NARATIF : Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika
Vol. 06 No. 02 Desember 2024
P-ISSN: 2656-7377 || E-ISSN: 2714-8467

PERINGKAS BERITA OTOMATIS BERBASIS *WEBSITE* DENGAN METODE *TEXT RANK*

Daflah Tsany Gusra[1], Kenneth Marlon Tan[2], Ricky Cangniago[3], Viny Christanti M[4].
Fakultas Teknologi Informasi, Teknik Informatika[1,2,3,4]
Universitas Tarumanagara[1,2,3,4]
Jakarta Barat, Indonesia
e-mail: daflah.535220177@stu.untar.ac.id[1] kenneth.535220183@stu.untar.ac.id[2]
ricky.535220210@stu.untar.ac.id[3]

Abstrak

Rangkuman berita otomatis menjadi semakin penting dalam era informasi saat ini yang dipenuhioleh banyak artikel berita *online*. Tujuannya adalah untuk menyajikan informasi yang relevan secara ringkas dari data teks yang besar. Metode seperti *TextRank* menjadi sebuah metode yang efisien dalam mengatasi tugas seperti ini dengan bantuan perpustakaan seperti Summa dan BeautifulSoup. Penelitian inibertujuan untuk menguji kinerja metode *TextRank* dalam menghasilkan rangkuman berita, dengan fokus pada akurasi, efisiensi, dan aplikabilitasnya di berbagai sumber berita dan topik. Dengan mengumpulkanartikel dari berbagai platform *online*, peringkas berita otomatis pada penelitian ini termasuk peringkasanekstraktif, program ini hanya dapat membaca dokumen tunggal. Akurasi program diuji dengan membandingkan hasil rangkuman dengan artikel aslinya melalui tautan URL. Dengan kata lain, program mampu merangkum sekitar 80% dari konten artikel, dengan panjang rata-rata 75 kata dari total 387 kata.Telah dilakukan pengujian untuk melihat berapa persentase akurasi hasil ringkasan dengan artikel asli dengan metode MMR *Accuracy* dengan menyebarkan responden, dari hasil tes yang menunjukkan akurasisekitar 71%. Secara kesimpulan, pendekatan ini menggabungkan berbagai metode untuk menyajikan versiringkas dari artikel berita, dengan *TextRank* untuk mengidentifikasi informasi kunci, Summa untukmerangkum, dan BeautifulSoup untuk pengambilan data web. Hasilnya adalah rangkuman singkat yang mencakup inti dari artikel-artikel tersebut.
Kata kunci: BeautifulSoup, Peringkas Berita, Summa, *TextRank.*

Abstract

Automatic news summarization is increasingly vital in today's information age inundated with numerous online news articles. Its primary aim is to provide concise, relevant information from vast textualdata. Techniques like TextRank have emerged as efficient methods in tackling such tasks, often aided by libraries like Summa and BeautifulSoup. This study seeks to evaluate the performance of TextRank in generating news summaries, with a specific focus on accuracy, efficiency, and applicability across diversenews sources and topics. By collecting articles from various online platforms, the automated news summarization in this study encompasses extractive summarization, albeit limited to processing single documents. Program accuracy is assessed by comparing the summary outputs with the original articles viaURL links. In essence, the program is capable of summarizing approximately 80% of the article content, averaging 75 words from a total of 387 words. Testing reveals an accuracy rate of around 71% when comparing the summary results with the original articles using the MMR Accuracy method with distributed respondents. In conclusion, this approach amalgamates various methods to present concise versions of news articles, leveraging TextRank for key information identification, Summa for summarization, and BeautifulSoup for web data extraction. The result is a succinct summary encapsulating the essence of the articles.
Keywords: BeautifulSoup, News Summazation, Summa, TextRank.

I. PENDAHULUAN

Era teknologi saat ini, *website* daring menjadi sumber alternatif untuk mencari artikel dan berita [1]. Banyaknya artikel dan berita membuat peringkas berita otomatis menjadi penting untuk efisiensi.Peringkasan berita terdiri dari dua tipe yaitu ekstraktif dan abstraktif. Ekstraktif memilih kalimat dari dokumen asli, sedangkan abstraktif menginterpretasi teks melalui transformasi kalimat [2]. Rangkuman berita otomatis menggunakan teknik pemrosesan bahasa alami untuk menyederhanakan pengambilan informasi [2]. Dengan menyusutkan volume teks menjadi rangkuman yang ringkas, metode rangkuman otomatis bertujuan untuk menyederhanakan proses pengambilan informasi, memungkinkan pengguna untuk mengakses konten relevan dengan cepat dan efektif.

Perkembangan artikel berita *online* di berbagai platform, telah memperburuk kebutuhan akan teknik rangkuman yang kuat. Dengan jutaan artikel yang dipublikasikan setiap hari, individu menghadapi tugas yang menakutkan untuk menyaring sejumlah besar informasi untuk mendapatkan wawasan kunci [3]. Dalam konteks ini, algoritma seperti *TextRank* dengan asisten perpustakaan seperti Summa, dan BeautifulSoup telah menjadi terkenal karena kemampuannya untuk menguraikan data teks kompleks menjadi rangkuman yang mudah dicerna [4].

PageRank merupakan algoritma yang digunakan oleh mesin pencarian *Google* yang memberikan bobot numerik pada setiap dokumen dengan tujuan untuk mengukur hubungan kepentingan dalam kumpulan dokumen, *PageRank* juga menjadi referensi untuk metode *TextRank* [5].

TextRank merupakan sebuah algoritma peringkat berbasis grafik, menggunakan representasi grafik dari teks untuk mengidentifikasi kalimat dan frasa penting, merankingnya berdasarkan kepentingan mereka dalam dokumen [6], [7]. Summa, di sisi lain, adalah perpustakaan *Python* yang dirancang khusus untuk tugas rangkuman teks. Ini menggunakan algoritma seperti *TextRank* untuk menghasilkan rangkuman dengan mengekstrak kalimat kunci dari teks masukan. BeautifulSoup, sementara itu, memfasilitasi *web scraping*, memungkinkan pengguna untuk mengekstrak data dari dokumen HTML dengan mudah [4]. Bersama-sama, metode ini menawarkan kumpulan alat untuk rangkuman berita otomatis, masing-masing dengan pendekatan dan keunggulan uniknya.

Adapun alasan menggunakan metode *TextRank* antara lain; tidak memerlukan data terlatih atau data *training*, menjadikannya ideal digunakan di berbagai bahasa dan domain tanpa memerlukan sumber daya dan waktu yang signifikan untuk pelatihan model; fleksibilitas, *TextRank* dapat digabungkan dengan berbagai teknik pra dan pasca-pemrosesan untuk meningkatkan kualitas rangkuman [4]. Ini termasuk penggunaan teknik Pemrosesan Bahasa Alami untuk mengidentifikasi entitas bernama, frasa penting, dan lainnya yang dapat dimasukkan ke dalam proses pemilihan kalimat.

Jika dibandingkan dengan metode berbasis *Deep Learning* seperti *transformer* dan LSTM, yang juga menghasilkan ringkasan yang koheren dan kontekstual, metode tersebut memerlukan sumber daya komputasi yang signifikan serta waktu yang lebih lama untuk dilatih. Di sisi lain, metode berbasis statistik atau *Supervised Learning*, seperti *Naive Bayes* atau SVM, sering kali membutuhkan data latih yang besar dan ekstensif. Sedangkan *TextRank* menawarkan alternatif yang sederhana namun efektif dengan struktur algoritma yang mudah diimplementasikan, menjadikannya pilihan yang baik untuk pengembangan yang cepat dan efektif, maka dari itu kami memilih metode *TextRank* untuk penelitian kami.

II. TINJAUAN PUSTAKA

Tinjauan pustaka dalam penelitian ini berfokus pada landasan teori dan kajian literatur yang mendukung pengembangan sistem peringkas berita otomatis menggunakan metode *TextRank*. Dalam era informasi yang terus berkembang, kebutuhan untuk menyaring dan merangkum informasi dari berbagai sumber berita menjadi semakin penting. Adapun beberapa aspek kunci yang perlu dipertimbangkan, seperti:

1. Peringkasan Berita Otomatis

   Peringkasan berita otomatis yang bertujuan untuk menyederhanakan pengambilan informasi dari teks panjang menjadi ringkasan yang lebih ringkas. Terdapat dua tipe peringkasan yaitu ekstraktif, yang memilih kalimat dari teks asli, dan abstraktif, yang menghasilkan kalimat baru berdasarkan interpretasi teks.

2. Metode *TextRank*

   TextRank merupakan algoritma berbasis graf yang digunakan untuk menilai kepentingan kalimat dalam dokumen. Algoritma ini terinspirasi dari *PageRank* dan mengidentifikasi kalimat kunci berdasarkan kesamaan antar kalimat, sehingga dapat menghasilkan ringkasan yang informatif.

3. *Library* Pendukung

   Dua perpustakaan penting dalam penelitian ini antara lain Summa, yang menerapkan algoritma *TextRank* untuk merangkum teks, dan BeautifulSoup, yang memfasilitasi pengambilan data dari halaman web. Kombinasi dari kedua *library* ini memungkinkan sistem untuk meringkas artikel dengan efisien.

4. Kinerja dan Akurasi

   Penelitian menunjukkan bahwa metode *TextRank* dapat merangkum sekitar 70-80% konten artikel dengan akurasi rata-rata 71% melalui pengujian dengan berbagai sumber berita.

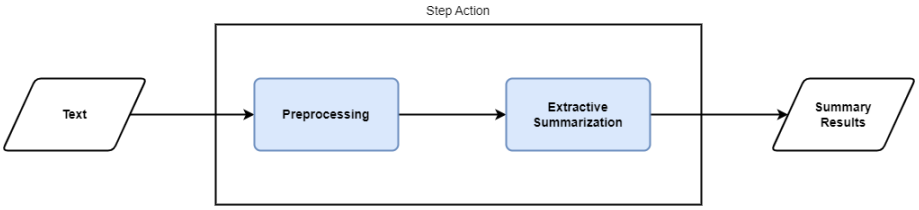
5. Penelitian Terkait

   Berbagai studi sebelumnya menunjukkan efektivitas *TextRank* dalam aplikasi *natural language processing*, termasuk analisis sentimen dan pengenalan entitas bernama, memperkuat relevansi metode ini dalam konteks peringkasan berita otomatis.

Dengan demikian, penelitian ini berkontribusi pada pemahaman lebih lanjut tentang penerapan metode *TextRank* dalam peringkasan berita otomatis, serta tantangan dan peluang yang ada di dalamnya.

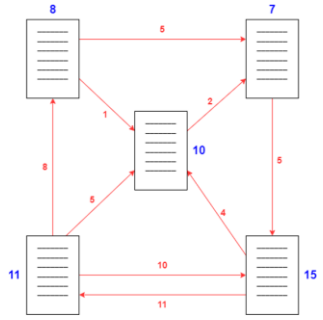
III. METODE PENELITIAN

Metode Penelitian dimulai dengan pra pemrosesan (*preprocessing*) yaitu tokenisasi, *lowercase*, stopword *removal, stemming,* dan pembuatan graf diikuti oleh ringkasan ekstraktif (*extractivesummarization*) dimana teks dianalisis dan disajikan sedemikian rupa sehingga siap untuk disajikan [8]. Ringkasan Ekstraktif bertujuan untuk mengekstrak kalimat penting dari seluruh dokumen.



Gambar 1. *Text Rank Diagram Processing*

Metode yang digunakan adalah *TextRank*. *TextRank* merupakan model peringkat berbasis graf dalam pemrosesan teks yang diusulkan oleh Mihalcea dan Paul. Ada dua pendekatan pembelajaran tanpa pengawasan yang diterapkan: satu untuk ekstraksi kata kunci dan satu untuk ekstraksi kalimat. *TextRank* itu sendiri adalah perluasan dari *PageRank*. Algoritma *TextRank* terinspirasi oleh algoritma *PageRank*, yang utamanya digunakan untuk menentukan peringkat halaman web dalam pencarian *online*.



Gambar 2. Algoritma Metode *Pagerank*

Algoritma *PageRank* menilai pentingnya halaman *web* berdasarkan jumlah dan kualitas tautan yang mengarah ke halaman tersebut. Halaman dengan lebih banyak "suara" dari halaman berkualitas tinggiakan mendapatkan peringkat lebih tinggi, dengan memperhitungkan juga *PageRank* halaman yang memberikan tautan [9].

Konteks *TextRank*, kesamaan antara kalimat-kalimat digunakan untuk menentukan bobot tepi (*edges*) dalam graf tidak berarah. menurut kriteria kesamaan yang dipilih, kalimat yang lebih mirip satu sama lain akan memiliki bobot tepi yang lebih besar, yang pada gilirannya mempengaruhi peringkat akhir dari setiap kalimat dalam graf. Algoritma *TextRank* kemudian mengidentifikasi kalimat-kalimat dengan peringkat tertinggi sebagai yang paling penting untuk dimasukkan dalam ringkasan.

Implementasi digunakan juga konsep *sentence similiarity* dimana *sentence similarity* merupakan konsep untuk menghitung kesamaan antara dua kalimat dengan melihat berapa banyak kata yang sama antara keduanya [10]. Untuk menghindari bias terhadap kalimat yang panjang, kita menggunakan logaritma dari jumlah kata di masing-masing kalimat sebagai pembagi. Ini membantu memberikan perbandingan yang lebih adil, terutama ketika membandingkan kalimat dengan panjang yangberbeda.

$$Similarity(S_i, S_j) = \frac{|w_k | w_k \epsilon S_i \ \& \ w_k \epsilon S_j|}{\log(|S_i|) + \log(|S_j|)} \quad (1)$$

$Similarity(S_i, S_j)$ merupakan skor kesamaan antara dua kata S_i dan S_j, S_i adalah kata pertama, S_j adalah kata kedua, w_k merupakan elemen yang ada didalam kata S_i dan S_j, $|w_k|$ adalah jumlah elemen w_k yang ada di dua kata, log merupakan fungsi logaritma, $|S_i|$ adalah jumlah elemen dari kata S_i, $|S_j|$ adalah jumlah elemen dari kata S_j. Rumus ini bekerja dengan cara pertama – tama menghitung jumlah elemen w_k yang terdapat di kedua kata S_i dan S_j. Setelah itu, jumlah elemen tersebut dibagi dengan jumlah logaritmadari ukuran masing – masing set S_i dan S_j. Skor yang dihasilkan menunjukkan tingkat kesamaan antara kedua kata tersebut dengan nilai yang lebih tinggi menunjukkan tingkat kesamaan yang lebih besar.

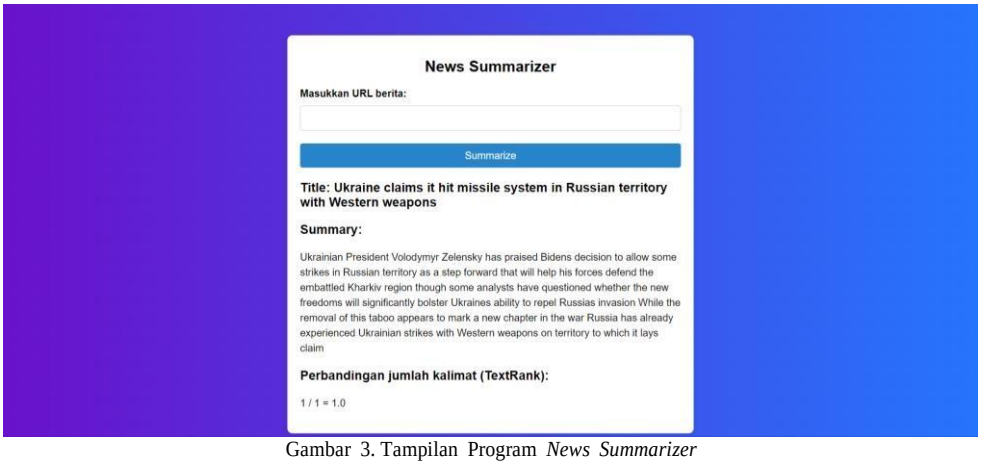
Pada pemrograman ini dipakai juga *library* Summa. Summa merupakan perpustakaan *Python* yang menggunakan algoritma *TextRank* untuk merangkum teks secara otomatis. Dengan Summa, Anda dapat menyediakan teks dan

menerima ringkasan yang berisi kalimat-kalimat kunci. Summa memperlakukan kalimat-kalimat sebagai *node* dalam graf, menilai pentingnya berdasarkan kesamaannya dengan kalimat- kalimat lain. Ini memungkinkan pembuatan ringkasan yang singkat namun informatif, menjadikannya berguna untuk berbagai tugas pemrosesan bahasa alami.

IV. HASIL DAN PEMBAHASAN

Algoritma *coding* akan menghitung peringkat untuk setiap *node* berdasarkan struktur graf dan bobot tepi untuk menentukan tingkat kepentingannya. Hasilnya adalah sebuah ringkasan yang mencakup informasi paling penting dalam teks. Dari total kata pada tautan URL adalah 387 dan program ini akan menghasilkan 75 kata dari 387 kata tersebut dan juga telah diuji dengan beberapa artikel, program ini akan merangkum sekitar 75% dari konten artikel secara keseluruhan.

Cara kerja program automatic news summarization yang dibuat dalam penelitian ini, pengguna menyalin dan menambahkan URL ke dalam program yang sudah bertampilkan *website*. Kemudian program akan berjalan dan menghasilkan ringkasan dari berita atau artikel yang diinginkan, namun artikel ataupun berita yang dapat dibaca oleh program adalah artikel berbahasa inggris saja.



Gambar 3. Tampilan Program *News Summarizer*

Gambar 3 merupakan hasil atau tampilan dari program *News Summarizer* kami, program akan menampilkan judul artikel atau berita beserta hasil ringkasan. Melalui beberapa *data testing* yang sudah dijalankan, menghasilkan beberapa tes dan hasil. Pada tes yang pertama dijalankan program dengan 4 artikel yang berbeda bertujuan untuk melihat persentase hasil ringkasan dari beberapa berita ataupun artikel.

TABEL I
TOTAL RATA – RATA RINGKASAN PROGRAM

Amount of Words in News Website	Result Summary	Summary Presentation (%)
387	75	80%
1046	340	70%
942	268	71%
1544	463	70%
2464	741	70%
Total Average (%)		**72%**

Dari tes yang pertama, dapat dilihat pada tabel 1 bahwa rata-rata persentase ringkasan dari empatartikel yang sudah dijalankan oleh program menunjukkan bahwa rata-rata program dapat meringkas artikelsebesar 70% dan menghasilkan ringkasan sebesar 30%. Hal ini menunjukkan bahwa program ini cukup efisien dalam mengurangi jumlah teks tanpa mengurangi informasi penting yang ada dalam artikel.

Adapun tes kedua yang berada pada tabel 2, telah diuji pada program adalah untuk mengecek tingkat akurasi kesamaan (similarity) dengan menyebarkan kuesioner berupa pertanyaan singkat. Responden diberikan pertanyaan dan menjawab "ya" atau "tidak", kemudian dilakukan perhitungan dari hasil yang didapatkan.

Rumus yang digunakan dalam perhitungan, digunakan *MMR Accuracy*, dengan rumus dibawah ini.

$$MMR\ Accuracy\ (\text{setiap dokumen}) = \left(\frac{Jawaban\ "ya"}{Total\ pertanyaan}\right)\ X\ 100\% \quad (2)$$

TABEL II
PENGUJIAN AKURASI MELALUI METODE *QNA*

Article	Yes Answer	Total Question	Accuracy (%)
1	5	7	71.42%
2	4	7	57.14%
3	5	7	71.42%
4	5	7	71.42%
5	6	7	85.71%
Total Accuracy (%)			**71.42%**

Seperti yang dapat dilihat pada tabel 2 hasil tes yang telah dilakukan, total keseluruhan akurasi yang didapatkan adalah 71.42% dari lima dokumen yang telah diuji. Selain itu, pengujian juga telah dilakukan dari berbagai sumber berita atau artikel luar negeri berbahasa Inggris. Hasil pengujian ini menunjukkan dua keberhasilan dan satu *error* terhadap sebuah sumber laman berita.

TABEL III
PENGUJIAN SUMBER HALAMAN BERITA

Source	Rendering
CNN	Success
CNA	Success
BBC	Unsuccess

Dapat dilihat pada tabel pengujian, bila berita atau artikel diambil dari sumber *website* seperti CNN dan CNA akan berhasil dibaca dan berjalan serta menghasilkan rangkuman yang diinginkan, namunapabila artikel diambil dari BBC tidak terbaca atau timbulnya indikasi *error*.

Kegagalan ini disebabkan oleh dua faktor utama yaitu struktur HTML BBC yang lebih kompleks dan berbeda dari situs lain menyebabkan *newspaper3k* tidak dapat mengekstrak semua konten penting dan BBC menggunakan JavaScript untuk memuat konten secara dinamis, yang tidak dapat ditangani oleh *newspaper3k* dan *Beautifulsoup* yang hanya bekerja dengan HTML statis. Untuk mengatasi masalah ini, ke depan program dapat ditingkatkan dengan mendukung rendering dinamis menggunakan teknologi seperti *Selenium* untuk menangani konten yang dimuat setelah halaman awa

V. KESIMPULAN

Kesimpulan dari penelitian yang sudah dilakukan, analisis peringkasan berita otomatis dengan menggunakan metode *TextRank* dengan bantuan pustaka Summa, dan BeautifulSoup dalam rangkuman berita mengungkap wawasan berharga tentang kinerja mereka melintasi berbagai metrik. *TextRank* menunjukkan akurasi dan efisiensi yang tangguh dalam menghasilkan rangkuman, dengan memanfaatkan algoritma peringkat berbasis grafiknya. Summa, dengan perpustakaan *Python*-nya untuk tugas rangkuman, menunjukkan kinerja yang memuaskan, sementara BeautifulSoup berkontribusi pada pengumpulan data melalui web *scraping*. Integrasi teknik-teknik ini menawarkan solusi komprehensif untuk mengekstrak informasi penting dari artikel berita *online*. Namun, penting untuk mengakui batasan dari masing-masing teknik, seperti ketergantungan *TextRank* pada kualitas input, potensi tantangan Summa dengan jenis teks tertentu, dan ketergantungan BeautifulSoup pada sumber daya *online* yang dapat diakses. Program rangkuman berita akan merangkum 70-80% dari sumber dan menghasilkan 20-30% rangkuman, juga telahdiuji untuk akurasi, di mana akurasi rata-rata program rangkuman berita adalah 71,422%. Dibalik keberhasilan program yang sudah dibuat adapun kelemahan dari program yang dibuat dalam penelitian seperti, program hanya bisa membaca dan meringkas artikel dengan menggunakan bahasa inggris, selain itu program juga hanya dapat memproses dokumen tunggal. Dengan kelemahan dari program dan penelitian tersebut diharapkan pada penelitian mendatang peneliti dapat mengatasi keterbatasan dengan mengintegrasikan dukungan multibahasa, dan juga dapat memproses multidokumen dengan baik dan akurat.

REFERENSI

[1] E. V. C. M dan J. Pragantha, "PENERAPAN ALGORITMA *TEXTRANK* UNTUK AUTOMATICSUMMARIZATION PADA DOKUMEN BERBAHASA INDONESIA," *Jurnal Ilmu Teknik dan Komputer*, vol. 1, p. 8, 2017.

[2] K. Yoko, V. C. M dan J. Hendryli, "SISTEM PERINGKAS OTOMATIS ABSTRAKTIF DENGAN MENGGUNAKAN RECURRENT NEURAL NETWORK," *Journal of ComputerScience and Information Systems*, p. 11, 2018.

[3] A. Ashari dan M. Riasetiawan, "Document Summarization using *TextRank* and SemanticNetwork," *Modern Education and Computer Science*, p. 8, 2017.

[4] R. Yanuarti dan H. A. Alfaruq, "Implementasi Text Summarization Pada Reading Comprehension," *JASIE "Jurnal Aplikasi Sistem Informasi Dan Elektronika"* , vol. 2, p. 9, 2022.

[5] P. Joshi, "An Introduction to Text Summarization using the *TextRank* Algorithm (with Pythonimplementation)," *Analytic Vidhya*, 2023.

[6] M. A. zamzam, C. Crysdian dan K. F. Hayati Holle, "Sistem Automatic Text Summarization," *MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi* , vol. 12, p. 6, 2020.

[7] L. N. Fadhila dan K. D. Nuryana, "Teks Ringkas Otomatis pada Portal Berita CNN IndonesiaMenggunakan Algoritma *TextRank*," *Journal of Emerging Information Systems and Business Intelligence*, vol. 5, p. 7, 2024.

[8] I. A. Fathoni, "IMPLEMENTASI PERINGKASAN TEKS OTOMATIS DENGAN ALGORITMA *TEXTRANK* UNTUK BERITA *ONLINE*," p. 51, 2023.

[9] S. Anand, "Understanding Page Rank," 2018.

[10] U. Rani dan K. Bidhan, "Comparative Assessment of Extractive Summarization: *TextRank*, TF-IDF and LDA," *Journal of Scientific Research*, vol. 65, no. 1, p. 8, 2021.