

THE INFLUENCE OF FEATURE EXTRACTION ON AUTOMATIC TEXT SUMMARIZATION USING GENETIC ALGORITHM

Fitrah Amalia Rahmadiani*¹, Nirwana Hendrastuty²

^{1,2,3}Information Systems, Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Indonesia

Email: ¹fitrah_amalia_rahmadiani@teknokrat.ac.id, ²nirwanahendrastuty@teknokrat.ac.id

(Article received: May 08, 2024; Revision: May 23, 2024; published: July 29, 2024)

Abstract

Text summarization using extraction methods is a technique that summarizes by retaining a subset of sentences to create a summary. There are two types of documents commonly used for summarization: single document and multi-document. Multi-document refers to documents originating from one or more sources that contain several main ideas. The data used in this research is obtained from the E-lapor DIY website, consisting of 1000 data entries. E-Lapor DIY is a website provided by the DIY government to accommodate all public aspirations and complaints, such as damaged roads, broken traffic lights, insufficient street lighting, litter in public places, and more. The accumulation of data and the delayed response time has become an issue for the government in addressing these complaints. This research aims to consider the impact of using feature extraction for text summarization using genetic algorithms. The feature extraction compared in this research is the influence of sentence position in feature extraction. The results obtained show that Precision testing using F1 is 0.64, and without using F1, it is 0.66. Recall testing using F1 is 0.65, and without using F1, it is 0.68. F-Measure testing using F1 is 0.65, and without using F1, it is 0.68. This testing using the algorithm can be an interesting alternative for more time-efficient text summarization.

Keywords: E-Lapor DIY, Genetic algorithm, Text summarization.

PENGARUH PENGGUNAAN EKSTRAKSI FITUR PADA PERINGKASAN TEKS OTOMATIS MENGGUNAKAN ALGORITMA GENETIKA

Abstrak

Peringkasan teks menggunakan metode ekstraksi merupakan teknik meringkas dengan mempertahankan subset dari kalimat untuk dijadikan ringkasan. Ada dua jenis dokumen yang sering digunakan untuk melakukan peringkasan yaitu single dokumen dan multi dokumen, Multi dokumen merupakan dokumen yang berasal dari satu sumber atau lebih yang memiliki beberapa ide pokok bahasan. Data yang digunakan pada penelitian ini adalah data yang dapat di dapat dari website E-lapor DIY sebanyak 1000 data. E-Lapor DIY merupakan website yang disediakan oleh pemerintah DIY untuk menampung seluruh aspirasi dan keluhan masyarakat seperti jalanan rusak, lampu apil yang mati, penerangan jalan yang kurang, sampah yang berserakan di tempat umum dan lain-lain. Penumpukan data lama untuk di respon menjadi masalah pemerintah dalam menangani pengaduan tersebut. Penelitian ini bertujuan mempertimbangkan pengaruh penggunaan ekstraksi fitur untuk melakukan peringkasan teks menggunakan algoritma genetika, Ekstraksi fitur yang dibandingkan pada penelitian ini adalah seberapa berpengaruhnya posisi antar kalimat dalam ekstraksi fitur. Hasil yang diperoleh, pengujian Precision menggunakan F1 sebesar 0,64 dan tidak menggunakan F1 sebesar 0,66 sedangkan Hasil pengujian Recall menggunakan F1 sebesar 0,65 dan tidak menggunakan F1 sebesar 0,68 serta Hasil pengujian F-Measure menggunakan F1 sebesar 0,65 dan tidak menggunakan F1 sebesar 0,68. Dalam pengujian menggunakan algoritma ini dapat menjadi alternatif menarik untuk peringkasan teks yang lebih efisien waktu.

Kata kunci: Algoritma genetika, E-Lapor DIY, Text summarization.

1. PENDAHULUAN

Saat ini pemerintah tidak hanya melakukan pelayanan dan menerima pengaduan terkait keluhan dari masyarakat secara tatap muka tetapi sudah

banyak *website* yang menerima pengaduan tersebut. Permasalahannya adalah, ketika banyak masyarakat melaporkan keluhannya ke *website* maka keluhan tersebut akan menumpuk dan lama untuk di respon

[1], [2]. Sehingga membutuhkan peringkasan secara otomatis untuk mempermudah hal tersebut. Peringkasan teks otomatis (*automated text summarization*) merupakan tahapan untuk mendapatkan informasi dari sebuah dokumen [3] atau teks menggunakan komputer [4]. Terdapat dua pendekatan untuk melakukan peringkasan teks otomatis, pendekatan ekstraksi dan pendekatan abstraksi [5], [6]. Pendekatan abstraksi melakukan peringkasan dengan mempertahankan klausa yang dianggap informatif pada suatu tulisan [7]. frasa yang disalin dapat berupa klausa utama, klausa utama, atau paragraf utama [8]. Metode eliminasi digunakan untuk melestarikan kata asli dalam laporan yang sebenarnya untuk meminimalkan perbedaan makna turunan. Salah satu metode yang banyak dikembangkan untuk melakukan peringkasan teks secara otomatis adalah algoritma genetika [9]. Algoritma Genetika merupakan salah satu algoritma komputasi yang mengadopsi teori evolusi, dimana teori tersebut yang akan digunakan untuk mencari solusi dari suatu permasalahan optimasi [10], [11]. Fatah dan Ren (2008) telah melakukan peringkasan teks secara otomatis pada dokumen tunggal berbahasa Indonesia menggunakan metode algoritma genetika [12]. Melakukan peringkasan teks secara otomatis pada multi dokumen menggunakan algoritma genetika [13]. Yang ketiga penelitian tersebut adalah ekstraksi fitur yang di gunakan [14]. Melakukan peringkasan teks menggunakan empat ekstraksi fitur sedangkan melakukan peringkasan teks menggunakan sepuluh ekstraksi fitur yaitu posisi antar kalimat, kata kunci positif, kata kunci negative, kemiripan kalimat, koneksi antar kalimat, panjang kalimat, kalimat mengandung angka, kalimat entitas, jumlah koneksi antar kalimat dan bobot kalimat [12]. Data yang digunakan pada penelitian ini adalah data yang di dapat dari website E-lapor DIY sebanyak 1000 data. E-Lapor DIY merupakan website yang di sediakan oleh pemerintah DIY untuk menampung seluruh aspirasi dan keluhan masyarakat seperti jalanan rusak, lampu apil yang mati, penerangan jalan yang kurang, sampah yang berserakan di tempat umum dan lain-lain [15],[16]. Penelitian ini bertujuan untuk mengetahui seberapa pengaruh posisi antar kalimat sebagai ekstraksi fitur pada peringkasan teks multi dokumen menggunakan metode abstraksi dan algoritma genetika.

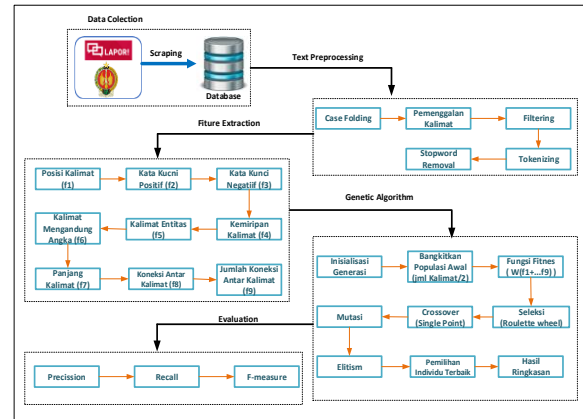
2. METODE PENELITIAN

2.1. Arsitektur Sistem

Arsitektur sistem yang di lakukan pada penelitian ini dapat di lihat pada gambar 1.

Penelitian ini memiliki enam tahap penelitian yaitu, *data collection*, *text preprocessing*, *fiture extraction*, *TF-ISF*, *genetic algorithm*, *evalustion*. Data yang di gunakan pada penelitian ini sebanyak 1000 data yang di dapatkan dari proses scraping pada website *e-Lapor DIY*. Dataset yang digunakan

merupakan aduan masyarakat berbahasa Indonesia dari bulan Februari 2018 sampai Juli 2020. Tahap preprocessing yang digunakan yaitu case folding, pemenggalan kalimat, filtering, tokenizing dan stopwords removal.



Gambar 1. Arsitektur Sistem

2.2. Ekstraksi Fitur

Salah satu proses yang sangat penting untuk mengubah format teks yang tidak terstruktur menjadi terstruktur adalah ekstraksi fitur. Penelitian ini menggunakan ekstraksi fitur yang sama dengan penelitian yang sudah di lakukan oleh [12] yaitu Posisi Antar Kalimat (F1), Kalimat Positif (F2), Kalimat Negatif (F3), Kemiripan Antar Kalimat (F4), Kalimat Mengandung Entitas (F5), Kalimat Mengandung Numerik (F6), Panjang Kalimat (F7), Koneksi Antar Kalimat (F8), Jumlah Koneksi Antar Kalimat (F9) [17]. Penggunaan ekstraksi fitur yang akan di bandingkan adalah pada penggunaan posisi antar kalimat (F1), seberapa berpengaruhnya posisi antar kalimat pada ekstraksi fitur untuk melakukan peringkasan teks [18]. Berikut merupakan persamaan ekstraksi fitur yang di gunakan pada penelitian ini:

a. Posisi Kalimat

Posisi kalimat dapat dihitung menggunakan persamaan (1) Dengan asumsi bahwa kalimat pertama dan terakhir dari setiap paragraf adalah yang paling penting.

$$Score_{f_1}(s) = \frac{x}{N} \quad (1)$$

b. Kata Kunci Positif (f2)

Menghitung kata kunci positif dapat di lakukan menggunakan persamaan (2). Kata yang paling sering digunakan dalam sebuah kalimat adalah kata kunci positif.

$$Score_{f_2}(s) = \frac{\text{Banyaknya kata kunci ke i dalam kalimat}}{\text{Banyaknya kata kunci dalam kalimat}} \quad (2)$$

c. Kata Kunci Negatif (f3)

Kata kunci negatif dapat di hitung menggunakan persamaan (3). Kata kunci negatif merupakan kata kunci yang paling sedikit muncul pada kalimat.

$$Score_{f_3}(s) = \frac{\text{kata kunci paling sedikit dalam kalimat}}{\text{Banyaknya kata kunci dalam kalimat}} \quad (3)$$

d. Kemiripan Antar Kalimat (f4)

Persamaan dapat digunakan untuk mengetahui seberapa mirip kalimat satu sama lain (4). Kemiripan antar kalimat adalah mencari kata yang sama yang ada pada kromosom pertama dengan kata yang ada pada kromosom lain.

$$Score_{f_4}(s) = \frac{\text{kata kunci dalam kalimat} \cap \text{kata kunci antar kalimat}}{\text{kata kunci dalam kalimat} \cup \text{kata kunci antar kalimat}} \quad (4)$$

e. Kalimat yang Mengandung Kata Entitas (f5)

Persamaan dapat digunakan untuk menghitung kalimat yang mengandung kata entitas (5). Kalimat yang mengandung entitas merupakan kalimat yang mengandung makna seperti nama pulau, nama orang, institusi, tempat, dan lain-lain.

$$Score_{f_5}(s) = \frac{\text{nama entitas dalam kalimat}}{\text{panjang kalimat}} \quad (5)$$

f. Kalimat yang Mengandung Angka (f6)

Persamaan menunjukkan kalimat yang mengandung angka (6). Kalimat yang mengandung angka biasanya dianggap penting.

$$Score_{f_6}(s) = \frac{\text{data yang mengandung angka dalam kalimat}}{\text{panjang kalimat}} \quad (6)$$

g. Panjang Kalimat (f7)

Panjang kalimat dapat dihitung menggunakan persamaan (7). Panjang kalimat bertujuan untuk mengetahui seberapa Panjang kalimat tersebut.

$$Score_{f_7}(s) = \frac{\text{jumlah kata dalam kalimat}}{\text{kata unik dalam dokumen}} \quad (7)$$

h. Koneksi Antar Kalimat (f8)

Koneksi antar kalimat dapat dihitung menggunakan persamaan (8). Banyak kalimat yang memiliki kata yang sama disebut "koneksi antar kalimat".

$$Score_{f_8}(s) = \text{koneksi antar kalimat} \quad (8)$$

i. Jumlah Koneksi Antar Kalimat (f9)

Jumlah koneksi antar kalimat dapat dihitung menggunakan persamaan (9). Jumlah kalimat yang memiliki kata yang sama disebut "koneksi antar kalimat".

$$Score_{f_9}(s) = \sum_{k=0}^n \text{koneksi antar kalimat} \quad (9)$$

Perbandingan ekstraksi fitur yang di lakukan pada penelitian ini sebagai berikut.

Tabel. 1 Perbandingan Penggunaan Ekstraksi Fitur		
Perbandingan Penggunaan Ekstraksi Fitur		
Perbandingan	Menggunakan F1	Tidak Menggunakan F1
Ke 1	F1, F2, F3, F4, F5, F6, F7, F8, F9	F2, F3, F4, F5, F6, F7, F8, F9

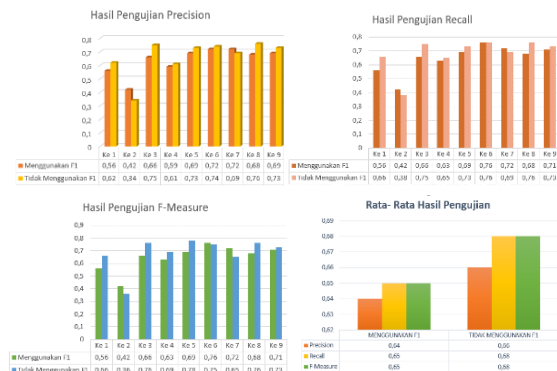
Ke 2	F1, F3, F4, F5, F6, F7, F8, F9	F3, F4, F5, F6, F7, F8, F9
Ke 3	F1, F2, F4, F5, F6, F7, F8, F9	F2, F4, F5, F6, F7, F8, F9
Ke 4	F1, F2, F3, F5, F6, F7, F8, F9	F2, F3, F5, F6, F7, F8, F9
Ke 5	F1, F2, F3, F4, F6, F7, F8, F9	F2, F3, F4, F6, F7, F8, F9
Ke 6	F1, F2, F3, F4, F5, F7, F8, F9	F2, F3, F4, F5, F7, F8, F9
Ke 7	F1, F2, F3, F4, F5, F6, F8, F9	F2, F3, F4, F5, F6, F8, F9
Ke 8	F1, F2, F3, F4, F5, F6, F7, F9	F2, F3, F4, F5, F6, F7, F9
Ke 9	F1, F2, F3, F4, F5, F6, F7, F8	F2, F3, F4, F5, F6, F7, F8

Fungsi fitness yang dihitung dengan menggunakan ekstraksi fitur teks digunakan untuk menentukan seberapa baik seseorang [19]. Setelah melakukan proses ekstraksi fitur tahap selanjutnya yaitu proses peringkasan teks menggunakan algoritma genetika. Algoritma genetika berhasil menyelesaikan masalah kompleks yang sulit diselesaikan oleh metode konvensional. Ini karena kemampuan algoritma untuk menjelajahi ruang pencarian kompleks dan menemukan solusi optimal untuk kompleksitas parameter [20]. Tahap algoritma genetika menggunakan 20 iterasi dengan 20 popsize yang di tetapkan di awal proses penentuan individu [21]. Nilai ekstraksi fitur teks ditunjukkan dalam isi kromosom, yang merupakan indeks kalimat yang menggabungkan seluruh fitur ekstraksi ($W_1, W_2, W_3, \dots, W_n$). Proses peringkasan menggunakan algoritma genetika akan di lakukan per satu minggu dengan jumlah kromosom di setiap individu adalah $n/2$ di mana n adalah jumlah kalimat yang ada pada data aduan masyarakat selama satu minggu.

Fungsi Fitness yang digunakan yaitu $Fitness = \sum_{i=1}^n W_i$ di mana W merupakan bobot kalimat [22]. Metode seleksi yang digunakan adalah *roulette wheel*. Kemungkinan bahwa seseorang akan dipilih sebagai orang terbaik lebih besar jika mereka memiliki nilai probabilitas yang lebih tinggi [22] [23]. Dengan ketentuan total dari probabilitas keseluruhan individu adalah 100%. Metode *crossover* yang di gunakan yaitu *single poin crossover* [24]. Kromosom yang akan di *crossover* diseleksi secara *random* untuk menghasilkan anak (*offspring*) [25]. Proses mutasi gen yang terpilih bobot kalimatnya akan di tambah dengan nilai *random* antara -1 hingga 1. *Elitism* dilakukan menggunakan metode *continous update* yaitu memungkinkan berkumpulnya *genome offspring* dan *parents* dalam satu generasi. Individu dengan nilai fitness tertinggi di populasi yang sama dianggap individu terbaik dan akan di jadikan hasil ringkasan [26]. Dari hasil ringkasan yang di anggap individu terbaik akan di bandingkan antara hasil ringkasan yang menggunakan ekstraksi fitur F1 dengan hasil ringkasan yang tidak menggunakan ekstraksi fitur F1.

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan menggunakan metode evaluasi intrinsik dengan mengukur kinerja sistem, meliputi pengujian akurasi, recall, dan f-measure berdasarkan hasil rangkuman dari keluhan pelanggan. Objek berbahasa Indonesia diimplementasikan dengan algoritma genetika. Tidak semua kombinasi parameter dapat dilakukan karena keterbatasan waktu dan sumber daya, sehingga parameter yang diuji dalam penelitian ini menggunakan 20 ukuran popsize dan 20 Iterasi.



Gambar 2. Grafik Hasil Pengujian

Bagan tersebut membandingkan dua konfigurasi, berjudul "Mengggunakan F1" (Mengggunakan F1) dan "Tidak Menggunakan F1" (Tidak Menggunakan F1). Ini mungkin mengacu pada pengaturan dalam *Machine Learning*. Bagan pengukuran presisi, Perolehan Kembali, dan Pengukuran F1 untuk setiap konfigurasi di sembilan eksperimen (diberi label "Ke 1" hingga "Ke 9"). Ini semua adalah metrik performa umum yang digunakan untuk mengevaluasi *Machine Learning*. Sisi kiri grafik kemungkinan menunjukkan trade-off antara presisi dan perolehan. Presisi menunjukkan berapa banyak kasus positif aktual yang diidentifikasi oleh model, sedangkan recall menunjukkan berapa banyak kasus positif sebenarnya yang diidentifikasi oleh model. Idealnya, suatu model memiliki presisi dan recall yang tinggi. Dengan grafik garis di sebelah kanan kemungkinan menunjukkan bagaimana Presisi, Perolehan, dan Pengukuran F1 berubah selama sembilan eksperimen untuk setiap konfigurasi.

Berdasarkan Hasil yang diperoleh, hasil pengujian Precision menggunakan F1 sebesar 0,64 dan tidak menggunakan F1 sebesar 0,66 sedangkan Hasil pengujian Recall menggunakan F1 sebesar 0,65 dan tidak menggunakan F1 sebesar 0,68 serta Hasil pengujian F-Measure menggunakan F1 sebesar 0,65 dan tidak menggunakan F1 sebesar 0,68..

4. DISKUSI

Sebelumnya terdapat beberapa penelitian yang melakukan peringkasan teks otomatis. Salah satunya adalah penelitian pembobotan fitur ekstraksi pada peringkasan teks Bahasa Indonesia menggunakan

algoritma genetika. Berdasarkan penelitian tersebut hasil pembobotan 2 bobot fitur adalah 39%. Terjadi peningkatan 2% ketika menggunakan 4 bobot fitur, peningkatan 9% ketika menggunakan 6 bobot fitur, peningkatan 2% ketika menggunakan 7 bobot fitur, peningkatan 1% ketika menggunakan 8 bobot fitur, mengalami penurunan 4% ketika menggunakan 9 bobot fitur, dan mengalami penurunan 3% ketika menggunakan 10 bobot fitur [12].

Kemudian Penelitian lainnya melakukan peringkasan teks artikel ilmiah berbahasa Indonesia dengan metode pembobotan kalimat menghasilkan tiga indikator yaitu desain sebesar 82,6% , segi efisiensi sebesar 80,2% dan segi kepuasan sebesar 83,7% [27].

Beberapa penelitian sebelumnya menunjukkan bahwa peringkasan teks memiliki metode yang berbeda – beda. Seperti penelitian ini menggunakan metode algoritma genetika mendapatkan hasil perbandingan yang menggunakan F1 dan tidak menggunakan F1. Selain itu penelitian ini memiliki perbedaan dari beberapa penelitian sebelumnya seperti data yang digunakan adalah data *E-Lapor* DIY Dataset yang digunakan merupakan aduan masyarakat berbahasa Indonesia dari bulan Februari 2018 sampai Juli 2020. Kemudian melakukan tahapan preprocessing yang digunakan yaitu case folding, pemenggalan kalimat, filtering, tokenizing dan stopword removal.

5. KESIMPULAN

Penelitian ini mengusulkan solusi untuk menangani penumpukan keluhan masyarakat di platform online dengan mengembangkan peringkasan teks otomatis menggunakan algoritma genetika. Dua pendekatan utama, yaitu ekstraksi dan abstraksi, digunakan untuk mencapai tujuan ini. Melalui ekstraksi fitur, penelitian ini mengeksplorasi pengaruh posisi antar kalimat sebagai salah satu faktor dalam peringkasan teks. Penelitian menggunakan data dari website E-lapor DIY dan melibatkan enam tahap penelitian, termasuk pengumpulan data, pra-pemrosesan teks, ekstraksi fitur, dan evaluasi. Pengujian dilakukan dengan mengukur akurasi, recall, dan f-measure dari hasil peringkasan teks dalam dua konfigurasi: dengan menggunakan ekstraksi fitur posisi antar kalimat (F1) dan tanpa F1. Hasil pengujian menunjukkan bahwa penggunaan F1 memberikan hasil yang sedikit lebih rendah dibandingkan tanpa F1. Meskipun demikian, penelitian ini memberikan kontribusi penting dalam memahami penggunaan ekstraksi fitur untuk peringkasan teks otomatis dan menyoroti pentingnya penyesuaian parameter untuk memperoleh hasil yang optimal.

Adapun saran untuk penelitian selanjutnya dapat mencari fitur ekstraksi baru kemudian menambahkannya dengan fitur ekstraksi yang ada pada penelitian ini dan pada penelitian selanjutnya dapat menggunakan algoritma lainnya untuk

pembobotan fitur ekstraksi teks pada peringkasan Bahasa Indonesia.

DAFTAR PUSTAKA

- [1] A. Annaafi, Engkus, and M. I. Nur, "Mekanisme Pengaduan Dan Responsivitas Kinerja Pegawai Pada Badan Pertanahan Nasional Kabupaten Bandung," *J. Ilm. Hosp.*, vol. 11, no. 2, pp. 443–455, 2022.
- [2] M. F. M. Hutasuhut and R. Rosnelly, "Perancangan Dan Implementasi Sistem Pengaduan Dan Pelayanan Masyarakat Berbasis Mobile," *J. Info Digit*, vol. 1, no. 2, pp. 756–769, 2023.
- [3] I. Muslim, K. Karo, A. Perdana, S. Dewi, and I. Komputer, "Implementasi Text Summarization Pada Review Aplikasi Digital Library System Menggunakan Metode Maximum Marginal Relevance," vol. 4, no. 1, 2024.
- [4] Halimah, Surya Agustian, and Siti Ramadhani, "Peringkasan teks otomatis (automated text summarization) pada artikel berbahasa indonesia menggunakan algoritma lexrank," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 3, no. 3, pp. 371–381, 2022, doi: 10.37859/coscitech.v3i3.4300.
- [5] A. A. Fattahila, A. Romadhony, and S. Al Faraby, "Peringkasan Artikel Berita Menggunakan Pendekatan Abstraktif Dengan Model Transformers," vol. 10, no. 5, pp. 4980–4986, 2023.
- [6] N. F. Saraswati, Indriati, and R. S. Perdana, "Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 5494–5502, 2018, doi: 10.1016/s1010-6030(01)00380-x.
- [7] G. A. Pradnyana and I. K. A. Mogi, "Implementasi Automated Text Summarization Untuk Dokumen Tunggal Berbahasa Indonesia Dengan Menggunakan Graph-Based," *J. Ilm. NERO*, vol. 1, no. 2, pp. 33–46, 2014.
- [8] W. Pratama, R. Ilyas, and F. Kasyidi, "Peringkasan Otomatis Makalah Menggunakan Maximum Marginal Relevance," *Informatics Digit. Expert*, vol. 3, no. 1, pp. 32–37, 2022, doi: 10.36423/index.v3i1.677.
- [9] Y. Yuliska and K. U. Syaliman, "Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia," *IT J. Res. Dev.*, vol. 5, no. 1, pp. 19–31, 2020, doi: 10.25299/itjrd.2020.vol5(1).4688.
- [10] M. K. Fajarlestari and I. B. Suban, "Kombinasi Crossover dan Mutasi Terbaik pada Algoritma Genetika dalam Penjadwalan Mata Kuliah," *Techno.Com*, vol. 22, no. 4, pp. 843–853, 2023, doi: 10.33633/tc.v22i4.9298.
- [11] S. B. Saputra and E. W. Pamungkas, "Development of Scheduling System With Genetic Algorithm in Website-Based Smk Negeri 1 Sine," *J. Tek. Inform.*, vol. 4, no. 4, pp. 797–806, 2023, doi: 10.52436/1.jutif.2023.4.4.784.
- [12] Z. Zulkifli, A. T. Wibowo, and G. Septiana, "Pembobotan Fitur Ekstraksi Pada Peringkasan Teks Bahasa Indonesia Menggunakan Algoritma Genetika | Zulkifli | eProceedings of Engineering," vol. 2, no. 2, pp. 6481–6489, 2015.
- [13] R. R. Putra *et al.*, "Peringkasan Teks Otomatis Pada Multi Dokumen Menggunakan Textrank," *J. Ilm. Komput. dan Inform.*, 2018.
- [14] N. Hendrastuty and A. SN, "Text Summarization in Multi Document Using Genetic Algorithm," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 15, no. 4, p. 327, 2021, doi: 10.22146/ijccs.66026.
- [15] D. Lestari, W. W. Winarno, and M. P. Kurniawan, "Model E-Readiness Untuk Pengukuran Kesiapan Pengelolaan Aduan E-Lapor DIY," *Creat. Inf. Technol. J.*, vol. 7, no. 2, p. 86, 2021, doi: 10.24076/citec.2020v7i2.249.
- [16] F. D. Agustyar, A. Aditya, S. Aminah, and A. Tirtana, "Design and Development an e-Lapor Application to Support Public Complaint Services in Tunjungtirta Village," *J. Informatics Telecommun. Eng.*, vol. 7, no. 1, pp. 164–173, 2023, doi: 10.31289/jite.v7i1.9762.
- [17] A. Tendi And K. E. Dewi, "Ilmiah Komputer Dan Relevance Vector Machine Dalam Peringkasan Multidokumen Ilmiah Komputer Dan," Vol. 6, No. 1, 2017.
- [18] A. Kurniawan, "Aplikasi Sistem Ekstraksi Kata Kunci Berbahasa Indonesia Menggunakan Algoritma Textrank Studi Kasus Data Wikipedia Indonesia," *Repository.Uinjkt.Ac.Id*, 2021.
- [19] W. Hadikristanto and I. Nasai, "Penerapan Algoritma Genetika Dalam Memprediksi Penerima Program Keluarga Harapan Dengan Metode Naïve Bayes," *SIGMA - J. Teknol. Pelita Bangsa 167*, vol. 10, no. 1, pp. 167–172, 2019.
- [20] L. A. Pangestu, S. H. Suryawan, and A. J. Latipah, "Penerapan Algoritma Genetika

- Dalam Penjadwalan Mata Pelajaran,” *J. Inform.*, vol. 10, no. 2, pp. 194–205, 2023, doi: 10.31294/inf.v10i2.16701.
- [21] L. Syakina, T. Bakhtiar, F. Hanum, and P. T. Supriyo, “Penentuan Rute Distribusi Rastra Menggunakan Algoritma Genetika,” *MILANG J. Math. Its Appl.*, vol. 19, no. 2, pp. 97–115, 2023, doi: 10.29244/milang.19.2.97-115.
- [22] L. D. Yulianti, S. Basuki, and Y. Azhar, “Implementasi Algoritma Graf dan Algoritma Genetika pada Peringkasan Single Document,” *J. Repos.*, vol. 2, no. 11, p. 1521, 2020, doi: 10.22219/repositor.v2i11.891.
- [23] D. A. Suprayogi and W. F. Mahmudy, “Penerapan Algoritma Genetika Traveling Salesman Problem with Time Window: Studi Kasus Rute Antar Jemput Laundry,” *J. Buana Inform.*, vol. 6, no. 2, pp. 121–130, 2015, doi: 10.24002/jbi.v6i2.407.
- [24] M. Kurniawan, “Optimasi Struktur Rangka Batang Menggunakan Metode Algoritma Genetika Dengan Kendala Tegangan Dan Probabilitas Kegagalan,” *J. Saintis*, vol. 19, no. 1, pp. 15–23, 2019, doi: 10.25299/saintis.2019.vol19(1).3043.
- [25] W. F. Mahmudy and M. A. Rahman, “Optimasi Fungsi Multi-Obyektif Berkendala Menggunakan Algoritma Genetika Adaptif Dengan Pengkodean Real,” *J. Ilm. KURSOR*, vol. 6, no. 1, pp. 19–26, 2011.
- [26] S. Ulya, M. A. Soeleman, and F. Budiman, “Optimasi Parameter K Pada Algoritma K-NN Untuk Klasifikasi Prioritas Bantuan Pembangunan Desa,” *Techno.Com*, vol. 20, no. 1, pp. 83–96, 2021, doi: 10.33633/tc.v20i1.4215.
- [27] D. Fatmalasari and F. R. Lumbanraja, “Peringkasan Teks Artikel Ilmiah Berbahasa Indonesia dengan Metode Pembobotan Kalimat,” *J. Pepadun*, vol. 3, no. 3, pp. 314–322, 2022, doi: 10.23960/pepadun.v3i3.127.