Implementasi LexRank dan BERT2GPT dalam Auto Summarization Teks Bahasa Indonesia

Tristan Bey Kusuma[a1], I Made Widiartha[a2], I Putu Gede Hendra Suputra[a3]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]tristanbeykusuma@gmail.com
[2]madewidiartha@unud.ac.id
[3]hendra.suputra@unud.ac.id

Abstract

In Indonesia, with the rapid growth of internet and social media usage, the amount of information produced in the Indonesian language has reached significant levels. This creates challenges in managing and understanding this information quickly and efficiently. Text summarization has emerged as a potential solution to help users organize and summarize information, enabling easier and more efficient access to relevant content. This study discusses the development of an Indonesian text summarization model using the LexRank algorithm. The results show that this model can produce accurate and concise summaries, with ROUGE-L result of 0.91 and also a ROUGE-1 result of 0.31. Developing an Indonesian text summarization model is important because it can help users manage and understand information quickly and efficiently. This study provides a positive contribution to the development of Indonesian text summarization models, by providing evidence that the LexRank model can produce accurate and concise summaries.

Keywords: *Text mining, Summarization, LexRank, BERT, ROUGE*

1. Pendahuluan

Di Indonesia, dengan pertumbuhan pesat dalam penggunaan internet dan media sosial, banyaknya informasi yang dihasilkan dalam bahasa Indonesia telah mencapai tingkat yang signifikan. Hal ini menciptakan tantangan dalam mengelola dan memahami informasi tersebut dengan cepat dan efisien. Text summarization atau peringkasan teks muncul sebagai solusi yang potensial untuk membantu pengguna menyusun dan meringkas informasi, memungkinkan akses yang lebih mudah dan efisien terhadap konten yang relevan [1]. Beberapa alasan mengapa proyek pemrosesan bahasa alami untuk text summarization di Indonesia penting antara lain: Volume besar teks dalam bahasa Indonesia yang tersedia di internet dan media sosial membuat sulit bagi pengguna untuk mengonsumsi semua informasi dengan cepat. Peningkatan aksesibilitas, dengan adanya ringkasan teks, orang yang memiliki keterbatasan waktu atau perhatian dapat dengan mudah mendapatkan pemahaman yang cepat. Aplikasi Bisnis dan Riset, dalam konteks bisnis dan riset, text summarization dapat digunakan untuk mengidentifikasi tren pasar dan menganalisis sentimen publik.

Sebelumnya, telah diteliti implementasi algoritma TextRank dalam melakukan peringkasan 50 artikel teks berita yang diambil dari laman berita digital. Algoritma TextRank melakukan pemeringkatan dokumen teks berdasarkan suatu graf yang diperoleh dari ekstraksi teks dokumen. Hasilnya menunjukkan nilai ROUGE yaitu 0.167. Algoritma LexRank juga memanfaatkan bentuk graf, namun disini juga didefinisikan sebuah threshold kesamaan dokumen, sehingga hanya dokumen diatas threshold yang akan disimpan [2].

Kami mengembangkan model text summarization berbahasa Indonesia dengan algoritma LexRank yang juga dapat dimodifikasi untuk melaksanakan peringkasan multi-dokumen untuk meningkatkan kualitas ringkasan bahasa Indonesia. Selain itu, permasalahan lainnya yang melandasi ini adalah untuk mengatasi fitur linguistik Bahasa Indonesia dalam proses peringkasan

dengan stop word removal, cleaning, dan menerapkan stemming atau lemmatization yang khusus. Teknik yang digunakan di sini adalah teknik peringkasan teks abstraktif dengan pemeringkatan kalimat berdasarkan skor graf [3]. Selain LexRank, kami juga dapat mengangkat metode abstraktif dalam meringkas dokumen. Model-model machine learning seperti jaringan syaraf tiruan atau transformer memiliki performa yang cukup baik dalam task-task terkait pemrosesan data tekstual. Maka kami juga membandingkan hasil ringkasan yang dihasilkan model ringkasan hanya dengan LexRank dengan sebuah model lainnya dimana hasil ekstraktif LexRank kemudian masuk kembali kedalam model abstraktif yaitu BERT2GPT.

2. Metode Penelitian

Untuk mencapai tujuan peringkasan teks berita Bahasa Indonesia ini, maka metode pelaksanaan melewati beberapa langkah yaitu Pengumpulan Data, Preprocessing, Training, dan Evaluation.

2.1 Pengumpulan Data

Pada proses pengumpulan data, data diperoleh dari benchmark dataset peringkasan teks bahasa indonesia, yaitu sekitar seribu artikel berita yang diberi token dari situs Shortir.com, situs agregator berita. Dataset teks berita ini berupa beberapa dokumen yang berisi paragraf-paragraf teks yang disertai ringkasan teks tiap dokumen tersebut. Peringkasan teks ini dilakukan secara manual.

2.2 Preprocessing

Text preprocessing merupakan sebuah metode yang digunakan untuk mempermudah representasi teks dan membuat dokumen menjadi konsisten. Proses text mining sangat bergantung dari metode text preprocessing yang digunakan [4]. Berikut ini adalah text preprocessing yang digunakan:

a. Case Folding
   Case folding melibatkan pengolahan data apa pun dalam teks atau opini dalam dataset menjadi karakter huruf kecil. Hal ini bertujuan agar semua data memiliki format yang sama sehingga pemrosesan data dapat memudahkan dalam tahap pemodelan.
b. Stopword Removal
   Stopword removal merupakan proses penghapusan kata-kata dalam teks yang memiliki makna minim, seperti kata penghubung, kata ganti, dan sebagainya.
c. Tokenization
   Tokenization adalah proses mengubah suatu sekuens teks menjadi bagian terkecilnya, pada konteks ini akan mengubah input kalimat menjadi token atau kata [4].

2.3 Algoritma LexRank

LexRank merupakan algoritma untuk peringkasan dokumen tunggal atau multi. LexRank menggunakan pendekatan berbasis centroid dalam proses peringkasan. Pemberian skor pada kalimat dilakukan dengan menggunakan metode grafik. LexRank digunakan untuk menghitung penting atau tidaknya suatu kalimat berdasarkan konsep sentralitas vektor eigen dalam representasi grafik kalimat. Untuk mengatasi masalah redundansi dalam peringkasan multi-dokumen, LexRank juga menerapkan langkah heuristik pemeringkatan ulang yang membuat ringkasan dengan menambahkan kalimat dalam urutan peringkat, namun membuang kalimat apa pun yang terlalu mirip dengan kalimat yang sudah ditempatkan dalam ringkasan [3]. Tahap-tahapnya meliputi :

- Hitung Term Frequency-Inverse Document Frequency (TF-IDF)
  Untuk mendapatkan frekuensi istilah pertama-tama kita menghitung frekuensi kata (menghitung jumlah kemunculan kata) di setiap kalimat. Kemudian kita membagi frekuensi kata. dengan jumlah kata pada tiap kalimat untuk mendapatkan TF tiap kata. Untuk menghitung IDF, pertama-tama kita harus menghitung jumlah kalimat yang memiliki kata tertentu (yaitu Menghitung kalimat yang memiliki kata tertentu). Kemudian menghitung IDF

dengan mencari Log(jumlah total kalimat / jumlah kalimat memiliki kata (w)). Hasilnya TF dikali IDF adalah nilai TF-IDF nya.

- Membuat Graf Kalimat
  Asumsikan grafik kalimat sebagai Matriks 2D di mana |s| adalah jumlah kalimat. Disini kita akan memiliki matriks[i][j] dengan ordo sebesar banyak kalimat dan akan mewakili bobot antara kalimat i dan j.
- Kalkulasi Bobot Graf
  Hitung bobot grafik (yaitu nilai matriks[i][j]), misalnya matrix[0][0] menunjukkan bobot antara kalimat 1 dan 1, demikian pula matriks[0][1] menunjukkan bobot antara kalimat 1 dan 2. Disini menggunakan rumus kesamaan cosinus untuk menghitung bobot.

$$\text{idf-modified-cosine}(x,y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}}$$

Gambar 1. Rumus Kalkulasi Bobot Graf

Rumus di atas dapat dipecah menjadi dua bagian, pembilang dan penyebut yang selanjutnya dapat dipecah menjadi dua bagian penyebut (d1 x d2).

$$pembilang(s_i, s_j) = \sum_{w \in (s_i \cap s_j)} \left(tf_{s_i}(w)\, tf_{s_j}(w)(idf(w))^2\right) \quad (1)$$

$$penyebut, d1(s_i) = \sqrt{\sum_{w \in s_i} \left(\text{tf}_{s_i}(w)\,(\text{idf}(w))^2\right)} \quad (2)$$

$$penyebut, d2(s_j) = \sqrt{\sum_{w \in s_j} \left(tf_{s_j}(w)(idf(w))^2\right)} \quad (3)$$

Pada persamaan tersebut w merepresentasikan satu kata. Jika nilai penyebut sama dengan nol atau kurang dari nol, maka matriks[i][j] bernilai nol. Pada persamaan, (tf(s[i])(w)) berarti nilai frekuensi *term* (tf) dari sebuah kata dalam kalimat ke-i. Demikian pula (tf(s[j](w)) berarti jumlah *term* (tf) kata dalam kalimat ke-j. Dan idf(w) berarti frekuensi dokumen terbalik (idf) dari sebuah kata. Nilai ini dapat kita peroleh dari perhitungan tf dan idf sebelumnya [3].

- Menghitung Valensi/Derajat Graf dengan Nilai Threshold
  Bandingkan bobot matriks dengan ambang batas, misalnya 0,1 dan pertahankan derajat untuk setiap kalimat. Kita dapat menyimpan derajat dalam list |s| lainnya (jumlah kalimat). Jika matriks[i][j] lebih besar dari ambang batas, kita ganti nilai matriks[i][j] dengan 1 dan tingkatkan jumlah derajat[i] sebanyak 1. Jika tidak, kita ganti nilai matriks[i][j] dengan 0. Jadi sekarang matriks kita hanya berisi 1 atau 0.
- Menyesuaikan nilai dengan nilai derajat
  Kami sekarang mengganti nilai matriks[i][j] kami sesuai dengan nilai derajat. Namun jika derajatnya 0, kita menggantinya dengan 1. Lalu menghitung nilai matriks sebagai matriks[i][j] / derajat[i]. Misalnya nilai 1 dengan derajat kalimat tersebut 2 maka menghasilkan nilai 0,5.
- Hitung sentralitas/centroid
  Kami sekarang menilai kalimat tersebut. Untuk menemukan kalimat yang akan menjadi ringkasan kami. Untuk menilai kalimat, kami menghitung sentralitas.

$$p = B^T P \quad (4)$$

B^T adalah transpos matriks (yaitu baris diganti dengan kolom) yang kita hitung sebelumnya dan P adalah array dengan panjang |s| (jumlah kalimat) masing-masing mempunyai nilai [1 / |s|](di

sini |s|=jumlah kalimat, misalnya jumlah kalimat=4 maka 1/4=0,25)]. Hasilnya adalah sebuah array dengan panjang |s| juga.

- Mendapatkan rata-rata
  Temukan skor rata-rata untuk memilih kalimat yang akan menjadi ringkasan dokumen. Misalnya hasil p akhir = [0,25 , 0,25 , 0,25 , 0,25] maka rata-ratanya 0,25
- Tampilkan Peringkasan
  Disini yaitu tahap menampilkan kalimat yang memiliki rata-rata p[i] akhir >= rata-rata.

Dalam penggunaannya algoritma LexRank memiliki dua hyperparameter yang penting. Pertama, *summarysize* yaitu ukuran peringkasan yang dihasilkan. Kemudian juga terdapat *threshold* yaitu nilai yang digunakan untuk menentukan apakah suatu kata atau frasa penting untuk ringkasan. Misalnya, jika *threshold* diatur menjadi 0,5, maka hanya kata atau frasa yang memiliki skor kepentingan lebih besar dari 0,5 yang akan dimasukkan ke dalam ringkasan [3].

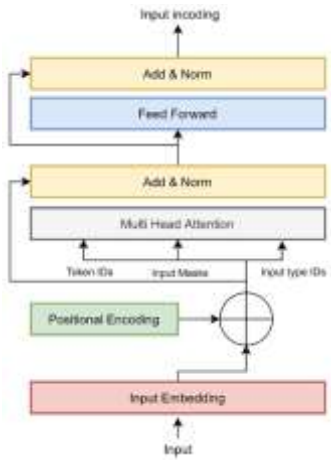
2.4 Algoritma BERT2GPT

Model BERT2GPT adalah alat yang ampuh untuk membuat ringkasan teks Indonesia yang ringkas dan akurat. Model ini bekerja dengan menggabungkan dua model pra-latih yang sudah disetel dengan baik, BERT-base-indonesian-1.5G dan GPT2-small-indonesian-522M [5].

a. BERT-base-indonesian
   Model pra-latih ini merupakan tahap pertama dalam proses ringkasan. Ia didasarkan pada arsitektur BERT (Bidirectional Encoder Representations from Transformers) yang populer, yang khusus disesuaikan untuk bahasa Indonesia. BERT unggul dalam memahami konteks dan makna kata-kata dalam sebuah kalimat, sehingga memberikan representasi yang kaya dari teks masukan.
b. GPT2-small-indonesian
   Model pra-latih kedua ini berfungsi sebagai "pembuat ringkasan" dalam sistem. Ia termasuk dalam keluarga GPT (Generative Pre-training Transformer), yang terampil dalam menghasilkan urutan teks baru mengikuti pola tertentu. Dalam hal ini, GPT2 dilatih secara khusus pada teks Indonesia dan disetel dengan baik untuk tugas-tugas pembuatan ringkasan menggunakan dataset yang disebutkan sebelumnya.

Menggunakan kedua model tersebut, model ini melalui beberapa proses:

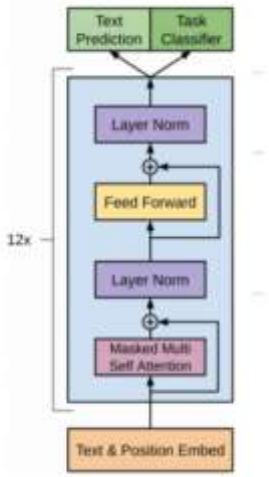
a. Pemrosesan Input
   BertTokenizer memproses teks tersebut, memecahnya menjadi unit-unit kecil yang disebut token. Token-token ini kemudian dikonversi menjadi representasi numerik.
b. Pengodean BERT



Gambar 2. Arsitektur Encoding BERT

Token-token yang telah diproses dimasukkan ke dalam model BERT-base-indonesian-1.5G. Tahap ini menganalisis hubungan antar kata dan kalimat, menangkap makna dan konteks keseluruhan dari teks masukan. BERT memproses teks melalui mekanisme perhatian multi-layernya kemudian menghasilkan representasi kontekstual yang komprehensif dari masukan.

c. Penerjemahan/Peringkasan dengan GPT-2



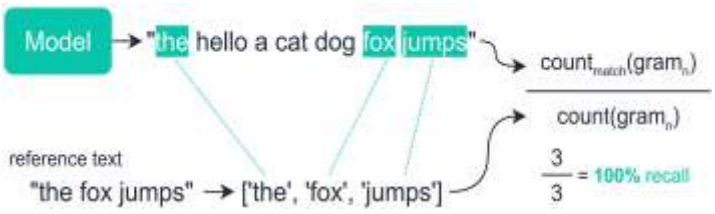
Gambar 3. Arsitektur Encoding GPT-2

Representasi yang dikodekan dari BERT kemudian diteruskan ke model GPT2-small-indonesian-522M. GPT2 memanfaatkan pengetahuannya tentang bahasa Indonesia dan pola pembuatan ringkasan untuk menghasilkan ringkasan yang ringkas dan informatif dari teks masukan. Sehingga, model ini akan menghasilkan ringkasan kata demi kata.

d. Pemformatan Keluaran
Ringkasan yang dihasilkan, dalam bentuk token, diubah kembali menjadi teks Indonesia yang dapat dibaca manusia. Akhirnya, ringkasan yang dihasilkan memberikan Anda gambaran umum yang cepat dan akurat dari konten asli.

Model tersebut telah di fine-tuning menggunakan dataset besar teks Indonesia. Model ini dilatih untuk meminimalkan fungsi loss tertentu, yang dirancang untuk mendorong ringkasan yang akurat dan ringkas [5].

2.5 Evaluasi Algoritma



Gambar 4 Evaluasi Peringkasan ROUGE

Evaluasi peringkasan ekstraktif dan abstraktif ini menggunakan metrik evaluasi ringkasan otomatis, ROUGE. ROUGE (Recall-Oriented Understanding for Gisting Evaluation) adalah metrik berbasis recall untuk ringkasan panjang tetap yang didasarkan pada n-gram. ROUGE dihitung dengan melihat kata pada peringkasan dengan model yang juga terdapat peringkasan asli atau manual. Terdapat Rogue-1 yang hanya menghitung jumlah unigram (satu token) yang sama

antara ringkasan dan referensi, serta Rogue-L yang melihat jumlah urutan token terpanjang yang muncul secara berurutan [6].

3. Hasil dan Pembahasan

Skenario evaluasi dilakukan dengan membandingkan hasil ROGUE pada tiap besaran hyperparameter yaitu melakukan tuning. Serta, juga dibandingkan performa antara model LexRank dengan model LexRank yang kemudian diproses BERT2GPT.

3.1 LexRank

Hasil hyperparameter tuning ukuran ringkasan atau *summarysize* menunjukkan hasil ROGUE yang dapat dilihat di Tabel 1.

Tabel 1. Hasil Nilai ROUGE Summary Size LexRank

Ukuran Summary	ROGUE-L	ROGUE-1
2	0.5265103685405912	0.30093451857904147
3	0.6500174653729165	0.3293875973088481
4	0.7265743748560927	0.3271475994537218
5	0.7920252878770672	0.325573663747843
6	0.8500251784278774	0.32332161418609934
7	0.8884659371209487	0.3132521000754712
8	0.9060894701138378	0.30056997706459976
9	0.9256160149138252	0.2947154690069601

Hasil ini menunjukkan bahwa nilai ukuran ringkasan yang menghasilkan performa terbesar adalah 6 hingga 8 dimana nilai ROGUE-L masih diatas 0.9 dan nilai ROGUE-1 masih diatas 0.3. Kemudian hasil hyperparameter tuning untuk nilai *threshold* menghasilkan nilai yang dapat dilihat pada Tabel 2 ini.

Tabel 2. Hasil Nilai ROUGE Threshold LexRank

Threshold	ROGUE-L	ROGUE-1
0.1	0.88846593712	0.31325210007
0.2	0.84567060275	0.29462353722
0.01	0.88369001727	0.30600628753
0.001	0.88369001727	0.30600628753

Dari 4 nilai *threshold* ini ditemukan bahwa nilai 0.1 menghasilkan nilai ROGUE paling tinggi yaitu 0.88. Ini menunjukkan bahwa nilai tersebut menghasilkan performa yang cukup baik. Kemudian setelah melakukan peringkasan dengan nilai *summarysize* 7 dan *threshold* 0.1, menghasilkan nilai ROGUE-L sebesar 0.91 dan ROGUE-1 sebesar 0.31.

3.2 BERT2GPT

Selain LexRank, kami juga membandingkan performa ringkasan dengan hasil ringkasan abstraktif dengan BERT2GPT. Hasil dari peringkasan ekstraktif dengan LexRank yang ukuran ringkasannya 15 kalimat diproses kembali oleh BERT2GPT untuk menghasilkan ringkasan yang lebih pendek. Hasilnya menunjukkan nilai ROGUE-L rata-rata sebesar 0.12606 dan ROGUE-1

rata-rata sebesar 0.04033. Hasil recall ini menunjukkan bahwa walaupun BERT2GPT dapat menghasilkan ringkasan yang lebih pendek dengan mudah serta ukuran ringkasannya dapat diubah sesuai jumlah token yang diinginkan, hasilnya tidak dapat memperoleh performa yang lebih tinggi.

4. Kesimpulan

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa model peringkasan teks bahasa Indonesia dengan algoritma LexRank dapat menghasilkan ringkasan yang akurat dan ringkas. Hasil evaluasi ROUGE menunjukkan bahwa model ini dapat menghasilkan nilai ROGUE-L sebesar 0.91 dan ROGUE-1 sebesar 0.31. Hasil penelitian ini juga menunjukkan bahwa model peringkasan abstraktif dengan BERT2GPT tidak dapat menghasilkan performa yang lebih tinggi dimana ROGUE-L menghasilkan nilai 0.12 dan ROGUE-1 senilai 0.04. Hal ini disebabkan oleh beberapa faktor, antara lain:

- BERT2GPT lebih cocok untuk menghasilkan ringkasan yang kreatif dan informatif, sedangkan LexRank lebih cocok untuk menghasilkan ringkasan yang ringkas dan akurat.
- BERT2GPT membutuhkan lebih banyak data untuk dilatih, sehingga sulit untuk diterapkan pada dataset yang kecil.

Secara keseluruhan, penelitian ini memberikan kontribusi positif bagi pengembangan model peringkasan teks bahasa Indonesia. Hasil penelitian ini dapat digunakan sebagai pedoman untuk meneliti metode peringkasan teks bahasa Indonesia yang lebih baik di masa mendatang.

Daftar Pustaka

[1] F. Retkowski, “The Current State of Summarization,” *Beyond Quantity: Research with Subsymbolic AI*, 2023, doi: 10.48550/arXiv.2305.04853.

[2] M. A. Zamzam, C. Crysdian, and K. F. H. Holle, “Sistem Automatic Text Summarization Menggunakan Algoritma TextRank,” *Jurnal Ilmu Komputer dan Teknologi Informasi*, vol. 12, no. 2, Sep. 2020, doi: 10.18860/mat.v12i2.8372.

[3] Halimah, S. Agustian, and S. Ramadhani, “Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank,” *Jurnal CoSciTech*, vol. 3, no. 3, pp. 371–381, Dec. 2022.

[4] T. Aksoy, S. Celik, and S. Gulsecen, “Data Pre-processing in Text Mining" in *Who Runs The World: Data*, Istanbul University Press, 2020, pp. 122–144, doi: 10.26650/B/ET06.2020.011.07.

[5] M. Nasari, A. Maulina, and A. Girsang, “Abstractive Indonesian News Summarization Using BERT2GPT,” in *2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Nov. 2023, pp. 369–375, doi: 10.1109/ICITISEE58992.2023.10405359.

[6] A. N. Vora, R. M. Jain, A. S. Shah, and S. Sonawane, “Extractive Summarization using Extended TextRank Algorithm,” in *Proc. 21st Int. Conf. Natural Language Processing (ICON)*, Chennai, India, Dec. 2024, pp. 462–471.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong