

# **PERBANDINGAN ALGORITMA SVM DAN NAIVE BAYES UNTUK ANALISIS SENTIMEN PENGGUNA APLIKASI DI PLAYSTORE BERBASIS WEBSITE**

## **TUGAS AKHIR**

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi  
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)  
dalam Teknologi Informasi



Oleh :

**Alfa Rifa Luky Achmad Rayendra**

NIM : 2108096099

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG  
2025**

# **PERBANDINGAN ALGORITMA SVM DAN NAIVE BAYES UNTUK ANALISIS SENTIMEN PENGGUNA APLIKASI DI PLAYSTORE BERBASIS WEBSITE**

## **TUGAS AKHIR**

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi  
Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1)  
dalam Teknologi Informasi



Oleh :

**Alfa Rifa Luky Achmad Rayendra**

NIM : 2108096099

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG  
2025**



## PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Alfa Rifa Luky Achmad Rayendra  
NIM : 2108096099  
Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

### **PERBANDINGAN ALGORITMA SVM DAN NAIVE BAYES UNTUK ANALISIS SENTIMEN PENGGUNA APLIKASI DI PLAYSTORE BERBASIS WEBSITE**

Secara keseluruhan adalah hasil penelitian/karya saya sendiri,  
kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 16 Juni 2025  
Pembuat Pernyataan,



Alfa Rifa Luky Achmad Rayendra  
NIM : 2108096099





## PENGESAHAN

Naskah proposal skripsi berikut ini:

Judul : PERBANDINGAN ALGORITMA SVM DAN NAIVE  
BAYES UNTUK ANALISIS SENTIMEN PENGGUNA  
APLIKASI DI PLAYSTORE BERBASIS WEBSITE

Penulis : **Alfa Rifa Luky Achmad Rayendra**

NIM : 2108096099

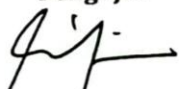
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji  
Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima  
sebagai salah satu syarat memperoleh gelar sarjana dalam  
Teknologi Informasi.

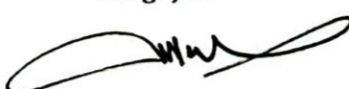
Semarang, 30 Juni 2025

### DEWAN PENGUJI

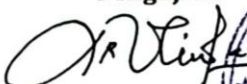
**Penguji I**

  
Dr. Khotibul Umam, M.Kom  
NIP. 197908272011011007


**Penguji II**

  
Adzhal Arwani Mahfudh, M.Kom  
NIP. 199107032019031006

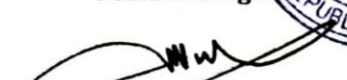
**Penguji III**

  
Masy Ari Ulinuha, MT  
NIP. 198108122011011007


**Penguji IV**

  
Mokhammad Ikil Mustofa, M.Kom  
NIP. 198808072019031010

**Pembimbing I**

  
Adzhal Arwani Mahfudh, M.Kom  
NIP. 199107032019031006

**Pembimbing II**

  
Siti Nur'aini, M.Kom  
NIP. 198401312018012001





## NOTA DINAS

Semarang, 16 /Juni/2025

Yth. Ketua Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
UIN Walisongo Semarang

*Assalamu'alaikum. wr. wb.*

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : PERBANDINGAN ALGORITMA SVM DAN NAIVE  
BAYES UNTUK ANALISIS SENTIMEN PENGGUNA  
APLIKASI DI PLAYSTORE BERBASIS WEBSITE

Penulis : **Alfa Rifa Luky Achmad Rayendra**

NIM : 2108096099

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

*Wassalamu'alaikum. wr. wb.*

Pembimbing I



Adzhal Arwani Mahfudh, M.Kom  
NIP. 199107032019031006





## NOTA DINAS

Semarang, 16 /Juni/2025

Yth. Ketua Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
UIN Walisongo Semarang

*Assalamu'alaikum. wr. wb.*

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : PERBANDINGAN ALGORITMA SVM DAN NAIVE  
BAYES UNTUK ANALISIS SENTIMEN PENGGUNA  
APLIKASI DI PLAYSTORE BERBASIS WEBSITE

Penulis : **Alfa Rifa Luky Achmad Rayendra**

NIM : 2108096099

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosyah.

*Wassalamu'alaikum. wr. wb.*

Pembimbing II



Siti Nur'aini, M.Kom  
NIP. 198401312018012001



## ABSTRAK

Play Store sebagai salah satu platform penyedia mengalami penurunan jumlah aplikasi sebesar 45,9% hingga awal tahun 2025 yang menandakan persaingan semakin ketat antara aplikasi yang mengharuskan pengembang untuk terus meningkatkan kualitas dan daya saing produknya. Melalui ulasan pengembang dapat mengukur kepuasan pelanggan dan memantau persaingan pada produk atau layanan serupa. Peningkatan jumlah pengguna suatu aplikasi akan berdampak pada volume ulasan yang mayoritasnya berupa data tidak terstruktur. Untuk dapat mengekstrak informasi penting diperlukan pendekatan yang secara otomatis, yaitu dengan analisis sentimen. Pada studi ini akan membandingkan performa dari dua algoritma klasifikasi yakni SVM dan NB dengan tahapan meliputi pengumpulan data, persiapan data, *preprocessing*, *modeling*, evaluasi model, visualisasi dan analisis. Data ulasan yang digunakan adalah ulasan dari aplikasi MyPertamina dengan jumlah 14.000 ulasan. Berdasarkan studi, didapatkan data ulasan hasil *preprocessing* didominasi oleh sentimen negatif sebesar 66%. Adapun model klasifikasi SVM lebih unggul dibandingkan NB di dua skenario data. Dari dua skenario tersebut model dengan kombinasi metode SMOTE memiliki akurasi yang lebih baik dibandingkan model awal. Model SVM memiliki performa tertinggi dengan akurasi 88,11%, *precision* 88%, *recall* 88% dan *f1-score* 88%, sedangkan model NB mendapat akurasi 86,52%, *precision* 86%, *recall* 87% dan *f1-score* 86%. Perbedaan akurasi sebesar 1,59%, *precision* 2%, *recall* 1% dan *f1-score* 2% membuat model SVM lebih unggul dibandingkan model NB. Integrasi sistem analisis sentimen kedalam aplikasi berbasis web telah berhasil dengan hasil 100% pada pengujian *Black Box Testing*.

**Kata kunci** : Analisis Sentimen, SVM, Naive Bayes, TF-IDF, SMOTE, Webiste



## KATA PENGANTAR

Puji syukur kehadiran Allah SWT, penguasa alam semesta karena segala rahmat, taufik dan hidayah-Nya. Tak lupa kita panjatkan shalawat dan salam kepada Nabi Besar Muhammad SAW, sehingga Tugas Akhir penulis yang berjudul **“Perbandingan Algoritma SVM dan Naive Bayes Untuk Analisis Sentimen Pengguna Aplikasi Di Playstore Berbasis Website”** dapat diselesaikan dengan baik. Adapun tujuan penulisan Tugas Akhir ini sebagai syarat kelulusan dalam menempuh Pendidikan di Progam Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Walisongo Semarang.

Penyusunan Tugas Akhir ini tidak terlepas dari bantuan beberapa pihak, oleh karena itu penulis hendak mengucapkan terimakasih kepada :

1. Dr. Khotibul Umam S.T, M.Kom., selaku Ketua Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Walisongo Semarang.
2. Bapak Adzhal Arwani Mahfudh, M.Kom., selaku Dosen Pembimbing I yang telah memberikan bimbingan, masukan, serta arahan kepada penulis.

3. Ibu Siti Nur'aini, M.Kom., selaku Dosen Pembimbing II yang telah memberikan bimbingan, masukan, serta arahan kepada penulis.
4. Seluruh pihak yang telah berkontribusi dalam penyusunan tugas akhir ini namun tidak dapat disebutkan satu per satu.

Akhir kata, penulis berharap segala bentuk kebaikan dan ketulusan yang telah diberikan mendapatkan balasan yang sebaik-baiknya dari Allah SWT. Semoga skripsi ini dapat memberikan manfaat bagi para pembaca serta menjadi referensi bagi penelitian-penelitian selanjutnya.

Semarang, 16 Juni 2025



Alfa Rifa Luky Achmad Rayendra  
NIM : 2108096099

## DAFTAR ISI

<b>PERNYATAAN KEASLIAN .....</b>	<b>iii</b>
<b>PENGESAHAN .....</b>	<b>v</b>
<b>NOTA DINAS .....</b>	<b>vii</b>
<b>NOTA DINAS .....</b>	<b>ix</b>
<b>ABSTRAK .....</b>	<b>xi</b>
<b>KATA PENGANTAR .....</b>	<b>xiii</b>
<b>DAFTAR ISI .....</b>	<b>xv</b>
<b>DAFTAR GAMBAR .....</b>	<b>xix</b>
<b>DAFTAR TABEL .....</b>	<b>xxiii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
A. Latar Belakang .....	1
B. Rumusan Masalah .....	7
C. Batasan Penelitian .....	7
D. Tujuan Penelitian .....	8
E. Manfaat Penelitian .....	8
<b>BAB II LANDASAN PUSTAKA .....</b>	<b>9</b>
A. Analisis Sentimen .....	9
B. <i>Preprocessing</i> .....	11
C. TF-IDF .....	12
D. SMOTE .....	13
E. SVM .....	14
F. Naive Bayes .....	15
G. Confusion Matrix .....	16



H.	Flask.....	18
I.	Kajian Penelitian yang Relevan.....	19
<b>BAB III METODOLOGI PENELITIAN.....</b>		<b>25</b>
A.	Data dan Sumber Data.....	25
B.	Metode Analisis Data .....	26
1.	Pengumpulan Data .....	27
2.	Persiapan Data .....	27
3.	<i>Preprocessing</i> .....	28
4.	Modeling .....	33
5.	Evaluasi Model .....	34
6.	Visualisasi.....	34
C.	Perancangan Sistem .....	35
1.	Analisis Kebutuhan.....	35
2.	Desain Sistem.....	37
3.	Implementasi.....	56
4.	Pengujian.....	56
5.	Maintenance.....	56
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN .....</b>		<b>57</b>
A.	Sistem Analisis Sentimen.....	57
1.	<i>Scraping Data</i> .....	57
2.	Persiapan Data .....	60
3.	<i>Preprocessing</i> .....	62
4.	Modeling .....	69
5.	Evaluasi Model .....	80
6.	Visualisasi.....	94

B.	Hasil Analisis Sentimen .....	101
C.	Implementasi Sistem .....	102
D.	Pengujian .....	113
<b>BAB V</b>	<b>PENUTUP .....</b>	<b>119</b>
A.	Kesimpulan .....	119
B.	Saran.....	120
<b>DAFTAR PUSTAKA</b> .....		<b>123</b>
<b>DAFTAR LAMPIRAN</b> .....		<b>127</b>
<b>LAMPIRAN 1 : Pengesahan Ujian Komprehensif</b> .....		<b>127</b>
<b>LAMPIRAN 2 : Riwayat Hidup</b> .....		<b>129</b>



## DAFTAR GAMBAR

<b>Gambar 3.1</b>	Flowchart Alur Analisis Sentimen .....	26
<b>Gambar 3.2</b>	Tahapan Metode Waterfall.....	35
<b>Gambar 3.3</b>	DFD Level 0 .....	38
<b>Gambar 3.4</b>	DFD Level 1 .....	39
<b>Gambar 3.5</b>	Halaman Login.....	42
<b>Gambar 3.6</b>	Halaman Dashboard .....	43
<b>Gambar 3.7</b>	Halaman Dataset Awal.....	44
<b>Gambar 3.8</b>	Modal Pencarian Aplikasi.....	45
<b>Gambar 3.9</b>	Modal Metode Upload.....	46
<b>Gambar 3.10</b>	Menampilkan Data Ulasan dalam Tabel .....	47
<b>Gambar 3.11</b>	Modal Konfirmasi Reset Dataset .....	48
<b>Gambar 3.12</b>	Modal Metode Scraping.....	49
<b>Gambar 3.13</b>	Halaman Preprocessing .....	50
<b>Gambar 3.14</b>	Halaman Modeling (Menu Metode) .....	51
<b>Gambar 3.15</b>	Halaman Hasil Modeling (Hasil Prediksi).....	52
<b>Gambar 3.16</b>	Halaman Hasil Modeling (Confusion Matrix)..	53
<b>Gambar 3.17</b>	Halaman Summary (Visualisasi Data) .....	54
<b>Gambar 3.18</b>	Halaman Summary (Perbandingan Model) .....	55
<b>Gambar 3.19</b>	Entity Relationship Diagram.....	56
<b>Gambar 4.1</b>	Install dan Import Module Scraping .....	57
<b>Gambar 4.2</b>	Proses Scraping Data.....	58
<b>Gambar 4.3</b>	Hasil Scraping Data.....	59
<b>Gambar 4.4</b>	Menghapus Atribut.....	60

<b>Gambar 4.5</b> Hasil Hapus Atribut .....	60
<b>Gambar 4.6</b> Menghapus Duplikat dan Pelabelan .....	61
<b>Gambar 4.7</b> Hasil Hapus Duplikat dan Pelabelan.....	61
<b>Gambar 4.8</b> Case Folding.....	62
<b>Gambar 4.9</b> Hasil Case Folding .....	62
<b>Gambar 4.10</b> Cleansing .....	63
<b>Gambar 4.11</b> Hasil Cleansing.....	63
<b>Gambar 4.12</b> Normalisasi .....	64
<b>Gambar 4.13</b> Hasil Normalisasi.....	64
<b>Gambar 4.14</b> Tokenisasi .....	65
<b>Gambar 4.15</b> Hasil Tokenisasi.....	65
<b>Gambar 4.16</b> Stopwords Removal.....	66
<b>Gambar 4.17</b> Hasil Stopword Removal.....	66
<b>Gambar 4.18</b> Lematization.....	67
<b>Gambar 4.19</b> Hasil Lematization .....	67
<b>Gambar 4.20</b> Menghapus Data Kosong.....	68
<b>Gambar 4.21</b> Hasil Menghapus Data Kosong.....	69
<b>Gambar 4.22</b> Filtering Sentimen.....	69
<b>Gambar 4.23</b> Spliting Data.....	70
<b>Gambar 4.24</b> Ekstraksi Fitur .....	71
<b>Gambar 4.25</b> Hasil Mean Score TF-IDF .....	72
<b>Gambar 4.26</b> Grafik Distribusi Kelas Data Training.....	76
<b>Gambar 4.27</b> Oversampling dengan SMOTE.....	77
<b>Gambar 4.28</b> Distibusi Kelas Setelah SMOTE .....	78
<b>Gambar 4.29</b> Pelatihan Model SVM dan NB .....	79

<b>Gambar 4.30</b>	Proses Pengujian Data Testing .....	80
<b>Gambar 4.31</b>	Menampilkan Tabel Hasil Prediksi .....	80
<b>Gambar 4.32</b>	Menampilkan Confusion Matrix .....	81
<b>Gambar 4.33</b>	Confusion Matrix SVM .....	82
<b>Gambar 4.34</b>	Confusion Matrix SVM (SMOTE) .....	82
<b>Gambar 4.35</b>	Confusion Matrix NB .....	83
<b>Gambar 4.36</b>	Confusion Matrix NB (SMOTE) .....	83
<b>Gambar 4.37</b>	Proses Menampilkan Laporan Klasifikasi .....	91
<b>Gambar 4.38</b>	Laporan Klasifikasi SVM .....	92
<b>Gambar 4.39</b>	Laporan Klasifikasi SVMS .....	92
<b>Gambar 4.40</b>	Laporan Klasifikasi NB .....	93
<b>Gambar 4.41</b>	Laporan Klasifikasi NBS .....	93
<b>Gambar 4.42</b>	Distribusi Sentimen Data Ulasan Bersih .....	94
<b>Gambar 4.43</b>	Wordcloud Ulasan Negatif .....	95
<b>Gambar 4.44</b>	Grafik Frekuensi Kata Ulasan Negatif .....	96
<b>Gambar 4.45</b>	Wordcloud Ulasan Netral .....	97
<b>Gambar 4.46</b>	Grafik Frekuensi Kata Ulasan Netral .....	98
<b>Gambar 4.47</b>	Wordcloud Ulasan Positif .....	99
<b>Gambar 4.48</b>	Frekuensi Kata Positif .....	100
<b>Gambar 4.49</b>	Implementasi Halaman Login .....	102
<b>Gambar 4.50</b>	Implementasi Halaman Dashboard Awal .....	103
<b>Gambar 4.51</b>	Implementasi Halaman Dashboard Akhir .....	103
<b>Gambar 4.52</b>	Implementasi Halaman Dataset Awal .....	104
<b>Gambar 4.53</b>	Implementasi Modal Pencarian Aplikasi .....	105
<b>Gambar 4.54</b>	Implementasi Modal Scraping Data .....	106

<b>Gambar 4.55</b> Implementasi Modal Upload File .....	107
<b>Gambar 4.56</b> Implementasi Halaman Dataset Akhir .....	107
<b>Gambar 4.57</b> Implementasi Modal Reset Dataset.....	108
<b>Gambar 4.58</b> Implementasi Halaman Preprocessing.....	109
<b>Gambar 4.59</b> Implementasi Halaman Menu Modeling .....	110
<b>Gambar 4.60</b> Implementasi Hasil Modeling Tab I.....	110
<b>Gambar 4.61</b> Implementasi Hasil Modeling Tab II .....	111
<b>Gambar 4.62</b> Implementasi Halaman Summary Tab I.....	112
<b>Gambar 4.63</b> Implementasi Halaman Summary Tab II .....	113

## DAFTAR TABEL

<b>Tabel 2.1</b> Confusion Matrix.....	17
<b>Tabel 2.2</b> Kajian Penelitian .....	21
<b>Tabel 3.1</b> Parameter Scraping.....	25
<b>Tabel 3.2</b> Penerapan Case Folding .....	29
<b>Tabel 3.3</b> Penerapan Cleansing .....	29
<b>Tabel 3.4</b> Penerapan Normalisasi .....	30
<b>Tabel 3.5</b> Penerapan Tokenization.....	31
<b>Tabel 3.6</b> Penerapan Stopword Removal .....	31
<b>Tabel 3.7</b> Penerapan Lemmatization.....	32
<b>Tabel 3.8</b> Kebutuhan Perangkat Keras .....	37
<b>Tabel 3.9</b> Kebutuhan Perangkat Lunak.....	37
<b>Tabel 4.1</b> Perhitungan TF .....	73
<b>Tabel 4.2</b> Perhitungan IDF .....	74
<b>Tabel 4.3</b> Perhitungan TF-IDF.....	75
<b>Tabel 4.4</b> Hasil Confusion Matrix .....	84
<b>Tabel 4.5</b> Perbandingan Akurasi Model.....	85
<b>Tabel 4.6</b> Pengujian.....	114





# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Pengguna telepon seluler di Indonesia mencapai 67,29% dari total populasi penduduk di tahun 2023 (Badan Pusat Statistik, 2024). Peningkatan ini mendorong pertumbuhan aplikasi seluler di sektor pendidikan, kesehatan, dan e-commerce, yang mengubah pola interaksi masyarakat di era digital.

Play Store sebagai salah satu platform penyedia aplikasi android menjadi sangat penting dalam ekosistem digital. Pada tahun 2021, terdapat tiga juta aplikasi di Play Store yang jumlah tersebut justru mengalami penurunan sebesar 45,9% hingga awal tahun 2025 (AppBrain, 2025). Penurunan ini dipengaruhi oleh berbagai sebab, salah satunya adalah persaingan semakin ketat antara aplikasi yang mengharuskan pengembang untuk terus meningkatkan kualitas dan daya saing produknya agar tetap relevan dan diminati pengguna.

Ulasan pengguna menjadi sumber daya penting bagi pengembang untuk meningkatkan sistem, layanan dan produk mereka (Nandwani & Verma, 2021). Melalui ulasan pengembang dapat mengukur kepuasan pelanggan dan

memantau persaingan pada produk atau layanan serupa. Bagi calon pengguna, ulasan berguna untuk memperoleh informasi lebih lanjut tentang produk sebelum membuat keputusan. Peningkatan jumlah pengguna suatu aplikasi akan berdampak pada volume ulasan yang mayoritasnya berupa data tidak terstruktur. Untuk dapat mengekstrak informasi penting melalui proses manual akan memakan sumber daya yang besar. Oleh karena itu, diperlukan pendekatan yang dapat secara otomatis menganalisis sentimen dari data ulasan, yaitu dengan menggunakan analisis sentimen (Handayanna & Ayu Nur Wulandari, 2024).

Dalam Islam, pentingnya pengolahan informasi dengan cermat juga ditekankan dalam Al-Quran, salah satunya dalam Surah Al-Isra'/17:36, sebagai berikut :

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهِ عِلْمٌ إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَٰئِكَ كَانَ عَنْهُ  
مَسْئُولًا

Terjemahan :

*“Janganlah engkau mengikuti sesuatu yang tidak kau ketahui. Sesungguhnya pendengaran, penglihatan, dan hati nurani, semua itu akan diminta pertanggungjawabannya (Quran NU Online, n.d.).”*

Dari ayat tersebut dapat dipahami bahwa Islam mengajarkan pentingnya dalam verifikasi dan analisis informasi yang cermat agar terhindar dari kesalahan atau bias. Hal tersebut sesuai dengan prinsip ilmiah yang menekankan metode yang tepat dan berbasis data yang valid untuk menghasilkan kesimpulan yang akurat dan dapat dipertanggungjawabkan.

Analisis sentimen atau klasifikasi sentimen merupakan proses mengekstraksi atau mengklasifikasi informasi penting dari opini pengguna untuk menentukan apakah bersifat positif, netral, atau negatif (Onyenwe et al., 2020). Analisis sentimen ini memiliki banyak penerapan seperti untuk memahami opini publik di media sosial, analisis produk, manajemen perusahaan, dan tren pasar saham (Raghunathan & Saravanakumar, 2023). Dalam melakukan analisis sentimen terdiri dari beberapa proses yang perlu dilakukan seperti persiapan data, *preprocessing*, ekstraksi fitur, pemodelan, dan evaluasi (Imelda & Arief Ramdhan Kurnianto, 2023).

Tahapan mengubah data tidak terstruktur menjadi kumpulan data bersih dan terstruktur disebut *preprocessing* (Bordoloi & Biswas, 2023). Proses ini sangat penting agar performa model lebih akurat dalam mengklasifikasikan data. Dengan proses yang tepat, fitur yang tidak relevan akan direduksi data menjadi

terstandarisasi sehingga memudahkan algoritma mengelompokkan data dengan kesalahan prediksi yang rendah. Selain *preprocessing*, terdapat beberapa metode yang dapat meningkatkan performa model salah satunya penggunaan metode ekstraksi fitur seperti TF-IDF.

Term Frequency-Inversion Document Frequency (TF-IDF) adalah metode yang digunakan untuk mengekstraksi fitur. TF-IDF bekerja dengan menghitung nilai frekuensi kata dan frekuensi invers pada dokumen. Menurut Alzami bersama koleganya pada penelitian ulasan Amazon Food menunjukkan bahwa ekstraksi fitur menggunakan TF-IDF lebih unggul dibandingkan menggunakan metode seperti Bag of Words (BoW), Word2Vec dan hibrid (Alzami et al., 2020). Selain penggunaan ekstraksi fitur, penggunaan metode untuk mengatasi ketidakseimbangan kelas data juga perlu ditambahkan.

Distribusi kelas tidak merata dapat memengaruhi proses pelatihan model, menyebabkan bias yang merugikan kelas mayoritas. Dalam mengatasi ketidakseimbangan kelas data, penggunaan metode oversampling diperlukan. Synthetic Minority Oversampling Technique (SMOTE) merupakan metode oversampling yang bekerja dengan mengambil sampel dari kelas minoritas yang kemudian dibuat menjadi sampel sintesis dengan cara menggabungkan antar sampel dari

kelas minoritas, sehingga tercipta sampel baru (Wang et al., 2021). Dalam penelitian yang dilakukan Ardianto bersama koleganya, penggunaan metode tambahan seperti ekstraksi fitur TF-IDF yang digabungkan dengan SMOTE berhasil meningkatkan performa model yang dibuat (Ardianto et al., 2020).

Pada analisis sentimen terdapat sejumlah algoritma klasifikasi yang sering digunakan, yaitu K-Nearest Neighbor (K-NN), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), dan lain sebagainya. Pada studi ini akan membandingkan antara dua algoritma, yakni SVM dan NB. Penggunaan kedua algoritma ini didasarkan pada performa model dalam mengklasifikasikan teks.

SVM merupakan metode *machine learning* yang mengklasifikasikan fitur data ke dalam vektor dengan menentukan garis linear *hyperplane* yang memisahkan data ke dalam kategori dan menghitung jarak terjauh antara *support vector* (Mehra & Choudhury, 2018). SVM cocok digunakan untuk jumlah data yang kecil dengan hasil akurasi yang tinggi. Namun, efisiensi SVM akan berkurang pada klasifikasi dengan data besar karena komputasi yang kompleks untuk menemukan *hyperplane*.

Adapun NB menggunakan probabilitas untuk mengklasifikasikan data berdasarkan *Teorema Bayes* yang

mengasumsikan tidak adanya korelasi antara fitur data. Keunggulan algoritma yang sederhana ini membuatnya cocok untuk mengolah data yang besar. NB dipilih pada analisis sentimen karena mudah diterapkan, tidak memerlukan banyak parameter, komputasi ringan dan akurasi yang cukup tinggi (Alzami et al., 2020).

Ulasan pengguna aplikasi di Play Store telah banyak dikaji pada berbagai studi klasifikasi sentimen khususnya yang menggunakan metode SVM dan NB. Sebagian besar studi tersebut memerlukan pemahaman dan keahlian dalam pemrograman. Berdasarkan informasi tersebut implementasi sistem klasifikasi sentimen pengguna aplikasi di Play Store berbasis website belum banyak dilakukan. Studi ini bertujuan merancang aplikasi web klasifikasi sentimen pengguna aplikasi secara berkelanjutan yang memungkinkan pengguna non-teknis melakukan proses analisis sentimen tanpa menulis kode pemrograman. Aplikasi ini diharapkan dapat memberikan solusi yang efisien untuk analisis sentimen ulasan pengguna, memfasilitasi pengembang aplikasi dalam memahami sentimen pengguna terhadap produk mereka dan membantu peneliti dalam memahami sentimen pengguna aplikasi yang mereka teliti.

## **B. Rumusan Masalah**

1. Bagaimana merancang sistem dan membuat model untuk mengklasifikasikan sentimen pengguna aplikasi di Play Store ?
2. Bagaimana perbandingan performa model dari algoritma SVM dan NB dalam mengklasifikasikan sentimen pengguna aplikasi di Play Store?
3. Bagaimana mengintegrasikan sistem ke dalam aplikasi berbasis web menggunakan Flask?

## **C. Batasan Penelitian**

1. Dataset dalam studi ini diperoleh dari ulasan pengguna aplikasi MyPertamina di Play Store yang tersedia dalam bahasa Indonesia dan dapat diunduh oleh pengguna di Indonesia.
2. Algoritma SVM dan NB digunakan untuk mengklasifikasikan sentimen pada studi ini.
3. Dalam mengoptimalkan performa model akan menggunakan metode TF-IDF dan SMOTE.
4. Model pada penelitian ini dibuat dengan bahasa pemrograman Python.



#### **D. Tujuan Penelitian**

1. Merancang sistem dan membuat model untuk mengklasifikasikan sentimen pengguna aplikasi di Play Store.
2. Membandingkan performa model yang telah dibuat dalam mengklasifikasikan sentimen pengguna aplikasi di Play Store.
3. Mengintegrasikan sistem yang telah dibuat ke dalam aplikasi berbasis website menggunakan Flask.

#### **E. Manfaat Penelitian**

1. Manfaat Teoritis

Studi ini berguna sebagai referensi pada penerapan teknik *preprocessing*, TF-IDF, dan SMOTE untuk meningkatkan performa model klasifikasi sentimen pengguna aplikasi di Google Play Store.

2. Manfaat Praktis

Sebagai alat analisis yang memudahkan pengguna dalam melakukan analisis sentimen dengan menyajikan hasil performa model dan visualisasi data yang mendukung pengambilan keputusan bagi pengembang maupun peneliti.

## **BAB II**

### **LANDASAN PUSTAKA**

#### **A. Analisis Sentimen**

Analisis sentimen erat kaitannya dengan beberapa bidang, seperti komputasi linguistik, pemrosesan bahasa alami (NLP), penambangan teks, dan analisis teks. Analisis sentimen adalah disiplin ilmu dari pemrosesan bahasa alami (NLP) yang mengkaji persepsi, perasaan dan emosi seseorang (Talaat, 2023) dari berbagai modalitas seperti teks, gambar, dan suara (Zhao et al., 2024). Analisis sentimen atau penambangan pendapat adalah tahapan menganalisis dan mengelompokkan sentimen pengguna sebagai positif, netral, atau negatif (Bordoloi & Biswas, 2023). Tujuan dari analisis sentimen untuk memahami emosi yang tersirat dalam informasi yang tidak terstruktur (Raghunathan & Saravanakumar, 2023).

Berdasarkan level sentimennya, analisis sentimen dapat dibedakan menjadi tiga, yaitu level aspek, level kalimat, dan level dokumen (Zhao et al., 2024). Pada level aspek setiap aspek yang muncul dalam kalimat akan diberikan nilai polaritas yang kemudian nilai-nilai tersebut digabungkan untuk menentukan sentimen keseluruhan kalimat (Wankhade et al., 2022). Adapun pada level kalimat polaritas ditentukan sesuai dengan hasil analisa setiap

kalimat. Terakhir pada level dokumen setiap dokumen dianggap hanya mewakili satu polaritas. Pada penelitian ini, analisis sentimen akan berfokus pada level dokumen dengan pendekatan berbasis machine learning.

Analisis sentimen memiliki dua pendekatan yang umum digunakan, yaitu pendekatan berbasis pembelajaran mesin (ML) dan pendekatan berbasis leksikal. Pendekatan berbasis pembelajaran mesin memungkinkan sistem untuk mempelajari pola pada dataset yang diberikan. Salah satu metode berbasis pembelajaran mesin seperti *supervised learning*, metode ini membutuhkan data berlabel untuk melatih model yang kemudian akan diuji untuk mengetahui performa model tersebut (Raghuathan & Saravanakumar, 2023). Berbeda dengan pendekatan sebelumnya, pendekatan leksikal menggunakan kamus yang berisi kata-kata dengan nilai sentimen yang telah ditentukan berdasarkan polaritasnya. Nilai-nilai ini akan dijumlahkan atau dirata-ratakan untuk mendapatkan sentimen keseluruhan dari kalimat atau dokumen (Nandwani & Verma, 2021). Perbedaan antara kedua pendekatan tersebut terletak pada cara kerjanya, metode pembelajaran mesin memerlukan proses training untuk mencari pola data, sedangkan pendekatan leksikal akan menyesuaikan teks yang ada dengan kamus dengan nilai sentimen yang telah ditentukan.

## **B. *Preprocessing***

Proses mengubah data tidak terstruktur menjadi kumpulan data yang bersih dan terstruktur disebut *preprocessing* (Bordoloi & Biswas, 2023). Terdapat enam tahapan yang harus dijalankan pada proses ini yang meliputi *case folding*, *cleansing*, normalisasi, *tokenization*, *stopword removal*, dan *lemmatization*. Proses *preprocessing* yang tepat, dapat meningkatkan performa dari model analisis sentimen. Dalam penelitian ini tahap *preprocessing* memiliki sejumlah proses, yakni :

### **1. *Case Folding***

Pada proses ini, seluruh ulasan akan diubah ke huruf kecil (*lowercase*) agar memiliki format penulisan yang konsisten sehingga terhindar dari duplikat kata.

### **2. *Cleansing***

Pada proses ini, ulasan akan dibersihkan dari bilangan angka, tanda baca, simbol, dan URL.

### **3. Normalisasi**

Pada proses ini, bahasa gaul (*slang*) dan kata yang tidak baku akan diubah ke bentuk baku yang sesuai dengan aturan KBBI.

#### 4. *Tokenization*

Pada proses ini, teks akan dipecah menjadi unit-unit kata dengan menghilangkan spasi. Hasil dari proses ini dikenal sebagai token.

#### 5. *Stopword Removal*

Pada proses ini, kata-kata yang sering muncul tapi tidak memiliki informasi penting seperti kata hubung, kata ganti, kata tanya dan lain sebagainya akan dihapus.

#### 6. *Lemmatization*

Pada tahap ini, seluruh kata yang berimbuhan dan kata perulangan pada kalimat akan diubah ke bentuk dasarnya.

### **C. TF-IDF**

Term Frequency-Inversion Document Frequency (TF-IDF) merupakan metode pembobotan yang memberikan bobot untuk setiap kata pada sebuah dokumen (Putra et al., 2024). TF-IDF umum digunakan dalam ekstraksi fitur. Metode ini mengubah teks ke dalam matrix yang dimana setiap angka mempresentasikan seberapa pentingnya informasi yang ada pada fitur tersebut dalam suatu dokumen (Nandwani & Verma, 2021). TF-IDF bertujuan untuk menentukan seberapa relevannya data di dalam dokumen dengan pembobotan kata . Pada TF sebuah kata

dianggap penting jika sering muncul dalam dokumen. Sebaliknya, IDF menilai semakin jarang kata muncul pada dokumen lain maka semakin tinggi tingkat kepentingannya. TF-IDF dapat dihitung menggunakan persamaan 2.1 berikut :

$$TFIDF(d, t) = TF(d, t) \cdot IDF(t) \quad (2.1)$$

$$TF(d, t) = \frac{\text{jumlah kata } t \text{ pada dokumen } d}{\text{total kata pada dokumen } d} \quad (2.2)$$

$$IDF(t) = \log \frac{\text{total dokumen}}{\text{jumlah dokumen yang mengandung kata } t} \quad (2.3)$$

Keterangan :

$TF-IDF(d, t)$  = Nilai  $TF-IDF$  untuk kata  $t$  pada dokumen  $d$

$TF(d, t)$  = Frekuensi relatif kemunculan kata  $t$  pada dokumen  $d$

$IDF(t)$  = Logaritma dari kebalikan frekuensi dokumen yang mengandung kata  $t$ , dihitung berdasarkan seluruh korpus dokumen

#### D. SMOTE

Synthetic Minority Oversampling Technique (SMOTE) merupakan metode oversampling diperkenalkan oleh

Chawla et al. (2002). SMOTE meningkatkan jumlah kelas minoritas dengan membuat sampel sintetis dari kelas tersebut. Pengambilan sampel kelas dilakukan secara acak dari  $k$  tetangga terdekat dari suatu sampel yang dilanjutkan dengan menambahkan data baru di sepanjang jalur linear yang menggabungkan sampel tersebut dengan tetangga terpilih. Tujuan dari oversampling ini adalah untuk meratakan distribusi kelas tanpa menghilangkan informasi seperti undersampling (Utami, 2022). Proses sintesis data mengacu pada persamaan 2.4.

$$x_{new} = x_i + (x_j | x_i) \delta \quad (2.4)$$

Keterangan :

$x_{new}$  = Data sintesis yang diciptakan

$x_i$  = Sampel kelas minoritas yang menjadi oversampling

$x_j$  = Data terdekat dari sampel minoritas

$\delta$  = Bilangan acak di rentang 0 dan 1

## E. SVM

Support Vector Machine (SVM) merupakan metode *supervised learning* berbasis *kernel* yang umum digunakan untuk klasifikasi dan regresi (Dimas Sulistiyo & Shihab, 2023; Prastyo et al., 2020). Metode ini diperkenalkan pertama kali oleh Vapnic yang digunakan untuk membuat

*hyperplane* yang memisahkan kelas data dalam ruang dimensi tertentu (Kusumaningrum et al., 2021). Tujuan dari metode ini adalah untuk menemukan *hyperplane* terbaik yang memaksimalkan jarak margin (Hasanli & Rustamov, 2019). Secara sederhana, metode ini merepresentasikan data dalam bentuk vektor, kemudian membangun *hyperplane* sebagai pemisah antar kelas dan memilih *hyperplane* terbaik dengan margin maksimal untuk pemisahan yang optimal.

Penggunaan SVM populer dalam klasifikasi karena performa generalisasi yang baik pada data pelatihan yang terbatas. Disisi lain, efisiensi metode ini akan berkurang pada klasifikasi dengan data besar karena komputasi yang kompleks untuk menemukan *hyperplane* .

#### **F. Naive Bayes**

Naïve Bayes (NB) menggunakan probabilitas untuk mengklasifikasikan data berdasarkan Teorema Bayes yang mengansumsikan tidak adanya korelasi antara fitur data. NB bekerja dengan mengklasifikasikan data pada kategori yang sesuai dengan menggunakan nilai probabilitas tertinggi. Keunggulan algoritma yang sederhana ini membuatnya cocok untuk mengalah data yang besar. *NB* dipilih pada analisis sentimen karena mudah diterapkan, tidak memerlukan banyak parameter, komputasi ringan



dan akurasi yang cukup tinggi (Sabilly & Agung Nugroho, 2023). NB dapat dihitung menggunakan persamaan 2.7.

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)} \quad (2.7)$$

Keterangan :

H = Hipotesis data E dari kelas spesifik

E = Data dengan kelas yang belum diketahui

$P(H|E)$  = Probabilitas hipotesis H berdasarkan kondisi E (*Posterior*)

$P(E|H)$  = Probabilitas E berdasarkan kondisi tersebut (*Likelihood*)

$P(H)$  = Probabilitas hipotesis H (*Prior*)

$P(E)$  = Probabilitas dari E (*Evidance*)

## G. Confusion Matrix

Confusion Matrix adalah metode evaluasi model untuk mengukur performa dari model yang dihasilkan (Sabilly & Agung Nugroho, 2023). Metode ini memvisualisasikan performa model dengan komponen seperti *accuracy*, *precision*, *recall*, dan *F1-score* yang memberikan gambaran komprehensif tentang kemampuan model (Madyatmadja et al., 2025). Adapun confusion matrix ditunjukkan pada tabel 2.1.

**Tabel 2.1** *Confusion Matrix*

		<i>Actual class</i>	
		P	N
<i>Predicted class</i>	P	TP	FP
	N	FN	TN

Keterangan:

TP (*True Positive*) = data positif yang diprediksi sebagai positif

FP (*False Positive*) = data negatif yang salah diprediksi sebagai positif

TN (*True Negative*) = data negatif yang diprediksi sebagai negatif

FN (*False Negative*) = data positif yang salah diprediksi sebagai data negatif

Berdasarkan nilai tersebut dapat dihitung performa model dengan *accuracy*, *precision*, *recall*, dan *F1-score*. *Accuracy* mengukur tingkat keakuratan model dengan prediksi yang benar. *Accuracy* dapat dihitung menggunakan persamaan 2.7.

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \times 100\% \quad (2.7)$$

*Precision* mengukur tingkat akurasi yang ada antara informasi yang diminta dengan respons sistem. *Precision* dapat dihitung menggunakan persamaan 2.8.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (2.8)$$

*Recall* mengukur tingkat akurasi semua informasi terkait yang telah diklasifikasikan dengan benar (Raghunathan & Saravanakumar, 2023). *Recall* dapat dihitung menggunakan persamaan 2.9.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.9)$$

*F1-score* merupakan parameter tunggal untuk mengevaluasi kinerja model secara seimbang dengan menggabungkan *precision* dan *recall*. *F1-score* dapat dihitung menggunakan persamaan 2.10.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.10)$$

## H. Flask

Flask adalah *microframework* yang menggunakan bahasa pemrograman Python untuk membuat aplikasi berbasis web (Zainudin & Yunant, 2019). Flask pertama kali diperkenalkan oleh Armin Ronacher pada 2004. Selain digunakan untuk membuat aplikasi berbasis web, Flask juga digunakan untuk membuat REST API. Flask cocok untuk pembuatan aplikasi berbasis web yang cepat dan mudah, hal ini dikarenakan fleksibilitas dan kemudahan

yang ditawarkan oleh framework ini. Pengembang dapat membuat user interface menggunakan *Jinja*, handle input dari user, dan berkomunikasi dengan model machine learning dengan lancar. Struktur Flask yang modular juga memudahkan untuk mengintegrasikan komponen lain seperti database atau layanan external lainnya untuk meningkatkan fungsi aplikasi.

Dalam penelitian ini, Integrasi machine learning dengan Flask dapat menciptakan user interface yang memungkinkan pengguna melakukan proses analisis sentimen dengan lebih mudah. Hal ini juga dapat mempersingkat waktu yang digunakan untuk melakukan analisa sentimen karena pada aplikasi yang dibuat seluruh proses dilakukan dalam satu aplikasi mulai dari proses pengumpulan data hingga pengujian model.

## **I. Kajian Penelitian yang Relevan**

Pada penelitian yang dilakukan oleh Madyatmadja dan koleganya menggunakan beberapa metode, yakni NB, SVM, dan RF dengan membandingkan metode mana yang paling efektif dalam mengklasifikasikan sentimen pengguna pada aplikasi Snapchat. Pada penelitian tersebut ditemukan ketidakseimbangan kelas pada dataset yang digunakan yang mana kelas positif lebih banyak daripada kelas negatif dan didapat hasil akurasi tertinggi menggunakan metode

RF sebesar 83% yang diikuti SVM sebesar 81% dan NB 80%.

Penelitian lain dilakukan oleh Ardianto dan koleganya pada 2020, meneliti tentang pendidikan *e-sport* di Indonesia melalui Twitter. Penelitian ini membandingkan dua algoritma klasifikasi NB dengan SVM yang dipadukan metode TF-IDF untuk ekstraksi fitur dan SMOTE untuk penanganan ketidakseimbangan data. Pada penelitian ini, penggunaan metode SMOTE berhasil meningkatkan akurasi dari model NB sebesar 19,58 % yang awalnya 50,74% menjadi 70.32% dan pada model SVM akurasi yang didapat sebesar 66,92%.

Penelitian berikutnya yang dilakukan oleh Mustasaruddin dan koleganya meneliti tentang klasifikasi sentimen pengguna aplikasi MyPertamina di Play Store. Pada penelitian tersebut menggunakan pendekatan leksikal dalam mengklasifikasikan ulasan dengan algoritma SVM serta metode undersampling dan FastText. Berdasarkan hasil uji yang dilakukan didapat hasil tertinggi akurasi sebesar 80% pada rasio split data 90:10 tanpa penggunaan undersampling, sedangkan penggunaan undersampling didapat penurunan akurasi menjadi 67% yang terjadi karena reduksi data pada kelas mayoritas yang menghilangkan informasi pada data latih.

Penelitian selanjutnya dilakukan oleh Bahtiar dan koleganya yang bereksperimen dengan metode pelabelan berdasarkan rating ulasan pada empat aplikasi *Marketplace* yaitu Shopee, Tokopedia, Lazada dan Blibli. Peneliti membandingkan hasil akurasi dan waktu proses algoritma NB dengan Logistic Regresion. Berdasarkan penelitian tersebut didapatkan penggunaan dua label (positif dan negatif) memiliki performa lebih baik dibanding menggunakan tiga label (positif, netral dan negatif) dengan akurasi terbaik sebesar 84,58% pada Logistic Regresion menggunakan dua label dan waktu pemrosesan NB lebih cepat yaitu dibawah 0.067 detik pada semua pengujian dibandingkan Logistic Regresion dengan waktu tercepatnya 0.47 detik.

**Tabel 2.2** *Kajian Penelitian*

Peneliti	Metode	Akurasi	Gap
Madyatmadja et al. (2025)	NB	80	Belum menggunakan metode untuk penanganan ketidakseimbangan data
	SVM	81	
	RF	83	
Ardianto et al. (2020)	NB + SMOTE	70	Pada <i>preprocessing</i> belum ada tahap normalisasi
	SVM + SMOTE	67	
Mustasaruddin et al. (2023)	SVM	80	Metode penanganan ketidakseimbangan data yang
	SVM+ Undersampling	67	

			digunakan belum berhasil meningkatkan performa model
Bahtiar et al. (2023)	NB 2 label	84	Pada <i>preprocessing</i> belum ada tahap normalisasi
	NB 3 label	71	
	LR 2 label	85	
	LR 3 label	73	

Berdasarkan tabel 2.2 dapat dilihat gap dari penelitian sebelumnya, yaitu penggunaan metode oversampling berhasil meningkatkan performa model, sementara penggunaan metode undersampling belum berhasil karena menurunkan performa model. Hal ini menunjukkan pemilihan metode penanganan ketidakseimbangan data yang tepat diperlukan. Selanjutnya, optimalisasi proses *preprocessing* belum dilakukan dengan normalisasi data dari kata slang. Terakhir belum adanya implementasi sistem yang dibuat tersebut ke dalam aplikasi berbasis web untuk memudahkan pengguna dalam melakukan analisis sentimen ulasan aplikasi secara berkelanjutan.

Selain mengidentifikasi gap penelitian, kajian tersebut juga memberikan wawasan kepada peneliti bahwa penggunaan metode *supervised learning* seperti SVM dan NB menunjukkan performa yang cukup baik dalam mengklasifikasikan sentimen. Penggunaan metode tambahan untuk menangani ketidakseimbangan data

seperti SMOTE dan undersampling memberikan dampak yang cukup signifikan pada performa model. Pendekatan lain dalam pelabelan ulasan berdasarkan rating dapat menjadi metode alternatif pada dataset yang besar.





## BAB III

### METODOLOGI PENELITIAN

#### A. Data dan Sumber Data

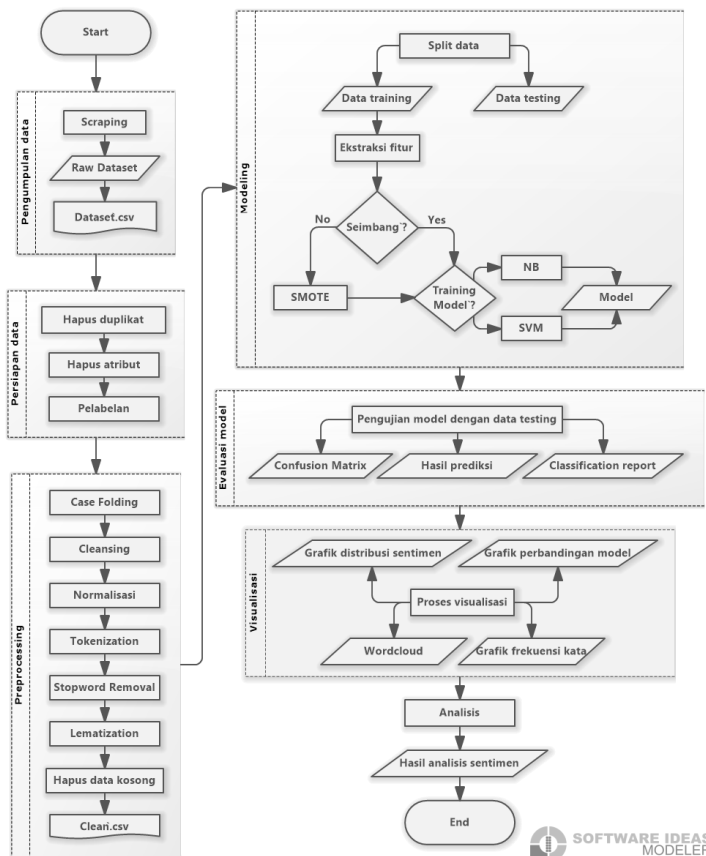
Data yang akan digunakan merupakan data Penelitian ini diambil dari API Google Play Store menggunakan library `google_play_scraper`. Dataset yang akan di uji menggunakan aplikasi yang tersedia di Play Store, yaitu aplikasi MyPertamina dengan atribut yang meliputi username, tanggal, ulasan, rating, dan versi. Proses scraping menggunakan parameter yang ditampilkan pada tabel 3.1.

***Tabel 3.1 Parameter Scraping***

Parameter	Keterangan	Nilai
appId	ID aplikasi pada Google Play Store	com.dafturn.my Pertamina
lang	Bahasa dari metadata yang diambil	id
country	Region aplikasi yang diambil	id
sort	Menyaring berdasarkan data terbaru	Newest
count	Jumlah ulasan yang diambil	14000

## B. Metode Analisis Data

Pada tahapan ini akan dilakukan proses analisis sentimen menggunakan algoritma SVM, NB, TF-IDF dan SMOTE. Untuk alur analisis sentimen ditunjukkan pada gambar 3.1.



**Gambar 3. 1** Flowchart Alur Analisis Sentimen

Berikut penjelasan terkait alur sistem analisis sentimen yang dilakukan :

### 1. Pengumpulan Data

Pada tahap ini, data akan diambil melalui proses *scraping* dengan menggunakan appId dan jumlah ulasan yang ingin diambil dari aplikasi. Data yang diambil melalui proses *scraping* ini adalah ulasan terbaru sesuai dengan jumlah *request* yang diberikan. Hasil dari proses ini berupa dataframe yang akan diubah ke format CSV untuk disimpan dan diproses ketahap berikutnya. Penggunaan *scraping* ini bergantung pada API Google Play Store aplikasi dengan rating rendah atau jumlah pengguna terbatas sering kali tidak tersedia melalui API publik.

### 2. Persiapan Data

Pada tahap ini, file CSV akan dibaca dan diubah menjadi dataframe, kemudian dilakukan pembersihan dari duplikat data, dilanjutkan dengan penghapusan atribut yang tidak digunakan seperti username, tanggal dan versi. Pada tahap terakhir data akan dilakukan pelabelan berdasarkan rating. Penggunaan metode pelabelan ini didasarkan pada efisiensi dan fleksibilitas. Dengan volume ulasan yang mencapai 14.000, pengolahan manual akan sangat memakan

waktu, sehingga metode otomatis ini mampu mempercepat proses secara signifikan. Selain itu, metode pelabelan tersebut dapat diintegrasikan secara konsisten ke dalam sistem analisis sentimen berbasis web yang siap diterapkan pada berbagai aplikasi di Play Store. Proses pelabelan dapat dilihat pada persamaan 3.1 berikut.

$$L(r) = \begin{cases} \text{"negatif"}, & r < 3, \\ \text{"netral"}, & r = 3, \\ \text{"positif"}, & r > 3. \end{cases} \quad (3.1)$$

Keterangan:

$L(r)$  = Label ulasan berdasarkan rating yang diberikan

$r$  = Nilai rating ulasan pengguna

### 3. *Preprocessing*

Pada bagian ini, data yang sudah dipersiapkan sebelumnya akan diproses untuk membersihkan dari *noise* agar data terstruktur.

#### a. *Case Folding*

Pada proses ini ulasan akan diubah ke bentuk huruf kecil (*lowercase*) tujuannya menyelaraskan format data agar tidak terjadi duplikasi fitur. Dalam tabel 3.2 menampilkan contoh penerapan proses *case folding*.

**Tabel 3.2** Penerapan Case Folding

Teks awal	Hasil
Apk keren banget memudahkan kita untuk membeli BBM menggunakan uang elektronik Tetapi sayang nya banyak SPBU Pertamina yang menolak pembayaran non tunai.. Alasan alat lagi rusak	apk keren banget memudahkan kita untuk membeli bbm menggunakan uang elektronik tetapi sayang nya banyak spbu pertamina yang menolak pembayaran non tunai.. alasan alat lagi rusak

b. *Cleansing*

Pada proses ini ulasan akan dibersihkan dari bilangan angka, tanda baca, simbol, dan URL. Dalam tabel 3.3 menampilkan contoh penerapan proses *cleansing*.

**Tabel 3.3** Penerapan Cleansing

Teks awal	Hasil
Apk keren banget memudahkan kita untuk membeli BBM menggunakan uang elektronik Tetapi sayang nya banyak SPBU Pertamina yang menolak pembayaran non tunai.. Alasan alat lagi rusak	apk keren banget memudahkan kita untuk membeli bbm menggunakan uang elektronik tetapi sayang nya banyak spbu pertamina yang menolak pembayaran non tunai alasan alat lagi rusak

c. Normalisasi

Pada proses ini bahasa gaul (slang) atau kata tidak baku akan di ubah ke bentuk yang sesuai dengan kaidah KBBI. Tujuannya agar model dapat lebih mudah mengenali pola data dan menghindari duplikat fitur. Contoh yang sering ditemukan seperti “apk”, “app” diubah menjadi “aplikasi”. Dalam tabel 3.4 menampilkan contoh penerapan proses normalisasi.

**Tabel 3.4** Penerapan Normalisasi

Teks awal	Hasil
Apk keren banget memudahkan kita untuk membeli BBM menggunakan uang elektronik Tetapi sayang nya banyak SPBU Pertamina yang menolak pembayaran non tunai.. Alasan alat lagi rusak	aplikasi keren banget memudahkan kita untuk membeli bbm menggunakan uang elektronik tetapi sayang banyak spbu pertamina yang menolak pembayaran non tunai alasan alat lagi rusak

d. *Tokenization*

Pada proses ini, teks akan dipecah menjadi unit-unit kata dengan menghilangkan spasi. Hasil dari proses ini berupa kumpulan token yang telah diberi indeks. Dalam tabel 3.5 menampilkan contoh penerapan proses *tokenization*.

**Tabel 3.5** Penerapan Tokenization

Teks awal	Hasil
Apk keren banget memudahkan kita untuk membeli BBM menggunakan uang elektronik Tetapi sayang nya banyak SPBU Pertamina yang menolak pembayaran non tunai.. Alasan alat lagi rusak	['aplikasi', 'keren', 'banget', 'mudah','beli', 'bbm', 'uang', 'elektronik', 'sayang', 'spbu', 'pertamina', 'tolak', 'bayar', 'non', 'tunai', 'alasan', 'alat', 'rusak']

e. *Stopword Removal*

Pada proses ini kata yang sering muncul dan tidak memiliki informasi penting seperti kata hubung, konjungsi akan dihapus untuk mengurangi *noise* pada data. Penerapan *stopword removal* dapat dilihat pada tabel 3.6.

**Tabel 3.6** Penerapan Stopword Removal

Teks awal	Hasil
Apk keren banget memudahkan kita untuk membeli BBM menggunakan uang elektronik Tetapi sayang nya banyak SPBU Pertamina yang menolak pembayaran non tunai.. Alasan alat lagi rusak	aplikasi keren banget memudahkan membeli bbm uang elektronik sayang spbu pertamina menolak pembayaran non tunai alasan alat rusak



f. *Lemmatization*

Pada proses ini semua kata yang berimbuhan akan diubah ke bentuk dasarnya dan menghapus emoji atau emoticon. Penerapan *lemmatization* dapat dilihat pada tabel 3.7.

**Tabel 3.7** Penerapan Lemmatization

Teks awal	Hasil
Apk keren banget memudahkan kita untuk membeli BBM menggunakan uang elektronik Tetapi sayang nya banyak SPBU Pertamina yang menolak pembayaran non tunai.. Alasan alat lagi rusak	aplikasi keren banget mudah beli bbm uang elektronik sayang spbu pertamina tolak bayar non tunai alasan alat rusak

g. Hapus data kosong

Setelah data selesai melewati proses *preprocessing* selanjutnya data akan dicek apakah terdapat data yang kosong dan jika ditemukan data tersebut akan dihapus. Data kosong ini berasal dari ulasan yang hanya mengandung *noise* seperti emoji, simbol, dan lain sebagainya.

#### 4. Modeling

Pada proses ini data ulasan dengan sentimen netral akan didrop, sehingga klasifikasi hanya melibatkan dua kelas sentimen, yakni positif dan negatif. Data yang telah dibersihkan akan dibagi menjadi dua, yaitu data *training* dan data *testing* dengan rasio 80:20. Ulasan pada data training akan diekstraksi dengan melakukan pembobotan setiap kata dalam dokumen menggunakan TF-IDF. Proses ini membantu model untuk menentukan fitur yang mengandung informasi penting dan mengurangi bobot pada fitur yang umum dalam dokumen. Selanjutnya, grafik distribusi sentimen pada data *training* akan ditampilkan untuk mengetahui sebaran sentimen dan jika kelas data tidak seimbang dapat dilakukan oversampling menggunakan SMOTE untuk meratakan distribusi data. Pembagian data dan pemilihan data pada SMOTE dilakukan secara acak dengan menetapkan parameter *random state* sebesar 42 untuk memastikan hasil konsisten pada setiap kali eksekusi. Proses pelatihan model akan menggunakan dua algoritma klasifikasi, yaitu SVM dan NB yang dilatih menggunakan data *training* yang sudah disiapkan sebelumnya.

## 5. Evaluasi Model

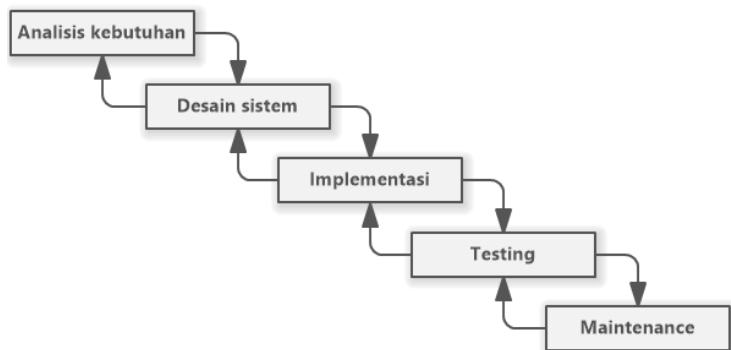
Pada tahap ini, model yang telah dibuat sebelumnya akan dites menggunakan data testing. Hasilnya dari proses ini berupa confusion matrix dalam bentuk gambar dan laporan klasifikasi berupa *accuracy*, *precision*, *recall*, dan *f1-score* yang akan disimpan pada hasil modeling. Hasil uji ini dapat menjadi indikator performa dari model yang dihasilkan.

## 6. Visualisasi

Pada tahap ini, hasil dari proses sebelumnya akan di visualisasikan. Pada visualisasi data, sentimen akan divisualisasikan menggunakan grafik distribusi sentimen, grafik frekuensi kata dan *wordcloud*. Pada grafik distribusi sentimen menampilkan presentase sebaran kelas sentimen, grafik frekuensi kata akan berisi sepuluh kata yang sering muncul pada setiap kelas sentimen, sedangkan pada *wordcloud* seluruh kata pada kelas sentimen tersebut ditampilkan dengan ukuran kata yang menyesuaikan dengan frekuensi kata tersebut. Pada visualisasi model selain menampilkan confusion matrix, hasil model akan dibandingkan dalam grafik.

### C. Perancangan Sistem

Untuk mengintergrasikan sistem analisis sentimen yang telah dibuat menjadi aplikasi berbasis website diperlukan perancangan sistem. Metode perancangan sistem yang akan digunakan dalam penelitian ini adalah metode *Waterfall*. Menurut Pressman & Maxim (2015) metode ini merupakan model tertua dalam pengembangan perangkat lunak yang sequential dan sistematis. Metode ini sering disebut juga sebagai *classic life cycle*. Proses metode ini memiliki tahap seperti gambar 3.2.



**Gambar 3.2** Tahapan Metode Waterfall

#### 1. Analisis Kebutuhan

Pada tahap ini kebutuhan aplikasi akan dirumuskan secara rinci dengan batasan dan tujuan yang telah ditetapkan. Hal ini penting agar aplikasi memiliki spesifikasi yang jelas dan tidak melenceng

dari tujuan awal. Analisis kebutuhan dapat dibagi menjadi dua yaitu sebagai berikut :

a. Analisis Kebutuhan Fungsional

Kebutuhan fungsional menguraikan aktivitas yang akan dilakukan dalam suatu sistem dan merinci kondisi yang diperlukan agar sistem dapat beroperasi secara efektif dan memenuhi tujuan yang dimaksudkan. Adapun kebutuhan fungsional pada sistem analisis sentimen ulasan aplikasi sebagai berikut :

- 1) Sistem memiliki fitur Login dan Logout.
- 2) Sistem dapat menambahkan dataset ulasan atau scraping data ulasan.
- 3) Sistem dapat mengolah atau membersihkan data ulasan dari *noise* sebelum digunakan untuk pelatihan model.
- 4) Sistem dapat mengklasifikasikan data ulasan menggunakan metode SVM, NB dan SMOTE.
- 5) Sistem dapat memberikan visualisasi data dan perbandingan performa dari metode yang digunakan.

b. Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional pada penelitian ini meliputi perangkat keras dan perangkat lunak

untuk mendukung proses penelitian. Berikut perangkat yang digunakan dalam penelitian ini :

***Tabel 3.8 Kebutuhan Perangkat Keras***

No	Hardware	Spesifikasi
1	Device	Swift x SFX14-41G
2	Processor	AMD R 5600U
3	Memory	16 GB
4	Display	14 inch
5	Mouse & Keyboard	Normal

***Tabel 3.9 Kebutuhan Perangkat Lunak***

No	Software	Spesifikasi
1	IDE	VS Code
2	PHP, Web server, dan Database server	XAMPP
3	UI Frameworks	Bootstrap
4	Design tools	Figma
5	Web Frameworks	Flask
6	Sistem Operasi	Windows 11

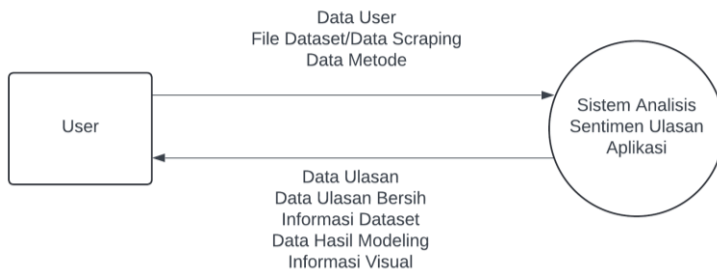
## 2. Desain Sistem

Pada tahap ini rancangan antarmuka pengguna, alur data dan komponen lainya dikumpulkan berdasarkan kebutuhan. Hal ini bertujuan untuk menguraikan langkah-langkah yang harus diambil dan mengilustrasikan bagaimana tampilan sistem yang dimaksud. Pada penelitian ini, beberapa desain yang

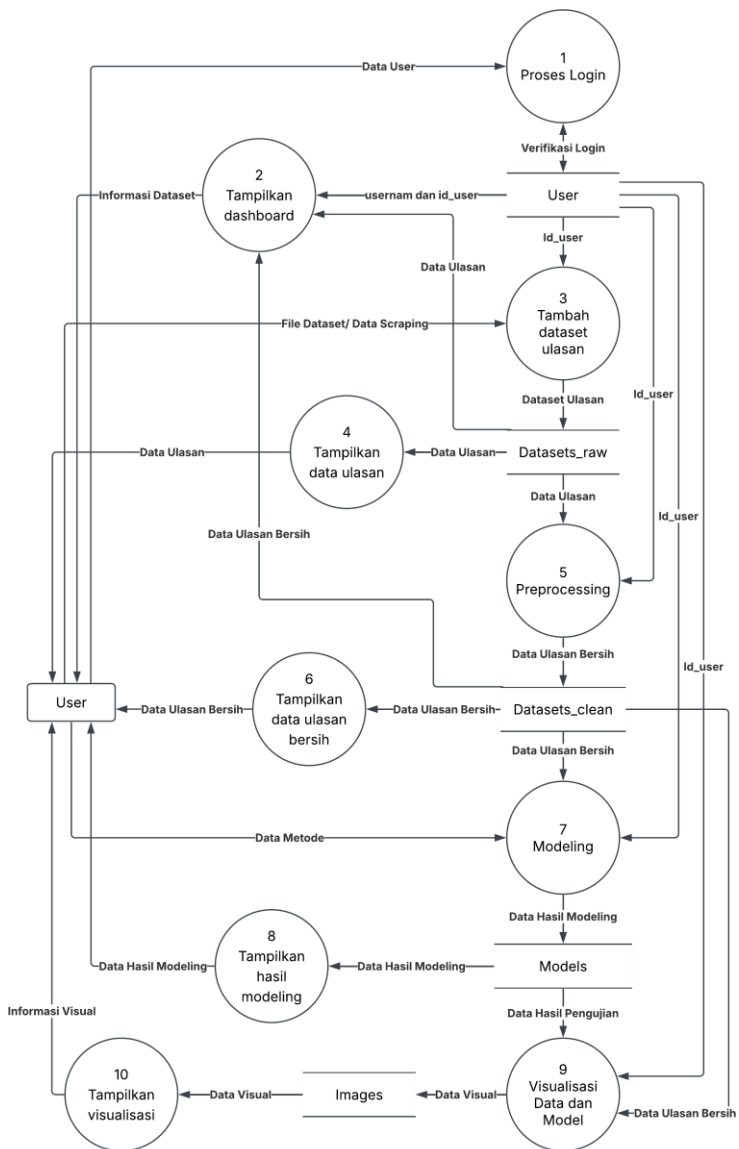
sudah dibuat meliputi *Data Flow Diagram*, *User Interface* dan *Entity Relationship Diagram*.

a. *Data Flow Diagram (DFD)*

Data Flow Diagram merupakan alat untuk memvisualisasikan aliran data yang ada pada suatu sistem (Chong & Diamantopoulos, 2020). Tujuan dari DFD ini untuk memudahkan dalam memahami proses dan aliran data dalam suatu sistem. Pada penelitian ini aliran data akan divisualisasikan dengan dua level DFD, yaitu level 0 dan level 1. Dalam DFD level 0 akan menggambarkan alur data sistem sebagai satu proses. Berikut pada gambar 3.3 yang menampilkan DFD level 0.



**Gambar 3.3** DFD Level 0



**Gambar 3.4 DFD Level 1**



Pada gambar 3.4 menampilkan DFD level 1 yang memvisualisasikan sistem ke dalam proses yang lebih detail. User akan memberikan input berupa data user, data ulasan dan data metode. Berikut penjelasan terkait proses pada DFD level 1, yakni:

- 1) Memasukkan data user untuk diverifikasi oleh sistem, setelah berhasil diverifikasi `id_user` dan `username` akan disimpan sebagai session dan akan dialihkan ke halaman *dashboard*.
- 2) Menampilkan halaman *dashboard* yang berisi informasi dataset meliputi jumlah sentimen setimen setiap kelas, jumlah data ulasan, jumlah data ulasan bersih, grafik distribusi rating, dan grafik distribusi sentimen.
- 3) Memasukkan data ulasan melalui input file CSV atau *scraping* data pada halaman dataset. Data yang telah dimasukkan akan disimpan ke dalam folder `datasets_raw` dengan menggunakan `id_user` sebagai awalan nama file CSV.

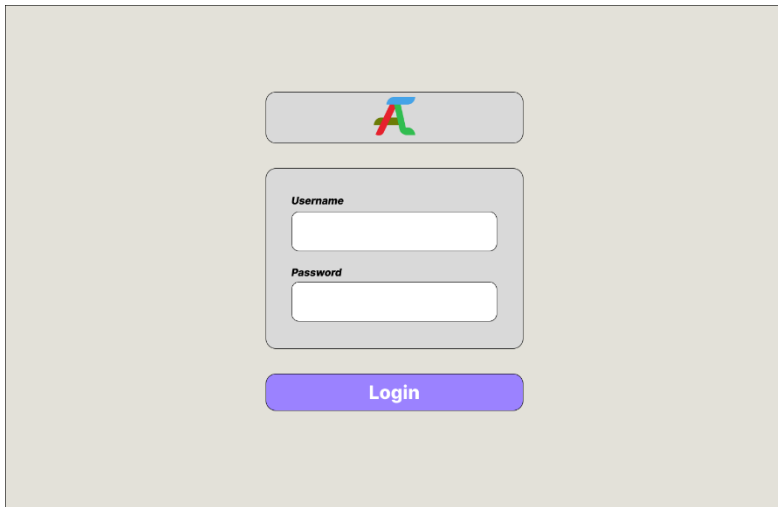
- 4) Menampilkan halaman data ulasan dalam bentuk tabel pada halaman dataset.
- 5) Melakukan *preprocessing* dengan membaca file data ulasan yang disimpan. Data hasil proses ini akan disimpan ke dalam folder `datasets_clean` dengan menggunakan `id_user` sebagai awalan nama file CSV.
- 6) Menampilkan data ulasan bersih dalam bentuk tabel pada halaman *preprocessing*.
- 7) Memasukkan data metode pada halaman modeling untuk proses modeling dengan menggunakan data ulasan bersih yang disimpan. Pada proses ini akan menghasilkan data hasil modeling.
- 8) Menampilkan data hasil modeling pada halaman modeling.
- 9) Melakukan visualisasi dari data ulasan bersih dan data hasil pengujian. Data visual yang berupa gambar akan disimpan pada folder `images` dengan menggunakan `id_user` sebagai awalan nama file gambar.
- 10) Menampilkan informasi visual pada halaman summary dari data visual yang telah disimpan sebelumnya.

b. *User Interface*

*User interface* memberikan gambaran tentang tampilan pengguna yang nantinya digunakan sebagai patokan dalam proses development aplikasi. Desain tampilan yang akan dibuat dalam penelitian ini yang meliputi halaman login, *dashboard*, *preprocessing*, modeling, dan *summary*.

1) Halaman Login

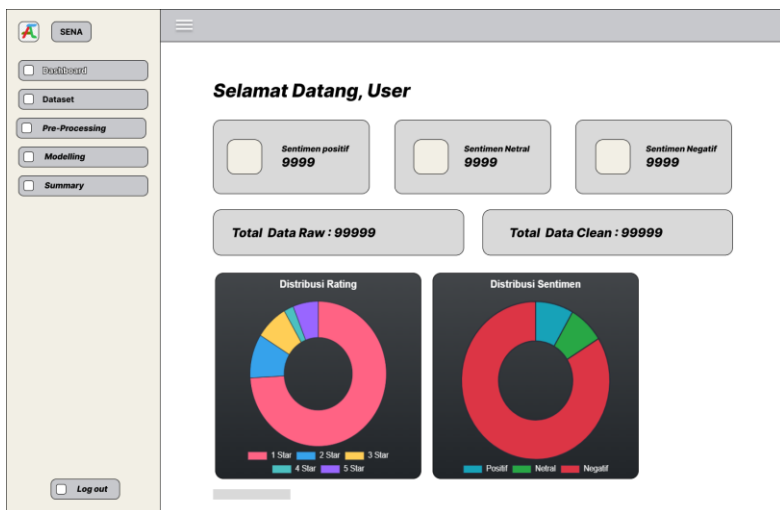
Pada halaman ini, user akan memasukan data user untuk bisa masuk ke *dashboard* aplikasi.

The image shows a login page with a light gray background. At the top center is a rounded rectangle containing a stylized 'A' logo with red, green, and blue segments. Below this is a gray-bordered box containing two white input fields. The first field is labeled 'Username' and the second is labeled 'Password'. Below the input fields is a purple rounded button with the text 'Login' in white.

**Gambar 3.5** Halaman Login

## 2) Halaman *Dashboard*

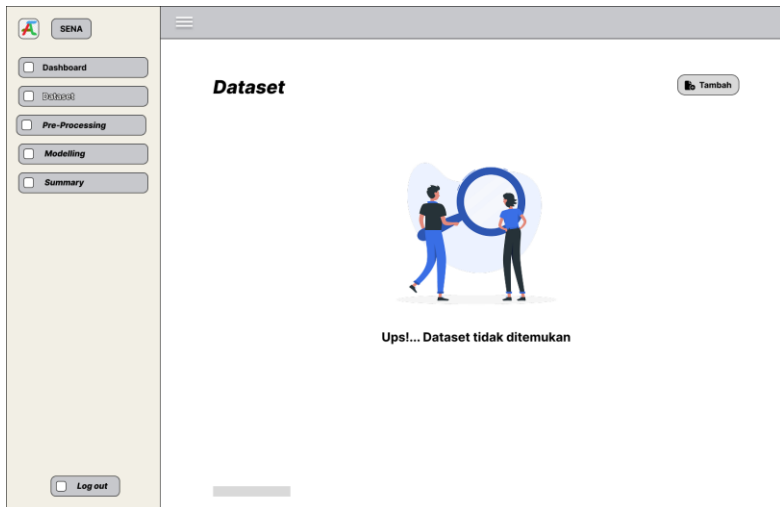
Pada halaman ini akan menampilkan username pengguna dan informasi dari dataset yang telah diproses meliputi jumlah sentimen setiap kelas, jumlah dataset awal, jumlah data ulasan bersih, grafik distribusi rating, dan grafik distribusi sentimen dari ulasan aplikasi. Berikut tampilan halaman *dashboard* yang dapat dilihat pada gambar 3.6.



**Gambar 3.6** Halaman *Dashboard*

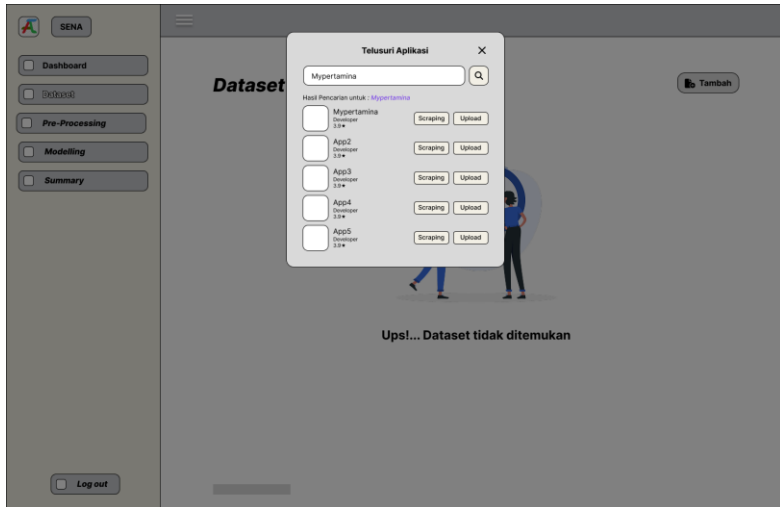
### 3) Halaman Dataset

Pada halaman ini user dapat memasukkan dataset ulasan atau melakukan *scraping*. Pada gambar 3.7 menampilkan halaman awal ketika user belum memiliki data ulasan. User dapat memasukkan data menggunakan tombol tambah.



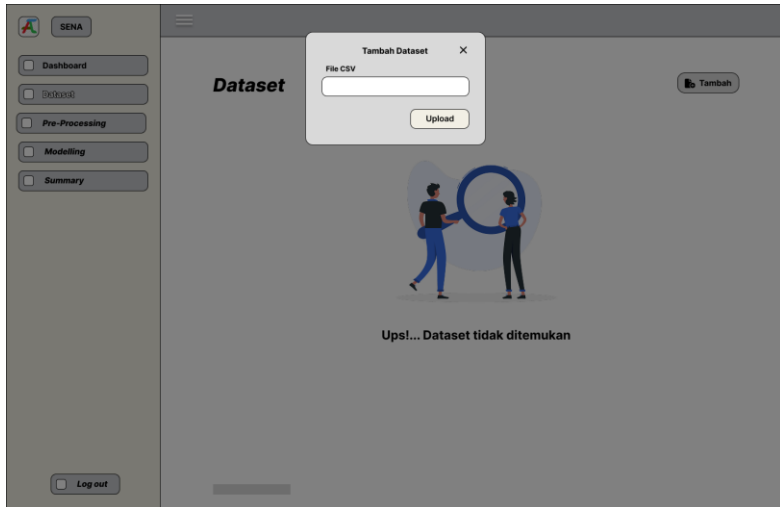
**Gambar 3.7** Halaman Dataset Awal

Setelah menekan tombol tambah user akan diarahkan ke modal pencarian mencari aplikasi yang akan digunakan untuk dilakukan analisis sentimen. Berikut pada gambar 3.8 menampilkan modal pencarian aplikasi.



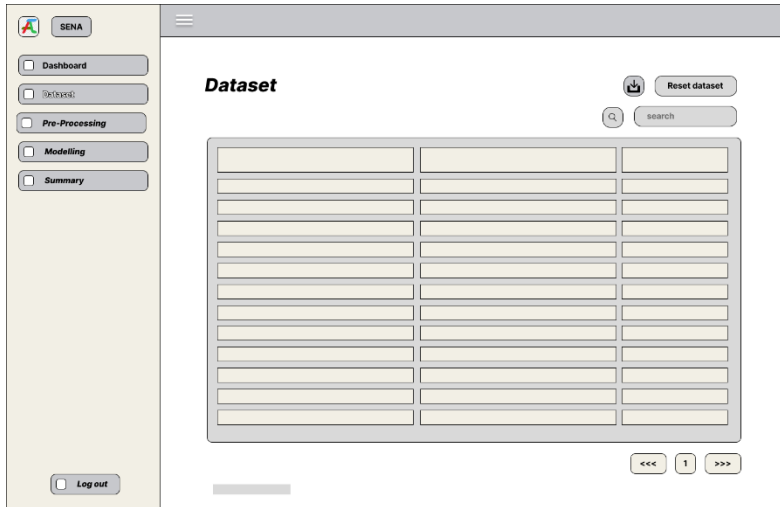
**Gambar 3.8** Modal Pencarian Aplikasi

Setelah memasukkan *keyword* aplikasi yang dicari, data hasil pencarian akan ditampilkan pada bagian bawah bar pencarian. Pada setiap data hasil pencarian tersedia pilihan untuk memasukkan data, yakni metode *scraping* dan *upload*. Tombol *scraping* digunakan untuk melakukan *scraping* data dan tombol *upload* digunakan untuk memasukkan file dataset yang telah dimiliki user. Aplikasi yang terpilih akan meneruskan detail aplikasi ke dalam modal metode input data yang dipilih.



**Gambar 3.9** Modal Metode Upload

Pada gambar 3.9 menampilkan modal metode *upload* untuk memasukkan file dataset yang berformat CSV. Sebelum memasukkan file CSV pastikan file tersebut memiliki atribut ulasan dan rating agar dapat diproses oleh sistem. Setelah memilih dataset yang akan digunakan tekan tombol *upload*, file CSV akan disimpan ke dalam folder server untuk kemudian ditampilkan dalam bentuk tabel seperti pada gambar 3.10.

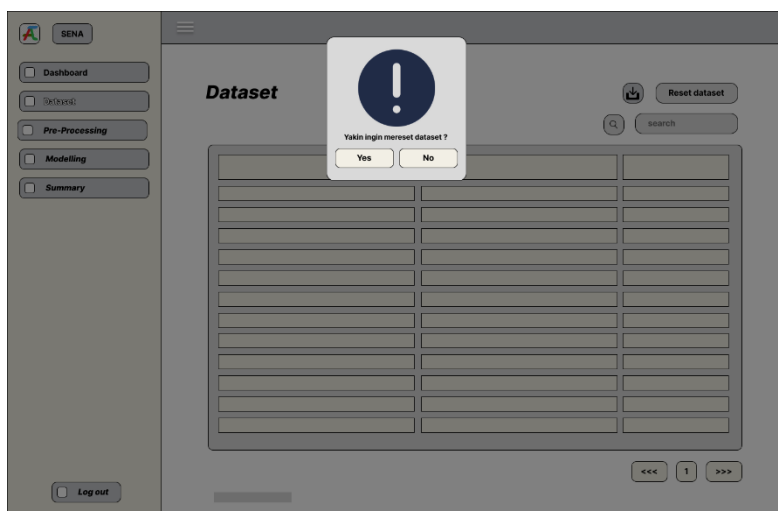


**Gambar 3.10** Menampilkan Data Ulasan dalam Tabel

Pada gambar 3.10 menampilkan data ulasan dalam bentuk tabel. Pada halaman dataset ini data ulasan dapat ditampilkan berdasarkan filter yang diinginkan user. Tombol *download* disediakan, jika user ingin mendownload data ulasan yang ditampilkan pada tabel dengan format file CSV. Setiap user hanya bisa melakukan proses analisis sentimen menggunakan satu dataset, jika ingin melakukan analisis sentimen menggunakan dataset lain seluruh data proses sebelumnya harus dihapus terlebih dahulu. Tombol reset



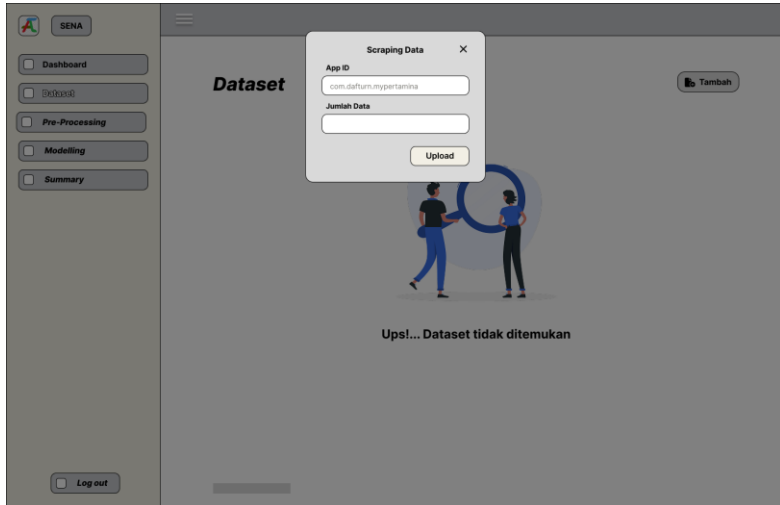
dataset digunakan untuk menghapus seluruh data hasil milik user yang tersimpan dalam server. Pada gambar 3.11 menampilkan modal konfirmasi reset dataset yang digunakan untuk menghapus seluruh data hasil proses yang tersimpan dalam server.



**Gambar 3.11** Modal Konfirmasi Reset Dataset

Proses sebelumnya harus dihapus terlebih dahulu. Tombol reset dataset digunakan menghapus seluruh data hasil milik user yang tersimpan dalam server. Setelah data berhasil dihapus, user akan diarahkan ke

halaman dataset awal pada gambar 3.7 untuk kembali memasukkan data ulasan.



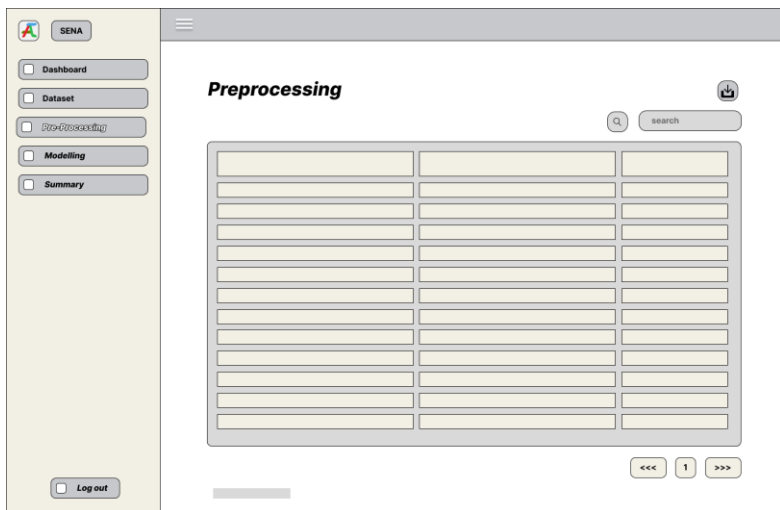
**Gambar 3.12** Modal Metode Scraping

Pada gambar 3.12 menampilkan modal metode *scraping* yang digunakan untuk melakukan *scraping* data ulasan. Data appId yang diteruskan dari modal pencarian aplikasi akan dimasukkan dalam form input secara otomatis. User dapat memasukkan jumlah data ulasan yang ingin diambil dengan batas minimal 200 dan maksimal 3000 data ulasan. Sistem akan melakukan proses *scraping* setelah

user menekan tombol *upload* dan data ulasan akan disimpan ke dalam folder server untuk kemudian ditampilkan dalam bentuk tabel seperti pada gambar 3.10.

#### 4) Halaman *Preprocessing*

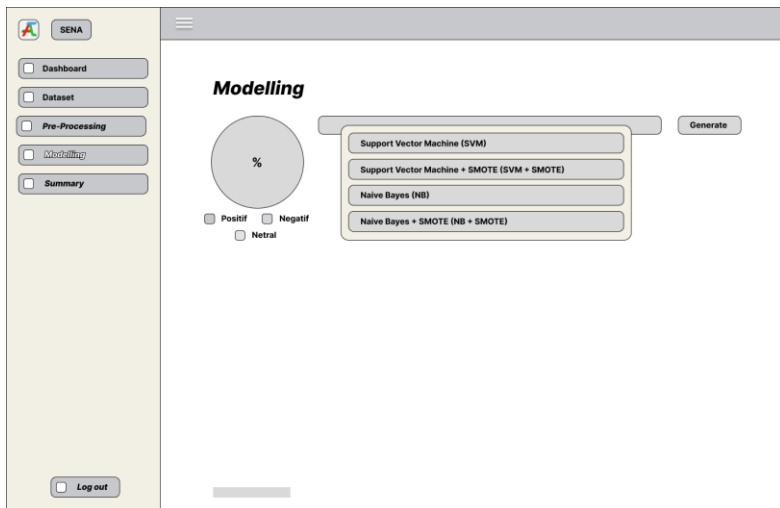
Data ulasan yang disimpan sebagai file CSV akan dibaca untuk memasuki tahap pelabelan dan pembersihan data. Setelah melalui proses ini, data ulasan bersih akan disimpan ke dalam folder server untuk ditampilkan dalam bentuk tabel seperti pada gambar 3.12.



**Gambar 3.13** Halaman *Preprocessing*

## 5) Halaman Modeling

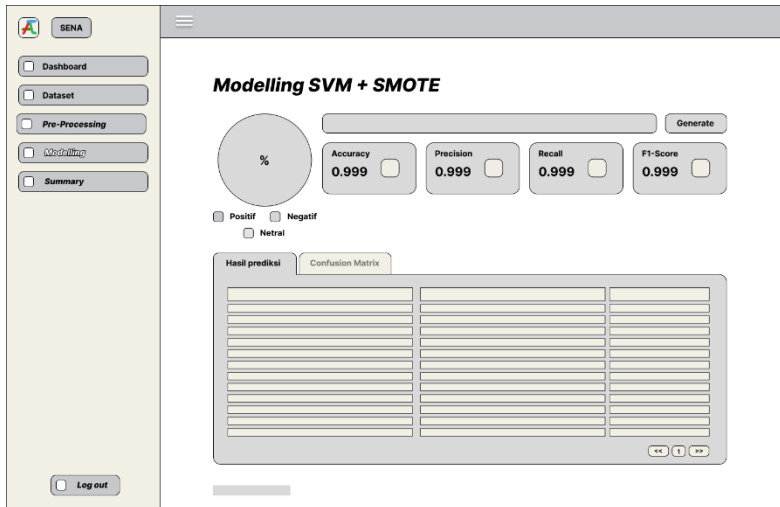
Pada halaman ini user dapat memilih metode untuk mengklasifikasikan sentimen ulasan aplikasi. Terdapat empat pilihan metode yang dapat digunakan seperti pada gambar 3.14.



**Gambar 3.14** Halaman Modeling (Menu Metode)

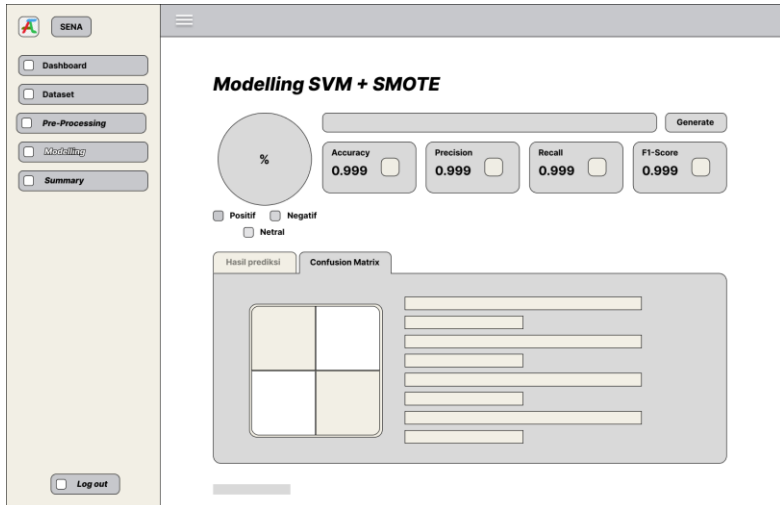
Pada halaman ini distribusi sentimen pada data *training* akan ditampilkan dalam bentuk grafik lingkaran. Setelah memilih metode modeling yang diinginkan, tekan tombol *generate* untuk mulai melatih model

dan akan secara otomatis dilanjutkan proses evaluasi model menggunakan data testing.



**Gambar 3.15** Halaman Hasil Modeling (Hasil Prediksi)

Pada gambar 3.15 menampilkan hasil proses modeling yang meliputi grafik distribusi sentimen, hasil evaluasi model, tabel data hasil prediksi, dan confusion matrix. Hasil evaluasi model yang ditampilkan meliputi *accuracy*, *precision*, *recall*, dan *f1-score*. Pada tab hasil prediksi menampilkan tabel perbandingan sentimen aktual dengan hasil prediksi.



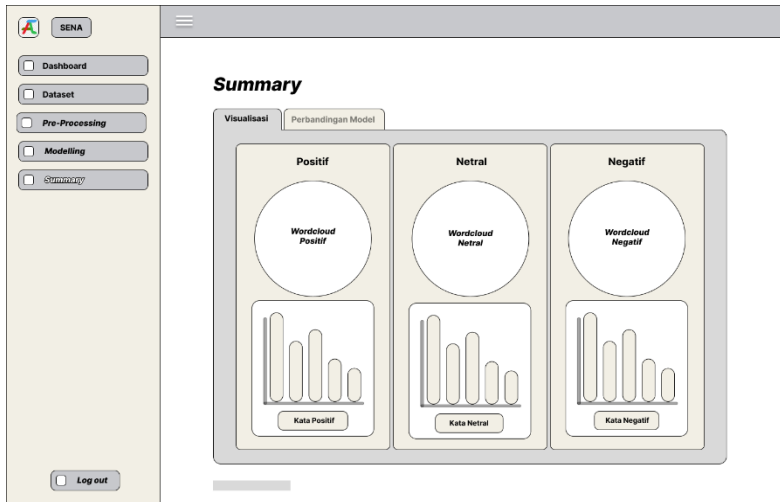
**Gambar 3.16** Halaman Hasil Modeling (*Confusion Matrix*)

Pada gambar 3.16 menampilkan tab confusion matrix berisi yang gambar confusion matrix dari model dipilih dengan keterangan singkat di sampingnya.

#### 6) Halaman Summary

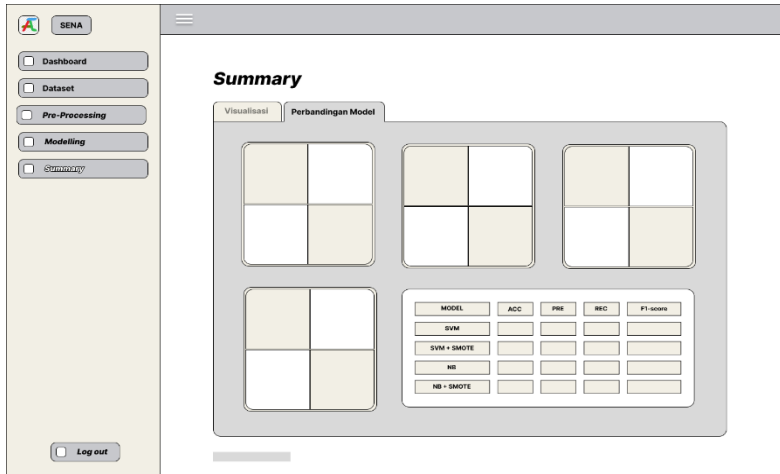
Pada halaman ini hasil proses sebelumnya akan disajikan dalam bentuk data visual yang meliputi *wordcloud*, diagram kata paling sering muncul, confusion matrix dan perbandingan performa model. Data visual berupa gambar seperti *wordcloud* dan confusion matrix akan disimpan pada folder

images di server. Adapun data visual pada halaman ini dapat dilihat pada gambar 3.17.



**Gambar 3.17** Halaman Summary (Visualisasi Data)

Pada gambar 3.17 menampilkan tab visualisasi yang berisi *wordcloud* dan grafik frekuensi kata dari setiap kelas sentimen. Pada tab perbandingan model menampilkan confusion matrix dan perbandingan performa model yang dapat dilihat pada gambar 3.18.

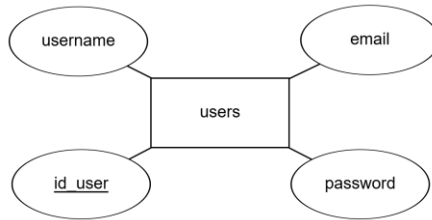


**Gambar 3.18** Halaman Summary (Perbandingan Model)

### c. ERD

ERD (*Entity Relationship Diagram*) adalah pemodelan data atau sistem dalam database. Fungsi ERD adalah untuk memodelkan struktur dan hubungan antar data yang relatif kompleks. Pada penelitian ini hanya terdapat satu entitas yaitu users. Berikut pada gambar 3.19 menampilkan ERD sistem analisis sentimen ulasan aplikasi.





**Gambar 3.19** Entity Relationship Diagram

### 3. Implementasi

Setelah desain dibuat selanjutnya akan diimplementasikan ke dalam aplikasi berbasis web menggunakan framework *Flask*. Pada penelitian ini pengembangan akan menggunakan *visual studio code* dan github sebagai versi kontrol aplikasi.

### 4. Pengujian

Pada tahap ini akan dilakukan pengujian pada aplikasi yang sudah dibuat. Pengujian akan menggunakan metode Black Box Testing untuk menguji fungsionalitas aplikasi yang telah dibuat.

### 5. Maintenance

Selanjutnya aplikasi akan disesuaikan dengan kebutuhan yang belum lengkap sehingga aplikasi dapat dideploy dan digunakan oleh user.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### A. Sistem Analisis Sentimen

##### 1. *Scraping Data*

Proses pengumpulan data ulasan aplikasi MyPertamina di Playstore dilakukan dengan menggunakan library `google_play_scraper` seperti pada gambar 4.1 berikut.

```
!pip install google_play_scraper
from google_play_scraper import Sort, reviews_all
import pandas as pd
```

**Gambar 4.1** *Install dan Import Module Scraping*

Pada gambar 4.1 library `google_play_scraper` akan diinstall kemudian modul dari library akan diimport, yakni modul `sort` dan `reviews_all`. `Sort` digunakan untuk melakukan filtering ulasan, sedangkan modul `reviews_all` berguna untuk mengambil semua ulasan dari aplikasi. Selain itu, modul `pandas` juga akan diimport untuk proses manipulasi data dalam sistem analisis sentimen ini. Setelah modul diimport selanjutnya akan dilakukan proses *scraping* dengan program yang ditampilkan pada gambar 4.2.

```

# Inisialisasi list untuk menyimpan semua review
all_reviews = []

# Set untuk menyimpan ID ulasan yang telah ditemukan
review_ids = set()

# Jumlah review yang ingin diambil
total_reviews = 14000
batch_size = 200 # Ukuran setiap batch review yang diambil

# Lakukan iterasi untuk mengambil review sebanyak yang dibutuhkan
while len(all_reviews) < total_reviews:
    # Ambil review dalam batch
    result = reviews_all(
        'com.dafturn.mypertamina',
        lang='id',
        country='id',
        sort=Sort.NEWEST,
        count=batch_size,
        filter_score_with=None
    )

    # Loop melalui hasil
    for review in result:
        # Jika ID ulasan belum ada di set, tambahkan ke list dan set
        if review['reviewId'] not in review_ids:
            # Simpan hanya kolom yang diinginkan
            formatted_review = {
                'Username': review['userName'],
                'Tanggal': review['at'].strftime("%Y-%m-%d"),
                'Ulasan': review['content'],
                'Rating': review['score'],
                'Versi': review['appVersion']
            }
            all_reviews.append(formatted_review)
            review_ids.add(review['reviewId'])

        # Jika sudah mencapai total yang diinginkan, keluar dari loop
        if len(all_reviews) >= total_reviews:
            break

# Konversi list review menjadi DataFrame
scrapped = pd.DataFrame(all_reviews)

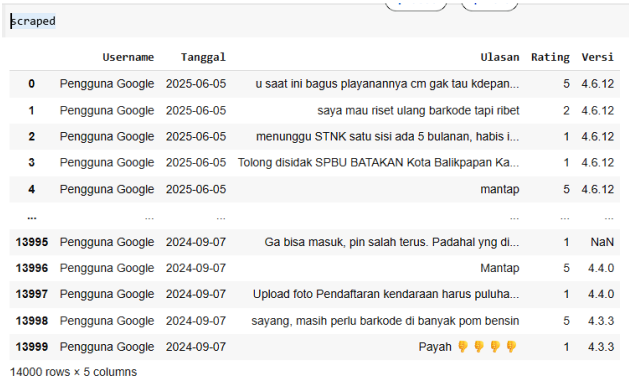
# Simpan DataFrame ke dalam file CSV
scrapped.to_csv('Dataset.csv', index=False)

```

**Gambar 4.2** Proses Scraping Data

Program pada gambar 4.2 dirancang untuk mengumpulkan total 14.000 ulasan unik. Tahapan awal dilakukan dengan menginisialisasi sebuah daftar kosong untuk menampung ulasan dan sebuah *set* untuk

mencatat ID setiap ulasan guna mencegah adanya data duplikat secara efisien. Proses pengambilan data berjalan dalam sebuah *loop* yang berlangsung terus-menerus hingga target 14.000 ulasan terpenuhi, di mana setiap iterasi mengambil 200 ulasan. Setiap ulasan yang didapat kemudian divalidasi keunikannya berdasarkan `reviewId`. Program akan menyeleksi dan mengambil hanya kolom-kolom data dengan format atribut, yakni username, tanggal, ulasan, rating, dan versi aplikasi. Setelah seluruh 14.000 ulasan unik berhasil terkumpul dan diformat, data tersebut dikonversi menjadi sebuah dataframe dan disimpan ke dalam sebuah file CSV. Adapun hasil dari proses *scraping* ini ditampilkan pada gambar 4.3 berikut.



	Username	Tanggal	Ulasan	Rating	Versi
0	Pengguna Google	2025-06-05	u saat ini bagus playananya cm gak tau kdepn...	5	4.6.12
1	Pengguna Google	2025-06-05	saya mau riset ulang barkode tapi ribet	2	4.6.12
2	Pengguna Google	2025-06-05	menunggu STNK satu sisi ada 5 bulanan, habis i...	1	4.6.12
3	Pengguna Google	2025-06-05	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	1	4.6.12
4	Pengguna Google	2025-06-05	mantap	5	4.6.12
...	...	...	...	...	...
13995	Pengguna Google	2024-09-07	Ga bisa masuk, pin salah terus. Padahal yng di...	1	NaN
13996	Pengguna Google	2024-09-07	Mantap	5	4.4.0
13997	Pengguna Google	2024-09-07	Upload foto Pendaftaran kendaraan harus puluha...	1	4.4.0
13998	Pengguna Google	2024-09-07	sayang, masih perlu barkode di banyak pom bensin	5	4.3.3
13999	Pengguna Google	2024-09-07	Payah 🤔🤔🤔🤔	1	4.3.3

14000 rows x 5 columns

Gambar 4.3 Hasil Scraping Data

## 2. Persiapan Data

```
# Membaca data hasil scraping
data = pd.read_csv('/content/Dataset.csv')

# Hapus atribut selain Ulasan dan Rating
data = data.drop(columns=[col for col in data.columns if col not in ['Ulasan', 'Rating']])
data
```

**Gambar 4.4** Menghapus Atribut

Pada gambar 4.4 menunjukkan tahapan awal dalam persiapan data, yakni dengan membaca file CSV yang telah disimpan sebelumnya menjadi dataframe dan dimasukkan ke dalam variabel *data*. Proses selanjutnya, atribut selain Ulasan dan Rating akan dihapus. Hasil dari proses ini ditampilkan pada gambar 4.5 berikut.

		Ulasan	Rating
0	u saat ini bagus playanannya cm gak tau kedepan...		5
1	saya mau riset ulang barkode tapi ribet		2
2	menunggu STNK satu sisi ada 5 bulanan, habis l...		1
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...		1
4	mantap		5
...		...	...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...		1
13996	Mantap		5
13997	Upload foto Pendaftaran kendaraan harus puluha...		1
13998	sayang, masih perlu barkode di banyak pom bensin		5
13999	Payah 🙄🙄🙄🙄		1

14000 rows x 2 columns

**Gambar 4.5** Hasil Hapus Atribut

Pada gambar 4.5 menampilkan tabel hasil penghapusan atribut yang berisi dua kolom, yakni ulasan dan rating. Selanjutnya, data akan dihapus dan dilakukan pelabelan seperti gambar 4.6 berikut.

```
# Hapus duplikat
data = data.drop_duplicates(subset=['Ulasan'])

# Buat kolom baru 'Sentimen' dengan label-label yang sesuai
data['Sentimen'] = data['Rating'].apply(lambda x: 'positif' if x > 3 else ('netral' if x == 3 else 'negatif'))

# Tampilkan Data Frame dengan kolom baru 'Sentimen' dan menghilangkan Rating
data = data.drop(columns='Rating')
data
```

**Gambar 4.6** Menghapus Duplikat dan Pelabelan

Pada gambar 4.6 data ulasan duplikat akan dihapus untuk selanjutnya dilakukan pelabelan berdasarkan rating dan menyimpan hasil pelabelan kedalam kolom sentimen. Setelah dilakukan pelabelan, kolom rating akan dihapus dan data akan ditampilkan seperti pada gambar 4.7 berikut.

	Ulasan	Sentimen
0	u saat ini bagus playanannya cm gak tau kdepn...	positif
1	saya mau riset ulang barkode tapi ribet	negatif
2	menunggu STNK satu sisi ada 5 bulanan, habis l...	negatif
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif
4	mantap	positif
...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif
13998	sayang, masih perlu barkode di banyak pom bensin	positif
13999	Payah 🙄🙄🙄🙄	negatif

11294 rows x 2 columns

**Gambar 4.7** Hasil Hapus Duplikat dan Pelabelan

Pada gambar 4.7 menunjukkan hasil proses penghapusan duplikat dan pelabelan. Data yang dihasilkan terdiri dari dua kolom, yakni ulasan dan sentimen, dengan jumlah akhir sebanyak 11.294 ulasan.

### 3. Preprocessing

```
# Mengubah text menjadi lower
data['Preprocessed'] = data['Ulasan'].str.lower()
data
```

**Gambar 4.8** Case Folding

Pada gambar 4.8 menampilkan program untuk mentransform data ulasan menjadi huruf kecil (*lower case*) yang disimpan dalam kolom preprocessed untuk kemudian ditampilkan seperti pada gambar 4.9 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playanannya cm gak tau kdepn...	postif	u saat ini bagus playanannya cm gak tau kdepn...
1	saya mau riset ulang barkode tapi ribet	negatif	saya mau riset ulang barkode tapi ribet
2	menunggu STNK satu sisi ada 5 bulanan, habis l...	negatif	menunggu stnk satu sisi ada 5 bulanan, habis l...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	tolong disidak spbu batakan kota balikpapan ka...
4	mantap	postif	mantap
...	...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	aplikasi dongo, lu nyuruh verifikasi tapi otp ...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	ga bisa masuk, pin salah terus. padahal yng di...
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif	upload foto pendaftaran kendaraan harus puluha...
13998	sayang, masih perlu barkode di banyak pom bensin	postif	sayang, masih perlu barkode di banyak pom bensin
13999	Payah 🙄🙄🙄🙄	negatif	payah 🙄🙄🙄🙄

11294 rows x 3 columns

**Gambar 4.9** Hasil Case Folding

Pada gambar 4.9 menampilkan hasil *case folding* yang mengubah semua teks ulasan menjadi ke bentuk huruf kecil. Tahap selanjutnya data ulasan akan dilakukan proses *cleansing* seperti yang ditampilkan pada gambar 4.10 berikut.

```

!pip install IndoNLP
import string
from string import punctuation
from IndoNLP.preprocessing import pipeline, replace_word_elongation, replace_slang, remove_url

# menghapus angka
data['Preprocessed'] = data['Preprocessed'].apply(lambda x: ''.join([char for char in x if not char.isdigit()]))

# Menghapus tanda baca
data['Preprocessed'] = data['Preprocessed'].apply(lambda x: ''.join([char for char in x if char not in string.punctuation]))

#Menghapus
data['Preprocessed'] = data['Preprocessed'].apply(remove_url)
data

```

**Gambar 4.10** *Cleansing*

Proses *cleansing* diawali dengan penginstalan library `IndoNLP` dan import modul. Selanjutnya data ulasan akan dibersihkan dari angka dengan metode `isdigit`, simbol atau tanda baca dengan mengacu pada konstanta `string.punctuation`, dan URL dengan fungsi `remove_url`. Adapun hasil proses *cleansing* ditunjukkan pada gambar 4.11 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playanannya cm gak tau kdepan...	positif	u saat ini bagus playanannya cm gak tau kdepan...
1	saya mau riset ulang barkode tapi ribet	negatif	saya mau riset ulang barkode tapi ribet
2	menunggu STNK satu sisi ada 5 bulanan, habis i...	negatif	menunggu stnk satu sisi ada bulanan habis itu...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	tolong disidak spbu batakan kota balikpapan ka...
4	mantap	positif	mantap
...	...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	aplikasi dongo lu nyuruh verifikasi tapi otp n...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	ga bisa masuk pin salah terus padahal yng din...
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif	upload foto pendaftaran kendaraan harus puluha...
13998	sayang, masih perlu barkode di banyak pom bensin	positif	sayang masih perlu barkode di banyak pom bensin
13999	Payah 🙄🙄🙄🙄	negatif	payah 🙄🙄🙄🙄

11294 rows x 3 columns

**Gambar 4.11** *Hasil Cleansing*



```
# Normalisasi
pipe = pipeline([replace_word_elongation, replace_slang])
data['Preprocessed'] = data['Preprocessed'].apply(lambda x: ' '.join([pipe(token) for token in x.split()])))
data
```

**Gambar 4.12** Normalisasi

Proses normalisasi menggunakan fungsi yang telah diimport dari library `indoNLP` sebelumnya, yaitu `replace_word_elongation` dan `replace_slang`. Fungsi `replace_word_elongation` digunakan untuk menangani kata-kata yang ditulis dengan pemanjangan karakter berulang, misalnya mengubah kata *"bagusss"* kembali menjadi *"bagus"*. Sementara itu, fungsi `replace_slang` bekerja dengan cara mengganti kata-kata gaul (slang) yang ada dalam teks ulasan dengan padanan katanya yang baku berdasarkan kamus yang dimilikinya. Adapun hasil proses normalisasi ditampilkan pada gambar 4.13 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playanannya cm gak tau kdepan...	positif	lu saat ini bagus playanannya cuma enggak tau ...
1	saya mau riset ulang barkode tapi ribet	negatif	saya mau riset ulang barkode tapi ribet
2	menunggu STNK satu sisi ada 5 bulanan, habis i...	negatif	menunggu stnk satu sisi ada bulanan habis itu ...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	tolong disidak spbu batakan kota Balikpapan ka...
4	mantap	positif	mantap
...	...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	aplikasi dongo lu nyuruh verifikasi tapi otp n...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	enggak bisa masuk pin salah terus padahal yang...
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif	upload foto pendaftaran kendaraan harus puluha...
13998	sayang, masih perlu barkode di banyak pom bensin	positif	sayang masih perlu barkode di banyak pom bensin
13999	Payah 🤔🤔🤔🤔	negatif	payah 🤔🤔🤔🤔

11294 rows x 3 columns

**Gambar 4.13** Hasil Normalisasi

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.tokenize import word_tokenize

# Tokenisasi
data['Preprocessed'] = data['Preprocessed'].apply(nltk.word_tokenize)
data
```

Gambar 4.14 Tokenisasi

Pada gambar 4.14 menampilkan proses tokenisasi yang diawali dengan mengimport modul nltk word\_tokenize, kemudian data akan di pecah menjadi unit-unit kata. Adapun hasil dari proses tokenisasi ditmapilkan pada gambar 4.15 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playannya cm gak tau kedepan...	positif	[lu, saat, ini, bagus, playannya, cuma, engg...
1	saya mau riset ulang barkode tapi ribet	negatif	[saya, mau, riset, ulang, barkode, tapi, ribet]
2	menunggu STNK satu sisi ada 5 bulanan, habis i...	negatif	[menunggu, stnk, satu, sisi, ada, bulanan, hab...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	[tolong, disidak, spbu, batakan, kota, balikpa...
4	mantap	positif	[mantap]
...	...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	[aplikasi, dongo, lu, nyuruh, verifikasi, tapi...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	[enggak, bisa, masuk, pin, salah, terus, padah...
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif	[upload, foto, pendaftaran, kendaraan, harus, ...
13998	sayang, masih perlu barkode di banyak pom bensin	positif	[sayang, masih, perlu, barkode, di, banyak, po...
13999	Payah 🙄🙄🙄🙄	negatif	[payah, 🙄🙄🙄🙄]

11294 rows x 3 columns

Gambar 4.15 Hasil Tokenisasi

Pada gambar 4.15 menunjukkan hasil proses tokenisasi dengan hasil ulasan yang sudah dipecah menjadi unit kata. Transformasi ini mengubah data dari format kalimat utuh menjadi daftar kata terstruktur yang siap untuk tahap analisis selanjutnya.

```

from nltk.corpus import stopwords

# Menghapus stopwords
data['Preprocessed'] = data['Preprocessed'].apply(lambda x: [token for token in x if token not in stopwords.words('Indonesian')])
data

```

**Gambar 4.16** Stopwords Removal

Tahapan berikutnya adalah penghapusan stopwords yang diawali dengan mengimport modul stopwords dari nltk dan dilanjutkan mencari token serupa dengan yang ada pada corpus stopwords. Adapun hasil penghapusan stopwords ditampilkan pada gambar 4.17 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playanannya cm gak tau kedepan...	positif	[lu, bagus, playanannya, tau, kdepantngkatkan...
1	saya mau riset ulang barkode tapi ribet	negatif	[riset, ulang, barkode, ribet]
2	menunggu STNK satu sisi ada 5 bulanan, habis l...	negatif	[menunggu, stnk, sisi, bulanan, habis, daftar,...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	[tolong, disidak, spbu, batakan, kota, balikpa...
4	mantap	positif	[mantap]
...	...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	[aplikasi, dongo, lu, nyuruh, verifikasi, otp,...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	[masuk, pin, salah, diinput, aplikasi]
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif	[upload, foto, pendaftaran, kendaraan, puluhan...
13998	sayang, masih perlu barkode di banyak pom bensin	positif	[sayang, barkode, pom, bensin]
13999	Payah 🙄🙄🙄🙄	negatif	[payah, 🙄🙄🙄🙄]

11294 rows x 3 columns

**Gambar 4.17** Hasil Stopword Removal

Hasil penghapusan stopwords akan menghilangkan kata yang sering muncul namun tidak memiliki sentimen kuat, seperti kata “saya”, “mau” dan “tapi” pada data ulasan “saya mau riset ulang kode tapi ribet” yang berhasil dihapus pada proses ini.

```

!pip install nlp-id
from nlp_id.lemmatizer import Lemmatizer

# Membuat objek Lemmatizer
lemmatizer = Lemmatizer()

# Menerapkan lemmatisasi pada setiap token dalam data tokenisasi
data['Preprocessed'] = data['Preprocessed'].apply(lambda x: [lemmatizer.lemmatize(token) for token in x])
data['Preprocessed'] = data['Preprocessed'].apply(lambda x: " ".join(x))
data

```

**Gambar 4.18** Lematization

Pada proses lemmatisasi token data akan diubah menjadi ke bentuk dasarnya menggunakan library `nlp-id` dengan modul `lemmatizer`. Selanjutnya data yang sudah di transform menjadi bentuk dasar akan digabungkan kembali menjadi satu kalimat. Adapun hasil lemmatisasi ditunjukkan pada gambar 4.19 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playanannya cm gak tau kdepan...	positif	lu bagus playanannya tau kdepantingkatkan ajat...
1	saya mau riset ulang barkode tapi ribet	negatif	riset ulang barkode ribet
2	menunggu STNK satu sisi ada 5 bulanan, habis i...	negatif	tunggu stnk sisi bulan habis daftar jadjadi m...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	tolong disidak spbu batak kota balikpapan kail...
4	mantap	positif	mantap
...	...	...	...
13993	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	aplikasi dongo lu nyuruh verifikasi otp lu kir...
13995	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	masuk pin salah diinput aplikasi
13997	Upload foto Pendaftaran kendaraan harus puluha...	negatif	upload foto daftar kendaraan puluh kali hasil
13998	sayang, masih perlu barkode di banyak pom bensin	positif	sayang barkode pom bensin
13999	Payah 🙄🙄🙄🙄	negatif	payah

11294 rows x 3 columns

**Gambar 4.19** Hasil Lematization

Pada gambar 4.19 menunjukkan hasil dari proses konversi kata-kata ke bentuk dasar yang berguna untuk model mengenali pola dengan lebih efektif karena tidak perlu mempelajari variasi kata yang berbeda-beda untuk makna yang sama.

```
# Hapus ulasan kosong dengan dropna() dan reset index
data = data.dropna(axis='rows')
data = data[data['Preprocessed'] != ""]
data.reset_index(drop=True, inplace=True)

# Simpan data kedalam csv
data.to_csv('Preprocessed.csv', index=False)
data
```

**Gambar 4.20** Menghapus Data Kosong

Proses terakhir dalam *preprocessing* adalah menghapus data kosong. Keberadaan baris data kosong dapat menyebabkan kesalahan pada tahap pemodelan, sehingga wajib untuk dihilangkan. Proses ini dilakukan melalui dua langkah penyaringan, yakni fungsi `dropna()` untuk menghapus baris yang mengandung nilai `NaN` dan menyaring data pada kolom 'Preprocessed' untuk membuang baris yang hanya berisi string kosong. Setelah itu, indeks dataframe diatur ulang untuk memastikan penomoran baris kembali berurutan mulai dari 0 dan membuang indeks lama yang tidak lagi relevan. Dataframe yang telah sepenuhnya bersih akan disimpan ke dalam sebuah file baru bernama 'Preprocessed.csv' untuk digunakan pada tahap selanjutnya. Adapun hasil proses menghapus data kosong ditunjukkan pada gambar 4.21 berikut.

	Ulasan	Sentimen	Preprocessed
0	u saat ini bagus playanannya cm gak tau kedepan...	positif	lu bagus playanannya tau kdepantingkatkan ajat...
1	saya mau riset ulang barkode tapi ribet	negatif	riset ulang barkode ribet
2	menunggu STNK satu sisi ada 5 bulanan, habis i...	negatif	tunggu stnk sisi bulan habis daftar jadijadi m...
3	Tolong disidak SPBU BATAKAN Kota Balikpapan Ka...	negatif	tolong disidak spbu batak kota balikpapan kali...
4		mantap	positif
...		...	...
11187	Aplikasi dongo, lu nyuruh verifikasi tapi otp ...	negatif	aplikasi dongo lu nyuruh verifikasi otp lu kir...
11188	Ga bisa masuk, pin salah terus. Padahal yng di...	negatif	masuk pin salah diinput aplikasi
11189	Upload foto Pendaftaran kendaraan harus puluha...	negatif	upload foto daftar kendaraan puluh kali hasil
11190	sayang, masih perlu barkode di banyak pom bensin	positif	sayang barkode pom bensin
11191	Payah 🙄🙄🙄🙄	negatif	payah

11192 rows x 3 columns

**Gambar 4.21** Hasil Menghapus Data Kosong

Pada gambar 4.21 menunjukkan hasil setelah dilakukan penghapusan baris data kosong yang diperoleh data akhir sebanyak 11.192 ulasan. Jumlah ini merupakan hasil reduksi dari data awal yang berjumlah 11.194 ulasan.

#### 4. Modeling

```
dc = pd.read_csv('Preprocessed.csv')

# Filtering rating
dc = dc[dc['Sentimen'] != 'netral']
data_positif = dc[dc['Sentimen'] == 'positif']
data_negatif = dc[dc['Sentimen'] == 'negatif']
```

**Gambar 4.22** Filtering Sentimen

Pada tahap awal modeling, file `Preprocessed.csv` akan dibaca dan dimuat pada variabel `dc` (data clean). Data ulasan dengan sentimen netral dibuang karena tidak digunakan pada proses

modeling, sedangkan data sentimen positif disimpan pada variabel `data_positif` dan data sentimen negatif pada variabel `data_negatif` untuk digunakan pada proses berikutnya. Proses selanjutnya data ulasan akan dibagi menjadi data *training* dan data *testing* dengan rasio 80:20 seperti pada gambar 4.23 berikut.

```
from sklearn.model_selection import train_test_split
from collections import Counter

# Split data menjadi data training dan data testing
X_train, X_test, y_train, y_test = train_test_split(dc['Preprocessed'], dc['Sentimen'],
                                                    test_size=0.2, stratify=dc['Sentimen'],
                                                    random_state=42)

print("Jumlah data training:", Counter(y_train))
print("Jumlah data testing:", Counter(y_test))

Jumlah data training: Counter({'negatif': 5920, 'positif': 2562})
Jumlah data testing: Counter({'negatif': 1480, 'positif': 641})
```

**Gambar 4.23** *Splitting Data*

Proses *splitting data* dilakukan dengan mengimport modul `train_test_split` dari library `sklearn` dan modul `Counter` untuk menghitung jumlah sentimen pada data *training* dan *testing*. Hasil yang didapatkan pada data *training*, terdiri atas 5.920 sentimen negatif dan 2.562 sentimen positif dengan total 8.482 sentimen. Pada data *testing* jumlah sentimen negatif sebanyak 1.480 ulasan dan sentimen positif sebanyak 641 ulasan dengan total sebanyak 2.121 ulasan. Proses selanjutnya data ulasan akan dilakukan ekstraksi fitur menggunakan TF-IDF yang ditunjukkan pada gambar 4.24 berikut.

```

from sklearn.feature_extraction.text import TfidfVectorizer

# Membuat TF-IDF vectorizer
vectorizer = TfidfVectorizer(decode_error='replace', encoding='utf-8')

# Transformasi teks menjadi representasi TF-IDF
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# Mengambil fitur names (kata-kata) dari vectorizer
feature_names = vectorizer.get_feature_names_out()

# Mendapatkan skor TF-IDF untuk setiap kata di setiap dokumen pada data latih
tfidf_scores = X_train

# Buat DataFrame dari skor TF-IDF dengan fitur names sebagai kolom
df_tfidf = pd.DataFrame(tfidf_scores.toarray(), columns=feature_names)

# Hitung rata-rata skor TF-IDF untuk setiap kata di seluruh dokumen
average_tfidf_scores = df_tfidf.mean(axis=0)

# Urutkan kata berdasarkan rata-rata skor TF-IDF secara menurun
ranked_words = average_tfidf_scores.sort_values(ascending=False)

# Tampilkan 10 kata teratas beserta skor TF-IDF rata-ratanya
print("Ranking 10 Kata Berdasarkan Rata-rata Skor TF-IDF:")
print(ranked_words.head(10))

```

**Gambar 4.24** Ekstraksi Fitur

Tahap awal ekstraksi fitur dimulai dengan mengimport modul `TfidfVectorizer` dari library `sklearn`, setelah itu data *training* akan dilakukan proses TF-IDF kemudian ditransform menjadi vektor dengan metode `fit_transform()` dan data *testing* akan ditransform ke dalam vektor dengan metode `transform()`. Selanjutnya hasil dari TF-IDF akan ditampilkan berdasarkan nilai rata-rata TFIDF tertinggi seperti pada gambar 4.25 berikut.



```

Ranking 10 Kata Berdasarkan Rata-rata Skor TF-IDF:
aplikasi      0.060774
daftar        0.042742
susah         0.031038
ribet         0.025469
mudah         0.022415
sulit         0.021289
barcode       0.019813
pertamina    0.019764
otp           0.019666
masuk         0.019602
dtype: float64

```

**Gambar 4.25** Hasil Mean Score TF-IDF

Berdasarkan hasil rata-rata TF-IDF pada gambar 2.5 menunjukkan rangking 10 kata dari tertinggi hingga terendah. Hasil rata-rata TF-IDF menunjukkan tingkat kepentingan sebuah kata secara keseluruhan di dalam korpus. Adapun contoh perhitungan TF-IDF pada empat dokumen sampel dari data ulasan sebagai berikut :

Doc1 = “terimakasih, aplikasinya bermanfaat sekali”

Doc2 = “aplikasi sering eror”

Doc3 = “gak bisa masuk trus ini apk gak jelas”

Doc4 = “aplikasi nya bagus membantu beli BBM lewat online”

Selanjutnya dokumen tersebut akan dilakukan proses *preprocessing*, sehingga menghasilkan data sebagai berikut :

Doc1 = “terimakasih aplikasi manfaat”

Doc2 = “aplikasi eror”

Doc3 = “masuk aplikasi”

Doc4 = “aplikasi bagus bantu beli bbm online”

Setelah dilakukan *preprocessing*, didapatkan 10 *term* dari empat dokumen sampel yang ada. Proses selanjutnya adalah menghitung TF seperti yang ditampilkan pada tabel 4.1 berikut.

**Tabel 4.1** Perhitungan TF

<i>Term (t)</i>	D1	D2	D3	D4	$df_t$
terimakasih	0,33	0	0	0	1
aplikasi	0,33	0,5	0,5	0,16	4
manfaat	0,33	0	0	0	1
eror	0	0,5	0	0	1
masuk	0	0	0,5	0	1
bagus	0	0	0	0,16	1
bantu	0	0	0	0,16	1
beli	0	0	0	0,16	1
bbm	0	0	0	0,16	1
online	0	0	0	0,16	1

Pada tabel 4.1 hasil TF merupakan probabilitas kemunculan *term (t)* pada suatu dokumen, sedangkan dokumen frekuensi (*df*) merupakan jumlah dokumen yang mengandung *term (t)*.

**Tabel 4.2** Perhitungan IDF

<i>Term (t)</i>	$IDF = \log \left( \frac{N_d+1}{df_t+1} \right) + 1$
terimakasih	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
aplikasi	$\log \left( \frac{4+1}{4+1} \right) + 1 = 1$
manfaat	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
eror	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
masuk	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
bagus	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
bantu	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
beli	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
bbm	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$
online	$\log \left( \frac{4+1}{1+1} \right) + 1 = 1,916$

Pada tabel 4.2 menampilkan proses perhitungan IDF sesuai dengan persamaan yang digunakan pada *library sklearn*, yang secara default mengaktifkan *smoothing* dengan parameter `smooth_idf=True`. Penggunaan *smoothing* ini bertujuan untuk mencegah skor IDF menjadi nol bagi kata-kata yang muncul di semua dokumen, sehingga memastikan semua kata tetap memiliki bobot dan perhitungan menjadi lebih stabil.

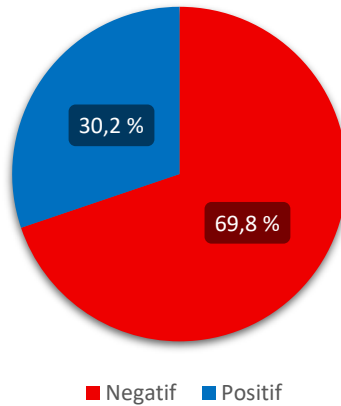
**Tabel 4.3** Perhitungan TF-IDF

Term (t)	TF-IDF = TF * IDF			
	D1	D2	D3	D4
terimakasih	0,63228	0	0	0
aplikasi	0,33	0,5	0,5	0,16
manfaat	0,63228	0	0	0
eror	0	0,958	0	0
masuk	0	0	0,958	0
bagus	0	0	0	0,30656
bantu	0	0	0	0,30656
beli	0	0	0	0,30656
bbm	0	0	0	0,30656
online	0	0	0	0,30656

Setelah nilai TF dan IDF didapat, nilai tersebut akan dikalikan untuk menghasilkan nilai TF-IDF. Tabel 4.3 menampilkan hasil TF-IDF yang mempresentasikan bobot dari fitur pada setiap dokumen sampel. Selanjutnya, dapat dihitung nilai rata-rata TF-IDF dengan menjumlahkan nilai TF-IDF pada *term (t)* dan membaginya dengan jumlah dokumen sampel. Adapun ranking nilai rata-rata TF-IDF dari tertinggi hingga terendah, yakni “aplikasi”, “eror”, “masuk”, “terimakasih”, “manfaat”, “bagus”, “bantu”, “beli”, “bbm”, dan “online”.

Berdasarkan proses *splitting data* sebelumnya, didapatkan bahwa distribusi sentimen pada data *training* tidak seimbang yang terdiri atas 5.920 sentimen negatif dan 2.562 sentimen positif. Adapun

grafik distribusi kelas pada data training ditampilkan pada gambar 4.26 berikut.



**Gambar 4.26** Grafik Distribusi Kelas Data Training

Distribusi kelas tidak merata ini dapat mempengaruhi proses pelatihan model yang dapat menyebabkan bias performa model. Kelas mayoritas yang sangat dominan akan membuat performa model sangat baik dalam memprediksi kelas mayoritas, namun model memiliki performa buruk dalam memprediksi kelas minoritas karena kekurangan data saat pelatihan model. Penggunaan SMOTE dapat membantu meratakan distribusi kelas, sehingga menghindari bias dari kelas mayoritas. Adapun proses SMOTE ditunjukkan pada gambar 4.27 berikut.

```

from imblearn.over_sampling import SMOTE

# Inisialisasi SMOTE
smote = SMOTE(random_state=42)

# Melakukan oversampling pada data training
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

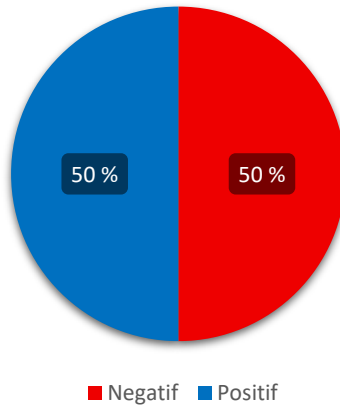
print("Jumlah data training resampled:", Counter(y_train_resampled))

```

Jumlah data training resampled: Counter({'negatif': 5920, 'positif': 5920})

**Gambar 4.27** Oversampling dengan SMOTE

Proses diawali dengan mengimport modul SMOTE dari *library* imblearn. Selanjutnya melakukan inisialisasi SMOTE dengan parameter `random_state=42` yang disimpan pada variabel `smote`. Setelah itu data training akan dilakukan oversample menggunakan metode `fit_resample()` yang kemudian ditampilkan distribusi sentimen setelah proses oversample seperti pada gambar 4.27. Berdasarkan hasil tersebut, distribusi ke dua kelas sentimen menjadi seimbang dengan jumlah data sentimen positif yang mengalami peningkatan sebanyak 3.358 ulasan. Berikut pada gambar 4.28 yang menampilkan grafik distribusi sentimen setelah dilakukan proses oversample.



**Gambar 4.28** *Distibusi Kelas Setelah SMOTE*

Proses oversampling dengan SMOTE menjadikan distribusi sentimen pada ke dua kelas menjadi seimbang, penambahan ini membuat data training yang awalnya terdiri dari 8.482 ulasan menjadi 11.840 ulasan. Selanjutnya, data training asli dan data training resampled akan digunakan dalam proses pelatihan model menggunakan algoritma klasifikasi SVM dan NB.

Proses klasifikasi dilakukan menggunakan modul SVC dan MultinomialNB dari *library* sklearn. Parameter yang digunakan dalam proses klasifikasi ini merupakan parameter default. Adapun proses pelatihan model klasifikasi SVM dan NB yang ditunjukkan gambar 4.29 berikut.

```

from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB

# Model SVM
svm = SVC()
svm.fit(X_train, y_train)

# Model SVMS
svms = SVC()
svms.fit(X_train_resampled, y_train_resampled)

# Model NB
nb = MultinomialNB()
nb.fit(X_train, y_train)

# Model NBS
nbs = MultinomialNB()
nbs.fit(X_train_resampled, y_train_resampled)

```

**Gambar 4.29** Pelatihan Model SVM dan NB

Pada gambar 4.29 menampilkan proses pelatihan model SVM dan NB yang menggunakan dua skenario data, yakni data *training* asli dan data *training* resampled. Proses untuk setiap skenario diawali dengan inisialisasi model ke dalam sebuah variabel, kemudian dilatih pada data *training* yang sesuai menggunakan metode `fit()`. Proses ini akan menghasilkan empat model klasifikasi, yakni SVM, SVMS (SVM & SMOTE), NB dan NBS (NB & SMOTE) yang siap digunakan untuk pengujian dengan data *testing*.



## 5. Evaluasi Model

Pada tahap ini model yang sudah dilatih akan dilakukan pengujian dan kemudian dievaluasi performanya. Proses pengujian model dengan data testing ditunjukkan pada gambar 4.30 berikut.

```
# Memprediksi kelas data testing
y_pred_svm = svm.predict(X_test)
y_pred_svms = svms.predict(X_test)
y_pred_nb = nb.predict(X_test)
y_pred_nbs = nbs.predict(X_test)
```

**Gambar 4.30** Proses Pengujian Data Testing

Pengujian model dilakukan menggunakan method `predict()` yang kemudian hasil dari prediksi tersebut disimpan dalam variabel prediksi model.

```
# Membuat dataframe untuk menampilkan hasil prediksi
prediction_df = pd.DataFrame({'Text Asli': dc['Ulasan'][y_test.index],
                             'Text clean': dc['Preprocessed'][y_test.index],
                             'Actual': y_test,
                             'P_SVM': y_pred_svm,
                             'P_SVMS': y_pred_svms,
                             'P_NB': y_pred_nb,
                             'P_NBS': y_pred_nbs})

# Menampilkan dataframe hasil prediksi
display(prediction_df.head())
```

	Text Asli	Text clean	Actual	P_SVM	P_SVMS	P_NB	P_NBS
1331	tolong lebih dipermudah	tolong mudah	positif	positif	positif	negatif	positif
10319	Saya daftar subsidi lepat kok sudah banget ya?	daftar subsidi banget ya?	positif	negatif	negatif	negatif	negatif
2777	transaksi cepat dan mudah, pegawai SPBU ug ramah	transaksi cepat mudah pegawai spbu ug ramah	positif	positif	positif	positif	positif
8095	Belum bisa mendapatkan QR	qr	positif	positif	positif	negatif	positif
8136	Memudahkan transaksi di SPBU	mudah transaksi spbu	positif	positif	positif	positif	positif

**Gambar 4.31** Menampilkan Tabel Hasil Prediksi

Gambar 4.31 menampilkan hasil dari pengujian setiap model dalam bentuk tabel. Data pengujian memiliki indeks acak karena parameter `random_state` yang digunakan pada saat proses *splitting data*. Selanjutnya data hasil pengujian ini akan dipetakan dalam confusion matrix yang ditunjukkan pada gambar 4.32 berikut.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

def confusionMatrix(model, X_train, y_train, X_test, y_test, y_pred, model_name):
    # Confusion matrix
    plt.figure(figsize=(5, 4))
    sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
    plt.title(f'{model_name}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.show()

# CM SVM
confusionMatrix(svm, X_train, y_train, X_test, y_test, y_pred_svm, 'SVM')

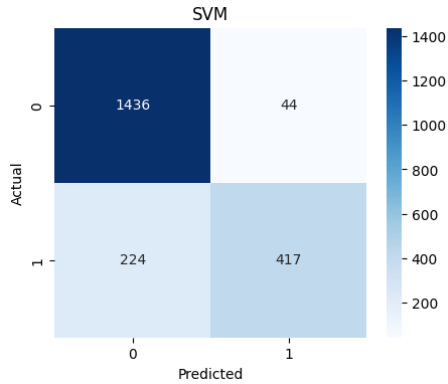
# CM SVM (SMOTE)
confusionMatrix(svms, X_train_resampled, y_train_resampled, X_test, y_test, y_pred_svms, 'SVM & SMOTE')

# CM Naive Bayes
confusionMatrix(nb, X_train, y_train, X_test, y_test, y_pred_nb, 'NBC')

# CM Naive Bayes (SMOTE)
confusionMatrix(nbs, X_train_resampled, y_train_resampled, X_test, y_test, y_pred_nbs, 'NBC & SMOTE')
```

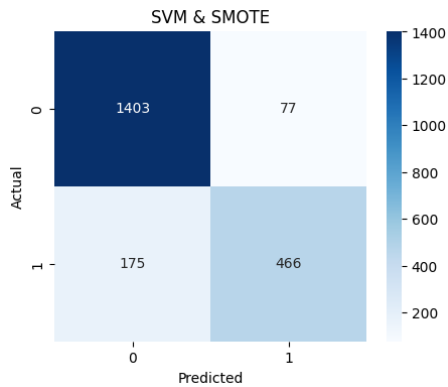
**Gambar 4.32** Menampilkan Confusion Matrix

Proses ini diawali dengan mengimport modul `confusion_matrix`, `seaborn` dan `matplotlib`. Fungsi `confusionMatrix()` digunakan untuk membuat confusion matrix dari setiap model. Adapun hasil pemanggilan fungsi tersebut ditunjukkan pada gambar berikut.



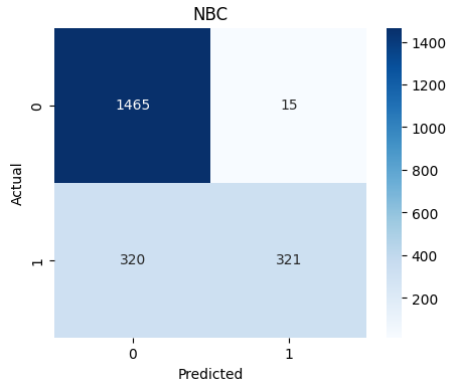
**Gambar 4.33** *Confusion Matrix SVM*

Gambar 4.33 menampilkan hasil confusion matrix dari model SVM.



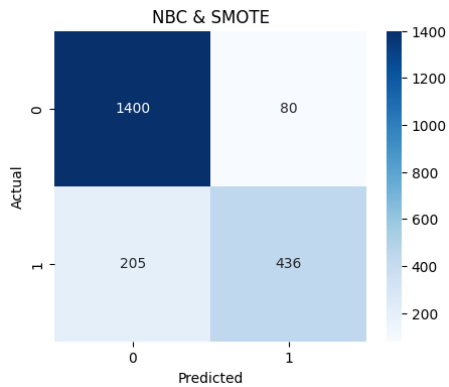
**Gambar 4.34** *Confusion Matrix SVM (SMOTE)*

Gambar 4.34 menampilkan hasil confusion matrix dari model SVMS.



**Gambar 4.35** Confusion Matrix NB

Gambar 4.35 menampilkan hasil confusion matrix dari model NB.



**Gambar 4.36** Confusion Matrix NB (SMOTE)

Gambar 4.36 menampilkan hasil confusion matrix dari model NBS.

Setelah confusion matrix didapatkan selanjutnya hasil tersebut akan dipetakan dalam tabel 4.4 berikut.

**Tabel 4.4** Hasil Confusion Matrix

Model	TP	FP	TN	FN
SVM	417	44	1436	224
SVMS	466	77	1403	175
NB	321	15	1465	320
NBS	436	80	1400	205

Proses selanjutnya adalah menghitung performa dari setiap model. Adapun hasil perhitungan akurasi dari setiap model yang dilakukan secara manual dijabarkan pada tahapan berikut.

- a. Perhitungan akurasi model SVM

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \times 100\%$$

$$Accuracy = \frac{1436 + 417}{1436 + 417 + 44 + 224} \times 100\%$$

$$Accuracy = \frac{1853}{2121} \times 100\%$$

$$Accuracy = 87,364 \%$$

- b. Perhitungan akurasi model SVMS

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \times 100\%$$

$$Accuracy = \frac{1403 + 466}{1436 + 417 + 44 + 224} \times 100\%$$

$$Accuracy = \frac{1869}{2121} \times 100\%$$

$$Accuracy = 88,118 \%$$

c. Perhitungan akurasi model NB

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \times 100\%$$

$$Accuracy = \frac{1465 + 321}{1436 + 417 + 44 + 224} \times 100\%$$

$$Accuracy = \frac{1786}{2121} \times 100\%$$

$$Accuracy = 84,205 \%$$

d. Perhitungan akurasi model NBS

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \times 100\%$$

$$Accuracy = \frac{1400 + 436}{1436 + 417 + 44 + 224} \times 100\%$$

$$Accuracy = \frac{1836}{2121} \times 100\%$$

$$Accuracy = 86,562 \%$$

Berdasarkan perhitungan manual tersebut, didapatkan hasil akurasi dari setiap model yang ditampilkan pada tabel 4.5 berikut.

**Tabel 4.5** Perbandingan Akurasi Model

Model	SVM	SVMS	NB	NBS
Akurasi	87,364	88,118	84,205	86,562

Berdasarkan tabel 4.5 didapatkan hasil akurasi dari model SVM lebih unggul dibandingkan model NB dengan selisih akurasi sebesar 3,15% pada data training awal. Berbeda dengan model yang menggunakan data training awal, model dengan data training oversample mendapatkan peningkatan nilai akurasi, yakni 0,75% pada metode SVM dan 2,35% pada metode NB. Hal ini menandakan pengaruh keseimbangan kelas data terhadap akurasi model dan menunjukkan efektifitas metode SMOTE dalam membantu meningkatkan akurasi model. Berdasarkan data tersebut didapat akurasi tertinggi pada model SVMS dengan akurasi sebesar 88,11 persen. Selain itu, penggunaan metode SMOTE berhasil meningkatkan akurasi dari model SVM dan NB. Meskipun demikian, metode SVM tetap unggul dibandingkan dengan metode NB dikedua skenario data. Adapun hasil perhitungan secara manual *precision*, *recall* dan *f1-score* dari setiap kelas menggunakan model kombinasi SMOTE dijabarkan pada tahapan berikut.

- a. Perhitungan performa kelas positif model SVMS

$$Pre = \frac{TP}{TP + FP} \times 100\%$$

$$Pre = \frac{466}{466 + 77} \times 100\%$$

$$Pre = \frac{466}{543} \times 100\%$$

$$Pre = 85,820\%$$

$$Rec = \frac{TP}{TP + FN} \times 100\%$$

$$Rec = \frac{466}{466 + 175} \times 100\%$$

$$Rec = \frac{466}{641} \times 100\%$$

$$Rec = 72,699\%$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec}$$

$$F1 = 2 \times \frac{85,820 \times 72,699}{85,820 + 72,699}$$

$$F1 = 2 \times \frac{6.239,028}{158,519}$$

$$F1 = 78,716\%$$

Jadi, didapatkan *presicion* 86%, *recall* 73% dan *f1-score* 79% untuk kelas positif model SVMS.



b. Perhitungan performa kelas negatif model SVMS

$$Pre = \frac{TN}{TN + FN} \times 100\%$$

$$Pre = \frac{1403}{1403 + 175} \times 100\%$$

$$Pre = \frac{1403}{1578} \times 100\%$$

$$Pre = 88,910 \%$$

$$Rec = \frac{TN}{TN + FP} \times 100\%$$

$$Rec = \frac{1403}{1403 + 77} \times 100\%$$

$$Rec = \frac{1403}{1480} \times 100\%$$

$$Rec = 94,797\%$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec}$$

$$F1 = 2 \times \frac{88,910 \times 94,797}{88,910 + 94,797}$$

$$F1 = 2 \times \frac{8.428,401}{183,707}$$

$$F1 = 91,759\%$$

Jadi, didapatkan *presicion* 89%, *recall* 95% dan *f1-score* 92% untuk kelas negatif model SVMS.

c. Perhitungan performa kelas positif model NBS

$$Pre = \frac{TP}{TP + FP} \times 100\%$$

$$Pre = \frac{436}{436 + 80} \times 100\%$$

$$Pre = \frac{466}{516} \times 100\%$$

$$Pre = 84,496\%$$

$$Rec = \frac{TP}{TP + FN} \times 100\%$$

$$Rec = \frac{436}{436 + 205} \times 100\%$$

$$Rec = \frac{436}{641} \times 100\%$$

$$Rec = 68,019\%$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec}$$

$$F1 = 2 \times \frac{84,496 \times 68,019}{84,496 + 68,019}$$

$$F1 = 2 \times \frac{5.747,333}{152,515}$$

$$F1 = 75,367\%$$

Jadi, didapatkan *presicion* 84%, *recall* 68% dan *f1-score* 75% untuk kelas positif model NBS.

d. Perhitungan performa kelas negatif model NBS

$$Pre = \frac{TN}{TN + FN} \times 100\%$$

$$Pre = \frac{1400}{1400 + 205} \times 100\%$$

$$Pre = \frac{1400}{1605} \times 100\%$$

$$Pre = 87,227 \%$$

$$Rec = \frac{TN}{TN + FP} \times 100\%$$

$$Rec = \frac{1400}{1400 + 80} \times 100\%$$

$$Rec = \frac{1400}{1480} \times 100\%$$

$$Rec = 94,595\%$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec}$$

$$F1 = 2 \times \frac{87,227 \times 94,595}{87,227 + 94,595}$$

$$F1 = 2 \times \frac{8.251,238}{181,822}$$

$$F1 = 90,762\%$$

Jadi, didapatkan *presicion* 87%, *recall* 95% dan *f1-score* 91% untuk kelas negatif model NBS.

Untuk mengetahui performa model secara lebih rinci, laporan klasifikasi akan ditampilkan dengan proses seperti pada gambar 4.37 berikut.

```
from sklearn.metrics import classification_report

def evalModel(model, X_train, y_train, X_test, y_test, y_pred, model_name):
    # Classification report
    print(f'----- Classification Report {model_name} ----- \n')
    print(classification_report(y_test, y_pred))

# Evaluasi model SVM
evalModel(svm, X_train, y_train, X_test, y_test, y_pred_svm, 'SVM')

# Evaluasi model SVM (SMOTE)
evalModel(svms, X_train_resampled, y_train_resampled,
            X_test, y_test, y_pred_svms, 'SVM & SMOTE')

# Evaluasi model Naive Bayes
evalModel(nb, X_train, y_train, X_test, y_test, y_pred_nb, 'NBC')

# Evaluasi model Naive Bayes (SMOTE)
evalModel(nbs, X_train_resampled, y_train_resampled,
            X_test, y_test, y_pred_nbs, 'NBC & SMOTE')
```

**Gambar4.37** Proses Menampilkan Laporan Klasifikasi

Proses diawali dengan melakukan import modul `classification_report` dari library `sklearn`. Laporan klasifikasi dibuat dengan memanggil fungsi `evalModel` untuk setiap metode. Adapun hasil dari pemanggilan fungsi tersebut ditampilkan pada gambar berikut.

----- Classification Report SVM -----				
	precision	recall	f1-score	support
negatif	0.87	0.97	0.91	1480
positif	0.90	0.65	0.76	641
accuracy			0.87	2121
macro avg	0.88	0.81	0.84	2121
weighted avg	0.88	0.87	0.87	2121

**Gambar 4.38** Laporan Klasifikasi SVM

Berdasarkan gambar 4.38 laporan klasifikasi model SVM didapat akurasi 87%, *precision* 88%, *recall* 87% dan *f1-score* 87% dengan jumlah data testing sebanyak 2.121 ulasan.

----- Classification Report SVM & SMOTE -----				
	precision	recall	f1-score	support
negatif	0.89	0.95	0.92	1480
positif	0.86	0.73	0.79	641
accuracy			0.88	2121
macro avg	0.87	0.84	0.85	2121
weighted avg	0.88	0.88	0.88	2121

**Gambar 4.39** Laporan Klasifikasi SVMS

Berbeda dengan laporan klasifikasi model SVM, pada SVMS didapatkan akurasi 88%, *precision* 88%, *recall* 88% dan *f1-score* 88%. Pada model SVMS terjadi peningkatan *f1-score* dikedua kelas data, yakni satu persen pada kelas negatif dan tiga persen pada kelas positif.

----- Classification Report NBC -----				
	precision	recall	f1-score	support
negatif	0.82	0.99	0.90	1480
positif	0.96	0.50	0.66	641
accuracy			0.84	2121
macro avg	0.89	0.75	0.78	2121
weighted avg	0.86	0.84	0.82	2121

**Gambar 4.40** Laporan Klasifikasi NB

Berdasarkan gambar 4.40 laporan klasifikasi model NB didapat akurasi 84%, *precision* 86%, *recall* 84% dan *f1-score* 82%.

----- Classification Report NBC & SMOTE -----				
	precision	recall	f1-score	support
negatif	0.87	0.95	0.91	1480
positif	0.84	0.68	0.75	641
accuracy			0.87	2121
macro avg	0.86	0.81	0.83	2121
weighted avg	0.86	0.87	0.86	2121

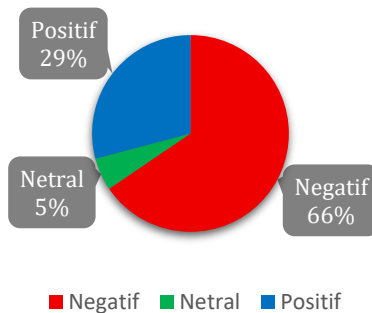
**Gambar 4.41** Laporan Klasifikasi NBS

Penggunaan SMOTE pada model NB juga meningkatkan performa model NB, didapatkan akurasi 87%, *precision* 86%, *recall* 87% dan *f1-score* 86%. Pada model NB akurasi model mengalami peningkatan sebesar tiga persen dengan penggunaan metode SMOTE. Adapun *f1-score* di kedua kelas tersebut mengalami peningkatan, pada kelas negatif meningkat satu persen dan pada kelas positif meningkat sembilan persen.

Berdasarkan data tersebut didapatkan penggunaan metode SMOTE pada kedua model memberikan peningkatan performa. Meskipun terjadi peningkatan performa, model SVM tetap unggul dikedua skenario data dibandingkan model NB.

## 6. Visualisasi

Pada tahap ini hasil dari proses yang telah dilakukan sebelumnya akan divisualisasikan menjadi informasi visual. Dalam tahap *preprocessing* menghasilkan data ulasan bersih sebanyak 11.192 ulasan yang terdiri dari 7.400 ulasan negatif, 589 ulasan netral dan 3.203 ulasan positif. Distribusi sentimen dari ulasan bersih divisualisasikan pada gambar 4.42 berikut.



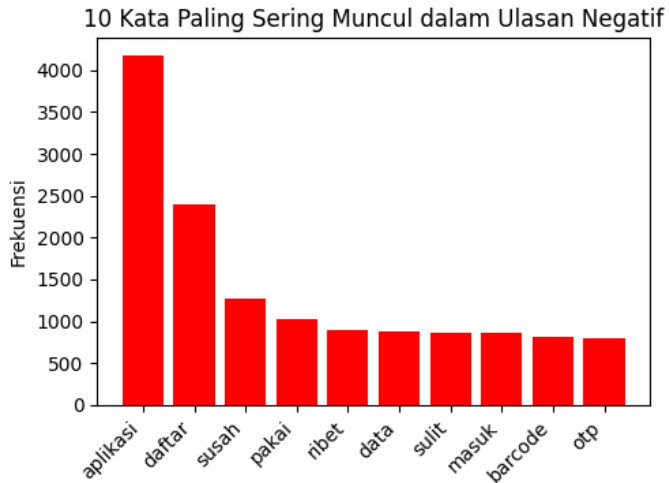
**Gambar 4.42** Distribusi Sentimen Data Ulasan Bersih

Pada gambar 4.42 menampilkan distribusi sentimen ulasan yang didominasi oleh sentimen





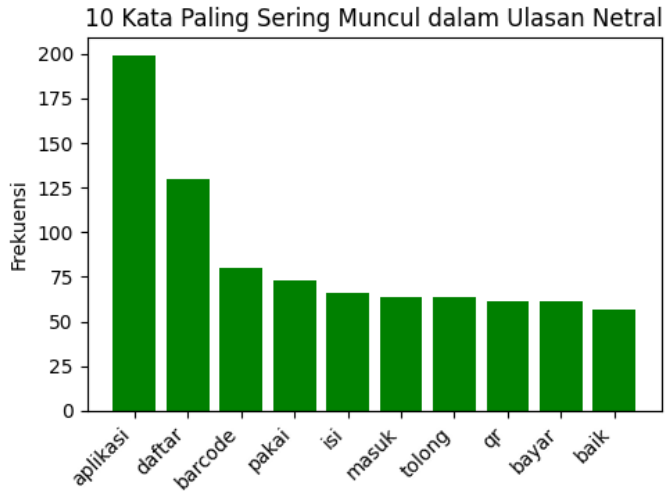
“aplikasi”, “daftar”, “susah”, “pakai”, dan “ribet”. Untuk memberikan gambaran yang lebih rinci terkait kata-kata yang sering muncul akan divisualisasikan dalam grafik frekuensi kata seperti pada gambar 4.44 berikut.



**Gambar 4.44** Grafik Frekuensi Kata Ulasan Negatif

Dalam grafik frekuensi kata ulasan negatif didapatkan kata yang paling sering yakni “aplikasi” (4174), “daftar” (2393), “susah” (1271), “pakai” (1018), “ribet” (890), “data” (876), “sulit” (864), “masuk” (855), “barcode” (814), dan “otp” (794). Data tersebut menunjukkan bahwa sebagian besar keluhan pengguna berfokus pada proses pendaftaran, yang dinilai rumit dan menyulitkan. Hal ini tercermin dari



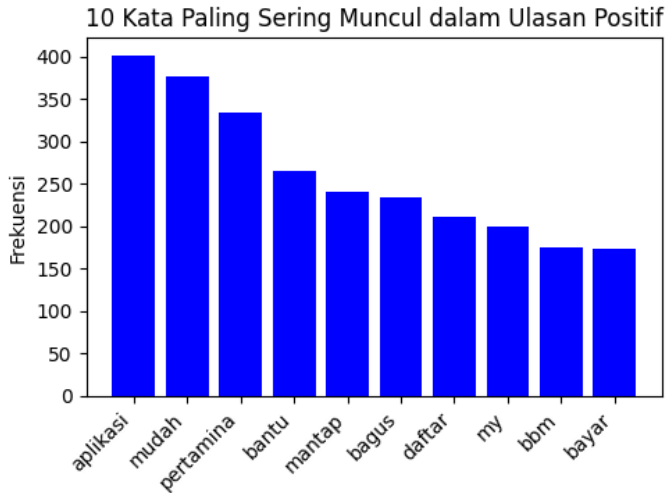


**Gambar 4.46** Grafik Frekuensi Kata Ulasan Netral

Dalam grafik frekuensi kata ulasan netral didapatkan kata yang paling sering yakni “aplikasi” (199), “daftar” (130), “barcode” (80), “pakai” (73), “isi” (66), “masuk” (64), “tolong” (64), “masuk” (61), “bayar” (61), dan “baik” (57). Data tersebut menunjukkan bahwa pengguna umumnya memberikan ulasan yang bersifat deskriptif atau menyampaikan pengalaman tanpa muatan emosi yang kuat. Kehadiran kata seperti “tolong” juga mengindikasikan adanya permintaan bantuan atau saran, namun disampaikan secara netral.

**Gambar 4.47** Wordcloud Ulasan Positif

Berdasarkan visualisasi *wordcloud* pada ulasan positif, kata-kata yang paling sering muncul antara lain “aplikasi”, “mudah”, “pertamina”, “bantu”, dan “mantap”. Untuk memberikan gambaran yang lebih rinci terkait kata-kata yang sering muncul akan divisualisasikan dalam grafik frekuensi kata seperti pada gambar 4.48 berikut.



**Gambar 4.48** Frekuensi Kata Positif

Dalam grafik frekuensi kata ulasan negatif didapatkan kata yang paling sering yakni “aplikasi” (402), “mudah” (377), “pertamina” (334), “bantu” (266), “mantap” (240), “bagus” (234), “daftar” (211), “my” (200), “bbm” (175), dan “bayar” (174). Berdasarkan data tersebut pengguna memberikan apresiasi terhadap kemudahan penggunaan aplikasi serta manfaat yang dirasakan, terutama dalam layanan pertamina yang tercermin dalam kata “mudah”, “bantu”, “mantap” dan “bagus”.

## B. Hasil Analisis Sentimen

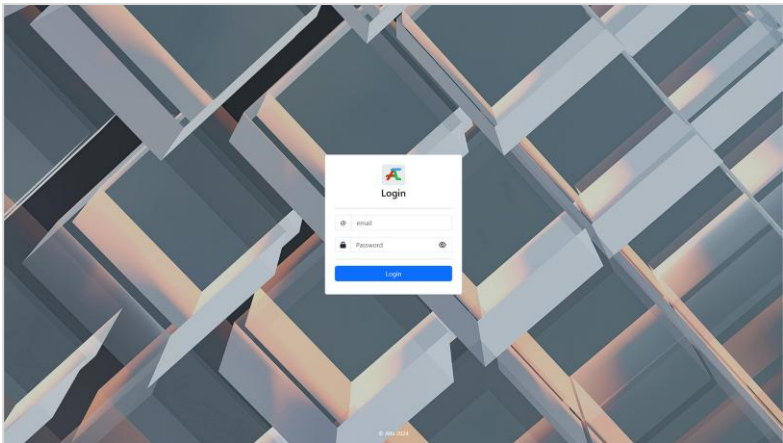
Berdasarkan data sentimen dari ulasan aplikasi MyPertamina dengan jumlah data ulasan bersih sebanyak 11,192 data yang didominasi sentimen negatif sebesar 66%. Hal ini menandakan bahwa pandangan pengguna terhadap aplikasi cenderung kurang baik secara umum. Adapun yang menjadi keluhan pengguna sebagian besar berfokus pada proses pendaftaran, yang dinilai rumit dan menyulitkan.

Pada model klasifikasi, algoritma SVM lebih unggul dibandingkan NB di dua skenario data. Dari dua skenario tersebut model dengan kombinasi metode SMOTE memiliki akurasi yang lebih baik dibandingkan model awal. Model SVMS memiliki performa tertinggi dengan akurasi 88,11%, *precision* 88%, *recall* 88% dan *f1-score* 88%, sedangkan model NBS mendapat akurasi 86,52%, *precision* 86%, *recall* 87% dan *f1-score* 86%. Perbedaan akurasi sebesar 1,59%, *precision* 2%, *recall* 1% dan *f1-score* 2% membuat model SVMS lebih unggul dibandingkan model NBS.

## C. Implementasi Sistem

### 1. Halaman Login

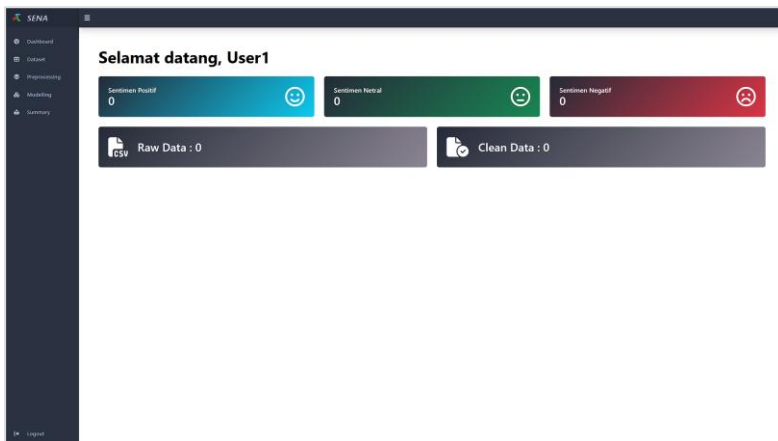
Pada halaman ini menampilkan form login untuk pengguna masuk ke dalam sistem. Adapun tampilan halaman login ditunjukkan pada gambar 4.49 berikut.



**Gambar 4.49** Implementasi Halaman Login

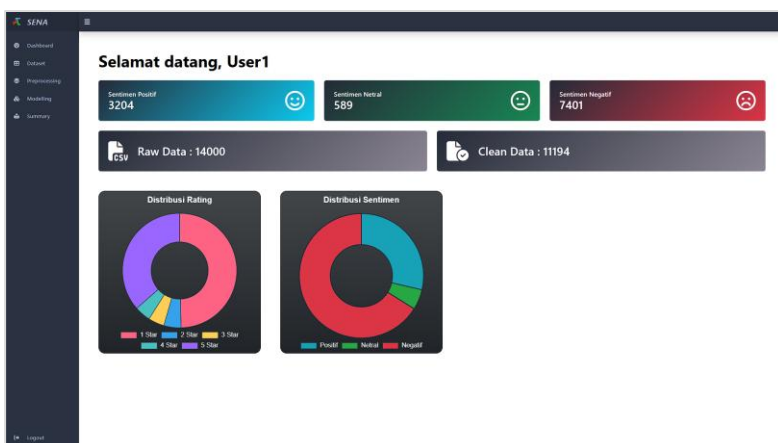
### 2. Halaman *Dashboard*

Halaman *dashboard* berisi informasi dari dataset awal (raw) dan dataset bersih (clean). Berikut hasil implementasi dari halaman *dashboard*.



**Gambar 4.50** Implementasi Halaman Dashboard Awal

Gambar 4.50 menampilkan *dashboard* awal ketika pengguna belum melakukan tahap *preprocessing*. Adapun tampilan *dashboard* ketika pengguna telah melakukan tahap *preprocessing* dapat dilihat pada gambar 4.51 berikut.



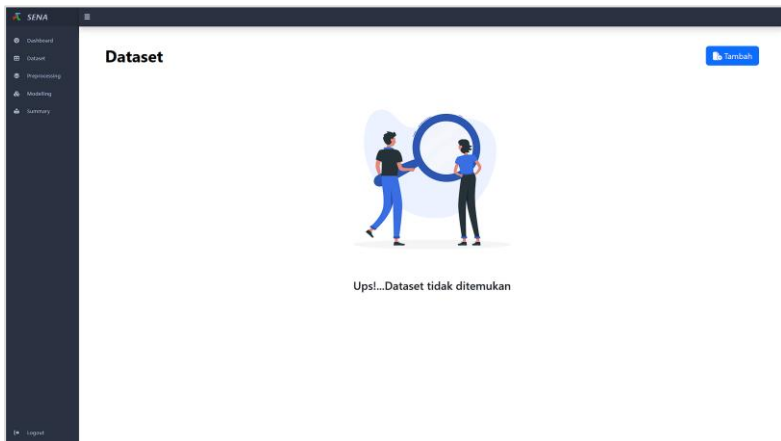
**Gambar 4.51** Implementasi Halaman Dashboard Akhir



Gambar 4.51 menampilkan informasi dari dataset setelah pengguna melakukan *preprocessing*. Grafik distribusi rating memvisualisasikan sebaran sentimen pada data mentah, sedangkan grafik distribusi sentimen memvisualisasikan sebaran sentimen pada data bersih.

### 3. Halaman Dataset

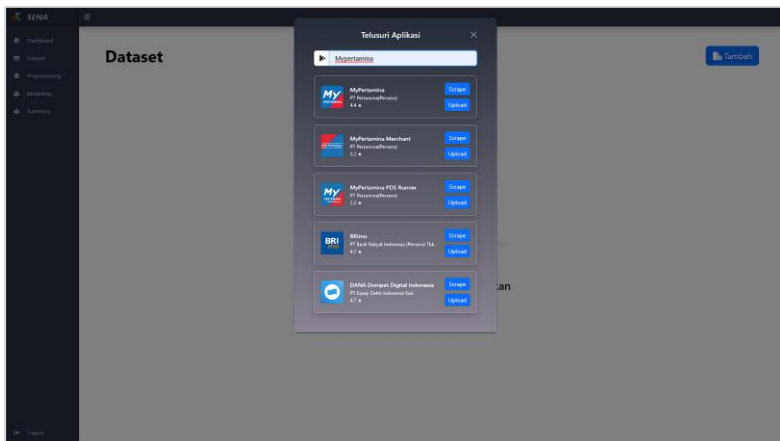
Pada halaman dataset pengguna dapat memasukkan data ulasan yang akan digunakan untuk analisis sentimen. Adapun hasil implementasi dari halaman dataset ditunjukkan pada gambar berikut.



**Gambar 4.52** Implementasi Halaman Dataset Awal

Gambar 4.52 merupakan tampilan awal ketika pengguna tidak memiliki data ulasan untuk diproses. Pada tampilan awal ini pengguna dapat memasukkan

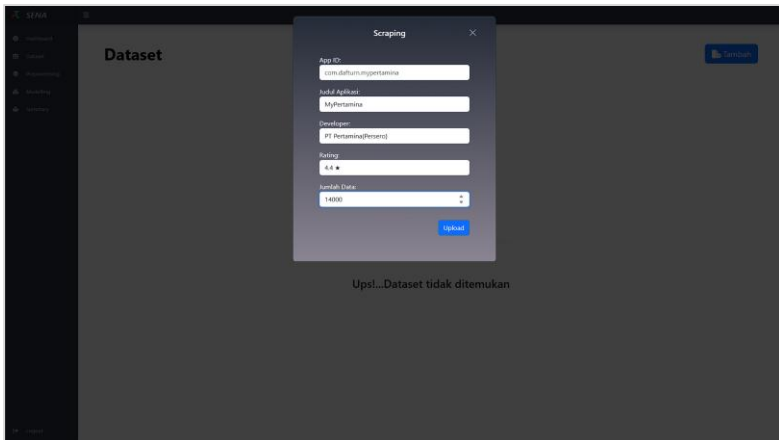
data ulasan melalui tombol tambah yang disediakan. Tombol tambah ini akan mengarahkan pengguna ke modal pencarian aplikasi. Modal ini digunakan pengguna untuk mencari aplikasi target untuk dilakukan analisis sentimen. Adapun tampilan modal pencarian aplikasi ditampilkan seperti pada gambar 4.53 berikut.



**Gambar 4.53** Implementasi Modal Pencarian Aplikasi

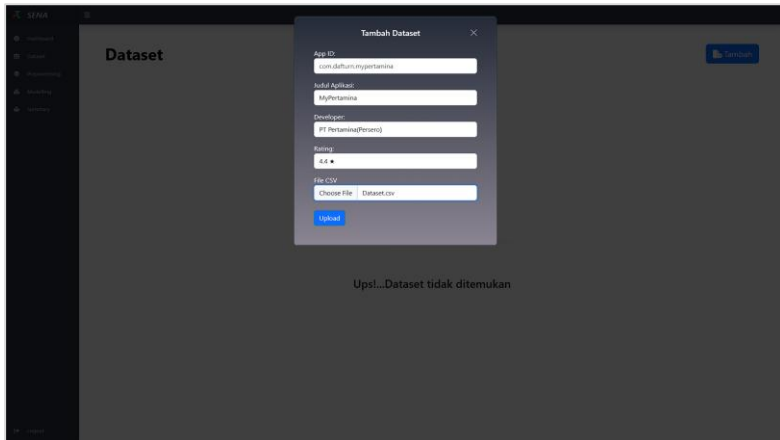
Pengguna dapat memasukkan nama aplikasi target pada *field* pencarian, kemudian sistem akan menampilkan lima aplikasi yang sesuai dengan input tersebut. Untuk menambahkan data ulasan dari aplikasi yang dipilih terdapat dua metode yang dapat digunakan pengguna, yakni *scraping* data dan *upload* file. Tombol scrape akan mengarahkan pengguna ke

modal *scraping* untuk melakukan proses *scraping* data, sedangkan tombol *upload* mengarahkan pengguna ke modal tambah dataset. Adapun tampilan dari modal kedua metode ini ditampilkan pada gambar berikut.



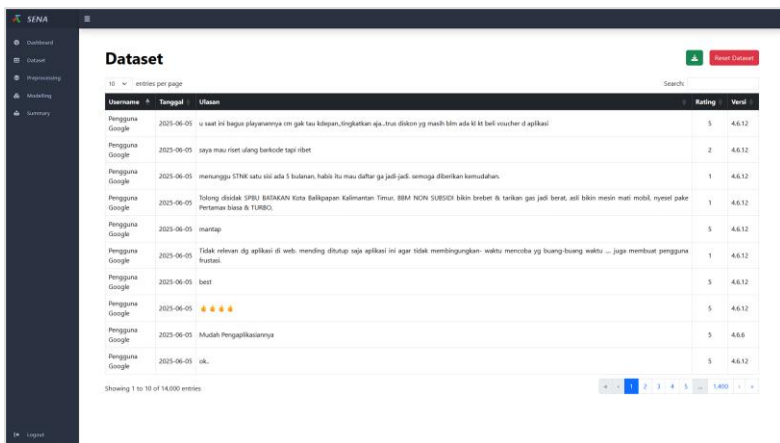
**Gambar 4.54** Implementasi Modal Scraping Data

Pada modal *scraping* akan menampilkan informasi dari aplikasi target yang telah dipilih sebelumnya dan form jumlah data ulasan yang akan diambil. Proses *scraping* akan dimulai ketika pengguna menekan tombol *upload* pada modal ini.



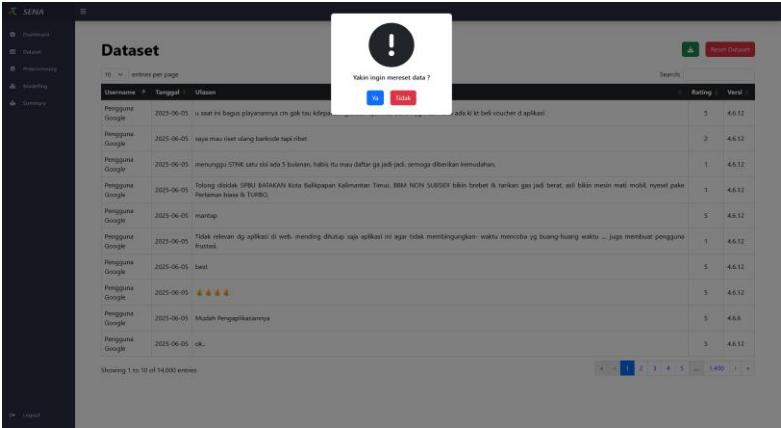
**Gambar 4.55 Implementasi Modal Upload File**

Pada modal tambah dataset pengguna menambahkan data ulasan melalui input file CSV. Setelah proses memasukkan data selesai, data ulasan akan ditampilkan seperti pada gambar 4.56 berikut.



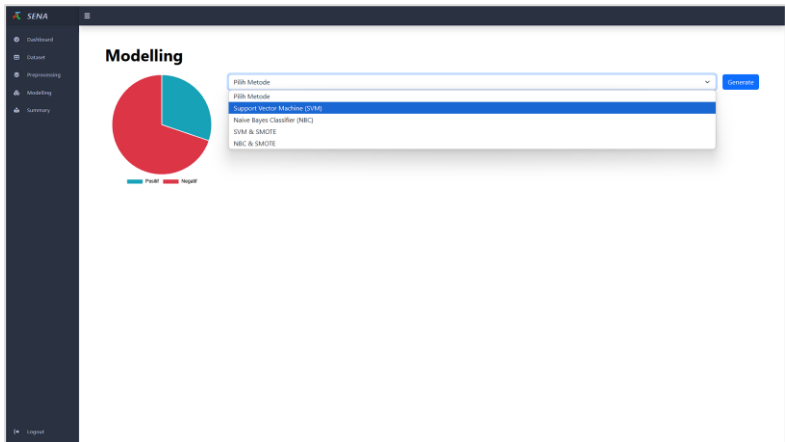
**Gambar 4.56 Implementasi Halaman Dataset Akhir**

Pada tahap ini, pengguna dapat melihat data ulasan dalam bentuk tabel yang dapat disorting berdasarkan keperluan. Pengguna dapat mengunduh data ulasan yang ditampilkan dengan tombol unduh dalam format file CSV. Untuk dapat melakukan proses analisis pada dataset lain pengguna dapat menghapus dataset saat ini dengan tombol reset dataset. Tombol reset dataset akan mengarahkan ke dalam modal untuk mengkonfirmasi tindakan pengguna. Adapun tampilan modal reset dataset seperti pada gambar 4.75 berikut.



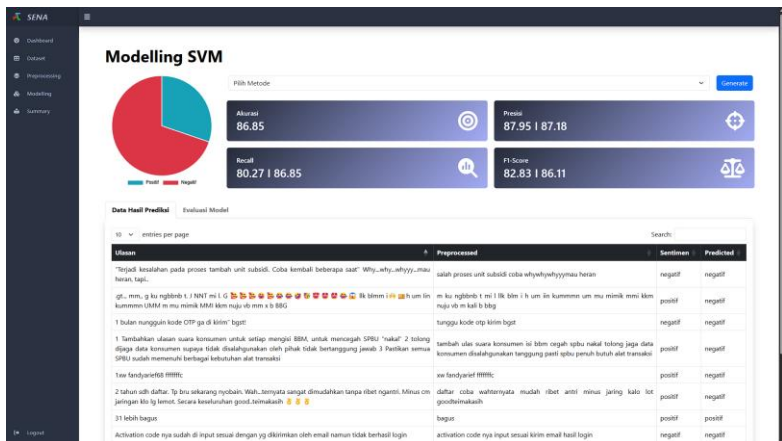
**Gambar 4.57 Implementasi Modal Reset Dataset**





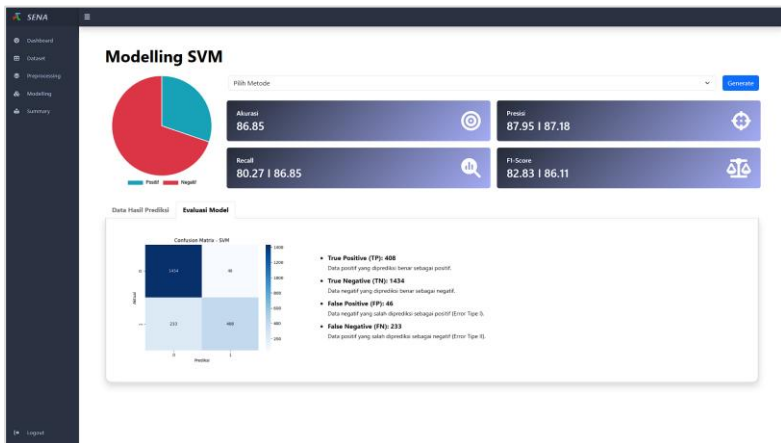
**Gambar 4.59** Implementasi Halaman Menu Modeling

Gambar 4.59 menampilkan halaman utama yang menampilkan grafik distribusi kelas data training dan menu metode modeling. Berikut hasil pengujian dan evaluasi model pada gambar 4.60 dan 4.61.



**Gambar 4.60** Implementasi Hasil Modeling Tab I

Gambar 4.60 menampilkan hasil evaluasi model, yakni akurasi, *precision*, *recall* dan *f1-score*. Hasil pengujian dari model akan ditampilkan pada tab hasil prediksi dalam bentuk tabel yang dapat disorting sesuai keperluan pengguna.



**Gambar 4.61** Implementasi Hasil Modeling Tab II

Pada gambar 4.61 menampilkan tab hasil evaluasi model, yakni confusion matrix. Confusion matrix dari model ditampilkan beserta penjelasan singkat dari gambar tersebut.



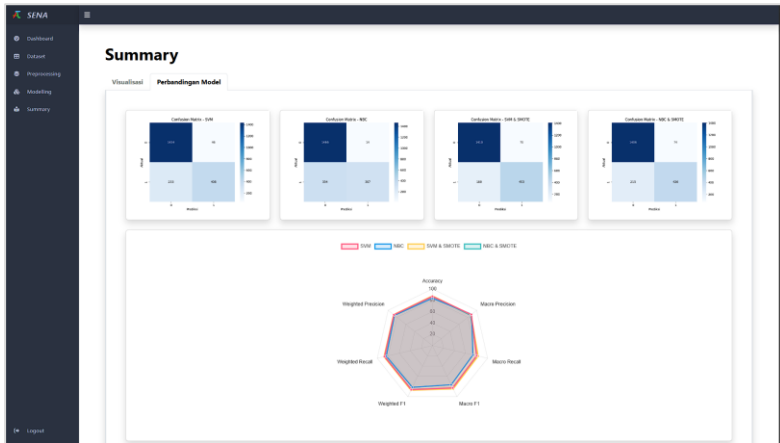
## 6. Halaman Summary

Pada halaman ini menampilkan visualisasi dari hasil proses yang telah dilakukan pada halaman sebelumnya. Dalam halaman summary terdapat dua tab, yakni tab visualisasi yang memberikan visualisasi dari data ulasan bersih dan tab perbandingan model yang memberikan visualisasi dari perbedaan dari model yang sudah dibuat. Adapun implementasi dari halaman ini dapat dilihat seperti pada gambar 4.62 dan 4.63 berikut.



**Gambar 4.62** Implementasi Halaman Summary Tab 1

Gambar 4.62 menampilkan tab visualisasi yang berisi data visual dari setiap kelas sentimen. Data visual tersebut berupa *wordcloud* dan grafik frekuensi kata dari data ulasan bersih.



**Gambar 4.63** Implementasi Halaman Summary Tab II

Pada tab perbandingan model berisi gambar dari confusion matrix yang telah dilatih dan perbandingan performa dari setiap model dalam bentuk grafik radar. Dalam grafik ini memiliki tujuh label yakni akurasi, *macro precision*, *macro recall*, *macro f1*, *weighted precision*, *weighted recall* dan *weighted f1* untuk perbandingan model yang komprehensif.

## D. Pengujian

Pada tahap pengujian dilakukan menggunakan *Black Box Testing*. *Black Box Testing* adalah metode pengujian sistem berdasarkan fungsionalitas tanpa perlu mengetahui sistem yang berjalan di belakang. Pengujian ini digunakan untuk mengetahui seberapa baik sistem bekerja. Adapun

hasil pengujian dari *Black Box Testing* ditampilkan pada tabel berikut.

**Tabel 4.6** Pengujian

No	Fitur	Skenario	Aksi	Status
1	Halaman Login			
	Login	Data sesuai	Pengguna berhasil masuk dan diarahkan ke halaman <i>Dashboard</i>	Berhasil
		Data tidak sesuai	Pengguna tetap di halaman Login dan muncul pesan peringatan (alert) "Username atau password salah."	Berhasil
	Akses ilegal ke halaman internal tanpa Login	<i>Dashboard</i>	Sistem secara otomatis mengarahkan kembali ke halaman Login dan menampilkan pesan peringatan (alert) "Harap login terlebih dahulu"	Berhasil
		Dataset		
		<i>Preprocessing</i>		
		Modeling		
<i>Summary</i>				
2	Layout			
	Menu navigasi (Sidebar)	<i>Dashboard</i>	Mengarahkan ke halaman Dashedboard	Berhasil
		Dataset	Mengarahkan ke halaman Dataset	Berhasil
		<i>Preprocessing</i>	Mengarahkan ke halaman <i>Preprocessing</i>	Berhasil
		Modeling	Mengarahkan ke halaman Modeling	Berhasil

		Summary	Mengarahkan ke halaman Summary	Berhasil
		Logout	Mengarahkan ke halaman Login	Berhasil
3	Halaman <i>Dashboard</i>			
	<i>Dashboard</i>	Menampilkan informasi dataset	Menampilkan jumlah distribusi sentimen setiap kelas	Berhasil
			Menampilkan jumlah data sebelum dan sesudah diproses	Berhasil
			Menampilkan grafik distribusi rating	Berhasil
			Menampilkan grafik distribusi sentimen	Berhasil
4	Halaman Dataset			
	Pencarian aplikasi	Aplikasi ditemukan	Menampilkan 5 aplikasi sesuai keyword	Berhasil
		Aplikasi tidak ditemukan	Menampilkan pesan peringatan "Aplikasi tidak ditemukan !!"	Berhasil
	Scraping data	Input data integer	Melakukan proses scraping, kemudian tampilkan tabel data ulasan hasil scraping	Berhasil
		Input data float	Menampilkan pesan peringatan	Berhasil
	Upload file	Input file CSV	Menampilkan tabel data ulasan	Berhasil
		Input file selain CSV	File tidak akan ditampilkan	Berhasil

	Download file	Dataset tersedia	Tombol download ditampilkan dan dapat digunakan	Berhasil
		Dataset belum tersedia	Tombol download disembunyikan	Berhasil
	Reset dataset	Konfirmasi “Ya” pada modal reset	Menghapus seluruh data hasil proses user dan menampilkan halaman dataset awal	Berhasil
		Konfirmasi “Tidak” pada modal reset	Kembali ke halaman dataset dan menampilkan tabel data ulasan	Berhasil
	Filter data	Filtering berdasarkan kategori	Data diurutkan sesuai filter yang diterapkan	Berhasil
	Search data	Data ditemukan	Menampilkan data yang ditemukan	Berhasil
		Data tidak ditemukan	Menampilkan pesan “No matching records found”	Berhasil
	Halaman <i>Preprocessing</i>			
	Auto Preprocessing	Navigasi ke halaman <i>preprocessing</i> setelah dataset dimasukkan	Melakukan <i>preprocessing</i> dan menampilkan hasil dalam tabel	Berhasil
5	Download file	Dataset tersedia	Tombol download ditampilkan dan dapat digunakan	Berhasil
		Dataset belum tersedia	Tombol download disembunyikan	Berhasil
	Filter data	Filtering berdasarkan kategori	Data diurutkan sesuai filter yang diterapkan	Berhasil

	Search data	Data ditemukan	Menampilkan data yang ditemukan	Berhasil
		Data tidak ditemukan	Menampilkan pesan “No matching records found”	Berhasil
	Halaman Modeling			
6	Visualisasi data training	Dataset bersih tersedia	Membuat dan menampilkan grafik distribusi kelas pada data training	Berhasil
	Modeling	Pilih metode modeling dan generate	Melatih model dilanjutkan dengan evaluasi model kemudian hasil evaluasi model akan ditampilkan	Berhasil
	Halaman Summary			
7	Auto generate visual data	Navigasi ke halaman summary setelah dilakukan <i>preprocessing</i>	Menampilkan visualisasi data ulasan bersih pada tab visualisasi	Berhasil
		Navigasi ke halaman summary setelah dilakukan modeling	Menampilkan visualisasi data perbandingan model pada tab perbandingan model	Berhasil

Berdasarkan hasil pengujian fungsionalitas tersebut, sistem analisis sentimen berbasis website yang dibuat telah berjalan dengan baik dan sesuai dengan kebutuhan yang ada.



## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Berdasarkan hasil studi yang dilakukan dapat disimpulkan sebagai berikut :

1. Sistem analisis sentimen yang digunakan untuk menganalisis ulasan pada aplikasi MyPertamina dengan membandingkan metode SVM dan NB berjalan dengan baik. Data ulasan yang digunakan sebesar 14.000 ulasan, jumlah ini berkurang menjadi 11.192 ulasan setelah dilakukan tahap preproccesing dengan jumlah sentimen negatif 7.400 ulasan, sentimen netral 589 ulasan dan sentimen positif 3.203 ulasan.
2. Model klasifikasi SVM lebih unggul dibandingkan NB di dua skenario data. Dari dua skenario tersebut model dengan kombinasi metode SMOTE memiliki akurasi yang lebih baik dibandingkan model awal. Model SVMS memiliki performa tertinggi dengan akurasi 88,11%, *precision* 88%, *recall* 88% dan *f1-score* 88%, sedangkan model NBS mendapat akurasi 86,52%, *precision* 86%, *recall* 87% dan *f1-score* 86%. Perbedaan akurasi sebesar 1,59%, *precision* 2%, *recall* 1% dan *f1-score* 2% membuat model SVMS lebih unggul dibandingkan model NBS. Berdasarkan data tersebut penggunaan



SMOTE berhasil meningkatkan performa dari model klasifikasi.

3. Sistem analisis sentimen telah berhasil diintegrasikan ke dalam aplikasi berbasis website, sehingga memungkinkan pengguna untuk melakukan seluruh proses analisis, mulai dari input data hingga visualisasi hasil, melalui antarmuka yang dirancang agar mudah digunakan oleh pengguna non-teknis. Berdasarkan hasil pengujian, sistem berjalan dengan baik dan sesuai dengan kebutuhan yang ada.

## **B. Saran**

Berdasarkan hasil penelitian yang telah dilakukan, peneliti menyadari bahwa masih terdapat beberapa keterbatasan dalam studi ini. Oleh karena itu, penelitian selanjutnya diharapkan dapat melakukan pengembangan lebih lanjut. Beberapa saran yang dapat diberikan antara lain sebagai berikut:

1. Metode labeling yang digunakan dalam studi ini berbasis rating yang memiliki keterbatasan dalam hal akurasi, karena tidak semua ulasan dengan rating tertentu mencerminkan sentimen yang sesuai. Pada penelitian selanjutnya perlu mengeksplorasi metode pelabelan otomatis berbasis *text classification* atau pendekatan leksikal untuk hasil yang lebih akurat.

2. Melakukan *custom hyperparameter tuning* untuk mengoptimalkan performa model. Pendekatan seperti *Grid Search* dapat digunakan untuk mencari kombinasi parameter terbaik, sehingga hasil klasifikasi menjadi lebih akurat.



## DAFTAR PUSTAKA

- Alzami, F., Udayanti, E. D., Prabowo, D. P., & Megantara, R. A. (2020). Document Preprocessing with TF-IDF to Improve the Polarity Classification Performance of Unstructured Sentiment Analysis. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 235–242. <https://doi.org/10.22219/kinetik.v5i3.1066>
- AppBrain. (2025). Number of Android applications on Google Play (Mar 2025). <https://www.appbrain.com/stats/number-of-android-apps>
- Ardianto, R., Rivanie, T., Alkhalifi, Y., Septia Nugraha, F., & Gata, W. (2020). SENTIMENT ANALYSIS ON E-SPORTS FOR EDUCATION CURRICULUM USING NAIVE BAYES AND SUPPORT VECTOR MACHINE. In *Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)* (Vol. 13, Issue 2).
- Badan Pusat Statistik. (2024). *Statistik Telekomunikasi Indonesia 2023* (Vol. 12).
- Bahtiar, S. A. H., Dewa, C. K., & Luthfi, A. (2023). Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling. *Journal of Information Systems and Informatics*, 5(3), 915–927. <https://doi.org/10.51519/journalisi.v5i3.539>
- Bordoloi, M., & Biswas, S. K. (2023). Sentiment analysis: A survey on design framework, applications and future scopes. *Artificial Intelligence Review*, 56(11), 12505–12560. <https://doi.org/10.1007/s10462-023-10442-2>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).

- Chong, H. Y., & Diamantopoulos, A. (2020). Integrating advanced technologies to uphold security of payment: Data flow diagram. *Automation in Construction*, 114. <https://doi.org/10.1016/j.autcon.2020.103158>
- Dimas Sulistiyo, R., & Shihab, M. R. (2023). Transformasi Digital dalam Pelayanan Surat Izin Mengemudi (SIM): Studi Kasus Korlantas Polri. *Technomedia Journal*, 8(2SP), 189–204. <https://doi.org/10.33050/tmj.v8i2sp.2064>
- Handayanna, F., & Ayu Nur Wulandari, D. (2024). *Penerapan Naive Bayes Terhadap Sentimen Analisis Media Sosial Twitter Pengguna Commuter Line Implementation of Naive Bayes for Sentiment Analysis of Commuter Line Twitter Social Media* (Vol. 23, Issue 4).
- Hasanli, H., & Rustamov, S. (2019). Sentiment Analysis of Azerbaijani tweets Using Logistic Regression, Naive Bayes and SVM. *2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT)*, 1–7. <https://doi.org/10.1109/AICT47866.2019.8981793>
- Imelda, I., & Arief Ramdhan Kurnianto. (2023). Naïve Bayes and TF-IDF for Sentiment Analysis of the Covid-19 Booster Vaccine. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 7(1), 1–6. <https://doi.org/10.29207/resti.v7i1.4467>
- Kusumaningrum, R., Nisa, I. Z., Nawangsari, R. P., & Wibowo, A. (2021). Sentiment analysis of Indonesian hotel reviews: from classical machine learning to deep learning. *International Journal of Advances in Intelligent Informatics*, 7(3), 292–303. <https://doi.org/10.26555/ijain.v7i3.737>
- Madyatmadja, E. D., Winata, B. C. S., Pradhan, E., Yasmina, F. P., Adrian, F. A., Mahardhika, R., Christian, & Sembiring, D. J. M. (2025). Sentiment Analysis on User Reviews of Snapchat in Indonesia. *Journal of Computer Science*, 21(1), 158–167. <https://doi.org/10.3844/jccsp.2025.158.167>
- Mehra, S., & Choudhury, T. (2018). Sentiment Analysis of User Entered Text. *2018 International Conference on*

- Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 457–461.  
<https://doi.org/10.1109/CTEMS.2018.8769136>
- Mustasaruddin, M., Budianita, E., Fikry, M., & Yanto, F. (2023). Klasifikasi Sentiment Review Aplikasi MyPertamina Menggunakan Word Embedding FastText dan SVM (Support Vector Machine). *Jurnal Sistem Komputer Dan Informatika (JSON)*, 4(3), 526.  
<https://doi.org/10.30865/json.v4i3.5695>
- Nandwani, P., & Verma, R. (2021). A review on sentiment analysis and emotion detection from text. In *Social Network Analysis and Mining* (Vol. 11, Issue 1). Springer.  
<https://doi.org/10.1007/s13278-021-00776-6>
- Onyenwe, I., Nwagbo, S., Mbeledogu, N., & Onyedima, E. (2020). The impact of political party/candidate on the election results from a sentiment analysis perspective using #AnambraDecides2017 tweets. *Social Network Analysis and Mining*, 10(1).  
<https://doi.org/10.1007/s13278-020-00667-2>
- Prastyo, P. H., Ardiyanto, I., & Hidayat, R. (2020, October 26). Indonesian Sentiment Analysis: An Experimental Study of Four Kernel Functions on SVM Algorithm with TF-IDF. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*.  
<https://doi.org/10.1109/ICDABI51230.2020.9325685>
- Pressman, R. S., & Maxim, B. R. (2015). *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH, EIGHTH EDITION*.
- Putra, R. A., Novita, R., Ahsyar, T. K., & Zarnelly. (2024). Implementation of Classification Algorithm for Sentiment Analysis: Measuring App User Satisfaction. *Teknika*, 13(2), 204–212. <https://doi.org/10.34148/teknika.v13i2.827>
- Quran NU Online. (n.d.). *Surat Al-Isra' Ayat 36: Arab, Latin, Terjemah dan Tafsir Lengkap | Quran NU Online*. <https://Quran.Nu.or.Id/al-Isra'/36>. Retrieved 13 February 2025, from <https://quran.nu.or.id/al-isra'/36>

- Raghunathan, N., & Saravanakumar, K. (2023). Challenges and Issues in Sentiment Analysis: A Comprehensive Survey. *IEEE Access*, 11, 69626–69642. <https://doi.org/10.1109/ACCESS.2023.3293041>
- Sabilly, N., & Agung Nugroho, F. (2023). *SENTIMENT ANALYSIS OF LEAGUE OF LEGENDS: WILD RIFT REVIEWS ON GOOGLE PLAY USING NAÏVE BAYES CLASSIFIER*. 12(1).
- Talaat, A. S. (2023). Sentiment analysis classification system using hybrid BERT models. *Journal of Big Data*, 10(1). <https://doi.org/10.1186/s40537-023-00781-w>
- 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>
- Wang, L., Han, M., Li, X., Zhang, N., & Cheng, H. (2021). Review of Classification Methods on Unbalanced Data Sets. *IEEE Access*, 9, 64606–64628. <https://doi.org/10.1109/ACCESS.2021.3074243>
- Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7), 5731–5780. <https://doi.org/10.1007/s10462-022-10144-1>
- Zainudin, Ahmad., & Yunant, A. Ampuh. (2019). *Proceedings, IES 2019: IES, International Electronics Symposium: Surabaya, Indonesia, September 27-28, 2019: the Role of Techno-intelligence in Creating an Open Energy System Towards Energy Democracy*. IEEE.
- Zhao, H., Yang, M., Bai, X., & Liu, H. (2024). A Survey on Multimodal Aspect-Based Sentiment Analysis. *IEEE Access*, 12, 12039–12052. <https://doi.org/10.1109/ACCESS.2024.3354844>

## DAFTAR LAMPIRAN

### LAMPIRAN 1 : Pengesahan Ujian Komprehensif



KEMENTERIAN AGAMA  
UNIVERSITAS ISLAM NEGERI WALISONGO  
FAKULTAS SAINS DAN TEKNOLOGI  
Jl. Prof. Dr. Hamka Ngaliyan Semarang  
Telp.024-7601295 Fax.7615387

#### PENGESAHAN UJIAN KOMPREHENSIF

Naskah proposal skripsi berikut ini:

Judul : PERBANDINGAN ALGORITMA SVM DAN NAIVE  
BAYES UNTUK ANALISIS SENTIMEN PENGGUNA  
APLIKASI DI PLAYSTORE BERBASIS WEBSITE  
Penulis : Alfa Rifa Luky Achmad Rayendra  
NIM : 2108096099  
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang komprehensif oleh Dewan Penguji  
Fakultas Sains dan Teknologi UIN Walisongo Semarang pada  
Kamis, 22 Mei 2025.

Semarang, 2 Juni 2025

#### DEWAN PENGUJI

Penguji I

Dr. Khotibul Umam, M.Kom  
NIP.197908272011011007

Penguji II

Adzhal Arwani Mahfudh, M.Kom  
NIP. 199107032019031006

Penguji III

Maya Rini Handayani, M.Kom  
NIP. 197605052011012007

Penguji IV

Mokhamad Iklil Mustofa, M.Kom  
NIP. 198808072019031010





## **LAMPIRAN 2 : Riwayat Hidup**

### **RIWAYAT HIDUP**

#### **A. Identitas Diri**

Nama lengkap : Alfa Rifa Luky Achmad Rayendra  
TTL : Bandar Jaya, 20 Januari 2003  
Alamat : Glagahwangi 05/03, Ds. Glagahwangi,  
Kec. Polanharjo, Kab. Klaten  
No. HP : 087862766601  
E-mail : [sshunter2013@gmail.com](mailto:sshunter2013@gmail.com)

#### **B. Riwayat Pendidikan Formal**

1. SD Negeri 2 Meluai
2. SMP Negeri 3 Semendawai Barat
3. SMA Negeri 1 Polanharjo