

## PERINGKASAN TEKS OTOMATIS ABSTRAKTIF MENGUNAKAN TRANSFORMER PADA TEKS BAHASA INDONESIA

Andika Bahari<sup>1</sup>, Kania Evita Dewi<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Universitas Komputer Indonesia

Jl. Dipati Ukur no.112-116, Lebakgede, Kecamatan Coblong, Kota Bandung, Jawa Barat 40132

E-mail: kania.evita.dewi@email.unikom.ac.id<sup>2</sup>

### Abstrak

Peringkasan teks abstraktif digunakan untuk menghasilkan ringkasan yang mirip dengan ringkasan buatan manusia. Untuk mencapai kemampuan tersebut, biasanya diterapkan arsitektur *deep learning* yang bersifat *recurrent*, seperti RNN, LSTM, dan GRU. Pada penelitian sebelumnya yang membahas peringkasan teks abstraktif dalam bahasa Indonesia, model *recurrent* banyak digunakan dan terdapat masalah pada kohesi dan tata bahasa dari ringkasan yang dihasilkan oleh model, sehingga hal ini bisa berdampak terhadap performa. Saat ini, terdapat arsitektur yang lebih baru bernama Transformer yang mengandalkan mekanisme *attention* secara keseluruhan. Karena sifatnya yang tidak *recurrent*, Transformer mengatasi masalah ketergantungan terhadap deretan *hidden state* yang terjadi pada model *recurrent* dan bisa mempertahankan informasi pada seluruh *input sequence*. Dengan demikian, pada penelitian ini, penulis menggunakan Transformer untuk mengevaluasi seberapa baik Transformer dalam melakukan peringkasan teks abstraktif dalam bahasa Indonesia. Pelatihan dilakukan menggunakan *pre-trained model* T5 dengan *dataset* Indosum yang berisi sekitar 19 ribu pasangan berita dan ringkasan. Skor evaluasi paling tinggi yang berhasil dicapai adalah ROUGE-1 sebesar 0.61 dan ROUGE-2 sebesar 0.51.

**Kata kunci:** peringkasan teks abstraktif, Transformer, T5, Indonesia

### Abstract

*Abstractive text summarization is used to generate summaries that are similar to human-written summaries. To achieve that capability, recurrent deep learning architectures were usually applied, such as RNN, LSTM, and GRU. In previous studies about abstractive text summarization in Indonesian, recurrent models were widely used and there were cohesion and grammar errors that appeared in the generated summaries—this could have an impact on performance. Currently, there is Transformer, a relatively new architecture that relies on the attention mechanism entirely. Due to its non-recurrent nature, Transformer overcomes the problem of dependency on the hidden states that occurs in recurrent models and it can retain information on all input sequences. In this study, we use Transformer to evaluate how good it is at abstractive text summarization in Indonesian. The training was conducted using the pre-trained T5 model with IndoSum dataset which contains around 19K news-summary pairs. We achieved evaluation scores of 0.61 ROUGE-1 and 0.51 ROUGE-2.*

**Keywords:** *abstractive text summarization, Transformer, T5, Indonesia*

## 1. PENDAHULUAN

Banyaknya dokumen yang tersebar di internet membutuhkan banyak riset di dalam ranah peringkasan teks otomatis [1]. Teks yang lebih pendek akan mengurangi waktu baca dan ukuran *file*. Sehingga, ini memberikan benefit baik bagi manusia maupun komputer. *Automatic text summarization* (peringkasan teks otomatis) adalah sebuah pendekatan untuk menghasilkan sebuah ringkasan yang pendek dengan mempertahankan informasi kunci dan keseluruhan makna [1]. Terdapat dua jenis metode untuk peringkasan teks otomatis, yaitu ekstraktif dan abstraktif. Metode ekstraktif meringkas dokumen dengan cara memilih bagian dari kata atau kalimat di dalam dokumen [2], metode ini bergantung pada ekstraksi kalimat-kalimat dari teks aslinya [1]. Sedangkan, metode abstraktif meringkas dokumen dengan cara membuat kalimat-kalimat baru yang mempunyai informasi yang sama dengan dokumen aslinya[2]. Namun, ada juga penelitian yang menggabungkan kedua metode tersebut [3].

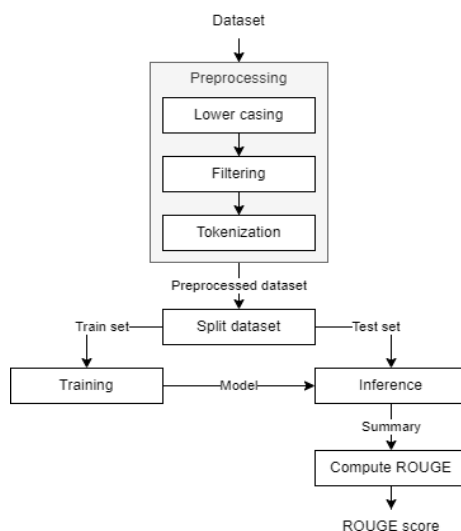
Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), dan Gated Recurrent Units (GRU) sebelumnya sudah pernah digunakan pada kasus peringkasan teks abstraktif dalam bahasa Indonesia [4][5][6]. Akan tetapi, ketergantungan terhadap deretan *hidden state* yang terjadi pada model *recurrent* menyebabkan masalah seperti menghalangi paralelisasi dan adanya *information loss* karena hasil komputasi  $h_t$  yang bergantung pada  $h_{t-1}$ . Rike Adelia dkk. [7] melakukan peringkasan teks abstraktif menggunakan Bidirectional Gated Recurrent Unit (BiGRU), salah satu arsitektur RNN, pada teks bahasa Indonesia. Dalam penelitiannya disebutkan bahwa model yang dibangun berhasil mempelajari kata-kata tunggal dari sumber teks aslinya. Namun, ringkasan yang dihasilkan masih terdapat kekurangan dalam kohesi dan tata bahasa.

Pada tahun 2017, A. Vaswani dkk. [8] membuat arsitektur bernama Transformer yang memanfaatkan mekanisme *self-attention*. Transformer berhasil mencapai skor BLEU sebesar 28,4 pada dataset WMT 2014 English-to-German dan 41,8 skor BLEU pada dataset WMT 2014 English-to-French, menjadikan Transformer sebagai *state of the art* yang baru pada tahun tersebut. Berdasarkan eksperimen tersebut, Transformer menunjukkan hasil yang bagus secara kualitas dan dapat melakukan paralelisasi sehingga membutuhkan waktu yang lebih sedikit untuk dilatih. Selain translasi, Transformer juga bisa digunakan pada peringkasan. Anushka dkk. [9] mengimplementasikan Transformer untuk melakukan peringkasan dengan menggunakan dataset yang berisikan kumpulan berita dari BBC. Pada penelitiannya, satuan pengukuran yang digunakan untuk mengukur kualitas ringkasan adalah ROUGE[10], dan berhasil mencapai skor ROUGE-1 sebesar 0.47 dan ROUGE-2 sebesar 0.33.

Berdasarkan performa yang dicapai oleh Transformer pada beberapa penelitian yang telah disebutkan, penulis tertarik untuk mengimplementasikan Transformer pada peringkasan teks abstraktif dalam bahasa Indonesia. Tujuan dari penelitian ini adalah mengukur seberapa baik performa Transformer pada peringkasan teks abstraktif dalam bahasa Indonesia. Adapun metode pengukuran yang digunakan pada penelitian ini adalah ROUGE untuk membandingkan ringkasan yang dihasilkan oleh model dengan referensi ringkasan yang dibuat oleh manusia.

## 2. METODOLOGI

Berikut ini adalah gambaran metodologi penelitian yang akan diterapkan untuk peringkasan teks abstraktif, ditunjukkan pada Gambar 1.



Gambar 1 Metodologi Penelitian

### 2.1 Dataset

IndoSum adalah dataset untuk peringkasan teks dalam bahasa Indonesia yang dikumpulkan dari artikel berita online[6]. Karena IndoSum tersedia secara publik, maka penelitian ini menggunakan dataset dari IndoSum yang berisikan 19 ribu pasangan berita dan ringkasan.

### 2.2 Preprocessing

*Preprocessing* adalah proses mengubah struktur teks agar sesuai dengan kebutuhan. Adapun metode *preprocessing* yang diterapkan adalah (1) *lower casing* untuk menyeragamkan semua huruf menjadi huruf

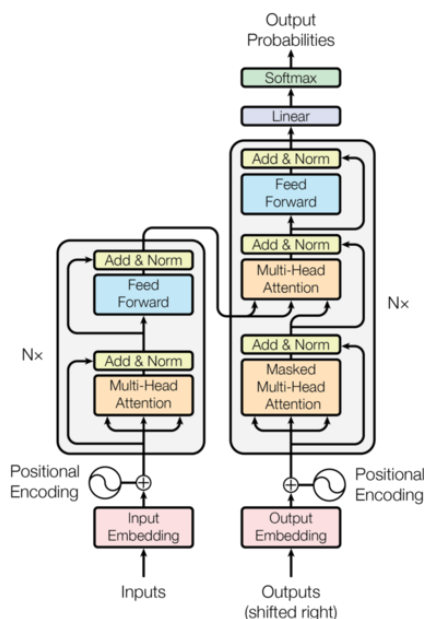
kecil, (2) *filtering* digunakan untuk menyaring teks yang memenuhi pola *regex*:  $([a-z0-9+, .\ \\/\ \ ] * [a-z0-9] [a-z0-9+ ] *)$ , dan (3) *tokenization* untuk mengubah teks menjadi bagian-bagian kecil (*token*). *Tokenization* dilakukan menggunakan SentencePiece [11] dengan metode *unigram tokenization* **Error! Reference source not found.** yang bisa mengurangi *out of vocabulary* karena satu kata tidak harus diubah menjadi satu *token*, melainkan bisa tersusun beberapa *token*. Sebelum diterapkan *tokenization*, teks berita diberikan *prefix* “summarize:” [12] terlebih dahulu. Hasil *tokenization* untuk setiap dokumen, baik berita maupun referensi ringkasan, menggunakan *token* “</s>” sebagai tanda akhir dokumen untuk memberi tahu algoritma *decoder* agar berhenti menghasilkan *token* ketika *inference*. Setiap *token* yang tidak terdaftar di dalam *vocab* akan dipetakan ke *token* “<unk>”.

**2.3 Split Dataset**

*Dataset* yang sudah melalui tahap *preprocessing* akan dipisah menjadi dua set, yaitu *train set* dan *test set*. *Train set* adalah data yang digunakan untuk pelatihan di mana model akan belajar berdasarkan sampel-sampel yang terkandung di dalamnya. Sedangkan, *test set* adalah data yang digunakan untuk pengujian di mana setiap sampel yang terkandung di dalam *test set* merupakan sampel yang belum pernah ditemui oleh model pada saat proses pelatihan, sehingga berguna untuk pengukuran performa.

**2.4 Transformer**

Transformer [8] adalah model *deep learning* dengan arsitektur *encoder-decoder*. *Encoder* menerima *token* berita sebagai masukan. Pada saat *training*, *decoder* menerima *token* label (referensi ringkasan) yang disisipkan *token* “<s>” di posisi paling depan, deretan *token* digeserkan ke sebelah kanan (*shifted right*) agar jumlah *token* yang masuk ke dalam *decoder* sama dengan jumlah *token* label. Sedangkan, pada saat *inference*, *decoder* menghasilkan ringkasan dengan melakukan iterasi untuk mengeluarkan prediksi *token* satu per satu sampai *decoder* menemukan *token* “</s>” atau sampai dengan batas iterasi yang ditentukan. Karena kasus dari penelitian ini adalah peringkasan, maka panjang ringkasan dibatasi maksimal 50% dari panjang dokumen aslinya. Arsitektur Transformer bisa dipelajari lebih detail pada [13].



**Gambar 2 Arsitektur Transformer**

**Encoder.** *Encoder* terdiri atas *N layer* di mana setiap *layer* memiliki *sublayer multi-head attention* dan *feed-forward network*. Setiap *sublayer* menerapkan *residual connection* dan *layer normalization* [14],  $LayerNorm(x + Sublayer(x))$ , di mana *x* adalah vektor *embedding* dan *Sublayer(x)* adalah fungsi yang diimplementasikan oleh *sublayer* [8].

**Decoder.** *Decoder* terdiri atas *N layer*. *Decoder* memiliki tiga *sublayer* di mana dua di antaranya merupakan *sublayer* yang sama dengan *sublayer* yang dimiliki oleh *encoder*, sedangkan satu *sublayer* lainnya merupakan *masked multi-head attention* [8].

**Embedding.** Transformer mengubah *token* menjadi vektor dengan dimensi  $d_{model}$ . Matriks bobot pada *embedding layer* dikalikan dengan  $\sqrt{d_{model}}$  [8].

**Positional Encoding.** Karena sifatnya yang tidak *recurrent*, Transformer menggunakan *positional encoding* untuk memberikan informasi mengenai posisi *token*. *Positional encoding* memiliki dimensi yang sama dengan vektor *input*,  $d_{model}$  [8].

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \tag{1}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \tag{2}$$

Di mana *pos* adalah posisi *token* dan *i* adalah dimensi vektor *input*.

**Self-Attention.** *Self-attention* adalah sebuah mekanisme untuk menghitung keterhubungan antarkata di dalam kalimat. Untuk menghitung nilai *attention*, vektor *input*  $x_i$  diproyeksikan menjadi *query* (*Q*), *key* (*K*), dan *value* (*V*). *Query* dan *key* memiliki dimensi  $d_k$ , sedangkan *value* memiliki dimensi  $d_v$  [8].

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{3}$$

**Multi-Head Attention.** Alih-alih menghitung satu nilai *attention* dengan  $d_k$  sama dengan  $d_{model}$ , *multi-head attention* memungkinkan nilai *attention* untuk bisa dihitung sebanyak *h* kali dengan  $d_k = d_v = d_{model}/h$  [8].

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \tag{4}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

Di mana matriks bobot  $W_i^Q \in R^{d_{model} \times d_k}$ ,  $W_i^K \in R^{d_{model} \times d_k}$ ,  $W_i^V \in R^{d_{model} \times d_v}$ , dan  $W_i^O \in R^{hd_v \times d_{model}}$ .

**Feed-Forward Network.** Setiap *layer* di dalam *encoder* dan *decoder* memiliki *feed-forward network* yang terdiri atas dua transformasi linear dengan fungsi aktivasi ReLU [15] di antara keduanya. Dimensi dari *input* dan *output* adalah  $d_{model}$ , sedangkan *inner-layer* memiliki dimensi  $d_{ff}$  [8].

$$FFN(x) = max(0, xW_1 + b_1)W_2 + W_2 \tag{6}$$

## 2.5 ROUGE

ROUGE-N adalah satuan yang mengukur jumlah n-gram yang cocok antara teks yang dihasilkan oleh model dengan referensi ringkasan yang dibuat oleh manusia [10]. Penelitian ini menggunakan ROUGE-1 (*unigram*) untuk mengukur kecocokan per satu kata dan ROUGE-2 (*bigram*) untuk mengukur kecocokan per dua kata.

$$ROUGE-N = \frac{\sum_{S \in \{References\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{References\}} \sum_{gram_n \in S} Count(gram_n)} \tag{7}$$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Konfigurasi Pelatihan

Transformer membutuhkan pelatihan sebelum dapat meringkas teks. Pada penelitian ini pelatihan dilakukan pada *platform* Google Colab menggunakan GPU NVIDIA T4 dengan *memory* sebesar 15 GB. Fitur yang digunakan didalam penelitian ini berupa potongan kata atau bisa jadi sebuah kata. Untuk membentuk fitur yang akan digunakan dalam peringkasan menggunakan dua jenis *vocab* yang digunakan, *vocab* yang dimiliki oleh T5 [12] (non-Indonesia) dan idT5 [16] (Indonesia). T5 memiliki *vocab* yang tidak mengandung bahasa Indonesia dengan jumlah suku kata sebanyak 32 ribu, sedangkan idT5 memiliki *vocab* yang mengandung bahasa Indonesia dengan jumlah suku kata sebanyak 30 ribu. Pelatihan menggunakan *pre-trained model* T5 [12] dengan parameter  $d_{model} = 512$ , 6 *layer encoder-decoder*, 8 *attention head*,  $d_{kv} = 64$ , dan  $P_{dropout} = 0.1$ . Terdapat dua jenis model dengan ukuran *token embeddings* yang disesuaikan dengan *vocab* yang digunakan (non-Indonesia dan Indonesia), ditunjukkan pada Tabel 1.

Tabel 1 Model

Model	Vocab	Token embeddings
T5-default	Default (non-Indonesia)	32 ribu
T5-id	Indonesia	30 ribu

Pelatihan dilakukan menggunakan komposisi 80% *train set* dan 20% *test set* [5]. *Optimizer* yang digunakan adalah *optimizer* Adam [17] dengan konfigurasi  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1e-8$ , dan *learning rate*:  $1e-3$ ,  $1e-4$ , dan  $1e-5$  [16][17][18]. Baik T5-default maupun T5-id, keduanya dilatih menggunakan

konfigurasi yang sama dengan 3 *epoch*, untuk membandingkan performa dari kedua model tersebut pada kondisi yang sama [16].

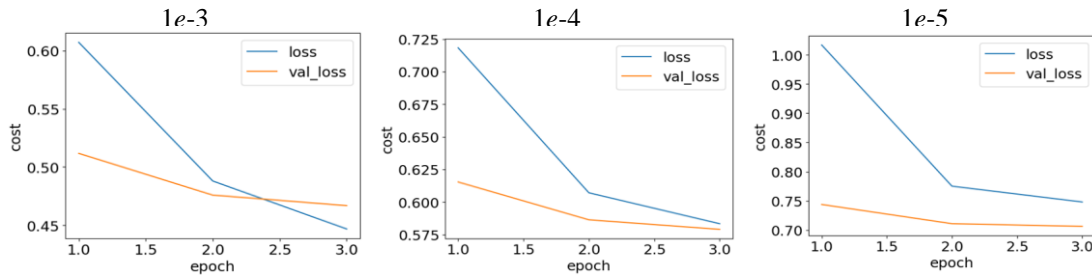
### 3.2 Hasil Pelatihan

Dalam proses pelatihan karena terdapat dua *vocab* yang berbeda, non-Indonesia dan Indonesia, akan terjadi perbedaan jumlah *token* yang dihasilkan oleh *tokenizer*. Tabel 2 menunjukkan perbandingan jumlah *token* yang dihasilkan oleh *tokenizer* dengan *truncation* sama dengan  $d_{model}$ . Penggunaan *vocab* Indonesia memberikan benefit dalam mengurangi jumlah *token*. Jumlah *token* yang lebih sedikit mengurangi jumlah vektor yang masuk ke dalam *neural network*, sehingga ini bisa mengurangi beban komputasi dan konsumsi *memory*.

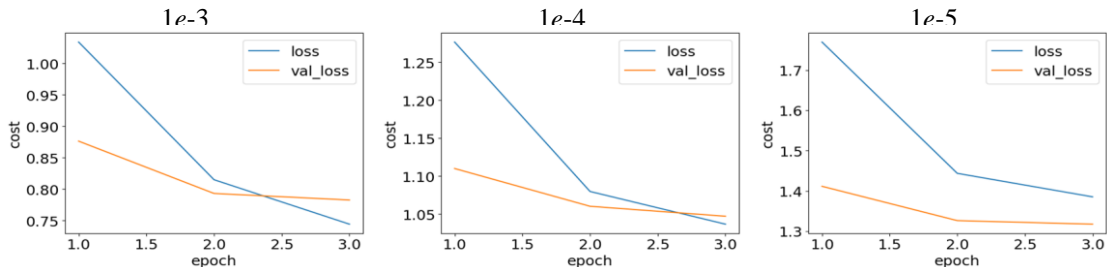
Tabel 2 Jumlah Token

Vocab	Jumlah token ( $10^6$ )
Default (non-Indonesia)	9.47
Indonesia	8.22

Gambar 3 dan Gambar 3 menunjukkan grafik *loss* dari pelatihan yang telah dilakukan. Terlihat bahwa T5-id memiliki *validation loss* yang lebih tinggi dari T5-default, hal ini terjadi karena model T5 sebelumnya dilatih menggunakan *vocab*-nya tersendiri. Sehingga, mengganti *vocab* dapat menyebabkan *loss* menjadi lebih tinggi.



Gambar 3 Grafik Loss (T5-default)



Gambar 34 Grafik Loss (T5-id)

Namun, mengganti *vocab* juga bisa memberikan benefit tertentu. Karena jumlah *token* yang lebih sedikit, T5-id memiliki waktu pelatihan yang lebih cepat dan konsumsi *memory* yang lebih rendah—lihat Tabel 3.

Tabel 3 Waktu Pelatihan dan Konsumsi Memory

Model	Learning rate	Waktu pelatihan	Konsumsi memory
T5-default	1e-3	2904 detik	14.7 GB
	1e-4	2852 detik	
	1e-5	2854 detik	
T5-id	1e-3	2018 detik	10.6 GB
	1e-4	2027 detik	
	1e-5	2023 detik	

### 3.3 Hasil Pengujian

ROUGE mengukur jumlah kata yang cocok antara ringkasan yang dihasilkan oleh model dan ringkasan yang dibuat oleh manusia. Karena judul dan isi ringkasan bisa ditulis dengan berbeda, skor ROUGE sebesar 0.5 dianggap sebagai hasil cukup tinggi [19]. Tabel 3 menunjukkan klasifikasi mutu ringkasan berdasarkan skor ROUGE-1 menurut [19].

**Tabel 3 Klasifikasi Mutu**

Mutu	ROUGE-1
A+	> 0.48
A	> 0.45
B	> 0.40
C	> 0.35
F	≤ 0.35

Berdasarkan klasifikasi mutu yang sudah ditentukan, berikut ini persentase dari jumlah ringkasan yang dikelompokkan berdasarkan mutu, ditunjukkan pada Tabel 4.

**Tabel 4 Jumlah Ringkasan Berdasarkan Mutu**

Model	Learning rate	Jumlah (%)				
		A+	A	B	C	F
T5-default	1e-3	49.43	10.28	13.56	10.60	16.14
	<b>1e-4</b>	<b>51.72</b>	8.31	11.93	9.69	18.35
	1e-5	24.79	6.44	12.17	12.46	44.13
T5-id	1e-3	72.38	4.71	5.81	5.81	11.29
	<b>1e-4</b>	<b>75.13</b>	4.66	5.43	4.15	10.63
	1e-5	73.93	4.42	5.89	5.17	10.60

Adapun hasil pengukuran rata-rata skor ROUGE-1 dan ROUGE-2 yang ditunjukkan pada Tabel 5.

**Tabel 5 Skor Evaluasi**

Model	Learning rate	ROUGE-1	ROUGE-2
T5-default	1e-3	0.47	0.32
	<b>1e-4</b>	<b>0.48</b>	<b>0.34</b>
	1e-5	0.37	0.30
T5-id	1e-3	0.59	0.49
	<b>1e-4</b>	<b>0.61</b>	<b>0.51</b>
	1e-5	0.60	0.50

Berdasarkan Tabel 4 dan Tabel 5, model yang memiliki performa paling tinggi adalah model yang dilatih dengan *learning rate* 1e-4 ditunjukkan dengan (1) jumlah ringkasan terbanyak dan masuk kategori A+ adalah 51.72% untuk T5-default dan 75.13% untuk T5-id, dan (2) skor evaluasi tertinggi adalah 0.48 ROUGE-1 dan 0.34 ROUGE-2 untuk T5-default; dan 0.61 ROUGE-1 dan 0.51 ROUGE-2 untuk T5-id.

Tingginya skor ROUGE dipengaruhi oleh *vocab* yang bisa berpengaruh terhadap performa dengan cukup signifikan. Meskipun *tokenizer* bisa saja menggunakan *vocab* yang sudah ada (non-Indonesia), karena sebagian besar suku kata yang digunakan pada bahasa Inggris juga digunakan pada bahasa Indonesia [16], namun penggunaan *vocab* yang mengandung bahasa Indonesia nyatanya menghasilkan peningkatan performa yang cukup tinggi karena *vocab* tersebut lebih relevan dengan bahasa yang digunakan.

Untuk meninjau struktur kalimat dari ringkasan yang dihasilkan oleh model, berikut ini hasil ringkasan oleh model menggunakan salah satu sampel yang diambil dari data uji, ditunjukkan pada Tabel 6.

**Tabel 6 Sampel Prediksi Ringkasan**

Teks berita		
rimanews penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat kpk baru saja menggeledah rumah di kawasan rafles hills tapos depok atas nama agus nugroho yang menjadi perantara ashanti sales inc ke pejabat pt pal tim satgas kpk menggeledah sejak pukul 09.00 wib hingga pukul 12.30 wib dari lokasi tim menyita dokumen kata juru bicara kpk feabri diansyah di gedung kpk jalan kuningan persada kuningan jakarta hari ini menurut mantan aktifis icw itu rumah di kawasan rafles hills tersebut merupakan tempat kesepuluh dari rangkaian pengeledahan yang telah dilakukan sejak sabtu lalu sebelumnya tim penyidik telah menggeledah lima lokasi di surabaya dan empat lokasi lainnya di jakarta dari sembilan lokasi pengeledahan sebelumnya tersebut disita sejumlah dokumen barang bukti elektronik dan uang tegasnya		
Model	Learning rate	Prediksi ringkasan
T5-default	1e-3	penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat baru saja menggeledasi perantara ashanti sales inc
	1e-4	penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat baru saja rumah ini mengadi perantara ashanti sales inc kesepuluh di gedung karena di jakarta
	1e-5	penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat baru saja rumah di di jakarta dari sembilan dikawasan rafles hills tapos depok atas nama nugroho yang menjadi perantara ashanti sales inc kes pejabat ppt tim minggu mengaku karena sejak pukul 09.00 wib hingga pengawab dan agus pembawa
T5-id	1e-3	rimanews penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat k gila baru saja menggeledah rumah di kawasan rafles hills tapos depok atas nama agus nugroho yang menjadi perantara ashanti sales inc ke pejabat Temple pal tim satgas k You Mert FL
	1e-4	penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat k gila baru saja menggeledah rumah di kawasan rafles hills tapos depok atas nama agus nugroho yang menjadi perantara ashanti sales inc ke pejabat Temple pal tim satgas kbitur mengge mindedah sejak pukul 09.00 wib hingga p100 12.30 w Wer dari lokasi tim menyita dokumen
	1e-5	penyidik kpk masih terus mengumpulkan barang bukti kasus pemberian fee agency penjualan kapal pt pal ke filipina dengan cara melakukan pengeledahan di berbagai tempat k gila baru saja menggeledah rumah di kawasan rafles hills tapos depok atas nama agus nugroho yang menjadi perantara ashanti sales inc ke pejabat awan pal tim satgas

Tabel 6 menunjukkan bahwa kedua jenis model mampu menghasilkan ringkasan struktur kalimat yang cukup bisa dipahami. Meskipun T5-id memiliki skor evaluasi yang lebih tinggi, ternyata T5-default menghasilkan ringkasan dengan struktur kalimat yang lebih konsisten. Karena pelatihan dilakukan dengan *transfer learning*, hal tersebut mungkin dipengaruhi oleh model T5 yang sebelumnya memang dilatih menggunakan *vocab*-nya sendiri.

Berikut ini adalah data skor evaluasi peringkasan teks abstraktif dalam bahasa Indonesia yang diperoleh dari penelitian ini dan juga beberapa penelitian lainnya yang dijadikan referensi [5][6][7], ditunjukkan pada Tabel 3.7.

**Tabel 3.7 Skor Evaluasi pada Beberapa Penelitian**

Model	Dataset	ROUGE-1	ROUGE-2
LSTM Error! Reference source not found.	IndoSum (10 ribu sampel)	0.14	Tidak tersedia
GRU Error! Reference source not found.	500 jurnal bahasa Indonesia	0.12	0.01
NeuralSum Error! Reference source not found.	IndoSum (19 ribu sampel)	0.46	0.48
T5-default	IndoSum (19 ribu sampel)	0.48	0.34

T5-id	IndoSum (19 ribu sampel)	0.61	0.51
-------	--------------------------	------	------

#### 4. PENUTUP

Berdasarkan hasil pengujian yang diperoleh, Transformer mampu melakukan peringkasan teks abstraktif dalam bahasa Indonesia dengan menghasilkan ringkasan yang cukup bisa dipahami, walaupun masih terdapat kekurangan seperti *typo* (salah eja) dan kesalahan pada tata bahasa. Performa peringkasan teks abstraktif diukur menggunakan ROUGE, skor tertinggi dicapai oleh T5-id dengan ROUGE-1 sebesar 0.61 dan ROUGE-2 sebesar 0.51. Adapun beberapa saran yang bisa diterapkan pada penelitian selanjutnya di antaranya ialah menggunakan *vocab* yang relevan dengan bahasa dari kasus yang diteliti agar mendapatkan performa yang lebih baik, menggunakan *pre-trained model* untuk membantu proses pelatihan, lakukan *fine-tuning* untuk menemukan model yang paling optimal, dan diharapkan penelitian selanjutnya bisa menghasilkan ringkasan dengan penulisan yang sesuai dengan Ejaan Bahasa Indonesia yang Disempurnakan (EYD).

#### DAFTAR PUSTAKA

- [1] M. Allahyari, E. D. Trippe, and J. B. Gutierrez, "Https://Arxiv.Org/Abs/1707.02268," no. 1.
- [2] S. Twinandilla, S. Adhy, B. Surarso, and R. Kusumaningrum, "Multi-Document Summarization Using K-Means and Latent Dirichlet Allocation (LDA) - Significance Sentences," *Procedia Comput. Sci.*, vol. 135, pp. 663–670, 2018, doi: 10.1016/j.procs.2018.08.220.
- [3] K. E. Dewi and N. I. Widiastuti, "The Design of Automatic Summarization of Indonesian Texts Using a Hybrid Approach," *J. Teknol. Inf. dan Pendidik.*, vol. 15, no. 1, pp. 37–43, 2022, doi: 10.24036/jtip.v15i1.451.
- [4] K. Ivanedra and M. Mustikasari, "Implementasi Metode Recurrent Neural Network pada Text Summarization dengan Teknik Abstraktif," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, p. 377, 2019, doi: 10.25126/jtiik.2019641067.
- [5] M. Alfhi Saputra, "Peringkasan Teks Otomatis Bahasa Indonesia secara Abstraktif Menggunakan Metode Long Short-Term Memory," *e-Proceeding Eng. Vol.8, No.2 April 2021* |, vol. 8, no. 2, pp. 3474–3488, 2021.
- [6] K. Kurniawan and S. Louvan, "IndoSum: A New Benchmark Dataset for Indonesian Text Summarization," *Proc. 2018 Int. Conf. Asian Lang. Process. IALP 2018*, pp. 215–220, 2018, doi: 10.1109/IALP.2018.8629109.
- [7] R. Adelia, S. Suyanto, and U. N. Wisesty, "Indonesian abstractive text summarization using bidirectional gated recurrent unit," *Procedia Comput. Sci.*, vol. 157, pp. 581–588, 2019, doi: 10.1016/j.procs.2019.09.017.
- [8] A. Vaswani *et al.*, "Attention Is All You Need arXiv:1706.03762v5," *Adv. Neural Inf. Process. Syst.*, vol. 31, no. Nips, pp. 5999–6009, 2017, [Online]. Available: <https://www.auteuriteitnvs.nl/onderwerpen/stralingsincidenten/inhoud/incidenten-met-a--en-of-b-objecten>
- [9] A. Gupta, D. Chugh, Anjum, and R. Katarya, "Automated News Summarization Using Transformers," *Lect. Notes Electr. Eng.*, vol. 840, pp. 249–259, 2022, doi: 10.1007/978-981-16-9012-9\_21.
- [10] G. Tsuchiya, "Postmortem Angiographic Studies on the Intercoronary Arterial Anastomoses.: Report I. Studies on Intercoronary Arterial Anastomoses in Adult Human Hearts and the Influence on the Anastomoses of Strictures of the Coronary Arteries.," *Jpn. Circ. J.*, vol. 34, no. 12, pp. 1213–1220, 1971, doi: 10.1253/jcj.34.1213.
- [11] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *EMNLP 2018 - Conf. Empir. Methods Nat. Lang. Process. Syst. Demonstr. Proc.*, pp. 66–71, 2018, doi: 10.18653/v1/d18-2012.
- [12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, M. Matena, and P. J. Liu, "1. Introduction," vol. 21, pp. 1–67, 2020.
- [13] A. M. Rush, "The annotated transformer," in *Proceedings of workshop for NLP open source software (NLP-OSS)*, 2018, pp. 52–60.
- [14] J. L. Ba, J. R. Kiros, and G. E. Hinton, "(LN) Layer Norm," *arXiv:1607.06450v1*, 2015.
- [15] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU) Deep Learning using Rectified Linear Units (ReLU)," no. March 2018, pp. 2–8, 2020.
- [16] M. Fuadi, A. D. Wibawa, and S. Sumpeno, "idT5: Indonesian Version of Multilingual T5



- Transformer,” 2023, [Online]. Available: <http://arxiv.org/abs/2302.00856>
- [17] D. P. Kingma and J. Lei Ba, “15iclr-ADAM,” *Iclr*, pp. 1–15, 2015, [Online]. Available: <https://arxiv.org/pdf/1412.6980.pdf> %22 entire document
- [18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv Prepr. arXiv1810.04805*, 2018.
- [19] “No Title.” <https://docs.oneai.com/docs/rouge-metrics-for-summary-headline> (accessed Apr. 10, 2023)..