TESIS

DETEKSI ULASAN PALSU MENGGUNAKAN ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE

Disusun oleh:

Nama	**: Zulpan Hadi**
NIM	**: 21.51.2090**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

TESIS

DETEKSI ULASAN PALSU MENGGUNAKAN ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE

DETECTION OF FAKE REVIEWS USING RANDOM FOREST ALGORITHM AND SUPPORT VECTOR MACHINE

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Zulpan Hadi**
NIM	**: 21.51.2090**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

HALAMAN PENGESAHAN

DETEKSI ULASAN PALSU MENGGUNAKAN ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE

DETECTION OF FAKE REVIEWS USING RANDOM FOREST ALGORITHM AND SUPPORT VECTOR MACHINE

Dipersiapkan dan Disusun oleh

Zulpan Hadi

21.51.2090

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Sabtu, 02 September 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 September 2023
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

DETEKSI ULASAN PALSU MENGGUNAKAN ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE

DETECTION OF FAKE REVIEWS USING RANDOM FOREST ALGORITHM AND SUPPORT VECTOR MACHINE

Dipersiapkan dan Disusun oleh

Zulpan Hadi

21.51.2090

Telah Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIK BSI Yogyakarta
pada hari Sabtu, 02 September 2023

Pembimbing Utama

Prof. Dr. Kusrini, S.Si., M.Kom
NIK. 190302037

Anggota Tim Penguji

Alva Hendi Muhammad, S.T., M.Eng., Ph.D
NIK. 190302206

Mei P. Kurniawan, S.Kom., M.Eng., Ph.D
NIK. 190302024

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom., Ph.D
NIK. 190302197

Prof. Dr. Kusrini, S.Si., M.Kom
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 02 September 2023
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Zulpan Hadi
NIM : 21.51.2090
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
DETEKSI ULASAN PALSU MENGGUNAKAN ALGORITMA RANDOM FOREST DAN SUPPORT VECTOR MACHINE

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Dhani Ariatmanto, M.Kom., Ph.D

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 02 September 2023
Yang Menyatakan,

Zulpan Hadi

HALAMAN PERSEMBAHAN

Tesis ini disajikan dengan tulus dan rasa terima kasih kepada Dosen Pembimbing yang telah memberikan bimbingan, dorongan, dan pengetahuan yang sangat berharga selama perjalanan akademis ini. Anda adalah sumber inspirasi bagi kami. Juga, terima kasih kepada teman-teman seperjuangan atas kerja keras, dukungan, dan momen-momen berharga yang telah kita bagikan selama perjalanan ini. Kita telah melewati banyak tantangan bersama, dan ini adalah awal dari banyak pencapaian yang akan datang.

HALAMAN MOTTO

Tak pernah ada yang tahu dengan pasti apa yang akan terjadi di masa depan. Namun, jangan biarkan ketakutan akan kesalahan atau ketidakpastian menghentikan langkah Anda. Cobalah aja dulu, jalani setiap pengalaman dengan penuh semangat, dan biarkan masalah benar atau salahnya menjadi bagian dari perjalanan Anda. Itulah cara kita tumbuh dan belajar

KATA PENGANTAR

Puji syukur atas kehadirat Allah SWT yang telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan laporan penelitian tesis ini dengan baik. Penulisan laporan tesis ini dapat terselesaikan berkat bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terimakasih kepada pihak-pihak yang terlibat dalam penelitian ini:

1. Prof. Dr. M. Suyanto, MM. selaku Rektor Universitas AMIKOM Yogyakarta.
2. Ibu Prof. Dr. Kusrini, M.Kom. selaku Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom. selaku Wakil Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta sekaligus selaku Pembimbing Utama.
4. Bapak Dhani Ariatmanto, M.Kom., Ph.D., selaku dosen Pembimbing Pendamping
5. Dr. Andi Sunyoto, M.Kom. selaku dosen Penguji Seminar Hasil
6. Emha Taufiq Luthfi, S.T., M.Kom. selaku dosen Penguji Seminar Hasil
7. Alva Hendi Muhammad, S.T., M.Eng., Ph.D. selaku dosen Penguji Ujian Tesis
8. Hanafi, S.Kom., M.Eng., Ph.D. selaku dosen Penguji Ujian Tesis

Dengan diiringi doa dan ucapan terimakasih, penulis berharap semoga tesis ini dapat bermanfaat. Saran, harapan, kritik yang membangun selalu penulis untuk perbaikan di masa yang akan datang. Terimakasih

Yogyakarta, 02 September 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR ISTILAH

Dataset adalah kumpulan data yang terstruktur yang berisi informasi atau pengamatan tentang suatu topik atau domain tertentu.

Kernel adalah metode sederhana yang digunakan untuk mengambil data nonlinier dengan dimensi rendah dan memetakkannya atau mengubahnya menjadi ruang dimensi lebih tinggi untuk menganalisis pola atau relasi yang lebih kompleks.

Data Training, juga dikenal sebagai dataset pelatihan (training dataset), merupakan data input yang diberikan kepada algoritma pembelajaran mesin agar ia dapat mempelajari pola atau hubungan antara fitur (atribut) input dan hasil output yang sesuai.

Data Testing, juga disebut sebagai dataset uji (test dataset), adalah sebagian data dari keseluruhan dataset yang digunakan untuk menguji performa model yang telah dilatih menggunakan data latih. Hal ini membantu dalam mengukur sejauh mana model mampu menggeneralisasi dan melakukan prediksi akurat pada data yang tidak pernah dilihat sebelumnya.

INTISARI

Dalam era digital, pengaruh ulasan online terhadap keputusan konsumen, terutama dalam reservasi hotel, menjadi signifikan. Ulasan memberikan wawasan tentang kualitas layanan dan fasilitas hotel. Namun, ulasan palsu yang sengaja dibuat untuk memanipulasi persepsi konsumen juga semakin umum. Ulasan palsu bisa bersifat positif untuk promosi atau negatif untuk merugikan reputasi. Ulasan ini mempengaruhi pandangan konsumen terhadap merek dan keputusan reservasi hotel.

Untuk mengatasi ini, penelitian ini menerapkan algoritma Support Vector Machine (SVM) dan Random Forest dalam mendeteksi ulasan palsu. Data yang digunakan adalah data sekunder dari studi sebelumnya. Melalui 6 skenario yang berbeda, dengan n-gram dan seleksi fitur, SVM menghasilkan akurasi tertinggi pada Skenario 3 (92.81%), sedangkan Random Forest mencapai akurasi tertinggi pada Skenario 3 (88.13%).

Berdasarkan eksperimen ini, SVM lebih unggul dalam deteksi ulasan palsu dibandingkan dengan Random Forest. Penelitian ini memberikan pandangan berharga tentang bagaimana algoritma pembelajaran mesin dapat membantu mengidentifikasi ulasan palsu, yang berpotensi meningkatkan integritas dan kepercayaan konsumen dalam reservasi hotel secara online.

Kata kunci: ulasan palsu, n-gram, feature selection, SVM, Random Forest

ABSTRACT

In the digital era, the impact of online reviews on consumer decisions, particularly in hotel reservations, has become significant. Reviews provide insights into the quality of hotel services and facilities. However, intentionally fabricated fake reviews aimed at manipulating consumer perceptions have also become increasingly common. Fake reviews can be positive for promotion or negative to harm reputation. These reviews influence consumers' views on brands and hotel reservation choices

To address this issue, this research applies Support Vector Machine (SVM) and Random Forest algorithms for detecting fake reviews. The data used is secondary data from previous studies. Through 6 different scenarios, utilizing n-gram and feature selection techniques, SVM achieves the highest accuracy in Scenario 3 (92.81%), while Random Forest reaches its highest accuracy in Scenario 3 (88.13%)

Based on this experiment, SVM outperforms Random Forest in detecting fake reviews. This study offers valuable insights into how machine learning algorithms can help identify fake reviews, potentially enhancing the integrity and trust of consumers in making hotel reservations online

Keyword: fake reviews, n-gram, feature selection, SVM, Random forest

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam era digital saat ini, ulasan online memiliki peran yang sangat penting dalam pengambilan keputusan konsumen. Ulasan ini memberikan konsumen informasi terkait produk, yang pada gilirannya meningkatkan kepercayaan mereka selama pembelian online. Ulasan ini ditawarkan oleh perusahaan e-commerce sebagai alternatif bagi konsumen untuk berinteraksi secara fisik atau visual dengan produk . Ulasan online semakin dianggap sebagai sumber informasi penting yang memfasilitasi konsumen untuk membuat keputusan pembelian, dan diketahui memiliki pengaruh yang kuat (Zhang et al., 2014).

Khususnya dalam konteks pemesanan hotel secara online, ulasan dari pengguna sebelumnya dapat menjadi acuan yang berharga. Ulasan tersebut memberikan informasi tentang kualitas layanan, fasilitas, kebersihan, dan pengalaman secara umum di hotel yang sedang dipertimbangkan. Saat konsumen terpapar ulasan hotel online, ini menciptakan kesan tentang merek hotel di pikiran konsumen, yang berpengaruh pada niat pemesanan hotel mereka (Chakraborty & Biswal, 2020)

Ulasan online memainkan peran penting dalam keputusan pemesanan konsumen (Yu et al., 2016). Namun, bersamaan dengan perkembangan ini, juga muncul fenomena ulasan palsu atau bias yang dapat mengaburkan pandangan konsumen dan mempengaruhi keputusan mereka. Ulasan tersebut sengaja dibuat

dengan tujuan untuk memanipulasi persepsi konsumen tentang suatu produk atau layanan. Ulasan palsu dapat berupa ulasan positif palsu yang ditulis oleh pihak yang berkepentingan untuk mempromosikan produknya, atau ulasan negatif palsu yang ditulis oleh pesaing atau pihak lain untuk merugikan reputasi produk atau layanan tertentu. Dalam satu studi, dampak ulasan online pada niat pemesanan hotel ditekankan oleh persepsi pelanggan. Selain itu, studi ini juga menunjukkan bahwa ada interaksi antara peringkat numerik yang diberikan kepada suatu produk atau layanan dan jumlah ulasan verbal yang diterimanya selama tahap pertama proses pengambilan keputusan saat memesan hotel (Gavilan et al., 2018).

Secara umum, ulasan online berfungsi sebagai alat penting untuk membantu konsumen membuat keputusan pembelian mereka, termasuk dalam konteks pemesanan hotel. Ulasan tersebut memberikan pelanggan gambaran tentang apa yang bisa mereka harapkan dari hotel, termasuk kualitas layanan, fasilitas, dan pengalaman secara umum. Oleh karena itu, ulasan online dapat berfungsi sebagai sumber informasi yang berharga bagi konsumen saat merencanakan perjalanan mereka (El-Said, 2020), (Vo et al., 2022).

Ulasan online memiliki peranan penting dalam pengambilan keputusan konsumen karena dapat mempengaruhi niat dan keputusan pembelian. Beberapa penelitian sebelumnya telah membahas tentang pengaruh ulasan palsu dalam konteks ini, seperti penelitian yang dilakukan oleh (Elmogy et al., 2021). Penelitian tersebut mengungkap bahwa saat ini, ulasan palsu atau manipulatif secara sengaja ditulis untuk membangun reputasi virtual dan menarik calon pelanggan. Ulasan positif umumnya menarik lebih banyak pelanggan dan meningkatkan penjualan

secara signifikan. Oleh karena itu, identifikasi ulasan palsu menjadi fokus penelitian yang penting dan terus berkembang. Dalam penelitian ini, (Elmogy et al., 2021) mengusulkan pendekatan pembelajaran mesin untuk mengidentifikasi ulasan palsu. Selain melakukan proses ekstraksi fitur-fitur dari ulasan, penelitian ini juga menerapkan teknik rekayasa fitur untuk mengekstrak berbagai perilaku penulis ulasan. Percobaan dilakukan pada dataset ulasan restoran Yelp yang nyata dengan dan tanpa menggunakan fitur perilaku pengguna yang diekstrak. Hasilnya dibandingkan menggunakan beberapa metode klasifikasi, seperti KNN, Naive Bayes (NB), SVM, Regresi Logistik, dan Random Forest. Selain itu, evaluasi juga mempertimbangkan penggunaan model bahasa n-gram, terutama bi-gram dan tri-gram. Hasil penelitian menunjukkan bahwa KNN dengan parameter K=7 memiliki kinerja terbaik dalam hal f-score, mencapai f-score tertinggi sebesar 82,40%. Hasil tersebut menunjukkan peningkatan f-score sebesar 3,80% ketika fitur perilaku penulis ulasan diekstrak dan dimasukkan dalam analisis. Algoritma SVM dan Random Forest yang menggunakan tri-gram berhasil mencapai akurasi sebesar 86,9% untuk SVM dan 86,8% untuk Random Forest. Meskipun hasil ini termasuk tinggi, namun peneliti merasa hasil akurasi ini bisa ditingkatkan lagi.

Menurut penelitian yang dilakukan oleh (Alsubari et al., 2021) ulasan produk online memiliki peran krusial dalam menentukan keberhasilan atau kegagalan bisnis E-commerce. Sebelum membeli produk atau layanan, konsumen cenderung membaca ulasan online dari pelanggan sebelumnya untuk mendapatkan rekomendasi mengenai detail produk dan membuat keputusan pembelian. Namun, ada risiko peningkatan atau penurunan kualitas produk E-business tertentu akibat

ulasan palsu yang ditulis oleh penipu. Ulasan palsu ini dapat menyebabkan kerugian finansial bagi bisnis E-commerce dan membingungkan konsumen dalam mencari produk alternatif yang tepat. Oleh karena itu, pengembangan sistem deteksi ulasan palsu menjadi sangat penting bagi bisnis E-commerce. Dalam metodologi yang diusulkan, penelitian ini menggunakan empat set data ulasan palsu standar dari berbagai domain, termasuk hotel, restoran, Yelp, dan Amazon. Selain itu, metode preprocessing seperti penghapusan kata penghubung, penghapusan tanda baca, dan tokenisasi dilakukan, serta metode pengisian urutan untuk menjaga panjang urutan masukan tetap selama pelatihan, validasi, dan pengujian model. Metodologi ini menggunakan berbagai ukuran set data, dan matriks word-embedding fitur n-gram dari teks ulasan dikembangkan dengan bantuan lapisan word-embedding sebagai salah satu komponen dalam model yang diusulkan. Teknik konvolusi dan max-pooling dari CNN diterapkan dalam pengurangan dimensi dan ekstraksi fitur. Dengan memanfaatkan mekanisme gerbang, lapisan LSTM digabungkan dengan teknik CNN untuk mempelajari dan mengelola informasi kontekstual dari fitur-fitur n-gram pada teks ulasan. Pada akhirnya, fungsi aktivasi sigmoid digunakan sebagai lapisan terakhir dalam model yang diusulkan, yang melakukan tugas klasifikasi biner terhadap teks ulasan untuk membedakan antara ulasan palsu atau jujur. Penelitian ini melakukan evaluasi model CNN-LSTM yang diusulkan dalam dua jenis eksperimen, yaitu eksperimen dalam domain dan eksperimen lintas domain. Pada eksperimen dalam domain, model diterapkan pada set data masing-masing secara individu, sementara pada eksperimen lintas domain, semua set data dikumpulkan dan dianalisis secara keseluruhan. Hasil pengujian model pada set

data eksperimen dalam domain menunjukkan akurasi 77%, 85%, 86%, dan 87% untuk set data restoran, hotel, Yelp, dan Amazon, masing-masing. Dalam eksperimen lintas domain, model yang diusulkan mencapai akurasi sebesar 89%. Selain itu, dilakukan analisis perbandingan hasil eksperimen dalam domain dengan pendekatan yang sudah ada berdasarkan metrik akurasi, dan hasilnya menunjukkan bahwa model yang diusulkan memiliki performa yang lebih baik dibandingkan metode-metode yang dibandingkan tersebut. Meskipun model yang diusulkan telah menunjukkan performa yang baik, masih ada ruang untuk perbaikan dan pembandingan dengan metode-metode yang sudah ada. Dalam pengembangan sistem deteksi ulasan palsu, perlu dilakukan studi lebih lanjut untuk membandingkan dan meningkatkan performa model dengan menggunakan pendekatan yang berbeda.

Menurut penelitian yang dilakukan oleh (Mohawesh et al., 2021), ditemukan bahwa dalam industri e-commerce, ulasan pengguna memiliki peran yang signifikan dalam menentukan pendapatan suatu organisasi. Pengguna online sangat mengandalkan ulasan sebelum membuat keputusan tentang produk dan layanan. Oleh karena itu, kepercayaan terhadap ulasan online memiliki kepentingan yang besar bagi bisnis dan dapat langsung mempengaruhi reputasi dan profitabilitas perusahaan. Sayangnya, beberapa bisnis menggaji spammer untuk memposting ulasan palsu yang mengeksploitasi keputusan pembelian konsumen. Hal ini mendorong penelitian tentang teknik deteksi ulasan palsu dalam dua belas tahun terakhir. Namun, hingga saat ini masih kurangnya survei yang dapat menganalisis dan merangkum pendekatan-pendekatan yang ada dalam deteksi ulasan palsu.

Untuk mengatasi masalah ini, makalah survei ini menjelaskan tugas deteksi ulasan palsu, merangkum dataset yang sudah ada beserta metode pengumpulannya. Selain itu, makalah ini juga menganalisis teknik ekstraksi fitur yang sudah ada dan secara kritis merangkum serta menganalisis teknik-teknik yang ada untuk mengidentifikasi kesenjangan dalam dua kelompok utama, yaitu metode pembelajaran mesin statistik tradisional dan metode pembelajaran mendalam. Selanjutnya, dilakukan studi benchmark untuk menginvestigasi kinerja berbagai model jaringan saraf dan transformer yang belum digunakan sebelumnya dalam deteksi ulasan palsu. Hasil eksperimen pada dua dataset benchmark menunjukkan bahwa RoBERTa memiliki kinerja sekitar 7% lebih baik daripada metode terkini dalam domain campuran untuk dataset penipuan, dengan akurasi tertinggi mencapai 91,2%, yang dapat dijadikan sebagai dasar untuk penelitian masa depan. Pada akhirnya, penelitian ini menyoroti kesenjangan yang masih ada dalam bidang penelitian ini dan mengidentifikasi kemungkinan arah penelitian masa depan.

Penelitian yang dilakukan oleh (Hassan & Islam, 2019) membahas tentang pendeteksian review palsu secara online dengan menggunakan model semi-supervised dan supervised text mining. Model tersebut menggabungkan algoritma Naive Bayes dan Support Vector Machine untuk meningkatkan performa klasifikasi. Selain itu, paper ini juga membandingkan efisiensi kedua teknik tersebut pada dataset yang berisi review hotel. Dengan menggunakan rasio 80:20, hasil akurasi terbaik dari kedua model yang digunakan adalah menggunakan Naive Bayes dengan akurasi sebesar 0.8521 untuk model semi-supervised dan 0.8632 untuk model supervised.

Penelitian yang dilakukan oleh (Abhinandan V. et al., 2020) menyatakan bahwa ulasan online memainkan peran yang sangat penting dalam e-commerce saat ini dalam pengambilan keputusan. Sebagian besar pelanggan membaca ulasan produk atau toko sebelum memutuskan apa yang akan dibeli dan dari mana membeli serta apakah akan membeli atau tidak. Karena menulis ulasan palsu memiliki keuntungan finansial, terjadi peningkatan yang besar dalam penipuan opini spam di situs ulasan online. Pada dasarnya, ulasan palsu atau penipuan opini adalah ulasan yang tidak jujur. Ulasan positif tentang suatu objek target dapat menarik lebih banyak pelanggan dan meningkatkan penjualan; ulasan negatif tentang suatu objek target dapat menyebabkan permintaan yang lebih rendah dan penurunan penjualan. Ulasan palsu ini ditulis dengan sengaja untuk menipu calon pelanggan agar mempromosikan atau mencemarkan reputasi mereka. Paper ini bertujuan untuk membandingkan empat algoritma klasifikasi, yaitu Naïve Bayes, Random Forest, Logistic Regression, dan SVM, dalam mendeteksi ulasan palsu. Dalam penelitian ini, digunakan metode ekstraksi fitur n-gram. Hasil penelitian menunjukkan bahwa SVM memiliki akurasi tertinggi, yakni sebesar 74,07%. Meskipun hasil ini masih relatif rendah, peneliti akan mencoba meningkatkan akurasi dengan mengombinasikan metode n-gram dan teknik seleksi fitur pada masing-masing algoritma yang digunakan.

Peneltian yang dilakukan oleh (Somantri & Apriliani, 2018) bertujuan untuk mengoptimalkan model SVM dengan menerapkan feature selection menggunakan algoritma Information Gain (IG) dan Chi Square. Hasil penelitian menunjukkan bahwa penggunaan feature selection dengan metode Information Gain (IG) pada

model SVM (SVM-IG) menghasilkan tingkat akurasi terbaik sebesar 72,45%, mengalami peningkatan sekitar 3,08% dari tingkat awal. Selain itu, penelitian ini juga menunjukkan bahwa SVM-IG memiliki tingkat akurasi yang lebih baik dibandingkan dengan SVM dan Chi Squared (SVM-CS). Dengan demikian, penelitian ini membuktikan bahwa penggunaan SVM dengan feature selection menggunakan Information Gain (IG) dapat meningkatkan akurasi dalam klasifikasi tingkat kepuasan pelanggan terhadap warung dan restoran kuliner di Kota Tegal. Dengan adanya temuan ini, peneliti akan mencoba menerapkan feature selection untuk meningkatkan akurasi dalam deteksi ulasan palsu menggunakan SVM dan Random Forest.

Untuk meningkatkan akurasi dalam deteksi ulasan palsu menggunakan algoritma SVM dan Random Forest, peneliti akan melakukan beberapa kombinasi metode. Salah satunya adalah dengan menggabungkan penggunaan n-gram sebagai metode ekstraksi fitur untuk menggambarkan pola dan karakteristik ulasan, serta teknik seleksi fitur untuk memilih fitur-fitur yang paling berpengaruh. Dengan menggunakan kombinasi ini, diharapkan peneliti dapat meningkatkan akurasi deteksi ulasan palsu dan memperoleh hasil yang lebih baik dalam membedakan ulasan palsu dan asli.

1.2. Rumusan Masalah

Berdasarkan latar belakang permasalahan diatas dapat disimpulkan permasalahan yang ada yaitu:

a. Bagaimana algoritma Random Forest dan SVM dapat digunakan untuk mendeteksi ulasan palsu dan apa perbandingan akurasi antara keduanya?

b. Apakah penggunaan n-gram dapat meningkatkan akurasi dalam mendeteksi ulasan palsu?

c. Apa saja teknik seleksi fitur yang dapat digunakan untuk meningkatkan akurasi dalam deteksi ulasan palsu dan bagaimana penerapannya pada algoritma Random Forest dan SVM?

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

a. Fokus pada penggunaan algoritma SVM dan Random Forest sebagai metode deteksi untuk mengidentifikasi ulasan palsu pada dataset yang diberikan.

b. Menggunakan dataset ulasan yang telah dikumpulkan sebelumnya dan terdiri dari ulasan yang telah dikategorikan sebagai palsu atau asli

c. Menerapkan teknik ekstraksi fitur, termasuk n-gram, untuk mengubah teks ulasan menjadi representasi fitur numerik yang dapat digunakan oleh SVM dan Random Forest.

d. Melakukan seleksi fitur untuk mengidentifikasi subset fitur yang paling informatif dan relevan dalam mendeteksi ulasan palsu. Teknik seleksi fitur yang digunakan akan dijelaskan secara lebih rinci

e. Mengukur kinerja metode deteksi dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Perbandingan kinerja antara SVM dan Random Forest akan dievaluasi sebelum dan setelah penerapan seleksi fitur

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

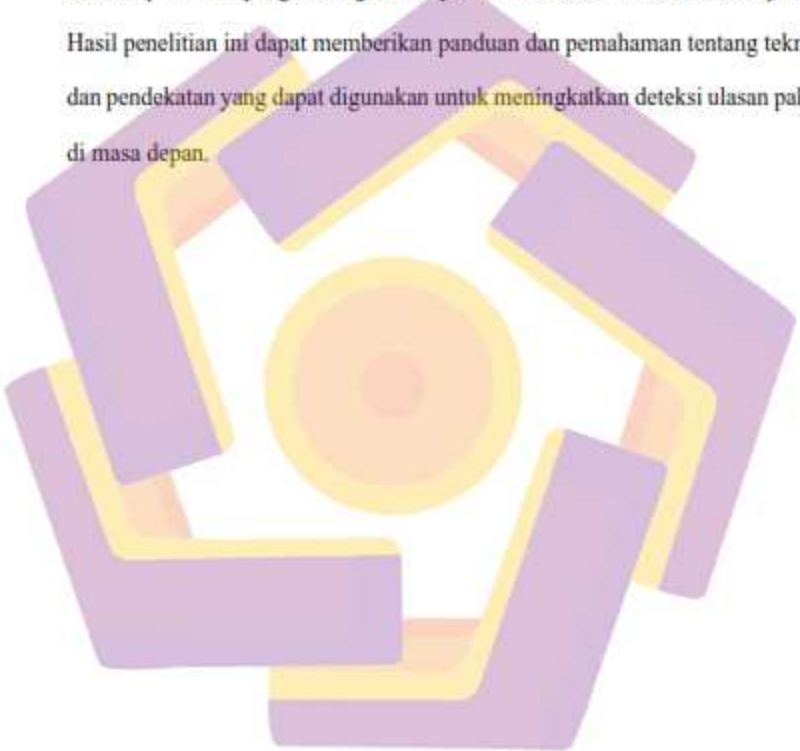
a. Menguji dan mengevaluasi akurasi algoritma Random Forest dan SVM dalam mendeteksi ulasan palsu.
b. Mengidentifikasi pengaruh penggunaan n-gram sebagai fitur terhadap akurasi deteksi ulasan palsu.
c. Menerapkan teknik seleksi fitur yang relevan pada algoritma Random Forest dan SVM untuk meningkatkan akurasi deteksi ulasan palsu.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah:

a. Diharapkan dengan menggunakan sistem yang handal untuk mendeteksi ulasan palsu, dapat memastikan bahwa ulasan yang ditampilkan kepada calon pelanggan adalah akurat dan dapat dipercaya. Ini akan membantu meningkatkan kepercayaan pelanggan terhadap produk atau layanan yang ditawarkan.
b. Diharapkan dengan menggunakan algoritma deteksi ulasan palsu yang akurat, dapat menjaga reputasi mereka dengan menghapus ulasan palsu dan menjaga integritas platform ulasan mereka
c. Diharapkan pengguna dapat membuat keputusan yang lebih baik berdasarkan ulasan yang valid dan dapat dipercaya. Selain itu, diharapkan hal ini akan memberikan kontribusi yang signifikan dalam pengembangan produk, perbaikan layanan, dan peningkatan kepuasan pelanggan.

d. Diharapkan penelitian ini akan memberikan kontribusi terhadap bidang deteksi ulasan palsu dengan membandingkan kinerja algoritma Random Forest dan SVM serta teknik seleksi fitur yang digunakan
e. Diharapkan penelitian ini dapat menjadi acuan bagi penelitian-penelitian berikutnya dalam pengembangan dan perbaikan metode deteksi ulasan palsu. Hasil penelitian ini dapat memberikan panduan dan pemahaman tentang teknik dan pendekatan yang dapat digunakan untuk meningkatkan deteksi ulasan palsu di masa depan.



BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian terdahulu yang pernah dilakukan, relevan dan dijadikan studi literatur adalah sebagai berikut :

Sebuah studi yang dilakukan oleh (Shuqin & Jing, 2019) mengusulkan untuk membangun model klasifikasi dengan mengintegrasikan fitur teks komentar dan perilaku pengguna. Akan tetapi, data komentar yang diperoleh pada kenyataannya sebagian besar tidak berlabel. Oleh karena itu, peneliti mengusulkan model MPINPUL (Pencampuran Populasi dan Pembelajaran Sifat Individu PU) berdasarkan beberapa fitur untuk membangun model klasifikasi ulasan palsu. Dalam tulisan ini, model MPINPUL dibagi menjadi empat langkah. Pertama, algoritma k-means yang dibatasi diusulkan untuk menghitung contoh kepercayaan negatif. Keuntungan dari k-means yang dibatasi adalah dapat memperluas kumpulan contoh positif sambil mengidentifikasi contoh negatif tepercaya. Kemudian, LDA dan k-means digunakan untuk menghitung beberapa sampel representatif untuk masing-masing contoh positif dan negatif. Kemudian kami menggunakan ide populasi dan individualitas untuk menentukan label kategori sampel. Akhirnya, pengklasifikasi didirikan. Hasil percobaan pada kumpulan data nyata menunjukkan bahwa tingkat pengenalan model MPINPUL yang diusulkan dalam makalah ini lebih tinggi daripada fitur tunggal lainnya dalam kondisi fitur fusi

Penelitian yang dilakukan oleh (Budhi et al., 2021) tentang kecurangan yang sering dilakukan oleh penjual online untuk memberikan ulasan palsu pada produk mereka, ini merusak kepercayaan pembeli terhadap ulasan pada produk. Untuk mendeteksi ulasan secara akurat penelitian ini menyelidiki beberapa metode textual-based featuring dan preprocessing Bersama dengan klasifikasi machine learning temasuk model tunggak dan assemble. Karena terjadi data yang tidak seimbang peneliti mengusulkan random sampling. Hasilnya menunjukan bahwa Teknik pengambilan sample dapat meningkatkan akurasi kelas ulasan palsu. Akurasi dapat ditingkatkan hingga maksimun 84,5% dan 75,6% untuk pengambilan sample random under and over-sampling. Namun untuk akurasi ulasan *real* turun menjadi 75% dan 58,8% untuk pengambilan sample random under and over-sampling. Pada penelitian ini ukuran data yang digunakan sangat berpengaruh dimana pada penelitian ini menggunakan random under and over-sampling untuk mengatasi masalah tersebut.

Penelitian yang dilakukan oleh (Hassan & Islam, 2019) memperkenalkan model semi-supervised dan supervised untuk mendeteksi ulasan palsu. Dimana peneliti menggunakan feature word frequency count, sentiment score, review size untuk meningkatkan akurasi dari penelitian sebelumnya dimana feature yang digunakan adalah bigram, sentiment score, POS, LIWC. Dengan hasil terbaik didapatkan oleh naïve bayes dengan menggunakan model supervised dengan ratio data training dan data test sebesar 80%:20% dan mendapatkan akurasi sebesar 86%. Pada penelitian ini ratio data training yang digunakan sangat berpengaruh dikarenakan dari 4 percoba yang dilakukan oleh peneliti dataset dengan ratio 80:20

mendapatkan akurasi tertinggi walaupun perbedaannya tidak signifikan. Penelitian selanjutnya peneliti juga akan menggunakan ratio 80:20 untuk pembagian datasetnya.

Penelitian yang dilakukan oleh (Hassan & Islam, 2020) dengan tujuan untuk melanjutkan penelitian sebelumnya yang dilakukan oleh (Hassan & Islam, 2019) juga. Kali ini peneliti menggunakan pendekatan supervised machine learning untuk mengklasifikasu ulasan palsu menggunakan kumpulan data ulasan hotel. Dengan menggunakan TF-IDF, Empath categories, sentiment score dapat meningkatkan akurasi dari penelitian sebelumnya dimana algoritma svm mendapatkan akursi terbaik sebesar 88%, dimana 2% lebih baik dari pada penelitian sebelumnya yang menggunakan algoritma naïve bayes. Pada penelitian selanjutnya peneliti akan menggunakan feature selection untuk memaksilakan lagi akurasi dari algoritma yang digunakan.

Penelitian yang dilakukan oleh (Elmogy et al., 2021) membandingkan kinerja dari beberapa percobaan yang dilakukan pada daraset restoran yelp dengan dan tanpa feature ekstrasi dari perilaku pengguna. Dalam kedua kasus tersebut, panelitian ini membandingkan beberapa klasifikasi: KNN, Naive Bayes (NB), SVM, Regresi Logistik, dan Random Forest. Hasilnya menunjuukan bahwa KNN(K=7) mengungguli pengklasifikasi lainnya dalam hal skor-f yang mencapai f1-score 82,40%. Pada penelitian ini fokus dalam menaikan f1-skor, padahal dalam klasifikasi text akurasi yang paling banyak menjadi patokan perbandingan. Pada penelitian selanjutnya peneliti akan lebih memfokuskan pada menaikan akurasi dari klasifikasi dan juga menambahkan feature sekection untuk meningkatkan akurasi

Penelitian lainnya yang dilakukan oleh (Alsubari et al., 2022) menggunakan ulasan hotel sebanyak 1600 yang dikumpulkan dari satu situs web pemesanan popular, *trip advisor*. Peneliti ini mengumpulkan data ini menyempurnakan semua ulasan peringkat bintang 5 dan 3 dari 20 hotel di Chicago. Dataset telah diproses sebelumnya dengan menambahkan fitur seperti panjang ulasan, *four-grams*, *sentiment score*, and *POS*. Sebelum melakukan langkah ekstraksi fitur, data perlu diekspos ke pembersihan tertentu. ing, seperti penghapusan tanda baca untuk menghilangkan tanda baca dari teks ulasan (? !:;," .), penghentian penghapusan kata untuk membersihkan kalimat ulasan dari kata artikel ('the,' 'a,' 'an,' 'in '), menghapus kata dan karakter yang tidak perlu dari seluruh kumpulan data, dan tokenisasi data untuk membagi setiap kalimat konten ulasan menjadi kata, kata kunci, frasa, dan potongan informasi yang terpisah. Setelah itu dilakukan konversi dataset dalam bentuk fitur *TF-IDF* dan sebelum melatih *classifier machine learning*, dataset dibagi menjadi 80% data *training* dan 20% data *testing*. Dalam percobaan ini digunakan empat pengklasifikasi yaitu, *Naïve Bayes, Support Vector Machine, Random Forest* dan *Adaptive Boosting*. Dimana Random Forest mendapatkan akurasi terbaik sebesar 95%. Penelitian selanjunya peneliti akan menggunakan algoritma Random Forest juga dan menambahkan feature selection untuk meningkatkan akurasinya.

Perbedaan penelitian jika di bandingkan dengan penelitian terdahulu adalah pada penggunaan fitur selection yang dimana tujuan dari penggunaan feature selection untuk meningkatkan akurasi dari masing-masing algoritma yang digunakan. Untuk lebih jelasnya dapat dilihat pada Tabel 2.1.

2.2. Keaslian Penelitian

Tabel 2.1 Matriks Literatur Review Dan Posisi Penelitian
Deteksi Ulasan Palsu Menggunakan Algoritma Random Forest Dan Support Vector Machine

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Fake Reviews Detection Based on Text Feature and Behavior Feature	Shuqin and Jing, Proceedings - 21st IEEE International Conference on High Performance Computing and Communications, 17th IEEE International Conference on Smart City and 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019 2019	Membangun model klasifikasi dengan menggunakan fitur teks komentar dan prilaku pengguna pada data tidak berlabel	Hasil percobaan pada penelitian menunjukkan bahwa tingkat pengenalan model MPINPUL yang diusulkan dalam makalah ini lebih tinggi daripada fitur tunggal lainnya.	Tidak dijelaskan fitur apa saja yang digabungkan pada model MPINPUL	Pada penelitian ini akan mencoba mengabungkan
2	Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features	Budhi, Chiong, and Wang , Multimedia Tools and Applications, 2021	Membangun sistem deteksi ulasan palsu menggunakan dua metode yaitu model tungal dan esamble	Dengan menggunakan data yang seimbang dapat meningkatkan akurasi hingga maksimal 84,5% dan 75,6% untuk data dengan kelas fake. Akan tetapi untuk data dengan kelas real turun menjadi 75% dan 58,8%	dengan menggunakan feature selection akan dapat meningkatkan akurasi dengan lebih maksimal	Penelitian selanjutnya akan menambahkan feature selection untuk memaksimalkan akurasi menggunakan algoritma yang diusulkan

Tabel 2. 1 Lanjutan Matriks Literatur Review Dan Posisi Penelitian

No	Judul	Penelitı, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Detection of fake online reviews using semi-supervised and supervised learning	Hassan and Islam, 2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019, 2019	Makalah ini menggunakan semi-supervised dan supervised untuk mendeteksi ulasan palsu serta membandingkan berapa efisien 4kedua teknik tersebut	Dengan menggunakan feature word frequency count, sentiment score, review size dapat meningkatkan akurasi dari penelitian sebelumnya dimana feature yang digunakan adalah bigram, sentiment score, POS, LIWC. Dengan hasil terbaik didapatkan oleh naive bayes	Alur penelitianya tidak digambarkan secara menyeluruh akan tetapi hanya menjelaskan tentang proses ekstraksi fitur yang diusulkan	Penelitian selanjutnya akan mengambarkan secara jelas alur penelitiannya dari awal sampai akhir
4	A Supervised Machine Learning Approach to Detect Fake Online Reviews	Hassan and Islam, ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings, 2020	Penelitian ini mengembangkan machine learning supervised yang efektif untuk melakukan klasifikasi ulasan palsu	Dengan menggunakan TF-IDF, Empath categories, sentiment score dapat meningkatkan akurasi dari penelitian sebelumnya dimana algoritma svm mendapatkan akursi terbaik sebesar 88%, dimana 2% lebih baik dari pada penelitian sebelumnya yang menggunakan algoritma naive bayes	Untuk memaksimalkan akurasinya bisa menggunakan feature selection	Penelitian selanjutnya akan mencoba menambahkan feature selection untuk meningkatkan akurasinya

Tabel 2.1 Lanjutan Matriks Literatur Review Dan Posisi Penelitian

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Fake Reviews Detection using Supervised Machine Learning	Elmogy, Ahmed M. Tariq, Usman Ibrahim, Atef Mohammed, Ammar, International Journal of Advanced Computer Science and Applications, 2021	Penelitian ini membandingkan kinerja beberapa percobaan yang dilakukan pada dataset ulasan restoran Yelp nyata dengan dan tanpa fitur yang diambil dari perilaku pengguna	Dengan menggunakan bi-gram, tri-gram klasifikasi KNN(K=7) mengungguli pengklasifikasi lainnya dalam ulasan palsu dengan f1-skor terbaik yaitu 82,40%	Penelitian ini fokus dalam menaikan f1-skor, padahal dalam klasifikasi text akurasi yang banyak jadi patokan perbandingan	Pada penelitian selanjutnya peneliti akan lebih memfokuskan pada menaikan akurasi dari klasifikasi dan juga menambahkan feature sekection untuk meningkatkan akurasi
6	Detection of fake online reviews using semi-supervised and supervised learning	Hassan, Rakibul Islam, Md Rabiul. 2nd International Conference on Electrical, Computer and Communication Engineering, ECCE, 2019	Untuk meningkatkan kinerja klasifikasi	Paper ini juga menemukan bahwa klasifikasi Naive Bayes memberikan akurasi tertinggi untuk data yang sudah dilabeli dengan bai	Untuk meningkatkan akurasinya, dapat menambahkan konteks kata pada proses ekstraksi kata. Untuk memperkaya representasi kata.	Pada penelitian selanjutnya akan menggunakan n-gram model untuk memperkaya representasi kata.

2.3. Landasan Teori

2.3.1 Natural Langue Processing (NLP)

NLP adalah kemampuan program komputer untuk memahami bahasa manusia seperti yang diucapkan dan ditulis. Inilah yang kita sebut bahasa alami. Dalam perkembangannya, *NLP* dikenal sebagai proses pengolah kata yang membutuhkan bantuan pemrosesan bahasa alami karena mesin tidak dapat memahami bahasa alami manusia (Munasatya & Novianto, 2020).

Dalam proses komputasi, bahasa harus direpresentasikan sebagai serangkaian *symbol* memenuhi aturan tertentu. Sederhananya, *NLP* mencoba mencoba untuk membuat komputer dapat mengerti perintah-perintah yang ditulis dalam standar bahasa manusia (Lisangan et al., n.d.).

Terdapat beberapa alasan yang menyulitkan NLP (Lisangan et al. n.d.). Secara khusus, masalah ambiguitas dan makna ganda, serta banyaknya kosakata yang berkembang dari waktu ke waktu. Jadi *NLP* tidak peduli bagaimana kalimat itu dimasukkan ke dalam komputer, tetapi menyalin informasi dari kalimat tersebut.

2.3.2 Deteksi Ulasan Palsu

Spam opini mengacu pada opini palsu yang mencoba menipu pembaca atau sistem otomatis dengan memberikan opini positif yang tidak semestinya kepada item target untuk mempromosikan item tersebut, atau dengan memberikan opini negatif yang berbahaya untuk menodai reputasinya. Mendeteksi spam semacam itu sangat penting untuk aplikasi. Salah satu jenis spam opini adalah opini palsu (Jindal & Liu, 2008), secara sengaja menyesatkan pembaca dengan memberikan peringkat positif yang tidak pantas pada properti target untuk mempromosikan properti

tersebut, atau dengan memberikan peringkat negatif yang tidak wajar pada properti lain untuk menodai reputasinya. Untuk mendeteksi ulasan palsu atau spam opini digunakan feature textual dimana metode yang dipakai adalah pos tagging distribusi. Dimana pos tagging digunakan untuk mendapatkan tag atau penanda dari sebuah kata. Pada ulasan yang jujur atau *real* cenderung memiliki banyak kata benda (NN) dan kata sifat (ADJ) sedangkan ulasan palsu memiliki banyak kata kerja (VERB) dan kata keterangan (ADV) (Pasaribu et al., 2019),(Alsubari et al., 2022). Selain menggunakan pos tagging, ada beberapa metode yang digunakan untuk mendeteksi ulasan palsu seperti kata-kata emosi, kata ganti orang dan panjang text. Untuk panjang teks ulasan juga bisa digunakan untuk mendeteksi ulasan palsu menurut (Alsubari et al., 2022) memaparkan bahwa 75% pelaku spam tidak dapat menulis lebih dari 136 kata per ulasan. Lebih dari 90% pengulas jujur menulis 200 kata per ulasan.

2.3.3 Preprocessing Data

Banyak faktor yang mempengaruhi keberhasilan *Machine Learning* pada tugas yang diberikan. Pertama-tama, kita membutuhkan data yang berkualitas. *Preprocessing* data merupakan faktor yang secara langsung mempengaruhi kualitas proses analisis intelektual karena pemecahan masalah dengan sampel awal yang tidak diproses tidak memberikan hasil yang diharapkan, yang dapat menyebabkan kesimpulan yang salah (Madrakhimov et al., 2021). *Preprocessing* data merupakan langkah penting dalam proses penemuan pengetahuan, karena keputusan yang berkualitas harus didasarkan pada data yang berkualitas (Kumar & Chadha, 2012). *Preprocessing* data sering digunakan untuk mengurangi kesalahan dan bias data

pada data mentah sebelum analisis dilakukan (Tong et al., 2011). Ada banyak faktor yang dapat menyebabkan masalah kinerja klasifikasi. Pertama dan terpenting adalah format dan kualitas data. Kondisi tersebut mempersulit proses ekstraksi fitur selama fase pelatihan ketika data mengandung *noise*, redundansi, dan data yang tidak relevan (Kotsiantis & Tzelepis, 2006). Oleh karena itu, *preprocessing* data merupakan langkah penting dalam proses *machine learning*. Algoritma pemilihan subset masa depan akan mengidentifikasi dan menghapus fitur yang tidak relevan dan berlebihan. Efek dari kondisi tersebut adalah mengurangi dimensi data, membuat algoritma pembelajaran bekerja lebih cepat dan lebih efisien. Ini juga dapat meningkatkan efisiensi data dan menghilangkan *noise* data untuk membantu mengidentifikasi *survival of the fittest* (Setyohadi & Kristiawan, 2017). Mereka menunjukkan bahwa pretreatment dapat membantu meningkatkan efisiensi secara signifikan.

Penelitian tentang manfaat *preprocessing* dalam klasifikasi juga telah dilakukan, khususnya pada algoritma jaringan syaraf tiruan (JST). Nawi dkk. (2013) melaporkan manfaat *preprocessing* data menggunakan berbagai teknik untuk meningkatkan konvergensi JST. Karyanya menyatakan bahwa pra-pemrosesan adalah langkah kunci dalam proses penambangan data, dan bahwa kualitas, keandalan, dan ketersediaan adalah beberapa faktor yang mengarah pada keberhasilan interpretasi data dalam JST. Peneliti memproses dataset dari repositori UCI (Wine, Iris, Haberman) menggunakan teknik *preprocessing* normalisasi *min-max*, normalisasi *z-score*, dan normalisasi penskalaan fraksional. Kesimpulannya, peneliti menemukan bahwa menggunakan teknik *preprocessing* meningkatkan

akurasi klasifikasi JST setidaknya 95%. Kondisi ini berarti untuk meningkatkan kinerja algoritma JST (Setyohadi & Kristiawan, 2017).

Preprocessing data terjadi dalam beberapa tahapan berurutan sebagai berikut:

1. Case Folding

   Case Folding digunakan untuk mengubah teks menjadi standar bentuk sehingga mudah dipahami oleh komputer, untuk contoh "Memakan(Makan)" menjadi "memakan(makan)" (Ramadani et al., n.d.).

2. Tokenizing

   Pada tahap ini, kata-kata dipotong menurut urutan yang telah ditentukan. Jika ada pengulangan/kata yang sama dalam struktur kalimat, tetap akan dipersingkat. Sebagai contoh: Pergi ke kebun Anda pagi ini jika Anda ingin memetik buah. Tokenisasi ini memisahkan kata-kata. Yang berikutnya adalah contohnya, dipisahkan oleh | masuk ke: Pergilah | diwaktu | pagi | ini | ke | kebunmu | jika | kamu | hendak | memetik | buahnya (Rifai & Winarko, 2019).

3. Filtering

   Pemfilteran menghilangkan kata-kata yang tidak berarti dan menghentikan kata-kata. *Filtering* digunakan untuk membuang kata-kata yang kurang penting, dan tidak ada artinya. Kondisi dari proses penyaringan adalah bahwa hal itu tidak dapat mengubah makna. Data kata yang dibuang terdapat dalam sebuah tabel yang disebut sebuah daftar berhenti (Ramadani et al., n.d.).

4. Cleansing

Pembersihan adalah proses menghapus semua karakter non-abjad dari posting untuk mengurangi karakter yang tidak perlu atau tidak berarti. \ Karakter tersebut dapat berupa angka, #, @, emoji, atau tautan dari situs web di postingan.

5. Stemming

Stemming adalah proses mencari kata dasar dari sebuah kata. Pilihan kata dasar yang salah dapat mengakibatkan informasi yang diterima salah. Selain itu, proses yang sebenarnya tidak selalu menghasilkan satu kata dasar, karena ada beberapa kata dalam bahasa Indonesia yang memiliki dua kemungkinan, yaitu sebagai kata dasar atau sebagai akhiran, seperti kata "beruang" (Rifai & Winarko, 2019).

2.3.4 Pos Tagging

Label atau *tag* yang diberikan ke suatu kata dalam sautu kalimat menunjukkan kelas kata *(word class)* dari kata yang bersangkutan, dalam konteks kalimat tersebut. Kelas kata ini juga disebut sebagai *part of speech*. Kumpulan atau koleksi label atau *tag part of speech* atau kelas kata disebut sebagai *tagset* (Mulyanto et al., 2017). Dalam proses *POS tagging* bahasa Indonesia tagset yang digunakan seperti ditunjukkan dalam Tabel I (Pisceldo et al., 2009).

Tabel 2.2 Tag Bahasa Indonesia

No	Tag	Deskripsi	Contoh
1	(	Opening Parenthesis	({ [
2	)	Closing parenthesis	) }]
3	,	Comma	,
4	.	Sentence terminator	. ? !
5	:	Colon or ellipsis	: ;

Tabel 2.3 Lanjutan Tag Bahasa Indonesia

No	Tag	Deskripsi	Contoh
6	--	Dash	--
7	"	Opening quotation mark	` "
8	"	Closing quotation mark	' "
9	$	Dollar	$
10	Rp	Rupiah	Rp
11	SYM	Symbols	% & ' " ".)). * + ,. < = > @A[fj] U.S U.S.S.R * ** ***
12	NN	common nouns	Buku, rumah, karyawan, air, gula, rumahnya, kambingmu
13	NNP	Proper nouns	Jakarta, Soekarno-Hatta, Australia, BCA
14	PRP	Personal pronouns	Saya, aku, dia, kami
15	PR	Common pronouns	Kedua-duanya, Ketiga-tiganya, sini, situ, sana
16	WH	WH	Apa, siapa, mengapa, bagaimana, berapa
17	VB	Verbs	Makan, tidur, menyanyi, bermain, terdiam, berputar-putar
18	MD	Modal or auxiliaries verbs	Sudah, bolch, harus, mesti, perlu
19	JJ	Adjectives	Mahal, kaya, besar, malas
20	CD	Cardinal numerals	Satu, juta, milyar, pertama, semua, bertiga
21	NEG	Negations	Bukan, tidak, belum, jangan
22	IN	Prepositions	Di, ke, dari, pada, dengan
23	CC	Coordinate conjunction	Dan, atau, tetapi
24	SC	Subordinate conjunction	Yang, ketika, setelah
25	RB	Adverbs	Sekarang, nanti, sementara, sebab, sehingga
26	WDT	WH-determiners	Apa, siapa, barangsiapa
27	FW	Foreign words	Absurd, deadline, list, science

2.3.5 TF-IDF

TFIDF (Term Frequency Inverse Document Frequency) adalah metode pembobotan berupa integrasi antara term frequency dan inverse document frequency. TFIDF merupakan metode yang merepresentasikan integrasi antara term frequency (TF) dan inverse document frequency (IDF). Term frequency dihitung menggunakan Persamaan (1), dimana term frequency adalah frekuensi kemunculan term dalam dokumen ke-j. Kepadatan dokumen terbalik (IDF) adalah logaritma rasio jumlah total dokumen dalam korpus dengan jumlah dokumen di mana ekspresi minat ditulis secara matematis dalam persamaan (2). Nilai tersebut diperoleh dengan mengalikan keduanya yang dirumuskan dalam persamaan (3).

$$tf_i = \frac{freq_i(d_j)}{\sum_{i=1}^{k} freq_i(d_j)} \quad (1)$$

$$idf_i = log\frac{|D|}{|\{d: t_i \in d\}|} \quad (2)$$

$$(tf - idf)_{ij} = tf_i(d_j) * idf_i \quad (3)$$

Tugas dari metode TFIDF adalah mencari representasi dari nilai setiap dokumen pada dataset pelatihan (training set), kemudian membentuk vektor antara dokumen dan kata (dokumen dengan term), yang kemudian ditentukan kesamaan dokumennya. dan cluster dengan Menggunakan vektor prototipe, juga disebut cluster centroid (Saadah et al., 2013).

2.3.6 N-Gram

Pada dasarnya, model N-gram adalah model probabilistik yang awalnya dikembangkan oleh ahli matematika Rusia pada awal abad ke-20 dan kemudian disempurnakan untuk memprediksi item berikutnya dalam urutan item. Item bisa

berupa huruf/karakter, kata atau yang lainnya tergantung aplikasinya. Salah satunya, model n-gram berbasis kata, digunakan untuk memprediksi kata berikutnya dalam urutan kata tertentu. Dalam artian n-gram hanyalah kumpulan kata-kata, masing-masing panjangnya n kata. Misalnya, n-gram berukuran 1 disebut unigram; ukuran 2 sebagai "bigram"; Ukuran 3 sebagai "trigram" dan seterusnya.

Dalam pembuatan karakter, n-gram adalah substring sepanjang n karakter dari sebuah string, definisi lainnya mgram adalah potongan sejumlah n karakter dari sebuah string. Metode n-gram digunakan untuk mengambil n karakter dari sebuah kata yang dibaca terus menerus dari sumber hingga akhir dokumen. Misalnya: Kata "TEXT" dapat dibagi menjadi n-gram berikut:

uni-gram : T, E, X, T

bi-gram : TE, EX, XT

tri-gram :, TEX, EXT

quad-gram : *TEXT*, EXT_

Dan seterusnya.

Pada saat yang sama, word generation menggunakan metode n-gram untuk mengambil potongan kata sejumlah n dari sebuah rangkaian kata (kalimat, paragraf, bacaan) yang dibaca terus menerus dari teks sumber hingga akhir dokumen. Misalnya: kalimat "Saya melihat cahaya". dapat dibagi menjadi n-gram berikut :

uni-gram : saya, dapat, melihat, cahaya, itu

bi-gram : saya dapat, dapat melihat, itu ada

tri-gram : saya dapat melihat, dapat melihat dia

dan seterusnya

Salah satu keuntungan menggunakan n-gram daripada seluruh kata adalah n-gram tidak terlalu sensitif terhadap kesalahan ketik dalam dokumen (Hanafi, 2009).

2.3.7 Feature Selection

Feature Selection adalah cara untuk membuat pengklasifikasi lebih efisien dan efektif dengan mengurangi jumlah data yang akan dianalisis atau dengan mengidentifikasi fitur yang cocok sebagai bahan pertimbangan pada proses pembelajaran (Moraes et al., 2013). Sehingga dapat mengurangi waktu yang diperlukan untuk memproses data pengklasifikasi dan dapat meningkatkan akurasi, juga karena fitur yang tidak relevan dapat menurunkan data, yang berdampak negatif pada akurasi klasifikasi (Doraisamy et al., 2008). Dengan feature selection dapat meningkatkan pemahaman dan mengurangi biaya data (Arauzo-Azofra et al., 2011).

Algoritma feature selection dibagi menjadi tiga kelompok: Filters, wrappers, dan embedded selectors. Filters menilai setiap fitur secara independen dari pengklasifikasi dan menetapkan skor ke fitur setelah mengevaluasinya dan memilih yang terbaik (Singh et al., 2011). Wrappers mengambil subset dari feature set, mengevaluasi kinerja pengklasifikasi untuk subset itu, dan kemudian pengklasifikasi mengevaluasi subset lainnya. Subset dengan kinerja klasifikasi terbaik dipilih. Jadi wrappers bergantung pada pengklasifikasi yang dipilih. Bahkan wrappers lebih andal karena algoritme klasifikasi memengaruhi akurasi (Novakovic, 2010). Teknik Embedded, di sisi lain, melakukan feature selection selama proses mempelajari data, seperti halnya jaringan saraf tiruan.

Pada teknik filters ada dua metode yang akan digunakan yaitu chi square dan information gain. Chi Square adalah metode seleksi fitur yang digunakan untuk meningkatkan hasil klasifikasi pada analisis sentimen. Metode ini dapat meningkatkan nilai recall, precision, F1-score, dan akurasi walaupun peningkatannya tidak terlalu signifikan, tapi dapat mempengaruhi hasil klasifikasi. Dalam penelitian yang dilakukan oleh (Luthfiana et al., 2020) Chi Square digunakan sebagai metode seleksi fitur dalam pengembangan model untuk klasifikasi otomatis feedback pengguna dengan menggunakan algoritma SVM. Metode chi square dapat mengevaluasi nilai atribut dengan menghitung nilai statistic berkaitan dengan kelas. Chi square statistic juga dilambangkan dengan (x^2) merupkan teknik statistic nonparametik dengan menggunakan data nominal (kategori) dengan tes frekuensi

$$X^2 = \sum_{i=1}^{n} \frac{(0i - Ei)^2}{E_i}$$

Dimana X^2 adalah statistik uji yang asimtotik mendekati X^2 distibusi, 0_i adalah frekuensi yang diamati, dan E_i adalah frekensi yang diharapkan. n adalah jumlah hasil yang mungkin dari setiap peristiwa.

Metode kedua pada teknik rilters yang akan digunakan adalah information gain (ig) metode ini menilai pentingnya atribut dengan mengukur information gain berkaitan dengan kelas. Secara umum IG memperkirakan perubhan entripi information sebelum keadaan yang mengambil beberapa information.

$$IG(Class, Attribute) = H(Class) - H(Class|Attribute)$$

Dimana H menentukan entropi. Lebih khusus lagi, misalkan A adalah himpunan semua atribut dan kelas menjadi atribut tergantung dari semua contoh pelatihan,

nilai (a, y) dengan $y \in A, V$ merupakan himpunan nilai-nilai atribut, yaitu $V = \{value(a, y) \mid a \in A \cap y \in Class\}$ dan | s | adalah jumlah elemen dalam set s. IG untuk atribut $a \in A$ didefinisikan sebagai berikut

$$IG(Class, a) = H(class) - \sum_{v \in V} \frac{|\{y \in class \mid value(a,y)=v\}|}{|Class|} X\ H(\{y \in class \mid value(a, y) = v\})$$

Pada teknik wrappers metode yang digunakan adalah RFE (Recursive Feature Elimination) dimana metode ini berkerja dengan cara menghapus atribut secara rekursif dan membangun model dengan atribut tersisa. Metode ini menghilangkan sifat yang tidak perlu dan lemah, atau fitur yang berkontribusi paling kecil terhadap keberhasilan model klasifikasi, dan pada saat yang sama fitur yang efektif dan kuat yang meningkatkan keberhasilan model. Metode ini menggunakan proses iteratif yang bekerja mirip dengan Backward Elimination. Metode ini pertama-tama membuat model dari seluruh set fitur dan memberi skor pada setiap fitur sesuai dengan efek dan kepentingannya pada variabel target. Model tersebut kemudian dibangun kembali, menghilangkan fitur yang paling tidak penting pada setiap langkah dan menghitung ulang pentingnya setiap fitur hingga kesuksesan terbesar dari model tercapai (Akkaya, 2021).

Pada teknik embedded method terdapat dua metode yang sering digunakan yaitu LASSO (Least Absolute Shrinkage and Selection Operator) dan Elastic Net. Pada teknik ini metode yang digunakan adalah LASSO, dimana LASSO ini adalah metode reduksi dimensi berdasarkan model regresi linier dengan fungsi penalty L1 yang telah menarik perhatian luas dibidang seleksi fitur karena kinerjanya yang efisien (Rajeswari et al., 2016).

2.3.8 Matode Random Forest

Random Forest adalah metode pembelajaran ensemble oleh Breiman pada tahun 2001. Hutan acak adalah kombinasi klasifikasi di mana setiap pohon bergantung secara independen pada nilai acak vektor. *Random Forest* sering digunakan Baik klasifikasi maupun regresi. Hal ini sesuai dengan pendapat (Christy et al., 2021) bahwa *random forest* juga merupakan proses klasifikasi yang tersusun dari *multiple decision tree*. Menurut (Syukron & Subekti, 2018).

Langkah-langkah dari metode *Random Forest* adalah.

1. Menghasilkan *training* baru dengan sampel yang acak.
2. Pada setiap *training* yang baru, maka dibangunlah sebuah pohon dengan pemilihan fitur secara acak juga.
3. Setelah menghasilkan sejumlah pohon, maka data segera diprediksi dengan menggabungkan hasil dari semua pohon dengan voting.

Pohon keputusan juga merupakan serangkaian pertanyaan yang disusun secara sistematis, setiap pertanyaan memutuskan cabang berdasarkan nilai atribut dan berhenti di daun pohon yang merupakan prediksi dari variabel kelas (Dinas et al., 2018).

1. Perhatikan label datanya. Jika semuanya sama, nilai di seluruh label data akan membentuk lembaran.
2. Menghitung nilai informasi dengan menggunakan semua data yang ada dengan formula:

$$inf\ 0(D) = -\sum_{i=1}^{m} log^{2(pi)} \quad (1)$$

Dimana merupakan probabilitas tuple dalam D yang menjadi kelas dengan asumsi atau disebut juga entropy dari D merupakan rata rata informasi yang diperlukan untuk identifikasi tuple dalam D (Kusrini, 2009).

Jika nilai A adalah diskrit, data D dipisahkan oleh rangkaian nilai data A, sehingga nilai di setiap cabang murni dan serupa. Setelah cabang pertama, jumlah kemungkinan cabang diukur dengan rumus:

$$in\,f(D)\sum_{j}^{v}\frac{|Dj|}{|D|}x\,info\,A(Dj) \qquad (2)$$

3. Menghitung nilai informasi dengan formula.
4. Untuk setiap atribut dengan memperhatikan isi data dari atribut. dimana $\frac{|Dj|}{|D|}$ merupakan bobot dari partisi j. $onfo_A(D)$ merupakan informasi yang diperlukan untuk mengklasifikasikan tuple dari D pada partisi A. Semakin kecil hasil persamaan ini, semakin baik pula partisi yang dihasilkan. Nilai dari sebuah atribut menentukan penting tidaknya atribut tersebut dalam penyusunan pohon keputusan. Jika atribut bernilai kontinyu, maka akan dicari *split_point* dengan cara mengurutkan seluruh data menurut atribut tersebut dari kecil ke besar, lalu di rata-rata antar satu data dengan data setelahnya. Nilai informasi akan dihitung menurut satu persatu calon *split_point* dan nilai *split_point* yang akan dipilih yang terkecil. (4) Nilai gain untuk setiap atribut akan diperhitungkan dengan formula (2.3), nilai dengan gain tertinggi akan dijadikan cabang dalam pohon keputusan.

$$Gain(A) = info(D) - info(D) \qquad (3)$$

5. Setelah membentuk cabang-cabang pohon keputusan, perhitungan dilakukan kembali seperti pada tahap 1-4. Namun, begitu cabang mencapai cabang maksimum yang diizinkan, daun terbentuk dengan sebagian besar nilai data.

2.3.9 Metode Support Vector Machine

Support Vector Machine (*SVM*) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi (Santosa, 2007). Meskipun SVM memiliki prinsip dasar pengklasifikasi , yaitu kasus klasifikasi yang dapat dipisahkan secara linear, *SVM* dikembangkan untuk mengatasi masalah nonlinear dengan mengintegrasikan konsep kernel ke dalam ruang kerja berdimensi tinggi. Fungsi kernel yang digunakan untuk memetakan dimensi awal (dimensi yang lebih rendah) himpunan data ke dimensi baru (dimensi yang relatif lebih tinggi) (Santosa, 2007).

Menurut (Prasetyo, 2012) macam fungsi kernel diantaranya:

1. Kernel linear:

$$K(u,v) = uv^T \quad (4)$$

2. Kernel polynomial

$$K(u,v) = (1 + uv^T)^d, d \geq 2 \quad (5)$$

3. Kernel RBF (Radial Basis Function):

$$K(u,v) = \exp(-y||u - v||^2), y > 0 \quad (6)$$

4. Kernel Sigmoid:

$$K(x,y) = tan(\sigma(x_i, x_j) + c) \quad (7)$$

Menurut (Santosa, 2007) *hyperplane* klasifikasi linear SVM dinotasikan:

$$f(x) = w^T.x + b \quad (8)$$

Sehingga diperoleh persamaan:

$$[(W^T.x_i) + b] \geq 1 \; untuk \; y_i = +1 \quad (9)$$

$$[(W^T.x_i) + b] \leq untuk \; y_i = -1 \quad (10)$$

Dengan x_i = himpunan data *training*, i= 1,2,...n dan y_i = label kelas dari x_i untuk mendapatkan *hyperplane* terbaik adalah dengan mencari *hyperplane* yang terletak ditengah-tengah antara dua bidang pembatas kelas.

Mencari *hyperplane* terbaik dapat digunakan metode *Quadratic Programming* (QP) *Problem* yaitu meminimalkan $\frac{1}{2}w^T W$, dengan menggunakan fungsi *Lagrange Multiplier* yang telah ditransformasi sebagai berikut: (Santosa, 2007)

$$L(w, b, a) = \frac{1}{2}w^T W - \sum_{i=1}^{n} a_i y_i \, (w^T.x_i - b \sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i \quad (11)$$

berdasarkan persamaan 11, maka persamaan 11 menjadi sebagai berikut:

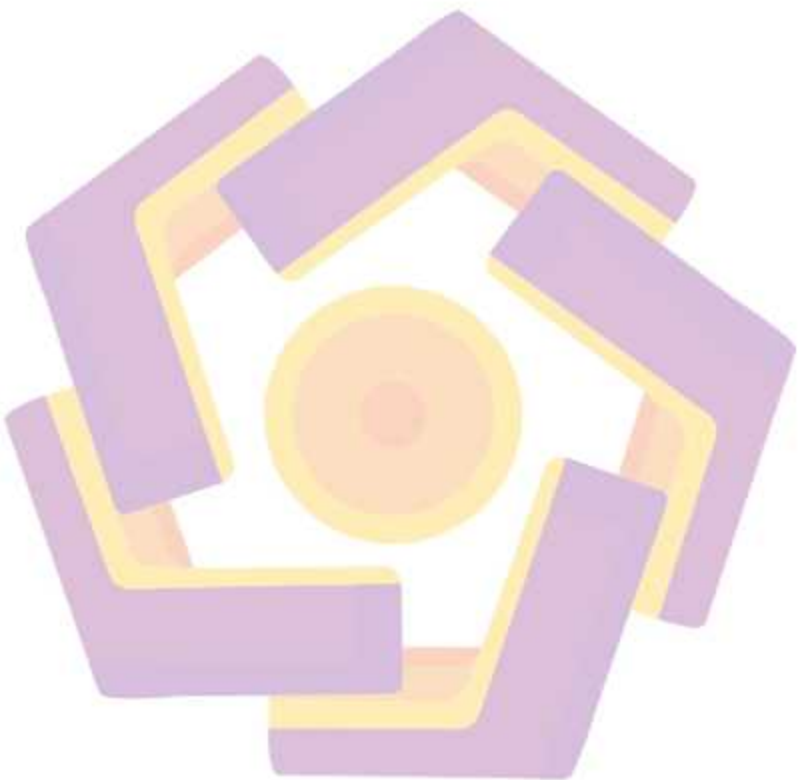
$$L_d = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j x_i^T x_j \quad (12)$$

dan diperoleh dual problem:

$$max_d L_d = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j x_i^T x_j \quad (13)$$

Dengan batasan $a_i \geq 0, i = 1,2, \ldots, n$ dan $\sum_{i=1}^{n} a_i y_i = 0$ data training dengan $a_i \geq 0$ terletak pada hyperplane disebut support vector, data training yang tidak terletak pada hyperplane tersebut mempunyai $a_i = 0$. Setelah solusi permasalahan quadratic progamming ditemukan (nilai a_i), maka kelas dari data

yang akan di prediksi atau data testing dapat ditentukan berdasarkan nilai fungsi tersebut (Santosa, 2007).



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian ini adalah penelitian eksperimental. Dimana penelitian ini melakukan pengujian efektivitas algorima *Random Forest* dan *SVM* dalam mendeteksi ulasan palsu dengan melakukan eksperimen dan membandingkan hasil klasifikasi dan label ulasan yang telah diketahui.

Penelitian ini berfsifat deskriptif, dimana penelitian yang menggambarkan, menjelaskan dan menjawab persoalan-persoalan tentang fenomena dan peristiwa yang terjadi saat ini (Muhammad Arsyam, 2021). Penelitian ini akan menggambarkan dan menjelaskan hasil dari pengujian-pengujian yang dilakukan pada dataset yang ada untuk dapat diketahui metode mana yang memiliki akurasi, presisi, recall, f1-score terbaik.

Penelitian ini menggunakan pendekatan kuantitatif. Untuk pendekatan kuantitatif dijelaskan oleh (Arikunto, 2013) bahwa pendekatan dengan menggunakan kuantitatif karena menggunakan angka, mulai dari pengumpulan data, penafsiran terhadap data tersebut, serta penampilan dari hasilnya. Dimana penelitian ini akan menghasilkan data numerik yang dapat diolah dan dianalisis secara statistic. Hasil penelitian ini akan mencakup angka-angka yang menggambarkan efektivitas algoritma *Random Forest* dan *SVM* dalam deteksi ulasan palsu, seperti akurasi, presisi, recall, f1-score.

3.2. Metode Pengumpulan Data

Metode yang digunakan dalam pengumpulan data yaitu metode sekunder. Menurut (Sugiyono, 2018) menyatakan bahwa data sekunder adalah sumber data yang diperoleh tidak langsung oleh peneliti, tetapi merupakan data yang sudah ada dan dikumpulkan oleh pihak lain sebelumnya. Dalam penelitian ini, data sekunder yang digunakan berasal dari berbagai sumber seperti buku referensi, jurnal penelitian, internet, dan sumber lainnya yang tersedia secara publik. Peneliti menggunakan data yang telah ada tersebut untuk mengumpulkan informasi yang relevan dengan tujuan penelitian, tanpa harus melakukan pengumpulan data primer secara langsung.

Dataset yang digunakan dalam penelitian ini adalah dataset ulasan hotel yang dikumpulkan oleh (Ott et al., 2013). Dataset ini terdiri dari total 1600 ulasan, dengan 800 ulasan palsu dan 800 ulasan asli.

3.3. Metode Analisis Data

Sebelum melakukan analisis data, langkah pertama yang dilakukan adalah preprocessing. Preprocessing ini melibatkan beberapa tahap seperti konversi data menjadi huruf kecil, pemisahan kata, penghapusan kata yang tidak memiliki arti, serta mengembalikan kata ke bentuk semula atau membuang imbuhan kata.

Setelah melakukan preprocessing, data akan dijalani beberapa tahap lebih lanjut. Pertama, dilakukan pembobotan kata menggunakan metode tfidf dan menambahkan konteks kata menggunakan n-gram. Selanjutnya, dilakukan seleksi fitur dengan menggunakan metode chi-square dan information gain.

Setelah fitur-fitur yang signifikan telah dipilih, langkah selanjutnya adalah melakukan klasifikasi menggunakan algoritma SVM (Support Vector Machine) dan Random Forest. Kedua algoritma ini akan digunakan untuk memprediksi kelas atau label dari data yang ada.

Setelah proses klasifikasi selesai, tahap terakhir adalah evaluasi performa untuk mendapatkan akurasi, presisi, recall, dan f1-score. Evaluasi ini akan memberikan pedoman dan acuan dalam menentukan hasil penelitian ini. Dengan melihat hasil pengukuran performa tersebut, peneliti dapat memahami sejauh mana model klasifikasi yang dibangun dapat menghasilkan prediksi yang akurat dan dapat diandalkan.

Dengan demikian, seluruh proses ini membentuk sebuah kerangka kerja yang komprehensif dalam melakukan analisis data dan mengevaluasi performa dari model klasifikasi yang telah dikembangkan dalam penelitian ini.

3.4. Dataset

Dalam penelitian ini, digunakan dataset ulasan hotel standar yang dikembangkan oleh (Ott et al., 2013). Dataset ini juga digunakan oleh (Rout et al., 2017) dan (Hassan & Islam, 2019). Pertama, (Ott et al., 2013) mengembangkan dataset yang hanya berisi ulasan palsu positif. Data ulasan palsu dikumpulkan menggunakan Amazon Mechanical Turk (AMT). Hotel-hotel yang menjadi target adalah 20 hotel paling populer di area Chicago, Amerika Serikat, yang terdaftar di situs TripAdvisor. Melalui platform AMT, manajer departemen pemasaran hotel-hotel tersebut meminta pengguna untuk menulis ulasan positif tentang hotel mereka. Sebanyak 400 ulasan palsu dengan sentimen positif berhasil dikumpulkan.

Selain itu, 400 ulasan jujur diambil dari beberapa situs seperti TripAdvisor dan Yelp. Hasil klasifikasi yang diperoleh bagus, namun dataset tersebut tidak seimbang karena tidak ada pengumpulan ulasan negatif yang menyesatkan. Kemudian, (Ott et al., 2013) memperbaiki dataset yang ada dengan mengumpulkan 400 ulasan menyesatkan dengan sentimen negatif. Selain itu, ditambahkan juga 400 ulasan jujur untuk menyeimbangkan datas.

Untuk proses pelabelan dilakukan dengan memperhatikan penggunaan bahasa spesial, Ulasan positif yang palsu cenderung kurang mendetail dalam hal bahasa spasial, seperti lantai, ukuran, atau lokasi, karena penulis ulasan tersebut mungkin tidak memiliki pengalaman langsung di hotel tersebut (Ott et al., 2013). Hal ini juga berlaku untuk ulasan negatif kami, dengan penggunaan bahasa spasial yang lebih sedikit pada ulasan negatif palsu dibandingkan dengan ulasan yang jujur. Demikian pula, ulasan negatif palsu kami memiliki lebih banyak kata kerja dibandingkan dengan kata benda daripada ulasan yang jujur, mengindikasikan gaya naratif yang lebih menggambarkan tulisan imajinatif.

Selain menggunakan bahasa special, emosi juga digunakan untuk mendeteksi ulasan palsu. para penulis ulasan negatif palsu lebih banyak menghasilkan kata-kata emosi negatif (misalnya, mengerikan, kecewa) dibandingkan dengan ulasan yang jujur dengan cara yang sama seperti para penulis ulasan positif palsu yang lebih banyak menghasilkan kata-kata emosi positif (misalnya, elegan, mewah). Jika digabungkan, data ini menunjukkan bahwa peningkatan frekuensi istilah emosi negatif dalam kumpulan data saat ini bukanlah hasil dari tekanan emosional yang muncul karena berbohong (Ott et al., 2013).

Sebaliknya, perbedaan-perbedaan ini menunjukkan bahwa para penulis ulasan hotel palsu membesar-besarkan sentimen yang ingin mereka sampaikan dibandingkan dengan ulasan yang jujur dengan valensi serupa.

Pola frekuensi kata ganti tidak sama antara ulasan positif dan negatif. Secara khusus, sementara kata ganti orang pertama tunggal diproduksi lebih sering dalam ulasan palsu daripada yang jujur, sesuai dengan kasus ulasan positif, peningkatan ini berkurang pada ulasan negatif yang dianalisis di sini. Dalam ulasan positif yang dilaporkan oleh (Ott et al., 2011). tingkat kata ganti orang pertama tunggal dalam ulasan palsu (rerata=4,36%, deviasi standar=2,96%) dua kali lipat dari tingkat yang diamati dalam ulasan jujur (rerata=2,18%, deviasi standar=2,04%). Sebaliknya, tingkat kata ganti orang pertama tunggal dalam ulasan negatif yang menipu (rerata=4,47%, deviasi standar=2,83%) hanya 57% lebih tinggi daripada ulasan yang jujur (rerata=2,85%, deviasi standar=2,23%). Hasil ini menunjukkan bahwa penekanan pada diri sendiri, mungkin sebagai strategi untuk meyakinkan pembaca bahwa penulis sebenarnya pernah ke hotel, tidak sejelas dalam ulasan negatif palsu, mungkin karena nada negatif ulasan membuat para penulis secara psikologis menjauhkan diri dari pernyataan negatif mereka, sebuah fenomena yang diamati dalam beberapa penelitian penipuan lainnya, misalnya, (Hancock et al., 2008)

Tabel 3.1 Sample Dataset

No	Text
0	We stayed for a one night getaway with family on a thursday. Triple AAA rate of 173 was a steal. 7th floor room complete with 44in plasma TV bose stereo, voss and evian water, and gorgeous bathroom(no tub but was fine for us) Concierge was very helpful. You cannot beat this location... Only flaw was breakfast was pricey and service was very very slow(2hours for four kids and four adults on a friday morning) even though there were only two other tables in the restaurant. Food was very good so it was worth the wait. I would return in a heartbeat. A gem in chicago...
1	Triple A rate with upgrade to view room was less than $200 which also included breakfast vouchers. Had a great view of river, lake, Wrigley Bldg. & Tribune Bldg. Most major restaurants, Shopping, Sightseeing attractions within walking distance. Large room with a very comfortable bed.
2	This comes a little late as I'm finally catching up on my reviews from the past several months:) A dear friend and I stayed at the Hyatt Regency in late October 2007 for one night while visiting a friend and her husband from out of town. This hotel is perfect, IMO. Easy check in and check out. Lovely, clean, comfortable rooms with great views of the city. I know this area pretty well and it's very convenient to many downtown Chicago attractions. We had dinner and went clubing with our friends around Division St.. We had no problems getting cabs back and forth to the Hyatt and there's even public transportation right near by but we didn't bother since we only needed cabs from and to the hotel. Parking, as is usual for Chicago, was expensive but we were able to get our car out quickly (however, we left on a Sunday morning, not exactly a high traffic time although it was a Bears homegame day, so a bit busier than usual I would think). No problems at all and the best part is that we got a rate of $100 through Hotwire, a downright steal for this area of Chicago and the quality of the hotel.
3	The Omni Chicago really delivers on all fronts, from the spaciousness of the rooms to the helpful staff to the prized location on Michigan Avenue. While this address in Chicago requires a high level of quality, the Omni delivers. Check in for myself and a whole group of people with me was under 3 minutes, the staff had plentiful recommendations for dining and events, and the rooms are some of the largest you'll find at this price range in Chicago. Even the 'standard' room has a separate living area and work desk. The fitness center has free weights, weight machines, and two rows of cardio equipment. I shared the room with 7 others and did not feel cramped in any way! All in all, a great property!
4	I asked for a high floor away from the elevator and that is what I got. The room was pleasantly decorated, functional and very clean. I didn't need a whole lot of service but when I did they were pleasant and prompt. I used the fitness center which was well equipped and everything was in working order. It is in a great location at one end of the Michigan Avenue shopping district.
1595	Problems started when I booked the InterContinental Chicago online at the hotel's site, and got a server error. Somehow I managed to get my reservation, and wish I had looked elsewhere on this great Chicago street, filled with plenty of other options. The server errors continued at check in and didn't let up. I was on a holiday weekend, but far too many others must have been pushy conventioneers, and I had trouble getting served. The hotel is enormous and doesn't give a sense of comfort. The health facilities are also just too big, so noisy and crowded. And Internet access was $18 a day, kind of surprising when I can go around the corner to a Starbucks! Next time, I'll pick a more intimate place nearby--without so many 'server errors'.

Tabel 3.1 Lanjutan Sample Dataset

No	Text
1596	The Amalfi Hotel has a beautiful website and interior decorating, but that's about it. When my wife and I got here, we were given keys to a room that had not even been cleaned! The Internet access promised on the hotel's website was down, so I couldn't catch up on any of the business I had intended to do, and my wife thought that the dark design details in the room made her feel claustrophobic, like she was sleeping inside a Salvador Dali painting. All in all, this hotel was not worth the money, especially since we spent most of our time enjoying the city--something more casual and comfortable would have been better, and it would probably have been cleaner, too!

Tahap selanjutnya dilakukan proses proprocessing dimana proses ini terdapat 4 tahap yaitu: case folding, tokenizing, filtering/stopword, stemming. Untuk lebih jelasnya alur proses preprocessing dapat dilihat pada gambar 3.3.

a. Case folding adalah proses konversi teks menjadi huruf kecil semua. Dalam penelitian ini, kami menerapkan case folding untuk memastikan konsistensi dan keseragaman dalam teks. Dengan melakukan case folding, kami mengubah semua huruf menjadi huruf kecil tanpa mengubah konten atau makna dari teks tersebut. Hal ini memungkinkan kami untuk mengurangi kompleksitas dan mempermudah proses analisis dan pemrosesan selanjutnya.
b. Tokenizing adalah proses memecah teks menjadi unit-unit yang lebih kecil yang disebut "token". Token bisa berupa kata, frasa, atau karakter tergantung pada aturan yang digunakan dalam proses tokenizing. Dalam penelitian ini, kami menggunakan tokenizing untuk memisahkan teks menjadi kata-kata individual. Dengan membagi teks menjadi token-token, kami dapat menganalisis dan memproses setiap kata secara terpisah. Ini memungkinkan kami untuk melakukan penghitungan statistik, analisis sentimen, atau pemodelan lainnya yang membutuhkan pemrosesan kata per kata.
c. Stopword removal adalah proses penghapusan kata-kata umum yang tidak memberikan kontribusi signifikan dalam pemahaman konten teks. Kata-kata tersebut seringkali merupakan kata-kata penghubung (seperti "dan", "atau", "juga") atau kata-kata yang sering muncul dalam bahasa tertentu tanpa memberikan makna khusus. Dalam pengolahan teks, penghapusan stopword dilakukan untuk memperbaiki akurasi dan efisiensi analisis teks. Misalnya,

dalam kalimat "Saya pergi ke toko dan membeli beberapa barang", kata-kata "saya", "ke", "dan", "beberapa" dapat dianggap sebagai stopword. Dalam proses stopword removal, kata-kata tersebut akan dihapus sehingga kalimat menjadi "pergi toko membeli barang". Tujuan dari stopword removal adalah untuk meningkatkan relevansi dan interpretasi hasil analisis teks. Dengan menghilangkan kata-kata yang tidak memiliki makna khusus, fokus analisis dapat lebih tertuju pada kata-kata yang memiliki kontribusi penting dalam pemahaman konten teks.

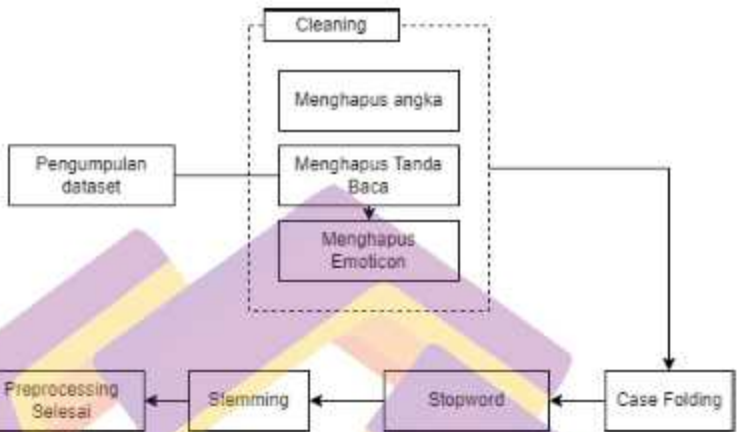
d. Stemming Porter merupakan metode dalam pengolahan teks yang digunakan untuk mengubah kata-kata menjadi kata dasar atau kata dasar yang serupa. Metode ini didasarkan pada algoritma Porter yang mengikuti aturan linguistik untuk menghilangkan imbuhan dan menerapkan pemangkasan pada kata-kata. Contoh penerapan Stemming Porter: Kata "menggunakan" akan diubah menjadi "guna" Kata "berlari" akan diubah menjadi "lari" Kata "menulis" akan diubah menjadi "tulis" Kata "bermain" akan diubah menjadi "main". Dengan menggunakan Stemming Porter, variasi kata-kata yang memiliki akar kata yang sama dapat direduksi menjadi bentuk kata dasar. Hal ini membantu dalam meningkatkan konsistensi dan keseragaman dalam pemrosesan teks, serta mengurangi dimensi fitur dalam analisis teks.

Tabel 3. 2 Proses Preprocessing

No	Text	Case Folding	Tokenizing	Stopword	Stemming
1	We stayed for a one night getaway with family on a thursday. Triple AAA rate of 173 was a steal. 7th floor room complete with 44in plasma TV bose stereo, voss and evian water, and gorgeous bathroom(no tub but was fine for us) Concierge was very helpful. You cannot beat this location... Only flaw was breakfast was pricey and service was very very slow(2hours for four kids and four adults on a friday morning) even though there were only two other tables in the restaurant. Food was very good so it was worth the wait. I would return in a heartbeat. A gem in chicago...	we stayed for a one night getaway with family on a thursday triple aaa rate of was a steal th floor room complete with in plasma tv bose stereo voss and evian water and gorgeous bathroomno tub but was fine for us concierge was very helpful you cannot beat this location only flaw was breakfast was pricey and service was very very slowhours for four kids and four adults on a friday morning even though there were only two other tables in the restaurant food was very good so it was worth the wait i would return in a heartbeat a gem in chicago	['we','stayed','for','a','one','night','getaway','with','family','on','a','thursday','triple','aaa','rate','of','was','a','steal','th','floor','room','complete','with','in','plasma','tv','bose','stereo','voss','and','evian','water','and','gorgeous','bathroomno','tub','but','was','fine','for','us','concierge','was','very','helpful','you','can','not','beat','this','location','only','flaw','was','breakfast','was','pricey','and','service','was','very','very','slowhours','for','four','kids','and','four','adults','on','a','friday','morning','even','though','there','were','only','two','other','tables','in','the','restaurant','food','was','very','good','so','it','was','worth','the','wait','i','would','return','in','a','heartbeat','a','gem','in','chicago']	['stayed','one','night','getaway','family','thursday','triple','aaa','rate','steal','th','floor','room','complete','plasma','tv','bose','stereo','voss','evian','water','gorgeous','bathroomno','tub','fine','us','concierge','helpful','beat','location','flaw','breakfast','pricey','service','slowhours','four','kids','four','adults','friday','morning','even','though','two','tables','restaurant','food','good','worth','wait','would','return','heartbeat','gem','chicago']	['stay','one','night','getaway','famili','thursday','tripl','aaa','rate','steal','th','floor','room','complet','plasma','tv','bose','stereo','voss','evian','water','gorgeou','bathroomno','tub','fine','us','concierg','help','beat','locat','flaw','breakfast','pricey','servic','slowhour','four','kid','four','adult','friday','morn','even','though','two','tabl','restaur','food','good','worth','wait','would','return','heartbeat','gem','chicago']
2	Triple A rate with upgrade to view room was less than $200 which also included breakfast vouchers. Had a great view of river, lake, Wrigley Bldg. & Tribune Bldg. Most major restaurants, Shopping, Sightseeing attractions within walking distance. Large room with a very comfortable bed.	triple a rate with upgrade to view room was less than which also included breakfast vouchers had a great view of river lake wrigley bldg tribune bldg most major restaurants shopping sightseeing attractions within walking distance large room with a very comfortable bed	['triple','a','rate','with','upgrade','to','view','room','was','less','than','which','also','included','breakfast','vouchers','had','a','great','view','of','river','lake','wrigley','bldg','tribune','bldg','most','major','restaurants','shopping','sightseeing','attractions','within','walking','distance','large','room','with','a','very','comfortable','bed']	['triple','rate','upgrade','view','room','less','also','included','breakfast','vouchers','great','view','river','lake','wrigley','bldg','tribune','bldg','major','restaurants','shopping','sightseeing','attractions','within','walking','distance','large','room','comfortable','bed']	['tripl','rate','upgrad','view','room','less','also','includ','breakfast','voucher','great','view','river','lake','wrigley','bldg','tribun','bldg','major','restaur','shop','sightse','attract','within','walk','distanc','larg','room','comfort','bed']

Tabel 3.2 Lanjutan Proses Preprocessing

No	Text	Case Folding	Tokenizing	Stopword	Stemming
3	This comes a little late as I'm finally catching up on my reviews from the past several months:) A dear friend and I stayed at the Hyatt Regency in late October 2007 for one night while visiting a friend and her husband from out of town. This hotel is perfect, IMO. Easy check in and check out. Lovely, clean, comfortable rooms with great views of the city. I know this area pretty well and it's very convenient to many downtown Chicago attractions. We had dinner and went clubing with our friends around Division St.. We had no problems getting cabs back and forth to the Hyatt and there's even public transportation right near by but we didn't bother since we only needed cabs from and to the hotel. Parking, as is usual for Chicago, was expensive but we were able to get our car out quickly (however, we left on a Sunday morning, not exactly a high traffic time although it was a Bears homegame day, so a bit busier than usual I would think).	this comes a little late as im finally catching up on my reviews from the past several months a dear friend and i stayed at the hyatt regency in late october for one night while visiting a friend and her husband from out of town this hotel is perfect imo easy check in and check out lovely clean comfortable rooms with great views of the city i know this area pretty well and its very convenient to many downtown chicago attractions we had dinner and went clubing with our friends around division st we had no problems getting cabs back and forth to the hyatt and theres even public transportation right near by but we didnt bother since we only needed cabs from and to the hotel parking as is usual for chicago was expensive but we were able to get our car out quickly however we left on a sunday morning not exactly a high traffic time although it was a bears homegame day so a bit busier than usual i would think	['this','comes','a','little','late','as','im','finally','catching','up','on','my','reviews','from','the','past','several','months','a','dear','friend','and','i','stayed','at','the','hyatt','regency','in','late','october','for','one','night','while','visiting','a','friend','and','her','husband','from','out','of','town','this','hotel','is','perfect','imo','easy','check','in','and','check','out','lovely','clean','comfortable','rooms','with','great','views','of','the','city','i','know','this','area','pretty','well','and','its','very','convenient','to','many','downtown','chicago','attractions','we','had','dinner','and','went','clubing','with','our','friends','around','division','st','we','had','no','problems','getting','cabs','back','and','forth','to','the','hyatt','and','theres','even','public','transportation','right','near','by','but','we','didnt','bother','since','we','only','needed','cabs','from','and','to','the','hotel','parking','as','is','usual','for','chicago','was','expensive','but','we','were','able','to','get','our','car','out','quickly','however','we','left','on','a','sunday','morning','not','exactly','a','high','traffic','time','although','it','was','a','bears','homegame','day','so','a','bit','busier','than','usual','i','would','think']	['comes','little','late','im','finally','catching','reviews','past','several','months','dear','friend','stayed','hyatt','regency','late','october','one','night','visiting','friend','husband','town','hotel','perfect','imo','easy','check','check','lovely','clean','comfortable','rooms','great','views','city','know','area','pretty','well','convenient','many','downtown','chicago','attractions','dinner','went','clubing','friends','around','division','st','problems','getting','cabs','back','forth','hyatt','theres','even','public','transportation','right','near','didnt','bother','since','needed','cabs','hotel','parking','usual','chicago','expensive','able','get','car','quickly','however','left','sunday','morning','exactly','high','traffic','time','although','bears','homegame','day','bit','busier','usual','would','think']	['come','littl','late','im','final','catch','review','past','sever','month','dear','friend','stay','hyatt','regenc','late','octob','one','night','visit','friend','husband','town','hotel','perfect','imo','easi','check','check','love','clean','comfort','room','great','view','citi','know','area','pretti','well','convent','mani','downtown','chicago','attract','dinner','went','clube','friend','around','divis','st','problem','get','cab','back','forth','hyatt','there','even','public','transport','right','near','didnt','bother','since','need','cab','hotel','park','usual','chicago','expens','abl','get','car','quickli','howev','left','sunday','morn','exactli','high','traffic','time','although','bear','homegam','day','bit','busier','usual','would','think"]

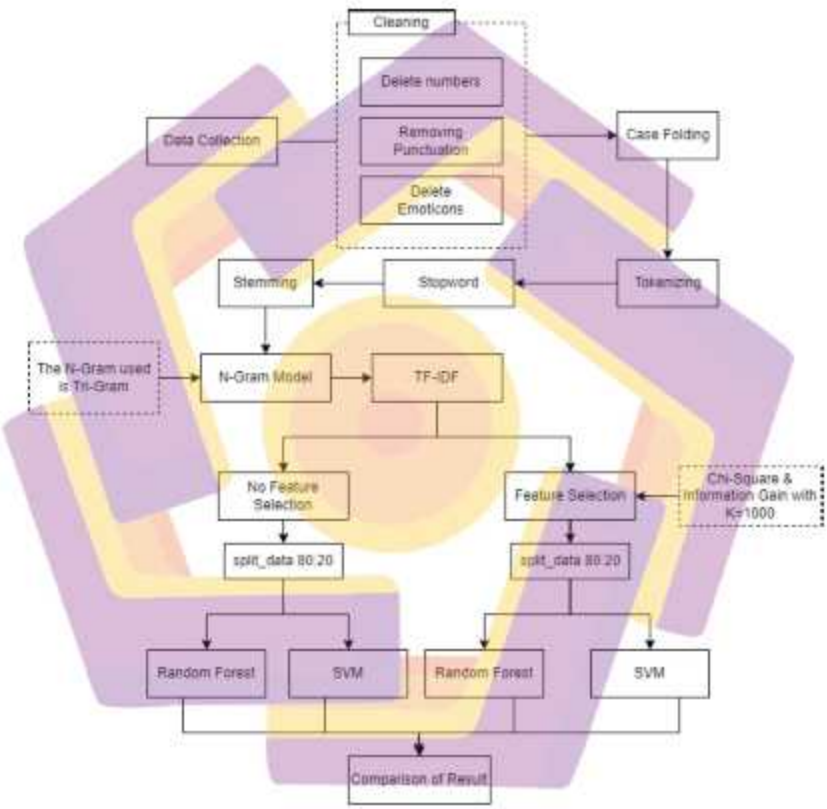


Gambar 3.1 Alur Processing Data

3.5. Alur Penelitian

Pada penelitian ini terdapat beberapa tahapan penelitian diantaranya; Tahapan pertama adalah preprocessing data, di mana data yang diperoleh akan melalui proses case folding, tokenizing, stopword removal, dan stemming Porter untuk menghasilkan teks yang lebih terstruktur. Selanjutnya, dilakukan ekstraksi fitur menggunakan metode TF-IDF untuk mengidentifikasi kata-kata penting dalam teks. Setelah itu, dilakukan feature selection dengan menggunakan algoritma Information Gain dan Chi-Square untuk memilih fitur-fitur yang paling relevan dan memiliki pengaruh signifikan dalam klasifikasi. Tahap berikutnya adalah classification, di mana algoritma SVM akan digunakan untuk mengklasifikasikan teks menjadi kategori yang tepat berdasarkan fitur-fitur yang telah diekstraksi. Evaluasi dilakukan untuk mengukur kinerja model klasifikasi, termasuk akurasi, presisi, recall, dan F1-

score. Pada akhirnya, kesimpulan akan diambil berdasarkan hasil evaluasi dan menunjukkan keberhasilan atau kekurangan model yang telah dikembangkan dalam penelitian ini.



Gambar 3.2 Alur Penelitian

Penelitian ini dimulai dengan melakukan pengumpulan data menggunakan dataset hotel yang diambil dari website https://www.kaggle.com/ dengan keyword "Deceptive Opinion Spam Corpus". Dataset ini dikembangkan oleh Ott, Cardie, dan Hancock pada

tahun 2013. Dataset ini telah menjadi sumber data yang populer dan telah digunakan oleh beberapa peneliti, seperti Rout et al. pada tahun 2017 dan Hassan dan Islam pada tahun 2019.

Tahapan selanjutnya adalah dilakukan penambahan konteks kata menggunakan tri-gram untuk membantu mengidentifikasi pola kata atau frase yang sering muncul dalam ulasan palsu dan membedakannya dari ulasan asli, dengan mempertimbangkan konteks kata-kata secara keseluruhan. Tujuan dari penggunaan n-gram adalah untuk memahami konteks dan pola yang muncul dalam teks. Dengan membagi teks menjadi potongan berurutan berisi n kata atau karakter, n-gram membantu mengidentifikasi kombinasi kata yang sering muncul bersama

Selanjutnya dilakukan proses pembobotan kan menggunakan tf-idf. Tujuan dari proses ini adaah untuk mendapatkan tingkat kepentingan atau bobot kata dalam sebuah dokumen. Dalam sebuah dokumen, ada banyak kata yang muncul berulang kali. Namun, tidak semua kata memiliki tingkat kepentingan yang sama dalam sebuah dokumen. TFIDF dapat membantu menentukan kata yang paling penting dalam dokumen atau korpus dengan memberi bobot lebih pada kata-kata yang jarang muncul di badan tetapi sering muncul di dokumen. Penggunaan TF-IDF dalam analisis teks dapat membantu meningkatkan akurasi dan akurasi klasifikasi dokumen, analisis sentimen, pengecek keaslian dokumen, dan pencarian informasi. Alasan penggunaan TF-IDF ini adalah dikarenakan memiliki proses yang lebih cepat bahkan dengan menggunakan sumber daya yang terbatas TF-IDF

dapat menghitung dengan cepat berbeda dengan algoritma word vector yang memerlukan sumber daya komputasi yang sangat besar.

Setelah proses ekstraksi data menggunakan tf-idf dan n-gram, langkah selanjutnya adalah melakukan feature selection. feature selection merupakan peroses pengurangan jumlah fitur sehingga dapat mempercepat proses klasifikasi (Saskia et al. n.d.). Pada tahapan ini, kami melakukan seleksi fitur untuk memilih subset fitur yang paling relevan dan berkontribusi dalam klasifikasi atau analisis data. Tujuannya adalah untuk mengurangi dimensi data dan mempertahankan fitur yang paling informatif. Dengan menggunakan metode feature selection, kami dapat meningkatkan efisiensi pemrosesan data, menghindari overfitting, dan mengoptimalkan kinerja model yang digunakan. feature selection yang akan digunakan adalah chi-square dan information gain. dimana kedua algoritma ini akan menggunakan 1000 fitur teratas atau K=1000. Selection feature ini nantinya akan mengambil hanya 1000 fitur dari jumlah fitur yang dihasilkan pada tahap pembobotan kata

Sebelum melakukan klasifikasi. Tahapan yang harus dilakukan adalah pembagian dataset menjadi data training dan data testing, dimana data training dan testing ini menggunakan rasio 80 untuk data training dan 20 data testing. Alasan penggunaan rasio 80 traning dan 20 testing ini adalah dikarenakan penelitian-penelitian sebelumnya juga menggunakan rasio ini.

Algoritma klasifikasi adalah algoritma yang digunakan untuk mengelompokkan data ke dalam sekumpulan kategori atau kelas

berdasarkan karakteristik atau atribut yang ada pada data. Algoritma klasifikasi digunakan dalam berbagai aplikasi seperti deteksi spam email, klasifikasi dokumen, pengenalan wajah dan banyak lainnya. Ada banyak Teknik berberda untuk melakukan klasifikasi *NB, DT-J48, SVM, K-NN, Neural Networks*, dan *Genetic Algorithm*. Dalam penelitian ini peneliti menggunakan *Random Forest* dan *SVM* untuk mengklasikasi ulasan apakah termasuk ulasan palsu atau asli.

Evaluasi performa digunakan untuk mengukur seberapa baik model klasifikasi dapat melakukan prediksi terhadap data yang belum pernah dilihat sebelumnya (data test). Beberapa metrik evaluasi performa umum yang digunakan dalam klasifikasi antara lain:

1. Akurasi (*Accuracy*): Akurasi mengukur seberapa sering model melakukan prediksi dengan benar. Akurasi dihitung sebagai rasio antara jumlah prediksi benar dengan total jumlah data. Formula akurasi: (TP+TN)/(TP+TN+FP+FN)
2. Presisi (*Precision*): Presisi mengukur seberapa sering prediksi positif (klasifikasi sebagai kelas positif) yang dilakukan oleh model benar. Formula presisi: TP/(TP+FP)
3. *Recall* (*Sensitivity*): *Recall* mengukur seberapa sering model dapat mengklasifikasikan kelas positif secara benar. Formula *recall*: TP/(TP+FN)
4. *F1-Score: F1-Score* adalah *harmonic mean* dari *precision* dan *recall*. *F1-Score* mengukur keseimbangan antara *precision* dan

recall. *Formula* *F1-Score*:

2*(*Precision***Recall*)/(*Precision*+*Recall*)

Ketika menggunakan evaluasi performa untuk membandingkan antara model yang berbeda, sangat penting untuk memperhatikan beberapa metrik evaluasi performa, karena suatu model dapat memberikan performa yang baik pada satu metrik namun buruk pada metrik lainnya

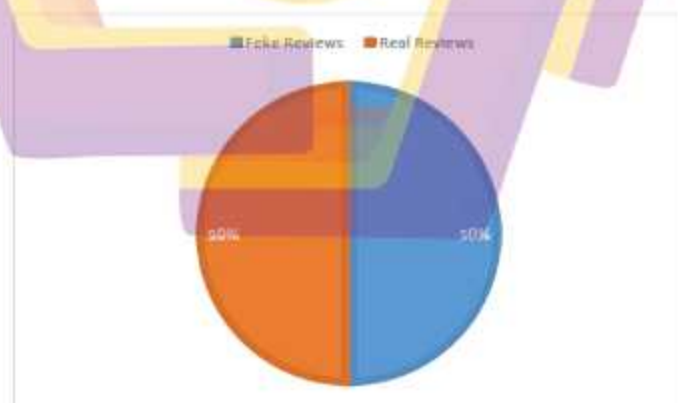
Tahapan terkahir pada penelitian ini mendapatkan hasil perbandingan antara dua metode algoritma untuk mendeteksi ulasan palsu, yaitu algortima *Random Forest* dan *Support Vector Machine*

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Dataset

Berdasarkan hasil pengumpulan data yang diambil dari website Kaggle.com dengan menggunakan keyword "Deceptive Opinion Spam Corpus", diketahui bahwa jumlah dataset ini mencapai 1600 ulasan. Dataset ini terdiri dari dua kategori utama, yaitu truthful (asli) dan deceptive (palsu), yang membedakan keabsahan ulasan tersebut. Menariknya, dataset ini telah banyak digunakan oleh beberapa peneliti, antara lain Rout et al. (2017) dan Hassan & Islam (2019), yang menunjukkan kepercayaan dan validitas dataset ini dalam penelitian terkait. Untuk memberikan gambaran yang lebih jelas, dataset dapat divisualisasikan melalui diagram yang terdapat pada gambar 4.1. Diagram tersebut menunjukkan persentase ulasan dalam dataset secara grafis.



Gambar 4.1 Persentase Jumlah Ulasan Palsu dan Asli

Pada gambar 4.1, ulasan palsu direpresentasikan menggunakan warna biru dengan jumlah 800 ulasan, yang merupakan 50% dari total ulasan. Di sisi lain, ulasan asli direpresentasikan dengan warna orange dan memiliki jumlah yang sama, yaitu 800 ulasan atau 50% dari total ulasan. Hal ini dilakukan untuk membedakan dengan jelas antara ulasan palsu dan ulasan asli dalam visualisasi tersebut.

4.2. Pembobotan Dataset

Tahapan penelitian ini dilakukan setelah proses pelabelan dataset selesai dilakukan. Proses pembobotan dataset dilakukan dengan dua cara yaitu: tanpa model n-gram dan dengan model n-gram.

4.2.1 Pembobotan Dataset Tanpa Model N-Gram.

Dalam metode pengolahan teks, salah satu teknik yang sering digunakan adalah TF-IDF. TF-IDF menghitung bobot kata berdasarkan frekuensi kemunculan kata dalam dokumen dan frekuensi kemunculan kata dalam seluruh koleksi dokumen. Bobot TF-IDF dapat digunakan untuk menunjukkan pentingnya kata dalam teks. Berikut ini ada gambar kode dari TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd

# Inisialisasi objek TfidfVectorizer
vectorizer = TfidfVectorizer()

# Menghitung nilai TF-IDF
tfidf = vectorizer.fit_transform(x)

# Mendapatkan daftar kata kunci yang digunakan sebagai fitur
feature_names = vectorizer.get_feature_names_out()

# Membuat DataFrame dari matriks TF-IDF
df = pd.DataFrame(tfidf.toarray(), columns=feature_names)

# Menampilkan DataFrame
df
```

Gambar 4.2 Kode Pembobotan Kata Dengan TF-IDF

Pada gambar 4.2, dapat dilihat proses dari pembobotan kata dengan tahapan pertama adalah dengan mengimpor modul "TfidfVectorizer" dari pustaka "sklearn.feature_extraction.text" yang digunakan untuk menghitung nilai TF-IDF. Kemudian, dilakukan inisialisasi objek "TfidfVectorizer" dengan "vectorizer = TfidfVectorizer()". Objek ini akan digunakan untuk menghitung nilai TF-IDF. Selanjutnya menghitung nilai TF-IDF dengan menggunakan "tfidf = vectorizer.fit_transform(x)", di mana "x" merupakan data teks yang akan dihitung TF-IDF-nya. Fungsi "fit_transform" akan menghasilkan matriks TF-IDF berdasarkan data teks yang diberikan. Setelah itu, kita mendapatkan daftar kata kunci yang digunakan sebagai fitur dalam matriks TF-IDF dengan menggunakan "feature_names = vectorizer.get_feature_names_out()." Fungsi "get_feature_names_out()" akan mengembalikan daftar kata kunci yang digunakan dalam proses transformasi TF-IDF. Selanjutnya, kita membuat DataFrame dari matriks TF-IDF dengan menggunakan "df = pd.DataFrame(tfidf.toarray(), columns=feature_names)". Fungsi "toarray()" mengonversi matriks TF-IDF menjadi "array numpy", dan kemudian kita membuat DataFrame dari array tersebut dengan kolom yang sesuai dengan daftar kata kunci. Dimana Dataframe merupakan tabel/data tabular dengan array dua dimensi yaitu baris dan kolom. Struktur data ini merupakan cara paling standar untuk menyimpan data. Terakhir, kita menampilkan DataFrame yang berisi nilai-nilai TF-IDF dengan menggunakan "print(df)". Ini akan mencetak DataFrame ke output. Berikut adalah hasil dari proses pembobotan kata.

Tabel 4.1 Output Dari Pembobotan Kata dengan TF-IDF

	aaa	aaah	aback	abaisador	abd	abil	abl	abound	abrupt	abruptli	. . .
0	0.181465	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
2	0.000000	0.0	0.0	0.0	0.0	0.0	0.07995	0.0	0.0	0.0	...
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1595	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
1596	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
1597	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
1598	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...
1599	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	...

Setelah melakukan proses ekstraksi fitur dari kumpulan data yang besar, pada Table 4.1 menghasilkan 7201 kolom yang masing-masing berisi informasi penting tentang karakteristik objek yang sedang dianalisis

4.2.2 Pembobotan Dataset Dengan Model N-Gram

TF-IDF dan n-gram adalah dua teknik umum yang digunakan dalam tugas klasifikasi teks. TF-IDF adalah singkatan dari term frequency-inverse document frequency. Ini adalah statistik numerik yang mencerminkan betapa pentingnya sebuah kata bagi dokumen dalam koleksi atau korpus. Ini sering digunakan sebagai fitur untuk tugas klasifikasi teks, seperti SVM dan Random Forest. Sedangkan N-gram adalah urutan n item yang berdekatan dari sampel teks atau ucapan yang diberikan. Ini digunakan untuk merepresentasikan teks sebagai vektor fitur untuk tugas klasifikasi teks. N-gram dapat digunakan sebagai fitur SVM dan Random Forest untuk meningkatkan kinerjanya. Dalam analisis bahasa alami dan pemodelan bahasa, trigram sering digunakan sebagai unit dasar untuk menggambarkan struktur dan hubungan antar kata. Dengan memperhatikan konteks tiga kata sekaligus, trigram dapat memberikan informasi yang lebih kaya tentang makna dan struktur kalimat. Misalnya, dengan menggunakan trigram, kita dapat menangkap pola dan

relasi seperti frasa yang sering muncul bersama, urutan kata yang umum, dan ketergantungan sintaksis antara tiga kata tersebut. Ini dapat membantu dalam tugas seperti pemrosesan bahasa alami, pemodelan topik, pemahaman kalimat, dan masih banyak lagi. Namun, perlu diingat bahwa penggunaan trigram juga dapat menyebabkan peningkatan kompleksitas komputasional karena jumlah kombinasi trigram yang mungkin jauh lebih banyak dibandingkan dengan unigram atau bigram. Oleh karena itu, dalam penggunaan trigram, diperlukan penyesuaian yang tepat untuk mengatasi masalah tersebut. Dalam rangka memanfaatkan trigram, biasanya dilakukan tokenisasi kalimat menjadi kata-kata individu dan kemudian membangun model n-gram berdasarkan urutan tiga kata yang terjadi dalam teks. Model ini dapat digunakan untuk analisis teks lebih lanjut, termasuk analisis sentimen, prediksi kata berikutnya, dan tugas-tugas lain yang melibatkan pemahaman konteks dan ketergantungan antar kata. Jadi, menggunakan trigram dapat membantu kita dalam menangkap lebih banyak informasi tentang konteks dan hubungan antar kata dalam kalimat (Violos et al., 2018). Berikut ini ada gambar kode dari TF-IDF dan Model N-Gram

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Inisialisasi objek TfidfVectorizer dengan trigram
vectorizer = TfidfVectorizer(ngram_range=(1, 3))

# Menghitung nilai TF-IDF dengan trigram
tfidf_matrix = vectorizer.fit_transform(x)

# Mendapatkan daftar trigram yang digunakan sebagai fitur
feature_names = vectorizer.get_feature_names_out()

df = pd.DataFrame(tfidf_matrix.toarray(), columns=feature_names)

# Menampilkan DataFrame
df
```

Gambar 4.3 Kode Pembobotan Kata Dengan TF-IDF dan Model N-Gram

Pada gambar 4.4, dapat dilihat proses dari pembobotan kata dengan tahapan pertama adalah dengan mengimpor modul "TfidfVectorizer" dari pustaka "sklearn.feature_extraction.text" yang digunakan untuk menghitung nilai TF-IDF. Kemudian, dilakukan inisialisasi objek "TfidfVectorizer" dengan "vectorizer = TfidfVectorizer(n-gram=(1,3)) Dalam kasus ini, kita mengatur parameter "n-gram_range" menjadi "(1, 3)", yang berarti kita akan menggunakan trigram sebagai fitur dalam perhitungan TF-IDF. Trigram adalah urutan tiga kata yang saling berdekatan dalam teks. Selanjutnya menghitung nilai TF-IDF dengan menggunakan "tfidf = vectorizer.fit_transform(x)", di mana "x" merupakan data teks yang akan dihitung TF-IDF-nya. Fungsi "fit_transform" akan menghasilkan matriks TF-IDF berdasarkan data teks yang diberikan. Setelah itu, kita mendapatkan daftar kata kunci yang digunakan sebagai fitur dalam matriks TF-IDF dengan menggunakan "feature_names = vectorizer.get_feature_names_out()." Fungsi "get_feature_names_out()" akan mengembalikan daftar kata kunci yang digunakan dalam proses transformasi TF-IDF. Selanjutnya, kita membuat DataFrame dari matriks TF-IDF dengan menggunakan "df = pd.DataFrame(tfidf.toarray(), columns=feature_names)". Fungsi "toarray()" mengonversi matriks TF-IDF menjadi "array numpy", dan kemudian kita membuat DataFrame dari array tersebut dengan kolom yang sesuai dengan daftar kata kunci. Terakhir, kita menampilkan DataFrame yang berisi nilai-nilai TF-IDF dengan menggunakan "print(df)". Ini akan mencetak DataFrame ke output. Berikut adalah hasil dari proses pembobotan kata.

Tabel 4.2 Output Dari Pembobotan Kata dengan TF-IDF dan Model N-gram

	aaa	aaa card	aaa card delight	aaa discount	aaa discount paid	aaa member	aaa member plenti	aaa member would	aaa membership	aaa membership get	...
0	0.074867	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1595	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1596	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1597	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1598	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1599	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

Pada Table 4.2, terdapat perbedaan hasil dari pembobotan kata dengan TF-IDF dan TF-IDF dengan n-gram, di mana jumlah kolom pada TF-IDF dengan n-gram mencapai 186611 kolom, sedangkan dengan TF-IDF saja hanya mendapatkan 7201 kolom. Perbedaan ini disebabkan oleh penggunaan n-gram yang tidak hanya mempertimbangkan kata secara individual, tetapi juga kombinasi kata-kata yang berdekatan dalam rangkaian n buah kata. Dengan memasukkan kombinasi kata ini sebagai fitur tambahan, ruang fitur menjadi lebih kaya dan kompleks, sehingga meningkatkan jumlah kolom dalam pembobotan. Sementara itu, penggunaan TF-IDF saja hanya mempertimbangkan frekuensi kata dalam dokumen dan di seluruh kumpulan data, sehingga menghasilkan ruang fitur yang lebih terbatas dan jumlah kolom yang lebih sedikit

4.3. Seleksi Fitur

Pada tahap ini, akan dilakukan proses seleksi fitur dengan menggunakan metode Chi-Square dan Information Gain. Jumlah fitur yang akan digunakan adalah sebanyak K=1000, di mana dari hasil pembobotan kata pada tabel 4.1 dan tabel 4.2, akan diambil 1000 fitur terbaik. Tujuan di balik pelaksanaan proses seleksi fitur ini

adalah untuk mereduksi dimensi data, dengan harapan dapat meningkatkan performa model dengan menghilangkan fitur-fitur yang tidak memberikan informasi yang berharga atau bahkan memperkenalkan noise ke dalam model. Proses seleksi fitur akan mengidentifikasi dan memilih fitur-fitur yang paling informatif dan relevan dari dataset. Dalam kasus ini, kita akan menggunakan dua metode, yaitu Chi-Square dan Information Gain. Metode Chi-Square akan membantu kita mengukur hubungan antara fitur-fitur dan variabel target dalam bentuk tabel kontingensi. Sementara itu, Information Gain akan membantu kita mengukur seberapa besar informasi yang diberikan oleh setiap fitur terhadap kelas target. Setelah kedua metode ini diterapkan, kita akan memilih 1000 fitur dengan skor tertinggi dari hasil pembobotan kata. Hal ini akan membantu mengurangi kompleksitas data dengan mempertahankan hanya fitur-fitur yang paling berpengaruh. Diharapkan bahwa dengan melakukan seleksi fitur ini, kita dapat meningkatkan kualitas model dengan memfokuskan perhatian pada fitur-fitur yang memiliki dampak signifikan pada hasil prediksi. Langkah ini merupakan bagian penting dalam pra-pemrosesan data, karena dapat membantu mewujudkan peningkatan dalam kinerja model dan mengurangi risiko overfitting akibat dimensi data yang tinggi.

4.4. Model SVM

Dalam klasifikasi, model SVM mencoba menemukan hiperplane yang membagi dua kelas data dengan sejauh mungkin dari titik-titik data yang terdekat dari kedua kelas. Tujuannya adalah untuk menciptakan batas keputusan yang optimal sehingga dapat mengklasifikasikan data baru dengan akurasi tinggi. Model

SVM dapat bekerja dengan baik untuk data yang linier maupun non-linier. Jika data tidak linier, SVM menggunakan fungsi kernel untuk mengubah ruang fitur menjadi ruang yang lebih tinggi sehingga hiperplane dapat memisahkan kelas dengan baik. Beberapa parameter yang penting dalam model SVM meliputi:

1. Kernel: Menentukan jenis fungsi kernel yang digunakan untuk memetakan data ke ruang fitur yang lebih tinggi. Beberapa kernel umum adalah linear, polinomial, sigmoid, dan RBF (Radial Basis Function).
2. C: Parameter penalti yang mengontrol tingkat toleransi terhadap kesalahan klasifikasi. Nilai C yang lebih tinggi akan memberikan penalti yang lebih tinggi terhadap kesalahan dan mendorong model untuk mencoba memisahkan data dengan benar.
3. Gamma (hanya untuk kernel non-linear): Parameter kernel yang mengontrol fleksibilitas dan kehalusan hiperplane. Nilai gamma yang lebih tinggi akan memberikan lebih banyak fleksibilitas, yang dapat menyebabkan overfitting.

Setelah dilatih, model SVM dapat digunakan untuk memprediksi kelas dari data baru berdasarkan posisi relatifnya terhadap hiperplane yang telah dibuat selama pelatihan. Model SVM juga dapat menghasilkan skor kepercayaan (confidence scores) untuk setiap prediksi.

Pada penelitian ini, kami akan mencoba semua parameter yang tersedia dan mencari parameter terbaik dengan menggunakan metode GridSearch. Dengan menggunakan GridSearch, kami akan menjelajahi kombinasi berbagai parameter untuk menemukan kombinasi yang menghasilkan performa terbaik pada model SVM yang digunakan.

4.3.1 Skenario Pengujian Model SVM

Pada tahap ini, kami merancang enam skenario pengujian yang berbeda untuk menguji model SVM yang telah kami bangun. Berikut ini adalah rancangan skenario yang akan digunakan untuk memprediksi ulasan palsu

Tabel 4.3 Rancangan Skenario Model SVM

Scenario	SVM	TF-IDF	N-gram	Feature selection	
				CS	IG
1	✓	✓			
2	✓	✓	✓		
3	✓	✓	✓	✓	
4	✓	✓	✓		✓
5	✓	✓		✓	
6	✓	✓			✓

Tabel diatas merupakan representasi dari kombinasi beberapa metode atau teknik yang akan digunakan dalam pengujian kinerja model klasifikasi menggunakan SVM.

1. Pada kolom pertama mununjukkan apakah SVM akan digunakan dalam pengujian performa atau tidak.
2. Kolom kedua menunjukkan apakah penggunaan TF-IDF akan digunakan dalam pengolahan teks atau tidak.
3. Kolom ketiga menunjukkan apakah teknik N-gram akan digunakan dalam pengolahan teks atau tidak
4. Kolom keempat menunjukkan apakah metode seleksi fitur berdasarkan uji statistik Chi-Squared akan digunakan atau tidak.
5. Kolom kelima menunjukkan apakah metode seleksi fitur berdasarkan Information Gain akan digunakan atau tidak.

Setelah pengujian dilakukan dengan kombinasi fitur dan metode yang ditentukan dalam tabel, selanjutnya akan melakukan evaluasi performa model SVM pada setiap skenario. Performa model biasanya diukur menggunakan berbagai metrik seperti akurasi, presisi, recall, F1-score. Berikut ini adalah beberapa tujuan dari matrik tersebut:

1. Accuracy adalah untuk mengukur seberapa banyak prediksi yang benar dari keseluruhan prediksi yang dilakukan oleh model.
2. Precision dilakukan untuk mengukur sejaug mana prediksi positif yang dibuat oleh model adalah benar.
3. Recall dilakukan untuk mengukur sejauh mana model dapat mendeteksu semua instance positif yang sebenarnya.
4. F1-score adalah penggabungan presisi dan recall ke dalam satu matrik Tunggal yang mencerminkan keseimbangan antara keduanya

Tujuan dari evaluasi performa model adalah untuk mengukur seberapa baik model tersebut bekerja dalam melakukan tugas yang telah ditugaskan, serta untuk memahami sejauh mana model tersebut dapat diandalkan dan berguna dalam konteks praktis.

Pada skenario pertama, kami menggunakan algoritma SVM untuk melakukan proses klasifikasi ulasan palsu. Kami menggunakan seluruh fitur yang ada tanpa melakukan proses feature selection. Dengan menggunakan SVM, kami dapat mengklasifikasikan ulasan-ulasan tersebut ke dalam kategori ulasan palsu atau bukan palsu berdasarkan pola-pola yang ada dalam data tersebut. Skenario ini

memberikan gambaran awal tentang performa SVM dalam mengklasifikasikan ulasan palsu sebelum melakukan proses penyempurnaan dengan metode feature selection lainnya.

Tabel 4.4 Hasil Performa Skenario 1 Model SVM

0	0.1	linear	scale	0.8438	0.7798	0.9097	0.8397
1	0.1	rbf	auto	0.8438	0.7798	0.9097	0.8397
2	0.1	poly	scale	0.4750	0.0000	0.0000	0.0000
3	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
4	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
5	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
6	0.1	poly	scale	0.8438	0.7679	0.9214	0.8377
7	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
8	1.0	linear	scale	0.8812	0.8690	0.9012	0.8848
9	1.0	rbf	auto	0.8812	0.8690	0.9012	0.8848
10	**1.0**	**poly**	**scale**	**0.8875**	**0.8810**	**0.9024**	**0.8916**
11	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
12	1.0	linear	scale	0.8531	0.8333	0.8805	0.8563
13	1.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
14	1.0	poly	scale	0.8844	0.8750	0.9018	0.8882
15	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
16	10.0	linear	scale	0.8812	0.8750	0.8963	0.8855
17	10.0	rbf	auto	0.8812	0.8750	0.8963	0.8855
18	10.0	poly	scale	0.8875	0.8810	0.9024	0.8916
19	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
20	10.0	linear	scale	0.8531	0.8333	0.8805	0.8563
21	10.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
22	10.0	poly	scale	0.8531	0.8512	0.8667	0.8589
23	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
0	0.1	linear	scale	0.8438	0.7798	0.9097	0.8397

Tabel tersebut menunjukkan hasil evaluasi performa model SVM pada beberapa kombinasi hyperparameter yang berbeda. Terdapat tiga hyperparameter

yang diatur pada tabel tersebut, yaitu parameter C, jenis kernel (kernel), dan gamma. Dari hasil evaluasi, dapat dilihat bahwa hasil terbaik diperoleh oleh hyperparameter dengan kombinasi C 1.0 dan 10.0, serta jenis kernel poly. Model SVM dengan kombinasi tersebut mencapai akurasi sebesar 0.8875, menunjukkan kinerja yang sangat baik dalam mengklasifikasikan data dan memprediksi dengan akurasi yang tinggi. Hasil ini membuktikan bahwa hyperparameter yang tepat sangat penting dalam perancangan model SVM untuk mendapatkan performa yang optimal dan dapat membantu dalam mendeteksi ulasan palsu.

Pada skenario kedua, kami memperluas fitur-fitur yang digunakan dalam proses klasifikasi ulasan palsu dengan menggunakan pendekatan n-gram. Kami menggabungkan metode TF-IDF dengan ekstraksi n-gram, yaitu mengambil sekumpulan kata berurutan sebanyak n dari teks ulasan. Hal ini membantu kami dalam menangkap konteks dan hubungan antar kata yang lebih kompleks dalam ulasan tersebut. Setelah itu, kami menerapkan algoritma SVM untuk melakukan klasifikasi ulasan palsu berdasarkan fitur-fitur n-gram yang telah diekstraksi. Dengan skenario ini, kami berharap dapat meningkatkan kemampuan SVM dalam mengenali pola ulasan palsu dengan mempertimbangkan konteks yang lebih lengkap melalui pendekatan n-gram

Tabel 4.5 Hasil Performa Skenario 2 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
0	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
1	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
2	0.1	poly	scale	0.4750	0.0000	0.0000	0.0000
3	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000

Tabel 4.5 Lanjutan Hasil Performa Skenario 2 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
4	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
5	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
6	0.1	poly	scale	0.4750	0.0000	0.0000	0.0000
7	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
8	1.0	linear	scale	0.8906	0.8869	0.9030	0.8949
9	1.0	rbf	auto	0.8906	0.8869	0.9030	0.8949
10	1.0	poly	scale	0.8812	0.8571	0.9114	0.8834
11	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
12	1.0	linear	scale	0.4812	0.0119	1.0000	0.0235
13	1.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
14	1.0	poly	scale	0.8938	0.8929	0.9036	0.8982
15	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
16	10.0	linear	scale	0.8906	0.8869	0.9030	0.8949
17	10.0	rbf	auto	0.8906	0.8869	0.9030	0.8949
18	10.0	poly	scale	0.8844	0.8750	0.9018	0.8882
19	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
20	10.0	linear	scale	0.4812	0.0119	1.0000	0.0235
21	10.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
22	**10.0**	**poly**	**scale**	**0.8938**	**0.8810**	**0.9136**	**0.8970**
23	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000

Pada skenario kedua, hasil terbaik diperoleh dengan menggunakan kombinasi hyperparameter C 10.0, Kernel Poly, dan Gamma Scale, yang menghasilkan akurasi sebesar 0.8938. Perlu diperhatikan bahwa akurasi awal sebesar 0.8875 diperoleh tanpa menggunakan n-gram model. Oleh karena itu, penambahan n-gram model berhasil meningkatkan akurasi model SVM. Penemuan ini juga sejalan dengan penelitian sebelumnya yang dilakukan oleh (Pacol & Palaoag, 2021). Dalam penelitian mereka, mereka juga menggunakan kombinasi n-gram model dan SVM

untuk meningkatkan akurasi. Hasil penelitian mereka menunjukkan bahwa penggunaan n-gram model secara signifikan meningkatkan akurasi SVM.

Dalam skenario ketiga, kami mengintegrasikan pendekatan n-gram dengan seleksi fitur menggunakan metode chi-square untuk meningkatkan kinerja klasifikasi ulasan palsu. Setelah melakukan ekstraksi fitur n-gram seperti pada skenario sebelumnya, kami menerapkan metode chi-square untuk mengevaluasi kepentingan setiap fitur n-gram dalam membedakan ulasan palsu dan bukan palsu. Fitur-fitur dengan skor chi-square yang tinggi dipilih sebagai fitur penting untuk digunakan dalam model klasifikasi. Kemudian, kami menggunakan algoritma SVM untuk melakukan klasifikasi ulasan palsu berdasarkan fitur-fitur n-gram yang telah diseleksi menggunakan chi-square. Dengan skenario ini, kami berharap dapat meningkatkan kemampuan SVM dalam mengenali pola ulasan palsu dengan memilih fitur-fitur yang paling informatif melalui pendekatan n-gram dan seleksi fitur chi-square.

Tabel 4.6 Hasil Performa Skenario 3 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
0	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
1	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
2	0.1	poly	scale	0.7844	0.7202	0.8462	0.7781
3	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
4	0.1	linear	scale	0.5969	1.0000	0.5657	0.7226
5	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
6	0.1	poly	scale	0.9000	0.9345	0.8820	0.9075
7	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
8	1.0	linear	scale	0.8906	0.9345	0.8674	0.8997
9	1.0	rbf	auto	0.8906	0.9345	0.8674	0.8997

Tabel 4.6 Lanjutan Hasil Performa Skenario 3 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
10	1.0	poly	scale	0.9094	0.9048	0.9212	0.9129
11	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
12	1.0	linear	scale	0.8094	0.9940	0.7357	0.8456
13	1.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
14	1.0	poly	scale	0.9188	0.9226	0.9226	0.9226
15	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
16	**10.0**	**linear**	**scale**	**0.9281**	**0.9405**	**0.9240**	**0.9322**
17	**10.0**	**rbf**	**auto**	**0.9281**	**0.9405**	**0.9240**	**0.9322**
18	10.0	poly	scale	0.9094	0.9048	0.9212	0.9129
19	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
20	10.0	linear	scale	0.8688	0.9464	0.8281	0.8833
21	10.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
22	10.0	poly	scale	0.8531	0.8631	0.8580	0.8605
23	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000

Pada skenario ketiga, terdapat dua kombinasi hyperparameter yang menghasilkan hasil terbaik dengan akurasi yang sama, yaitu C=10.0, Kernel=Linear dengan Gamma=scale, dan C=10.0, Kernel=rbf dengan Gamma=auto. Kedua kombinasi ini memiliki akurasi sebesar 0.9281, yang merupakan peningkatan yang signifikan dibandingkan dengan akurasi awal sebesar 0.8875. Hal ini menunjukkan bahwa penggunaan n-gram dan chi-square pada model SVM dengan kombinasi hyperparameter C=10.0, Kernel=Linear atau rbf, dan Gamma=scale atau auto dapat menghasilkan performa yang lebih baik dalam memprediksi kelas target. Penelitian yang dilakukan oleh (S. Fachrurrozi, 2021). Dalam penelitian tersebut, dilakukan beberapa eksperimen dengan variasi pemilihan fitur dan reduksi fitur menggunakan teknik bigram dan unigram. Hasilnya menunjukkan bahwa penerapan unigram dan

Chi-square dengan reduksi fitur sebesar 90% memberikan akurasi tertinggi sebesar 97,98%. Ini merupakan peningkatan yang signifikan dibandingkan dengan SVM tanpa penerapan N-gram dan tanpa seleksi fitur yang hanya memiliki akurasi sebesar 76,80%. Penelitian ini juga menekankan bahwa kinerja SVM dipengaruhi oleh penerapan N-gram dan Chi-square, yang memiliki dampak pada jumlah fitur yang digunakan. Oleh karena itu, penting untuk menentukan nilai N-gram dan jumlah fitur secara tepat agar dapat mencapai performansi klasifikasi teks yang optimal.

Dalam skenario keempat, kami menggabungkan pendekatan n-gram dengan metode information gain untuk meningkatkan klasifikasi ulasan palsu menggunakan SVM. Setelah melakukan ekstraksi fitur n-gram seperti pada skenario sebelumnya, kami menerapkan metode information gain untuk mengevaluasi tingkat informatifitas setiap fitur n-gram dalam membedakan ulasan palsu dan bukan palsu. Fitur-fitur dengan informasi gain yang tinggi dipilih sebagai fitur penting untuk digunakan dalam model klasifikasi. Selanjutnya, kami menggunakan algoritma SVM untuk melakukan klasifikasi ulasan palsu berdasarkan fitur-fitur n-gram yang telah diseleksi menggunakan information gain. Dengan skenario ini, kami berharap dapat meningkatkan kinerja SVM dalam mengenali ulasan palsu dengan memilih fitur-fitur yang memiliki tingkat informatifitas yang tinggi melalui pendekatan n-gram dan metode information gain.

Tabel 4.7 Hasil Performa Skenario 4 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
0	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
1	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
2	0.1	poly	scale	0.5781	0.2202	0.9024	0.3541
3	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
4	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
5	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
6	0.1	poly	scale	0.8781	0.8929	0.8772	0.8850
7	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
8	1.0	linear	scale	0.8781	0.8988	0.8728	0.8856
9	1.0	rbf	auto	0.8781	0.8988	0.8728	0.8856
10	1.0	poly	scale	0.8812	0.8869	0.8869	0.8869
11	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
12	1.0	linear	scale	0.7812	0.9821	0.7112	0.8250
13	1.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
14	**1.0**	**poly**	**scale**	**0.8938**	**0.8810**	**0.9136**	**0.8970**
15	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
16	**10.0**	**linear**	**scale**	**0.8938**	**0.8988**	**0.8988**	**0.8988**
17	**10.0**	**rbf**	**auto**	**0.8938**	**0.8988**	**0.8988**	**0.8988**
18	10.0	poly	scale	0.8844	0.8750	0.9018	0.8882
19	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
20	10.0	linear	scale	0.8094	0.9643	0.7465	0.8416
21	10.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
22	10.0	poly	scale	0.8406	0.8333	0.8589	0.8459
23	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000

Pada skenario keempat, terdapat tiga kombinasi hyperparameter yang memberikan hasil terbaik dengan akurasi yang sama. Kombinasi pertama adalah C=10.0 dengan kernel Linear dan Gamma=scale, kombinasi kedua adalah C=1.0 dengan kernel poly dengan Gamma=scale, dan terakhir kombinasi dengan C=10.0

dengan kernel=rbf dengan gamma auto. ketiga kombinasi ini memiliki akurasi yang mencapai 0.8938, menunjukkan peningkatan yang signifikan dibandingkan dengan akurasi awal sebesar 0.8875. Pada skenario ini, meskipun peneliti belum menemukan jurnal yang secara khusus menggunakan kombinasi n-gram, information gain, dan SVM untuk klasifikasi, namun peneliti memilih skenario ini berdasarkan penelitian sebelumnya. Penelitian yang dilakukan oleh (S. Fachrurrozi, 2021) menggunakan n-gram dan chi-square telah berhasil meningkatkan akurasi SVM. Oleh karena itu, peneliti memutuskan untuk menggabungkan kedua teknik ini dalam penelitiannya untuk mencapai peningkatan akurasi SVM yang lebih baik.

Dalam skenario kelima, kami menggunakan pendekatan chi-square untuk melakukan seleksi fitur sebelum melakukan klasifikasi ulasan palsu dengan menggunakan SVM. Kami menghitung skor chi-square untuk setiap fitur yang ada, yang mencerminkan tingkat hubungan antara keberadaan fitur tersebut dengan label ulasan palsu atau bukan palsu. Fitur-fitur dengan skor chi-square yang tinggi dianggap memiliki hubungan yang signifikan dengan ulasan palsu dan digunakan sebagai fitur penting dalam model klasifikasi. Selanjutnya, kami menerapkan algoritma SVM untuk melakukan klasifikasi berdasarkan fitur-fitur yang dipilih melalui seleksi chi-square. Dengan skenario ini, kami berharap dapat meningkatkan kemampuan SVM dalam mengklasifikasikan ulasan palsu dengan memilih fitur-fitur yang memiliki hubungan yang signifikan dengan ulasan palsu melalui pendekatan chi-square.

Tabel 4.8 Hasil Performa Skenario 5 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
0	0.1	linear	scale	0.8406	0.7560	0.9270	0.8328
1	0.1	rbf	auto	0.8406	0.7560	0.9270	0.8328
2	0.1	poly	scale	0.7188	0.4643	1.0000	0.6341
3	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
4	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
5	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
6	0.1	poly	scale	0.8938	0.8988	0.8988	0.8988
7	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
8	1.0	linear	scale	0.9094	0.9048	0.9212	0.9129
9	1.0	rbf	auto	0.9094	0.9048	0.9212	0.9129
10	1.0	poly	scale	0.9031	0.9048	0.9102	0.9075
11	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
12	1.0	linear	scale	0.8719	0.9226	0.8470	0.8832
13	1.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
14	1.0	poly	scale	0.9156	0.9167	0.9222	0.9194
15	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
16	10.0	linear	scale	0.9156	0.9226	0.9172	0.9199
17	10.0	rbf	auto	0.9156	0.9226	0.9172	0.9199
18	**10.0**	**poly**	**scale**	**0.9188**	**0.9226**	**0.9226**	**0.9226**
19	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
20	10.0	linear	scale	0.8938	0.8988	0.8988	0.8988
21	10.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
22	10.0	poly	scale	0.9000	0.9107	0.9000	0.9053
23	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000

Pada skenario kelima, terlihat bahwa terdapat kombinasi hyperparameter yang memberikan hasil terbaik dalam klasifikasi. Kombinasi tersebut adalah dengan menggunakan C=10.0, kernel poly, dan gamma=scale. Hasil yang dihasilkan adalah akurasi sebesar 0.9188, yang menunjukkan peningkatan yang signifikan

dibandingkan dengan akurasi awal sebesar 0.8875. Dalam konteks ini, kombinasi hyperparameter tersebut berhasil memberikan akurasi yang lebih baik dalam klasifikasi data. Terkait dengan hal ini, terdapat penelitian sebelumnya yang dilakukan oleh (Somantri & Apriliani, 2018), yang juga mengadopsi skenario yang sama dan berhasil meningkatkan akurasi SVM dan Chi-square. Temuan ini mendukung hasil penelitian sebelumnya dan menunjukkan potensi penggunaan kombinasi hyperparameter yang optimal dalam meningkatkan akurasi klasifikasi data. Dalam konteks ini, penggabungan SVM dan Chi-square dapat menjadi pendekatan yang efektif untuk meningkatkan performa model klasifikasi.

Dalam skenario enam, kami menggunakan pendekatan information gain untuk melakukan seleksi fitur sebelum melakukan klasifikasi ulasan palsu dengan menggunakan SVM. Kami menghitung informasi gain dari setiap fitur dalam dataset, yang menggambarkan seberapa informatifnya fitur tersebut dalam membedakan antara ulasan palsu dan bukan palsu. Fitur-fitur dengan nilai informasi gain yang tinggi dianggap memiliki kontribusi yang signifikan dalam klasifikasi ulasan palsu. Kami kemudian menggunakan fitur-fitur yang dipilih melalui seleksi information gain sebagai input untuk algoritma SVM, yang digunakan untuk melakukan klasifikasi ulasan menjadi dua kategori: palsu atau bukan palsu. Dengan menerapkan skenario ini, kami berharap dapat meningkatkan kemampuan SVM dalam mengklasifikasikan ulasan palsu dengan memanfaatkan fitur-fitur yang memiliki nilai informasi yang tinggi dalam membedakan antara ulasan palsu dan bukan palsu melalui pendekatan information gain.

Tabel 4.9 Hasil Performa Skenario 6 Model SVM

	C	Kernel	Gamma	Accuracy	Recall	Precision	F1-score
0	0.1	linear	scale	0.8344	0.7560	0.9137	0.8274
1	0.1	rbf	auto	0.8344	0.7560	0.9137	0.8274
2	0.1	poly	scale	0.4906	0.0357	0.8571	0.0686
3	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
4	0.1	linear	scale	0.4750	0.0000	0.0000	0.0000
5	0.1	rbf	auto	0.4750	0.0000	0.0000	0.0000
6	0.1	poly	scale	0.8594	0.8333	0.8917	0.8615
7	0.1	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
8	1.0	linear	scale	0.8781	0.8750	0.8909	0.8829
9	1.0	rbf	auto	0.8781	0.8750	0.8909	0.8829
10	**1.0**	**poly**	**scale**	**0.8906**	**0.8810**	**0.9080**	**0.8943**
11	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
12	1.0	linear	scale	0.7969	0.9762	0.7289	0.8346
13	1.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
14	1.0	poly	scale	0.8781	0.8690	0.8957	0.8822
15	1.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
16	10.0	linear	scale	0.8531	0.8631	0.8580	0.8605
17	10.0	rbf	auto	0.8531	0.8631	0.8580	0.8605
18	10.0	poly	scale	0.8812	0.8750	0.8963	0.8855
19	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000
20	10.0	linear	scale	0.8281	0.9702	0.7653	0.8556
21	10.0	rbf	auto	0.4750	0.0000	0.0000	0.0000
22	10.0	poly	scale	0.8594	0.8810	0.8555	0.8680
23	10.0	sigmoid	auto	0.4750	0.0000	0.0000	0.0000

Dalam skenario keenam, kami melakukan eksplorasi kombinasi hyperparameter untuk mencari hasil terbaik dalam proses klasifikasi. Setelah melakukan berbagai percobaan, kami menemukan kombinasi hyperparameter yang

optimal, yaitu C=1.0, kernel poly, dan gamma=scale. Dengan konfigurasi tersebut, kami berhasil mencapai akurasi sebesar 0.8906, yang menunjukkan peningkatan yang signifikan dibandingkan dengan akurasi awal sebesar 0.8875. Meskipun peningkatannya relatif kecil, yaitu sebesar 0.0031, temuan ini tetap sejalan dengan penelitian sebelumnya yang dilakukan oleh (Somantri & Apriliani, 2018), yang menggunakan skenario SVM dan IG (Information Gain). Hasil kami mengindikasikan bahwa penggunaan SVM dan IG dalam kombinasi hyperparameter yang tepat dapat memberikan peningkatan yang konsisten dalam akurasi klasifikasi data. Dengan demikian, hasil penelitian ini memberikan kontribusi yang berharga dalam memahami pentingnya pemilihan hyperparameter yang optimal dalam meningkatkan performa SVM dan penggunaan IG dalam analisis klasifikasi data.

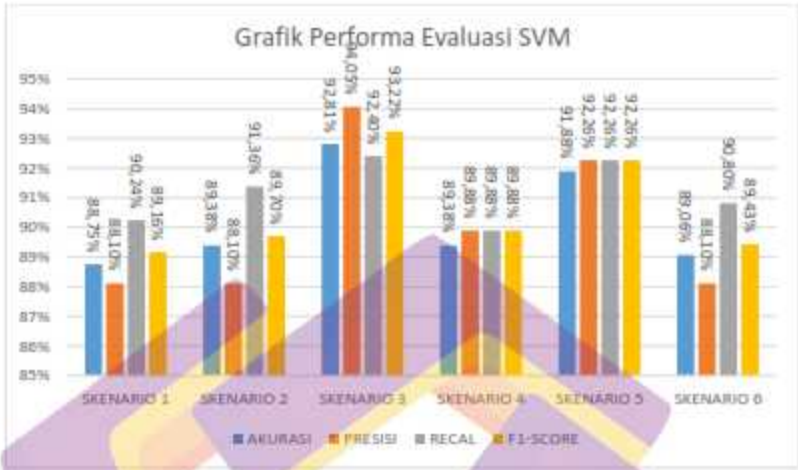
Setelah mencari hasil terbaik dari masing-masing skenario pada model SVM, langkah selanjutnya adalah membandingkan hasil tersebut antara skenario-skenario untuk menentukan skenario mana yang memberikan performa terbaik pada model SVM. Berikut adalah hasil klasifikasi model svm:

Tabel 4.10 Hasil klasifikasi algoritma SVM

Skenario	SVM	TF-IDF	N-gram	Feature selection		Accuracy	Precision	Recall	F1-Score
				CS	IG				
1	✓	✓				0.8875	0.8810	0.9024	0.8916
2	✓	✓	✓			0.8938	0.8810	0.9136	0.8970
3	✓	✓	✓	✓		**0.9281**	**0.9405**	**0.9240**	**0.9322**
4	✓	✓	✓		✓	0.8938	0.8988	0.8988	0.8988
5	✓	✓		✓		0.9188	0.9226	0.9226	0.9226
6	✓	✓			✓	0.8906	0.8810	0.9080	0.8943

Dalam tabel ini, terdapat enam skenario yang telah dievaluasi menggunakan model SVM dengan kombinasi berbeda dari fitur dan metode seleksi fitur. Skenario satu menggunakan SVM dengan fitur TF-IDF dan tanpa penggunaan N-gram dan seleksi fitur. Skenario dua menggunakan SVM dengan fitur TF-IDF, N-gram, dan tanpa seleksi fitur. Skenario tiga menggunakan SVM dengan fitur TF-IDF, N-gram, dan seleksi fitur menggunakan CS (chi-square). Skenario empat menggunakan SVM dengan fitur TF-IDF, N-gram, dan seleksi fitur menggunakan IG (information gain). Skenario lima menggunakan SVM dengan fitur N-gram dan seleksi fitur menggunakan IG. Skenario enam menggunakan SVM dengan seleksi fitur menggunakan IG tanpa menggunakan fitur TF-IDF atau N-gram. Berdasarkan hasil evaluasi, skenario tiga mencapai akurasi tertinggi sebesar 0.9281 dengan precision 0.9405, recall 0.9240, dan F1-score 0.9322. Ini menunjukkan bahwa penggunaan fitur TF-IDF, N-gram, dan seleksi fitur menggunakan chi-square dalam skenario tiga memberikan performa yang lebih baik dalam klasifikasi menggunakan model SVM.

Untuk mempermudah visualisasi informasi, data tersebut akan disajikan dalam bentuk grafik. Grafik tersebut akan memberikan representasi yang lebih jelas dan mudah dipahami tentang performa evaluali yang telah dilakukan. Berikut ini adalah visualisasi performa evaluasi SVM dalam bentuk grafik



Gambar 4.4 Grafik Performa Evaluasi SVM

4.5. Model Random Forest

Dalam klasifikasi, model Random Forest menggunakan kumpulan pohon keputusan yang bekerja secara ensemble untuk mengklasifikasikan data. Model ini bertujuan untuk menciptakan batas keputusan yang optimal dengan membangun banyak pohon keputusan yang berbeda dan menggabungkan prediksi mereka. Model Random Forest dapat bekerja dengan baik untuk data linier maupun non-linier, serta mampu mengatasi masalah overfitting. Beberapa parameter penting dalam model Random Forest antara lain:

1. Jumlah Pohon (`n_estimators`): Menentukan jumlah pohon keputusan yang akan dibangun dalam ensemble. Jumlah yang lebih besar dapat meningkatkan akurasi model, tetapi juga memperbesar waktu pelatihan.
2. Kedalaman Maksimum (`max_depth`): Parameter ini mengontrol kedalaman maksimum setiap pohon dalam ensemble. Mengatur nilai `max_depth` dapat membantu mencegah overfitting atau underfitting.

Pohon yang terlalu dalam dapat mempelajari detail yang berlebihan dari data pelatihan, sementara pohon yang terlalu dangkal mungkin tidak dapat menangkap pola yang kompleks.

3. Jumlah Fitur Maksimum (`max_features`): Parameter ini menentukan jumlah fitur yang akan dipertimbangkan saat mencari pemisah terbaik dalam setiap pemisahan simpul dalam pohon. Memilih nilai yang lebih kecil dapat meningkatkan keragaman dan kekuatan prediktif setiap pohon, sementara nilai yang lebih besar mungkin menghasilkan korelasi yang lebih tinggi antar-pohon.
4. Jumlah Sampel Minimum untuk Membagi (`min_samples_split`): Parameter ini menentukan jumlah sampel minimum yang diperlukan untuk membagi simpul dalam pohon keputusan. Menentukan nilai yang lebih tinggi dapat membantu mencegah pembentukan simpul yang terlalu spesifik yang hanya mempelajari kejadian langka atau noise dalam data.

Setelah dilatih, model Random Forest dapat digunakan untuk memprediksi kelas dari data baru dengan menggabungkan prediksi dari setiap pohon dalam ensemble. Selain itu, model Random Forest juga dapat memberikan estimasi probabilitas atau skor kepercayaan untuk setiap kelas prediksi.

Dalam penelitian ini, peneliti mencoba semua parameter model Random Forest untuk mencari kombinasi yang menghasilkan performa terbaik untuk klasifikasi ulasan palsu.

4.4.1 Skenario Pengujian Model Random Forest

Pada tahap ini, kami merancang enam skenario pengujian yang berbeda untuk menguji model Random Forest yang telah kami bangun. Berikut ini adalah rancangan skenario model Random Forest yang akan digunakan.

Tabel 4.11 Rancangan Skenario Model Random Forest

Scenario	RF	TF-IDF	N-gram	Feature selection	
				CS	IG
1	✓	✓			
2	✓	✓	✓		
3	✓	✓	✓	✓	
4	✓	✓	✓		✓
5	✓	✓		✓	
6	✓	✓			✓

Tabel diatas merupakan representasi dari kombinasi beberapa metode atau teknik yang akan digunakan dalam pengujian kinerja model klasifikasi menggunakan Random Forest.

1. Pada kolom pertama mununjukkan apakah Random Forest akan digunakan dalam pengujian performa atau tidak.
2. Kolom kedua menunjukkan apakah penggunaan TF-IDF akan digunakan dalam pengolahan teks atau tidak.
3. Kolom ketiga menunjukkan apakah teknik N-gram akan digunakan dalam pengolahan teks atau tidak
4. Kolom keempat menunjukkan apakah metode seleksi fitur berdasarkan uji statistik Chi-Squared akan digunakan atau tidak.
5. Kolom kelima menunjukkan apakah metode seleksi fitur berdasarkan Information Gain akan digunakan atau tidak.

Setelah pengujian dilakukan dengan kombinasi fitur dan metode yang ditentukan dalam tabel, selanjutnya akan melakukan evaluasi performa model Random Forest pada setiap skenario. Performa model biasanya diukur menggunakan berbagai metrik seperti akurasi, presisi, recall, F1-score. Berikut ini adalah beberapa tujuan dari matrik tersebut:

1. Accuracy adalah untuk mengukur seberapa banyak prediksi yang benar dari keseluruhan prediksi yang dilakukan oleh model.
2. Precision dilakukan untuk mengukur sejaug mana prediksi positif yang dibuat oleh model adalah benar.
3. Recall dilakukan untuk mengukur sejauh mana model dapat mendeteksu semua instance positif yang sebenarnya.
4. F1-score adalah penggabungan presisi dan recall ke dalam satu matrik Tunggal yang mencerminkan keseimbangan antara keduanya

Tujuan dari evaluasi performa model adalah untuk mengukur seberapa baik model tersebut bekerja dalam melakukan tugas yang telah ditugaskan, serta untuk memahami sejauh mana model tersebut dapat diandalkan dan berguna dalam konteks praktis.

Pada skenario pertama, kami menggunakan algoritma Random Forest untuk melakukan proses klasifikasi ulasan palsu. Kami menggunakan seluruh fitur yang ada tanpa melakukan proses feature selection. Dengan menggunakan Random Forest, kami dapat mengklasifikasikan ulasan-ulasan tersebut ke dalam kategori ulasan palsu atau bukan palsu berdasarkan pola-pola yang ada dalam data tersebut.

Skenario ini memberikan gambaran awal tentang performa Random Forest dalam mengklasifikasikan ulasan palsu sebelum melakukan proses penyempurnaan dengan metode feature selection lainnya.

Tabel 4.12 Hasil Performa Skenario 1 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
0	10	None	auto	2	0.7313	0.5952	0.8475	0.6993
1	10	5	auto	2	0.7563	0.7321	0.7885	0.7593
2	10	10	sqrt	5	0.7406	0.6429	0.8244	0.7224
3	10	None	sqrt	5	0.7656	0.7262	0.8079	0.7649
4	10	5	auto	2	0.7156	0.6964	0.7452	0.7200
5	10	10	auto	2	0.6906	0.5119	0.8350	0.6347
6	10	None	sqrt	5	0.7063	0.6429	0.7606	0.6968
7	10	5	sqrt	5	0.6750	0.5952	0.7353	0.6579
8	10	10	auto	2	0.7656	0.7083	0.8207	0.7604
9	10	None	auto	2	0.7719	0.7321	0.8146	0.7712
10	10	5	sqrt	5	0.7500	0.6845	0.8099	0.7419
11	10	10	sqrt	5	0.7438	0.6964	0.7905	0.7405
12	50	None	auto	2	0.8281	0.7798	0.8792	0.8265
13	50	5	auto	2	0.8344	0.7976	0.8758	0.8349
14	50	10	sqrt	5	0.8344	0.7738	0.8966	0.8307
15	50	None	sqrt	5	0.8344	0.8036	0.8710	0.8359
16	50	5	auto	2	0.8156	0.7679	0.8658	0.8139
17	50	10	auto	2	0.7625	0.6726	0.8433	0.7483
18	50	None	sqrt	5	0.7719	0.7202	0.8231	0.7683
19	50	5	sqrt	5	0.7875	0.7262	0.8472	0.7821
20	50	10	auto	2	0.8219	0.7917	0.8581	0.8235
21	50	None	auto	2	0.8281	0.8095	0.8553	0.8318
22	50	5	sqrt	5	0.8063	0.7619	0.8533	0.8050
23	50	10	sqrt	5	0.8469	0.8452	0.8606	0.8529
24	100	None	auto	2	0.8375	0.8155	0.8671	0.8405
25	100	5	auto	2	0.8469	0.8274	0.8742	0.8502
26	**100**	**10**	**sqrt**	**5**	**0.8594**	**0.8333**	**0.8917**	**0.8615**
27	100	None	sqrt	5	0.8469	0.8155	0.8839	0.8483

Tabel 4.12 Lanjutan Hasil Performa Skenario 1 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
28	100	5	auto	2	0.7844	0.7024	0.8613	0.7738
29	100	10	auto	2	0.8219	0.7619	0.8828	0.8179
30	100	None	sqrt	5	0.8438	0.8214	0.8734	0.8466
31	100	5	sqrt	5	0.8281	0.7798	0.8792	0.8265
32	100	10	auto	2	0.8406	0.8333	0.8589	0.8459
33	100	None	auto	2	0.8281	0.7917	0.8693	0.8287
34	100	5	sqrt	5	0.8469	0.8274	0.8742	0.8502
35	100	10	sqrt	5	0.8531	0.8095	0.9007	0.8527

Pada skenario pertama, kami melakukan evaluasi terhadap 36 kombinasi skenario yang menggunakan model Random Forest dengan variasi parameter yang signifikan. Dalam evaluasi ini, kami mempertimbangkan parameter-parameter penting seperti jumlah estimator (n_estimators), kedalaman maksimum (max_depth), jumlah fitur maksimum (max_features), dan jumlah sampel minimum untuk membagi node internal (min_samples_split). Dari 36 kombinasi skenario yang kami uji, kami berhasil mencapai akurasi tertinggi sebesar 0.8594. Hasil ini memberikan gambaran awal tentang kemampuan model Random Forest dalam mengklasifikasikan ulasan palsu sebelum melibatkan langkah-langkah penyempurnaan lebih lanjut. Evaluasi ini memberikan pandangan awal mengenai performa model Random Forest dalam konteks pengklasifikasian ulasan palsu.

Pada skenario kedua, peneliti menggunakan pendekatan yang lebih canggih dalam melakukan klasifikasi ulasan palsu dengan menggabungkan metode TF-IDF, n-gram model, dan algoritma Random Forest. Dengan menggunakan n-gram model, peneliti dapat memperhatikan konteks dan urutan kata dalam ulasan,

sehingga dapat mengidentifikasi pola-pola yang penting dalam mengklasifikasikan ulasan palsu. Pendekatan ini memungkinkan peneliti untuk menangkap penggunaan kata yang khas dan hubungan antara kata-kata dalam ulasan. Terakhir, dengan menerapkan algoritma Random Forest, peneliti memanfaatkan kekuatan ensemble dari sekumpulan pohon keputusan. Setiap pohon dalam Random Forest memberikan kontribusi dalam mengklasifikasikan ulasan berdasarkan fitur-fitur yang mereka tangkap dari dataset. Melalui proses voting mayoritas, hasil akhir klasifikasi diperoleh.

Tabel 4.13 Hasil Performa Skenario 2 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
0	10	None	auto	2	0.6938	0.5357	0.8182	0.6475
1	10	5	auto	2	0.7563	0.7262	0.7922	0.7578
2	10	10	sqrt	5	0.7563	0.6786	0.8261	0.7451
3	10	None	sqrt	5	0.7313	0.6964	0.7697	0.7313
4	10	5	auto	2	0.6500	0.4524	0.7917	0.5758
5	10	10	auto	2	0.7031	0.6310	0.7626	0.6906
6	10	None	sqrt	5	0.6375	0.4286	0.7826	0.5538
7	10	5	sqrt	5	0.6594	0.6786	0.6746	0.6766
8	10	10	auto	2	0.6906	0.5774	0.7760	0.6621
9	10	None	auto	2	0.7438	0.7083	0.7829	0.7438
10	10	5	sqrt	5	0.6469	0.6131	0.6821	0.6458
11	10	10	sqrt	5	0.7063	0.6607	0.7500	0.7025
12	50	None	auto	2	0.8188	0.8214	0.8313	0.8263
13	50	5	auto	2	0.8438	0.8214	0.8734	0.8466
14	50	10	sqrt	5	0.8188	0.7679	0.8716	0.8165
15	**50**	**None**	**sqrt**	**5**	**0.8719**	**0.8690**	**0.8848**	**0.8769**
16	50	5	auto	2	0.7688	0.6964	0.8357	0.7597
17	50	10	auto	2	0.7063	0.5655	0.8190	0.6690
18	50	None	sqrt	5	0.7406	0.6548	0.8148	0.7261
19	50	5	sqrt	5	0.7906	0.6726	0.9040	0.7713

Tabel 4.13 Lanjutan Hasil Performa Skenario 2 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
20	50	10	auto	2	0.8094	0.7679	0.8543	0.8088
21	50	None	auto	2	0.8094	0.7381	0.8794	0.8026
22	50	5	sqrt	5	0.7969	0.7679	0.8323	0.7988
23	50	10	sqrt	5	0.8000	0.7560	0.8467	0.7987
24	100	None	auto	2	0.8500	0.8333	0.8750	0.8537
25	100	5	auto	2	0.8438	0.8214	0.8734	0.8466
26	100	10	sqrt	5	0.8781	0.8452	0.9161	0.8793
27	100	None	sqrt	5	0.8781	0.8750	0.8909	0.8829
28	100	5	auto	2	0.7906	0.6607	0.9174	0.7682
29	100	10	auto	2	0.8250	0.7679	0.8836	0.8217
30	100	None	sqrt	5	0.7563	0.5833	0.9245	0.7153
31	100	5	sqrt	5	0.8031	0.7024	0.9008	0.7893
32	100	10	auto	2	0.8188	0.7679	0.8716	0.8165
33	100	None	auto	2	0.8125	0.7560	0.8699	0.8089
34	100	5	sqrt	5	0.8469	0.8095	0.8889	0.8474
35	100	10	sqrt	5	0.8531	0.8036	0.9060	0.8517

Pada skenario kedua, peneliti kami mengadopsi pendekatan yang menggabungkan metode n-gram dan algoritma Random Forest untuk melakukan klasifikasi ulasan palsu. Pendekatan ini merupakan adaptasi dari penelitian sebelumnya yang dilakukan oleh (Pacol & Palaoag, 2021), Dalam penelitian mereka, mereka juga berhasil meningkatkan akurasi model Random Forest, meskipun hanya sebesar 0.01. Dalam skenario yang kami lakukan, kami menjalankan serangkaian percobaan dengan menggunakan kombinasi metode n-gram dan algoritma Random Forest. Hasil percobaan tersebut menunjukkan peningkatan signifikan dalam akurasi model Random Forest. Awalnya, model Random Forest memiliki akurasi sebesar 0.8594. Namun, setelah menerapkan

metode n-gram dan algoritma Random Forest, akurasi meningkat menjadi 0.8719. Temuan ini konsisten dengan hasil penelitian sebelumnya dan menunjukkan bahwa penggunaan metode n-gram dalam kombinasi dengan algoritma Random Forest dapat memberikan peningkatan yang signifikan dalam akurasi model. Meskipun peningkatan akurasi hanya sebesar 0.01, hal ini tetap menjadi kontribusi penting dalam meningkatkan kemampuan model dalam mengklasifikasikan ulasan palsu. Dengan demikian, penggabungan metode n-gram dan algoritma Random Forest telah terbukti efektif dalam meningkatkan akurasi model klasifikasi ulasan palsu.

Pada skenario ketiga, peneliti mengadopsi pendekatan yang lebih kompleks dan akurat dalam melakukan klasifikasi ulasan palsu. Mereka menggunakan kombinasi metode TF-IDF, n-gram model, chi-square, dan algoritma Random Forest. Dalam tahap awal, peneliti menerapkan metode TF-IDF untuk mengekstraksi fitur dari dataset teks. Dengan menggunakan TF-IDF, mereka dapat mengukur seberapa penting suatu kata dalam ulasan berdasarkan frekuensi kemunculan kata tersebut di seluruh dataset. Hal ini membantu dalam mengidentifikasi kata-kata kunci yang dapat membedakan ulasan palsu dan bukan palsu. Selanjutnya, peneliti menggunakan n-gram model untuk mempertimbangkan konteks dan urutan kata dalam ulasan. Dengan menggabungkan beberapa kata menjadi satu kesatuan, n-gram model dapat menangkap pola-pola yang lebih kompleks dalam teks, yang membantu dalam mengklasifikasikan ulasan dengan lebih baik. Selanjutnya, peneliti menerapkan metode chi-square untuk seleksi fitur. Chi-square digunakan untuk mengukur hubungan antara setiap fitur (kata) dengan label klasifikasi (ulasan palsu atau bukan palsu). Dengan menggunakan metode ini,

peneliti dapat mengidentifikasi fitur-fitur yang memiliki hubungan yang signifikan dengan label klasifikasi dan mengabaikan fitur-fitur yang kurang relevan. Terakhir, peneliti menggunakan algoritma Random Forest. Random Forest adalah metode ensemble yang terdiri dari beberapa pohon keputusan. Setiap pohon dalam Random Forest memberikan kontribusi dalam mengklasifikasikan ulasan berdasarkan fitur-fitur yang mereka tangkap dari dataset. Melalui proses voting mayoritas, hasil akhir klasifikasi diperoleh.

Tabel 4.14 Hasil Performa Skenario 3 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
0	10	None	auto	2	0.8125	0.7798	0.8506	0.8137
1	10	5	auto	2	0.8188	0.8155	0.8354	0.8253
2	10	10	sqrt	5	0.8250	0.8214	0.8415	0.8313
3	10	None	sqrt	5	0.8031	0.8036	0.8182	0.8108
4	10	5	auto	2	0.7844	0.7798	0.8037	0.7915
5	10	10	auto	2	0.7156	0.6964	0.7452	0.7200
6	10	None	sqrt	5	0.7813	0.8452	0.7634	0.8023
7	10	5	sqrt	5	0.7125	0.6964	0.7405	0.7178
8	10	10	auto	2	0.7906	0.7976	0.8024	0.8000
9	10	None	auto	2	0.7938	0.7917	0.8110	0.8012
10	10	5	sqrt	5	0.8000	0.7738	0.8333	0.8025
11	10	10	sqrt	5	0.7813	0.8095	0.7816	0.7953
12	50	None	auto	2	0.8625	0.8750	0.8647	0.8698
13	50	5	auto	2	0.8594	0.8571	0.8727	0.8649
14	50	10	sqrt	5	0.8656	0.8750	0.8698	0.8724
15	50	None	sqrt	5	0.8656	0.8750	0.8698	0.8724
16	50	5	auto	2	0.7813	0.7679	0.8063	0.7866
17	50	10	auto	2	0.8281	0.8036	0.8599	0.8308
18	50	None	sqrt	5	0.7906	0.7262	0.8531	0.7846
19	50	5	sqrt	5	0.7906	0.7798	0.8137	0.7964
20	50	10	auto	2	0.8406	0.8512	0.8462	0.8487

Tabel 4.14 Lanjutan Hasil Performa Skenario 3 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
21	50	None	auto	2	0.8594	0.8810	0.8555	0.8680
22	50	5	sqrt	5	0.8438	0.8333	0.8642	0.8485
23	50	10	sqrt	5	0.8313	0.8393	0.8393	0.8393
24	100	None	auto	2	0.8688	0.8929	0.8621	0.8772
25	100	5	auto	2	0.8688	0.9048	0.8539	0.8786
26	**100**	**10**	**sqrt**	**5**	**0.8813**	**0.9048**	**0.8736**	**0.8889**
27	100	None	sqrt	5	0.8750	0.8750	0.8855	0.8802
28	100	5	auto	2	0.8250	0.7917	0.8636	0.8261
29	100	10	auto	2	0.8125	0.7917	0.8418	0.8160
30	100	None	sqrt	5	0.8250	0.8214	0.8415	0.8313
31	100	5	sqrt	5	0.8188	0.7917	0.8526	0.8210
32	100	10	auto	2	0.8531	0.8690	0.8538	0.8614
33	100	None	auto	2	0.8469	0.8690	0.8439	0.8563
34	100	5	sqrt	5	0.8375	0.8631	0.8333	0.8480
35	100	10	sqrt	5	0.8688	0.8929	0.8621	0.8772

Pada skenario ketiga, peneliti telah mengadopsi pendekatan yang menggabungkan metode n-gram dan chi-square untuk melakukan klasifikasi ulasan palsu. Dalam pendekatan ini, terlihat peningkatan signifikan pada akurasi model Random Forest, dari 0.8594 menjadi 0.8813. Pendekatan ini sejalan dengan saran yang diajukan oleh (Arum Sari, 2018), yang menganggap kata-kata slang dalam dokumen dapat mempengaruhi nilai keterkaitan fitur atau kata tersebut. Oleh karena itu, penggunaan metode n-gram membantu dalam seleksi fitur dan meningkatkan akurasi secara keseluruhan. Hasil penelitian ini menunjukkan bahwa penggabungan metode n-gram dan chi-square dalam model Random Forest secara signifikan meningkatkan akurasi. Peningkatan sebesar 0.0219 menunjukkan peningkatan

kemampuan model dalam mengklasifikasikan ulasan palsu. Pendekatan ini memberikan kontribusi penting dalam meningkatkan kualitas klasifikasi ulasan palsu dengan mempertimbangkan konteks kata dan hubungan fitur secara lebih baik. Dengan demikian, pendekatan ini dapat dianggap sebagai langkah yang efektif dalam meningkatkan kinerja model dalam mengenali dan membedakan ulasan palsu.

Pada skenario keempat, peneliti mengadopsi pendekatan yang lebih canggih dalam klasifikasi ulasan palsu dengan menggunakan kombinasi metode TF-IDF, n-gram model, information gain, dan algoritma Random Forest. Pertama, peneliti menggunakan metode TF-IDF untuk mengukur tingkat kepentingan kata-kata dalam ulasan. Dengan memperhitungkan frekuensi kemunculan kata tersebut dalam seluruh dataset, metode TF-IDF membantu dalam mengidentifikasi kata-kata kunci yang dapat membedakan ulasan palsu dan bukan palsu. Selanjutnya, peneliti menerapkan n-gram model untuk memperhatikan konteks dan urutan kata dalam ulasan. Dengan memperhitungkan beberapa kata secara bersama-sama, n-gram model dapat menangkap pola-pola yang lebih kompleks dalam teks, memungkinkan pengklasifikasian ulasan dengan lebih akurat. Selanjutnya, peneliti menggunakan metode information gain untuk seleksi fitur. Information gain digunakan untuk mengukur sejauh mana suatu fitur (kata) memberikan informasi yang berguna dalam mengklasifikasikan ulasan. Dengan menggunakan metode ini, peneliti dapat mengidentifikasi fitur-fitur yang memiliki nilai informasi tinggi dan memilih hanya fitur-fitur yang paling informatif untuk digunakan dalam model klasifikasi. Terakhir, peneliti menerapkan algoritma Random Forest. Random

Forest adalah metode ensemble yang terdiri dari beberapa pohon keputusan. Setiap pohon dalam Random Forest memberikan kontribusi dalam mengklasifikasikan ulasan berdasarkan fitur-fitur yang mereka tangkap dari dataset. Melalui proses voting mayoritas, hasil akhir klasifikasi diperoleh.

Tabel 4.15 Hasil Performa Skenario 4 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
0	10	None	auto	2	0.7969	0.7262	0.8652	0.7896
1	10	5	auto	2	0.8375	0.8274	0.8580	0.8424
2	10	10	sqrt	5	0.7813	0.7143	0.8451	0.7742
3	10	None	sqrt	5	0.7656	0.7083	0.8207	0.7604
4	10	5	auto	2	0.7438	0.7083	0.7829	0.7438
5	10	10	auto	2	0.7344	0.7024	0.7712	0.7352
6	10	None	sqrt	5	0.7375	0.6845	0.7877	0.7325
7	10	5	sqrt	5	0.7406	0.7143	0.7742	0.7430
8	10	10	auto	2	0.7656	0.7321	0.8039	0.7664
9	10	None	auto	2	0.7688	0.7440	0.8013	0.7716
10	10	5	sqrt	5	0.7250	0.7321	0.7410	0.7365
11	10	10	sqrt	5	0.7563	0.6845	0.8214	0.7468
12	50	None	auto	2	0.8219	0.7738	0.8725	0.8202
13	50	5	auto	2	0.8563	0.8512	0.8720	0.8614
14	50	10	sqrt	5	0.8344	0.8214	0.8571	0.8389
15	50	None	sqrt	5	0.8469	0.8333	0.8696	0.8511
16	50	5	auto	2	0.8375	0.8512	0.8412	0.8462
17	50	10	auto	2	0.8156	0.7917	0.8471	0.8185
18	50	None	sqrt	5	0.8188	0.7738	0.8667	0.8176
19	50	5	sqrt	5	0.8406	0.8274	0.8634	0.8450
20	50	10	auto	2	0.8438	0.8274	0.8688	0.8476
21	50	None	auto	2	0.8500	0.8333	0.8750	0.8537
22	50	5	sqrt	5	0.8594	0.8512	0.8773	0.8640
23	50	10	sqrt	5	0.8500	0.8571	0.8571	0.8571
24	100	None	auto	2	0.8656	0.8571	0.8834	0.8701
25	100	5	auto	2	0.8594	0.8631	0.8683	0.8657

Tabel 4.15 Lanjutan Hasil Performa Skenario 4 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
26	**100**	**10**	**sqrt**	**5**	**0.8688**	**0.8631**	**0.8841**	**0.8735**
27	100	None	sqrt	5	0.8563	0.8512	0.8720	0.8614
28	100	5	auto	2	0.8344	0.8214	0.8571	0.8389
29	100	10	auto	2	0.8438	0.8393	0.8598	0.8494
30	100	None	sqrt	5	0.8406	0.8155	0.8726	0.8431
31	100	5	sqrt	5	0.8375	0.8274	0.8580	0.8424
32	100	10	auto	2	0.8250	0.8095	0.8500	0.8293
33	100	None	auto	2	0.8438	0.8393	0.8598	0.8494
34	100	5	sqrt	5	0.8500	0.8452	0.8659	0.8554
35	100	10	sqrt	5	0.8281	0.8274	0.8424	0.8348

Setelah melihat keberhasilan skenario ketiga, peneliti kami memutuskan untuk melanjutkan dengan skenario keempat yang menggunakan metode information gain sebagai alternatif untuk meningkatkan akurasi dari model Random Forest. Information gain digunakan untuk mengukur seberapa besar informasi yang diberikan oleh fitur dalam pemisahan kelas-kelas yang berbeda. Dalam implementasinya, fitur-fitur dengan information gain tertinggi dipilih untuk membangun model Random Forest. Namun, hasil dari skenario keempat menunjukkan akurasi sebesar 0.8688, sedikit lebih rendah dibandingkan dengan skenario ketiga yang mencapai akurasi sebesar 0.8813. Meskipun demikian, skenario keempat masih dapat dianggap berhasil karena mampu meningkatkan akurasi dari nilai awal Random Forest sebesar 0.8594 menjadi 0.8688. Meskipun tidak mencapai akurasi tertinggi, penggunaan metode information gain tetap memberikan kontribusi positif dalam meningkatkan kinerja model Random Forest dalam mengklasifikasikan ulasan palsu.

Pada skenario kelima, peneliti mengadopsi pendekatan menggunakan metode TF-IDF, Chi-square, dan algoritma Random Forest dalam melakukan klasifikasi ulasan palsu. Metode TF-IDF (Term Frequency-Inverse Document Frequency) digunakan untuk mengevaluasi kepentingan kata-kata dalam ulasan. Dengan mempertimbangkan frekuensi kemunculan kata dalam ulasan dan seluruh dataset, metode ini membantu mengidentifikasi kata-kata yang memiliki peran penting dalam membedakan ulasan palsu dan ulasan yang bukan palsu. Selain itu, peneliti menerapkan metode Chi-square sebagai metode seleksi fitur. Chi-square digunakan untuk mengukur keterkaitan antara keberadaan suatu fitur (kata) dengan kelas klasifikasi (ulasan palsu atau bukan palsu). Dengan menggunakan metode ini, peneliti dapat menemukan fitur-fitur yang memiliki korelasi yang signifikan dengan klasifikasi ulasan palsu. Terakhir, peneliti menggunakan algoritma Random Forest untuk membangun model klasifikasi. Algoritma ini adalah metode ensemble yang menggabungkan beberapa pohon keputusan. Setiap pohon dalam Random Forest memberikan kontribusi dalam mengklasifikasikan ulasan berdasarkan fitur-fitur yang mereka tangkap dari dataset. Melalui kombinasi hasil voting dari pohon-pohon tersebut, keputusan klasifikasi akhir diambil.

Tabel 4.16 Hasil Performa Skenario 5 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
0	10	None	auto	2	0.7844	0.6964	0.8667	0.7723
1	10	5	auto	2	0.8125	0.7976	0.8375	0.8171
2	10	10	sqrt	5	0.7938	0.7202	0.8643	0.7857
3	10	None	sqrt	5	0.7813	0.7500	0.8182	0.7826
4	10	5	auto	2	0.7750	0.7083	0.8380	0.7677
5	10	10	auto	2	0.7531	0.8155	0.7405	0.7762

Tabel 4.16 Lanjutan Hasil Performa Skenario 5 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
6	10	None	sqrt	5	0.7594	0.7440	0.7862	0.7645
7	10	5	sqrt	5	0.7750	0.7560	0.8038	0.7791
8	10	10	auto	2	0.7625	0.7143	0.8108	0.7595
9	10	None	auto	2	0.7688	0.7321	0.8092	0.7688
10	10	5	sqrt	5	0.7750	0.7679	0.7963	0.7818
11	10	10	sqrt	5	0.8000	0.7500	0.8514	0.7975
12	50	None	auto	2	0.8469	0.8036	0.8940	0.8464
13	50	5	auto	2	0.8625	0.8452	0.8875	0.8659
14	50	10	sqrt	5	0.8281	0.8095	0.8553	0.8318
15	50	None	sqrt	5	0.8344	0.8214	0.8571	0.8389
16	50	5	auto	2	0.8281	0.8214	0.8466	0.8338
17	50	10	auto	2	0.8219	0.8274	0.8323	0.8299
18	50	None	sqrt	5	0.7656	0.7143	0.8163	0.7619
19	50	5	sqrt	5	0.8281	0.8274	0.8424	0.8348
20	50	10	auto	2	0.8188	0.8095	0.8395	0.8242
21	50	None	auto	2	0.8438	0.8452	0.8554	0.8503
22	50	5	sqrt	5	0.8406	0.8512	0.8462	0.8487
23	50	10	sqrt	5	0.8281	0.8393	0.8343	0.8368
24	100	None	auto	2	0.8594	0.8512	0.8773	0.8640
25	100	5	auto	2	0.8625	0.8690	0.8690	0.8690
26	100	10	sqrt	5	0.8469	0.8274	0.8742	0.8502
27	100	None	sqrt	5	0.8563	0.8631	0.8631	0.8631
28	100	5	auto	2	0.8250	0.7976	0.8590	0.8272
29	100	10	auto	2	0.8406	0.8155	0.8726	0.8431
30	100	None	sqrt	5	0.8438	0.8274	0.8688	0.8476
31	100	5	sqrt	5	0.8156	0.7976	0.8428	0.8196
32	**100**	**10**	**auto**	**2**	**0.8750**	**0.8750**	**0.8855**	**0.8802**
33	100	None	auto	2	0.8375	0.8393	0.8494	0.8443
34	100	5	sqrt	5	0.8344	0.8393	0.8443	0.8418
35	100	10	sqrt	5	0.8625	0.8571	0.8780	0.8675

Pada skenario kelima, peneliti melakukan eksperimen yang melibatkan gabungan antara metode chi-square dan Random Forest dengan tujuan meningkatkan akurasi dari model Random Forest yang digunakan. Pendekatan ini didasarkan pada penelitian sebelumnya yang dilakukan oleh (Gbenga et al., 2021) yang berhasil meningkatkan akurasi model Random Forest. Dalam eksperimen tersebut, ditemukan bahwa penggunaan kombinasi metode chi-square dan Random Forest berhasil meningkatkan akurasi model Random Forest. Awalnya, akurasi dari Random Forest sebesar 0.8594. Namun, setelah menerapkan pendekatan tersebut, akurasi meningkat menjadi 0.8750. Hasil ini sejalan dengan temuan penelitian sebelumnya dan menunjukkan bahwa penggabungan metode chi-square dan Random Forest merupakan pendekatan yang efektif dalam meningkatkan performa model klasifikasi. Penemuan ini memberikan kontribusi penting dalam pengembangan metode klasifikasi yang lebih baik dan dapat diandalkan dalam mengklasifikasikan data dengan akurasi yang lebih tinggi.

Pada skenario keenam, peneliti menggunakan metode TF-IDF dan Information Gain dalam kombinasi dengan algoritma Random Forest untuk melakukan klasifikasi ulasan palsu. Metode TF-IDF (Term Frequency-Inverse Document Frequency) digunakan untuk mengukur pentingnya kata-kata dalam ulasan. Dengan mempertimbangkan frekuensi kemunculan kata dalam ulasan dan seluruh dataset, metode ini membantu mengidentifikasi kata-kata kunci yang dapat membedakan ulasan palsu dan ulasan yang bukan palsu. Selanjutnya, peneliti menerapkan metode Information Gain untuk seleksi fitur. Information Gain digunakan untuk mengukur seberapa informatifnya suatu fitur (kata) dalam

mengklasifikasikan ulasan. Dengan menggunakan metode ini, peneliti dapat mengidentifikasi fitur-fitur yang memberikan kontribusi signifikan dalam membedakan ulasan palsu. Terakhir, peneliti menggunakan algoritma Random Forest untuk membangun model klasifikasi. Random Forest adalah metode ensemble yang terdiri dari beberapa pohon keputusan. Setiap pohon memberikan kontribusi dalam mengklasifikasikan ulasan berdasarkan fitur-fitur yang mereka tangkap dari dataset. Melalui proses voting mayoritas dari pohon-pohon tersebut, hasil klasifikasi akhir diperoleh.

Tabel 4.17 Hasil Performa Skenario 6 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
0	10	None	auto	2	0.7813	0.7202	0.8403	0.7756
1	10	5	auto	2	0.8094	0.7679	0.8543	0.8088
2	10	10	sqrt	5	0.7719	0.7262	0.8188	0.7697
3	10	None	sqrt	5	0.7594	0.6905	0.8227	0.7508
4	10	5	auto	2	0.7563	0.7560	0.7744	0.7651
5	10	10	auto	2	0.7281	0.7024	0.7613	0.7307
6	10	None	sqrt	5	0.7531	0.6964	0.8069	0.7476
7	10	5	sqrt	5	0.7406	0.6905	0.7891	0.7365
8	10	10	auto	2	0.7750	0.7560	0.8038	0.7791
9	10	None	auto	2	0.7813	0.7500	0.8182	0.7826
10	10	5	sqrt	5	0.7563	0.7679	0.7679	0.7679
11	10	10	sqrt	5	0.7438	0.7262	0.7722	0.7485
12	50	None	auto	2	0.8594	0.8274	0.8968	0.8607
13	50	5	auto	2	0.8469	0.8512	0.8563	0.8537
14	50	10	sqrt	5	0.8406	0.8393	0.8545	0.8468
15	50	None	sqrt	5	0.8406	0.8095	0.8774	0.8421
16	50	5	auto	2	0.8031	0.7738	0.8387	0.8050
17	50	10	auto	2	0.8063	0.7500	0.8630	0.8025
18	50	None	sqrt	5	0.8094	0.7917	0.8365	0.8135
19	50	5	sqrt	5	0.8063	0.7560	0.8581	0.8038

Tabel 4.17 Lanjutan Hasil Performa Skenario 6 Model Random Forest

	n_estimators	max depth	max features	min samples split	Accuracy	Recall	Precision	F1-Score
20	50	10	auto	2	0.8375	0.8333	0.8537	0.8434
21	50	None	auto	2	0.8156	0.7857	0.8516	0.8173
22	50	5	sqrt	5	0.8250	0.8571	0.8182	0.8372
23	50	10	sqrt	5	0.8313	0.8214	0.8519	0.8364
24	**100**	**None**	**auto**	**2**	**0.8625**	**0.8631**	**0.8735**	**0.8683**
25	100	5	auto	2	0.8469	0.8452	0.8606	0.8529
26	100	10	sqrt	5	0.8500	0.8452	0.8659	0.8554
27	**100**	**None**	**sqrt**	**5**	**0.8656**	**0.8512**	**0.8882**	**0.8693**
28	100	5	auto	2	0.8031	0.7798	0.8344	0.8062
29	100	10	auto	2	0.8313	0.8036	0.8654	0.8333
30	100	None	sqrt	5	0.8219	0.7857	0.8627	0.8224
31	100	5	sqrt	5	0.8531	0.8452	0.8712	0.8580
32	100	10	auto	2	0.8469	0.8274	0.8742	0.8502
33	100	None	auto	2	0.8281	0.8393	0.8343	0.8368
34	100	5	sqrt	5	0.8469	0.8155	0.8839	0.8483
35	100	10	sqrt	5	0.8344	0.8155	0.8616	0.8379

Pada skenario keenam, peneliti telah mengadopsi pendekatan yang melibatkan penggunaan metode information gain untuk meningkatkan akurasi dari model Random Forest yang digunakan. Pendekatan ini telah sebelumnya diuji dan terbukti berhasil dalam penelitian yang dilakukan oleh (Prasetiyowati et al., 2022) yang juga menggunakan metode information gain untuk meningkatkan akurasi dari Random Forest. Hasil penelitian ini konsisten dengan temuan sebelumnya, menunjukkan bahwa penggunaan metode information gain dalam kombinasi dengan Random Forest dapat signifikan meningkatkan akurasi model Random Forest. Awalnya, akurasi model Random Forest sebesar 0.8594. Namun, setelah menerapkan metode information gain untuk seleksi fitur yang relevan, terjadi

peningkatan yang signifikan sebesar 0.0062, yaitu dari 0.8594 menjadi 0.8656. Temuan ini menunjukkan bahwa penggunaan metode information gain memiliki dampak positif dalam meningkatkan kemampuan prediksi dan klasifikasi dari model Random Forest yang digunakan dalam penelitian ini. Hal ini memberikan pemahaman yang lebih mendalam tentang pentingnya seleksi fitur yang tepat dan penggunaan metode yang sesuai dalam meningkatkan performa model klasifikasi.

Setelah mencari hasil terbaik dari masing-masing skenario pada model Random Forest, langkah selanjutnya adalah membandingkan hasil tersebut antara skenario-skenario untuk menentukan skenario mana yang memberikan performa terbaik pada model Random Forest. Berikut adalah hasil klasifikasi model Random Forest:

Tabel 4.18 Performa evaluasi algoritma Random Forest

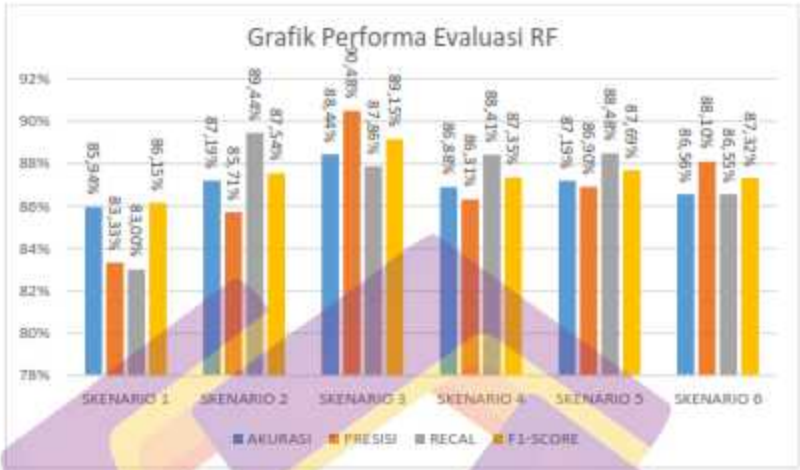
Skenario	Random Forest	TF-IDF	N-gram	Feature selection		Akurasi	Precision	Recall	F1-Score
				CS	IG				
1	✓	✓				0.8594	0.8333	0.8917	0.8615
2	✓	✓	✓			0.8719	0.8690	0.8848	0.8769
3	✓	✓	✓	✓		**0.8813**	**0.9048**	**0.8736**	**0.8889**
4	✓	✓	✓		✓	0.8688	0.8631	0.8841	0.8735
5	✓	✓		✓		0.8750	0.8750	0.8855	0.8802
6	✓	✓			✓	0.8656	0.8512	0.8882	0.8693

Berdasarkan hasil tabel, dapat disimpulkan bahwa kombinasi fitur dan teknik feature selection dalam Random Forest berpengaruh signifikan terhadap kinerja model dalam klasifikasi ulasan palsu. Hasil evaluasi menunjukkan bahwa skenario ke-tiga merupakan skenario terbaik dengan menggunakan fitur TF-IDF, N-gram, dan Chi-Square (CS), menghasilkan akurasi tertinggi sebesar 0.8813.

Model ini juga memiliki nilai precision sebesar 0.9048, recall sebesar 0.8736, dan F1-Score sebesar 0.8889.

Pada skenario lainnya, terdapat variasi dalam performa model. Skenario ke-dua, yang menggunakan fitur TF-IDF dan N-gram dengan akurasi 0.8719, juga menunjukkan hasil yang cukup baik dengan nilai precision sebesar 0.8690, recall sebesar 0.8848, dan F1-Score sebesar 0.8769. Sementara itu, skenario ke-empat dengan fitur TF-IDF, N-gram, dan feature selection (IG) memiliki akurasi 0.8688 dan skenario ke-lima dengan fitur TF-IDF dan feature selection (IG) memiliki akurasi 0.8750. Dalam skenario lainnya, seperti skenario pertama dan ke-enam, meskipun terdapat perbedaan performa, penelitian ini tetap berhasil meningkatkan akurasi model Random Forest dari akurasi awal 0.8594 menjadi 0.8656. Hasil ini menunjukkan bahwa penggunaan metode feature selection, seperti Chi-Square dan Information Gain, memiliki dampak positif dalam meningkatkan kemampuan prediksi dan klasifikasi dari model Random Forest yang digunakan dalam penelitian ini. Secara keseluruhan, temuan ini mendukung penelitian sebelumnya dan menunjukkan bahwa penggunaan kombinasi fitur dan teknik feature selection yang optimal dapat memberikan peningkatan yang signifikan dalam akurasi klasifikasi ulasan palsu menggunakan model Random Forest.

Untuk mempermudah visualisasi informasi, data tersebut akan disajikan dalam bentuk grafik. Grafik tersebut akan memberikan representasi yang lebih jelas dan mudah dipahami tentang performa evaluali yang telah dilakukan. Berikut ini adalah visualisasi performa evaluasi Random Forest dalam bentuk grafik



Gambar 4.5 Grafik Performa Evaluasi Random Forest

4.6. Analisis hasil klasifikasi

Seluruh hasil pengujian model SVM dan Random Forest akan dijabarkan secara detail pada gambar 4.4.

Gambar 4.6 Perbandingan SVM dan Random Forest

Berdasarkan hasil pada gambar , dapat disimpulkan bahwa pada skenario-skenario yang ada, SVM memiliki performa yang lebih baik dalam hal akurasi dibandingkan dengan Random Forest. Pada skenario terbaiknya (Skenario 3), SVM mencapai akurasi sebesar 0.928125, sedangkan Random Forest hanya mencapai akurasi 0.884375 dalam skenario terbaiknya (Skenario 3). Hal ini menunjukkan bahwa SVM memiliki kemampuan yang lebih baik dalam melakukan klasifikasi dengan tingkat akurasi yang lebih tinggi. Meskipun terdapat variasi dalam performa antara skenario-skenario yang berbeda, SVM cenderung memberikan akurasi yang lebih tinggi secara konsisten dibandingkan dengan Random Forest. Salah satu alasan mengapa SVM memiliki performa yang lebih baik dapat dikaitkan dengan kekuatan SVM dalam menangani data yang kompleks dan memiliki pemisahan yang jelas antara kelas-kelas yang berbeda. Dalam konteks ini, SVM mampu membangun batas keputusan yang lebih baik, sehingga menghasilkan akurasi yang lebih tinggi dalam klasifikasi data.

4.7. Membandingkan Dengan Penelitian Sebelumnya

Sebagai bagian dari perkembangan penelitian dibidang deteksi ulasan palsu, penting untuk melihat penelitian-penelitian sebelumnya sebagai pembanding. Dalam konteks ini, penelitian yang dilakukan memiliki tujuan untuk meningkatkan hasil akurasi yang didapat oleh penelitian sebelumnya dengan menambahkan feature selection. Untuk lebih jelas dapat dilihat pada tabel 4.20

Tabel 4.19 Perbandingan dengan penelitian sebelumnya

Peneliti	Feature	Algoritma	Classifier	Accuracy	Precesion	Recall	F1-Score
(Rout et al., 2017)	Bigrams, Sentimen Score, POS, LIWC	Semi Supervised	Logistic Regresion (PU)	0.8375	0.8313	0.8418	0.8365
			K-NN with (EM)	0.8313	0.8063	0.8487	0.8269
(Hassan & Islam, 2019)	Word requency count, review length, entiment score	Semi-Supervised	Naïve Bayes	0.8521	-	-	-
			SVM	0.8134	-	-	-
		Supervised	Naïve Bayes	0.8632	-	-	-
			SVM	0.8228	-	-	-
(Hassan & Islam, 2020)	TF-IDF, Empath Categories, Score Sentimen	supervised	Naïve Bayes	0.8437	0.9453	0.7378	0.8287
			Logistic Regresion	0.8843	0.8802	0.8963	0.8882
			SVM	0.8875	0.9050	0.8719	0.8881
Proposed Work	TF-IDF	supervised	SVM	0.8875	0.8810	0.9024	0.8916
			Random Forest	0.8594	0.8333	0.8917	0.8615
	TF-IDF Trigram	supervised	SVM	0.8938	0.8810	0.9136	0.8970
			Random Forest	0.8719	0.8690	0.8848	0.8769
	TF-IDF, Trigram, CS	supervised	SVM	**0.9281**	**0.9405**	**0.9240**	**0.9322**
			Random Forest	**0.8813**	**0.9048**	**0.8736**	**0.8889**
	TF-IDF, Trigram, IG	supervised	SVM	0.8938	0.8988	0.8988	0.8988
			Random Forest	0.8688	0.8631	0.8841	0.8735
	TF-IDF, CS	supervised	SVM	0.9188	0.9226	0.9226	0.9226
			Random Forest	0.8750	0.8750	0.8855	0.8802
	TF-IDF, IG	supervised	SVM	0.8906	0.8810	0.9080	0.8943
			Random Forest	0.8656	0.8512	0.8882	0.8693

Pada tabel di atas, terlihat dengan jelas bahwa penelitian yang sedang dilakukan saat ini menghasilkan hasil yang lebih baik dibandingkan dengan penelitian yang dilakukan oleh (Hassan & Islam, 2020). Hasan menggunakan metode TF-IDF, Empath Categories, dan sentimen score dalam penelitiannya, yang menghasilkan akurasi sebesar 0.8875. Namun, penelitian yang sedang berlangsung saat ini telah mengadopsi pendekatan yang lebih maju dan canggih.

Dalam penelitian ini, peneliti menggunakan metode TF-IDF sebagai dasar ekstraksi fitur, namun tidak berhenti di situ. peneliti juga mengintegrasikan pendekatan n-gram dan feature selection untuk meningkatkan akurasi klasifikasi. Hasil tertinggi didapatkan oleh pendekatan tri-gram dan chi-square dengan akurasi

sebesar 0.9281 dengan menggunakan algoritma klasifikasi SVM. Pendekatan tersebut berhasil meningkatkan akurasi dari penelian yang dilakukan oleh (Hassan & Islam, 2020) sebesar 0.0406. Hasil menunjukkan peningkatan yang signifikan dibandingkan dengan penelitian(Hassan & Islam, 2020). Peningkatan ini menunjukkan bahwa pendekatan yang diusulkan dalam penelitian saat ini mampu mengatasi beberapa kelemahan dari metode sebelumnya. Dengan menggabungkan TF-IDF dengan n-gram dan chi-square, peneliti dapat mengakses informasi lebih lengkap dan kompleks dari teks ulasan, sehingga meningkatkan kemampuan sistem dalam mengklasifikasikan ulasan dengan lebih akurat. Penemuan ini memiliki implikasi penting dalam penelitian ulasan palsu di bidang ecommerce. Dengan menerapkan pendekatan yang lebih maju, penelitian saat ini memberikan kontribusi signifikan dalam pengembangan metode dan teknik untuk memahami dan menganalisis ulasan palsu pada sebuah diproduk.

Agar informasi lebih mudah dipahami, data tersebut akan dipresentasikan dalam bentuk grafik. Grafik tersebut akan memberikan representasi yang jelas tentang hasil evaluasi tertinggi dari setiap penelitian sebelumnya. Berikut ini adalah gambar grafik perbandingan dengan penelitian sebelumnya.

Gambar 4.7 Perbandingan dengan penelitian sebelumnya

Gambar di atas hanya menampilkan hasil akurasi tertinggi yang didapatkan dari serangkaian percobaan yang telah dilakukan oleh masing-masing penelitian sebelumnya. Tujuan dari visualisasi ini adalah untuk menghadirkan informasi yang paling relevan dan signifikan, sehingga memungkinkan pembaca untuk dengan cepat dan mudah memahami pencapaian tertinggi yang telah dicapai dalam berbagai percobaan.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil yang telah dicapai dalam penelitian mengenai deteksi ulasan palsu menggunakan random forest dan SVM, dapat disimpulkan beberapa hal sebagai berikut:

1. Algoritma SVM menunjukkan tingkat akurasi yang lebih tinggi dibandingkan dengan algoritma Random Forest. Hasil ini dapat dilihat pada gambar 4.4. pada scenario 1. Perbedaan akurasinya sebesar 0.0281 (2.81%), dengan Random Forest mencapai akurasi sebesar 0.8594 (85.94%) dan SVM sebesar 0.8875 (88.75%). Hal ini menunjukkan bahwa SVM lebih efektif dalam mengenali dan membedakan ulasan palsu dari ulasan asli.
2. Berdasarkan hasil eksperimen, penggunaan model n-gram berhasil meningkatkan akurasi deteksi ulasan palsu pada algoritma Random Forest dan SVM. Hasil ini dapat dilihat pada gambar 4.4. pada scenario 2. Penggunaan n-gram model meningkatkan akurasi SVM sebesar 0.0063 (0.63%) dari 0.8875 (88.75%) menjadi 0.8938 (89.38%), sedangkan akurasi algoritma Random Forest meningkat sebesar 0.0125 (1.25%), dari 0.8594 (85.94%) menjadi 0.8719 (87.19%). Hal ini menunjukkan bahwa dengan mempertimbangkan konteks kata-kata secara keseluruhan, baik Random Forest maupun SVM dapat mengenali pola dan fitur yang relevan untuk mengklasifikasikan ulasan palsu dengan lebih akurat.

3. Berdasarkan hasil eksperimen, dapat disimpulkan bahwa penggunaan seleksi fitur berhasil secara signifikan meningkatkan akurasi deteksi ulasan palsu pada algoritma Random Forest dan SVM. Hasil ini dapat dilihat pada gambar 4.4. pada scenario 3 Penggunaan metode Chi-Square pada SVM menghasilkan peningkatan akurasi sebesar 0.0406 (4.06%), dari 0.8875 (88.75%) menjadi 0.9281 (92.81%), sedangkan pada algoritma Random Forest juga terjadi peningkatan sebesar 0.0219 (2.19%), dari 0.8594 (85.94%) menjadi 0.8813 (88.13%). Selain itu, peneliti juga melakukan percobaan menggunakan metode Information Gain untuk meningkatkan akurasi kedua algoritma tersebut, dan hasilnya juga berhasil meningkatkan akurasi keduanya. Pada SVM, Hasil ini dapat dilihat pada gambar 4.4. pada scenario 4. penggunaan Information Gain menghasilkan peningkatan akurasi sebesar 0.0063 (0.63%), dari 0.8875 (88.75%) menjadi 0.8938 (89.38%), sementara pada algoritma Random Forest terjadi peningkatan sebesar 0.0094 (0.94%), dari 0.8594 (85.94%) menjadi 0.8688 (86.88%). Meskipun peningkatan akurasi yang dihasilkan oleh Information Gain tidak terlalu signifikan, hasil ini tetap dapat dianggap berhasil. Hasil ini menunjukkan bahwa penggunaan seleksi fitur, baik dengan metode Chi-Square maupun Information Gain, berhasil secara signifikan meningkatkan akurasi deteksi ulasan palsu pada algoritma Random Forest dan SVM. Hal ini membuktikan bahwa seleksi fitur memainkan peran penting dalam meningkatkan kemampuan algoritma dalam mengenali dan membedakan ulasan palsu dengan lebih akurat

5.2. Saran

Ada beberapa saran yang direkomendasikan untuk penelitian selanjutnya pada topik penelitian yang sama, antara lain:

1. Penelitian selanjutnya dapat memfokuskan pada eksplorasi metode feature selection lainnya: Selain chi-square dan information gain, ada berbagai metode feature selection yang dapat dipertimbangkan, seperti mutual information, Recursive Feature Elimination (RFE), LASSO (Least Absolute Shrinkage and Selection Operator), dan principal component analysis (PCA). Melakukan eksplorasi terhadap metode-metode tersebut dapat membantu mengidentifikasi fitur-fitur yang lebih relevan dan meningkatkan akurasi deteksi ulasan palsu.
2. Penelitian selanjutnya dapat menerapkan pendekatan majority voting dalam kombinasi algoritma: Menggabungkan hasil dari beberapa algoritma, seperti Random Forest, Naïve Bayes dan SVM, melalui teknik majority voting, dapat meningkatkan kehandalan dan performa deteksi ulasan palsu. Dengan menggunakan majority voting, prediksi dari berbagai algoritma ensemble dapat digabungkan untuk menghasilkan keputusan akhir yang lebih konsisten dan akurat.

DAFTAR PUSTAKA

PUSTAKA BUKU

Arikunto, S. (2013). *Prosedur Penelitian Suatu Pendekatan Praktik.* Jakarta: PT. Rineka Cipta.

Basrowi, S. (2018). *Memahami penelitian kualitatif.* Jakarta: Rineka Cipta.

Hanafi, A. (2009). Pengenalan bahasa suku bangsa indonesia berbasis teks menggunakan metode n-gram. *IT TELKOM.*

Kusrini, E. T. (2009). *Algoritma data mining.* Yogyakarta: CV Andi Offset.

Muhammad Arsyam, M. Y. (2021). Ragam Jenis Penelitian dan Perspektif. *AUJPSI*, 8.

Prasetyo, E. (2012). *Data mining : konsep dan aplikasi menggunakan MATLAB.* Yogyakarta: CV Andi Offset.

S. Fachrurrozi, M. G. (2021). Increasing Accuracy of Support Vector Machine (SVM) By Applying N-Gram and Chi-Square Feature Selection for Text Classification. *021 International Seminar on Application for Technology of Information and Communication (iSemantic).* Semarangin, Indonesia: IEEE.

Santosa, B. (2007). *Data mining : Teknik pemanfaatan data untuk keperluan bisnis.* Yogyakarta: Garah Ilmu.

Sugiyono. (2013). *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif dan R&D.* Bandung: Alfabeta.

Sugiyono. (2018). *Metode Penelitian Kuantitatif, Kualitatif dan R&D.* Bandung: PT. Alfabet.

Sugiyono. (2018). *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: PT. Alfabet.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Abhinandan V., Aishwarya C. A., & Sultana, A. (2020). Fake Review Detection Using Machine Learning Techniques. *International Journal of Fog Computing*, *3*(2), 46–54. https://doi.org/10.4018/ijfc.2020070104

Akkaya, B. (2021). The Effect of Recursive Feature Elimination with Cross-Validation Method on Classification Performance with Different Sizes of Datasets. *IV International Conference on Data Science and Applications (ICONDATA'21)*, *June*, 4–6. https://www.researchgate.net/publication/354253728_The_Effect_of_Recursive_Feature_Elimination_with_Cross-Validation_Method_on_Classification_Performance_with_Different_Sizes_of_Datasets

Alsubari, S. N., Deshmukh, S. N., Al-Adhaileh, M. H., Alsaade, F. W., & Aldhyani, T. H. H. (2021). Development of Integrated Neural Network Model for Identification of Fake Reviews in E-Commerce Using Multidomain Datasets. *Applied Bionics and Biomechanics*, *2021*. https://doi.org/10.1155/2021/5522574

Alsubari, S. N., Deshmukh, S. N., Alqarni, A. A., Alsharif, N., Aldhyani, T. H. H., Alsaade, F. W., & Khalaf, O. I. (2022). Data analytics for the identification of fake reviews using supervised learning. *Computers, Materials and Continua*,

70(2), 3189–3204. https://doi.org/10.32604/cmc.2022.019625

Arauzo-Azofra, A., Aznarte, J. L., & Benítez, J. M. (2011). Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications*, *38*(7), 8170–8177. https://doi.org/10.1016/j.eswa.2010.12.160

Arum Sari, Y. (2018). *Analisis Sentimen Pada Review Konsumen Menggunakan Metode Naive Bayes Dengan Seleksi Fitur Chi Square Untuk Rekomendasi Lokasi Makanan Tradisional Melanoma Identification View project Food Image Classification, Retrieval, and Analysis View project*. *July*. http://j-ptiik.ub.ac.id

Budhi, G. S., Chiong, R., & Wang, Z. (2021). Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features. *Multimedia Tools and Applications*, *80*(9), 13079–13097. https://doi.org/10.1007/s11042-020-10299-5

Chakraborty, U., & Biswal, S. K. (2020). Impact of Online Reviews on Consumer's Hotel Booking Intentions: Does Brand Image Mediate? *Journal of Promotion Management*, *26*(7), 943–963. https://doi.org/10.1080/10496491.2020.1746465

Christy, E., Suryowati, K., Statistika, J., Sains Terapan, F., & AKPRIND Yogyakarta, I. (2021). ANALISIS KLASIFIKASI STATUS BEKERJA PENDUDUK DAERAH ISTIMEWA YOGYAKARTA MENGGUNAKAN METODE RANDOM FOREST. *Jurnal Statistika Industri Dan Komputasi*, *6*(1), 69–76.

Dinas, K., Hidup, L., Kebersihan, D., Selong, K., Lombok, K., Yahya, T., & Zuliana, R. (2018). Prediksi Jumlah Penggunaan BBM Perbulan Menggunakan Algoritma Decition Tree (C4.5) Pada. *Jurnal Informatika Dan Teknologi*, *1*(1), 56–63. https://doi.org/10.29408/jit.v1i1.895

Doraisamy, S., Golzari, S., Norowi, N. M., Sulaiman, M. N. B., & Udzir, N. I. (2008). A study on feature selection and classification techniques for automatic genre classification of traditional malay music. *ISMIR 2008 - 9th International Conference on Music Information Retrieval*, 331–336.

El-Said, O. A. (2020). Impact of online reviews on hotel booking intention: The moderating role of brand image, star category, and price. *Tourism Management Perspectives*, *33*(March 2019), 100604. https://doi.org/10.1016/j.tmp.2019.100604

Elmogy, A. M., Tariq, U., Ibrahim, A., & Mohammed, A. (2021). Fake Reviews Detection using Supervised Machine Learning. *International Journal of Advanced Computer Science and Applications*, *12*(1), 601–606. https://doi.org/10.14569/IJACSA.2021.0120169

Forester, J. (1983). The geography of planning practice. *Environment & Planning D: Society & Space*, *1*(2), 163–180. https://doi.org/10.1068/d010163

Gavilan, D., Avello, M., & Martinez-Navarro, G. (2018). The influence of online ratings and reviews on hotel booking consideration. *Tourism Management*, *66*, 53–61. https://doi.org/10.1016/j.tourman.2017.10.018

Gbenga, *Fadare Oluwaseun, Olusola, A. A., Eloho, (Mrs) Oyinloye Oghenerukevwe, & Alaba, M. S. (2021). Towards Optimization of Malware

Detection using Chi-square Feature Selection on Ensemble Classifiers. *International Journal of Engineering and Advanced Technology*, *10*(4), 254–262. https://doi.org/10.35940/ijeat.d2359.0410421

Hancock, J. T., Curry, L. E., Goorha, S., & Woodworth, M. (2008). On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, *45*(1), 1–23. https://doi.org/10.1080/01638530701739181

Hassan, R., & Islam, M. R. (2019). Detection of fake online reviews using semi-supervised and supervised learning. *2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019*, 1–5. https://doi.org/10.1109/ECACE.2019.8679186

Hassan, R., & Islam, M. R. (2020). A Supervised Machine Learning Approach to Detect Fake Online Reviews. *ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings*, 19–21. https://doi.org/10.1109/ICCIT51783.2020.9392727

Jindal, N., & Liu, B. (2008). Opinion spam and analysis. *WSDM'08 - Proceedings of the 2008 International Conference on Web Search and Data Mining*, 219–229. https://doi.org/10.1145/1341531.1341560

Kotsiantis, S., & Tzelepis, D. (2006). *Forecasting fraudulent financial statements using data mining Ensemble methods View project A Big Data Scale Analysis Framework to Support Customized and Personalized Learning Environments View project*. https://www.researchgate.net/publication/228084523

Lisangan, E. A., Informasi, F. T., Atma, U., & Makassar, J. (n.d.). *NATURAL*

LANGUAGE PROCESSING DALAM MEMPEROLEH INFORMASI AKADEMIK MAHASISWA UNIVERSITAS ATMA JAYA MAKASSAR. 1–9.

Luthfiana, L., Young, J. C., & Rusli, A. (2020). Young, 2020. *Ultimatics*, *XII*(2), 125–128.

Madrakhimov, S., Makharov, K., & Lolaev, M. (2021). Data preprocessing on input. *AIP Conference Proceedings*, *2365*(July). https://doi.org/10.1063/5.0058132

Mohawesh, R., Xu, S., Tran, S. N., Ollington, R., Springer, M., Jararweh, Y., & Maqsood, S. (2021). Fake Reviews Detection: A Survey. *IEEE Access*, *9*, 65771–65802. https://doi.org/10.1109/ACCESS.2021.3075573

Moraes, R., Valiati, J. F., & Gavião Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621–633. https://doi.org/10.1016/j.eswa.2012.07.059

Mulyanto, A., Nurhuda, Y. A., & Wiyanto, N. (2017). Penyelesaian Kata Ambigu Pada Proses POS Tagging Menggunakan Algoritma Hidden markov Model (HMM). *Prosiding Seminar Nasional Metode Kuantitatif*, *978*, 347–358.

Munasatya, N., & Novianto, S. (2020). Natural Language Processing untuk Sentimen Analisis Presiden Jokowi Menggunakan Multi Layer Perceptron. *Techno.Com*, *19*(3), 237–244. https://doi.org/10.33633/tc.v19i3.3630

Novakovic, J. (2010). The Impact of Feature Selection on the Accuracy of Naive Bayes Classifier. *18th Telecommunication Forum TELFOR*, *2*, 1113–1116.

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative deceptive opinion spam.

NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference, *June*, 497–501.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *1*, 309–319.

Pacol, C. A., & Palaoag, T. D. (2021). Enhancing Sentiment Analysis of Textual Feedback in the Student-Faculty Evaluation using Machine Learning Techniques. *European Journal of Engineering Science and Technology*, *4*(1), 27–34. https://doi.org/10.33422/ejest.v4i1.604

Pasaribu, B. E., Herdiani, A., & Astuti, W. (2019). *Deteksi Fake Reviews Menggunakan Support Vector Machine*. *6*(2), 8788–8797.

Pisceldo, F., Adriani, M., & Manurung, R. (2009). Probabilistic Part of Speech Tagging for Bahasa Indonesia. *Proceedings of the 3rd International MALINDO Workshop, Colocated Event ACL-IJCNLP, January 2009*.

Prasetiyowati, M. I., Maulidevi, N. U., & Surendro, K. (2022). The accuracy of Random Forest performance can be improved by conducting a feature selection with a balancing strategy. *PeerJ Computer Science*, *8*, 1–15. https://doi.org/10.7717/PEERJ-CS.1041

Rajeswari, R., Satheesh Kumar, J., Devi, T., Bharathiar University. Department of Computer Applications, & Institute of Electrical and Electronics Engineers. (2016). *2016 IEEE International Conference on Advances in Computer*

Applications : 2016 IEEE ICACA : 24th October 2016. 18–20.

Ramadani, R. A., Ketut, I., Darma Putra, G., Ayu, I., & Giriantari, D. (n.d.). S Stemming Algorithm for Indonesian Signaling Systems (SIBI). *International Journal of Engineering and Emerging Technology*, *5*(1).

Rifai, W., & Winarko, E. (2019). Modification of Stemming Algorithm Using A Non Deterministic Approach To Indonesian Text. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *13*(4), 379. https://doi.org/10.22146/ijccs.49072

Rout, J. K., Dalmia, A., Choo, K. K. R., Bakshi, S., & Jena, S. K. (2017). Revisiting semi-supervised learning for online deceptive review detection. *IEEE Access*, *5*(c), 1319–1327. https://doi.org/10.1109/ACCESS.2017.2655032

Saadah, M. N., Atmagi, R. W., Rahayu, D. S., & Arifin, A. Z. (2013). Sistem Temu Kembali Dokumen Teks Dengan Pembobotan Tf-Idf Dan Lcs. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *11*(1), 19. https://doi.org/10.12962/j24068535.v11i1.a16

Setyohadi, D. B., & Kristiawan, F. A. (2017). PREPROCESSING ITERATIVE PARTITIONING FILTER ALGORITHM. In *TELEMATIKA* (Vol. 14, Issue 01).

Shuqin, Y., & Jing, F. (2019). Fake reviews detection based on text feature and behavior feature. *Proceedings - 21st IEEE International Conference on High Performance Computing and Communications, 17th IEEE International Conference on Smart City and 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019*, 2007–2012.

https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00277

Singh, K. P., Basant, N., & Gupta, S. (2011). Support vector machines in water quality management. *Analytica Chimica Acta*, *703*(2), 152–162. https://doi.org/10.1016/j.aca.2011.07.027

Somantri, O., & Apriliani, D. (2018). Support Vector Machine Berbasis Feature Selection Untuk Sentiment Analysis Kepuasan Pelanggan Terhadap Pelayanan Warung dan Restoran Kuliner Kota Tegal. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(5), 537. https://doi.org/10.25126/jtiik.201855867

Syukron, A., & Subekti, A. (2018). Penerapan Metode Random Over-Under Sampling dan Random Forest untuk Klasifikasi Penilaian Kredit. *JURNAL INFORMATIKA*, *5*(2).

Tong, X., Deacon, S. H., Kirby, J. R., Cain, K., & Parrila, R. (2011). Morphological awareness: A key to understanding poor reading comprehension in english. *Journal of Educational Psychology*, *103*(3), 523–534. https://doi.org/10.1037/a0023495

Violos, J., Tserpes, K., Varlamis, I., & Varvarigou, T. (2018). Text Classification Using the N-Gram Graph Representation Model Over High Frequency Data Streams. *Frontiers in Applied Mathematics and Statistics*, *4*(September), 1–19. https://doi.org/10.3389/fams.2018.00041

Vo, N. T., Hung, V. V., Tuckova, Z., Pham, N. T., & Nguyen, L. H. L. (2022). Guest Online Review: An Extraordinary Focus on Hotel Users' Satisfaction, Engagement, and Loyalty. *Journal of Quality Assurance in Hospitality and*

Tourism, *23*(4), 913–944. https://doi.org/10.1080/1528008X.2021.1920550

Yu, Y., Guo, X., Zhang, Y., & Zhao, H. (2016). *Online Review Impacts on Hotel Online Booking Decision*. *Emim*, 1370–1375. https://doi.org/10.2991/emim-16.2016.279

Zhang, K. Z. K., Zhao, S. J., Cheung, C. M. K., & Lee, M. K. O. (2014). Examining the influence of online reviews on consumers' decision-making: A heuristic-systematic model. *Decision Support Systems*, *67*, 78–89. https://doi.org/10.1016/j.dss.2014.08.005

LAMPIRAN