

3. ANALISIS DAN DESAIN SISTEM

Bab ini akan membahas tentang analisis dan desain sistem yang berkaitan dengan pengembangan *chatbot* yang diharapkan dapat membantu mahasiswa Informatika Universitas Kristen Petra yang mengambil mata kuliah *Artificial Intelligence and Machine Learning*.

3.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk memastikan bahwa konten yang dikembangkan benar-benar sejalan dengan arah pembelajaran mata kuliah *Artificial Intelligence and Machine Learning* (AIML), baik dari segi kedalaman materi maupun keterpaduannya dengan kurikulum yang berlaku. Untuk itu, analisis ini merujuk pada struktur topik dan subtopik yang akan menjadi dasar sistem, yang disusun berdasarkan pokok-pokok pembahasan dalam *Buku Pedoman Program Studi Informatika Universitas Kristen Petra tahun 2024* (Program Studi Informatika, 2024). Pokok-pokok pembahasan tersebut dipilih karena mewakili inti dari silabus AIML dan dianggap paling tepat untuk dijadikan sebagai modul dalam sistem *chatbot*. Dengan mengacu pada struktur pembelajaran yang telah ditetapkan ini, *chatbot* diharapkan mampu menyajikan informasi dan jawaban yang selaras dengan materi akademik yang diajarkan dalam perkuliahan. Pokok-pokok pembahasan tersebut kemudian dijadikan sebagai modul-modul inti yang membentuk struktur dasar dalam sistem *chatbot* yang dapat dilihat pada Tabel 3.1.

Tabel 3.1

Modul-modul yang akan digunakan berdasarkan pokok pembahasan mata kuliah *Artificial Intelligence and Machine Learning*

Nama Modul	Materi Pembelajaran
<i>Introduction to AI</i>	Apa itu AI?, Mengapa AI penting?, Aplikasi AI, Konsep agen dalam AI
<i>Heuristic Search</i>	<i>Backtracking, Hill Climbing, Greedy Best-First Search, A*, Simulated Annealing</i>

<i>Genetic Algorithm dan Nature Inspired Algorithms</i>	<i>Genetic Algorithm, Evolutionary Strategy, Ant Colony Optimization (ACO), PSO</i>
<i>Rule Based AI dan Fuzzy Logic</i>	<i>Rule-Based System, Fuzzy Logic</i>
<i>Machine Learning</i>	<i>Pengantar Machine Learning</i>
<i>Supervised, Unsupervised and Reinforcement Learning</i>	<i>K-NN classifier & regressor, Gradient Descent, K-Means, DBScan, Q-Learning</i>
<i>Ensemble Learning</i>	<i>Bagging, Boosting, Random Forest</i>
<i>Deep Learning dan Generative AI</i>	<i>Neural Network, Perceptron, MLP, SOM, Back propagation, CNN, RNN, LSTM, Pengantar AI Generatif</i>

Selain berdasarkan kebutuhan internal kurikulum program studi, topik dan subtopik ini juga disusun dengan mengacu pada kurikulum internasional *ACM/IEEE Computer Science 2023* (Kumar et al., 2024), yang merepresentasikan konsensus global dalam pendidikan Ilmu Komputer. Dalam dokumen *ACM/IEEE*, pembelajaran mengenai *Artificial Intelligence* dan *Machine Learning* mencakup area-area seperti *Search and Optimization, Evolutionary Computation, Supervised and Unsupervised Learning, Neural Networks and Deep Learning*, serta *Knowledge Representation and Reasoning*.

Setiap topik utama yang diajarkan dalam mata kuliah ini memiliki keterkaitan langsung dengan bagian-bagian dari kurikulum *ACM/IEEE*. Misalnya, pada topik *Pengenalan AI*, subtopik seperti pengertian AI, pentingnya AI, dan agen dalam AI sesuai dengan area *Foundations of Artificial Intelligence* dan *Intelligent Agents*, yang merupakan materi dasar dalam memahami sistem kecerdasan buatan. Topik *Heuristic Search*, dengan subtopik seperti *Greedy Best-First Search* dan *A**, termasuk dalam *Search Algorithms* dan *Search and Constraint Satisfaction*, yang merupakan bagian penting dalam pendekatan pemecahan masalah. Selanjutnya, *Algoritma Genetika dan Nature Inspired Algorithms* seperti *Genetic Algorithm, Evolutionary Strategy*, serta *PSO* dan *ACO*, sejajar dengan bagian *Metaheuristics and Evolutionary Computation* dalam kurikulum *ACM/IEEE*, yang menekankan metode optimasi berbasis populasi dan inspirasi alam. Topik *Rule Based AI* dan *Logika Fuzzy* berkaitan dengan area *Knowledge Representation and*

Reasoning serta *Reasoning Under Uncertainty*, yang berfokus pada teknik representasi pengetahuan dan penalaran dalam kondisi tidak pasti. Pada topik *Machine Learning*, subtopik seperti K-NN classifier dan regressor termasuk dalam *Machine Learning Foundations*, yang mencakup teknik dasar *supervised learning*. Topik *Supervised, Unsupervised, and Reinforcement Learning* mencakup *gradient descent, clustering* (K-Means, DBScan), dan Q-Learning, yang seluruhnya sesuai dengan bagian *Supervised and Unsupervised Models, Unsupervised Learning, dan Sequential Decision Making*. Kemudian, *Ensemble Learning* yang mencakup teknik seperti *bagging* dan *boosting* selaras dengan *Model Evaluation and Improvement*, yang membahas penggabungan model untuk meningkatkan akurasi dan ketahanan sistem. Sementara itu, *Deep Learning dan AI Generatif*, yang memuat *neural networks* (Perceptron, MLP, SOM), CNN, RNN, LSTM, dan pengantar *AI generatif*, sejalan dengan *Neural and Deep Learning Models* dan *Generative Models*, yang menjadi bagian penting dalam tren terbaru pengembangan AI.

Dengan cakupan materi yang telah terstruktur, *chatbot* diharapkan mampu menyajikan jawaban yang relevan dan sesuai dengan topik yang diajarkan dalam perkuliahan, sehingga mahasiswa dapat memperoleh pemahaman yang lebih sistematis dan selaras dengan kurikulum yang berlaku. Pembelajaran dengan *chatbot* ini juga mendukung kemandirian mahasiswa karena memungkinkan mereka untuk mengeksplorasi materi secara bebas dan mandiri tanpa harus menunggu sesi kelas atau arahan langsung dari dosen. Selain itu, kemampuan *chatbot* untuk merespons berbagai pertanyaan dalam cakupan topik yang telah ditentukan juga mendorong mahasiswa untuk berpikir kreatif dan aktif dalam mengajukan pertanyaan-pertanyaan lanjutan yang dapat memperluas pemahaman. Pembelajaran melalui *chatbot* ini juga dapat disebut efektif karena seluruh materi yang digunakan telah dikurasi dan dipilih berdasarkan kesesuaiannya dengan silabus perkuliahan, meskipun sumbernya berasal dari platform terbuka seperti *Medium, Wikipedia, dan arXiv*. Sistem *retrieval* yang diterapkan dalam *chatbot* membatasi pencarian hanya pada dokumen-dokumen yang relevan dengan modul yang sedang dipelajari, sehingga jawaban yang diberikan tetap terfokus dan kontekstual terhadap materi akademik yang sedang diajarkan. Sementara itu, dari segi efisiensi, mahasiswa tidak perlu membuka banyak referensi yang tersebar di berbagai sumber, karena seluruh informasi penting yang berkaitan dengan materi pembelajaran telah dihimpun dalam satu platform yang mudah diakses dan dapat digunakan kapan saja. Dengan cara ini, mahasiswa dapat memanfaatkan waktu mereka secara lebih optimal untuk mengulang atau memperdalam pemahaman terhadap materi tanpa harus bergantung pada momen atau sesi pembelajaran

tertentu di kelas. Kemampuan *chatbot* dalam memberikan saran pertanyaan tambahan juga menjadi aspek penting dalam mendorong eksplorasi materi lebih lanjut, sehingga mahasiswa dapat menggali konsep secara lebih mendalam. Dengan demikian, keberadaan *chatbot* ini diharapkan dapat meningkatkan fleksibilitas dan aksesibilitas dalam proses pembelajaran, memungkinkan mahasiswa untuk memahami materi secara mandiri tanpa bergantung sepenuhnya pada jadwal dosen. Selain meningkatkan aksesibilitas pembelajaran, *chatbot* ini juga diharapkan dapat meningkatkan keterlibatan mahasiswa dalam proses belajar dengan menyediakan pengalaman interaktif yang disesuaikan dengan kebutuhan individu, sehingga mereka dapat lebih aktif dalam mengeksplorasi materi yang diajarkan.

3.2 Analisis Dataset

Dalam pengembangan *chatbot* berbasis *Retrieval-Augmented Generation* (RAG) untuk mata kuliah *Artificial Intelligence and Machine Learning* (AIML), *dataset* yang digunakan harus dirancang agar sesuai dengan kebutuhan pembelajaran mahasiswa. *Dataset* ini disusun berdasarkan silabus pembelajaran AIML, yang mencakup topik-topik utama yang diajarkan dalam perkuliahan. Silabus ini menjadi acuan dalam menentukan cakupan materi yang digunakan dalam *chatbot* agar tetap relevan dengan kurikulum yang berlaku. Namun, *dataset* yang digunakan dalam *chatbot* ini tidak berasal langsung dari materi perkuliahan seperti PPT atau catatan dosen, melainkan diambil dari sumber referensi yang menjadi dasar dalam pengajaran mata kuliah tersebut. Sumber utama *dataset* terdiri dari buku teks yang menjadi referensi dalam pengajaran perkuliahan, jurnal ilmiah, serta artikel dari *Wikipedia*, *arXiv*, dan *Medium*. Dengan menggunakan referensi akademik yang lebih luas, *chatbot* dapat memberikan jawaban yang lebih mendalam dan berbasis literatur yang kredibel. *Dataset* ini termasuk dalam kategori *unstructured data*, karena terdiri dari dokumen teks dalam format PDF. Format PDF dipilih karena merupakan format standar yang umum digunakan dalam publikasi akademik dan artikel ilmiah. Selain itu, PDF memungkinkan pelestarian struktur asli dokumen, sehingga memudahkan proses ekstraksi teks secara konsisten untuk keperluan *preprocessing* dan sistem *retrieval* dalam *chatbot*. Secara keseluruhan, *dataset* yang digunakan berjumlah 1.246 dokumen, sehingga diperlukan proses pemrosesan lanjutan agar dapat digunakan secara efektif dalam sistem *chatbot*.

Heuristic (computer science)

In mathematical optimization and computer science, **heuristic** (from Greek εὕρισκω "I find, discover"^[1]) is a technique designed for problem solving more quickly when classic methods are too slow for finding an exact or approximate solution, or when classic methods fail to find any exact solution in a search space. This is achieved by trading optimality, completeness, accuracy, or precision for speed. In a way, it can be considered a shortcut.

A **heuristic function**, also simply called a **heuristic**, is a function that ranks alternatives in search algorithms at each branching step based on available information to decide which branch to follow. For example, it may approximate the exact solution.^[2]

Definition and motivation

The objective of a heuristic is to produce a solution in a reasonable time frame that is good enough for solving the problem at hand. This solution may not be the best of all the solutions to this problem, or it may simply approximate the exact solution. But it is still valuable because finding it does not require a prohibitively long time.

Heuristics may produce results by themselves, or they may be used in conjunction with optimization algorithms to improve their efficiency (e.g., they may be used to generate good seed values).

Results about NP-hardness in theoretical computer science make heuristics the only viable option for a variety of complex optimization problems that need to be routinely solved in real-world applications.

Heuristics underlie the whole field of Artificial Intelligence and the computer simulation of thinking, as they may be used in situations where there are no known algorithms.^[3]

Trade-off

The trade-off criteria for deciding whether to use a heuristic for solving a given problem include the following:

- *Optimality*: When several solutions exist for a given problem, does the heuristic guarantee that the best solution will be found? Is it actually necessary to find the best solution?
- *Completeness*: When several solutions exist for a given problem, can the heuristic find them all? Do we actually need all solutions? Many heuristics are only meant to find one solution.
- *Accuracy and precision*: Can the heuristic provide a confidence interval for the purported solution? Is the error bar on the solution unreasonably large?
- *Execution time*: Is this the best-known heuristic for solving this type of problem? Some heuristics converge faster than others. Some heuristics are only marginally quicker than classic

Gambar 3.1 Contoh Data Dokumen Dari Wikipedia



Generative AI is a type of artificial intelligence technology that can produce various types of content including text, imagery, audio and synthetic data.

What is Artificial Intelligence? How is it different from Machine Learning?

AI is a discipline like Physics for example. AI is a branch of Computer Science that deals with the creation of intelligent agents which are systems that can reason and learn and act autonomously. Essentially AI has to do with the theory and methods to build machines that think and act like humans.

Machine Learning which is a subfield of AI is a program or system that trains a model from input data. The trained model can make useful predictions from new or never before seen data drawn from the same one used to train the model. Machine Learning gives the computer the ability to learn without explicit programming. Two of the most common classes of machine learning models are

1. Unsupervised Machine Learning
2. Supervised Machine Learning

The key difference between the two is that with Supervised models we have labels. Labelled data is data that comes with a tag like a name, a type or a number. Unlabeled data is data that comes with no such tag.

Gambar 3.2 Contoh data dokumen dari Medium

Towards Scientific Discovery with Generative AI: Progress, Opportunities, and Challenges

Chandan K Reddy, Parshin Shojaee

Virginia Tech
reddy@cs.vt.edu, parshinshojaee@vt.edu

Abstract

Scientific discovery is a complex cognitive process that has driven human knowledge and technological progress for centuries. While artificial intelligence (AI) has made significant advances in automating aspects of scientific reasoning, simulation, and experimentation, we still lack integrated AI systems capable of performing autonomous long-term scientific research and discovery. This paper examines the current state of AI for scientific discovery, highlighting recent progress in large language models and other AI techniques applied to scientific tasks. We then outline key challenges and promising research directions toward developing more comprehensive AI systems for scientific discovery, including the need for science-focused AI agents, improved benchmarks and evaluation metrics, multimodal scientific representations, and unified frameworks combining reasoning, theorem proving, and data-driven modeling. Addressing these challenges could lead to transformative AI tools to accelerate progress across disciplines towards scientific discovery.

Introduction

Scientific discovery - the process of formulating and validating new concepts, laws, and theories to explain natural phenomena - is one of humanity's most intellectually demanding and impactful pursuits. For decades, AI researchers have sought to automate aspects of scientific reasoning and discovery. Early work focused on symbolic AI approaches to replicate the formation of scientific hypotheses and laws in symbolic forms (Segler, Preuss, and Waller 2018; MacColl 1897). More recently, deep learning and large language models (LLMs) have shown promise in tasks like literature analysis and brainstorming (Ji et al. 2024; Lu et al. 2024; Si, Yang, and Hashimoto 2024), experiment design (Boiko et al. 2023; Arlt et al. 2024), hypothesis generation (Wang et al. 2024; Ji et al. 2024), and equation discovery (Shojaee et al. 2024b; Ma et al. 2024).

Despite this progress, we still lack AI systems capable of integrating the diverse cognitive processes involved in sustained scientific research and discovery. Most work has focused on narrow aspects of scientific reasoning in isolation. Developing more comprehensive AI discovery systems capable of supporting the full cycle of scientific in-

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

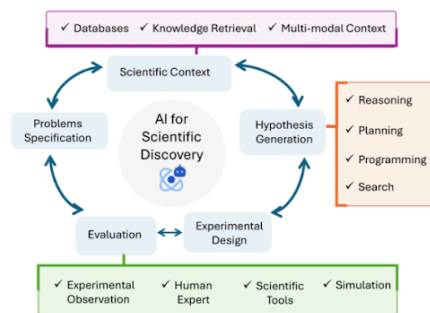


Figure 1: **Overview of the AI-driven scientific discovery framework.** The cycle illustrates the iterative process of scientific inquiry. The framework begins with user-defined problem specifications, retrieves relevant scientific context from literature and databases, and utilizes generative AI systems to produce new hypotheses and experimental designs. These AI-generated concepts are then evaluated and refined through experimental observation, expert input, and scientific tools, driving further iterations of the discovery cycle.

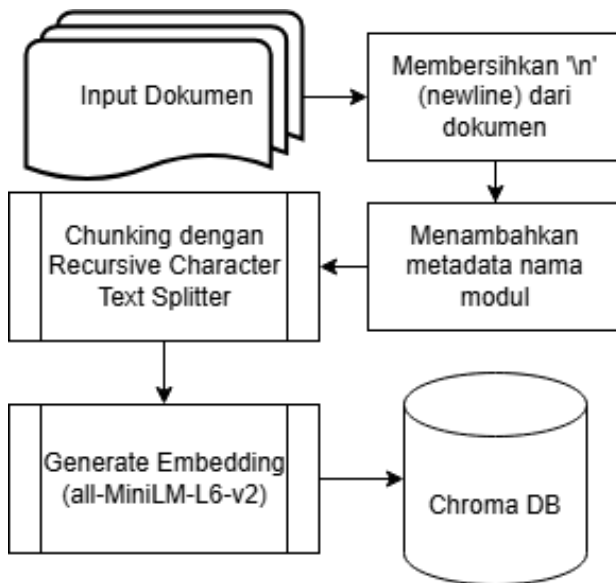
quiry—from context retrieval and hypothesis generation to experiment design and evaluation (Figure 1)—could dramatically accelerate progress across scientific disciplines. This paper examines the current state and future potential of generative AI for scientific discovery. We highlight recent advances, particularly in scientific understanding and discovery frameworks, while identifying critical gaps. We then outline key research challenges and directions towards more unified AI systems for discovery, including: (i) Creating improved benchmarks and evaluation frameworks for scientific discovery; (ii) Developing science-focused AI agents that leverage scientific knowledge and reasoning capabilities; (iii) Advancing multimodal scientific representations beyond text; and (iv) Unifying automated reasoning, theorem proving, and data-driven modeling. By tackling these challenges, the AI and Science community can work towards systems that serve as collaborative partners to human scientists, accelerating the pace of discovery in science.

Gambar 3.3 Contoh data dokumen dari arXiv

Sebelum dokumen-dokumen ini dimasukkan ke dalam sistem *Retrieval-Augmented Generation* (RAG), dilakukan tahapan *pre-processing* untuk meningkatkan kualitas data dan efisiensi pencarian. Dokumen-dokumen ini dikelompokkan ke dalam folder berdasarkan nama modul agar memudahkan proses penambahan *metadata* dan memastikan hanya informasi

relevan yang diambil saat *retrieval*. *Metadata* berupa nama modul ini nantinya berfungsi sebagai filter pencarian sehingga *chatbot* hanya menarik informasi dari modul yang sesuai dengan pertanyaan mahasiswa. Dalam implementasinya, dua modul, yaitu *Machine Learning* dan *Supervised, Unsupervised, and Reinforcement Learning*, digabung menjadi satu karena memiliki keterkaitan materi yang erat, sering diajarkan bersamaan, serta untuk menghindari redundansi informasi. Setelah dikelompokkan, dokumen melalui tahap pembersihan teks, termasuk menghapus karakter yang tidak diperlukan seperti *newline* ("*\n*"), tanda baca berlebihan, serta simbol khusus yang dapat mengganggu pemrosesan teks. Langkah ini memastikan bahwa struktur dokumen tetap konsisten dan siap untuk pemrosesan lebih lanjut. Selanjutnya, dilakukan *chunking* menggunakan *Recursive Character Text Splitter*. Metode ini dipilih karena mampu membagi dokumen berdasarkan panjang karakter tertentu secara hierarkis, memastikan bahwa setiap potongan teks tetap memiliki konteks yang cukup tanpa kehilangan keterkaitan antarbagian. Hal ini penting untuk meningkatkan kualitas *retrieval* karena sistem dapat mengenali keterkaitan antarbagian teks dengan lebih baik.

Setelah teks dibagi menjadi *chunk*, setiap potongan dikonversi menjadi *vector embedding* menggunakan model *all-MiniLM-L6-v2*. Model ini dipilih karena keseimbangannya antara efisiensi dan akurasi dalam merepresentasikan teks secara semantik. *Embedding* yang dihasilkan memungkinkan *chatbot* untuk melakukan pencarian berbasis makna, bukan sekadar pencocokan kata kunci. *Dataset* yang telah dikonversi ke dalam bentuk *embedding* kemudian disimpan dalam *ChromaDB*, sebuah *vector database* yang mendukung pencarian berbasis vektor dengan performa tinggi. *ChromaDB* dipilih karena memiliki struktur *indexing* yang efisien, memungkinkan pencarian informasi dengan *latency* rendah, serta kompatibel dengan berbagai model *embedding*. Selain itu, *ChromaDB* mendukung kueri berbasis *similarity search*, yang sangat penting dalam sistem RAG untuk mendapatkan dokumen yang paling relevan berdasarkan makna semantik. Dengan pendekatan ini, *chatbot* dapat memberikan jawaban yang lebih relevan dan berbasis sumber akademik, membantu mahasiswa memahami konsep AI & ML secara lebih efektif dalam konteks pembelajaran.

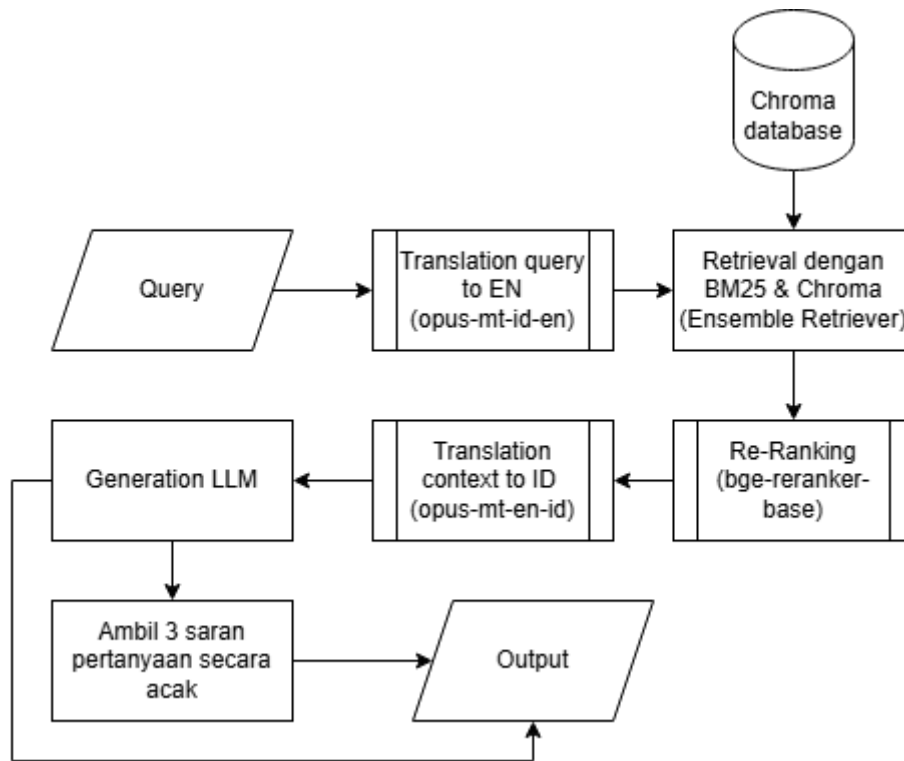


Gambar 3.4 Diagram Proses *Pre-processing Dataset*

3.3 Desain Sistem

Sistem RAG dalam penelitian ini dirancang untuk memberikan jawaban berbasis *retrieval* dengan mengombinasikan pencarian berbasis kata kunci dan semantik. Proses dimulai ketika pengguna memasukkan *query* ke dalam sistem. Jika *query* diberikan dalam Bahasa Indonesia, sistem akan menerjemahkannya ke dalam Bahasa Inggris menggunakan model *opus-mt-id-en* dari *Helsinki-NLP*. Translasi ini diperlukan karena sebagian besar *dataset* yang digunakan dalam *retrieval* berbahasa Inggris, sehingga meningkatkan peluang pencocokan dengan dokumen yang relevan. Setelah *query* siap, sistem melakukan *retrieval* menggunakan pendekatan *ensemble retriever* yang menggabungkan dua metode utama, yaitu *BM25Retriever* dari *Langchain* dan *ChromaDB Retriever*. *BM25* digunakan untuk mencocokkan kata kunci dalam *query* dengan dokumen dalam *database*, sedangkan *ChromaDB* menggunakan pencarian berbasis *embedding* untuk memahami makna semantik *query*. Kedua metode diberi bobot seimbang (masing-masing 50%) sehingga sistem dapat menangkap aspek literal dan semantik dari *query* secara bersamaan. Selain itu, sistem juga menyaring dokumen berdasarkan *metadata* modul yang sesuai dengan modul aktif pengguna, sehingga hanya informasi yang kontekstual dan sesuai dengan materi pembelajaran yang akan diambil.

Setelah memperoleh hasil retrieval awal, sistem menerapkan re-ranking menggunakan model bge-reranker-base dari BAAI yang diintegrasikan melalui Langchain. Model ini menyusun ulang dokumen berdasarkan skor relevansi dan mengeliminasi dokumen dengan relevansi rendah. Dari sepuluh dokumen yang diperoleh dari ensemble retriever, diambil beberapa teratas yang dipertahankan untuk memastikan kualitas konteks yang diberikan kepada model generatif. Langkah selanjutnya adalah pembuatan jawaban oleh model Nusantara-0.8b-Indo-Chat (Kusworo, 2024), yang telah dioptimalkan untuk menghasilkan teks dalam Bahasa Indonesia. Namun, sebelum membahas hal tersebut secara lebih lanjut, konteks yang didapatkan ditranslate ke dalam bahasa Indonesia terlebih dahulu agar chatbot dapat menyerap konteks dengan baik. Sebelum diteruskan ke model, sistem menyusun prompt dalam bahasa Indonesia yang berisi instruksi tugas, pertanyaan pengguna, serta dokumen hasil re-ranking. Pendekatan ini dirancang untuk meminimalkan potensi hallucination dan menjaga jawaban tetap berbasis sumber. Selain jawaban utama, sistem juga menyajikan saran pertanyaan tambahan yang dipilih secara acak dari daftar pertanyaan relevan per modul. Tujuannya adalah mendorong pengguna untuk mengeksplorasi materi lebih lanjut dan membangun pemahaman yang lebih dalam terhadap konsep yang sedang dipelajari. Dengan pendekatan ini, sistem tidak hanya menjawab pertanyaan secara langsung, tetapi juga berperan sebagai fasilitator pembelajaran aktif dan adaptif bagi mahasiswa.



Gambar 3.6 Diagram Proses Retrieval-Augmented Generation

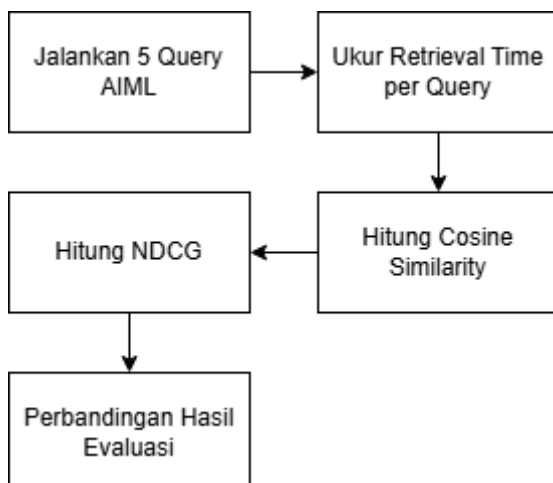
3.4 Pengujian

Pengujian dilakukan dalam dua tahap untuk mengevaluasi performa sistem chatbot berbasis *Retrieval-Augmented Generation (RAG)* yang dikembangkan dalam penelitian ini. Tahap pertama difokuskan pada pengujian kualitas lima konfigurasi *dataset* yang dibedakan berdasarkan ukuran *chunking* dokumen, yaitu 2000, 3000, 4000, 5000, dan 6000 karakter. Pemilihan ukuran ini didasarkan pada pertimbangan kapasitas *context* maksimal *model*, serta kebutuhan untuk menyeimbangkan cakupan informasi dalam satu *chunk* dengan efisiensi proses pencarian. *Chunk* yang terlalu pendek dapat menyebabkan fragmentasi informasi, sedangkan *chunk* yang terlalu panjang berpotensi menurunkan presisi karena *context* menjadi terlalu luas. Oleh karena itu, pengujian dilakukan terhadap variasi ukuran sedang hingga besar untuk mengetahui ukuran paling optimal.

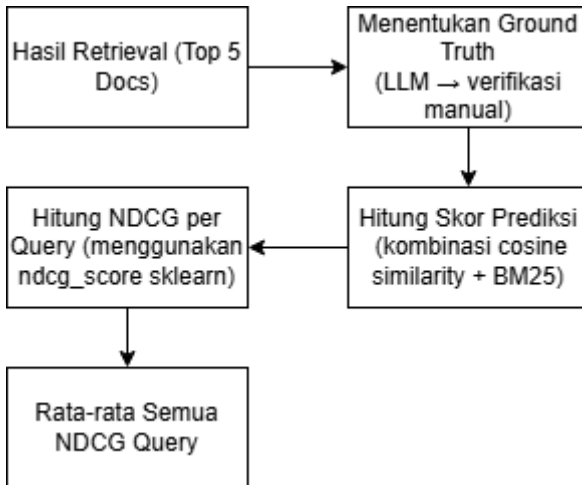
Evaluasi dilakukan menggunakan tiga metrik utama, yaitu *cosine similarity*, *retrieval time*, dan *Normalized Discounted Cumulative Gain (NDCG)*. *Cosine similarity* digunakan untuk mengukur kedekatan semantik antara *query* dan dokumen hasil *retrieval*. Nilai ini dihitung menggunakan *model embedding all-MiniLM-L6-v2* dari *SentenceTransformer*. Metrik ini dipilih karena

representasi *vector embedding* secara langsung mencerminkan kemiripan makna antar kalimat. *Retrieval time* diukur untuk menilai efisiensi sistem dalam mengambil dokumen. Sistem yang digunakan merupakan *hybrid retriever* (gabungan BM25 dan ChromaDB) yang dipadukan dengan *reranker* bge-reranker-base. Waktu pencarian dihitung menggunakan fungsi `time.time()` dari Python. Untuk memperoleh hasil yang stabil dan merepresentasikan performa sistem yang sesungguhnya, setiap *query* diuji sebanyak 20 kali, dan hasil waktu pengambilan dokumen dirata-ratakan.

Sementara itu, *NDCG* digunakan untuk mengukur kualitas peringkat dokumen hasil *retrieval* berdasarkan relevansi terhadap *query*. Nilai *y-score* untuk *NDCG* diperoleh dari kombinasi skor BM25 dan *cosine similarity* yang telah dinormalisasi dengan metode *min-max*, lalu digabungkan dengan bobot rata (50:50). Kombinasi ini digunakan karena skor BM25 bersifat leksikal dan skor *cosine similarity* bersifat semantik, sehingga penggabungannya diharapkan memberikan penilaian yang lebih representatif. *Ground truth* relevansi dokumen terhadap *query* awalnya dinilai oleh ChatGPT sebagai referensi awal, kemudian diverifikasi dan disesuaikan secara manual oleh peneliti. Dosen pembimbing turut dilibatkan dalam validasi akhir untuk memastikan akurasi dan validitas akademik dari penilaian tersebut. Perhitungan *NDCG* dilakukan menggunakan fungsi `ndcg_score` dari pustaka *scikit-learn*, dan hasilnya dirata-ratakan untuk tiap konfigurasi *chunk*. Ketiga metrik ini digunakan secara terpadu untuk membandingkan performa antar konfigurasi *chunk* dan menentukan ukuran *chunk* yang paling optimal dalam hal ketepatan hasil pencarian dan efisiensi waktu.



Gambar 3.6 Diagram Proses Pengujian Dataset



Gambar 3.7 Diagram Proses Pengujian NDCG

Pada tahap kedua, dilakukan pengujian *chatbot* secara kualitatif dengan melibatkan mahasiswa di Universitas Kristen Petra. Mahasiswa diminta mengajukan pertanyaan terkait materi dari mata kuliah *Artificial Intelligence and Machine Learning*, dan *chatbot* akan memberikan jawaban berdasarkan sistem RAG yang telah dibangun. Setelah beberapa interaksi dengan *chatbot*, mahasiswa diminta memberikan penilaian terhadap kejelasan jawaban, relevansi jawaban,, dan tingkat kepuasan mereka dalam menggunakan *chatbot* dengan menggunakan Skala *Likert* 1-5, dengan kategori: 1 (sangat tidak puas), 2 (tidak puas), 3 (netral), 4 (puas), dan 5 (sangat puas). Tujuan dari pengujian ini adalah untuk mengevaluasi sejauh mana *chatbot* mampu mendukung pemahaman materi secara interaktif. Hasil pengujian dianalisis dengan menghitung rata-rata skor *Likert* dari seluruh responden untuk memperoleh gambaran umum mengenai kepuasan pengguna terhadap sistem yang dikembangkan.