Abstractive Text Summarization Berita Bahasa Indonesia Menggunakan Retrieval-Augmented Generation

Antonius Sakti Wiradinata [1] Viny Christanti Mawardi [2]

[1] [2] Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara
Jl. Letjen S. Parman No.1, Jakarta
email : antonius.535210070@stu.untar.ac.id [1], vinym@fti.untar.ac.id [2]

ABSTRACT

This research discusses the application of Abstractive Text Summarization (ATS) to Indonesian language news using the Retrieval-Augmented Generation (RAG) method. Increased access to news through various digital platforms often causes users to have difficulty identifying relevant information among the large amount of news available. RAG integrates retrieval and generation techniques to produce coherent and informative news summaries. In this research, news from the CNN and CNBC sites was collected via web scraping to form a dataset. The data is processed through several stages, including preprocessing, embedding, information retrieval, and summary generation. Summary quality evaluation was carried out using the ROUGE metric, where the test results show that this system has good performance in the precision aspect, with a ROUGE-1 Precision value of 0.7432 and ROUGE-2 Precision of 0.6174. However, a lower ROUGE Recall value indicates that there is important information that is not fully included in the summary. These results indicate that the RAG method in ATS is effective in helping users obtain core information concisely, but there needs to be improvement in capturing the entire news context

Key words

Abstractive Text Summarization, embedding, information, Retrieval-Augmented Generation, ROUGE, web scraping

1. Pendahuluan

Perkembangan teknologi menunjukkan perubahan dalam cara mengakses dan mengonsumsi informasi, terutama dalam bentuk berita. Dengan kemajuan teknologi, berita dapat diakses dengan mudah melalui berbagai platform digital seperti situs web, aplikasi berita, dan media sosial. Kemudahan ini telah mengakibatkan peningkatan volume informasi yang tersedia bagi publik, dengan ribuan berita yang diunggah setiap hari. Di satu sisi, hal ini memberikan keuntungan bagi pengguna untuk memperoleh berita terkini secara cepat. Namun di sisi lain, pengguna sering kali dihadapkan pada masalah, di mana banyaknya berita yang tersedia membuat sulit bagi pengguna untuk menyaring dan mengidentifikasi informasi yang paling relevan dalam waktu singkat [1].

Masalah ini tidak hanya mengurangi efisiensi dalam mendapatkan informasi penting tetapi juga dapat menyebabkan pengguna merasa kewalahan dan akhirnya mengabaikan informasi yang sebenarnya penting. Oleh karena itu, diperlukan solusi yang dapat membantu pengguna dalam menyaring dan meringkas informasi secara otomatis, sehingga esensi dari berita dapat diperoleh tanpa harus membaca keseluruhan teks.

Salah satu solusi yang diusulkan untuk mengatasi masalah ini adalah dengan menggunakan Abstractive Text Summarization (ATS), sebuah teknik yang memungkinkan proses untuk meringkas teks secara otomatis dengan menyusun ulang dan memparafrasekan isi berita. Melalui abstractive summarization, pengguna dapat memperoleh ringkasan yang tidak hanya berisi kalimat-kalimat penting dari teks asli tetapi juga disusun secara koheren dan dengan cara yang lebih alami, mirip dengan cara manusia menyusun ringkasan.

Sebagian besar metode yang digunakan dalam ringkasan teks saat ini adalah extractive summarization, yang hanya memilih kalimat-kalimat penting dari teks asli dan menyusunnya kembali menjadi ringkasan. Meskipun proses ini lebih sederhana dan cepat, hasil ringkasan yang dihasilkan seringkali kurang berhubungan, karena kalimat-kalimat yang dipilih tidak selalu terhubung secara logis. Untuk mengatasi keterbatasan tersebut, proses abstractive summarization menjadi pilihan yang lebih baik, karena mampu menghasilkan ringkasan yang lebih alami dan informatif [2].

Dalam beberapa tahun terakhir, salah satu metode terbaru dalam abstractive summarization adalah Retrieval-Augmented Generation (RAG). RAG menggabungkan kemampuan retrieval, yaitu pencarian informasi relevan dari sejumlah besar teks, dengan kemampuan generasi teks, yaitu menghasilkan ringkasan yang koheren dan informatif. Dengan proses ini, RAG tidak hanya dapat menghasilkan ringkasan dari teks yang diberikan, tetapi juga dapat memperkaya ringkasan

tersebut dengan informasi tambahan yang relevan dari sumber eksternal.

Penerapan RAG dalam abstractive summarization untuk berita berbahasa Indonesia masih jarang dilakukan, dan juga memiliki potensi manfaatnya yang besar. Proses ini dapat membantu pengguna dalam mendapatkan ringkasan berita yang relevan secara cepat dan efisien, terutama di tengah maraknya berita yang beredar saat ini. Penggunaan RAG diharapkan mampu menghasilkan ringkasan yang lebih akurat dan informatif dibandingkan dengan metode sebelumnya, sehingga dapat memberikan solusi yang efektif untuk mengatasi masalah overload informasi dalam konsumsi berita.

Seiring dengan bertambahnya jumlah berita yang tersedia, metode konvensional dalam menyaring dan meringkas informasi mulai menunjukkan keterbatasan. Metode extractive summarization, yang sering digunakan, hanya dapat mengidentifikasi dan menyusun ulang kalimat-kalimat penting dari teks asli tanpa memahami konteks atau menyajikan informasi dengan cara yang koheren dan alami. Hal ini sering kali mengakibatkan ringkasan yang tidak sepenuhnya representatif terhadap keseluruhan isi berita dan dapat kehilangan alur serta detail penting yang membentuk konteks berita [2].

Di antara teknik extractive summarization yang umum digunakan adalah pendekatan berbasis frekuensi, seperti Term Frequency-Inverse Document Frequency (TF-IDF), yang menghitung frekuensi kemunculan kata dalam teks untuk menentukan kalimat yang paling penting. Selain itu, metode berbasis graf, seperti TextRank, membangun graf di mana node mewakili kalimat dan edge mewakili kemiripan antara kalimat. Kalimat dengan ranking tertinggi dalam graf ini kemudian dipilih untuk dimasukkan dalam ringkasan [2].

Meskipun teknik-teknik ini telah digunakan secara luas, proses itu juga memiliki keterbatasan dalam memahami konteks dan menyajikan informasi dengan cara yang koheren dan alami. Ringkasan yang dihasilkan sering kali tidak sepenuhnya representatif terhadap keseluruhan isi berita dan dapat kehilangan alur serta detail penting yang membentuk konteks berita.

Untuk mengatasi masalah ini, proses abstractive summarization yang lebih canggih seperti Retrieval-Augmented Generation (RAG) muncul sebagai solusi potensial. Dengan kemampuan RAG untuk menggabungkan teknik retrieval dan generation, proses ini dapat melakukan pencarian informasi relevan secara lebih mendalam dan menyajikan ringkasan yang tidak hanya lebih informatif tetapi juga lebih terstruktur [3].

RAG memungkinkan untuk mengakses dan memanfaatkan informasi dari berbagai sumber eksternal, meningkatkan kualitas dan keberagaman informasi yang disajikan dalam ringkasan. proses ini tidak hanya memberikan ringkasan yang lebih koheren dan menyeluruh tetapi juga mampu memberikan nilai tambah dengan memasukkan informasi relevan yang mungkin tidak langsung tersedia dalam teks asli [3].

Dalam konteks berita berbahasa Indonesia, Penerapan RAG diharapkan dapat meningkatkan kualitas ringkasan berita, membuatnya lebih akurat, informatif, dan relevan bagi pengguna. Teknologi ini bertujuan untuk membantu pengguna dalam mengelola informasi yang akan diterima, serta memberikan pemahaman yang lebih baik tentang konten berita.

2. Landasan Teori

2.1 Rancangan Sistem

Sistem yang dirancang bertujuan untuk menghasilkan ringkasan teks berita berbahasa Indonesia secara abstraktif menggunakan proses Retrieval-Augmented Generation (RAG). Proses ini mencakup beberapa tahapan utama, yaitu pengumpulan data, preprocessing, embedding, pencarian informasi relevan, generasi ringkasan, serta evaluasi kualitas ringkasan menggunakan metrik ROUGE.

Pengumpulan Data dimulai dengan menggunakan UiPath untuk melakukan web scraping pada situs berita CNN dan CNBC yang diambil dari 500 link berita dari website CNN dan 500 dari link website CNBC, dengan total 1000 link berita. Perangkat lunak ini digunakan untuk mengumpulkan artikel berita dan menyimpannya dalam file Excel yang berisi informasi seperti judul, URL, konten, dan kategori berita. Data yang dikumpulkan menjadi dasar bagi sistem dalam menghasilkan ringkasan [8].

Preprocessing dan Ekstraksi Teks adalah langkah berikutnya setelah data terkumpul. Program membaca file Excel dan melakukan permintaan HTTP untuk mengambil konten dari URL yang terdaftar. Teks yang diperoleh kemudian dibersihkan dari elemen yang tidak diperlukan seperti iklan, tautan, dan elemen multimedia. Teks juga diolah untuk menghilangkan spasi berlebihan dan karakter-karakter yang tidak relevan. Setelah dibersihkan, teks dipecah menjadi bagian-bagian yang lebih kecil melalui proses chunking untuk memudahkan proses embedding dan generasi ringkasan [9].

Chunking adalah teknik yang membagi teks menjadi segmen-segmen yang lebih kecil, sehingga masing-masing segmen dapat diproses dengan lebih efisien oleh model RAG. Dengan membagi teks menjadi chunk yang lebih kecil, sistem dapat lebih mudah menangani dan menganalisis informasi, serta meningkatkan akurasi dalam menghasilkan ringkasan yang relevan. Proses ini juga memungkinkan model untuk lebih baik memahami konteks dalam setiap bagian teks, sehingga informasi

yang dihasilkan dalam ringkasan menjadi lebih koheren dan informatif.

Embedding Teks dengan RAG merupakan tahapan penting dalam sistem ini. Teks yang telah diproses diubah menjadi representasi vektor menggunakan model RAG. Model ini menggabungkan teknik retrieval dan generation. Pada tahap retrieval, sistem mencari informasi relevan dari koleksi teks berdasarkan query atau bagian teks yang sedang diproses. Teknik generation kemudian digunakan untuk menghasilkan ringkasan dengan menyusun ulang dan memparafrasekan informasi relevan. RAG Tokenizer dan RAG Sequence For Generation digunakan untuk mengubah teks menjadi embedding dan menghasilkan ringkasan [10].

Pencarian Informasi Relevan dilakukan setelah embedding. Sistem menggunakan kemampuan retrieval dari RAG untuk menemukan informasi yang paling relevan dari koleksi teks yang ada. Dalam sistem ini, mode retrieval menggunakan Dense Passage Retrieval (DPR) untuk memaksimalkan pencarian informasi yang relevan. DPR adalah teknik retrieval berbasis neural yang menggunakan model dual-encoder untuk membuat representasi teks dalam bentuk vektor. Model ini mengubah query dan teks ke dalam representasi vektor terpisah, memungkinkan sistem untuk menghitung cosine similarity antara embedding query dan embedding teks dalam dataset. Cosine similarity digunakan untuk mengidentifikasi teks yang paling relevan berdasarkan nilai similarity tertinggi, dan teks yang relevan ini digunakan untuk membangun konteks bagi generasi ringkasan.

Generasi Ringkasan dengan RAG dan Pegasus merupakan tahapan generatif. Setelah informasi relevan diambil, model Pegasus digunakan sebagai komponen generatif dalam RAG untuk menghasilkan ringkasan. Pegasus, yang telah dilatih khusus untuk tugas summarization, memiliki kemampuan unik dalam menyusun ulang dan memparafrasekan informasi sehingga menghasilkan ringkasan yang lebih koheren dan informatif. Dibandingkan dengan metode ekstraktif yang hanya memilih kalimat penting dari teks asli, penggunaan Pegasus memungkinkan sistem untuk menghasilkan ringkasan abstraktif yang lebih alami, dengan mempertimbangkan konteks yang lebih luas dari informasi yang tersedia. Pegasus membantu menghasilkan ringkasan yang lebih tersusun dengan menjaga alur dan relevansi informasi secara lebih baik.

Evaluasi kualitas ringkasan dilakukan dengan menggunakan metrik ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROUGE mengukur kesamaan n-gram, yaitu kesamaan kata individu antara ringkasan yang dihasilkan dan referensi. ROUGE-2 mengukur kesamaan dua kata berturut-turut (bigram), sementara ROUGE-L mengukur panjang longest common subsequence (LCS), yang menunjukkan seberapa baik urutan kata dalam ringkasan sesuai dengan referensi. ROUGE-S mengukur kesamaan skip-bigram, yaitu dua kata dengan satu atau lebih jarak antar kata. Evaluasi ini memastikan bahwa ringkasan yang dihasilkan mencakup informasi penting dan relevan serta memiliki kesamaan struktur dan kata dengan ringkasan referensi. Setelah evaluasi, ringkasan disimpan dalam format .docx yang dapat diunduh oleh pengguna, menyediakan solusi efisien untuk mendapatkan ringkasan berita yang relevan dan berguna.

2.2 Berita Online

Berita online adalah penyajian informasi melalui platform digital yang dapat diakses langsung melalui internet. Berita online, seperti yang disediakan oleh situs web CNN dan CNBC, menawarkan berbagai macam informasi, mulai dari politik, ekonomi, hingga hiburan, dalam bentuk teks, gambar, video, dan infografis yang dapat diakses kapan saja melalui perangkat elektronik.

Keunggulan utama dari berita online adalah kecepatannya dalam memperbarui informasi secara real-time, memungkinkan untuk selalu mendapatkan berita terkini. Selain itu, berita online juga sering dilengkapi dengan fitur interaktif yang memungkinkan pembaca untuk berpartisipasi aktif dalam diskusi melalui kolom komentar atau berbagi berita di media sosial. Dalam penelitian ini, berita online digunakan sebagai sumber data utama untuk peringkasan teks berita bahasa indonesia, mengingat kemudahan akses dan keandalan informasi yang disajikan secara real-time, sehingga data yang digunakan selalu relevan dengan situasi terkini.

2.3 Web Scraping

Web scraping adalah teknik pengumpulan data dari situs web secara otomatis. Dalam proses ini, web scraping digunakan untuk mengumpulkan artikel berita dari website berita CNN Indonesia https://www.cnnindonesia.com/nasional/indeks/3 dan CNBC Indonesia https://www.cnbcindonesia.com/indeks berdasarkan mengambil dari indeks berita supaya dapat semua kategori beritanya dengan menggunakan UiPath dan BeautifulSoup. UiPath digunakan untuk mengotomatisasi proses pengumpulan data, sedangkan BeautifulSoup membantu dalam parsing HTML dan ekstraksi informasi yang relevan dari halaman web. Data yang terkumpul kemudian disimpan dalam bentuk CSV yang siap digunakan untuk proses summarization. Kombinasi dari kedua alat ini memastikan bahwa data yang diperoleh bersih, terstruktur, dan siap diproses lebih lanjut sesuai kebutuhan.

2.4 Abstractive Text Summarization

Abstractive summarization adalah teknik yang menghasilkan ringkasan dengan menyusun ulang informasi dari teks sumber dan menghasilkan kalimat baru. Alih-alih hanya menyeleksi kalimat yang sudah

ada, sistem abstraktif menciptakan kalimat baru yang lebih ringkas, dengan tetap mempertahankan esensi informasi yang terkandung di dalam dokumen asli. Pendekatan ini meniru cara manusia dalam merangkum teks: mencerna informasi, memahami konteks, lalu menyajikan ulang informasi dengan gaya bahasa yang lebih ringkas dan mudah dipahami.

Proses ini lebih kompleks karena sistem harus memahami makna, hubungan antar kalimat, serta konteks secara keseluruhan. Selain itu, sistem abstraktif harus dapat menyusun ulang informasi secara akurat tanpa mengubah esensi informasi. Pendekatan ini sangat mengandalkan teknik pembelajaran mesin modern seperti deep learning, terutama model Transformer yang mampu menghasilkan teks yang lebih alami dan sesuai konteks.

Sistem abstraktif yang efektif memerlukan arsitektur yang canggih, seperti Transformer-based models (misalnya Pegasus, GPT, RAG, dan BART) yang dapat memproses konteks dalam skala besar dan menyusun ulang informasi dari teks yang lebih panjang dan kompleks. Model Retrieval-Augmented Generation (RAG) juga memberikan peningkatan signifikan, dengan menggabungkan pencarian informasi dengan pembuatan teks yang menghasilkan ringkasan lebih tepat sasaran dan relevan.

Extractive text summarization adalah teknik yang menghasilkan ringkasan dengan cara memilih dan mengekstrak kalimat-kalimat atau frasa-frasa penting dari teks sumber tanpa mengubah struktur atau kata-kata yang diambil. Pendekatan ini berfokus pada identifikasi bagian-bagian teks yang dianggap paling relevan untuk disusun menjadi ringkasan. Metode ini biasanya menggunakan teknik berbasis statistik, seperti tf-idf (term frequency-inverse document frequency), yang menghitung seberapa sering sebuah kata muncul dalam dokumen dibandingkan dengan frekuensinya dalam keseluruhan kumpulan dokumen. Teknik ini juga sering dipadukan dengan algoritma TextRank, yang mirip dengan cara kerja algoritma pencarian, di mana kalimat-kalimat diberi bobot berdasarkan konektivitas dan pentingnya hubungan antar kalimat dalam teks.

Meskipun metode ini sederhana dan cepat diterapkan, ringkasan yang dihasilkan sering kali kurang kohesif atau terasa terfragmentasi. Ini karena kalimat yang dipilih dari teks asli tidak selalu terhubung secara alami ketika disusun kembali. Alhasil, hasil ringkasan dari extractive summarization dapat terasa kurang halus dan tidak mengalir seperti ringkasan yang ditulis oleh manusia. Meskipun demikian, metode ini sangat efektif untuk menghasilkan ringkasan dalam waktu singkat, terutama untuk dokumen panjang di mana efisiensi pemrosesan menjadi prioritas.

2.5 Large Language Model (LLM)

Large Language Model (LLM) merupakan suatu jenis model pembelajaran mesin yang dibangun untuk memahami, menganalisis, dan menghasilkan teks dalam bahasa alami. Model ini menggunakan arsitektur jaringan saraf yang kompleks, khususnya yang berbasis Transformer, yang memungkinkan pemrosesan dan pengolahan data dalam skala besar. LLM dilatih pada korpus data teks yang sangat luas, mencakup berbagai sumber informasi seperti buku, artikel, dan dokumen daring, sehingga dapat menangkap nuansa, konteks, serta hubungan antar kata dalam kalimat dengan lebih baik. Melalui proses pelatihan yang mendalam, LLM mampu menghasilkan teks yang koheren dan relevan, menjawab pertanyaan, dan melakukan berbagai tugas pemrosesan bahasa alami (Natural Language Processing/NLP) yang membutuhkan pemahaman konteks yang mendalam.

Dalam konteks pengembangan Abtractive Text Summarization, Large Language Models (LLM) dapat diintegrasikan dengan Retrieval-Augmented Generation (RAG) untuk meningkatkan kualitas dan relevansi output yang dihasilkan. Proses RAG memadukan kemampuan pencarian informasi dari sumber eksternal dengan kemampuan generatif LLM, sehingga tidak hanya mengandalkan pengetahuan yang diperoleh dari pelatihan sebelumnya. Pada tahap retrieval, sistem mencari dan mengekstraksi informasi yang relevan berdasarkan query pengguna, yang selanjutnya digunakan oleh LLM untuk menghasilkan teks yang lebih informatif dan kontekstual.

Dengan demikian, LLM dalam RAG tidak hanya berfungsi sebagai generator teks, tetapi juga sebagai alat yang mampu memberikan respons yang lebih tepat dan terkini, menjawab kebutuhan pengguna dengan cara yang lebih efektif dan responsif terhadap informasi terbaru.

2.6 Transformers

Transformers adalah arsitektur deep learning yang menggunakan mekanisme self-attention untuk memproses dan memahami sekuens teks dengan lebih efisien tanpa harus memperhatikan urutan input secara eksplisit. Arsitektur ini terdiri dari dua komponen utama: encoder dan decoder. Encoder berfungsi untuk memproses sekuens input dan menghasilkan representasi mendalam dari teks, sementara decoder menggunakan representasi tersebut untuk menghasilkan output, seperti pada tugas terjemahan atau peringkasan teks.

Setiap blok dalam encoder mengandung dua komponen penting: multi-head self-attention mechanism dan feed-forward neural network. Mekanisme self-attention memungkinkan model untuk fokus pada kata-kata penting dalam sekuens input dengan menghitung attention score antara setiap kata dalam kalimat. Karena proses ini dilakukan secara paralel, transformer mampu menangani dependensi jarak jauh antar kata dengan lebih efisien dibandingkan dengan model seperti RNN. Selain

itu, penggunaan multi-head attention memungkinkan model untuk memahami berbagai konteks kata secara bersamaan melalui beberapa fungsi perhatian yang berbeda diterapkan pada sekuens yang sama.

Selain mekanisme self-attention, setiap blok encoder juga dilengkapi dengan lapisan feed-forward yang terhubung penuh, yang diaplikasikan secara terpisah pada setiap token dalam sekuens. Output dari mekanisme attention dan feed-forward ini digabungkan menggunakan residual connections dan layer normalization, yang membantu meningkatkan stabilitas gradien selama pelatihan dan mempercepat proses konvergensi model.

Bagian decoder memiliki arsitektur serupa dengan encoder, tetapi terdapat beberapa modifikasi. Salah satu modifikasi utama adalah adanya masked multi-head attention, yang mencegah model untuk melihat token-token yang belum dihasilkan selama proses pelatihan autoregressive. Setelah itu, decoder melakukan self-attention pada output dari encoder, dan secara bertahap menghasilkan prediksi teks baru berdasarkan input yang telah diproses.

Keunggulan utama dari transformer terletak pada kemampuannya memproses sekuens panjang secara paralel, sehingga membuatnya sangat efisien. Selain itu, model ini sangat fleksibel dan telah menjadi dasar bagi model-model NLP canggih seperti BERT, GPT, dan T5, yang unggul dalam berbagai tugas pemrosesan bahasa alami.

Gambar 1 Arsitektur Encoder-Decoder dengan Masking pada Model Transformer

2.7 Teks Embedding

Embedding teks adalah proses mengubah teks menjadi representasi vektor numerik yang dapat dipahami oleh model machine learning. Representasi ini memungkinkan sistem untuk melakukan pencarian dan pemrosesan teks dengan lebih efektif. Dalam konteks RAG, embedding teks digunakan untuk mencocokkan teks yang sedang diproses dengan teks dalam dataset, memastikan bahwa hanya informasi yang paling relevan yang digunakan dalam generasi ringkasan [11].

(1)

alam proses embedding teks, setiap kata, frasa, atau dokumen yang menjadi input diubah menjadi vektor numerik melalui fungsi embedding Fungsi embedding ini memetakan teks input ke dalam representasi vektor berdimensi tetap yang dapat dipahami oleh model machine learning. Vektor embedding ini kemudian digunakan untuk berbagai tujuan seperti pencarian informasi atau pemrosesan lebih lanjut dalam model, termasuk dalam konteks Retrieval-Augmented Generation (RAG) [12].

2.8 Retrieval-Augmented Generation (RAG)

RAG adalah proses yang menggabungkan dua komponen utama retrieval (pencarian informasi) dan generation (generasi teks). Dalam RAG, sistem pertama-tama mencari informasi relevan dari kumpulan teks yang besar berdasarkan query atau bagian teks yang sedang diproses. Setelah itu, informasi yang ditemukan digunakan sebagai dasar untuk menghasilkan ringkasan atau teks baru. RAG menawarkan keunggulan dalam menghasilkan ringkasan yang tidak hanya relevan tetapi juga kaya akan informasi, karena mampu menarik informasi dari sumber eksternal [12].

Pada tahap pencarian informasi (retrieval) dalam Retrieval-Augmented Generation (RAG), berfungsi untuk menemukan dokumen atau bagian teks yang relevan dari kumpulan data yang besar berdasarkan query yang diberikan. Proses ini dimulai dengan mengubah query menjadi vektor representasi menggunakan model embedding yang digunakan dalam RAG. Vektor ini kemudian dibandingkan dengan vektor-vektor representasi dari dokumen yang ada dalam database. Untuk mengukur kesamaan antara query dan dokumen adalah Cosine Similarity. Cosine Similarity menghitung derajat kesamaan antara dua vektor dengan mengukur sudut kosinus , yang dapat diwakili oleh rumus:

(2)

Di mana adalah hasil perkalian dot product antara vektor query dan vektor dokumen dan adalah magnitudo dari masing-masing vektor. Hasil dari perhitungan ini akan memberikan nilai antara -1 dan 1, di mana nilai 1 menunjukkan kesamaan yang sempurna. Dokumen-dokumen dengan nilai Cosine Similarity tertinggi dipilih sebagai informasi yang paling relevan untuk digunakan dalam tahap berikutnya.

Proses augmented dalam Retrieval-Augmented Generation (RAG) terjadi melalui penggabungan dua komponen utama: pencarian informasi dan generasi teks. Pada tahap retrieval, sistem mencari informasi relevan dari sumber eksternal berdasarkan query yang diberikan, bukan hanya mengandalkan pengetahuan yang ada di dalam model. Proses ini menambah informasi terkini dan

spesifik yang tidak selalu ada dalam model generatif. Setelah informasi relevan ditemukan, data tersebut digunakan dalam proses generasi, sehingga output tidak hanya berasal dari pengetahuan yang telah dilatih sebelumnya, tetapi juga memperkaya konteks dengan data terbaru. Dengan cara ini, RAG dapat menghasilkan teks yang lebih informatif dan relevan, meningkatkan kualitas dan akurasi hasil akhir, seperti dalam merangkum berita dengan mengambil data terbaru dari berbagai artikel untuk menghasilkan ringkasan yang lebih lengkap dan kontekstual.

Proses Retrieval-Augmented Generation (RAG) lanjut ke tahap generasi teks. Dalam tahap ini, model generatif digunakan untuk menghasilkan teks atau ringkasan baru berdasarkan informasi yang telah diambil dan query yang diberikan. Model generatif bekerja dengan menghitung probabilitas kata-kata dalam teks keluaran secara bertahap, dengan mempertimbangkan kata-kata yang sudah dihasilkan sebelumnya, query, dan informasi yang ditemukan pada tahap retrieval. Proses ini dapat diformulasikan melalui model probabilitas kondisi di mana setiap kata dalam keluaran dihasilkan berdasarkan kata-kata sebelumnya dan konteks yang diperoleh dari dokumen yang diambil, ditulis sebagai:

(3)

Hasilnya adalah menghasilkan kata berikutnya dalam teks. adalah kata yang akan dihasilkan, adalah kata-kata sebelumnya yang memberikan konteks lokal, Query adalah permintaan yang memandu fokus teks, dan Retrieved Information adalah informasi tambahan dari retrieval yang memperluas teks. Dengan menggabungkan konteks lokal dan informasi eksternal, RAG dapat menghasilkan teks yang lebih akurat dan relevan.

Gambar 2 Arsitektur dan Alur Kerja RAG

2.9 Dense Passage Retrieval (DPR)

Dense Passage Retrieval (DPR) merupakan salah satu metode retrieval yang digunakan dalam konteks Retrieval-Augmented Generation (RAG), yang bertujuan untuk mengambil informasi relevan dari korpus teks atau basis data yang sangat besar. Proses ini terjadi pada tahap retrieval, di mana sistem RAG harus memilih dokumen atau bagian teks yang paling relevan dengan permintaan pengguna, seperti pertanyaan atau permintaan ringkasan. DPR memanfaatkan representasi vektor untuk mencocokkan query (pertanyaan) dan dokumen dalam ruang dimensi yang sama, sehingga memungkinkan perhitungan relevansi secara efisien.

Metode DPR menggunakan pendekatan dual-encoder, di mana terdapat dua encoder yang beroperasi secara paralel Question Encoder dan Context Encoder. Question Encoder bertugas mengubah pertanyaan dari pengguna menjadi vektor query, sementara Context Encoder mengubah dokumen atau konteks yang ada dalam korpus menjadi vektor konteks. Dengan memetakan query dan konteks ke dalam ruang vektor yang sama, relevansi antar query dan dokumen dapat dihitung menggunakan metode similarity measurement, seperti cosine similarity. Hal ini memungkinkan DPR untuk menilai dokumen mana yang memiliki kecocokan atau kemiripan paling tinggi dengan query yang diberikan.

DPR memiliki tiga komponen utama dalam arsitekturnya, yaitu Question Encoder, Context Encoder, dan Reader.

1. Question Encoder bertugas untuk mengubah pertanyaan pengguna menjadi representasi vektor yang padat (dense vector) sehingga pertanyaan tersebut dapat diproses dalam ruang vektor yang sama dengan konteks dokumen.
2. Context Encoder berfungsi untuk mengonversi dokumen atau bagian teks dari korpus ke dalam vektor yang relevan. Setiap dokumen dalam korpus teks diubah menjadi vektor yang mewakili kontennya secara komprehensif.
3. Reader kemudian bekerja dengan menggunakan kedua vektor ini (vektor query dan vektor konteks) untuk menghitung relevansi antara pertanyaan dan dokumen, serta memilih dokumen yang memiliki kecocokan tertinggi. Setelah dokumen relevan ditemukan, Reader dapat memberikan teks yang kemudian digunakan sebagai input untuk model generatif pada tahap generation dalam RAG, seperti model Pegasus.

DPR memanfaatkan pre-trained language models, seperti BERT, untuk melakukan encoding query dan konteks ke dalam vektor. Keunggulan utama dari pendekatan ini adalah kemampuan untuk melakukan pencarian teks secara dense (padat), yang jauh lebih efisien daripada pendekatan pencarian berbasis kata kunci atau token sederhana. DPR tidak hanya mempertimbangkan kemunculan kata yang sama antara query dan dokumen, tetapi juga makna semantik yang lebih dalam, yang memungkinkan pencarian yang lebih akurat dan kontekstual.

Dalam implementasinya pada sistem RAG, Dense Passage Retrieval (DPR) menjadi elemen penting yang memungkinkan pengambilan informasi yang lebih presisi dari korpus teks yang besar, sehingga sistem dapat menghasilkan ringkasan atau jawaban yang sesuai dengan kebutuhan pengguna. Integrasi DPR dalam RAG

berperan dalam memperkuat relevansi antara input pengguna dan hasil yang dihasilkan oleh model generatif, memastikan bahwa output yang dihasilkan tidak hanya koheren tetapi juga sangat relevan dengan konteks yang diminta.

2.9 Pegasus

Pegasus adalah salah satu model deep learning yang dikembangkan oleh Google Research dengan fokus pada tugas abstractive text summarization. Model ini tergolong dalam kategori sequence-to-sequence models (seq2seq) dan menggunakan teknik transformer, yang memungkinkan Pegasus untuk menghasilkan ringkasan teks dengan mempertahankan makna dan struktur yang alami. Pegasus dirancang secara khusus untuk menangani skenario ringkasan teks dalam berbagai domain, termasuk teks berita, artikel ilmiah, hingga dokumen teknis [13].

Dalam konteks Retrieval-Augmented Generation (RAG), Pegasus dapat diintegrasikan sebagai model generatif pada tahap generation. Pada tahap ini, Pegasus memanfaatkan informasi yang telah diambil atau retrieved oleh retriever, seperti Dense Passage Retrieval (DPR), untuk kemudian menghasilkan teks baru yang lebih koheren, alami, dan relevan. Tahap retrieval berperan penting dalam menyediakan konteks tambahan yang berkaitan dengan teks awal, sehingga model Pegasus memiliki referensi yang lebih kaya dan bermakna dalam menghasilkan ringkasan. Hal ini menjadikan model generatif seperti Pegasus mampu menghasilkan ringkasan yang lebih komprehensif dibandingkan metode extractive summarization yang hanya mengambil potongan teks langsung dari dokumen tanpa melakukan proses abstraksi.

Secara khusus, Pegasus menggunakan teknik inovatif yang disebut gap-sentence generation (GSG). Teknik ini melibatkan penghapusan kalimat kunci dalam dokumen sumber, di mana model kemudian dilatih untuk memprediksi dan menghasilkan kalimat yang hilang tersebut. Proses pelatihan ini mempersiapkan model Pegasus untuk memahami konteks yang lebih luas dan memberikan keluaran yang mencerminkan inti dari keseluruhan dokumen secara abstraktif, bukan sekadar menyalin kalimat dari teks asli.

Dalam penerapannya pada berita berbahasa Indonesia, penggunaan model RAG dengan integrasi Pegasus dapat memberikan manfaat yang signifikan. Pegasus, dengan kemampuannya dalam menghasilkan teks abstraktif, dapat memperbaiki kualitas ringkasan dengan menyertakan informasi tambahan yang relevan dari retriever, sehingga hasil ringkasan menjadi lebih lengkap dan kontekstual. Selain itu, model ini juga memastikan bahwa ringkasan yang dihasilkan tetap mudah dipahami, menjaga koherensi, dan mengikuti alur logis dari dokumen asli. Penggunaan Pegasus dalam RAG memungkinkan terciptanya ringkasan teks yang lebih kaya dan bermakna, menjawab kebutuhan informasi secara lebih efektif dibandingkan dengan model ringkasan ekstraktif yang hanya menyajikan potongan teks tanpa interpretasi.

2.10 Evaluasi

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) adalah metrik evaluasi yang digunakan untuk menilai kualitas ringkasan teks otomatis dengan membandingkan ringkasan yang dihasilkan oleh sistem dengan ringkasan referensi. Metrik ini berguna untuk mengukur seberapa baik ringkasan yang dihasilkan mencerminkan konten penting dari teks aslinya. ROUGE sering digunakan dalam penelitian text summarization karena kesederhanaannya dan kemampuannya untuk memberikan gambaran tentang kualitas ringkasan berdasarkan beberapa aspek.

1. ROUGE-N adalah metrik yang mengukur kesamaan antara n-gram (kelompok kata berurutan) dalam ringkasan yang dihasilkan dengan n-gram dalam ringkasan referensi.

(4)

(5)

(6)

(7)

(8)

Jumlah n-gram yang cocok: Ini adalah jumlah kelompok kata yang sama antara ringkasan yang dihasilkan dan ringkasan referensi. Jumlah total n-gram: Ini adalah jumlah semua kelompok kata dalam ringkasan referensi

2. ROUGE-L mengukur panjang subsequence umum terpanjang (Longest Common Subsequence, LCS) antara ringkasan yang dihasilkan dan ringkasan referensi. LCS adalah urutan kata yang muncul dalam urutan yang sama dalam kedua ringkasan, meskipun tidak harus berurutan

(9)

Panjang subsekuens umum terpanjang: Ini adalah jumlah kata dalam urutan yang sama yang ditemukan di kedua ringkasan. Panjang ringkasan

referensi: Ini adalah jumlah total kata dalam ringkasan referensi.

3. Hasil Percobaan

Hasil pengujian diperoleh menunjukkan bahwa sistem dan proses berhasil mengambil dan merangkum berita.

3.1 Blackbox Testing

Dalam melakukan pengujian Black Box Testing terdapat beberapa modul tampilan yang sudah diuji. Berikut adalah modul-modul yang diuji.

1. Modul Scraping
   Modul ini merupakan langkah pertama dalam proses sebelum input link. Pada modul ini nanti akan ada berisi file-file atau link berita dari website CNN Indonesia dan CNBC Indonesia yang nantinya akan di ekstrasi lagi menggunakan Beautifulsoup untuk mengambil link, judul, dan kategori berita yang akan mejadi dataset dalam berbentuk excel.
2. Modul Beranda
   Modul ini merupakan modul utama yang tampil pertama kali ketika pengguna membuka website. Modul utama berisi tampilan untuk memberikan untuk memberikan input link berita yang akan diproses untuk mendapatkan hasil ringkasan dari website berita CNN Indonesia dan CNBC Indonesia.
3. Modul Proses Ringkasan
   Modul ini merupakan tentang hasil ringkasan setelah melakukan input link berita dari website CNN Indonesia dan CNBC Indonesia.

3.2 Hasil Evaluasi

Dalam proses evaluasi hasil ringkasan, telah dilakukan perhitungan menggunakan metrik ROUGE serta analisis kesamaan antara isi berita dan hasil ringkasan. Uji coba ini melibatkan 50 berita yang diambil dari situs web CNN Indonesia dan CNBC Indonesia, dengan hasil yang menunjukkan performa yang cukup baik. Dari table L 7.1, rata-rata nilai kemiripan antara berita dan ringkasan mencapai 0.9284, mendekati angka 1, yang mengindikasikan tingkat kesamaan yang tinggi.

Metrik ROUGE juga memberikan hasil evaluasi kualitas ringkasan. Dari table L.6.2 ,rata-rata untuk ROUGE-1 Precision adalah 0.7432, menunjukkan kemampuan sistem dalam menangkap kata yang relevan, sementara ROUGE-1 Recall yang rendah di angka 0.0561 menandakan bahwa banyak informasi penting tidak tertangkap. Selain itu, rata-rata ROUGE-2 Precision mencapai 0.6174, tetapi dengan recall yang sangat rendah pada 0.0368, menunjukkan tantangan dalam mencakup keseluruhan informasi dari berita asli. Secara keseluruhan, meskipun hasil evaluasi menunjukkan beberapa kekuatan dalam hal precision, terdapat kebutuhan untuk meningkatkan recall agar ringkasan dapat lebih efektif dalam merepresentasikan informasi dari sumber berita.

3.3 Hasil Waktu Proses Ringkasan

Pada proses ringkasan diuji dengan lima puluh link berita percobaan.

Tabel 1 Waktu Respon Ringkasan

No	Waktu Respon
1	25.57
2	23.45
3	25.98
4	25.04
5	22.23
6	21.02
7	26.04
8	22.86
9	26.93
10	27.56
11	22.60
12	26.58
13	23.62
14	24.06
15	22.94
16	18.68
17	25.40
18	27.75
19	27.12
20	22.61
21	27.93
22	22.14
23	26.62
24	25.39
25	23.95
26	43.04
27	34.34
28	27.87
29	24.52
30	26.74
31	21.78
32	21.50
33	26.62
34	21.83
35	21.97
36	26.55
37	25.83
38	22.46
39	26.74
40	22.15
41	26.49
42	27.95
43	26.04
44	12.10
45	26.63
46	24.83
47	24.30
48	23.50

49	22.90
50	17.76

3.4 Hasil Url Berita dan Ringkasan

Dalam melakukan pengujian ringkasan dibuat kusioner untuk mencari responden agar menilai apakah hasil ringkasan sudah bisa dipahami dari berita.

Dari hasil pengujian yang dilakukan proses scraping yang dilakukan UiPath 20 menit – 40 menit proses ini hanya dilakukan sekali pada saat diawal pembuatan proses dan nantinya akan menjadi 1 file link url berita, dan nantinya akan diproses lagi untuk pengambilan isi dari berita dari link berita yang sudah di Scraping menggunakan Beautifulsoup. Dan akhirnya pada proses ringkasan dapat memperoleh 15 detik – 30 detik untuk mendapatkan hasil ringkasan.

REFERENSI

[1] [1] M. Indriyani, "Efektivitas Penggunaan Media Online Tirto.Id terhadap Pemenuhan Kebutuhan Informasi Berita Livi Zheng," Jurnal Studi Jurnalistik, vol. 2, no. 2, pp. 157–167, Dec. 2020, doi: 10.15408/jsj.v2i2.15065.

[2] [2] U. Rani and K. Bidhan, "Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA," Journal of scientific research, vol. 65, no. 01, pp. 304–311, 2021, doi: 10.37398/jsr.2021.650140.

[3] [3] I. O. William and M. Altamimi, "Text Embedding Implementation Using Retrieval Augmented Generation (RAG) Model Combined With Large Language Model," 2024.

[4] [4] D. Fitrianah and R. N. Jauhari, "Extractive text summarization for scientific journal articles using long short-term memory and gated recurrent units," Bulletin of Electrical Engineering and Informatics, vol. 11, no. 1, pp. 150–157, Feb. 2022, doi: 10.11591/eei.v11i1.3278.

[5] [5] G. Keswani, W. Bisen, H. Padwad, Y. Wankhedkar, S. Pandey, and A. Soni, "Abstractive Long Text Summarization using Large Language Models." [Online]. Available: www.ijisae.org

[6] [6] F. Amalia Rahmadianti and N. Hendrastuty, "THE INFLUENCE OF FEATURE EXTRACTION ON AUTOMATIC TEXT SUMMARIZATION USING GENETIC ALGORITHM," vol. 5, no. 4, pp. 79–84, 2024, doi: 10.52436/1.jutif.2024.5.4.2064.

[7] [7] I. Muslim et al., "Implementasi Text Summarization Pada Review Aplikasi Digital Library System Menggunakan Metode Maximum Marginal Relevance," 2024.

[8] [8] Y. A. Hafiz and E. Sudarmilah, "IMPLEMENTASI WEB SCRAPING PADA PORTAL BERITA ONLINE," 2023.

[9] [9] F. Koto, J. H. Lau, and T. Baldwin, "Liputan6: A Large-scale Indonesian Dataset for Text Summarization," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.00679

[10] [10] Y. E. Işıkdemir, "NLP TRANSFORMERS: ANALYSIS OF LLMS AND TRADITIONAL APPROACHES FOR ENHANCED TEXT SUMMARIZATION," Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, vol. 32, no. 1, pp. 1140–1151, Apr. 2024, doi: 10.31796/ogummf.1303569.

[11] [11] S. Liu, J. Wu, J. Bao, W. Wang, N. Hovakimyan, and C. G. Healey, "Towards a Robust Retrieval-Based Summarization System," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.19889

[12] [12] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," May 2020, [Online]. Available: http://arxiv.org/abs/2005.11401

[13] [13] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020, November). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In International conference on machine learning (pp. 11328-11339). PMLR.

[14] [14] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, "A Survey of Automatic Text Summarization: Progress, Process and Challenges," IEEE Access, vol. 9, pp. 156043–156070, 2021, doi: 10.1109/ACCESS.2021.3129786.

[15] [15] A. Pradhan and K. Kumar Todi, "Understanding Large Language Model Based Metrics for Text Summarization," 2023.

[16] [16] E. Malinen, "INTERACTIVE DOCUMENT SUMMARIZER USING LLM TECHNOLOGY," 2024.

[17] S. Gaddam BTech Scholar, "ADVANCED SEARCH AND SUMMARIZATION OF EDUCATIONAL DOCUMENTS USING MACHINE LEARNING," Journal of Nonlinear Analysis and Optimization, vol. 15, no. 6, p. 2024, 2024.