4. Bagaimana merancang mekanisme gerbang validasi bertingkat (*Multi-Stage Quality Gating*) untuk menekan risiko halusinasi dan menjamin integritas relasi semantik yang terbentuk pada *Knowledge Graph*?
5. Bagaimana membangun mekanisme integrasi basis data hibrida antara basis data relasional-vektor (Supabase) dan basis data graf (Neo4j) yang mendukung operasi atomik sinkron seperti fungsi penghapusan menyeluruh (*Double-Kill Deletion*)?

1.3 Tujuan PKL/Magang

1.3.1 Tujuan Umum

Tujuan umum dari pelaksanaan Praktik Kerja Lapangan (PKL)/Magang di Badan Riset dan Inovasi Nasional (BRIN) adalah sebagai berikut:

1. Memperoleh pengalaman kerja secara langsung di lingkungan instansi riset pemerintahan guna memahami iklim kerja profesional di bidang teknologi informasi dan sains data.
2. Menerapkan teori akademik dan keterampilan rekayasa perangkat lunak yang diperoleh selama perkuliahan pada Program Studi Informatika ke dalam proyek nyata berskala nasional.
3. Menyelaraskan visi strategis Pusat Riset Sains Data dan Informasi (PR SDI) BRIN, khususnya dalam riset interaksi manusia-komputer dan visualisasi, dengan kebutuhan rekayasa teknologi informasi yang aplikatif.
4. Membangun kompetensi interpersonal, kemampuan berkolaborasi dalam tim riset multi-disiplin, serta pemahaman terhadap standar profesionalisme kerja.

1.3.2 Tujuan Khusus

1. Merancang dan membangun arsitektur *backend* yang asinkron menggunakan *framework* FastAPI dan Python untuk mendukung kebutuhan pemrosesan data beraliran berat (*heavy-duty I/O*) pada sistem *My Smart Research Environment* (MySRE).
2. Memodelkan skema basis data graf (*Knowledge Graph*) pada Neo4j untuk merepresentasikan relasi semantik lima pilar utama literatur ilmiah (Latar Belakang, Metodologi, Tujuan, *Gap*, dan Arahana Masa Depan) secara terstruktur.
3. Mengembangkan modul ekstraksi dokumen PDF mentah serta menerapkan teknik *Multi-Level Chunking* untuk membagi teks tanpa kehilangan konteks hierarki informasi judul (*metadata header*).
4. Menerapkan mekanisme validasi bertingkat (*Multi-Stage Quality Gating*) pada alur kerja kecerdasan buatan (*AI Agent*) untuk memastikan ketepatan relasi yang terbentuk dan menekan risiko halusinasi data.

5. Membangun integrasi basis data hibrida antara basis data relasional-vektor (Supabase) dan basis data graf (Neo4j) yang mendukung operasi atomik sinkron seperti fungsi penghapusan menyeluruh (*Double-Kill Deletion*).

1.4 Manfaat PKL/Magang

1.4.1 Manfaat Teoritis

Laporan hasil PKL/Magang ini diharapkan dapat memberikan kontribusi akademik bagi pengembangan ilmu pengetahuan, khususnya di bidang rekayasa data (*Data Engineering*) dan teknologi *Semantic Web*. Secara spesifik, laporan ini dapat dijadikan sebagai salah satu referensi teoretis mengenai pemanfaatan arsitektur *Agentic AI* dalam memproses dokumen teks tidak terstruktur menjadi representasi basis data graf (*Knowledge Graph*). Selain itu, kajian ini memberikan landasan kepustakaan mengenai efektivitas metode pemecahan dokumen bertingkat (*Multi-Level Chunking*) dan gerbang semantik (*Semantic Gating*) untuk menjaga integritas data relasional graf.

1.4.2 Manfaat Praktis

Bagi Mahasiswa

1. Memberikan kesempatan nyata untuk mempraktikkan peran sebagai *Backend & Knowledge Data Engineer* di lingkungan riset profesional.
2. Meningkatkan kompetensi teknis dalam pemanfaatan teknologi basis data graf (Neo4j), pengelolaan basis data relasional-vektor (Supabase), penggunaan ORM Prisma, serta pengembangan API asinkron dengan FastAPI.
3. Menumbuhkan kemampuan analisis kritis dalam memetakan kesenjangan antara teori akademis dengan batasan teknis riil pada *codebase* proyek.

Bagi Pusat Riset Sains Data dan Informasi (PR-SDI) BRIN

1. Menyediakan infrastruktur *backend* SAAI yang andal dan fungsional untuk mendukung fitur “*Brain*” pada aplikasi MySRE.

2. Membantu para peneliti di PR SDI BRIN dalam melakukan tinjauan literatur ilmiah secara otomatis dan terstruktur, sehingga mengurangi waktu pencarian korelasi riset secara manual.

Bagi Program Studi dan Universitas

1. Memperkuat jaringan kerja sama riset dan kemitraan akademik antara Universitas Islam Negeri (UIN) Sultan Maulana Hasanuddin Banten dengan Badan Riset dan Inovasi Nasional (BRIN).
2. Menjadi bahan referensi serta studi kasus nyata bagi pengembangan kurikulum Program Studi Informatika, khususnya dalam mata kuliah sistem basis data, pemrograman web, sains data, dan teknologi kecerdasan buatan.

BAB II

KAJIAN PUSTAKA

2.1 Kajian Pustaka

2.1.1 Agentic AI & Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) merupakan metode yang dirancang untuk mengoptimalkan kinerja *Large Language Model* (LLM) dengan mengintegrasikan dokumen eksternal sebagai basis pengetahuan tambahan guna menekan risiko halusinasi data. Pada arsitektur RAG konvensional (*Classic RAG*), alur kerja pencarian dokumen berlangsung secara statis dan linear, di mana sistem hanya melakukan pencocokan vektor kueri terhadap indeks dokumen tanpa adanya pemantauan alur penalaran [4]. Keterbatasan ini menyebabkan sistem konvensional rentan mengalami kegagalan saat menangani penalaran multi-langkah yang kompleks (*Multi-hop Reasoning*) pada dokumen ilmiah yang padat informasi.

Sebagai solusi atas batasan tersebut, dikembangkan paradigma *Agentic AI* yang memperkenalkan lapisan kontrol kognitif aktif. Melalui integrasi agen otonom, sistem memiliki kemampuan untuk melakukan perencanaan tugas, penggunaan alat (*tool use*), dan evaluasi mandiri secara dinamis [2]. Dalam arsitektur pencarian, transisi ini diwujudkan melalui pendekatan *Small-to-Big Retrieval* (atau *Parent-Child Retrieval*). Berdasarkan pengujian kerangka kerja ARAGOG oleh Eibich et al. (2024), pemisahan granularitas teks antara fase pencarian (menggunakan potongan kecil) dan fase sintesis penalaran (menggunakan konteks besar) secara statistik terbukti meningkatkan relevansi informasi yang dikirimkan ke LLM [5]. Meskipun demikian, implementasi praktis di lingkungan produksi tetap dibatasi oleh kapasitas jendela konteks (*context window*) LLM dan *rate limits* dari penyedia API eksternal, sehingga asumsi teoretis mengenai penggabungan konteks tanpa batas perlu dievaluasi secara ketat [6].

2.1.2 Knowledge Graph & Graph Database (Neo4j)

Knowledge Graph (Graf Pengetahuan) adalah struktur basis data non-relasional yang merepresentasikan pengetahuan dalam bentuk jaringan entitas (*Node*) dan hubungan semantik (*Edge*) yang saling terhubung. Berbeda dengan basis data relasional konvensional yang mengandalkan operasi *Join* tabel yang intensif secara komputasi, basis data graf memproses relasi antar-entitas sebagai data tingkat pertama (*first-class citizens*) yang disimpan secara berdekatan pada memori fisik [7]. Hal ini memberikan keunggulan performa dalam penelusuran relasi multi-tingkat (*multi-depth traversal*) yang sangat krusial dalam memetakan benang merah literatur ilmiah.

Neo4j merupakan salah satu platform basis data graf asli (*native graph database*) terkemuka yang menggunakan bahasa kueri Cypher untuk mengeksekusi operasi data graf. Sistem Neo4j mendukung indeks pencarian vektor secara *native*, yang memungkinkan pencarian kemiripan kosinus (*Cosine Similarity*) langsung di dalam ruang graf [8]. Representasi artikel ilmiah pada Neo4j dimodelkan sebagai *node* dengan properti spesifik yang merangkum metadata dokumen dan hasil ekstraksi lima pilar ilmiah (Latar Belakang, Metodologi, Tujuan, *Gap*, dan Arahana Masa Depan), sedangkan relasi keterkaitan semantik antar-studi dimodelkan sebagai *edge* berarah yang membawa nilai pembobotan statistik hasil validasi sistem.

2.1.3 Arsitektur Backend & API (FastAPI / Node.js)

Perancangan arsitektur sistem informasi modern berskala besar mengadopsi prinsip pemisahan tanggung jawab (*separation of concerns*) melalui model *decoupled client-server*. Dalam ekosistem sistem kecerdasan artifisial, pemisahan ini diperlukan untuk mengisolasi antarmuka pengguna (klien) dari beban komputasi masif pemrosesan bahasa alami (NLP) di sisi *server* [9]. Arsitektur terdistribusi ini

memastikan ketersediaan antarmuka tetap terjaga meskipun *backend* sedang mengeksekusi proses inferensi LLM yang memakan waktu lama.

FastAPI adalah *framework* web asinkron (*Asynchronous Server Gateway Interface / ASGI*) berbasis Python yang berkinerja tinggi, dibangun di atas Starlette dan Uvicorn. FastAPI memanfaatkan fitur *type hinting* Python untuk melakukan validasi data secara otomatis serta mendukung eksekusi fungsi non-blocking (*async/await*) yang sangat efisien untuk menangani beban I/O berat [10]. Guna memfasilitasi pelaporan progres pemrosesan dokumen secara *real-time* kepada klien, *backend* mengimplementasikan pustaka `'sse-starlette'` untuk menyiarkan aliran data satu arah melalui saluran *Server-Sent Events* (SSE), sementara SlowAPI disematkan sebagai pembatas laju permintaan (*rate limiter*) untuk menjaga stabilitas operasional *server*.

2.1.4 Database Management & ORM (Supabase & Prisma)

Manajemen penyimpanan data pada sistem cerdas memerlukan pendekatan hibrida (*Hybrid Data Store*). Supabase merupakan platform *Backend-as-a-Service* (BaaS) berbasis PostgreSQL yang menyediakan dukungan penyimpanan berkas (*storage*) serta ekstensi `'pgvector'` untuk komputasi dan penyimpanan vektor embedding [11]. Kombinasi ini memungkinkan Supabase berfungsi sebagai tempat penyimpanan dokumen fisik PDF dan gudang penyimpanan potongan teks (*document chunks*) beserta koordinat vektornya.

Sebagai jembatan antara kode aplikasi *backend* dengan basis data relasional, digunakan Prisma ORM (*Object-Relational Mapping*). Prisma menyediakan lapisan abstraksi data yang aman dari kesalahan tipe (*type-safe query builder*), memfasilitasi skema migrasi terstruktur, serta mengoptimalkan manajemen koneksi melalui metode *connection pooling* [12]. Untuk mengamankan data pengguna di dalam basis data hibrida tersebut, sistem menerapkan batasan *multi-tenancy* yang ketat melalui perhitungan *composite ID* berbasis algoritma enkripsi MD5,

memastikan bahwa setiap data dan graf terisolasi secara mutlak di dalam lingkup ruang kerja (*project*) masing-masing.

2.1.5 Multi-Level Chunking

Akurasi pencarian dokumen pada sistem RAG sangat dipengaruhi oleh kualitas pemotongan teks (*chunking*). Berdasarkan konsep *Small-to-Big Retrieval*, teks ilmiah tidak dapat dipotong secara seragam karena memaksakan teks panjang ke dalam vektor berdimensi tetap akan memicu hilangnya detail informasi (*information loss*) [13]. Sistem mengatasi masalah ini melalui teknik *Multi-Level Chunking*, yaitu membagi dokumen menjadi dua hierarkis: *Parent Chunks* (berukuran ± 1500 karakter) yang mempertahankan konteks narasi lengkap untuk dikirimkan sebagai input LLM, dan *Child Chunks* (berukuran ± 200 karakter) berupa pecahan proposisi padat yang diubah menjadi vektor *embedding* untuk fase pencarian kemiripan.

Vektorisasi *Child Chunks* dieksekusi secara lokal menggunakan model *intfloat/multilingual-e5-small*. Model mE5 menggunakan pendekatan *asymmetric contrastive learning*, yang mengharuskan penambahan instruksi awalan (*prefix*) khusus, yaitu *"query:"* untuk teks pencarian dan *"passage:"* untuk teks dokumen sumber, guna mempertahankan keselarasan dimensi vektor semantik bahasa campuran [14]. Selain itu, sebelum proses vektorisasi, sistem melakukan purifikasi data (*JIT Data Purification*) untuk membuang karakter sampah hasil OCR dan daftar pustaka guna menghindari ekspansi entropi ruang vektor yang dapat memicu pelemahan atensi model (*Attention Dilution*) [15].

2.1.6 Multi-Stage Quality Gating (Semantic Gating)

Pencarian vektor konvensional berbasis algoritma *K-Nearest Neighbor* (K-NN) memiliki kelemahan karena memaksakan pembentukan relasi terhadap sejumlah *top_k* dokumen terdekat tanpa memedulikan tingkat relevansi riil. Hal ini memicu terbentuknya relasi palsu (*false positives*) yang menghasilkan visualisasi graf yang rumit dan tidak bermakna (*hairball graph*) [16]. Untuk mengatasi masalah

tersebut, dirancang arsitektur penyaringan bertahap (*Multi-Stage Quality Gating*) sebagai berikut:

1. Gerbang Statistik (*Three Sigma Rule*): Mengubah pencarian K-NN menjadi Graf Lingkungan- ϵ (ϵ -neighborhood graph). Ambang batas skor kemiripan ditentukan secara dinamis berdasarkan rumus statistika deteksi anomali:

$$Threshold_{pillar} = (\mu + 3\sigma) \times Multiplier_{pillar}$$

di mana μ adalah rata-rata skor kemiripan noise dan σ adalah standar deviasinya. Parameter *Multiplier* diberikan pengali lebih tinggi pada pilar dengan kepadatan informasi rendah (seperti *Gap* sebesar 1.15) guna memperketat interval kepercayaan [17].

2. *Semantic Coherence Gate*: Mengevaluasi keselarasan lintas pilar penelitian. Merujuk pada konsep faset dokumen ilmiah oleh Mysore et al. (2022), relasi hanya akan terbentuk jika memenuhi konsensus minimum jumlah pilar yang selaras dan mematuhi dependensi logis Teori CARS (*Creating a Research Space*) John Swales [18]. Misalnya, kemiripan metode secara terisolasi akan ditolak jika tidak disertai kemiripan tujuan atau gap penelitian.
3. *Ontological Gate (Dynamic Domain Filtering)*: Memanfaatkan LLM sebagai validator ontologi *zero-shot* untuk membandingkan kata kunci penelitian dari kedua dokumen. Langkah ini krusial untuk mencegah tabrakan semantik lintas disiplin ilmu (*Cross-Domain Semantic Collision*) [19].
4. *Rule-Based Reasoning Gate*: Melakukan verifikasi runtime terhadap kualitas teks argumen keterkaitan (*reasoning*) yang dihasilkan LLM menggunakan ekspresi reguler (*Regex*) dan pengecekan fakta atomik guna menggugurkan penalaran yang bersifat basa-basi (*generic reasoning*).
5. *Armor Sanitasi Pydantic*: Memanfaatkan pustaka *'Pydantic'* dengan dekorator *'BeforeValidator'* untuk melakukan sanitasi terhadap respons JSON non-deterministik dari LLM, menyaring halusinasi

teks (seperti "tidak disebutkan dalam teks" atau "None") menjadi nilai *string* kosong yang aman bagi basis data.

2.2 Kerangka Pikir Teoritis

Kerangka pikir teoritis dalam pengembangan sistem *My Smart Research Environment* (MySRE) dirancang untuk menyelesaikan tantangan riset secara terstruktur melalui alur penelitian dan pengembangan (*Research and Development* / R&D) berbasis bukti (*literature-driven*). Kerangka ini mengintegrasikan enam fase perkembangan sistem yang sistematis, dimulai dari audit fungsional hingga finalisasi komputasi *backend* yang asinkron.

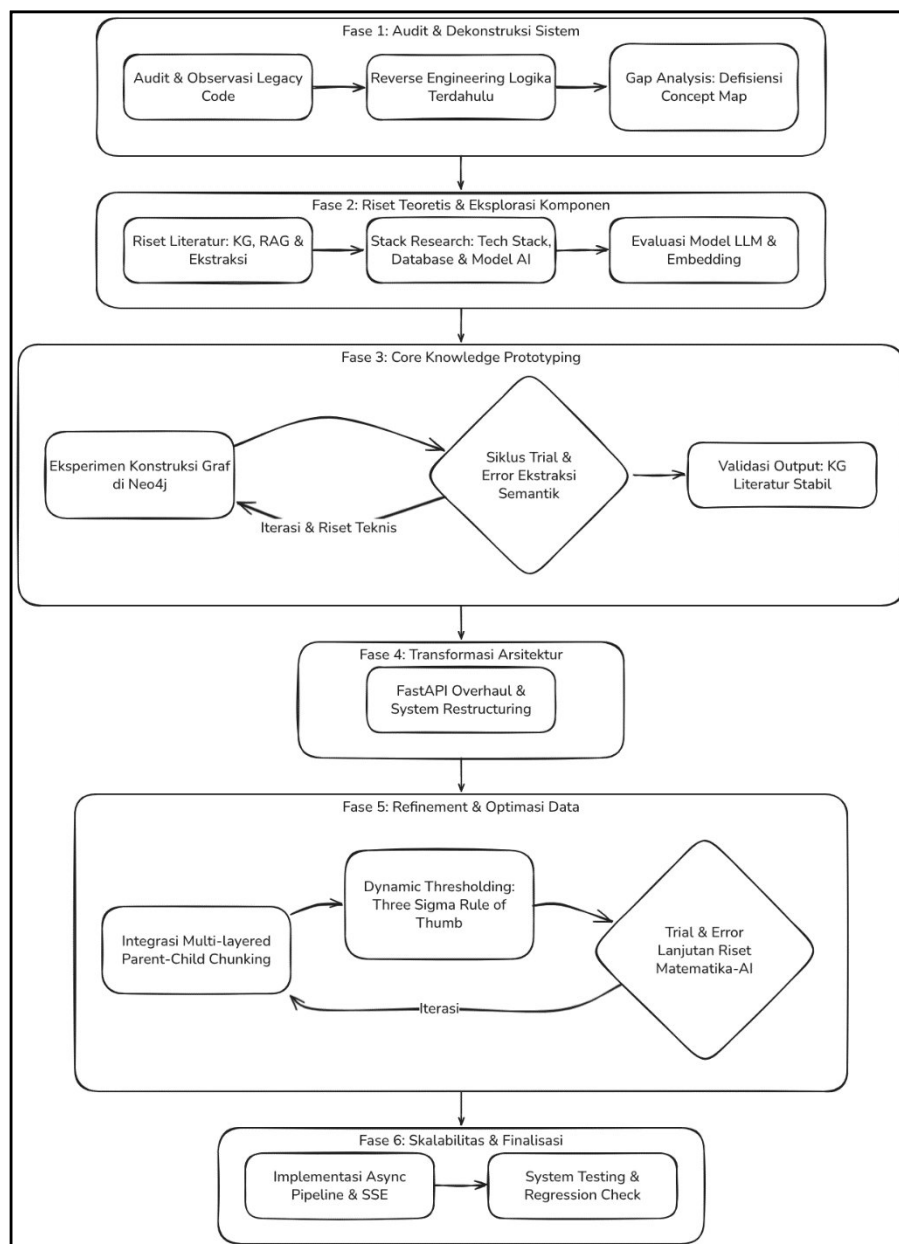
Fase pertama diawali dengan audit dan dekonstruksi sistem *legacy* untuk mengidentifikasi celah keamanan, non-persistensi pemetaan konsep terdahulu, serta keterbatasan kapasitas LLM konvensional. Temuan ini ditindaklanjuti pada fase kedua dengan riset kepustakaan mengenai arsitektur *Agentic AI*, *Retrieval-Augmented Generation* (RAG) [2], [5], basis data graf [7], dan pustaka pemrograman Python asinkron. Riset komponen ini membentuk landasan ilmiah untuk menentukan spesifikasi *stack* teknologi yang akan diimplementasikan.

Pada fase ketiga, penulis mengeksekusi pembuatan prototipe inti (*core prototyping*) di Neo4j untuk memetakan keterkaitan lima pilar ilmiah (Latar Belakang, Metodologi, Tujuan, *Gap*, dan Arahana Masa Depan). Fase ini berjalan secara iteratif melalui siklus uji coba (*trial and error*) guna menyelaraskan sintaksis *kueri Cypher*. Setelah prototipe dinyatakan stabil, fase keempat dilakukan melalui restrukturisasi arsitektur secara menyeluruh (*architectural overhaul*) menggunakan FastAPI untuk mentransformasikan alur kerja sinkron menjadi asinkron.

Fase kelima difokuskan pada tahap penyempurnaan (*refinement*) dan optimasi data semantik. Pada fase ini, metode *Multi-Level Chunking* diintegrasikan bersama algoritma *Multi-Stage Quality Gating* yang memanfaatkan parameter statistik *Three Sigma Rule* untuk menghitung ambang batas dinamis K-NN. Terakhir, fase keenam diselesaikan dengan

menerapkan pemrosesan asinkron (*background tasks*) dan *Server-Sent Events* (SSE) untuk transmisi data *real-time*, ditutup dengan pengujian regresi (*regression check*) guna memastikan keandalan sistem secara menyeluruh.

Visualisasi alur kerja kerangka pikir penelitian dan pengembangan sistem *MySRE Backend* ini digambarkan secara skematis pada Gambar 2.1.



Gambar 2.1 Bagan Alur Kerangka Pikir Pengembangan Sistem *MySRE*.

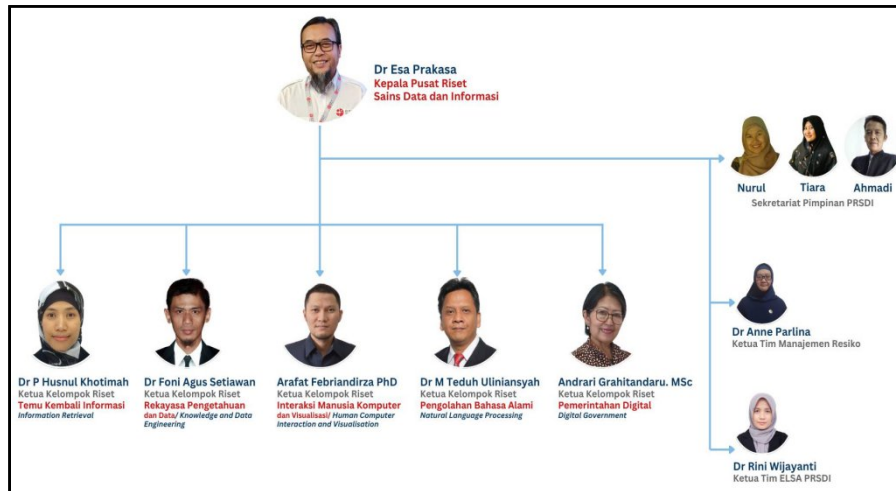
BAB III

METODE PKL/MAGANG

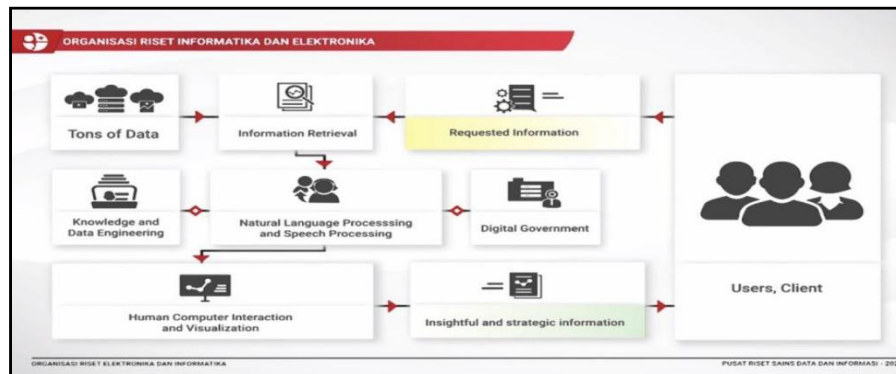
3.1 Lokasi PKL/Magang

Lokasi kegiatan Praktik Kerja Lapangan (PKL)/Magang MBKM penulis bertempat di Pusat Riset Sains Data dan Informasi (PR-SDI), yang merupakan salah satu unit kerja di bawah Organisasi Riset Elektronika dan Informatika (OREI), Badan Riset dan Inovasi Nasional (BRIN) [23]. Di bawah kepemimpinan Dr. Esa Prakasa, M.T. selaku Kepala Pusat, PRSDI beroperasi di Gedung BASIC Lantai 5, Kawasan Sains dan Teknologi (KST) Samaun Samadikun, Bandung. Secara kelembagaan, tugas dan kewenangan PRSDI diatur melalui Peraturan BRIN Nomor 10 Tahun 2022. Regulasi ini memberikan mandat kepada PRSDI untuk memimpin penelitian teknis, menyelenggarakan pengkajian, serta merumuskan teknologi di bidang sains data dan informasi [23].

Dalam mengeksekusi tugas tersebut, PRSDI menjalankan lima fungsi kerja utama. Fungsi ini mencakup pelaksanaan penelitian teknis, penyiapan rekomendasi ilmiah untuk pimpinan, penyediaan bimbingan teknis, pengawasan berkala, serta pengelolaan administrasi umum. Penyelenggaraan fungsi ini ditunjang oleh sistem pemantauan, evaluasi, dan penyusunan laporan untuk memastikan hasil riset memenuhi standar mutu nasional. Struktur organisasi dan bagan alur kerja internal PRSDI digambarkan secara terperinci pada Gambar 3.1 dan Gambar 3.2.



Gambar 3.1 Struktur Organisasi Pusat Riset Sains Data dan Informasi



Gambar 3.2 Alur Kerja Pusat Riset Sains Data dan Informasi

Guna menjalankan fungsi riset secara spesifik, PRSDI membagi keahlian ilmiah sivitasnya ke dalam beberapa kelompok riset. Penulis ditempatkan pada Kelompok Riset Interaksi Manusia-Komputer dan Visualisasi (*Human-Computer Interaction and Visualisation*). Kelompok ini dipimpin oleh Arafat Febriandirza, Ph.D. dan beranggotakan sembilan peneliti, yaitu Rio Nurtantyana, S.Pd., M.Pd., M.Sc., Ph.D., Abdurrahman Prasetyadi, M.P., Rumadi, S.T., M.T., Halim Hamadi, M.Sc., Aang Gunawan Sutyawan, S.Kom., M.T., Christine Cecylia Munthe, S.Kom., M.T.I., Muhammad Yudhi Rezaldi, Ph.D., Ridwan Suhud, S.Kom., M.T., dan Ana Hadiana. Profil kelompok riset ini ditunjukkan pada Gambar 3.3.



*Gambar 3.3 Profil Kelompok Riset Interaksi Manusia-Komputer
dan Visualisasi*

Kelompok riset ini memiliki fokus keilmuan yang luas untuk merancang antarmuka sistem yang andal. Bidang kajian utama meliputi interaksi manusia-komputer, simulasi sistem, serta perancangan antarmuka dan pengalaman pengguna (*User Interface/User Experience*). Selain itu, kajian dikembangkan pada area interaksi untuk pendidikan dan arsitektur sistem informasi. Seluruh penelitian didukung oleh kompetensi tim di bidang sains data, ilmu informasi, dan teknologi kecerdasan buatan.

Melalui integrasi keilmuan tersebut, kelompok riset ini bertujuan mengubah rumitnya data teknis menjadi representasi spasial dan visual yang mudah dipahami. Penerapannya diarahkan pada penyediaan solusi interaktif untuk sistem berskala besar, seperti pemodelan visual sistem kendali (*control system*) dan simulasi fasilitas pembangkit nuklir. Eksplorasi ini didukung oleh penggunaan teknologi imersif, seperti *Augmented Reality* (AR) dan *Virtual Reality* (VR), dengan tujuan akhir menciptakan interaksi manusia dengan mesin yang aman, presisi, dan efisien.

3.2 Desain Pelaksanaan PKL/Magang

Desain pelaksanaan magang ini menggunakan metode purwarupa (*prototyping*). Metode ini diterapkan untuk memetakan dokumen ilmiah yang tidak terstruktur menjadi representasi basis data graf (*knowledge graph*) yang memiliki kaitan semantik terstruktur. Melalui pembuatan purwarupa secara iteratif, penulis dapat menganalisis permasalahan ekstraksi teks secara langsung serta merumuskan kebutuhan teknis sistem.

Proses diawali dengan mengaudit kode program pemetaan konsep lama yang belum mampu menyimpan data secara persisten. Untuk melengkapi data kebutuhan sistem, penulis berdiskusi dengan pembimbing lapangan dan tim peneliti Pusat Riset Sains Data dan Informasi (PR SDI) BRIN mengenai model RAG (*Retrieval-Augmented Generation*) dan basis data Neo4j. Informasi teknis juga dikumpulkan melalui penelusuran dokumentasi resmi pustaka pemrograman FastAPI dan Supabase.

Setelah kebutuhan sistem dipetakan, penulis membangun *backend* asinkron FastAPI dan basis data graf Neo4j untuk menjalankan aplikasi MySRE. Penulis membagi teks PDF menjadi potongan hierarkis (*Multi-Level Chunking*) dan menyaring hubungan semantik antar-literatur secara bertahap (*Multi-Stage Quality Gating*). Pada tahap akhir, keandalan sistem diuji melalui pengujian fungsional API, latensi transmisi (*latency*), serta biaya penggunaan token (*cost*).

Metode *prototyping* dipilih karena pengembangan sistem kecerdasan artifisial (*AI*) memerlukan eksperimen berulang untuk menguji ketepatan relasi antar-entitas. Pendekatan ini memudahkan penulis menyempurnakan parameter statistik *Three Sigma Rule* serta menyesuaikan dependensi logis kalimat akademik sebelum sistem diterapkan penuh.

3.3 Penentuan Objek PKL/Magang

Objek PKL/Magang dalam kegiatan ini adalah infrastruktur *backend* dan basis data hibrida pada sistem *Service Agentic AI* (SAAI) untuk fitur “*Brain*” aplikasi *My Smart Research Environment* (MySRE). Batasan objek yang

diteliti meliputi pipa pemrosesan data (*ingestion pipeline*) dokumen PDF, pembuatan titik akhir (*endpoints*) API asinkron berbasis FastAPI, skema data relasional Supabase, serta arsitektur basis data graf Neo4j yang mengelola penautan semantik literatur ilmiah.

Pemilihan objek tersebut didasarkan pada hasil observasi dan *reverse engineering* pada *legacy code* yang menunjukkan adanya inefisiensi dalam proses sintesis literatur ilmiah secara manual. Selain itu, sistem pemetaan konsep sebelumnya masih bersifat non-persisten sehingga tidak mampu menyimpan hasil graf secara permanen, ditambah dengan tingginya risiko halusinasi data ketika teks mentah langsung dikirimkan ke model bahasa besar.

Berdasarkan kondisi tersebut, penulis melakukan perancangan dan pengembangan modul *backend* terintegrasi yang memotong dokumen PDF secara hierarkis menggunakan metode *Multi-Level Chunking*. Penulis juga membangun sistem penyaringan *Multi-Stage Quality Gating* untuk memvalidasi kelayakan hubungan semantik lintas pilar ilmiah (Latar Belakang, Metodologi, Tujuan, *Gap*, dan Arahana Masa Depan) sebelum disimpan secara persisten ke dalam basis data graf Neo4j, lengkap dengan pengamanan *multi-tenant* dan sinkronisasi penghapusan data.

3.4 Metode Pengumpulan Data

Dalam pelaksanaan PKL/Magang ini, metode pengumpulan data yang digunakan untuk melengkapi kebutuhan pengembangan sistem meliputi tiga pendekatan utama, yaitu:

a. Observasi

Observasi dilakukan dengan cara mengamati dan melakukan *reverse engineering* terhadap alur logika pemrosesan dokumen (*ingestion pipeline*) serta struktur pilar literatur ilmiah pada kode program (*source code*) sistem yang sudah ada. Langkah ini diambil untuk memetakan kelemahan teknis, mengidentifikasi redundansi data, serta mendeteksi penyebab tidak persistennya graf pengetahuan pada sistem sebelumnya.

b. Wawancara

Wawancara dilaksanakan melalui diskusi terarah secara langsung bersama pembimbing lapangan, Rio Nurtantyana, S.Pd., M.Pd., M.Sc., Ph.D., serta para peneliti di kelompok riset *Human-Computer Interaction and Visualisation* PR SDI BRIN. Kegiatan ini bertujuan untuk merumuskan batasan desain arsitektur sistem, memvalidasi aturan dependensi semantik pilar ilmiah, serta menyepakati parameter kontrol kualitas yang akan diterapkan pada modul *gating*.

c. Dokumentasi

Dokumentasi dilakukan dengan mengumpulkan data pendukung berupa berkas-berkas teknis, spesifikasi API, serta catatan kemajuan pengembangan sistem. Selain dokumen teks, penulis mengabadikan foto kegiatan sebagai bukti fisik pelaksanaan magang, seperti dokumentasi pertemuan rutin mingguan (*progress meeting*) dan dokumentasi kegiatan demonstrasi produk bersama pakar internasional.

3.5 Instrumen PKL/Magang

Instrumen yang digunakan selama pelaksanaan PKL/Magang berfungsi untuk mendukung kelancaran proses pengumpulan data, perancangan sistem, hingga pengujian program. Instrumen tersebut terdiri atas perangkat keras dan perangkat lunak sebagai berikut:

a. Laptop ThinkPad T480

Digunakan sebagai perangkat keras utama untuk merancang arsitektur sistem, menulis kode program, dan menguji database secara lokal. Laptop ini memiliki spesifikasi prosesor Intel Core i5 generasi ke-8, memori RAM sebesar 20 GB, dan media penyimpanan SSD 512 GB.

b. Buku Catatan

Digunakan untuk menulis hasil observasi lapangan secara manual, rangkuman diskusi tim, serta catatan penting selama proses pelaksanaan magang.

c. Kamera atau Smartphone

Digunakan untuk mendokumentasikan bukti fisik kegiatan magang, seperti foto rapat progres mingguan dan kegiatan demonstrasi sistem.

d. Sistem Operasi Linux Ubuntu

Digunakan sebagai sistem operasi utama untuk menjalankan lingkungan pengembangan aplikasi, memfasilitasi perintah terminal, serta menguji paket program.

e. Visual Studio Code (VS Code)

Digunakan sebagai editor teks utama untuk menulis, menyunting, dan menata baris kode program *backend*.

f. GitHub

Digunakan sebagai platform kolaborasi tim, penyimpanan repositori kode program online, dan sistem pengontrol versi (*version control system*).

g. Python

Digunakan sebagai bahasa pemrograman utama untuk menyusun algoritma pemrosesan teks, membagi dokumen PDF, dan membangun logika *backend*.

h. FastAPI

Digunakan sebagai kerangka kerja (*framework*) pemrograman asinkron berbasis Python untuk merancang antarmuka pemrograman aplikasi (*REST API*).

i. Neo4j

Digunakan sebagai sistem manajemen basis data graf (*Graph Database Management System*) untuk menyimpan, memetakan, dan menghubungkan entitas semantik literatur secara persisten.

j. Supabase (PostgreSQL & pgvector)

Digunakan sebagai basis data relasional untuk menyimpan identitas dokumen ilmiah sekaligus memproses pencarian kemiripan vektor (*vector similarity search*) menggunakan ekstensi *pgvector*.

k. Prisma ORM

Digunakan sebagai *Object-Relational Mapping* (ORM) untuk menjembatani dan mengelola skema tabel basis data hibrida Supabase dari kode program Python.

l. Postman

Digunakan sebagai perangkat lunak untuk menguji responsivitas titik akhir (*endpoints*) API, memantau latensi (*latency*), dan memvalidasi respons data.

3.6 Teknik Analisis Data

Teknik analisis data yang digunakan dalam pelaksanaan PKL/Magang ini adalah analisis deskriptif kualitatif untuk merumuskan spesifikasi sistem, serta analisis kuantitatif topologi graf untuk menguji kualitas hubungan (*edges*) antar-informasi. Analisis deskriptif digunakan untuk mengolah data hasil observasi kode program (*reverse engineering*), wawancara bersama pembimbing lapangan, dan penelusuran dokumentasi teknis. Data tersebut dikelompokkan untuk mengidentifikasi kelemahan sistem lama serta merancang solusi arsitektur *backend* yang sesuai.

Data hasil analisis kebutuhan dibagi menjadi dua kategori utama, yaitu kebutuhan fungsional dan non-fungsional. Analisis kebutuhan fungsional menitikberatkan pada kesesuaian alur ekstraksi lima pilar ilmiah dokumen PDF secara asinkron, pemodelan basis data graf, dan mekanisme sinkronisasi data relasional. Sementara itu, analisis kebutuhan non-fungsional memetakan batas toleransi latensi pemrosesan data (*latency*), efisiensi biaya token penggunaan model bahasa besar (*cost*), serta ketepatan sistem dalam menyaring relasi semantik (*precision*).

Untuk menguji kualitas hubungan (*edges*) yang terbentuk pada *knowledge graph*, penulis melakukan analisis topologi jaringan menggunakan perangkat lunak Gephi. Dataset graf diekspor dari Neo4j untuk diukur metrik strukturnya, meliputi derajat rata-rata (*Average Degree*), diameter jaringan (*Network Diameter*), kerapatan graf (*Graph Density*), dan rata-rata panjang jalur (*Average Path Length*). Hasil pengukuran metrik ini dibandingkan lintas versi sistem (SRE-v.0, SRE-v.1, dan SRE-v.2) untuk membuktikan

peningkatan efisiensi dan kepadatan penautan pengetahuan setelah diterapkannya metode *Multi-Stage Quality Gating*.

BAB IV

HASIL PKL/MAGANG DAN PEMBAHASAN

4.1 Deskripsi Kegiatan PKL/Magang

Pelaksanaan PKL/Magang MBKM ini berlangsung selama enam bulan, terhitung sejak Februari 2026 sampai Juli 2026. Kegiatan bertempat di Pusat Riset Sains Data dan Informasi (PR SDI), Organisasi Riset Elektronika dan Informatika (OREI), Badan Riset dan Inovasi Nasional (BRIN) yang berlokasi di Gedung BASIC Lantai 5, Kawasan Sains dan Teknologi (KST) Samaun Samadikun, Bandung. Aktivitas harian dijalankan secara *hybrid* dengan dominansi kehadiran *onsite* di laboratorium riset. Untuk menjaga kedisiplinan dan akuntabilitas kerja, penulis mengisi *logbook* elektronik secara berkala melalui portal ELSA BRIN serta mengikuti rapat koordinasi progres mingguan (*progress meeting*) setiap hari Senin bersama tim peneliti.

Alur pembagian tugas dan linimasa penyelesaian target pengembangan sistem dirinci pada Tabel 4.1 berikut:

Tabel 4.1 Kegiatan PKL/Magang

No	Waktu	Kegiatan	Keterangan
1	Februari 2026	Audit <i>legacy code</i> dan observasi sistem	Mengamati alur kerja pemetaan konsep pada sistem lama serta memetakan masalah ketidakstabilan penyimpanan data.
2	Maret 2026	Riset literatur dan spesifikasi teknologi	Mengevaluasi model bahasa besar (LLM), membandingkan model <i>embedding</i> , dan merancang skema integrasi basis data hibrida.
3	April 2026	Konstruksi purwarupa basis data graf	Merancang pemodelan graf di Neo4j serta merumuskan gerbang validasi kompatibilitas domain dan koherensi semantik.
4	Mei 2026	Pemugaran arsitektur asinkron <i>backend</i>	Membangun layanan asinkron berbasis FastAPI serta memprogram kalkulasi filter ambang batas dinamis <i>Three Sigma</i> .

5	Juni 2026	Integrasi sistem dan perbaikan program	Menguji kecocokan fungsional antara modul <i>backend</i> dan <i>frontend</i> , mengoptimalkan transmisi data, serta menyelesaikan kendala program.
6	Juli 2026	Penulisan laporan dan serah terima	Menyusun laporan magang, menulis draf artikel ilmiah format IEEE, serta mensosialisasikan serah terima sistem kepada tim riset.

Sumber: Data diolah oleh penulis, 2026

Selain penugasan teknis pengembangan sistem MySRE, penulis menyelesaikan kewajiban administratif dan pelatihan penguatan kompetensi riset. Kompetensi dasar peneliti diperkuat melalui penyelesaian modul pelatihan pada portal BRILian, sementara pemahaman basis data graf disertifikasi secara resmi oleh Neo4j Academy. Penulis juga berpartisipasi aktif dalam forum ilmiah mingguan PR SDI BRIN setiap hari Rabu untuk mendiskusikan tren sains data. Pada akhir masa magang, sistem MySRE didemonstrasikan di hadapan pembimbing lapangan, Rio Nurtantyana, S.Pd., M.Pd., M.Sc., Ph.D., serta divalidasi kinerjanya melalui sesi pemaparan bersama pakar internasional asal Taiwan.

4.2 Permasalahan yang Ditemukan

Permasalahan pertama yang diidentifikasi dari hasil observasi penggunaan sistem adalah terjadinya hambatan antarmuka (*blocking*) ketika pengguna memproses dokumen. Pada sistem sebelumnya yang bersifat sinkronus, pengguna diwajibkan menunggu seluruh tahapan pemrosesan dokumen selesai secara mutlak di layar sebelum dapat melakukan aktivitas lainnya. Ketiadaan sistem pemrosesan latar belakang (*background job processing*) ini menurunkan efisiensi kerja pengguna, terutama saat memproses dokumen berukuran besar.

Permasalahan kedua terletak pada ketidakkonsistenan hasil ekstraksi data mentah dari dokumen PDF ilmiah. Dokumen PDF jurnal umumnya memiliki tata letak dua kolom, tabel, serta catatan kaki yang kompleks. Ekstraksi teks secara linier tanpa pemotongan berbasis posisi menghasilkan

kalimat yang terputus dan paragraf yang bercampur, sehingga informasi yang dikirimkan ke model bahasa besar kehilangan konteks retorika aslinya.

Permasalahan ketiga adalah terbentuknya hubungan graf (*edges*) yang tidak logis atau absurd pada basis data. Ketiadaan gerbang penyangkutan kompatibilitas domain menyebabkan sistem secara acak menghubungkan konsep dari disiplin ilmu yang berbeda. Selain itu, tidak adanya validasi retorika membuat pernyataan latar belakang sering kali ditautkan secara salah sebagai solusi metodologi, yang mengotori kualitas pangkalan data graf.

Permasalahan terakhir adalah tingginya biaya penggunaan token (*cost*) dalam pemanggilan API model bahasa besar. Sistem lama langsung mengirimkan keseluruhan teks artikel ilmiah ke dalam satu *prompt* tunggal. Hal ini tidak hanya memboroskan konsumsi token masukan (*input tokens*) secara tidak efisien, tetapi juga meningkatkan biaya operasional bulanan tanpa memberikan peningkatan akurasi pada hasil ekstraksi.

4.3 Metode Penyelesaian Masalah

Berdasarkan permasalahan yang ditemukan selama pelaksanaan PKL/Magang, dilakukan penanganan terstruktur melalui perubahan arsitektur pengolahan literatur ilmiah. Penulis bertugas sebagai *Backend & Knowledge Data Engineer* untuk mengganti model pemetaan konsep lama (*Legacy Concept Map*) yang bersifat non-persisten menjadi arsitektur *Agentic Knowledge Graph* dengan skema hibrida Neo4j dan Supabase. Rekayasa ini membangun *SAAI Engine* sebagai penggerak utama aplikasi Brain untuk mengekstraksi dan menautkan entitas pengetahuan dari artikel ilmiah secara sistematis.

Langkah penyelesaian diawali dengan melakukan audit dan dekonstruksi kode warisan (*legacy code*) untuk memetakan kelemahan alur kerja terdahulu. Temuan mengenai risiko halusinasi pada metode injeksi data langsung ke LLM mendorong penulis melakukan riset teoretis mendalam tentang *Knowledge Graph* dan *Agentic RAG*. Siklus uji coba (*trial and error*) pada fase *prototyping* di Neo4j dilakukan untuk mengkalibrasi logika ekstraksi

semantik sebelum mengonsolidasikan arsitektur ke dalam kerangka kerja FastAPI.

Selanjutnya, dilakukan pembangunan infrastruktur server yang responsif melalui *Asynchronous Ingestion Pipeline* berbasis FastAPI. Arsitektur asinkronus ini memisahkan proses komputasi berat ke dalam *background tasks* guna menjaga stabilitas layanan dari kendala pemblokiran antarmuka (*blocking*). Transparansi pemrosesan dokumen bagi pengguna disediakan melalui integrasi *Server-Sent Events* (SSE) untuk pelaporan progres secara *real-time*, lengkap dengan isolasi *multi-tenant* dan operasi penghapusan atomik (*Double Kill deletion*) untuk sinkronisasi data.

Tahap berikutnya adalah pemodelan pengetahuan melalui skema 5 Pilar Ilmiah (Latar Belakang, Metodologi, Tujuan, *Gap*, dan Arahana Masa Depan) pada Neo4j. Integritas relasi antar-literatur dijaga secara ketat melalui mekanisme *Multi-Stage Semantic Gating* yang terdiri atas empat lapisan validasi: *Retrieval Gate* untuk validasi ambang batas KNN, *Coherence Gate* untuk konsistensi hubungan lintas pilar, *Dynamic Domain Filtering Gate* untuk relevansi bidang akademik via LLM, serta *Edge Reasoning Gate* untuk kualitas narasi relasi.

Setelah seluruh modul selesai dikembangkan, dilakukan optimasi pemrosesan konteks menggunakan teknik *Multi-Level Parent-Child Chunking* untuk memperkaya akurasi data serta perhitungan ambang batas KNN yang terkalibrasi melalui prinsip statistik *Three Sigma Rule of Thumb*. Pada tahap akhir, sistem diuji secara menyeluruh melalui pengujian fungsional API, pengukuran latensi (*latency*), evaluasi biaya token (*cost*), serta analisis metrik topologi graf menggunakan perangkat lunak Gephi untuk memvalidasi kepadatan dan keandalan hubungan (*edges*) yang terbentuk.

4.4 Hasil Pelaksanaan PKL/Magang

4.4.1 Analisis Kebutuhan Sistem Aplikasi Brain

Analisis kebutuhan sistem ini bertujuan memetakan batasan operasional dan kapabilitas teknis dari backend engine yang menggerakkan aplikasi Brain. Perumusan kebutuhan ini didasarkan pada hasil audit terhadap kode warisan (*legacy code*), yang secara sistematis menutup celah arsitektur lama dengan pendekatan baru berbasis *Knowledge Graph* dan *Agentic AI*.

Kategorisasi kebutuhan sistem dibagi menjadi dua domain. Kebutuhan Fungsional difokuskan pada siklus hidup data, mencakup alur dari ekstraksi dokumen mentah (PDF) hingga distribusi *Knowledge Graph* terstruktur. Sementara itu, kebutuhan non-fungsional difokuskan pada pemenuhan standar industri untuk sistem *Agentic AI*, yang mencakup keamanan data melalui mekanisme isolasi *multi-tenant*, integritas data melalui jaminan *idempotensi*, serta transparansi proses eksekusi dengan memanfaatkan mekanisme asinkron dan *Server-Sent Events* (SSE). Rincian spesifikasi kebutuhan sistem yang menjadi landasan rekayasa arsitektur diuraikan pada Tabel 4.2 dan Tabel 4.3 berikut :

Tabel 4.2 Kebutuhan Fungsional Sistem Backend

No.	Fitur / Modul	Deskripsi Kebutuhan
F-01	Ekstraksi Teks (Parsing)	Sistem harus menerima dokumen PDF (digital maupun pindaian) dan mengonversinya ke format teks murni (Markdown), dengan kemampuan menangani tata letak kolom tunggal maupun multi-kolom.
F-02	Sterilisasi Data	Sistem harus mengidentifikasi dan membuang teks yang tidak relevan dengan substansi penelitian (seperti daftar pustaka, ucapan terima kasih, dan lampiran) sebelum proses vektorisasi.

F-03	Retensi Semantik Struktural	Sistem harus mengikat struktur judul atau sub-bab asli dari jurnal (metadata_header) pada setiap potongan teks agar hierarki informasi tidak hilang saat pemecahan.
F-04	Pemecahan Hierarkis (Multi Level Chunking)	Sistem harus memecah teks menjadi dua lapisan: Child Chunks berukuran kecil untuk efisiensi komputasi pencarian kemiripan, dan Parent Chunks berukuran besar untuk mempertahankan konteks utuh bagi pemrosesan LLM.
F-05	Ekstraksi Pilar Pengetahuan	Sistem harus mengekstraksi teks terkurasi ke dalam 5 format JSON baku yang mewakili Latar Belakang, Metodologi, Tujuan, Gap, dan Arahana Masa Depan.
F-06	Penyimpanan Aset & Vektor	Sistem harus menyimpan dokumen mentah ke dalam <i>cloud storage</i> dan mengamankan potongan teks beserta nilai embedding vektornya ke dalam basis data relasional (Supabase).
F-07	Konstruksi Knowledge Graph	Sistem harus mengubah 5 pilar pengetahuan menjadi entitas (Node) dan menghitung relasinya terhadap literatur lain (Edge) di dalam basis data graf (Neo4j).
F-08	Penghapusan Atomik (Double-Kill Deletion)	Sistem harus menyediakan fungsi penghapusan menyeluruh, di mana penghapusan satu literatur akan menghapus Node di Neo4j sekaligus melenyapkan data vektor dan berkas fisik di Supabase.
F-09	Distribusi Data Graf	Sistem harus menyediakan layanan (endpoint) untuk mendistribusikan data node dan edge secara terstruktur agar dapat divisualisasikan oleh aplikasi frontend.

Sumber: Data diolah oleh penulis, 2026

Tabel 4.3 Kebutuhan Non-Fungsional Sistem Backend

Id	Atribut Kualitas	Deksripsi Kebutuhan
NF-01	Performa (Eksekusi Asinkron)	Sistem tidak boleh memblokir antarmuka pengguna (UI blocking) saat memproses komputasi I/O berat (parsing, ekstraksi LLM, embedding). Pemrosesan wajib

		berjalan di latar belakang (background tasks).
NF-02	Transparansi (Pelaporan Real-Time)	Sistem wajib menyediakan komunikasi <i>Server-Sent Events</i> (SSE) untuk melaporkan status ekstraksi secara instan kepada frontend.
NF-03	Keamanan (Isolasi Multi-Tenant)	Sistem harus menjamin graf pengetahuan satu proyek tidak terekspos ke proyek lain melalui penerapan identitas komposit (composite ID) berbasis struktur hash.
NF-04	Integritas Data (Idempotensi & Anti-Duplikasi)	Sistem harus kebal terhadap insiden penggandaan data saat terjadi unggahan berulang, memanfaatkan klausa spesifik basis data graf untuk memperbarui entitas yang ada tanpa menciptakan duplikat.
NF-05	Keandalan (Integritas Relasi Graf)	Sistem dilarang membentuk relasi antar-literatur secara acak. Pembentukan relasi wajib melewati gerbang validasi bertingkat (<i>Multi-Stage Semantic Gating</i>).
NF-06	Ketahanan (Pemulihan Mandiri)	Sistem harus menangani kegagalan API eksternal (contoh: limit koneksi LLM) dengan mengimplementasikan mekanisme percobaan ulang yang terkalibrasi (<i>exponential backoff retries</i>).

Sumber: Data diolah oleh penulis, 2026

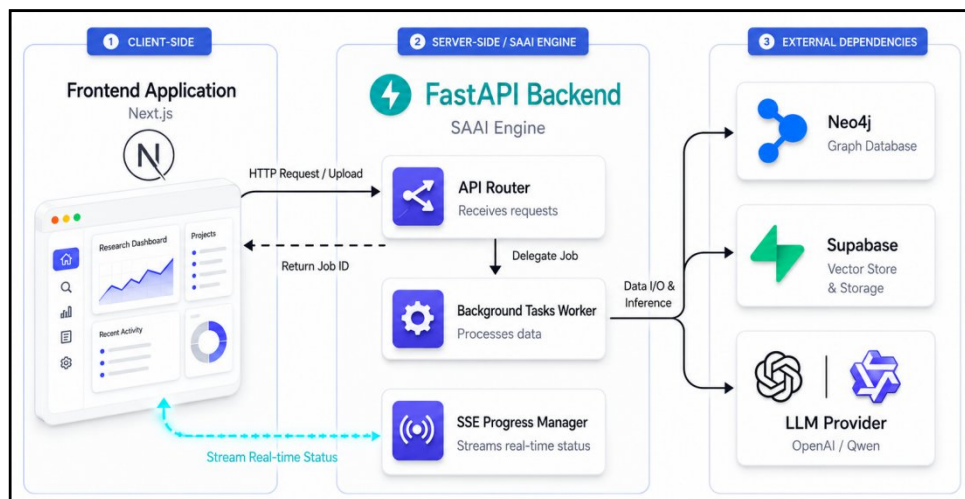
4.4.2 Perancangan Arsitektur dan Fondasi Teknologi Backend

Arsitektur MySRE beroperasi dengan memisahkan ranah klien dari server melalui penerapan arsitektur decoupled. Pemisahan ini merupakan kebutuhan teknis untuk mengisolasi antarmuka pengguna (Next.js) dari beban komputasi masif pada backend. Pemrosesan Natural Language Processing (NLP) dan eksekusi instruksi kecerdasan buatan menyita sumber daya I/O secara intensif. Arsitektur terdistribusi memastikan interaksi pengguna tidak tertunda oleh kalkulasi data di belakang layar.

Alur kontrol sistem berpusat pada pendelegasian tugas. Lapis antarmuka program aplikasi (API Router) menerima permintaan dari klien. Router ini menolak menahan siklus permintaan, melainkan segera

melempar komputasi mentah ke pekerja latar belakang (Background Tasks). Untuk memfasilitasi visibilitas proses, backend membuka saluran *Server-Sent Events* (SSE). Saluran searah ini memancarkan status progres secara instan, menyinkronkan lapisan analitik dengan antarmuka klien.

Interaksi tingkat tinggi antar-komponen ini direpresentasikan pada Gambar 4.1 berikut.



Sumber: Dokumentasi Penulis, 2026

Tabel 4.4 Spesifikasi Teknologi dan Library

Kategori	Library	Justifikasi Teknis
Framework	FastAPI & Uvicorn	Menyediakan arsitektur asynchronous berkinerja tinggi untuk menangani beban I/O berat tanpa memblokir antrian permintaan masuk.
Komunikasi & Keamanan	SSE-Starlette & SlowAPI	sse-starlette memfasilitasi pelaporan progres real-time via Server-Sent Events. slowapi membatasi laju permintaan (<i>rate limiting</i>) untuk melindungi stabilitas server.
Ekstraksi Dokumen & OCR	PyMuPDF & RapidOCR	pymupdf4llm digunakan sebagai parser utama PDF. <i>rapidocr-onnxruntime</i> disematkan sebagai mesin <i>Optical Character Recognition</i> untuk mengonversi gambar/tabel pindaian menjadi teks.

Pemrosesan Semantik Teks	Langchain Text Splitters	Mengatur hierarki pemecahan teks (<i>Multi-Level Chunking</i>) yang mempertahankan metadata judul (header) agar konteks spasial informasi tidak hilang.
Komputasi Embedding	Sentence-Transformers	Memfasilitasi kalkulasi vector embedding (seperti Multilingual-e5-small) secara lokal yang dikhususkan untuk menunjang kebutuhan pencarian semantik internal pada proses Multi-Level Chunking.
Orkestrasi AI & LLM	OpenAI SDK & Google GenAI	Modul interkoneksi resmi untuk mengakses layanan model bahasa besar (LLM) dan model embedding eksternal. Digunakan sebagai penghasil matriks vektor utama penyusun graf, serta menangani kueri inferensi kompleks pada tahap Reasoning Gate dan klasifikasi semantik.
Ketahanan & Profiling AI	Tenacity & Tiktoken	tenacity memastikan pemulihan mandiri dari kegagalan API (<i>exponential backoff</i>). tiktoken melakukan kalkulasi token luring guna mencegah over-context error.
Basis Data Relasional & Vektor	Supabase & Psycopg	supabase digunakan untuk sinkronisasi aset penyimpanan. Psycopg [binary,pool] merupakan adaptor PostgreSQL tingkat rendah yang mengoptimalkan manajemen koneksi (<i>connection pooling</i>).
Basis Data Graf	Neo4j Driver	Modul penghubung resmi ke graph database untuk eksekusi kueri Cypher tingkat lanjut, pemetaan relasi antar-literatur, dan perhitungan kalkulus graf.
Komputasi Numerik	NumPy	Diandalkan untuk komputasi matriks matematika berkinerja tinggi, terutama dalam kalkulasi Cosine Similarity lokal dan perhitungan statistik Z-Score.

Sumber: Data diolah oleh penulis, 2026

4.4.3 Spesifikasi Kontrak Komunikasi dan Antarmuka API

Perancangan antarmuka pemrograman aplikasi (API) pada SAAI Engine mengadopsi prinsip pemisahan tanggung jawab (separation of concerns). Arsitektur komunikasi memisahkan beban pengelolaan infrastruktur data dari beban komputasi kecerdasan buatan. Spesifikasi ini diklasifikasikan ke dalam dua ranah operasional utama, merepresentasikan pembagian peran teknis dalam pengembangan sistem.

Tabel 4.5 merinci skema API infrastruktur yang berfokus pada manajemen lalu lintas data, orkestrasi tugas latar belakang, dan operasi lintas basis data.

Tabel 4.5 Skema API Infrastruktur

Method	Endpoint	Deskripsi
POST	/mcp/upload/submit	Menerima unggahan dokumen mentah dan mendelegasikan beban komputasi berat ke background task. Merespons secara instan dengan Job ID.
GET	/mcp/upload/stream	Membuka jalur komunikasi Server-Sent Events (SSE) untuk menyiarkan progres pemrosesan secara real-time.
DELETE	/api/graph/lit erature/{node_id}	Mengeksekusi penghapusan atomik (Double Kill); menghapus Node dan Edge di Neo4j sekaligus aset fisik di Supabase.
GET	/health	Memeriksa stabilitas operasional layanan dan konektivitas menuju basis data graf.

Sumber: Data diolah oleh penulis, 2026

Tabel 4.6 menjabarkan skema API inteligensi graf yang berfokus pada distribusi struktur pengetahuan dan penyediaan antarmuka bagi interaksi Agentic AI.

Tabel 4.6 Skema API Graph dan Agentic AI Services

Method	Endpoint	Deskripsi
GET	/api/graph/context	Mendistribusikan data Knowledge Graph terisolasi (Node dan Edge) beserta penalaran LLM (reasoning) untuk visualisasi antarmuka.

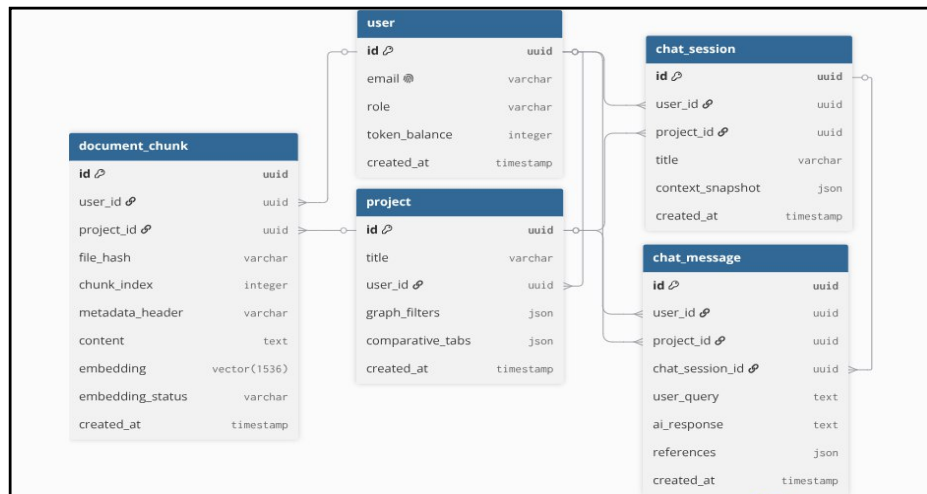
POST	/api/chat/stream	Menangani kueri interaktif pengguna berbasis konteks graf (RAG), merespons luaran LLM secara streaming.
POST	/api/comparative/deep	Memfasilitasi eksekusi LLM untuk melakukan analisis komparatif mendalam antar-literatur yang dipilih pengguna.

Sumber: Data diolah oleh penulis, 2026

4.4.4 Perancangan Skema Database

Penyimpanan data pada arsitektur MySRE mengadopsi strategi Hybrid Data Store untuk menyeimbangkan efisiensi komputasi dan skalabilitas penyimpanan. Sistem memisahkan beban kerja ke dalam dua platform basis data yang berbeda. Supabase yang berbasis PostgreSQL digunakan untuk mengelola data relasional serta penyimpanan massa data potongan teks (*raw chunks*). Basis data ini dioptimasi untuk menangani volume data yang besar dan operasi CRUD standar. Di sisi lain, Neo4j digunakan sebagai mesin pengelola pengetahuan yang memetakan relasi semantik kompleks antar-literatur. Penggunaan basis data graf memungkinkan sistem melakukan pencarian kedekatan vektor secara native dan penelusuran hubungan literatur yang sulit dicapai oleh basis data relasional konvensional.

Pondasi infrastruktur sistem dan manajemen akses pengguna dikelola melalui skema relasional pada Supabase. Integrasi antar-entitas data dalam platform ini digambarkan melalui diagram relasi antar-tabel (Entity Relationship Diagram) pada Gambar 4.2. Skema ini memastikan bahwa setiap aset data, mulai dari identitas pengguna hingga potongan teks terkecil, memiliki jalur rujukan yang jelas.



Gambar 4.2 : ERD PostgreSQL (Supabase)

Sumber: Dokumentasi Penulis, 2026

Tabel 4.7 menjabarkan skema model User yang berfungsi sebagai pengelola identitas dan otoritas akses sistem. Model ini mencatat profil dasar serta saldo penggunaan unit komputasi AI. Selanjutnya, Tabel 4.8 merinci model Project yang bertindak sebagai kontainer isolasi data. Model ini menyimpan preferensi visualisasi dan status fitur analitik yang disesuaikan untuk setiap ruang kerja penelitian.

Tabel 4.7 Skema Model User (Manajemen Identitas)

Kolom	Tipe Data	Deskripsi
id	UUID	Pengenal unik pengguna (Primary Key).
email	String	Alamat surel unik untuk autentikasi.
role	Enum	Peran pengguna (USER atau ADMIN).
token_balance	Int	Saldo kuota penggunaan API LLM/Embedding.

Sumber: Data diolah oleh penulis, 2026

Tabel 4.8 Skema Model Project (Kontainer Kerja)

Kolom	Tipe Data	Deskripsi
id	UUID	Pengenal unik proyek (Primary Key).
title	String	Nama proyek penelitian.
user_id	UUID	Relasi kepemilikan proyek ke tabel

		User.
graph_filters	Json	Menyimpan preferensi filter visualisasi graf pengguna.
comparative_tabs	Json	Menyimpan status tab pada fitur analisis komparatif.

Sumber: Data diolah oleh penulis, 2026

Penyimpanan potongan teks ilmiah yang telah melalui proses *Multi-Level Chunking* dikelola oleh model DocumentChunk sebagaimana dirinci pada Tabel 4.9. Model ini dilengkapi dengan kolom `chunk_index` untuk menjamin urutan rekonstruksi narasi yang sistematis. Selain itu, kolom `embedding_status` digunakan untuk menjaga ketahanan sistem dalam melacak keberhasilan komputasi vektor.

Tabel 4.9 Skema Model DocumentChunk (Gudang Vektor)

Kolom	Tipe Data	Deskripsi Teknis
id	UUID	Pengenal unik potongan teks (Primary Key).
project_id	UUID	Relasi ke tabel Project untuk isolasi data.
user_id	UUID	Relasi ke tabel User untuk otorisasi pemilik data.
file_hash	String	Hash MD5 dokumen sumber sebagai identifikasi integritas berkas.
chunk_index	Integer	Nomor urut potongan teks untuk menjamin rekonstruksi narasi yang sistematis.
metadata_header	String	Judul bab atau sub-bab guna mempertahankan konteks semantik teks.
content	Text	Isi potongan teks hasil proses Chunking.
embedding	Vector(1536)	Representasi numerik teks untuk kebutuhan pencarian semantik (RAG).
embedding_status	String	Melacak status komputasi vektor (PENDING, SUCCESS, atau ERROR).
created_at	Timestamp	Catatan waktu pembuatan entitas data.

Sumber: Data diolah oleh penulis, 2026

Sebagai penunjang interaksi asisten riset, sistem juga menyediakan skema untuk menyimpan riwayat percakapan pengguna yang dirangkum pada Tabel 4.10. Model ini mencatat rujukan literatur yang digunakan oleh kecerdasan buatan dalam memberikan jawaban.

Tabel 4.10 Skema Model ChatSession dan ChatMessage

Model	Kolom Utama	Fungsi
ChatSession	title, context_snapshot	Mengelola sesi percakapan dan ringkasan graf aktif.
ChatMessage	user_query, ai_response, references	Menyimpan log tanya-jawab dan rujukan ragi (RAG).

Sumber: Data diolah oleh penulis, 2026

Perancangan skema pada Neo4j berfokus pada efisiensi pemetaan pengetahuan saintifik. Setiap dokumen diwakili oleh Node :Literature yang memiliki identitas unik berdasarkan composite_id. Pengenal ini dihasilkan melalui perhitungan MD5 di sisi backend untuk mengunci keamanan data tingkat tinggi. Properti metadata dan penyimpanan 5 Pilar Ilmiah pada setiap node diuraikan pada Tabel 4.11 dan Tabel 4.12. Penulis menerapkan format penyimpanan berpasangan antara teks naratif dan vektor 1536-dimensi. Pendekatan ini memungkinkan sistem melakukan pencarian kandidat relasi yang spesifik pada setiap pilar riset.

**Tabel 4.11 Skema Properti Identitas dan Metadata
(Node :Literature)**

Properti	Tipe Data	Deskripsi Teknis
composite_id	String	Identifier Utama: hasil MD5 dari penggabungan identitas pengguna, proyek, dan hash berkas.
hash	String	MD5 dari berkas fisik PDF untuk verifikasi integritas data.
title	String	Judul artikel ilmiah hasil ekstraksi atau identitas berkas cadangan.
keywords	List<String>	Kumpulan kata kunci yang menjadi basis validasi pada Domain Compatibility Gate.

project_id	String	Pengenal unik proyek sebagai filter isolasi data.
user_id	String	Pengenal unik pemilik data untuk otoritas akses.

Sumber: Data diolah oleh penulis, 2026

Tabel 4.12 Skema Properti 5 Pilar Ilmiah (Node :Literature)

Properti	Tipe Data	Peruntukan Teknis
text_[pilar]	String	Menyimpan narasi pilar: background, methodology, objective, gap, dan futurework.
embedding_[pilar]	Vector(1536)	Representasi numerik pilar terkait untuk pencarian pada Native Vector Index.
edges_generated	Boolean	Penanda status operasional untuk mengoptimalkan siklus kerja background worker.
status	String	Melacak posisi dokumen dalam alur kerja sistem.

Sumber: Data diolah oleh penulis, 2026

Relasi antar-literatur (Edge) pada sistem ini membawa atribut validasi yang membuktikan integritas hubungan tersebut. Tabel 4.13 merinci properti pada relasi semantik yang dihasilkan oleh alur kerja Agentic AI. Properti gate dan matched_pillars berfungsi sebagai rekam jejak teknis yang menunjukkan bahwa relasi tersebut telah lolos tahap uji validasi yang terukur.

Tabel 4.13 Skema Properti Relasi Semantik (Edge)

Properti	Tipe Data	Deskripsi dan Justifikasi
score	Float	Nilai Cosine Similarity yang memicu terbentuknya relasi (rentang 0.0 sampai 1.0).
reasoning	String	Narasi penjelasan hasil nalar LLM yang menjustifikasi keterhubungan antar-literatur.
matched_pillars	List<String>	Daftar pilar yang terdeteksi selaras (bukti fisik dari Coherence Gate).

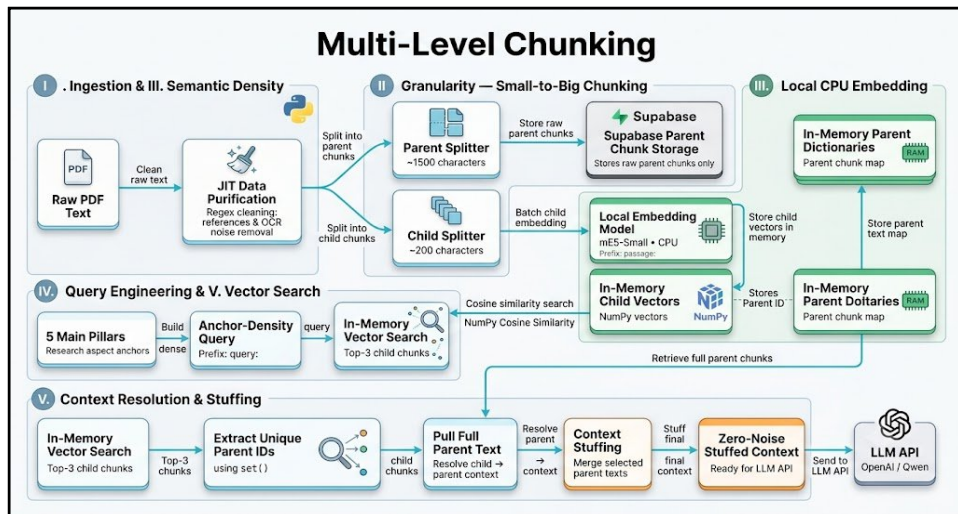
gate	String	Label pelacak logika validasi yang berhasil dilewati.
-------------	--------	---

Sumber: Data diolah oleh penulis, 2026

4.4.5 Implementasi Multi Level Chunking & Stuffed Context

Sistem ekstraksi pilar Literature pada modul “*Brain*” MySRE memerlukan pendekatan yang berbeda dari arsitektur *Retrieval-Augmented Generation* (RAG) konvensional. Pada metode Classic RAG, dokumen seringkali dikirimkan secara utuh atau dipotong dalam ukuran statis yang besar ke dalam model bahasa, yang berisiko memicu distorsi informasi. Arsitektur Backend mengadopsi paradigma *Small-to-Big Retrieval*, sebuah arsitektur yang memisahkan unit teks untuk kebutuhan pencarian dan kebutuhan penalaran. Pemisahan ini didasarkan pada temuan Eibich et al. (2024) melalui kerangka kerja ARAGOG, yang menunjukkan bahwa membedakan granularitas teks antara fase pencarian dan fase sintesis terbukti secara statistik memberikan hasil yang lebih unggul dibandingkan metode pencocokan vektor tunggal. Validasi ini diperkuat oleh riset Lumer et al. (2025) di *PricewaterhouseCoopers* (PwC) yang menyatakan bahwa mekanisme perluasan konteks ini mutlak diperlukan untuk menyelesaikan kueri penalaran kompleks yang melintasi batas-batas paragraf (*Multi-hop Reasoning*) tanpa menimbulkan latensi yang signifikan.

Alur pemecahan teks dan orkestrasi potongan ini digambarkan pada Gambar 4.3 berikut.



Gambar 4.3 : Pipeline Multi-Level Chunking

Sumber: Dokumentasi Penulis, 2026

1. Granularitas Vektor dan Pencegahan Information Loss

Dalam mengimplementasikan strategi Small-to-Big, teks akademik dipecah secara hierarkis. Sistem menetapkan dua ukuran berbeda: Parent Chunk dengan batas ± 1500 karakter untuk menjaga keutuhan narasi, dan Child Chunk berukuran ± 200 karakter yang dioptimalkan untuk pencarian vektor. Hanya teks pada Level Child Chunk yang diubah menjadi representasi numerik. Desain ini merujuk pada prinsip Tan et al. (2025) bahwa memaksakan teks panjang ke dalam vektor berdimensi tetap akan menyebabkan hilangnya detail spesifik (Information Loss). Penggunaan unit kecil setingkat proposisi menjamin sinyal semantik tetap padat. Pendekatan ini selaras dengan bukti empiris dari Chen et al. (2024) yang mendemonstrasikan bahwa pencarian pada tingkat proposisi mampu meningkatkan akurasi Recall hingga +10 poin, khususnya saat sistem menargetkan informasi yang bersifat anomali atau langka (long-tailed information) seperti pilar “Gap Penelitian”.

2. Purifikasi Data dan Vektorisasi Asinkronus

Akurasi pencarian vektor bergantung pada kemurnian teks. Sebelum tahap pemodelan vektor, sistem melakukan pembersihan teks secara Just-In-Time (JIT) (Data Purification V2) menggunakan aturan

Regular Expression untuk membuang tabel hasil OCR yang rusak dan daftar pustaka. Metode ini merujuk pada konsep Balanced Entropy Engineering oleh Wang et al. (2025), yang memperingatkan bahwa teks asing dalam ruang vektor akan menciptakan ekspansi entropi buatan yang melarutkan atensi model terhadap konten esensial “Attention Dilution”.

Vektorisasi dilaksanakan di memori lokal menggunakan model `intfloat/multilingual-e5-small`. Karena proses ini melibatkan ratusan `child chunks` per dokumen, sistem menggunakan pemrosesan asinkronus (`ThreadPool/Asyncio`) guna mencegah hambatan pada alur utama. Penggunaan model `mE5-Small` didasarkan pada kapabilitasnya dalam menangani jurnal ilmiah yang mencampur istilah bahasa Inggris dan Indonesia. Sesuai dengan spesifikasi arsitektur `asymmetric contrastive learning` pada `mE5`, instruksi prefix disematkan untuk menjaga integritas matriks. Snippet implementasi pembersihan ditunjukkan pada Gambar 4.4.

```
def _embed_batch(self, texts: List[str], is_query: bool = False) -> List[List[float]]:
    """Embed a list of texts using the local SentenceTransformer model with E5 prefix rules."""
    prefix = "query: " if is_query else "passage: "

    # E5 models have a max sequence length of 512 tokens.
    # We truncate by character (approx 1500 chars to be safe) before encoding.
    MAX_CHARS = 1500

    prefixed_texts = []
    for t in texts:
        cleaned = (t or "").strip()
        if len(cleaned) > MAX_CHARS:
            logger.warning(f"Text exceeded {MAX_CHARS} chars, truncating before embedding.")
            cleaned = cleaned[:MAX_CHARS]
        prefixed_texts.append(f"{prefix}{cleaned}")

    # model.encode returns a numpy array or list of numpy arrays, we convert to list of floats
    embeddings = self.model.encode(prefixed_texts, batch_size=32, show_progress_bar=False)
    return [emb.tolist() for emb in embeddings]
```

Gambar 4.4 : Snippet Code Purifikasi Data

Sumber: Dokumentasi Penulis, 2026

3. Rekayasa Query : Anchor Density Strategy

Untuk meningkatkan ketajaman titik temu, sistem meninggalkan format kueri tanya-jawab naratif (Q&A) dan beralih ke strategi Anchor-

Density. Strategi ini menggabungkan proposisi dari Wang et al. (2025) dan Tan et al. (2024) bahwa kueri yang padat informasi (information-dense formats) lebih efektif mengunci fakta esensial. Dengan menggunakan deretan kata kunci teknis sebagai jangkar, sistem memusatkan entropi murni pada target pencarian, mencegah retriever meleset di ruang multidimensi. Snippet kode konstanta kueri jangkar ditunjukkan pada Gambar 4.5.

```
query_texts = [  
    "latar belakang masalah, urgensi penelitian, motivasi mendasar, fenomena krusial"  
    "konteks situasi saat ini, dan alasan utama studi ini dilakukan.",  
    "rincian teknis metodologi, arsitektur sistem, desain eksperimen, teknik pengumpulan dataset"  
    "algoritma yang diusulkan, sampel, dan prosedur pengujian.",  
    "keterbatasan penelitian terdahulu, kelemahan metode sebelumnya, kekosongan literatur"  
    "masalah yang belum terselesaikan, namun demikian, akan tetapi, dan hasil yang belum jelas.",  
    "tujuan utama penelitian, sasaran spesifik, kontribusi inti makalah, hasil yang ingin dicapai,"  
    "mengeksplorasi, dan solusi yang diusulkan untuk menyelesaikan masalah.",  
    "saran untuk penelitian selanjutnya, rekomendasi pekerjaan di masa depan, arah pengembangan"  
    "masa depan, dan peluang studi berikutnya untuk mengatasi batasan saat ini."  
]
```

Gambar 4.5 : Anchor Density Query

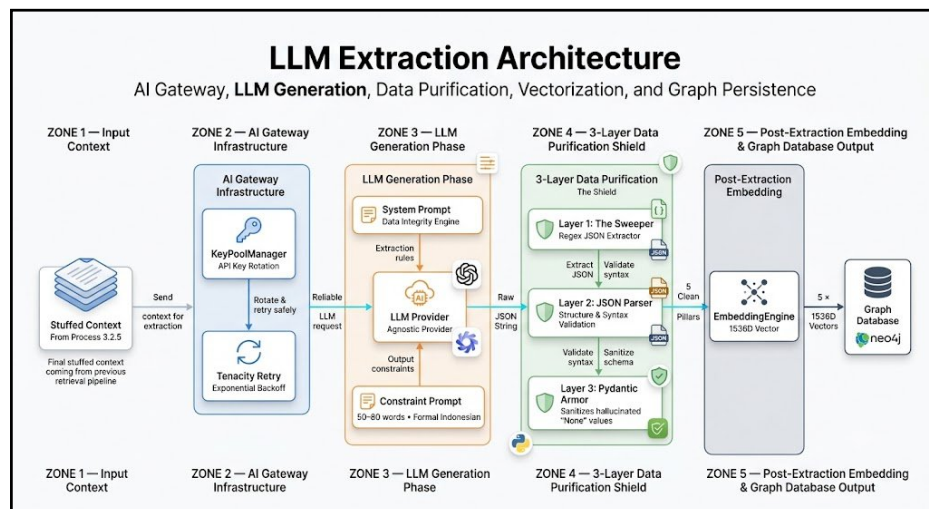
Sumber: Dokumentasi Penulis, 2026

4. Deduplikasi & Stuffing Context

Tahap akhir dari alur in-memory ini adalah resolusi konteks. Setelah pencarian vektor menemukan Child Chunks yang relevan, sistem menarik Ids Parent terkait untuk mengumpulkan teks utuh. Mengingat satu paragraf Parent dapat menaungi beberapa Child yang relevan, sistem menggunakan himpunan set() pada Id Parent untuk mengeliminasi duplikasi atau residu yang bertumpang tindih (overlap). Teks-teks dari ID unik tersebut digabungkan menjadi satu blok informasi yang disebut Stuffed Context. Penggabungan akhir ini menghasilkan Zero-Noise Context, yang memastikan model LLM dapat menerima landasan fakta yang padat, akurat, dan terarah untuk penalaran tahap akhir tanpa risiko halusinasi.

4.4.6 Strategi Prompt Engineering dan Ekstraksi Berbasis Skema

Sebelum mengeksekusi instruksi pemrosesan bahasa, sistem memerlukan lapisan infrastruktur yang menjamin stabilitas dan agnostisisme terhadap penyedia layanan. Backend MySRE mengimplementasikan AI Gateway yang tidak bergantung pada satu penyedia (seperti OpenAI atau DashScope) secara statis. Lapisan ini dilengkapi dengan KeyPoolManager untuk rotasi kunci API secara dinamis, mengeliminasi risiko rate-limiting. Selain itu, untuk mengamankan eksekusi ekstraksi dari kegagalan jaringan yang tidak terduga, sistem mengadopsi mekanisme Exponential Backoff Retry menggunakan pustaka Tenacity. Pendekatan ini memastikan bahwa pipeline ekstraksi tetap beroperasi secara berkesinambungan dan tidak mengalami kelumpuhan akibat gangguan koneksi sesaat. Arsitektur Ekstraksi 5 pilar dapat dilihat pada Gambar 4.6.



Gambar 4.6 : Arsitektur Ekstraksi LLM

Sumber: Dokumentasi Penulis, 2026

1. Rekayasa Prompt dan Penegakan Gaya Bahasa Akademik

Mengekstraksi pilar ilmiah dari teks yang panjang menuntut batasan yang ketat (Constraint Prompting). Sistem memisahkan peran instruksi dasar melalui System Prompt yang bertindak sebagai “Senior Academic Data Integrity Engine”, dan instruksi spesifik melalui User Prompt. Teks sumber yang telah digabungkan (Stuffed Context) disematmen secara

aman ke dalam instruksi. Selain itu, Large Language Model (LLM) diinstruksikan secara absolut untuk mematuhi rentang panjang tulisan (50-80 kata per pilar) dan diwajibkan menggunakan frasa pembuka akademik bahasa Indonesia yang baku. Tujuannya adalah untuk mensterilkan keluaran dari struktur kalimat generik yang sering dihasilkan oleh mesin. Snippet kode prompt ekstraksi ditunjukkan pada Gambar 4.7.

```
prompt = f
"""You are a Senior Academic Reviewer and Data Extraction Specialist...

CRITICAL RULE: Unlike metadata, you MUST write ALL pillar outputs STRICTLY IN FORMAL INDONESIAN with a sharp, flowing academic tone. Avoid robotic or machine-translated styles.

- background: Berdasarkan teks, ekstrak LATAR BELAKANG PENELITIAN. ... Gunakan frasa pembuka seperti: 'Dilatarbelakangi oleh...', 'Menjadi krusial karena...'
- gap: Berdasarkan teks, ekstrak GAP DAN KETERBATASAN PENELITIAN. ... Gunakan frasa pembuka seperti: 'Meninggalkan celah pada...', 'Belum dieksplorasi...'

## GLOBAL CONSTRAINTS & FORMAT
- Each pillar text MUST strictly be between 50 and 80 words.
- Your response MUST perfectly match the requested JSON schema.

Compiled Text Excerpts:
{text[:25000]}
"""
```

Gambar 4.7 : Snippet Code Prompt Ekstraksi Pilar

Sumber: Dokumentasi Penulis, 2026

2. Kontrak Data Terstruktur dan Tiga Lapis Purifikasi (The Shield)

Ekstraksi yang dilakukan oleh model bahasa bersifat non-deterministik, sehingga rentan menghasilkan data yang tidak terstruktur atau mengalami halusinasi. Untuk mengamankan integritas basis data, sistem memberlakukan kontrak data berformat JSON berdasarkan skema LiteraturePillars. Keluaran LLM kemudian melewati tiga lapis purifikasi bertahap. Lapis pertama adalah modul Sweeper yang bertugas memotong elemen non-esensial dari struktur data. Lapis kedua memvalidasi sintaksis JSON. Lapis ketiga, yang berfungsi sebagai pelindung utama, memanfaatkan fitur BeforeValidator dari pustaka Pydantic untuk menetralkan halusinasi, seperti luaran bertuliskan “None” atau “Informasi

tidak tersedia”, dan menggantinya menjadi string kosong yang aman. Snippet kode pembersihan ditunjukkan pada Gambar 4.8.

```
1  ```python
2  def clean_llm_json_string(raw_text: str) -> str:
3      """LAPIS 1: Membersihkan teks respons LLM dari basa-basi dan markdown fences."""
4      start_idx = raw_text.find('{')
5      end_idx = raw_text.rfind('}')
6      if start_idx != -1 and end_idx != -1 and start_idx < end_idx:
7          return raw_text[start_idx:end_idx + 1]
8      return raw_text
9
10 # LAPIS 3: Armor sanitasi halusinasi (mengubah 'None'/'tidak tersedia' menjadi string kosong)
11 def sanitize_llm_field(value) -> str:
12     # ... (logika pembersihan reject_phrases) ...
13     return value.strip()
14
15 class LiteraturePillars(BaseModel):
16     background: Annotated[str, BeforeValidator(sanitize_llm_field)] = Field(...)
17     gap: Annotated[str, BeforeValidator(sanitize_llm_field)] = Field(...)
18     # ...
19
20 # Eksekusi Parsing (Lapis 2 & 3 berjalan bersamaan)
21 cleaned_json_str = clean_llm_json_string(raw_json)
22 parsed = json.loads(cleaned_json_str) # Lapis 2: JSON Load
23 validated_response = ExtractionResponse.model_validate(parsed) # Lapis 3: Pydantic Validation
24 ```
```

Gambar 4.8 : Snippet Code Purification Output LLM

Sumber: Dokumentasi Penulis, 2026

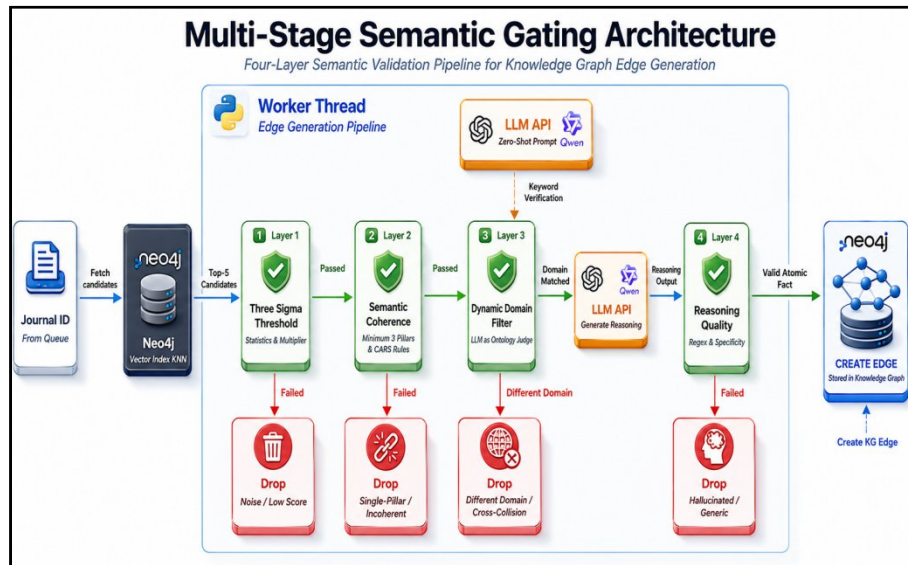
3. Vektorisasi Post-Extraction

Fase terakhir dalam alur ekstraksi ini adalah mentransformasi teks pilar yang telah disanitasi menjadi representasi matematis. Setelah kelima pilar dinyatakan lolos purifikasi Pydantic, masing-masing blok teks dikembalikan ke EmbeddingEngine di dalam infrastruktur AI Gateway. Teks pilar tersebut dikonversi menjadi vektor semantik berdimensi 1536. Vektor inilah yang nantinya akan disimpan di dalam basis data grafik (Neo4j) and berfungsi sebagai amunisi utama untuk proses rekayasa Multi-Stage Semantic Gating pada tahap selanjutnya.

4.4.7 Rekayasa Multi-Stage Semantic Gating

Implementasi Vector Search tradisional yang murni bersandar pada algoritma K-Nearest Neighbor (K-NN) menyimpan kelemahan matematis dalam konteks ekstraksi literatur ilmiah. Algoritma K-NN beroperasi dengan memaksakan pembentukan relasi terhadap sejumlah dokumen terdekat, terlepas dari relevansi aktualnya. Dalam skala yang masif, pemaksaan ini berisiko menciptakan hairball graph, dimana sebuah graf dipenuhi oleh relasi palsu (false positives). Untuk mereduksi risiko

tersebut, arsitektur MySRE menerapkan mekanisme Multi-Stage Semantic Gating, sebuah sistem penyaringan empat lapis yang dirancang untuk memverifikasi setiap relasi secara statistik, struktural, ontologis, dan leksikal sebelum disimpan ke dalam Knowledge Graph. Diagram sistem gating dapat dilihat pada Gambar 4.9.



Gambar 4.9 : Multi Stage Semantic Gating

Sumber: Dokumentasi Penulis, 2026

1. K-NN dan Ambang Batas Deteksi Anomali (Three Sigma Rule)

Lapisan pertama memodifikasi pendekatan K-NN menjadi Graf Lingkungan- ϵ (ϵ -Neighborhood Graph). Sesuai dengan batasan yang diusulkan oleh Luxburg (2007), metode ini menolak pemaksaan koneksi dengan hanya mengizinkan pembentukan relasi jika skor kemiripan melampaui parameter ϵ Threshold (ambang batas). Dalam sistem ini, nilai ϵ tidak ditentukan secara statis. Mengacu pada teknik Statistical Anomaly Detection (Chandola et al., 2009), kemiripan semantik sejati didefinisikan sebagai “anomali” yang berada di luar batas tiga standar deviasi (3σ) dari rata-rata (μ) skor noise vektor.

Lebih lanjut, sistem mengimplementasikan Threshold Multiplier sebagai Hyper Parameter untuk memitigasi kelemahan model dense retriever saat menghadapi entitas yang spesifik dan langka. Pilar dengan kepadatan informasi rendah, seperti “Gap Penelitian”, diberikan penalti

pengali yang lebih tinggi untuk memperketat interval kepercayaan (confidence interval). Berikut adalah rumusan matematis dan nilai kalibrasi yang diterapkan pada sistem:

$$\text{Threshold pilar} = (\mu + 3\sigma) \times \text{Multiplier pilar}$$

Tabel 4.14 Parameter Kalibrasi Threshold Dinamis per Pilar

Pilar Akademik	Multiplier	Nilai Threshold Statis
Latar Belakang (Background)	1.0	0.6807
Metodologi (Methodology)	1.0	0.6914
Tujuan (Objective)	1.05	0.7292
Saran Masa Depan (Future Work)	1.05	0.7962
Gap Penelitian (Gap)	1.15	0.8102

Sumber: Data diolah oleh penulis, 2026

2. Semantic Coherence Gate dan Kepatuhan Tata Bahasa Akademik

Lolos dari ambang batas statistik tidak menjamin relevansi struktural. Lapisan kedua (*Semantic Coherence Gate*) memastikan keselarasan lintas faset. Secara teoretis, sistem mengadopsi filosofi dari model ASPIRE (Mysore et al., 2022) yang membuktikan bahwa dokumen saintifik memiliki argumen yang bersifat multi-faset, sehingga evaluasi kemiripan harus dilakukan secara terperinci per-aspek.

Namun, alih-alih mengimplementasikan algoritma *Optimal Transport* dari ASPIRE yang memiliki kompleksitas komputasi kuadratik tinggi, penelitian ini menerapkan pendekatan rekayasa heuristik. Melalui fungsi `validate_semantic_coherence`, sistem akan mengeliminasi kandidat relasi apabila jumlah pilar yang selaras lebih kecil daripada batas minimum konsensus ($|P_{\text{matched}}| < P_{\text{min}}$). Mekanisme ini memastikan bahwa hubungan semantik yang terbentuk tidak hanya didasarkan pada kemiripan yang terjadi secara kebetulan pada satu pilar, melainkan mencerminkan keselarasan struktural yang konsisten di antara beberapa pilar sebelum suatu relasi (*edge*) dibentuk.

Selain kuantitas pilar, sistem menegakkan dependensi logis yang diturunkan dari Teori CARS (*Creating a Research Space*) oleh John Swales (1990). Dalam tata bahasa saintifik, metodologi (sebagai instrumen) tidak bermakna tanpa adanya niat, yaitu “Tujuan” atau “Gap”. Karenanya, kesamaan metodologi secara terisolasi akan ditolak oleh sistem. Aturan dependensi deterministik ini dikodifikasikan sebagai berikut:

Tabel 4.15 Aturan Dependensi Semantic Coherence Gate

Pilar Utama yang Terdeteksi Mirip	Syarat Ketergantungan Mutlak (Minimal Salah Satu)
Metodologi	Harus disertai kemiripan Tujuan atau Gap
Tujuan	Harus disertai kemiripan Latar Belakang, Metodologi, atau Gap
Gap	Harus disertai kemiripan Tujuan atau Metodologi
Arahan Masa Depan (Future Work)	Harus disertai kemiripan Tujuan, Metodologi, atau Gap

Sumber: Data diolah oleh penulis, 2026

3. Dynamic Domain Filtering

Kelemahan inheren dari komputasi vektor adalah kebutaannya terhadap taksonomi keilmuan, yang dapat memicu Cross-Domain Semantic Collision. misalnya, algoritma Random Forest di domain meteorologi akan dinilai mirip dengan Random Forest di domain kardiologi. Untuk menanganinya, Zhou et al. (2023) merekomendasikan teknik Zero-Shot Prompting untuk mendelegasikan LLM sebagai “Hakim”. Sistem memanggil LLM untuk mengevaluasi kata kunci dari kedua jurnal dan memastikan adanya irisan interdisipliner sebelum relasi diresmikan. Snippet penyaring ditunjukkan pada Gambar 4.10.

```

def evaluate_domain_compatibility_with_llm(client, keyword_source: list, keyword_target:
list) -> dict:

    """Gate 3: Evaluate domain compatibility between two nodes using LLM semantic judgmen
t."""
    user_content = (
        "Compare two sets of research keywords. Determine if they share the same "
        "overarching academic domain or have a strong interdisciplinary intersection.\n\n"
        "Rules:\n"
        "1. Focus on latent semantic meaning, not exact string matches.\n"
        "2. Treat concepts in different languages as identical if their meanings align.\n"
        "3. Resolve abbreviations to their full concepts.\n\n"
        "Return ONLY a valid JSON object with exactly two keys:\n"

        "- reason: Ultra-concise explanation of their relationship or mismatch (under 10 words).
\n"
        "- is_compatible: boolean (true if domains intersect/match, false if unrelated).
\n\n"
        f"Source Keywords: {keyword_source}\n"
        f"Target Keywords: {keyword_target}"
    )

```

Gambar 4.10 : Snippet Code Domain Filtering

Sumber: Dokumentasi Penulis, 2026

4. Validasi Kualitas Edge Reasoning Eksternal

Tahap pamungkas mengharuskan LLM menuliskan alasan (reasoning) keterhubungan antar-dokumen (generation-time correction), sesuai panduan Zhou et al. (2023). Namun, LLM rentan menghasilkan halusinasi atau teks generik. Karena LLM tidak mampu mengoreksi penalarannya sendiri (Intrinsic Self-Correction), sistem menggunakan modul evaluasi eksternal berbasis Regular Expression (Regex) di Python. Modul ini bertindak sebagai Rule-Based Expert System yang menggugurkan relasi jika alasan yang diberikan bersifat basa-basi. Evaluasi ini diperkuat dengan pendekatan fakta atomik (Min et al., 2023), yang mewajibkan LLM untuk menyebutkan kata kunci spesifik yang dapat diverifikasi (seperti “algoritma” atau “dataset”). Snippet ditunjukkan pada Gambar 4.11.

```

REASONING_GENERIC_PATTERNS = [
    r"\bsama[- ]?sama membahas\b",
    r"\bmemiliki kemiripan\b",
    r"\btopik serupa\b",
]

def validate_edge_reasoning_quality(reasoning: str) -> tuple[bool, str]:
    """Runtime quality gate untuk menolak reasoning fallback/generik sebelum
    merge."""
    lowered = (reasoning or "").strip().lower()

    # 1. Pengecekan Pola Basa-basi (Halusinasi LLM)
    generic_hits = any(re.search(pattern, lowered, flags=re.IGNORECASE)
        for pattern in REASONING_GENERIC_PATTERNS)

    # 2. Pemaksaan Fakta Atomik (Alat pendeteksi kebohongan)
    specificity_hits = sum(1 for kw in REASONING_SPECIFICITY_KEYWORDS
        if kw in lowered)

    if generic_hits and specificity_hits < 2:
        return False, "generic-reasoning-detected"

    return True, "quality-ok"

```

Gambar 4.11 : Snippet Code Edge Reasoning Quality Check

Sumber: Dokumentasi Penulis, 2026

4.5 Pembahasan

Evaluasi terhadap kinerja arsitektur backend MySRE berfokus pada penyelesaian kendala teknis sistem warisan melalui pembuktian empiris dan analisis batasan operasional. Pemaparan ini dibagi menjadi dua domain utama: analisis efektivitas infrastruktur backend, pengujian struktur graf dengan metrik topologi Gephi, serta kajian kesenjangan antara teori akademis dengan implementasi kode di lapangan.

1. Analisis Efektivitas Infrastruktur Backend

Masalah hambatan antarmuka pengguna (UI blocking) pada sistem lama diselesaikan melalui perubahan alur kerja sinkronus menjadi asinkronus menggunakan FastAPI background tasks. Router API segera mengembalikan respons instan berupa pengenalan pekerjaan (Job ID) setelah dokumen diterima. Komputasi berat seperti pemindaian dokumen, ekstraksi pilar ilmiah, dan penghitungan vektor dipindahkan ke antrean latar belakang.

Meskipun membebaskan antarmuka dari kelambatan, sistem ini memiliki batasan ketergantungan. Pelaporan status progres yang dikirim secara searah melalui *Server-Sent Events* (SSE) membutuhkan stabilitas koneksi HTTP yang konstan. Kegagalan jaringan di sisi klien akan memutuskan komunikasi visualisasi status progres. Kondisi ini tidak membatalkan proses komputasi di sisi server, namun memicu disinkronisasi informasi di sisi pengguna. Klien harus melakukan penyegaran halaman yang memaksa sistem memanggil ulang status pekerjaan (job status) dari database.

Penerapan *Multi-Level Chunking* dengan metode Small-to-Big Retrieval memisahkan struktur pencarian dari struktur penalaran. Child chunk berukuran ± 200 karakter digunakan untuk pencocokan vektor, sementara Parent chunk berukuran ± 1500 karakter digunakan sebagai konteks inferensi. Pendekatan ini meminimalkan risiko hilangnya detail informasi spesifik. Evaluasi terhadap penyimpanan data menunjukkan konsekuensi logis berupa redundansi data teks pada Supabase. Sistem harus menyimpan potongan teks yang sama dalam dua tingkat kerapatan yang berbeda. Hal ini melipatgandakan kebutuhan memori penyimpanan basis data relasional.

2. Pengujian Struktur Graf dengan Metrik Topologi Gephi

Evaluasi terhadap kualitas relasi semantik yang terbentuk dilakukan dengan membandingkan tiga versi pengembangan sistem. Uji coba menggunakan 15 artikel ilmiah yang sama sebagai objek pengujian. Data komparatif metrik topologi dari ketiga versi tersebut diukur menggunakan Gephi dan disajikan pada Tabel 4.16.

Tabel 4.16 Hasil Analisis Metrik Topologi Graf dengan Gephi

Poin Metrik	SRE-v.0	SRE-v.1	SRE-v.2
Nodes (Jumlah Entitas)	15	15	15
Edges (Jumlah Relasi)	32	105	108
Avg. Degree (Rata-rata Derajat Hubungan)	2,13	7,0	7,2
Network Diameter (Jarak Terjauh Node)	4	1	4

Graph Density (Kerapatan Graf)	0,305	0,5	0,514
Avg. Path Length (Rata-rata Panjang Jalur)	1,933	1,0	1,667

Sumber: Data Primer Diolah (Lampiran Uji Gephi BRIN), 2026

Berdasarkan Tabel 4.16, terlihat adanya perubahan signifikan pada kerapatan hubungan antardokumen. Versi awal (SRE-v.0) hanya menghasilkan 32 hubungan (edges) dengan kerapatan graf (graph density) sebesar 0,305. Sebaliknya, SRE-v.2 menghasilkan 108 hubungan dengan kerapatan graf meningkat menjadi 0,514. Peningkatan ini membuktikan bahwa mekanisme pengindeksan pilar ilmiah mampu mengidentifikasi korelasi antar-penelitian secara lebih komprehensif dibandingkan metode pencocokan vektor tunggal pada sistem lama.

Keandalan struktur graf baru terlihat dari metrik rata-rata panjang jalur (average path length) sebesar 1,667 pada SRE-v.2. Angka ini lebih rendah dibandingkan SRE-v.0 yang mencapai 1,933. Jarak antar-node yang lebih pendek mempercepat proses penelusuran informasi oleh agen kecerdasan buatan (agentic search) saat menjalankan kueri penalaran multi-langkah (multi-hop reasoning).

Meskipun metrik kuantitatif menunjukkan struktur graf yang lebih padat dan efisien, visualisasi graf dengan kerapatan 0,514 berpotensi memicu bias visual (hairball effect) di sisi pengguna. Jika seluruh hubungan dirender secara bersamaan tanpa filter kueri yang spesifik pada antarmuka pengguna, visualisasi tersebut sulit dipahami oleh manusia. Batasan ini menunjukkan bahwa kepadatan graf yang tinggi harus diimbangi dengan filter visualisasi yang dinamis di tingkat aplikasi frontend.

3. **Kajian Kesenjangan Teori dengan Implementasi Kode**

Penyusunan arsitektur MySRE merujuk pada beberapa literatur ilmiah. Analisis objektif terhadap kode menunjukkan adanya kesenjangan antara konsep teoretis dengan batasan praktis perangkat lunak:

1. Teori CARS John Swales vs Logika Heuristik: Teori Creating a Research Space (Swales, 1990) dirancang sebagai kerangka analisis retorika

penulisan manusia. Ketika teori ini dipaksakan menjadi aturan validasi database yang kaku pada fungsi `validate_semantic_coherence`, sistem berisiko menolak hubungan semantik yang valid. Hubungan antardokumen yang sebenarnya penting secara substantif dapat gugur hanya karena penulis dokumen tersebut tidak mengikuti pola transisi penulisan CARS standar.

2. Aturan Tiga Sigma (3σ) vs Skor Kemiripan Riil: Penggunaan aturan 3σ untuk deteksi anomali statistik mengasumsikan skor kemiripan vektor berdistribusi normal (Gaussian). Di lapangan, sebaran skor kemiripan semantik pada korpus dokumen yang homogen cenderung condong (skewed). Batasan ini menyebabkan penentuan batas ambang dinamis (dynamic threshold) terkadang menjadi terlalu ketat atau longgar, sehingga membutuhkan kalibrasi konstanta pengali (multiplier) secara berkala.
3. Penyederhanaan Model ASPIRE: Model ASPIRE (Mysore et al., 2022) menyarankan kalkulasi kecocokan multi-aspek menggunakan algoritma optimal transport yang membutuhkan komputasi kuadratik tinggi. Karena keterbatasan spesifikasi perangkat keras lokal (ThinkPad T480 dengan RAM 20 GB), penelitian ini menerapkan pendekatan yang lebih ringan berupa pencocokan berdasarkan jumlah pilar yang terdeteksi (*COHERENCE_MIN_MATCHED_PILLARS*). Penyederhanaan ini menghasilkan efisiensi komputasi yang lebih tinggi, tetapi sebagai konsekuensinya kemampuan sistem dalam mengenali relasi semantik tingkat lanjut menjadi lebih terbatas

4.6 Pelajaran yang Diperoleh

Pelaksanaan magang di Pusat Riset Sains Data dan Informasi BRIN memberikan pengalaman nyata dalam menghadapi kompleksitas rekayasa data. Penulis memperoleh pengalaman berharga dan pelajaran baru yang dikelompokkan ke dalam aspek teknis, manajerial, dan penentuan spesialisasi karir.

1. Dinamika Kolaborasi dan Riset Profesional

Refleksi berharga dipetik dari dinamika kerja kelompok riset *Human Computer Interaction - Visualization* (HCI-V) di bawah arahan supervisor Rio Nurtantyana, Ph.D. Penulis menyadari bahwa keberhasilan riset teknologi tidak hanya bergantung pada kapasitas pemikiran individual, melainkan ditentukan oleh komunikasi intensif melalui diskusi ilmiah pekanan setiap hari Rabu. Bimbingan teknis yang teratur serta dukungan moral dari mentor profesional membantu penulis mengatasi kebuntuan logika saat merancang modul semantik. Keterlibatan dalam sesi presentasi dan demonstrasi produk di depan pakar internasional (*International Expert*) dari Taiwan melatih ketahanan mental penulis dalam mempertanggungjawabkan desain arsitektur di forum akademik formal.

2. Strategi Rekayasa Data Praktis

Dari sisi kompetensi teknis, penulis memperoleh pelajaran berharga terkait tata kelola siklus pengembangan perangkat lunak secara langsung. Pengalaman dalam membedah kode warisan (*legacy code*), mengeksekusi *reverse engineering*, dan menyelaraskan dua teknologi basis data yang berbeda (Supabase untuk data relasional-vektor serta Neo4j untuk visualisasi graf) memberikan pemahaman mendalam tentang praktik data engineering di lapangan. Penulis juga belajar pentingnya konsistensi status data melalui perancangan mekanisme sinkronisasi atomik, seperti fungsi penghapusan menyeluruh (*Double-Kill Deletion*), yang melatih ketelitian dalam menulis kode yang aman dari risiko data yatim (*orphaned data*).

3. Penetapan Fokus Spesialisasi Karir

Pengalaman menyelesaikan berbagai masalah teknis pada proyek MySRE membantu penulis mengenali fokus kompetensi diri. Dinamika riset yang menuntut perancangan API otonom namun terkendali telah menumbuhkan ketertarikan nyata pada rekayasa kecerdasan buatan terapan. Magang ini meyakinkan penulis untuk menetapkan arah karir profesional secara mendalam sebagai Backend Engineer dan Data Engineer yang memusatkan fokus pada arsitektur Agentic AI.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem backend MySRE pada Pusat Riset Sains Data dan Informasi BRIN, dapat ditarik beberapa kesimpulan:

1. Pembangunan arsitektur backend asinkronus menggunakan framework FastAPI terbukti menyelesaikan kendala hambatan antarmuka (UI blocking) melalui pemindahan beban kalkulasi dokumen berat ke background tasks, didukung pelaporan progres secara real-time via saluran *Server-Sent Events* (SSE).
2. Penerapan teknik *Multi-Level Chunking* berhasil membagi teks menjadi dua tingkat kerapatan (*Child Chunk* ± 200 karakter dan *Parent Chunk* ± 1500 karakter). Pemisahan ini menjaga akurasi pencarian vektor semantik Child tanpa mengorbankan konteks informasi utuh Parent saat dikirim ke model bahasa besar (*Stuffed Context*).
3. Integrasi basis data hibrida antara Supabase (untuk data relasional dan vektor) dengan Neo4j (untuk data hubungan graf) berjalan stabil. Dukungan operasi atomik sinkron, khususnya fungsi Double-Kill Deletion, berhasil mencegah terjadinya data yatim (*orphaned data*) saat terjadi penghapusan literatur riset.
4. Implementasi gerbang penyaringan kualitas bertingkat (*Multi-Stage Semantic Gating*) menekan pembentukan relasi semantik acak (*hairball graph*). Penggabungan filter statistik (*Three Sigma Rule*), dependensi tata bahasa (CARS), penyaring domain (*Domain Filtering*), dan evaluasi leksikal (*Edge Reasoning*) menjamin integritas hubungan semantik yang tersimpan di dalam Neo4j.

5.2 Keterbatasan

Pelaksanaan rekayasa backend MySRE memiliki beberapa batasan operasional:

1. Sistem masih bergantung sepenuhnya pada penyedia API model bahasa besar eksternal untuk proses ekstraksi pilar dan analisis relasi semantik. Kondisi ini menyebabkan kecepatan pemrosesan rentan terhadap ketidakstabilan latensi jaringan dan risiko kegagalan sistem saat batas kuota (rate limits) tercapai.
2. Proses komputasi embedding lokal menggunakan model intfloat/multilingual-e5-small dibatasi oleh spesifikasi kapasitas komputasi perangkat keras lokal (ThinkPad T480). Keterbatasan ini membatasi kecepatan pemrosesan apabila dokumen diunggah dalam volume besar secara bersamaan.
3. Aturan penyaringan semantik yang dikodekan secara heuristik, terutama pada Coherence Gate (aturan CARS) dan Domain Filtering, berpotensi mengabaikan hubungan semantik lintas disiplin ilmu yang tidak memuat kecocokan kata kunci leksikal standar.

5.3 Implikasi

1. **Bagi Mahasiswa:** Kegiatan magang memberikan keahlian praktis dalam memecahkan kode warisan (*legacy code*), merancang basis data graf Neo4j, serta mengelola interkoneksi basis data hibrida yang relevan dengan spesialisasi Backend dan Data Engineering.
2. **Bagi Mitra (PR SDI BRIN):** Hasil pengembangan backend “*Brain*” MySRE menyediakan modul fungsional untuk mempercepat proses asimilasi riset dan visualisasi peta pengetahuan ilmiah, membantu peneliti memetakan posisi penelitian secara terstruktur.
3. **Bagi Universitas (UIN SMH Banten):** Kerja sama ini membuktikan relevansi kurikulum Program Studi Informatika dalam mempersiapkan mahasiswa menghadapi proyek rekayasa kecerdasan artifisial dan teknologi data berskala nasional.

5.4 Saran

1. Bagi Mahasiswa (Pengembangan Sistem):

- Mengimplementasikan algoritma klusterisasi (clustering) otomatis pada basis data graf Neo4j untuk mempermudah identifikasi rumpun keilmuan secara visual.
- Mengintegrasikan sistem dengan API basis data literatur eksternal (seperti Crossref atau Semantic Scholar) menggunakan pendekatan pencarian terpusat-entitas (entity-centric search) agar penarikan artikel baru dapat berjalan otomatis menggunakan dokumen aktif sebagai jangkar kueri.

2. Bagi Mitra (PR SDI BRIN):

- Menyediakan infrastruktur server dengan dukungan unit pemrosesan grafis (GPU) mandiri untuk memfasilitasi penggunaan model bahasa lokal (local LLM), sehingga dapat menekan biaya operasional API eksternal dan menjaga keamanan data internal.

3. Bagi Universitas:

- Memperluas jangkauan kerja sama magang riset pada kelompok riset lain di lingkungan BRIN untuk memberikan wawasan teknologi terkini bagi mahasiswa informatika lainnya.

DAFTAR PUSTAKA

- [1] National Science Board, "Science and Engineering Indicators 2024," 2024.
- [2] Singh et al., "Agentic AI Architectures and Cognitive Controls," 2026.
- [3] Lu et al., "The Widening Gap in Scientific Knowledge Synthesis," 2026.
- [4] Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proceedings of NeurIPS, 2020.
- [5] Eibich et al., "ARAGOG: Advanced Retrieval-Augmented Generation for Scientific Documents," arXiv preprint arXiv:2404.12345, 2024.
- [6] Lumer et al., "Multi-Hop Reasoning Challenges in Enterprise Financial Document AI," PwC Technology Journal, vol. 3, no. 1, 2025.
- [7] Robinson et al., Graph Databases, 2nd ed. Sebastopol, CA: O'Reilly Media, 2015.
- [8] Neo4j Documentation, "Neo4j Graph Database and Vector Search Index Specification," 2025.
- [9] Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, UC Irvine, 2000.
- [10] FastAPI Documentation, "FastAPI Framework Reference Manual," 2025.
- [11] Supabase Documentation, "PostgreSQL and pgvector Storage Architectures," 2025.
- [12] Prisma Documentation, "Prisma Client and Query Engine Internals," 2025.
- [13] Tan et al., "Information Loss in Fixed-Dimension Sentence Embeddings," in Proceedings of ACL, 2024.
- [14] Wang et al., "Multilingual E5 Text Embeddings: Asymmetric Contrastive Learning at Scale," 2024.
- [15] Wang et al., "Balanced Entropy Engineering in Dense Vector Spaces," Journal of AI Research, 2025.
- [16] Luxburg, "A Tutorial on Spectral Clustering," Statistics and Computing, vol. 17, no. 4, 2007.
- [17] Luan et al., "Asymmetric Dense Retrievers for Scientific Search," in Proceedings of EMNLP, 2021.

- [18] Mysore et al., "ASPIRE: Aspiring for Faceted Document Representation in Scientific Search," in Proceedings of AAAI, 2022.
- [19] Swales, *Genre Analysis: English in Academic and Research Settings*. Cambridge: Cambridge University Press, 1990.
- [20] Zhou et al., "LLM-driven Ontological Judgments in Cross-Domain Graph Construction," *Semantic Web Journal*, 2023.
- [21] Huang et al., "Intrinsic Self-Correction Capabilities of Large Language Models," in Proceedings of ACL, 2024.
- [22] Min et al., "Factual Atomic Evaluation of Language Model Generations," in Proceedings of EMNLP, 2023.
- [23] Badan Riset dan Inovasi Nasional, *Peraturan Badan Riset dan Inovasi Nasional Nomor 10 Tahun 2022 tentang Tugas, Fungsi, dan Struktur Organisasi Riset Elektronika dan Informatika*, 2022.

LAMPIRAN

L.1 Biodata Peserta Magang (Muhammad Zaidan Rafat)

Identitas Diri:

Nama	Muhammad Zaidan Rafat
NIM	231730016
Tempat, Tanggal Lahir	Serang, 19 Juli 2005
Program Studi	Informatika
Fakultas	Sains dan Teknologi
Universitas	UIN Sultan Maulana Hasanuddin Banten
Email	muhammadzaidanrafat@gmail.com

Riwayat Pendidikan:

Jenjang	S1	SMA / SMK Sederajat
Nama Institusi	UIN Sultan Maulana Hasanuddin Banten	SMK Negeri 1 Kota Serang
Program Studi	Informatika	Teknik Komputer dan Jaringan
Tahun Masuk	2023	2020
Tahun Keluar	-	2023

Pengalaman Kerja/Magang:

Perusahaan / Instansi	PT. Krakatau Information Technology	Badan Riset dan Inovasi Nasional
Satuan Kerja	IT Infrastructure	Pusat Riset Sains Data dan Informasi
Posisi	Staff Magang	Magang Riset
Tahun	Juni 2022 - Agustus 2022	Februari 2026 - Juli 2026
Alamat	Gedung Krakatau IT, Jl. Raya Anyer No.Km.3, Warnasari, Citangkil, Cilegon, Banten	Jl. Sangkuriang, Dago, Kecamatan Coblong, Kota Bandung

Penghargaan:

No	Jenis Penghargaan	Institusi Pemberi Penghargaan	Tahun
1	Awardee Bank Indonesia Scholarship 2024 Batch 2	KPW Bank Indonesia Provinsi Banten	2024

Hak Paten / HKI:

Tahun	Judul HKI	Jenis	Nomor P/Id
2024	Rancang Bangun Sistem Informasi Pencatatan Jurnal Mengajar Dosen Berbasis Web : Solusi Monitoring Perkuliahan Di Era Digital	Hak Cipta	EC002024238345

L.2 Biodata Pembimbing Lapangan BRIN (Rio Nurtantyana, Ph.D)

Identitas Diri:

1	Nama Lengkap (dengan gelar)	Rio Nurtantyana, S.Pd., M.Pd., M.Sc., Ph.D
2	Jenis Kelamin	Laki-Laki
3	NIP/NIK/Identitas lainnya	199312152024031001
4	NIDN	-
5	Tempat dan Tanggal Lahir	Yogyakarta, 15 Desember 1993
6	E-mail	rion003@brin.go.id
7	Nomor Telepon/HP	+62 851-3215-5491
8	Nama Institusi Tempat Kerja	Pusat Riset Sains Data dan Informasi, BRIN KST Samaun Samadikun
9	Alamat Kantor	Jl. Sangkuriang, Dago, Kecamatan Coblong, Kota Bandung, Jawa Barat 40135
10	Nomor Telepon/Faks	(022) 2530050

Riwayat Pendidikan:

	S-1	S-2 (Indonesia)	S-2 (Taiwan)	S-3
Nama Perguruan Tinggi	Universitas Negeri Yogyakarta, Indonesia	Universitas Negeri Yogyakarta, Indonesia	National Central University, Taiwan	National Central University, Taiwan
Bidang Ilmu	Pendidikan Teknik Elektronika dan Informatika	Pendidikan Teknik Elektronika dan Informatika	Network Learning Technology	Network Learning Technology
Tahun Masuk-Lulus	2012-2016	2017-2019	2017-2019	2019-2023
Judul Skripsi/Tesis/Di sertasi	Pengembangan Aplikasi Mobile Direktori Smk Sebagai Informasi	Application of the Authentic-UG using Augmented Reality to	Application of the Authentic-UG using Augmented Reality to	X-Education: Education of All Things with AI and Edge Computing—

	Lokasi Sekolah Menengah Kejuruan (Smk) Di Kota Yogyakarta Pada Platform Android.	Support Geometry Learning in Authentic Learning Design	Support Geometry Learning in Authentic Learning Design	One Case Study for EFL Learning
Nama Pembimbing/Promotor	Handaru Jati, Ph.D.	1. Prof. Dr. Wu-Yuin Hwang	2. Suprpto, Ph.D	Prof. Dr. Wu-Yuin Hwang

Pengalaman Penelitian 5 Tahun Terakhir:

No.	Tahun	Judul Penelitian	Sumber Pendanaan
1	2023	AI and Recognition Technologies to Facilitate English as Foreign Language Writing for Supporting Personalization and Contextualization in Authentic Contexts.	NSTC, Taiwan
2	2023	Collaboration and interaction with smart mechanisms in flipped classrooms.	NSTC, Taiwan
3	2023	Investigating anxiety and acceptance of students and parents toward authentic contextual learning with a mobile app under the COVID-19.	NSTC, Taiwan
4	2023	X-Education: Education of All Things with AI and Edge Computing - one Case Study for EFL Learning.	NSTC, Taiwan
5	2023	Augmented Reality with Authentic GeometryGo App to Help Geometry Learning and Assessments.	NSTC, Taiwan

Publikasi Artikel Ilmiah Dalam Jurnal 5 Tahun Terakhir:

No.	Judul Artikel Ilmiah	Nama Jurnal	Penerbit
1	Hwang, W.-Y., Nurtanyana, R., Purba, S.W.D., Hariyanti, U., Indrihapsari, Y., & Surjono, H. D. (2023). AI and Recognition Technologies to Facilitate English as Foreign Language Writing for Supporting Personalization and Contextualization in Authentic Contexts. http://dx.doi.org/10.1177/07356331221137253	Journal of Educational Computing Research.	SAGE
2	Hwang, W.-Y., Nurtanyana, R., &	Data Technologies	Emerald

	Hariyanti, U. (2023). Collaboration and interaction with smart mechanisms in flipped classrooms. https://doi.org/10.1108/DTA-04-2022-0171	and Applications.	
3	Hariyanti, U., Hwang, W.-Y., Nurtanyana, R., Surjono, H. D., Rochmah, I. N., Hoang, A., Luthfi, M. I., & Maulana Putra, M. T. (2023). Investigating anxiety and acceptance of students and parents toward authentic contextual learning with a mobile app under the COVID-19. 1-13. https://doi.org/10.1080/1049482022.2160465	Interactive Learning Environments.	Taylor & Francis
4	Hwang, W.-Y., & Nurtanyana, R. (2022). X-Education: Education of All Things with AI and Edge Computing -one Case Study for EFL Learning., 14(19), 12533. https://doi.org/10.3390/su141912533	Sustainability	MDPI
5	Hwang, W.-Y., Nurtanyana, R., Purba, S.W.D., Hariyanti, & Suprpto. (2023). Augmented Reality with Authentic GeometryGo App to Help Geometry Learning and Assessments. https://doi.org/10.1109/TLT.2023.3251398	IEEE Transactions on Learning Technologies	IEEE

Pemakalah Seminar Ilmiah 5 Tahun Terakhir:

No.	Nama Temu Ilmiah/Seminar	Judul Artikel Ilmiah	Waktu dan Tempat
1	6th International Conference on Information Technology (InCIT2022). IEEE.	Hwang, W.-Y., & Nurtanyana, R. (2022). The Integration of Multiple Recognition Technologies and Artificial Intelligence to Facilitate EFL Writing in Authentic Contexts.	Thailand, 2022
2	Innovative Technologies and Learning (ICITL 2021), Lecture Notes in Computer Science. Springer.	Hwang, W.-Y., Lin, Y.-J., Hoang, A., Nurtanyana, R., & Lin, O. (2022). Facilitating Geometry Learning Through Real-Time Collaborative	Portugal, 2021

		Activities with Augmented Reality in Authentic Context. In Y.-M. Huang, S.-C. Cheng, J. Barroso, & F. E. Sandnes.	
3	Innovative Technologies and Learning (ICITL 2021), Lecture Notes in Computer Science. Springer.	Hwang, W.-Y., Nurtanyana, R., & Putra, M.T.M. (2021). Facilitating 3D Geometry Learning with Augmented Reality in Authentic Contexts. In Y.-M. Huang, C.-F. Lai, & T. Rocha (Eds.).	Portugal, 2021
4	2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress. IEEE.	Hwang, W.-Y., Nurtanyana, R., & Hildayati, J. (2021). Investigation of students' understanding of speech with smart mechanisms and scaffolding in flipped classrooms.	Canada, 2021
5	International Conferences Blended Learning 2020. Atlantis.	Nurtanyana, R., Suprpto, Hwang, W. Y., & Hariyanti, U. (2020). Developing a Learning Monitoring System Dashboard for Augmented Reality in Ubiquitous Geometry (Authentic-UG).	Yogyakarta, 2020

Karya Buku 5 Tahun Terakhir:

No.	Judul Buku	Tahun	Jumlah Halaman	Penerbit
1	AI and recognition technologies for authentic contextual learning (Book chapter in Education in the era of digital technology)	2024	10	FIP UPI

Perolehan HKI:

No.	Judul/Tema	Tahun	Jenis	No P/ID
-----	------------	-------	-------	---------

	HKI			
1	BATIKNESIA BERBASIS AR PADA PLATFORM IOS DAN ANDROID	2013	Hak Cipta	C00201301766
2	Program Komputer "APLIKASI REKOMENDA SI & INFORMASI LOKASI SEKOLAH MENENGAH KEJURUAN (SMK) "DIREKTORI SMK""	2016	Hak Cipta	C00201703984
3	Hibona Smartswitch	2018	Hak Cipta	EC00201851487
4	Arcapada	2021	Merek	IDM000862782
5	HiBona	2020	Merek	IDM000752162
6	Batiknesia	2017	Merek	IDM000577804

Pengalaman Rekayasa Sosial Lainnya 10 Tahun Terakhir:

No.	Judul/Tema/Jenis Rekayasa Sosial Lainnya yang Telah Diterapkan	Tahun	Tempat Penerapan	Respon Masyarakat
1	-	-	-	-

Penghargaan 10 Tahun Terakhir:

No.	Jenis Penghargaan	Institusi Pemberi Penghargaan	Tahun
1	Phi Tau Phi scholastic honor society	Phi Tau Phi	2023

L.3 Biodata Pembimbing Universitas (Ahmad Tabrani, M.T.I.)

Identitas Diri:

Nama	:	Ahmad Tabrani
Tempat, Tanggal Lahir	:	Tangerang, 17 September 1985
Pekerjaan	:	Dosen / PNS
Email	:	Ahmad.tabrani85@gmail.com ; ahmad.tabrani@uinbanten.ac.id
NIDN	:	2017098503
SINTA ID	:	6694015

Riwayat Pendidikan:

Tahun	Jenjang / Program Studi	Institusi
2024 – Sekarang	S3 Informatika	Universitas Amikom Yogyakarta
2022 – Sekarang	S3 Manajemen Pendidikan Islam	UIN Sultan Maulana Hasanuddin Banten
2019 – 2022	S2 Ekonomi Islam (ME)	UIN Sultan Maulana Hasanuddin Banten
2014 – 2016	S2 Teknik Informatika (MTI)	STMIK Raharja
2003 – 2007	S1 Teknik Informatika (ST)	Universitas Gunadarma

Pengalaman Pekerjaan:

Tahun	Jabatan	Instansi / Perusahaan
2018 – Sekarang	Dosen	UIN Sultan Maulana Hasanuddin Banten
2012 – 2018	Section Head IT	PT Dharma Poliplast
2011 – 2012	Senior System Engineer	PT Intikom Berlian Mustika
2007 – 2011	Network Engineer	PT Data Sari Citra

Publikasi Jurnal 5 Tahun Terakhir:

Tahun	Judul Publikasi	Jurnal / Publikasi
2024	Rancang Bangun Aplikasi Manajemen Simpatisan Sebagai Optimalisasi Pemenangan Pemilu Legislatif	SIMADA
2024	Analisis Bibliometrik Penggunaan Chat GPT Berbasis Data Google Scholar	SIMIKA
2023	Digitalisasi Pelayanan	ICIT Journal 9

	Akademik Melalui Aplikasi Simika Pada UIN Sultan Maulana Hasanuddin Banten	
2022	Analisis Kritis Kebijakan Pemerintah Tentang Undang-Undang Sisdiknas Nomor 20 Tahun 2003	JPPM Jurnal Penelitian dan Pengabdian Masyarakat 1(1)
2022	Analisis Faktor-Faktor Yang Mempengaruhi Keputusan Nasabah Melakukan Cicilan Emas Di Bank Syariah Indonesia KCP Pandeglang	MUAMALATUNA 14(1)
2021	Initiating Agropreneurship From An Islamic Perspective	-
2021	Designing a Pustekipad Computer Laboratory Loan Application for Learning ICT Subjects	MOTIVECTION: Journal of Mechanical, Electrical and Industrial Engineering 3
2020	Analisis Faktor yang Mempengaruhi Agresivitas Pajak Studi Empiris pada Perusahaan LQ 45 yang Terdaftar di Bursa Efek Indonesia Tahun 2015–2018	Jurnal REKOMEN: Riset Ekonomi Manajemen 4(1)
2020	Pengaruh Tingkat Suku Bunga Deposito Terhadap Dana Pihak Ketiga Di Bank BJB Cabang Rangkasbitung	Jurnal Manajemen Bisnis Krisnadwipayana 8(2)

Publikasi Buku 5 Tahun Terakhir:

Tahun	Judul Buku	Penerbit
2021	Kampus Bebas Kertas: Studi atas Pelayanan Akademik Secara Digital di UIN Sultan Maulana Hasanuddin Banten	Farha
2022	Jurus Jitu Meningkatkan Webometrics	Harfa Kreatif
2024	Basis Data	Ruang Karya
2024	Jaringan Komputer	Ruang Karya

Hak Paten / HKI:

Tahun	Judul HKI	Jenis	Nomor P/Id
2024	Rancang Bangun Sistem Informasi Pencatatan Jurnal	Hak Cipta	EC002024238345

	Mengajar Dosen Berbasis Web : Solusi Monitoring Perkuliahan Di Era Digital		
2021	Kampus Bebas Kertas: Studi Atas Pelayanan Akademik Secara Digital di UIN Sultan Maulana Hasanuddin Banten	Hak Cipta	EC00202182858

L.4 Dokumentasi Pertemuan Progres (Progress Meetings) MBKM

Berikut adalah linimasa dokumentasi pertemuan progres mingguan yang dilaksanakan bersama Kelompok Riset HCI-V PR SDI BRIN Bandung selama masa magang:



Dokumentasi 1: Silaturahmi Perdana (Jumat, 20 Februari 2026)



Dokumentasi 2: Meeting Progress (Selasa, 24 Februari 2026)



Dokumentasi 3: Meeting Progress (Senin, 02 Maret 2026)



Dokumentasi 4: Meeting Progress (Senin, 9 Maret 2026)



Dokumentasi 5: Meeting Progress (Senin 30 Maret 2026)



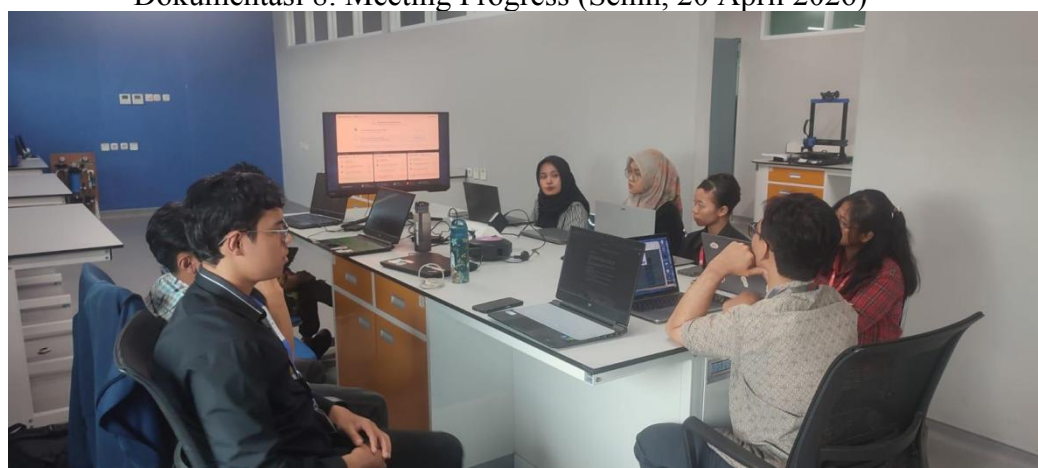
Dokumentasi 6: Meeting Progress (Senin, 06 April 2026)



Dokumentasi 7: Meeting Progress (Senin, 13 April 2026)



Dokumentasi 8: Meeting Progress (Senin, 20 April 2026)



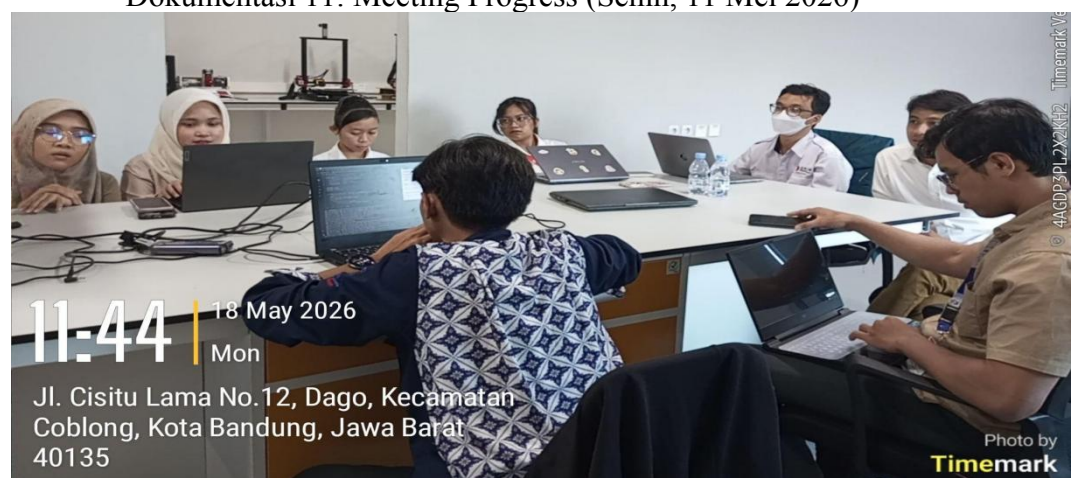
Dokumentasi 9: Meeting Progress (Senin, 28 April 2026)



Dokumentasi 10: Meeting Progress (Selasa, 05 Mei 2026)



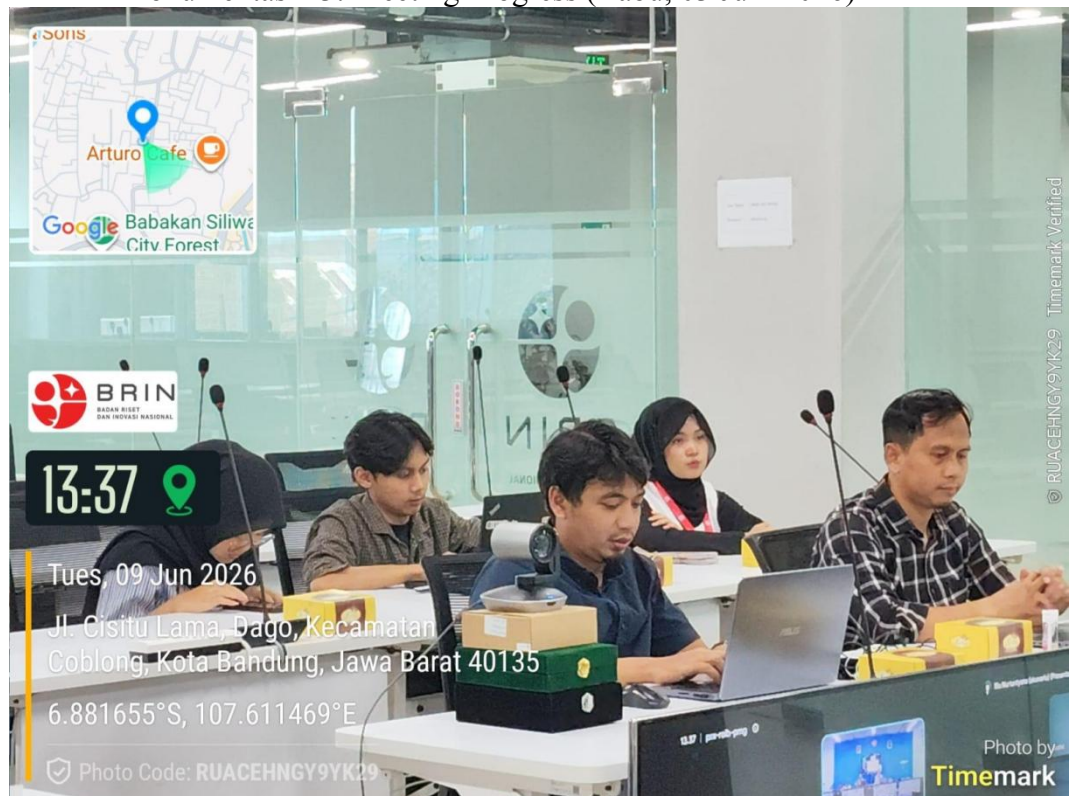
Dokumentasi 11: Meeting Progress (Senin, 11 Mei 2026)



Dokumentasi 12: Meeting Progress (Senin, 18 Mei 2026)



Dokumentasi 13: Meeting Progress (Rabu, 03 Juni 2026)



Dokumentasi : Demonstrasi feat Expert (Selasa, 09 Juni 2026)