

## Peringkasan Teks Berita Berbahasa Indonesia Menggunakan LSTM dan Transformer

Christina Prilla Rosaria Ardyanti\*, Yudi Wibisono, Rani Megasari  
Program Studi Ilmu Komputer Universitas Pendidikan Indonesia  
\*Email: prillarosaria@upi.edu

### Info Artikel

#### Kata Kunci :

Peringkasan Teks, *Natural Language Processing*, *Deep Learning*, LSTM, *Transformer*, Mekanisme Atensi, ROUGE

#### Keywords :

*Text Summarization*, *Natural Language Processing*, *Deep Learning*, LSTM, *Transformer*, *Attention Mechanism*, ROUGE

#### Tanggal Artikel :

Dikirim : 16 September 2023  
Direvisi : 22 Maret 2024  
Diterima : 30 Mei 2024

### Abstrak

Pertumbuhan informasi di internet membuat volume data tekstual semakin besar. Hal ini membuat manusia kesulitan dalam mengolah informasi dengan cepat. Peringkasan teks dapat membantu manusia untuk memahami informasi dalam jumlah yang banyak dengan cepat. Pada penelitian ini, arsitektur *encoder-decoder* akan diimplementasikan pada *dataset* Indosum menggunakan *Long Short Term Memory* (LSTM) dengan tambahan mekanisme atensi dan *Transformer*. Ujicoba juga dilakukan menggunakan *fine-tuning* pada pre-trained model T5-Small dan BART-Small. Eksperimen juga dilakukan dengan membandingkan *dataset* yang menggunakan praproses dan tanpa praproses. Berdasarkan eksperimen, model LSTM-Atensi memiliki kinerja rendah dengan nilai ROUGE-L sebesar 13.0 pada *dataset* yang menggunakan praproses. Sedangkan nilai ROUGE-tertinggi didapatkan dari hasil *fine-tuning* T5-Small dengan nilai sebesar 66.2.

### Abstract

*The proliferation of information on the internet has led to an increasing volume of textual data. This presents a challenge for humans in processing information rapidly. Text summarization can aid humans in quickly comprehending large amounts of information. In this research, an encoder-decoder architecture will be implemented on the Indosum dataset using Long Short-Term Memory (LSTM) along with attention mechanisms and Transformer. Experiments will also involve fine-tuning pre-trained models T5-Small and BART-Small. The influence of preprocessing will also be studied through experiments. Based on the experiments, the LSTM-Attention model demonstrates poor performance with an ROUGE-L score of 13.0 on the preprocessed dataset. Conversely, the highest ROUGE score was achieved through fine-tuning T5-Small, scoring 66.2.*

## 1. PENDAHULUAN

Peringkasan teks otomatis merupakan salah satu masalah yang dapat diselesaikan oleh *Natural Language Processing* (NLP). Dalam bidang NLP, peringkasan teks berbahasa Indonesia masih belum banyak dieksplorasi karena masalah *dataset* [1]. Peringkasan yang dilakukan secara manual memang menghasilkan ringkasan yang baik dan efektif untuk menangkap inti informasi dari teks, tetapi pekerjaan ini memerlukan waktu lama karena memerlukan pemahaman pembaca terhadap teks yang akan diringkas [2]. Peringkasan teks menjadi hal yang tepat untuk diotomatisasi karena pertumbuhan informasi pada internet menghasilkan volume data tekstual yang besar dan membuat manusia kesulitan dalam mengolah informasi tersebut [3].

Terdapat dua metode untuk melakukan peringkasan teks, yaitu peringkasan abstraktif dan ekstraktif. Metode peringkasan abstraktif mengambil informasi penting dari teks yang diringkas dan melakukan parafrase sehingga membentuk kalimat baru sebagai ringkasan. Sementara metode peringkasan ekstraktif hanya menyatukan kalimat yang mengandung informasi penting untuk membentuk ringkasan tanpa mengubah kalimat asli. Penelitian ini akan menggunakan metode abstraktif untuk peringkasan teks secara otomatis.

*Deep Learning* yang menggunakan *Encoder-Decoder* merupakan arsitektur yang cocok untuk permasalahan *Sequence-to-Sequence* seperti mesin translasi, peringkasan teks, dan *chatbot*. Arsitektur *encoder-decoder* dapat diaplikasikan pada algoritma *recurrence* seperti *Long Short Term Memory* (LSTM) atau *Recurrent Neural Network* (RNN) dan *transformer*.

Namun *encoder-decoder* yang menggunakan algoritma *recurrence* sulit menangani input dengan sekuens yang panjang. Hal ini disebabkan oleh konsep *encoder-decoder* yang mengubah *input* menjadi *fixed-length vector*. Penelitian [4] mencoba menyelesaikan masalah ini dengan memperkenalkan mekanisme atensi. Mekanisme atensi bekerja dengan melakukan pembobotan pada tiap *hidden states*, sehingga *decoder* dapat fokus pada *hidden states* dengan informasi yang paling relevan, dalam hal ini *hidden states* yang memiliki bobot paling besar. Penelitian ini mencoba mengkombinasikan algoritma LSTM dan mekanisme atensi Bahdanau [4].

Selain itu, perkembangan *transformer* menghasilkan banyak arsitektur baru seperti T5 [6] dan BART [7]. Kedua *pre-trained model* ini dilatih menggunakan *dataset* yang besar dan menghasilkan performa yang baik, sehingga penelitian ini juga mencoba melakukan *fine-tuning* pada *pre-trained model* T5 dan BART menggunakan variasi arsitektur keduanya yang lebih kecil, yaitu T5-Small dan BART-Small.

Tabel 1. Hasil Perhitungan Nilai ROUGE

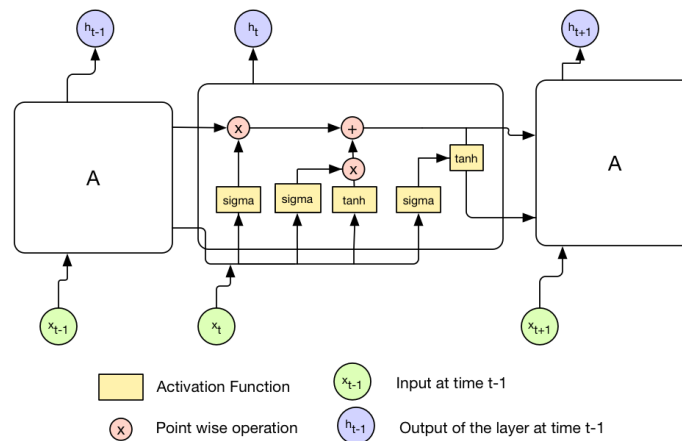
Model	R-1	R-2	R-L
NEURALSUM [5]	<b>68.0</b>	<b>61.7</b>	<b>67.2</b>
TEXTRANK [5]	42.9	29.0	41.0
LSA [5]	41.4	28.4	39.6
LEXRANK [5]	62.9	54.4	62.1
BERTSUM (INDOBERT) [3]	55.0	45.0	54.0
BERTSUM (CAHYA) [3]	60.0	51.0	59.0
PGN [10]	67.0	61.0	65.0
BERTSUM (INDOBERT) [10]	55.0	45.0	54.0

Penelitian ini akan menggunakan *dataset* Indosum untuk melatih model peringkasan teks. *Dataset* Indosum [5] merupakan *dataset* peringkasan teks berbahasa Indonesia yang berisi sekitar 19 ribu artikel berita dan ringkasan dari artikel tersebut. Artikel berita menjadi kandidat yang baik untuk *dataset* peringkasan karena sudah memiliki struktur kalimat yang baik dan tersedia dalam jumlah yang banyak [1]. Tabel 1 merupakan rangkuman kinerja dari penelitian-penelitian peringkasan teks terkait Indosum sebelumnya.

## 2. METODE PENELITIAN

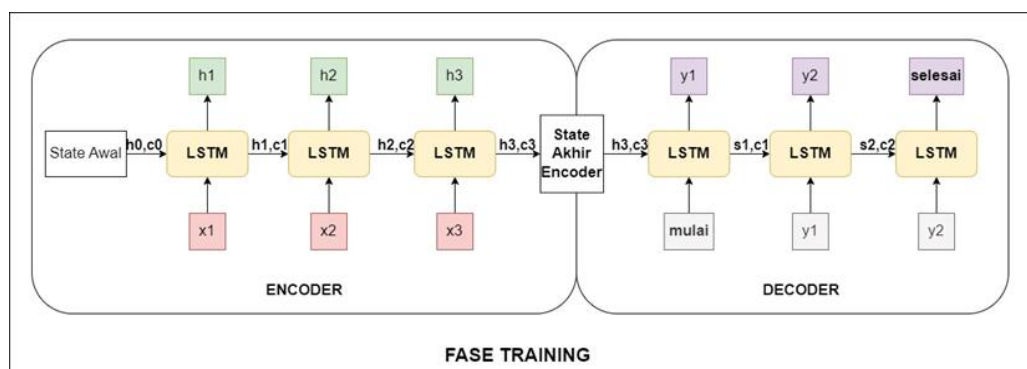
### 2.1 LSTM-Atensi

LSTM merupakan modifikasi dari RNN yang dibuat agar model dapat menyimpan informasi lebih lama dan mempertimbangkan informasi mana yang akan disimpan atau dilupakan. Struktur LSTM terdiri atas tiga gerbang, yaitu *input gate*, *output gate*, dan *forget gate* beserta *cell state* yang berfungsi sebagai memori yang menyimpan informasi. Gambar 1 merupakan gambaran dari arsitektur LSTM.



Gambar 1. Struktur LSTM [11]

Untuk peringkasan teks, LSTM diaplikasikan ke dalam arsitektur *encoder-decoder*. Arsitektur ini dibuat untuk menangani masalah representasi secara sekuensial dengan panjang vektor yang tidak diketahui dan tidak ditentukan [12], sehingga cocok untuk menangani permasalahan *sequence-to-sequence* seperti peringkasan teks dan mesin translasi yang memiliki *output* tidak tetap. Bagian *encoder* berfungsi membaca *input* sekuens di setiap *timestep* dan menangkap informasi dari *input* tersebut. Sementara *decoder* berfungsi untuk memprediksi kata yang menjadi *output* berdasarkan *output* sebelumnya. Arsitektur *encoder-decoder* digambarkan oleh Gambar 2.

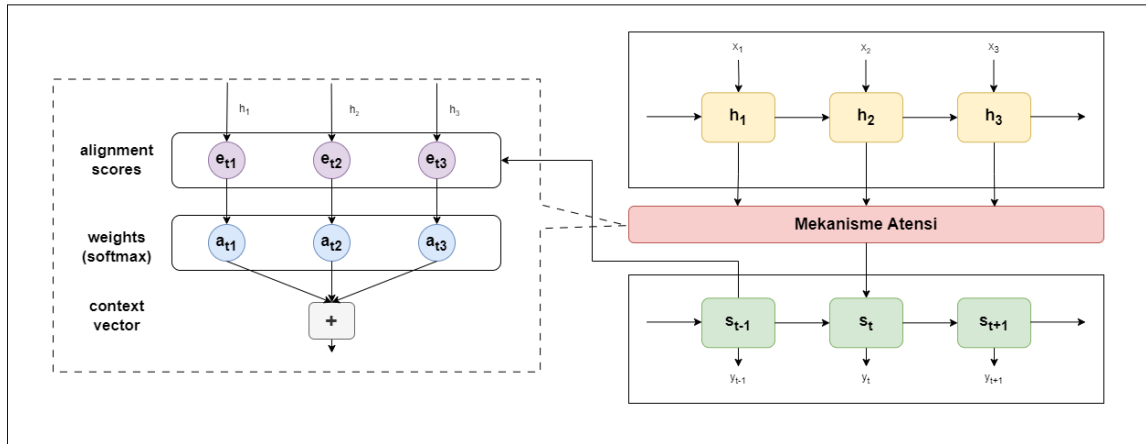


Gambar 2. Arsitektur Encoder-Decoder

*Encoder-decoder* mengubah *input sequence* menjadi *fixed-length vector*. Namun, *fixed-length vector* pada *encoder-decoder* menjadi masalah ketika mendapat *input* yang memiliki sekuens panjang. Representasi kata pada *fixed-length vector* ini tidak memiliki kemampuan untuk melakukan *encode* kalimat yang panjang dengan struktur dan konteks yang rumit [13].

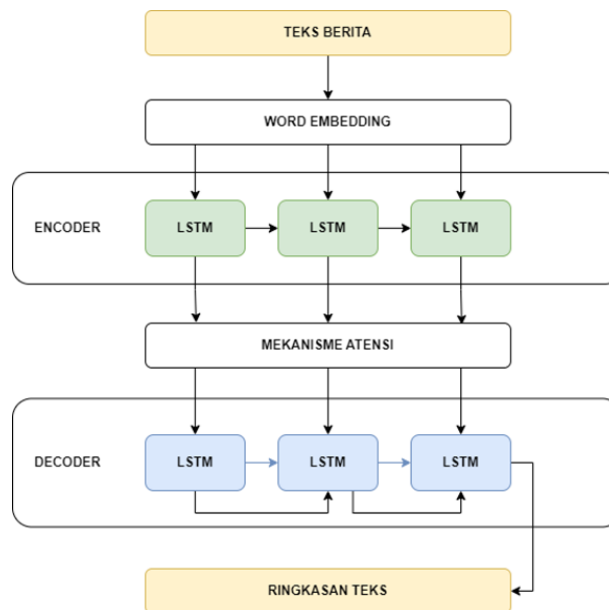
Mekanisme atensi Bahdanau [4] dapat digunakan untuk mengatasi masalah tersebut. Teknik ini bekerja dengan memilih bagian dari urutan vektor *input* secara fleksibel pada saat proses *generate* kata. Pemilihan vektor ini didasarkan pada vektor yang memiliki informasi paling relevan dengan melakukan pembobotan pada *hidden state* di *encoder* dan *hidden state* di *decoder*. Hasil pembobotan ini akan membantu *decoder* untuk fokus pada *state* dengan bobot paling besar. Gambar 3 merupakan diagram mekanisme atensi Bahdanau. Perhitungan bobot ini diawali

dengan menghitung *alignment scores* dengan input *hidden state encoder* dan *hidden state decoder* pada *timestep* sebelumnya. *Alignment scores* akan digunakan untuk menghitung bobot dengan menggunakan fungsi *softmax*. Bobot tersebut akan dijumlahkan dengan semua *hidden state* di *encoder* dan menjadi *context vector*.



Gambar 3. Diagram Mekanisme Atensi [14]

Konsep atensi ini kemudian diaplikasikan pada arsitektur *encoder-decoder*. Gambar 4 menunjukkan arsitektur *encoder-decoder* untuk peringkasan teks yang sudah diaplikasikan layer atensi.



Gambar 4. Arsitektur Encoder-Decoder dengan Mekanisme Atensi

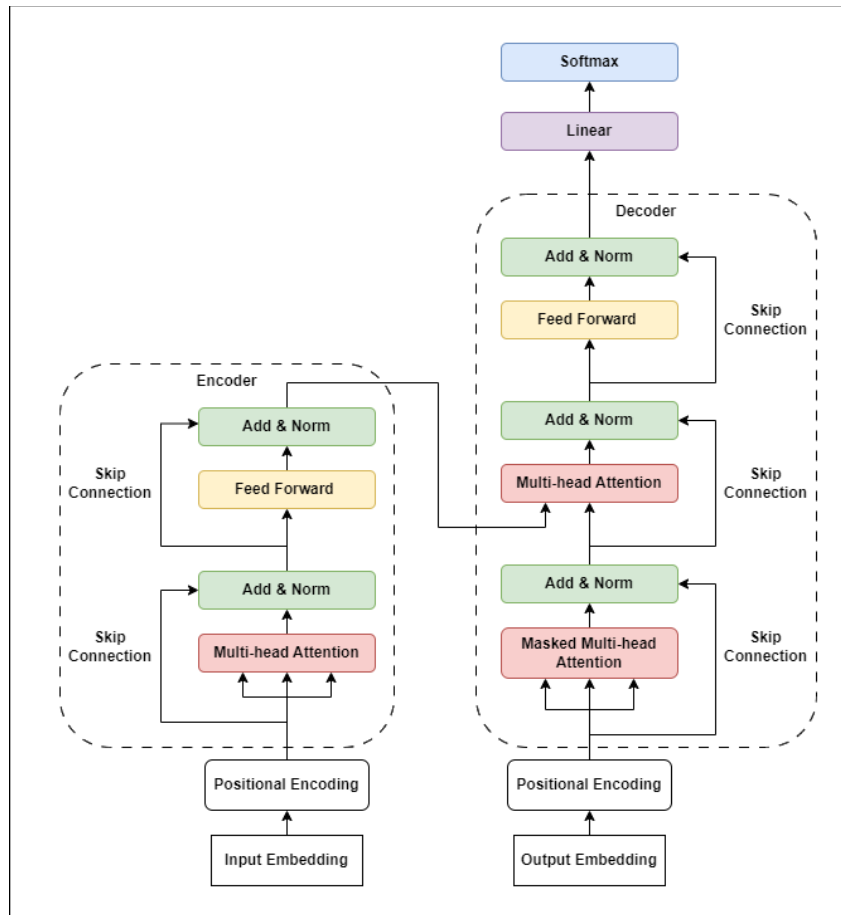
## 2.2 Transformer

*Transformer* [15] mengimplementasikan konsep *encoder-decoder* pada arsitekturnya. *Transformer* merupakan arsitektur *neural network* tetapi tidak menggunakan algoritma *recurrence* seperti LSTM dan RNN. Arsitektur *transformer* terdiri dari layer atensi bernama *multi-head attention*, *layer feedforward network*, dan *layer normalisasi*. *Input embedding* dioperasikan dengan vektor posisi sehingga menghasilkan *positional encoding* sebelum dimasukkan pada *layer* selanjutnya. Gambar 5 menunjukkan arsitektur *transformer* konvensional.

Masing-masing *encoder* dan *decoder* memiliki enam *layer*. Setiap *layer* pada *encoder* mengandung dua *sublayer* yaitu *multi-head attention* dan *feedforward network*. Hasil kedua *sublayer* ini kemudian dinormalisasi dan diaplikasikan pada *add function* yang mengoperasikan koneksi residual dari *input sublayer* yang belum diproses.

Sedangkan pada *decoder*, tiap *layer* memiliki tiga *sublayer*, yaitu *masked multi-head attention*, *multi-head attention*, dan *feedforward network*. Ketiga *sublayer* ini juga akan dinormalisasi.

*Multi-head attention* berbeda dengan mekanisme atensi Bahdanau. *Multi-head attention* mengadaptasi *self-attention* yang bekerja dengan memproyeksikan *token embedding* menjadi *key*, *query*, dan *value* yang kemudian mengoperasikannya ke perhitungan *scaled dot-product*. Namun, proses ini dikerjakan secara paralel di setiap *attention head*. Sedangkan *masked multi-head attention* adalah *layer* pada *decoder* dengan *token embedding* yang disamarkan (*masking*). *Masking* dilakukan agar *decoder* memprediksi *output* berdasarkan *output-output* sebelumnya. Lalu, pada *layer feedforward network* terdapat 2 *layer* linear yang mengaplikasikan fungsi aktivasi ReLU.



Gambar 5. Arsitektur *Transformer* [15]

Pengembangan arsitektur *transformer* menghasilkan variasi-variasi seperti BART dan T5. T5 memiliki arsitektur yang mirip dengan arsitektur *transformer* konvensional, dengan beberapa perbedaan implementasi layer normalisasi dan jenis *positional embedding* yang digunakan [5]. Model T5 memiliki variasi arsitektur yang lebih kecil, yaitu T5-Small yang akan digunakan pada penelitian ini. Model BART menggunakan arsitektur *transformer* konvensional dan merupakan gabungan dari arsitektur *encoder* BERT dan *decoder* GPT [6]. BART memiliki variasi arsitektur yang lebih kecil, yaitu BART-Small yang juga akan digunakan pada penelitian ini. Kedua *pre-trained model* ini diimplementasikan melalui proses *fine tuning*.

### 2.3 Dataset

Penelitian ini menggunakan *dataset* Indosum yang merupakan *dataset* ringkasan teks yang diperkenalkan oleh Kurniawan dan Louvan [9]. *Dataset* ini berisi sekitar 19 ribu teks berita beserta ringkasannya. Terdapat 7 kolom pada *dataset* ini yaitu *category*, *gold\_labels*, *id*, *paragraphs*, *source*, *source\_url*, dan *summary*. Kolom yang akan digunakan untuk pelatihan model hanya kolom *paragraphs* yang berisi teks asli berita dan kolom *summary* yang berisi ringkasan asli teks berita.

*Dataset* Indosum awalnya mengandung *5-folds data* yang masing-masing terbagi menjadi data *training*, *validation*, dan *testing*. Tetapi berdasarkan [3], *dataset* tersebut digabungkan menjadi *one fold* dan dibagi menjadi data 70% data *training*, 10% *validation* dan 20% *testing*. Karena keterbatasan komputasi eksperimen, pada penelitian ini hanya digunakan data teks berita yang panjangnya kurang dari 500 kata. Jumlah data keseluruhan yang digunakan adalah 17.545 artikel dengan distribusi yang sama untuk *training*, *validation* dan *testing*. Distribusi data ini berdasarkan penelitian [3] dan penelitian [10].

Penelitian ini juga menggunakan dua skenario untuk *dataset*, yaitu *dataset* dengan praproses dan *dataset* dengan tidak menggunakan praproses. Praproses pada *dataset* dilakukan dengan proses *lowercasing* kata dan menghapus nama platform berita yang ada di awal artikel. Lalu, kedua skenario juga menambahkan token “sostok” pada awal kalimat dan “eostok” pada akhir kalimat ringkasan.

## 2.4 Evaluasi

*Recall-Oriented Understanding for Gisting Evaluation* (ROUGE) akan digunakan pada penelitian ini untuk mengevaluasi hasil ringkasan. Beberapa jenis metode evaluasi ROUGE yang akan digunakan adalah ROUGE-1, ROUGE-2, dan ROUGE-L. ROUGE-1 dan ROUGE-2 digunakan untuk mengukur jumlah kesamaan antara teks ringkasan asli dan ringkasan yang diprediksi oleh model per satu kata dan per dua kata. Sementara ROUGE-L digunakan untuk mengukur sekuens terpanjang pada ringkasan yang diprediksi yang sama dengan ringkasan asli. Nilai yang akan digunakan adalah F1-score dari ketiga jenis metode ROUGE tersebut.

## 2.5 Konfigurasi Eksperimen

Arsitektur untuk LSTM-Atensi menggunakan dua *layer encoder* LSTM dengan ukuran *units* sebesar 300 dimensi, *dropout* dan *recurrent dropout* sebesar 0.4. *Input shape* untuk *encoder* diatur menjadi sejumlah jumlah kata terbesar yang ada pada artikel berita, yaitu 500. *Layer embedding* memiliki dimensi *input* sebesar jumlah *vocabulary* pada data *training* dan dimensi *output* sebesar 300. Jumlah *units* dan dimensi *output* pada *embedding* diatur menjadi hanya 300 karena faktor keterbatasan komputasi. Sementara untuk *decoder*, hanya menggunakan satu *layer* LSTM yang memiliki parameter yang sama dengan *layer* LSTM di *encoder*. Setelah itu, *output hidden state* dari *encoder* dan *decoder* akan dimasukkan ke *layer* Atensi Bahdanau dan menghasilkan bobot atensi. Bobot ini akan dimasukkan juga ke *layer concatenate* untuk disatukan dengan *output decoder*. Model ini kemudian dilatih sebanyak 20 *epochs* dengan *batch size* sebesar 64, *optimizer rmsprop*, dan *loss sparse\_categorical\_crossentropy*.

Sementara untuk *fine-tuning* BART-Small dan T5-Small, *hyperparameter* yang digunakan adalah *batch size* sebesar 16, *learning rate*  $2e-5$ , *weight decay* 0.01, *floating point* 16, *epoch* sebanyak 10 dan *save total limit* sebesar 3. Jumlah *batch size* yang digunakan juga sudah disesuaikan dengan kapasitas perangkat.

## 3. HASIL DAN PEMBAHASAN

Setiap model diuji untuk memprediksi ringkasan menggunakan data *test* yang telah tersedia. Kemiripan hasil ringkasan asli dan ringkasan prediksi dihitung menggunakan metrik ROUGE-1, ROUGE-2, dan ROUGE-L. Tabel 2 menunjukkan hasil perhitungan nilai ROUGE setiap model menggunakan skenario *dataset* yang menggunakan praproses dan *dataset* yang tidak menggunakan praproses.

Tabel 2. Hasil Perhitungan Nilai ROUGE

Model	R-1	R-2	R-L
LSTM-Atensi (Dengan Praproses)	19.1	3.6	14.2
LSTM-Atensi (Tanpa Praproses)	17.7	3.2	13.1
BART-Small (Dengan Praproses)	63.5	53.3	59.8
BART-Small (Tanpa Praproses)	64.8	55.1	61.2

BART-Small (Tanpa Fine-Tuning)	59.2	39.2	44.2
T5-Small (Dengan Praproses)	69.3	61.9	<b>66.2</b>
T5-Small (Tanpa Praproses)	<b>69.5</b>	<b>62.0</b>	<b>66.2</b>
T5-Small (Tanpa Fine-Tuning)	45.5	36.7	41.5

Berdasarkan Tabel 2, hasil ringkasan yang diprediksi oleh T5-Small dengan *dataset* yang tidak dipraproses mendapat nilai paling tinggi di setiap perhitungan metrik ROUGE. Sementara model LSTM-Atensi mendapat nilai terendah di antara ketiga model yang dilatih. Contoh hasil prediksi ringkasan dapat dilihat pada Tabel 3. Hal ini dapat disebabkan oleh beberapa hal seperti kompleksitas arsitektur model dan penggunaan *word embedding*. Beberapa kendala yang ditemui pada penelitian ini adalah terkait sumber daya komputasi yang digunakan untuk melatih model. Keterbatasan ini membuat arsitektur model yang digunakan pada proses pelatihan tidak terlalu kompleks sehingga berpotensi membuat model kurang optimal untuk menghasilkan ringkasan yang baik dengan konteks ringkasan yang tepat.

Sedangkan hasil prediksi ringkasan dari *pre-trained model* T5-Small dan BART-Small memiliki nilai ROUGE yang tinggi. Hal ini menandakan bahwa ringkasan yang diprediksi mirip dengan ringkasan asli. Ini dapat disebabkan oleh penggunaan *pre-trained model* yang sebelumnya telah dilatih menggunakan sumber daya yang baik dan *dataset* yang besar sehingga ketika dilakukan proses *fine-tuning* dan dilatih menggunakan *dataset* Indosum, kedua *pre-trained model* ini menghasilkan ringkasan yang baik karena sudah memiliki *knowledge* pada pelatihan sebelumnya.

Namun, ketika tidak dilakukan *fine-tuning*, *pre-trained model* T5-Small dan BART-Small memiliki performa di bawah *pre-trained model* T5-Small dan BART-Small yang dilakukan *fine-tuning*, sehingga dapat disimpulkan proses *fine-tuning* penting dilakukan untuk menghasilkan ringkasan yang lebih baik.

Berdasarkan hasil prediksi ringkasan, model LSTM-Atensi belum dapat menghasilkan prediksi ringkasan yang baik. Kalimat ringkasan yang dihasilkan oleh model LSTM-Atensi memiliki konteks ringkasan yang berbeda dengan ringkasan aslinya. Kalimat ringkasan yang dihasilkan pun tampak belum tuntas. Kesalahan ini juga dapat disebabkan oleh *word embedding* yang digunakan saat pelatihan model. *Word embedding* yang digunakan dilatih dari *scratch* menggunakan *dataset* yang terbatas sehingga model masih belum bisa menangkap setiap konteks kata dengan baik. Model LSTM-Atensi juga masih kesulitan dalam memproses *input* yang memiliki *long sequences* seperti *input* pada *dataset* ini yang memiliki jumlah sekuens terpanjang sebesar 500 kata.

**Tabel 3. Hasil Prediksi Ringkasan**

**Ringkasan Manual :**

Gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat Negara Chile pada Senin 25 4. Gempa berpusat di sekitar 35 kilometer sebelah barat kota pesisir Valparaiso atau 137 kilometer dari Ibu Kota Santiago. Gedung gedung di Santiago terguncang dan jaringan telepon di Valparaiso sempat mati. Warga yang berada di sekitar pusat gempa bergegas lari ke tempat tinggi saat gempa mengguncang.

**Model LSTM-Atensi (Dengan Praproses):**

gempa berkekuatan 5,3 magnitudo mengguncang wilayah pesisir timur laut Kabupaten Karangasem Kabupaten Karangasem Jawa Timur sabtu dini hari ini senin 11 11 WIB gempa berkekuatan 2,3 SR dan mengguncang wilayah tersebut menjadi pusat gempa yang mengguncang wilayah timur laut laut timur laut laut timur laut laut laut laut setempat gempa berkekuatan 5,3 kilometer dan timur

**Model LSTM-Atensi (Tanpa Praproses):**

badan energi dan sumber daya manusia mineral menilai pihaknya akan melakukan perjalanan kerja sama dengan kasus korupsi yang dilakukan oleh Kementerian Agama Dan Informatika RI di kantor Kementerian Kesehatan Dan Perumahan Rakyat Jakarta Pusat terkait kasus ini dilakukan oleh ketua DPR RI Muhammad Kurniawan dia mengatakan pihaknya akan melakukan pengawasan terhadap pihak yang disampaikan oleh Kementerian Perhubungan

**Model BART-Small (Tanpa Fine Tuning):**

Gempa berpusat di sekitar 35 kilometer sebelah barat kota pesisir Valparaiso atau 137 kilometer dari Ibu Kota Santiago. Gedung gedung di Santiago terguncang dan jaringan telepon di Valparaiso sempat mati. Gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat Negara Chile pada Senin 25 4 Sekitar pukul 18.40 waktu setempat.

**Model BART-Small (Dengan Praproses):**

gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat negara chile pada senin 25 4 sekitar pukul 18.40 waktu setempat. gempa berpusat disekitar 35 kilometer sebelah barat kota pesisir Valparaiso atau 137 kilometer dari ibu kota santiago. salah seorang warga paloma salamo 26 mengatakan warganya yang berada di sekit pusat gempa.

**Model BART-Small (Tanpa Praproses):**

Gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat Negara Chile pada Senin 25 4. Gempa berpusat di sekitar 35 kilometer sebelah barat kota pesisir Valparaiso atau 137 kilometer dari Ibu Kota Santiago. Gedung gedung di Santiago terguncang dan jaringan telepon di Valparaiso sempat mati. Salah seorang warga setempat Paloma Salamo 26 mengatakan gempa bergegas lari ke tempat tinggi.

**Model T5-Small (Dengan Praproses):**

gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat negara chile pada senin 25 4 sekitar pukul 18.40 waktu setempat. gempa berpusat di sekitar 35 kilometer sebelah barat kota pesisir valparaiso atau 137 kilometer dari ibu kota santiago. gedung gedung di santiago terguncang dan jaringan telepon di valparaiso sempat mati. salah seorang warga setempat paloma salamo 26 mengatakan warga yang berada di sekitar pusat gempa bergegas lari ke tempat tinggi saat gempa mengguncang.

**Model T5-Small (Tanpa Praproses):**

Gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat Negara Chile pada Senin 25 4 sekitar pukul 18.40 waktu setempat. Gempa berpusat di sekitar 35 kilometer sebelah barat kota pesisir Valparaiso atau 137 kilometer dari Ibu Kota Santiago. Gedung gedung di Santiago terguncang dan jaringan telepon di Valparaiso sempat mati. Salah seorang warga setempat Paloma Salamo 26 mengatakan warga yang berada di sekitar pusat gempa bergegas lari ke tempat tinggi saat gempa mengguncang.

**Model T5-Small (Tanpa Fine-Tuning):**

gempa berkekuatan 6,9 skala richter melanda lepas pantai bagian barat Negara Chile pada Senin 25 4 waktu setempat. gedung gedung di Santiago terguncang dan jaringan telepon di Valparaiso sempat mati.

---

Hasil prediksi ringkasan dari proses *fine-tuning pre-trained model* BART-Small dan T5-Small sudah menghasilkan ringkasan baik, meskipun masih terdapat beberapa kesalahan kecil seperti pada hasil ringkasan BART-Small yang memiliki kata yang tidak dikenal dalam bahasa Indonesia seperti “sekit”. Konteks ringkasan yang diprediksi juga sudah sesuai dengan ringkasan asli. Hasil ringkasan yang diprediksi T5-Small menjadi ringkasan yang memiliki nilai ROUGE tertinggi. Hal itu juga dapat dibuktikan kalimat yang dihasilkan pada prediksi ringkasan T5-Small yang mirip dengan ringkasan asli.

Namun, jika dibandingkan dengan hasil ringkasan BART-Small dan T5-Small yang dilakukan tanpa proses *fine tuning*, hasil ringkasan keduanya masih memiliki nilai ROUGE yang lebih rendah dibandingkan hasil ringkasan dari *pre-trained model* yang melalui proses *fine-tuning*. Secara konteks, kedua ringkasan tersebut sudah sama dengan ringkasan asli dan struktur kalimat yang dihasilkan sudah baik.

#### 4. KESIMPULAN

Peringkasan teks secara otomatis bermanfaat untuk membantu manusia dalam meringkas teks secara cepat dan efisien. Peringkasan teks pada penelitian ini menggunakan arsitektur *encoder-decoder* LSTM yang ditambahkan mekanisme atensi. Model kedua menggunakan *fine-tuning* pada *pre-trained model transformer* T5-Small dan BART-Small. Model LSTM-Atensi menghasilkan ringkasan yang kurang baik, karena terdapat kesalahan konteks dan kalimat yang tidak tuntas.

Sementara hasil ringkasan T5-Small dan BART-Small memiliki konteks yang sama dengan ringkasan yang asli, walaupun terdapat beberapa kata yang tidak dikenali. Nilai ROUGE tertinggi didapatkan oleh hasil ringkasan T5-Small. Hal yang masih dapat ditingkatkan dari penelitian ini adalah menggunakan *dataset* yang lebih abstraktif untuk melatih model karena setelah ditinjau, *dataset* Indosum cenderung memiliki ringkasan asli yang persis dengan kalimat-kalimat awal pada artikel berita, sehingga lebih mengarah pada peringkasan ekstraktif. Lalu, kendala lain dalam penelitian ini adalah spesifikasi perangkat yang digunakan dalam melatih model. Hal ini menyebabkan keterbatasan saat mencoba arsitektur-arsitektur model yang lebih kompleks sehingga kompleksitas arsitektur model juga dapat lebih ditingkatkan. Penggunaan *pre-trained word embedding* juga dapat dilakukan agar model dapat menangkap konteks kalimat dengan lebih baik.

#### DAFTAR PUSTAKA

- [1] F. Koto, J. H. Lau, and T. Baldwin, “Liputan6: {A} Large-scale Indonesian *Dataset* for Text Summarization,” CoRR, vol. abs/2011.0, 2020, [Online]. Available: <https://arxiv.org/abs/2011.00679>
- [2] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, “A survey of automatic text summarization: Progress, process and challenges,” IEEE Access, vol. 9, pp. 156043–156070, 2021.



- [3] R. Wijayanti, M. L. Khodra, and D. H. Widyantoro, "Indonesian Abstractive Summarization using Pre-trained Model," in 2021 3rd East Indonesia Conference on Computer and Information Technology (EIconCIT), 2021, pp. 79–84.
- [4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate." 2016.
- [5] K. Kurniawan and S. Louvan, "Indosum: A new benchmark *dataset* for Indonesian text summarization," 2018 Int. Conf. Asian Lang. Process., 2018, doi: 10.1109/ialp.2018.8629109.
- [6] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020, [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>
- [7] P. M. Hanunggul and S. Suyanto, "The impact of local attention in lstm for abstractive text summarization," in 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2019, pp. 54–57.
- [8] S. Song, H. Huang, and T. Ruan, "Abstractive text summarization using LSTM-CNN based deep learning," *Multimed. Tools Appl.*, vol. 78, pp. 857–875, 2019.
- [9] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," arXiv Prepr. arXiv1908.08345, 2019.
- [10] R. Wijayanti, M. L. Khodra, and D. H. Widyantoro, "Single Document Summarization Using BertSum and Pointer Generator Network," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 4, pp. 916–930, 2021.
- [11] R. Dua and M. S. Ghotra, *Keras Deep Learning Cookbook: Over 30 recipes for implementing deep neural networks in python*. Packt Publishing, 2018.
- [12] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to Sequence Learning with Neural Networks." 2014.
- [13] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches." 2014.
- [14] I. Vasilev, *Advanced deep learning with python: Design and implement advanced next-generation AI solutions using tensorflow and pytorch*. Packt, 2019.
- [15] A. Vaswani et al., "Attention Is All You Need." 2017.