

# Ringkasan Ekstraktif Otomatis pada Berita Berbahasa Indonesia Menggunakan Metode BERT

Franky Halim, Liliana, Kartika Gunadi.

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031)-2983455, Fax. (031)-8417658

E-mail: frankyhalim1@gmail.com, lilian@petra.ac.id, kgunadi@petra.ac.id

## ABSTRAK

Pada era yang semakin modern, informasi menjadi bagian yang penting dalam kehidupan sehari-hari. Dalam mendapatkan informasi terdapat beberapa hal yang dapat dilakukan dimana salah satunya adalah dengan membaca. Dengan semakin banyaknya informasi yang tersedia di internet membuat manusia kerepotan untuk terus mengikuti perkembangannya. Selain itu, berita *online* juga merupakan salah satu sumber informasi yang ada di internet dengan jumlah yang sangat banyak serta topik yang beragam. Membaca keseluruhan informasi tersebut juga terkadang memerlukan waktu yang lama. Oleh karena itulah, diperlukan suatu pembuatan ringkasan dari informasi berita yang tersedia secara *online* untuk mengurangi waktu baca dan mendapatkan informasi yang relevan.

Pada penelitian ini akan dilakukan pembuatan ringkasan berita dengan memilih kalimat penting dari teks berita. Metode yang digunakan adalah *Bidirectional Encoder Representations from Transformers* dengan tambahan *transformer encoder layer*.

Berdasarkan hasil pengujian yang telah dilakukan pada penelitian ini menunjukkan bahwa *pre-trained* model indolem/indobert-base-uncased dapat menghasilkan nilai *F1-Score* terbaik untuk *ROUGE-1* sebesar 57.17, *ROUGE-2* sebesar 51.27, dan *ROUGE-L* sebesar 55.20 pada referensi abstraktif serta *ROUGE-1* sebesar 84.46, *ROUGE-2* sebesar 83.21, dan *ROUGE-L* sebesar 83.40 pada referensi ekstraktif.

**Kata Kunci:** *Text Summarization, Online News, Bidirectional Encoder Representations from Transformers.*

## ABSTRACT

*In this modern era, information has become an important part of everyday life. In getting information several things can be done where one of them is by reading. With the increasing amount of information available on the internet, it is difficult for humans to keep abreast of developments. Online news is also one of the sources of information on the internet with a very large number and various topics. Reading the whole information sometimes also takes a long time. Therefore, it is necessary to make a summary of the available online news to reduce reading time and obtain relevant information. In this research, a summary of the news will be made by selecting important sentences from the news text.*

*The method used in this research is Bidirectional Encoder Representations from Transformers with the addition of a transformer encoder layer.*

*Based on the results of the tests that have been carried out, the pre-trained indolem/indobert-base-uncased model can produce the best F1-Score 57.17 for ROUGE-1, 51.27 for ROUGE-2, and 55.20 for ROUGE-L using abstractive reference and 84.46 for ROUGE-1,*

*83.21 for ROUGE-2, and 83.40 for ROUGE-L using extractive reference.*

**Keywords:** *Text Summarization, Online News, Bidirectional Encoder Representations from Transformers.*

## 1. PENDAHULUAN

Informasi merupakan bagian penting dalam kehidupan sehari-hari. Dalam mendapatkan informasi terdapat beberapa hal yang dapat dilakukan dimana salah satunya adalah dengan membaca. Semakin banyaknya informasi yang ada di internet membuat manusia kerepotan untuk terus mengikuti perkembangannya [6]. Maka dari itu, diperlukan suatu pembuatan ringkasan terhadap informasi-informasi yang tersedia secara *online*.

Ringkasan merupakan suatu teks singkat yang dihasilkan dari kumpulan teks panjang namun tetap menyimpan informasi penting dari teks asalnya [9]. Dengan adanya ringkasan akan membantu dalam membaca dengan waktu yang lebih cepat dibandingkan dengan harus membaca informasi secara utuh. Selain itu, ringkasan juga membantu dalam mengabaikan informasi yang tidak relevan tanpa harus kehilangan makna dari informasi yang dibaca. Dalam melakukan pembuatan ringkasan juga terkadang masih dilakukan secara manual oleh manusia yang memakan waktu lama karena jumlah informasi yang ada sangat banyak. Sehingga pembuatan ringkasan secara otomatis diperlukan untuk mengatasi hal ini.

Berita *online* merupakan salah satu sumber informasi yang ada di internet dengan jumlah yang sangat banyak serta topik yang beragam [14]. Topik berita dapat meliputi mengenai politik, ekonomi, olahraga, teknologi, dan masih banyak lagi. Adanya variasi informasi yang luas dan sering digunakan sehingga membuat berita menjadi objek pada penelitian ini. Kemudian, pada penelitian ini dilakukan pada berita berbahasa Indonesia karena untuk pembuatan ringkasan secara otomatis pada berita berbahasa Indonesia belum terlalu berkembang seperti berita berbahasa Inggris. Selain itu, bahasa Indonesia juga masih tergolong ke dalam *low-resource language* bila dibandingkan dengan *high-resource language* seperti bahasa Inggris sehingga perlu dilakukan penelitian untuk pengembangan *natural language processing* berbahasa Indonesia [7].

*Bidirectional Encoder Representations from Transformers* (BERT) merupakan model yang dikembangkan oleh Google untuk keperluan *Natural Language Processing* seperti *text classification*, *question answering*, dan lain-lain [3][5]. Selain itu, pada BERT tersusun atas *transformer encoder* yang digunakan untuk mempelajari hubungan antar kalimat [15]. Pada penelitian ini akan menggunakan BERT untuk melakukan peringkasan berita *online*. Terdapat beberapa penelitian yang telah melakukan peringkasan secara otomatis pada berita *online* berbahasa Inggris dengan menggunakan *pre-trained encoder* BERT. Pada penelitian yang dilakukan dengan BERT digunakan tambahan susunan *transformer*

*encoder* setelah dihasilkan *embedding* dari BERT. Evaluasi terhadap model yang diusulkan mampu menghasilkan ringkasan ekstraktif berita dengan skor yang paling tinggi bila dibandingkan dengan metode lainnya [13].

Pada penelitian yang telah dilakukan oleh Liu dan Lapata untuk penerapan dan perbandingan metode BERT dengan metode lain seperti *Recurrent Neural Network* diperoleh bahwa metode BERT dapat memperoleh hasil yang baik [13]. Kemudian, BERT juga dapat mempelajari kata yang tidak dikenali dengan mengubahnya menjadi sub-kata. Penerapan metode BERT pada penelitian juga dilakukan untuk mengatasi keterbatasan *Recurrent Neural Network* dalam memproses kata demi kata yang dilakukan secara berurutan dimana pada metode BERT akan secara langsung memproses seluruh input kata secara bersamaan. Maka dari itu, pada penelitian yang akan dilakukan ini akan menggunakan BERT dalam menghasilkan ringkasan ekstraktif pada berita berbahasa Indonesia.

Dalam pembuatan ringkasan otomatis berdasarkan pada hasilnya dapat dibedakan menjadi dua yaitu abstraktif dan ekstraktif [6]. Pada penelitian ini akan digunakan metode ekstraktif dalam pembuatan ringkasannya karena pengguna internet lebih memilih teks ringkasan yang mirip dengan teks asli yang dibuat oleh penulis [14]. Metode yang akan digunakan dalam penelitian ini untuk menghasilkan ringkasan ekstraktif adalah *Bidirectional Encoder Representations from Transformers* (BERT).

## 2. LANDASAN TEORI

### 2.1 News Summarization

Ringkasan merupakan suatu teks singkat yang dihasilkan dari kumpulan teks panjang namun tetap menyimpan informasi penting dari teks asalnya [9]. Informasi yang menunjukkan ide utama dapat diberikan penilaian dari seberapa tersambungannya informasi dengan tema dan judul berita serta penilaian dari pembaca. Lalu, untuk menentukan informasi penting maka lebih akurat bila dipilih oleh manusia. Karena itulah, perlu dasar untuk melakukan pelatihan terhadap ringkasan yang dibuat oleh manusia serta sudah dipercaya sebagai ringkasan yang baik dengan mengandung informasi penting dari dokumen. Berdasarkan dari hasilnya, metode untuk peringkasan otomatis dapat dibedakan menjadi dua kategori yaitu abstraktif dan ekstraktif. Metode abstraktif memiliki tujuan untuk melakukan *paraphrase* dari informasi yang dipilih pada dokumen yang akan diringkas. Kemudian, untuk metode ekstraktif memiliki tujuan untuk membuat ringkasan dengan memilih kalimat maupun kata-kata penting dari dokumen yang akan diringkas. Dengan adanya ringkasan dapat mempersingkat waktu untuk menemukan informasi penting dari bacaan [6].

### 2.2 Indosum

Indosum merupakan *dataset* berita berbahasa Indonesia yang merupakan hasil penelitian yang dilakukan oleh Kurniawan dan Louvan [11]. *Dataset* pada penelitian dapat dijadikan sebagai tolak ukur dalam pembuatan ringkasan otomatis berbahasa Indonesia. *Dataset* ini berbentuk *json*. Selain itu, *dataset* ini juga berisi 18.774 artikel berita yang diambil dari berbagai portal berita seperti CNN Indonesia, Kumparan, dan portal berita lainnya. Dalam *dataset* ini terdapat *category* yang menunjukkan kategori berita, *gold label* yang merupakan label *gold summarization* untuk peringkasan ekstraktif, *id* unik tiap berita dari berita yang bersangkutan, *source* yang berisi sumber berita, *source url* yang berisi *url* dari berita, *paragraph* yang berisi paragraf teks dari berita, dan *summary* yang berisi ringkasan berita yang dibuat oleh manusia dan bersifat abstraktif [11].

## 2.3 Bidirectional Encoder Representations from Transformers (BERT)

Dalam BERT terdapat dua tahapan yang dilakukan yaitu *pre-training* dan *fine-tuning*. Selama *pre-training*, akan dilakukan dengan *Masked Language Model* (MLM) dimana secara *random* melakukan *mask* terhadap beberapa *token* dari input kalimat dan tujuannya untuk memprediksi kata yang di-*masked* berdasarkan pada konteks kalimat tersebut. Pada BERT, input dari suatu teks kalimat akan diubah menjadi *token sequence*. Dimana terdapat *token* [CLS] yang merupakan *token* untuk melambangkan mulainya suatu kalimat dan *token* [SEP] yang merupakan *token* untuk memisahkan antar kalimat. Selain itu, juga terdapat *token* [PAD] untuk melakukan penambahan *padding* atau *token* kosong ke dalam kalimat untuk memenuhi *fixed length* dari suatu kalimat yang diinputkan. *Fixed length* pada model BERT adalah 512. Untuk melakukan pelatihan terhadap representasi *bidirectional* maka dilakukan *mask* dengan persentase tertentu dari input *token* secara *random* dan dilanjutkan dengan memprediksi *masked token* yang dilatih pada suatu *pre-training task* dengan *Next Sentence Prediction* (NSP). Lalu, untuk *fine-tuning* model BERT akan melakukan inialisasi parameter yang sudah dilatih sebelumnya [5]. Mengenai penerapan untuk pembuatan ringkasan dilakukan beberapa perubahan dari model asli BERT. Perubahan terletak pada *token* [CLS] dan [SEP] yang disisipkan pada setiap awal dan akhir kalimat. Kemudian, saat input dokumen melebihi *fixed length* dari BERT maka akan dilakukan pemotongan terhadap kalimat yang berikutnya [5][13].

## 2.4 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

*ROUGE* merupakan salah satu cara untuk melakukan evaluasi kualitas dari suatu ringkasan yang dibuat secara otomatis. Ringkasan akan dibandingkan dengan *gold summary* yang ada. *Gold summary* merupakan ringkasan dibuat oleh manusia dan menjadi standar bagi teks yang akan diringkas. Pemilihan *ROUGE* dalam evaluasi ringkasan dikarenakan sudah menjadi standar evaluasi dan telah banyak digunakan oleh penelitian lain. Terdapat 2 jenis *ROUGE* yang digunakan dalam melakukan peringkasan yaitu *ROUGE-N* dan *ROUGE-L*. *ROUGE-N* merupakan pencarian nilai *recall* berdasarkan *n-gram* sedangkan *ROUGE-L* menggunakan *longest common subsequence* [12]. *ROUGE-N* digunakan untuk menilai dari sisi *informativeness* atau seberapa banyak kesamaan informasi yang terkandung dalam hasil ringkasan sistem dengan referensi ringkasan. *ROUGE-L* akan menilai dari sisi *fluency* atau seberapa lancar hasil ringkasan yang dihasilkan dengan melihat urutan kesamaan kata dalam suatu kalimat terhadap referensi ringkasan [12].

## 3. ANALISIS DAN DESAIN SISTEM

### 3.1 Analisa Data

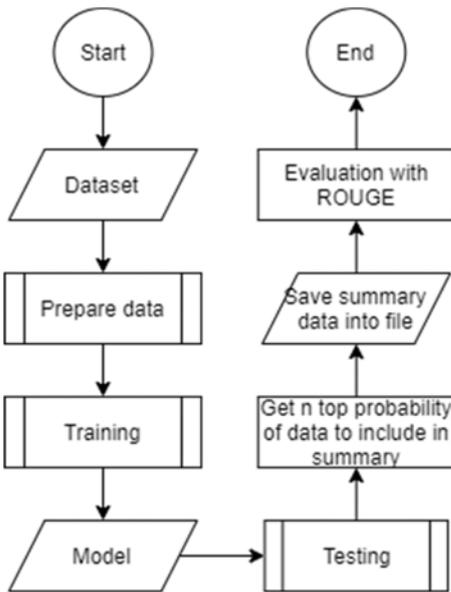
Data yang akan diambil dari *dataset* terdapat 3 bagian yaitu '*gold\_labels*', '*paragraphs*', dan '*summary*'. *Dataset* berita berbahasa Indonesia yang digunakan diambil dari penelitian yang telah disediakan oleh Kurniawan dan Louvan [11]. *Dataset* memiliki format *json* dan dibagi menjadi 3 bagian yaitu untuk *train*, *test*, dan *dev*. Selain itu, *dataset* juga dibagi menjadi 5 *fold* dimana tiap *fold* akan memiliki data sebanyak 18.774 berita. *Dataset train* digunakan untuk proses *training*, *test* untuk proses *test*, dan *dev* untuk proses evaluasi *training*.

Sebelum data digunakan ke dalam program, data akan diolah terlebih dahulu sehingga memudahkan dalam implementasi ke dalam program. Terdapat beberapa hal yang akan dilakukan dalam

pengolahan data. Pertama, mengambil bagian 'gold\_labels', 'paragraphs', dan 'summary' dari dataset. Kemudian dilanjutkan dengan membuka array pada tiap bagian 'paragraphs', 'gold\_labels', 'summary' untuk diambil isinya. Terakhir, isi pada masing-masing bagian akan digabungkan dengan delimiter '<q>'. Pada bagian 'paragraphs' dan 'summary', delimiter ini bertujuan untuk memisahkan tiap kalimat sedangkan pada 'gold\_labels' bertujuan untuk memisahkan label dari tiap kalimat yang akan digunakan sebagai referensi ekstraktif.

### 3.2 Desain Sistem

Dalam melakukan peringkasan secara ekstraktif, gambaran keseluruhan sistem yang akan diterapkan pada penelitian ini dapat dilihat pada Gambar 1. Library yang akan digunakan adalah pytorch dan pytorch-lightning untuk penerapan encoder transformer dan feed forward neural network. Kemudian, untuk model BERT akan menggunakan library huggingface.



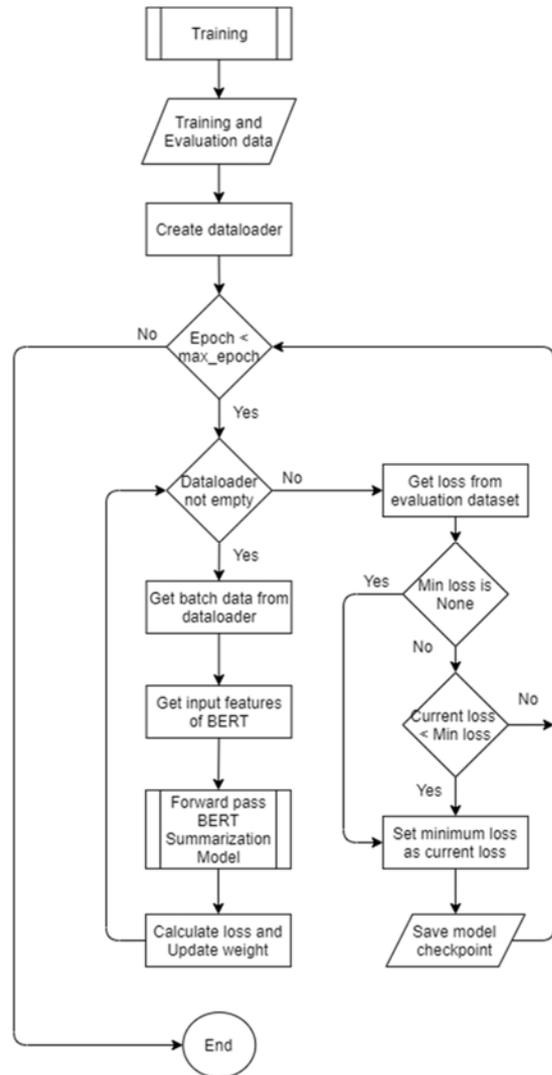
Gambar 1. Alur sistem secara keseluruhan

Sebelum dilakukan proses training, akan dilakukan persiapan terhadap data terlebih dahulu. Mengenai proses persiapan yang dimaksud disini adalah melakukan preprocessing sederhana dengan mengubah tiap data teks menjadi huruf kecil dan dilanjutkan dengan merapikan data dari karakter spasi, tab, dan newline yang berlebihan.

Selanjutnya, proses training akan dimulai dengan input model BERT yang digunakan dan dataset sebagai training dan evaluasi. Kemudian, akan dilakukan inialisasi parameter transformer encoder, optimizer, dan lain-lain. Optimizer yang digunakan adalah AdamW optimizer dan fungsi loss yang akan digunakan adalah binary cross entropy. AdamW optimizer dipilih karena memiliki performa yang baik dengan melakukan modifikasi terhadap weight decay. Setelah itu, training akan dilakukan dalam beberapa epoch. Dalam satu epoch, proses akan dijalankan beberapa kali bergantung pada jumlah batch.

Pencarian loss menggunakan fungsi loss dengan target label sebagai ukuran kebenaran. Selama melakukan proses training, bila suatu epoch terpenuhi maka akan dilakukan proses evaluasi dimana model yang telah dilatih pada semakin baik atau tidak. Ketika dilakukan evaluasi model semakin baik maka model akan

disimpan. Baik tidaknya suatu ditentukan berdasarkan dari loss yang didapat dari proses evaluasi dimana loss kecil menunjukkan bahwa model semakin membaik. Proses training dapat dilihat pada Gambar 2.

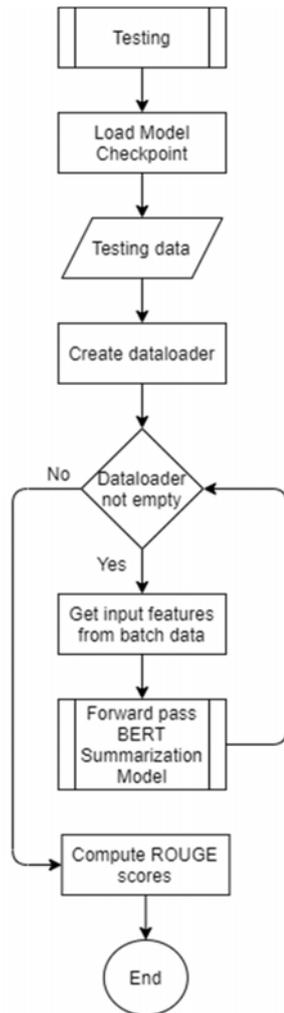


Gambar 2. Alur proses training

Tambahan susunan transformer encoder yang akan digunakan mengikuti dari penelitian Liu dan Lapata [13], dimana model yang digunakan untuk peringkasan adalah transformer encoder untuk mengetahui relasi antar kalimat. Pada model yang digunakan akan mengambil pada level kalimat untuk diteruskan ke dalam layer klasifikasi. Layer klasifikasi akan dihitung berdasarkan dari saliency suatu kalimat yaitu seberapa penting kalimat dari artikel berita yang bersangkutan. Setelah itu, akan dilakukan penghitungan dengan menggunakan fungsi aktivasi sigmoid untuk menghasilkan probabilitas dari tiap kalimat yang akan masuk ke dalam hasil ringkasan [4].

Proses evaluasi akan mirip dengan training hanya saja model tidak dilakukan training, namun langsung digunakan untuk mencari loss dari dataset evaluasi. Semua loss dari dataset akan digunakan untuk menentukan apakah model semakin baik atau tidak. Proses evaluasi bertujuan untuk menghindari terjadinya overfitting pada dataset training.

Setelah model telah terbuat, maka akan dilanjutkan dengan proses *testing* dengan dilakukan pengujian untuk mengetahui performa dari model yang telah dilatih. Mengenai alur proses *testing* dapat dilihat melalui Gambar 3.



Gambar 3. Alur proses *testing*

#### 4. PENGUJIAN SISTEM

Pengujian sistem ini akan dilakukan untuk mencari tahu konfigurasi yang tepat dengan melakukan percobaan *training*. Selain itu, karena adanya keterbatasan *resource* yang dimiliki oleh peneliti maka untuk penelitian ini akan digunakan *resource* yang berasal dari *google colab* untuk melakukan proses *training* dan *testing* dari model. Tiap pengujian yang akan dilakukan akan dibandingkan terhadap referensi ekstraktif dan abstraktif. Selain itu, untuk beberapa pengujian juga akan dibandingkan model IndoBERT dengan model Multilingual BERT [5][10]. Kemudian, untuk pengujian lain selain pengujian awal dan akhir hanya akan diimplementasikan pada *fold* yang pertama saja untuk mempersingkat waktu pengujian pada penelitian ini. Jumlah kalimat yang akan dijadikan sebagai ringkasan yaitu sebanyak 3 kalimat teratas dengan nilai probabilitas tertinggi. Alasan pengambilan 3 kalimat teratas adalah untuk menyamakan dengan pengujian yang telah dilakukan pada penelitian sebelumnya [2][11]. Pengujian pada penelitian ini akan diambil nilai *F1-Score* dari *ROUGE* mengikuti penilaian standar pada penelitian-

penelitian lain yang serupa [1][2][8]. Mengenai pembagian data untuk tiap *fold* yang digunakan untuk pengujian awal dan akhir dapat dilihat pada Tabel 1. Lalu, untuk *hyperparameter* konstan yang tidak akan diubah-ubah pada pengujian ini dapat dilihat pada Tabel 2.

Tabel 1. Pembagian data untuk tiap *fold*

Fold	Train	Dev	Test
1	14262	750	3762
2	14263	749	3762
3	14290	747	3737
4	14272	750	3752
5	14266	747	3761

Tabel 2. Konfigurasi *hyperparameter* konstan

Batch size	Max Epochs	Epsilon	Weight Decay	Optimizer
8	4	1e-8	0.01	AdamW

#### 4.1 Pengujian Awal

Pada pengujian awal ini akan dilakukan dengan menggunakan konfigurasi awal *learning rate* 1e-5, *dropout* 0.1, tumpukan *transformer encoder* sebanyak 2, dan menggunakan *token type ids*. Pengujian awal ini dilakukan untuk mengetahui performa awal dari model sebelum diujikan menggunakan konfigurasi *hyperparameter* lainnya. Mengenai hasil dari pengujian awal untuk referensi ekstraktif dan abstraktif model IndoBERT dapat dilihat pada Tabel 3 dan Tabel 5 sedangkan untuk model Multilingual BERT dapat dilihat pada Tabel 4 dan Tabel 6.

Tabel 3. Pengujian awal pada model IndoBERT menggunakan referensi ekstraktif

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	83.64	82.33	82.53
2	83.17	81.84	82.07
3	84.86	83.56	83.90
4	83.48	82.17	82.41
5	84.73	83.54	83.69
Average	83.97	82.68	82.92

Tabel 4. Pengujian awal pada model Multilingual BERT menggunakan referensi ekstraktif

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	82.99	81.64	81.94
2	84.37	83.07	83.23
3	84.38	83.07	83.43
4	83.76	82.48	82.84
5	83.96	82.64	82.98
Average	83.89	82.58	82.88

**Tabel 5. Pengujian awal pada model IndoBERT menggunakan referensi abstraktif**

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	56.95	51.13	54.89
2	56.81	50.82	54.85
3	57.21	51.19	55.27
4	56.88	50.89	54.94
5	56.94	51.19	55.06
Average	56.95	51.04	55.00

**Tabel 6. Pengujian awal pada model Multilingual BERT menggunakan referensi abstraktif**

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	56.47	50.55	54.46
2	57.22	51.22	55.23
3	57.21	51.20	55.27
4	56.70	50.74	54.83
5	56.61	50.74	54.76
Average	56.84	50.89	54.91

## 4.2 Pengujian Kombinasi Learning Rate dan Dropout

Pada pengujian ini akan dilakukan percobaan training dengan mengubah nilai *learning rate* dan *dropout*. Tujuan dari pengujian ini untuk mencari tahu konfigurasi *learning rate* yang tepat pada model BERT yang digunakan. Selain itu, *learning rate* juga akan menentukan seberapa cepat model belajar dimana semakin tinggi nilai dari *learning rate* maka model akan belajar lebih cepat dan sebaliknya. Kemudian, *learning rate* akan dikombinasikan dengan *dropout* sehingga diharapkan dapat menghindari terjadinya *overfitting*. Mengenai hasil yang didapatkan menggunakan referensi ekstraktif dan abstraktif pada model IndoBERT dapat dilihat pada Tabel 7 dan Tabel 9 sedangkan untuk model Multilingual BERT dapat dilihat pada Tabel 8 dan Tabel 10.

**Tabel 7. Pengujian kombinasi *learning rate* dan *dropout* pada model IndoBERT menggunakan referensi ekstraktif**

Learning rate	Dropout	ROUGE-1	ROUGE-2	ROUGE-L
1e-5	0.1	83.64	82.33	82.53
	0.3	84.12	82.85	83.02
	0.5	83.83	82.51	82.73
2e-5	0.1	84.11	82.84	83.00
	0.3	83.68	82.37	82.62
	0.5	84.24	83.01	83.14
3e-5	0.1	83.74	82.45	82.64
	0.3	84.50	83.22	83.40
	0.5	84.64	83.45	83.52

**Tabel 8. Pengujian kombinasi *learning rate* dan *dropout* pada model Multilingual BERT menggunakan referensi ekstraktif**

Learning rate	Dropout	ROUGE-1	ROUGE-2	ROUGE-L
1e-5	0.1	82.99	81.64	81.94
	0.3	83.84	82.53	82.78
	0.5	83.08	81.72	82.03
2e-5	0.1	84.02	82.77	82.95
	0.3	83.62	82.34	82.52
	0.5	82.49	81.13	81.46
3e-5	0.1	83.69	82.47	82.61
	0.3	82.19	80.76	81.18
	0.5	82.51	81.10	81.45

**Tabel 9. Pengujian kombinasi *learning rate* dan *dropout* pada model IndoBERT menggunakan referensi abstraktif**

Learning rate	Dropout	ROUGE-1	ROUGE-2	ROUGE-L
1e-5	0.1	56.95	51.13	54.89
	0.3	57.21	51.40	55.12
	0.5	57.09	51.24	55.02
2e-5	0.1	57.30	51.45	55.22
	0.3	56.94	51.09	54.90
	0.5	57.38	51.56	55.29
3e-5	0.1	56.69	50.84	54.65
	0.3	57.28	51.44	55.18
	0.5	57.45	51.64	55.33

**Tabel 10. Pengujian kombinasi *learning rate* dan *dropout* pada model Multilingual BERT menggunakan referensi abstraktif**

Learning rate	Dropout	ROUGE-1	ROUGE-2	ROUGE-L
1e-5	0.1	56.47	50.55	54.46
	0.3	57.06	51.26	55.03
	0.5	56.56	50.74	54.57
2e-5	0.1	57.10	51.25	55.06
	0.3	56.91	51.10	54.83
	0.5	56.03	50.17	54.05
3e-5	0.1	56.85	51.07	54.81
	0.3	55.95	50.04	53.97
	0.5	56.07	50.20	54.08

### 4.3 Pengujian Token Type Ids

Pada pengujian ini akan dicoba melatih model tanpa menggunakan *token type ids* yang bertujuan untuk membedakan antara kalimat dengan urutan ganjil dan kalimat dengan urutan genap pada teks. Pengujian ini akan dilakukan menggunakan konfigurasi *learning rate* dan *dropout* terbaik dari pengujian sebelumnya. Tujuan dilakukannya pengujian ini adalah untuk mengetahui apakah hasil ringkasan akan menjadi lebih baik tanpa menggunakan *token type ids* yang berperan sebagai segmen antar kalimat. Mengenai hasil pengujian dengan referensi ekstraktif dan abstraktif yang didapatkan untuk model IndoBERT dapat dilihat pada Tabel 11 dan Tabel 13 sedangkan untuk model Multilingual BERT dapat dilihat pada Tabel 12 dan Tabel 14.

**Tabel 11. Pengujian token type ids pada model IndoBERT menggunakan referensi ekstraktif**

Token type ids	ROUGE-1	ROUGE-2	ROUGE-L
No token type ids	84.36	83.13	83.26
Use token type ids	84.64	83.45	83.52

**Tabel 12. Pengujian token type ids pada model Multilingual BERT menggunakan referensi ekstraktif**

Token type ids	ROUGE-1	ROUGE-2	ROUGE-L
No token type ids	82.99	81.56	81.99
Use token type ids	84.02	82.77	82.95

**Tabel 13. Pengujian token type ids pada model IndoBERT menggunakan referensi abstraktif**

Token type ids	ROUGE-1	ROUGE-2	ROUGE-L
No token type ids	57.23	51.41	55.15
Use token type ids	57.45	51.64	55.33

**Tabel 14. Pengujian token type ids pada model Multilingual BERT menggunakan referensi abstraktif**

Token type ids	ROUGE-1	ROUGE-2	ROUGE-L
No token type ids	56.44	50.50	54.43
Use token type ids	57.10	51.25	55.06

### 4.4 Pengujian Stacked Transformer Encoder

Pada pengujian ini akan dilakukan terhadap jumlah tumpukan *transformer encoder* yang digunakan. Pengujian ini dilakukan untuk mengetahui hasil ringkasan sistem dari klasifikasi yang dilakukan oleh *transformer encoder*. Hasil yang didapatkan untuk pengujian menggunakan referensi ekstraktif dan abstraktif model IndoBERT dapat dilihat pada Tabel 15 dan Tabel 17 sedangkan untuk model Multilingual BERT dapat dilihat pada Tabel 16 dan Tabel 18.

**Tabel 15. Pengujian Stacked Transformer Encoder pada model IndoBERT menggunakan referensi ekstraktif**

N-Stacked Transformer Encoder	ROUGE-1	ROUGE-2	ROUGE-L
1	83.89	82.60	82.75
2	84.64	83.45	83.52
3	84.43	83.16	83.36

**Tabel 16. Pengujian Stacked Transformer Encoder pada model Multilingual BERT menggunakan referensi ekstraktif**

N-Stacked Transformer Encoder	ROUGE-1	ROUGE-2	ROUGE-L
1	84.14	82.92	83.11
2	84.02	82.77	82.95
3	83.61	82.34	82.59

**Tabel 17. Pengujian Stacked Transformer Encoder pada model IndoBERT menggunakan referensi abstraktif**

N-Stacked Transformer Encoder	ROUGE-1	ROUGE-2	ROUGE-L
1	57.08	51.23	54.99
2	57.45	51.64	55.33
3	57.13	51.28	55.07

**Tabel 18. Pengujian Stacked Transformer Encoder pada model Multilingual BERT menggunakan referensi abstraktif**

N-Stacked Transformer Encoder	ROUGE-1	ROUGE-2	ROUGE-L
1	57.17	51.36	55.14
2	57.10	51.25	55.06
3	56.86	51.06	54.84

### 4.5 Pengujian Akhir

Pada pengujian ini akan dilakukan dengan menggunakan konfigurasi terbaik pada seluruh *fold*. Pengujian ini akan dilakukan pada model IndoBERT saja dikarenakan untuk *ROUGE* terbaik

pada pengujian-pengujian sebelumnya adalah saat menggunakan model IndoBERT. Mengenai hasil akhir dari model pada referensi ekstraktif dan abstraktif dapat dilihat pada Tabel 19 dan Tabel 20.

**Tabel 19. Pengujian Akhir Terbaik model BERT menggunakan referensi ekstraktif**

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	84.64	83.45	83.52
2	83.68	82.31	82.60
3	85.03	83.79	84.03
4	84.31	83.08	83.25
5	84.64	83.43	83.62
Average	84.46	83.21	83.40

**Tabel 20. Pengujian Akhir Terbaik model BERT menggunakan referensi abstraktif**

Fold	ROUGE-1	ROUGE-2	ROUGE-L
1	57.45	51.64	55.33
2	56.92	50.90	54.97
3	57.37	51.46	55.39
4	57.16	51.21	55.24
5	56.96	51.18	55.07
Average	57.17	51.27	55.20

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Metode BERT yang diterapkan pada penelitian ini mendapatkan *ROUGE* terbaik dengan konfigurasi *learning rate*  $3e-5$ , *dropout* 0.5, tumpukan *transformer encoder* sebanyak 2, dan menggunakan *token type ids* pada model IndoBERT. Hasil *ROUGE* terbaik lewat pengujian yang telah dilakukan dapat menghasilkan *F1-Score* untuk *ROUGE-1* sebesar 57.17, *ROUGE-2* sebesar 51.27, dan *ROUGE-L* sebesar 55.20 pada referensi abstraktif serta *ROUGE-1* sebesar 84.46, *ROUGE-2* sebesar 83.21, dan *ROUGE-L* sebesar 83.40 pada referensi ekstraktif. Perbedaan nilai *ROUGE* antara referensi ekstraktif dan abstraktif dari pengujian yang telah dilakukan masih cukup signifikan yaitu mendekati sekitar 30%.

### 5.2 Saran

Menggunakan model transformer lain seperti OpenAI GPT, ALBERT, dan BART. Lalu, juga dapat dicoba menggunakan layer klasifikasi yang lain diatas model BERT seperti LSTM.

## 6. DAFTAR REFERENSI

- [1] Al-Maleh, M., & Desouki, S. (2020). Arabic text summarization using deep learning approach. *Journal of Big Data*, 7(1). <https://doi.org/10.1186/s40537-020-00386-7>
- [2] Cheng, J., & Lapata, M. (2016). Neural summarization by extracting sentences and words. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1, 484–494. <https://doi.org/10.18653/v1/p16-1046>
- [3] Clark, K., Khandelwal, U., Levy, O., & Manning, C. D. (2019). What does BERT look at? An analysis of BERT's attention. *ArXiv*. <https://doi.org/10.18653/v1/w19-4828>
- [4] Deng, L., & Liu, Y. (2018). *Deep learning in natural language processing*. Springer. [https://doi.org/10.1007/978-981-10-5209-5\\_11](https://doi.org/10.1007/978-981-10-5209-5_11)
- [5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [6] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165, 113679. <https://doi.org/10.1016/j.eswa.2020.113679>
- [7] Hirschberg, J., & Manning, C. D. (2015). Advances in Natural Language Processing. *Science*, 349(6245), 261–266. <https://doi.org/10.1126/science.aaa8685>
- [8] Ismi, D. P., & Ardianto, F. (2019). Peringkasan Ekstraktif Teks Bahasa Indonesia dengan Pendekatan Unsupervised Menggunakan Metode Clustering. *CYBERNETICS*, 3(02), 90–99. <http://dx.doi.org/10.29406/cbn.v3i02.2290>
- [9] Joshi, A., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). SumCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129, 200–215. <https://doi.org/10.1016/j.eswa.2019.03.045>
- [10] Koto, F., Rahimi, A., Lau, J. H., & Baldwin, T. (2020). IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP. *Proceedings of the 28th International Conference on Computational Linguistics*, 757–770. <https://doi.org/10.18653/v1/2020.coling-main.66>
- [11] Kurniawan, K., & Louvan, S. (2018). Indosum: A new benchmark dataset for Indonesian text summarization. *2018 International Conference on Asian Language Processing (IALP)*, 215–220. <https://doi.org/10.1109/IALP.2018.8629109>
- [12] Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 74–81. [doi: aalanthology.org/W04-1013](https://aclanthology.org/W04-1013)
- [13] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3730–3740. <https://doi.org/10.18653/v1/d19-1387>
- [14] Schmitt, J. B., Debbelt, C. A., & Schneider, F. M. (2017). Too much information? Predictors of information overload in the context of online news exposure. *Information Communication and Society*, 21(8), 1151–1167. <https://doi.org/10.1080/1369118X.2017.1305427>
- [15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips), 6000–6010. <https://dl.acm.org/doi/10.5555/3295222.3295349>