BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1. Knowledge Management System

Knowledge Management System (KMS) merupakan sistem untuk mengidentifikasi, mengelola dan membagikan pengetahuan bagi perusahaan atau organisasi untuk mencapa tujuan (Fathi Azzumar & Nurmiati, 2021).

Komponen penting dari *knowledge management* adalah *people, process*, dan *technology* (Lumbantobing et al., 2007). Masing-masing dari komponen ini saling berhubungan dan terikat satu sama lain sebagai landasan kuat untuk *knowledge management*

Fathi Azzumar & Nurmiati (2021) menjelaskan tiga pendekatan untuk mengelola pengetahuan dalam *knowledge management* sebagai berikut :

1. *People* (orang), berarti merupakan individu yang memiliki pengetahuan, mencari pengetahuan dan menyalurkan pengetahuan serta memiliki komitmen untuk pengetahuan organisasi atau perusahaan
2. *Process* (Proses), proses ini menjelaskan bagaimana *knowledge* di akuisisi, distribusi dan utilisasi dengan baik seperti semestinya agar selaras dengan prinsip dan tujuan dari organisasi atau perusahaan.
3. *Technology* (Teknologi), merupakan komponen penting yang menjadi fokus pada infrastruktur untuk memfasilitasi *knowledge* dalam implementasinya seperti dokumentasi, kolaborasi dan distribusi antar *people.*

2.1.2. Large Language Model (LLM)

Large Language Model (LLM) adalah sebuah tipe kecerdasan buatan (AI) yang dipergunakan untuk mengenali, menghasilkan, meringkas, dan menerjemahkan bahasa manusia. Mereka berbeda dari pendekatan sebelumnya terhadap *Natural Language Processing* (NLP) karena mereka berdasarkan pada kumpulan data yang sangat besar dan dirancang untuk mengekstrak dan mereplikasi aturan bahasa (Radford et al., n.d.).

GPT-3, salah satu LLM terbesar yang pernah dikembangkan, memiliki kinerja yang lebih baik dan perilaku baru meskipun satu-satunya perbedaan dari model sebelumnya adalah ukurannya (Brown et al., 2020).

Munculnya LLM mengubah paradigma pada *Natural Language Processing* (NLP) menjadi peningkatan klasifikasi, *generation* dan pemahaman teks. Namun, LLM seringkali membutuhkan adaptasi lanjut agar dapat digunakan pada tugas yang spesifik. Hal ini membuat pendekatan yang berbeda dalam meningkatkan limitasi dari model yang telah dilatih sebelumnya. Pendekatan tersebut dapat dilakukan dengan cara *full fine-tuning*, *parameter-efficient fine-tuning* (PEFT) , *prompt engineering* dan *retrieval augmented generation* (RAG) (Radeva et al., 2024).

2.1.3. Generative Pre-Trained Transformer-3.5 (ChatGPT-3.5)

Menurut Setiawan et al (2023), Chat GPT *(Generative Pre-trained Transformer)* adalah sebuah model bahasa berbasis AI yang dibuat dan dikembangkan oleh OpenAI. Model ini dapat memahami dan menghasilkan bahasa yang mudah dikenali menggunakan pendekeatan *deep* learning yang disebut dengan

transformer. GPT dibangun pada arsitektur *transformer* yang memahami hubungan antara kata-kata dalam teks.

Mulai tahun 2018, OpenAI menciptakan GPT-1, GPT-2, dan merilis GPT-3 (Abate et al., 2023). ChatGPT 3.5 adalah versi upgrade dari ChatGPT 3, dengan beberapa peningkatan dalam hal akurasi, keamanan, dan kegunaan. ChatGPT 3.5 secara umum dianggap lebih akurat daripada ChatGPT 3. GPT 3.5 digunakan dalam penelitian ini karena gratis

GPT-3.5 mampu menghasilkan teks yang koheren dan konsisten dengan konteks yang disediakan. GPT-3.5 dapat digunakan untuk berbagai tugas, termasuk *text summarization*, menjawab pertanyaan, dan *text generation*.(Oğuz “oz” Buruk, n.d.)

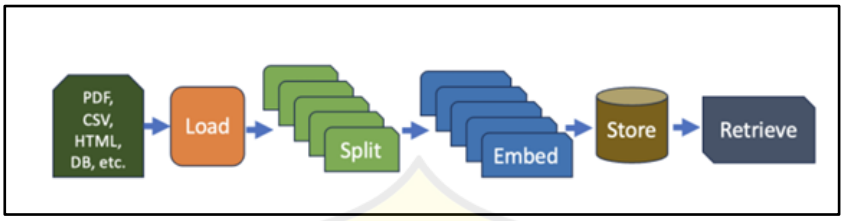
2.1.4. Retrieval-Augmented Generation (RAG)

Retrieval Augmented Generation meningkatkan kinerja *language models* dengan menggabungkan *prompt engineering* dan *query database* untuk memberikan jawaban yang kaya akan konteks, mengurangi errors dan beradaptasi dengan data yang baru secara efisien (Radeva et al., 2024). Menurut Radeva et al (2024), konsep utama RAG adalah kombinasi antara language models yang telah dilatih sebelumnya dengan menerima database dari luar untuk memperluas pengetahuan dan menghasilkan jawaban yang dinamis.

2.1.4.1. LangChain

LangChain adalah framework untuk mengembangkan aplikasi yang menggunakan Large Language Model dengan tujuannya adalah untuk memungkinkan pengembang dengan mudah memanfaatkan sumber data lain dan berinteraksi dengan aplikasi lain (Topsakal & Akinci, 2023). LangChain telah

mendapatkan banyak perhatian dari komunitas AI karena dapat memberikan solusi untuk membangun aplikasi custom AI yang memanfaatkan Large Language Model. Salah satu dari banyak sistem yang dibuat dengan LLM adalah sistem yang dapat menjawab pertanyaan dengan data dari suatu dokumen.



Gambar 2. 1 Langkah untuk menjawab pertanyaan dari dokumen (Topsakal & Akinci, 2023)

2.1.4.2. ChromaDB

ChromaDB adalah database *vector* berbasis AI yang bersifat *open-source*. ChromaDB menawarkan kemudahan untuk membangun aplikasi LLM dengan memiliki *tools* seperti menyimpan *embedding* beserta metadata hingga melakukan pencarian *embedding* dengan *query* menggunakan *semantic search*. ChromaDB memiliki prioritas seperti sederhana, cepat dan produktivitas bagi *developer*. (*Chroma Docs*, n.d.). Penggunaan ChromaDB memiliki kelebihan seperti *Open Source*, kapabilitas pencarian yang lebih fleksibel dan komunitas pengguna yang berkembang. Selain itu, ChromaDB juga telah di-dukung oleh *framework* pengembangan LLM seperti LangChain.

2.1.5. Chatbot

Chatbot adalah salah satu bentuk sistem menggunakan *Natural Language Processing* (NLP) yang dapat mempelajari komunikasi antara manusia dengan komputer menggunakan bahasa alami (Nila & Afrianto, 2015).

Chatbot memiliki cara yang baru dalam penyampaian suatu informasi. Pengguna dapat langsung menanyakan chatbot terkait berita acara, orang, tempat dengan jawaban yang relevan. (Soyusiawaty & Ganda Putra, 2023).

2.1.6. Telegram

Telegram adalah aplikasi pesan berbasis cloud yang menawarkan fitur unggulan seperti gratis, ringan dan multi-platform. (Fitriansyah, n.d.). Telegram tidak sebatas platform chat saja, telegram juga menyediakan API untuk dipakai oleh developer sesuai dengan kebutuhan. Biasanya API tersebut digunakan untuk membuat chatbot, Telegram menawarkan dua metode API untuk menjalankan chatbot yaitu long-polling dan webhook. Long-polling cocok digunakan jika kita tidak memiliki hosting atau web server. Sedangkan webhook, kita harus memiliki web server yang aktif karena telegram akan menyalurkan ke server jika terdapat pesan baru. (Hasyim et al., 2021).

2.1.7. Django

Menurut situs resmi Django Django Software Foundation, n.d.), Django adalah *framework* website berlisensi *open-source* dengan bahasa pemrograman python. Django menawarkan keunggulan seperti *framework* yang cepat, fitur lengkap, keamanan yang serius, *scalable* dan juga *versatile.*

Django menerapkan *design pattern* Model-View-Controller (MVC) namun saat implementasinya memiliki logic tersendiri yang lebih dikenal dengan Model-View-Template (MVT). Struktur MVT memiliki 3 bagian seperti :

1. *Model* sebagai representasi dari data yang digunakan. Pada model ini kita dapat melakukan bagaimana melakukan validasi data, perilaku data dan hubungan antar data dengan yang lain.

2. *View* adalah *user interface* atau tampilan yang dilihat pada *user* ketika mengakses *website* pada *browser*. Pada bagian *view* akan mengatur alur antara *model* dengan *template*. Pada *view* juga kita dapat menulis logika bisnis.
3. *Template* berisi bagian statis dari HTML *output* dan beberapa syntax untuk mendeskripsikan bagaimana konten akan ditampilkan. (Ramesh et al., n.d.).

2.1.8. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-*user* (Fitri, 2020). Database *MySQL* memiliki beberapa keunggulan seperti kecepatan, *open source*, bahasa *query*, *multiple user*, dan mudah digunakan. Database MySQL juga memiliki *query* yang dilengkapi dengan banyak operator seperti SELECT dan WHERE untuk memudahkan mengolah data yang diinginkan.

2.1.9. Rapid Application Development (RAD)

Rapid Aplication Development (RAD) adalah sebuah metode pengembangan perangkat lunak yang dikenal dengan tahapan proses yang cepat biasanya membutuhkan waktu sekitar 60 sampai 90 hari dalam membangun sistem. Model RAD menggunakan metode tahapan berulang dan tahapan yang fleksibel tanpa menunggu suatu tahapan harus selesai terlebih dahulu .(Pricillia & Zulfachmi, 2021). Model RAD memiliki tahapan sebagai berikut :

1. Rencana Kebutuhan (*Requirement Planning*): Tahapan ini untuk mengidentifikasi tujuan dan kebutuhan dari sistem yang ingin dibangun.
2. Proses Desain Sistem *(Design System*): Pada tahapan ini akan menentukan bagaimana sistem berjalan secara umum, struktur data dan lain – lain. Peran

user yang aktif sangat dibutuhkan pada tahapan ini, jika terdapat desain yang tidak sesuai *user* dapat langsung mengubah atau mengomentari secara berulang – ulang.

3. Implementasi (*Implementation*): Tahapan ini adalah proses bagi *programmer* untuk membangun sistem berdasarkan desain yang telah dibuat. Pada tahapan ini juga *user* bisa melakukan uji coba, memberikan tanggapan dan mendapatkan persetujuan akan sistem yang dibangun.

2.1.9.1.Unified Model Language (UML)

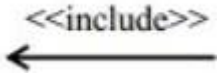
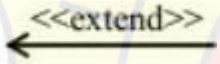
UML atau *Unified Model Language* merupakan pemodelan visual secara standar untuk dokumentasi, spesifikasi dalam Pembangunan *software* yang berbasis *Object Programming Language* (Kurniawan et al, 2021).

2.1.10.1. Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan interkasi antara aktor dan proses apa saja yang dikerjakan pada sistem. *Use case* diagram popular digunakan karena dapat melihat secara gambaran umum hubungan antara aktor dan *use case* pada sistem. (Kurniawan et al., 2021).

Tabel 2. 1 *Use case* diagram (Hendini, 2016)

Simbol	Nama	Keterangan
	Aktor	Orang atau sistem lain yang berinteraksi dengan sistem
	*Use case*	Fungsionalitas sistem yang dinyatakan dengan kata kerja

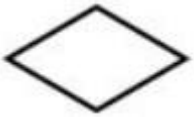
	*Association*	Hubungan antara *use case* dan aktor
	Generalisasi	Bentuk generalisasi lain dari suatu aktor atau *use case*
	Relasi *include*	Menunjukan bahwa suatu *use case* merupakan bagian dari *use case* lainnya
	Relasi *extend*	Tambahan fungsionalitas dari *use case* lainnya

2.1.10.2. Activity Diagram

Menurut Muhammad (2019) *Activity Diagram* merupakan rancangan diagram aliran aktivitas terhadap sistem.

Tabel 2. 2 *Activity* Diagram (Hendini, 2016)

Simbol	Nama	Keterangan
	Status awal	Awal dari aktivitas

	Aktivitas	Aktivitas atau proses yang terjadi pada sistem
	Percabangan/ *decision*	Percabangan untuk pemilihan keputusan
	Penggabungan/ *join*	Penggabungan lebih dari satu aktivitas menjadi satu atau dekomposisi
	Status akhir	Akhir dari aktivitas
	*Swimlane*	*Swimlane*, untuk menunjukan siapa yang melakukan apa

2.1.10.3. Entity Relationship Diagram

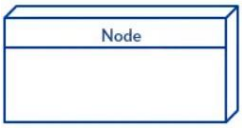
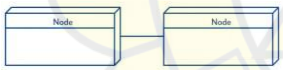
ERD (*Entity Relationship Diagram)* merupakan diagram untuk menggambarkan relasi atau hubungan antar entitas yang ada pada *database* beserta atributnya secara detail. Dengan adanya ERD ini diharapkan dapat membantu untuk menampilkan struktur *database* secara lebih rapi dan mudah dibaca bagi *programmer* yang membangun sistem.

2.1.10.4. Deployment Diagram

Deployment diagram menggambarkan secara detail komponen akan disebar atau di-*deploy* ke suatu infrastruktur sebuah sistem. *Deployment diagram*

menggambarkan *device, node* serta *hardware* yang digunakan untuk memodelkan sistem pada *client server* atau sistem tambahain lain. (Irianto et al., 2021).

Tabel 2. 3 Deployment Diagram (Hendini, 2016)

Simbol	Nama	Keterangan
	Node	Entitas fisik yang mengeksekusi lebih dari satukomponen, subsistem, atau *executables* lain.
	Artifak	Elemen konkret yang menyebabkan proses pengembangan, seperti libraries, configuration file, dan sebagainya
	*Communication Association*	Menggambarkan jalan komunikasi antar 2 node

2.2 Kajian Penelitian Terdahulu

2.2.1. Paper 1 : UJI PERFORMA CHATBOT DENGAN RETRIEVAL AUGMENTEDGENERATION DAN MODEL GPT-4 UNTUK DOMAIN TAHARAHBERDASARKAN EMPAT IMAM MAZHAB FIKIH (STUDI KASUSKITAB RAHMAH AL UMMAH FI IKHTILAF AL A'IMMAH). ROYYAN ABDURROHMAN, 2024.

2.1.1.1. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengimplementasikan *chatbot* dengan metode *Retrieval Augmented Generation* (RAG) untuk domain taharah berdasarkan empat imam mazhab fikih. Dengan adanya penelitian ini diharapkan *chatbot* dapat membantu dalam memberikan informasi terkait taharah berdasarkan perbedaan mazhab fikih.

2.1.1.2. Metode Penelitian

Metode yang dipakai dalam penelitian ini menggunakan *Waterfall* untuk pengembangan sistem dan *Retrieval Augmented Generation* (RAG) untuk pengembangan Large Language Models (LLM). Dalam tahapan *requirement* dalam metodologi *waterfall*, peneitian ini melakukan proses mengakuisisi data yangdibutuhkan dalam perancangan chatbot. Karena chatbot yang akan dibangun ini dikhususkan pada permasalahan taharah perbedaan mazhab fikih, maka diperlukan data-data mengenai hal tersebut dari berbagai kalangan mazhab, dalam hal ini berfokus pada empat imam besar mazhab (Syafii, Maliki, Hambali, dan Hanafi). Dalam teks-teks yang belum terstruktur tersebut, penelitimengorganisasi teks tersebut supaya data menjadi lebih terstruktur dan siap untukdimasukkan ke dalam basis data graf. Dalam basis data graf sendiri, data disimpandalam bentuk node dan relationship yang dapat memiliki properti.

Tahapan selanjutnya adalah pemodelan *chatbot* dengan menggunakan *framework* LangChain yang mengintegrasikan LLM seperti ChatGPT-4 terhadap *knowledge base* yang diinginkan. Proses integrasi ini menggunakan metode *Retrieval Augmented Generation* (RAG) yang dapat diharapkan untuk memberikan jawaban yang akurat dan meminimalkan tingkat halusinasi dari model GPT-4 dengan dukungan sumber yang kredibel.

2.1.1.3. Temuan Utama

Temuan utama dalam penelitian ini adalah pada proses pengujian *chatbot* diuji dengan berbagai pernyataan, baik yang di luar maupun dari *knowledge base* yang terintegrasi. Jumlah pernyataan yang diujikan berjumlah 40. Pertanyaan tersebut diuji dengan *chatbot* dengan integrasi RAG dan hanya *chatbot* biasa saja

Dalam pengujian yang dilakukan, dari 40 pertanyaan terdapat 36 jawaban jawaban yang dihasilkan berhasil melakukan RAG dan 4 jawaban yang tidak bisa dijawab dengan benar. Sedangkan pada pengujian yang hanya memakai model GPT-4, dari 40 pertanyaan, terdapat 28 jawaban yang mampu dijawab dengan baik dan 12 pertanyaan dengan tidak tepat.