Jurnal Teknologi Informasi, Komputer dan Aplikasinya (JTIKA)
Vol. 7, No. 1, Maret 2025, (Terakreditasi Sinta-4, SK No:164/E/KPT/2021) ISSN:2657-0327

Optimalisasi Layanan Sistem Informasi Mahasiswa dengan Integrasi Telegram : Chatbot Retrieval-Augmented-Generation berbasis Large Language Model

(*Optimization of Student Information System Services with Telegram Integration : Chatbot Retrieval-Augmented Generation based on Large Language Model*)

Lalu Ramdoni Hidayat*[1],I Gede Pasek Suta Wijaya[2], Ramaditia Dwiyansaputra[3]
[1]Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: laluramdoni3@gmail.com, gpsutawijaya@unram.ac.id, rama@unram.ac.id

Abstract

Kemajuan teknologi telah memberikan dampak yang cukup signifikan dalam berbagai bidang, termasuk salah satunya Pendidikan. Dalam aspek Pendidikan permasalahan yang dihadapi adalah keterbatasan akses mahasiswa terhadap informasi akademik secara cepat dan efisien. Untuk mengatasi hal ini, penelitian ini bertujuan mengembangkan chatbot berbasis Telegram yang mampu memberikan respons informatif, akurat, dan ringkas terhadap pertanyaan pengguna terkait akademik di program studi Teknik Informatika. Chatbot ini memanfaatkan metode Retrieval-Augmented-Generation (RAG) untuk memproses informasi dari dokumen teks secara efisien. Metode RAG digunakan untuk menemukan jawaban yang relevan dari dokumen berdasarkan pertanyaan pengguna, sementara Large Language Model memahami konteks pertanyaan dan menghasilkan jawaban yang sesuai. Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan tahapan meliputi survei questioner kebutuhan mahasiswa, preprocessing data, Pembangunan indeks pencarian berbasis vektor, konfigurasi model LLM, serta integrasi chatbot dengan Telegram. Hasil pengujian menunjukkan bahwa chatbot mampu memberikan jawaban dengan akurasi tinggi dan waktu respons rata-rata 60 detik untuk pertanyaan sederhana hingga kompleks, sehingga chatbot berbasis RAG cukup efektif meningkatkan aksesibilitas informasi secara real-time. Pengembangan lebih lanjut dapat difokuskan pada peningkatan pemahaman terhadap beragam pertanyaan dan personalisasi respons.

Keywords: *Deep Learning, Chatbot, Telegram, Retrieval-Augmented-Generation, Large Language Model*

**Correspondence Author*

1. PENDAHULUAN

Perkembangan zaman beriringan dengan kemajuan teknologi, yang berarti teknologi terus berkembang seiring perubahan waktu. Kemajuan teknologi memberikan pengaruh besar pada berbagai sektor kehidupan, termasuk Pendidikan. Dalam era digital saat ini, mahasiswa tidak lagi kesulitan mencari sumber belajar atau membawa banyak buku untuk perkuliahan. Semua informasi yang dibutuhkan tersedia secara mudah melalui internet. Mahasiswa kini dapat memanfaatkan media sosial, situs web, dan platform digital lainnya untuk menunjang pembelajaran[1].

Chatbot menjadi salah satu solusi inovatif untuk memberikan layanan informasi, termasuk dalam konteks pendidikan. Dengan chatbot, mahasiswa dapat berinteraksi langsung dengan sistem berbasis bot untuk mendapatkan jawaban atas berbagai pertanyaan. Teknologi ini memungkinkan mahasiswa memperoleh informasi dengan cepat tanpa harus bertanya dahulu kepada staf program studi. Keunggulan chatbot terletak pada kemampuannya menyediakan berbagai konten menarik, seperti gambar, video, dan informasi relevan lainnya yang dapat meningkatkan minat pengguna[2].

Dalam penerapannya di program studi Teknik Informatika, chatbot dapat digunakan untuk

memberikan layanan sistem informasi. Mahasiswa dapat menanyakan berbagai hal terkait perkuliahan, jadwal, kurikulum, dan prosedur administrasi lainnya. Chatbot akan merespons pertanyaan secara spesifik sesuai dengan basis data yang telah disediakan, sehingga mahasiswa mendapatkan solusi yang tepat tanpa harus menanyakannya langsung di loket program studi atau mengirimkan email.

Teknologi di balik chatbot melibatkan kecerdasan buatan (AI), pembelajaran mesin (*Machine Learning*), pembelajaran mendalam (*Deep Learning*), dan pemrosesan bahasa alami (Natural Language Processing atau NLP). Cara kerjanya adalah dengan menganalisis kata kunci dari input pengguna, lalu memberikan respons yang paling sesuai berdasarkan pola yang ada dalam basis data. Saat mahasiswa mengajukan pertanyaan, chatbot akan mencari jawaban yang relevan dan menyajikannya dalam format teks atau media lainnya. Dengan begitu, mahasiswa hanya perlu memberikan perintah atau pertanyaan, dan chatbot akan secara otomatis mencari data yang diperlukan[3].

Implementasi chatbot di program studi Teknik Informatika bertujuan untuk meningkatkan efisiensi layanan informasi bagi mahasiswa. Dengan teknologi ini, mahasiswa dapat memperoleh akses informasi kapan saja tanpa tergantung pada jam operasional staf. Penggunaan chatbot juga mendorong proses belajar menjadi lebih efektif, karena mahasiswa dapat memperoleh penjelasan yang relevan dengan cepat dan mudah. Oleh karena itu, pengembangan chatbot dengan data yang sesuai kebutuhan mahasiswa sangatlah penting untuk mendukung kemajuan layanan pendidikan berbasis teknologi.

Chatbot yang dirancang menggunakan pendekatan *Retrieval Augmented Generation* (RAG) dan *Large Language Models* (LLM) memberikan keunggulan dalam hal pengolahan dan penyediaan informasi. Metode RAG memungkinkan chatbot untuk menggabungkan kekuatan basis data yang terstruktur dengan kemampuan generatif dari LLM, sehingga jawaban yang diberikan tidak hanya akurat tetapi juga relevan dengan konteks pertanyaan. Proses ini bekerja dengan cara mengakses basis data yang telah disusun secara sistematis dan menggunakan model bahasa besar untuk menyusun respons yang komprehensif[4].

Penggunaan metode RAG pada chatbot ini juga memastikan bahwa informasi yang disampaikan tetap terkini dan sesuai dengan kebutuhan pengguna. Model ini dapat menyesuaikan respons berdasarkan pembaruan data terbaru, sehingga mahasiswa selalu mendapatkan jawaban yang relevan dengan kondisi terkini. Selain itu, integrasi teknologi LLM memungkinkan chatbot untuk memahami pertanyaan yang kompleks dan memberikan jawaban yang mendalam, sehingga mendukung kebutuhan akademik mahasiswa secara lebih baik[5].

Untuk meningkatkan aksesibilitas, chatbot ini dirancang agar dapat diintegrasikan langsung dengan platform Telegram. Telegram dipilih karena popularitasnya di kalangan mahasiswa dan kemampuannya untuk mendukung percakapan real-time secara efektif. Dengan integrasi ini, mahasiswa dapat mengakses layanan chatbot kapan saja dan di mana saja hanya melalui aplikasi Telegram yang tersedia di perangkat mereka. Hal ini tidak hanya mempermudah komunikasi tetapi juga meningkatkan efisiensi waktu, karena mahasiswa tidak perlu lagi menunggu balasan dari staf prodi[1].

2. TINJAUAN PUSTAKA

2.1. Perkembangan Chatbot

Istilah "chatbot" berasal dari gabungan kata "*chat*" dan "*robot*," awalnya merujuk pada sistem dialog berbasis teks yang mensimulasikan bahasa manusia. Pada masa awal, chatbot menggunakan teknik pencocokan pola untuk menciptakan pengalaman percakapan sederhana melalui aturan dan template yang telah ditentukan. Pionir seperti Eliza dan ALICE menggunakan pendekatan ini untuk mencocokkan pola masukan dengan keluaran template tertentu. Namun, metode ini memiliki keterbatasan, seperti respons yang cenderung repetitif dan kurang personal, sehingga tidak ideal untuk percakapan yang kompleks. Meski sederhana, chatbot berbasis pola ini tetap populer untuk aplikasi skala kecil yang tidak memerlukan konfigurasi komputasi tinggi. Pada era modern, chatbot berkembang jauh melampaui interaksi berbasis teks sederhana. Teknologi seperti *Artificial Intelligence* (AI), *Natural Language Processing* (NLP), dan *Machine Learning* (ML) memungkinkan chatbot untuk belajar dan beradaptasi seiring waktu, memberikan interaksi

yang lebih personal dan menarik. Kemajuan teknologi ini terlihat jelas dengan munculnya chatbot seperti ChatGPT dan Bard yang menggunakan *Transformer Neural Network Architecture*. Arsitektur ini unggul dalam memproses urutan masukan yang panjang dan kompleks, sehingga menghasilkan respons yang koheren dan kontekstual. Melalui pelatihan berbasis data tekstual yang sangat besar, model ini mampu mengenali pola bahasa dan hubungan antar kata dengan presisi tinggi[6].

2.2. Natural Language Processing

Natural Language Processing (NLP) adalah teknologi yang memungkinkan komputer memproses dan memahami bahasa manusia, baik lisan maupun tulisan, yang digunakan dalam percakapan sehari-hari. Untuk menjalankan proses ini, bahasa manusia perlu direpresentasikan dalam bentuk simbol-simbol yang mengikuti aturan tertentu[7]. Dengan NLP, komputer dapat memahami perintah dan standar bahasa yang biasa digunakan manusia. Hasil atau keluaran dari sistem ini dihasilkan berdasarkan makna yang telah diringkas dari input yang diberikan oleh pengguna sebelumnya[3].

2.3. Large Language Model

Kecerdasan Buatan (AI) telah menjadi katalis utama dalam inovasi teknologi, termasuk pengembangan sistem informasi berbasis chatbot[8]. Dalam konteks layanan di Program Studi Teknik Informatika, pengintegrasian metode *Retrieval-Augmented Generation* (RAG) dengan *Large Language Models* (LLM), seperti ChatGPT 3.5, ke dalam platform Telegram Bot, menghadirkan solusi praktis untuk mempermudah mahasiswa memperoleh informasi tanpa harus menghubungi staf secara langsung. RAG memanfaatkan kombinasi pencarian dokumen yang relevan dan kemampuan generatif LLM, memastikan chatbot mampu memberikan jawaban yang akurat, relevan, dan berbasis data terkini[9].

Python, sebagai bahasa pemrograman dengan ekosistem perpustakaan yang luas seperti TensorFlow, PyTorch, dan scikit-learn, menjadi tulang punggung pengembangan chatbot berbasis RAG. Dengan dukungan teknologi ini, chatbot dapat diintegrasikan secara efisien ke dalam Telegram Bot, sehingga mahasiswa dapat mengakses informasi kapan saja dan di mana saja. Model LLM berfungsi untuk memahami dan merespons pertanyaan mahasiswa berdasarkan konteks, sementara metode RAG memastikan jawaban yang diberikan didukung oleh sumber data yang valid. Pendekatan ini tidak hanya meningkatkan kecepatan akses informasi, tetapi juga memastikan kualitas dan keakuratan layanan chatbot[9].

2.4. Retrieval-Augmented-Generation

Retrieval-Augmented Generation (RAG) adalah metode yang digunakan untuk meningkatkan kinerja *Large Language Models* (LLMs) dalam menjawab pertanyaan spesifik yang membutuhkan informasi di luar data pelatihan model[10]. Metode ini bekerja dengan cara mengambil informasi dari sumber data eksternal dan memberikannya sebagai konteks tambahan kepada LLM untuk menghasilkan respons. Dengan pendekatan ini, RAG dapat meningkatkan akurasi faktual dan relevansi jawaban, terutama pada pertanyaan yang bersifat spesifik atau memerlukan data terkini. Meski RAG dapat diterapkan pada tahap *pre-training*, penggunaannya lebih umum dilakukan pada tahap *inference* karena lebih praktis dan efisien[4].

Metode RAG sangat penting dalam aplikasi domain tertentu, seperti bahasa Indonesia, yang sering menghadapi keterbatasan data pelatihan. Dengan memberikan akses langsung ke sumber informasi tambahan, RAG memungkinkan model untuk menghasilkan respons yang lebih relevan dan tepat guna, mengatasi masalah seperti kesalahan atau imajinasi (*hallucinations*) yang sering terjadi pada LLM ketika menghadapi pertanyaan di luar lingkup data latihannya[4].

2.5. Optimalisasi Chatbot dengan Metode Retrieval-Augmented Generation (RAG) dalam Layanan Informasi Akademik

Metode *Retrieval-Augmented Generation* (RAG) menawarkan pendekatan inovatif untuk mengembangkan chatbot yang melayani kebutuhan informasi akademik di Program Studi Teknik Informatika. Dengan metode ini, chatbot mampu mengakses dan mengintegrasikan data eksternal yang spesifik, seperti jadwal kuliah, informasi administrasi, serta panduan akademik, sehingga dapat memberikan respons yang akurat dan relevan berdasarkan data

terkini. RAG bekerja dengan menggabungkan kemampuan pencarian dokumen yang relevan dengan fitur generatif dari *Large Language Models* (LLM)[4]. Pada tahap inferensi, data eksternal disisipkan sebagai konteks tambahan untuk memperkaya kualitas respons yang diberikan. Hal ini memungkinkan mahasiswa mendapatkan informasi secara cepat dan efisien, tanpa perlu berkomunikasi langsung dengan staf program studi, sekaligus memastikan jawaban yang diberikan didukung oleh sumber data yang valid dan terintegrasi dengan sistem[11].

2.6. Integrasi Telegram Bot sebagai Antarmuka Chatbot

Penggunaan Telegram Bot sebagai antarmuka utama untuk chatbot berbasis RAG memberikan kemudahan akses yang optimal bagi mahasiswa Teknik Informatika. Telegram Bot dipilih karena kemampuannya yang mendukung lintas platform, proses implementasi yang sederhana, dan kapasitasnya untuk menangani percakapan interaktif dalam jumlah besar[12]. Dalam pengembangannya, Python digunakan sebagai bahasa pemrograman utama dengan dukungan pustaka seperti TensorFlow dan PyTorch untuk memastikan performa yang andal[1]. Telegram Bot memungkinkan mahasiswa untuk memperoleh informasi akademik dengan mudah, kapan saja, dan di mana saja, melalui respons yang dihasilkan oleh model RAG yang terintegrasi dengan data spesifik program studi[13].

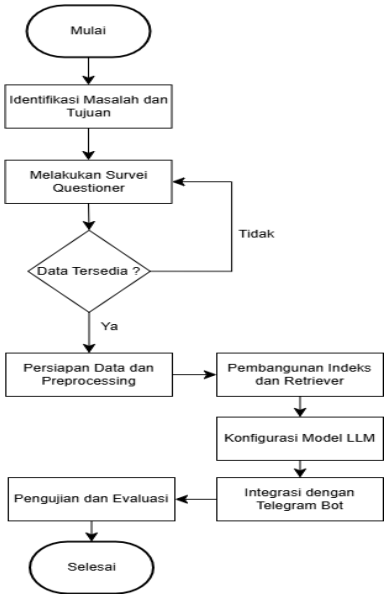
2.7. Kebaruan dan Efisiensi Metode Retrieval-Augmented Generation (RAG) dalam Integrasi Large Language Models (LLM)

Metode *Retrieval-Augmented Generation* (RAG) dan Large Language Models (LLM) menawarkan pendekatan inovatif dalam pengolahan bahasa alami, terutama untuk tugas-tugas yang memerlukan penggabungan antara kemampuan pencarian informasi dan generasi teks yang relevan. RAG mengintegrasikan mekanisme retrieval untuk mendapatkan informasi terkait dari basis data eksternal dengan kemampuan generatif LLM, sehingga menghasilkan jawaban yang lebih akurat dan kontekstual. Kebaruan metode ini terletak pada kemampuannya untuk mengatasi keterbatasan LLM dalam memahami konteks yang memerlukan referensi eksternal secara langsung. Dalam konteks penelitian ini, penerapan RAG memberikan peluang untuk meningkatkan kualitas respons sistem berbasis NLP tanpa harus melatih ulang model secara menyeluruh, sehingga lebih efisien dalam hal waktu dan sumber daya. Sebagai contoh, penelitian sebelumnya menunjukkan bahwa LLM dapat digunakan untuk menghasilkan data sintetik dan meningkatkan akurasi tugas spesifik melalui integrasi dengan metode retrieval yang relevan (Said Al Faraby, 2021) [8]. Dengan demikian, metode RAG tidak hanya memperluas cakupan aplikasi LLM tetapi juga menghadirkan solusi yang lebih adaptif dan efisien untuk berbagai kebutuhan pengolahan bahasa alami.

3. METODE PENELITIAN

Metode penelitian ini menggunakan tahapan alur *Research and Development*[14], Dimana sebelum melakukan *processing* terhadap data yang akan digunakan nantinya, dilakukan survei kelengkapan data kepada beberapa mahasiswa teknik informatika untuk mendapatkan pandangan mereka terkait dengan layanan sistem informasi di program studi teknik informatika. Setelah mengumpulkan hasil survei, kemudian data dapat di tindak lanjuti dan diproses untuk mendapatkan informasi lebih spesifik terkait dengan program studi teknik informatika. Selanjutnya dilakukan proses Pembangunan Chatbot dengan alur tahapan seperti pada Gambar 1.[15].

3.1. Alur Penelitian



Gambar 1. Flowchart Alur Penelitian

Flowchart penelitian dimulai dengan identifikasi masalah terkait keterbatasan akses infromasi akademik oleh mahasiswa, setelah itu dilanjutkan dengan melakukan proses survei questioner mengenai kebutuhan apa saja yang menjadi kendala mahasiswa terutama terkait dengan informasi akademik, diikuti oleh *decision* untuk memastikan data yang diperlukan sudah tersedia. Jika data belum lengkap atau data *input* dari hasil survei tidak lengkap dokumennya pada program studi untuk diproses ke tahap selanjutnya, maka dari hasil survei akan dilakukan pemilihan yang lebih berpotensi dan spesifik hanya terhadap informasi akademik. Setelah melihat kebutuhan mahasiswa melalui survei yang dilakukan maka selanjutnya persiapan data yaitu berupa dokumen program studi terkait dengan akademik mahasiswa untuk digunakan pada preprocessing, setelah itu melakukan Pembangunan indeks dan retriever, konfigurasi model LLM, integrasi dengan telegram dan terakhir adalah pengujian dan evaluasi.

3.2. Persiapan Data dan Preprocessing

Tahapan awal adalah mempersiapkan data yang relevan, dalam hal ini adalah dokumen PDF berisi panduan akademik atau informasi terkait program studi Teknik Informatika. Data ini diolah menggunakan *SimpleDirectoryReader* untuk dimuat sebagai kumpulan dokumen[4].

Setiap dokumen dipecah menjadi unit yang lebih kecil (node) menggunakan *SentenceSplitter* dengan parameter ukuran potongan 128 token. Untuk memastikan dokumen terstruktur dengan baik sehingga memudahkan proses penarikan informasi oleh model.

3.3. Pembangunan Indeks dan Retriever

Setelah data dipecah menjadi node, langkah berikutnya adalah membangun *VectorStoreIndex*. Indeks ini memungkinkan penyimpanan representasi vektor dari dokumen sehingga mempermudah proses pencarian informasi berdasarkan tingkat kesamaan (*similarity*)[4]. Penggunaan *retriever* memungkinkan penarikan dokumen relevan berdasarkan kueri pengguna.

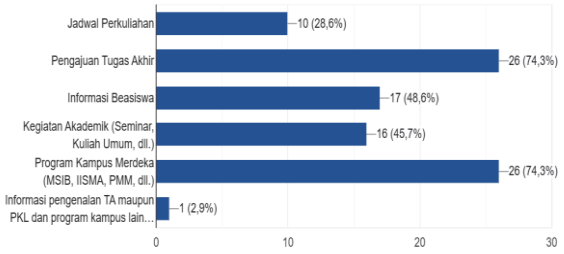
3.4. Konfigurasi Model dan LLM

Pada tahap ini, LLM atau dalam Pembangunan chatbot ini menggunakan model dari HuggingFace yaitu Zephyr-7B-beta dan BAAI/bge-base-en-v1.5 sebagai awal pelatihannya, dikonfigurasi untuk mendukung interaksi dengan data yang telah diproses. Selain itu, model dilengkapi dengan fungsi-fungsi khusus untuk merespons percakapan pengguna secara ringkas dan relevan, serta mampu menangani kueri yang kompleks dengan efisiensi tinggi[4].

Pemilihan model Zephyr-7B-beta dan BAAI/bge-base-en-v1.5 didasarkan pada kemampuan keduanya untuk memenuhi kebutuhan spesifik dalam implementasi *Retrieval-Augmented Generation* (RAG). Zephyr-7B-beta adalah model bahasa yang telah dioptimalkan melalui *Direct Preference Optimization* (DPO), menjadikannya unggul dalam menghasilkan respons kontekstual yang relevan dengan performa tinggi pada benchmark seperti MT-Bench dan AlpacaEval. Model ini tidak memerlukan *finetuning* tambahan untuk aplikasi berbasis percakapan, sehingga dapat langsung diterapkan dalam sistem RAG untuk menghasilkan teks dengan respons yang akurat dan efisien[16]. Sementara itu, BAAI/bge-base-en-v1.5 adalah model embedding yang dirancang untuk tugas-tugas seperti *semantic search* dan *passage retrieval*. Dengan arsitektur berbasis BERT dan kemampuan untuk menangani query panjang maupun pendek, model ini menawarkan distribusi kemiripan yang lebih baik tanpa perlu proses finetuning tambahan. Kombinasi kedua model ini memungkinkan integrasi langsung ke metode RAG tanpa memerlukan biaya dan waktu tambahan untuk finetuning, menjadikannya solusi yang efisien dan efektif untuk aplikasi berbasis NLP yang kompleks[16].

4. HASIL DAN PEMBAHASAN

Setelah sebelumnya melakukan survei dengan penyebaran questioner terhadap beberapa mahasiswa, didapatkan sekitar 40 pertanyaan relevan campuran beberapa angkatan mahasiswa terkait dengan akademik di program studi teknik informatika dan yang mendominasi adalah yang menanyakan terkait dengan informasi akademik. Survei ini dilakukan tentunya hanya untuk mengetahui permasalahan atau kendala mengenai akademik yang sulit didapatkan oleh mahasiswa selama menjalankan perkuliahan. Membahas terkait penerapan *Retrieval-Augmented Generation* yang diintegrasikan ke Chatbot telegram. Metode ini menggunakan data panduan mahasiswa di program studi teknik informatika dalam format pdf yang nanti digunakan untuk melakukan *processing*.



Gambar 2. *Sample* Data Hasil Survei

4.1. Pengujian Terdahulu

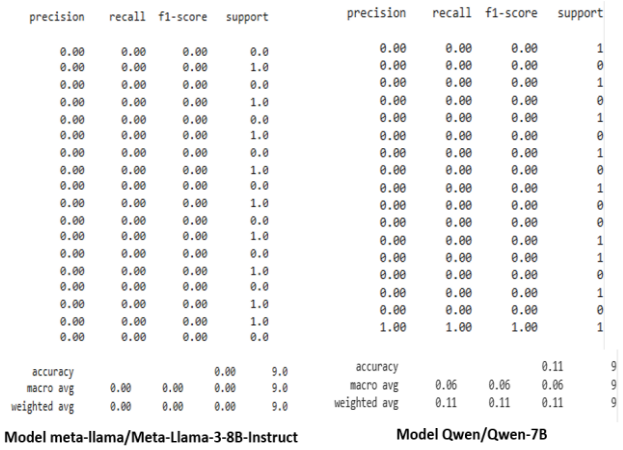
Sebelum melakukan *development* pada model yang akan diterapkan yaitu *Retrieval Augmented Generation*, terlebih dahulu dilakukan pengujian terhadap beberapa model yang akan digunakan sebagai acuan utama LLM untuk proses selanjutnya, dua model dari huggingface seperti meta-llama/Meta-Llama-3-8B-Instruct dan juga Qwen/Qwen-7B untuk memilih mana model yang cocok untuk di gunakan pada proses Pembangunan chatbot.

```
# Fungsi evaluasi
def evaluate(y_true, y_pred):
    # Hitung akurasi
    accuracy = accuracy_score(y_true=y_true, y_pred=y_pred)
    print(f'Accuracy: {accuracy:.3f}')

    # Generate classification report
    class_report = classification_report(y_true=y_true, y_pred=y_pred)
    print('\nClassification Report:')
    print(class_report)

    # Generate confusion matrix
    conf_matrix = confusion_matrix(y_true=y_true, y_pred=y_pred)
    print('\nConfusion Matrix:')
    print(conf_matrix)
```

Kode ini menunjukkan fungsi evaluasi yang digunakan untuk menilai performa model prediksi. Fungsi pada kode menerima dua parameter, yaitu *y_true* (label sebenarnya) dan *y_pred* (label prediksi).

```
from transformers import pipeline

def predict(test, model, tokenizer, temperature=0.7, do_sample=True):

    y_pred = []
    for i in tqdm(range(len(test))):
        prompt = test.iloc[i]["text"]
        pipe = pipeline(task="text-generation",
                        model=model,
                        tokenizer=tokenizer,
                        max_new_tokens=1,
                        temperature=temperature,
                        do_sample=do_sample,
                        )
        result = pipe(prompt)
        answer = result[0]['generated_text'].split("=" )[-1].strip()
        y_pred.append(answer)
    return y_pred

y_pred = predict(X_test, model, tokenizer)
# Evaluasi hasil prediksi
evaluate(X_test['Completion'].tolist(), y_pred)
```

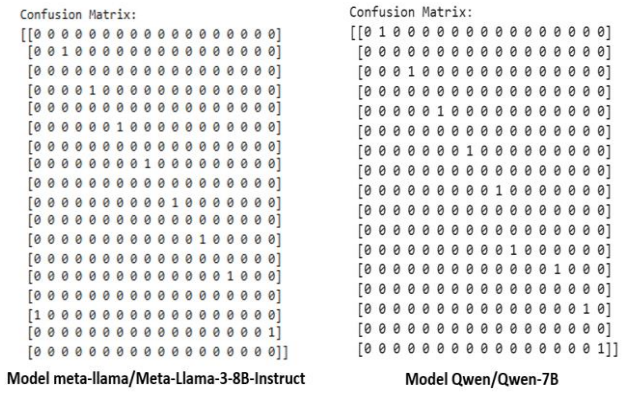
Kode tersebut memiliki fungsi *predict* yang bertujuan menghasilkan prediksi berdasarkan data uji. Setiap baris data uji, *prompt* diambil dari kolom "*text*", kemudian pipeline menghasilkan teks prediksi dengan parameter *max_new_tokens* (jumlah token baru yang dihasilkan) dan hasil prediksi diproses untuk mengambil jawaban yang relevan.



Model meta-llama/Meta-Llama-3-8B-Instruct Model Qwen/Qwen-7B

Gambar 3. Evaluasi Kedua Model *Huggingface*

Pada Gambar 3, Hasil evaluasi menunjukkan bahwa model Meta-llama/Meta-llama-3-8B-Instruct memiliki akurasi 0,00 dengan nilai *precision*, *recall*, dan *f1-score* yang juga 0,00 untuk semua kelas, mengindikasikan performa yang sangat buruk tanpa prediksi yang benar. Sebaliknya, model Qwen/Qwen-7B menunjukkan performa yang sedikit lebih baik dengan akurasi 0,11 serta nilai rata-rata tertimbang (*weighted avg*) untuk *precision, recall,* dan *f1-score* masing-masing sebesar 0,11. Meskipun demikian, performa kedua model masih tergolong rendah dan memiliki kelemahan signifikan untuk nantinya dapat di tindak lanjuti untuk digunakan sebagai model LLM untuk chatbot.



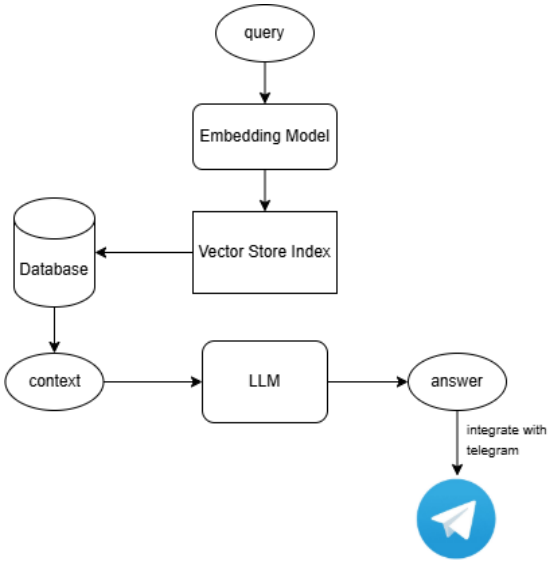
Model meta-llama/Meta-Llama-3-8B-Instruct Model Qwen/Qwen-7B

Gambar 4. *Confusion Matrix* kedua Model *Huggingface*

Hasil evaluasi matriks kebingungan menunjukkan bahwa model Meta-llama/Meta-llama-3-8B-Instruct gagal sepenuhnya, karena tidak ada prediksi yang benar untuk semua kelas, dengan hanya distribusi nol di seluruh matriks. Sementara itu, model Qwen/Qwen-7B menunjukkan sedikit perbaikan dengan beberapa prediksi yang benar, tetapi masih jauh dari memadai

untuk tugas klasifikasi yang kompleks, sebagaimana terlihat dari distribusi prediksi yang salah di sebagian besar kelas. Performa buruk ini menunjukkan bahwa kedua model tidak layak digunakan untuk tahap lebih lanjut tanpa perbaikan signifikan. Finetuning model mungkin diperlukan untuk meningkatkan performa dengan menyesuaikan parameter berdasarkan dataset spesifik. Namun, proses finetuning ini memakan waktu, sumber daya komputasi, dan biaya yang tidak efisien jika dibandingkan dengan menggunakan model lain yang lebih sesuai atau sudah dioptimalkan untuk tugas serupa. Oleh karena itu *Retrieval Augmented Generation* digunakan untuk Solusi yang efektif.

4.2. Arsitektur Model



Gambar 5. Alur Kerja *Retrieval Augmented Generation*

Proses dimulai dengan *query* (pertanyaan) yang diberikan oleh pengguna. *Query* ini diproses oleh *Embedding* Model, yang mengubah teks *query* menjadi representasi vektor. Vektor tersebut kemudian digunakan untuk mencari data relevan dari *VectorStore Index*, yaitu sebuah penyimpanan yang mengorganisasikan dokumen berbasis vektor untuk pencarian cepat. *VectorStoreIndex* ini mendapatkan datanya dari *Database*, yang menjadi sumber utama informasi. Setelah data relevan ditemukan, konteks dari data ini disiapkan dan diberikan kepada LLM untuk diproses. LLM menggunakan konteks tersebut untuk menghasilkan jawaban yang sesuai dengan *query*. Jawaban ini kemudian dikirim ke pengguna melalui integrasi dengan Telegram.

4.3. Preprocessing Dokumen

Agar chatbot dapat menjawab pertanyaan berbasis dokumen, *preprocessing* dilakukan pada file PDF. Langkah ini melibatkan pemecahan dokumen menjadi *node* kecil menggunakan *SentenceSplitter*, dengan *overlap* antar *node* untuk menjaga kesinambungan konteks.

```
from llama_index.core import SimpleDirectoryReader

documents = SimpleDirectoryReader(input_files=["/content/Buku-Panduan-Kurikulum-PSTI-2022.pdf"]).load_data()
parser = SentenceSplitter(chunk_size=128, chunk_overlap=20)
nodes = parser.get_nodes_from_documents(documents)
```

Kode tersebut bertujuan untuk membaca data dari dokumen PDF, memecahnya menjadi potongan teks berdasarkan ukuran *chunk* tertentu dengan *overlap*, dan menghasilkan representasi node untuk pengolahan lebih lanjut. Teknik ini memastikan bahwa setiap *query* dapat menemukan informasi yang relevan secara kontekstual .

4.4. Pembentukan Indeks Pencarian

Hasil *Preprocessing* digunakan untuk membangun *VectorStoreIndex* yang memungkinkan chatbot melakukan pencarian berbasis kesamaan semantik. Setiap node direpresentasikan dalam ruang vektor menggunakan *embedding* dari model *HuggingFace*.

```
from llama_index.embeddings.huggingface import HuggingFaceEmbedding
from llama_index.core import VectorStoreIndex

Settings.embed_model = HuggingFaceEmbedding(model_name="BAAI/bge-base-en-v1.5")
index = VectorStoreIndex(nodes)
retriever = index.as_retriever(similarity_top_k=3)
```

Kode tersebut menginisialisasi sebuah model *embedding* menggunakan *Hugging Face Embedding* dengan nama "BAAI/bge-base-en-v1.5" lalu membuat sebuah indeks vektor (*VectorStoreIndex*) untuk menyimpan representasi vektor dari data teks, dan terakhir membuat sebuah *retriever* untuk mengambil data yang paling relevan berdasarkan kemiripan vektor.

	Skor Relevansi	Konten
0	0.683768	Setiap \npelaksanaan perkuliahan di lingkungan...
1	0.682998	D. Pelaksanaan Perkuliahan \n Perkuliahan di...
2	0.673495	• Pengetahuan dan Teknologi. Memiliki pengetah...

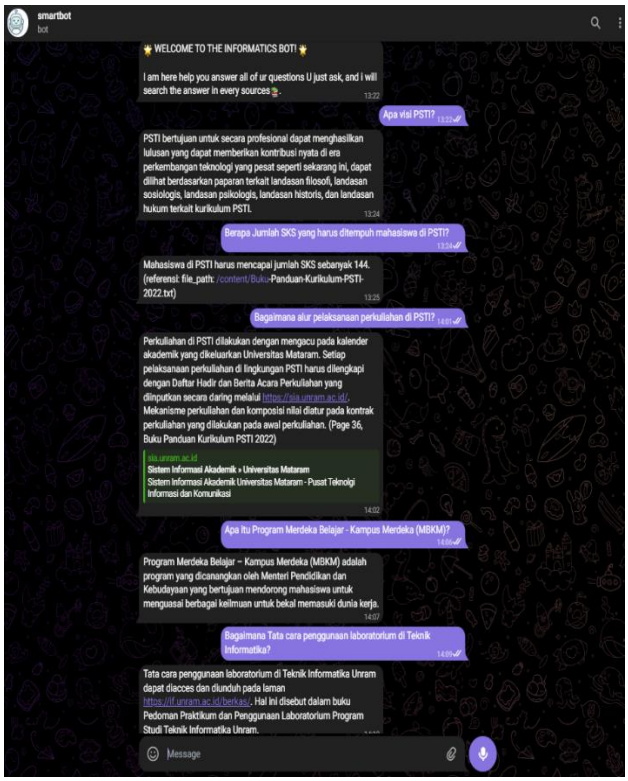
Gambar 6. Hasil *Retrieval*

Menunjukkan hasil *retrieval* berdasarkan relevansi dari *query*. Proses ini menggunakan *vector* yang menghasilkan tiga hasil teratas yang dianggap paling relevan dengan *query*. Setiap hasil dalam table memiliki dua kolom yaitu Relevansi dan Konten, Skor Relevansi menunjukkan tingkat kesesuaian hasil dengan *query*, di

mana nilai yang lebih tinggi menandakan tingkat relevansi yang lebih baik. Dalam contoh ini, skor tertinggi adalah 0.683768, diikuti oleh 0.682998, dan 0.673495. Kolom Konten menampilkan cuplikan dari hasil pencarian, memberikan gambaran singkat mengenai isi informasi yang ditemukan.

4.5. Integrasi dengan Telegram Bot

Integrasi Telegram bot memungkinkan interaksi antara pengguna dengan chatbot, bot menerima *query* dari pengguna, memproses *query* melalui *query engine*, dan mengirimkan hasil pencarian setelah diolah oleh model LLM.



Gambar 7. Tampilan Bot di Telegram

Proses ini dimulai dengan inisialisasi bot menggunakan Pustaka *python-telegram-bot*. Setiap pesan yang diterima diteruskan ke *engine* untuk dilakukan pencarian dan analisis semantik, lalu respons dikembalikan ke pengguna melalui bot.

Salah satu kendala teknis dalam implementasi sistem adalah ketergantungan pada konektivitas *network* yang stabil untuk memastikan respons chatbot berjalan optimal. Selain itu, proses inferensi model LLM membutuhkan sumber daya komputasi yang lumayan untuk dapat *generate* jawaban, sehingga dapat menyebabkan latensi dalam memberikan jawaban jika server tidak dioptimalkan dengan baik. Tantangan lain adalah memastikan relevansi data yang diambil oleh metode RAG agar sesuai dengan konteks pertanyaan pengguna, karena kesalahan dalam *retrieval* dapat menghasilkan jawaban yang kurang akurat. Untuk mengatasi hambatan ini, diperlukan pengujian intensif dan optimasi pada *pipeline* sistem, termasuk penggunaan server GPU untuk mempercepat waktu respons.

4.6. Evaluasi Kinerja

	Query	Waktu Respons (detik)	\
0	Bagaimana alur pelaksanaan perkuliahan di PSTI?	60.099248	
1	alur Pengisian Kartu Rencana Studi di Teknik I...	58.625910	
2	Bagaimana rincian Sistem penilaian yang diguna...	55.454994	

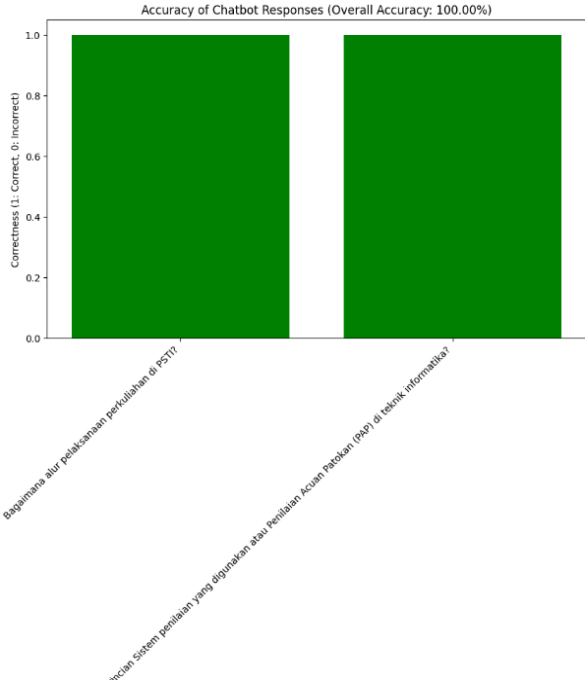
	Jawaban
0	Perkuliahan di PSTI dilakukan dengan mengacu ...
1	I do not have prior knowledge or context abou...
2	Di teknik informatika, sistem penilaian yang ...

Gambar 8. Tabel Evaluasi Performa

Pada Gambar 8 tersebut menampilkan evaluasi performa sistem dalam menjawab berbagai *query* dengan tingkat kompleksitas berbeda. Evaluasi dilakukan dengan mengukur waktu respons untuk setiap *query* dan merekam hasil jawaban. *Query* yang digunakan memiliki tingkat kesulitan berbeda, mulai dari pertanyaan sederhana seperti pelaksanaan perkuliahan hingga yang lebih kompleks terkait detail sistem penilaian di Teknik Informatika. Proses evaluasi dimulai dengan mencatat waktu awal sebelum *query* dieksekusi, dilanjutkan dengan pemrosesan oleh mesin *query*, dan terakhir mencatat waktu selesai. Selisih antara waktu awal dan akhir menentukan durasi waktu respons. Contohnya waktu respons untuk *query* pertama adalah sekitar 60 detik. Data juga menyebutkan bahwa jawaban yang dihasilkan cenderung lebih rinci untuk *query* kompleks. Dan saat menjalankan proses *query engine* dibutuhkan penggunaan server GPU untuk mempercepat proses respons dari pertanyaan yang di-*input* kan.

```python
from fuzzywuzzy import fuzz
import pandas as pd
import matplotlib.pyplot as plt

queries = [
    "Bagaimana alur pelaksanaan perkuliahan di PSTI?",
    "Bagaimana rincian Sistem penilaian yang digunakan atau Penilaian Acuan Patokan (PAP) di teknik informatika?"
]

ground_truth = [
    "Perkuliahan di PSTI dilakukan dengan mengacu pada kalender akademik yang dikeluarkan Universitas Mataram.
    Setiap pelaksanaan perkuliahan di lingkungan PSTI harus dilengkapi dengan Daftar Hadir dan Berita Acara Perkuliahan
    yang diinputkan secara daring melalui https://sia.unram.ac.id/. Mekanisme perkuliahan dan komposisi nilai diatur pada
    kontrak perkuliahan yang dilakukan pada awal perkuliahan. ",
    "Di teknik informatika, sistem penilaian yang digunakan adalah Penilaian Acuan Patokan (PAP)
    dengan derajat penguasaan yang diberikan sebagai berikut: > 85 (nilai A), 80 - < 85 (nilai B+), 75 - < 80 (nilai B),
    dan 70 - < 75 (nilai C+)."

]

accuracy_data = []

for query, correct_answer in zip(queries, ground_truth):
    response = query_engine.query(query)
    bot_answer = response.response

    # Compare bot's answer with the ground truth using fuzzywuzzy
    similarity_score = fuzz.ratio(bot_answer.lower(), correct_answer.lower())
    is_correct = similarity_score >= 80

    accuracy_data.append({
        "Query": query,
        "Bot Answer": bot_answer,
        "Ground Truth": correct_answer,
        "Similarity Score": similarity_score,
        "Correct": is_correct
    })

df_accuracy = pd.DataFrame(accuracy_data)

accuracy_score = df_accuracy["Correct"].mean()

plt.figure(figsize=(10, 6))
plt.bar(df_accuracy["Query"], df_accuracy["Correct"].astype(int), color=['green' if x else 'red' for x in df_accuracy["Correct"]])
plt.xlabel('Queries')
plt.ylabel('Correctness (1: Correct, 0: Incorrect)')
plt.title(f'Accuracy of Chatbot Responses (Overall Accuracy: {accuracy_score * 100:.2f}%)')
plt.xticks(rotation=45, ha='right')
plt.show()
print(df_accuracy)
```

Kode tersebut bertujuan untuk mengevaluasi akurasi respons chatbot dengan membandingkan jawaban model terhadap *ground truth* menggunakan metrik kesamaan (*similarity*) berbasis pustaka "fuzzywuzzy". Prosesnya dimulai dengan mendefinisikan daftar pertanyaan (*queries*) dan jawaban benar (*ground truth*). Setiap respons model dibandingkan dengan jawaban benar menggunakan skor kesamaan berbasis string, di mana skor di atas 80 dianggap benar.



Gambar 9. Evaluasi Akurasi Respons Chatbot

Pada Gambar 9 menampilkan evaluasi akurasi respons chatbot terhadap dua *query* sebagai contoh *sample* dengan pendekatan pengukuran berbasis skor kesamaan. *Query* yang diuji mencakup pertanyaan dengan detail operasional, seperti pelaksanaan perkuliahan dan sistem penilaian di Teknik Informatika. Untuk mengevaluasi akurasi, respons chatbot dibandingkan dengan jawaban yang dianggap sebagai kebenaran dasar (*ground truth*). Skor kesamaan dihitung menggunakan metrik *similarity* dengan skala 0 hingga 100.

```
                                                Queries
                                                  Query  \
0    Bagaimana alur pelaksanaan perkuliahan di PSTI?
1  Bagaimana rincian Sistem penilaian yang diguna...

                                             Bot Answer  \
0   Perkuliahan di PSTI dilakukan dengan mengacu ...
1   Di teknik informatika, sistem penilaian yang ...

                                          Ground Truth  Similarity Score  \
0  Perkuliahan di PSTI dilakukan dengan mengacu p...                95
1  Di teknik informatika, sistem penilaian yang d...               100

   Correct
0     True
1     True
```

Gambar 10. Detail Evaluasi Akurasi Respons

Hasil evaluasi menunjukkan bahwa respons chatbot berhasil memberikan jawaban yang identik dengan *ground truth* untuk kedua *query* yang diuji, sehingga memperoleh skor akurasi keseluruhan sebesar 100%. Grafik batang vertikal yang ditampilkan mengilustrasikan keberhasilan ini, di mana semua respons dianggap benar (ditandai dengan batang hijau penuh pada setiap *query*).

Penelitian sebelumnya menyoroti bahwa integrasi sistem informasi akademik dengan Telegram Bot mampu meningkatkan efisiensi layanan informasi bagi mahasiswa[1]. Dalam konteks penelitian ini, pendekatan tersebut diperluas dengan mengintegrasikan metode *Retrieval-Augmented Generation* (RAG) dan *Large Language Models* (LLM), yang memungkinkan chatbot tidak hanya memberikan respons yang lebih kontekstual dan relevan, tetapi juga mendukung kebutuhan secara lebih efektif. Dengan memanfaatkan RAG untuk menemukan dokumen relevan dan LLM untuk memahami serta menghasilkan jawaban berdasarkan konteks, penelitian ini memberikan kontribusi signifikan terhadap pengembangan chatbot berbasis *Artificial Intelligence* dalam domain pendidikan[17].

5. KESIMPULAN DAN SARAN

Pengembangan chatbot berbasis *Retrieval Augmented Generation* (RAG) dengan model dari HuggingFace dapat memberikan hasil yang sangat baik dalam memproses data menggunakan retriever menjadi terstruktur. Hasil evaluasi menunjukkan tingkat akurasi respons chatbot mencapai 100% untuk dua query uji sebagai *sample*, dengan skor kesamaan sempurna terhadap jawaban yang dianggap benar sesuai data (*ground truth*). Selain itu, waktu respons rata-rata adalah sekitar 60 detik, bahkan untuk pertanyaan kompleks, menunjukkan efisiensi sistem. Keberhasilan ini didukung oleh tiga komponen utama yaitu preprocessing data menggunakan teknik pembagian menjadi node kecil dengan overlap untuk menjaga kesinambungan infromasi, pembentukan indeks pencarian berbasis vektor yang memungkinkan pencarian berdasarkan kesamaan *semantic*, dan pemanfaatan *Large Lanauage Model* untuk menghasilkan jawaban yang relevan dan kontekstual.

Implikasi dari penelitian ini adalah bahwa metode RAG dapat diadopsi secara lebih luas dalam pengembangan chatbot akademik untuk meningkatkan aksesibilitas informasi bagi mahasiswa. Namun, beberapa aspek masih perlu ditingkatkan, seperti kemampuan memahami beragam jenis pertanyaan atau multi-topik yang terkadang menghasilkan jawaban yang kurang optimal.

Sistem berbasis *Retrieval Augmented Generation* dan *Large Language Model* ini dapat menjadi model referensi untuk mengembangkan chatbot lain yang mampu mengakses dan memproses informasi dari berbagai jenis data dokumen secara efisien. Dengan peningkatan lebih lanjut pada aspek pemahaman konteks dan personalisasi atau kemampuan sistem untuk menyesuaikan respons berdasarkan data dan preferensi individu pengguna dengan mengumpulkan informasi dari interaksi sebelumnya, seperti Riwayat pertanyaan, pola penggunaan, atau preferensi spesifik pengguna. Chatbot semacam ini berpotensi menjadi sistem yang lebih cerdas dan adaptif, tidak hanya untuk aspek Pendidikan tetapi juga untuk berbagai aspek lainnya seperti layanan pelanggan atau manajemen informasi dan lainnya.

UCAPAN TERIMA KASIH

Puji syukur saya ucapkan atas kemudahan dan Rahmat Tuhan yang Maha Esa karena atas berkat-nya saya dapat menyelesaikan *Paper* dalam Tugas Akhir Kuliah di Program Studi Teknik Informatika. Dengan judul yang saya angkat yaitu "*Chatbot Retrieval Augmemted Generation berbasis Large Language Model*". *Paper* ini memperlihatkan pemahaman yang mendalam tentang Analisa dan bagaimana penerapannya dapat mencari akurasi dari metode yang diterapkan untuk pengembangan chatbot.

Saya ingin mengucapkan terima kasih khusus kepada dosen pembimbing tugas akhir saya yaitu Bapak Prof.Dr.Eng I Gede Pasek Suta Wijaya S.T., M.T. dan Bapak Ramaditia Dwiyansaputra, S.T., M.Eng. yang telah memberikan arahan, bimbingan, dan wawasan berharga sepanjang penyusunan dan pengembangan tugas akhir.

DAFTAR PUSTAKA

[1] A. Zubaidi, A. Z. Mardiansyah, W. Wedashwara, and A. H. Jatmika, "Integrasi Sistem Informasi Akademik Dan Bot Telegram Sebagai Media Pengaksesan Informasi Di Universitas Mataram," *Jtika*, vol. 3, no. 2, pp. 253–268, 2021, [Online]. Available:http://jtika.if.unram.ac.id/index.php/JTIKA/

[2] Nuzul Hikmah, Dyah Ariyanti, and Ferry Agus Pratama, "Implementasi Chatbot Sebagai Virtual Assistant di Universitas Panca Marga Probolinggo menggunakan Metode TF-IDF," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 4, no. 2, pp. 133–148, 2022, doi: 10.35746/jtim.v4i2.225.

[3] D. Radhian, I. Afrianto, P. Studi, and T. I. Komputer, "Pembangunan Aplikasi Chatbot Sebagai Media Pencarian Informasi Dalam Bidang Peternakan," *Progr. Stud. Tek. Inform. Komput. Indones.*, 2019.

[4] H. Tohir, N. Merlina, and M. Haris, "UTILIZING RETRIEVAL-AUGMENTED GENERATION IN LARGE LANGUAGE MODELS TO ENHANCE INDONESIAN LANGUAGE NLP," vol. 10, no. 2, pp. 352–360, 2024, doi: 10.33480/jitk.v10i2.5916.INTRODUCTION.

[5] H. Sultan, U. Kristen, I. Toraja, and U. N. Madano, "Sistem Pembelajaran Berbasis Chatbot Untuk Pelatihan Online : Meningkatkan Efektivitas Pembelajaran," vol. 3, no. 6, pp. 7146–7150, 2024.

[6] M. Al-Amin *et al.*, "History of generative Artificial Intelligence (AI) chatbots: past, present, and future development," no. February, 2024, doi:

10.48550/arXiv.2402.05122.

[7] R. C. Noor Santi, "Perancangan Interaksi Pengguna (User Interaction Design) Menggunakan Metode Prototyping," *J. Tek. Inform.*, vol. 9, no. 2, pp. 108–113, 2018, doi: 10.15408/jti.v9i2.5599.

[8] M. A. Nasution *et al.*, "Implementasi NLP Dalam Pembuatan Chatbot Customer Service Publisher Jurnal Studi Kasus LARISMA," *SAINTEK J. Sains, Teknol. Komput.*, vol. 1, no. 1, pp. 13–17, 2024.

[9] Q. Rizqie, N. Afifah, and A. Bardadi, "NetPLG Journal of Network and Computer Applications Eksplorasi Penggunaan Large Language Model (LLM) dalam Pembangunan Permainan Minesweeper dengan Python Programming," *J. Netw. Comput.*, vol. 2, no. 3, pp. 63–70, 2023, [Online]. Available: https://jurnal.netplg.com/jnca

[10] E. Eldi and H. Syaputra, "Implementasi Chatbot Untuk Mendukung Sistem Informasi Pada Rumah Sakit Muhamadiyah Palembang," *J. Nas. Ilmu Komput.*, vol. 1, no. 3, pp. 139–148, 2020, doi: 10.47747/jurnalnik.v1i3.160.

[11] D. Rahayu, M. Mukrodin, and R. Hariyono, "Penerapan Artificial Intelligence Dalam Aplikasi Chatbot Sebagai Helpdesk Objek Wisata Dengan Permodelan Simple Reflex-Agent (Studi Kasus : Desa Karangbenda)," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 9, no. 1, pp. 7–21, 2020, doi: 10.30591/smartcomp.v9i1.1813.

[12] G. C. Lenardo, Herianto, and Y. Irawan, "Pemanfaatan Bot Telegram sebagai Media Informasi Akademik di STMIK Hang Tuah Pekanbaru," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 1, no. 4, pp. 351–357, 2020, doi: 10.35746/jtim.v1i4.59.

[13] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning," *Karya Ilm. Mhs. Bertauhid (KARIMAH TAUHID)*, vol. 2, no. 1, pp. 1–6, 2023.

[14] A. A. Chandra, V. Nathaniel, F. R. Satura, and F. D. Adhinata, "Pengembangan Chatbot Informasi Mahasiswa Berbasis Telegram dengan Metode Natural Language Processing," *J. ICTEE*, vol. 3, no. 1, p. 20, 2022, doi: 10.33365/jictee.v3i1.1886.

[15] P. K. Laksana Utama, "Bot Chat : Customer Relation dengan Teknologi Artificial Intelligence," *Widya Duta J. Ilm. Ilmu Agama dan Ilmu Sos. Budaya*, vol. 13, no. 2, p. 81, 2018, doi: 10.25078/wd.v13i2.692.

[16] M. Weber and M. Reichardt, "Evaluation is all you need. Prompting Generative Large Language Models for Annotation Tasks in the Social Sciences. A Primer using Open Models," 2023, [Online]. Available: http://arxiv.org/abs/2401.00284

[17] Muhammad Irfan Syah, Nazruddin Safaat Harahap, Novriyanto, and Suwanto Sanjaya, "Penerapan Retrieval Augemented Generation Menggunakan Langchain Dalam Pengembangan Sistem Tanya Jawab Hadis Berbasis Web," *Zo. J. Sist. Inf.*, vol. 6, no. 2, pp. 370–379, 2024, doi: 10.31849/zn.v6i2.19940.