

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem pendidikan saat ini sedang mengalami pergeseran besar, dari model pengajaran konvensional yang bersifat kaku atau "*one size fits all*" menjadi pendekatan yang lebih personal [1], [2]. Evolusi ini bermula dari digitalisasi materi pembelajaran yang dulunya hanya sekadar tumpukan dokumen statis seperti PDF, namun kini mulai memanfaatkan data pendidikan dalam skala besar untuk memahami kebutuhan pelajar [3]. Di era kecerdasan buatan (AI) saat ini, sistem pembelajaran adaptif menjadi kunci untuk menciptakan jalur belajar yang dinamis dan sesuai dengan kecepatan pemahaman tiap individu secara mandiri [4], [5]. Namun, tantangan terbesarnya bukan lagi sekadar memberikan jawaban instan, melainkan bagaimana AI mampu memahami urutan logika atau logika pengajaran yang tepat seperti yang dilakukan seorang tutor [6]. Oleh karena itu, AI masa kini dituntut untuk mengadopsi metode *Socratic*, untuk proses berpikir pengguna secara sistematis daripada hanya memberikan solusi akhir tanpa proses belajar [7]. Namun, efektivitas tetap bergantung pada infrastruktur basis data yang dapat menyimpan dan menjaga urutan logika materi secara konsisten.

Kebutuhan ini menjadi semakin mendesak khususnya pada pembelajaran pemrograman Python, mengingat skala adopsi AI dalam aktivitas pembelajaran secara umum yang terus meningkat dalam beberapa tahun terakhir [8], termasuk secara spesifik pada aktivitas belajar coding [9]. Kurikulum pemrograman seperti Python pada dasarnya tersusun sebagai rantai keterampilan dan konsep yang berjenjang, di mana penguasaan satu keterampilan menjadi prasyarat bagi keterampilan berikutnya [10]. Konteks pemrograman Python inilah yang dipilih sebagai studi kasus utama penelitian ini untuk menguji bagaimana arsitektur basis data dapat menjaga konsistensi urutan tersebut.

Meskipun sistem pembelajaran digital terus berkembang, infrastruktur penyimpanan data yang digunakan sebagian besar sistem saat ini masih mengandalkan *database* relasional atau RDBMS. Struktur tabel yang kaku

membuat data kurikulum yang saling berkaitan harus dipaksa ke dalam baris dan kolom yang terpisah, sehingga ketika relasi antar materi semakin kompleks, sistem harus menggabungkan banyak tabel secara berulang dan performa berpotensi terbebani seiring bertambahnya volume data [3], [11], [12]. Salah satu cara yang umum digunakan untuk menyiasati keterbatasan ini adalah *recursive Common Table Expression* (CTE), namun stabilitasnya cenderung menurun seiring bertambahnya kedalaman dan volume data kurikulum [11], [12].

Selain kendala infrastruktur, tantangan lain muncul dari cara AI mencari informasi. Sistem *Retrieval-Augmented Generation* (RAG) yang berbasis *vector database* memang bagus dalam mencocokkan kemiripan makna teks, namun lemah dalam memahami urutan logis antar materi yang bersifat hierarkis [13], [14], [15]. Ini terjadi karena LLM pada dasarnya bekerja secara probabilitas, yaitu menebak respons berdasarkan kemiripan teks tanpa mengenal aturan prasyarat [7], [16].

Akibatnya, banyak platform AI yang sudah menyediakan panduan belajar justru mengalami halusinasi urutan materi, yaitu kondisi di mana sistem menyajikan materi lanjutan sebelum konsep dasar dikuasai karena tidak ada mekanisme yang mengunci urutan secara deterministik di level *database* [7], [16]. Masalah ini semakin besar ketika sistem juga tidak mampu mengenali kemampuan awal setiap pengguna, sehingga bimbingan yang diberikan tetap seragam dan materi yang disajikan bisa terlalu mudah atau terlalu sulit bagi masing-masing pelajar, yang pada akhirnya berpotensi menimbulkan kelelahan belajar dan kebiasaan bergantung pada jawaban instan tanpa benar-benar memahami konsepnya [1], [2], [17], [18].

Berbagai penelitian telah mencoba menjawab tantangan ini dengan menggabungkan *Knowledge Graph* dan *Vector RAG* untuk meningkatkan akurasi *retrieval*, atau mengintegrasikan *knowledge graph* dengan LLM untuk memperkuat model dialog berbasis metode *Socratic* [7], [16], [18]. Namun, solusi tersebut mayoritas masih memisahkan antara logika pembelajaran dan lapisan penyimpanan data, sehingga pelacakan penguasaan materi berjalan sebagai modul eksternal yang terpisah dari arsitektur *database*-nya dan menimbulkan potensi latensi tinggi saat memproses urutan materi kompleks dalam skala besar [3], [5], [13].

Di tengah keterbatasan tersebut, minat publik dan akademik terhadap integrasi basis data graf dengan RAG serta metode pelacakan penguasaan materi berbasis

probabilistik justru meningkat tajam dalam dua tahun terakhir. Data Google Trends menunjukkan indeks pencarian "knowledge graph RAG" naik dari rata-rata 2,5 menjadi 100 (40,6 kali lipat), "graph database" naik dari 10,6 menjadi 100 (9,4 kali lipat), dan "bayesian knowledge tracing" yang sebelumnya nyaris nol kini mencapai indeks 100, seluruhnya pada paruh pertama 2026 [19]. Sejalan dengan temuan dari trend di sisi akademik, dimana survei integrasi graf dan RAG telah dikutip lebih dari 250 kali dalam kurang dari dua tahun, Neo4j menjadi sistem peringkat pertama Graph DBMS menurut DB-Engines Ranking, dan pemodelan berbasis graf menjadi satu dari enam arah riset pelacakan penguasaan materi yang berkembang, melengkapi kontribusi pyBKT dan survei metode pelacakan penguasaan materi [20], [21], [22], [23], [24].

Untuk menjawab keterbatasan pada penelitian terdahulu, penelitian ini mengusulkan arsitektur *hierarchical graph database* yang memanfaatkan fitur *native vector indexing* pada Neo4j untuk menyatukan logika urutan belajar langsung ke dalam lapisan penyimpanan data [25], [26]. Algoritma *Bayesian Knowledge Tracing* (BKT) ditanamkan langsung di level basis data sebagai mekanisme pelacak kemajuan belajar pengguna, bukan sebagai modul terpisah di lapisan aplikasi [16], [27]. Dengan memanfaatkan indeks vektor bawaan di dalam struktur *graph*, arsitektur ini memungkinkan database menentukan jalur belajar secara langsung berdasarkan profil kognitif pengguna, tanpa perlu sinkronisasi data antar platform [18], [28].

Penelitian ini berbeda dari penelitian sebelumnya karena menggeser kendali logika urutan belajar dari lapisan aplikasi (*backend*) langsung ke tingkat basis data (*storage-level logic*), di mana aturan prasyarat (*prerequisite*) menjadi bagian utama dari struktur data itu sendiri [12], [13]. Dengan cara ini, respons tutor tidak lagi sepenuhnya bergantung pada probabilitas teks, melainkan dibatasi oleh kondisi aktual kemajuan belajar pengguna, sehingga risiko halusinasi urutan materi dapat diminimalkan [7], [16].

Dalam prosesnya penelitian ini mengadaptasi metodologi *Database System Development Life Cycle* (DBSDLC) untuk menjamin pembangunan arsitektur yang sistematis, mulai dari pemodelan *graph* hingga implementasi fisik pada Neo4j. Untuk mengevaluasi kesiapan arsitektur yang diusulkan, evaluasi dilakukan

melalui tiga dimensi utama, yaitu performa sistem, akurasi adaptivitas, dan kualitas *retrieval*. Dari sisi performa, kesesuaian performa Neo4j akan dievaluasi dalam mendukung *pipeline* RAG adaptif, diukur dari rata-rata latensi kueri, konsistensi waktu respons, dan stabilitasnya saat menelusuri rantai prasyarat materi pemrograman, dengan PostgreSQL + pgvector sebagai baseline referensi struktural [11], [12]. Keberhasilan fitur *adaptive learning* akan divalidasi melalui kemampuan *Bayesian Knowledge Tracing* (BKT) dalam memetakan profil kognitif pengguna secara akurat dan dinamis [5], [27]. Selain itu, kualitas *retrieval* sistem akan dievaluasi menggunakan framework RAGAS untuk mengukur sejauh mana pendekatan Graph-Augmented RAG mampu menghasilkan konteks yang relevan dan sesuai urutan kurikulum, dengan *Vector-Only* sebagai baseline referensi struktural.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana performa Neo4j dalam mendukung *pipeline* RAG adaptif, diukur dari rata-rata latensi kueri, konsistensi waktu respons, dan stabilitasnya saat menelusuri rantai prasyarat materi pemrograman Python?
2. Sejauh mana sistem Graph-Augmented RAG mampu menghasilkan konteks yang relevan dan akurat untuk materi pemrograman Python yang bersifat hierarkis, diukur menggunakan metrik *Faithfulness*, *Context Precision*, dan *Answer Relevance*?
3. Seberapa akurat implementasi algoritma *Bayesian Knowledge Tracing* di level basis data Neo4j dalam memperbarui *mastery state* pengguna berdasarkan simulasi skenario jawaban yang telah ditentukan?

1.3 Batasan Masalah

Agar penelitian tetap fokus pada optimalisasi arsitektur *database* dan kualitas pembelajaran adaptif, batasan masalah ditetapkan sebagai berikut:

1. Data yang digunakan adalah dataset kurikulum pemrograman python *open source* materi dari platform edukasi teknologi terpercaya dengan volume minimal 1.000 *record* berupa potongan materi atau *learning objectives*.
2. Sistem menggunakan model bahasa Qwen2.5:7B yang berjalan secara lokal tanpa koneksi layanan internet eksternal, sehingga hasil pengukuran kecepatan pada RQ1 murni mencerminkan perbedaan arsitektur basis data, bukan gangguan dari kondisi jaringan.
3. Penggunaan layanan AI eksternal dalam penelitian ini mencakup dua peran yang terpisah. Pertama, Claude Sonnet 4.6 berperan sebagai alat bantu pada tahap pembangunan dataset, mencakup ekstraksi dan penataan konten materi, penerjemahan ke Bahasa Indonesia, serta validasi relasi prasyarat. Kedua, Claude Sonnet 4.6 juga digunakan sebagai model penilai independen (*RAGAS judge*) pada tahap evaluasi kualitas *retrieval* (RQ2), terpisah dari *pipeline* sistem utama. Dalam kedua peran tersebut, Claude Sonnet 4.6 tidak terlibat dalam proses inferensi sistem saat runtime, yang sepenuhnya dijalankan oleh Qwen2.5:7b secara lokal. Penggunaan model yang sama pada kedua peran tersebut, yaitu sebagai penyusun dataset dan sebagai judge evaluasi, diakui sebagai batasan dalam penelitian ini karena berpotensi membawa bias preferensi penilaian yang konsisten dengan gaya penataan datanya sendiri. Implikasi batasan ini dibahas lebih lanjut pada bagian Diskusi beserta saran mitigasinya pada bagian akhir penelitian.
4. Fokus utama implementasi adalah Neo4j sebagai basis data graf yang menyimpan aturan urutan materi secara eksplisit. PostgreSQL digunakan secara terbatas hanya sebagai pembanding referensi pada RQ1 saja, bukan sebagai sistem utama penelitian.
5. Pelacakan kemajuan belajar pengguna dibatasi pada estimasi status penguasaan materi menggunakan algoritma BKT yang berjalan langsung di dalam basis data graf, sehingga seluruh logika adaptivitas tidak bergantung pada modul pemrosesan di luar basis data.
6. Interaksi pengguna dengan sistem dibatasi pada pendekatan *Socratic Tutoring*, di mana setiap respons LLM dikontrol oleh jalur urutan materi dan status penguasaan yang tersimpan di basis data. Sistem hanya

memberikan pertanyaan pemantik dan petunjuk berpikir, bukan jawaban langsung, dan pembatasan ini diterapkan di level struktur data bukan sekadar aturan teks pada model LLM.

7. Sistem menggunakan Bahasa Indonesia sebagai bahasa pengantar utama, dengan tetap mempertahankan istilah teknis pemrograman dalam Bahasa Inggris untuk menjaga akurasi terminologi.
8. Evaluasi sistem dibatasi pada tiga dimensi yang selaras langsung dengan rumusan masalah: kecepatan dan stabilitas basis data graf (RQ1), kualitas konteks yang dihasilkan sistem pencarian (RQ2), dan ketepatan pelacakan *progress* belajar pengguna (RQ3).
9. Validasi manual terhadap relasi *prerequisite* antar materi dilakukan secara iteratif oleh satu peneliti sepanjang proses pembangunan dataset, tanpa keterlibatan reviewer lain, sehingga keputusan kelayakan suatu relasi *prerequisite* berpotensi dipengaruhi oleh interpretasi subjektif peneliti.

1.4 Tujuan dan Manfaat Penelitian

Berdasarkan identifikasi masalah yang telah dipaparkan pada bagian sebelumnya, penelitian ini diarahkan untuk memberikan solusi teknis terhadap keterbatasan sistem pembelajaran saat ini dalam mengelola data kurikulum yang kompleks dan bercabang. Adapun tujuan dari penelitian ini adalah sebagai berikut:

1.4.1 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan, tujuan dari penelitian ini adalah:

1. Membangun sistem *graph database* hirarkis menggunakan Neo4j yang mengorganisasi materi pemrograman secara bertingkat dengan aturan urutan materi (*prerequisite*) yang tersimpan langsung di dalam basis data, sebagai fondasi *pipeline* RAG adaptif pada sistem *Socratic Tutor*.
2. Merancang struktur data berlapis yang memisahkan antara logika urutan kurikulum dan konten materi, serta menyimpan profil kemampuan pengguna sebagai bagian dari arsitektur basis data untuk mendukung penyesuaian konteks pembelajaran secara personal.

3. Mengukur kesiapan Neo4j dalam mendukung sistem tutor adaptif melalui pengujian kecepatan dan konsistensi waktu respons saat menelusuri rantai urutan materi pemrograman, dengan PostgreSQL sebagai pembanding referensi.
4. Memvalidasi bahwa pendekatan pencarian informasi yang dipandu struktur graf hirarkis mampu menghasilkan konteks yang relevan dan akurat dari sisi pembelajaran untuk materi pemrograman Python yang bersifat hierarkis, diukur menggunakan framework RAGAS.
5. Memvalidasi ketepatan algoritma *Bayesian Knowledge Tracing* (BKT) dalam melacak dan memperbarui status penguasaan materi pengguna yang tersimpan langsung di dalam basis data graf, melalui simulasi skenario belajar dengan jawaban yang telah ditentukan hasilnya.

Perlu ditegaskan bahwa ketiga rumusan masalah yang ditetapkan tidak mengevaluasi perilaku Socratic Tutor secara langsung, melainkan menguji kesiapan infrastruktur basis data sebagai fondasi deterministik yang menjamin perilaku Socratic dapat berjalan secara konsisten dan terstruktur. RQ1 memvalidasi bahwa Neo4j mampu menelusuri rantai prasyarat materi secara real-time dengan latensi yang stabil, RQ2 membuktikan bahwa konteks yang diambil oleh sistem sesuai urutan kurikulum dan tidak memicu halusinasi urutan, serta RQ3 mengonfirmasi bahwa status penguasaan pengguna terlacak secara akurat sehingga sistem mengetahui kapan transisi ke materi berikutnya dapat dilakukan. Dengan demikian, ketiga RQ bersifat saling melengkapi dan membentuk satu rantai validasi utuh terhadap arsitektur database yang menjadi dasar berjalannya sistem *Socratic Tutor* adaptif ini.

1.4.2 Manfaat Penelitian

Manfaat yang diharapkan dari hasil penelitian ini adalah sebagai berikut, yang mencakup manfaat akademis dan manfaat praktis.

1. Bagi Pengguna, Memberikan pengalaman belajar yang lebih personal dan mudah dipahami melalui analogi yang relevan dengan latar belakang harian mereka, sehingga dapat menekan beban kognitif (*cognitive load*).

2. Bagi Bidang Pendidikan, Menyediakan solusi teknologi yang dapat mengurangi risiko miskonsepsi dalam belajar mandiri melalui penerapan *deterministic constraints* yang menjaga urutan logika materi.
3. Bagi Pengembang & Akademisi, Memberikan referensi arsitektur *Integrated Multi-modal Database* yang lebih efisien dalam menangani hubungan data yang kompleks dan bercabang banyak dibandingkan dengan basis data relasional tradisional.
4. Bagi Instansi/Organisasi, Menyediakan model sistem bimbingan cerdas yang mampu mengidentifikasi hambatan belajar pengguna secara otomatis guna meningkatkan efektivitas penyampaian materi.

1.5 Sistematika Penulisan

Untuk mencapai tujuan penelitian dan memudahkan pembaca dalam memahami isi penelitian, laporan skripsi ini disusun dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini menguraikan latar belakang penelitian mengenai pentingnya sistem pembelajaran yang dapat menyesuaikan diri dengan kemampuan setiap individu secara mandiri. Di dalamnya dijelaskan mengenai tantangan dalam penggunaan cara penyimpanan data tradisional yang sering mengalami kendala saat memproses hubungan materi pelajaran yang kompleks dan bercabang. Bab ini juga memuat rumusan masalah mengenai perancangan struktur data yang lebih efektif, batasan masalah agar penelitian tetap fokus pada bidang materi pemrograman, serta tujuan dan manfaat penelitian dalam memberikan solusi bagi proses bimbingan belajar mandiri yang lebih terarah dan akurat.

2. BAB II LANDASAN TEORI

Bab ini menyajikan dasar-dasar teori serta hasil penelitian terdahulu yang mendukung topik arsitektur penyimpanan data berbasis *graph* untuk mengoptimalkan proses pencarian informasi. Pembahasan mencakup teori mengenai cara kerja sistem *graph database*, metode untuk melacak tingkat pemahaman pengguna secara otomatis, serta teknik bimbingan belajar yang menuntun logika berpikir tanpa langsung memberikan jawaban jawaban instan.

Selain itu, dijelaskan pula perangkat lunak dan peralatan pendukung yang digunakan untuk membangun sistem tersebut.

3. BAB III METODOLOGI PENELITIAN

Bab ini memaparkan langkah-langkah sistematis yang diambil dalam pelaksanaan penelitian. Bagian ini diawali dengan penjelasan mengenai objek penelitian, yaitu kumpulan data materi pembelajaran pemrograman. Kemudian, dijelaskan tahapan pengembangan sistem, mulai dari tahap perencanaan, pemodelan, hingga pembuatan struktur penyimpanan data yang diinginkan. Terakhir, bab ini menguraikan teknik pengumpulan dan analisis data untuk mengukur kecepatan respon sistem, ketepatan dalam menyesuaikan materi dengan kemampuan pengguna, serta kualitas pengajaran yang dihasilkan.

4. BAB IV ANALISIS DAN HASIL PENELITIAN

Bab ini menguraikan inti dari penelitian yang dilakukan, mulai dari tahap analisis hingga evaluasi hasil. Mulai dengan Analisa Masalah, yang mencakup pemetaan proses belajar saat ini serta kebutuhan pengguna agar sistem dapat berfungsi dengan baik. Selanjutnya, dibahas mengenai Perancangan Sistem, yang meliputi pemodelan alur logika, perancangan struktur penyimpanan data berbasis *graph*, serta desain antarmuka bagi pengguna. Bagian berikutnya menjelaskan proses Uji Coba dan Implementasi untuk memastikan sistem berjalan sesuai rencana. Bab ini diakhiri dengan Hasil dan Diskusi, yang memaparkan temuan penelitian terkait kesiapan arsitektur.

5. BAB V SIMPULAN DAN SARAN

Bab ini merupakan bagian penutup yang menyajikan Simpulan dan Saran. Bagian kesimpulan merangkum seluruh hasil penelitian dan menjawab pertanyaan yang diajukan pada rumusan masalah, khususnya mengenai kelayakan arsitektur graf sebagai fondasi deterministik sistem pembelajaran adaptif. Sementara itu, bagian saran memberikan masukan bagi pengembangan sistem di masa mendatang agar dapat mencakup cakupan materi yang lebih luas atau fitur-fitur tambahan yang belum tersentuh dalam penelitian ini.