Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2024118088
Vol. 11, No. 6, Desember 2024, hlm. 1273-1280 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

ANALISIS PERFORMA EKSTRAKSI KONTEN *GPT-3* DENGAN MATRIK *BERTSCORE* DAN *ROUGE*

Yetti Yuniati*[1] , Kaira Milani Fitria[2], Melvi[3], Sri Purwiyanti[4], Emir Nasrullah[5], Meizano A Muhammad[6]

[1,3,4,5,6] Universitas Lampung, Lampung
[2] Institut Informatika dan Bisnis Darmajaya, Lampung
Email: *[1] yetti.yuniati@eng.unila.ac.id , [2]kairaamilanii@gmail.com, [3]melvi@eng.unila.ac.id, [4]sri.purwiyanti@eng.unila.ac.id, [5]emir.nasrullah@eng.unila.ac.id, [6]meizano@eng.unila.ac.id
*Penulis Korespondensi

(Naskah masuk: 23 November 2023, diterima untuk diterbitkan: 21 November 2024)

Abstrak

Integrasi model bahasa canggih dalam tugas-tugas pembangkitan teks telah menampilkan beberapa aplikasi yang luas di berbagai bidang, termasuk ekstraksi konten. Penelitian ini memanfaatkan model bahasa *OpenAI GPT-3* untuk mengembangkan aplikasi yang membantu dalam proses persiapan konten penulisan kreatif dengan menerapkan fitur ekstraksi konten. Fitur-fitur ini mencakup ekstraksi informasi, meringkas paragraf, mengidentifikasi topik utama, dan menafsirkan teks untuk presentasi terminologi yang optimal. Penelitian ini menggunakan pendekatan *'few-shot learning'* yang melekat pada model *GPT-3*. Kinerja aplikasi ini dievaluasi secara ketat melalui uji coba, membandingkan efektivitasnya dengan mesin pembangkitan teks komersial yang banyak digunakan saat ini. Tujuannya adalah menganalisis tingkat kelayakan sistem yang telah kami bangun terhadap aplikasi lain yang populer. Metrik evaluasi termasuk *BERTscore* dan *ROUGE* digunakan sebagai pengujian. Aplikasi ini mencapai *BERTscore* sebesar 86% untuk *precision*, 88% untuk *recall*, dan 87% untuk *F1-Score*. Selain itu, evaluasi *ROUGE* menghasilkan skor *ROUGE-L* sebesar 55% pada *precision*, 60% pada *recall*, dan 57% pada *F1-Score*, hasil tersebut menunjukkan kekuatan model dalam tugas ekstraksi konten. Hasil ini memberikan gambaran bahwa model *GPT-3* berpotensi baik dalam meningkatkan efisiensi dan akurasi untuk tugas persiapan konten tulisan dalam industri penulisan kreatif.

Kata kunci: *GPT-3, Model Bahasa, Ekstraksi Konten, Pembuatan Teks, NLP*

PERFORMANCE ANALYSIS OF GPT-3 CONTENT EXTRACTION WITH BERTSCORE AND ROUGE MATRICES

Abstract

The integration of advanced language models in text generation tasks has featured some extensive applications in various fields, including content extraction. This research utilises the OpenAI GPT-3 language model to develop an application that assists in the content preparation process of creative writing by implementing content extraction features. These features include information extraction, summarising paragraphs, identifying main topics, and interpreting text for optimal terminology presentation. This research utilises the 'few-shot learning' approach inherent to the GPT-3 model. The performance of this application was rigorously evaluated through trials, comparing its effectiveness with commercial text generation engines widely used today. The aim is to analyse the feasibility of the system we have built against other popular applications. Evaluation metrics including BERTscore and ROUGE were used as tests. The application achieved a BERTscore of 86% for precision, 88% for recall, and 87% for F1-Score. In addition, the ROUGE evaluation resulted in ROUGE-L scores of 55% in precision, 60% in recall, and 57% in F1-Score, these results show the strength of the model in the content extraction task. These results illustrate that the GPT-3 model has good potential in improving efficiency and accuracy for the task of writing content preparation in the creative writing industry.

Keywords: *GPT-3, Language Model, Content Extraction, Text Generation, NLP*

1. PENDAHULUAN

Kecerdasan buatan (*AI*) telah menjadi hal yang umum dalam kehidupan sehari-hari dan mengalami perkembangan pesat (Latinovic & Chatterjee, 2022).

Salah satu cabang *AI*, Pemrosesan Bahasa Alami (*NLP*), memungkinkan pemrosesan teks manusia secara otomatis, termasuk menghasilkan teks yang tampak alami (Bahja, 2020). Model bahasa adalah jenis model pembelajaran mesin yang dilatih untuk memprediksi probabilitas urutan kata atau token (Scao et al., 2022). Model bahasa memiliki sejarah panjang dalam penerapannya dalam *NLP* (Mikolov et al., 2010), dengan tujuan memprediksi token-token berikutnya (Chen & Goodman, 1999). Beberapa model bahasa, seperti model berbasis *transformer*, telah menunjukkan efisiensi yang lebih tinggi dibandingkan pendekatan lainnya (Vaswani et al., 2017). Model bahasa terlatih yang paling representatif saat ini termasuk *BERT* dari *Google* dan *GPT-2* serta *GPT-3* dari *OpenAI* (Brown et al., 2020), yang semuanya dibangun di atas arsitektur *transformer*. Model *GPT*, yang pertama kali menggunakan *transformator unidirectional* sebagai kerangka kerja untuk model bahasa *generative pre-training*, menunjukkan potensi signifikan dari teknik pra-pelatihan untuk berbagai aplikasi lanjutan (Radford et al., 2018). Namun, kemajuan ini juga menciptakan tantangan-tantangan baru yang perlu diatasi (Gevaert et al., 2021). Model *GPT-3*, sebagai generasi ketiga dari *Generative Pre-trained Transformer*, melakukan pembelajaran tanpa pengawasan pada korpus data teks yang sangat besar, memungkinkan respons dalam bahasa tertulis yang mencerminkan respons manusia (Chan, 2023). *GPT-3* adalah model bahasa yang dapat digunakan untuk ekstraksi konten (Ramachandran et al., 2022). Penelitian dibidang ini telah mencapai kinerja terobosan pada tugas-tugas NLP dalam beberapa tahun terakhir (Taylor et al., 2022). Model bahasa dapat dilatih pada berbagai tugas, termasuk pemodelan bahasa autoregresif, pemodelan bahasa lintas-bahasa, dan pembelajaran *few-shot* (Alayrac et al., 2022; Howard & Ruder, 2018; Lample & Conneau, 2019; Scao et al., 2022)

Salah satu masalah utama dalam ekstraksi konten dari teks panjang adalah memastikan akurasi dan relevansi informasi yang dihasilkan. Meskipun model seperti *GPT-3* telah menunjukkan kemampuan luar biasa dalam menghasilkan teks yang menyerupai tulisan manusia, masih terdapat tantangan dalam mengintegrasikan fitur-fitur seperti ringkasan paragraf, identifikasi topik utama, dan koreksi kalimat dengan tingkat presisi yang tinggi. Tantangan ini menciptakan kebutuhan untuk penelitian lebih lanjut guna meningkatkan kemampuan model dalam tugas-tugas tersebut.

GPT-3 merupakan model auto-regresif yang dapat menjawab pertanyaan, meringkas, dan menerjemahkan teks (Chiu et al., 2021). Arsitektur utama *GPT* adalah *decoder* (Radford et al., 2018). Model *GPT* memiliki 175 miliar parameter dan ukuran kosakata sebesar 50.257 (Yadin, 2001). *GPT-3* dapat diaplikasikan untuk tugas-tugas baru *few-shot learning* pada *prompting*-nya, dan telah menunjukkan kemampuannya untuk menghasilkan teks yang dapat dipahami oleh manusia (Dale, 2021). *GPT-3* mengetahui hampir semua domain secara alami. Hanya perlu sedikit instruksi tentang apa yang harus dilakukan hasil yang diberikan akan digunakan untuk pembelajaran kembali (*few-shot learning*) (Wang et al., 2020). Model "*text-davinci-003*" pada *GPT-3* menggunakan dataset pelatihan yang terdiri dari 45 juta halaman web, buku, dan sumber-sumber lainnya (Haluza & Jungwirth, 2023). Meskipun model bahasa ini mengetahui hampir semua domain secara alami, tantangan tetap ada dalam memastikan akurasi dan relevansi teks yang dihasilkan (Singh et al., 2021; Thomas J Ackermann, 2020). *GPT-3* meningkatkan skala data (45 TB vs 40 GB) dan ruang parameter (175 miliar vs 1,5 miliar), menjadikannya model bahasa yang paling signifikan yang pernah dibuat (Zhang & Li, 2021). Model ini juga dapat dihubungkan ke platform seperti situs web melalui *API* dan digunakan untuk berbagai fungsi (Haluza & Jungwirth, 2023), termasuk membuat anotasi data untuk pelatihan model pembelajaran mesin (Ding et al., 2022). *Prompt-Learning* memberikan wawasan tentang apa yang dapat disertakan dalam pemrosesan bahasa alami (*NLP*) di masa depan (Lester et al., 2021).

Penelitian ini bertujuan untuk mengembangkan aplikasi yang mendukung proses persiapan konten penulisan kreatif menggunakan model *GPT-3* dari *OpenAI*. Aplikasi ini mengimplementasikan fitur-fitur seperti ringkasan paragraf, poin utama, ekstraksi kata kunci, dan korektor kalimat *AI* untuk membantu pengguna dalam menghasilkan teks yang lebih terstruktur dan bermakna. Metode yang digunakan dalam penelitian ini adalah pendekatan *'few-shot learning*' dari model *GPT-3*. Evaluasi dilakukan dengan membandingkan kinerja aplikasi terhadap mesin pembangkitan teks komersial lainnya menggunakan metrik *BERTscore* dan *ROUGE*. Urgensi penelitian ini terletak pada meningkatnya kebutuhan akan alat yang dapat mendukung penulis dalam menghasilkan konten yang berkualitas tinggi dengan efisiensi yang lebih baik. Dalam dunia yang semakin digital, di mana konten teks menjadi bagian integral dari komunikasi, pendidikan, dan bisnis, aplikasi yang mampu meningkatkan proses penulisan kreatif akan sangat bermanfaat. Penelitian ini memberikan solusi inovatif untuk target pengguna agar dapat dengan mudah mengakses fitur kami, sehingga dibentuklah hasil penelitian ini dalam bentuk aplikasi. Penelitian ini juga bertujuan untuk menganalisis efisiensi dan akurasi suatu model bahasa dalam ekstraksi konten, yang diharapkan dapat memberikan kontribusi signifikan pada perkembangan teknologi *NLP* di masa depan.

2. METODE PENELITIAN

Metode penelitian dilakukan secara kuantitatif berdasarkan fokus penelitian dengan mengembangkan aplikasi yang bekerja untuk tugas

copywriting pada perancangan sistem dan pengembangan aplikasi. Penggunaan model *GPT-3* pada aplikasi yang dibuat dilakukan dengan *training* dan menggunakan kode *API* dari *GPT-3* tersebut pada kode program web yang telah dibuat.

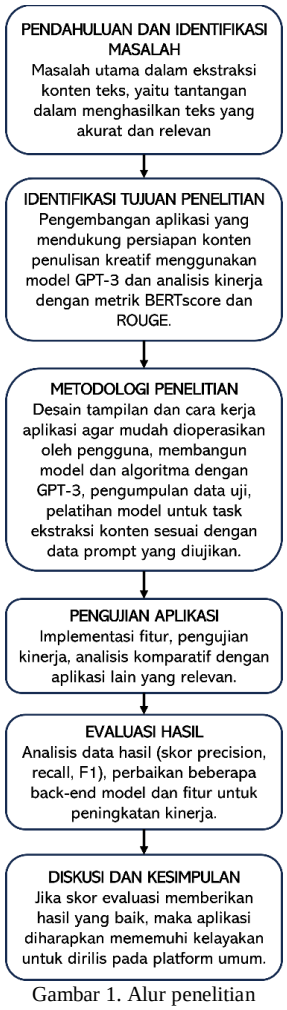
Berdasarkan solusi yang dirancang, dihasilkanlah sebuah demonstrasi untuk menguji aplikasi dan mulai melihat kesesuaian rancangan dengan ekspektasi yang ingin dicapai. Mendemonstrasikan sebuah aplikasi sebelum dirilis memiliki beberapa tujuan penting. Simulasi untuk sebuah aplikasi sebelum dirilis melibatkan perancangan lingkungan virtual, pembuatan dan implementasi algoritme dan model yang diperlukan, integrasi dengan kepuasan interaksi pengguna, dan memastikan simulasi berjalan sesuai dengan yang diinginkan. Untuk mencapai hal tersebut, seringkali dibutuhkan keahlian *developer* aplikasi dalam mewujudkan tampilan yang interaktif bagi pengguna, contohnya adalah kemampuan dalam grafis komputer, memodelkan persamaan, kemampuan memahami implementasi kecerdasan buatan, hal-hal berikut akan mendukung kemampuan dalam merancang desain antar muka pengguna pada aplikasi yang akan dirilis (Lohr, 2000). Diagram alir pada Gambar 1 dibawah ini memuat informasi terkait alur penelitian ini secara umum.

Secara keseluruhan, membuat simulasi dalam aplikasi menambahkan elemen dinamis dan interaktif, yang memungkinkan pengguna untuk terlibat dengan lingkungan virtual, skenario, atau sistem yang nyaman untuk digunakan. Umpan balik dari simulasi ini akan meningkatkan keterlibatan pengguna, memfasilitasi pembelajaran, memungkinkan eksperimen, dan menyediakan fungsionalitas yang dapat ditingkatkan.

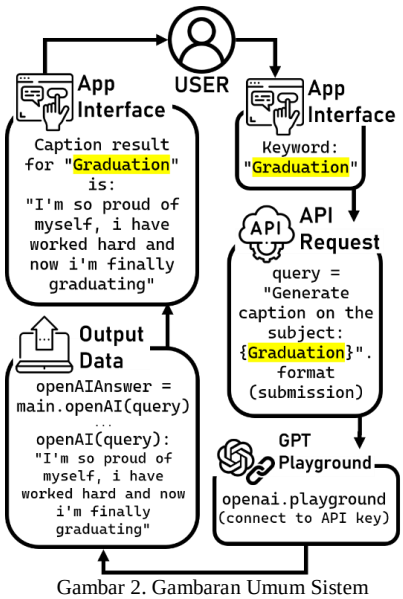
Pengujian program melibatkan evaluasi aplikasi secara sistematis untuk mengidentifikasi cacat, kesalahan, atau masalah dan memastikan aplikasi berfungsi sebagaimana mestinya. Tujuan utama pengujian program adalah untuk meningkatkan kualitas, keandalan, dan kegunaan aplikasi sebelum dirilis ke pengguna. Pengujian penerimaan pengguna dilakukan dengan pengujian aplikasi terhadap pengguna akhir (*end-user*) untuk memastikan aplikasi tersebut memenuhi persyaratan dan harapan mereka. Hal ini sering kali dilakukan di lingkungan dunia nyata untuk mensimulasikan skenario penggunaan yang sebenarnya.

Implementasi sistem melibatkan penerapan dan pengintegrasian sistem perangkat lunak yang dikembangkan ke dalam lingkungan operasional. Implementasi yang sukses melibatkan perencanaan yang matang, penyiapan infrastruktur yang tepat, pengujian yang ketat, pelatihan pengguna, dan transisi yang lancar.

Tujuan penelitian ini untuk merancang aplikasi web yang mampu terhubung dengan model *GPT-3* dan *user* melalui *user interface*. Antarmuka situs web akan mengumpulkan masukan dari pengguna, membuat input yang diberikan menjadi *prompt* pada model *GPT-3*, kemudian teks yang dihasilkan oleh model bahasa pada antarmuka web yang dapat dilihat oleh pengguna. Gambaran umum sistem yang dibangun dapat dilihat pada Gambar 2.



Gambar 1. Alur penelitian



Gambar 2. Gambaran Umum Sistem

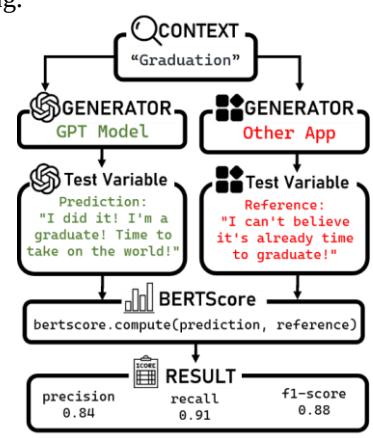
Proses pengembangan aplikasi termasuk pembuatan prototipe aplikasi, demonstrasi fungsionalitas aplikasi, dan melakukan pengujian aplikasi. Konsep desain aplikasi diimplementasikan dengan *training* pada model bahasa melalui *API* atau layanan dari model *GPT-3* pada web yang dibuat. Tugas utamanya adalah membuat koneksi antara aplikasi web, layanan *API* dari *GPT-3*, dan *user* melalui *User Interface*. *User* akan memberikan *input* melalui web, dan aplikasi akan mengubahnya sebagai *prompt* untuk dilanjutkan ke *API* dari *GPT-3*, dan kemudian menampilkan hasil yang dihasilkan oleh model bahasa kembali kepada pengguna pada antarmuka web yang terlihat.

Pelatihan *GPT-3* dalam aplikasi ini menggunakan metode pembelajaran *few-shot*. Metode pembelajaran *few-shot learning* dimulai dengan memberikan *prompt* yang berisi perintah. Kemudian hasilnya diarahkan sesuai dengan yang diinginkan. Spesifikasi model yang digunakan untuk proses pelatihan menggunakan *Complete Mode*, model *text-davinci-002*, Temperatur 0.5 dengan panjang kata Maksimum 256, dan *Top-P* 1.

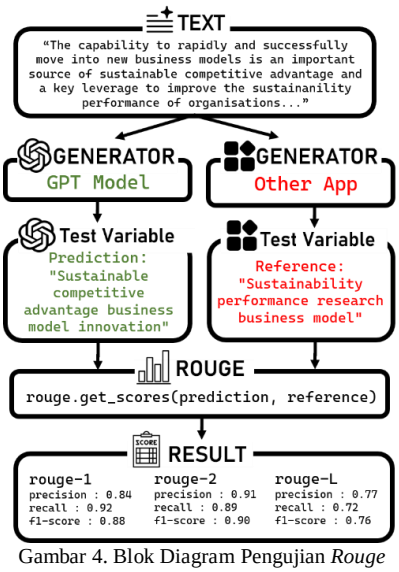
Seluruh aplikasi ditulis dalam perangkat lunak *Visual Studio Code*. Bahasa pemrograman yang digunakan adalah *Python* versi 3.8. *Library* pendukung yang digunakan adalah *Flask* dan *OpenAI*. Pemrograman frontend menggunakan *HTML*, *CSS*, dan *JavaScript*. Perancangan layout website menggunakan aplikasi *Adobe Illustrator* untuk pembuatan komponen grafis. Aplikasi *Figma* digunakan untuk membuat prototipe layout website. Pengujian aplikasi sebelum tahap *deployment* dilakukan dengan akses melalui *localhost* dengan *running* program *script* di *Command Prompt* atau *VS Code* yang terbuka pada *browser Chrome*. Jika hasil uji pada *localhost* tidak ditemukan masalah, folder program akan dilanjutkan ke tahap *deployment*. Tahap *deployment* dilakukan di layanan *PythonAnywhere*, tujuan *deployment* pada *server* tertentu adalah agar *URL* website dapat diakses pada berbagai perangkat. Pembuatan versi aplikasi android dilakukan dengan aplikasi *Android Studio* dengan penyesuaian *layout* website ke bentuk tampilan *mobile*. Perilisan aplikasi yang berbasis Android dilakukan dengan akun *developer* milik peneliti pada platform *Google Play Store*. Aplikasi dirilis hanya untuk melakukan pengujian terhadap kepuasan beberapa responden terpilih, sehingga hanya dapat diakses pada saat periode penelitian berlangsung, mengingat implementasi aplikasi ini dalam skala besar juga memerlukan support sumber daya yang cukup besar.

Terdapat dua pengujian dalam tahap uji aplikasi ini. Kedua uji tersebut dengan melalui matriks evaluasi yang berbeda. Terdapat dua matriks evaluasi yang dirasa paling sesuai dengan uji model bahasa yang digunakan pada aplikasi yang dirancang, yaitu matriks *ROUGE* dan matriks *BERTscore*. Hasil evaluasi dihitung dengan mempertimbangkan fitur-fitur dan mengumpulkan data pengujian untuk setiap fitur untuk menilai kinerja. Setiap fitur kemudian dikategorikan dan diuji menggunakan *BERTscore*, terutama ketika mengukur skor kesamaan "konteks" atau makna kalimat.

Hasil evaluasi matrik dihitung dengan menguji fitur aplikasi agar data pengujian pada setiap fitur dapat digabungkan. Setiap fitur akan ditentukan kategori matriknya antara *ROUGE* atau *BERTscore*. Jika skor *similarity context* atau makna kalimat yang ingin diukur, maka digunakan *BERTscore*. Matrik *ROUGE* digunakan untuk mengukur skor kemiripan susunan kata dalam sebuah kalimat. Gambar 3 dan 4 mengilustrasikan pengujian dengan matriks evaluasi BERTscore dan ROUGE pada aplikasi yang dirancang.



Gambar 3. Diagram Pengujian *Bertscore*



Gambar 4. Blok Diagram Pengujian *Rouge*

Terdapat dua variabel data yang digunakan pada pengujian, yaitu variabel "prediksi" yang berasal dari hasil kalimat oleh aplikasi yang kami hasilkan, dan variabel "referensi" yang didapatkan dari hasil kalimat oleh aplikasi komersial lain yang sudah rilis sebelumnya. Dalam pengujian, kami juga mempersiapkan beberapa *prompt* atau perintah yang sama untuk dimasukkan kedalam kedua sistem, untuk fitur *paragraph summary* digunakan *prompt* dari data

paragraf yang ada pada sejumlah buku atau novel, pada fitur *key points* dilakukan dengan *random prompt*, pada fitur *keyword extractor* didapatkan data paragraf dari teks berita, dan pada uji *AI sentence corrector* menggunakan beberapa *sample* soal *toefl* yang dimuat dalam sebuah buku latihan *toefl*.

Masing-masing *prompt* yang telah disiapkan, akan diperintahkan ke kedua aplikasi yang dibandingkan, yaitu aplikasi yang kami buat, dan aplikasi pembanding yaitu *Quillbot*, dengan berbagai fiturnya yang relevan dengan aplikasi yang kami buat. Pemilihan aplikasi pembanding *Quillbot* sesuai dengan minat *user* yang tinggi pada aplikasi tersebut karena meningkatkan kualitas penulisan dan pemikiran logis pada pembuatan teks oleh para siswa, serta meningkatkan kemampuan akademis dalam penulisan karya ilmiah (Aladini, 2023; Kurniati & Fithriani, 2022).

Jika data uji sudah didapatkan untuk setiap fitur, maka dilakukan pengujian sesuai dengan matriks yang relevan pada masing-masing fitur aplikasinya. Pengujian dilakukan dengan *import* beberapa *library python* yaitu *ROUGE* dan *BERTscore* dan dijalankan pada *Jupyter Notebook*. *Running* program dilakukan untuk menampilkan hasil uji setiap matriksnya, dengan begitu dapat disimpulkan kualitas kalimat yang dihasilkan oleh aplikasi yang sudah dibuat dan menganalisis kesetaraan kualitasnya dengan aplikasi *text generator* lainnya.

3. HASIL DAN PEMBAHASAN

Hasil pengujian aplikasi mendapatkan skor *ROUGE* dan *BERTscore* untuk masing-masing fitur yang dibandingkan, seperti fitur *Paragraph Summary*, *Key Points*, *Keyword Extractor*, dan *AI Sentence Corrector.*

3.1. Paragraph Summary

Fitur peringkasan paragraf dalam ekstraksi konten mencakup proses pembuatan ringkasan koheren secara otomatis dari blok teks. Pada pengujian ringkasan paragraf, data teks yang berhasil diringkas oleh aplikasi dibandingkan kinerjanya dengan sistem yang serupa. Saat ini, salah satu aplikasi pemrosesan teks berbasis *AI* adalah *Quillbot*, yang salah satu fiturnya adalah *summarize*, sehingga data hasil pengujian aplikasi ini akan dibandingkan dengan hasil dari aplikasi *quillbot summarizer*. Teks yang dihasilkan oleh aplikasi berada pada variabel "prediksi", dan teks yang dihasilkan oleh *web quillbot* berada di variabel "referensi".

Data hasil pengujian diambil dari sepuluh kali percobaan fitur dengan teks uji yang berbeda dari beberapa paragraf data buku atau novel, salah satu contohnya adalah teks berikut:

"*But man is not destined to vanish. He can be killed, but he cannot be destroyed, because his soul is deathless and his spirit is irrepressible. Therefore, though the situation seems dark in the context of the confrontation between the superpowers, the silver lining is provided by amazing phenomenon that the very nations which have spent incalculable resources and energy for the production of deadly weapons are desperately trying to find out how they might never be used*". Hasil yang diberikan oleh aplikasi adalah kalimat *"The situation between the superpowers is dark, but there is a silver lining. The nations which have spent resources and energy on deadly weapons are trying to find out how they might never be used"*

Hasil perhitungan *BERTscore* dan *ROUGE* pada fitur *Paragraph Summary* berupa *Precision*, *Recall*, dan *F1-score* dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1. Hasil *BERTscore* dari *Paragraph Summary*

Precision	Recall	F1-score
0,907984078	0,879039526	0,893277407
0,944557011	0,893484354	0,918311179
0,904390216	0,916899145	0,910601735
0,944914281	0,939181328	0,942039073
0,898536742	0,889627755	0,894060016
0,845633864	0,841136932	0,843379378
0,905197263	0,876652658	0,890696347
0,878227592	0,875805497	0,877014875
0,908895135	0,919851184	0,914340317
0,839232445	0,838209093	0,838720441
$\bar{x}_{pr}$	$\bar{x}_{re}$	$\bar{x}_{F1}$
0,897756863	0,886988747	0,892244077

Performa aplikasi dalam meringkas teks setara dengan performa *quillbot summarizer* dengan persentase *BERTscore* untuk *precision* 89%, *recall* 88%, dan *F1-score* 89%.

Tabel 2. Hasil *ROUGE* dari *Paragraph Summary*

	Precision	Recall	F1-score
ROUGE-1	0,392497502	0,364579913	0,37121377
ROUGE-2	0,158064186	0,151428488	0,152692872
ROUGE-L	0,343533839	0,321175444	0,325876348

Hasil *ROUGE* pada *Paragraph Summary* adalah *precision* (0.39), *recall* (0.36), dan *f1-score* (0.37). Kemiripan hasil ringkasan aplikasi dalam meringkas teks dengan kinerja *quillbot summarizer* memiliki persentase *precision* 39%, *recall* 36%, dan *f1-score* 37%.

3.2. Key Points

Fitur penentuan poin-poin penting dalam ekstraksi konten membutuhkan kemampuan identifikasi dan ekstraksi informasi yang relevan dari teks atau dokumen tertentu. Proses ini penting untuk ringkasan, pencarian informasi, dan analisis data. Teks yang dihasilkan oleh aplikasi yang kami uji berada di variabel "prediksi", dan teks yang diinputkan sebagai perintah masuk ke variabel "referensi". Tujuannya adalah untuk mengukur apakah hasil yang diberikan oleh aplikasi memiliki relevansi dengan perintah yang diinputkan.

Salah satu pengujian dilakukan dengan menginputkan kata *"publish journal"*, dan hasil yang diberikan oleh aplikasi adalah beberapa poin yang memuat tahapan yang dibutuhkan dalam publikasi

penelitian dengan detail hasil sebagai berikut *"There are few key things to keep in mind when publishing journal articles: Make sure your research is high of quality and meets the journal's standards. Then, pay attention to the journal's submission guidelines, and follow them carefully. Next, be prepared to revise your article based on feedback from the journal's editors and reviewers. Once your article is accepted, it will be published in the journal and made available to readers"*.

Pengujian dilakukan dengan mengambil 10 hasil test pada teks yang berbeda. Hasil perhitungan *BERTscore* pada fitur *Key Point* dapat dilihat pada Tabel 3.

Tabel 3. Hasil *BERTscore* dari *Key Points*

Precision	Recall	F1-score
0,779897749	0,843341649	0,810379863
0,815431595	0,897724867	0,854601681
0,790260077	0,89763701	0,840533078
0,812404156	0,92686379	0,865867734
0,798271835	0,858163536	0,827134967
0,79590416	0,873222709	0,832772672
0,747204781	0,820692658	0,782226503
0,812427282	0,903312325	0,85546267
0,774516702	0,839854479	0,80586344
0,802556157	0,88449955	0,841537774
$\bar{x}_{pr}$	$\bar{x}_{re}$	$\bar{x}_{F1}$
0,792887449	0,874531257	0,831638038

Performa aplikasi dalam membuat poin-poin penting memiliki kecocokan konteks dengan *BERTscore precision* 79%, *recall* 87%, dan *F1-score* 83%.

3.3. Keyword Extractor

Fitur yang menghasilkan kata kunci dalam ekstraksi konten melibatkan proses mengidentifikasi dan mengekstrak kata atau frasa yang paling relevan dan signifikan secara otomatis dari teks yang diberikan. Kata kunci ini berfungsi sebagai representasi dari topik atau tema utama yang ada di dalam konten. Pada pengujian ekstraktor kata kunci, kata kunci dari aplikasi dibandingkan kinerjanya dengan kata kunci yang dihasilkan oleh aplikasi penghasil teks lain (*quillbot*). Teks yang dihasilkan oleh aplikasi berada di variabel "prediksi", dan teks yang dihasilkan oleh web *quillbot* berada di variabel "referensi".

Pengujian dilakukan dengan mengambil teks hasil sebanyak sepuluh kali dengan teks yang berbeda. Hasil perhitungan *ROUGE* pada fitur *Keyword Extractor* berupa *Precision, Recall, dan F1-score* pada setiap jenis parameter *ROUGE-1, ROUGE-2,* dan *ROUGE-L* yang dapat dilihat pada Tabel 4.

Tabel 4. Hasil *ROUGE* dari *Keyword Extractor*

	Precision	Recall	F1-score
ROUGE-1	0,392497502	0,364579913	0,37121377
ROUGE-2	0,158064186	0,151428488	0,152692872
ROUGE-L	0,343533839	0,321175444	0,325876348

Hasil *ROUGE* pada *Keyword Extracto*r adalah *precision* (0.27), *recall* (0.45), dan *f1-score* (0.34). Kemiripan teks hasil aplikasi dalam mengekstrak kata kunci dari sebuah teks dengan aplikasi *quillbot* memiliki persentase *precision* 27%, recaill 45%, dan *f1-score* 34%.

3.4. AI Sentence Corrector

Fitur perbaikan kalimat dalam ekstraksi konten melibatkan proses mengidentifikasi dan memperbaiki kesalahan tata bahasa, sintaksis, dan semantik secara otomatis dalam kalimat atau teks tertentu. Tugas ini bertujuan untuk meningkatkan koherensi, kejelasan, dan keakuratan konten secara keseluruhan. Pada pengujian korektor kalimat, teks yang telah berhasil dikoreksi kalimatnya oleh aplikasi dibandingkan keakuratannya dengan soal *TOEFL* dengan kunci jawaban. Teks yang dihasilkan aplikasi berada pada variabel "prediksi" dan kalimat soal *TOEFL* beserta kunci jawaban berada pada variabel "referensi". Pengujian dilakukan dengan memasukkan soal *TOEFL* (dengan tata bahasa yang salah) sebagai input ke aplikasi sehingga memberikan hasil yang menunjukkan apakah sesuai dengan kunci jawaban soal *TOEFL*.

Pengujian dilakukan dengan mengambil teks hasil sebanyak sepuluh kali dengan teks yang berbeda. Hasil perhitungan *ROUGE* pada fitur *Sentence Corrector* berupa *Precision, Recall*, dan i pada setiap jenis parameter *ROUGE-1, ROUGE-2,* dan *ROUGE-L* yang dapat dilihat pada Tabel 5.

Tabel 5. Hasil *ROUGE* dari *Sentence Corrector*

	Precision	Recall	F1-score
ROUGE-1	1	1	0,999999994
ROUGE-2	1	1	0,999999994
ROUGE-L	1	1	0,999999994

Hasil *ROUGE* pada *Sentence Corrector* adalah *precision* (1.0), *recall* (1.0), dan *f1-score* (0.99). Kemiripan teks hasil aplikasi pada kalimat yang benar dengan soal *TOEFL* memiliki persentase *precision* 100%, *recall* 100%, dan *f1-score* 99%.

Skor keseluruhan dari pengujian fitur evaluasi matriks hasil aplikasi dapat dilihat pada Tabel 6 untuk *BERTscore* dan Tabel 7 untuk *ROUGE*. Fitur yang diuji pada *BERTscore* adalah *Paragraph Summary* dan *Key Points*, dan fitur yang diuji pada *ROUGE* adalah *Paragraph Summary, Keyword Extractor*, dan *AI Sentence Corrector*.

Tabel 6. Hasil *BERTscore* total

Fitur	Precision	Recall	F1-score
Paragraph Summary	0,897756863	0,886988747	0,892244077
Key Points	0,792887449	0,874531257	0,831638038
Rata-Rata	0,845322156	0,880760002	0,8619410575

Hasil pengukuran *BERTscore* pada fitur aplikasi memberikan rata-rata 84% untuk presisi, 88% untuk *recall*, dan 86% untuk skor *F1*.

Hasil pengukuran matriks *ROUGE* pada fitur aplikasi memberikan rata-rata 55% untuk presisi, 60% untuk *recall*, dan 57% untuk skor *F1*.

Tabel 7. Hasil *ROUGE* total

Fitur	Precision	Recall	F1-score
Paragraph Summary	0,392497502	0,364579913	0,37121377
Sentence Corrector	1	1	0,999999994
Keyword Extractor	0,276331169	0,454404762	0,340048543
Rata-Rata	0,556276224	0,606328225	0,570420769

Penggunaan *BERTScore* dan *ROUGE* untuk analisis kinerja teks memberikan evaluasi yang komprehensif dan bernuansa terhadap teks yang dihasilkan. *BERTScore* memanfaatkan model berbasis *transformator* seperti *BERT*, memberikan metrik pengujian mutakhir untuk menilai kemiripan semantik antara teks yang dihasilkan dan teks referensi. Kemampuannya untuk menangkap informasi kontekstual dan nuansa semantik membuatnya menjadi metrik pengujian yang berkaitan dengan tugas-tugas seperti penerjemahan mesin, peringkasan, dan pembuatan teks.

Sedangkan metrik *ROUGE*, rangkaian metrik yang sudah digunakan sejak lama untuk menangkap adanya tumpang tindih dan kemiripan pada tingkat leksikal. *ROUGE* mengevaluasi tumpang tindih *n-gram*, yang sangat berguna untuk tugas-tugas seperti peringkasan dan ekstraksi konten. *ROUGE* juga memungkinkan evaluasi panjang *n-gram* yang berbeda dan bahkan ukuran yang berorientasi pada penarikan seperti *ROUGE-L*.

Dengan menggabungkan kedua metrik ini, akan membantu memberikan pemahaman yang lebih menyeluruh tentang kualitas teks. Penekanan metrik *BERTScore* yaitu pada kesamaan *semantic* dan *ROUGE* pada tumpang tindih leksikal. Pendekatan gabungan ini memungkinkan evaluasi yang lebih kuat yang memperhitungkan kebenaran tingkat permukaan dan koherensi semantik yang lebih dalam. Sehingga pendekatan ini juga memberikan pandangan yang seimbang mengenai kinerja teks di berbagai tugas pemrosesan bahasa alami.

4. KESIMPULAN

Hasil dari pengujian performa ekstraksi konten *GPT-3* dengan matrik *BERTScore* dan *ROUGE* menggunakan dua variabel yang digunakan sebagai pembanding, yaitu "*prediction*" yang didapat dari model bahasa GPT-3, dan variabel "*reference*" yang didapatkan dari model bahasa lain pada suatu aplikasi *text generator*. Pengujian terhadap fitur *Paragraph Summary, Key Points, Keyword Extractor*, dan *AI Sentence Correcto*r pada aplikasi yang dibuat memberikan hasil rata-rata *ROUGE* sebesar 55% untuk *precision*, 60% untuk *recall*, dan 57% untuk *F1-score* serta hasil *BERTscore* sebesar 84% untuk *precision*, 88% untuk *recall*, dan 86% untuk *F1-score*. Skor hasil pengujian ini merupakan skor yang cukup baik bagi aplikasi yang berfokus pada tugas pemrosesan kata dengan mekanisme ekstraksi konten yang relevan dengan perintah yang diberikan oleh *user*.

Saran untuk penelitian selanjutnya adalah peningkatan kriteria untuk penentuan *variable* yang akan diuji, pengembangan fitur aplikasi yang lebih spesifik dengan konsep ekstraksi konten, dan pemilihan tipe model bahasa yang diuji disesuaikan dengan kondisi terbaru. Serta, untuk meningkatkan kekuatan aplikasi, disarankan untuk implementasi secara luas dan terbuka untuk umum dengan sumber daya yang memadai, agar mendapatkan *feedback* yang variatif dengan jangkauan yang luas.

DAFTAR PUSTAKA

ALADINI, Dr. A. 2023. AI applications impact on improving EFL University Academic writing skills and their logical thinking. https://doi.org/10.21608/ssj.2023.320166

ALAYRAC, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., & Reynolds, M. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, *35*, 23716–23736.

BAHJA, M. 2020. Natural Language Processing Applications in Business. In R. M. X. Wu & M. Mircea (Eds.), *E-Business* (p. Ch. 4). IntechOpen. https://doi.org/10.5772/intechopen.92203

BROWN, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. 2020. Language Models are Few-Shot Learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

CHAN, A. 2023. GPT-3 and InstructGPT: technological dystopianism, utopianism, and "Contextual" perspectives in AI ethics and industry. *AI and Ethics*, *3*(1), 53–64. https://doi.org/10.1007/s43681-022-00148-6

CHEN, S. F., & Goodman, J. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, *13*(4), 359–394. https://doi.org/https://doi.org/10.1006/csla.1999.0128

CHIU, K.-L., Collins, A., & Alexander, R. 2021. *Detecting Hate Speech with GPT-3*. *March*, 1–29. http://arxiv.org/abs/2103.12407

DALE, R. 2021. GPT-3: What's it good for? *Natural Language Engineering*, *27*(1), 113–118. https://doi.org/DOI: 10.1017/S1351324920000601

DING, B., Qin, C., Liu, L., Bing, L., Joty, S., & Li, B. 2022. *Is GPT-3 a Good Data Annotator?* http://arxiv.org/abs/2212.10450

GEVAERT, C. M., Carman, M., Rosman, B., Georgiadou, Y., & Soden, R. 2021. Fairness and accountability of AI in disaster risk management: Opportunities and challenges. *Patterns*, *2*(11), 100363. https://doi.org/https://doi.org/10.1016/j.patter.2021.100363

HALUZA, D., & Jungwirth, D. 2023. Artificial Intelligence and Ten Societal Megatrends: An Exploratory Study Using GPT-3. *Systems*, *11*(3). https://doi.org/10.3390/systems11030120

HOWARD, J., & Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:40100965

KURNIATI, E. Y., & Fithriani, R. 2022. Post-Graduate Students' Perceptions of Quillbot Utilization in English Academic Writing Class. *Journal of English Language Teaching and Linguistics*, *7*(3), 437. https://doi.org/10.21462/jeltl.v7i3.852

LAMPLE, G., & Conneau, A. 2019. Cross-lingual language model pretraining. *ArXiv Preprint ArXiv:1901.07291*.

LATINOVIC, Z., & Chatterjee, S. C. 2022. Achieving the promise of AI and ML in delivering economic and relational customer value in B2B. *Journal of Business Research*, *144*, 966–974. https://doi.org/https://doi.org/10.1016/j.jbusres.2022.01.052

LESTER, B., AL-RFOU, R., & CONSTANT, N. 2021. *The Power of Scale for Parameter-Efficient Prompt Tuning*. https://doi.org/10.18653/v1/2021.emnlp-main.243

LOHR, L. L. 2000. Designing the instructional interface. *Computers in Human Behavior*, *16*(2), 161–182. https://doi.org/https://doi.org/10.1016/S0747-5632(99)00057-6

MIKOLOV, T., KARAFIÁT, M., BURGET, L., ERNOCKÝ, J. H., & KHUDANPUR, S. 2010. Recurrent neural network based language model. *Interspeech*. https://api.semanticscholar.org/CorpusID:17048224

RADFORD, A., NARASHIMAN, K., SALIMANS, T., & SUTSKEVER, I. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI*. https://openai.com/blog/language-unsupervised/

RAMACHANDRAN, R., RAMASUBRAMANIAN, M., KOIRALA, P., GURUNG, I., & MASKEY, M. 2022. Language Model for Earth Science: Exploring Potential Downstream Applications as well as Current Challenges. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 4015–4018. https://api.semanticscholar.org/CorpusID:252590791

SCAO, T. Le, Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., & Gallé, M. (2022). Bloom: A 176b-parameter open-access multilingual language model. *ArXiv Preprint ArXiv:2211.05100*.

SINGH, R., Garg, V., & GPT-3. 2021. Human Factors in NDE 4.0 Development Decisions. *Journal of Nondestructive Evaluation*, *40*(3), 71. https://doi.org/10.1007/s10921-021-00808-3

TAYLOR, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., & Stojnic, R. 2022. Galactica: A large language model for science. *ArXiv Preprint ArXiv:2211.09085*.

THOMAS J Ackermann. (2020, November 29). *GPT-3: a robot wrote this entire article. Are you scared yet, human?* Artificial Intelligence: ANI, LogicGate Computing, AGI, ASI.

VASWANI, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. 2017. Attention is All you Need, Advances in neural information processing systems. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

WANG, Y., YAO, Q., KWOK, J. T., & NI, L. M. 2020. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Comput. Surv.*, *53*(3). https://doi.org/10.1145/3386252

YADIN, D. L. 2001. *Creative Marketing Communications: A Practical Guide to Planning, Skills and Techniques* (3rd ed.). Kogan Page Publishers.

ZHANG, M., & LI, J. 2021. A commentary of GPT-3 in MIT Technology Review 2021. *Fundamental Research*, *1*(6), 831–833. https://doi.org/https://doi.org/10.1016/j.fmre.2021.11.011