# BAB II

# LANDASAN TEORI

## 2.1 Penelitian Terkait

Tabel 2.1 Penelitian Terkait

| Informasi Literatur & Referensi | Hasil Akhir Penelitian | Hubungan Relevansi dengan Arsitektur AuditRAG |
|---|---|---|
| [9] FinQA: A *Dataset* of Numerical Reasoning over Financial Data<br><br>Penulis: Z. Chen et al.<br><br>Publikasi: EMNLP (Association for Computational Linguistics)<br><br>Tahun: 2021 | Studi ini mengungkap bahwa model bahasa standar hanya mencapai akurasi 50% (setara tebakan acak) pada soal keuangan yang kompleks, namun meningkat menjadi 61,24% dengan modul penalaran program. | Dasar Justifikasi Algoritma Penalaran: Penelitian ini menjadi landasan urgensi utama mengapa AuditRAG tidak bisa langsung menyuruh LLM menebak angka. Hasil ini mendasari keputusan untuk membangun *pipeline* penalaran eksplisit berbasis struktur Audit-*Chain-of-Thought* (Audit-CoT) guna menjembatani kesenjangan ekstraksi teks dan kalkulasi numerik. |
| [19] TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance<br><br>Penulis: F. Zhu et al.<br><br>Publikasi: ACL (Association for Computational Linguistics)<br><br>Tahun: 2021 | Model TagOp yang menggabungkan ekstraksi tabel dan teks secara hibrida mencapai F1-Score 58,5%, jauh mengungguli model yang hanya membaca teks saja. | Justifikasi Pendekatan Retrieval hibrida: Mengonfirmasi bahwa laporan keuangan PDF tidak bisa dibaca sepotong-sepotong. Paper ini memvalidasi keputusan teknis AuditRAG untuk menerapkan arsitektur *Hybrid retrieval* yang mengekstrak dan memproses struktur matriks tabel dan teks narasi secara simultan. |
| [20] LoRA: *Low-rank adaptation* of Large Language Models<br><br>Penulis: E. J. Hu et al.<br><br>Publikasi: ICLR (International Conference on Learning Representations)<br><br>Tahun: 2022 | Teknik LoRA mampu mengurangi jumlah parameter yang perlu dilatih hingga 10.000x dan kebutuhan memori GPU hingga 3x lipat, sambil mempertahankan performa yang setara dengan full *fine-tuning* pada model GPT-3. | Landasan Efisiensi Komputasi Lokal: Menjadi landasan teknis utama yang diadopsi AuditRAG untuk memproses model bahasa berukuran besar seperti Qwen-3 8B secara lokal. Paper ini melandasi penggunaan parameter efisien untuk memangkas VRAM komputasi tanpa mendegradasi performa model generatif. |

| Informasi Literatur & Referensi | Hasil Akhir Penelitian | Hubungan Relevansi dengan Arsitektur AuditRAG |
|---|---|---|
| [21] FinGPT: *Open-source* Financial Large Language Models<br><br>Penulis: H. Yang, X.-Y. Liu, dan C. D. Wang<br><br>Publikasi: IJCAI Symposium on Large Language Models for Financial Services<br><br>Tahun: 2023 | Menunjukkan bahwa model *open-source* (seperti Llama/ChatGLM) yang di-*fine-tune* dengan LoRA pada data keuangan dapat mencapai skor sentimen dan ekstraksi yang sebanding dengan BloombergGPT, dengan biaya pelatihan kurang dari $300. | Validasi Strategi Domain-*Specific Tuning*: Memvalidasi hipotesis AuditRAG bahwa optimasi model skala kecil yang dispesialisasi secara lokal (*Domain-Specific Fine-tuning*) mampu menandingi atau mengungguli kapabilitas model komersial masif yang bersifat umum dalam penanganan data finansial. |
| [22] Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding<br><br>Penulis: Z. Wang et al.<br><br>Publikasi: ICLR (International Conference on Learning Representations)<br><br>Tahun: 2024 | Metode transformasi tabel bertahap (*step-by-step*) meningkatkan akurasi pada *benchmark* TabFact dan WikiTableQuestions secara signifikan, mengungguli metode Chain-of-Thought standar pada data terstruktur. | Inspirasi Pengembangan Modul Parser: Logika transformasi tabel secara sekuensial ini diadopsi ke dalam perancangan arsitektur Query Intent Parser dan penulisan struktur penataan *chunk* berbasis elemen terstruktur pada komponen pra-pemrosesan data AuditRAG. |
| [15] Lost in the Middle: How Language Models Use Long Contexts<br><br>Penulis: N. F. Liu et al.<br><br>Publikasi: Transactions of the Association for Computational Linguistics<br><br>Tahun: 2024 | Kinerja LLM menurun drastis (lebih dari 20%) ketika informasi kunci berada di tengah-tengah dokumen input yang panjang, bukan di awal atau akhir. | Justifikasi Penggunaan Komponen *Reranker*: Menjadi dasar pembenaran ilmiah mengapa AuditRAG wajib menyertakan modul *Reranker* (BGE-M3). Komponen ini krusial untuk memaksa dokumen sumber yang paling relevan naik ke urutan teratas (top-k) agar tidak tenggelam di tengah konteks panjang. |
| [23] Chain-of-Verification Reduces Hallucination in Large Language Models<br><br>Penulis: S. Dhuliawala et al.<br><br>Publikasi: Findings of the Association for Computational Linguistics (ACL)<br><br>Tahun: 2024 | Mekanisme verifikasi mandiri (*Self-Verification*) berhasil mengurangi tingkat halusinasi pada tugas berbasis fakta (Wikidata) dengan margin 20-30% dibandingkan *prompting* standar. | Dasar Pengembangan Kerangka Kerja Audit-CoT: Paradigma Chain-of-Verification ini dimodifikasi dan diintegrasikan langsung ke dalam inti instruksi penalaran Audit-CoT, di mana model diwajibkan melakukan pencarian baris, penyalinan verbatim, dan validasi silang sebelum merilis angka final. |

| Informasi Literatur & Referensi | Hasil Akhir Penelitian | Hubungan Relevansi dengan Arsitektur AuditRAG |
|---|---|---|
| [24] Self-Consistency Improves Chain of Thought Reasoning in Language Models<br><br>Penulis: X. Wang et al.<br><br>Publikasi: ICLR (International Conference on Learning Representations)<br><br>Tahun: 2023 | Strategi pengambilan suara mayoritas (*majority voting*) dari beberapa output model meningkatkan akurasi matematika pada *dataset* GSM8K dari 56,5% menjadi 74,4%. | Landasan Pemilihan Metrik Evaluasi: Teori konsistensi jalur penalaran ini diadopsi secara langsung menjadi dasar pengembangan metrik evaluasi internal AuditRAG, yaitu *Consistency Score* (CS), untuk mengukur tingkat stabilitas deterministik sistem keuangan. |
| [25] PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance<br><br>Penulis: Q. Xie et al.<br><br>Publikasi: ACL (Association for Computational Linguistics)<br><br>Tahun: 2023 | Model FinMA 7B yang dilatih dengan instruksi (instruction tuning) mengungguli BloombergGPT 50B pada tugas prediksi kredit dan QA keuangan. | Validasi Strategi Pembangunan *Dataset* Sintetik: Memberikan pembenaran ilmiah terhadap efektivitas rekayasa data instruksi finansial. Referensi ini mendasari keputusan pembuatan *dataset* sintetik QGA (Question-Guidance-Answer) berstruktur tinggi menggunakan Gemini sebagai generator data latih. |
| [26] Program of Thoughts *Prompting*: Disentangling Computation from Reasoning for Numerical Reasoning Tasks<br><br>Penulis: W. Chen, X. Ma, X. Wang, dan W. W. Cohen<br><br>Publikasi: Transactions on Machine Learning Research<br><br>Tahun: 2023 | Memisahkan logika penalaran (teks) dari komputasi (kode) meningkatkan akurasi numerik rata-rata sebesar 12% dibandingkan metode standar. | Tolok Ukur Target Peningkatan Kinerja: Digunakan oleh AuditRAG sebagai landasan komparatif konseptual dalam memetakan target peningkatan akurasi ekstraksi nilai numerik laporan keuangan melalui pemisahan langkah logika analitik yang ketat. |
| [27] AuditRAG: An ARAG Approach for Auditing Dutch *Annual report*s: Ensuring Compliance<br><br>Penulis: L. Bijvoet<br><br>Publikasi: Master's thesis, Vrije Universiteit Amsterdam<br><br>Tahun: 2025 | Sistem ARAG multi-agen terbukti efektif untuk verifikasi kepatuhan laporan tahunan Belanda dengan F1-score 0,74 dan presisi 89% pada pengujian model Llama 3.3-70B-Instruct. | Justifikasi Diferensiasi Pendekatan: Penelitian rujukan ini menggunakan pendekatan orkestrasi multi-agen (ARAG) berskala besar untuk menilai kepatuhan dokumen hukum. Hal ini mempertegas novelty AuditRAG yang berfokus pada ekstraksi nilai numerik presisi menggunakan arsitektur Entity-Centric RAG tunggal dengan logika Audit-CoT yang di-*fine-tune* menggunakan QLoRA pada model berukuran ringkas (Qwen-3 8B). |

Sebelum membahas aspek teknis sistem, penting untuk memahami mengapa metrik keuangan tertentu seperti *Net income* (Laba Bersih), *Total Asset* (Total Aset), dan *Revenue* (Pendapatan) menjadi fokus utama dalam penelitian ini. Pemahaman ini menjadi landasan bisnis (*business knowledge*) yang mendasari perancangan sistem AuditRAG, karena sistem ini dirancang untuk membantu analis keuangan dalam mengekstrak dan memverifikasi metrik-metrik tersebut dari laporan tahunan.

*Net income* (Laba Bersih) merupakan indikator utama profitabilitas perusahaan. Penelitian terbaru oleh Abujabal (2025) menunjukkan bahwa *Income from Continuing Operations* (yang komponen utamanya adalah *Net income*) memiliki relevansi tertinggi dibandingkan lima indikator kinerja lainnya dalam memprediksi return saham di Bursa Palestina [28]. Temuan ini memperkuat pentingnya verifikasi laba bersih dalam proses audit, karena angka ini secara langsung mempengaruhi persepsi investor terhadap kinerja perusahaan. Lebih lanjut, Singh dan Khatua (2025) menemukan bahwa profitabilitas secara konsisten berasosiasi dengan respons pasar positif dan peningkatan valuasi perusahaan berdasarkan analisis 353 perusahaan di Bursa Efek India selama periode 2014-2024 [29].

*Total Asset* (Total Aset) menunjukkan seluruh sumber daya yang dikuasai perusahaan, baik yang bersifat lancar (kas, piutang, persediaan) maupun tidak lancar (pabrik, tanah, peralatan). Noordermeer dan Vorst (2025) dalam penelitian mereka di *Management Science* membuktikan bahwa model yang menggunakan informasi aset terdisagregasi (memisahkan properti, pabrik, peralatan, aset tidak berwujud, dan aset operasi lancar) memiliki akurasi yang lebih tinggi dalam memprediksi pertumbuhan aset operasi dan pendapatan masa depan dibandingkan model yang hanya menggunakan informasi agregat [30]. Temuan ini menunjukkan bahwa detail komponen aset sangat informatif bagi analis keuangan. Selain itu, Zhang dan Wang (2025) juga menjelaskan bahwa evaluasi mendalam terhadap

komponen aset dan kinerja korporasi merupakan kunci utama dalam menilai kesehatan keuangan suatu perusahaan [31].

*Revenue* (Pendapatan) adalah garis depan (top line) dari laporan laba rugi. Hunt, Myers, dan Sougiannis (2025) dalam *Journal of Accounting and Economics* menemukan bahwa pendapatan dan profitabilitas dapat diprediksi secara lebih akurat menggunakan pendekatan *machine learning* dibandingkan model statistik tradisional, dengan peningkatan akurasi hingga 6,977% di luar sampel [32]. Penelitian oleh Petinga (2025) menganalisis 50 perusahaan Eropa terbesar dan menemukan bahwa KPI terkait pendapatan, profitabilitas, dan efisiensi operasional menjadi instrumen kunci dalam pengambilan keputusan manajemen, terutama di sektor padat modal seperti keuangan dan utilitas [33]. Temuan ini memperkuat pentingnya verifikasi pendapatan dalam proses audit, karena pendapatan sering menjadi target manipulasi.

Dalam konteks audit laporan keuangan, ketiga metrik ini saling terkait. Seorang auditor tidak cukup hanya mengetahui angka satu per satu; ia harus mampu memverifikasi apakah laba bersih yang dilaporkan konsisten dengan total aset yang dimiliki, serta apakah pertumbuhan pendapatan didukung oleh peningkatan aset produktif. Zhang dan Wang (2025), menyatakan bahwa untuk mengambil keputusan strategis, kita perlu menyatukan berbagai data keuangan agar mendapatkan gambaran analisis yang menyeluruh [31]. Oleh karena itu, sistem AuditRAG dirancang untuk tidak hanya mengekstrak angka-angka tersebut, tetapi juga mempertahankan hubungan logis antar metrik melalui metode *Audit-Chain-of-Thought* yang mensimulasikan proses berpikir seorang auditor profesional.

Selain penelitian-penelitian di atas, pendekatan *Retrieval-Augmented Generation* (RAG) di ranah audit finansial juga dieksplorasi oleh Bijvoet (2025) yang mengembangkan sistem dengan penamaan serupa, yaitu AuditRAG, untuk mengevaluasi kepatuhan laporan tahunan terhadap regulasi Belanda [27]. Namun, terdapat perbedaan fundamental pada arsitektur dan fokus penyelesaian masalah. Sistem yang dikembangkan oleh Bijvoet (2025) menggunakan pendekatan *Agentic*

RAG (ARAG) yang sangat bergantung pada orkestrasi multi-agen (seperti agen ekspansi kueri dan agen evaluasi konteks) untuk memproses regulasi berbahasa Belanda [27]. Sebaliknya, penelitian ini menawarkan kebaruan (*novelty*) dengan mengusulkan arsitektur *Entity-Centric* Financial RAG yang dioptimasi secara spesifik untuk memitigasi halusinasi ekstraksi numerik. Alih-alih menggunakan sistem multi-agen yang kompleks, penelitian ini menanamkan logika penalaran terstruktur bernama *Audit-Chain-of-Thought* (Audit-CoT) secara langsung ke dalam model berukuran ringkas (Qwen-3 8B) melalui teknik *fine-tuning* QLoRA, guna mengekstrak metrik finansial pada laporan emiten di Bursa Efek Indonesia (BEI).

Berdasarkan pemetaan terhadap berbagai penelitian terdahulu pada Tabel 2.1, ditemukan adanya kesenjangan (*research* gap) yang signifikan dalam implementasi LLM untuk domain finansial. Mayoritas penelitian terdahulu berfokus pada model komersial berskala masif yang membutuhkan biaya komputasi tinggi, atau hanya mengandalkan pencarian hibrida linier tanpa mekanisme isolasi entitas yang ketat. Hal ini menyebabkan sistem masih rentan terhadap kontaminasi data antar-periode laporan keuangan serta halusinasi angka pada perangkat keras lokal. Guna memperjelas posisi akademis dan kebaruan (*novelty*) yang diusulkan, Tabel 2.2 merangkum perbandingan komparatif antara keterbatasan penelitian terdahulu dengan solusi yang diterapkan pada arsitektur AuditRAG.

Tabel 2.2 Gap dan kebaruan (*Novelty*) Penelitian

| Dimensi Komparasi | Fokus Keterbatasan Penelitian Terdahulu | Kebaruan (Novelty) pada Penelitian AuditRAG |
|---|---|---|
| Metode Pencarian Dokumen (*Retrieval*) | Menggunakan pendekatan RAG konvensional linier (hanya pencarian kata kunci atau vektor saja). Sering gagal membedakan konteks angka yang mirip jika kode saham dan tahun laporan tercampur di dalam korpus dokumen panjang. | Entity-Centric Filtering: Menerapkan komponen Query Intent Parser deterministik untuk mengekstrak kode emiten dan tahun fiskal secara langsung dari kueri, lalu menguncinya sebagai Hard Metadata Filters sebelum pencarian dijalankan untuk memitigasi kontaminasi lintas entitas. |

| | | |
|---|---|---|
| Strategi Penalaran Model (*Generation*) | Mengandalkan arsitektur *Chain-of-Thought* (CoT) umum atau verifikasi mandiri pasca-generasi (*Chain-of-Verification*). Model rentan terjebak dalam monolog naratif yang panjang, sehingga memicu halusinasi aritmatika pada angka laporan keuangan. | Algoritma Audit-Chain-of-Thought (Audit-CoT): Membangun *dataset* instruksi sintetik berstruktur QGA (*Question-Guidance-Answer*). Melatih model untuk mengeksplorasi jejak logika secara kaku dengan pola *Locate-Extract-Verify* dan memaksa luaran akhir dalam format JSON yang bersih. |
| Kebutuhan Sumber Daya Komputasi | Studi berskala industri (seperti BloombergGPT) sangat bergantung pada foundation model raksasa (50B+ parameter). Memerlukan infrastruktur server atau klaster GPU yang sangat mahal, sehingga sulit diadopsi secara lokal oleh analis keuangan. | Optimasi Low-Resource *Hardware*: Membuktikan bahwa model bahasa berukuran kecil (Qwen-3 8B) yang dioptimasi secara efisien melalui metode kuantisasi 4-bit NormalFloat (NF4) dan QLoRA dapat berjalan stabil pada VRAM 8,0 GB dengan pemangkasan waktu inferensi hingga 61,2%. |
| Stabilitas Validasi Keluaran | Evaluasi model umumnya hanya mengandalkan metrik akurasi kebahasaan tradisional (seperti F1-Score atau ROUGE), yang tidak sensitif terhadap kesalahan konversi skala satuan nominal keuangan (jutaan vs miliaran). | Metrik Evaluasi Deterministik Finansial: Menerapkan pengujian *Dual-run inference* dengan metrik Consistency Score (CS) untuk menguji stabilitas deterministik, serta metrik Matched Accuracy (MA) dengan penalti ambiguitas satuan via *LLM-as-a-Judge*. |

## 2.2 Teori yang berkaitan

Sub-bab ini menguraikan kerangka kerja teoretis dan teknis yang mendasari perancangan sistem *Entity-Centric Financial* RAG. Pembahasan mencakup teori dasar LLMs, mekanisme adaptasi hemat parameter, arsitektur pencarian informasi hibrida, serta strategi pemrosesan dokumen tabular yang kompleks.

### *2.2.1 Large Language Models dan Parameter-efficient fine-tuning*

Penerapan model generatif skala besar (*Large Language Models*) pada lingkungan dengan sumber daya perangkat keras yang terbatas menghadirkan tantangan komputasi yang signifikan. Model-model ini umumnya

memerlukan memori GPU (VRAM) yang sangat besar untuk memuat parameter bobot, belum termasuk kebutuhan memori tambahan untuk menyimpan status pengoptimal dan gradien selama proses pelatihan. Oleh karena itu, skenario ini menuntut metode optimasi khusus untuk mencapai keseimbangan yang presisi antara efisiensi memori dan performa model. Jika melakukan pelatihan ulang model besar dari awal (*training from scratch*) yang memakan biaya komputasi sangat mahal dan waktu yang lama, penelitian terbaru dalam literatur *Deep Learning* menyoroti efektivitas teknik efisiensi parameter. Teknik ini bekerja dengan cara membekukan sebagian besar parameter model pra-latih (*pre-trained weights*) dan hanya melatih sejumlah kecil parameter tambahan untuk mengadaptasi model tersebut terhadap tugas-tugas spesifik domain, seperti analisis keuangan [34]. Pendekatan ini tidak hanya mengurangi hambatan masuk untuk pengembangan AI tingkat lanjut tetapi juga mencegah fenomena *catastrophic forgetting*, di mana model kehilangan pengetahuan umumnya saat mempelajari data baru[35].

Tabel 2.3 berikut menyajikan rincian komprehensif mengenai model fondasi yang dipilih dalam penelitian ini, serta strategi *fine-tuning* dan komponen pendukung lainnya. Pemilihan komponen-komponen ini didasarkan pada strategi untuk menyeimbangkan efisiensi komputasi pada perangkat keras kelas konsumen dengan kebutuhan akan keandalan numerik yang tinggi dalam audit laporan keuangan.

Tabel 2.3 Ringkasan Model Fondasi dan Teknik Optimasi

| Konsep / Komponen | Deskripsi & Peran dalam Sistem | Referensi Utama |
|---|---|---|
| Model Fondasi | **Qwen-3 8B** dipilih sebagai generator dasar (*base generator*) utama dalam arsitektur ini. Berbeda dengan model sejenis seperti Llama-2 atau Mistral, Qwen-3 menunjukkan superioritas dalam *benchmark* penalaran matematika dan pemahaman konteks panjang. Kemampuan ini sangat krusial karena model bertugas memproses potongan tabel keuangan yang diambil (*retrieved chunks*) yang sering kali memiliki struktur kompleks—untuk kemudian diekstraksi menjadi jawaban yang berlandaskan data numerik yang presisi, serta mendukung input multibahasa (Indonesia-Inggris). | [16] |
| *Parameter-efficient* | **LoRA (*Low-rank adaptation*)** digunakan sebagai strategi adaptasi utama. Secara teknis, LoRA menyuntikkan matriks dekomposisi berperingkat rendah (*low-rank decomposition matrices*) ke dalam | [35] |

| | | |
|---|---|---|
| *fine-tuning* (PEFT) | lapisan attention model transformator, sementara bobot model pra-latih tetap dibekukan. Hal ini memfasilitasi adaptasi model terhadap domain keuangan tanpa perlu melatih ulang miliaran parameter secara penuh. Dampaknya adalah pengurangan drastis pada kebutuhan memori GPU dan penyimpanan, memungkinkan proses *fine-tuning* dilakukan dengan lebih efisien tanpa mengorbankan kualitas luaran model. | |
| Quantized *Fine-tuning* | **QLoRA (*Quantized Low-rank adaptation*)** diterapkan untuk memperluas efisiensi LoRA lebih jauh dengan memperkenalkan tipe data 4-bit NormalFloat (NF4). Teknik ini melakukan kuantisasi pada model dasar menjadi presisi 4-bit, namun tetap mempertahankan performa yang sebanding dengan *tuning* presisi penuh 16-bit (*full-precision*) melalui mekanisme *Double Quantization* dan *Paged Optimizers*. Metode ini adalah kunci yang memungkinkan *fine-tuning* model Qwen-3 8B secara spesifik pada *dataset Question Answering* (QA) keuangan Indonesia dapat dijalankan dalam batas kemampuan perangkat keras kelas konsumen (seperti GPU dengan VRAM terbatas). | [17] |
| Model *Embedding* | **E5-multilingual (v2-base)** berfungsi sebagai penyandi semantik yang mengodekan query pengguna dan potongan dokumen laporan keuangan ke dalam ruang vektor padat (*dense vector space*) yang sama. Model ini dipilih karena dilatih menggunakan teknik *contrastive learning* pada data multibahasa yang masif, sehingga sangat efektif dalam menangkap nuansa semantik lintas bahasa. Kemampuan ini memungkinkan sistem melakukan pencarian relevansi (*similarity search*) yang akurat baik pada konten keuangan berbahasa Indonesia maupun Inggris, mengatasi keterbatasan pencarian kata kunci tradisional. | [36] |
| Model Evaluasi | **Gemini 2.5 Flash** diintegrasikan sebagai pengawas eksternal (LLM-as-a-Judge) untuk memverifikasi jawaban Qwen-3. Dengan context window besar dan logika kuat, model ini membandingkan prediksi dengan *ground truth* memakai rubrik ketat, serta sensitif terhadap ambiguitas, kesalahan satuan (mis. "jutaan" vs "miliaran"), dan halusinasi angka. | [37] |

Penerapan model generatif skala besar pada perangkat keras dengan sumber daya terbatas menuntut metode optimasi khusus. Alih-alih melatih ulang seluruh parameter model (*full fine-tuning*) yang memakan biaya komputasi masif, penelitian ini menerapkan teknik *Parameter-efficient fine-tuning* (PEFT).

Metode PEFT utama yang digunakan dalam penelitian ini adalah LoRA. Sebagaimana dijelaskan oleh Hu et al. (2022), LoRA bekerja dengan membekukan bobot model pra-latih (pre-trained weights) $W_0 \in \mathbb{R}^{d \times d}$ dan

menyuntikkan pasangan matriks dekomposisi rank-rendah yang dapat dilatih ke dalam lapisan *Transformer* [20].

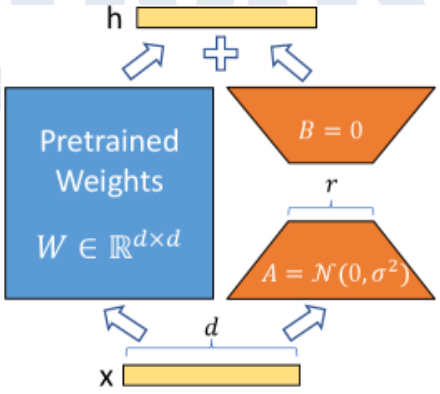
Jika x adalah input dan h adalah output, maka pembaruan bobot pada LoRA adalah berikut:

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

Rumus 2.1 LORA [20]

- $h$ = vektor representasi output tersembunyi (hidden state output).
- $W_0$ = matriks bobot dasar model pra-latih yang dibekukan (frozen pre-trained weights).
- x = vektor representasi *input* (input vector).
- ΔW = akumulasi perubahan matriks bobot hasil adaptasi.
- $B$ = matriks dekomposisi adaptor pertama dengan dimensi d×r.
- $A$ = matriks dekomposisi adaptor kedua dengan dimensi r×d.

Pada Rumus 2.1 di mana matriks dengan dimensi *rank* yang jauh lebih kecil daripada dimensi model asli $(r \ll d)$. Matriks diinisialisasi dengan distribusi Gaussian acak, sedangkan matriks diinisialisasi dengan nol, sehingga pada awal pelatihan $\Delta W = 0$. Arsitektur ini memungkinkan efisiensi memori GPU hingga 3 kali lipat tanpa mengorbankan kemampuan model dalam menangkap pola bahasa baru.



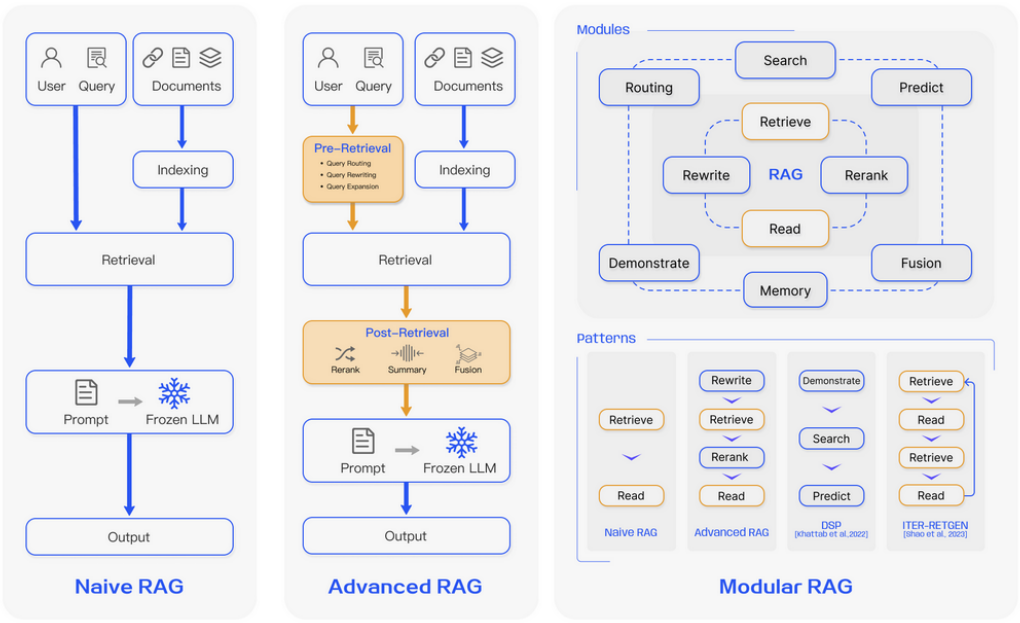

Gambar 2.1 LoRA *Architecture* [20]

Pada Gambar 2.1 yang merupakan *Architecture* dari LoRA, menunjukkan bahwa hanya matriks A dan B yang dilatih, sedangkan bobot model awal (kotak biru) dibekukan untuk menghemat memori [20].

### 2.2.2 Retrieval-Augmented Generation dan *Hybrid retrieval*

Dalam mengurangi risiko halusinasi yang terkait dengan memori parametrik pada LLM, sistem keuangan modern mengadopsi arsitektur RAG. Arsitektur ini membantu dalam mengambil bukti yang tepat dan memastikan bahwa luaran yang dihasilkan tetap akurat, terutama untuk pertanyaan yang menuntut angka pasti. Strategi hibrida diterapkan dengan menggabungkan pencocokan istilah yang tepat (*exact term matching*) dengan pemahaman semantik guna meningkatkan recall, sebagaimana dirinci dalam Tabel 2.4.

Menurut Gao et al. (2024) [6] mendefinisikan pendekatan ini sebagai Modular RAG, di mana sistem tidak hanya melakukan pencarian satu arah, tetapi melibatkan modul tambahan seperti *rewriting, reranking*, dan *filtering* untuk meningkatkan presisi dokumen yang diambil [6].




Gambar 2.2 Arsitektur Modular RAG [6]

Sistem ini menggunakan mekanisme *Hybrid retrieval* yang menggabungkan dua metode perhitungan skor:

a. *Sparse retrieval* (BM25): Digunakan untuk menangkap kata kunci eksak (seperti "Aset Lancar" atau "2023"). Skor BM25 dihitung menggunakan rumus probabilitas berikut:

$$\text{Score}(D,Q) = \sum_{i=1}^{n} \text{IDF}\,(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{\text{avgdl}}\right)}$$

Rumus 2.2 BM25 [13]

Di mana:

- *Score(D, Q)*: skor relevansi leksikal akhir antara dokumen *D* dan kueri *Q*.
- IDF($q_i$): nilai *Inverse Document Frequency* dari kata kunci ke-*i* ($q_i$).
- *f*($q_i$, *D*): frekuensi kemunculan kata kunci $q_i$ di dalam dokumen *D*.
- $k_1$: parameter konstanta untuk mengontrol batas saturasi frekuensi kata kunci.
- *b*: parameter konstanta untuk mengontrol tingkat penalti panjang dokumen.
- |*D*| = panjang dokumen *D* berdasarkan hitungan jumlah kata.
- avgdl = rata-rata panjang dokumen dari keseluruhan korpus data.

b. *Dense retrieval* (*Vector*): Menggunakan model *embedding* E5-multilingual untuk menangkap makna semantik. Kemiripan antara vektor query (A) dan dokumen (B) diukur menggunakan Cosine Similarity:

$$\text{similarity}(A,B) = \frac{A\cdot B}{\|A\|B\|}$$

Rumus 2.3 *Vector* [14]

- similarity (*A,B*) = nilai tingkat kemiripan kosinus antara vektor *A* dan vektor *B*.
- *A* . *B* = perkalian titik (*dot product*) antara vektor kueri *A* dan vektor dokumen *B*.
- ||*A*|| = magnitudo atau panjang spasial dari vektor kueri *A*.
- ||*B*|| = magnitudo atau panjang spasial dari vektor dokumen *B*.

Rincian komponen teknis yang digunakan dalam kerangka kerja RAG pada penelitian ini:

Tabel 2.4 Komponen Kerangka Kerja RAG

| Konsep / Komponen | Deskripsi & Peran dalam Sistem | Referensi Utama |
|---|---|---|
| *Core RAG* | Menjelaskan pendekatan arsitektur yang mengintegrasikan komponen pencarian (retrieval) dengan model generatif. Mekanisme ini bekerja dengan menyuntikkan bagian teks (*passages*) relevan yang diambil dari basis data eksternal ke dalam *prompt* pengguna, memberikan konteks faktual tambahan untuk meminimalisir halusinasi model. | [6] |
| *Entity-Centric Filtering* | Sebuah *Query Intent Parser* deterministik mendeteksi entitas spesifik seperti kode saham (*ticker*) dan tahun fiskal dari pertanyaan pengguna, lalu menerapkannya sebagai Hard *Filters* pada penyimpanan vektor. Mekanisme isolasi ini secara ketat mencegah kontaminasi lintas data, memastikan misalnya data tahun 2022 tidak terambil untuk pertanyaan mengenai tahun 2023. | [6] |
| *Sparse retrieval* | BM25 berfungsi sebagai metode pencarian leksikal dasar (*baseline*), yang menghitung skor relevansi probabilistik melalui pencocokan kata kunci eksak. Komponen ini krusial dalam domain keuangan untuk memastikan istilah teknis spesifik, seperti nama akun akuntansi atau angka unik, dapat ditemukan secara presisi tanpa distorsi semantik. | [38] |
| *Dense retrieval* | Model E5-multilingual menghasilkan representasi vektor padat (*dense vectors*) untuk memfasilitasi pencarian semantik. Metode ini mampu menangkap kesamaan konseptual dan sinonim antar bahasa (Indonesia-Inggris), memungkinkan pengambilan konteks yang relevan bahkan ketika frasa pertanyaan pengguna tidak memiliki kecocokan kata kunci secara harfiah. | [36] |
| *RRF* | Algoritma RRF menggabungkan hasil pencarian sparse dan dense berdasarkan posisi peringkat (rank) dokumen, bukan skor mentahnya. Pendekatan ini menyeimbangkan kedua sinyal secara optimal tanpa terpengaruh oleh perbedaan skala skor metrik bawaan, memastikan konteks yang relevan diprioritaskan. | [39] |
| *Reranking* | Sebuah *Cross-encoder* (BGE) digunakan untuk memberi skor ulang (*rescoring*) pada kandidat dokumen teratas. Model ini memproses kueri dan dokumen secara bersamaan menggunakan lapisan interaksi mendalam (*deep interaction layers*), menghasilkan penilaian relevansi yang jauh lebih akurat dibandingkan kesamaan kosinus sederhana untuk menyaring noise sebelum masuk ke generator. | [40] |

### *2.2.3* Pengindeksan Vektor dan Pencarian *Approximate nearest neighbor*

Penerapan struktur pengindeksan yang efisien sangat diperlukan untuk mendukung pengambilan kembali (*retrieval*) secara real-time pada koleksi

potongan dokumen (*document chunks*) yang besar. Algoritma *Approximate nearest neighbor* (ANN) menyediakan alternatif praktis terhadap pencarian eksak yang memakan biaya komputasi tinggi, dengan memungkinkan pencarian kesamaan (*similarity search*) yang skalabel dan berlatensi rendah. Infrastruktur ini mempertahankan kinerja yang efisien sembari tetap mendukung kompleksitas dimensi tinggi dari dense *embedding*s (lihat Tabel 2.5).

Tabel 2.5 Teknologi untuk Pengindeksan dan Pencarian Vektor yang Efisien

| Teknologi | Peran dalam Sistem | Referensi Utama |
|---|---|---|
| FAISS | Pustaka *open-source* Facebook AI *Similarity Search* menyediakan standar industri untuk pencarian kesamaan vektor dan pengelompokan (*clustering*) vektor padat yang terakselerasi oleh GPU. Teknologi ini diimplementasikan untuk mengindeks ribuan representasi *embedding* E5 dari potongan dokumen PDF, memfasilitasi pengambilan data (*retrieval*) dengan latensi setingkat milidetik tanpa mengorbankan akurasi pencarian. | [14] |
| HNSW | Struktur data *Hierarchical Navigable Small World* berfungsi sebagai referensi konseptual utama untuk algoritma pencarian *Approximate nearest neighbor* (ANN) berbasis grafik modern. Mekanisme navigasi "dunia kecil" yang bertingkat pada algoritma ini menjadi dasar perancangan indeks vektor yang sangat efisien, menawarkan keseimbangan optimal antara kecepatan kueri (*query speed*) dan akurasi pemanggilan (*recall*) pada *dataset* berdimensi tinggi. | [41] |

### 2.2.4 *Chunking* dan Pengayaan Semantik untuk Berkas Keuangan

Struktur berkas keuangan menciptakan tantangan tersendiri, mengingat tabel multi-kolom sering kali rusak selama proses ekstraksi teks konvensional. Pengambilan kembali (*retrieval*) yang akurat sangat bergantung pada pelestarian hubungan posisi antara label baris dan nilai numerik yang bersangkutan. Sistem ini menggunakan alur kerja (*pipeline*) pra-pemrosesan yang berpusat pada tabel (*table-centric*) yang mengintegrasikan OCR tingkat lanjut dan pengayaan semantik untuk memastikan integritas data, sebagaimana diuraikan dalam Tabel 2.6.

Tabel 2.6 Strategi Pra-pemrosesan dan Pengayaan Semantik untuk Dokumen Keuangan

| Strategi | Metodologi | Referensi Utama |
|---|---|---|
| Filosofi Pra-pemrosesan | Mengadopsi pendekatan FinSage dalam mengonversi konten multi-modal (tabel/teks) menjadi teks linear yang ramah terhadap proses retrieval sembari tetap mempertahankan struktur tabel secara ketat. | [8] |
| Mesin OCR & Tata Letak | Alur kerja memanfaatkan Marker (didukung oleh Surya OCR) untuk mengurai laporan PDF. Mesin ini dipilih karena kemampuan deteksi tingkat baris yang unggul dan kemampuannya mempertahankan struktur tabel dalam lingkungan sumber daya terbatas (*low-resource*), sebagaimana divalidasi oleh pengujian (*benchmarking*) terbaru. | [42] |
| Pengayaan Metadata | Metadata tingkat berkas (simbol *ticker*, tahun fiskal) disuntikkan ke dalam metadata potongan dokumen (*chunk*), guna memfasilitasi penyaringan yang presisi. Studi terbaru mengonfirmasi bahwa injeksi metadata semacam ini secara signifikan meningkatkan presisi retrieval dalam sistem RAG. | [43] |
| Pengayaan Semantik | Kesadaran Bagian (*Section-Awareness*): Mempropagasi tajuk bagian (*headers*) ke dalam teks *chunk* sehingga setiap fragmen dokumen memiliki konteks yang mandiri (*Contextual Chunking*). | [44] |

### 2.2.5 Sistem Tanya Jawab Finansial dan Tolok Ukur Domain

Mengevaluasi sistem finansial memerlukan tolok ukur (*benchmarks*) yang melampaui sekadar pemeriksaan fakta dasar guna mengukur kemampuan penalaran numerik. *Dataset* konvensional sering kali kurang memadai dalam memodelkan tugas audit yang melibatkan perhitungan numerik yang bersumber dari tabel yang diambil (*retrieved tables*).

Tabel 2.7 merangkum tolok ukur penelitian utama yang menjadi panduan dalam pengembangan protokol evaluasi sistem ini.

Tabel 2.7 Benchmarks dan *Dataset* untuk Sistem Tanya Jawab Finansial

| Benchmarks/ *Dataset* | Relevansi dengan Studi | Referensi Utama |
|---|---|---|
| FinQA | Mendemonstrasikan perlunya penalaran numerik (penjumlahan, pengurangan, perhitungan rasio) atas data keuangan, yang mendorong kebutuhan akan generator yang memiliki kapabilitas matematika (*math-capable generator*). | [45] |
| FinanceBench | Digunakan sebagai garis dasar (*baseline*) untuk menyoroti keterbatasan model generik dalam menjaga akurasi faktual saat membaca berkas perusahaan yang kompleks. | [5] |

| | | |
|---|---|---|
| FinGPT | Mengilustrasikan efektivitas adaptasi LLM *open-source* dengan instruksi keuangan yang spesifik pada domain tersebut (*domain-specific instructions*). | [21] |

### 2.2.6 Teori Akuntansi Fondasional dan Signifikansi Metrik Finansial

Pemrosesan, ekstraksi, dan penalaran atas informasi kuantitatif di dalam domain finansial menuntut tingkat akurasi yang absolut karena dampaknya yang masif terhadap keputusan strategis korporasi. Model kecerdasan artifisial umum sangat rentan terhadap halusinasi numerik di domain ini, sehingga diperlukan pemahaman mendalam mengenai relevansi metrik keuangan utama yang menjadi target ekstraksi sistem [31]. Dalam penelitian ini, pengujian keandalan arsitektur AuditRAG difokuskan pada tiga metrik finansial fundamental akuntansi:

1. *Net Income* (Laba Bersih): Merupakan indikator paling krusial untuk mengukur profitabilitas bersih dan kinerja operasional menyeluruh suatu emiten setelah dikurangi seluruh beban dan pajak. Secara teoretis, komponen laba bersih memiliki relevansi nilai (*value relevance*) tertinggi bagi investor publik karena secara langsung memengaruhi ekspektasi *return* saham, pergerakan harga instrumen pasar modal, serta menjadi sinyal utama tingkat kepercayaan para *stakeholder* [28], [29].
2. *Total Asset* (Total Aset): Merepresentasikan akumulasi seluruh sumber daya ekonomis yang dikuasai oleh entitas hukum perusahaan, baik yang bersifat aset lancar maupun aset tidak lancar, sebagai akibat dari transaksi masa lalu. Penilaian detail dan pemisahan komponen aset yang dilaporkan secara terdisagregasi di dalam lembar Informasi Umum sangat informatif bagi analis keuangan dalam memprediksi arah pertumbuhan kapasitas operasi bisnis serta keberlanjutan pendapatan masa depan korporasi [30], [31].
3. *Revenue* (Pendapatan): Menunjukkan aliran masuk bruto dari manfaat ekonomi yang timbul dari aktivitas normal entitas selama

suatu periode (*top-line growth*). Di dalam praktik audit konvensional maupun otomatis, metrik pendapatan merupakan area yang paling krusial untuk diperiksa secara deterministik karena sering kali menjadi target manipulasi pelaporan keuangan, sehingga sistem RAG menuntut adanya kapabilitas pengenalan skala satuan nominal kaku untuk mencegah kesalahan pembacaan [32], [33].

Penyatuan dan pemahaman mendalam atas ketiga metrik finansial fundamental ini di dalam lapisan landasan teori berfungsi sebagai parameter bisnis kaku (*business knowledge constraints*) yang menjustifikasi keandalan jalur penalaran algoritma Audit-CoT dalam menjaga hubungan logis dan integritas aritmatika antar-akun laporan keuangan emiten di Indonesia [31].

### 2.2.7 Strategi Penalaran dan Audit-Chain-of-Thought

Pendekatan penalaran standar cenderung menyebabkan halusinasi aritmatika dalam konteks keuangan. Untuk meningkatkan presisi, sistem ini menggunakan metode *prompting* khusus yang mengharuskan adanya verifikasi eksplisit terhadap nilai numerik sebelum perhitungan dilakukan. Metode *Audit-Chain-of-Thought* dirancang menyerupai proses penalaran seorang auditor keuangan, dengan memanfaatkan strategi implementasi yang ditunjukkan pada Tabel 2.8.

Tabel 2.8 Strategi Penalaran dan Kerangka Kerja Audit-Chain-of-Thought

| Strategi | Detail Implementasi | Referensi Utama |
|---|---|---|
| *Self-Consistency CoT* | Meningkatkan ketahanan (*robustness*) dengan mengambil sampel dari beberapa jalur penalaran dan mengagregasi hasilnya, sehingga mengurangi kemungkinan terjadinya kesalahan perhitungan tunggal. | [24] |
| *Audit-Chain-of-Thought (Audit-CoT)* | Sebuah kerangka kerja *prompting* kustom yang menginstruksikan model untuk menemukan baris tabel yang tepat, menyalin nilai secara verbatim (persis sesuai aslinya), dan memverifikasinya sebelum menjawab. Metode ini mengadaptasi paradigma "*Chain-of-Verification*" ke dalam audit keuangan. | [23] |
| *LLM-as-a-Judge* | Memanfaatkan Gemini 2.5 Flash untuk menilai respons berdasarkan kesetaraan numerik yang ketat dengan kebenaran | [46] |

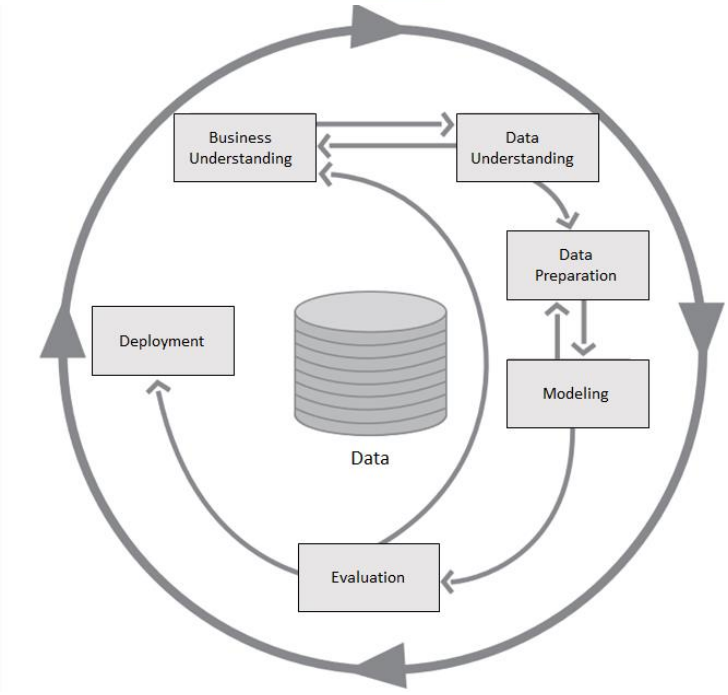
| | | |
|---|---|---|
| | dasar (*ground truth*), yang mengotomatisasi proses evaluasi akurasi. | |
| *Chain of Guidance* | Menjadi landasan konseptual dalam merancang format *dataset* instruksi sintetik menjadi struktur QGA (*Question-Guidance-Answer*). Menyisipkan panduan langkah demi langkah untuk memastikan model memverifikasi logika ekstraksi tabelnya sebelum menyimpulkan jawaban akhir. | [12] |

## 2.3 *Framework*/Algoritma yang digunakan

Penelitian ini mengintegrasikan berbagai kerangka kerja (*framework*) metodologis dan algoritma komputasi untuk membangun sistem *Entity-Centric Financial* RAG. Kerangka kerja ini mencakup metodologi pengembangan sistem secara keseluruhan, arsitektur *retrieval* (temu kembali), hingga algoritma *fine-tuning* model bahasa.

### 2.3.1 CRISP-DM

Sebagai kerangka kerja metodologi utama, penelitian ini mengadopsi standar CRISP-DM. Meskipun konsep ini sudah mapan, penerapannya dalam proyek *Generative* AI modern tetap relevan karena strukturnya yang iteratif, memungkinkan perbaikan berkelanjutan pada kualitas data dan model [47].




Gambar 2.3 Siklus Hidup CRISP-DM yang terdiri dari enam tahapan iteratif. [47]

a. *Business Understanding* (Pemahaman Bisnis): Tahap ini mendefinisikan masalah "halusinasi aritmatika" pada LLM dan menetapkan tujuan audit otomatis. Fokus utamanya adalah memetakan kebutuhan auditor terhadap kemampuan sistem untuk memverifikasi angka secara presisi [47].
b. *Data Understanding* (Pemahaman Data): Melibatkan pengumpulan Laporan Tahunan emiten BEI (2021–2024) dan analisis struktur tabel yang kompleks. Tantangan seperti format bilingual dan tabel lintas-halaman diidentifikasi di sini [47].
c. *Data Preparation* (Persiapan Data): Tahap rekayasa paling krusial yang mencakup parsing PDF menggunakan OCR, pemotongan teks (*chunking*) dengan ukuran tetap sebesar 2048 token per segmen, dan pengayaan metadata (*Ticker*/Tahun) untuk memastikan data siap dilatih [47].
d. *Modeling* (Pemodelan): Penerapan algoritma *Fine-tuning* QLoRA pada model Qwen-3 8B serta konfigurasi sistem RAG hibrida [47].
e. *Evaluation* (Evaluasi): Mengukur kinerja model menggunakan metrik *Matched Accuracy (MA), Audit Reasoning Quality* (ARQ) dan *Consistency Score* (CS) pada data uji terisolasi (2023–2024) [47].
f. *Deployment* (Penyebaran): Integrasi model ke dalam *inference pipeline* yang dapat diakses melalui antarmuka pengguna untuk kueri *real-time* [47].

### 2.3.2 Entity-Centric RAG (Modular RAG)

Berbeda dengan RAG tradisional yang bersifat linear, penelitian ini menerapkan arsitektur Modular RAG. Algoritma ini dirancang untuk menangani ambiguitas entitas dalam dokumen keuangan (misalnya membedakan "Laba 2022" dan "Laba 2023").

Berdasarkan survei komprehensif tahun 2024, pendekatan modular ini terbukti meningkatkan adaptabilitas sistem melalui penambahan modul fungsional khusus [6]. Komponen algoritmanya meliputi:

a. *Deterministic Entity Routing*: Algoritma hard-filtering yang membatasi ruang pencarian vektor hanya pada simbol saham dan Tahun Fiskal yang relevan, mencegah kontaminasi silang data antar perusahaan [48].
b. Hybrid Score Fusion: Algoritma penggabungan peringkat secara heuristik menggunakan metode RRF yang menyeimbangkan hasil dari pencarian vektor dan kata kunci secara proporsional dan kebal terhadap distorsi skala [13].
c. *Audit-Chain-of-Thought* (Audit-CoT): Algoritma penalaran yang memaksa model untuk memvalidasi angka yang diambil sebelum melakukan kalkulasi, mengadaptasi teknik verifikasi mandiri terbaru [23].

### 2.3.3 QLoRA

Pelatihan model bahasa besar pada perangkat keras konsumen (GPU tunggal), penelitian ini menggunakan algoritma QLoRA. Algoritma ini merupakan pengembangan dari LoRA yang memperkenalkan tipe data 4-bit NormalFloat (NF4) dan *Double Quantization*.

QLoRA mampu mengurangi kebutuhan memori GPU secara drastis sambil mempertahankan performa yang setara dengan pelatihan presisi penuh 16-bit [17]. Ini memungkinkan model Qwen-3 8B diadaptasi secara efisien menggunakan data instruksi keuangan Indonesia.

## 2.4 Tools/software yang digunakan

Bagian ini merinci perangkat lunak, pustaka, dan model yang digunakan sebagai instrumen penelitian. Pemilihan alat didasarkan pada kompatibilitas dengan ekosistem *open-source* terkini dan performa pada benchmark akademis.

### 2.4.1 Model Fondasi (Foundation Models)

a. Qwen-3 8B (*Base Generator*): Model bahasa ini dipilih sebagai fondasi utama karena keunggulan performanya dalam matematika

dan pemahaman multibahasa dibandingkan model sekelasnya seperti Llama. Laporan teknis terbaru menunjukkan superioritas arsitektur Qwen dalam menangani reasoning tasks [16].

b. Gemini 2.5 Flash (*Synthetic Data & Evaluator*): Model ini digunakan untuk dua fungsi: menghasilkan data latih sintetik berkualitas tinggi dan bertindak sebagai *LLM-as-a-Judge* untuk menilai akurasi jawaban. Kemampuannya dalam memproses konteks panjang menjadikannya ideal untuk audit dokumen [37].

### 2.4.2 Komponen Retrieval dan *Embedding*

a. intfloat/multilingual-e5-base (*Embedding Model*): Model ini digunakan untuk mengubah teks laporan keuangan menjadi vektor. E5 dipilih karena menggunakan teknik *weakly-supervised contrastive pre-training* yang menghasilkan representasi semantik berkualitas tinggi untuk teks multibahasa [36].

b. FAISS (*Vector Indexing*): Pustaka Facebook AI Similarity Search digunakan untuk manajemen indeks vektor. Versi terbarunya mendukung pencarian tetangga terdekat (*Nearest Neighbor*) yang sangat cepat pada GPU, sangat krusial untuk sistem RAG skala besar [14].

c. BAAI/bge-*reranker*-v2-m3 (*Cross-encoder*): Digunakan pada tahap *reranking* untuk mengurutkan ulang dokumen yang diambil. Model BGE terbukti menempati peringkat teratas pada benchmark C-MTEB untuk kemampuan pemahaman relevansi dokumen [49].

### 2.4.3 Pemrosesan Dokumen

a. Marker (PDF to *Markdown Engine*): Alat ini digunakan untuk mengonversi PDF menjadi format Markdown. Marker, yang ditenagai oleh model Surya OCR, dipilih karena akurasinya yang tinggi dalam mendeteksi tata letak tabel (*layout analysis*) dibandingkan alat tradisional seperti PyPDF [42].