PROPOSAL SKRIPSI

Analisis Sentimen Ulasan Aplikasi Bareksa dengan Pendekatan *Lexicon-Based*, Seleksi Fitur *Chi-square*, dan *Support Vector Machine*

Disusun 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 TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2024

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

Daftar Gambar

Daftar Tabel

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 Aplikasi

Dapat dilihat 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 penelitian terkait yang pernah dilakukan sebelumnya lebih tepatnya pada tahun 2020, Reza Hermansyah *et al* [3], dalam penlitian ini peneliti 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].

Setahun kemudian terdapat penelitian tentang analisis sentimen lebih tepatnya tahun 2021, 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 yang sama, 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].

Pada Tahun 2022 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], melakukan penelitian pada tahun 2023 yang 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 *fi socre* 90.51%. Sedangkan hasil dari analysis sentimen dengan menggunakan bantuan vader lexicon meningkat menjadi akuras 86.57%, presisi 89.71%, recal 96.32%, dan *f-1 score* 92.89% [11].

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 edngan 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 tesebut bisa menjadi bahan evaluasi untuk bareksa sendiri, tetapi tidak adanya sistem yang bisa menganalisis hal tersebut. Terkait permasalahan ini dibutuhkan suatu sistem sentiment 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, bagaimana pengaruh dalam peningkatan performa *accuracy, precision, recall* metode *support vector machine* dengan menggunakan seleksi fitur *Chi Square* serta menggunakan smote.

1.3. Tujuan dan Manfaat

1.3.1. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Mengetahui pengaruh peningkatan performa *accuracy*, *precision recall* dan penggunaan *support vector machine* dengan menggunakan seleksi fitur serta.
2. Membuat bahan referensi untuk evaluasi aplikasi bareksa melalui sistem sentiment analisis.

1.3.2. Manfaat Penelitian

Adapun manfaat dilakukannya penelitian ini adalah :

1. Dapat menjadi bahan referensi untuk bahan evaluasi aplikasi bareksa dalam mempertahankan dan meningkatkan pelayanan aplikasi bareksa
2. 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 1007 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 sentiment dijadikan ke dalam 2 Kategori yaitu sentiment positif sebanyak 740 ulasan dan sentiment negatif sebanyak 267 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 serta proses klasifikasi menggunakan *support vector machine*.

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 memilik 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 memliki 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 pemorsesan 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 berhimbuhan 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 sensitiem 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 TF-IDF. Frasa yang jarang digunakan pada dokumen yang ada memiliki bobot *Ineverse 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 :

$$\mathbf{tfidf(}\boldsymbol{i, j}\mathbf{)} = \mathrm{tf}(i,j)\times(\log(\frac{N}{df(\mathrm{j})}) + 1 \quad \text{.....(2.1)}$$

Keterangan :

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

i = 1,2,3,…N.

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

df(*i*) = frekuensi.dokumen yang mengandung.*term* j dari.semua.koleksi.dokumen.

2.8. SMOTE

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

Metode ini menggunakan teknik statistic untuk menambah jumlah kasus dalam suatu kumpulan data yang tidak seimbang dengan cara mensisntesis ssasmple baru dari kelas minoritas. metode Smote dapat digunakan untuk mengatasi permasalahan ketidakseimbangan kelas (*imbalance class*) dalam kalsifikasi machine learning. SMOTE tidak menjamin akan menghasilkan model yang lebih akurat, sehingga perlu dilakukan percobaan dengan presentase yang berbeda, set fitur yang berbeda dan jumlah tetangga terdekat yang berbeda untuk melihat bagaimana penambahan kasus mempengaruhi model. Selain itu SMOTE juga bisa digunakan pada praproses untuk meningkatkan akurasi pada model yang digunakan. Untuk menggunakan metode SMOTE dapat digunakan persamaan 2.2.

Xsyn = $X_i + (X_{knn} - X_I) \times \delta$.....(2.2)

Keterangan :

Xsyn = data sintesis yang akan dibuat

Xi = data yang akan direpilikasi.

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:

$$X^2(t, c) = \frac{N \times (A \times D - C \times B)^2}{(A + C)(B + D)(A + B)(C + D)} \quad \text{.....(2.3)}$$

Keterangan :

X^2 = 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 *flate 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 :

$$K(x, y) = \sum_{i=1}^{n} xi \times yi \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.4)$$

4. Hitung matriks dengan formula sebagai berikut :

$$D_{ij} = Y_i Y_j (K (X_i X_j) + \lambda^2) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.5)$$

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:

$$E_i = \sum_{j=1}^{i} \alpha_j D_{ij} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.6)$$

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 :

   $\delta\alpha_i = \min\{\max\ [\gamma(1-E_i)-E_i\ ]\ C-\ \alpha_i$……..(2.7)

   Keterangan:

   $\alpha_i =$ alfa nilai ke – i

   $\gamma =$ gamma.untuk.mencari.kecepatan

   $E_{ij} =$ rata-rata eror

   $C =$ untuk menentukan batas nilai alfa

7. Hitung nilai alpha baru dengan formula

   $\alpha_{\ i} =\ \alpha_i + \delta\alpha_i$…….(2.8)

   Keterangan.:

   $\alpha_i =$ alfa nilai ke – *i*

   $\delta\alpha_i =$ delta.alfa nilai ke – *i*

8. Hitung nilai bias dengan formula

   $b = -\frac{1}{2}(\text{w}.x^{+} +\ \text{w}.x^{x})$……(2.9)

   Keterangan:

   Wi^{+}.adalah.bobot.dot.product.data.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:

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

Keterangan :

α_i = alfa.nilai.ke-i

xi = 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 dalam klasifikasi digunakan sebagai dasar untuk mengevaluasi algoritma klasifikasi dengan menggunakan metode *confusion matrix*. Cara kerja dalam kalsifikasi akan dipengarui 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 memilliki beberapa pengukuran yang dapat digunakan yang akan menghasilkan *accurasi, precision, recall, specifity, fi-score* berikut rumus yang dimaksud :

1. Akurasi

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

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

2. Precicion

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

   $$Precision = \frac{TP}{TP+FP} \text{.....(2.12)}$$

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* :

   $$Recall = \frac{TP}{TP+FN} \text{.....(2.13)}$$

4. *Specificity*

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

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

5. *F1 Score*

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

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

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 swuare 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, pengucian parameter c, pengujian parameter lambda, pengujian konstanta gamma, pengujian parameter epsilon, dan pengujian parameter iterasi, dan menggunakanpengujian threshold chi square. Hasil dari penelitian ini yakni menghasilkan akurasi yang cukup baik sebesar Ketika menggunakan pengujian threshold chisquare 80% denga 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 Analsis 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 inni

menghasilkan akurasi yang cukup baik bagi keduaanya 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 socre* 90.51%. Sedangkan hasil dari analysis sentimen dengan menggunakan batnuan vader lexicon meningkat menjadi akuras 86.57%, presisi 89.71%, recal 96.32%, dan *fi score* 92.89% [11].

Tabel 2. 2 Penelitian Terkait

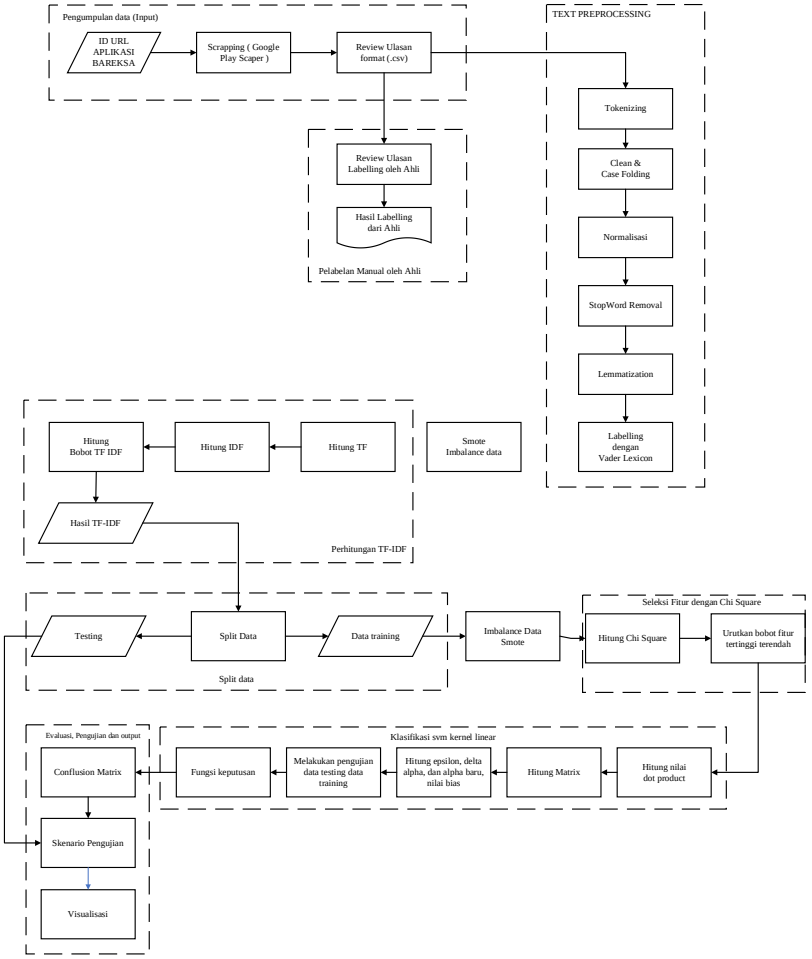
Nama Peneliti	**Judul Penelitian**	**Metode**	**Hasil Penelitian**
reza hermansyah, riyanarto sarno (2020)	sentiment analysis about product and service evaluation of pt telekomunikasi indonesia tbk from tweets using textblob, naive bayes & k-nn method	textblob, knn dan naïve bayes	penelitian ini menghasilkan akurasi *textblob* 54,67%, *naïve bayers* 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 bibit 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 *countvectorizer*	dengan menggunakan textblob dan vader lexicon sebaagai

			pelabelan, countvectorizer 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 neighbo	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 ,	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

ahossam faris (2022)			89%.
mahendra dwifebri purbolaksono, deninsyah tiya bella pratama, fahmi hamzah (2023)	perbandingan gini index dan chi square pada sentimen analsis ulasan film menggunakan support vector machine classifie	svm, gini index dan chi square	gini index sebesar 86.8% dan chi square sebesar 89.2%
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, svm 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 : (besok diganti total)

a. 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.
b. 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 menyeimbangkalan 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 chisqueare.
h. lalu diproses klasifikasi dengan metode *Support Vector Machine*.
i. Terakhir proses yang dilakukan yakni pengujian menggunakan *confusion matrix* dan visualisasi sentiment ulasan dengan menggunakan *wordcloud*.

3.2. Web Scraping

Langkah pertama dalam penelitan 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 menendapatkan ulasan sesuai kebutuhan yang telah ditentukan. Dalam proses web scrarpping ini mendapatkan 1007 ulasan dari aplikasi bareksa yang berbentuk csv.

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	Secara umum sistem aplikasinya Kalo digunakan ya standar aplikasi

	Fadhilla	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

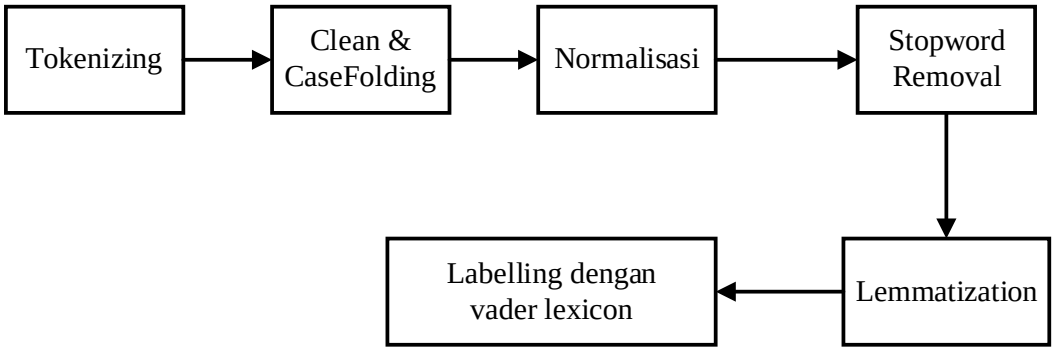
Dataset yang digunakan dalam penelitian ini adalah data ulasan aplikasi bareksa, data tersebut diperoleh dari ulasan pengguna yang telah tertera di *google play store*. Pengambilan data ulasan menggunakan metode *scraping* dengan bantuan *API google play scaper* yang merupakan salah satu *library python* 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 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 puntuk proses selanjutnya. Berikut adalah alur dari tahap *text preprocessing*.



Gambar 3. 2 Alur text preprocessing

Pada gambar 3.2 dapat dilihat alur dari *text preprocessing*.

a. Pertama proses *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.
f. proses terakhir yakni proses *labelling* yaitu untk memberikan label sentiment positif atau negatif.

3.4.1. Tokenizing

Tokenizing merupakan sebuah proses pemisahan teks menjadi potongan token baik menjadi potongan kata, huruf maupun kalimat sebelum di proses tahap selanjutnya. Data yang telah dilakukan proses *cleaning & casefolding* dan normalisasi setelah itu dipisahkan menjadi token. Proses *tokenizing* dilihatkan seperti gambar berikut.

Pada proses *tokenizing* dimulai dari pembacaan data yang telah dilakukan normalisasi lalu dilanjutkan ke proses *tokenizing* yang mememcahkan kalimat dalam berbentuk perkata atau berbentuk token berdasarkan spasi. Dalam proses ini menggunakan NLTK. Berikut hasil dari proses *tokenizing*.

Tabel 3. 3 *Tokenizing*

Ulasan	**Cleaning CaseFolding**
Aplikasinya ringan, fiturnya lengkap untuk reksadana cocok untuk pemula yang mau investasi, tampilannya bagus	Aplikasinya,ringan,,fiturnya,lengkap, untuk,reksadana,cocok,untuk,pemula, yang,mau,investasi,,tampilannya,bagus
bareksa aplikasi investasi yang bagus, ringan dan jelas informasinya fiturnya lengkap dan mudah dimengerti	bareksa,aplikasi,investasi,yang, bagus,,ringan,dan,jelas, informasinya,fiturnya, lengkap,dan,mudah,dimengerti
fitur aplikasinya kurang lengkap di saham tampilannya biasa, tapi aplikasinya lumayan ringan	fitur,aplikasinya,kurang,lengkap,di,saham ,tampilannya,biasa,,tapi, aplikasinya,lumayan,ringan

3.4.2. Tokenizing

Pada proses ini peneliti menggabungkan 2 proses yakni proses *cleaning* dan proses *casefolding*. Pada proses cleaning digunakan untuk menghapus simbol, tanda baca, *emoticon* agar mengurangi noise saat proses klasifikasi. Dalam proses cleaning menggunakan *library regular expretion*. Sedangkan proses *casefolding*

digunakan untuk merubah dokumen menjadi format tertentu. Proses ini memanfaatkan fitur NLTK. Alur dari proses case folding yakni pembacaan data lalu masuk dalam proses *casefolding*. Dalam penelitian ini peneliti menggunakan format *lowercase* yang merubah semua format dataset tersebut kedalam huruf kecil.

Tabel 3. 4. Clean & Case Folding

Ulasan	Clean & Case Folding
Aplikasinya,ringan,,fiturnya,lengkap, untuk,reksadana,cocok,untuk,pemula, yang,mau,investasi,,tampilannya,bagus	aplikasinya ringan fiturnya lengkap untuk reksadana cocok untuk pemula yang mau investasi tampilannya bagus
bareksa,aplikasi,investasi,yang, bagus,,ringan,dan,jelas, informasinya,fiturnya, lengkap,dan,mudah,dimengerti	bareksa aplikasi investasi yang bagus ringan dan jelas informasinya fiturnya lengkap dan mudah dimengerti
fitur,aplikasinya,kurang,lengkap,di,saham ,tampilannya,biasa,,tapi, aplikasinya,lumayan,ringan	fitur aplikasinya kurang lengkap di saham tampilannya biasa tapi aplikasinya lumayan ringan

3.4.3. Normalisasi

Dalam tahap normalisasi berguna untuk memperbaiki kata-kata yang kurang dikenal seperti singkatan (yg, kpd, app, kl dan lain sebagainya), memperbaiki kalimat yang salah dalam pengejaan atau *typo* memisahkan kata yang tersambung.

Alur normalisasi pada penelitian ini menggunakan normalisasi manual, peneliti memasukkan variable yang berisikan kata yang disingkat, kata yang *typo* dengan menggunakan *regular expresion*. Tahap normalisasi penting dalam analisis sentimen karena dengan menggunakan normalisasi dapat membantu meningkatkan akurasi maka dari itu peneliti menambahkan tahap ini kedalam *text preprocessing*. Berikut hasil dari proses normalisasi yang telah dilakukan.

Tabel 3. 5. Tabel Normalisasi

Ulasan	Normalisasi
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	mau nambahin rekening ditolak terus dengan alasan ktp tidak jelas padahal ktp jelas dan bisa terbaca jadi uang yang sudah masuk ke bareksa gabisa diambil dong kalau gini ini baru nyoba uang dikit saja gini susahnya apalagi kalau sudah masukin uang banyak bisa ilang tuh duit
performa aplikasi dn tampilan sngat bgus krn ringan fitur juga bagus terutama fitur robo advisornya cocok pemula seperti saya untuk investast	performa aplikasi dn tampilan sangat bagus karena ringan fitur juga bagus terutama fitur robo advisornya cocok pemula seperti saya untuk investasi

3.4.4. Stopword Removal

Stopword removal merupakan proses yang berguna untuk menghapus kata yang dianggap tidak penting atau tidak memiliki informasi yang mempengaruhi sentimen kata tersebut seperti “dan”, “di”, “yang” dan lain sebagainya. Dalam tahap ini peneliti menggunakan library sastrawi dan membuat variable baru yang berfungsi untuk menambahkan kata tambahan untuk di hilangkan.

Dalam proses ini data yang telah dilakukan beberapa proses sebelumnya daya yang telah berbentuk token dibaca terlebih dahulu lalu dimasukkan kedalam suatu kondisi yakni jika kata tersebut merupakan bagian dari kata(stoplist) maka kata tersebut akan dihapus sebaliknya jika bukan bagian dari kata(stoplist) maka data tersebut akan disimpan dan ditampilkan. Berikut hasil dari proses stopword removal.

Tabel 3. 6. Stopword Removal

Ulasan	Removal
bareksa aplikasi investasi yang bagus ringan dan jelas informasinya fiturnya lengkap dan mudah dimengerti	bareksa aplikasi investasi bagus ringan jelas informasinya fiturnya lengkap mudah dimengerti
aplikasinya ringan fiturnya lengkap untuk reksadana cocok untuk pemula yang mau investasi tampilannya bagus	aplikasinya ringan fiturnya lengkap reksadana cocok pemula mau investasi tampilannya bagus

3.4.5. Lemmatization

Pada tahap *lemmatization* merupakan proses untuk mengubah kata-kata ke bentuk kata dasarnya. Dataset yang telah diproses *tokenizing* dan *stopword removal* akan di inputkan kedalam proses *stemming*. Dalam proses ini peneliti menggunakan *library sastrawi* yang disediakan oleh bahasa pemograman python.

pada proses lemmatization ini data yang telah dilakukan proses sebelumnya normalisasi dan *stopword removal* akan di inputkan kedalam suatu kondisi yakni apakah data tersebut adalah bentuk kata dasar atau tidak jika kata tersebut bukan merupakan kata dasar atau ada imbuhan maka pesan tersebut diubah menjadi kata dasarnya dan disimpan dengan. Proses *stemming* ini berguna untuk memudahkan data saat di proses di tahap klasifikasi dan bisa menambah akurasi. Berikut hasil dari pemrosesan stemming.

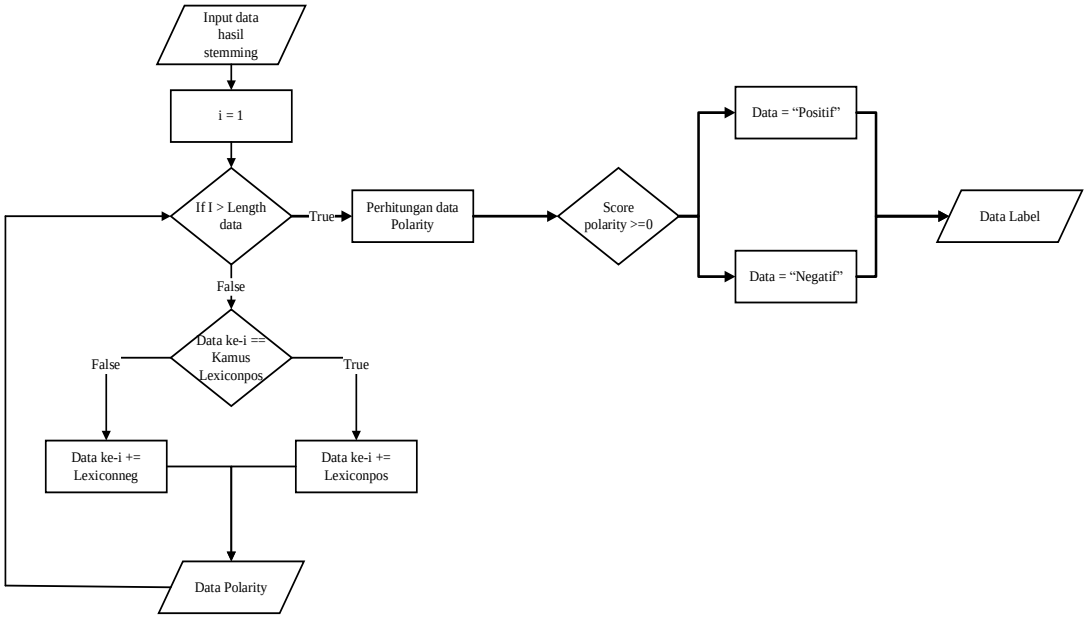
Tabel 3. 7. *lemmatization*

Ulasan	Lemmatization
mau nambahin rekening ditolak terus alasan ktp jelas padahal ktp jelas terbaca jadi uang masuk bareksa gabisa diambil dong kalau gini baru nyoba uang dikit gini susahnya kalau sudah masukin uang banyak bisa ilang tuh duit	mau nambahin rekening tolak terus alas ktp jelas padahal ktp jelas baca jadi uang masuk bareksa gabisa ambil dong kalau gin baru nyoba uang dikit gin susah kalau sudah masukin uang banyak bisa ilang tuh duit

performa aplikasi tampilan sangat bagus ringan fitur bagus terutama fitur robo advisornya cocok pemula saya investasi	performa aplikasi tampil sangat bagus ringan fitur bagus utama fitur robo advisornya cocok mula saya investasi

3.4.6. 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

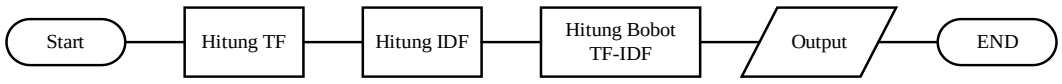
Dalam gambar tersebut dapat dilihat bahwasanya data yang telah dilakukan proses *stemming* akan dimasukkan kedalam sebuah kondisi apakah data tersebut ada pada kamus lexicon yang positif atau negatif, lalu setelah itu diproses dan hasil prosesnya yakni dikumpulkan menjadi satu sehingga menghasilkan dataset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

proses ini menghasilkan label sentiment positif sebanyak 740 dan label sentiment negatif sebanyak 267. Berikut hasil dari labelling.

Tabel 3. 8. Hasil Labelling

Ulasan	Compound Socre	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 sihgak bibit integrasi stockbitjadi gak usah foto ulang e ktpketika hasil buram jadi gabisa	-2	Negatif
tampil gajelas kalau mode gelap metode bayar cuma ovo sama danamon jadi ribet	-13	Negatif

3.5. Ekstraksi Fitur TF-IDF



Gambar 3. 4 Ektraksi 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. 9. 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 advisornya 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. 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 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
advisornya		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

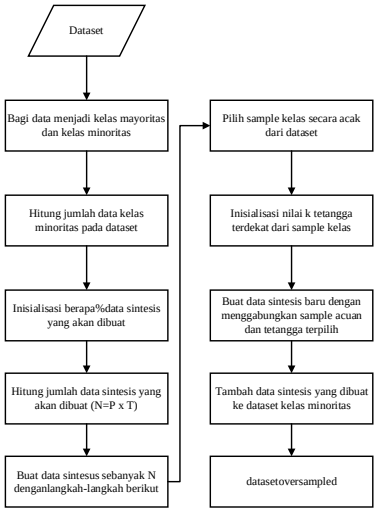
Gambar 3. 5 TF-IDF

3.6. Pembagian data

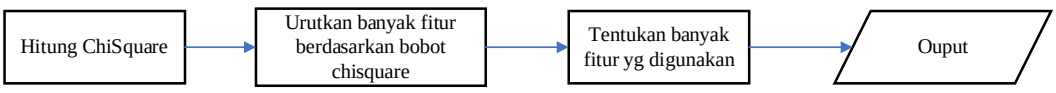
Pada tahap ini dilakukan pembagian data *training* dan data *testing*. Teknik yang digunakan untuk membagi data disini menggunakan teknik *train test split*. Teknik ini membagi data menjadi dua bagian yaitu data *traing* dan data *test* sesuai presentase yang ditentukan. Pada penelitian ini menggunakan 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].

3.7. Metode Smote belom

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



3.8. Seleksi fitur *Chi-Square*



Gambar 3. 6 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(aplikasi, positif) = $\frac{(5 \times (3 \times 1 - 0 \times 1)^2}{(3+0)(1+1)(3+1)(0+1)} = 1{,}875$

X^2(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 3.10.

Tabel 3. 10. Seleksi Fitur Chi Square

Fitur	Kelas == Positif				Kelas == Negatif				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

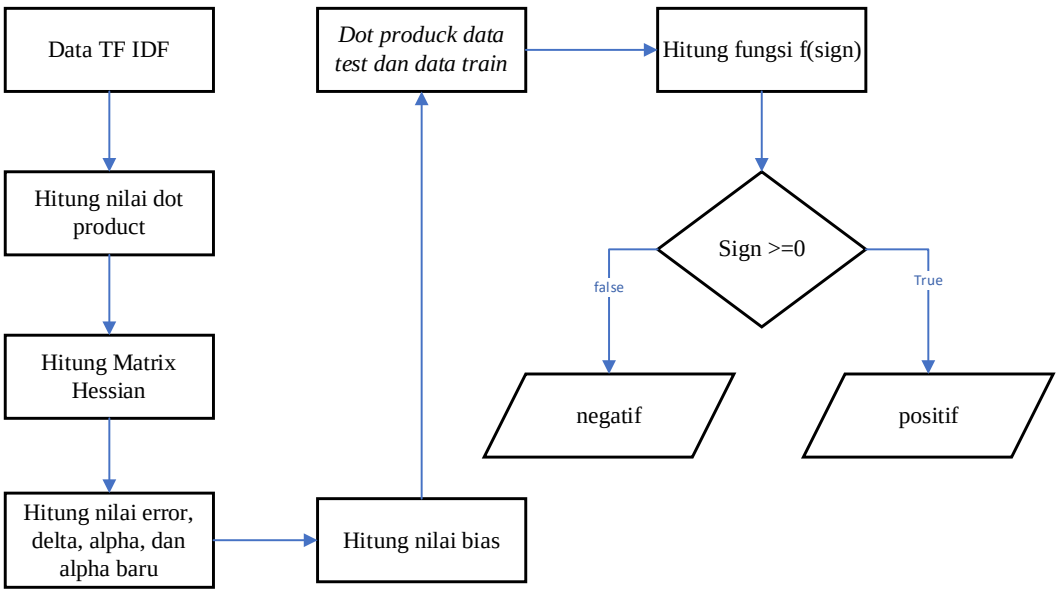
Kemudian semua fitur telah diketahui nilai bobot chi square-nya, maka semua fitur akan di urutkan nilai bobot chi square yang tertinggi ke rendah.

Tabel 3. 11. 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.

3.9. Proses Klasifikasi SVM



Gambar 3. 7 Seleksi Fitur Chi-Square

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 α = 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. 12. 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. 13. 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 :

   E1 = (0,5*1,498592) + (0,5*0,970208) + (0,5*0,476356) + (0,5*0,123392) + (0,5*0,240608) = 1,654578

Tabel 3. 14. 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 =MIN(MAX(0,5*(1-1,654578);-0,5);1-0,5) = -0,32729

Tabel 3. 15. 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. 16. 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* dikelas positif dan negatif yang tertinggi :

   $w x^{+}$ = (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

   $w x^{-}$ = (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. 17. 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. 18. 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) = Sign(SUM(0,047828+0,056561+0,006177+0+0,000495+0+0+0 = 1

    f(x) = 1 >= 0 (Positif)

3.10. Evaluasi

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

3.11. Skenario 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.

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 pemograman *python*. Dalam melakukan crawling hingga proses evaluasi pada penelitian ini menggunakan *library* yang dibutuhkan seperti yang diuraikan pada tabel 4.1.

No.	*Library*	Kegunaan
1.	*Google play scaper*	Digunakan dalam melakukan proses *crawling* 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 peneyimbangan 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
```

```
    start_idx = 0
    while len(all_reviews) < num_reviews:
        try:
            batch_reviews, _ = reviews(
                app_id,
                lang='id',
                country='id',
                sort=Sort.MOST_RELEVANT,
                count=min(batch_size, num_reviews - len(all_reviews)),
                filter_score_with=None if len(all_reviews) == 0 else 3,)
            all_reviews.extend(batch_reviews)
            start_idx += batch_size
            time.sleep(1)
        except Exception as e:
            print(f"Error occurred: {e}")
            break
    return all_reviews[:num_reviews]
df_busu = pd.DataFrame(np.array(app_reviews),columns=['review'])
df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))
df_busu
data = df_busu
df = pd.DataFrame(data)
file_path = '/content/drive/MyDrive/Skripsi/Program Skripsi/datasetbaru.csv'
df.to_csv(file_path, index=False)
```

Penjelasan dari program pada kode 4.1. sebagai berikut :

1. Pada baris 1-3 berfungsi sebagai untuk memanggil library yang dibutuhkan yakni google play scaper 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 29-32 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-13e593766f33	Maulina Nila Rahma	https://play-lh.googleusercontent.com/a-/ALV-U...	Aplikasinya ringan, fiturnya lengkap untuk rek...	2	17	4.0.1	19/11/2023 2:12:44	Terima kasih telah memberikan feedback terhada...	2023-11-23 1:59:26	4.0.1
1	aeb81eb7-5004-4a27-8ded-07e593866e33	Rachman Fadhilla	https://play-lh.googleusercontent.com/a-/ALV-U...	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-lh.googleusercontent.com/a-/ALV-U...	bareksa aplikasi investasi yang bagus, ringan ...	4	5	4.0.1	2023-11-14 6:06:12	Terima kasih telah memberikan feedback terhada...	2023-11-16 3:50:04	4.0.1
3	ec3d76d6-3acf-4463-885a-b0b69e135307	Rachmawati Ariningsih	https://play-lh.googleusercontent.com/a/ACg8oc...	Mw nambahin rekening ditolak 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-d9a0af0c6b51	Muhammad Nur	https://play-lh.googleusercontent.com/a-/ALV-U...	Performa Aplikasi dan tampilan sangat bagus ka...	5	0	4.0.1	2023-11-16 8:53:10	Terima kasih telah memberikan feedback terhada...	2023-11-18 3:18:06	4.0.1

Dari data ulasan yang didapatkan, hanya data ulasan dengan atribut content yang menggunakan bahasa Indonesia yang akan digunaan 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* melaui 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
   StopWordRemoverFactory
   from nltk.corpus import stopwords
```

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

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

Penjelasan dari potongan program kode 4.6. sebagai berikut :

1. Pada baris 1 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.

```
1  Import re
2  def clean_text(df, text_field, new_text_field_name):
3      df[new_text_field_name] = df[text_field].apply(lambda x: '
4  '.join(x)).str.lower()
5      df[new_text_field_name] = df[new_text_field_name].apply(lambda elem:
6  re.sub(r"(@[A-Za-z0-9_]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|^rt|http.+?", "",
7  elem))
8      df[new_text_field_name] = df[new_text_field_name].apply(lambda elem:
9  re.sub(r"\d+", "", elem))
10     df[new_text_field_name] = df[new_text_field_name].apply(lambda elem:
11 separate_punctuation(elem))
12     df[new_text_field_name] = df[new_text_field_name].apply(lambda x: x.split())
```

```
13      return df
```

Penjelasan pada potongan kode 4.4 sebagai berikut :

1. Pada baris pertama mengimport library re (regular expression)
2. Pada baris 2-13 fungsi clean_text yang berisikan beberapa fungsi yang diperlukan untuk *clean & casefolding*
3. pada baris 3-4 berfungsi sebagai merubah format seluruh huruf menjadi *lowercase*.
4. pada baris 5-7 berfungsi sebagai menghapus tanda baca, mention, emoji dan karakter khusus lainnya.
5. pada baris 8-9 berfungsi sebagai menghapus angka.
6. pada baris 10-11 berrfungsi sebagai memisahkan tanda baca yang terhubung dengan kata.
7. lalu baris 12 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.

```
1   data_kamus = pd.read_csv('/content/drive/MyDrive/Skripsi/Program Skripsi/Kamus
2   Lexicon/kamus_normalization.csv')
3   def normalisasi(teks, kamus):
4       kalimat_final = []
5       for kata in teks:
6           kata_benar = kamus[kamus['Tidak Baku'] == kata]['Baku'].values
7           if len(kata_benar) > 0:
8               kalimat_final.append(kata_benar[0])
9           else:
10              kalimat_final.append(kata)
11      return kalimat_final
12  df['content_norm'] = df['text_clean'].apply(lambda x: normalisasi(x,data_kamus))
13  df
```

Penjelasan pada potongan kode 4.5. sebagai berikut :

1. Pada baris pertama berfungsi sebagai Membaca kamus bahasa dari file CSV.
2. Pada baris 3-13 berfungsi sebagai fungsi normalisasi yang berguna untuk mengubah kata typo atau salah penulisan maka akan terdeteksi dan diubah.
3. pada baris 14-15 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

```
1  more_stopwords = {
2      'dar', 'hai', 'txffzhybv', 'bg', 'bot', 'yg', 'deh', 'ypdhl', 'tidak', 'nic',
3  'bos', 'hmmm', 'ky', 'yaa', 'mo', 'fb', 'laah', 'br', 'blg', 'da', 'x', 'jt', 'dan',
4  'y', 'b', 't', 'yang', 'sj', 'faq', 'jsajan', 'aja', 'mis', 'mf', 'hmm', 'jii',
5  'issi', 'the', 'kok', 'ng', 'di', 'nih', 'lah', 'adm', 'nig', 'min', 'y', 'kak',
6  'k', 'va','dong', 'ai', 'nya', 'e', 'tuh', 'nih', 'di' , 'min','ke', 'dgn', 'nya',
7  'jadi', 'ada', 'nya', 'ah', 'aamiin'}
8  stop_words_factory = StopWordRemoverFactory()
9  stop_words = stop_words_factory.get_stop_words()
10 stop_words = stop_words.extend(more_stopwords)
11 stopword_remover = stop_words_factory.create_stop_word_remover()
12 def remove_stopwords(text):
13     if isinstance(text, list):
14         text = ' '.join(text)
15     return stopword_remover.remove(text)
16 #proses stopword
17 df['text_stopword'] = df['content_norm'].apply(remove_stopwords)
18 df
19
```

Penjelasan pada potongan kode 4.6. sebagai berikut :

1. Pada baris 1-7 merupakan Daftar kata-kata stop words tambahan
2. Pada baris 8-11 berfungsi sebagai Membuat daftar kata-kata stop words
3. Pada baris 12-15 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

```
1  from spacy.tokens import Token
2  from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
3  factory = StemmerFactory()
4  stemmer = factory.create_stemmer()
5  def lemmatize_indonesian(token):
6      return stemmer.stem(token.text)
7  Token.set_extension('lemma_indonesian', getter=lemmatize_indonesian, force=True)
8  nlp = spacy.blank('id')
9  # Tambahkan pipeline tokenizer spaCy
10 def custom_tokenizer(nlp):
11     return spacy.tokenizer.Tokenizer(nlp.vocab)
12 nlp.tokenizer = custom_tokenizer(nlp)
13 data = df['text_stopword']
14 df = pd.DataFrame(data)
15 df['hasil_lemma'] = df['text_stopword'].apply(lambda x: '
16 '.join([token._.lemma_indonesian for token in nlp(x)]))
17 # Tampilkan dataframe hasil
18 Df
```

Penjelasan pada potongan kode 4.7. sebagai berikut :

1. Pada baris 1-4 Instalisasi dan inisialisasi library spacy dan sastrawi untuk lemmatisasi
2. Pada baris 5-6 fungsi untuk lemmatisasi
3. pada baris ke 7 melakukan penambahan ekstensi spacy dan mengatur model bahasa 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.

```
1  pip install vaderSentiment
2  import pandas as pd
3  from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
4  lexiconpos = pd.read_csv('/content/drive/MyDrive/Skripsi/Program Skripsi/Kamus
5  Lexicon/positifbersih.csv')
6  lexiconneg = pd.read_csv('/content/drive/MyDrive/Skripsi/Program Skripsi/Kamus
7  Lexicon/negatifbersih.csv')
8  analyzer = SentimentIntensityAnalyzer()
9  def label_sentiment_vader(text, lexiconpos, lexiconneg):
10     compound_score = analyzer.polarity_scores(text)['compound']
11     words = text.split()
12     score = 0
13     for word in words:
14         if word in lexiconpos['word'].values:
15             score += lexiconpos[lexiconpos['word'] == word]['weight'].values[0]
16         elif word in lexiconneg['word'].values:
17             score += lexiconneg[lexiconneg['word'] == word]['weight'].values[0]
18     polarity = 'Positif' if score >= 0 else 'Negatif'
19     return score, polarity
20 labels = df['text_prepro'].apply(label_sentiment_vader, lexiconpos=lexiconpos,
21 lexiconneg=lexiconneg)
22 jumlah_positif = df[df['Sentimen'] == 'Positif'].shape[0]
23 jumlah_negatif = df[df['Sentimen'] == 'Negatif'].shape[0]
24 jumlah_total = df.shape[0]
```

Penjelasan pada potongan kode 4.8. sebagai berikut :

1. Pada baris 1-3 berfungsi sebagai instalasi library vader lexicon.
2. Pada baris 4-7 berfungsi untuk membaca kamus lexicon positif dan negatif.
3. Pada baris 9-19 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 22-24 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.

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 pembobotan kata menggunakan TF-IDF.

```
1  import pandas as pd
2  from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
3  from sklearn.preprocessing import normalize
4  data = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
```

```
Skripsi/Hasil/labelling2.csv')
column = "text_prepro"
data[column] = data[column].fillna('')
count_vectorizer = CountVectorizer()
tfidf_vectorizer = TfidfVectorizer()
TF_vector = count_vectorizer.fit_transform(data[column])
normalized_tf_vector = normalize(TF_vector, norm='l1', axis=1)
tfs = tfidf_vectorizer.fit_transform(data[column])
IDF_vector = tfidf_vectorizer.idf_
tfidf_mat = normalized_tf_vector.multiply(IDF_vector).toarray()
```

Penjelasan dari potongan program pada kode 4.10. sebagai berikut :

1. Pada baris 1-3 berfungsi sebagai perintah memanggil *library* yang dibutuhkan.
2. Pada baris 4-7 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 8-13 berfungsi sebagai Membuat objek CountVectorizer dan TfidfVectorizer. Transformasi teks dengan CountVectorizer
4. Pada baris 14 berfungsi sebagai Mengalikan matriks TF yang sudah dinormalisasi dengan IDF.

4.4. Metode SMOTE belum

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. merupakan script untuk melakukan smote.

```
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
data = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
Skripsi/Hasil/labellingfix.csv')
column = "text_prepro"
y = data['Sentimen']
tf_idf = pd.read_csv('/content/drive/MyDrive/Skripsi/Program
Skripsi/Hasil/tfidffix.csv')
```

```
# Membagi data menjadi data latih (80%) dan data uji (20%)
X_train, X_test, y_train, y_test = train_test_split(tf_idf, y, test_size=0.2,
shuffle=True, random_state=42)
# Penerapan SMOTE
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
# Menyimpan hasil resampling untuk digunakan nanti
pd.DataFrame(X_train_resampled).to_csv('/content/drive/MyDrive/Skripsi/Program
Skripsi/Hasil/X_train_resampled.csv', index=False)
pd.Series(y_train_resampled).to_csv('/content/drive/MyDrive/Skripsi/Program
Skripsi/Hasil/y_train_resampled.csv', index=False)
# Menampilkan hasil SMOTE
print("Sebelum SMOTE:")
print(y_train.value_counts())
print("\nSetelah SMOTE:")
print(pd.Series(y_train_resampled).value_counts())
```

penjelasan dari potongan program pada kode 4.10 sebagai berikut:

1. Pada baris 1-2 pemabnggilan library yang dibutuhkan

4.5. 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 4.11 merupakan contoh *script* untuk melakukan Chi Square.

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

Penjelasan dari potongan program pada kode 4.10. sebagai berikut :

1. Pada baris 1-5 berfungsi sebagai perintah memanggil *library* yang dibutuhkan.

2. Pada baris 6 Daftar persentase fitur yang akan dievaluasi
3. Pada baris 7-13 memiliki fungsi yakni Menentukan jumlah fitur yang akan dipilih, lalu Seleksi fitur menggunakan SelectKBest untuk memilih seleksi fitur yang terbaik.

4.6. 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 4.11 merupakan contoh *script* untuk melakukan Chi Square.

```
1  from sklearn.svm import SVC
2  from sklearn.model_selection import train_test_split, cross_val_score,
3  cross_val_predict
4  from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
5  precision_score, recall_score, f1_score
6  for persen_fitur in persen_fitur_list:
7      num_feature_to_select = int(persen_fitur * tf_idf.shape[1] / 100)
8      selector = SelectKBest(f_classif, k=num_feature_to_select)
9      X_selected = selector.fit_transform(tf_idf, y)
10     X_train, X_test, y_train, y_test = train_test_split(X_selected, y,
11 test_size=0.2, shuffle=True)
12     modelsvm = SVC(kernel='linear', gamma=0.01, C=1)
```

Penjelasan dari potongan program pada kode 4.10. sebagai berikut :

1. Pada baris 1-5 berfungsi sebagai perintah memanggil *library* yang dibutuhkan.
2. Pada baris 6-9 berfungsi sebagai seleksi fitur chi square.
3. Pada baris 10-12 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.

4.7. 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 4.11 merupakan script untuk melakukan evaluasi model menggunakan confusion matrix

```
1  from sklearn.metrics import
2  classification_report,confusion_matrix,ConfusionMatrixDisplay, accuracy_score,
3  precision_score, recall_score, f1_score
4  import matplotlib.pyplot as plt
5  conf_matrix = confusion_matrix(y_train, y_train_pred)
6  class_label = ["negative", "positive"]
7  df_cm = pd.DataFrame(conf_matrix, index=class_label, columns=class_label)
8  accuracy = accuracy_score(y_train, y_train_pred)
9  precision = precision_score(y_train, y_train_pred, average='weighted')
10 recall = recall_score(y_train, y_train_pred, average='weighted')
11 f1 = f1_score(y_train, y_train_pred, average='weighted')
12 print(f"Accuracy: {accuracy}")
13 print(f"Precision: {precision}")
14 print(f"Recall: {recall}")
15 print(f"F1 Score: {f1}")
16 plt.figure(figsize=(10, 7))
17 sns.heatmap(df_cm, annot=True, fmt='d')
18 plt.title(f"Confusion Matrix - Cross-Validation (k={k})")
19 plt.xlabel("Predicted Label")
20 plt.ylabel("True Label")
21 plt.show()
22
```

Berikut penjelasan dari potongan code 4.11 sebagai berikut :

1. Baris 1-4 merupakan pemanggilan *library* yang akan dignakan.
2. baris 5 merupakan sebagai fungsi untuk melakukan confusion matrix
3. baris 6-7 untuk menyimpan hasil confusion matrix dengan dataframe
4. baris 8-15 untuk menampilkan hasil *confusion matrix*
5. Baris 16-22 yakni menampilkan visualisasi pada *confusion matrix.*

4.8. Hasil Skenario Uji Coba 1

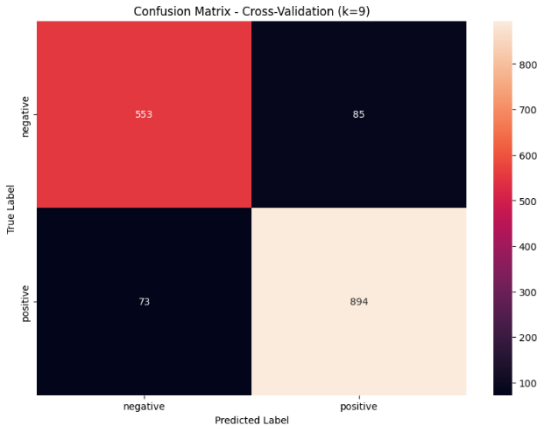
Skenario uji coba pertama ialah pengujian nilai akurasi nilai model *support vector machine* (SVM) tanpa menggunakan seleksi fitur chi square dan metode smote. Pada skenario uji cob ini melakukan dengan menggunakan parameter c=1,

gamma==0,01 serta menggunakan k-fold validation dengan menggunakan nilai k = 5, 7, 9. Berikut hasil skenario uji coba 1 :

Skenario Uji coba 1	Parameter C	Parameter gamma	K-fold	Akurasi	presisi	Recall
Support Vector Machine tanpa seleksi fitur, tanpa smote	1	0,01	5	89,28%	89,25%	89,28%
			7	90,15%	90,12%	90,15%
			9	90,15%	90,13%	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 dengan menggunakan confusion matrix sebagai berikut.

(a) (b)



(c)

Dapat dilihat dari gambar 4. (a) bahwa hasil yang diprediksi benar dengan menggunakan nilai K = 5 sebanyak 811 diprediksi *True* positif (TP), 463 diprediksi *True* negatif (TN), dan 285 *False* positif (FP) dan 46 *false* negatif dengan akurasi 89,28%. Lalu pada ggambar 4. (b) menggunakan K=7 menghasilkan 807 *True* positif (TP), 485 *True* negatif (TN), 263 *False* Positif (FP), dan 50 *False* Negatif (FN) menghasilkan akurasi sebesar 90,15%. Sedangkan jika menggunakan nilai K=9 menghasilkan 809 *True* positif (TP), 485 *True* negatif (TN), 263 *False* positif dan 40 *False* negatif (FN) dengan akurasi sebesar 90,15%.

4.9. Hasil Skenario Uji Coba 2

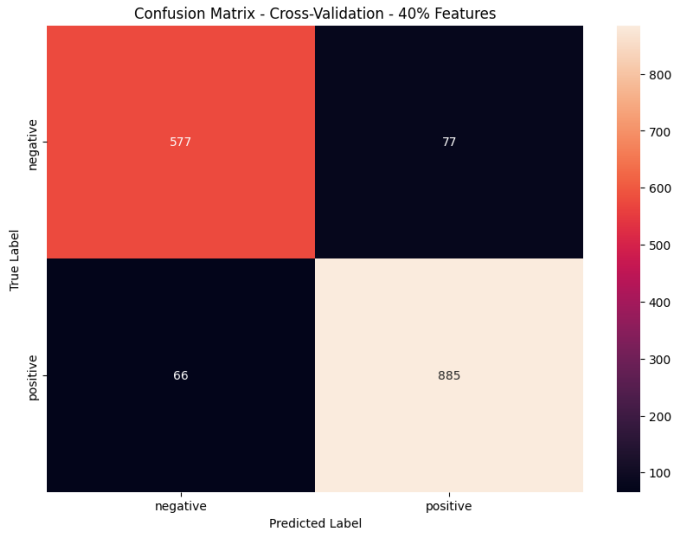
Skenario uji coba 2 ialah pengujian nilai akurasi nilai model *support vector machine* (SVM) menggunakan seleksi fitur chisquare. Pada skenario uji cob ini melakukan dengan menggunakan parameter c=1, gamma==0,01 serta menggunakan k-fold validation dengan menggunakan nilai k = 5, 7, 9, serta menggunakan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

No.	Fitur & SVM	Nilai K-Fold	Akurasi	Presisi	Recall
1.	5%	5	84,17%	84,54%	84,57
		7	83,61%	84,70%	83,83%
		9	83,92%	83,12%	82,83%
2.	10%	5	86,29%	85,15%	85,07%
		7	85,17%	87,54%	87,56%
		9	85,66%	88,40%	88,05%
3.	15%	5	87,35%	85,62%	85,57%
		7	87,72%	88,51%	88,55%
		9	87,04	89,30%	89,30%
4.	20%	5	88,09%	86,79%	86,81%
		7	88,22%	88,04%	88,05%
		9	87,91%	92,35%	92,28%
5.	25%	5	88,66%	90,53%	90,54%
		7	89,96%	89,36%	89,30%
		9	90,21%	88,83%	88,80%
6.	30%	5	89,40%	92,29%	92,28%
		7	90,46%	90,31%	90,29%
		9	90,52%	91,56%	91,54%
		5	89,53%	88,34%	88,30%

7.	35%	7	89,28%	93,03%	93,03%
		9	89,03%	91,27%	91,29%
8.	40%	5	89,84%	88,99%	89,05%
		7	89,65%	90,41%	90,29
		9	91,09%	89,63%	89,55%
9.	45%	5	88,72%	91,56%	91,54%
		7	90,15%	89,55%	89,55%
		9	89,16%	91,53%	91,54%
10.	50%	5	89,28%	90,28%	90,29%
		7	89,78%	91,32%	91,04%
		9	90,09%	89,81%	89,80%
11.	55%	5	90,40%	90,03%	90,04%
		7	91,09%	88,74%	88,80%
		9	89,77%	88,07%	88,05
12.	60%	5	89,84%	90,51%	90,54%
		7	90,15%	89,10%	89,05%
		9	90,21%	90,53%	90,54%
13.	65%	5	89,15%	88,29%	88,30%
		7	90,03%	89,09%	89,05%
		9	89,28%	91,06%	91,04%
14.	70%	5	89,22%	90,54%	90,54%
		7	89,28%	92,53%	92,53%
		9	89,22%	92,03%	92,03%
15.	75%	5	90,40%	89,82%	89,80%
		7	90,09%	90,04%	90,04%
		9	90,40%	89,77%	89,80%
16.	80%	5	89,40%	91,54%	91,54%
		7	89,90%	91,32%	91,29%
		9	89,84%	89,76%	89,80%
		5	89,65%	89,29%	89,30%

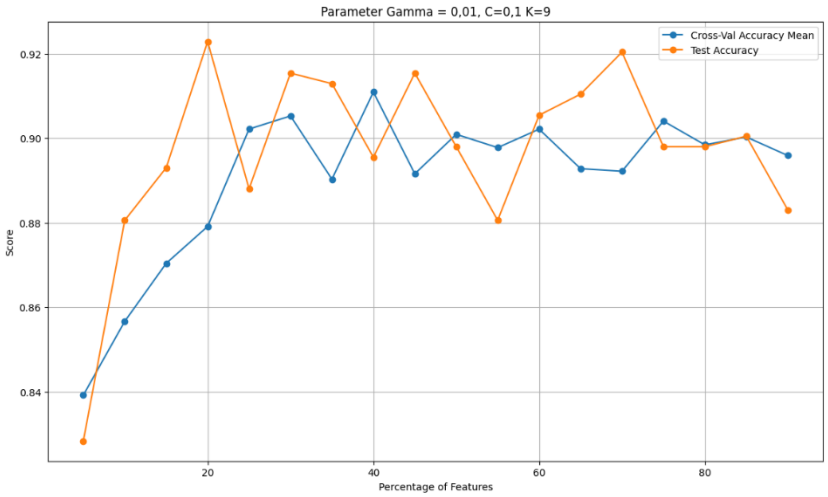
17.	85%	7	89,78%	91,42%	91,29%
		9	90,03%	90,01%	90,04%
18.	90%	5	88,97%	89,28%	89,30%
		7	89,22%	91,06%	91,04%
		9	89,59%	88,28%	88,30%

Pada Tabel 4. dapat dilihat dengan melakukan seleksi fitur dengan chi square pada algoritma *support vector machine* dengan menggunakan parameter c=1 dan gamma=0,01 serta menambahkan K-fold validation dengan nilai K=5,7,6 mampu meningkatkan performa model *support vector mahine* dengan melihat peningkatan nilai akurasi yang telah didapatkan.Dalam hasil tabel diatas hasil yang terbaik dan optimal dihasilkan dengan menggunakan seleksi fitur sebesar 40% dengan menggunakan nilai K=9 yang menghaslkan akurasi 91,09%, presision 89,63 dan recall 89,55% dengan menggunakan K=9 memiliki tingkat variansi pada data semakon rendah dan menyebabkan semakin optimal suatu model. Sedangkan untuk evaluasi *confusion matrix support vector machine* dengan seleksi fitur chi square yang paling optimal ditunjukkan pada gambar 4.



Pada gambar 4. 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%, presition 89,63% dan recall 89,55%.



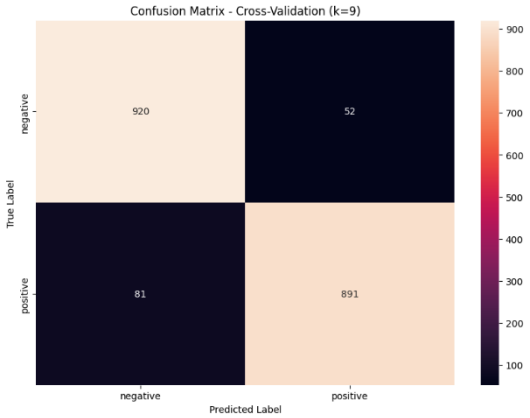
Pada gambar 4. merupakan hasil grafik yang paling optimal dengan menggunakan parameter gamma = 0,01 c=0,1 serta nilai K=9. Karena semakin tinggi nilai K maka semakin rendah untuk variansi data yang akan di proses. Dalam grafik tersebut juga tidak terdapat indikasi overfitting karena jarak gap antara cross validation accuracy dan test accuracy tidak terlalu jauh cenderung stabil.

4.10. Hasil Skenario Uji Coba 3

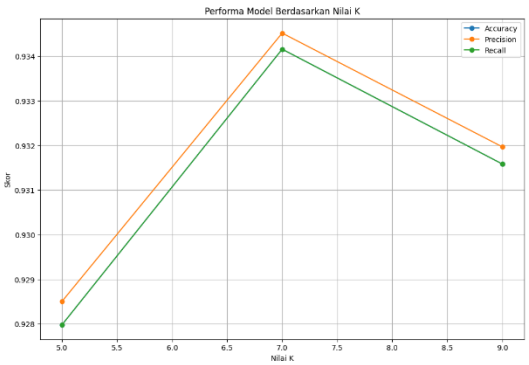
Skenario pengujian ke 3 ialah pengujian model *Support Vector Machine* dengan menggunakan dataset yang seimbang dengan melakukan teknik oversampling SMOTE. Metode smote hanya dilakukan pada data train saja dengan menggunakan nilai 5,7 dan 9 sebagai nilai k-fold validation. Berikut hasil dari pengujian ketiga ini pada tabel 4.

Skenario Uji coba 1	Parameter C	Parameter gamma	K-fold	Akurasi	presisi	Recall
Support Vector Machine + Smote	1	0,01	5	92,79%	92,85%	92,79%
			7	93,41%	93,45%	93,41%
			9	93,15%	93,19%	93,15%

Pada tabel 4. 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. 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 penggunakan nilai K=5,7,9.



Dapat dilihat dari gambar 4. terdapat penurunan yang tidak terlalu siginifikan saat *support vector machine* dengan smote menggunakan nilai K=9. Selisih sebesar 0,26% dan dapat dilihat bahwasanya pada skenario pengujian ini lebih optimal dengan menggunakan nilai k-fold sebesar 7.

4.11. Hasil Skenario Uji Coba 4

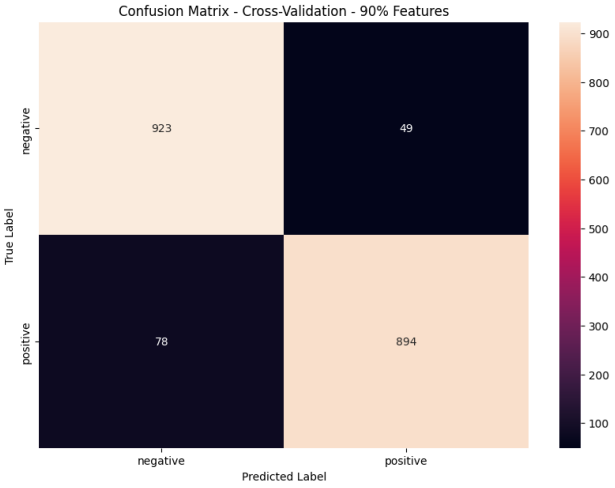
Pada skenario keempat yakni dengan mengkombinasikan seluruh skenario uji coba sebelumnya sehingga menggunakan algoritma *Suppport Vector Machine* dengan menggunakan metode smote untuk imbalance data serta seleksi fitur dengan chi square. Dengan menggunakan parameter c=1, gamma =0,01 dengan menggunakan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

No.	Fitur & SVM	K-Fold	Akurasi	Presisi	Recall
1.	5%	5	83,12%	79,33%	79,35%
		7	83,79%	79,33%	79,35%
		9	83,69	79,33%	79,35%
2.	10%	5	87,60%	81,63%	81,59%
		7	87,55%	81,63%	81,59%
		9	87,70%	81,63%	81,59%
3.	15%	5	88,89%	83,51%	83,58%

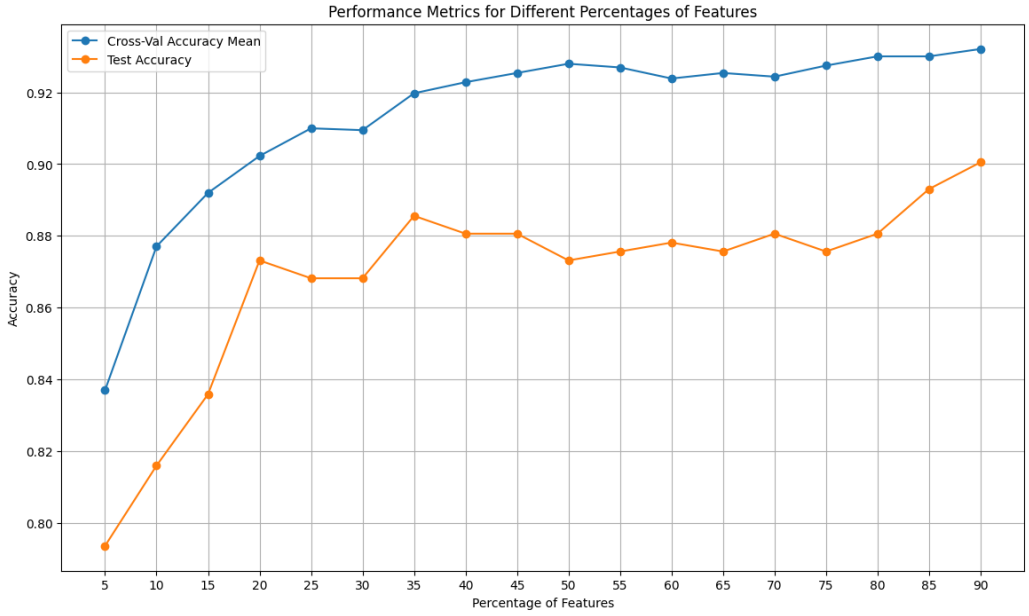
		7	89,45%	83,51%	83,58%
		9	89,19%	83,51%	83,58%
10	20%	5	90,58%	87,44%	87,31%
		7	90,79%	87,44%	87,31%
		9	90,22%	87,44%	87,31%
13.	25%	5	91%	86,90%	86,81%
		7	90,74%	86,90%	86,81%
		9	90,99%	86,90%	86,81%
16.	30%	5	90,69%	86,86%	86,81%
		7	90,94%	86,86%	86,81%
		9	90,94	86,86%	86,81%
19.	35%	5	91,61%	88,65%	88,55%
		7	91,56%	88,65%	88,55%
		9	91,97%	88,65%	88,55%
22.	40%	5	92,02%	88,21%	88,05%
		7	92,02%	88,21%	88,05%
		9	92,28%	88,21%	88,05%
25.	45%	5	92,13%	88,05%	88,05%
		7	92,54%	88,05%	88,05%
		9	92,54%	88,05%	88,05%
28.	50%	5	92,02%	87,35%	87,31%
		7	92,59	87,35%	87,31%
		9	92,79%	87,35%	87,31%
31.	55%	5	92,18%	87,62%	87,56%
		7	92,80%	87,62%	87,56%
		9	92,69%	87,62%	87,56%
34.	60%	5	92,44%	87,85%	87,81%
		7	92,95%	87,85%	87,81%
		9	92,38%	87,85%	87,81%

37.	65%	5	92,59%	87,58%	87,56%
		7	93%	87,58%	87,56%
		9	92,54%	87,58%	87,56%
40.	70%	5	93%	88,12%	88,05%
		7	92,64%	88,12%	88,05%
		9	92,43%	88,12%	88,05%
43.	75%	5	93%	87,62%	87,56%
		7	93%	87,62%	87,56%
		9	92,74%	87,62%	87,56%
46.	80%	5	92,54%	88,12%	88,05%
		7	92,80%	88,12%	88,05%
		9	93%	88,12%	88,05%
49.	85%	5	93,10%	89,42%	89,30%
		7	92,90%	89,42%	89,30%
		9	93%	89,42%	89,30%
52	90%	5	92,74%	90,14%	90,04%
		7	93,46%	89,42%	89,30%
		9	93,20%	89,42%	89,30%

Pada tabel 4. diatas merupakan hasil pengujian model dengan menggunakan algoritma *support vector machine* dengan seleksi fitur chi square serta smote untuk *imbalance* data. Dapat dilihat bahwasanya dengan digabungkan fitur chi square dan smote model mengalami suatu peningkatan serta ditambahnya nilai k-fold dari 5,7,9. Dapat dilihat dari masing-masing K-fold telah mencapai 93% yang merupakan hasil terbaik. Haisl yang optimal dan mencapai diatas 90% yakni dimulai menggunakan seleksi fitur sebesar 15% . berikut merupakan hasil evaluasi *confusion matrix* yang paling optimal yakni menggunakan 90% dengan menggunakan nilai k-fold 7.



Dalam gambar 4. 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-fold = 7.



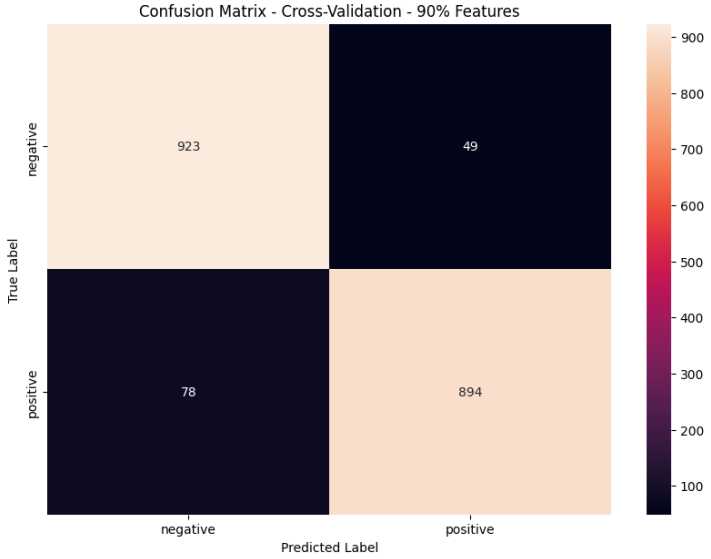
Pada gambar 4. merupakan sebuah grafik hasil dari pengujian dengan parameter c=1, gamma=0,01 dengan menggunakan k-fold=7. Dapat dilihat bahwasanya Ketika menggunakan seleksi fitur dan metode SMOTE hasilnya terdapat peningkatan optimal dalam akurasinya. Lalu dengan menggunakan nilai k-fold = 7 memiliki peningkatan yang optimal pada saat penggunaan presentase fitur sekitar 30% berdasarkan rata-rata cross validation sementara pada penggunaan 70% mengalami peningkatan yang signiffikan.

4.12. Analisa Skenario Uji coba belum .

Berdasarkan dari implementasi yang telah dilakukan, dengan membandingkan literatur yang ada dengan melakukan analisis sentiment 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-fold bisa mempengaruhi suatu performa model. Pada pengujian menghasilkan performa terbaik dengan menggunakan nilai k-fold = 9 dengan akurasi sebesar 90,15%, presisi 90,13 dan recall 90,15%. Hasil uji coba menunjukkan bahwasanya menambahkan nilai k-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 SMOT 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 meggunakan k-fold =7. Berikut merupakan hasil dari evaluasi *confusion matrix* pada gambar 4.



berdasarkan gambar 4. 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.

4.13. Implementasi Antarmuka

Bab V

PENUTUP

5.1. Kesimpulan

Berdasarkan dari implementasi hasil penelitian yang telah dilakukan dengan menggunakan algoritma *support vector machine* menggunakan seleksi fitur chi square serta smote untuk *imbalance* data dalam melakukan analisis sentiment ulasan pada aplikasi bareksa mampu meningkatkan performa kinerja modelnya. Hasil kesimpulan pada penelitian ini sebagai berikut :

a. Penggunaan algoritma *support vector machine* dalam memproses data berbentuk teks tergolong optimal tetapi terdapat beberapa yang harus diperhatikan yakni menambahkan parameter c dan gamma serta menggunakan k-fold validation karena untuk mengurangi indikasi overfitting. Dalam penelitian ini menggunakan parameter c=1, gamma=0,01 serta menggunakan k-fold = 7, 9 menghasilkan akurasi yang optimal yakni sebesar 90,15%
b. Penggunaan seleksi fitur dalam *support vector machine* mampu meningkatkan performa dengan bukti yakni pada beberapa skenario percobaan yang menggunakan seleksi fitur dimulai dari 20% mengalami peningkatan performa serta yang paling optimal pada penelitian ini menggunakan parameter c=1, gamma=0,01 k-fold = 9 dengan seleksi fitur 40% menghasilkan akurasi yang tertinggi sebesar 90,09%, presisi 89,63% dan 89,55%.
c. Penggunaan metode smote untuk *imbalance* data dalam penelitian ini memiliki pengaruh yang optimal dalam meningkatkan performa dibuktikan dengan saat hanya menggunakan svm tanpa smote akurasi yang tertinggi yakni 90,15% dengan k-fold = 7 dan 9. Tetapi saat menggunakan smote mengalami peningkatan sebesar 93% dengan menggunakan semua nilai K yakni 5,7 dan 9.

d. Sedangkan dalam penggunaan svm dengan smote serta seleksi fitur mengalami peningkatan performa dengan meningkatnya akurasi sebesar 93,46% dengan menggunakan parameter c=1, gamma=0,01 dan menggunakan nilai k-fold = 9.

5.2. Saran

Dari penelitian yang penulis lakukan, penulis mengetahui beberapa kekurangan yang bisa dapat diperbaiki untuk penelitian selanjutnya agar menghasilkan performa yang optimal dan tidak ada indikasi *overfitting* ataupun *underfitting*. Menambahkan dataset yang lebih banyak,menambahkan algoritma atau sistem yang berguna untuk mengatasi dan memperbaiki penggunaan kata baku karena hal tersebut juga membantu dalam mengurangi indikasi *overfitting*. Serta untuk mengurangi indikasi hal tersebut bisa menambahkan penggunaan *hyperparamter tunning* agar lebih optimal.

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 Analsis 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%0Ahttps://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.