SKRIPSI

IMPLEMENTASI *TEXT-TO-TEXT TRANSFER TRANSFORMER* BERBASIS ABSTRAKSI DAN EKSTRAKSI DALAM *TEXT SUMMARIZATION* BERITA *ONLINE*

Disusun dan diajukan oleh:

KISANA ADZAN SITORUS
D121 20 1106

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024

LEMBAR PENGESAHAN SKRIPSI

Implementasi *Text-to-Text Transfer Transformer* berbasis Abstraksi dan Ekstraksi dalam *Text Summarization* Berita *Online*

Disusun dan diajukan oleh

Kisana Adzan Sitorus
D121 20 1106

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 22 Juli 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Elly Warni, S.T., M.T.
NIP 19820216 200812 2 001

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, S. T., M. T., M. Bus., Sys., IPM, ASEAN. Eng.
NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Kisana Adzan Sitorus
NIM : D121201106
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Implementasi *Text-to-Text Transfer Transformer* berbasis Abstraksi dan Ekstraksi dalam *Text Summarization* Berita Online

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 1 Agustus 2024

Yang Menyatakan

Kisana Adzan Sitorus

ABSTRAK

KISANA ADZAN SITORUS. Implementasi *Text-to-Text Transfer Transformer* berbasis Abstraksi dan Ekstraksi dalam *Text Summarization* Berita *Online* (dibimbing oleh Elly Warni)

Di era digital, banyaknya informasi memerlukan metode efektif untuk merangkum teks secara otomatis guna membantu pembaca menyerap detail penting. Peringkasan teks adalah tugas penting dalam Natural Language Processing (NLP), yang memungkinkan pengurangan teks panjang menjadi ringkasan singkat tanpa kehilangan informasi esensial. Dengan meningkatnya konsumsi berita digital, menyediakan ringkasan yang singkat dan informatif dapat meningkatkan pengalaman pengguna dan penyebaran informasi.

Penelitian ini bertujuan untuk menganalisis performa model Text-to-text transfer transformer (T5) dalam menghasilkan ringkasan artikel berita Indonesia dan membandingkan efektivitas pendekatan abstraksi dan ekstraksi. Model abstraksi menghasilkan teks baru yang koheren, sedangkan model ekstraksi memilih segmen penting dari teks asli tanpa mengubah struktur kalimat.

Metodologi penelitian ini melibatkan beberapa langkah kunci. Pertama, *dataset* komprehensif artikel berita online berbahasa Indonesia dikumpulkan menggunakan teknik web *scraping* kemudian digabungkan dengan dataset Indosum. *Dataset* ini kemudian dibagi menjadi set pelatihan, validasi, dan uji. Langkah-langkah preprocessing mencakup tokenisasi, pembersihan, dan normalisasi data teks. Kedua model dievaluasi menggunakan metrik ROUGE dan BLEU untuk menilai performa dalam hal presisi, *recall*, dan kualitas keseluruhan ringkasan yang dihasilkan.

Hasil penelitian menunjukkan bahwa model T5 berbasis abstraksi memiliki kinerja lebih baik dibandingkan model ekstraksi. Pada metrik ROUGE-1, model abstraksi memiliki precision 0.838, recall 0.544, dan F1-score 0.662, sedangkan model ekstraksi memiliki precision 0.757, recall 0.444, dan F1-score 0.56. Pada metrik ROUGE-2, model abstraksi memiliki precision 0.778, recall 0.509, dan F1-score 0.617, sementara model ekstraksi memiliki precision 0.889, recall 0.516, dan F1-score 0.651. Pada metrik ROUGE-L, model abstraksi memiliki precision 0.811, recall 0.526, dan F1-score 0.639, sedangkan model ekstraksi memiliki precision 0.919, recall 0.54, dan F1-score 0.679. Selain itu, model abstraksi memiliki skor BLEU 0.375, lebih tinggi dibandingkan model ekstraksi yang memiliki skor BLEU 0.338. Temuan ini menunjukkan bahwa pendekatan abstraksi lebih efektif dalam menghasilkan ringkasan yang koheren dan informatif.

Kata Kunci: *text summarization,* abstraksi, ekstraksi, *text-to-text transfer transformer* (T5), ROUGE

ABSTRACT

KISANA ADZAN SITORUS. *Implementation of Abstraction and Extraction-based Text-To-Text Transfer Transformer in Online News Text Summarization* (*supervised by* Elly Warni)

In the digital era, the vast amount of information requires effective methods for automatically summarizing text to help readers absorb important details efficiently. Text summarization is a crucial task in Natural Language Processing (NLP) that reduces long texts into brief summaries without losing essential information. With the increasing consumption of digital news, providing concise and informative summaries can enhance user experience and information dissemination.

This study aims to analyze the performance of the Text-to-text transfer transformer (T5) model in summarizing Indonesian news articles and compare the effectiveness of abstractive and extractive approaches. The abstractive model generates new coherent text, while the extractive model selects important segments from the original text without altering the sentence structure.

The methodology involves collecting a comprehensive dataset of Indonesian news articles using web scraping and combining it with the Indosum dataset. The dataset is divided into training, validation, and test sets. Preprocessing includes tokenization, cleaning, and normalization of text data. Both models are evaluated using ROUGE and BLEU metrics to assess precision, recall, and summary quality.

Results show that the T5 abstractive model outperforms the extractive model. On the ROUGE-1 metric, the abstractive model has a precision of 0.838, recall of 0.544, and F1-score of 0.662, while the extractive model has a precision of 0.757, recall of 0.444, and F1-score of 0.56. For ROUGE-2, the abstractive model has a precision of 0.778, recall of 0.509, and F1-score of 0.617, while the extractive model has a precision of 0.889, recall of 0.516, and F1-score of 0.651. For ROUGE-L, the abstractive model has a precision of 0.811, recall of 0.526, and F1-score of 0.639, while the extractive model has a precision of 0.919, recall of 0.54, and F1-score of 0.679. Additionally, the abstractive model has a BLEU score of 0.375, higher than the extractive model's BLEU score of 0.338. These findings indicate that the abstractive approach is more effective in producing coherent and informative summaries.

Keywords*:* text summarization, abstraction, extraction, text-to text transfer transformer (T5), ROUGE

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
NLP	*Natural Language Processing*
T5	*Text-to-text transfer transformer*
ROUGE	*Recall-oriented understudy for gisting evaluation*
BLEU	*Bilingual evaluation understudy*
N-Gram	Urutan kata atau karakter yang berurutan
BP	Brevity Penalty (Penalti Keringkasan)
Pn	Precision untuk n-gram
LCS	Longest Common Subsequence (Urutan Subterpanjang yang Sama)
CNN	Convolutional Neural Network (Jaringan Saraf Konvolusi)
Pytorch	Library Deep Learning Python
LSTM	Long Short-Term Memory (Memori Jangka Panjang Pendek)

DAFTAR LAMPIRAN

KATA PENGANTAR

Segala puji hanya milik Allah *Subhanallahu Wa Ta'ala* atas segala limpahan rahmat dan hidayah-Nya. Shalawat dan salam senantiasa tercurahkan kepada baginda Rasulullah *Shallallahu 'Alaihi Wa sallam*. *Alhamdulillahirabbil'aalamiin*, berkat *ridha'* dan kemudahan yang diberikan oleh Allah *Subhanallahu Wa Ta'ala*, sehingga penulis dapat menyelesaikan tugas akhir dengan judul **"Implementasi *Text-To-Text Transfer Transformer* Berbasis Abstraksi dan Ekstraksi dalam *Text Summarization* Berita *Online*"** sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan tugas akhir ini tidak dapat terselesaikan dengan baik tanpa adanya bimbingan dan bantuan baik materi maupun non-materi dari berbagai pihak. Bimbingan dan bantuan tersebut adalah alasan penulisan dan penyusunan tugas akhir ini dapat terselesaikan dengan lancar. Oleh karena itu sebagai salah bentuk penghargaan yang setinggi-tingginya, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Allah Subhanallahu Wa Ta'ala, atas izin rahmat, nikmat dan kasih sayang-Nya sehingga skripsi ini dapat terselesaikan.
2. Kedua orang tua penulis, Bapak Amron Sitorus dan Ibu Hasnawati yang selalu mendoakan untuk kebaikan penulis, selalu memberikan kasih saying, cinta, dukungan dan motivasi. Terima kasih kepada Bapak dan Ibu yang selalu sabar dan semangat tiada henti dalam menghadapi dan mendidik penulis.
3. Ibu Elly Warni, S.T., M.T. selaku pembimbing yang senantiasa meluangkan waktu, tenaga dan pikiran serta perhatian yang luar biasa untuk membimbing penulis dalam proses penyelesaian tugas akhir.
4. Segenap dosen dan staf Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang telah membantu penulis selama masa perkuliahan.

5. Kakak penulis, Asti Asa Sitorus dan Aswati Sahaja Sitorus serta adik penulis Seprilia Gebril Sitorus sebagai tempat berkeluh kesah, pemberi semangat dan motivasi.
6. Teman-teman dan mahasiswa tanpa lab Agung, Rischa, Avil, Calvin, May, Ray, Rizal, Tasya, Shekinah Firman, Muti, Thesa, Dewa, Rifqy, Alex dan teman-teman dekat penulis yang tidak dapat penulis sebutkan satu persatu yang senantiasa selalu membantu dalam menyusun tugas akhir penulis.
7. Teman-teman Geng Asep, Livia, Fadly, Rara, Ismi, Isma, Walda, Nazifah, Indra dan Arya yang menjadi bagian dari perjalanan studi penulis.
8. Teman-teman perantauan dari Papua yang juga sedang menggapai cita-cita, Muliadi, Kesya, Anin, Aging yang juga sebagai tempat berkeluh kesah.
9. Teman-teman Rezolver 20 yang telah membersamai penulis selama masa perkuliahan, atas waktu bersama dan hiburan-hiburan yang telah diberikan.
10. Seluruh pihak yang tidak disebutkan dan tanpa sadar telah menjadi inspirasi dan membantu penulis dalam menyelesaikan tugas akhir

Penulis berharap semoga Tuhan Yang Maha Esa berkenan membalas segala bantuan dan dukungan dari semua pihak yang membantu dalam penyelesaian tugas akhir ini. Penulis menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan yang membangun dari berbagai pihak. Semoga tugas akhir ini dapat berguna bagi para pembaca.

Gowa, 21 Juni 2024

Kisana Adzan Sitorus

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini kita menghadapi *overload* informasi di mana tersedia terlalu banyak informasi dari berbagai sumber seperti situs berita, blog, dan media sosial yang tidak dapat diserap dan diproses dengan cepat (Sposteru, 2022). Di Indonesia, 89% responden mengakses berita dari portal media *online* (Tobergte and Curtis, 2023). Diperkirakan ada 11.000 postingan baru per detik di media sosial dan 2 juta email baru per menit di seluruh dunia (Desjardins, 2022). Ledakan informasi ini menyebabkan kesulitan bagi pengguna internet untuk menyerap, memahami, dan mengolah semua informasi yang tersedia, yang disebut sebagai information *overload* (Edmunds & Morris, 2000).

Salah satu solusi yang ditawarkan untuk mengatasi masalah ini adalah *text summarization* atau peringkasan teks otomatis (Radev et al., 2020). *Text summarization* merupakan teknik meringkas informasi penting dari satu atau beberapa sumber teks menjadi ringkasan yang lebih singkat dan padat (Allahyari, Trippe and Gutierrez, 2017). Dengan membaca ringkasan yang dihasilkan, pembaca diharapkan dapat memahami ide dan informasi utama dari teks asli secara lebih efisien. Teknik ini memiliki potensi besar untuk membantu mengatasi beban informasi yang berlebihan di era digital saat ini.

Text summarization dinilai penting terutama untuk meringkas berita dan informasi daring yang semakin banyak. Sejumlah penelitian menunjukkan *text summarization* berita daring yang berkualitas dapat meningkatkan pemahaman pembaca dan penghematan waktu hingga 70% dibandingkan membaca keseluruhan berita (Moratanch & Chitrakala, 2016).

Text Summarization menghasilkan ringkasan teks berdasarkan metode abstraksi dan ekstraksi (Allahyari et al., 2017). Dalam ringkasan ekstraktif, ringkasan dari teks yang diberikan dibuat dengan memilih subset dari keseluruhan kalimat. Frase atau kalimat yang paling penting dari teks tersebut diidentifikasi dan dipilih berdasarkan skor yang dihitung tergantung pada kata-kata dalam kalimat tersebut (Moratanch and Chitrakala, 2017). Dalam metode ringkasan abstraktif,

interpretasi pertama-tama dibuat dengan menganalisis dokumen teks. Berdasarkan interpretasi ini, mesin memprediksi ringkasan. Metode ini mengubah teks dengan meringkas bagian dari dokumen asli (Moratanch and Chitrakala, 2016). Metode abstraksi dan ekstraksi dipilih karena karakteristik data berita *online* yang panjang dan detail. Metode abstraksi akan menangkap inti berita dan membuat ringkasan global. Metode ekstraksi akan mengekstrak bagian penting dari detail berita. Kombinasi keduanya dapat menghasilkan ringkasan berita yang baik. *Text-to-text transfer transformer* (T5) sangat cocok untuk tugas *text summarization*. Model ini dilatih dengan pasangan teks sumber dan ringkasan, sehingga dapat mempelajari pola untuk meringkas teks dengan baik.

Penelitian terkait mengenai *text summarization* pernah dilakukan oleh Itsnaini, dkk (2023) dengan judul *Abstractive Text Summarization using Pre-trained* Language Model *Text-to-text transfer transformer* dimana penulis menggunakan metode abstraksi dengan model *Text-to-text transfer transformer* dalam *text summarization* berita *online*. Penelitian tersebut mengumpulkan *dataset* sebanyak 9,387 data dari total 18,774 berita berbahasa indonesia dari IndoSum. Hasil dari penelitian menjelaskan bahwa model T5 efektif dalam peringkasan teks abstraktif dimana skor ROUGE-1 sebesar 0,387, skor ROUGE-2 sebesar 0,174 dan skor ROUGE-L sebesar 0,334. Namun Penelitian tidak membandingkan kinerja model T5 dengan model atau metode lain yang dapat memberikan pemahaman yang lebih baik tentang efektivitas model T5.

Penelitian lainnya juga dilakukan oleh Fitriana dan Jauhari (2022) dengan judul *Extractive Text Summarization For scientific journal article using long short term memory and gated recurrent units* dimana penulis menggunakan metode ekstraksi dengan model LSTM dalam *text summarization* Jurnal atau artikel dengan mengumpulkan *dataset* sebanyak 41.000 jurnal *scientific* dalam format PDF. Penelitian ini menghasilkan performa akurasi yang sangat mirip, dengan akurasi sekitar 98,40% dan 98,68% berturut-turut. Namun penelitian ini juga tidak membandingkan kinerja model LSTM dengan model atau metode lain yang dapat memberikan pemahaman yang lebih baik tentang efektivitas model LSTM ataupun metode ekstraksi.

Oleh karena itu, penelitian ini bertujuan untuk menganalisis algoritma *Text-to-text transfer transformer* berbasis abstraksi dan ekstraksi dalam *text summarization* berita *online*. Melalui analisis ini, pembaca dapat memahami sejauh mana algoritma NLP dapat menjaga integritas informasi, serta apakah ringkasan teks berbasis abstraksi atau ekstraksi lebih sesuai dalam menghadapi tantangan ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan, maka rumusan masalah pada penelitian ini yaitu:

1. Bagaimana penerapan algoritma *text-to-text transfer transformer* dalam menghasilkan ringkasan teks berita *online* berbasis abstraksi dan ekstraksi?
2. Bagaimana kinerja algoritma *text-to-text transfer transformer* berdasarkan nilai ROUGE dan BLEU dalam menghasilkan ringkasan teks berita *online* berbasis abstraksi dan ekstraksi?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan dapat diperoleh tujuan dari penelitian ini yaitu:

1. Menganalisis penerapan algoritma *text-to-text transfer transformer* dalam menghasilkan ringkasan teks berita *online* berbasis abstraksi dan ekstraksi.
2. Menganalisis kinerja algoritma *text-to-text transfer transformer* berdasarkan nilai ROUGE dan BLEU dalam menghasilkan ringkasan teks berita *online* berbasis abstraksi dan ekstraksi.

1.4 Manfaat Penelitian

Melalui penelitian ini, diharapkan dapat memberikan manfaat antara lain:

1. Memberikan wawasan tentang kinerja algoritma *text-to-text transfer transformer* berbasis abstraksi dan ekstraksi dalam meringkas teks berita *online*.
2. Hasil penelitian berpotensi untuk dikembangkan menjadi aplikasi praktis *text summarization* khusus berita *online*, yang dapat dimanfaatkan oleh pembaca maupun platform berita *online*.

3. Secara umum penelitian ini dapat memperkaya kajian akademik terkait penerapan *text summarization* pada berita *online* dengan memanfaatkan *text-to-text transfer transforme*.

1.5 Ruang Lingkup

Ruang lingkup dalam penelitian ini adalah sebagai berikut:

1. Fokus pada berita *online* yang menggunakan Bahasa Indonesia.
2. *Penggunaan algoritma NLP yang sudah tersedia dan terdokumentasi yaitu dengan menggunakan algoritma text-to-text transfer transformer T5 untuk abstraksi dan ekstraksi*.

BAB II
TINJAUAN PUSTAKA

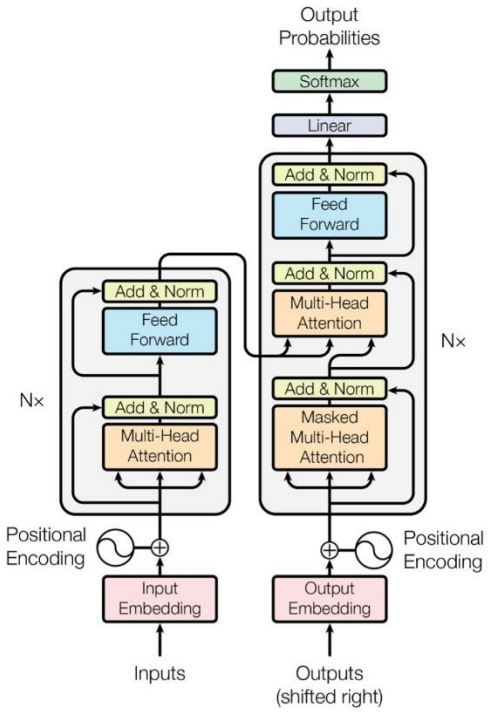
2.1 *Text Summarization*

Text summarization adalah proses menghasilkan ringkasan otomatis dari teks asli yang mempertahankan informasi penting dan relevan. Ada dua pendekatan utama dalam *text summarization*: abstraksi dan ekstraksi.

a. Abstraksi (*Abstractive Summarization*): Pendekatan ini melibatkan pemahaman dan pembuatan kembali teks dalam bahasa alami, menghasilkan ringkasan yang mungkin menggunakan kata-kata yang berbeda dari teks asli namun tetap menjaga makan utamanya (Lewis et al., 2020).
b. Ekstraksi (*Extractive Summarization*): Pendekatan ini melibatkan pemilihan kalimat atau frasa penting dari teks asli disusun menjadi ringkasan. Ringkasan yang dihasilkan biasanya terdiri dari potongan-potongan teks asli yang dipilih berdasarkan kriteria tertentu seperti frekuensi kata, posisi kalimat, atau skor keunikan (See, Liu, & Manning, 2017).

2.2 *Transformer Model*

Pendekatan *Transformer adalah salah satu inovasi terbesar dalam pemrosesan bahasa alami (NLP) yang memungkinkan berbagai tugas NLP diformulasikan dalam format "text-to-text". Pada dasarnya, model ini menerima teks sebagai konteks atau kondisi dan menghasilkan teks sebagai output. Pendekatan ini menyediakan tujuan pelatihan yang konsisten baik untuk pra-pelatihan maupun fine-tuning. Model dilatih dengan tujuan maksimum likelihood tanpa memperhatikan tugas spesifik yang sedang dilakukan (Vaswani et al., 2017)*.



Sumber: (Zolotareva, 2020)

Gambar 1. Model arsitektur *transformer*

Pada Gambar 1 pendekatan *Transformer* menggunakan *multi-head attention* untuk memungkinkan model berbagi informasi dari berbagai subruang representasi pada posisi yang berbeda. Dengan satu kepala perhatian, ini dicegah oleh perataan. Berikut adalah penjelasan rinci tentang arsitektur *transformer*:

a. *Encoder*

Encoder memetakan urutan *input* dari representasi simbol (x1, ..., xn) ke urutan representasi kontinu z = (z1, ..., zn). *Encoder* terdiri dari tumpukan N = 6 lapisan identik. Setiap lapisan memiliki mekanisme *multi-head self-attention* dan jaringan *feed-forward* sederhana yang sepenuhnya terhubung secara posisi. Koneksi residual digunakan di sekitar masing-masing dari dua sub-lapisan diikuti dengan normalisasi lapisan. *Output* dari setiap sub-lapisan adalah LayerNorm(x + Sublayer(x)), di mana Sublayer(x) adalah fungsi yang diimplementasikan oleh sub-lapisan itu sendiri (Vaswani et al., 2017).

b. *Decoder*

Decoder kemudian menghasilkan urutan *output* (y1, ..., ym) dari simbol satu elemen pada suatu waktu. *Decoder* juga terdiri dari tumpukan N = 6 lapisan identik. *Decoder* menambahkan sub-lapisan ketiga yang, selain dari

dua sub-lapisan, menyediakan *multi-head attention* ke *output* dari tumpukan *encoder*. Sama seperti *encoder*, koneksi residual di sekitar masing-masing dari dua sub-lapisan digunakan, diikuti dengan normalisasi lapisan. Untuk mencegah posisi memperhatikan posisi berikutnya, sub-lapisan *self-attention* yang dimodifikasi digunakan di *decoder* (Vaswani et al., 2017).

c. *Attention*

Fungsi perhatian dapat dijelaskan sebagai pemetaan *query* dan satu set pasangan *key-value* ke *output*, di mana *query*, *key*, *value*, dan *output* semuanya adalah vektor. *Output* dihitung sebagai jumlah berbobot dari *value*, di mana bobot yang diberikan pada setiap *value* dihitung oleh fungsi kompatibilitas dari *query* dengan *key* yang sesuai. Keuntungan menggunakan *multi-head attention* memungkinkan model untuk berbagi informasi dari representasi subspasi yang berbeda di posisi yang berbeda. Dengan satu kepala perhatian, ini dicegah oleh perataan (Vaswani et al., 2017).

d. *Encoder-decoder attention*

Query berasal dari lapisan *decoder* sebelumnya, dan memori *key* serta *value* berasal dari *output encoder*. Hal ini memungkinkan setiap posisi di *decoder* untuk memperhatikan semua posisi dalam urutan *input* (Vaswani et al., 2017; Wu et al., 2016).

e. Lapisan *self-attention*

Lapisan *self-attention* di *decoder* memungkinkan setiap posisi di *decoder* untuk memperhatikan semua posisi di *decoder* hingga dan termasuk posisi itu (Vaswani et al., 2017). Pendekatan ini memberikan fleksibilitas dan efisiensi dalam memproses urutan *input* dan menghasilkan urutan *output* yang lebih akurat.

2.3 *Attention*

Attention adalah mekanisme yang memungkinkan model untuk fokus pada bagian-bagian tertentu dari input saat menghasilkan output. Ini sangat penting dalam pemrosesan bahasa alami karena memungkinkan model untuk mempertimbangkan konteks yang relevan dan mengabaikan informasi yang tidak relevan (Vaswani et al., 2017).

Mekanisme perhatian diperkenalkan dalam konteks terjemahan mesin oleh Vaswani et al. (2017) melalui model Transformer. Attention dalam Transformer memungkinkan setiap bagian dari input (misalnya, kata dalam kalimat) untuk berinteraksi dan "memperhatikan" setiap bagian lainnya, menciptakan representasi kontekstual yang lebih kaya (Vaswani et al., 2017).

Attention bekerja dengan mengonversi tiga komponen utama: Query (Q), Key (K), dan Value (V). Setiap token dalam input dipetakan menjadi vektor Query, Key, dan Value. Mekanisme perhatian menghitung skor kesamaan antara Query dan Key untuk menentukan seberapa banyak perhatian yang harus diberikan pada setiap token dalam Value (Vaswani et al., 2017).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (1)$$

Di mana dkd_kdk adalah dimensi dari vektor Key, dan softmax digunakan untuk normalisasi skor kesamaan menjadi probabilitas.

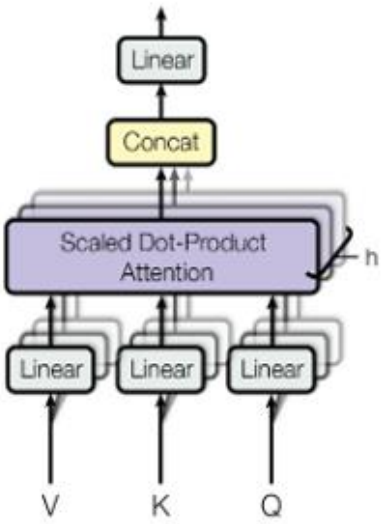
Untuk menangkap berbagai jenis hubungan semantik, T5 menggunakan multi-head attention. Ini berarti beberapa set Query, Key, dan Value digunakan secara paralel, dan hasilnya digabungkan . Setiap "head" dalam multi-head attention dapat menangkap aspek berbeda dari hubungan semantik dalam data (Vaswani et al., 2017; Raffel et al., 2020).

Dalam model T5, perhatian memungkinkan transformasi teks-ke-teks yang efektif dengan menangkap konteks dari input teks secara dinamis dan adaptif. Self-attention dalam encoder membantu membangun representasi kontekstual dari input, sedangkan cross-attention dalam decoder memastikan bahwa output yang dihasilkan mempertimbangkan informasi kontekstual dari input (Raffel et al., 2020).

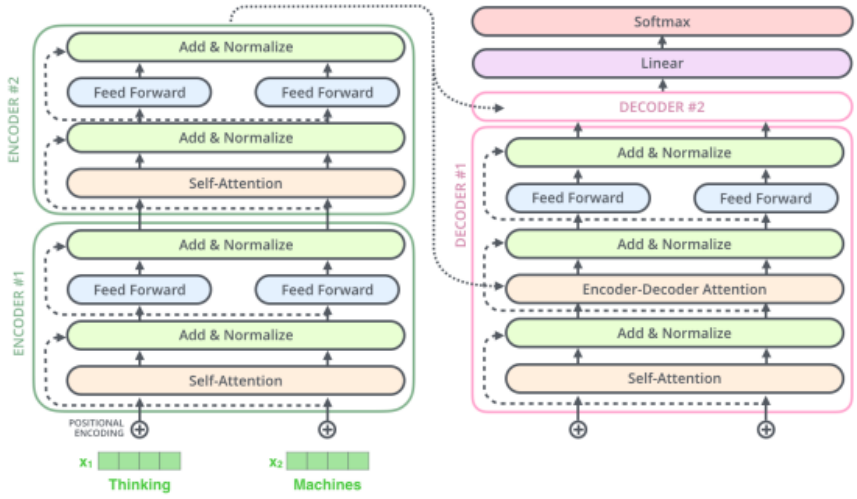
2.4 *Text-to-text Transfer Transformer (T5)*

Pendekatan *Transformer Transfer* Teks-ke-Teks (T5) adalah salah satu inovasi besar dalam pemrosesan bahasa alami (NLP) yang dikembangkan oleh Google pada tahun 2020. Google AI juga mengembangkan *Colossal clean crawled*

corpus (C4) sebagai *dataset* pra-pelatihan. T5 kemudian dilatih ulang pada C4 untuk tujuan *denoising* dan *corrupting* span dengan arsitektur *Encoder-decoder*. Setelah itu, model ini di-*fine-tuning* untuk tugas *downstream* dengan menggunakan tujuan terawasi (Raffel et al., 2020).



(a)



(b)

Sumber: (Zolotareva, 2020)

Gambar 2. Arsitektur t5: (a) *multi head attention;* (b) arsitektur t5 untuk *summarization*

Gambar 2 menunjukkan arsitektur dari T5 dimana terdapat faktor pembeda utama untuk arsitektur yaitu "*mask*" yang digunakan oleh berbagai mekanisme

perhatian dalam model. Operasi *self-attention* dalam *Transformer* menerima urutan sebagai *input* dan menghasilkan urutan baru dengan panjang yang sama. Setiap entri dari urutan *output* dihasilkan dengan menghitung rata-rata berbobot dari entri urutan *input*. Secara khusus, biarkan y_i merujuk pada elemen ke-i dari urutan *output* dan x_j merujuk pada entri ke-j dari urutan *input* (Zolotareva, 2020).

Dalam praktiknya, fungsi perhatian dihitung pada satu set *query* secara simultan, dikemas bersama menjadi matriks Q. *Key* dan *value* juga dikemas bersama menjadi matriks K dan V. Matriks *output* dihitung sebagai:

$$y_i = \sum_j w_i, x_j \quad (2)$$

di mana,

$YiYi$ adalah output dari unit ke-i,

wi,jwi,j adalah bobot antara unit ke-i dan input ke-j,

$xjxj$ adalah nilai dari input ke-j,

$\sum j\sum j$ menunjukkan penjumlahan di atas semua input jj,

Encoder pada model T5 terdiri dari blok-blok dengan lapisan *self-attention* dan jaringan *feedforward* yang *input*nya telah dinormalisasi menggunakan *layer normalization* (Ba et al., 2016). Versi *layer normalization* yang lebih sederhana digunakan, di mana aktivasi hanya di-*rescale* dan dihitung tanpa bias tambahan. Setelah itu, residual *skip connection* diperkenalkan untuk memetakan *input* ke *output* (He et al., 2016). *Dropout* diterapkan pada jaringan *feedforward*, *skip connection*, bobot *attention*, dan *input*/*output* dari seluruh *stack*. *Decoder* hampir serupa dengan tambahan lapisan *attention*. *Self-attention* dalam *decoder* memungkinkan model untuk memanfaatkan *output* sebelumnya. *Output* akhir dari *decoder* melewati *dense softmax output*. Dengan demikian, semua tugas berada dalam format teks-ke-teks. Kerangka ini memungkinkan penggunaan model yang sama untuk berbagai tugas (Pal, Kanupriya, Fan, & Igodifo, 2024).

Transformer encoder-decoder terdiri dari dua lapisan *stack*: *encoder*, yang diberi urutan *input*, dan *decoder*, yang menghasilkan urutan *output* baru. *Encoder* menggunakan masker "*fully visible*", yang memungkinkan mekanisme *self-attention* untuk memperhatikan setiap *input* dari *output*-nya. Bentuk masking ini

cocok ketika perhatian berada di atas "*prefix*", yaitu konteks yang diberikan ke model yang kemudian akan digunakan untuk membuat prediksi. Operasi *self-attention* dalam *decoder transformer* menggunakan pola *masking* "*causal*". Pendekatan dengan *mask* "*causal*" memungkinkan *decoder* mencegah model dari memperhatikan entri ke-j saat menangani urutan *input* ke-i untuk $j > i$. Ini digunakan selama pelatihan sehingga model tidak dapat "melihat masa depan" saat menghasilkan *output*-nya (Raffel et al., 2019).

Panjang maksimum ringkasan ditetapkan sepertiga dari panjang teks asli. Karena terdapat batasan komputasi maka digunakan T5-*small*. T5-*small* memiliki 6 blok (setiap blok terdiri dari *self-attention*, *attention encoder-decoder*, dan *network feedforward*). *Network feedforward* dalam setiap blok memiliki lapisan *dense* dengan 2048 dimensi *output*, diikuti oleh ReLU dan lapisan dense lainnya. Matriks "*key*" dan "*value*" dari semua mekanisme *attention* memiliki dimensi dalam 64 dan semua mekanisme perhatian memiliki 8 *head*. Semua sub-lapisan dan *embeddings* lainnya memiliki dimensi 512. Oleh karena itu, model ini memiliki sekitar 60 juta parameter. Parameter *early-stopping* diatur sebagai "True" (Pal, Kanupriya, Fan, & Igodifo, 2024).

2.5 Teknik *Scraping* dengan BeautifulSoup4

Teknik *scraping* adalah proses mengumpulkan data dari situs web. Salah satu alat yang popular digunakan untuk *scraping* dalam BeautifulSoup4, sebuah Pustaka Python yang mempermudah parsing halaman HTML dan XML (Richardson, 2013). BeautifulSoup4 memungkinkan pengguna untuk menavigasi, mencari, dan memodifikasi parse tree dari halaman web yang diambil menggunakan HTTP requests (Mitchell, 2018). Teknik ini biasanya digunakan untuk mengumpulkan artikel berita dari berbagai situs web berita untuk diolah lebih lanjut dalam penelitian *text summarization* (Richardson, 2013).

2.6 IndoSum

Data yang diambil dari dataset terdiri dari 3 bagian yaitu 'gold labels', 'paragraphs', dan 'summary'. Dataset berita berbahasa Indonesia ini berasal dari penelitian yang dilakukan oleh Kurniawan dan Louvan (Liu & Lapata, 2019).

"Dataset ini berisi sekitar 20 ribu artikel berita. Setiap artikel memiliki judul, kategori, sumber (misalnya, CNN Indonesia, Kumparan), URL ke artikel asli, dan ringkasan abstraktif yang dibuat secara manual oleh dua penutur asli bahasa Indonesia." (Kurniawan & Louvan, 2018). Dataset ini memiliki format json dan dibagi menjadi 3 bagian yaitu train, test, dan dev. Dataset ini mencakup 18.774 berita. Bagian train digunakan untuk proses pelatihan, test untuk pengujian, dan dev untuk evaluasi pelatihan. Sebelum data diimplementasikan ke dalam program, data perlu diolah terlebih dahulu untuk memudahkan proses implementasi. Beberapa langkah akan dilakukan dalam pengolahan data. Pertama, mengambil bagian 'gold labels', 'paragraphs', dan 'summary' dari dataset. Selanjutnya, membuka array pada setiap bagian 'paragraphs', 'gold labels', dan 'summary' untuk mengambil isinya. Terakhir, isi dari masing-masing bagian akan digabungkan dengan delimiter '<q>'. Pada bagian 'paragraphs' dan 'summary', delimiter ini digunakan untuk memisahkan setiap kalimat, sementara pada 'gold labels' digunakan untuk memisahkan label dari setiap kalimat yang akan digunakan sebagai referensi ekstraktif (Liu & Lapata, 2019).

2.7 *Torch* dan Pytorch

Torch adalah *library* komputasi ilmiah yang mendukung *machine learning* dengan kemampuan untuk menjalankan algoritma *neural network*. *Pytorch*, penerus dari *Torch*, adalah *library* yang lebih modern dan popular untuk penelitian pengembangan dalam *deep learning* (Paszke et al., 2019). *Pytorch* menyediakan *tensor computation* (seperti NumPy) dengan akselerasi GPU dan mendukung *dynamic computation graphs* yang memudahkan penelitian dan pengembangan model *machine learning* (Paszke et al., 2019). *Pytorch* digunakan secara luas dalam penelitian NLP untuk melatih model-model seperti T5. *Pytorch* menyediakan modul-modul seperti *torch*.nn untuk membangun *neural networks* dan *torch.optim* untuk optimisasi model (Paszke et al., 2019).

2.8 *Pre-processing*

Proses *preprocessing* adalah langkah awal yang sangat penting dalam *text summarization* dan tugas-tugas pemrosesan bahasa alami lainnya. *Preprocessing*

melibatkan berbagai teknik yang digunakan untuk mempersiapkan teks mentah sebelum dimasukkan ke dalam model *machine learning*. Beberapa tahapan utama dalam *preprocessing* adalah *filtering*, *case folding*, *stopword and numbers removal*, normalisasi, dan *tokenizing*.

a. *Filtering*

Filtering adalah langkah awal dalam *preprocessing* yang bertujuan untuk menghilangkan karakter atau simbol yang tidak relevan dari teks. Karakter-karakter ini bisa termasuk tanda baca, simbol-simbol khusus, atau elemen HTML dalam teks yang diambil dari web. *Filtering* membantu dalam mengurangi kebisingan dalam data dan memfokuskan pada konten yang relevan (Manning et al., 2008).

b. *Case folding*

Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil. Tujuan dari *case folding* adalah untuk memastikan bahwa perbandingan antara kata-kata tidak peka terhadap huruf besar atau kecil. Misalnya, kata "AI" dan "ai" akan dianggap sama setelah *case folding*. Hal ini penting untuk konsistensi dan untuk mengurangi jumlah fitur unik dalam data (Jurafsky & Martin, 2020).

c. *Stopwords removal*

Stopword adalah kata-kata umum yang sering muncul dalam teks tetapi tidak memberikan informasi penting untuk analisis, seperti "dan", "atau", "adalah", dan sebagainya. Proses ini juga melibatkan penghapusan angka yang mungkin tidak relevan untuk analisis tertentu. Menghapus *stopword* dan angka membantu dalam mengurangi dimensi data dan fokus pada kata-kata yang lebih bermakna (Manning et al., 2008).

d. Normalisasi

Normalisasi adalah proses mengubah variasi bentuk kata ke bentuk dasar yang konsisten. Ini termasuk mengubah kata-kata dengan imbuhan atau variasi morfologis ke bentuk dasarnya. Contohnya, kata-kata seperti "berlari", "berlari-lari", dan "lari" akan diubah menjadi "lari". Normalisasi membantu

dalam mengkonsolidasikan variasi kata yang berbeda menjadi satu representasi yang standar (Jurafsky & Martin, 2020).

e. *Tokenizing*

Tokenizing adalah proses membagi teks menjadi unit-unit yang lebih kecil yang disebut token. Token ini biasanya berupa kata, frasa, atau kalimat. *Tokenizing* adalah langkah dasar yang memungkinkan analisis lebih lanjut seperti menghitung frekuensi kata, n-gram, dan lain-lain. Proses ini sangat penting untuk mempersiapkan data teks untuk model NLP yang memerlukan *input* dalam bentuk token (Manning et al., 2008).

2.9 Evaluasi Metrik

Evaluasi kualitas ringkasan otomatis adalah aspek penting dalam *text summarization*. Dua metrik yang paling umum digunakan untuk mengevaluasi kinerja model *summarization* adalah ROUGE dan BLEU.

a. ROUGE (*Recall-oriented understudy for gisting evaluation*)

ROUGE adalah keluarga metrik evaluasi yang membandingkan n-gram, LCS (*Longest common subsequence*), dan penghitungan token antar teks yang dihasilkan dengan referensi ringkasan manusia. Metrik ini termasuk ROUGE-N, ROUGE-L, dan ROUGE-W, yang masing-masing mengevaluasi berdasarkan n-gram, LCS, dan panjang kata (Lin, 2004). ROUGE sering digunakan untuk mengevaluasi kinerja model *summarization* karena kemampuannya mencerminkan keberhasilan model dalam menangkap informasi penting dari teks asli (Lin, 2004).

b. BLEU (*Bilingual evaluation understudy*)

BLEU awalnya dikembangkan untuk mengevaluasi terjemahan mesin tetapi juga digunakan untuk evaluasi *text summarization*. BLEU mengukur kesamaan antara teks yang dihasilkan dan referensi dengan menghitung n-gram presisi, yang memberikan penilaian tentang seberapa mirip teks yang dihasilkan dengan referensi manusia (Papineni et al., 2002). BLEU terutama berguna untuk mengukur akurasi pada tingkat kata atau frasa kecil tetapi kurang efektif dalam menangkap konteks dan koherensi teks yang lebih panjang (Papineni et al., 2002).

2.10 Penelitian Terkait

Berikut penelitian terkait yang dijadikan bahan referensi dalam penelitian ini seperti pada Tabel 1 berikut.

Tabel 1. Penelitian terkait

Judul	**Penulis**	**Metode**	**Hasil**
Abstractive Text Summarization using Pre-trained Language Model "*Text-to-text transfer transformer"*	Qurrota A'yuni Itsnaini, Mardhiya Hayaty, Andriyan Dwi Putra, Nidal A.M Jabari (2023)	Model *text-to-text transfer transformer* (T5) dengan *dataset* sebanyak 9,387 data dari total 18,774 berita berbahasa indonesia dari IndoSum dengan data pelatihan 90% = 8,448 dan data pengujian 10% = 939 serta 10% dari data pelatihan digunakan untuk validasi data Evaluasi menggunakan matriks ROUGE.	Model T5 efektif dalam peringkasan teks abstraktif dimana skor ROUGE-1 sebesar 0,387, skor ROUGE-2 sebesar 0,174 dan skor ROUGE-L sebesar 0,334
Extractive text summarization for scientific journal article using long short term memory and gated recurrent units	Devi Fitrianah, Raihan Nugroho Jauhari (2022)	Model long short term *memory* dan gated reccurent units. Membangun *dataset* sebanyak 41.000 jurnal *scientific* dalam format PDF Evaluasi dilakukan dengan menghitung	Metode baru yang menggabungkan one-hot *encoding* untuk setiap kalimat dan rekayasa fitur untuk mengekstrak makna semantik menghasilkan ringkasan ekstraktif yang efektif, setara dengan hasil yang dicapai oleh algoritma LSA. Metode ini menggunakan pendekatan *deep learning*,

Judul	Penulis	Metode	Hasil
		akurasi dengan mendapatkan nilai presisi, *recall* dan nilai F-measure	yang memungkinkan penyesuaian dengan mengubah metode pelabelan data, seperti pengelompokan atau metode manual. Model GRU dan LSTM memberikan performa akurasi yang sangat mirip, dengan akurasi sekitar 98,40% dan 98,68% berturut-turut. LSTM memiliki sedikit keunggulan karena memiliki arsitektur gerbang yang lebih kompleks, sementara GRU memiliki keunggulan dalam waktu yang dibutuhkan dalam proses pelatihan karena setiap sel hanya memiliki dua gerbang. Penyetelan parameter juga berdampak positif, meningkatkan akurasi dari 98,33% menjadi 98,68% untuk model LSTM. Nilai konfigurasi terbaik pada parameter penyetelan termasuk ukuran batch=32, *epoch* 100, tingkat pembelajaran=0,001, optimisasi Adam, dan fungsi aktivasi neuron hard sigmoid
Implementasi Metode *Recurrent Neural Network* pada *Text Summarization* dengan Teknik abstraktif	Kasyfi Ivanedra, Metty Mustikasari (2019)	Metode recurrent *neural network* dengan jenis RNN long short term *memory*. Teknik pengolahan data menggunakan pra-proses data.	Sistem mampu menghasilkan ringkasan yang lebih alami dan lebih relevan secara kontekstual dibandingkan dengan Teknik ekstraktif. Hasil rata-rata dari berita tanpa stemming (penyederhanaan kata) adalah: *Recall* 41%, presisi 81%, F-measure 54,27%

Judul	Penulis	Metode	Hasil
		Dataset yang digunakan adalah data artikel berita dengan jumlah sebanyak 4515 buah artikel. Pengujian performa menggunakan *precision, recall* dan f-measure dengan membandingkan ringkasan yang dibuat oleh manusia.	Hasil dengan Teknik *stemming*: *recall* 44%, presisi 88%, dan F-measure 58,20%.
Implementasi Peringkas Dokumen Berbahasa Indonesia Menggunakan Metode *Text* To *Text Transfer Transformer* (T5)	I Nyoman Purnama, Ni Nengah Widya Utami (2023)	Menggunakan *dataset* berita bahasa Indonesia INDOSUM. Dilakukan *preprocessing* data. Kemudian meringkas menggunakan model T5 dengan 3 skenario berbeda. Hasilnya dievaluasi dengan ROUGE.	Skenario 2 dengan implementasi stemming tanpa *stopwords* removal mendapatkan hasil ROUGE-1 terbaik yaitu 0.17568.