

**ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK ULASAN  
APLIKASI SIREKAP 2024 MENGGUNAKAN CART DAN METODE  
WORDNET**

**SKRIPSI**

**Diajukan Oleh :**

**Muhammad Fariz Ijlal Rafi**

**2011102441124**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR  
SAMARINDA**

**2024**

**ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK ULASAN  
APLIKASI SIREKAP 2024 MENGGUNAKAN CART DAN METODE  
WORDNET**

**SKRIPSI**

Diajukan Sebagai Salah Satu Persyaratan Untuk Memenuhi Gelar Sarjana Fakultas Sains dan  
Teknologi Universitas Muhammadiyah Kalimantan Timur

**Diajukan Oleh :**

**Muhammad Fariz Ijlal Rafi**

**2011102441124**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR  
SAMARINDA**

**2024**

## LEMBAR PERSETUJUAN

### ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK ULASAN APLIKASI SIREKAP 2024 MENGGUNAKAN CART DAN METODE WORDNET

Diajukan Oleh :

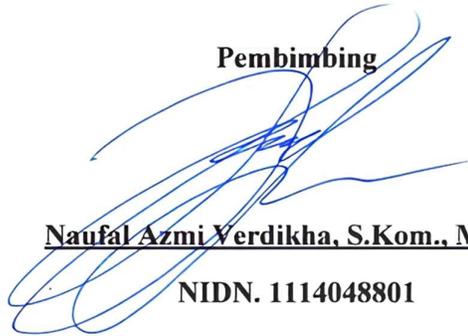
**MUHAMMAD FARIZ IJLAL RAFI**

2011102441124

Disetujui untuk diujikan

Pada tanggal 28 Juni 2024

**Pembimbing**



**Naufal Azmi Verdikha, S.Kom., M.Eng.**

NIDN. 1114048801

**Mengetahui,**

**Koordinator Tugas Akhir/Skripsi/Tesis/Disertasi**



**Abdul Rahim, S.Kom., M.Cs.**

NIDN. 0009047901

# LEMBAR PENGESAHAN

## ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK ULASAN APLIKASI SIREKAP 2024 MENGGUNAKAN CART DAN METODE WORDNET

### SKRIPSI

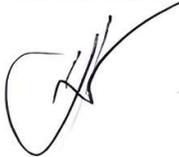
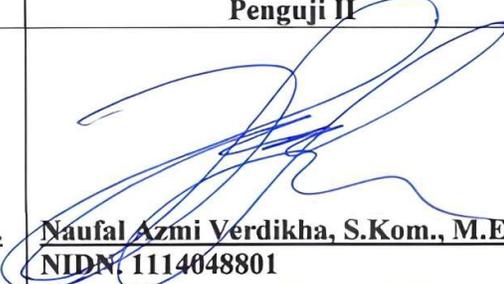
Diajukan Oleh:

Muhammad Fariz Ijlal Rafi

2011102441124

Diseminarkan dan Diujikan

Pada tanggal 11 Juli 2024

Penguji I	Penguji II
 <u>Taghfirul Azhima Yoga Siswa, S.Kom., M.Kom.</u> NIDN. 1118038805	 <u>Naufal Azmi Verdikha, S.Kom., M.Eng.</u> NIDN. 1114048801

Mengetahui,

Ketua

Program Studi Teknik Informatika



Arbansyah, S.Kom., M.T.I.

NIDN. 1118019203

## PERNYATAAN KEASLIAN PENELITIAN

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Fariz Ijlal Rafi  
NIM : 2011102441124  
Program Studi : S1 Teknik Informatika  
Judul Penelitian : ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK ULASAN APLIKASI SIREKAP 2024 MENGGUNAKAN CART DAN METODE *WORDNET*

Menyatakan bahwa **skripsi** yang saya tulis ini benar-benar hasil karya saya sendiri, dan bukan merupakan hasil plagiasi/falsifikasi/fabrikasi baik sebagian atau seluruhnya.

Atas pernyataan ini, saya siap menanggung resiko atau sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam **skripsi** saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Samarinda, 18 Juni 2024  
Yang membuat pernyataan



Muhammad Fariz Ijlal Rafi  
NIM. 2011102441124

## ABSTRAK

Model klasifikasi sentimen adalah model yang digunakan untuk mengkategorikan teks berdasarkan sentimen yang terkandung di dalamnya. Terdapat beberapa jenis model klasifikasi sentimen seperti model *Wordnet* dan algoritma klasifikasi *Machine Learning* seperti model *Decision Tree* CART. Penelitian ini bertujuan untuk membandingkan dua jenis model klasifikasi sentimen yaitu *Wordnet* dan *Decision Tree* dengan algoritma CART dalam mengklasifikasi sentimen dari sebuah ulasan yang dievaluasi menggunakan *F1-Score*. Dataset yang digunakan adalah data ulasan aplikasi SIREKAP 2024 dari *Google Playstore* dengan *rating* 1 hingga 5 bintang yang diambil pada awal bulan februari 2024. Hasil penelitian menunjukkan, model *Decision Tree* dengan algoritma CART memiliki nilai *F1-Score* yang lebih tinggi yaitu sebesar 31,62%, sedangkan model *Wordnet* memiliki nilai *F1-Score* sebesar 17,70%. Penelitian ini membuktikan bahwa kedua model dapat digunakan untuk mengklasifikasi sentimen terhadap ulasan aplikasi SIREKAP 2024, dan model *Decision Tree* dengan algoritma CART terbukti lebih baik dibandingkan model *Wordnet* dalam mengklasifikasi sentimen dari ulasan.

**Kata Kunci:** Analisis Sentimen, Ulasan, SIREKAP 2024, *Wordnet*, *Decision Tree*, CART

## ***ABSTRACT***

Sentiment classification model is a model used to categorize text based on the sentiment contained in it. There are several types of sentiment classification models such as the *Wordnet* model and *Machine Learning* classification algorithms such as the *Decision Tree* CART model. This study aims to compare two types of sentiment classification models, namely *Wordnet* and *Decision Tree* with the CART algorithm in classifying sentiment from a review evaluated using *F1-Score*. The dataset used is the SIREKAP 2024 application review data from *Google Playstore* with a *rating* of 1 to 5 stars taken in early February 2024. The results of the study showed that the *Decision Tree* model with the CART algorithm had a higher *F1-Score* value of 31.62%, while the *Wordnet* model had an *F1-Score* value of 17.70%. This study proves that both models can be used to classify sentiment towards the SIREKAP 2024 application review, and the *Decision Tree model* with the CART algorithm is proven to be better than the *Wordnet* model in classifying sentiment from reviews.

**Keywords:** Sentiment Analysis, Reviews, SIREKAP 2024, *Wordnet*, *Decision Tree*, CART

## PRAKATA

Alhamdulillah, dengan mengucapkan puji syukur kehadiran Allah SWT karena atas petunjuk-Nya, penulis dapat menyelesaikan skripsi ini dengan judul “ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK ULASAN APLIKASI SIREKAP 2024 MENGGUNAKAN CART DAN METODE *WORDNET*” sebagai salah satu syarat untuk menyelesaikan Program Sarjana (S1) Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Muhammadiyah Kalimantan Timur.

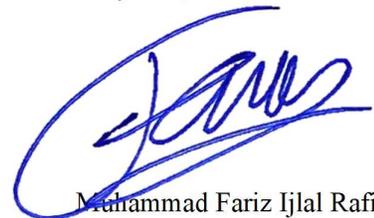
Segala kekurangan dalam penulisan skripsi ini sangat disadari oleh penulis, namun atas bantuan dan dukungan moral maupun materi dari sebagian pihak dapat terpenuhi segala kelengkapan dalam penulisan. Pada kesempatan ini penulis menyampaikan ucapan terimakasih serta penghargaan yang setinggi-tingginya untuk berbagai pihak yang telah memberikan bantuan baik secara langsung atau tidak langsung.

1. Kedua orang tua tercinta Ayahanda Kusnadi dan Ibunda Surtinah yang telah memberikan dukungan serta semangat, doa, nasehat, motivasi kepada penulis sehingga proposal ini dapat diselesaikan dengan baik.
2. Bapak Naufal Azmi Verdikha, S.kom., M.Eng sebagai dosen pembimbing yang telah memberikan waktu bimbingan baik masukan, saran dan pengetahuan yang sangat membantu dalam menyelesaikan skripsi ini.
3. Bapak Taghfirul Azhima Yoga Siswa, S.Kom., M.Kom., sebagai dosen penguji yang telah memberikan saran-saran bermanfaat, sehingga penulis dapat menyempurnakan isi proposal skripsi ini.
4. Bapak Arbansyah, S.Kom., M.TI selaku ketua Program Studi Teknik Informatika.
5. Teman-teman seperjuangan baik dari prodi Teknik Informatika maupun dari prodi lain yang tidak bisa disebutkan satu-satu.

Semoga semua do'a, bimbingan dan dukungan yang telah diberikan kepada penulis mendapat balasan dari Allah SWT. Penulis berharap semoga laporan ini memberikan kesan yang bermanfaat kepada pembaca.

Samarinda, 18 Juni 2024

Penyusun,



Muhammad Fariz Ijlal Rafi

# DAFTAR ISI

	<b>Halaman</b>
HALAMAN JUDUL.....	ii
LEMBAR PERSETUJUAN.....	iii
LEMBAR PENGESAHAN.....	iv
PERNYATAAN KEASLIAN PENELITIAN.....	v
ABSTRAK .....	vi
<i>ABSTRACT</i> .....	vii
PRAKATA.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL .....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN.....	xiii
BAB I PENDAHULUAN.....	14
1.1    Latar Belakang Masalah.....	14
1.2    Rumusan Masalah .....	15
1.3    Tujuan Penelitian.....	15
1.4    Manfaat Penelitian.....	15
BAB II METODE PENELITIAN .....	16
2.1    Objek Penelitian .....	16
2.2    Alat dan Bahan .....	16
2.3    Prosedur Penelitian.....	17
2.3.1    Pengambilan Data.....	17
2.3.2 <i>Pra-proses</i> Data.....	18
2.3.3 <i>Wordnet</i> .....	19
2.3.4    Klasifikasi <i>Decision Tree</i> CART.....	23
BAB III HASIL DAN PEMBAHASAN.....	27
3.1    Hasil .....	27
3.1.1    Pengambilan Data.....	27
3.1.2 <i>Pra-proses</i> Data.....	30
3.1.3 <i>Wordnet</i> .....	33
3.1.4    Klasifikasi <i>Decision Tree</i> CART.....	35
3.2    Pembahasan.....	40
3.2.1    Pengambilan dan Analisis Data.....	40
3.2.2    Analisis Hasil <i>Pra-proses</i> .....	40
3.2.3    Analisis Komparasi Model Klasifikasi <i>Wordnet</i> dan CART.....	40
BAB IV PENUTUP .....	42

4.1	Kesimpulan.....	42
4.2	Saran.....	42
	DAFTAR PUSTAKA.....	43
	LAMPIRAN.....	45
	RIWAYAT HIDUP.....	60

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
2.1 Library <i>Python</i> .....	16
2.2 Pelabelan Kelas <i>Wordnet</i> .....	20
3.1 Jumlah Kemunculan Data Pada Setiap <i>Rating</i> .....	28
3.2 <i>Fold 1 Cross-Validation</i> .....	37
3.3 <i>Fold 10 Cross-Validation</i> .....	38
3.4 Hasil <i>F1-Score</i> pada Setiap <i>Fold</i> .....	38
3.5 Hasil <i>F1-Score</i> Kedua Metode .....	41

## DAFTAR GAMBAR

Gambar	Halaman
2.1 Alur Penelitian.....	17
2.2 Penggunaan <i>Deep-translator</i> .....	20
2.3 Penggunaan <i>TextBlob</i> dan Klasifikasi <i>Wordnet</i> .....	20
2.4 <i>Confusion Matrix</i> .....	21
2.5 <i>Confusion Matrix Multiclass</i> Pada Kelas 1 <i>Star</i> .....	21
2.6 <i>K-fold Cross-Validation</i> K=10 .....	26
3.1 Hasil Pengambilan Data Ulasan SIREKAP 2024 .....	27
3.2 Hasil Input Data.....	27
3.3 Distribusi <i>Rating</i> .....	28
3.4 <i>Wordcloud Rating</i> 1.....	29
3.5 <i>Wordcloud Rating</i> 5.....	29
3.6 Hasil Input Id.....	30
3.7 Hasil <i>Lower Case</i> .....	30
3.8 Hasil <i>Remove Unnecessary Character</i> .....	31
3.9 Kamus Tidak Baku.....	31
3.10 Hasil <i>Spell Checker</i> .....	32
3.11 Hasil <i>Stemming</i> .....	32
3.12 Hasil Hapus Data Kosong .....	33
3.13 Hasil <i>Pra-proses</i> .....	33
3.14 Hasil <i>Deep-translator</i> .....	34
3.15 Hasil <i>Wordnet</i> .....	34
3.16 <i>Confusion Matrix</i> dengan <i>Wordnet</i> .....	35
3.17 komentar <i>_stemming</i> .....	36
3.18 <i>Rating</i> .....	36
3.19 Hasil <i>TF-IDF</i> .....	37
3.20 <i>F1-Score</i> 10 <i>Fold</i> .....	39
3.21 <i>Confusion Matrix Overall</i> dengan <i>Decision Tree</i> CART .....	39
3.22 Komparasi Hasil <i>F1-Score</i> .....	40

## DAFTAR LAMPIRAN

	<b>Halaman</b>
<b>Lampiran 1</b> Proses Pengambilan Data.....	45
<b>Lampiran 2</b> Memasukan Data Ke <i>Dataframe</i> Pandas .....	45
<b>Lampiran 3</b> Filtering dan Sorting Kolom .....	45
<b>Lampiran 4</b> Ekspor Data Ke File CSV .....	45
<b>Lampiran 5</b> Input File CSV .....	45
<b>Lampiran 6</b> Proses Distribusi <i>Rating</i> .....	46
<b>Lampiran 7</b> Proses <i>Wordcloud</i> .....	46
<b>Lampiran 8</b> Proses Pemberian ID Disetiap Data .....	47
<b>Lampiran 9</b> Proses <i>Lower Case</i> .....	47
<b>Lampiran 10</b> Proses <i>Remove Unnecessary Character</i> .....	47
<b>Lampiran 11</b> Source Code Kamus "4 lebih list kata KBBI.txt" .....	48
<b>Lampiran 12</b> Cek Kata Tidak Baku.....	48
<b>Lampiran 13</b> Cek List Kata Tidak Baku.....	49
<b>Lampiran 14</b> Proses Input "kamus_tidak_baku" .....	49
<b>Lampiran 15</b> Proses <i>Spell Checker</i> .....	49
<b>Lampiran 16</b> Proses <i>Stemming</i> .....	49
<b>Lampiran 17</b> Cek Data Kosong .....	50
<b>Lampiran 18</b> Hapus Data Kosong .....	50
<b>Lampiran 19</b> Proses Penerjemahan Data Ulasan ke dalam Bahasa Inggris.....	50
<b>Lampiran 20</b> Fungsi Hitung Score.....	51
<b>Lampiran 21</b> Proses <i>Wordnet</i> .....	51
<b>Lampiran 22</b> Proses Evaluasi <i>F1-Score Wordnet</i> .....	52
<b>Lampiran 23</b> Proses Penentuan Input dan Kelas .....	53
<b>Lampiran 24</b> Proses Ekstraksi Fitur TF-IDF .....	53
<b>Lampiran 25</b> Proses <i>Cross-Validation 10 Fold</i> .....	55
<b>Lampiran 26</b> Menampilkan Jumlah Data Setiap Pembagian <i>Fold</i> .....	55
<b>Lampiran 27</b> Proses Input Model Klasifikasi <i>Decision Tree</i> CART .....	55
<b>Lampiran 28</b> Proses Visualisasi Model dengan Menjalankan <i>Cross Validation</i> .....	55
<b>Lampiran 29</b> Surat Keterangan Pengambilan Data Sekunder .....	56
<b>Lampiran 30</b> Kartu Kendali Bimbingan Laporan Karya Ilmiah.....	57

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Menjelang pemilihan umum tahun 2024, aplikasi SIREKAP 2024 menjadi trending di Platform X karena banyak kritik pada ulasan terkait keandalannya dan masalah aksesibilitas<sup>1</sup>. SIREKAP 2024 adalah Sistem yang diterapkan oleh Komisi Pemilihan Umum (KPU) dan Kelompok Penyelenggara Pemungutan Suara (KPPS) yang bertujuan untuk memfasilitasi proses input dan pengumpulan hasil pemilu tahun 2024. KPPS menggunakan aplikasi versi mobile yang bisa di unduh di *Goggle Play Store* (Kasim, 2024). Namun, pada awal bulan Februari 2024 (sebelum Pemilu 2024) aplikasi tersebut memiliki sekitar delapan ribuan ulasan dengan rata-rata *rating* 2,7 dari 5 bintang. Hal ini menunjukkan adanya ketidakpuasan pengguna dan perlu dianalisis lebih lanjut.

Pada sebuah aplikasi atau produk, pengguna mengungkapkan emosi dan sentimennya dalam sebuah ulasan. Ulasan dari pengguna lain dianggap mempunyai informasi yang penting karena ulasan tersebut mempunyai objektivitas pada penilaian sebuah produk (Lestari & Putra, 2022). Hal tersebut juga diperkuat oleh penelitian (Darmawan, 2023) yang dimana ulasan terbukti mempunyai fungsi penting dalam memilih produk atau aplikasi. Namun, ulasan dari pengguna tidak dapat langsung dipastikan apakah aplikasi tersebut baik atau buruk karena sering terdapat ulasan yang memiliki *rating* tidak konsisten dengan komentar yang diberikan oleh pengguna aplikasi (Safitri & Widiati, 2022).

Permasalahan tersebut membuat pengembang aplikasi kesulitan mengetahui cara meningkatkan kinerja aplikasi SIREKAP 2024 berdasarkan sentimen pengguna yang terdapat dalam delapan ribu lebih ulasan. Selain itu, hal ini menjadi tantangan bagi calon pengguna untuk membaca semua ulasan dan memutuskan apakah akan menggunakan aplikasi SIREKAP 2024. Oleh karena itu, diperlukan alat untuk memahami sentimen publik dalam ulasan. Analisis Sentimen, salah satu cabang *Natural Language Processing* (NLP), berfokus pada identifikasi dan pelabelan otomatis emosi dan sentimen yang diungkapkan dalam teks tertulis (Tan et al., 2023)

Salah satu metode yang dapat digunakan untuk menganalisis sentimen dari ulasan pada sebuah aplikasi adalah *Wordnet*. *Wordnet* seringkali digunakan dalam hal pelabelan otomatis dan klasifikasi. Penelitian (Fatihin, 2022), menggunakan metode *Wordnet* untuk mengklasifikasi ulasan pengguna aplikasi PeduliLindungi. Hasil penelitian menunjukkan dari 1000 sampel ulasan pengguna PeduliLindungi, diperoleh 620 ulasan positif dan 380 ulasan negatif. Penelitian (Nengsih et al., 2021), menggunakan metode *Wordnet* dalam mengklasifikasikan ulasan pelanggan Hotel. Hasil penelitian menunjukkan nilai akurasi kurva *Receiver Operating Characteristic* (ROC) sebesar 90%, yaitu 68% ulasan positif dan 32% ulasan negatif.

Terdapat metode lain dalam mengklasifikasikan ulasan. Pada penelitian (Dangi et al., 2022), menggunakan model klasifikasi *Logistic Regression*, *Random Forest*, *Multinomial Naïve Bayes*, *Support Vector Machine* (SVM), dan *Decision Tree* dengan algoritma *Classification and Regression Tree* (CART) dalam menganalisis sentimen sosial media twitter terkait COVID-19. Hasil akurasi dari masing masing model adalah 94%, 95%, 98%, 97%, dan 99% yang menunjukkan *Decision Tree*

---

<sup>1</sup> <https://teknologi.bisnis.com/read/20240210/84/1739635/aplikasi-kpu-sirekap-trending-di-platform-x-ini-penyebabnya>

dengan algoritma CART memiliki hasil akurasi yang lebih tinggi dibandingkan model klasifikasi lainnya pada dataset sentimen pasca lockdown di twitter.

Penelitian lain yang memperkuat algoritma CART lebih baik dalam klasifikasi adalah (Bayu Setiawan & Sulisty Nugroho, 2023), menggunakan metode *Decision Tree* dengan 3 algoritma yang dibandingkan yaitu CART, *Iterative dichotomiser 3* (ID3), dan C4.5. Hasilnya, CART memiliki hasil evaluasi tertinggi dengan nilai *accuracy*, *Precision*, dan *Recall* masing-masing sebesar 72%, 92,59%, dan 74,62%. Sedangkan ID3 dan C4.5 memiliki hasil evaluasi yang sama yaitu masing-masing sebesar 67,26%, 85,71%, dan 71,47%.

Dalam mengklasifikasi ulasan berbentuk data teks, dibutuhkan metode untuk mengekstraksi fitur teks. Penelitian (Ahuja et al., 2019) menggunakan ekstraksi fitur *term frequency-inverse document frequency* (TF-IDF) dan *N-gram* dalam menganalisis sentimen pada Twitter. Hasil penilitan menunjukkan ekstraksi fitur TF-IDF mempunyai kinerja 3-4% lebih tinggi dibandingkan *N-gram*. Penelitian (Munir et al., 2022) menggunakan ekstraksi fitur TF-IDF dan Unigram dengan metode *Support Vector Machine* (SVM) dalam menganalisis sentimen terhadap review hotel pada website portal. Hasilnya ekstraksi fitur TF-IDF memiliki akurasi yang lebih tinggi sebesar 95%, sedangkan ekstraksi fitur Unigram memiliki akurasi sebesar 94%.

Penelitian ini bertujuan untuk membandingkan hasil evaluasi dua tahapan analisis sentimen pada aplikasi SIREKAP 2024 dan melihat metode mana yang lebih baik. Tahapan pertama digunakan metode *Wordnet* yang dibagi menjadi lima kelas sentimen berdasarkan *rating* yang terdapat di ulasan SIREKAP 2024. Tahapan kedua menggunakan *Decision Tree* dengan algoritma CART. Fitur ekstraksi TF-IDF akan digunakan untuk mengekstraksi fitur data ulasan pada SIREKAP 2024 untuk diklasifikasikan menggunakan algoritma CART. Kedua tahapan penelitian tersebut dievaluasi menggunakan nilai evaluasi *F1-Score*.

## 1.2 Rumusan Masalah

Bagaimana komparasi hasil evaluasi *F1-Score* pada tahapan penggunaan metode *Wordnet* dan tahapan penggunaan klasifikasi *Decision Tree* dengan algoritma CART dan ekstraksi fitur TF-IDF untuk analisa sentimen ulasan SIREKAP 2024?

## 1.3 Tujuan Penelitian

Mengkomparasi hasil evaluasi *F1-Score* pada tahapan penggunaan metode *Wordnet* dan tahapan penggunaan klasifikasi *Decision Tree* dengan algoritma CART dan ekstraksi fitur TF-IDF untuk analisa sentimen ulasan SIREKAP 2024.

## 1.4 Manfaat Penelitian

Dari hasil penelitian ini diharapkan mampu memberikan dampak sebagai berikut.

1. Memberikan wawasan kepada organisasi dan pengembang aplikasi tentang pendekatan terbaik dalam analisis sentimen ulasan SIREKAP 2024 untuk meningkatkan strategi bisnis dan pemasaran.
2. Menyediakan panduan praktis bagi praktisi dan analis sentimen dalam memilih metode analisis yang paling sesuai untuk mengoptimalkan sumber daya dan akurasi evaluasi ulasan.
3. Menambah referensi dan pengetahuan bagi peneliti dan akademisi dalam domain analisis sentimen serta mendorong penelitian lanjutan dan inovasi metode baru.

## BAB II

### METODE PENELITIAN

#### 2.1 Objek Penelitian

SIREKAP 2024 adalah aplikasi yang disiapkan oleh KPU untuk melakukan pencatatan dan pendokumentasian hasil penghitungan suara di TPS. Aplikasi SIREKAP 2024 telah tersedia di *Google Play Store* dengan total unduhan mencapai satu juta lebih dan delapan ribuan ulasan di awal bulan februari 2024. Setiap ulasan di aplikasi tersebut sangat bermacam-macam dan memiliki *rating* dari 1 sampai 5 bintang dengan jumlah ulasan yang berbeda pada setiap *rating*-nya. Dari delapan ribuan ulasan tersebut akan diambil dan menjadi objek penelitian ini.

#### 2.2 Alat dan Bahan

Alat yang akan digunakan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak. Perangkat keras terdiri dari Laptop Acer NITRO V15 dengan spesifikasi prosesor 13th Gen Intel(R) Core(TM) i5-13420H 2.10 GHz, kapasitas memori 16GB, dan hardisk SSD 512GB. Sedangkan perangkat lunak yang akan digunakan berupa sistem operasi *Microsoft Windows 11*, *Visual Studio Code* versi 1.87.2, *Jupyter Notebook*

versi 2024.2.0, *Python* versi 3.12.1, dan *Library Python*. *Library Python* yang akan digunakan dapat dilihat pada Tabel 2.1.

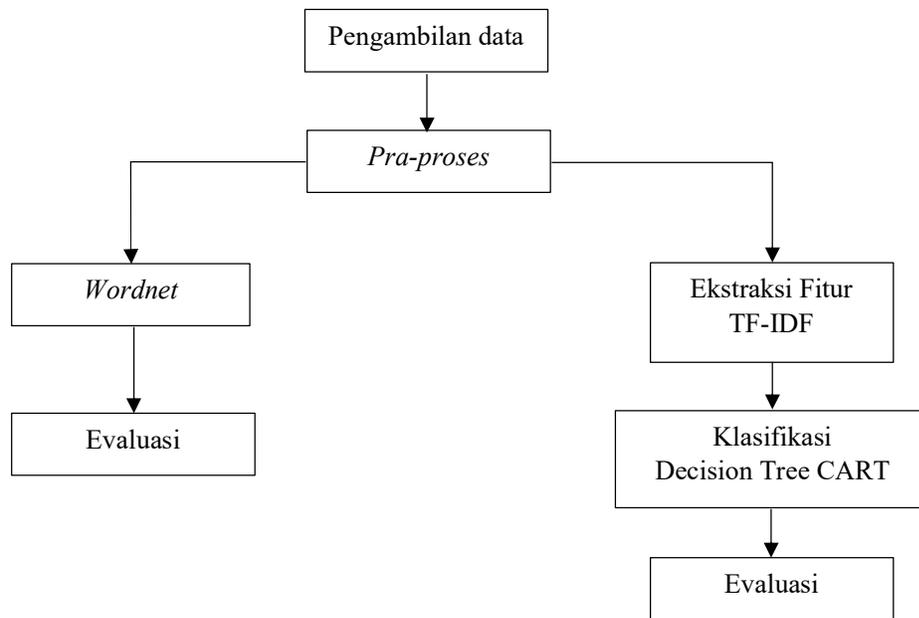
Tabel 2.1 *Library Python*

<i>Library Python</i>	Versi	Keterangan
<i>Pandas</i>	2.2.0	untuk mengelola dan menganalisis data
<i>Numpy</i>	1.26.3	pustaka <i>Python</i> yang dirancang khusus untuk manipulasi data numerik
<i>Matplotlib</i>	3.8.2	pustaka plot populer dengan <i>Python</i> yang digunakan untuk membuat visualisasi dan grafik berkualitas tinggi
NLTK ( <i>Natural Language Toolkit</i> )	3.8.1	untuk membangun program <i>Python</i> agar dapat bekerja dengan data bahasa manusia
<i>Scikit-learn</i>	1.4.0	pustaka <i>Machine Learning</i> yang menyediakan berbagai algoritma <i>Machine Learning</i>
<i>TextBlob</i>	0.18.0	untuk menganalisa sentimen teks, untuk mengetahui apakah sebuah teks bersifat positif, negatif, atau netral pada sentimen analisis
<i>Sastrawi</i>	1.0.1	mengubah kata berimbuhan ke bentuk dasarnya atau yang biasa disebut <i>root</i>
<i>Deep-translator</i>	1.11.4	untuk menerjemahkan berbagai bahasa dengan cara sederhana menggunakan banyak penerjemah
<i>Wordcloud</i>	1.9.3	mengvisualisasikan data kedalam bentuk gambar
<i>Google-play-scraper</i>	1.2.6	API untuk mengekstraksi data informasi aplikasi dan ulasan aplikasi dari <i>Google Play Store</i> dengan lebih mudah tanpa ketergantungan eksternal

Adapun bahan yang akan digunakan dalam penelitian ini adalah data yang terdiri dari *rating* dan ulasan yang diambil dari aplikasi SIREKAP 2024 di *Google Playstore* pada tanggal 6 Februari 2024 jam 22:00 WITA.

### 2.3 Prosedur Penelitian

Tahap penelitian yang akan digunakan untuk membandingkan hasil evaluasi dua tahapan analisis sentimen pada aplikasi SIREKAP 2024 dapat ditunjukkan pada Gambar 2.1.



**Gambar 2.1** Alur Penelitian

Tahap awal yang akan dilakukan dalam penelitian ini yaitu pengambilan data. Setelah itu, data akan melalui tahap *Pra-proses*. Kemudian, data dari tahapan *Pra-proses* akan digunakan untuk klasifikasi dan analisis ulasan yakni *Wordnet* dan *Decision Tree* CART. Dalam tahapan *Wordnet*, tahapan setelah *Pra-proses* akan diklasifikasikan ke dalam kelas *rating*. Lalu dalam tahapan *Decision Tree* CART, tahapan setelah *Pra-proses* akan melalui tahapan ekstraksi fitur TF-IDF terlebih dahulu sebelum diklasifikasikan menggunakan model *Decision Tree* CART. Terakhir, masing-masing model akan di evaluasi dan di analisis perbandingan dari hasil kedua model tersebut.

#### 2.3.1 Pengambilan Data

Data yang akan digunakan dalam penelitian ini merupakan data ulasan pada aplikasi SIREKAP 2024 di *Google Play Store*<sup>2</sup>. Dataset diambil pada tanggal 6 februari 2024 menggunakan software *Visual Studio Code* dengan library *Python* yakni *Numpy*, *Pandas*, dan *Google-play-scraper*. Library ini bertujuan untuk mengambil data ulasan dari aplikasi SIREKAP 2024 di *Google Play Store*.

Dalam mengambil data ulasan terdapat tahapan yang akan dilakukan. Tahap pertama yaitu mengimpor library yang akan digunakan ke dalam *Visual Studio Code*. Selanjutnya, memasukan link

<sup>2</sup> <https://play.google.com/store/apps/details?id=id.go.kpu.sirekap2024>

id dari aplikasi SIREKAP 2024 untuk mengambil data ulasan dari aplikasi tersebut. Setelah mengambil data ulasan, langkah selanjutnya membuat *dataframe* (df) untuk menampilkan seluruh data dalam bentuk tabel dan menampilkan jumlah data yang terambil. Kemudian df akan diatur untuk filter data yang diperlukan pada penelitian ini. Selanjutnya, data diekspor dengan format CSV dan menggunakan separator ‘|’ sebagai pembatas pada data tersebut.

### 2.3.2 *Pra-proses Data*

*Pra-proses* data merupakan langkah di mana data diubah, digabungkan, atau disesuaikan agar dapat dimengerti oleh sistem dengan optimal (Cholis & Ulinnuha, 2023). Dataset yang didapat dari tahap pengambilan data merupakan data mentah dan perlu dilakukan *Pra-proses* data. Pada tahapan penelitian ini, menggunakan beberapa tahapan dalam *Pra-proses* data sebagai berikut.

#### 1. *Lower Case*

*Lower Case* akan merubah data ulasan berupa teks menjadi bentuk huruf kecil untuk membantu mesin dalam menghindari karakter berbeda yang seharusnya sama.

#### 2. *Remove Unnecessary Character*

*Remove Unnecessary Character* akan menghapus atribut spesifik dan semua karakter non-alfanumerik atau karakter yang tidak diperlukan pada ulasan seperti simbol, emoji, karakter yang berulang, tanda baca, dan karakter spesial lainnya.

#### 3. *Spell Checker*

*Spell Checker* akan digunakan untuk memastikan konsistensi tingkat tinggi dalam teks yang digunakan dalam penelitian dengan menstandarkan kata-kata yang berlebihan dan tidak baku ke dalam satu konteks. *Spell Checker* dapat membantu tidak hanya dalam menemukan dan memperbaiki kesalahan ejaan tetapi juga dalam menghilangkan variasi dan redundansi kata, yang dapat berdampak pada interpretasi dan analisis temuan penelitian.

Pada tahap ini, standarisasi akan menggunakan *Kamus Bahasa Indonesia* (Bahasa, 2008) untuk mendeteksi kata yang tidak baku. Sebelum dilakukan *Spell Checker*, akan dibuat kamus bernama “*kamus\_tidak\_baku*” dalam format *Comma Separated Value* (CSV) untuk standarisasi kata-kata tidak jelas dan berlebih seperti kata “*ancurrrrrrr*” yang distandarisasi menjadi “*hancur*”, kata “*lemot*” menjadi “*lambat*” dan sebagainya.

Tahap pertama dalam membuat kamus yang akan digunakan dalam penelitian ini adalah mengkonversi Kamus Bahasa Indonesia dari bentuk *Portable Document Format* (PDF), menjadi *Text File* (TXT) dan diimpor ke dalam *Python*. Selanjutnya, karakter non-alfanumerik yang terdapat pada kamus akan dihapus dan teks dipecah menjadi kata-kata. Kata-kata tersebut kemudian difilter untuk menghapus kata yang mengandung numerik dan mengambil kata yang berjumlah lebih dari sama dengan 4 huruf. Hal ini dilakukan karena kata-kata yang hanya terdiri dari 1 – 3 huruf masih ada yang memiliki konteks yang kurang jelas. Setelah itu, hasilnya akan diekspor dengan nama “*4 list lebih kata KBBI.txt*”

Selanjutnya, “*4 list lebih kata KBBI.txt*” ini akan dijadikan acuan untuk mendeteksi kata tidak baku atau kata yang tidak terdeteksi oleh acuan tersebut di dalam data ulasan. Kata yang tidak tidak baku kemudian di ekspor kedalam format CSV dan setiap kata akan diubah secara manual dengan cara menambahkan kolom baru yang berisi kata baku dari kata tidak baku tersebut. Setelah semua

kata-kata yang tidak baku telah dibakukan, hasilnya akan diekspor dengan nama “kamus\_tidak\_baku.csv” dan siap digunakan untuk proses *Spell Checker*.

#### 4. *Stemming*

*Stemming* merupakan tahap untuk mengubah kata imbuhan menjadi kata dasar menggunakan library *Sastrawi* seperti ‘mencobanya’ yang berasal dari kata dasar ‘coba’, ‘dipakai’ yang memiliki kata dasar ‘pakai’, dan ‘menjalankan’ menjadi ‘jalan’. *Stemming* pada penelitian ini akan menggunakan *Sastrawi.Stemmer.StemmerFactory* yang terdapat pada library *Sastrawi*.

#### 5. Hapus Data Kosong

Tahap akhir dari *Pra-proses* data adalah hapus data kosong. Hapus data kosong dilakukan untuk membantu menghilangkan bias yang mungkin disebabkan oleh nilai yang hilang, sehingga analisis sentimen diharapkan dapat memberikan hasil yang lebih akurat dan dapat diandalkan. Proses awal hapus data kosong yaitu mengecek nilai yang memiliki kosong yang terdapat pada hasil *Stemming* dengan menggantinya dengan ‘NaN’ menggunakan fungsi ‘*replace*’. Selanjutnya, ‘NaN’ dihapus menggunakan fungsi ‘*df.dropna*’. Hasilnya akan digunakan untuk proses selanjutnya.

### 2.3.3 *Wordnet*

#### 1. *Wordnet*

*Wordnet* adalah basis data leksikal yang menggambarkan hubungan semantik antar kata, yang digunakan untuk mengukur polaritas dan subjektivitas dalam suatu teks. Nilai polaritasnya adalah desimal dalam rentang [-1, 1], sementara nilai subjektivitasnya adalah desimal dalam rentang [0, 1]. Nilai 0 mencerminkan objektivitas yang tinggi, dan nilai 1 menunjukkan subjektivitas yang tinggi. (Fjærli & Larsen, 2022).

Pada penelitian ini, data akan diterjemahkan terlebih dahulu ke dalam bahasa Inggris menggunakan API *Google Translate* yang terdapat pada library *Deep-translator* karena *Wordnet* hanya dapat memproses teks dalam bahasa Inggris. Implementasi kode pada tahap ini dapat dilihat pada gambar 2.2. Selanjutnya, *Wordnet* menggunakan library *TextBlob* untuk mencari nilai polaritas kata pada data ulasan.

Nilai polaritas yang diperoleh akan diberikan kelas yang dibedakan menjadi lima sentimen berdasarkan *rating* bintang satu sampai lima dari ulasan aplikasi SIREKAP 2024. Lima kelas akan dinyatakan sebagai tingkat ulasan pengguna terhadap aplikasi tersebut dengan rentang nilai kurang dari -0,5 sampai lebih dari 0,5 seperti pada tabel 2.2. Implementasi kode pada tahap penggunaan *TextBlob* dan pengklasifikasiannya dapat dilihat pada gambar 2.3.

**Tabel 2.2** Pelabelan Kelas *Wordnet*

Values	Peringkat
$<(-0,5)$	1 Star
$(-0,5) - (-0,2)$	2 Star
$(-0,1) - 0,1$	3 Star
$0,2 - 0,5$	4 Star
$>0,5$	5 Star

```
1 from deep_translator import GoogleTranslator
2 import time
3
4 def translate_komentar(text):
5     text_translated = GoogleTranslator(source='id', target='en').translate(text)
6     return text_translated
7
```

**Gambar 2.2** Penggunaan *Deep-translator*

```
1 from textblob import TextBlob
2
3 def scoring_sentiment(text):
4     analysis = TextBlob(text)
5     sentiment_score = analysis.sentiment.polarity
6     # print("Sentiment polarity:", sentiment_score)
7     score_wordnet = 0
8
9     if sentiment_score > 0.5:
10        score_wordnet = '5 star'
11    elif 0.2 <= sentiment_score <= 0.5:
12        score_wordnet = '4 star'
13    elif -0.2 < sentiment_score < 0.2:
14        score_wordnet = '3 star'
15    elif -0.5 <= sentiment_score <= -0.2:
16        score_wordnet = '2 star'
17    else:
18        score_wordnet = '1 star'
19
20    return score_wordnet, sentiment_score
```

**Gambar 2.3** Penggunaan *TextBlob* dan Klasifikasi *Wordnet*

Hasil dari implementasi diatas akan memperoleh nilai polaritas dari setiap teks ulasan. Nilai polaritas kemudian diklasifikasikan dalam lima kelas sentimen berdasarkan rentang nilai yang telah ditentukan. Setiap ulasan akan diberikan skor yang sesuai berdasarkan nilai polaritas yang didapatkan. Proses ini memberikan gambaran tentang tingkat kepuasan pengguna terhadap aplikasi SIREKAP 2024, yang selanjutnya akan digunakan sebagai data untuk evaluasi menggunakan *F1-Score*.

## 2. Evaluasi *F1-Score Wordnet*

Evaluasi pada penelitian ini menggunakan *F1-Score*. untuk mendapatkan *F1-Score* menggunakan *Confusion Matrix*. *Confusion Matrix* merupakan metode pengujian untuk klasifikasi yang bekerja dengan membandingkan hasil prediksi dengan hasil aktual dari dataset (Kurniawati & Arianto, 2019). Contoh Pengujian dengan *Confusion Matrix* dapat dilihat pada Gambar 2.4.

		Predicted	
		Positive	Negative
A c t u a l	Positive	TP	FN
	Negative	FP	TN

**Gambar 2.4** *Confusion Matrix*

*True Positif* (TP) adalah jumlah data positif yang diklasifikasikan sebagai nilai positif dan *True Negatif* (TN) adalah jumlah data negatif yang diklasifikasikan sebagai nilai negatif, sedangkan *False Positif* (FP) merupakan jumlah data positif yang diklasifikasikan sebagai nilai negatif dan *False Negatif* (FN) merupakan jumlah data negatif yang diklasifikasikan sebagai nilai positif.

Namun gambar 2.4 hanya dapat digunakan untuk klasifikasi biner yakni klasifikasi dengan dua kelas. Untuk mengklasifikasi data yang lebih dari dua kelas, memerlukan *Confusion Matrix* dengan klasifikasi *Multiclass*. klasifikasi multi class merupakan klasifikasi lebih dari dua kelas dimana setiap sampel dibebankan untuk satu label (Husin, 2023). Contoh *Confusion Matrix* dengan *Multiclass* yang akan digunakan dalam penelitian ini dapat dilihat pada Gambar 2.5.

		Predicted				
		1 Star	2 Star	3 Star	4 Star	5 Star
A c t u a l	1 Star	TP	FN	FN	FN	FN
	2 Star	FP	TN	TN	TN	TN
	3 Star	FP	TN	TN	TN	TN
	4 Star	FP	TN	TN	TN	TN
	5 Star	FP	TN	TN	TN	TN

**Gambar 2.5** *Confusion Matrix Multiclass Pada Kelas 1 Star*

Dari nilai yang terdapat pada gambar 2.5, yakni TP, FP, TN, dan FN, akan digunakan untuk menghitung nilai *Precision*, *Recall*, dan *F1-Score*. *Precision* merupakan persentase kejadian serangan yang diklasifikasikan dengan benar sebagai serangan, atau nilai prediksi positif, Sedangkan *Recall* merupakan efektivitas suatu model dalam mengidentifikasi suatu serangan (Bagui & Li, 2021). Nilai *Precision* setiap kelas akan diperoleh dari persamaan (1) dan nilai *Recall* setiap kelas akan diperoleh dari persamaan (2).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Setelah nilai *Precision* dan *Recall* pada setiap kelas diperoleh, selanjutnya akan dicari nilai *F1-Score* pada setiap kelas. *F1-Score* merupakan nilai tengah yang diperoleh dari nilai *Precision* dan *Recall*. *F1-Score* berfungsi untuk menguji model pada data yang tidak seimbang (Rizki et al., 2023). nilai *Precision* dan *Recall* setiap kelas yang telah diperoleh akan digunakan untuk mencari nilai *F1-Score* pada masing-masing kelas menggunakan persamaan (3).

$$F1\ Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (3)$$

Nilai *F1-Score* pada setiap kelas yang diperoleh dari persamaan (3) akan digunakan untuk mencari rata-rata dari nilai keseluruhan *F1-Score* menggunakan *Average Macro*. *F1-Score Average Macro* menjadi pilihan terbaik jika dataset tidak seimbang dan tanpa mempertimbangkan proporsi kelasnya karena memperlakukan semua kelas secara setara (Siregar et al., 2023). *F1-Score Average Macro* akan menggunakan persamaan (4).

$$Macro\ F1\ Score = \frac{\sum F1\ Score}{n} \quad (4)$$

dimana:

- $\sum F1\ Score$  : jumlah seluruh nilai *F1-Score* per kelas
- $n$  : jumlah kelas

Nilai yang di hasilkan dari evaluasi *F1-Score Average Macro* ini akan menjadi hasil evaluasi dari model *Wordnet* dalam klasifikasi sentimen dari ulasan aplikasi SIREKAP 2024. Evaluasi pada penelitian ini akan menggunakan *sklearn.metrics* di library *Scikit-learn*.

### 2.3.4 Klasifikasi *Decision Tree* CART

#### 1. TF-IDF

Ekstraksi fitur merupakan suatu teknik untuk mengenali ciri-ciri yang berkaitan dengan klasifikasi. Terdapat dua jenis ekstraksi fitur, yakni melalui pendekatan *Machine Learning* dan metode *lexicon based* (Keerthi Kumar et al., 2019). Salah satu ekstraksi fitur *Machine Learning* yang sering digunakan yakni TF-IDF. TF-IDF adalah ekstraksi fitur yang bekerja dimana kata pada kalimat umum akan dihitung dengan bobot pada setiap kata (Pratomo et al., 2021). Tingkat kepentingan kata dalam dokumen atau kumpulan kata tertentu diukur dengan TF-IDF sebagai ukuran statistik.

TF-IDF menghitung dua faktor penting yakni *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). TF adalah metode untuk menilai seberapa sering frekuensi kata yang muncul. sedangkan IDF adalah metode untuk menilai seberapa penting kata dalam konteks koleksi dokumen yang lebih besar dengan membagi jumlah total dokumen dalam koleksi dan jumlah dokumen yang mengandung kata tersebut. Hasilnya kemudian diambil logaritma untuk meratakan skala (Septiani & Isabela, 2022).

Pada penelitian ini, ekstraksi fitur TF-IDF akan menggunakan `sklearn.feature_extraction` yang terdapat pada library *Scikit-learn*. Untuk menghitung ekstraksi TF-IDF digunakan persamaan (5).

$$B_{xy} = tf(x, y) \times idf(x) \quad (5)$$

dimana:

- $B(x, y)$  : bobot term ke-x pada dokumen y
- $tf(x, y)$  : frekuensi kemunculan term ke-x pada dokumen y

Kemudian untuk nilai IDF, didapatkan melalui persamaan berikut.

$$idf(x) = \log \frac{1 + n}{1 + dfx} + 1 \quad (6)$$

dimana:

- $idf(x)$  : nilai idf term x
- $dfx$  : jumlah dokumen dalam kumpulan dokumen yang mengandung term x
- $n$  : jumlah dokumen

Setelah nilai TF-IDF dihasilkan, data akan di normalisasi untuk mengatur skala nilai data dalam rentang 0 sampai 1. Penelitian ini akan menggunakan *Euclidean norm* atau *L2-norm* dengan persamaan (7).

$$v_{norm} = \frac{v}{\|v\|_2} = \frac{v}{\sqrt{v_1^2 + v_2^2 + v_3^2 + \dots + v_n^2}} \quad (7)$$

dimana:

- $v$  : nilai tf-idf yang akan di normalisasi
- $\|v\|_2$  : jumlah kuadrat nilai tf-idf dari semua dokumen

Berikut beberapa parameter ekstraksi fitur TF-IDF dalam library *Scikit-learn* yang akan digunakan pada penelitian ini.

- a. `ngram_range`: Parameter yang akan digunakan untuk menentukan rentang panjang n-gram atau nilai n selama proses ekstraksi fitur, pengguna dapat mengatur nilai minimum dan maximum dari n-gram sesuai keinginan. Nilai yang akan digunakan dalam penelitian ini (1,1) dengan parameter `ngram_range = (1,1)`, artinya hanya menghitung unigram.
- b. `norm`: Parameter yang akan digunakan untuk menentukan jenis normalisasi pada hasil vektor. Nilai yang akan digunakan l2, menghitung jumlah kuadrat dari setiap elemen yang terletak pada vektor, mengambil akar kuadrat dari hasil penjumlahan, dan kemudian membagi nilai setiap elemen yang terletak pada vektor dengan nilai akar kuadrat tersebut akan membuat dokumen menjadi normal.

## 2. *Decision Tree* CART

*Decision Tree* melakukan klasifikasi dimana nilai dan atribut digambarkan berdasarkan *node* yang ada (Nurfauzan & Maharani, 2021). Algoritma *Decision Tree* yang dibuat untuk masalah klasifikasi dan regresi diwakili oleh metode CART. CART adalah sistem pengambilan keputusan berbasis pohon biner yang bercabang menjadi dua *node* anak di setiap persimpangan (Romadhonia et al., 2023).

CART digunakan untuk membagi objek ke dalam dua atau lebih kelompok. Dalam sebuah dataset yang memiliki p variabel bebas dan satu variabel terikat, jika variabel terikat tersebut merupakan kategori, CART akan membentuk pohon klasifikasi. Sebaliknya, jika variabel terikat tersebut bersifat kontinu atau numerik, CART akan membentuk pohon regresi (Bayu Setiawan & Sulisty Nugroho, 2023). Pohon keputusan dengan algoritma CART dibangun melalui perhitungan indeks Gini untuk setiap kelas (Suryani et al., 2022), menggunakan rumus yang diberikan dalam persamaan 8.

$$Gini\ j(t) = 1 - \sum_{i=1}^c p_i^2(j|t) \quad (8)$$

dimana:

- $c$  : jumlah kelas
- $P_i(j|t)$  : probabilitas dalam *node* t yang termasuk kelas j

Dalam persamaan 8, *node* t dibagi menjadi dua subset, yakni *L* dan *R*, dengan ukuran masing-masing  $N_L$  dan  $N_R$ . Setelah itu, indeks Gini total untuk pembagian ini dihitung menggunakan subset tersebut, dan hasilnya dapat ditemukan dalam persamaan 9.

$$Gini\ split\ (t) = \frac{N_L}{N} giniL + \frac{N_R}{N} giniR \quad (9)$$

dimana:

- $N_L$  : banyaknya data dalam subset  $N_L$
- $N_R$  : banyaknya data dalam subset  $N_R$
- $N$  : banyaknya data dalam sebuah variabel
- $\text{gini}_L$  : nilai indeks gini dari subset L untuk setiap variabel
- $\text{gini}_R$  : nilai indeks gini dari subset R untuk setiap variabel

Apabila jumlah sampel dalam suatu kelas adalah 5 atau kurang, *node* tersebut akan dijadikan sebagai *node* terminal (Bayu Setiawan & Sulisty Nugroho, 2023). Penentuan label untuk *node* terminal ini dilakukan sesuai persamaan 10, dengan mempertimbangkan jumlah terbanyak.

$$P(j_0|t) = \max_j P(j|t) + \max_j \frac{m_j(t)}{m(t)} \quad (10)$$

Tahapan ini akan dilakukan proses klasifikasi dengan *Decision Tree* dengan algoritma CART menggunakan *DecisionTreeClassifier* yang ada pada library *Scikit-learn*. Beberapa parameter yang dibutuhkan adalah sebagai berikut.

- Criterion*: merupakan parameter yang digunakan untuk menentukan metode pengukuran kualitas split pada setiap *node* pohon keputusan. Dalam penelitian ini digunakan “*gini*” yang mengukur ketidakmurnian (*impurity*) dari kelas di dalam *node*.
- Max Depth*: merupakan parameter untuk menentukan kedalaman maximum dari pohon keputusan. Kedalaman maksimum akan diatur menjadi “*None*” agar Pohon akan terus tumbuh hingga setiap *leaf* adalah murni atau berisi kurang dari ‘*min\_samples\_split*’.
- Min Samples Split*: merupakan parameter untuk menentukan jumlah minimum sampel yang diperlukan untuk melakukan split pada *node* internal. Pada penelitian ini akan menggunakan 2 sampel pada suatu *node* untuk melakukan split.
- Min Samples Leaf*: merupakan parameter untuk menentukan jumlah minimum sampel yang diperlukan pada *leaf*. Penelitian ini akan menggunakan 1 sampel pada suatu *leaf*.
- Max Features*: merupakan parameter untuk menentukan jumlah fitur yang dipertimbangkan untuk melakukan split pada setiap *node*. Penelitian ini akan menggunakan semua fitur yang tersedia (*None*).
- Class Weight*: merupakan parameter untuk menentukan bobot yang diberikan pada setiap kelas. Untuk memulai penelitian ini, tidak ada penyesuaian bobot kelas yang dilakukan (*None*).

Dalam penggunaan *DecisionTreeClassifier*, yang pertama adalah mengimpor modul dari *Scikit-learn*. Modul ini termasuk *DecisionTreeClassifier* untuk membangun pohon keputusan dan parameter yang telah disebutkan diatas (Source code terlampir pada lampiran 27). Setelah mengimpor modul yang diperlukan, Selanjutnya adalah mengevaluasi menggunakan *F1-Score* dengan teknik *K-fold Cross-Validation* dengan  $k=10$ .

### 3. Evaluasi *F1-Score* CART

Evaluasi pada tahapan ini menggunakan *F1-Score* yang sama metode pencariannya dengan metode evaluasi *F1-Score Wordnet*. Namun, evaluasi *F1-Score* pada tahapan ini menggunakan *K-fold Cross-Validation*. *K-fold Cross-Validation* merupakan teknik validasi dimana kumpulan data independen akan digeneralisasi dengan statistik analisis untuk dinilai bagaimana hasilnya (Sari et al., 2023). Teknik statistik yang disebut *K-fold Cross-Validation* membagi data menjadi data train dan data test untuk menilai dan membandingkan algoritma pembelajaran (Nurnawati et al., 2023).

Teknik *K-fold Cross-Validation* berguna untuk mengevaluasi performa dari proses suatu algoritma dengan cara data sampel yang digunakan akan dibagi secara random dan kemudian akan dikelompokkan sebanyak nilai K yang digunakan. Salah satu kelebihan dari menggunakan *K-Fold Cross-Validation* dalam pengujian adalah kemampuannya untuk mengungkap model dengan akurasi tertinggi. Ini terjadi karena data sampel dibagi secara acak ke dalam K bagian, memungkinkan untuk melihat variasi model yang paling unggul dalam komposisi yang optimal (Nurainun et al., 2023). *K fold Cross Validation* pada penelitian ini akan menggunakan *Kfold* yang terdapat pada library *Scikit-learn*. Penggunaan *K-fold Cross-Validation* dengan nilai k=10 seperti Gambar 2.6.

Fold 1	Test	Train								
Fold 2	Train	Test	Train							
Fold 3	Train	Train	Test	Train						
Fold 4	Train	Train	Train	Test	Train	Train	Train	Train	Train	Train
Fold 5	Train	Train	Train	Train	Test	Train	Train	Train	Train	Train
Fold 6	Train	Train	Train	Train	Train	Test	Train	Train	Train	Train
Fold 7	Train	Train	Train	Train	Train	Train	Test	Train	Train	Train
Fold 8	Train	Test	Train	Train						
Fold 9	Train	Test	Train							
Fold 10	Train	Test								

**Gambar 2.6** *K-fold Cross-Validation* K=10

Evaluasi *F1-Score* tahapan ini akan menggunakan *sklearn.metrics* yang terdapat pada library *Scikit-learn*, dengan parameter untuk setiap *fold* dan perhitungan rata-rata dijelaskan dibawah ini.

- y\_target*: nilai sebenarnya dari sebenarnya dari target atau label dari klasifikasi
- y\_pred*: nilai yang telah diprediksi oleh model klasifikasi
- f1*: fungsi untuk menilai *F1-Score* dari setiap *fold*

## BAB III HASIL DAN PEMBAHASAN

### 3.1 Hasil

#### 3.1.1 Pengambilan Data

Data ulasan aplikasi SIREKAP 2024 diambil menggunakan library *Google-play-scrapper* yang diambil pada tanggal 6 februari 2024. Berikut hasil data ulasan yang terambil dapat dilihat pada gambar 3.1.

	userName	score	at	content	thumbsUpCount
6671	Nency Miranda	5	2024-02-05 14:07:12	Mantap	0
5783	ichal sabiel	1	2024-02-05 14:06:17	Sudah tau pemakainya masyarakat biasa Malah sp...	0
7909	Evandra Aditya	1	2024-02-05 14:06:16	Apk nya nggk bisa buat log in	0
6307	Ganesh insann	1	2024-02-05 14:05:38	Susah masuk dih	0
8008	Yohana Frediana	1	2024-02-05 14:04:32	Tidak bisa masuk inisiliasi	0

**Gambar 3.1** Hasil Pengambilan Data Ulasan SIREKAP 2024

Hasil data yang telah terambil diekspor menggunakan format *Comma Separated Values (CSV)* dengan nama “scrapped\_data.csv” untuk dianalisis lebih lanjut. Selanjutnya, data ulasan diinput menggunakan format CSV ke *Python* dan hasilnya ditunjukkan pada Gambar 3.2.

```

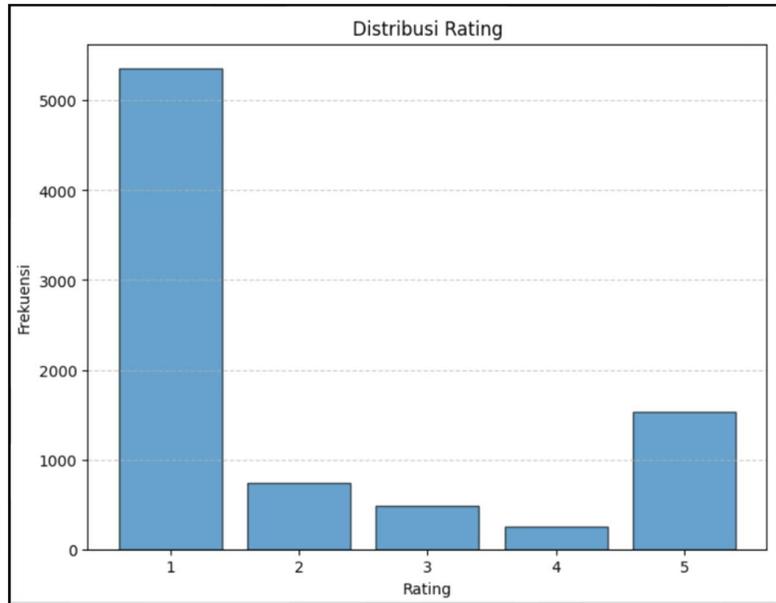
      Nama Rating Waktu \
0      Nency Miranda 5 05/02/2024 14:07
1      ichal sabiel 1 05/02/2024 14:06
2      Evandra Aditya 1 05/02/2024 14:06
3      Ganesh insann 1 05/02/2024 14:05
4      Yohana Frediana 1 05/02/2024 14:04
...
8353      Tn A Muntaha 5 23/01/2024 09:05
8354      Adhi Nugroho 1 23/01/2024 08:51
8355      TATANG RUSDIANA 5 23/01/2024 07:03
8356      vie 5 23/01/2024 06:50
8357      Ahmad ferdiansyah 5 23/01/2024 04:26

      Komentar Like
0      Mantap 0.0
1      Sudah tau pemakainya masyarakat biasa Malah sp... 0.0
2      Apk nya nggk bisa buat log in 0.0
3      Susah masuk dih 0.0
4      Tidak bisa masuk inisiliasi 0.0
...
8353      Semoga Tambah Baik,Mudah Dalam Input Dan Share... 21.0
8354      Cara masuk nya bagaimana? Pakai email kok gag ... 448.0
8355      Mbuh 6.0
8356      Semoga tidak seperti sirekap 2020 yang saat di... 42.0
8357      Semoga dengan adanya aplikasi ini pemilu akan ... 186.0

[8358 rows x 5 columns]
```

**Gambar 3.2** Hasil Input Data

Dapat dilihat pada gambar diatas, data yang digunakan terdiri dari 8358 baris dan 5 kolom. Yang berisikan mulai dari Nama, *Rating*, Waktu, Komentar, dan Like. Setelah data diinput, tahap pertama dalam menganalisis data yang dilakukan adalah Distribusi *Rating*. Distribusi *rating* dilakukan untuk melihat berapa banyak ulasan yang terdapat pada setiap *rating*. Hasil distribusi *rating* dapat dilihat pada gambar 3.3.



**Gambar 3.3** Distribusi *Rating*

Terlihat pada gambar 3.3, *rating* 1 memiliki frekuensi kemunculan yang sangat tinggi dibandingkan *rating* lainnya. Selain itu, frekuensi kemunculan pada *rating* 2, 3, dan 4 yang berada diantara *rating* 1 dan *rating* 5 sangat rendah. Ini menunjukkan bahwa data ulasan memiliki data yang tidak seimbang. Jumlah kemunculan data pada setiap *rating* dapat dilihat pada tabel 3.1.

**Tabel 3.1** Jumlah Kemunculan Data Pada Setiap *Rating*

<i>Rating</i>	Jumlah kemunculan data
1	5361
2	736
3	478
4	255
5	1528

Tabel 3.1 menunjukkan jumlah data pada setiap *rating*. *Rating* 1 berisi 5361 data, *Rating* 2 berisi 736 data, *Rating* 3 berisi 478 data, *Rating* 4 berisi 255 data, dan *Rating* 5 berisi 1528 data. Selanjutnya digunakan *Wordcloud* untuk memvisualisasikan data ulasan pada setiap *rating* dalam bentuk gambar dan melihat frekuensi kata yang sering muncul.



### 3.1.2 Pra-proses Data

Setiap data diberikan id untuk mempermudah pelacakan dan identifikasi setiap baris secara individual. Hasilnya dapat dilihat pada gambar 3.6.

	ID	Nama	Rating	Waktu \	
0	d0001	Nency Miranda	5	05/02/2024 14:07	
1	d0002	ichal sabiel	1	05/02/2024 14:06	
2	d0003	Evandra Aditya	1	05/02/2024 14:06	
3	d0004	Ganesh insann	1	05/02/2024 14:05	
4	d0005	Yohana Frediana	1	05/02/2024 14:04	
...	...	...	...	...	
8353	d8354	Tn A Muntaha	5	23/01/2024 09:05	
8354	d8355	Adhi Nugroho	1	23/01/2024 08:51	
8355	d8356	TATANG RUSDIANA	5	23/01/2024 07:03	
8356	d8357	vie	5	23/01/2024 06:50	
8357	d8358	Ahmad ferdiansyah	5	23/01/2024 04:26	
				Komentar Like	
0				Mantap	0.0
1		Sudah tau pemakainya masyarakat biasa		Malah sp...	0.0
2		Apk nya nggk bisa buat log in			0.0
3		Susah masuk dih			0.0
4		Tidak bisa masuk inisiliasi			0.0
...		...		...	...
8353		Semoga Tambah Baik,Mudah Dalam Input Dan Share...			21.0
8354		Cara masuk nya bagaimana? Pakai email kok gag ...			448.0
8355		Mbuh			6.0
8356		Semoga tidak seperti sirekap 2020 yang saat di...			42.0
8357		Semoga dengan adanya aplikasi ini pemilu akan ...			186.0

[8358 rows x 6 columns]

Gambar 3.6 Hasil Input Id

Terlihat pada gambar 3.6 telah ditambahkan kolom id kumpulan data. Kolom id terdapat pada kolom ke 1 dengan menambahkan id “d0001” dan seterusnya yang kemudian ditambahkan pada data ulasan masing-masing. Tahap berikutnya hanya menggunakan kolom Komentar, karena hanya ulasan pengguna dalam kolom Komentar yang diperlukan pada proses klasifikasi teks.

#### 1. Lower Case

Data ulasan diubah dari kata yang mengandung huruf kapital menjadi huruf kecil. Gambar 3.7 merupakan perbandingan antara sebelum dilakukan *Lower Case* dan setelah dilakukan *Lower Case*.

	Komentar \	komentar_lowercase
0	Mantap	mantap
1	Sudah tau pemakainya masyarakat biasa Malah sp...	sudah tau pemakainya masyarakat biasa malah sp..
2	Apk nya nggk bisa buat log in	apk nya nggk bisa buat log in
3	Susah masuk dih	susah masuk dih
4	Tidak bisa masuk inisiliasi	tidak bisa masuk inisiliasi
...	...	...
8353	Semoga Tambah Baik,Mudah Dalam Input Dan Share...	semoga tambah baik,mudah dalam input dan share...
8354	Cara masuk nya bagaimana? Pakai email kok gag ...	cara masuk nya bagaimana? pakai email kok gag ...
8355	Mbuh	mbuh
8356	Semoga tidak seperti sirekap 2020 yang saat di...	semoga tidak seperti sirekap 2020 yang saat di...
8357	Semoga dengan adanya aplikasi ini pemilu akan ...	semoga dengan adanya aplikasi ini pemilu akan ...

Gambar 3.7 Hasil Lower Case

Terlihat pada gambar diatas, *komentar\_lowercase* berisi data ulasan yang telah dilakukan proses *lower\_case* yang mengubah huruf kapital menjadi huruf kecil.

#### 2. Remove Unnecessary Character

Atribut spesifik dan semua karakter alphanumeric atau karakter yang tidak diperlukan pada data ulasan seperti simbol, emoji, karakter yang berulang, tanda baca, dan karakter spesial lainnya dihapus menggunakan *remove\_unnecessary\_character*. Hasil dari *Remove Unnecessary Character* dapat dilihat pada gambar 3.8.

komentar_lowercase \		komentar_remove_char	
0	mantap	0	mantap
1	sudah tau pemakainya masyarakat biasa malah sp...	1	sudah tau pemakainya masyarakat biasa malah sp...
2	apk nya nggk bisa buat log in	2	apk nya nggk bisa buat log in
3	susah masuk dih	3	susah masuk dih
4	tidak bisa masuk inisiliasi	4	tidak bisa masuk inisiliasi
...	...	...	...
8353	semoga tambah baik,mudah dalam input dan share...	8353	semoga tambah baik mudah dalam input dan share...
8354	cara masuk nya bagaimana? pakai email kok gag ...	8354	cara masuk nya bagaimana pakai email kok gag b...
8355	mbuh	8355	mbuh
8356	semoga tidak seperti sirekap 2020 yang saat di...	8356	semoga tidak seperti sirekap 2020 yang saat di...
8357	semoga dengan adanya aplikasi ini pemilu akan ...	8357	semoga dengan adanya aplikasi ini pemilu akan ...

**Gambar 3.8** Hasil *Remove Unnecessary Character*

*komentar\_remove\_char* adalah data ulasan yang telah dilakukan proses *Remove Unnecessary Character* pada data *komentar\_lowercase*. Contoh perubahan yang terlihat pada *komentar\_remove\_char* adalah tanda baca seperti “,” dan “?” yang sebelumnya terdapat di *komentar\_lowercase* pada baris 8353 dan baris 8354 berhasil dihapus.

### 3. *Spell Checker*

Hasil dari proses pembuatan kamus bernama “*kamus\_tidak\_baku*” yang digunakan sebagai referensi proses *Spell Checker* ditampilkan pada gambar berikut.

	Original	Replacement
0	Kata_Tidak_Baku	Kata_Baku
1	0	0
2	00	0
3	000d	000d
4	1	1
...	...	...
4404	hahahaha	ketawa
4405	oalah	ambigu
4406	ayok	ayo
4407	system	sistem
4408	geogle	google
[4409 rows x 2 columns]		

**Gambar 3.9** Kamus Tidak Baku

Pada gambar 3.9, terlihat isi dari “*kamus\_tidak\_baku*” yang terdiri dari 4409 baris dan 2 kolom. Kolom Original merupakan kata – kata tidak baku atau kata yang tidak terdeteksi oleh file “4 lebih list kata KBBI.txt”. Sedangkan kolom Replacment merupakan kata – kata yang telah dibakukan.

Selanjutnya, dilakukan *Spell Checker* untuk memeriksa ejaan pada data ulasan dan mengubah ejaan menjadi lebih baik. Hasilnya dapat dilihat pada gambar 3.10.

komentar_remove_char \		komentar_spellchecker	
0	mantap	0	mantap
1	sudah tau pemakainya masyarakat biasa malah sp...	1	sudah tahu pemakainya masyarakat biasa malah s...
2	apk nya ngggk bisa buat log in	2	aplikasi nya tidak bisa buat masuk ini
3	susah masuk dih	3	susah masuk dih
4	tidak bisa masuk inisiliasi	4	tidak bisa masuk inisialisasi
...	...	...	...
8353	semoga tambah baik mudah dalam input dan share...	8353	semoga tambah baik mudah dalam memasukkan dan ...
8354	cara masuk nya bagaimana pakai email kok gag b...	8354	cara masuk nya bagaimana pakai email kok tidak...
8355	mbuh	8355	tidak tahu
8356	semoga tidak seperti sirekap 2020 yang saat di...	8356	semoga tidak seperti sirekap 2020 yang saat di...
8357	semoga dengan adanya aplikasi ini pemilu akan ...	8357	semoga dengan adanya aplikasi ini pemilu akan ...

**Gambar 3.10** Hasil *Spell Checker*

Proses *Spell Checker* menggunakan “kamus\_tidak\_baku” yang telah diinput. Seperti yang ditunjukkan Gambar 3.10, *komentar\_spellchecker* merupakan data ulasan setelah dilakukan *Spell Checker* seperti "apk" menjadi "aplikasi", "inisiliasi" menjadi "inisialisasi", dan perubahan lainnya.

#### 4. *Stemming*

*Stemming* menggunakan library *Sastrawi* untuk mengubah kata-kata yang ada pada data ulasan menjadi kata dasar. Hasil proses ini dapat dilihat pada gambar berikut.

komentar_spellchecker		komentar_stemming	
0	mantap	0	mantap
1	sudah tahu pemakainya masyarakat biasa malah s...	1	sudah tahu maka masyarakat biasa malah spesifi...
2	aplikasi nya tidak bisa buat masuk ini	2	aplikasi nya tidak bisa buat masuk ini
3	susah masuk dih	3	susah masuk dih
4	tidak bisa masuk inisialisasi	4	tidak bisa masuk inisial
...	...	...	...
8353	semoga tambah baik mudah dalam memasukkan dan ...	8353	moga tambah baik mudah dalam masuk dan bagi data
8354	cara masuk nya bagaimana pakai email kok tidak...	8354	cara masuk nya bagaimana pakai email kok tidak...
8355	tidak tahu	8355	tidak tahu
8356	semoga tidak seperti sirekap 2020 yang saat di...	8356	moga tidak seperti sirekap 2020 yang saat guna...
8357	semoga dengan adanya aplikasi ini pemilu akan ...	8357	moga dengan ada aplikasi ini milu akan lebih b...

**Gambar 3.11** Hasil *Stemming*

Terlihat pada gambar 3.11, *komentar\_stemming* adalah data ulasan setelah dilakukan proses *Stemming* dimana kata-kata pada data ulasan diubah menjadi kata dasar seperti kata “semoga” mejadi “moga”, “memasukkan” menjadi “masuk”, “inisialisasi” menjadi “inisial”, dan perubahannya lainnya.

#### 5. Hapus Data Kosong

Setelah data ulasan melalui tahap *stemming*, dilakukan cek data kosong ‘NaN’ untuk melihat berapa data yang memiliki nilai ‘NaN’ yang terdapat pada *komentar\_stemming*. Didapatkan sebanyak 60 data yang memilki nilai ‘NaN’ pada *komentar\_stemming*. Selanjutnya, ‘NaN’ dihapus untuk menghilangkan data kosong tersebut. Hasilnya dapat dilihat pada gambar 3.12.

ID	Nama	Rating	Waktu	Komentar	Like	komentar_lowerCase	komentar_remove_char	komentar_spellchecker	komentar_stemming
0	d0001	Nency Miranda	5	05/02/2024 14:07	Mantap	0.0	mantap	mantap	mantap
1	d0002	ichal sabiel	1	05/02/2024 14:06	Sudah tau pemakainya masyarakat biasa Malah sp...	0.0	sudah tau pemakainya masyarakat biasa malah sp...	sudah tau pemakainya masyarakat biasa malah sp...	sudah tahu maka masyarakat biasa malah spesifik...
2	d0003	Evandra Aditya	1	05/02/2024 14:06	Apk nya ngkk bisa buat log in	0.0	apk nya ngkk bisa buat log in	apk nya ngkk bisa buat log in	aplikasi nya tidak bisa buat masuk ini
3	d0004	Ganesh insann	1	05/02/2024 14:05	Susah masuk dih	0.0	susah masuk dih	susah masuk dih	susah masuk dih
4	d0005	Yohana Frediana	1	05/02/2024 14:04	Tidak bisa masuk inisiliasi	0.0	tidak bisa masuk inisiliasi	tidak bisa masuk inisiliasi	tidak bisa masuk inisial
...	...	...	...	...	...	...	...	...	...
8353	d8354	Tn A Muntaha	5	23/01/2024 09:05	Semoga Tambah Baik,Mudah Dalam Input Dan Share...	21.0	semoga tambah baik,mudah dalam input dan share...	semoga tambah baik mudah dalam memasukkan dan ...	moga tambah baik mudah dalam masuk dan bagi data
8354	d8355	Adhi Nugroho	1	23/01/2024 08:51	Cara masuk nya bagaimana? Pakai email kok gag ...	448.0	cara masuk nya bagaimana? pakai email kok gag ...	cara masuk nya bagaimana pakai email kok tidak...	cara masuk nya bagaimana pakai email kok tidak...
8355	d8356	TATANG RUSDIANA	5	23/01/2024 07:03	Mbuh	6.0	mbuh	mbuh	tidak tahu
8356	d8357	vie	5	23/01/2024 06:50	Semoga tidak seperti sirekap 2020 yang saat di...	42.0	semoga tidak seperti sirekap 2020 yang saat di...	semoga tidak seperti sirekap 2020 yang saat di...	moga tidak seperti sirekap 2020 yang saat guna...
8357	d8358	Ahmad ferdiansyah	5	23/01/2024 04:26	Semoga dengan adanya aplikasi ini pemilu akan ...	196.0	semoga dengan adanya aplikasi ini pemilu akan ...	semoga dengan adanya aplikasi ini pemilu akan ...	moga dengan ada aplikasi ini milu akan lebih b...

8298 rows x 10 columns

Gambar 3.12 Hasil Hapus Data Kosong

Dapat dilihat pada gambar 3.12, data ulasan menjadi 8298 baris setelah dilakukan proses hapus data kosong. Ini menandakan bahwa penghapusan data kosong berhasil dilakukan. Perbandingan data ulasan sebelum dan setelah dilakukan tahap *Pra-proses* dapat dilihat pada gambar berikut.

0		Mantap
1	Sudah tau pemakainya masyarakat biasa Malah sp...	
2	Apk nya ngkk bisa buat log in	
3	Susah masuk dih	
4	Tidak bisa masuk inisiliasi	
...	...	
8353	Semoga Tambah Baik,Mudah Dalam Input Dan Share...	
8354	Cara masuk nya bagaimana? Pakai email kok gag ...	
8355	Mbuh	
8356	Semoga tidak seperti sirekap 2020 yang saat di...	
8357	Semoga dengan adanya aplikasi ini pemilu akan ...	
Name: Komentar, Length: 8358, dtype: object		
0		mantap
1	sudah tahu maka masyarakat biasa malah spesifik...	
2	aplikasi nya tidak bisa buat masuk ini	
3	susah masuk dih	
4	tidak bisa masuk inisial	
...	...	
8353	moga tambah baik mudah dalam masuk dan bagi data	
8354	cara masuk nya bagaimana pakai email kok tidak...	
8355	tidak tahu	
8356	moga tidak seperti sirekap 2020 yang saat guna...	
8357	moga dengan ada aplikasi ini milu akan lebih b...	
Name: komentar stemming, Length: 8298, dtype: object		

Gambar 3.13 Hasil *Pra-proses*

Dapat dilihat pada gambar 3.13, gambar disisi kiri adalah data ulasan sebelum dilakukan Tahap *Pra-proses*, Sedangkan gambar disisi kiri merupakan data ulasan yang telah melalui tahap *Pra-proses*.

### 3.1.3 Wordnet

Tahapan selanjutnya setelah *Pra-proses* adalah melakukan tahapan analisis sentimen menggunakan metode *Wordnet*. Sebelum melakukan analisis sentimen menggunakan *Wordnet*, data ulasan diterjemahkan ke dalam bahasa Inggris. Hasil penerjemahan dapat dilihat pada gambar 3.14.

komentar_stemming \		Translated	
0	mantap	0	Excellent
1	sudah tahu maka masyarakat biasa malah spesifik...	1	You already know that ordinary people even hav...
2	aplikasi tidak bisa buat masuk in	2	the application can't do this entry
3	susah masuk dih	3	it's hard to get in
4	tidak bisa masuk inisial	4	Can't enter initials
...	...	...	...
8297	moga tambah baik mudah dalam masuk dan share data	8293	Hopefully it will be easier to enter and share...
8298	cara masuk bagaimana pakai email kok tidak bis...	8294	How come you can log in using email? How come ...
8299	tidak tahu	8295	Don't know
8300	moga tidak seperti sirekap 2020 yang saat guna...	8296	Hopefully it won't be like the 2020 recap wher...
8301	moga dengan ada aplikasi ini milu akan lebih b...	8297	Hopefully with this application Milu will be m...
		[8298 rows x 2 columns]	
		Waktu proses: 6387.12087059021 detik	

**Gambar 3.14** Hasil *Deep-translator*

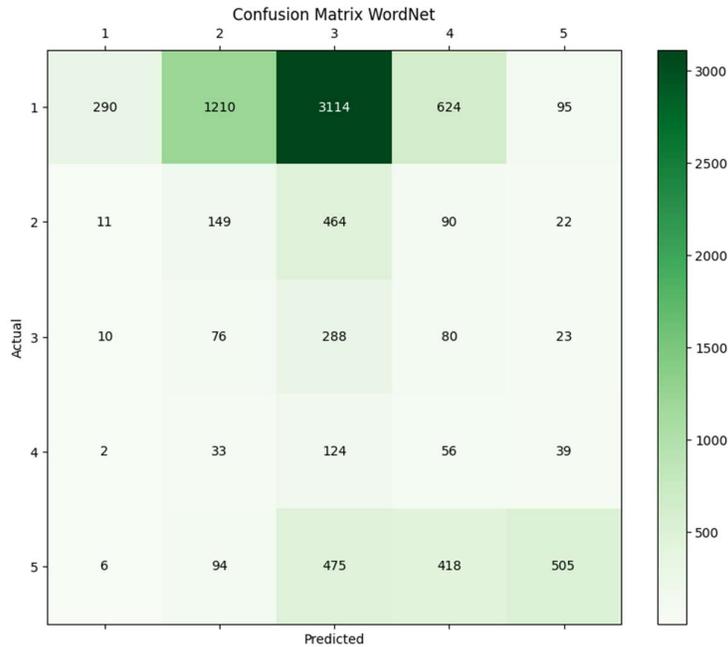
Terlihat pada gambar 3.14, *Translated* merupakan data ulasan dari *komentar\_stemming* yang telah melalui tahap penerjemahan dari bahasa Indonesia ke bahasa Inggris. Proses ini membutuhkan waktu sekitar 107 menit untuk menerjemahkan data ulasan. Selanjutnya, mencari nilai polaritas menggunakan *TextBlob* dengan memanfaatkan *Wordnet* dan mengklasifikasinya dalam 5 kelas peringkat berdasarkan tabel 2.2. Hasilnya dapat dilihat pada gambar berikut.

ID	Rating	komentar_stemming	Translated	Score_Wordnet	Sentiment_Score	
0	d0001	5	mantap	Excellent	5 star	1.000000
1	d0002	1	sudah tahu maka masyarakat biasa malah spesifik...	You already know that ordinary people even hav...	3 star	-0.045000
2	d0003	1	aplikasi nya tidak bisa buat masuk ini	the application can't do this entry	3 star	0.000000
3	d0004	1	susah masuk dih	it's hard to get in	2 star	-0.291667
4	d0005	1	tidak bisa masuk inisial	Can't enter initials	3 star	0.000000

**Gambar 3.15** Hasil *Wordnet*

Terlihat pada gambar 3.15, setiap komentar memiliki nilai polaritas masing – masing yang ada pada kolom “*Sentiment\_Score*” dan kemudian diklasifikasikan ke dalam 5 kelas peringkat yang ditampilkan pada kolom “*Score\_Wordnet*”. Seperti pada komentar “mantap” yang memiliki nilai polaritas 1 dan masuk kedalam kelas peringkat 5 *star* karena memiliki nilai lebih dari 0,5. Terlihat juga bahwa *rating* yang diberikan pengguna dengan kelas peringkat berdasarkan nilai polaritas ada yang tidak sama. Contohnya pada komentar “tidak bisa masuk inisial” yang diberikan *rating* 1 oleh pengguna namun memiliki nilai polaritas 0 dan masuk kedalam kelas peringkat 3 *star* karena memiliki nilai yang kurang dari 0,2 dan lebih dari -0,2.

*Wordnet* dievaluasi menggunakan *Confusion Matrix* untuk mencari nilai *F1-Score*. *Confusion Matrix* terdiri dari *Actual* dan *Predicted*. *Actual* merupakan label kelas sebenarnya dari data ulasan, sedangkan *Predicted* merupakan kelas yang diklasifikasi oleh model pada data ulasan. Hasil *Confusion Matrix* pada klasifikasi menggunakan *Wordnet* dapat dilihat pada gambar 3.16.



**Gambar 3.16** *Confusion Matrix* dengan *Wordnet*

Dapat dilihat pada gambar 3.16, warna hijau yang lebih intens menandakan data test yang paling banyak diklasifikasi, sedangkan warna hijau yang lebih pudar menandakan data test yang paling sedikit diklasifikasi. data test yang paling banyak diklasifikasi adalah 3114 data test yang seharusnya dilabelkan pada *rating* 1, namun diklasifikasikan kedalam kelas peringkat 3 *star*. Sedangkan data test yang paling sedikit diklasifikasi adalah 2 data test yang seharusnya dilabelkan pada *rating* 4, namun diklasifikasikan kedalam kelas peringkat 1 *star*.

Dari gambar 3.16 dapat disimpulkan bahwa, dari 5361 data pada *rating* 1, hanya 298 data test yang berhasil diklasifikasi benar. 736 data pada *rating* 2, hanya 149 data test yang berhasil diklasifikasi benar. 478 data pada *rating* 3, yang berhasil diklasifikasi benar hanya 288 data test. 255 data pada *rating* 4, hanya 56 data test yang berhasil diklasifikasi benar. Sedangkan, 1528 data pada *rating* 5, hanya 505 data test yang berhasil diklasifikasi benar.

Hasil dari *Confusion Matrix*, dilakukan perhitungan performa dan didapatkan hasil *F1-Score* yaitu 17.70%. Proses ini membutuhkan waktu sekitar 1 detik untuk memproses evaluasi *F1-Score* pada model *Wordnet*. Nilai tersebut menunjukkan hasil evaluasi *F1-Score* yang didapatkan dari *Wordnet* dalam menganalisis sentimen pada ulasan aplikasi SIREKAP 2024.

### 3.1.4 Klasifikasi *Decision Tree* CART

Dalam ekstraksi fitur TF-IDF, data dibagi menjadi 2 bagian, yaitu X (Gambar 3.17) sebagai kolom komentar\_ *stemming* dan y (Gambar 3.18) sebagai kolom *rating*. Berikut gambar dari kolom X dan y.

```

1 X
✓ 0.0s
0
1 mantap
2 sudah tahu maka masyarakat biasa malah spesifi...
3 aplikasi nya tidak bisa buat masuk ini
4 susah masuk dih
5 tidak bisa masuk inisial
6 ...
7
8293 moga tambah baik mudah dalam masuk dan bagi data
8294 cara masuk nya bagaimana pakai email kok tidak...
8295 tidak tahu
8296 moga tidak seperti sirekap 2020 yang saat guna...
8297 moga dengan ada aplikasi ini milu akan lebih b...
Name: komentar_stemming, Length: 8298, dtype: object

```

**Gambar 3.17** komentar\_stemming

```

1 y
✓ 0.0s
0 5
1 1
2 1
3 1
4 1
5 ..
6
8293 5
8294 1
8295 5
8296 5
8297 5
Name: Rating, Length: 8298, dtype: int64

```

**Gambar 3.18** Rating

Dapat dilihat pada gambar 3.17 dan gambar 3.18, data telah dibagi menjadi dua bagian yang ditandai dengan variabel X sebagai data komentar dan variabel y sebagai data *rating* dari data ulasan. Selanjutnya dilakukan perhitungan TF-IDF pada variabel X untuk mengubah . Hasil perhitungan TF-IDF dapat dilihat pada gambar 3.19.

(0, 1376)	1.0
(1, 259)	0.6792391025684885
(1, 887)	0.10383584445945196
(1, 206)	0.07652109295763325
(1, 1614)	0.14506596822529783
(1, 2410)	0.13373232476819363
(1, 2320)	0.27802569670255944
(1, 825)	0.16356998608982817
(1, 2155)	0.27307728845508056
(1, 1359)	0.13781584933616398
(1, 380)	0.25004926716628667
(1, 1394)	0.29022619285877926
(1, 1348)	0.2746602935493359
(1, 2207)	0.22574459106729391
(1, 2173)	0.11088639682045809
(2, 1393)	0.3060772188862647
(2, 439)	0.5368957909043959
(2, 400)	0.3187697434660648
(2, 2307)	0.287055929249031
(2, 1573)	0.4683618676203409
(2, 887)	0.3729945196202908
(2, 206)	0.27487567956070985
(3, 600)	0.9096105750431945
(3, 2193)	0.34717119421206155
(3, 1393)	0.22821210239371564
...	
(8297, 1061)	0.26376030223663394
(8297, 887)	0.15294509486441463
(8297, 206)	0.11271180855185466
(8297, 1394)	0.4274889160867539

**Gambar 3.19** Hasil TF-IDF

Hasil pada gambar 3.19 menunjukkan nilai yang telah dihitung menggunakan TF-IDF. Contohnya (0, 1376), yang menunjukkan pada indeks 0 terdapat kata dengan term 1376 yang berisi kata ‘mantap’ dengan nilai TF-IDF yaitu 1,0 dan seterusnya.

Setelah menghitung ekstraksi fitur TF-IDF, performa model dievaluasi dengan membagi data menjadi 10 *fold*. *Fold* 1 sampai *fold* 8 membagi data menjadi 7468 sampel sebagai data train dan data 830 sampel sebagai data test. Sedangkan, *fold* 9 dan *fold* 10 membagi data menjadi 7469 sampel sebagai data train dan 829 sampel sebagai data test. Berikut pembagian indeks data yang digunakan pada *fold* 1.

**Tabel 3.2** *Fold* 1 Cross-Validation

Train Index	Test Index
[830, 831, 832, ..., 8295, 8296, 8297]	[0, 1, 2, ..., 827, 828, 829]

Tabel 3.2 menunjukkan *index* yang digunakan dalam melatih dan menguji model pada *fold* 1. *Train Index* merupakan sampel yang digunakan pada data train *fold* 1, yaitu *index* 830 sampai 8279. Sedangkan *Test Index* merupakan sampel yang digunakan pada data test *fold* 1, yaitu *index* 0 sampai 829. Selanjutnya, pembagian indeks data yang digunakan pada *fold* 10 yang memiliki perbedaan penggunaan sampel data dengan *fold* 1.

**Tabel 3.3** *Fold 10 Cross-Validation*

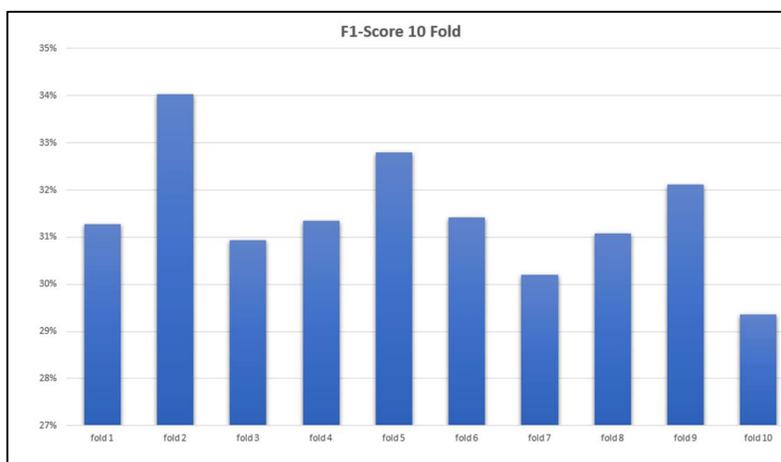
<i>Train Index</i>	<i>Test Index</i>
[0, 1, 2, ..., 7466, 7467, 7468]	[7469, 7470, 7471, ..., 8295, 8296, 8297]

Tabel 3.3 menunjukkan *index* yang digunakan dalam melatih dan menguji model pada *fold* 10. *Train Index* merupakan sampel yang digunakan pada data train *fold* 10, yaitu *index* 0 sampai 7468. Sedangkan *Test Index* merupakan sampel yang digunakan pada data test *fold* 10, yaitu *index* 7469 sampai 8297. Setelah itu, dilakukan perhitungan performa menggunakan *F1-Score* pada masing – masing *fold*. Hasilnya dapat dilihat pada tabel 3.4.

**Tabel 3.4** Hasil *F1-Score* pada Setiap *Fold*

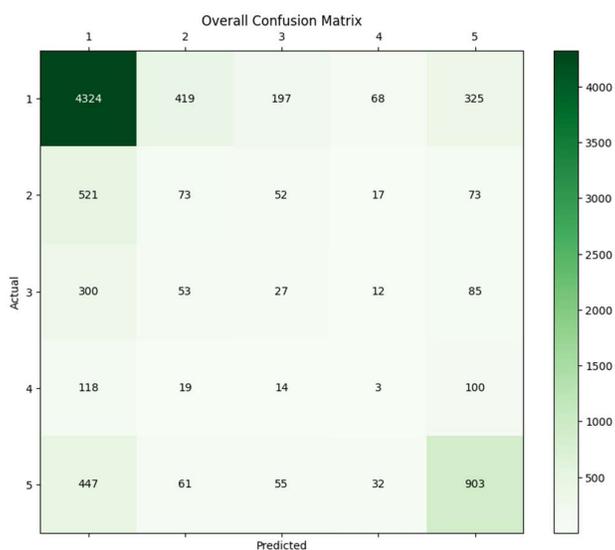
<i>Fold</i>	<i>F1-Score</i>
1	31.27%
2	34.03%
3	30.94%
4	31.35%
5	32.79%
6	31.42%
7	30.19%
8	31.07%
9	32.11%
10	29.36%

Tabel diatas menunjukkan hasil *F1-Score* dari masing – masing *fold* menggunakan *Cross-Validation* 10 *fold*. dapat dilihat *fold* yang memiliki nilai *F1-Score* tertinggi adalah *fold* 2 dengan nilai *F1-Score* sebesar 34.03% dan *fold* yang memiliki nilai *F1-Score* terendah adalah *fold* 10 dengan nilai *F1-Score* sebesar 29.36%. Perbandingan nilai *F1-Score* pada setiap *fold* dapat dilihat pada gambar 3.20.



**Gambar 3.20** *F1-Score 10 Fold*

Gambar 3.20 merupakan grafik performa pada setiap *fold* yang dihitung menggunakan *F1-Score*. Dari 10 *fold* tersebut, *fold 2* memiliki performa tertinggi, sementara *fold 10* memiliki performa terendah dibandingkan *fold* lainnya. Ini menunjukkan bahwa pembagian data train dan data test yang menghasilkan performa tertinggi berada pada *fold 2*, sedangkan pembagian data dengan performa terendah berada pada *fold 10* berdasarkan hasil *K-fold Cross-Validation*  $K=10$ . Selanjutnya nilai dari setiap *fold* digunakan untuk menghitung hasil keseluruhan nilai *fold*. Berikut hasil keseluruhan nilai *fold* yang divisualisasikan dengan *Confusion Matrix*.



**Gambar 3.21** *Confusion Matrix Overall* dengan *Decision Tree CART*

Gambar 3.21 merupakan tampilan *Confusion Matrix* dari hasil keseluruhan tahapan klasifikasi dari *Decision Tree* dengan algoritma *CART*. Terlihat data ulasan yang paling banyak benar diklasifikasikan oleh *Decision Tree* dengan algoritma *CART* ada pada ulasan dengan *rating 1* dan *5* yaitu sebanyak 4324 data untuk *rating 1* dan 903 data untuk *rating 5*. Dari *Confusion Matrix* tersebut, didapatkan hasil dari *Macro F1-Score*, yaitu 31.62%. Waktu yang dibutuhkan untuk proses ini adalah 32 detik. Hal ini menunjukkan hasil evaluasi *F1-Score* menggunakan metode *Decision Tree* dengan algoritma *CART* dalam menganalisis sentimen ulasan aplikasi SIREKAP 2024.

## 3.2 Pembahasan

### 3.2.1 Pengambilan dan Analisis Data

Penelitian ini menggunakan data ulasan dari pengguna aplikasi SIREKAP 2024 di *Google Playstore* yang diambil pada tanggal 6 februari 2024. Setelah dilakukan analisis data, data pada penelitian ini merupakan data tidak seimbang dimana jumlah kemunculan data pada *rating* 1 dan 5 jauh lebih banyak dibandingkan *rating* 2, 3, dan 4. Menurut (Provost, 2000; Utami, 2022), data yang tidak seimbang menyebabkan kesalahan klasifikasi pada kelas minoritas, karena data cenderung lebih mendukung kelas mayoritas.

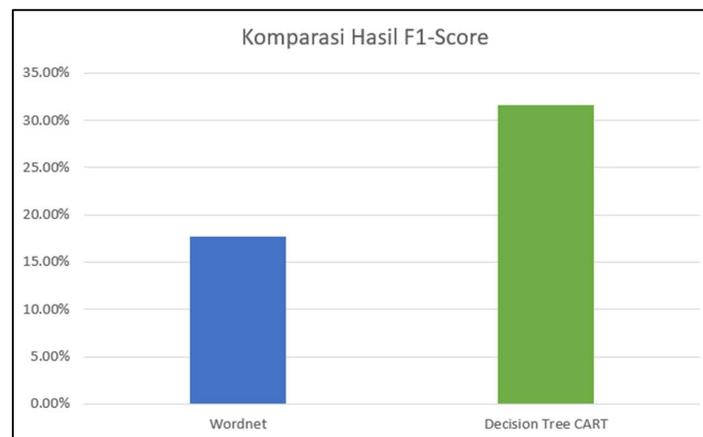
### 3.2.2 Analisis Hasil *Pra-proses*

Pada hasil tahap *Pra-proses*, tahap ini berhasil menghapus karakter non-alfanumerik, membenarkan sebagian besar kesalahan ejaan kata, serta membersihkan data kosong yang pada data ulasan. Namun, masih terdapat masalah pada hasil *Pra-proses* dipenelitian ini. Dapat dilihat pada gambar 3.13, terdapat kata “dih” yang tidak memiliki arti atau disebut dengan kata *slank* yang seharusnya dihapus. Selain itu, terdapat juga kesalahan pengubahan kata imbuhan menjadi kata dasar, seperti kata “pemainnya” menjadi “maka” dimana dua kata tersebut memiliki makna yang berbeda. Hal ini dapat mempengaruhi hasil klasifikasi *Wordnet* dan *Decision Tree* CART karena menggunakan data ulasan yang telah melalui tahap *Pra-proses*.

### 3.2.3 Analisis Komparasi Model Klasifikasi *Wordnet* dan CART

Model *Wordnet* dan *Decision Tree* dengan algoritma CART memiliki kelebihan dan kekurangannya masing-masing dalam mengklasifikasi sentimen pada data ulasan di penelitian ini. Pada model *Wordnet*, sebenarnya klasifikasi dan evaluasi *F1-Score* yang dilakukan lebih unggul dari segi kecepatan dibandingkan *Decision Tree* dengan algoritma CART. Namun, model *Wordnet* memerlukan waktu yang lama dalam menerjemahkan data ulasan dari bahasa Indonesia ke bahasa Inggris sehingga model *Decision Tree* dengan algoritma CART menjadi lebih cepat.

Selain itu, berdasarkan hasil evaluasi *F1-Score* yang didapatkan dari masing masing model, performa yang dihasilkan model *Decision Tree* dengan algoritma CART lebih tinggi dibandingkan model *Wordnet*. perbandingannya dapat dilihat pada gambar 3.22 dan tabel 3.5.



**Gambar 3.22** Komparasi Hasil *F1-Score*

**Tabel 3.5** Hasil *F1-Score* Kedua Metode

Metode	F1 - Score
<i>Wordnet</i>	17.70%
<i>Decision Tree</i> CART	31.62%

Berdasarkan tabel 3.5 dan gambar 3.22, dapat dilihat bahwa model *Decision Tree* dengan algoritma CART memiliki performa yang lebih tinggi dengan nilai *F1-Score* 31.62%, dibandingkan dengan model *Wordnet* yang memiliki nilai *F1-Score* 17.70% dalam menganalisis sentimen pada ulasan aplikasi SIREKAP 2024.

Rendahnya performa yang dihasilkan oleh hasil evaluasi *F1-Score* pada model *Wordnet* dibandingkan model *Decision Tree* CART, menandakan klasifikasi pada model *Wordnet* kurang baik dibandingkan model *Decision Tree* CART. Secara keseluruhan, model *Wordnet* banyak melakukan kesalahan dalam mengklasifikasi data ulasan, terutama pada data test yang seharusnya berada pada *rating* 1 namun dilabelkan ke dalam kelas peringkat 3 *star*, yaitu sebanyak 3114 data. Sedangkan data ulasan yang diklasifikasi benar pada *Wordnet* berada pada data test yang memiliki *rating* 5 dan dilabelkan ke dalam kelas peringkat 5 *star*, yaitu 505 data. Ini menandakan bahwa klasifikasi yang dilakukan pada model *Wordnet* lebih banyak yang salah dibandingkan yang benar.

Selanjutnya pada model *Decision Tree* CART, data ulasan yang diklasifikasi benar lebih banyak dibandingkan yang klasifikasi salah. Ini ditunjukkan pada gambar 3.21, dimana data test yang paling banyak diklasifikasi benar berada pada data test yang seharusnya memiliki *rating* 1 dan diklasifikasi ke dalam *rating* 1, yaitu sebanyak 4324 data. Sedangkan data ulasan yang paling banyak diklasifikasi salah berada pada data test yang seharusnya memiliki *rating* 5 namun diklasifikasi ke dalam *rating* 1, yaitu sebanyak 447 data. Hal ini membuktikan bahwa klasifikasi yang dilakukan pada model *Decision Tree* CART lebih banyak yang benar dibandingkan yang salah.

Hasil klasifikasi yang didapatkan dari *Confussion Matrix* pada masing-masing model menunjukkan bahwa, klasifikasi ulasan yang dilakukan oleh *Decision Tree* dengan algoritma CART lebih unggul dibandingkan model *Wordnet*. Hal ini menjadi alasan mengapa performa *F1-Score* pada model *Wordnet* lebih rendah dibandingkan performa *F1-Score* pada model *Decision Tree* dengan algoritma CART.

Hasil penelitian ini menunjukkan bahwa model *Wordnet* dan *Decision Tree* dengan algoritma CART terbukti dapat digunakan dalam menganalisis sentimen pada ulasan aplikasi SIREKAP 2024 yang memiliki data yang tidak seimbang. Setelah dianalisis komparasi dari kedua model, metode *Decision Tree* dengan algoritma CART menjadi metode yang lebih baik dibandingkan dengan metode *Wordnet* dalam mengklasifikasi sentimen ulasan pada aplikasi SIREKAP 2024.

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Penerapan metode *Decision Tree* dengan algoritma CART dan metode *Wordnet* dapat digunakan dalam analisis sentimen ulasan aplikasi SIREKAP 2024. *Decision Tree* dengan algoritma CART memiliki hasil evaluasi *F1-Score* dengan nilai 31.62%, sedangkan *Wordnet* memiliki hasil evaluasi *F1-Score* dengan nilai 17.70%. Ini membuktikan bahwa *Decision Tree* dengan algoritma CART memiliki performa yang lebih baik dalam menganalisis sentimen ulasan aplikasi SIREKAP 2024 dibandingkan *Wordnet*.

Walaupun kedua model klasifikasi pada penelitian ini dapat digunakan dalam analisis sentimen ulasan aplikasi SIREKAP 2024, performa yang didapatkan dari kedua model masih kurang baik. Hal ini disebabkan karena data yang tidak seimbang dan tahap *Pra-proses* yang kurang maksimal terhadap data ulasan aplikasi SIREKAP 2024. Hasil penelitian ini diharapkan mampu membantu pengembang aplikasi dalam meningkatkan kinerja aplikasi SIREKAP 2024 dan diharapkan dapat menjadi referensi untuk penelitian selanjutnya.

#### **4.2 Saran**

Terdapat saran untuk penelitian selanjutnya, diantaranya:

- Untuk menangani data yang tidak seimbang pada data ulasan, penelitian selanjutnya dapat melakukan tahapan seperti *Oversampling* dan *Undersampling* agar model klasifikasi dapat menghindari kesalahan klasifikasi terhadap data ulasan.
- Pada penelitian selanjutnya, diharapkan memaksimalkan tahap *Pra-proses* dengan lebih memperhatikan lagi setiap proses-proses pada *Pra-proses* agar data ulasan dapat digunakan lebih baik lagi.

## DAFTAR PUSTAKA

- Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). ScienceDirect ScienceDirect The Impact of Features Extraction on the Sentiment Analysis. *Procedia Computer Science*, 152, 341–348. <https://doi.org/10.1016/j.procs.2019.05.008>
- Bagui, S., & Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data*, 8(1). <https://doi.org/10.1186/s40537-020-00390-x>
- Bahasa, T. P. K. P. (2008). *Kamus Bahasa Indonesia*. Pusat Bahasa.
- Bayu Setiawan, D., & Sulisty Nugroho, Y. (2023). Perbandingan Performa Algoritma Decision Tree untuk Klasifikasi Penerima Beasiswa Bank Indonesia. *Indonesian Journal of Computer Science*, 12(4), 2108–2123. <https://doi.org/10.33022/ijcs.v12i4.3339>
- Cholis, D. N., & Ulinuha, N. (2023). An Ensemble Voting Approach for Dropout Student Classification Using Decision Tree C4.5, K-Nearest Neighbor and Backpropagation. *Indonesian Journal of Artificial Intelligence and Data Mining*, 6(1), 107. <https://doi.org/10.24014/ijaidm.v6i1.23412>
- Dangi, D., Dixit, D. K., & Bhagat, A. (2022). Sentiment analysis of COVID-19 social media data through machine learning. *Multimedia Tools and Applications*, 81(29), 42261–42283. <https://doi.org/10.1007/s11042-022-13492-w>
- Darmawan, D. (2023). Pengaruh Promosi, Ulasan Produk, Dan Persepsi Harga Terhadap Pemilihan Toko Online. *Jurnal Industri Kreatif Dan Kewirausahaan*, 6(1), 95–105. <https://doi.org/10.36441/kewirausahaan.v6i1.1382>
- Fatihin, A. (2022). *Analisis Sentimen Terhadap Ulasan Aplikasi Mobile Menggunakan Metode Support Vector Machine (Svm) Dan Pendekatan Lexicon Based*. 103.
- Fjærli, M., & Larsen, J. E. (2022). Sentiment Analysis of Nasdaq News. *Master's Thesis in Economics and Business Administration*, May. <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3029450%0Ahttps://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/3029450/no.ntnu%3Ainspera%3A111624424%3A114108169.zip?sequence=2>
- Husin, N. (2023). Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN). *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, 7(1), 75–84. <https://doi.org/10.55886/infokom.v7i1.608>
- Kasim, Y. U. (2024). *Sirekap KPU Pemilu 2024: Link, Fungsi, dan Mekanismenya*. Detik.Com. <https://www.detik.com/sulsel/berita/d-7195718/sirekap-kpu-pemilu-2024-link-fungsi-dan-mekanismenya>
- Keerthi Kumar, H. M., Harish, B. S., & Darshan, H. K. (2019). Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(5), 109–114. <https://doi.org/10.9781/ijimai.2018.12.005>
- Kurniawati, F., & Arianto, D. B. (2019). *Analisis Implementasi Seleksi Fitur Pada Klasifikasi Diabetes dengan Metode Corellation Matrix dan Algoritma Logistic Regression*. 4221(2020), 157–164.
- Lestari, U. P., & Putra, A. R. (2022). Brand Switching Behavior on Smartphone Product. *Journal of Science, Technology and Society (SICO)*, 3(2).
- Munir, A., Atika, E. P., & Indraswari, A. D. (2022). Analisis Sentimen pada review hotel menggunakan metode pembobotan dan klasifikasi. *Jnanaloka*, 3(1), 33–38. <https://doi.org/10.36802/jnanaloka.2022.v3-no1-33-38>
- Nengsih, W., Zein, M. M., & Hayati, N. (2021). Coarse-Grained Sentiment Analysis Berbasis Natural

- Language Processing – Ulasan Hotel. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 10(1), 41–48. <https://doi.org/10.22146/jnteti.v10i1.548>
- Nurainun, N., Haerani, E., Syafria, F., & Oktavia, L. (2023). Penerapan Algoritma Naïve Bayes Classifier Dalam Klasifikasi Status Gizi Balita dengan Pengujian K-Fold Cross Validation. *Journal of Computer System and Informatics (JoSYC)*, 4(3), 578–586. <https://doi.org/10.47065/josyc.v4i3.3414>
- Nurfauzan, A., & Maharani, W. (2021). Klasifikasi Emosi Pada Pengguna Twitter Menggunakan Metode Klasifikasi Decision Tree. *Universitas Telkom*.
- Nurnawati, E. K., Sholeh, M., Ariyan, R. Y., & Almuntaha, E. (2023). Comparison of Decision Tree and Naïve Bayes Algorithms in Classification Models To Determine Lecturer Performance Using K Fold Cross. *Jurnal TAM (Technology Acceptance Model) Volume*, 14(2), 152–157.
- Pratomo, S. A., Al Faraby, S., & Purbolaksono, M. D. (2021). Analisis Sentimen Pengaruh Kombinasi Ekstraksi Fitur TF-IDF dan Lexicon Pada Ulasan Film Menggunakan Metode KNN. *E-Proceeding of Engineering*, 8(5), 10116–10126. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15726>
- Provost, F. (2000). Machine Learning from Imbalanced Data Sets 101 Extended Abstract. *Machine Learning*, 3.
- Rizki, F., Kharisma Putra, M. P., Assuja, M. A., & Ariany, F. (2023). Implementasi Deep Learning Lenet Dengan Augmentasi Data Pada Identifikasi Anggrek. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 4(3), 357–366. <https://doi.org/10.33365/jatika.v4i3.3652>
- Romadhonia, R. W., Sofro, A., Ariyanto, D., Maulana, D. A., & Prihanto, J. B. (2023). *Application of Decision Trees in Athlete Selection : A Cart Algorithm Approach*. 14.
- Safitri, K., & Widiati, E. (2022). Pengaruh Online Customer Review, Rating, dan Kepercayaan terhadap Niat Beli Konsumen pada Aplikasi Tokopedia. *Manajerial: Jurnal Manajemen Dan Sistem Informasi*, 21(2), 153–160. <http://ejournal.upi.edu/index.php/manajerial/>
- Sari, Y. V., Muallifah, Z., & Fanani, A. (2023). Klasifikasi Kualitas Air Menggunakan Metode Extreme Learning Machine (ELM). *Jurnal JUPITER*, 15(2), 983–994. <https://jurnal.polsri.ac.id/index.php/jupiter/article/view/6995>
- Septiani, D., & Isabela, I. (2022). Analisis Term Frequency Inverse Document Frequency (Tf-Idf) Dalam Temu Kembali Informasi Pada Dokumen Teks. *SINTESIA: Jurnal Sistem Dan Teknologi Informasi Indonesia*, 1(2), 81–88.
- Siregar, D., Ladayya, F., Albaqi, N. Z., & Wardana, B. M. (2023). Penerapan Metode Support Vector Machines (SVM) dan Metode Naïve Bayes Classifier (NBC) dalam Analisis Sentimen Publik terhadap Konsep Child-free di Media Sosial Twitter. *Jurnal Statistika Dan Aplikasinya*, 7(1), 93–104.
- Suryani, Rahmadani, D., Muzafar, A. A., Hamid, A., Annisa, R., & Mustakim. (2022). Analisis Perbandingan Algoritma C4.5 dan CART untuk Klasifikasi Penyakit Stroke. *SENTIMAS: Seminar Nasional Penelitian Dan Pengabdian Masyarakat*, 197–206. <https://journal.irpi.or.id/index.php/sentimas>
- Tan, K. L., Lee, C. P., & Lim, K. M. (2023). A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. *Applied Sciences (Switzerland)*, 13(7). <https://doi.org/10.3390/app13074550>
- Utami, H. (2022). Analisis Sentimen dari Aplikasi Shopee Indonesia Menggunakan Metode Recurrent Neural Network. *Indonesian Journal of Applied Statistics*, 5(1), 31. <https://doi.org/10.13057/ijas.v5i1.56825>

## LAMPIRAN

### 1. Pengambilan Data

```
1 from google_play_scraper import Sort, reviews
2
3 result, continuation_token = reviews(
4     'id.go.kpu.sirekap2024',
5     lang='id',
6     country='id',
7     sort=Sort.MOST_RELEVANT,
8     count=10000000,
9     filter_score_with=None
10 )
```

#### Lampiran 1 Proses Pengambilan Data

```
1 import pandas as pd
2 import numpy as np
3
4 df = pd.DataFrame(np.array(result), columns=['review'])
5 df = df.join(pd.DataFrame(df.pop('review').tolist()))
6
7 df.head()
```

#### Lampiran 2 Memasukan Data Ke *Dataframe* Pandas

```
1 df = df[['userName', 'score', 'at', 'content', 'thumbsUpCount']]
2 df.sort_values(by='at', ascending=False)
3
4 df.head()
```

#### Lampiran 3 *Filtering* dan *Sorting* Kolom

```
5 df.to_csv("scrapped_data.csv", index = False, sep='|')
```

#### Lampiran 4 Ekspor Data Ke File CSV

```
1 import pandas as pd
2
3 file_csv = 'scrapped_data.csv'
4
5 df = pd.read_csv(file_csv, sep='|')
6
7 print(df)
```

#### Lampiran 5 Input File CSV

```

1  import matplotlib.pyplot as plt
2
3  plt.figure(figsize=(8, 6))
4  ratings, counts =
5  zip(*sorted(dict(df['Rating'].value_counts()).items()))
6  plt.bar(ratings, counts, alpha=0.7, edgecolor='black', align='center')
7  plt.title('Distribusi Rating')
8  plt.xlabel('Rating')
9  plt.ylabel('Frekuensi')
10 plt.xticks(ratings)
11 plt.grid(axis='y', linestyle='--', alpha=0.7)
12 plt.show()
13
14 rating_counts = df['Rating'].value_counts().sort_index()
15
16 for rating, count in rating_counts.items():
17     print(f"Rating {rating}: {count} kali")

```

### Lampiran 6 Proses Distribusi *Rating*

```

17  import matplotlib.pyplot as plt
18
19  plt.figure(figsize=(8, 6))
20  ratings, counts =
21  zip(*sorted(dict(df['Rating'].value_counts()).items()))
22  plt.bar(ratings, counts, alpha=0.7, edgecolor='black', align='center')
23  plt.title('Distribusi Rating')
24  plt.xlabel('Rating')
25  plt.ylabel('Frekuensi')
26  plt.xticks(ratings)
27  plt.grid(axis='y', linestyle='--', alpha=0.7)
28  plt.show()
29
30 rating_counts = df['Rating'].value_counts().sort_index()
31
32 for rating, count in rating_counts.items():
33     print(f"Rating {rating}: {count} kali")

```

### Lampiran 7 Proses *Wordcloud*

## 2. *Pra-proses Data*

```
1 digit_count = len(str(len(df)))
2
3 df.insert(0, 'ID', df.index.map(lambda x: 'd' +
4 str(x+1).zfill(digit_count)))
5 print(df)
```

### **Lampiran 8** Proses Pemberian ID Disetiap Data

```
1 import time
2
3 def lowercase(text):
4     return text.lower()
5
6 start_time = time.time()
7 df['komentar_lowercase'] = df['komentar'].apply(lowercase)
8 end_time = time.time()
9 time_taken = end_time - start_time
10
11 print(df[['komentar', 'komentar_lowercase']])
12 print("Waktu proses:", time_taken, "detik")
```

### **Lampiran 9** Proses *Lower Case*

```
1 import re
2
3 def remove_unnecessary_char(text):
4     text = re.sub('((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))', '
5 ',text)
6     text = re.sub('\n', ' ',text)
7     text = re.sub('\r', ' ',text)
8     text = re.sub(r'\\x\.', ' ',text)
9     text = re.sub(' +', ' ', text)
10    text = re.sub('[^0-9a-zA-Z]+', ' ', text)
11    return text
12
13 start_time = time.time()
14 df['komentar_remove_char'] =
15 df['komentar_lowercase'].apply(remove_unnecessary_char)
16 end_time = time.time()
17 time_taken = end_time - start_time
18
19 print(df[['komentar_lowercase', 'komentar_remove_char']])
20 print("Waktu proses:", time_taken, "detik")
```

### **Lampiran 10** Proses Remove Unnecessary Character

```

12 import re
13
14 # Path untuk file input dan output
15 file_path = "kbi.txt"
16 output_file_path = "4 kata lebih list kata.txt"
17
18 # Membaca konten file
19 with open(file_path, "r", encoding="utf-8") as file:
20     text = file.read()
21
22 # Ekstraksi kata-kata dan konversi ke huruf kecil
23 words = re.findall(r'\b\w+\b', text.lower())
24
25 # Menghapus kata-kata yang mengandung karakter numerik
26 words = [word for word in words if not any(char.isdigit() for char in
word)]
27
28 # Menghapus kata-kata yang hanya terdiri dari satu hingga tiga huruf
29 words = [word for word in words if len(word) >= 4]
30
31 # Menghapus duplikat kata dengan menggunakan set
32 unique_words = set(words)
33
34 # Mengurutkan kata-kata sesuai dengan abjad
35 sorted_words = sorted(unique_words)
36
37 # Membuka file untuk menulis hasil output
38 with open(output_file_path, "w", encoding="utf-8") as output_file:
39     for word in sorted_words:
40         output_file.write(word + "\n")
41
42 print(f"Kata-kata unik telah disimpan di file {output_file_path}")

```

**Lampiran 11** Source Code Kamus "4 lebih list kata KBBI.txt"

```

1 with open('4 lebih list kata KBBI.txt', 'r', encoding='utf-8') as file:
2     kamus_baku = set(file.read().splitlines())
3
4 def detect_and_insert_unknown_words(text, baku_dict):
5     words = text.split()
6     unknown_words = [word for word in words if word.lower() not in
baku_dict]
7     return ', '.join(unknown_words)
8
9 df['Kata_Tidak_Baku'] = df['komentar_stemming'].apply(lambda x:
detect_and_insert_unknown_words(x, kamus_baku))
10
11 print(df['Kata_Tidak_Baku'])

```

**Lampiran 12** Cek Kata Tidak Baku

```

1 def detect_and_create_dataframe(text, baku_dict):
2     words = text.split()
3     kata_tidak_baku = set(word.lower() for word in words if word.lower()
4     not in baku_dict)
5     return pd.DataFrame({'Kata_Tidak_Baku': list(kata_tidak_baku)})
6
7 df_kata_tidak_baku = pd.concat([detect_and_create_dataframe(row,
8     kamus_baku) for row in df['komentar_stemming']], ignore_index=True)
9
10 df_kata_tidak_baku = df_kata_tidak_baku.drop_duplicates()
11 df_kata_tidak_baku_sorted =
12     df_kata_tidak_baku.sort_values(by='Kata_Tidak_Baku')
13
14 print(df_kata_tidak_baku_sorted)

```

### Lampiran 13 Cek List Kata Tidak Baku

```

43 alay_dict = pd.read_csv('kamus_tidak_baku.csv', sep="," ,
44     encoding='latin-1', header=None)
45 alay_dict = alay_dict.rename(columns={0: 'original',
46     1: 'replacement'})
47
48 alay_dict_map = dict(zip(alay_dict['original'],
49     alay_dict['replacement']))

```

### Lampiran 14 Proses Input "kamus\_tidak\_baku"

```

12 def normalize_alay(text):
13     return ' '.join([alay_dict_map[word] if word in alay_dict_map
14     else word for word in text.split(' ')])
15
16 start_time = time.time()
17 df['komentar_spellchecker'] = df['komentar_remove_char'].apply
18     (normalize_alay)
19 end_time = time.time()
20 time_taken = end_time - start_time
21
22 print(df[['komentar_remove_char', 'komentar_spellchecker']])
23 print("Waktu proses:", time_taken, "detik")

```

### Lampiran 15 Proses *Spell Checker*

```

1 from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
2 factory = StemmerFactory()
3 stemmer = factory.create_stemmer()
4
5 def stemming(text):
6     return stemmer.stem(text)
7
8 start_time = time.time()
9
10 df['komentar_stemming'] = df['komentar_spellchecker'].apply(stemming)
11
12 end_time = time.time()
13 time_taken = end_time - start_time

```

### Lampiran 16 Proses *Stemming*

```

1 import numpy as np
2
3 df['komentar_stemming'] = df['komentar_stemming'].replace('', np.nan)
4
5 baris_nan = df[df['komentar_stemming'].isnull()]
6 jumlah_nan = df['komentar_stemming'].isnull().sum()
7 print(f"Jumlah NaN pada kolom 'komentar_stemming': {jumlah_nan}")
8 print(baris_nan)

```

#### Lampiran 17 Cek Data Kosong

```

1 df_cleaned = df.dropna(subset=['komentar_stemming'])
2 df_cleaned

```

#### Lampiran 18 Hapus Data Kosong

### 3. Proses Klasifikasi dan Evaluasi *F1-Score Wordnet*

```

1 !pip install Deep-translator
2
3 from deep_translator import GoogleTranslator
4 langs_dict = GoogleTranslator().get_supported_languages(as_dict=True)
5 import time
6
7 def translate_komentar(text):
8     text_translated = GoogleTranslator(source='id',
9     target='en').translate(text)
10     return text_translated
11
12 from tqdm import tqdm
13
14 tqdm.pandas()
15
16 start_time = time.time()
17 df['Translated'] =
18 df['komentar_stemming'].progress_apply(translate_komentar)
19 end_time = time.time()
20 time_taken = end_time - start_time
21
22 print(df[['komentar_stemming', 'Translated']])
23 print("Waktu proses:", time_taken, "detik")

```

#### Lampiran 19 Proses Penerjemahan Data Ulasan ke dalam Bahasa Inggris

```

1  from TextBlob import TextBlob
2
3  def scoring_sentiment(text):
4      analysis = TextBlob(text)
5      sentiment_score = analysis.sentiment.polarity
6      # print("Sentiment polarity:", sentiment_score)
7      score_Wordnet = 0
8
9      if sentiment_score > 0.5:
10         score_Wordnet = '5 star'
11     elif 0.2 <= sentiment_score <= 0.5:
12         score_Wordnet = '4 star'
13     elif -0.2 < sentiment_score < 0.2:
14         score_Wordnet = '3 star'
15     elif -0.5 <= sentiment_score <= -0.2:
16         score_Wordnet = '2 star'
17     else:
18         score_Wordnet = '1 star'
19
20     return score_Wordnet, sentiment_score

```

### Lampiran 20 Fungsi Hitung Score

```

1  start_time = time.time()
2  score_Wordnet, sentiment_score =
zip(*df['Translated'].apply(scoring_sentiment))
3  df['Score_Wordnet'] = score_Wordnet
4  df['Sentiment_Score'] = sentiment_score
5  end_time = time.time()
6  time_taken = end_time - start_time
7
8  print(df[['komentar_stemming', 'Score_Wordnet', 'Sentiment_Score']])
9  print("Waktu proses:", time_taken, "detik")

```

### Lampiran 21 Proses Wordnet

```

1  import pandas as pd
2  import numpy as np
3  import matplotlib.pyplot as plt
4  from sklearn.metrics import confusion_matrix, f1_score
5  from matplotlib.ticker import FixedLocator
6
7  cm = confusion_matrix(df['Rating'], df['Score_Wordnet'])
8
9  Macro_f1 = f1_score(df['Rating'], df['Score_Wordnet'], Average='Macro')
10 print(f"Macro F1-Score: {Macro_f1}")
11
12 fig, ax = plt.subplots(figsize=(10, 8), dpi=100)
13 cax = ax.matshow(cm, cmap=plt.cm.Greens)
14 fig.colorbar(cax)
15
16 tick_marks = np.arange(len(np.unique(df['Rating'])))
17 ax.set_xticks(tick_marks)
18 ax.set_yticks(tick_marks)
19 ax.set_xticklabels(list(range(1, 6)))
20 ax.set_yticklabels(list(range(1, 6)))
21
22 plt.xlabel('Predicted')
23 plt.ylabel('Actual')
24 plt.title('Confusion Matrix Wordnet')
25
26 thresh = cm.max() / 2
27 for i in range(len(cm)):
28     for j in range(len(cm[i])):
29         value = cm[i][j]
30         text_color = 'white' if value > thresh else 'black'
31         ax.text(j, i, str(value), va='center', ha='center',
32               color=text_color)
33 plt.show()

```

**Lampiran 22** Proses Evaluasi *F1-Score Wordnet*

#### 4. Proses Klasifikasi dan Evaluasi *F1-Score Decision Tree CART*

```
1 X = df['komentar_stemming']
2 y = df['Rating']
```

##### Lampiran 23 Proses Penentuan Input dan Kelas

```
1 from sklearn.feature_extraction.text import TfidfVectorizer
2
3 vectorizer = TfidfVectorizer()
4 features = vectorizer.fit_transform(X)
```

##### Lampiran 24 Proses Ekstraksi Fitur TF-IDF

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.metrics import confusion_matrix, Precision_score,
4 Recall_score, f1_score
5 from sklearn.model_selection import KFold, cross_val_predict
6
7 def cross_validation(model, _X, _y, _cv):
8     kf = KFold(n_splits=_cv, shuffle=True, random_state=42)
9     fold_count = 1
10    results = {}
11    fold_predictions = []
12    overall_cm = np.zeros((len(np.unique(_y)), len(np.unique(_y))),
13                          dtype=int)
14
15    for train_index, test_index in kf.split(_X):
16        X_train, X_test = _X[train_index], _X[test_index]
17        y_train, y_test = _y[train_index], _y[test_index]
18
19        model.fit(X_train, y_train)
20        y_pred = model.predict(X_test)
21        fold_predictions.append(y_pred)
22        cm = confusion_matrix(y_test, y_pred)
23
24        results[f"Fold {fold_count}"] = {
25            "Confusion Matrix": cm,
26            "Precision": Precision_score(y_test, y_pred,
27                                       Average='Macro'),
28            "Recall": Recall_score(y_test, y_pred, Average='Macro'),
29            "F1 Score": f1_score(y_test, y_pred, Average='Macro')
30        }
31        overall_cm += cm
32        fold_count += 1
33
34    for key, value in results.items():
35        print(key + ":")
36        print("Confusion Matrix:")
37        print(value["Confusion Matrix"])
38        print("Precision:", value["Precision"])
39        print("Recall:", value["Recall"])
40        print("F1 Score:", value["F1 Score"])
41        print()
42
43    fig, ax = plt.subplots(figsize=(10, 8), dpi=100)
```

```

41     im = ax.imshow(value["Confusion Matrix"],
42 interpolation='nearest', cmap=plt.cm.Greens)
43     ax.figure.colorbar(im, ax=ax)
44
45     class_names = np.unique(_y)
46     ax.set(xticks=np.arange(len(class_names)),
47           yticks=np.arange(len(class_names)),
48           xticklabels=class_names, yticklabels=class_names,
49           title=f'Confusion Matrix - {key}',
50           ylabel='True label',
51           xlabel='Predicted label')
52
53     thresh = value["Confusion Matrix"].max() / 2.
54     for i in range(value["Confusion Matrix"].shape[0]):
55         for j in range(value["Confusion Matrix"].shape[1]):
56             ax.text(j, i, format(value["Confusion Matrix"][i, j],
57 'd'),
58                   ha="center", va="center",
59                   color="white" if value["Confusion Matrix"][i, j]
60 > thresh else "black")
61
62     plt.show()
63
64     print("Overall Confusion Matrix:")
65     print(overall_cm)
66     print()
67
68     fig, ax = plt.subplots(figsize=(10, 8), dpi=100)
69     cax = ax.matshow(overall_cm, cmap=plt.cm.Greens)
70     fig.colorbar(cax)
71
72     class_names = np.unique(_y)
73     ax.set_xticks(np.arange(len(class_names)))
74     ax.set_yticks(np.arange(len(class_names)))
75     ax.set_xticklabels(class_names)
76     ax.set_yticklabels(class_names)
77     plt.xlabel('Predicted')
78     plt.ylabel('Actual')
79     plt.title('Overall Confusion Matrix')
80
81     thresh = overall_cm.max() / 2.
82     for i in range(overall_cm.shape[0]):
83         for j in range(overall_cm.shape[1]):
84             ax.text(j, i, format(overall_cm[i, j], 'd'), ha='center',
85 va='center',
86                   color='white' if overall_cm[i, j] > thresh else
87 'black')
88
89     plt.show()
90
91     y_pred = cross_val_predict(model, _X, _y, cv=_cv)
92     cm_overall = confusion_matrix(_y, y_pred)
93
94     Precision_Macro = Precision_score(_y, y_pred, Average='Macro')
95     print(f"\nPrecision Macro: {Precision_Macro}")
96
97     Recall_Macro = Recall_score(_y, y_pred, Average='Macro')
98     print(f"Recall Macro: {Recall_Macro}")

```

```

94
95     f1_score_Macro = f1_score(_y, y_pred, Average='Macro')
96     print(f"F1-Score Macro: {f1_score_Macro}")
97
98     return results, fold_predictions

```

### Lampiran 25 Proses Cross-Validation 10 Fold

```

1     kfold = KFold(n_splits=10, shuffle=False)
2     for fold, (train_index, test_index) in enumerate(kfold.split(X, y), 1):
3         X_train, X_test = X[train_index], X[test_index]
4         y_train, y_test = y[train_index], y[test_index]
5
6         print(f'Fold {fold}:')
7         print(f' - Train data: {len(X_train)} samples')
8         print(f' - Test data: {len(X_test)} samples')
9         print(f' - Train Index: {train_index
10    }')

```

### Lampiran 26 Menampilkan Jumlah Data Setiap Pembagian Fold

```

1     from sklearn.tree import DecisionTreeClassifier
2
3     cart_model = DecisionTreeClassifier(
4         criterion='gini',
5         max_depth=None,
6         min_samples_split=2,
7         min_samples_leaf=1,
8         max_features=None,
9         class_weight=None,
10        random_state=0
11    )

```

### Lampiran 27 Proses Input Model Klasifikasi Decision Tree CART

```

1     import time
2
3     start_time = time.time()
4     cv_results, fold_predictions = cross_validation(cart_model, features, y,
5         _cv=10)
6
7     end_time = time.time()
8     time_taken = end_time - start_time
9     print("Waktu proses:", time_taken, "detik")

```

### Lampiran 28 Proses Visualisasi Model dengan Menjalankan Cross Validation

## 5. Dokumen

 UNIVERSITAS MUHAMMADIYAH Kalimantan Timur Berakhlak   Berwawasan   Berkemajuan	<b>UMKT</b> Program Studi Teknik Informatika Fakultas Sains dan Teknologi	Telp. 0541-748511 Fax. 0541-766832 Website <a href="http://informatika.umkt.ac.id">http://informatika.umkt.ac.id</a> email: <a href="mailto:informatika@umkt.ac.id">informatika@umkt.ac.id</a>
---	--	--

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Nomor : 055-001/KET/FST.1/A/2024  
Lampiran : -  
Perihal : **Keterangan Pengambilan Data Sekunder**

*Assalamu'alaikum Warrahmatullahi Wabarrakatuh*

Puji Syukur kepada Allah Subhanahu wa ta'ala yang senantiasa melimpahkan Rahmat-Nya kepada kita sekalian. Amin.

Dengan surat ini, kami menerangkan bahwa mahasiswa berikut:

No	Nama	NIM
1	Emyzar Hafliida Tanjung	2011102441240
2	Lilis Sagita	2011102441198
3	Sri Ramadani	2011102441177
4	Muhammad Fariz Ijlal Rafi	2011102441124
5	Nurlita	2011102441070

Melakukan penelitian dengan pengambilan data sekunder di Google Playstore data yang diambil yaitu data ulasan aplikasi "Sirekap" dari rating 1-5.

Demikian hal ini disampaikan, atas kerjasamanya kami ucapkan terima kasih.

*Wassalamu'alaikum Warrahmatullahi Wabarrakatuh*

Samarinda, 18 Dzulhijjah 1445 H  
25 Juni 2024 M

Ketua Program Studi S1 Teknik Informatika  
  
**Arbansyah, S.Kom., M.TI**  
IDN. 1118019203



Kampus 1 : Jl. Ir. H. Juanda, No.15, Samarinda  
Kampus 2 : Jl. Pelita, Pesona Mahakam, Samarinda

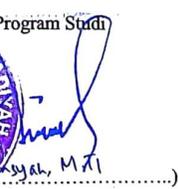
### Lampiran 29 Surat Keterangan Pengambilan Data Sekunder

### KARTU KENDALI BIMBINGAN LAPORAN KARYA ILMIAH

Nama : Muhammad Fariz Ijlal Rafi  
 NIM : 2011102441124  
 Nama Dosen Pembimbing : Naufal Azmi Verdikha, S.Kom., M.Eng.  
 Judul : ANALISIS KOMPARASI MODEL KLASIFIKASI UNTUK  
 ULASAN APLIKASI SIREKAP 2024 MENGGUNAKAN  
 CART DAN METODE *WORDNET*

No	Tanggal	Uraian Pembimbingan	Paraf Dosen
1	26/01/2024	Diskusi mengenai data yang akan digunakan dalam penelitian	
2	27/01/2024	Pemilihan metode yang akan digunakan dalam penelitian	
3	28/01/2024	Evaluasi awal data ulasan yang akan digunakan dan literatur terkait metode CART dan <i>Wordnet</i> .	
4	29/01/2024	Pembahasan metode pengumpulan data dan teknik preprocessing yang relevan.	
5	07/02/2024	Review hasil sementara preprocessing data dan perbaikan berdasarkan saran.	
6	15/02/2024	Diskusi mengenai penerapan model CART pada dataset dan evaluasi awal hasilnya.	
7	20/03/2024	Evaluasi penerapan metode <i>Wordnet</i> untuk analisis sentimen pada data ulasan.	
8	18/04/2024	Perbandingan hasil klasifikasi antara model CART dan metode <i>Wordnet</i> , serta interpretasi hasilnya.	
9	30/04/2024	Review dan saran perbaikan laporan hasil penelitian	
10	10/05/2024	Pembahasan mengenai integrasi temuan dalam laporan dan persiapan dokumen akhir.	
11	26/06/2024	Finalisasi laporan penelitian dan persiapan untuk presentasi akhir.	

Mengetahui

Dosen Pembimbing  Ketua Program Studi 



**Lampiran 30** Kartu Kendali Bimbingan Laporan Karya Ilmiah

# SKRIPSI MUHAMMAD FARIZ IJLAL RAFI

by Teknik Informatika Universitas Muhammadiyah Kalimantan Timur



---

**Submission date:** 25-Jul-2024 10:56AM (UTC+0800)

**Submission ID:** 2422080893

**File name:** SKRIPSI\_MUHAMMAD\_FARIZ\_IJLAL\_RAfi.docx (1.96M)

**Word count:** 7128

**Character count:** 44307

# SKRIPSI MUHAMMAD FARIZ IJLAL RAFI

## ORIGINALITY REPORT

14%

SIMILARITY INDEX

13%

INTERNET SOURCES

7%

PUBLICATIONS

4%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://repository.its.ac.id">repository.its.ac.id</a> Internet Source	2%
2	<a href="http://ijcs.stmikindonesia.ac.id">ijcs.stmikindonesia.ac.id</a> Internet Source	1%
3	<a href="http://repository.uin-suska.ac.id">repository.uin-suska.ac.id</a> Internet Source	1%
4	<a href="http://jurnal.umk.ac.id">jurnal.umk.ac.id</a> Internet Source	<1%
5	<a href="http://dspace.uii.ac.id">dspace.uii.ac.id</a> Internet Source	<1%
6	<a href="http://dspace.umkt.ac.id">dspace.umkt.ac.id</a> Internet Source	<1%
7	<a href="http://gudangjurnal.com">gudangjurnal.com</a> Internet Source	<1%
8	<a href="http://text-id.123dok.com">text-id.123dok.com</a> Internet Source	<1%
9	<a href="http://pdfcookie.com">pdfcookie.com</a> Internet Source	<1%

## RIWAYAT HIDUP



Penulis lahir di Suliliran Baru pada 9 Agustus 2002 sebagai anak pertama dari tiga bersaudara, dari pasangan Kusnadi dan Surtinah. Saat ini, penulis tinggal di Jl. K.H. Wahid Hasyim, Gg. Sungai II, RT 011, Sempaja Selatan, Samarinda Utara. Penulis menyelesaikan pendidikan di SD Negeri 013 Pasir Belengkong pada tahun 2014, kemudian melanjutkan ke SMP Negeri 03 Pasir Belengkong dan lulus pada tahun 2017, serta menamatkan pendidikan di SMA Negeri 01 Pasir Belengkong pada tahun 2020. Saat ini, penulis sedang menempuh pendidikan di Universitas Muhammadiyah Kalimantan Timur di Kota Samarinda dengan jurusan Teknik Informatika Internasional dan belajar di gedung Fakultas Sains dan Teknologi. Penulis pernah menjalani program magang pada semester 7 di PT. Barqun Digital Teknologi, sebuah perusahaan software house di Samarinda. Saat ini, penulis sedang menyelesaikan tugas akhir atau skripsi.