

SKRIPSI

**ANALISIS SENTIMEN ULASAN APLIKASI BAREKSA DENGAN
PENDEKATAN *LEXICON-BASED*, SELEKSI FITUR *CHI-SQUARE*,
DAN *SUPPORT VECTOR MACHINE***

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI

UNIVERSITAS TRUNOJOYO MADURA

Oleh :

FATHONY SYAENNULLOH

200411100073

Dosen Pembimbing 1 : Dr. Fika Hastarita Rachman, S.T., M.Eng.

Dosen Pembimbing 2 : Abdullah Basuki Rahmat, S.Si., M.T.

PROGRAM STUDI INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2024

HALAMAN JUDUL

ANALISIS SENTIMEN ULASAN APLIKASI BAREKSA DENGAN PENDEKATAN LEXICON-BASED, SELEKSI FITUR CHI-SQUARE, DAN SUPPORT VECTOR MACHINE

SKRIPSI

Diajukan untuk Memenuhi Persyaratan Penyelesaian
Studi Strata (S1) dan Memperoleh Gelar Sarjana Komputer (S.Kom)
di Universitas Trunojoyo Madura

Disusun Oleh :

FATHONY SYAENNULLOH

200411100073

PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

2024

HALAMAN PENGESAHAN

ANALISIS SENTIMENT ULASAN APLIKASI BAREKSA DENGAN
PENDEKATAN LEXICON-BASED, SELEKSI FITUR CHI-SQUARE, DAN
SUPPORT VECTOR MACHINE
SKRIPSI

Skripsi ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer (S.Kom) di Universitas Trunojoyo Madura

Oleh:

Nama : Fathony Syaennulloh
NIM : 200411100073

Tanggal Sidang : 12 Agustus 2024
Disetujui oleh Tim Penguji Skripsi

Dr. Fika Hastarita Rachman, S.T., M.Eng.
NIP. 19830305 200604 2 002

 (Pembimbing I)

Abdullah Basuki Rahmat, S.Si., M.T.
NIP. 19740610 200812 1 002

 (Pembimbing II)

HUSNI, S.Kom., MT.
NIP. 19790722 200312 1 006

 (Penguji I)

Ika Oktavia Suzanti, S.Kom., M.Cs.
NIP. 19881018 201504 2 004

 (Penguji II)

Yudha Dwi Putra Negara, S.Kom., M.Kom.
NIP. 19890530 201903 1 012

 (Penguji III)

Bangkalan, 18 Oktober 2024

Mengetahui,
Ketua Jurusan Teknik Informatika

Dr. Yeni Kustiyahningsih, S.Kom., M.Kom
NIP. 19770921 200812 2 002

HALAMAN PERNYATAAN ORSINARALITAS

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi yang saya tulis dengan judul :

**“ANALISIS SENTIMEN ULASAN APLIKASI BAREKSA DENGAN
PENDEKATAN *LEXICON-BASED*, SELEKSI FITUR *CHI-SQUARE*, DAN
SUPPORT VECTOR MACHINE”.**

1. Adalah asli, bukan merupakan karya pihak lain serta belum pernah diajukan untuk mendapatkan gelar akademik Sarjana Komputer baik di Universitas Trunojoyo Madura maupun di perguruan tinggi lain di Indonesia.
2. Tidak terdapat karya atau pendapat pihak lain yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis telah dijadikan acuan dan disebutkan dalam referensi pada naskah ini.

Apabila dikemudian hari skripsi ini terbukti sebagian dan seluruhnya merupakan hasil plagiasi atau terdapat hal - hal yang tidak sesuai dalam pernyataan diatas, maka saya sanggup menerima sanksi akademis yang berlaku dengan segala akibat hukumnya sesuai dengan peraturan di Universitas Trunojoyo Madura dan atau peraturan perundang - undangan yang berlaku.

Bangkalan, 18 Oktober 2024

Yang Menyatakan,

Fathony Syaennulloh
200411100073

KATA PENGANTAR

Alhamdulillah segala puji dan syukur penulis panjatkan kepada Allah SWT atas segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “ANALISIS SENTIMEN ULASAN APLIKASI BAREKSA DENGAN PENDEKATAN *LEXICON-BASED*, SELEKSI FITUR *CHI-SQUARE*, DAN *SUPPORT VECTOR MACHINE*”. Penyusunan skripsi ini dapat terselesaikan tidak lepas dari bimbingan, dan arahan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar - besarnya kepada :

1. Allah SWT atas semua segala rahmat, ridho dan merestui segala apa yang saya panjatkan kepadanya, sehingga saya dapat menyelesaikan skripsi ini dengan lancar
2. Kedua Orang tua saya yakni bapak Arifin dan ibu Suriyah mengucapkan terimakasih banyak telah menjadi *support system* saya dalam perkuliahan tanpa doa dan restu serta dukungan orang tua saya tidak bisa dengan lancar dalam penggerjaan skripsi ini.
3. Saudara saya yang bernama kakak Choirun Nisa’, S.Pd., M.Pd, Nur Aini, S.Psi serta adik tersayang saya yakni Rafaizan Ikbar Alfarizqi yang telah menjadi alasan saya untuk semangat menyelesaikan skripsi ini.
4. Ibu Dr. Fika Hastarita Rachman, S.T., M.Eng dan Bapak Abdullah Basuki Rahmat, S.Si., M.T. selaku Dosen pembimbing. Saya mengucapkan terimakasih banyak atas bimbingan yang telah memberikan arahan, dukungan serta ilmu dan pandangan terhadap saya baik itu dalam skripsi maupun masa depan saya sehingga saya bisa menyelesaikan skripsi saya.

5. Saya ucapkan terimakasih kepada para dosen dan staf fakultas teknik lebih khususnya dosen prodi teknik informatika yang telah memberikan banyak ilmu kepada saya selama perkuliahan.
6. Saya ucapkan terimakasih banyak kepada Fieza Naurah Aprilia selaku pasangan saya sekaligus teman, sahabat dan rekan saya dalam berprestasi yang telah menemani saya saat awal sampai akhir penggerjaan skripsi saya. Semangat untuk skripsimu dan semoga dilancarkan dalam penggerjaannya serta semoga kita sukses bersama.
7. Saya ucapkan terimakasih juga kepada organisasi yang saya cintai dan saya banggakan yakni UKM Triple-C beserta seluruh anggota khususnya pada masa creative 2022 semoga kalian semua sukses *see u on top guys!*.
8. Kepada teman sircle saya THE CLOWNS yang terkadang menjadi penghambat dan terkadang juga menjadi penyemangat saya juga dalam penggerjaan skripsi saya. Semoga kita sukses Bersama ya guys, aamiin!.

ABSTRAK

Investasi merupakan pengumpulan aset untuk memperoleh keuntungan di masa depan. Dalam beberapa tahun terakhir, Indonesia mengalami peningkatan signifikan dalam bidang investasi digital, sehingga banyak bermunculan platform investasi digital. Namun, banyak masyarakat yang kebingungan dalam memilih platform investasi digital yang tepat. Oleh karena itu, penelitian ini mengusulkan pendekatan untuk menganalisis aplikasi investasi digital, salah satunya Bareksa, dengan menggunakan pendekatan lexicon-based untuk pelabelan, Support Vector Machine (SVM) dengan kernel linear, seleksi fitur Chi-Square, dan metode Synthetic Minority Over-sampling Technique (SMOTE). Penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi Bareksa. Salah satu tantangan utama dalam analisis sentimen adalah banyaknya fitur tidak relevan (noisy features) dalam data teks yang dapat menurunkan kinerja model. Untuk mengatasi masalah ini, digunakan seleksi fitur Chi-Square untuk menyaring fitur-fitur yang tidak signifikan. Selain itu, penelitian ini juga menghadapi masalah ketidakseimbangan data (imbalance data), di mana ulasan positif lebih banyak daripada ulasan negatif. Untuk mengatasi ketidakseimbangan ini, diterapkan metode SMOTE. Hasil penelitian menunjukkan bahwa penggunaan seleksi fitur Chi-Square, SMOTE, dan SVM memberikan hasil yang optimal dengan akurasi 93,46%, presisi 89,42%, dan recall 89,30% menggunakan k-fold 7 dan seleksi 90% fitur. Sebagai perbandingan, algoritma SVM tanpa penerapan seleksi fitur dan SMOTE hanya memberikan akurasi tertinggi sebesar 90,15%, presisi 90,13%, dan recall 90,15%. Temuan ini menunjukkan bahwa kombinasi pendekatan ini efektif dalam meningkatkan kinerja analisis sentimen ulasan aplikasi Bareksa.

Kata kunci : Analisis Sentimen, TF-IDF, *Lexicon Based*, *Chi Square*, *Support Vector Machine*, *smote*

Daftar Isi

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN ORSINARALITAS	iv
KATA PENGANTAR	v
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR KODE PROGRAM.....	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	4
1.2.1. Permasalahan.....	4
1.2.2. Metode Usulan	4
1.2.3. Pertanyaan Penelitian.....	5
1.3. Tujuan dan Manfaat.....	5
1.3.1. Tujuan Penelitian	5
1.3.2. Manfaat Penelitian	5
1.4. Batasan Masalah.....	6
1.5. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	8
2.1. Bareksa	8
2.2. Analisis Sentimen.....	8
2.3. <i>Text Mining</i>	8
2.4. <i>Web Scraping</i>	9
2.5. <i>Google Play Scraper</i>	9
2.6. Text Preprocessing	10
2.6.1. <i>Handling Duplicate</i>	10
2.6.2. <i>Cleaning & Case Folding</i>	10

2.6.3.	Normalisasi	10
2.6.4.	Tokenization.....	11
2.6.5.	<i>Stopword Removal</i>	11
2.6.6.	Lemmatization.....	11
2.6.7.	<i>Labelling</i>	11
2.7.	Pembobotan TF-IDF.....	12
2.8.	SMOTE.....	13
2.9.	Seleksi fitur Chi-Square	13
2.10.	<i>Support Vector Machine (SVM)</i>	14
2.11.	<i>Confusion Matrix</i>	16
2.12.	WordCloud.....	18
2.13.	Penelitian Terkait.....	18
Bab III METODE USULAN		23
3.1.	Arsitektur Sistem.....	23
3.2.	Web Scraping	24
3.3.	Dataset	25
3.4.	Text Preprocessing	26
3.5.	Labelling.....	28
3.6.	Pembobotan TF-IDF.....	29
3.7.	Pembagian data.....	31
3.8.	Metode Smote.....	31
3.9.	Seleksi fitur <i>Chi-Square</i>	32
3.10.	Proses Klasifikasi SVM	34
3.11.	Evaluasi.....	38
3.12.	Senario Uji Coba.....	38
Bab IV Hasil dan Pembahasan.....		40
4.1.	Pengumpulan data	40
4.2.	<i>Text Preprocessing</i>	42
4.2.1.	<i>Tokenizing</i>	43
4.2.2.	<i>Clean & Case Folding</i>	43
4.2.3.	Normalisasi	44
4.2.4.	<i>Stopword Removal</i>	45

4.2.5.	Lemmatisasi	46
4.2.6.	Labelling	47
4.3.	Pembobotan Kata TF-IDF	48
4.5.	Metode SMOTE	50
4.6.	Seleksi Fitur Chi Square.....	51
4.7.	Klasifikasi SVM	51
4.8.	Evaluasi	52
4.9.	Hasil Skenario Uji Coba 1	53
4.10.	Hasil Skenario Uji Coba 2	54
4.11.	Hasil Skenario Uji Coba 3	58
4.12.	Hasil Skenario Uji Coba 4	59
4.13.	Analisa Skenario Uji coba.	63
4.11.	Implementasi Antarmuka.....	65
Bab V	PENUTUP	71
5.1.	Kesimpulan.....	71
	Daftar Pustaka	73
	LAMPIRAN	77
	DAFTAR RIWAYAT HIDUP.....	78

Daftar Gambar

Gambar 3. 1 Arsitektur Sistem.....	23
Gambar 3. 2 Alur text preprocessing	27
Gambar 3. 3 Labelling.....	28
Gambar 3. 4 TF-IDF	30
Gambar 3. 5 Smote.....	31
Gambar 3. 6 Hasil Selisih antar dokumen.....	32
Gambar 3. 7 Hasil Nilai TF-IDF Sintesis.....	32
Gambar 3. 8 Seleksi Fitur Chi-Square	32
Gambar 4. 1 Hasil Crawling.....	57
Gambar 4. 1 Hasil Crawling.....	42
Gambar 4. 2 Confusion Matrix K=7.....	54
Gambar 4. 3. Confusion Matrix K=9.....	54
Gambar 4. 4 Confusion Matrix 55% K=7.....	57
Gambar 4. 5. Confusion Matrix K=7.....	57
Gambar 4. 6 Grafik performa svm dan chisquare.....	58
Gambar 4. 7 Confusion Matrix.....	59
Gambar 4. 8 Confusion Matrix.....	62
Gambar 4. 10 Confusion Matrix.....	64
Gambar 4. 11 Halaman Home.....	65
Gambar 4. 12 Halaman Tokenizing.....	66
Gambar 4. 13 Clean & CaseFolding.....	66
Gambar 4. 14 Halaman Normalisasi.....	67
Gambar 4. 15 Halaman Stopword Removal.....	67
Gambar 4. 16 Halaman Word Cloud.....	68
Gambar 4. 17 Halaman Feature Extraction.....	68
Gambar 4. 18 Halaman Smote.....	69
Gambar 4. 19 Halaman Performa Kinerja Model.....	69
Gambar 4. 20 Halaman Uji Coba.....	70

Daftar Tabel

Tabel 3. 1 Hasil Scrapping.....	25
Tabel 3. 2 Dataset.....	26
Tabel 3. 3 Hasil Text Preprocessing	27
Tabel 3. 4 Hasil Labelling.....	29
Tabel 3. 5 Hasil Text Preprocessing	30
Tabel 3. 6 Seleksi Fitur Chi Square	33
Tabel 3. 7 Hasil Chi Square terurut.....	34
Tabel 3.8 Hasil Keseluruhan Perhitungan dot product kernel linear	35
Tabel 3. 9. Hasil Perhitungan Matrik	35
Tabel 3. 10. Hasil Perhitungan Epsilon.....	36
Tabel 3. 11. Hasil Perhitungan Delta Alpha	36
Tabel 3. 12. Hasil Perhitungan Alpha Baru	36
Tabel 3. 13. Pengujian data test	37
Tabel 3. 14. Hasil dot data train dan data test	37
Tabel 3. 15 Tabel Skenario Uji coba.....	39
Tabel 4. 1 Library yang dipakai	40
Tabel 4. 2 Hasil proses pelabelan.....	48
Tabel 4. 3 Hasil Skenario Uji Coba 1	53
Tabel 4. 4 Hasil Skenario Uji Coba 2	54
Tabel 4. 5 Hasil Skenario Uji Coba 3	58
Tabel 4. 6 Hasil Skenario Uji Coba 4	59

Daftar Kode Program

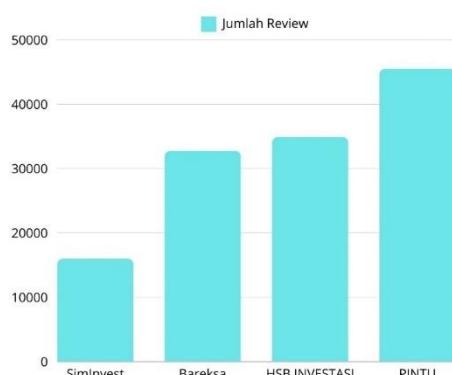
Kode Program 4. 1 Crawling	41
Kode Program 4. 2 Library untuk Preprocessing	43
Kode Program 4. 3 Tokenizing	43
Kode Program 4. 4 script clean & case folding	44
Kode Program 4. 5 Normalisasi	45
Kode Program 4. 6 Stopword Removal	45
Kode Program 4. 7 Lematisasi	46
Kode Program 4. 8 Labelling vader Lexikon	47
Kode Program 4. 9 pembobotan kata TF-IDF	49
Kode Program 4. 10 Kode Spliting Data	49
Kode Program 4. 11 SMOTE	50
Kode Program 4. 12 Seleksi Fitur Chi-Square	51
Kode Program 4. 13 Klasifikasi SVM	52
Kode Program 4. 14 Evaluasi	53

BAB I

PENDAHULUAN

1.1. Latar Belakang

Istilah investasi mengacu pada investasi pada aset atau kekayaan yang dikumpulkan dalam suatu opsi investasi dengan harapan keuntungan di masa depan [1]. Berdasarkan data KSIE per September 2023 berdasarkan single investor identifikasi (SID), jumlah investor pasar modal di Indonesia mencapai 11,72 juta, dengan investor reksa dana naik 14,47%, investor surat berharga negara (SBN) naik 15,45%, dan saham investor naik 13,27%. Rinciannya, investor pasar modal meliputi 10,99 juta investor reksa dana, 5,02 juta investor saham, dan 959 ribu investor surat berharga nasional (SBN). Jika data ini digabungkan dengan SID peserta Tabungan Perumahan Rakyat (TAPERA), maka total SID-nya sebanyak 16 juta. Menurut data KSIE, pertumbuhan pasar modal Indonesia meningkat signifikan, dari 9,2 juta pada tahun 2020 menjadi 28,7 juta pada September 2023, termasuk investasi pada saham, reksadana, dan surat berharga negara (SBN) [2]. Berdasarkan data tersebut, minat masyarakat untuk berinvestasi cukup kuat sehingga banyak bermunculan aplikasi investasi digital. Menurut website CNBC ada beberapa aplikasi investasi terbaik yang telah diawasi oleh OJK yaitu SimInvest, HSB Investasi, Bareksa, Pintu.[1]. Berikut ini merupakan hasil dari visualisasi perbandingan jumlah review di google play.



Gambar 1. 1 Perbandingan Grafik Aplikasi

Dapat dilihat pada gambar 1.1. bahwasanya bareksa memiliki review yang lebih rendah daripada hsb investasi dan pintu tetapi jika dilihat dari rating bareksa unggul sebesar 4.6/5 sedangkan HSB investasi memiliki rating 4.2/5, Pintu memiliki rating 4.1/5. Namun banyaknya aplikasi investasi terbaik membutuhkan informasi yang sangat banyak dan sulit untuk menarik kesimpulan karena banyak pendapat yang berbeda. Untuk memperoleh informasi tersebut, harus dilakukan review data pada aplikasi. Pada pembelajaran *computer science* masalah tersebut bisa diartikan analisis sentimen untuk metode yang bisa digunakan dalam analisis sentimen yakni Naive Bayes Classifier, Support Vector Machine dan Maximum Entropy [2].

Pada paragraf ini terdapat beberapa penelitian yang telah melakukan dengan menggunakan beberapa metode untuk melakukan sebuah pelabelan serta penggunaan metode SVM. Reza Hermansyah *et al* [3] melakukan sebuah penelitian dengan menganalisis produk dan evaluasi pelayanan dalam PT Telekomunikasi Indonesia dengan menggunakan *TextBlob* untuk pelabelan dan *naïve bayers & K-NN*. Penelitian ini menghasilkan akurasi *TextBlob* 54,67%, *naïve bayers* 69,44% dan K-NN sebesar 75% [3]. Sri Lestari *et al*[4], melakukan analisis sentimen terhadap aplikasi saham di *GoolePlay* Store dengan menggunakan algoritma *Support Vector Machine* dengan menggunakan bantuan *Rapid Miner* untuk melakukan pelabelan dengan label positif, negatif dan netral tanpa melakukan ekstraksi fitur. Dalam penelitian ini terdapat 5 aplikasi yang di analisis beserta hasilnya yakni HSB Investasi 88,70%, Ajaib 61,89%, Pluang 68,25%, stockbit 66,95% dan bibit 64,89%. Peneliti menyatakan bahwasanya rapid miner sangat membantu dan mudah untuk digunakan berdasarkan ulasan [4].Pada tahun 2021, Prasoon Gupt *et al* [5], melakukan sebuah penelitian sentiment analysis terhadap kegiatan Lockdown di India selama covid-19 dengan dataset dari twitter dengan menggunakan *TextBlob* dan Vader Lexicon sebagai pelabelan dengan menggunakan ekstraksi fitur *CountVectorizer* atau biasa dikenal sebagai *Bag Of Words*. Penelitian ini menghasilkan akurasi yang cukup baik yakni 84,4% [5]. Penelitian sama yang dilakukan M. N. Muttaqin *et al* [6], dengan melakukan

perbandingan metode analisis sentimen aplikasi gojek dengan menggunakan lexicon untuk melakukan pelabelan serta menggunakan ekstraksi dengan menggunakan algoritma SVM dan K-NN. Penelitian ini menghasilkan 2 akurasi yang cukup baik K-NN 82,14% dengan K=22 dan SVM 87,98% menggunakan parameter c=1. Peneliti menyimpulkan bahwasanya SVM melakukan klasifikasi secara lebih baik di bandingkan K-NN [6]. Dari beberapa penelitian diatas bahwasanya kombinasi metode *Support Vector Machine* dengan penggunaan *vader lexicon* menghasilkan hasil yang cukup baik.

Pada beberapa penelitian terkait dibawah ini memuat tentang penggunaan metode seleksi fitur antaralain yakni Gientry Rachma Ditami *et al* [7], dalam penelitian ini menggunakan TI-IDF sebagai ekstraksi fitur menggunakan algoritma *Support Machina Vector* (SVM) dengan bantuan grid search untuk menentukan parameter terbaik dalam pengujian. Dalam penelitian ini membandingkan event yang dimiliki oleh aplikasi Tokopedia dan Shopee dan menghasilkan akurasi yakni tokopedia 66,23% naik menjadi 67,67%, shopee 66,93% menjadi 67,47 kenaikan tersebut karena menggunakan grid search [7]. Ruba Obiedat *et al* [8], melakukan penelitian menggunakan teknik *crowdsourcing* untuk pelabelan dan menggunakan 2 ekstraksi fitur yakni N-gram dan *Bag Of Words*. Peneliti juga menambahkan SVM PSO dan SMOTE untuk mendapatkan hasil terbaik. Peneliti juga membandingkan beberapa metode yakni SVM *default*, KNN, Naïve Bayes dan 4 lainnya. Penelitian ini menghasilkan akurasi yang paling bagus yakni SVM dengan bantuan PSO dan Smote sebesar 89% [8].M. D. Purbolaksono *et al* [9], membandingkan seleksi fitur yang berbeda yakni gini index dan chi square dengan menggunakan ekstraksi fitur TF-IDF dan menghasilkan akurasi sebesar 85.8% untuk Gini Index dan 89.2% untuk Chi Square [9]. Maka dari itu kombinasi yang terbaik yakni SVM dengan Chi Square karena SVM memiliki kelemahan dalam pemilihan fitur. Pada tahun yang sama, P. R. B. Putra *et al* , melakukan penelitian dengan menggunakan TF-IDF sebagai ekstraksi fiturnya dan memberikan hasil yang terbaik yakni akurasi 93,06%, presisi 92,11%, *recall* 93,06%, dan *fi score* 93,04% [10]. V. Nurcahyawati *et al* [11] Peneliti membandingkan analisis sentimen dengan manual notasi SVM

tanpa bantuan vader lexicon mendapatkan akurasi 86%, presisi 86.06%, recal 95.43% dan *f1 score* 90.51%. Sedangkan hasil dari analisis sentimen dengan menggunakan bantuan vader lexicon meningkat menjadi akurasi 86.57%, presisi 89.71%, recal 96.32%, dan *f1 score* 92.89% [11]. Dari penelitian terkait yang telah dijelaskan bahwasanya penggunaan kombinasi seleksi fitur chi square memiliki hasil yang cukup baik.

Dari hasil penelitian tersebut dapat peneliti simpulkan bahwasanya metode *Support Vector Machine* merupakan metode yang paling cocok digunakan untuk penelitian ini dengan menggunakan *Vader Lexicon Based* dengan ekstraksi fitur TF-IDF serta seleksi fitur menggunakan fitur *Chi Square* dapat menghasilkan performa yang optimal jika dibandingkan dengan metode klasifikasi lainnya dalam pengklasifikasian data teks.

1.2. Rumusan Masalah

1.2.1. Permasalahan

Bareksa merupakan salah satu aplikasi investasi yang terbanyak dipakai oleh masyarakat indonesia dan sudah terverifikasi aman oleh pemerintah. Hal ini tentunya banyak menuai tanggapan oleh masyarakat Indonesia khususnya pengguna aplikasi bareksa. Ada banyak komentar yang didapatkan aplikasi bareksa baik itu positif maupun negatif. Komentar dan tanggapan tersebut bisa menjadi bahan evaluasi untuk bareksa sendiri, tetapi tidak adanya sistem yang bisa menganalisis hal tersebut. Terkait permasalahan ini dibutuhkan suatu sistem sentimen analisis guna memberikan referensi untuk menjadi bahan evaluasi. Pemilihan algoritma *support vector machine* memiliki akurasi yang cukup baik. Akan tetapi, algoritma tersebut memiliki kekurangan dalam pemilihan fitur, oleh karena itu dibutuhkan *chisquare* untuk mendapatkan performa yang optimal.

1.2.2. Metode Usulan

Berdasarkan hal tersebut dalam penelitian ini melakukan analisis sentimen dengan lexicon based sebagai alat untuk pelabelan serta menggunakan SVM dengan seleksi fitur *Chi Square*.

1.2.3. Pertanyaan Penelitian

Berdasarkan permasalahan dan metode usulan dalam penelitian ini didapatkan pertanyaan penelitian, seperti berikut :

- a. Berapa nilai peningkatan performa *accuracy* pada metode *Support Vector Machine* dengan menggunakan seleksi fitur *chi square* ?
- b. Berapa nilai peningkatan performa *accuracy* pada metode *Support Vector Machine* dengan menggunakan metode *smote*?
- c. Berapa nilai peningkatan performa *accuracy* pada metode *Support Vector Machine* dengan menggunakan metode *smote* serta seleksi fitur *chi square*?

1.3. Tujuan dan Manfaat

1.3.1. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui nilai peningkatan performa *accuracy* pada metode *support vector machine* dengan menggunakan seleksi fitur *square*
2. Mengetahui nilai peningkatan performa *accuracy* pada metode *support vector machine* dengan menggunakan metode *smote*
3. Mengetahui nilai peningkatan performa *accuracy* pada metode *support vector machine* dengan menggunakan metode *smote* serta seleksi fitur *chi square*

1.3.2. Manfaat Penelitian

Adapun manfaat dilakukannya penelitian ini adalah :

1. Dapat mengetahui peningkatan performa metode *support vector machine* dengan menggunakan metode *smote* dan seleksi fitur *chi square*
2. Dapat menjadi bahan referensi untuk bahan evaluasi aplikasi bareksa dalam mempertahankan dan meningkatkan pelayanan aplikasi bareksa
3. Dapat menjadi referensi dan sebagai rujukan untuk penelitian serupa.

1.4. Batasan Masalah

Berikut adalah batasan-batasan penelitian yang dilakukan dalam penelitian ini :

1. Data dalam penelitian ini adalah data ulasan aplikasi bareksa yakni PT. Bareksa Marketplace Indonesia dengan jumlah 2008 ulasan yang diambil pada tahun 2023 melalui google play.
2. Pengambilan data dengan menggunakan library *google-play-scapper*
3. Pelabelan dilakukan dengan menggunakan *Vader Lexicon* dengan bantuan kamus lexicon Indonesia.
4. Metode yang digunakan pada penelitian ini yaitu metode *Support Vector Machine* dengan seleksi fitur *chi square* dan tidak dengan algoritma lain.
5. Pengklasifikasian sentimen dijadikan ke dalam 2 Kategori yaitu sentimen positif sebanyak 1240 ulasan dan sentimen negatif sebanyak 977 ulasan.
6. Bahasa Pemograman menggunakan bahasa pemograman *Python*.

1.5. Sistematika Penulisan

Berikut merupakan beberapa aturan dalam penulisan yang digunakan dalam penyusunan laporan. Adaapun sistematika proposal yang digunakan terdiri dari beberapa bagian yaitu :

BAB I PENDAHULUAN

Bab ini membahas latar belakang usulan, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika.

BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai beberapa teori untuk menunjang penelitian, menguraikan tinjauan pustaka, penelitian terdahulu, serta teori penunjang lainnya sebagai bahan acuan.

BAB III METODE USULAN

Pada bab ini akan menguraikan arsitektur sistem yang digunakan dalam penelitian ini, yang melibatkan beberapa proses penting. Dimulai dari pengumpulan data melalui proses scrapping, lalu proses *text preprocessing*, serta menguraikan tahapan pada seleksi fitur menggunakan chi square dan metode smote serta proses klasifikasi menggunakan *support vector machine*.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian yang telah dilakukan dan analisis terhadap hasil penelitian yang didapatkan.

BAB V PENUTUP

Bab ini berisi kesimpulan yang merangkum dan menjawab keseluruhan penelitian serta saran yang memberikan pandangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1. Bareksa

Bareksa merupakan perusahaan PT. Bareksa Marketplace Indonesia yang bergerak di bidang pasar keuangan dan platform investasi terintegrasi pertama di Indonesia yang mendapat izin resmi sebagai penyalur reksa dana dari Otoritas Jasa Keuangan sejak tahun 2016 [12]. Bareksa merupakan forum investasi dengan produk dan fitur lengkap yang aman, sederhana, transparan dan merupakan media edukasi mengenai investasi dan pasar modal. Bagi investor yang masih ragu dalam menentukan keputusan investasinya, Bareksa menawarkan Robo Advisor yang merupakan solusi yang membantu mencari investor dengan strategi yang sesuai [13].

2.2. Analisis Sentimen

Analisis Sentimen adalah proses mengekstraksi sebuah opini public atau ulasan tentang topik, produk, atau layanan tertentu dari teks tidak terstruktur [14]. Analisis sentimen juga bisa diartikan sebuah program yang menganalisis opini atau sentiment seorang pengguna dari sebuah teks, contohnya seperti teks ulasan suatu produk [15]. Tujuan adanya analisis sentimen adalah untuk mengetahui pendapat seseorang atau ulasan pengguna memiliki nilai positif dan negatif yang nantinya nilai tersebut dapat digunakan untuk pengambilan keputusan [16].

Analisis Sentimen juga berfungsi sebagai bahan evaluasi sebuah produk atau jasa karena analisis sentimen bisa mengidentifikasi sebuah keluhan, opini terhadap produk ataupun jasa [16]. Dengan adanya analisis sentimen bisa mempermudah untuk mendapatkan kesimpulan suatu produk atau jasa. Teknik yang umum digunakan untuk menyelesaikan tugas yang melibatkan pengklasifikasi termasuk Naïve Bayes, Support Vector Machines, dan Entropy Maximum (EM).

2.3. *Text Mining*

Text Mining adalah sebuah teknik yang dilakukan untuk mengatasi permasalahan *clustering*, klasifikasi, *information extraction*, dan *information*

retrival. Klasifikasi teks, ekstraksi informasi, dan ekstraksi kata adalah contoh-contoh pra-pemrosesan dokumen yang selalu terlibat dalam text mining. Teknik ini melibatkan pencarian dan pemeriksaan pola-pola yang menarik untuk mengekstrak informasi dari sumber data[17].

Text Mining memiliki tujuan untuk mencari informasi bernilai yang tersembunyi baik dari sumber informasi terstruktur dan tidak terstruktur. *Text Mining* terdiri dari 3 proses yakni *preprocessing*, operasi penggalian teks dan *postprocessing* [17].

2.4. *Web Scrapping*

Web Scrapping merupakan sebuah metode cara pengambilan suatu informasi atau data yang telah ditentukan dengan skala besar yang bertujuan untuk berbagai kepentingan seperti riset, analisis dan lainnya[18]. Tahapan yang dilakukan saat web scrapping yakni [19]:

1. Meminta HTTP dalam mendapatkan sebuah informasi atau data yang ditargetkan.
2. *Request* tersebut akan diproses oleh server dengan format URL.
3. Setelah menerima informasi atau data yang telah sesuai akan diambil

Dalam penelitian ini menggunakan *API Google Play Scraper* untuk memudahkan menemukan data yang telah ditargetkan

2.5. *Google Play Scraper*

Google Play Scraper adalah sebuah alat antarmuka pemograman aplikasi atau biasa disebut dengan API yang memungkinkan pengguna mendapatkan suatu informasi atau data tanpa memerlukan ketergantungan sumber pihak ke-3. *Google Play Scraper* merupakan salah satu metode untuk pengambilan data dari *google play* store yang independent tanpa melibatkan pihak eksternal dengan menggunakan pemograman bahasa python [20].Proses menggunakan *library Google Play Scraper* membutuhkan beberapa komponen yakni nama *package* aplikasi bahasa dan negara. *Google Play Scraper* bisa mendapatkan data ulasan yang banyak hanya saja tergantung banyaknya ulasan aplikasi di *Play Store*.

2.6. Text Preprocessing

Tahapan awal untuk melakukan *text mining* yakni tahap *preprocessing*. Tahap *preprocessing* ini bertujuan untuk mengolah data mentah menjadi data yang bersih dan berkualitas yang siap untuk diolah. Pada tahap ini terdapat beberapa proses dengan menghilangkan tanda baca, merubah huruf uppercase menjadi huruf lowercase, memperbaiki kata yang ada kerusakan, menurunkan volume kata. Beberapa tahapan *text preprocessing* dalam penelitian ini yaitu : *Handling Duplicate, Cleaning, Case Folding, Normalisasi, Stop Removal, Tokenizing, Stemming, Labelling*.

2.6.1. Handling Duplicate

Pada tahap ini berfungsi sebagai untuk menghapus data atau informasi yang duplikat untuk menjaga akurasi dan menghindari statistik yang salah.

2.6.2. Cleaning & Case Folding

Tahapan cleaning berfungsi untuk menghapus semua tanda baca yang terdapat di sebuah ulasan tersebut tanpa terkecuali. Contohnya “Payload validation error. Sampai ulasan ini diunggah, udh 10 hari lebih eror ini muncul” menjadi “Payload validation error Sampai ulasan ini diunggah udh 10 hari lebih eror ini muncul”. *Case Folding* menggunakan operator transform case. Pada tahap ini bertujuan untuk merubah semua karakter menjadi huruf besar atau huruf kecil, untuk penelitian ini menggunakan format huruf kecil dengan menggunakan case folding dapat memudahkan perubahan tersebut. Contohnya “Payload validation error. Sampai ulasan ini diunggah, udh 10 hari lebih eror ini muncul” menjadi “payload validation error sampai ulasan ini diunggah udh 10 hari lebih eror ini muncul”.

2.6.3. Normalisasi

Normalisasi merupakan salah satu dari tahap *preprocessing* yang berfungsi sebagai proses merubah teks menjadi format standar atau normal untuk mempermudah pengolahan data selanjutnya. Normalisasi yang saya pakai dalam penelitian ini merubah kata tidak baku menjadi kata yang baku, memperbaiki kata yang salah dalam penulisan. saya menggunakan library regular expression (re)

untuk melakukan normalisasi dan menambahkan beberapa kata yang salah secara manual. Contohnya “Mw nambahin rekening” menjadi “Mau nambahin rekening”.

2.6.4. Tokenization

Tokenization merupakan sebuah proses pembongkaran kalimat menjadi kata-kata yang lebih sederhana dan lebih bermakna. Tokenization berfungsi untuk memecah kata dari sebuah kalimat menjadi perkata. Dengan menggunakan tahapan ini dapat memudahkan untuk membedakan mana antara pemisah kata atau bukan. Contohnya “secara umum sistem aplikasinya” menjadi “secara, umum, sistem, aplikasinya,“. Tokenizing digunakan untuk pembentukan suatu fitur.

2.6.5. Stopword Removal

Stopword Removal merupakan tahapan yang membuang kata-kata yang tidak bermakna atau tidak memiliki arti penting. Dalam tahap ini saya menggunakan beberapa cara yakni menggunakan kamus stopword Indonesia dan menambahkan kata-kata yang tidak ada pada kamus tersebut dengan manual. Contoh *stopword* yang akan dihapus yakni : itu, dari, yang, dan, ke dan masih banyak lainnya.

2.6.6. Lemmatization

Lemmatization adalah suatu teknik dalam pemrosesan bahasa alami yang digunakan untuk mengubah kata-kata berbentuk dasarnya atau lemma. Dalam proses ini lemmatization mengubah kata-kata berbentuk dasar berdasarkan kamus yang ada. Jadi *lemmatization* berfungsi sebagai mengubah kata berimbuhan ke bentuk dasar yang berfungsi agar lebih mempermudah pemrosesan pengolahan data. contoh dari lemmatization yakni : membayar : bayar, terbaik : baik, tampilan : tampil.

2.6.7. Labelling

Tahap preprocessing terakhir dalam penelitian ini yakni tahap labelling. Pada penelitian ini memiliki 3 kategori untuk pelabelan yakni positif dan negatif. Tahap Labelling atau pelabelan berfungsi untuk menentukan ulasan tersebut memiliki sentiment positif dan negatif. Peneliti menggunakan library Vader Lexicon untuk memberikan label tersebut.

VADER (Valance Aware Dictionary Sentiment Reasoner) Lexicon sangat cocok jika dalam konteks media sosial karena leksikal ini lebih sensitif terhadap

eksperimen sentiment[21]. *Vader* menggunakan perpaduan kombinasi leksikal sentiment dalam daftar leksikal seperti kata-kata *general* diberikan label dengan mengacu pada semantic orientasi positif dan negatif. Dalam *Vader Lexicon* peneliti juga menggunakan kamus lexicon yang tersedia di github.

2.7. Pembobotan TF-IDF

TF-IDF (*Term Frequency Inverse Document Frequency*) adalah suatu metode untuk ekstraksi fitur (pembobotan) dengan memberikan nilai pada masing-masing kata yang ada pada dokumen, Bobot setiap kata atau istilah untuk setiap dokumen ditentukan oleh pTF-IDF. Frasa yang jarang digunakan pada dokumen yang ada memiliki bobot *Inverse Document Frequency* yang relative tinggi ini berbeda dengan *Term Frequency*.

Pada tahap ini frasa dalam bentuk vector dan TF-IDF. Menggunakan pendekatan TF-IDF dalam tahap pembobotan dapat menghasilkan vektor dengan beberapa frasa, sehingga setiap frasa digabungkan sebagai sebuah fitur untuk dilakukan verifikasi. TF-IDF mempunyai sebuah rumus untuk sebagai berikut :

Keterangan :

$tf(i,j)$ = Frekuensi kemunculan *term* dalam dokumen

$$i = 1, 2, 3, \dots N.$$

N = Jumlah.seluruh.dokumen.yang.ada.di.koleksi dokumen.

$df(i)$ = frekuensi.dokumen yang mengandung $term_i$ dari semua.koleksi.dokumen.

2.8. SMOTE

Metode smote singkatan dari *Synthetic Minority Oversampling Technique* adalah sebuah salah satu teknik oversampling atau penambahan data minoritas dengan cara menambahkan sintesis yang bertujuan untuk menyeimbangkan data minoritas dengan data mayoritas. Teknik ini dapat meningkatkan performa dari proses klasifikasi. Hal ini dikarenakan performa proses klasifikasi akan menurun Ketika data tidak seimbang antara data minoritas dan data mayoritas tidak seimbang.

Keterangan :

Xsyn = data sintesis yang akan dibuat

X_i = data yang akan direplikasi.

Xknn = data tetangga terdekat

δ = nilai random antara 0 dan 1

2.9. Seleksi fitur Chi-Square

Seleksi fitur adalah proses menghilangkan fitur-fitur yang mungkin mengganggu proses klasifikasi. Chi Square berguna untuk menentukan hubungan atau pengaruh dua variabel nominal tambahan ($C =$ koefisien kontingensi). Uji Chi

Square menggunakan teori statistik untuk menentukan independensi suatu suku dari kesalahan. Rumus chi-kuadratnya adalah sebagai berikut:

Keterangan :

X² = Chi Square

t = term

c = kelas

A = Banyaknya dokumen yang dimiliki term t pada kelas c

B = Banyaknya dokumen yang dimiliki term t pada kelas selain kelas c

C = Banyaknya dokumen tanpa term t di kelas c

D = Banyaknya dokumen tanpa term t di bukan kelas c

N = Banyaknya dokumen train

2.10. Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah metode prediksi yang dapat diterapkan pada masalah regresi dan klasifikasi. Cara kerja svm yakni mencari *hyperlane* yang terbaik dengan menggunakan margin maksimum yang berfungsi untuk memisahkan dua buah kelas pada ruang input. Dalam *two-dimension*, *hyperline* berupa garis dan dapat berupa *flat plane* dan multi plane[6]. SVM merupakan salah satu algoritma yang sering digunakan untuk melakukan klasifikasi atau mendukung regresi vector. SVM memiliki konsep yang terstruktur dan lebih jelas daripada metode klasifikasi lainnya [22].

Dalam melakukan klasifikasi SVM dengan kernel linear terdapat beberapa langkah Berikut ini proses klasifikasi menggunakan metode *Support Vector Machine* menggunakan kernel linear yaitu:

1. Inisialisasi α , C, epsilon, lamda dan gamma.
 2. masukan data latih berdasarkan kemunculan satu kata kunci dalam kalimat.
 3. Hitung dot product untuk setiap data dengan menggunakan fungsi kernel [K].

Formula fungsi kernel linear adalah sebagai berikut :

4. Hitung matriks dengan formula sebagai berikut :

Keterangan:

D_{ij} = Elemen matriks ke- I,j

Y_i = Kelas data ke-i

Y_j = Kelas data ke- j

5. Hitung nilai eror dengan formula:

Keterangan :

E_i = nilai eror data ke-i

α_j = nilai alfa ke-j

D_{ij} = Matriks Hessian

6. Hitung nilai dari delta alpha dengan formula :

Keterangan:

α_i = alfa nilai ke - i

γ = gamma untuk mencari kecepatan

E_{ij} = rata-rata eror

C = untuk menentukan batas nilai alfa

7. Hitung nilai alpha baru dengan formula

Keterangan.:

α_i = alfa nilai ke - i

$\delta\alpha_i$ = delta alfa nilai ke - i

8. Hitung nilai bias dengan formula

Keterangan:

W_i^+ adalah bobot dot product dengan alpha terbesar di kelas positif.

Wi⁻ adalah bobot dot product dengan alpha terbesar di kelas negatif

9. sesudah nilai α , w dan b diketahui, maka dilanjutkan ketahap pengujian. Untuk melakukan pengujian dilakukan perhitungan dot product antara *data test* dengan semua *data train* dengan fungsi kernel rbf pada formula 2.3. Setelah itu pengujian dilakukan dengan fungsi keputusan:

$$f(x) = w \cdot x + b \text{ atau } f(x) = \sum_{i=1}^m sign(\alpha_i x_i K(x, x_i)) + b \dots \dots \dots (2.10)$$

Keterangan :

α_i = alfa nilai ke-i

x_i = data nilai dari kelas.ke-i

m.= data jumlah dari SV

$K(x, x_i)$ = fungsi kernel yang digunakan

b = nilai bias

2.11. *Confusion Matrix*

Sebuah ketepatan akurasi dalam klasifikasi digunakan sebagai dasar untuk mengevaluasi algoritma klasifikasi dengan menggunakan metode *confusion matrix*. Cara kerja dalam klasifikasi akan dipengaruhi oleh akurasi klasifikasi. Dengan menggunakan perhitungan *confusion matrix* hasil yang diperoleh berupa

nilai akurasi, presisi, *recall*, dan *f-measure*. Berikut representasi dari tabel *confusion matrix*:

Tabel 2. 1 *Confusion Matrix*

Nilai Sesungguhnya	Nilai Prediksi	
	Positive	Negative
Positive	True Positif (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

Keterangan :

TP (True Positive) = Jumlah Dokumen dari kelas 1 yang benar diklasifikasikan sebagai kelas 1.

TP (True Positive) = Jumlah Dokumen dari kelas 0 yang benar diklasifikasikan sebagai kelas 0.

FP (False Positive) = Jumlah Dokumen dari kelas 0 yang benardiklasifikasikan sebagai kelas 1.

FP (False Negative) = Jumlah Dokumen dari kelas 1 yang benardiklasifikasikan sebagai kelas 0.

Confusion Matrix memiliki beberapa pengukuran yang dapat digunakan yang akan menghasilkan *accurasi*, *precision*, *recall*, *specificity*, *f1-score* berikut rumus yang dimaksud :

1. Akurasi

Akurasi adalah suatu tingkat kebenaran antara nilai prediksi dengan nilai actual (nilai kebenaran), dengan rumus :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \dots \dots \dots (2.11)$$

2. Preciccion

merupakan hasil dari perbandingan antara rasio prediksi *true positive* dengan semua hasil yang diprediksi *true positive* ditambah *false positive*. Berikut rumusnya :

3. Recall

Recall merepresentasikan keberhasilan suatu model dalam mengambil informasi. *Recall* merupakan hasil perbandingan dari rasio prediksi *true positive* dengan keseluruhan data yang *true positive*. Berikut umus *Recall* :

4. Specificity

Specificity merupakan perbandingan kebenaran memprediksi *true negative* dengan keseluruhan *true negative* ditambah *false negative*. Berikut adalah rumus *specificity*

$$Specificity = \frac{TP}{TP+FP} \dots \dots \dots (2.14)$$

5. *F1 Score*

F1 score merupakan perbandingan rata-rata presisi dan recall yang telah dibobotkan :

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \dots \dots \dots (2.15)$$

2.12. WordCloud

Wordcloud merupakan salah satu library python yang berfungsi sebagai alat visualisasi berbentuk awan kata. Wordcloud sering digunakan dalam topik analisis sentiment karena dapat mempermudah peneliti dalam mengidentifikasi dan memahami kata-kata kunci yang muncul dari proses analisis sentiment tersebut.

2.13. Penelitian Terkait

Berikut penelitian sebelumnya mengenai SVM dan Chi square yang digunakan penulis sebagai referensi antara lain yaitu Klasifikasi Judul Berita Online menggunakan Metode Support Vector Machine (SVM) dengan Seleksi Fitur Chi-square. Pada penelitian ini data yang digunakan yaitu judul berita online dengan jumlah 2400 judul dengan melakukan beberapa pengujian yakni pengujian kernel svm, pengujian parameter c, pengujian parameter lambda, pengujian konstanta gamma, pengujian parameter epsilon, dan pengujian parameter iterasi, dan

menggunakan pengujian threshold chi square. Hasil dari penelitian ini yakni menghasilkan akurasi yang cukup baik sebesar Ketika menggunakan pengujian threshold chisquare 80% dengan hasil akurasi 93, 06%, presisi 93,11%, recall 93,06%, fi-score 93,04%. Selanjutnya penelitian yang berjudul “Perbandingan Gini Index dan Chi Square pada Sentimen Analisis Ulasan Film menggunakan Support Vector Machine Classifier” pada penelitian ini membandingkan 2 seleksi fitur Gini Index dan Chi Square. Peneliti menggunakan dataset sebesar 5000 data ulasan film didapatkan dari internet movie database (IMD). Penelitian ini menghasilkan akurasi yang cukup baik bagi keduanya yakni Ketika menggunakan gini index menghasilkan akurasi sebesar 85,8%, sedangkan untuk chi-square mengalami peningkatan akurasi sebesar 89,2%. V. Nurcahyawati *et al* [11], melakukan penelitian yang berjudul “Vader Lexicon and Support Vector Machine Algorithm to Detect Customer Sentiment Orientation”. Peneliti membandingkan analisis sentimen dengan manual notasi SVM tanpa bantuan vader lexicon mendapatkan akurasi 86,% presisi 86.06%, recal 95.43% dan *fi score* 90.51%. Sedangkan hasil dari analysis sentimen dengan menggunakan batuan vader lexicon meningkat menjadi akuras 86.57%, presisi 89.71%, recal 96.32%, dan *fi score* 92.89% [11].

Tabel 2. 2 Penelitian Terkait

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
reza hermansyah, riyanarto sarno (2020)	sentiment analysis about product and service evaluation of pt telekomunikasi indonesia tbk from tweets using textblob, naive bayes & k-nn method	textblob, knn dan naïve bayes	penelitian ini menghasilkan akurasi <i>textblob</i> 54,67%, <i>naïve</i> <i>bayers</i> 69,44% dan k-nn sebesar 75%

sri lestari, sudin saepudin*	support vector machine: analisis sentimen aplikasi saham di google play store	svm dan rapid miner	hsb investasi 88,70%, ajaib 61,89%, pluang 68,25%, stockbit 66,95% dan babit 64,89%.
prasoon gupta , sanjay kumar, r. r. suman, and vinay kumar (2021)	sentiment analysis of lockdown in india during covid-19: a case study on twitter	textblob, vader lexicon, n-gram dan <i>countvectorizer</i>	dengan menggunakan textblob dan vader lexicon sebaagai pelabelan, <i>countvectorizer</i> sebagai ekstraksi fitur dan n-gram menghasilkan akurasi yang cukup baik yakni 84,4%
m. nurul muttaqin*, iqbal kharisudi (2021)	analisis sentimen aplikasi gojek menggunakan support vector machine dan k nearest neighbor	svm dan knn	knn menghasilkan akurasi 82.14% dan mengalami kenaikan ketika menggunakan metode svm

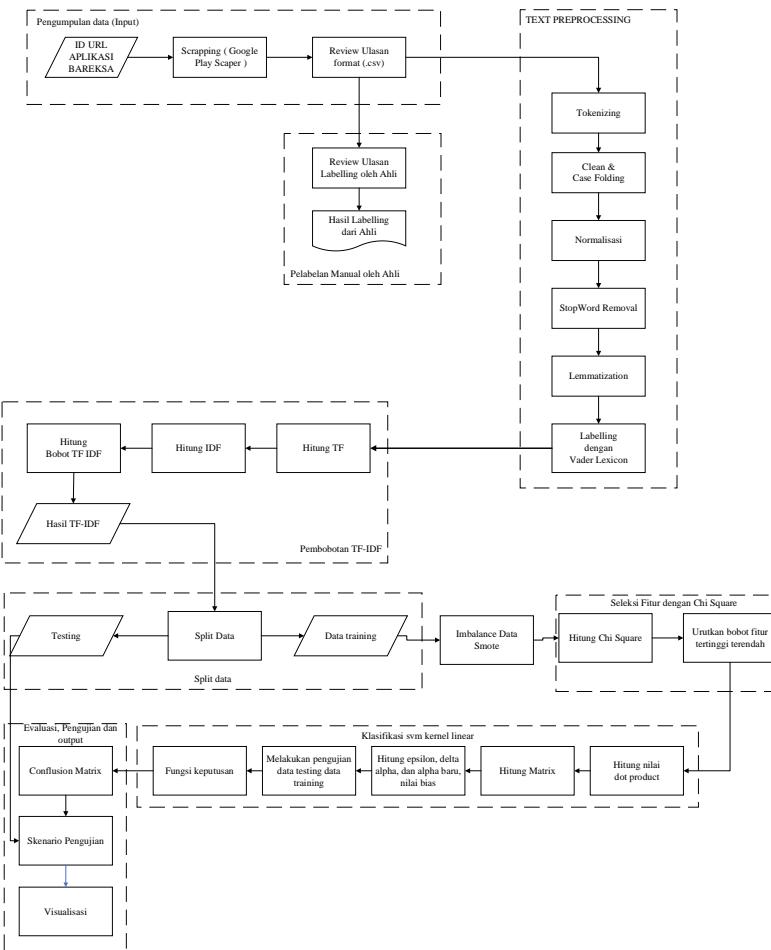
			sebesar 87.98%.
gientry rachma ditami, eva faja ripanti, herry sujaini (2022)	implementasi support vector machine untuk analisis sentimen terhadap pengaruh program promosi event belanja pada marketplace	svm dan grid search	aplikasi tokopedia dan shopee dan menghasilkan akurasi yakni tokopedia 66,23% naik menjadi 67,67%, shopee 66,93% menjadi 67,47 kenaikan tersebut karena menggunakan grid search .
ruba obiedat, raneem qaddoura2, ala' m. al-zoubi1, laila al-qaisi, osama harfoushi, mo'ath alrefai1 , ahossam faris (2022)	sentiment analysis of customers reviews using a hybrid evolutionary svm-based approach in an imbalanced data distribution	svm, pso, smote, bow dan n-gram	penelitian ini menggunakan svm dengan bantuan pso dan smote menghasilkan akurasi sebesar 89%.
mahendra dwifebri purbolaksono, deninsyah tiya	perbandingan gini index dan chi square pada sentimen analisis ulasan film menggunakan	svm, gini index dan chi square	gini index sebesar 86.8% dan chi square sebesar 89.2%

bella pratama, fahmi hamzah (2023)	support vector machine classifie		
selamet riadi, ema utami, ainul yaqin (2023)	comparison of nb and svm in sentiment analysis of cyberbullying using feature selection	svm, chi square	svm tanpa seleksi fitur sakurasi sebesar 82.3%, svm dengan seleksi fitur sebesar 90%.
amalia nur anggraeni, khabib mustofa2 , sigit priyanta3	comparison of filter and wrapper based feature selection methods on spam comment classification	mnb, dengan chi square	dalam penelitian ini menghasilkan 96% untuk mnb dan 100% untuk svm dengan seleksi 500 fitur.
mohammed hadwan 1,2,* , mohammed al- sarem 3,4 , faisal saeed 3,5 and mohammed a. al- hagery 6 (2023)	an improved sentiment classification approach for measuring user satisfaction toward governmental services' mobile apps using machine learning methods with feature engineering and smote technique	svm, lexicon, word2vec, bag of word, tf-idf	dalam penelitian ini menghasilkan akurasi yang tertinggi yakni 94,38%.

Bab III

METODE USULAN

3.1. Arsitektur Sistem



Gambar 3. 1 Arsitektur Sistem

Pada gambar 3.1. merupakan sebuah proses dalam melakukan analisis sentimen pada penelitian ini :

- Pada tahapan pertama yakni melakukan pengumpulan data dengan menyiapkan id url aplikasi bareksa untuk dilakukan proses scrapping yang menghasilkan dokumen yang berformat csv yang saya berinama dataset.
- Setelah melakukan scrapping melakukan secara manual dari para ahli.

- c. Selanjutnya setelah dataset telah didapatkan data tersebut akan dilakukan sebuah proses agar menjadi data yang bersih dan dapat digunakan yakni tahap *text preprocessing* yang terdiri dari beberapa proses *handling duplicate, clean & casefolding*, normalisasi, *tokenizing, stopword removal, stemming* dan *labelling*.
- d. Kemudian setelah melakukan pelabelan dan hasil label tersebut tidak seimbang antara positif dan negatif maka dari itu ditambahkan *smote* untuk menyeimbangkan pelabelan.
- e. Lalu setelah *text preprocessing* tahap selanjutnya yakni ekstraksi fitur menggunakan metode TF-IDF.
- f. Tahapan selanjutnya yakni pembagian data menjadi dua yaitu data *train* dan data *test*.
- g. Selanjutnya dilakukan proses seleksi fitur chisquare.
- h. lalu diproses klasifikasi dengan metode *Support Vector Machine*.
- i. Terakhir proses yang dilakukan yakni pengujian menggunakan *confusion matrix* dan visualisasi sentimen ulasan dengan menggunakan *wordcloud*.

3.2. Web Scraping

Langkah pertama dalam penelitian ini yakni melakukan perolehan data dengan cara *scraping* menggunakan *api* yang bernama *google play scraper*. Dalam penggunaan *api* tersebut hanya memerlukan beberapa komponen yakni membutuhkan url id aplikasi yang terdapat dalam google play yang telah ditentukan. Dalam *google play scraper* bisa mendapatkan ulasan sesuai kebutuhan yang telah ditentukan. Dalam proses web scrarpping ini mendapatkan 1007 ulasan dari aplikasi bareksa yang berbentuk csv. berikut hasil web scraping terdapat pada tabel 3.1.

Tabel 3. 1 Hasil Scrapping

No.	Userrname	Ulasan
1.	Maulina Nia Rahma	Aplikasinya ringan, fiturnya lengkap untuk reksadana cocok untuk pemula yang mau investasi, tampilannya bagus
2.	Rachman Fadhilla	Secara umum sistem aplikasinya Kalo digunakan ya standar aplikasi sejenis dengan antar muka standar. Tp layanan CS nya, ampun buruk sekali. Kl bisa kasih bintang minus, saya kasih bintang minus lima. CS nya : LEMOT, GA NYAMBUNG, GA SOLUTIF, RIBET. Kalau ditanya, jawabannya template, kayak robot. Sekarang aja robot yang pakai AI lebih cerdas cara jawabnya dibandingkan CS BAREKSA. Cma buat buka blokir aplikasi aja harus nunggu sebulan lebih, kl di bank, kyk gini ga sampe sehari selesai.
3.	Maudi sintia	bareksa aplikasi investasi yang bagus, ringan dan jelas informasinya fiturnya lengkap dan mudah dimengerti
4.	Rachmawati Ariningsih	Mw nambahin rekening ditolak terus dengan alasan ktp ga jelas, padahal ktp jelas dan bisa terbaca, jadi uang yg sudah masuk ke bareksa ga bisa diambil dong Kalau gini, ini baru nyoba uang dikit aja gini susahnya, apalagi Kalau sudah masukin uang banyak. Bisa ilang tuh duit
5.	Muhammad Nur	Performa Aplikasi dan tampilan sangat bagus karena ringan, fitur juga bagus terutama fitur robo advisornya cocok pemula seperti saya untuk investast
6.	Sekar Arum	Payload validation error. Sampai ulasan ini diunggah, udh 10 hari lebih eror ini muncul dan kusampaikan ke CS tp belum ada solusi apapun yg kuterima. Error ini kudapatkan ketika mau registrasi SBN, dan pendaftaran akan berakhir tapi belum ada solusi apapun. Selain ke CS, sudah kucoba update aplikasi tapi masih sama ya. Kalau sampe pendaftaran berakhir tp masih belum ada update, sungguh tidak solutif!!! Kalau bisa kasih bintang 0, sudah kukasih itu!

3.3. Dataset

Dataset yang digunakan dalam penelitian ini adalah data ulasan aplikasi bareksa, data tersebut diperoleh dari ulasan pengguna yang telah tertera di *google play store*. Pengambilan data ulasan menggunakan metode *scraping* dengan bantuan *API google play scaper* yang merupakan salah satu *library python* yang dapat melakukan ekstraksi suatu data ulasan aplikasi di *google play store* secara

otomatis. Jumlah data yang diperoleh dari hasil scraping sebanyak 1000 ulasan, dengan jumlah 740 ulasan positif dan 267 ulasan negatif. Berikut tabel 3.2. yang menjelaskan data yang digunakan dalam penelitian ini :

Tabel 3. 2 Dataset

No.	Ulasan
1.	Aplikasinya ringan, fiturnya lengkap untuk reksadana cocok untuk pemula yang mau investasi, tampilannya bagus
2.	Secara umum sistem aplikasinya Kalo digunakan ya standar aplikasi sejenis dengan antar muka standar. Tp layanan CS nya, ampun buruk sekali. Kl bisa kasih bintang minus, saya kasih bintang minus lima. CS nya : LEMOT, GA NYAMBUNG, GA SOLUTIF, RIBET. Kalau ditanya, jawabannya template, kayak robot. Sekarang aja robot yang pakai AI lebih cerdas cara jawabnya dibandingkan CS BAREKSA. Cma buat buka blokir aplikasi aja harus nunggu sebulan lebih, kl di bank, kyk gini ga sampe sehari selesai.
3.	bareksa aplikasi investasi yang bagus, ringan dan jelas informasinya fiturnya lengkap dan mudah dimengerti
4.	Mw nambahin rekening ditolak terus dengan alasan ktp ga jelas, padahal ktp jelas dan bisa terbaca, jadi uang yg sudah masuk ke bareksa ga bisa diambil dong Kalau gini, ini baru nyoba uang dikit aja gini susahnya, apalagi Kalau sudah masukin uang banyak. Bisa ilang tuh duit
5	Performa Aplikasi dan tampilan sangat bagus karena ringan, fitur juga bagus terutama fitur robo advisornya cocok pemula seperti saya untuk investast

3.4. Text Preprocessing

Selanjutnya setelah mendapatkan dataset yang diperlukan yakni tahap *text preprocessing*. Dalam tahap ini data yang telah di scraping diolah menjadi data yang berkualitas dan bersih dan siap diolah puktuk proses selanjutnya. Berikut adalah alur dari tahap *text preprocessing*.



Gambar 3. 2 Alur text preprocessing

Pada gambar 3.2. merupakan alur dari text preprocessing yang terdiri dari beberapa proses seperti *tokenizing* *clean & casefolding*, Normalisasi *stopword removal* dan *lemmatization*. Pada proses *text preprocessing* berfungsi sebagai mengelola data mentah menjadi data yang siap untuk diproses. Pada tabel 3.3 dibawah ini merupakan hasil dari proses *text preprocessing* berikut merupakan hasil text preprocessing

Tabel 3. 3 Hasil Text Preprocessing

Dataset	<i>Tokenizing</i>	<i>Clean & Casefolding</i>	Normalisasi	<i>Stopword Removal</i>	<i>lemmatization</i>
Bareksa adalah aplikasi investasi yang bagus, ringan dan jelas, informasinya lengkap dan mudah dipahami	Bareksa, adalah, aplikasi, investasi, yang, ringan, jelas, informasinya lengkap dan mudah dipahami	bareksa, adalah, aplikasi, investasi, yang, ringan, jelas, informasiny a, lengkap,dan ,	bareksa, adalah, aplikasi, investasi, yang, ringan, jelas, informasiny a, lengkap,dan ,	bareksa aplikasi investasi bagus ringan jelas informasi lengkap nya mudah dipahami	bareksa aplikasi investasi bagus ringan jelas informasi lengkap mudah paham

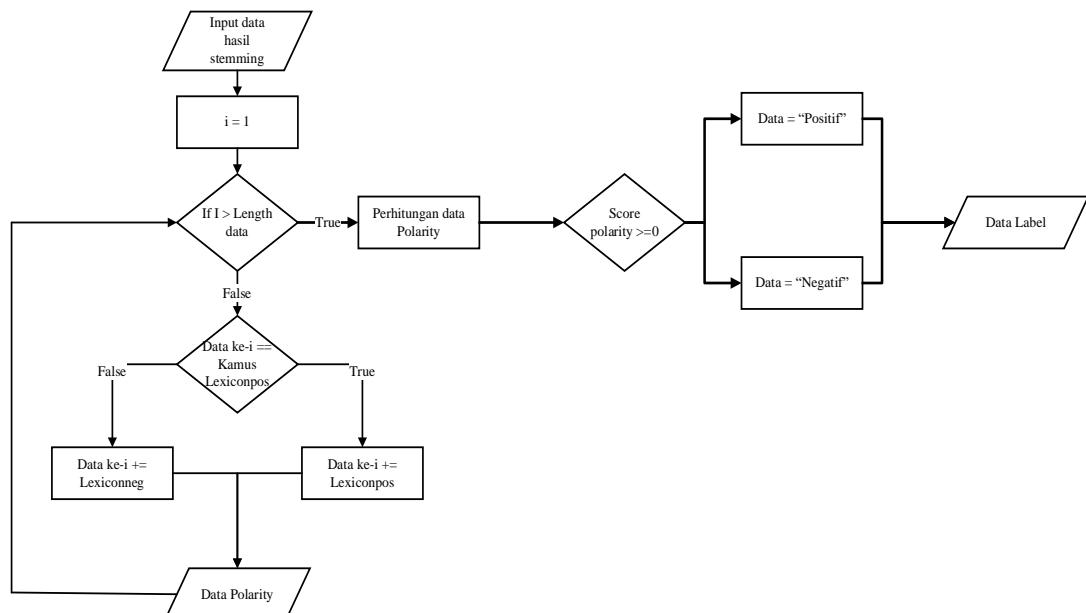
Pada tabel 3.3. menunjukkan hasil *text propressing* untuk penjelasan tahapan *text preprocessing* sebagai berikut :

- a. *Tokenizing* untuk memisahkan kalimat menjadi kata dalam bentuk token.

- b. Selanjutnya *Clean & Case Folding* yang berguna untuk menghapus symbol-simbol dan merubah dokumen ke dalam format tertentu.
- c. Kemudian proses normalisasi yang berguna untuk merubah kata-kata yang tidak benar atau typo.
- d. Selanjutnya proses stopword removal yang berguna untuk menghilangkan kata-kata yang tidak penting.
- e. Setelah itu proses lemmatization yaitu merubah kata kedalam bentuk kata dasar.

3.5. Labelling

Labelling adalah sebuah proses yang melakukan suatu penentuan label sentiment apakah data tersebut positif atau negatif. Peneliti menggunakan bantuan kamus lexicon dan *vader lexicon* sebagai *library python* untuk memberikan label pada dataset yang telah dilakukan proses stemming.



Gambar 3. 3 Labelling

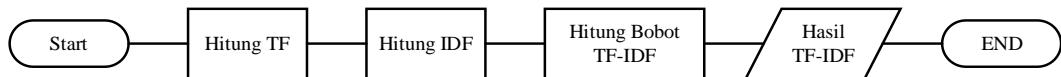
Dalam gambar 3.3. tersebut dapat dilihat bahwasanya data yang telah dilakukan proses *stemming* akan dimasukkan kedalam sebuah kondisi apakah data tersebut ada pada kamus lexicon yang positif atau negatif, lalu setelah itu diproses dan hasil prosesnya yakni dikumpulkan menjadi satu sehingga menghasilkan dataset yang telah memiliki hasil polarity, polarity tersebut diberikan kondisi Ketika

polarity tersebut ≥ 0 maka data tersebut memiliki sentimen positif, sedangkan polaritas <0 maka data tersebut memiliki sentimen negatif. Pada proses ini menghasilkan label sentimen positif sebanyak 740 dan label sentimen negatif sebanyak 267. Berikut hasil dari labelling pada tabel 3.4.

Tabel 3. 4 Hasil Labelling

Ulasan	Compound Score	Sentimen
aplikasi ringan fiturnya lengkap reksadana cocok mula mau investasi tampil bagus	9	Positif
terima kasih bareksa bantu atur uang moga bareksa makin sukses terus untuk bantu lain amin	18	Positif
buka rekening saham tunda terlalu ribet sih gak bibit integrasi stockbit jadi gak usah foto ulang e ktp ketika hasil buram jadi gabisa	-2	Negatif
tampil gajelas kalau mode gelap metode bayar cuma ovo sama danamon jadi ribet	-13	Negatif

3.6. Pembobotan TF-IDF



Gambar 3.4 Pembobotan Fitur TF-IDF

Setelah dilakukan *text preprocessing* data tersebut akan dilakukan ekstraksi fitur menggunakan tf-idf. Dalam tahapan ini, kumpulan kata/term akan diubah menjadi bentuk numerik yang menghasilkan matrik vector. Dataset yang telah dilakukan *text preprocessing* diinputkan kemudian dicari frekuensi kemunculan kata pada setiap dokumen. Selanjutnya mencari nilai IDF dengan rumus yang ada dan dilanjutkan dengan menghitung bobot tf-idf untuk mencari nilai dari kumpulan kata yang dilatih dimana hasilnya dalam bentuk vektor. Berikut adalah dataset yang akan digunakan.

Tabel 3. 5 Hasil Text Preprocessing

Ulasan	Sentimen
aplikasi ringan fitur lengkap reksadana cocok mula mau investasi tampil bagus	Positif
performa aplikasi tampil sangat bagus ringan fitur bagus utama fitur robo advisoranya cocok mula saya investasi	Positif
aplikasi sangat mudah guna fitur lengkap	Positif
fitur aplikasi kurang lengkap saham tampil biasa aplikasi lumayan ringan	Negatif
tampil gajelas kalau mode gelap metode bayar cuma ovo sama danamon jadi ribet	Negatif

Dalam proses tf-idf peneliti menggunakan 5 ulasan yang telah dilakukan *text preprocessing*, dataset tersebut sudah tertera tabel diatas. Setelah dilakukan pembobotan total fitur pada penelitian ini yakni 2567. Ekstraksi fitur dilakukan dengan memberikan bobot pada setiap kata dengan menghitung jumlah frekuensi kemunculan kata pada setiap data, untuk lebih jelasnya dapat dilihat dari gambar 3.4. berikut :

Term	Dokumen					DF	D/DF	IDF	Bobot (W)				
	D1	D2	D3	D4	D5				wd1	wd2	wd3	wd4	wd5
aplikasi	1	1	1	1		4	1,3	0,1	0,1	0,1	0,1	0,1	0
ringan	1	1		1		3	1,7	0,2	0,2	0,2	0	0,2	0
fitur	1	2	1	1		4	1,3	0,1	0,1	0,2	0,1	0,1	0
lengkap	1		1	1		3	1,7	0,2	0,2	0	0,2	0,2	0
cocok	1	1				2	2,5	0,4	0,4	0,4	0	0	0
mula	1	1				2	2,5	0,4	0,4	0,4	0	0	0
mau	1					1	5	0,7	0,7	0	0	0	0
investasi	1	1				2	2,5	0,4	0,4	0,4	0	0	0
tampil	1	1			1	3	1,7	0,2	0,2	0,2	0	0	0,2
bagus	1	1				2	2,5	0,4	0,4	0,4	0	0	0
performa		1				1	5	0,7	0	0,7	0	0	0
sangat		1				1	5	0,7	0	0,7	0	0	0
utama		1				1	5	0,7	0	0,7	0	0	0
robo		1				1	5	0,7	0	0,7	0	0	0
advisoranya		1				1	5	0,7	0	0,7	0	0	0
saya		1				1	5	0,7	0	0,7	0	0	0
mudah		1				1	5	0,7	0	0	0,7	0	0
guna		1				1	5	0,7	0	0	0,7	0	0
saham			1			1	5	0,7	0	0	0	0,7	0
biasa			1			1	5	0,7	0	0	0	0,7	0
lumayan			1			1	5	0,7	0	0	0	0,7	0
gajelas				1		1	5	0,7	0	0	0	0	0,7
kalau				1		1	5	0,7	0	0	0	0	0,7
mode				1		1	5	0,7	0	0	0	0	0,7
gelap				1		1	5	0,7	0	0	0	0	0,7
metode				1		1	5	0,7	0	0	0	0	0,7
bayar				1		1	5	0,7	0	0	0	0	0,7
Cuma				1		1	5	0,7	0	0	0	0	0,7
ovo				1		1	5	0,7	0	0	0	0	0,7
sama				1		1	5	0,7	0	0	0	0	0,7
danamon				1		1	5	0,7	0	0	0	0	0,7
jadi					1	1	5	0,7	0	0	0	0	0,7
ribet					1	1	5	0,7	0	0	0	0	0,7

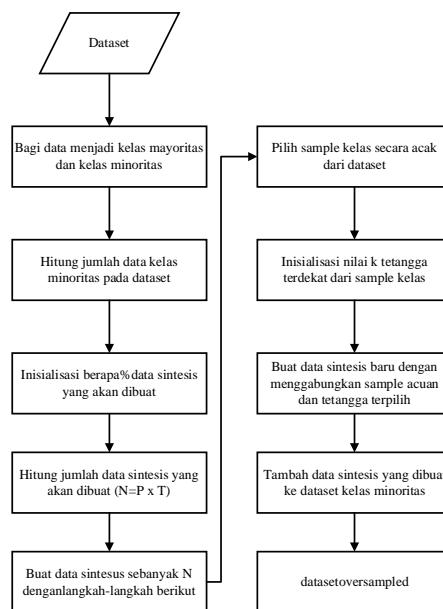
Gambar 3. 4 TF-IDF

3.7. Pembagian data

Pada tahap ini dilakukan pembagian data *training* dan data *testing*. Teknik yang digunakan untuk membagi data disini menggunakan teknik *train test split*. Teknik ini membagi data menjadi dua bagian yaitu data *traing* dan data *test* sesuai presentase yang ditentukan. Pada penelitian ini menggunakan 80%, 20%. Pada penelitian ini menggunakan 80%:20% didasarkan pada penelitian sebelumnya tentang “Sentiment Analysis in Finance: From Transformers Back to eXplainable Lexicons (XLex)” [23].

3.8. Metode Smote

Teknik smote ini digunakan untuk mengatasi ketidak seimbangan data kelas mayritas dan kelas minoritas. pada gambar 3.6. mendeskripsikan langkah-langkah pada smote.



Gambar 3. 5 Smote

Untuk membuat data sintesis dengan menggunakan metode SMOTE, dataset terlebih dahulu dilakukan pembobotan kata. Pembobotan kata yang dilakukan menggunakan TF-IDF seperti yang ditampilkan pada gambar 3.5 Sebagai contoh perhitungan, gunakan data ulasan dokumen 1, 2 dan 3 yang bersentimen positif. Dari ke-3 dokumen tersebut pilih salah satu dokumen sebagai data sample secara acak untuk menghasilkan data sintesis. Misal memilih dokumen pertama

maka data kedua dan ketiga dijadikan sebagai tetangga terdekatnya. Kemudian hitung selisih atau perbedaan antara tetangga terdekat dari data sample dengan data sample. Seperti yang ditunjukkan pada tabel 3.6.

d1	aplikasi	ringan	reksadana	fitur	lengkap	cocok	mula	mau	investasi	tampil	bagus				
d2-d1	0,0000	0,0000	-0,6990	0,0969	-0,2218	0,0000	0,0000	-0,6990	0,0000	0,0000	0,0000	0,6990	0,3979	0,6990	0,6990
d3-d1	0,0000	-0,2218	-0,6990	0,0000	0,0000	-0,3979	-0,3979	-0,6990	-0,3979	-0,0969	0,0000	0,6990	0,6990	0,6990	0,6990

Gambar 3. 6 Hasil Selisih antar dokumen

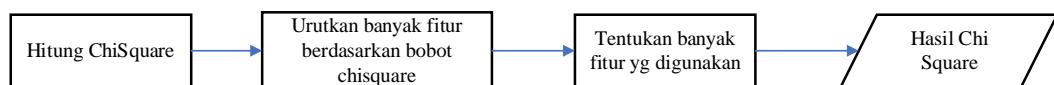
Dapat dilihat dari gambar 3.7. selisih antara dokumen 2 dan dokumen 3 sebagai tetangga terdekat dari dokumen 1 sebagai data sample. Setelah itu buat data sintesis dengan menggunakan persamaan 2.3 sehingga menghasilkan data sintesis dokumen 4 (D4) dengan dokumen 2 sebagai tetangga terdekatnya sehingga vektor fitur yang dihasilkan sebagai berikut :

Term	D1	D2	D3	D4 Sintesis
aplikasi	0,09691	0,09691	0,09691	0,09691
ringan	0,221849	0,221849	0	0,1109245
fitur	0,09691	0,19382	0,09691	0,145365
lengkap	0,221849	0	0,2218487	0,110925
cocok	0,39794	0,39794	0	0,19897
mula	0,39794	0,39794	0	0,19897
mau	0,69897	0	0	0,349485
investasi	0,39794	0,39794	0	0,19897
tampil	0,09691	0,09691	0	0,09691
bagus	0,221849	0,221849	0,2218487	0,22185
performa	0	0,69897	0	0,349485
sangat	0	0,39794	0,39794	0,19897
utama	0	0,69897	0	0,349485
robo	0	0,69897	0	0,349485
advisornya	0	0,69897	0	0,349485
saya	0	0,69897	0	0,349485
mudah	0	0	0,69897	0,349485
guna	0	0	0,69897	0,349485

Gambar 3. 7 Hasil Nilai TF-IDF Sintesis

Hasil data sintesis tersebut ditambahkan kedalam dataset, sehingga menghasilkan dataset oversampling

3.9. Seleksi fitur Chi-Square



Gambar 3. 8 Seleksi Fitur Chi-Square

Pada tahapan yang pertama dalam seleksi fitur chi square menempatkan fitur atau term yang akan dihitung nilai jumlah kemunculan data tersebut menggunakan tabel kontingensi untuk kelas positif dan negatif.

Kemudian hasil nilai chi square fitur dari kedua kelas tersebut akan dibandingkan dan akan dipilih nilai chi square maksimal dari fitur tersebut. Untuk melakukan perhitungan manual peneliti menggunakan dataset yang sama dengan digunakan saat proses tfidf. Untuk contohnya mengambil kata “aplikasi” sebagai berikut :

$$X^2(\text{aplikasi, positif}) = \frac{(5 \times (3 \times 1 - 0 \times 1))^2}{(3+0)(1+1)(3+1)(0+1)} = 1,875$$

$$X^2(\text{aplikasi, negatif}) = \frac{(5 \times (1 \times 0 - 1 \times 3))^2}{(1+1)(3+0)(1+3)(1+0)} = 1,875$$

Berikut merupakan hasil dari perhitungan nilai bobot chi square yang telah dihitung secara manual di tabel 3.6.

Tabel 3. 6 Seleksi Fitur Chi Square

Fitur	Kelas == Positif				Kelas == egatif				Bobot Chi Square		Max W
	A	B	C	D	A	B	C	D	Positif	Negatif	
aplikasi	3	1	0	1	1	3	1	0	1,875	1,875	1,875
ringan	2	1	1	1	1	2	1	1	0,139	0,139	0,139
fitur	3	1	0	1	1	3	1	0	1,875	1,875	1,875
lengkap	2	1	1	1	1	2	1	1	0,139	0,139	0,139
cocok	2	0	1	2	0	2	2	1	2,222	2,222	2,222
...
jadi	0	1	3	1	1	0	1	3	1,875	1,875	1,875
ribet	0	1	3	1	1	0	1	3	1,875	1,875	1,875

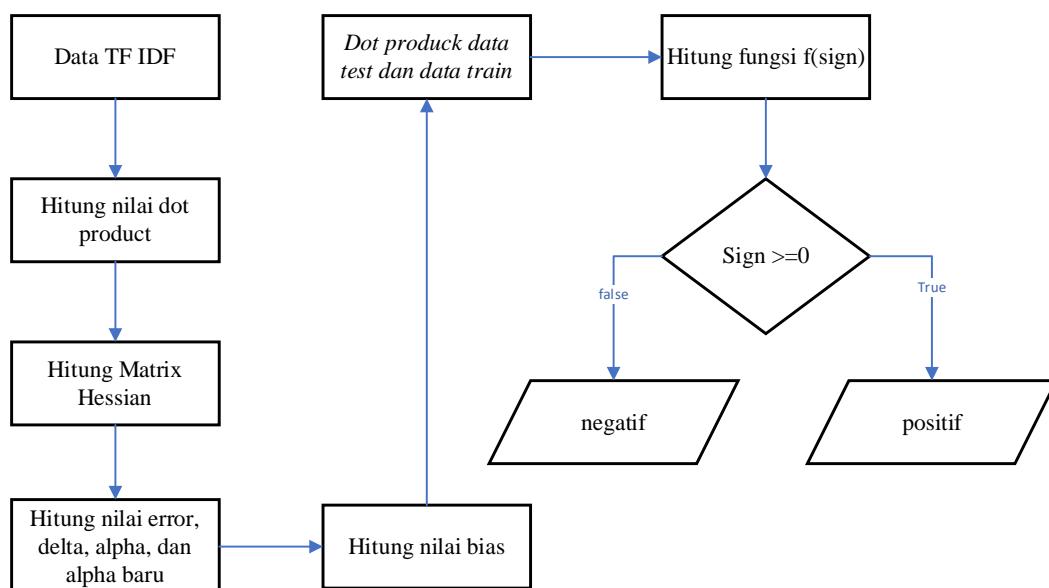
Kemudian semua fitur telah diketahui nilai bobot chi square-nya, maka semua fitur akan di urutkan nilai bobot chi square yang tertinggi ke rendah.

Tabel 3. 7 Hasil Chi Square terurut

Fitur	Max Bobot Chi Square
Bagus	5
Investasi	2,222
Cocok	2,222
...	...
lengkap	0,139

Kemudian setelah mengurutkan fitur yang digunakan untuk proses klasifikasi. Pada penelitian ini menggunakan fitur dengan bobot 965 fitur dengan bobot chi square tertinggi.

3.10. Proses Klasifikasi SVM



Gambar 3. 8 Proses Klasifikasi SVM

Pada proses klasifikasi yang akan digunakan dalam penelitian ini menggunakan model klasifikasi *Support Vector Machine*. Data yang telah diproses perhitungan bobot atau TF-IDF sebelumnya dibagi menjadi dua yaitu data *train* dan data *test*.

1. Langkah pertama menentukan nilai awal dari $\alpha = 0.5$, $C=1$, $\lambda = 0.6$ dan $\gamma = 1$.

2. Selanjutnya menghitung dot product disetiap data dengan menggunakan fungsi kernel. Fungsi kernel yang digunakan yaitu kernel linear. Untuk menghitung nilai *dot product* dilakukan dengan formula 2.3, berikut adalah contoh perhitungan *dot product* :

$$D_{1,1} = (0,09691*0,09691) + (0,22184*0,22184) + (0,09691*0,09691) + (0,22184*0,22184) + (0,39794*0,39794) + (0,39794*0,39794) + (0,69897 *0,69897) + (0,39794*0,39794) + (0,09691*0,09691) + (0,39794*0,39794)$$

Tabel 3.8 Hasil Keseluruhan Perhitungan dot product kernel linear

	D1	D2	D3	D4	D5	B
D1	1,248592	0,720208	0,226356	0,126608	0,009392	1
D2	0,720208	3,670346	0,186531	0,086783	0,009392	1
D3	0,226356	0,186531	1,203474	0,068	0	1
D4	0,126608	0,086783	0,068	1,592286	0,009392	-1
D5	0,009392	0,009392	0	0,009392	5,8721	-1

3. Menghitung Matik Heissan dengan formula 2.4 berikut adalah contoh perhitungannya:

$$D_{1,1} = (1*1)*1,248592 + (0,5*0,5) = 1,498592$$

Tabel 3. 9. Hasil Perhitungan Matrik

	D1	D2	D3	D4	D5
D1	1,498592	0,970208	0,476356	0,123392	0,240608
D2	0,970208	3,920346	0,436531	0,163217	0,240608
D3	0,476356	0,436531	1,453474	0,182	0,25
D4	0,123392	0,163217	0,182	1,842286	0,259392
D5	0,240608	0,240608	0,25	0,259392	6,1221

4. Mencari nilai epsilon dengan menggunakan formula 2.5 berikut contoh perhitungannya :

$$E_1 = (0,5*1,498592) + (0,5*0,970208) + (0,5*0,476356) + (0,5*0,123392) + (0,5*0,240608) = 1,654578$$

Tabel 3. 10. Hasil Perhitungan Epsilon

Epsilon	Hasil
E1	1,654578
E2	2,865455
E3	1,399181
E4	1,285143
E5	3,556354

5. Menghitung nilai delta alpha dengan formula 2.6 berikut contoh perhitungannya :

$$a = \text{MIN}(\text{MAX}(0,5 * (1 - 1,654578); -0,5); 1 - 0,5) = -0,32729$$

Tabel 3. 11. Hasil Perhitungan Delta Alpha

Delta Alpha	Hasil
A1	- 0,32729
A2	-0,5
A3	-0,19959
A4	-0,14257
A5	-0,5

6. menghitung nilai alpha baru dengan formula 2.7 berikut contoh perhitungannya:

$$a1 = 0,5 + (-0,32729) = 0,172711$$

Tabel 3. 12. Hasil Perhitungan Alpha Baru

Alpha Baru	Hasil
A1	0,132426
A2	0
A3	0,303086
A4	0,235289

A5	0
----	---

7. Mencari nilai bias dengan menggunakan formula 2.8 dengan menghitung bobot *dot product* di kelas positif dan negatif yang tertinggi :

$$wx^+ = (0,611069*1*0,132426) + (3,231003*1*0) + (0,235748*1*0,30386) \\ + (0,086783*-1*0,235289) + (0,009392*-1*0) = 0,131954$$

$$wx^- = (0,009392*1*0,132426) + (0,009392*1*0) + (0*1*0,30386) + (0,009392*-1*0,235289) + (5,8721*-1*0) = -0,00097$$

Maka nilai b sebagai berikut :

$$b = -1/2(0,131954 + (-0,00097)) = -0,6549$$

8. Melakukan pengujian data test dimana data test tersebut dilakukan pembobotan. Berikut adalah hasil bobot test:

Tabel 3. 13. Pengujian data test

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11
0,2	0,7	0,4	0	0,4	0,2	0,7	0			

9. Menghitung *dot* produk data *test* dan data *train* menggunakan kernel linear berikut contoh perhitungan :

$$d1 = (0,132426*1(0,221849*1,628012)) + (0*1(0,69897*0,611069)) + (0,303086*1(0,39794*0,117217)) + (0,235289*1(0*0,126608)) + (0*1(0,39794*0,009392)) + (0*1(0,221849*0)) + (0*1(0,69897*0)) = 0,47828$$

Tabel 3. 14. Hasil dot data train dan data test

D1							
0,047828	0,056561	0,006177	0	0,000495	0	0	0

10. Menghitung fungsi keputusan yang sesuai dengan rumus 2,9, berikut perhitungannya :

$$F(x) = \text{Sign}(\text{SUM}(0,047828+0,056561+0,006177+0+0,000495+0+0+0) = 1)$$

$$f(x) = 1 \geq 0 \text{ (Positif)}$$

3.11. Evaluasi

Confusion Matrix merupakan metode perhitungan performa suatu metode klasifikasi dalam memprediksi sentimen positif dan negatif terhadap dataset. Pada *confusion matrix* akan dilakukan tahap evaluasi dengan melakukan perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

3.12. Senario Uji Coba

Skenario uji coba bertujuan untuk mengetahui apakah sistem tersebut sesuai dengan yang diharapkan. Pada penelitian ini dilakukan beberapa skenario uji coba sebagai berikut :

1. Pengujian pertama dilakukan yakni melakukan pengujian nilai kinerja model dalam klasifikasi svm. Pengujian ini dilakukan dengan cara menginputkan data yang telah di *crawling* serta dilakukan *text preprocessing* yang dilanjutkan dengan proses pembobotan kata dengan menggunakan TF-IDF, setelah itu akan di klasifikasikan svm dengan menggunakan kernel linear serta menggunakan parameter $c=1$, $\gamma=0,01$ dan menggunakan k-fold validation dengan menggunakan k-fold 5, 7, 9 tanpa menggunakan chisquare dan metode smote dengan menggunakan 80% data latih, dan 20% data uji.
2. Pengujian kedua dilakukan yakni melakukan pengujian nilai kinerja model dalam penelitian ini dengan menggunakan svm serta seleksi fitur *chi square*, Pengujian ini dilakukan dengan cara menginputkan data yang telah di *crawling* serta dilakukan *text preprocessing* yang dilanjutkan dengan proses pembobotan kata dengan menggunakan TF-IDF. Kemudian dilakukan pengujian seleksi fitur chi square dengan menggunakan *threshold 5%* sampai 90% fitur yang bedasarkan penelitian A. Purnawati *et al.* Lalu setelah itu dilakukan pembagian data menjadi data latih 80% dan data uji 20%. Kemudian dilakukan pengujian model dengan menggunakan svm dengan menggunakan kernel linear serta menggunakan parameter $c=1$, $\gamma=0,01$ dan menggunakan k-fold validation dengan menggunakan k-fold 5, 7, 9.
3. Pengujian ketiga dilakukan yakni melakukan pengujian nilai kinerja model dalam klasifikasi svm dengan menggunakan metode smote. Pengujian ini dilakukan dengan cara menginputkan data yang telah di *crawling* serta

- dilakukan text preprocessing yang dilanjutkan dengan proses pembobotan kata dengan menggunakan TF-IDF, lalu melakukan pembagian data dengan 80% data latih dan 20% data uji, kemudian setelah itu dilakukan metode SMOTE pada data train untuk menyeimbangkan data train pada dataset dengan menggunakan parameter K 5 , 7 dan 9 dengan parameter c=1 dan gamma=0,01.
4. Pengujian keempat dilakukan yakni melakukan pengujian nilai kinerja model dalam penelitian ini dengan menggunakan svm serta seleksi fitur chi square dan menggunakan metode smote, Pengujian ini dilakukan dengan cara menginputkan data yang telah di crawling serta dilakukan text preprocessing yang dilanjutkan dengan proses pembobotan kata dengan menggunakan TF-IDF. Kemudian dilakukan pengujian seleksi fitur chi square dengan menggunakan threshold 5% sampai 90% fitur yang bedasarkan penelitian A. Purnawati et al. Lalu setelah itu dilakukan pembagian data menjadi data latih 80% dan data uji 20% lalu digunakan metode smote pada data train. Kemudian dilakukan pengujian model dengan menggunakan svm dengan menggunakan kernel linear serta menggunakan parameter c=1, gamma=0,01 dan menggunakan k-fold validation dengan menggunakan k-fold 5, 7, 9 tanpa menggunakan.

Tabel 3. 15 Tabel Skenario Uji coba

Skenario	Skenario Uji coba				
	Metode	Kernel	Paramter C	Parameter gamma	K-fold Validation
Skenario 1	SVM	Linear	C=1	Gamma = 0,01	K= 5, 7, 9
Skenario 2	SVM+CHISQUARE				
Skenario 3	SVM+SMOTE				
Skenario 4	SVM+CHISQUARE +SMOTE				

Bab IV

Hasil dan Pembahasan

Pada bab ini diuraikan hasil dari penelitian yang telah dilakukan. Pembuatan penelitian ini menggunakan *google collabs* sebagai editor kode dengan menggunakan bahasa pemrograman *python*. Dalam melakukan crawling hingga proses evaluasi pada penelitian ini menggunakan *library* yang dibutuhkan seperti yang diuraikan pada tabel 4.1.

Tabel 4. 1 Library yang dipakai

No.	Library	Kegunaan
1.	<i>Google play scaper</i>	Digunakan dalam melakukan proses <i>crawling</i> data.
2.	Nltk, Spacy, dan Sastrawi	Digunakan dalam melakukan proses preprocessing
3.	Pandas	Digunakan dalam melakukan analisis data
4.	Matplotlib	Digunakan dalam pembuatan plot atau graph untuk visualisasi
5.	imblearn	Digunakan dalam melakukan proses penyeimbangan dataset
6.	sklearn	Digunakan dalam melakukan proses ekstraksi fitur, seleksi fitur, klasifikasi dan evaluasi

4.1. Pengumpulan data

Pengumpulan data pada penelitian ini didapatkan dengan melakukan teknik crawling data ulasan aplikasi bareksa di *goole play store* dengan menggunakan bahasa python serta menggunakan bantuan google play scaper untuk melakukan pengumpulan data atau biasa disebut dengan *crawling* data. Data ulasan yang didapatkan berupa data mentah sebanyak 1007 ulasan dalam format csv. Untuk implementasi dari proses *crawling* tersebut ditunjukan pada kode 4.1.

```

1  from google_play_scraper import Sort, reviews, reviews_all
2  import pandas as pd
3  import time
4  app_id = 'com.bareksa.app'
5  num_reviews = 12005
6  def collect_reviews(app_id, num_reviews):
7      all_reviews = []
8      batch_size = 200
9      start_idx = 0
10     while len(all_reviews) < num_reviews:
11         try:
12             batch_reviews, _ = reviews(
13                 app_id,
14                 lang='id',
15                 country='id',
16                 sort=Sort.MOST_RELEVANT,
17                 count=min(batch_size, num_reviews - len(all_reviews)),
18                 filter_score_with=None if len(all_reviews) == 0 else 3,
19             )
20             all_reviews.extend(batch_reviews)
21             start_idx += batch_size
22             time.sleep(1)
23         except Exception as e:
24             print(f"Error occurred: {e}")
25             break
26     return all_reviews[:num_reviews]
27 df_busu = pd.DataFrame(np.array(app_reviews), columns=['review'])
28 df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))
29 df_busu
30 data = df_busu
31 df = pd.DataFrame(data)
32 file_path = '/content/drive/MyDrive/Skripsi/Program
33 Skripsi/datasetbaru.csv'
34 df.to_csv(file_path, index=False)

```

Kode Program 4. 1 Crawling

Penjelasan dari program pada kode 4.1. sebagai berikut :

1. Pada baris 1-3 berfungsi sebagai untuk memanggil library yang dibutuhkan yakni google play scraper untuk API *crawling*, pandas untuk menganalisis data serta time untuk memberikan jeda panggilan API tersebut.
2. Pada baris 4-5 berfungsi untuk menentukan ID aplikasi dan jumlah ulasan yang akan dilakukan *crawling*.

3. Pada baris 6-25 berfungsi sebagai proses pengumpulan data ulasan dimana ada beberapa parameter yang telah ditentukan.
4. pada baris 26-28 berfungsi sebagai membuat data frame hasil dari crawling data tersebut
5. pada baris 29-33 berfungsi sebagai menyimpan dataset hasil crawling tersebut menjadi format csv. Hasil ulasan yang didapatkan dari crawling dapat dilihat pada gambar 4.1.

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	eic90ad-1204-4a77-8ded-13e59376633	Maulina Nia Rahma	https://play.googleusercontent.com/a/ALVU...	Aplikasinya ringan, fiturnya lengkap untuk rekomendasi...	2	17	4.0.1	19/11/2023 2:12:44	Terima kasih telah memberikan feedback terhadap aplikasi...	2023-11-23 1:59:26	4.0.1
1	aed81eb7-5004-4a27-8ded-07e593866e33	Rachman Fadhillah	https://play.googleusercontent.com/a/ALVU...	Secara umum sistem aplikasinya kalo digunakan ...	1	24	4.0.1	2023-11-21 6:21:44	Mohon maaf atas ketidaknyamanannya, untuk kend...	2023-11-23 2:59:26	4.0.1
2	9a98caaa-0125-0862-1029-ae19bd93cb76	Maudi sintia	https://play.googleusercontent.com/a/ALVU...	bareknya aplikasi investasi yang bagus, ringan ...	4	5	4.0.1	2023-11-14 6:06:12	Terima kasih telah memberikan feedback terhadap aplikasi...	2023-11-16 3:50:04	4.0.1
3	ec3d76d6-5acf-4463-885a-bb69e135307	Rachmawati Arningisih	https://play.googleusercontent.com/a/ACgBooc...	Mw nambahin rekening dilolak terus dengan alas...	1	12	4.1.2000	2023-11-28 7:06:17	Mohon maaf atas ketidaknyamanannya, untuk kend...	2023-12-01 4:19:58	4.1.2000
4	77ec6c12-1890-5f19-b760-d9a0af0cd6051	Muhammad Nur	https://play.googleusercontent.com/a/ALVU...	Performa Aplikasi dan tampilan sangat bagus ka...	5	0	4.0.1	2023-11-16 8:53:10	Terima kasih telah memberikan feedback terhadap aplikasi...	2023-11-18 3:18:06	4.0.1

Gambar 4. 1 Hasil Crawling

Pada gambar 4.1. menunjukkan beberapa fitur awal setelah *crawling* data, hanya data ulasan dengan atribut content yang menggunakan bahasa Indonesia yang akan digunakan untuk proses ke tahap selanjutnya.

4.2. *Text Preprocessing*

Hasil crawling tersebut dinamakan dataset yang akan dilakukan *preprocessing* guna untuk mengelola teks mentah menjadi lebih terstruktur untuk diproses ketahapan selanjutnya. Tahapan teks *preprocessing* melalui beberapa tahapan yang meliputi *tokenizing*, *clean & case folding*, *normalization*, *stopword removal*, *lemmatization*. *Library* yang dibutuhkan untuk melakukan tahapan *preprocessing* dapat dilihat pada kode 4.2.

```

1 import nltk
2 nltk.download('stopwords')
3 import string
4 import re
5 import spacy
6 from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
7 from Sastrawi.StopWordRemover.StopWordRemoverFactory import
8 StopWordRemoverFactory
9 from nltk.corpus import stopwords

```

Kode Program 4. 2 Library untuk Preprocessing

Pada potongan kode 4.2. terdapat beberapa library yang digunakan dalam *preprocessing* yakni *nltk*, *spacy* dan *sastrawi*. Untuk implementasi tahapan *preprocessing* ialah sebagai berikut:

4.2.1. Tokenizing

Dalam penelitian ini *tokenizing* merupakan tahapan awal untuk melakukan *preprocessing* yang berfungsi untuk memisahkan teks menjadi potongan-potongan token. Kode 4.3. merupakan script meakukan tokenizing terhadap ulasan Bareksa.

```
8. df['text_token'] = df['content'].apply(lambda x: x.split())
```

Kode Program 4. 3 Tokenizing

Penjelasan dari potongan program kode 4.3. sebagai berikut :

1. Pada nomor code 8 membuat colomn baru yang berguna untuk menyimpan hasil tokenisasi yang bernama “text_token” dengan berisikan perintah tokenisasi menggunakan metode split pada string memecah menjadi daftar list berdasarkan spasi pemisah.

4.2.2. Clean & Case Folding

Langkah selanjutnya ialah tahapan *clean & case folding*. Dalam penelitian ini *clean & case folding* digabung menjadi satu. *clean* berfungsi untuk menghilangkan karakter special atau simbol, angka dan tanda baca yang tidak dibutuhkan. *Case folding* berfungsi untuk mengubah seluruh huruf dengan satu format, dalam penelitian ini menggunakan format *lowercase*. Kode 4.4. merupakan script *clean & case folding* pada data ulasan bareksa.

```

9   Import re
10  def clean_text(df, text_field, new_text_field_name):
11      df[new_text_field_name] = df[text_field].apply(lambda x: ' '
12          .join(x)).str.lower()
13      df[new_text_field_name] = df[new_text_field_name].apply(lambda elem:
14          re.sub(r"([A-Za-z0-9_]+)|([^\wA-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", '',
15          "", elem))
16      df[new_text_field_name] = df[new_text_field_name].apply(lambda elem:
17          re.sub(r"\d+", "", elem))
18      df[new_text_field_name] = df[new_text_field_name].apply(lambda elem:
19          separate_punctuation(elem))
20      df[new_text_field_name] = df[new_text_field_name].apply(lambda x:
x.split())
21      return df

```

Kode Program 4. 4 script clean & case folding

Penjelasan pada potongan kode 4.4 sebagai berikut :

1. Pada nomor 9 mengimport library re (regular expression)
2. Pada baris 10-12 fungsi `clean_text` yang berisikan beberapa fungsi yang diperlukan untuk *clean & casefolding* berfungsi sebagai merubah format seluruh huruf menjadi *lowercase*.
3. pada baris 13-15 berfungsi sebagai menghapus tanda baca, mention, emoji dan karakter khusus lainnya.
4. pada baris 16-17 berfungsi sebagai menghapus angka.
5. pada baris 17-18 berfungsi sebagai memisahkan tanda baca yang terhubung dengan kata.
6. lalu baris 20 yakni melakukan tokenisasi kembali.

4.2.3. Normalisasi

Tahap selanjutnya ialah tahap normalisasi yang berfungsi untuk memperbaiki kata typo, kata singkatan berdasarkan kamus yang telah dibuat serta dibantu dengan kamus normalisasi dengan Kode 4.5. merupakan *script* untuk melakukan normalisasi pada data ulasan bareksa.

```

21  data_kamus = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
22  Skripsi/Kamus Lexicon/kamus_normalization.csv')
23  def normalisasi(tekst, kamus):
24      kalimat_final = []
25      for kata in tekst:
26          kata_benar = kamus[kamus['Tidak Baku'] == kata]['Baku'].values
27          if len(kata_benar) > 0:

```

```

28         kalimat_final.append(kata_benar[0])
29     else:
30         kalimat_final.append(kata)
31     return kalimat_final
32 df['content_norm'] = df['text_clean'].apply(lambda x:
33     normalisasi(x,data_kamus))
34 df

```

Kode Program 4. 5 Normalisasi

Penjelasan pada potongan kode 4.5. sebagai berikut :

1. Pada 21-22 berfungsi sebagai Membaca kamus bahasa dari file CSV.
2. Pada baris 23-31 berfungsi sebagai fungsi normalisasi yang berguna untuk mengubah kata typo atau salah penulisan maka akan terdeteksi dan diubah.
3. pada baris 32-33 berfungsi sebagai menambahkan dan melakukan normalisasi text tersebut dan menampilkan hasilnya.

4.2.4. Stopword Removal

Tahap selanjutnya ialah tahap *stopword removal* yang berfungsi sebagai penghapusan atau menghilangkan kata yang dianggap tidak penting dan tidak memiliki pengaruh pada token seperti kata penghubung sesuai dengan corpus bahasa Indonesia yang disediakan serta menambahkan beberapa stopword

```

34 more_stopwords = {
35     'dar', 'hai', 'txfffzhybv', 'bg', 'bot', 'yg', 'deh', 'ypdhl', 'tidak',
36     'nic', 'bos', 'hmmm', 'ky', 'yaa', 'mo', 'fb', 'laah', 'br', 'blg', 'da',
37     'x', 'jt', 'dan', 'y', 'b', 't', 'yang', 'sj', 'faq', 'jsajan', 'aja',
38     'mis', 'mf', 'hmm', 'ji', 'issi', 'the', 'kok', 'ng', 'di', 'nih', 'lah',
39     'adm', 'nig', 'min', 'y', 'kak', 'k', 'va', 'dong', 'ai', 'nya', 'e', 'tuh',
40     'nih', 'di', 'min', 'ke', 'dgn', 'nya', 'jadi', 'ada', 'nya', 'ah',
41     'aamiin'}
42 stop_words_factory = StopWordRemoverFactory()
43 stop_words = stop_words_factory.get_stop_words()
44 stop_words = stop_words.extend(more_stopwords)
45 stopword_remover = stop_words_factory.create_stop_word_remover()
46 def remove_stopwords(text):
47     if isinstance(text, list):
48         text = ' '.join(text)
49     return stopword_remover.remove(text)
50 #proses stopword
51 df['text_stopword'] = df['content_norm'].apply(remove_stopwords)
52 df

```

Kode Program 4. 6 Stopword Removal

Penjelasan pada potongan kode 4.6. sebagai berikut :

1. Pada baris 34-40 merupakan Daftar kata-kata stop words tambahan
2. Pada baris 41-44 berfungsi sebagai Membuat daftar kata-kata stop words
3. Pada baris 45-50 sebagai fungsi stopwrod Fungsi untuk menghapus stop words dari teks

4.2.5. Lemmatisasi

Tahap selanjutnya ialah tahap *stopword removal* yang berfungsi sebagai penghapusan atau menghilangkan kata yang dianggap tidak penting dan tidak memiliki pengaruh pada token seperti kata penghubung sesuai dengan corpus bahasa Indonesia yang disediakan serta menambahkan beberapa stopword

```
51 from spacy.tokens import Token
52 from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
53 factory = StemmerFactory()
54 stemmer = factory.create_stemmer()
55 def lemmatize_indonesian(token):
56     return stemmer.stem(token.text)
57 Token.set_extension('lemma_indonesian', getter=lemmatize_indonesian,
58 force=True)
59 nlp = spacy.blank('id')
60 # Tambahkan pipeline tokenizer spaCy
61 def custom_tokenizer(nlp):
62     return spacy.tokenizer.Tokenizer(nlp.vocab)
63 nlp.tokenizer = custom_tokenizer(nlp)
64 data = df['text_stopword']
65 df = pd.DataFrame(data)
66 df['hasil_lemma'] = df['text_stopword'].apply(lambda x: '
67 '.join([token._.lemma_indonesian for token in nlp(x)]))
68 # Tampilkan dataframe hasil
Df
```

Kode Program 4. 7 Lematisasi

Penjelasan pada potongan kode 4.7. sebagai berikut :

1. Pada baris 50-54 Instalisasi dan inisialisasi library spacy dan sastrawi untuk lemmatisasi
2. Pada baris 55-56 fungsi untuk lemmatisasi
3. pada baris ke 57 melakukan penambahan ekstensi spacy dan mengatur model bahaa Indonesia dan tambah pipline spacy.

4.2.6. Labelling

Tahap Selanjutnya yakni tahapan labelling yang berfungsi sebagai menetukan sentiment ulasan tersebut kedalam sentiment positif atau negatif. Dalam penelitian ini menggunakan pelabelan dengan menggunakan vader lexicon sertaakan divalidasi oleh ahli.Berikut code untuk pelabelan dengan menggunakan vader lexicon.

```
69 pip install vaderSentiment
70 import pandas as pd
71 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
72 lexiconpos = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
73 Skripsi/Kamus Lexicon/positifbersih.csv')
74 lexiconneg = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
75 Skripsi/Kamus Lexicon/negatifbersih.csv')
76 analyzer = SentimentIntensityAnalyzer()
77 def label_sentiment_vader(text, lexiconpos, lexiconneg):
78     compound_score = analyzer.polarity_scores(text) ['compound']
79     words = text.split()
80     score = 0
81     for word in words:
82         if word in lexiconpos['word'].values:
83             score += lexiconpos[lexiconpos['word'] ==
84 word]['weight'].values[0]
85         elif word in lexiconneg['word'].values:
86             score += lexiconneg[lexiconneg['word'] ==
87 word]['weight'].values[0]
88     polarity = 'Positif' if score >= 0 else 'Negatif'
89     return score, polarity
90 labels = df['text_prepro'].apply(label_sentiment_vader,
91 lexiconpos=lexiconpos, lexiconneg=lexiconneg)
92 jumlah_positif = df[df['Sentimen'] == 'Positif'].shape[0]
jumlah_negatif = df[df['Sentimen'] == 'Negatif'].shape[0]
jumlah_total = df.shape[0]
```

Kode Program 4. 8 Labelling vader Lexikon

Penjelasan pada potongan kode 4.8. sebagai berikut :

1. Pada baris 69-71 berfungsi sebagai instalasi library vader lexicon.
2. Pada baris 72-75 berfungsi untuk membaca kamus lexicon positif dan negatif.
3. Pada baris 76-88 berfungsi sebagai Fungsi untuk melakukan pelabelan sentimen menggunakan VADER dengan mengecek apakah kata tersebut ada di kamus positif atau negatif lalu dihitung compound score untuk menentukan positif dan negatifnya ulasan tersebut.

4. baris 90-92 berfungsi sebagai untuk mengetahui berapa ulasan positif dan negatif serta total ulasan yang telah dilakukan pelabelan.

Dalam proses ini menghasilkan sentiment positif sebanyak 1087 dan negatif sebanyak 920. Berikut hasil dari proses pelabelan menggunakan.

Tabel 4. 2 Hasil proses pelabelan

Ulasan	Compound Score	Sentimen
aplikasi ringan fiturnya lengkap reksadana cocok mula mau investasi tampil bagus	9	Positif
umum sistem aplikasi kalau guna standard aplikasi jenis antar muka standard layan customer service nya ampun buruk sekali kl kasih bintang minus kasih bintang minus lima customer service nya lot nyambung solutif ribet kalau tanya jawab template kayak robot sekarang aja robot pakai ai lebih cerdas cara jawab banding customer service bareksa cuma buat buka blokir aplikasi aja nunggu bulan lebih kl bank kayak gin sampai hari selesai	-32	Negatif
bareksa aplikasi investasi bagus ringan jelas informasi fiturnya lengkap mudah erti	11	Positif

4.3. Pembobotan Kata TF-IDF

Tahapan selanjutnya yakni pembobotan kata dengan metode TF-IDF. Dalam proses ini setiap kata atau term dihitung jumlah kemunculannya dalam tiap dokumen dan nilai tersebut dijadikan bobot sebagai kata. Kode 4.9. merupakan *script* untuk melakukan tfidf.

```

93 import pandas as pd
94 from sklearn.feature_extraction.text import TfidfVectorizer,
95 CountVectorizer
96 from sklearn.preprocessing import normalize
97 data = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
98 Skripsi/Hasil/labelling2.csv')

```

```

99 column = "text_prepro"
100 data[column] = data[column].fillna('')
101 count_vectorizer = CountVectorizer()
102 tfidf_vectorizer = TfidfVectorizer()
103 TF_vector = count_vectorizer.fit_transform(data[column])
104 normalized_tf_vector = normalize(TF_vector, norm='l1', axis=1)
105 tfs = tfidf_vectorizer.fit_transform(data[column])
106 IDF_vector = tfidf_vectorizer.idf_
    tfidf_mat = normalized_tf_vector.multiply(IDF_vector).toarray()

```

Kode Program 4. 9 pembobotan kata TF-IDF

Penjelasan dari potongan program pada kode 4.9. sebagai berikut :

1. Pada baris 93-95 berfungsi sebagai perintah memanggil *library* yang dibutuhkan.
2. Pada baris 96-99 berfungsi sebagai membaca file csv dan inisialisasi nama kolom yang akan di lakukan tf-idf dan Mengganti nilai NaN dengan string kosong.
3. Pada baris 100-105 berfungsi sebagai Membuat objek CountVectorizer dan TfidfVectorizer. Transformasi teks dengan CountVectorize.
4. Pada baris 106 berfungsi sebagai Mengalikan matriks TF yang sudah dinormalisasi dengan IDF

4.4. *Split Data*

Tahapan selanjutnya merupakan pembagian data dengan menggunakan *library split data* dalam python. Pada tahap ini pembagian data menjadi 2 yakni data training dan data testing dimana pembagiannya 20% untuk data testing dan 80% data training. berikut merupakan code untuk *splitting data*.

```

107 X_train, X_test, y_train, y_test = train_test_split(X_selected, y,
    test_size=0.2, shuffle=True)

```

Kode Program 4. 10 Kode Spliting Data

Penjelasan Potongan kode 4.10 yakni pada baris 107 membuat variable untuk X_train, X_test, y_train, y_test setelah itu akan di bagi untuk data train dan data tesnya menggunakan split data untuk pembagiannya yakni pada code test_size=0.2 berarti untuk data *testing* 20% serta data *training* sebanyak 80%.

4.5. Metode SMOTE

Smote merupakan tahapan untuk mengatasi masalah ketidakseimbangan kelas. Pada tahapan ini dilakukan pembuatan data sintesis dengan menggunakan data *sample* kelas minoritas dan tetangga terdekatnya dan nilai random dari 0 dan 1. Kode 4.10 merupakan script untuk melakukan smote.

```
107 from sklearn.model_selection import train_test_split
108 from imblearn.over_sampling import SMOTE
109 data = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
110 Skripsi/Hasil/labellingfix.csv')
111 column = "text_prep"
112 y = data['Sentimen']
113 tf_idf = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
114 Skripsi/Hasil/tfidffix.csv')
115 X_train, X_test, y_train, y_test = train_test_split(tf_idf, y, test_size=0.2,
116 shuffle=True, random_state=42)
117 smote = SMOTE(random_state=42)
118 X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
119 # Menyimpan hasil resampling untuk digunakan nanti
120 pd.DataFrame(X_train_resampled).to_csv('/content/drive/MyDrive/Skripsi/Program
121 Skripsi/Hasil/X_train_resampled.csv', index=False)
122 pd.Series(y_train_resampled).to_csv('/content/drive/MyDrive/Skripsi/Program
123 Skripsi/Hasil/y_train_resampled.csv', index=False)
124 # Menampilkan hasil SMOTE
125 print("Sebelum SMOTE:")
126 print(y_train.value_counts())
127 print("\nSetelah SMOTE:")
128 print(pd.Series(y_train_resampled).value_counts())
```

Kode Program 4. 11 SMOTE

penjelasan dari potongan program pada kode 4.11 sebagai berikut:

1. Pada baris 107-114 pemanggilan library yang dibutuhkan serta membaca dokumen dataset.
2. pada baris 115-116 melakukan pembagian data dengan split data dengan 80% data latih dan 20% data uji.
3. Pada baris 117-118 proses smote dengan melatih dengan smote
4. pada baris 119-123 proses menyimpan hasil dari sampling dengan smote.

4.6. Seleksi Fitur Chi Square

Dalam tahap selanjutnya yakni pemilihan fitur dengan seleksi fitur chi square yang digunakan untuk menyeleksi fitur berdasarkan nilai chisquare yang dihasilkan pada setiap fitur. Kode 4.12 merupakan contoh *script* untuk melakukan Chi Square.

```
131 import pandas as pd
132 import numpy as np
133 from sklearn.feature_selection import SelectKBest, f_classif
134 import seaborn as sns
135 import matplotlib.pyplot as plt
136 persen_fitur_list = range(5, 95, 5)
137 for persen_fitur in persen_fitur_list:
138     num_feature_to_select = int(persen_fitur * tf_idf.shape[1] / 100)
139     selector = SelectKBest(f_classif, k=num_feature_to_select)
140     X_selected = selector.fit_transform(tf_idf, y)
```

Kode Program 4. 12 Seleksi Fitur Chi-Square

Penjelasan dari potongan program pada kode 4.12. sebagai berikut :

1. Pada baris 131-135 berfungsi sebagai perintah memanggil *library* yang dibutuhkan.
2. Pada baris 136 Daftar persentase fitur yang akan dievaluasi
3. Pada baris 137-140 memiliki fungsi yakni Menentukan jumlah fitur yang akan dipilih, lalu Seleksi fitur menggunakan SelectKBest untuk memilih seleksi fitur yang terbaik.

4.7. Klasifikasi SVM

Dalam tahap selanjutnya yakni pemilihan fitur dengan seleksi fitur chi square yang digunakan untuk menyeleksi fitur berdasarkan nilai chisquare yang dihasilkan pada setiap fitur. Kode 4.13 merupakan contoh *script* untuk melakukan klasifikasi SVM.

```
141 from sklearn.svm import SVC
142 from sklearn.model_selection import train_test_split, cross_val_score,
143 cross_val_predict
144 from sklearn.metrics import confusion_matrix, classification_report,
145 accuracy_score, precision_score, recall_score, f1_score
146 for persen_fitur in persen_fitur_list:
147     num_feature_to_select = int(persen_fitur * tf_idf.shape[1] / 100)
148     selector = SelectKBest(f_classif, k=num_feature_to_select)
```

```

149     X_selected = selector.fit_transform(tf_idf, y)
150     X_train, X_test, y_train, y_test = train_test_split(X_selected, y,
151     test_size=0.2, shuffle=True)
152     modelsvm = SVC(kernel='linear', gamma=0.01, C=1)

```

Kode Program 4. 13 Klasifikasi SVM

Penjelasan dari potongan program pada kode 4.13. sebagai berikut :

1. Pada baris 141-145 berfungsi sebagai perintah memanggil *library* yang dibutuhkan.
2. Pada baris 146-149 berfungsi sebagai seleksi fitur chi square.
3. Pada baris 150-152 berufngsi sebagai Membagi data menjadi data latih dan data uji serta membuat model svm dengan menggunakan kernel linear dengan parameter gamma = 0,01 dan c =1.

4.8. Evaluasi

Pada tahap ini dilakukan evaluasi terhadap kinerja model dengan melakukan perhitungan seberapa akurat model memprediksi sentimen guna menghitung akurasi yang didapatkan model dengan menggunakan metode confusion matrix. Kode 4.14 merupakan script untuk melakukan evaluasi model menggunakan confusion matrix

```

153 from sklearn.metrics import
154 classification_report,confusion_matrix,ConfusionMatrixDisplay, accuracy_score,
155 precision_score, recall_score, f1_score
156 import matplotlib.pyplot as plt
157 conf_matrix = confusion_matrix(y_train, y_train_pred)
158 class_label = ["negative", "positive"]
159 df_cm = pd.DataFrame(conf_matrix, index=class_label, columns=class_label)
160 accuracy = accuracy_score(y_train, y_train_pred)
161 precision = precision_score(y_train, y_train_pred, average='weighted')
162 recall = recall_score(y_train, y_train_pred, average='weighted')
163 f1 = f1_score(y_train, y_train_pred, average='weighted')
164 print(f"Accuracy: {accuracy}")
165 print(f"Precision: {precision}")
166 print(f"Recall: {recall}")
167 print(f"F1 Score: {f1}")
168 plt.figure(figsize=(10, 7))
169 sns.heatmap(df_cm, annot=True, fmt='d')
170 plt.title(f"Confusion Matrix - Cross-Validation (k={k})")
171 plt.xlabel("Predicted Label")
172 plt.ylabel("True Label")
173

```

```
plt.show()
```

Kode Program 4. 14 Evaluasi

Berikut penjelasan dari potongan code 4.14 sebagai berikut :

1. Baris 153-156 merupakan pemanggilan *library* yang akan digunakan.
2. baris 157 merupakan sebagai fungsi untuk melakukan confusion matrix
3. baris 158-159 untuk menyimpan hasil confusion matrix dengan dataframe
4. baris 164-168 untuk menampilkan hasil *confusion matrix*
5. Baris 169-173 yakni menampilkan visualisasi pada *confusion matrix*.

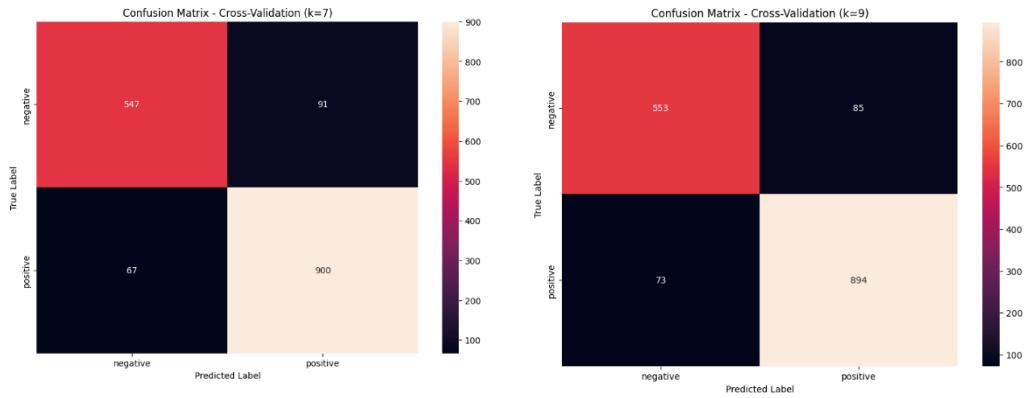
4.9. Hasil Skenario Uji Coba 1

Skenario uji coba pertama ialah pengujian nilai akurasi nilai model *support vector machine* (SVM) tanpa menggunakan seleksi fitur chi square dan metode smote. Pada skenario uji cob ini melakukan dengan menggunakan parameter $c=1$, $\gamma=0,01$ serta menggunakan k-fold validation dengan menggunakan nilai $k = 5, 7, 9$. Berikut hasil skenario uji coba 1 pada tabel 4.3. :

Tabel 4. 3 Hasil Skenario Uji Coba 1

Skenario Uji coba 1	Kernel	Parameterr C	Parameterr γ	K-fold	Akurasi
Support Vector Machine tanpa seleksi fitur, tanpa smote	Linear	1	0,01	5	89,28%
				7	90,15%
				9	90,15%

Pada tabel diatas merupakan hasil dari skenario uji coba yang pertama dapat dilihat seiring ditambah nilai k maka terdapat beberapa peningkatan pada nilai model yang menggunakan svm tanpa menggunakan seleksi fitur dan metode smote. Kemudian untuk hasil evaluasi semua skenario pertama pada gambar 4.2. serta gambar 4.3. dengan menggunakan confusion matrix :



Gambar 4. 2 Confusion Matrix K=7

Gambar 4. 3. Confusion Matrix K=9

Dapat dilihat dari gambar 4.2 dan Gambar 4.3. merupakan evaluasi dari hasil yang optimal dengan menggunakan *support vector machine* dengan menggunakan cross-validation dengan K=7 dengan menghasilkan *True* positif sebanyak 900, *True* negatif 547, *False* positif 91 dan *False* negatif 67 dengan menghasilkan akurasi sebesar 90,15, dan k=9 dengan menghasilkan *True* positif sebanyak 894, *True* negatif 553, *False* positif 85 dan *False* negatif 73 dengan menghasilkan akurasi sebesar 90,15%.

4.10. Hasil Skenario Uji Coba 2

Skenario uji coba 2 ialah pengujian nilai akurasi nilai model *support vector machine* (SVM) menggunakan seleksi fitur chisquare. Pada skenario uji cob ini melakukan dengan menggunakan parameter *c*=1, *gamma*=0,01 serta menggunakan k-fold validation dengan menggunakan nilai k = 5, 7, 9, serta menggunakan seleksi fitur dengan *threshold* 5% sampai dengan 90%. Berikut tabel 4.4. sebagai hasil dari skenario uji coba 2 :

Tabel 4. 4 Hasil Skenario Uji Coba 2

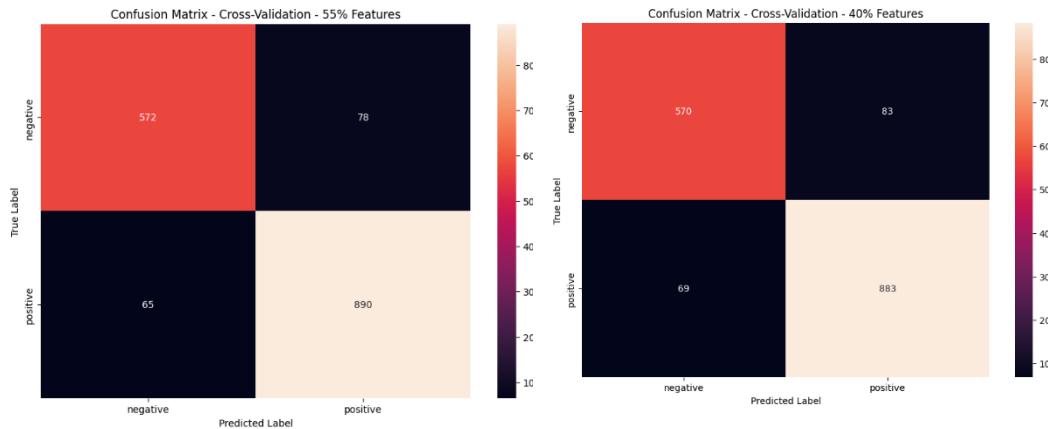
No.	Fitur & SVM	Nilai K-Fold	Akurasi
1.	5%	5	84,17%
		7	83,61%
		9	83,92%

		5	86,29%
2.	10%	7	85,17%
		9	85,66%
3.	15%	5	87,35%
		7	87,72%
		9	87,04
4.	20%	5	88,09%
		7	88,22%
		9	87,91%
5.	25%	5	88,66%
		7	89,96%
		9	90,21%
6.	30%	5	89,40%
		7	90,46%
		9	90,52%
7.	35%	5	89,53%
		7	89,28%
		9	89,03%
8.	40%	5	89,84%
		7	89,65%
		9	91,09%
9.	45%	5	88,72%
		7	90,15%
		9	89,16%
10.	50%	5	89,28%
		7	89,78%
		9	90,09%
11.	55%	5	90,40%
		7	91,09%
		9	89,77%

12.	60%	5	89,84%
		7	90,15%
		9	90,21%
13.	65%	5	89,15%
		7	90,03%
		9	89,28%
14.	70%	5	89,22%
		7	89,28%
		9	89,22%
15.	75%	5	90,40%
		7	90,09%
		9	90,40%
16.	80%	5	89,40%
		7	89,90%
		9	89,84%
17.	85%	5	89,65%
		7	89,78%
		9	90,03%
18.	90%	5	88,97%
		7	89,22%
		9	89,59%

Pada tabel 4.4 dapat dilihat bahwasanya skenario uji coba dengan menggunakan *support vector machine* dengan kombinasi seleksi fitur chi square dengan menggunakan parameter $c=1$, $\gamma = 0.01$ serta nilai $K=5,7,9$ dengan menggunakan *threshold* sebesar 5% hingga 90%. Pada penggunaan fitur 5%-10% fitur menghasilkan akurasi yang cukup rendah berkisar 83%-86% karena terbatasnya jumlah fitur yang sedikit yang digunakan untuk *training data* sehingga model tidak bisa melakukan proses klasifikasi dengan baik. Pada penggunaan fitur 15%-55% mengalami peningkatan akurasi yang konsisten dengan puncak fitur 40%

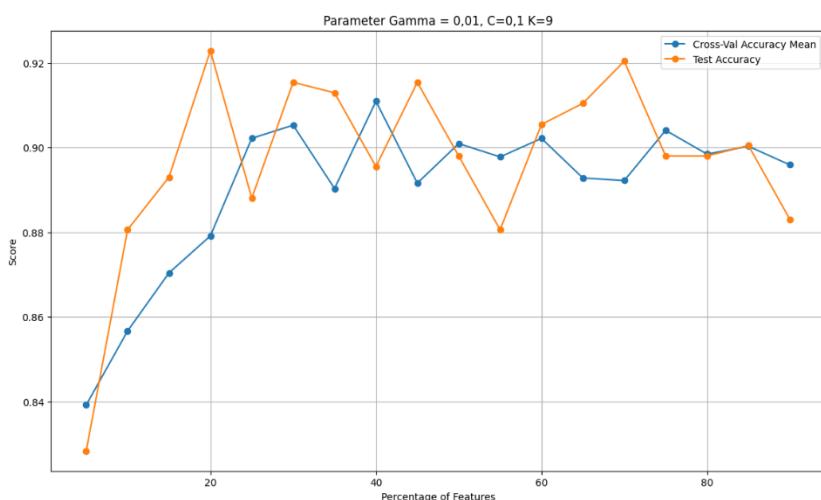
dengan menggunakan $k=9$ dengan hasil akurasi 91,09%, presisi 89,63% dan 89,55%. Ini menunjukkan bahwa penambahan fitur memiliki pengaruh dalam meningkatkan akurasi. Pada tabel 4.4. menunjukkan beberapa yang di *highlight* hal ini karena peneliti ingin menunjukkan hasil performa yang optimal pada setiap penggunaan nilai k dengan membandingkan hasil akurasi.



Gambar 4. 4 Confusion Matrix 55%
K=7

Gambar 4. 5. Confusion Matrix 40%
K=9

Pada gambar 4.4 serta gambar 4.5. merupakan evaluasi *confusion matrix* yang optimal saat mengugnakan 40% fitur dengan nilai $K=9$ dengan mendapatkan jumlah *True* positif sebanyak 885, jumlah *True* Negatif sebanyak 577 dan jumlah *False* positif sebanyak 77 serta *False* negatif sebanyak 66. yang dihasilkan yakni akurasi 91,09%, presision 89,63% dan recall 89,55%.



Gambar 4. 6 Grafik performa svm dan chisquare

Pada gambar 4.6 merupakan hasil grafik yang paling optimal dengan menggunakan parameter gamma = 0,01 c=0,1 serta nilai K=9. Karena semakin tinggi nilai K maka semakin rendah untuk variansi data yang akan di proses. Dalam grafik tersebut juga tidak terdapat indikasi overfitting karena jarak gap antara cross validation accuracy dan test accuracy tidak terlalu jauh cenderung stabil.

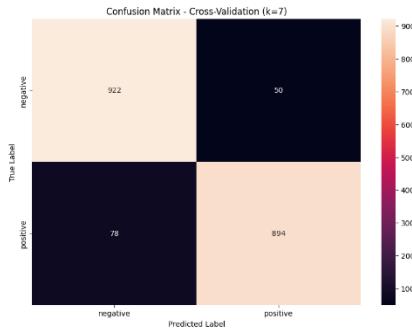
4.11. Hasil Skenario Uji Coba 3

Skenario pengujian ke 3 ialah pengujian model *Support Vector Machine* dengan menggunakan dataset yang seimbang dengan melakukan teknik oversampling SMOTE. Metode smote hanya dilakukan pada data train saja dengan menggunakan nilai 5,7 dan 9 sebagai nilai k-fold validation. Berikut hasil dari pengujian ketiga ini pada tabel 4.5.

Tabel 4. 5 Hasil Skenario Uji Coba 3

Skenario Uji coba 1	Kernel	Parameterr C	Parameterr gamma	K-fold	Akurasi
Support Vector Machine + Smote	Linear	1	0,01	5	92,79%
				7	93,41%
				9	93,15%

Pada tabel 4.5. diatas merupakan hasil dari *support vector machine* dengan menambahkan metode smote mampu meningkatkan performa model. Dimana mengalami peningkatan yakni K=7 yang optimal pada skenario uji coba ketiga. Kemudian hasil evaluasi *confusion matrix suppot vector machine* dengan menggunakan metode smote untuk imbalance data berikut hasilnya.



Gambar 4. 7 Confusion Matrix

Gambar 4.7 merupakan hasil dari evaluasi yang optimal yakni menggunakan nilai K=7 dengan menghasilkan sebanyak 891 *True* positif, 920 *True* negatif, 52 *false* positif dan 81 *false* negatif dengan menghasilkan akurasi sebesar 93,41%, presisi sebesar 93,45% dan recall sebesar 93,41%. Kemudian berikut grafik performa penggunaan nilai K=5,7,9.

4.12. Hasil Skenario Uji Coba 4

Pada skenario keempat yakni dengan mengkombinasikan seluruh skenario uji coba sebelumnya sehingga menggunakan algoritma *Support Vector Machine* dengan menggunakan metode smote untuk imbalance data serta seleksi fitur dengan chi square. Dengan menggunakan parameter c=1, gamma =0,01 dengan menggunakan nilai k-fold = 5,7,9. Berikut merupakan hasil dari skenario uji coba yang keempat sebagai berikut.

Tabel 4. 6 Hasil Skenario Uji Coba 4

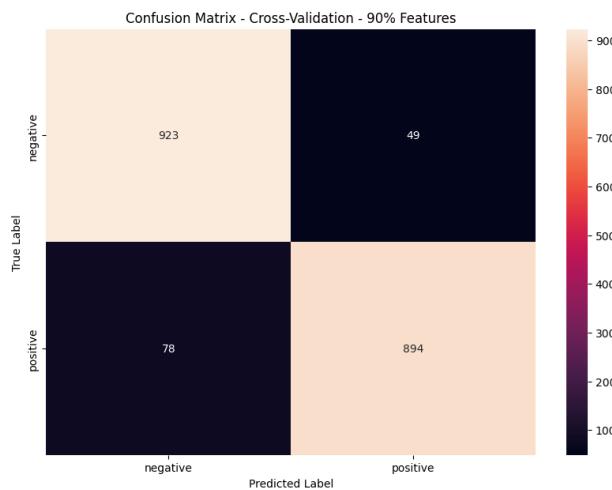
No.	Fitur & SVM	K-Fold	Akurasi
1.	5%	5	83,12%
		7	83,79%
		9	83,69
2.	10%	5	87,60%
		7	87,55%
		9	87,70%
3.	15%	5	88,89%

		7	89,45%
		9	89,19%
4	20%	5	90,58%
		7	90,79%
		9	90,22%
5	25%	5	91%
		7	90,74%
		9	90,99%
6.	30%	5	90,69%
		7	90,94%
		9	90,94
7.	35%	5	91,61%
		7	91,56%
		9	91,97%
8.	40%	5	92,02%
		7	92,02%
		9	92,28%
9.	45%	5	92,13%
		7	92,54%
		9	92,54%
10.	50%	5	92,02%
		7	92,59
		9	92,79%
11.	55%	5	92,18%
		7	92,80%
		9	92,69%
12.	60%	5	92,44%
		7	92,95%
		9	92,38%
13.	65%	5	92,59%

		7	93%
		9	92,54%
14.	70%	5	93%
		7	92,64%
		9	92,43%
15.	75%	5	93%
		7	93%
		9	92,74%
16.	80%	5	92,54%
		7	92,80%
		9	93%
17.	85%	5	93,10%
		7	92,90%
		9	93%
18.	90%	5	92,74%
		7	93,46%
		9	93,20%

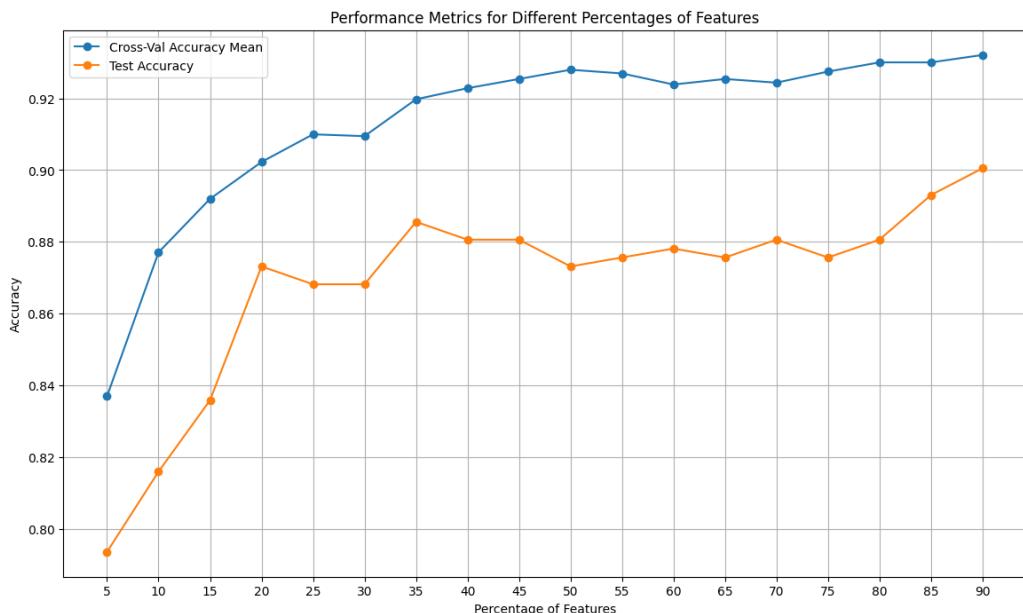
Pada tabel 4.6 dapat dilihat bahwasanya skenario uji coba dengan menggunakan *support vector machine* dengan kombinasi seleksi fitur chi square dan metode smote dengan menggunakan parameter $c=1$, $\gamma = 0.01$ serta nilai $K=5,7,9$ dengan menggunakan *threshold* sebesar 5% hingga 90%. Pada penggunaan fitur 5%-10% mengalami peningkatan dibandingkan dengan skenario uji yang ke-2 tanpa menggunakan smote menghasilkan 83% - 88% menunjukkan bahwasanya meskipun menggunakan fitur yang terbatas model dapat meningkatkan performanya. Pada penggunaan fitur 15%-60% mengalami peningkatan signifikan hingga mencapai 91% sampai 93% hal ini menunjukkan peningkatan jumlah fitur dapat meningkatkan sebuah akurasi pada model. Penggunaan fitur 85-90% merupakan performa yang paling tinggi dengan akurasi sebesar 93,46 dengan menggunakan nilai $k=7$. Pada tabel 4.4. diatas menunjukkan beberapa yang di

highlight hal ini karena peneliti ingin menunjukkan hasil performa yang optimal pada setiap penggunaan nilai k dengan membandingkan hasil akurasi.



Gambar 4. 8 Confusion Matrix

Dalam gambar 4.8 merupakan hasil dari evaluasi *confusion matrix* yang menghasilkan *True Positif* sebesar 894, *True Negatif* sebesar 923, *False Positif* 49, *False Negatif* 78 serta menghasilkan akurasi sebesar 93.46%, presisi 89.42%, recall 89.30%. berikut merupakan grafik dari hasil uji coba dengan menggunakan parameter $c=1$, $\gamma=0,01$ dan $k\text{-fold} = 7$.



Gambar 4. 9 Grafik Performa Svm dengan smote dan Chi Square

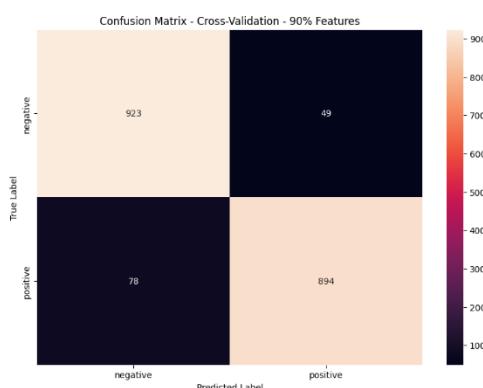
Pada gambar 4.9 merupakan sebuah grafik hasil dari pengujian dengan parameter $c=1$, $\gamma=0,01$ dengan menggunakan $k\text{-fold}=7$. Dapat dilihat bahwasanya Ketika menggunakan seleksi fitur dan metode SMOTE hasilnya terdapat peningkatan optimal dalam akurasinya. Lalu dengan menggunakan nilai $k\text{-fold} = 7$ memiliki peningkatan yang optimal pada saat penggunaan presentase fitur sekitar 30% berdasarkan rata-rata cross validation sementara pada penggunaan 70% mengalami peningkatan yang signifikan.

4.13. Analisa Skenario Uji coba.

Berdasarkan dari implementasi yang telah dilakukan, dengan membandingkan literatur yang ada dengan melakukan analisis sentimen dengan menggunakan *support vector machine* tanpa seleksi fitur dan metode chi square[24]. Pada penelitian ini dilakukan dengan pengujian menggunakan menggunakan nilai $k=5,7,9$ dengan menggunakan parameter $c=1$ dan $\gamma=0,01$. Hasil uji coba menunjukkan bahwasanya penambahan nilai $k\text{-fold}$ bisa mempengaruhi suatu performa model. Pada pengujian menghasilkan performa terbaik dengan menggunakan nilai $k\text{-fold} = 9$ dengan akurasi sebesar 90,15%, presisi 90,13 dan recall 90,15%. Hasil uji coba menunjukkan bahwasanya menambahkan nilai $k\text{-fold}$ bisa meningkatkan performa. Kemudian pengujian selanjutnya menggunakan

support vector machine dengan menggunakan seleksi fitur. Pada pengujian ini setiap nilai k-fold memiliki peningkatan dengan menggunakan seleksi fitur chisquare. Untuk K-fold yang mengalami kondisi peningkatan performa dengan optimal yakni menggunakan presentase fitur mulai dari 30% sampai 50%. Pada pengujian menggunakan chisquare hasil yang paling optimal yakni menggunakan presentase fitur 40% dengan k-fold = 9 dengan akurasi sebesar 91,09%, presisi 89,63% dan recall 89,55%. Kemudian pengujian selanjutnya penggunaan oversampling dengan menggunakan SMOTE juga mampu meningkatkan performa yang optimal semua nilai k-fold nilai akurasinya mencapai 93% dikarenakan sudah dilakukan *imbalance data* menambahkan sintesis berdasarkan k tetangga dari data kelas minoritas pada data *train*. Hasil yang paling optimal dalam pengujian ini dengan menggunakan k-fold=7 sebesar 93,41% akurasi, 93,45 presisi dan 93,41 recall.

Serta pada penelitian ini melakukan kombinasi dengan menggabungkan seleksi fitur *chisquare* dan metode *SMOTE* pada algoritma *Support Vector Machine*. Pada pengujian ini mampu meningkatkan performa meskipun hanya bertambah 0,05 pada akurasi, jika dibandingkan dengan pengujian dengan smote. Pengujian ini menghasilkan akurasi sebesar 93,46%, presisi 89,42%, recall 89,30% dengan menggunakan k-fold =7. Berikut merupakan hasil dari evaluasi *confusion matrix* pada gambar 4.10



Gambar 4. 10 Confusion Matrix

berdasarkan gambar 4.10. didapatkan *True* positif sebanyak 894, *True* negatif sebanyak 923, *False* Positif 49 dan *False* Negatif sebanyak 78 sehingga

menghasilkan akurasi sebesar 93,46%, presisi sebesar 89,42% dan recall sebesar 89,30%. Kemudian jika dilihat dari gambar 4 diketahui bahwa ini adalah kasus di mana kelas aktual adalah positif, tetapi diprediksi sebagai negatif oleh model. Ini menunjukkan bahwa meskipun SMOTE membantu menyeimbangkan data, beberapa contoh positif mungkin tidak cukup terwakili dalam fitur yang dipilih atau model masih mengalami kesulitan dalam mendeteksi semua contoh positif.

4.11. Implementasi Antarmuka

1. Halaman Awal

The screenshot shows a web application interface. On the left, there is a sidebar with a 'home' button highlighted in blue. Below it are links: Preprocessing, Word Cloud, Feature Extraction, smote, performa kinerja model, and View more. Under 'Menu', there is a 'Home' link. The main content area has a title: 'Website Analisis Sentimen Ulasan Aplikasi Bareksa dengan Pendekatan Lexicon-Based, Seleksi Fitur Chi-square, dan Support Vector Machine'. Below the title is a paragraph of text explaining the platform's purpose. At the bottom, there is a table with two rows of data samples:

Data	Sentiment	At Date
0 : aplikasinya ringan fitur reksa dananya lengkap cocok pemula mau investasi tampilan	Positif	2023-11-19
1 : umum sistem aplikasi digunakan aplikasi standard mirip antarmuka standart pelayan	Negatif	2023-11-21

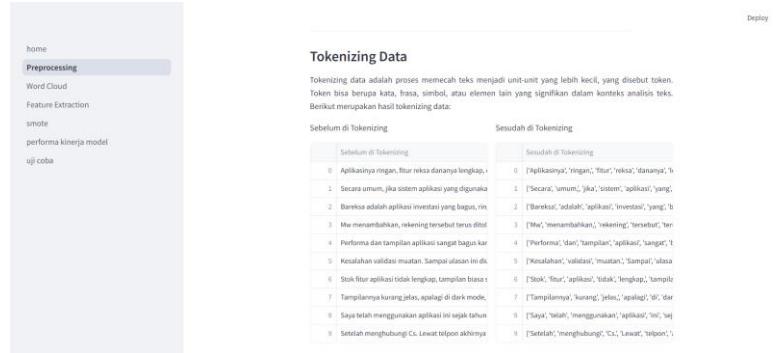
Gambar 4. 11 Halaman Home

Pada gambar 4.11. halaman awal berisikan beberapa kata yang menjelaskan kegunaan dan deskripsi dari website, serta terdapat dataset yang digunakan dalam penelitian ini.

2. Halaman Preprocessing

Pada halaman ini berisikan beberapa tahapan yang berfungsi untuk mengelola data mentah menjadi data yang siap untuk diproses atau *text preprocessing*. Pada halaman ini berisikan 5 tahapan yakni *tokenizing*, *clean&casefolding*, normalisasi, *stopword removal*, lemmatisasi.

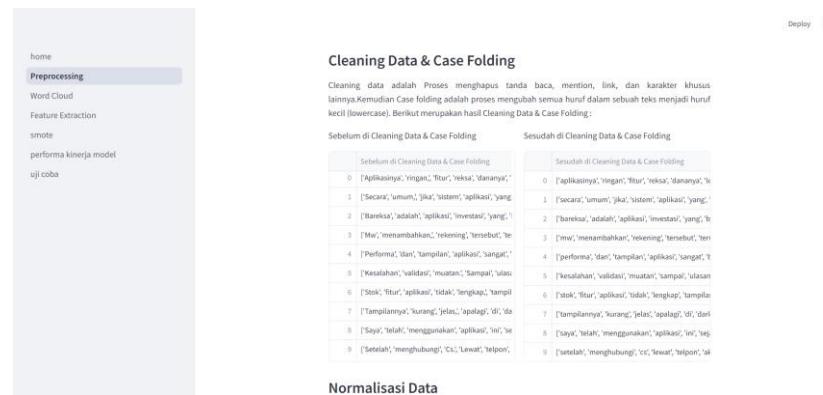
a. Tokenizing



Gambar 4. 12 Halaman Tokenizing

Pada gambar 4.12 halaman ini menampilkan proses dari tokenizing tersebut yang berisikan dataset asli yang belum dilakukan tokenizing dan hasil dari tokenizing seperti yang ditunjukkan pada gambar 4.10 diatas.

b. Clean & Case Folding



Gambar 4. 13 Clean & CaseFolding

Pada gambar 4.13 merupakan halaman yang menampilkan proses *cleaning* & *case folding* yang berguna sebagai menghapus karakter yang unik seperti emoji, menghapus tanda baca dan mengubah format menjadi *lowercase*.

c. Normalisasi

Sebelum di Normalisasi Data	Sesudah di Normalisasi Data
0 ['aplikasinya', 'ringan', 'blor', 'reksa', 'dananya', 'ik']	0 ['aplikasinya', 'ringan', 'blor', 'reksa', 'dananya', 'ik']
1 ['secara', 'umum', 'jika', 'sistem', 'aplikasi', 'yang', '']	1 ['secara', 'umum', 'jika', 'sistem', 'aplikasi', 'yang', '']
2 ['bereka', 'adalah', 'aplikasi', 'investasi', 'yang', 'b']	2 ['bereka', 'adalah', 'aplikasi', 'investasi', 'yang', 'b']
3 ['mau', 'menambahkan', 'rekening', 'tersebut', 'ter']	3 ['mau', 'menambahkan', 'rekening', 'tersebut', 'ter']
4 ['performa', 'dan', 'tampilan', 'aplikasi', 'sangat', 't']	4 ['performa', 'dan', 'tampilan', 'aplikasi', 'sangat', 't']
5 ['kesalahan', 'validasi', 'mustan', 'sampa', 'vlasan']	5 ['kesalahan', 'validasi', 'mustan', 'sampa', 'vlasan']
6 ['stok', 'fitur', 'aplikasi', 'tidak', 'lengkap', 'tampila']	6 ['stok', 'fitur', 'aplikasi', 'tidak', 'lengkap', 'tampila']
7 ['tampilannya', 'kurang', 'jelas', 'aplagi', 'di', 'dari']	7 ['tampilannya', 'kurang', 'jelas', 'aplagi', 'di', 'dari']
8 ['taya', 'telah', 'menggunakan', 'aplikasi', 'ini', 'sej']	8 ['taya', 'telah', 'menggunakan', 'aplikasi', 'ini', 'sej']
9 ['setelah', 'menghubungi', 'customer service', 'lew']	9 ['setelah', 'menghubungi', 'customer service', 'lew']

Gambar 4. 14 Halaman Normalisasi

Pada gambar 4.14 menunjukkan halaman normalisasi yang berfungsi sebagai memperbaiki kata yang tidak baku menjadi baku seperti yang ditunjukkan gambar diatas

d. Stopword Removal

Sebelum di Stopword Removal	Sesudah di Stopword Removal
0 ['aplikasinya', 'ringan', 'blor', 'reksa', 'dananya', 'ik']	0 ['aplikasinya', 'ringan', 'blor', 'reksa', 'dananya', 'ik']
1 ['secara', 'umum', 'jika', 'sistem', 'aplikasi', 'yang', '']	1 ['umum', 'sistem', 'aplikasi']
2 ['bereka', 'adalah', 'aplikasi', 'investasi', 'yang', 'b']	2 ['bereka', 'adalah', 'aplikasi', 'investasi']
3 ['mau', 'menambahkan', 'rekening', 'tersebut', 'ter']	3 ['mau', 'menambahkan', 'rekening', 'tersebut']
4 ['performa', 'dan', 'tampilan', 'aplikasi', 'sangat', 't']	4 ['performa', 'dan', 'tampilan', 'aplikasi', 'sangat']
5 ['kesalahan', 'validasi', 'mustan', 'sampa', 'vlasan']	5 ['kesalahan', 'validasi', 'mustan', 'sampa']
6 ['stok', 'fitur', 'aplikasi', 'tidak', 'lengkap', 'tampila']	6 ['stok', 'fitur', 'aplikasi', 'tidak', 'lengkap']
7 ['tampilannya', 'kurang', 'jelas', 'aplagi', 'di', 'dari']	7 ['tampilannya', 'kurang', 'jelas', 'aplagi']
8 ['taya', 'telah', 'menggunakan', 'aplikasi', 'ini', 'sej']	8 ['taya', 'telah', 'menggunakan', 'aplikasi']
9 ['setelah', 'menghubungi', 'customer service', 'lew']	9 ['setelah', 'menghubungi', 'customer service']

Gambar 4. 15 Halaman Stopword Removal

Pada gambar 4.15 menunjukkan proses *StopWord Removal* yang berfungsi untuk menghapus kata yang tidak penting dan tidak bermakna, seperti yang ditunjukkan pada gambar diatas.

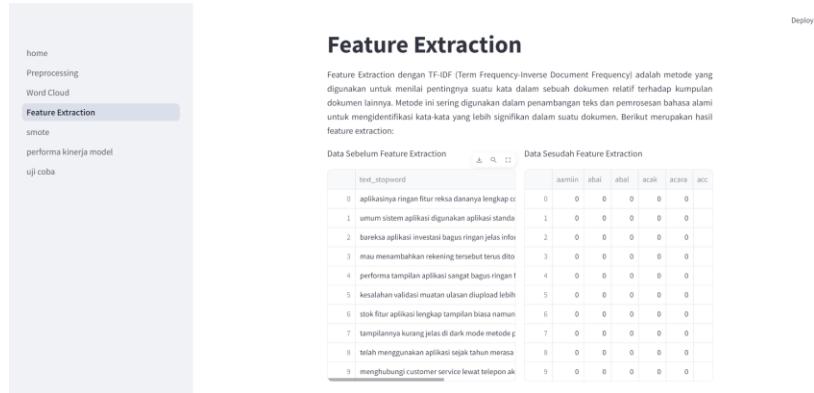
3. Word Cloud



Gambar 4. 16 Halaman Word Cloud

Pada gambar 4.16 menunjukkan proses visualisasi yang menggunakan *word cloud* yang berisikan kata yang paling sering muncul dalam dataset.

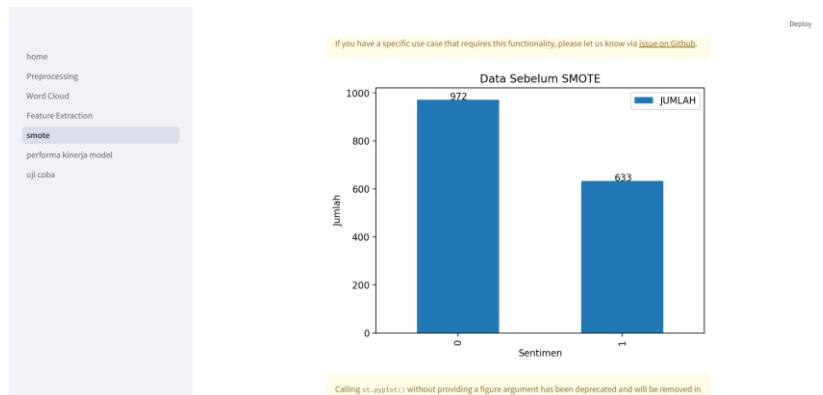
4. Feature Extraction



Gambar 4. 17 Halaman Feature Extraction

Pada gambar 4.17 halaman yang menunjukkan proses fitur ekstraksi menggunakan tf-idf yang berfungsi menghitung kata yang paling sering muncul, seperti yang ditunjukkan gambar diatas.

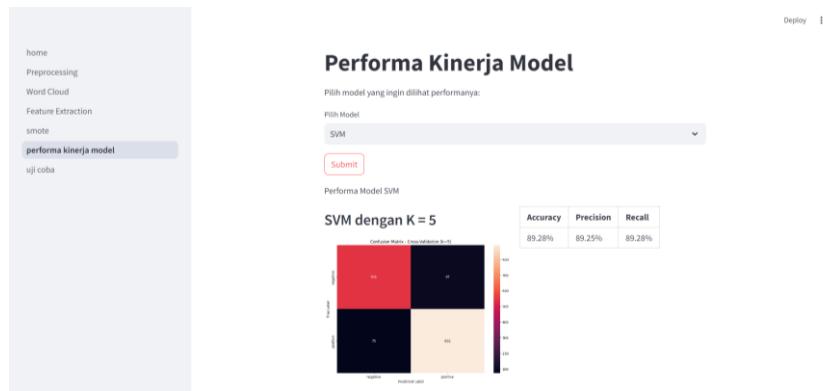
5. SMOTE



Gambar 4. 18 Halaman Smote

Pada halaman gam bar 4.18 menunjukkan fungsi smote yang berfungsi sebagai *imbalance data* dimana menyeimbangkan data agar bisa memoptimalkan model seperti yang ditunjukkan seperti gambar diatas.

6. Performa kinerja model



Gambar 4. 19 Halaman Performa Kinerja Model

Pada gambar 4.19 menunjukkan halaman yang berisikan performa kinerja model yang berisikan hasil performa disetiap k-fold validation seperti yang ditunjukkan gambar diatas.

7. Uji Coba

The screenshot shows a web-based sentiment analysis tool. On the left, a sidebar menu includes 'home', 'Preprocessing', 'Word Cloud', 'Feature Extraction', 'smote', 'performa kinerja model', and 'ujicoba' (which is highlighted). The main content area has a title 'Uji Data Dengan Data Baru'. Below it, a 'Data Input' section contains a text input field with the text 'Aplikasi sangat bagus dan sangat membantu untuk pemula' and a 'Submit' button. To the right, there are two columns of prediction results. The first column, under 'Model SVM', shows 'value' and 'Positif'. The second column, under 'Model SVM + SMOTE', also shows 'value' and 'Positif'. Below these, another pair of columns shows results for 'Model SVM + Chi Square' and 'Model SVM + SMOTE + Chi Square', both also showing 'value' and 'Positif'. A 'Deploy' button is located at the top right of the main content area.

Gambar 4. 20 Halaman Uji Coba

Pada gambar 4.20 halaman ini menunjukkan hasil dari prediksi data baru apakah ulasan tersebut memiliki sentiment positif atau negatif seperti gambar diatas.

Bab V

PENUTUP

5.1. Kesimpulan

Berdasarkan dari implementasi hasil penelitian yang dilakukan menunjukkan bahwa penggunaan algoritma Support Vector Machine (SVM) dalam analisis data teks memberikan performa yang optimal, terutama ketika memperhatikan aspek-aspek seperti penambahan parameter C dan gamma serta penerapan validasi silang k-fold untuk mengurangi indikasi overfitting. Dalam penelitian ini, dengan parameter $c=1$, $\gamma = 0,01$, dan $k\text{-fold} = 7$ dan 9 , model mencapai akurasi tertinggi sebesar 90,15%. Selain itu, penggunaan seleksi fitur Chi-Square terbukti mampu meningkatkan performa model, di mana penggunaan seleksi fitur mulai dari 20% menghasilkan peningkatan akurasi, presisi, dan recall. Peningkatan paling signifikan terjadi pada 40% seleksi fitur, dengan akurasi tertinggi mencapai 91,09%, presisi 89,63%, dan recall 89,55%, yang menunjukkan peningkatan sebesar 0,94% dibandingkan tanpa seleksi fitur.

Pada skenario penggunaan metode SMOTE dalam menangani data tidak seimbang memberikan kontribusi yang signifikan terhadap peningkatan performa model. Tanpa SMOTE, akurasi tertinggi yang dicapai adalah 90,15%, tetapi setelah penerapan SMOTE, akurasi meningkat hingga 93%, dengan peningkatan sebesar 2,85% untuk semua nilai k-fold yang digunakan (5, 7, dan 9). Kombinasi antara SVM, SMOTE, dan seleksi fitur Chi-Square memberikan peningkatan performa yang paling signifikan. Pada skenario uji coba ke-4, dengan kombinasi SMOTE dan 90% seleksi fitur, serta parameter $C=1$, $\gamma = 0,01$, dan $k\text{-fold} = 9$, akurasi tertinggi yang dicapai adalah 93,46%, yang menunjukkan peningkatan sebesar 3,31% dibandingkan baseline tanpa SMOTE dan seleksi fitur.

Maka dari itu dapat disimpulkan bahwasanya pengaruh peningkatan *accuracy*, *precision* dan *recall* dengan menggunakan *support vector machine*, seleksi fitur chi square serta penggunaan dari SMOTE memiliki peningkatan yang cukup signifikan sebesar 3,31% maka dari itu dapat disimpulkan bahwasanya

penggunaan seleksi fitur chi square dan smote memberikan pengaruh dalam peningkatan *accuracy*, *precision* dan *recall*.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat saran untuk penelitian selanjutnya yakni :

- a. Menggunakan data yang lebih banyak dan seimbang sehingga pada proses training menghasilkan performa yang optimal.
- b. Menggunakan Hyperparameter tuning agar meningkatkan akurasi yang lebih tinggi serta menghindari *overfitting* dan *underfitting*.
- c. Memperhatikan kamus yang digunakan untuk melakukan pelabelan.
- d. Menggunakan dataset yang berbeda dengan menggunakan metode yang telah diusulkan.

Daftar Pustaka

- [1] C. I. tech-Tim, “No Tipu-Tipu! Ini 8 Aplikasi Investasi Aman & Terdaftar OJK,” *CNBC INDONEISA*, 2021.
<https://www.cnbcindonesia.com/tech/20211109125635-37-290122/no-tipu-tipu-ini-8-aplikasi-investasi-aman-terdaftar-ojk> (accessed Mar. 05, 2024).
- [2] N. M. S. Hadna, I. S. Paulus, and W. Winarno, “Studi Literatur Tentang Perbandingan Metode Untuk Proses Analisis Sentimen Di Twitter,” *Semin. Nas. Teknol. Inf. dan Komun.*, no. March, pp. 1–8, 2016.
- [3] R. Hermansyah and R. Sarno, “Sentiment analysis about product and service evaluation of pt telekomunikasi Indonesia tbk from tweets using textblob, naive bayes & K-NN Method,” *Proc. - 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Secur. Age Digit. Disruption, iSemantic 2020*, pp. 511–516, 2020, doi: 10.1109/iSemantic50169.2020.9234238.
- [4] S. Lestari and S. Saepudin, “Support Vector Machine: Analisis Sentimen Aplikasi Saham di Google Play Store,” *JUSIFO (Jurnal Sist. Informasi)*, vol. 7, no. 2, pp. 81–90, 2021, doi: 10.19109/jusifo.v7i2.9825.
- [5] P. Gupta, S. Kumar, R. R. Suman, and V. Kumar, “Sentiment Analysis of Lockdown in India during COVID-19: A Case Study on Twitter,” *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 4, pp. 939–949, 2021, doi: 10.1109/TCSS.2020.3042446.
- [6] M. N. Muttaqin and I. Kharisudin, “Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor,” *UNNES J. Math.*, vol. 10, no. 2, pp. 22–27, 2021, [Online]. Available: <http://journal.unnes.ac.id/sju/index.php/ujm>
- [7] G. R. Ditami, E. F. Ripanti, and H. Sujaini, “Implementasi Support Vector Machine untuk Analisis Sentimen Terhadap Pengaruh Program Promosi

- Event Belanja pada Marketplace,” *J. Edukasi dan Penelit. Inform.*, vol. 8, no. 3, p. 508, 2022, doi: 10.26418/jp.v8i3.56478.
- [8] R. Obiedat *et al.*, “Sentiment Analysis of Customers’ Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution,” *IEEE Access*, vol. 10, pp. 22260–22273, 2022, doi: 10.1109/ACCESS.2022.3149482.
- [9] M. D. Purbolaksono, D. T. B. Pratama, and ..., “Perbandingan Gini Index dan Chi Square pada Sentimen Analisis Ulasan Film menggunakan Support Vector Machine Classifier,” *JEPIN (Jurnal Edukasi dan ...)*, vol. 9, no. 3, pp. 528–534, 2023, [Online]. Available: <https://jurnal.untan.ac.id/index.php/jepin/article/view/68845> <https://jurnal.untan.ac.id/index.php/jepin/article/download/68845/75676600511>
- [10] P. Rama, B. Putra, and R. S. Perdana, “Klasifikasi Judul Berita Online menggunakan Metode Support Vector Machine (SVM) dengan Seleksi Fitur Chi-square,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 5, pp. 2132–2141, 2023.
- [11] V. Nurcahyawati and Z. Mustaffa, “Vader Lexicon and Support Vector Machine Algorithm to Detect Customer Sentiment Orientation,” *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 1, pp. 108–118, 2023, doi: 10.20473/jisebi.9.1.108-118.
- [12] Bareksa, “Grow With Bareksa,” *PT. Bareksa Marketplace Indonesia*, 2024. <https://www.bareksa.com/>
- [13] Aftech Indonesia, “PT. BAREKSA PORTAL INVESTASI,” *Aftech Indonesia*, 2023. <https://fintech.id/id/member/detail/71>
- [14] S. S. Salim and J. Mayary, “Analisis Sentimen Pengguna Twitter Terhadap Dompet Elektronik Dengan Metode Lexicon Based Dan K – Nearest Neighbor,” *J. Ilm. Inform. Komput.*, vol. 25, no. 1, pp. 1–17, 2020, doi:

10.35760/ik.2020.v25i1.2411.

- [15] R. Kurz, C. Sheya, K. Brun, and H. R. Simmons, “Journal Of Southwest Jiaotong University,” *High Temp.*, vol. 57, no. No. 6, pp. 20–25, 2022.
- [16] A. D. Adhi Putra, “Analisis Sentimen pada Ulasan pengguna Aplikasi Bibit Dan Bareksa dengan Algoritma KNN,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 2, pp. 636–646, 2021, doi: 10.35957/jatisi.v8i2.962.
- [17] A. Firdaus, W. I. Firdaus, P. Studi, T. Informatika, M. Digital, and P. N. Sriwijaya, “Text Mining,” vol. 13, no. 1, pp. 66–78, 2021.
- [18] TIM CODINGSTUDIO, “Web Scraping: Pengertian, Fungsi, Cara Kerja dan Contohnya,” *PT Semua Mahir Teknologi (SMART)*, 2023.
<https://codingstudio.id/blog/web-scraping-adalah/>
- [19] V. A. Flores, P. A. Permatasari, and L. Jasa, “Penerapan Web Scraping Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan Keyword,” *Maj. Ilm. Teknol. Elektro*, vol. 19, no. 2, p. 157, 2020, doi: 10.24843/mite.2020.v19i02.p06.
- [20] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest,” ... *Teknol. Inf. dan ...*, vol. 6, no. 9, pp. 4305–4313, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [21] Y. Asri, W. N. Suliyanti, D. Kuswardani, and M. Fajri, “Pelabelan Otomatis Lexicon Vader dan Klasifikasi Naive Bayes dalam menganalisis sentimen data ulasan PLN Mobile,” *Petir*, vol. 15, no. 2, pp. 264–275, 2022, doi: 10.33322/petir.v15i2.1733.
- [22] L. Luthfiana, J. C. Young, and A. Rusli, “Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback

- Aplikasi,” *Ultim. J. Tek. Inform.*, vol. 12, no. 2, pp. 125–126, 2020, doi: 10.31937/ti.v12i2.1828.
- [23] M. Rizinski, H. Peshov, K. Mishev, M. Jovanovik, and D. Trajanov, “Sentiment Analysis in Finance: From Transformers Back to eXplainable Lexicons (XLex),” *IEEE Access*, vol. 12, no. December 2023, pp. 7170–7198, 2024, doi: 10.1109/ACCESS.2024.3349970.
- [24] F. E. Kavabilla, T. Widiharih, and B. Warsito, “Analisis Sentimen Pada Ulasan Aplikasi Investasi Online Ajaib Pada Google Play Menggunakan Metode Support Vector Machine Dan Maximum Entropy,” *J. Gaussian*, vol. 11, no. 4, pp. 542–553, 2023, doi: 10.14710/j.gauss.11.4.542-553.

LAMPIRAN

1. Surat Validasi Data

SURAT KETERANGAN VALIDASI

Yang bertanda tangan dibawah ini:

Nama : Ersa Alami, S.Pd., M.Pd.

Pendidikan : S2 Pendidikan Bahasa dan Sastra Indonesia

Jabatan : Dosen Program Studi Pendidikan Bahasa dan Sastra Indonesia

Telah membaca, memeriksa, dan memvalidasi label data dari ulasan Pantai Jawa Timur pada aplikasi Google Maps pada penelitian skripsi dengan judul “Analisis Sentimen Ulasan Aplikasi Bareksa dengan Pendekatan *Lexicon-Based*, Seleksi Fitur *Chi-square*, dan *Support Vector Machine*”, oleh peneliti:

Nama : Fathony Syaennulloh

NIM : 200411100073

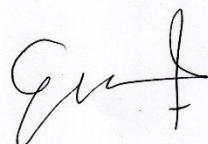
Program Studi : Teknik Informatika

Universitas : Universitas Trunojoyo Madura

Menyatakan bahwa label data dari ulasan Pantai Jawa Timur pada aplikasi Google Maps ini layak dengan revisi / ~~tidak layak untuk digunakan~~. Berikut adalah saran atau komentar yang validator berikan:

.....
.....
.....
.....

Bangkalan, 22 Juli 2024



Ersa Alami, S.Pd., M.Pd

NIDN. 19900428202412001

DAFTAR RIWAYAT HIDUP



FATHONY SYAENNULLOH

Alamat: Jl. Letnan Abdullah, Desa Tanjung Jati, Kec. Kamal, Kab. Bangkalan
Ponsel: +6285853600656
Email: fathony404@gmail.com
LinkeDin: <https://www.linkedin.com/in/syaennulloh>

TENTANG SAYA

Saya adalah seorang mahasiswa dengan minat yang kuat dalam bidang IT, khususnya dalam pengolahan data dan analisis data. Saya memiliki kemampuan yang baik dalam mengolah dan menganalisis data menggunakan Microsoft Excel dan memvisualisasikannya dengan alat seperti Tableau dan Looker Studio. Selain itu, saya memiliki pengalaman dalam pengembangan aplikasi website menggunakan framework Laravel, serta kemampuan UI/UX Design yang telah menghasilkan berbagai portofolio yang diakui. Saya adalah individu yang berorientasi pada detail dan mampu bekerja dalam tim, selalu siap untuk belajar hal baru, dan berkontribusi dalam memecahkan masalah dengan pendekatan berbasis data.

PENDIDIKAN

Universitas Trunojoyo Madura	Aug 2020 - Sep 2024
Predikat Puji dengan IPK 3.60	
<ul style="list-style-type: none">Jurusan Teknik Informatika.Judul Skripsi "Analisis Sentimen Ulasan Aplikasi Bareksa Dengan Pendekatan Lexicon Based, Metode Smote serta seleksi fitur Chi-Square dengan penggunaan Support Vector Machine .	

PENGALAMAN KERJA DAN ORGANISASI

Digistar Class by Telkom Indonesia sebagai Hipster	Juli 2024 - Present
<ul style="list-style-type: none">Terpilih dari 12.176 pendaftar sebagai salah satu dari 500 pesertaMendapatkan pelatihan intensif untuk mengembangkan berbagai soft skill dan hard skill terutama pada bidang Hipster (Ui/Ux)Memiliki Project Akhir dengan Use Case Website Looge	
Enca Production sebagai Content Creator	Nov 2023 - Present
<ul style="list-style-type: none">Membantu mensukseskan acara perlombaan terbesar se-Kabupaten Bangkalan divisi ITMembuat 10 konten dalam acara pemilu 2024 dengan insight tertinggi 20k lebih viewers	
Beasiswa Djrum Plus	Sep 2022 - Sep 2023
<ul style="list-style-type: none">Terpilih dari 13.703 pendaftar sebagai salah satu dari 521 pesertaMendapatkan pelatihan <i>character building</i>, Leadhership Development dan National BuildingMendapatkan Softskill antara lain pengembangan kepemimpinan, pembentukan karakter, dan pembangunan nasional serta mendapatkan relasi yang cukup membantu pada masa depan saya.	
Sribu.com Freelace Ui/Ux Designer	Jan 2022- Jan 2023
<ul style="list-style-type: none">Mendapatkan sekitar 10 Customer dalam periode 1 Tahun	
Internship Solite Kids Studio sebagai Ui/UX Designer	Jan 2022 - Mar 2023
<ul style="list-style-type: none">Ikut serta membangun Super Apps Solite KidsRedisgn Profile Company Website Solite Kids StudioMembuat 10 Desain Game	

Unit Kegiatan Mahasiswa Creative Computer Club (UKM TRIPLE-C)	Jan 2022- Des 2022
<ul style="list-style-type: none"> • Terpilih sebagai Kepala Divisi Edukasi yang beranggotakan 8 orang • Mendapatkan sekitar 10 prestasi anggota yang berhasil dibimbing • Mendapatkan rewards Organisasi Ter-Inovasi dalam Kampus dengan membuat dan mengembangkan website pendaftaran organisasi pada seluruh mahasiswa 	

PRESTASI

Mandalika Essai Competition	Sep 2024
<ul style="list-style-type: none"> • Terpilih Sebagai Finalis 10 Besar dari 105 Team Tingkat Nasional • Judul Essai "Mahaltra (Madura Halal Nusantara) : Digitalisasi Pariwisata Halal Melalui Community Based Tourism Berbasis Mobile dengan metode <i>User Centered Design</i> (UCD) dan <i>Geolocation</i> Guna Mewujudkan Kesejahteraan Masyarakat Madura • Saya Sebagai Ui/Ux Designer dalam pembuatan aplikasi Mahaltra 	
Festival Ajang Sinergitas Teknologi (FAST)	Sep 2022
<ul style="list-style-type: none"> • Juara 1 Lomba Kategori Lomba Ui/Ux Desain • Membuat Desain Aplikasi Berbasis Mobile Growth PKL (aplikasi khusus pedagang Kaki Lima) 	

Perlombaan Kompetisi Web Desain, Universitas Jambi	Sep 2022
<ul style="list-style-type: none"> • Juara 2 Lomba Kategori Webdesain • Membuat Aplikasi berbasis website TambalBan • Sebagai Front End Developer 	

ADDITIONAL INFORMATION

- **Technical Skills:** Web Developer, Full stack Ui/Ux Design, Data Science atau Data Analys,
- **Languages:** Inggris
- **Certifications:** Professional Engineer (PE) License, Project Management Professional (PMP).