Perbandingan Metode SVM dengan *Decision Tree* untuk Analisis Sentimen Ulasan Aplikasi Carousell

Prissely Pravasstifany Mediana[1], Yuni Yamasari[2]

[1,3] S1 Teknik Informatika, Universitas Negeri Surabaya
[1]prissely.20023@mhs.unesa.ac.id
[3]yuniyamasari@unesa.ac.id

Abstrak*— Carousell merupakan salah satu platform jual beli yang cukup populer. Hal ini ditunjukkan dengan pengguna lebih dari 10 juta, rating 4,7 dan 337 ribu ulasan di *Google Play Store*. Ulasan tersebut dapat memberikan informasi penting terkait kepuasan pengguna terhadap aplikasi. Karena jumlah ulasan yang sangat banyak, metode-metode dapat diterapkan untuk menganalisisnya. Oleh karena itu, penelitian ini memfokuskan untuk memecahkan permasalahn tersebut dengan melakukan analisis sentiment terhadap ulasan aplikasi Carousell. Penelitian ini melakukan analisis sentimen dengan menggunakan metode SVM dan *Decision tree*. Klasifikasi ulasan dikategorikan menjadi 2 yaitu: positif dan negatif. Data ulasan yang digunakan sebanyak 10.000 yang dikumpulkan melalui scraping. Proses pelabelan berdasarkan skor ulasan: skor 4 dan 5 sebagai positif, sedangkan skor 1, 2, dan 3 sebagai negatif. Metode fitur seleksi *information gain* dan dua metode pembagian data (*split data* dan *k-fold cross validation*) juga diterapkan dalam penelitian ini. Hasil penelitian memperlihatkan hasil terbaik pada algoritma SVM dengan kernel RBF yaitu dengan performa akurasi 93,1%, presisi 93,42%, recall 93,1%, dan f1-*score* 93,22% menggunakan metode pembagian data *split data* rasio 90:10 tanpa kombinasi fitur seleksi. Sedangkan, algoritma *Decision tree* menghasilkan performa terbaik dengan kombinasi fitur seleksi menggunakan metode pembagian data *split data* rasio 90:10 max_depth=10 yaitu akurasi sebesar 91%, presisi 92,39%, recall 91% dan f1-score 91,49%. Hasil-hasil ini mengindikasikan algoritma SVM memiliki performa lebih baik dibandingkan *Decision Tree.* Hal ini ditunjukkan dengan nilai perfoma dari SVM lebih tinggi dibanding dengan *Decision tree* dengan perbedaan nilai akurasi sebesar 2,1%, presisi 1,03%, recall 2,1%, f1-score 1,73%.

***Kata Kunci*— Analisis sentimen, SVM, *Decision tree, Information gain*.**

I. PENDAHULUAN

Dalam era digital layaknya di masa sekarang, smartphone menjadi sangat krusial untuk manusia pada keseharian hidup. Melalui adanya smartphone manusia bisa mengakses internet tanpa batasan waktu. Menurut laporan *We Are Social*, pada bulan januari 2023 Indonesia memiliki sebanyak 213 juta pengguna internet [1]. Jumlah tersebut mencapai 77% dari jumlah populasi manusia di Indonesia yang mencapai 276,4 juta orang pada awal tahun ini. Angka ini meningkat dalam jumlah 5,44% dari tahun sebelumnya, dimana pada januari 2022, total orang yang menggunakan internet di Indonesia adalah 202 juta orang. Kemajuan teknologi inilah yang dapat memberikan kemudahan kepada manusia untuk melakukan berbagai hal salah satunya yaitu dalam proses jual beli. Dengan adanya akses internet manusia dapat melakukan proses jual beli secara *online* atau sering kita sebut dengan transaksi *online*. Transaksi *online* di Indonesia tumbuh dengan begitu pesat, sejumlah survei melaporkan bertumbuhnya grafik *e-commerce* tertinggi di dunia. Menurut laporan PPRO Payments & E-Commerce Report in Asia 2018, Indonesia mencatatkan pertumbuhan tertinggi sebesar 78% per tahun [2]. Berbeda dengan *e-commerce* yang hanya menjual barang dari satu toko, *online marketplace* adalah sebuah platform yang menyediakan fasilitas pada proses jual beli dari berbagai toko. Salah satu *online marketplace* yang cukup populer di Indonesia adalah Carousell. Carousell merupakan sebuah platform perdagangan *online* dimana penggunanya dapat menjual dan membeli berbagai produk baru dan barang-barang bekas layak pakai mulai dari elektronik, fashion pria & wanita, peralatan rumah tangga dan masih banyak lainnya. hingga Oktober 2023, aplikasi carousell telah diunduh oleh 10 juta lebih pengguna dan diulas oleh 337 ribu pengguna serta mendapat rating sebesar 4,7.

Sebagai platform perdagangan *online* yang populer, *online* marketplace seringkali menjadi subjek beragam komentar dari pengguna. Ulasan di *Google Play Store* menjadi wadah bagi pengguna untuk memberikan penilaian, serta menyampaikan keluhan mereka terhadap pengalaman menggunakan aplikasi tersebut. Untuk menganalisis dan meneliti hal tersebut, diperlukan metode untuk mengklasifikasikan ulasan pengguna pada kelompok positif dan negatif yang menjadi komponen dari penelitian ini.

Data penelitian ini diperoleh dari ulasan yang ditinggalkan oleh pengguna terkait aplikasi Carousell di Google Play Store. Studi ini menggunakan teknik data mining untuk mengklasifikasikan sentimen dari ulasan pengguna. Data yang diperoleh kemudian diolah, hasil dari pengolahan data tersebut berupa analisis sentimen. Analisis ini merupakan teknik yang dilaksankaan guna melakukam ekstrak data opini, memahami, dan mengelola data tekstual dengan cara otomatis guna mengidentifikasi sentimen dalam sebuah opini [3]. Analisis sentimen ini bertujuan untuk mengkategorikan sentimen pengguna menjadi positif ataupun negatif.

Penelitian ini menerapkan pendekatan *Support Vector Machine* (SVM) dan *Decision Tree*. Dalam penelitian ini, penulis memilih metode *Support Vector Machine* karena dianggap sebagai salah satu metode yang unggul dalam *machine learning* dengan kinerja yang lebih baik dalam klasifikasi dan prediksi. [4]. Selain itu, penulis menggunakan metode *Decision tree* sebagai perbandingan karena metode pembelajaran mesin. *Decision Tree* adalah metode *unsupervised* berbentuk pohon yang mampu

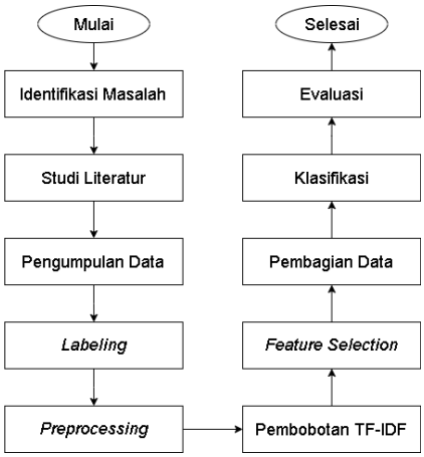
mengelompokkan data berukuran besar menjadi data berukuran kecil [5].

Beberapa penelitian terdahulu menunjukkan sejumlah keunggulan penggunaan algoritma *Support Vector Machine* (SVM) dan *Decision tree*. Penelitian sebelumnya mengindikasikan bahwasanya analisis sentimen menggunakan data ulasan dari Google Play Store menghasilkan akurasi dalam angka 90.20% untuk algoritma SVM dan 89.80% untuk algoritma Decision Tree. [6]. Studi lain mengatakan bahwa metode *SVM* mempunyai tingkat akurasi lebih tinggi jika daripada dengan metode *Decision Tree* dengan hasil klasifikasi menggunakan SVM sebesar 94,36% sedangkan *Decision Tree* sebesar 87,45% [7]. Menurut Destitus & Suryasari pada penelitiannya mengatakan bahwa hasil klasifikasi metode SVM kombinasi *Information gain* mendapatkan hasil yang lebih tinggi yakni 86% sedangkan metode SVM tanpa kombinasi *Information gain* mendapatkan hasil yang lebih rendah yakni 80% [8]. Penelitian lain mengatakan bahwa metode *SVM* berhasil mencapai akurasi sebesar 97,5% dengan menggunakan kernel linear. Penelitian ini memanfaatkan skor ulasan untuk menentukan label, di mana skor 1 hingga 3 diberi label negatif, sementara skor 4 dan 5 diberi label positif. Penelitian terdahulu menunjukkan bahwa akurasi algoritma *Support Vector Machine* lebih besar jika dibandingkan dengan akurasi algoritma *Decision tree* [9].

Tujuan penelitian ini adalah untuk membandingkan akurasi analisis sentimen antara algoritma *Support Vector Machine* dan *Decision Tree*. Peneliti menggunakan dataset dari Google Play yang diperoleh melalui teknik *scraping* ulasan pengguna. Proses pelabelan data dilakukan berdasarkan skor ulasan, dimana skor 1 hingga 3 dinilai sebagai sentimen negatif, sementara skor 4 dan 5 dinilai sebagai sentimen positif. Setelah itu, dataset ulasan akan melewati tahap *preprocessing* untuk persiapan analisis lebih lanjut.

II. METODOLOGI PENELITIAN

Jenis penelitian yang penulis terapkan yaitu menggunakan jenis penelitian kuantitatif untuk menentukan seberapa banyak ulasan positif dan negatif. Untuk mengumpulkan data, penelitian ini menggunakan teknik scraping ulasan di *Google Play Store*. Selanjutnya, data akan diklasifikasikan dengan metode *SVM* dan *Decision tree*. Kemudian, hasil pengukuran kedua metode tersebut akan dibandingkan. Berikut di bawah ini merupakan alur penelitian yang digunakan.



Gbr 1. Alur Penelitian

A. *Identifikasi Masalah*

Tahap pertama dalam penelitian adalah mengenali permasalahan terkait perbandingan antara metode SVM dan *Decision Tree* dalam menganalisis sentimen aplikasi Carousell. Tujuan penelitian ini yaitu dalam rangka menentukan metode yang paling efektif untuk melakukan analisis sentimen di platform tersebut.

B. *Studi Literatur*

Tahap kedua yaitu melakukan studi literatur, hal ini dilakukan untuk mencapai tujuan penelitian serta memecahkan masalah dalam penelitian dengan mempelajari teori-teori terkait analisis sentimen, *Support Vector Machine*, *Decision tree*. Selain itu peneliti juga menggunakan referensi tambahan seperti buku, artikel, jurnal international maupun nasional.

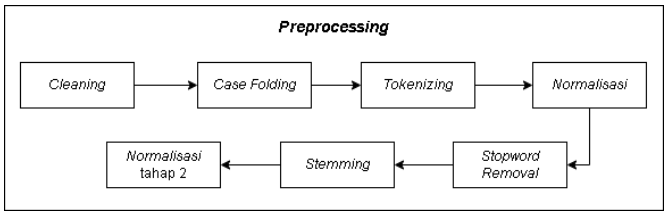
C. *Pengumpulan Data*

Tahap ketiga pada penelitian ini adalah mengumpulkan data, yang berasal dari ulasan pengguna di *Google Play* tentang aplikasi Carousell. Data dihimpun menggunakan teknik *scraping*, dengan tujuan untuk mengumpulkan sebanyak 10.000 ulasan.

D. *Labeling Data*

Tahap keempat dalam penelitian ini adalah melakukan pelabelan data. Ulasan *Google Play* yang telah dikumpulkan lalu diberikan label menjadi dua kategori, yakni positif dan negatif. Proses pelabelan dilakukan berdasarkan skor ulasan yang diberikan oleh pengguna. Skor 1 hingga 3 akan diberi label negatif, sementara skor 4 dan 5 akan diberi label positif.

E. *Preprocessing*



Gbr 2. Tahapan *Preprocessing*

Gambar 2. Menujukkan alur preprocessing dengan mencakup*cleaning*, *case folding*, *tokenizing*, *normalisasi*, *stopword removal*, *stemming*. Pada proses ini, data yang sudah dikumpulkan akan dikelola guna memperoleh data yang relevan serta menyingkirkan data yang tidak dibutuhkan dalam analisis. Berikut adalah teknik-teknik pengelolaan data dalam tahap *preprocessing*:

- *Data Cleaning*, tahap penting yang melibatkan proses menghilangkan elemen-elemen yang tidak relevan dari teks, seperti tag delimiter, kode HTML, angka, simbol, koma, titik, spasi berlebihan, dan kata-kata yang terdiri dari satu huruf. Proses ini menggunakan pustaka re dan string pada bahasa pemrograman Python. Tujuan dari pembersihan data adalah untuk mengurangi *noise* atau gangguan yang mungkin ada dalam dokumen, sehingga meningkatkan kualitas analisis yang dilakukan.
- *Case Folding*, tahap ini olah teks yang bertujuan untuk melakukan pengubahan seluruh huruf menjadi huruf kecil (*lowecase*).
- *Tokenizing*, pada tahap ini dokumen dipotong menjadi bagian-bagian kata atau karakter yang sesuai dengan kebutuhan sistem.
- *Normalisasi*, dalam tahap ini dilaksankaan proses perubahan kata-kata singkatan atau kata tidak baku sebagai bentuk kata yang baku atau asli. Tujuan dari proses ini adalah untuk memudahkan pengolahan data dengan menggunakan kata-kata yang baku.
- *Stopword Removal*, pada tahap ini kata-kata umum yang biasanya tidak berkontribusi secara signifikan pada proses klasifikasi, seperti kata penghubung, kata depan, dan kata sambung, akan dihapus dari teks yang sedang diproses.
- *Stemming* dilakukan dengan menggunakan library Sastrawi, yang dirancang khusus untuk bahasa Indonesia. Pada tahap *stemming*, semua imbuhan yang melekat pada kata akan dihapus, sehingga kata tersebut menjadi bentuk dasarnya.
- *Normalisasi* tahap kedua dilakukan untuk menyempurnakan kata-kata yang dianggap penting yang telah melalui proses stemming agar memiliki makna yang sama seperti sebelum dilakukan proses stemming.

F. Pembobotan TF-IDF

TF (*Term Frequency*) adalah intensitas suatu kata muncul pada sebuah dokumen. [10]. Sedangkan IDF (*Inverse Document Frequency*) memperhitungkan luas sebuah term tersebar dalam koleksi dokumen. TF-IDF dipergunakan dalam menghitung frekuensi kemunculan kata pada dokumen serta mengubahnya menjadi bentuk numerik.[11]. Semakin sering suatu term muncul dalam dokumen (TF tinggi), semakin besar pula bobotnya atau nilai kesesuaiannya. [4]. Adanya proses TF-idf ini karena algoritma klasifikasi hanya dapat memproses data numerik.

$$w_{td} = tf_{td} \times (\log\left(\frac{N}{dft}\right) + 1) \quad (1)$$

G. Feature Selection

Tahapan penyeleksian fitur tujuannya adalah guna menurunkan jumlah *term* yang diperoleh dari *preprocessing* untuk dipegunakan dalam analisis sentimen. *Information gain* berguna untuk mengurangi dimensi data dan meningkatkan akurasi klasifikasi. Selain itu, *information gain* menunjukkan keselarasan kata ataupun fitur dengan standarisasi yang telah ditentukan. [12]. Pada tahap ini, pemilihan fitur (kata) dilakukan dengan metode *Information Gain*. *Information Gain* mengukur seberapa banyak informasi yang diperoleh dari kehadiran atau ketidakhadiran sebuah kata dalam membantu keputusan klasifikasi yang benar dalam setiap kelas. Information Gain berperan sebagai alat untuk menyeleksi fitur paling esensial untuk meningkatkan akurasi sistem. Pada tahap ini, dilakukan penghapusan *stopword* (*stopword removal*) dan *stemming* untuk kata yang berimbuhan. [13].

Dengan memanfaatkan nilai *Information Gain*, setiap kata diberi peringkat sehingga dapat dihasilkan fitur terbaik. Bobot ini dipergunakan alam mengklasifikasikan data uji, di mana fitur yang memiliki nilai tertinggi akan dipilih. Ini memungkinkan untuk memprioritaskan kata-kata yang paling informatif dan relevan dalam proses klasifikasi.

H. Pembagian Data

Pada tahap ini terdapat 2 metode pembagian data yang digunakan yaitu *split data* dan *k-fold cross validation* . Tujuan dari penggunaan kedua metode pembagian data tersebut yakni untuk mengetahui metode manakah yang lebih efektif untuk digunakan dalam penelitian ini dan juga untuk membandingkan hasil akhir terbaik dari kedua metode tersebut.

- *Split Data*

  Split Data, juga dikenal sebagai pemisahan data, adalah langkah penting dalam proses pengembangan model. Secara umum, pemisahan data dilaksanakan melalui pembagian data dalam dua bagian, di mana salah satunya digunakan untuk menguji atau mengevaluasi data, sementara yang lainnya untuk melatih model. Pemisahan data bertujuan untuk menghasilkan dua atau lebih subset data yang berbeda untuk keperluan pelatihan dan pengujian model [14]. Pada studi ini data yang telah didapatkan akan terbagi dalam dua jenis yakni *data training* dan *data testing*.
- *K-Fold Cross Validation*

  K-Fold Cross Validation adalah salah satu teknik penting dalam evaluasi kinerja model. Prosesnya melibatkan pembagian data ke dalam k kelompok yang

seimbang. Setiap kelompok secara bergantian berperan menjadi data uji, sementara golongan yang tersisa dipergunakan sebagai data latih. Ini memungkinkan evaluasi model yang lebih reliabel karena setiap sampel data digunakan untuk pengujian dan pelatihan, sehingga mengurangi risiko overfitting atau kesalahan evaluasi yang disebabkan oleh penggunaan satu set data tertentu. [15].

I. *Klasifikasi*

Dalam tahap pemodelan penelitian ini, digunakan dua metode yaitu *Support Vector Machine* (SVM) dan *Decision Tree* :

- Klasifikasi menggunakan metode *Support Vector Machine* (SVM), tujuan utamanya yaitu mencari hyperplane terbaik menjadi pemisah dua kelas, yaitu ulasan positif dan ulasan negatif, dari data ulasan. SVM menerapkan algoritma khusus untuk mencari batas keputusan yang optimal. Setelah hyperplane ditemukan, model yang diperoduksi mampu dipergunakan dalam melakukan prediksi terhadap data uji, sehingga memungkinkan untuk mengklasifikasikan ulasan baru ke dalam salah satu dari dua kelas yang telah ditentukan.
- Klasifikasi dengan metode *Decision tree*, tujuan utamanya adalah menemukan batas keputusan yang terbaik dalam menjadi pemisah ulasan menjadi dua kelas: ulasan positif serta negatif. *Decision Tree* menggunakan algoritma spesifik untuk menentukan batas keputusan yang optimal. Setelah batas keputusan terbentuk, model yang diperoleh bsia dipergunakan dalam melakukan prediksi terhadap data uji, memungkinkan pengklasifikasian ulasan baru ke dalam salah satu dari dua kelas yang telah ditetapkan.

J. *Evaluasi*

Dalam tahap evaluasi ini diambil berdasarkan hasil *confussion matrix* yang dapat menghitung *performance matrix*. *Performance matrix* dalam mengukur nilai *accuracy, precision, recall,* dan *f1-score* dengan tujuan menguji performa model.

- *Accuracy*

  Accuracy yaitu perbandingan antara prediksi nilai yang benar (baik nilai positif maupun nilai negatif) dengan keseluruhan data. [16].

- *Precission*

  Precission yaitu perbandingan antara prediksi yang tepat positif dengan total hasil yang diprediksikan positif. [16].

- *Recall*

  Recall yaitu perbandingan antara prediksi yang tepat positif dengan total data yang benar positif. [16].

- *F1-Score*

  F1-score yaitu hasil penghitungan yang menggabungkan nilai precision dan recall, yang lalu digunakan menjadi ukuran evaluasi. [16].

III. HASIL DAN PEMBAHASAN

Pada bagian ini, dilakukan pembahasan hasil dari olah data yang sudah dilaksanakan menurut metode penelitian yang sudah dijelaskan dalam bagian sebelum ini.

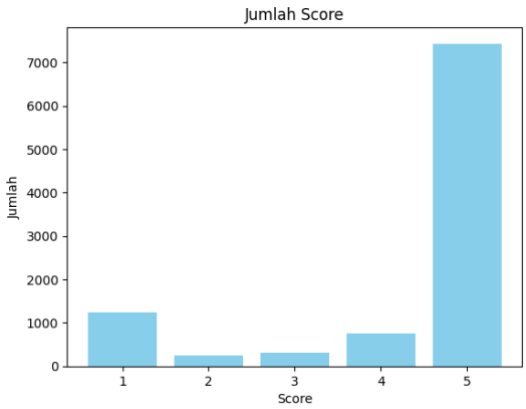
A. *Deskripsi Data*

Proses mengumpulkan data ulasan dari *Google Play Store* dilakukan dengan melalui teknik *scraping* yang dijalankan melalui pustaka Python bernama 'google-play-scraper'. Fokus pengambilan data adalah pada ulasan yang diberikan oleh pengguna untuk aplikasi Carousell. Dengan menggunakan teknik *scraping* ini, berhasil diperoleh sebanyak 10.000 ulasan dari berbagai pengguna.

Setelah data ulasan berhasil diambil, langkah selanjutnya adalah menghasilkan dataset yang terstruktur dalam format Excel. Di dalam dataset terdapat 2 kolom yaitu *content* dan *score* masing-masing kolom tersebut berisi:

- Kolom *content*, berisi teks ulasan pengguna aplikasi carousell. Data pada kolom *content* memiliki tipe data string mencakup berbagai ekspresi, mulai dari ungkapan singkat seperti ‘Good!’ hingga komentar panjang yang memperingatkan potensi penipuan dalam aplikasi.
- Kolom *score*, merupakan representasi numerik dari sentimen atau kualitas yang terkait dengan konten, yang dinyatakan dalam skala 1 sampai 5. Kolom *score* memiliki tipe data int64 yang berisi nilai-nilai integer.

Berdasarkan data yang telah diambi jumlah masing-masing *score* yang diperoleh yaitu 1.242 untuk *score* 1, 249 untuk *score* 2, 311 untuk *score* 3, 764 untuk *score* 4, 7.434 untuk *score* 5. Kemudian jumlah *score* tersebut divisualisasikan menjadi diagram batang seperti pada gambar 3.



Gbr 3. Diagram Jumlah score

Sampel data yang telah diperoleh dicantumkan pada tabel 1 di bawah ini :

TABEL I
SAMPEL DATA ULASAN

Content	Score
Hati hati masih banyak modus penipuan menggunakan aplikasi aplikasi yang belum jelas saya mengalami sendiri semoga tidak terulang ke orang lain	1
Keren dan mudah .sangat membantu dan responsif....terhubung dengan luar kota juga dan tidak banyak masalah. Ini pengalaman pertama saya pakai aplikasi ini. Sdmoga aplikasi bisa bertahan dan terus berkembang	5
Sebagai seller sy rekomendasi kan aplikasi Carousell, cara berjualan simple, transaksi mudah, namun kita juga harus berhati-hati karena Carousell tidak menggunakan rekening bersama, penipuan dapat terjadi baik pada pembeli maupun penjual, sukses selalu tambah maju utk Carousell dan kita semua.	4
Sistem pencariannya gak maksimal. Dannn.. Makin banyak akun penipu di area motor dan mobil.. Harus dibersihkan nih min.. Saya naikkan bintang jika sudah minim akun penipunya..	2
Hati hati di sini banyak penjual penipunya	3

B. Labeling Data

Data ulasan *Google Play* aplikasi carousell yang telah didapatkan melalui teknik *scraping* lalu diberikan label dalam 2 kelas yakni positif serta negatif. Pengelompokkan label tersebut berdasarkan atas *score* yang diberikan pihak yang menggunakan. *Score* 4 dan 5 termasuk dalam label 'Positif', *score* 1, 2 dan 3 termasuk dalam label 'Negatif'.

Dari hasil pelabelan yang telah dilakukan didapatkan 8.189 data dan label positif dan 1.491 data dengan label negatif. Sample hasil dari proses pelabelan ulasan ditunjukkan dalam table berikut:

TABEL II
HASIL PELABELAN

Content	*Score*	Label
apk ga banget dengan pembeli yg php , sekedar nanya² , maupun yg mau nipu pun ada ga banget. penjualan susah.	1	Negatif
sangat membantu sekali dalam berniaga	5	Positif
Mantap buyer bersahabat	4	Positif
Tidak bisa membalas pesan	2	Negatif
Hati hati di sini banyak penjual penipunya	3	Negatif

C. Preprocessing

Pada tahap preprocessing, data mengalami serangkaian proses mencakup *cleaning*, *case folding*, *tokenizing*, *normalisasi*, *stopword removal* serta *stemming*. Tujuannya adalah mempersiapkan data dengan mengolahnya untuk mempertahankan informasi yang relevan dan menghilangkan yang tidak dibutuhkan pada analisis berikutnya. Hasil dari proses *preprocessing* ini yaitu data yang sudah siap dipergunakan pada analisis lebih lanjut. Berikut hasil dari tahap *preprocessing* :

1) Data cleaning

TABEL III
HASIL PROSES CLEANING

Sebelum	Sesudah
diperbaiki lagi !!, cuman login aja masih masalah, dari sisi pengguna, udah sulit buat login aja, batasan akun per device juga mempersulit, not even sure server nya mendeteksi batasan akun dengan benar atau tidak wkwk, kalo udah sulit di pengalaman pengguna gitu ya malas mau make, apalagi pas login aja udah sulit	diperbaiki lagi cuman login aja masih masalah dari sisi pengguna udah sulit buat login aja batasan akun per device juga mempersulit not even sure server nya mendeteksi batasan akun dengan benar atau tidak wkwk kalo udah sulit di pengalaman pengguna gitu ya malas mau make apalagi pas login aja udah sulit

2) Case folding

TABEL IV
HASIL PROSES CASE FOLDING

Sebelum	Sesudah
APK GA BANGET DENGAN PEMBELI YG PHP , SEKEDAR NANYAÂ² , MAUPUN YG MAU NIPU PUN ADA GA BANGET. PENJUALAN SUSAH . TRANSAKSI RUMIT MENDING OREN MAKASI	apk ga banget dengan pembeli yg php sekedar nanya² maupun yg mau nipu pun ada ga banget penjualan susah transaksi rumit mending oren makasi

3) Tokenizing

TABEL V
HASIL PROSES TOKENIZING

Sebelum	Sesudah
Aplikasi bagus untuk mulai usaha namun aplikasinya kurang karena masih suka ngelag barang ga bisa di upload harus di coba berkali kali bikin ga mood katanya bermasalah trs	['aplikasi', 'bagus', 'untuk', 'mulai', 'usaha', 'namun', 'aplikasinya', 'kurang', 'karena', 'masih', 'suka', 'ngelag', 'barang', 'ga', 'bisa', 'di', 'upload', 'harus', 'di', 'coba', 'berkali', 'kali', 'bikin', 'ga', 'mood', 'katanya', 'bermasalah', 'trs']

4) Normalisasi

TABEL VI
HASIL PROSES NORMALISASI

Sebelum	Sesudah
Mau login kena pembatasan akun pdhl akunnya ya Cuma pakai itu agak aneh deh	['mau', 'login', 'terkena', 'pembatasan', 'akun', 'padahal', 'akunnya', 'ya', 'cuma', 'pakai', 'itu', 'agak', 'aneh', 'deh']

5) Stopword Removal

TABEL VII
HASIL PROSES STOPWORD REMOVAL

Sebelum	Sesudah
Perlu ditingkatkan terkait keamanan bagi buyer, sehingga terhindar dari penipuan.	[ditingkatkan, terkait, kemanan, buyer, terhidar, penipuan]

6) Stemming

TABEL VIII
HASIL PROSES STEMMING

Sebelum	Sesudah
Aplikasi yang simple dan sangat membantu menjual brg second dgn kondisi masih bagus :)	aplikasi simple bantu jual brg second dgn kondisi bagus

7) Normalisasi tahap 2

TABEL IX
HASIL PROSES NORMALISASI TAHAP 2

Sebelum	Sesudah
Alam salesman	Pengalaman salesman

D. Pembobotan TF-IDF

Sebelum dilakukan pembangunan model klasifikasi, dilakukan proses pembobotan fitur pada setiap *term* atau kata menggunakan TF-IDF (*Term Frequency – Inverse Document Frequency*). pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) dilaksanakan melalui perhitungan setiap kata (*term*) yang terdapat pada dokumen dari setiap kata. Pada proses TF-IDF disini meamnfaatkan library TfidfTransformer dan TfidfVectorizer. Tahapan ini merupakan salah satu tahap yang krusial karena model *machine learning* yang tersedia di pustaka 'sklearn' hanya mampu memproses data yang berupa angka.

Parameter yang digunakan untuk TfidfVectorizer adalah max_features=350. Artinya, matriks TF-IDF yang dihasilkan hanya akan melibatkan 350 kata yang paling sering muncul di seluruh dokumen. Hal ini membantu dalam mengurangi jumlah data yang diolah dan fokus pada kata-kata yang paling relevan untuk analisis. Berikut hasil dari pembobottan TF-IDF :

```
(1, 3)        1.0
(2, 209)      0.49029869067157017
(2, 144)      0.18808411668235087
(2, 229)      0.3815146450355839
(2, 43)       0.24082762625927023
(2, 25)       0.6157902191013092
(2, 11)       0.3761417220814821
(3, 115)      1.0
(4, 244)      0.3489228257164905
(4, 213)      0.3138800463048692
(4, 13)       0.39140067838695336
(4, 324)      0.22576343033942134
(4, 212)      0.38338423929842547
(4, 124)      0.6546641014771241
(5, 85)       0.3194087309870295
(5, 9)        0.3555203194766777
(5, 249)      0.29915565650888143
(5, 254)      0.3857912920312242
(5, 2)        0.4337486174319711
(5, 33)       0.3810240039581687
(5, 156)      0.29118314519637056
(5, 190)      0.3393369393643488
(6, 28)       1.0
(7, 155)      0.46858508649999114
(7, 248)      0.3448665267853739
```

Gbr 4. Hasil Proses TF-IDF

E. Feature Selection

Setelah melalui proses pembobotan fitur kemudian dilakukan proses seleksi fitur menggunakan *Information Gain*. *Information gain* adalah suatu teknik seleksi fitur yang dipergunakan dalam menghilangkah fitur yang kurang relevan atau menurunkan angka dimensi fitur dalam data. Berikut dibawah ini merupakan source code tahap pertama yaitu mengubah teks menjadi vector fitur menggunakan CountVectorizer dan menghitung nilai *information gain* antara fitur dan label. Dalam hal ini digunakan beberapa parameter yaitu max_df, min_df, dan max_features. Parameter min_df digunakan untuk mengabaikan beberapa kata yang timbul pada dokumen lebih dari 95%, parameter min_df dipergunakan dalam mengabaikan beberapa kata yang terdapat pada dokumen kurang dari 2 kali, parameter max_features digunakan untuk membatasi jumlah kata yang dijadikan fitur dalam pembentukan vektor fitur. Dalam hal ini, hanya 350 kata dengan frekuensi tertinggi yang akan dipilih sebagai fitur.Berikut output yang dihasilkan dari source code tersebut :

```
      Word  Information Gain
0  account          0.000335
1    admin          0.001001
2    aktif          0.000622
3     akun          0.054076
4   alasan          0.003091
```

Gbr 5. Hasil Perhitungan Nilai *Information Gain*

F. Pembagian Data

Setelah melalui tahap *labeling*, *preprocessing*, dan TF-IDF, dataset dibagi menjadi dua menggunakan dua metode: *split*

data dan *k-fold cross validation*. Proses ini penting dalam mempersiapkan data untuk pelatihan dan evaluasi model *machine learning*.

- *Split data*

  Pada tahap ini, dataset terbagi dalam dua subset yang berbeda, yakni data latih serta data uji, untuk keperluan pembangunan model. Terdapat lima skenario percobaan dalam penggunaan split data, di mana data dibagi dengan rasio 90:10, 80:20, 70:30, 60:40, dan 50:50. Dalam setiap skenario, persentase data latih dan data uji berbeda, dimana persentase data latih lebih besar daripada data uji. Ini dilaksanakan dalam rangka menguji dan melatih model pada berbagai kondisi pembagian data.

- *K-Fold Cross Validation*

  K-fold cross validation merupakan teknik yang dipergunakan dalam melakukan evaluasi kinerja model dalam pembelajaran mesin. Tujuannya adalah untuk membagi dataset menjadi k bagian yang memiliki ukuran yang setara. Pada penerapan *k-fold cross validation* terdapat 5 skenario percobaan yaitu menggunakan k=3 yang artinya data akan dibagi menjadi 3 lipatan, k=5 data akan dibagi menjadi 5 lipatan, k=7 data akan dibagi menjadi 7 lipatan, k=10 data akan dibagi menjadi 3 lipatan, k=12 data akan dibagi menjadi 3 lipatan.

G. Klasifikasi

Setelah data dipersiapkan, langkah selanjutnya adalah membangun model klasifikasi. Dalam penelitian ini, ada dua jenis model yang dikembangkan, yaitu menerapkan teknik klasifikasi *Support Vector Machine* (SVM) dan *Decision Tree*. Proses ini penting dalam mengevaluasi performa dan akurasi model terhadap data yang sudah diproses sebelum itu.

- *Support Vector Machine*

  Dalam penelitian ini, SVM diterapkan dengan memanfaatkan 4 jenis kernel yang berbeda, yakni kernel *linear*, RBF (*Radial Basis Function*), *sigmoid*, serta *polynomial*. Tujuannya adalah untuk membandingkan performa model dengan menggunakan kernel yang beragam. Hal ini dilaksanakan dalam rangka mengevaluasi mana kernel yang paling cocok dan memberikan hasil terbaik untuk data yang diberikan.

- *Decision tree*

  Dalam pengembangan model Decision Tree, digunakan kelas 'DecisionTreeClassifier' dari modul 'sklearn.tree'. Dalam penelitian ini, nilai max_depth yang digunakan berkisar antara 7 hingga 10. Tujuannya adalah untuk membandingkan hasil pemodelan Decision Tree dengan max_depth yang berbeda, sehingga dapat memilih parameter yang memberikan kinerja model terbaik.

H. Evaluasi

Setelah melewati tahap pelatihan, model yang sudah dibuat dipergunakan pada pelaksanaan prediksi data uji menggunakan metode 'predict'. Hasil prediksi ini kemudian dibandingkan dengan label sebenarnya dari data uji untuk mengevaluasi performa model. Dalam proses evaluasi klasifikasi ini, metrik-metrik penting seperti akurasi (accuracy), presisi (precision), recall, dan F1-score dihitung. Beberapa metrik ini menyediakan gambaran mengenai sebaik apa model mampu melakukan pengklasifikasian data uji ke pada kelas-kelas secara benar. Evaluasi ini membantu dalam memahami seberapa baik model dapat melakukan prediksi dan memberikan wawasan tentang kekuatan dan kelemahan model yang telah dikembangkan.

Berikut dibawah ini ditampilkan tabel hasil pengujian yang telah dilakukan.

TABEL X
HASIL PENGUJIAN SVM TANPA KOMBINASI FITUR SELEKSI

Split Data					***K-Fold Cross Validation***				
Rasio	**Akurasi**	**Presisi**	**Recall**	**F1-Score**	**Nilai k**	**Akurasi**	**Presisi**	**Recall**	**F1-Score**
Kernel RBF									
90:10	**93,1%**	**93,42%**	**93,1%**	**93,22%**	5	91,71%	91,43%	91,71%	91,5%
80:20	91,7%	91,99%	91,7%	91,82%	10	91,75%	91,48%	91,75%	91,55%
70:30	91,97%	92,48%	91,97%	92,17%	**15**	**91,79%**	**91,53%**	**91,79%**	**91,6%**
60:40	91,55%	92,18%	91,55%	91,8%	20	91,73%	91,46%	91,73%	91,53%
50:50	91,34%	92,12%	91,34%	91,63%	25	91,76%	91,49%	91,76%	91,57%
Kernel Linear									
90:10	92%	92,17%	92%	92,08%	5	91,38%	91,09%	91,38%	91,18%
80:20	91,8%	92,03%	91,8%	91,9%	10	91,54%	91,27%	91,54%	91,35%
70:30	**92%**	**92,5%**	**92%**	**92,2%**	**15**	**91,62%**	**91,36%**	**91,62%**	**91,45%**
60:40	91,68%	92,2%	91,68	91,89%	20	91,46%	91,18%	91,46%	91,26%
50:50	91,3%	92,05%	91,3%	91,58%	25	91,51%	91,24%	91,51%	91,32%
Kernel Polynomial									
90:10	**91,7%**	**93,27%**	**91,7%**	**92,21%**	5	89,7%	89,17%	89,7%	88,9%
80:20	90,1%	92,04%	90,1%	90,76%	10	89,84%	89,3%	89,84%	89,09%

70:30	90,47%	92,8%	90,47%	91,23%	15	89,91%	89,38%	89,91%	89,18%
60:40	90,03%	92,63%	90,03%	90,9%	20	89,89%	89,36%	89,89%	89,16%
50:50	89,2%	92,9%	89,2%	90,41%	**25**	**90,01%**	**89,5%**	**90,01%**	**89,31%**
Kernel Sigmoid									
90:10	89,9%	89,97%	89,9%	89,93%	**5**	**89,97%**	**89,67%**	**89,97%**	**89,78%**
80:20	90,05%	90,23%	90,05%	90,13%	10	89,74%	89,47%	89,74%	89,58%
70:30	**90,87%**	**91,32%**	**90,87%**	**91,06%**	15	89,31%	89,01%	89,31%	89,13%
60:40	90,8%	91,47%	90,8%	91,06%	20	89,53%	89,24%	89,53%	89,36%
50:50	90,12%	90,89%	90,12%	90,42%	25	89,46%	89,2%	89,46%	89,3%

Menurut hasil uji dalam tabel 10, kesimpulannya adalah algoritma SVM tanpa kombinasi fitur seleksi mencapai kinerja yang lebih baik saat menggunakan metode pembagian data *split data*. Pada rasio pembagian data 90:10 dan menggunakan kernel RBF, SVM mencapai akurasi sebesar 93,1%, presisi 93,42%, recall 93,1%, dan f1-score 93,22%. Sedangkan pada uji *k-fold cross validation* dengan k=15 dan kernel RBF, SVM mencapai akurasi tertinggi sebesar 91,79%, presisi 91,53%, recall 91,79%, dan f1-score 91,6%. Hal ini menunjukkan bahwa metode pembagian data *split data* dan penggunaan kernel RBF menghasilkan data yang lebih baik dalam hal performa SVM tanpa fitur seleksi.

TABEL XI
HASIL PENGUJIAN SVM DENGAN KOMBINASI FITUR SELEKSI (*INFORMATION GAIN*)

Split Data					***K-Fold Cross Validation***				
Rasio	**Akurasi**	**Presisi**	**Recall**	**F1-Score**	**Nilai k**	**Akurasi**	**Presisi**	**Recall**	**F1-Score**
Kernel RBF									
90:10	**91,9%**	**92,05%**	**91,9%**	**91,97%**	5	90,97%	90,75%	90,97%	90,84%
80:20	91,35%	91,4%	91,35%	91,4%	10	91,03%	90,84%	91,03%	90,92%
70:30	91,77%	91,9%	91,77%	91,84%	**15**	**91,13%**	**90,94%**	**91,13%**	**91,02%**
60:40	91,55%	91,8%	91,55%	91,66%	**20**	**91,13%**	**90,94%**	**91,13%**	**91,02%**
50:50	90,98%	91,46%	90,98%	91,17%	25	91,09%	90,9%	91,09%	90,98%
Kernel Linear									
90:10	**91,7%**	**92,26%**	**91,7%**	91,9%	5	90,47%	90%	90,47%	90,02%
80:20	90,9%	91,8%	90,9%	91,25%	10	90,62%	90,17%	90,62%	90,19%
70:30	90,9%	91,99%	90,9%	91,3%	15	90,66%	90,21%	90,66%	90,23%
60:40	90,8%	91,96%	90,8%	91,22%	**20**	**90,72%**	**90,28%**	**90,72%**	**90,3%**
50:50	89,98%	91,34%	89,98%	90,48%	25	90,65%	90,2%	90,65%	90,23%
Kernel Polynomial									
90:10	**89,2%**	**92,37%**	**89,2%**	**90,28%**	5	87,8%	86,9%	87,8%	86,51%
80:20	88,2%	91,4%	88,2%	89,3%	10	88,04%	87,22%	88,04%	86,83%
70:30	88,27%	91,98%	88,27%	89,55%	**15**	**88,05%**	**87,22%**	**88,05%**	**86,85%**
60:40	87,88%	91,77%	87,88%	89,24%	20	88,01%	87,18%	88,01%	86,79%
50:50	87,58%	91,75%	87,58%	89,03%	25	88%	87,16%	88%	86,8%
Kernel Sigmoid									
90:10	85%	85,47%	85%	85,22%	5	86%	84,95%	86%	85,25%
80:20	85,95%	86,47%	85,95%	86,19%	**10**	**86,26%**	**85,21%**	**86,26%**	**85,48%**
70:30	85,3%	85,75%	85,3%	85,53%	15	86,02%	84,97%	86,02%	85,26%
60:40	86,13%	86,63%	86,13%	86,35%	20	86,12%	85,04%	86,12%	85,32%
50:50	**86,34%**	**88,95%**	**86,34%**	**87,34%**	25	85,96%	84,93%	85,96%	85,23%

Berdasarkan data pada tabel 11, diketahui bahwa algoritma SVM dengan kombinasi fitur seleksi mencapai kinerja tertinggi saat menggunakan metode pembagian data *split data*. Pada rasio pembagian data 90:10 dan menggunakan kernel RBF, SVM mencapai akurasi dalam angka 91,9%, presisi 92,05%, recall 91,9%, dan f1-score 91,97%. Sedangkan pada uji *k-fold cross validation*, nilai tertinggi diperoleh saat k=15 serta k=20 dengan kernel RBF, yaitu akurasi 91,13%, presisi 90,94%, recall 91,13%, dan f1-score 91,02%. Hasil ini menunjukkan bahwa metode pembagian data *split data* lebih efektif digunakan daripada *k-fold cross validation* dalam hal performa algoritma SVM dengan kombinasi fitur seleksi.

TABEL XII
HASIL PENGUJIAN *DECISION TREE* TANPA KOMBINASI FITUR SELEKSI

Split Data	***K-Fold Cross Validation***

Rasio	Akurasi	Presisi	Recall	F1-Score	Nilai k	Akurasi	Presisi	Recall	F1-Score
max_depth=7									
90:10	**90,1%**	**91,42%**	**90,1%**	**90,59%**	5	89,11%	88,55%	89,11%	88,69%
80:20	89,35%	90,43%	89,35%	89,77%	10	89,46%	88,94%	89,46%	89,07%
70:30	89,77%	90,73%	89,77%	90,14%	15	89,62%	89,11%	89,62%	89,23%
60:40	89,43%	90,54%	89,43%	89,86%	20	89,53%	89,00%	89,53%	89,12%
50:50	88,94%	90,12%	88,94%	89,4%	**25**	**89,63%**	**89,11%**	**89,63%**	**89,22%**
max_depth=8									
90:10	**90,2%**	**91,56%**	**90,2%**	**90,7%**	5	89,13%	88,62%	89,13%	88,77%
80:20	89,25%	90,08%	89,25%	89,6%	**10**	**89,64%**	**89,12%**	**89,64%**	**89,23%**
70:30	90,33%	91,07%	90,33%	90,43%	15	89,50%	88,97%	89,50%	89,09%
60:40	89,48%	90,7%	89,48%	89,95%	20	89,59%	89,05%	89,59%	89,16%
50:50	89%	90,34%	89%	89,5%	25	89,55%	89,01%	89,55%	89,13%
max_depth=9									
90:10	**90,2%**	**91,56%**	**90,2%**	**90,69%**	5	89,34%	88,85%	89,34%	88,99%
80:20	89,55%	90,5%	89,55%	89,9%	10	89,57%	89,04%	89,57%	89,15%
70:30	89,9%	91%	89,9%	90,3%	15	89,48%	88,96%	89,48%	89,08%
60:40	89,45%	90,67%	89,45%	89,9%	**20**	**89,63%**	**89,09%**	**89,63%**	**89,19%**
50:50	89,3%	90,45%	89,3%	89,74%	25	89,62%	89,09%	89,62%	89,21%
max_depth=10									
90:10	**90,6%**	**91,9%**	**90,6%**	**91,07%**	5	89,31%	88,81%	89,31%	88,96%
80:20	89,4%	90,3%	89,4%	89,76%	10	89,79%	89,28%	89,79%	89,38%
70:30	89,87%	90,99%	89,87%	90,3%	**15**	**89,82%**	**89,31%**	**89,82%**	**89,40%**
60:40	89,9%	91,13%	89,9%	90,36%	20	89,72%	89,19%	89,72%	89,29%
50:50	89,24%	90,68%	89,24%	89,78%	25	89,57%	89,04%	89,57%	89,15%

Berdasarkan data pada tabel 12, diketahui bahwa algoritma *decision tree* tanpa kombinasi fitur seleksi mencapai kinerja tertinggi saat menggunakan metode pembagian data *split data*. Pada rasio pembagian data 90:10 dengan nilai max_depth=10, *decision tree* mencapai akurasi sebesar 90,6%, presisi 91,9%, recall 90,6%, dan f1-score 91,07%. Sedangkan pada uji pembagian data *k-fold cross validation*, nilai tertinggi diperoleh saat k=15 dengan max_depth=10, yaitu akurasi 89,82%, presisi 89,31%, recall 89,82%, dan f1-score 89,4%. Hasil ini menunjukkan bahwa metode pembagian data *split data* lebih efektif digunakan daripada *k-fold cross validation* dalam hal performa algoritma *decision tree* tanpa kombinasi fitur seleksi.

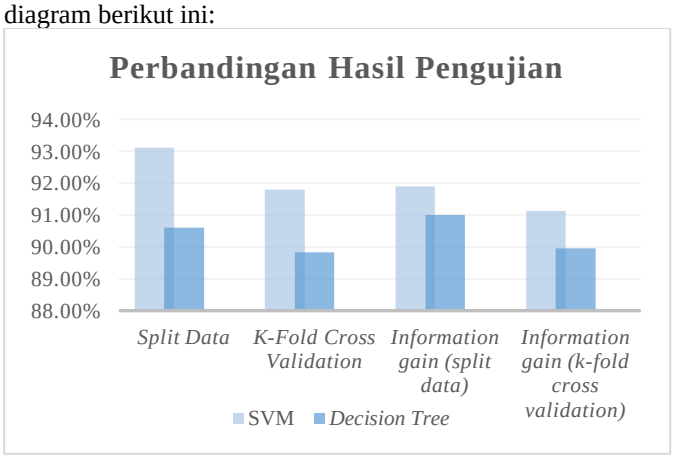
TABLE I
HASIL PENGUJIAN *DECISION TREE* DENGAN KOMBINASI FITUR SELEKSI (*INFORMATION GAIN*)

Split Data					***K-Fold Cross Validation***				
Rasio	**Akurasi**	**Presisi**	**Recall**	**F1-Score**	**Nilai k**	**Akurasi**	**Presisi**	**Recall**	**F1-Score**
max_depth=7									
90:10	**90,9%**	**92,25%**	**90,9%**	**91,38%**	5	89,55%	89,03%	89,55%	89,14%
80:20	89,95%	91,3%	89,95%	90,44%	**10**	**89,79%**	**89,27%**	**89,79%**	**89,37%**
70:30	89,77%	90,7%	89,77%	90,13%	15	89,77%	89,27%	89,77%	89,38%
60:40	89,68%	90,6%	89,68%	90,03%	20	89,70%	89,17%	89,70%	89,28%
50:50	89,16%	90,4%	89,16%	89,64%	**25**	89,71%	89,19%	89,71%	89,29%
max_depth=8									
90:10	**90,8%**	**92,3%**	**90,8%**	**91,3%**	5	89,64%	89,13%	89,64%	89,25%
80:20	89,9%	90,92%	89,9%	90,29%	**10**	**89,74%**	**89,21%**	**89,74%**	**89,31%**
70:30	90,07%	91,11%	90,07%	90,46%	15	89,66%	89,15%	89,66%	89,26%
60:40	89,73%	90,62%	89,73%	90,08%	20	89,66%	89,13%	89,66%	89,24%
50:50	89,32%	90,52%	89,32%	89,78%	25	89,64%	89,11%	89,64%	89,22%
max_depth=9									
90:10	**90,8%**	**92,3%**	**90,8%**	**91,3%**	5	89,66%	89,14%	89,66%	89,25%
80:20	90%	91,28%	90%	90,47%	10	89,72%	89,21%	89,21%	89,32%
70:30	90,3%	91,09%	90,3%	90,61%	15	89,71%	89,20%	89,71%	89,31%
60:40	90,13%	90,88%	90,13%	90,43%	20	89,77%	89,27%	89,77%	89,38%

50:50	89,56%	90,68%	89,56%	89,89%	**25**	**89,82%**	**89,31%**	**89,82%**	**89,40%**
max_depth=10									
90:10	**91%**	**92,39%**	**91%**	**91,49%**	5	89,69%	89,19%	89,69%	89,32%
80:20	90,1%	90,9%	90,1%	90,41%	10	89,92%	89,44%	89,92%	89,54%
70:30	90,37%	91,05%	90,37%	90,64%	**15**	**89,95%**	**89,48%**	**89,95%**	**89,59%**
60:40	90,15%	90,97%	90,15%	90,47%	20	89,83%	89,34%	89,83%	89,46%
50:50	89,6%	90,69%	89,6%	90,01%	25	89,71%	89,19%	89,71%	89,30%

Berdasarkan table 13, dapat diketahui bahwa algoritma *decision tree* dengan kombinasi fitur seleksi mencapai kinerja tertinggi saat menggunakan metode pembagian data *split data*. Pada rasio pembagian data 90:10 dengan nilai max_depth=10, decision tree mencapai akurasi dalam presentase 91%, presisi 92,39%, recall 91%, dan f1-score 91,49%. Sedangkan pada uji pembagian data *k-fold cross validation*, nilai tertinggi diperoleh saat k=15 dengan max_depth=10, yaitu akurasi 89,95%, presisi 89,48%, recall 89,95%, dan f1-score 89,59%. Hal ini menunjukkan bahwa metode pembagian data *split data* lebih efektif digunakan daripada *k-fold cross validation* dalam hal performa algoritma decision tree dengan kombinasi fitur seleksi.

Perbandingan hasil pengujian tersebut ditampilkan dalam diagram berikut ini:



Gbr 6. Diagram Perbandingan Hasil Pengujian

Berdasarkan gambar 6, dapat disimpulkan bahwa secara keseluruhan, algoritma SVM menunjukkan kinerja yang lebih baik daripada dengan algoritma *Decision Tree*. Hasil pengujian tertinggi pada penelitian ini dicapai oleh algoritma SVM tanpa menggunakan fitur seleksi, dengan menerapkan metode pembagian data split data dengan rasio 90:10 dan menggunakan kernel RBF.

I. Visualisasi Data

Visualisasi hasil analisis yang telah diperoleh ditampilkan dalam bentuk *wordcloud*, yang memperlihatkan beberapa kata yang paling acapkali muncul pada dataset komentar positif. Ini membantu dalam memahami tren atau tema yang dominan dalam komentar yang dianggap positif. Berikut hasil *wordcloud* komentar positif :

Gbr 7. Wordcloud Positif

Berdasarkan hasil visualisasi kata positif tersebut dapat diketahui bahwa banyak pengguna aplikasi Carousell yang mengakui bahwa aplikasi Carousell merupakan aplikasi bagus yang membantu dalam proses jual beli. Hal tersebut ditunjukkan melalui kata-kata seperti “bagus”, “membantu”, ‘’jual”, ‘’beli” dalam *wordcloud*.

Berikut hasil *wordcloud* komentar positif :

Gbr 8. Wordcloud Negatif

Berdasarkan hasil visualisasi kata negatif tersebut dapat diketahui bahwa banyak pengguna aplikasi Carousell yang menganggap bahwa aplikasi Carousell merupakan aplikasi yang banyak terdapat penipuan di dalamnya. Hal tersebut ditunjukkan melalui kata-kata seperti “penipu”, “penipuan”, ”ketipu”, “tipu” dalam *wordcloud*.

IV. KESIMPULAN

Berdasarkan hasil pengujian, metode SVM dan *Decision Tree* dapat diterapkan dengan performa yang baik terhadap ulasan pengguna aplikasi Carousell pada *Google Play*. Pengujian algoritma SVM tanpa fitur seleksi menghasilkan akurasi tertinggi yaitu sebesar 93,1% dengan menggunakan kernel RBF pada pembagian data 90:10. Sedangkan, SVM dengan kombinasi fitur seleksi *Information Gain* akurasinya sedikit menurun menjadi 91,9%. Selanjutnya, pada pengujian

metode *Decision Tree* tanpa fitur seleksi mencapai akurasi 90,6% dengan nilai max_depth = 10 pada pembagian data 90:10, sedangkan dengan kombinasi fitur seleksi (*Information Gain*) mengalami peningkatan menjadi 91% pada penggunaan rasio 90:10 max_depth=10. Kombinasi fitur seleksi *Information Gain* tidak mempengaruhi akurasi SVM secara signifikan. Namun, Teknik ini dapat meningkatkan akurasi *Decision Tree*. Secara keseluruhan, algoritma SVM lebih efektif dibanding *Decision Tree*. Penelitian juga menunjukkan bahwasanya penggunaa *split data* untuk metode pembagian data mendapatkan hasil pengujian yang lebih besar daripada penggunaan *k-fold cross validation.*

V. SARAN

Mengacu pada hasil penelitian yang telah dilaksanakan, disarankan agar penelitian analisis sentimen berikutnya menggunakan dataset terkini yang berasal dari aplikasi Carousell. Selain itu, penelitian berikutnya dapat mengeksplorasi metode klasifikasi lain untuk membandingkan kinerjanya, serta mencoba berbagai teknik fitur seleksi untuk memperoleh hasil yang lebih komprehensif.

REFERENSI

[1] C. M. Annur, "Pengguna Internet di Indonesia Tembus 213 Juta Orang hingga Awal 2023," Databooks. [Online]. Available: https://databoks.katadata.co.id/Datapublish/2023/09/20/Pengguna-Internet-Di-Indonesia-Tembus-213-Juta-Orang-Hingga-Awal-2023

[2] R. N. Handayani, "Optimasi Algoritma SVM untuk Analisis Sentimen pada Ulasan Produk Tokopedia Menggunakan PSO," *Media Inform.*, vol. 2, no. 2, pp. 97–108, 2021, [Online]. Available: https://doi.org/10.37595/mediainfo.v20i2.59

[3] F. V. Sari and A. Wibowo, "Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi," *J. SIMETRIS*, vol. 10, no. 2, pp. 681–686, 2019.

[4] A. Z. Rasyida, I. D. Wijaya, and Y. Yunhasnawa, "Analisis Sentimen Kualitas Layanan Online Marketplace di Indonesia Menggunakan Metode Support Vector Machine," *Semin. Inform. Apl. Polinema 2020*, pp. 70–75, 2020.

[5] P. B. N. Setio, D. R. S. Saputro, and Bowo Winarno, "Klasifikasi Dengan Pohon Keputusan Berbasis Algoritme C4.5," *Prism. Pros. Semin. Nas. Mat.*, vol. 3, pp. 64–71, 2020.

[6] K. A. Rokhman, B. Berlilana, and P. Arsi, "Perbandingan Metode Support Vector Machine dan Decision Tree untuk Analisis Sentimen Review Komentar pada Aplikasi Transportasi Online," *JOISM J. Inf. Syst. Manag.*, vol. 2, no. 2, pp. 1–7, 2021, doi: 10.24076/joism.2021v3i1.341.

[7] M. F. Asshidiqi and K. M. Lhaksmana, "Perbandingan Metode Decision Tree dan Support Vector Machine untuk Analisis Sentimen pada Instagram Mengenai Kinerja PSSI," *e-Proceeding Eng.*, vol. 7, no. 3, pp. 9936–9948, 2020.

[8] C. Destitus, W. Wella, and S. Suryasari, "Support Vector Machine VS Information Gain: Analisis Sentimen Cyberbullying di Twitter Indonesia," *Ultim. InfoSys J. Ilmu Sist. Inf.*, vol. 11, no. 2, pp. 107–111, 2020, doi: 10.31937/si.v11i2.1740.

[9] T. Tinaliah and T. Elizabeth, "Analisis Sentimen Ulasan Aplikasi PrimaKu Menggunakan Metode Support Vector Machine," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 4, pp. 3436–3442, 2022, doi: 10.35957/jatisi.v9i4.3586.

[10] F. Rahmadayana and Y. Sibaroni, "Sentiment Analysis of Work from Home Activity using SVM with Randomized Search Optimization," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 5, pp. 936–942, 2021, doi: 10.29207/resti.v5i5.3457.

[11] A. Aziz and F. Fauziah, "Analisis Sentimen Identifikasi Opini Terhadap Produk, Layanan dan Kebijakan Perusahaan Menggunakan Algoritma TF-IDF dan SentiStrength," *J. Sains Komput. Inform.*, vol. 6, no. 1, pp. 115–125, 2022, doi: 10.30645/j-sakti.v6i1.430.

[12] A. W. Attabi, L. Muflikhah, and M. A. Fauzi, "Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode Naïve Bayes Classifier dan Information Gain," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 11, pp. 4548–4554, 2018.

[13] A. Z. Amrullah, A. S. Anas, and M. A. J. Hidayat, "Analisis Sentimen Movie Review Menggunakan Naive Bayes Classifier Dengan Seleksi Fitur Chi Square," *J. Bumigora Inf. Technol.*, vol. 2, no. 1, pp. 40–44, 2020, doi: 10.30812/bite.v2i1.804.

[14] Trivusi, "Data Splitting: Pengertian, Metode, dan Kegunaannya." Accessed: Sep. 20, 2022. [Online]. Available: https://www.trivusi.web.id/2022/08/Datasplitting.html

[15] D. Cahyanti, A. Rahmayani, and S. A. Husniar, "Analisis Performa Metode Knn pada Dataset Pasien Pengidap Kanker Payudara," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 39–43, 2020, doi: 10.33096/ijodas.v1i2.13.

[16] S. Clara, D. L. Prianto, R. Al Habsi, E. F. Lumbantobing, and N. Chamidah, "Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learning untuk Prediksi Penghasilan Pada Adult Income Dataset," *Semin. Nas. Mhs. Ilmu Komput. dan Apl.*, vol. 2, no. 1, pp. 741–747, 2021.