INOTEK, Vol. 9
ISSN: 2580-3336 (Print) / 2549-7952 (Online)
Url: https://proceeding.unpkediri.ac.id/index.php/inotek/

Penerapan *Regular Expression* dan *Cosine Similarity* pada Uji Kemiripan Kalimat Bahasa Indonesia

[1*]Ahmad Dzaky Hafidz Musta'in, [2]Ardi Sanjaya, [3]Ahmad Bagus Setiawan
[1,2,3] Teknik Informatika, Universitas Nusantara PGRI Kediri
E-mail: *[1]dzakymustain43@gmail.com, [2]dersky@gmail.com, [3]ahmadbagus@unpkediri.ac.id
Penulis Korespondens : Ardi Sanjaya

Abstrak— Penelitian ini membahas sistem analisis kemiripan kalimat menggunakan metode *cosine similarity* dengan fokus pada optimasi tahap pra-pemrosesan. Masalah utama yang diangkat adalah kebutuhan untuk mengenali pola angka Romawi dalam teks yang sering muncul dalam penamaan kelas atau bab dokumen. Metode yang digunakan melibatkan proses *case folding*, *tokenizing*, *filtering*, *stemming*, serta penggunaan *regular expression* untuk mendeteksi angka Romawi. Hasil pengujian menunjukkan sistem berhasil mengonversi angka Romawi dengan akurat. Namun, ditemukan kelemahan dalam konteks linguistik, seperti kesalahan interpretasi huruf pada nama khas daerah yang menyerupai pola angka Romawi. Hal ini menunjukkan perlunya integrasi pendekatan berbasis konteks untuk meningkatkan akurasi sistem. Untuk pengembangan lebih lanjut, disarankan penggunaan metode pembobotan tambahan atau pendekatan berbasis BERT guna meningkatkan pemahaman semantik kalimat.

Kata Kunci—Angka Romawi, *Cosine Similarity,* Kemiripan Kalimat, *Pre-Processing, Regular Expression*

Abstract— *This study discusses a sentence similarity analysis system using the cosine similarity method with a focus on optimizing the text preprocessing stage. The main issue addressed is the need to recognize Roman numeral patterns in texts, which frequently appear in class names or document sections. The method involves processes such as case folding, tokenizing, filtering, stemming, and the use of regular expressions to detect Roman numerals. The test results show that the system successfully converts Roman numerals accurately. However, a linguistic limitation was identified, such as misinterpreting characters in regional names that resemble Roman numeral patterns. This highlights the need for context-aware approaches to improve system accuracy. For future development, the use of additional weighting methods or BERT-based approaches is recommended to enhance semantic understanding of sentences.*
Keywords— *Cosine Similarity, Pre-Processing, Roman Numerals, Sentence Similarity, Regular Expression*

I. PENDAHULUAN

Pemrosesan Bahasa Alami (*Natural Language Processing*/NLP) merupakan cabang kecerdasan buatan yang berfokus pada pemahaman dan analisis bahasa manusia secara otomatis. Salah satu aplikasi penting dalam bidang ini adalah deteksi kemiripan kalimat. Salah satu metode populer untuk mengukur kesamaan antar kalimat adalah *cosine similarity*, yaitu dengan menghitung tingkat kesamaan (*similarity*) antar dua buah dokumen [1]. Namun, pendekatan ini menghadapi kendala ketika kalimat mengandung elemen numerik, terutama angka dalam format yang tidak lazim seperti angka Romawi.

Penelitian sebelumnya telah berhasil menangani angka numerik dengan mengonversinya menjadi teks menggunakan fungsi terbilang, yang terbukti meningkatkan nilai kemiripan. Namun,

pendekatan tersebut belum mampu mengatasi angka Romawi, yang dimana tidak adanya fungsi untuk menanganinya [2]. Pada artikel ini, peneliti mencoba mengembangkan proses *pre-processing* dengan *regular expression (regex)* untuk mengenali dan mengonversi angka Romawi ke bentuk numerik atau teks terbilang. Dengan pendekatan ini, sistem diharapkan dapat memproses kalimat secara lebih optimal dalam pengukuran kemiripan menggunakan *cosine similarity*.

II. METODE

Penelitian ini menggunakan metode pengembangan sistem untuk merancang alat pemeriksa kemiripan kalimat dalam bahasa Indonesia. Fokus utama dari pengembangan sistem ini adalah optimalisasi proses *pre-processing* teks, khususnya dengan mengintegrasikan fungsi *regular expression* untuk mengidentifikasi angka Romawi dalam kalimat. Proses *pre-processing* mencakup normalisasi teks, tokenisasi, penghapusan tanda baca, dan pengenalan angka Romawi menggunakan *regular expression*.

Sistem ini menggunakan metode *Cosine Similarity* untuk menghitung nilai kemiripan antar kalimat. Metode ini merepresentasikan kalimat sebagai vektor berdasarkan frekuensi kata (*term frequency*), dan menghitung sudut kosinus antara dua vektor sebagai ukuran kemiripan. Nilai *Cosine Similarity* berkisar antara 0 (tidak mirip) hingga 1 (identik) [2]. Untuk eksperimen, dilakukan pengujian terhadap pasangan kalimat yang telah diberi label kemiripan secara manual sebagai data pembanding (*ground truth*). Teknik evaluasi yang digunakan meliputi penghitungan akurasi sistem berdasarkan kesesuaian antara hasil sistem dan label manual.

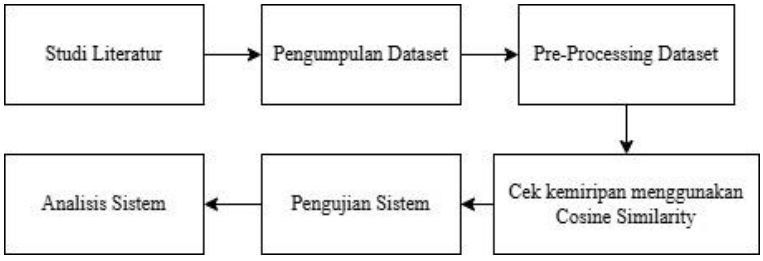
2.1 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini terdapat beberapa kumpulan pasangan kalimat Bahasa Indonesia yang dibuat oleh mahasiswa jurusan Pendidikan Bahasa Indonesia. Kalimat-kalimat tersebut disusun untuk mewakili beragam struktur kalimat dan mencakup unsur angka, termasuk angka Romawi, agar sesuai dengan kebutuhan pengujian kemiripan kalimat.

2.2 Pengolahan Data dan Metode yang Digunakan

Data diolah melalui beberapa tahap *preprocessing* menggunakan *Python*, seperti *lowercasing*, penghapusan tanda baca, dan tokenisasi. Angka Romawi dikenali dan dikonversi menggunakan *regular expression*, lalu diubah menjadi teks terbilang. Proses *tokenizing* dilakukan dengan *library* NLTK, lalu untuk proses *stemming* dan *stopword removal* dilakukan dengan *library* Sastrawi.

2.3 Prosedur Penelitian



Gambar 1. Prosedur Penelitian

Proses pengembangan sistem dimulai dengan studi literatur, yaitu tahapan untuk memahami teori-teori dasar dan penelitian sebelumnya yang relevan dengan analisis kemiripan kalimat serta metode *cosine similarity*. Setelah memiliki landasan teori yang kuat, dilakukan pengumpulan dataset berupa kalimat-kalimat yang akan dianalisis. Dataset ini kemudian memasuki tahap *pre-processing*, yaitu pembersihan dan standarisasi teks melalui beberapa proses seperti *case folding*, *tokenizing*, pengenalan angka Romawi dan terbilang, *stopword*, serta *stemming*. Setelah data selesai diproses, langkah selanjutnya adalah cek kemiripan menggunakan metode *cosine similarity*, yang bertujuan untuk mengukur sejauh mana kemiripan antara dua kalimat berdasarkan representasi vektornya. Hasil dari proses ini kemudian diuji dalam tahap pengujian sistem untuk melihat performa dan keakuratan metode yang digunakan. Tahapan terakhir adalah analisis sistem, di mana hasil pengujian dievaluasi guna mengidentifikasi kelebihan dan kekurangan sistem.

2.4 NLP (*Natural Language Processing*)

Natural Language Processing (NLP) adalah suatu pendekatan terkomputerisasi yang diterapkan untuk memahami dan menganalisis suatu teks atau bahasa alami [3]. Terdapat beberapa tahapan dari *Preprocessing* NLP yaitu :

a) *Case Folding*

Proses untuk mengubah huruf kapital yang ada pada kalimat menjadi huruf kecil [4]. Contoh: "Data SCIENCE" menjadi "data science". Tujuannya adalah agar tidak ada perbedaan antara kata yang ditulis dengan huruf besar dan kecil [5].

Teks sebelum diproses:

“Raja Louis XVI memerintah Prancis pada abad XVIII.”

Teks sesudah diproses:

“raja louis xvi memerintah prancis pada abad xviii.”

b) *Tokenizing*

Proses pemecahan kalimat menjadi kata-kata atau token [6]. Tahapan *tokenizing* bertujuan untuk memecah kalimat menjadi sebuah kata atau token dengan cara membelah kata dan mendefinisikan struktur atau token dengan cara menentukan unsur sintaksis per kata [7].

Teks sebelum diproses:

“raja louis xvi memerintah prancis pada abad xviii.”

Teks sesudah diproses:

[“raja”, “louis”, “xvi”, “memerintah”, “prancis”, “pada”, “abad”, “xviii”, “.”]

c) Fungsi Angka Romawi

Suatu proses untuk mendeteksi angka Romawi, seperti “XVI”, dan diubah ke bentuk angka numerik dengan pola tertentu menggunakan *regular expression*.

Teks sebelum diproses:

[“raja”, “louis”, “xvi”, “memerintah”, “prancis”, “pada”, “abad”, “xviii”, “.”]

Teks sesudah diproses:

[“raja”, “louis”, “16”, “memerintah”, “prancis”, “pada”, “abad”, “18”, “.”]

d) Fungsi Terbilang

Merupakan suatu proses untuk mengubah angka menjadi suatu teks [2]. Pada proses ini juga menghapus tanda baca seperti titik, koma, dll.

Teks sebelum diproses:

[“raja”, “louis”, “16”, “memerintah”, “prancis”, “pada”, “abad”, “18”, “.”]

Setelah diproses:

["raja", "louis", "enam belas", "memerintah", "prancis", "pada", "abad", "delapan belas", ""]

e) ***Stopword***

Kata-kata yang sering mucul namun tidak penting dan relevan, seperti kata hubung (konjungsi), kata kepunyaan, dan kata ganti orang dilakukan penghapusan dan penghilangan [6]. Contoh: "ke", "pada", "di".

Teks sebelum diproses:

["raja", "louis", "enam belas", "memerintah", "prancis", "pada", "abad", "delapan belas"]

Setelah diproses:

["raja", "louis", "enam belas", "memerintah", "prancis", "abad", "delapan belas"]

f) *Stemming*

Proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (*stem*) [8]. Contoh: "memakan" menjadi "makan".

Teks sebelum *stemming*:

["raja", "louis", "enam belas", "memerintah", "prancis", "abad", "delapan belas"]

Setelah *stemming*:

["raja", "louis", "enam belas", "perintah", "prancis", "abad", "delapan belas"]

2.5 Regex (*Regular Expression*)

Regular expression (*regex*) adalah notasi yang digunakan untuk mendeskripsikan pola dari kata yang ingin dicari [9]. *Regex* banyak digunakan dalam pemrosesan teks karena kemampuannya mengenali pola secara fleksibel dan efisien. Dalam sistem ini, *regex* dimanfaatkan untuk mengenali angka Romawi seperti "XII", "IV", dan "IX" yang umum ditemukan pada nama kelas atau bab. Pengenalan ini memungkinkan angka Romawi dikonversi ke bentuk numerik, sehingga tidak mengganggu analisis kemiripan kalimat. Salah satu pola regex yang digunakan adalah **\b[MCDLXVI]+\b**, yang mencocokkan kata berbasis huruf Romawi. Selain itu, *regex* juga membantu membersihkan teks dari simbol atau karakter yang tidak relevan, sehingga proses analisis menjadi lebih akurat dan konsisten.

2.6 *Cosine Similarity*

Perhitungan *cosine similarity* merupakan komponen dasar yang banyak digunakan pada aplikasi *data mining*. Secara garis besar metode *cosine similarity* ini didasarkan pada perhitungan sudut antara dua buah objek (contohnya dokumen 1 dan dokumen 2) yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran [10].

$$Similarity\ (A,B) = \frac{A.B}{\|A\|.\|B\|} = \frac{\sum_{i=1}^{n} A.B}{\sqrt{\sum_{i=1}^{n}(A)^2}.\sqrt{\sum_{i=1}^{n}(B)^2}} \qquad (1)$$

III. HASIL DAN PEMBAHASAN

3.1 Proses Pengolahan Data Kalimat (*Pre-Processing*)

Pada tabel 1 berisikan contoh kalimat yang telah melalui serangkaian tahapan *pre-processing* seperti *case folding, tokenizing,* konversi angka Romawi ke angka numerik, fungsi terbilang, *stopwords,* dan *stemming.*

Tabel 1. Hasil *Pre-Processing* Data Kalimat

No.	***Pre-Processing Dataset***	
	Kalimat Asli	**Kalimat Setelah *Pre-Processing***
1.	Bab V pada buku bahasa Indonesia membahas materi puisi.	bab lima buku bahasa indonesia bahas materi puisi
2.	Kelas XII akan mengadakan purna studi pada bulan Mei.	kelas dua belas ada purna studi bulan mei
3.	Tahun ini kelas X SMA Negeri II Kota menjuarai olimpiade tingkat Nasional.	tahun kelas puluh sma negeri dua kota juara olimpiade tingkat nasional
4.	Kerajaan Mataram berdiri pada akhir abad ke XVI.	raja mataram diri akhir abad enam belas
5.	Perang dunia II merupakan peristiwa bersejarah.	perang dunia dua rupa peristiwa sejarah
6.	Seluruh siswa kelas VI mempersiapkan diri untuk mengikuti ujian.	seluruh siswa kelas enam siap diri ikut uji
7.	Pertunjukan teater diadakan oleh angkatan X di hall kampus.	tunjuk teater ada angkat puluh hall kampus
8.	I Made Agus mengikuti ujian akhir kelas XII di aula sekolah.	satu made agus ikut uji akhir kelas dua belas aula sekolah
9.	Olimpiade Musim Panas XXVI akan diselenggarakan di kota Kediri.	olimpiade musim panas dua puluh enam selenggara kota diri
10.	DN LXV SMAN 2 Kediri mengundang Bernadya.	dn enam puluh lima sman dua diri undang bernadya

3.2 Hasil Perhitungan dengan *Cosine Similarity*

Pada tahap ini dilakukan perhitungan dengan *cosine similarity* untuk cek kemiripan menggunakan pasangan kalimat yang sudah melalui tahap *pre-processing*. Disajikan beberapa data kalimat pada tabel 2 dibawah ini untuk pengujian.

Tabel 2. Pengujian Cek Kemiripan Kalimat

No.	**Hasil Perhitungan *Cosine Similarity* dengan Kalimat *Pre-Processing***		
	Kalimat Pertama	**Kalimat Kedua**	**Skor Kemiripan (%)**
1.	bab lima buku bahasa indonesia bahas materi puisi	bab lima buku bahasa indonesia isi materi puisi	87,50

2.	kelas dua belas ada purna studi bulan mei	purna studi ada bulan mei kelas dua belas	100
3.	tahun kelas puluh sma negeri dua kota juara olimpiade tingkat nasional	tahun kelas puluh sma negeri dua kota raih juara olimpiade tingkat nasional	95,74
4.	raja mataram diri akhir abad enam belas	raja mataram diri akhir abad enam belas	100
5.	perang dunia dua rupa peristiwa sejarah	perang dunia dua jadi peristiwa sejarah dunia	81,65
6.	seluruh siswa kelas enam siap diri ikut uji	kelas enam beri waktu tambah ikut bimbing ajar siap uji	55,90
7.	tunjuk teater ada angkat puluh hall kampus	antusias tonton sangat tinggi saksi tunjuk teater angkat puluh	50,40
8.	dua belas mahasiswa ikut giat tanam hutan gundul	hutan bengkalai tangan dua belas mahasiswa	57,74
9.	belanja pasar tradisional nyaman banding pasar modern	pasar tradisional mudah ibu belanja proses tawar laku pasar tradisional laku	56,59
10.	mahasiswa tingkat empat universitas nusantara pgri diri sedang fokus susun skripsi	syarat lulus mahasiswa tingkat empat skripsi ikut tes toefl komputer	38,14
……			
35.	episode sembilan tokoh utama meni lelaki cinta	film seri capai musim sembilan balut cerita makin tarik	12,60

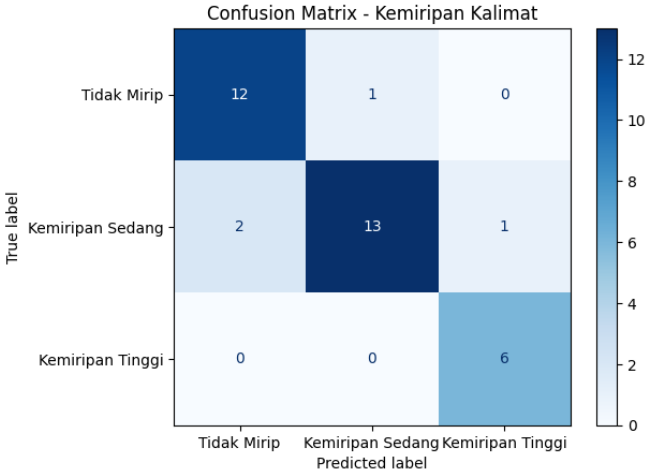
Pada pengujian di atas, sistem berhasil mendeteksi dan mengonversi angka Romawi menjadi angka numerik secara optimal. Proses ini dilakukan pada tahap pra-pemrosesan menggunakan *regular expression* (regex) yang dirancang khusus untuk mengenali pola angka Romawi. Sistem mampu mengenali angka Romawi dalam berbagai bentuk huruf (kapital maupun kecil) secara akurat, terutama setelah melalui proses *case folding*.

Keberhasilan ini menunjukkan bahwa *regex* efektif dalam menstandarkan input berbasis pola karakter. Namun, metode ini memiliki keterbatasan karena hanya bekerja secara struktural tanpa mempertimbangkan konteks linguistik. Sebagai contoh, dalam kalimat "I Made Agus mengikuti ujian akhir kelas XII*"*, kata "I" adalah bagian dari nama "I Made Agus", nama khas masyarakat Bali. Karena *regex* hanya mengenali huruf "I" sebagai bagian dari pola angka Romawi, sistem berpotensi salah mengidentifikasinya sebagai angka jika tidak dilengkapi dengan logika kontekstual tambahan. Hal ini menunjukkan bahwa meskipun *regex* sangat berguna dalam normalisasi pola tertentu, ia tidak memiliki kemampuan semantik untuk membedakan elemen linguistik seperti nama orang, sehingga perlu integrasi pendekatan berbasis konteks dalam sistem yang lebih kompleks.

3.3 Evaluasi *Confusion Matrix*

Pada Gambar 2 menunjukkan *confusion matrix* hasil pengujian sistem dalam mengevaluasi kemiripan kalimat menggunakan 35 pasangan kalimat yang telah diberi label manual (*ground truth*) dengan tiga kategori, yaitu 'tidak mirip', 'kemiripan sedang', dan

'kemiripan tinggi'. Berdasarkan *confusion matrix* tersebut, diperoleh hasil evaluasi sebagai berikut:



Gambar 2. Hasil Visual Evaluasi *Confusion Matrix*

a) Akurasi

$$Akurasi = \frac{\sum TP_i}{Jumlah\ Data} = \frac{12+13+6}{35} = \frac{31}{35} \approx 0.8857 = 0.89 \quad (2)$$

b) Presisi, Recall, dan F1-Score per Kelas

1) Kelas: Tidak Mirip

True Positive (TP) = 12, *False Positive* (FP) = 2, *False Negative* (FN) = 1

$$Presisi = \frac{TP}{TP + FP} = \frac{12}{12+2} = \frac{12}{14} \approx 0.857 \quad (3)$$

$$Recall = \frac{TP}{TP + FN} = \frac{12}{12+1} = \frac{12}{13} \approx 0.923 \quad (4)$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.857 \times 0.923}{0.857 + 0.923} \approx 0.889 \quad (5)$$

2) Kelas: Kemiripan Sedang

True Positive (TP) = 13, *False Positive* (FP) = 1, *False Negative* (FN) = 3

$$Presisi = \frac{TP}{TP + FP} = \frac{13}{13+1} = \frac{13}{14} \approx 0.929 \quad (6)$$

$$Recall = \frac{TP}{TP + FN} = \frac{13}{13+3} = \frac{13}{16} \approx 0.812 \quad (7)$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.929 \times 0.812}{0.929 + 0.812} \approx 0.866 \quad (8)$$

3) Kelas: Kemiripan Tinggi

True Positive (TP) = 6, *False Positive* (FP) = 1, *False Negative* (FN) = 0

$$Presisi = \frac{TP}{TP + FP} = \frac{6}{6+1} = \frac{6}{7} \approx 0.857 \quad (9)$$

$$Recall = \frac{TP}{TP + FN} = \frac{6}{6+0} = \frac{6}{6} \approx 1.00 \quad (10)$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} = 2 \times \frac{0.857 \times 1.00}{0.857 + 1.00} \approx 0.923 \quad (11)$$

c) Macro Average

$$Presisi_{Macro} = \frac{0.857+0.929+0.857}{3} \approx 0.881 \quad (12)$$

$$Recall_{Macro} = \frac{0.923+0.812+1.00}{3} \approx 0.911 \quad (13)$$

$$F1 - Score_{Macro} = \frac{0.889+0.866+0.923}{3} \approx 0.893 \quad (14)$$

IV. KESIMPULAN

Sistem analisis kemiripan kalimat yang dibangun dengan metode *cosine similarity* dan didukung oleh proses pra-pemrosesan seperti *regex* terbukti mampu mengenali pola-pola tertentu secara teknis, termasuk angka Romawi, dengan tingkat keberhasilan yang tinggi. Namun, meskipun sistem bekerja secara optimal dalam mengenali struktur permukaan teks, masih terdapat kelemahan mendasar, terutama dalam menangkap konsep linguistik atau makna semantik yang lebih dalam.

Metode *cosine similarity* hanya menghitung kemiripan berdasarkan representasi kata dalam bentuk vektor, tanpa memahami konteks, sinonim, makna kalimat secara keseluruhan, atau struktur sintaksis. Demikian pula, penggunaan *regex* hanya efektif untuk pola teks yang eksplisit dan tidak fleksibel terhadap variasi makna bahasa. Mungkin tidak dianggap mirip secara signifikan oleh sistem, karena urutan kata dan bentuk angka berbeda, meskipun secara semantik keduanya menyampaikan pesan yang sama.

Oleh karena itu, meskipun sistem ini efektif untuk kemiripan berbasis bentuk teks, pendekatan ini belum mampu menangani analisis makna linguistik secara menyeluruh, dan masih memerlukan pengembangan lebih lanjut, seperti integrasi dengan *word embedding* atau *semantic similarity models* (misalnya BERT atau Word2Vec) untuk hasil yang lebih kontekstual dan akurat.

DAFTAR PUSTAKA

[1] Sugiyamta, "Sistem Deteksi Kemiripan Dokumen Dengan Algoritma Cosine Similarity Dan Single Pass Clustering," *Dinamika Informatika*, vol. 7, no. 2, hlm. 85–91, 2015.

[2] A. Sanjaya dan S. D. Sasongko, "Uji Kemiripan Kalimat Menggunakan Fungsi Terbilang Pada Pre-Processing Dan Cosine Similarity Dalam Bahasa Indonesia Sentences Similarity Test Using Countable Function On Pre-Processing And Cosine In Indonesian," *Jurnal Ilmiah NERO*, vol. 7, no. 2, hlm. 95–104, 2022.

[3] D. O. Sihombing, "Implementasi Natural Language Processing (NLP) dan Algoritma Cosine Similarity dalam Penilaian Ujian Esai Otomatis," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 2, hlm. 396, Des 2022, doi: 10.30865/json.v4i2.5374.

[4] R. S. Amardita, A. Adiwijaya, dan M. D. Purbolaksono, "Analisis Sentimen terhadap Ulasan Paris Van Java Resort Lifestyle Place di Kota Bandung Menggunakan Algoritma KNN," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 1, hlm. 62, Feb 2022, doi: 10.30865/jurikom.v9i1.3793.

[5] N. Nurwanda, N. Suarna, dan W. Prihartono, "PENERAPAN NLP (NATURAL LANGUAGE PROCESSING) DALAM ANALISIS SENTIMEN PENGGUNA TELEGRAM DI PLAYSTORE," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, hlm. 1841–1846, Apr 2024, doi: 10.36040/jati.v8i2.8469.

[6] S. J. Angelina, A. B. Putra Negara, dan H. Muhardi, "Analisis Pengaruh Penerapan Stopword Removal Pada Performa Klasifikasi Sentimen Tweet Bahasa Indonesia," *Jurnal Aplikasi dan Riset Informatika*, vol. 02, no. 01, hlm. 165–173, Agu 2023.

[7] Y. Yuhandri, R. Sovia, A. Syaiffullah, F. Yenila, dan R. Permana, "Penerapan Natural Language Processing Pada Sistem Chatbot Sebagai Helpdesk Obyek Wisata Menggunakan Metode Naïve Bayes," *Jurnal Infortech*, vol. 5, no. 2, hlm. 210–218, Jan 2024, doi: 10.31294/infortech.v5i2.20911.

[8] M. S. H. Simarangkir, "STUDI PERBANDINGAN ALGORITMA - ALGORITMA STEMMING UNTUK DOKUMEN TEKS BAHASA INDONESIA," *Jurnal Inkofar*, vol. 1, no. 1, Agu 2017, doi: 10.46846/jurnalinkofar.v1i1.2.

[9] D. Nur Fadhillah dan A. Rachman, "Implementasi Regex Pada Pemberian Komentar Kode Program Html," *Jurnal Advance Research Informatika*, vol. 2, no. 1, 2023, [Daring]. Tersedia pada: https://www.ejournalwiraraja.com/index.php/JARS

[10] S. Lumbansiantar, S. Dwiasnati, dan N. S. Fatonah, "Penerapan Metode Cosine Similarity dalam Mendeteksi Plagiarisme pada Jurnal," 2023.