BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Tinjauan terhadap penelitian terkait dilakukan untuk memahami lanskap penelitian analisis sentimen, khususnya pada konteks aplikasi layanan keuangan digital seperti Kredivo. Tinjauan ini bertujuan untuk mengidentifikasi metodologi yang umum digunakan serta celah penelitian yang ada untuk diisi oleh penelitian ini. Beberapa penelitian terkait yang mendukung penelitian ini.

Tabel 2. 1 Penelitian Terkait

No	Nama Jurnal	Penulis	Tahun	Akurasi	Hasil
1.	**Jurnal Rekayasa Perangkat Lunak -** Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma SVM Dan NBC[8]	A. Muhammadin, I. A. Sobari	2024	SVM: 83,3%	SVM lebih unggul dari NBC. Penelitian ini juga menyebutkan kelemahan terkait data yang tidak seimbang.
2.	**Journal of Mandalika Literature** - Analisis Sentimen Pinjaman Online Akulaku dan Kredivo dengan Metode Support Vector Machine (SVM)[5]	K. A. Dasuki, S. S. Hilabi, dkk.	2023	SVM: 88,20%	Dalam perbandingan dengan Akulaku, Kredivo menunjukkan akurasi SVM yang lebih tinggi.

No	Nama Jurnal	Penulis	Tahun	Akurasi	Hasil
3.	**Jurnal Inovasi Global** - Analisis Sentimen Pengguna Aplikasi Kredivo Menggunakan Algoritma K-Nearest Neighbor[9]	Saepudin, Sutisna	2024	KNN: 79,36%	Algoritma KNN dinilai dapat mengklasifikasikan sentimen pada ulasan Kredivo dengan cukup baik.
4.	**MALCOM: Indonesian Journal of Machine Learning and Computer Science** - Implementasi Algoritma SVM untuk Analisa Sentimen Data Ulasan Aplikasi Pinjaman Online di *Google Play Store* [10]	M. Iqbal, M. Afdal, R. Novita	2024	SVM: 62%	Kredivo memiliki sentimen positif terbanyak (46%) dibandingkan 4 aplikasi lain, namun akurasi SVM-nya tercatat 62% pada studi ini.
5.	**HOAQ: Jurnal Teknologi Informasi** – *Analisis Sentimen Aplikasi Halo BCA di Google Play Store Menggunakan Metode Naïve Bayes, Support Vector Machine, dan Random Forest [6]*	Mola SAS, Baun DLB, Nunes IO, Sani MMA	2024	Random Forest: 91,28% SVM: 87,55% Naïve Bayes: 81,73%	Random Forest unggul dalam klasifikasi sentimen, sementara ulasan negatif (37,5%) menyoroti masalah teknis dan kendala akses.

No	Nama Jurnal	Penulis	Tahun	Akurasi	Hasil
6.	**Digital Business Intelligence Journal** – *Analisis Sentimen Ulasan Aplikasi Shopee Menggunakan Algoritma Random Forest, Naïve Bayes, dan Support Vector Machine di Kota Semarang [11]*	Adiguna VB, Pramudya RA	2024	Random Forest: 96,19% SVM: 95,71% Naïve Bayes: 84,76%	Mayoritas ulasan Shopee bersifat netral (69,7%), sementara sentimen negatif (11,8%) dan positif (18,9%).
7.	**Bangkit Indonesia** – *Komparasi Algoritma Support Vector Machine Dengan Naïve Bayes Untuk Analisis Sentimen Pada Aplikasi BRImo [3]*	Astuti AP, Alam S, Jaelani I	2024	SVM: 97,69% Naïve Bayes: 96,53%	SVM lebih unggul dalam mengklasifikasikan ulasan pengguna aplikasi BRImo.
8.	**JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)** – *Analisis Sentimen Aplikasi BCA Mobile Menggunakan Algoritma Naïve Bayes Dan Support Vector Machine [4]*	Bhatara DW, Suryono RR	2024	SVM: 85% Naïve Bayes: 83%	SVM lebih akurat dalam mengklasifikasikan ulasan positif, sementara Naïve Bayes lebih baik untuk ulasan negatif.

No	Nama Jurnal	Penulis	Tahun	Akurasi	Hasil
9.	**Edumatic: Jurnal Pendidikan Informatika** – *Penerapan Algoritma Random Forest untuk Menganalisis Ulasan Aplikasi Spotify pada Google Play [12]*	Insany GP, Kharisma IL, Rosmawati	2024	Random Forest: 88,4% (Indonesia) 93,6% (Inggris)	Algoritma Random Forest efektif untuk analisis sentimen dalam konteks multibahasa dan dapat membantu pengembang aplikasi.
10.	**Jurnal Ilmiah Penelitian dan Pembelajaran Informatika** – *Analisis Sentimen Ulasan Aplikasi M-Paspor Menggunakan Naïve Bayes dan Support Vector Machine [13]*	Maheri R, Salisah FN, Muttakin F, Megawati	2024	SVM: 80,76% Naïve Bayes: 78,12%	SVM lebih akurat dalam mengklasifikasikan ulasan pengguna aplikasi M-Paspor.

Tinjauan terhadap penelitian terkait menunjukkan berbagai pendekatan dan hasil dalam analisis sentimen pada aplikasi layanan keuangan. Secara khusus pada aplikasi Kredivo, penerapan algoritma *Support Vector Machine* (SVM) berhasil mencapai akurasi sebesar 88,20% dalam satu studi [4], dan studi lain mengonfirmasi keunggulan SVM (83,3%) atas Naïve Bayes [3]. Penelitian berbeda yang menggunakan *K-Nearest Neighbor* (KNN) pada ulasan Kredivo mencatatkan akurasi sebesar 79,36% [5]. Ketika perbandingan diperluas ke aplikasi sejenis, terlihat pola yang menarik. Pada aplikasi Halo BCA dan Shopee, algoritma *ensemble Random Forest* menunjukkan performa tertinggi dengan akurasi masing-masing 91,28% dan 96,19% [7] [8]. Namun, pada aplikasi perbankan digital seperti BRImo dan aplikasi M-Paspor, SVM kembali terbukti

lebih unggul dibandingkan *Naïve Bayes* [9] [12]. Temuan lain pada aplikasi BCA Mobile bahkan merinci bahwa SVM lebih efektif untuk sentimen positif, sementara *Naïve Bayes* lebih baik untuk sentimen negatif [10]. Keunggulan *Random Forest* juga terlihat pada kasus ulasan multibahasa di aplikasi Spotify, yang menegaskan kapabilitasnya dalam berbagai skenario [11].

Meskipun memiliki kesamaan tema dengan penelitian sebelumnya, penelitian ini membedakan diri melalui beberapa aspek fundamental. Pertama, penelitian ini secara spesifik berfokus pada perbandingan tiga algoritma: *Support Vector Machine*, *Naïve Bayes*, dan *Random Forest* pada ulasan aplikasi Kredivo, dengan menggunakan dataset terbaru sebanyak 10.000 ulasan dari tahun 2025. Kedua, pendekatan yang digunakan terstruktur secara sistematis mengikuti kerangka kerja *CRISP-DM*. Namun, pembeda utama penelitian ini terletak pada fokusnya untuk secara langsung menjawab celah penelitian yang teridentifikasi, yaitu masalah ketidakseimbangan kelas data, dengan menguji efektivitas teknik oversampling *SMOTE*.

Dengan demikian, penelitian ini tidak hanya melanjutkan tren perbandingan algoritma yang ada, tetapi juga memberikan kontribusi baru yang aplikatif. Kontribusi tersebut diwujudkan melalui pengujian solusi terhadap masalah data tidak seimbang dan pengembangan sebuah *dashboard* interaktif. Pendekatan ini diharapkan mampu menghasilkan analisis yang lebih seimbang dan andal, serta menyajikan solusi yang dapat digunakan untuk memonitor opini publik pada layanan keuangan digital secara efektif

2.2 Teori Penelitian

2.2.1 Analisis Sentimen Pada Ulasan Aplikasi Pinjaman Tunai

Sentiment analysis atau *opinion mining* merupakan metode yang digunakan untuk memahami, mengekstrak, dan menganalisis opini dari data berbasis teks [14]. Sentimen ini umumnya diklasifikasikan ke dalam dua kategori utama, yakni positif dan negatif. Di era digital saat ini, teknik ini menjadi semakin relevan seiring dengan meningkatnya jumlah ulasan aplikasi di platform seperti *Google Play Store*. Ulasan tersebut dapat memberikan wawasan berharga bagi perusahaan dalam menilai tingkat

kepuasan pengguna terhadap aplikasi mereka. Analisis sentimen juga memungkinkan pengolahan data dalam jumlah besar yang tidak dapat dilakukan secara manual [15]. Dalam penelitian ini, teknik tersebut dimanfaatkan untuk mengungkap persepsi pengguna terhadap aplikasi pinjaman tunai berdasarkan ulasan yang mereka tinggalkan di Google Play Store.

2.2.2 Kredivo

Kredivo adalah salah satu platform layanan keuangan digital yang menyediakan fasilitas kredit tanpa kartu bagi penggunanya [16]. Aplikasi ini memungkinkan pengguna untuk melakukan pembelian dengan sistem cicilan serta mengakses pinjaman tunai secara instan. Kredivo merupakan salah satu penyedia layanan kredit digital yang cukup populer di Indonesia dan beroperasi di bawah pengawasan Otoritas Jasa Keuangan (OJK) [17]. Dengan proses pengajuan yang cepat dan persyaratan yang relatif mudah, Kredivo menarik minat berbagai kalangan, terutama mereka yang membutuhkan akses kredit tanpa harus memiliki kartu kredit dari bank konvensional[18]. Selain itu, Kredivo telah terintegrasi dengan sejumlah platform e-commerce dan merchant, sehingga memberikan kemudahan dan fleksibilitas bagi pengguna dalam bertransaksi.

Sebagai penyedia layanan kredit digital, Kredivo menerapkan sistem penilaian kredit berbasis teknologi untuk menentukan kelayakan pengguna dalam mengakses layanan mereka. Pengguna dapat mengajukan limit kredit yang dapat digunakan untuk berbelanja di berbagai mitra Kredivo atau dicairkan dalam bentuk pinjaman tunai[19]. Meskipun menawarkan berbagai kemudahan, Kredivo juga memiliki tantangan, seperti transparansi biaya layanan dan bunga yang masih menjadi perhatian pengguna. Dengan semakin berkembangnya layanan keuangan digital, Kredivo terus berinovasi dalam memberikan pengalaman kredit yang lebih mudah dan aman bagi penggunanya.

2.2.3 Google Play Store

Google Play Store merupakan platform distribusi digital yang dikembangkan dan dikelola oleh Google. Platform ini menyediakan berbagai aplikasi, permainan, buku elektronik, film, dan konten digital lainnya yang dapat diunduh oleh pengguna perangkat berbasis Android[20]. Google Play Store berfungsi sebagai sarana utama bagi pengembang untuk mempublikasikan aplikasi mereka serta menjangkau pengguna di seluruh dunia. Dengan jutaan aplikasi yang tersedia, Google Play Store telah menjadi ekosistem yang mendukung inovasi di berbagai sektor, termasuk keuangan, pendidikan, hiburan, dan kesehatan.

Selain menjadi sarana distribusi aplikasi, Google Play Store turut menyediakan fitur rating dan review, yang memungkinkan pengguna memberikan umpan balik atas pengalaman mereka dalam menggunakan suatu aplikasi. Sistem ini membantu pengguna lain dalam menilai kualitas suatu aplikasi sebelum mengunduhnya, serta memberikan informasi berharga bagi pengembang untuk meningkatkan layanan mereka. Dengan meningkatnya jumlah pengguna smartphone di seluruh dunia, Google Play Store terus berkembang menjadi platform yang lebih canggih dan aman dalam mendukung distribusi aplikasi digital secara global.

2.3 Framework dan Algoritma Penelitian

2.3.1 CRISP-DM

Kerangka kerja CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah sebuah metodologi standar yang populer dan banyak digunakan untuk memandu pelaksanaan proyek *data mining* atau analisis data. Metodologi ini menyediakan pendekatan yang terstruktur dan sistematis, memastikan bahwa setiap tahapan penting dalam proyek tercakup secara menyeluruh dari awal hingga akhir [21]. Salah satu karakteristik utama dari CRISP-DM adalah sifatnya yang siklis dan iteratif, yang berarti peneliti dapat kembali ke tahap-tahap sebelumnya jika diperlukan penyesuaian atau ditemukan pemahaman baru, sehingga prosesnya fleksibel dan tidak harus berjalan secara linear[22].

CRISP-DM terdiri dari enam tahapan utama yang saling berhubungan, yaitu:

1. ***Business Understanding*** **(Pemahaman Bisnis):** Tahap awal yang berfokus pada pemahaman tujuan dan kebutuhan proyek dari sudut pandang bisnis. Hasilnya adalah perumusan masalah bisnis dan tujuan analisis data yang jelas.
2. ***Data Understanding*** **(Pemahaman Data):** Melibatkan pengumpulan data awal dan eksplorasi untuk memahami isi, kualitas, dan karakteristik data. Pada tahap ini, wawasan pertama atau hipotesis awal dapat terbentuk.
3. ***Data Preparation*** **(Persiapan Data):** Mencakup semua aktivitas untuk membangun dataset akhir yang akan digunakan untuk pemodelan dari data mentah. Tahapan ini seringkali menjadi yang paling memakan waktu, meliputi proses pembersihan data, integrasi, dan transformasi fitur.
4. ***Modeling*** **(Pemodelan):** Pada tahap ini, berbagai teknik pemodelan dipilih dan diterapkan. Prosesnya sering kali bersifat iteratif dengan tahap persiapan data karena beberapa model memiliki kebutuhan format data yang spesifik.
5. ***Evaluation*** **(Evaluasi):** Sebelum model diterapkan, dilakukan evaluasi menyeluruh untuk mengukur performa dan kualitas model. Tujuannya adalah untuk memastikan bahwa model yang dibangun benar-benar mencapai tujuan bisnis yang telah ditetapkan di awal.
6. ***Deployment*** **(Penerapan):** Tahap akhir di mana hasil analisis dan model yang telah divalidasi diimplementasikan ke dalam lingkungan operasional. Bentuknya bisa bervariasi, mulai dari laporan sederhana hingga integrasi model ke dalam sebuah aplikasi.

Penerapan kerangka kerja ini dalam sebuah penelitian memberikan alur kerja yang jelas dan terarah. Dengan mengikuti tahapan CRISP-DM, peneliti dapat memastikan bahwa solusi yang dikembangkan benar-benar menjawab permasalahan yang ada dan telah melalui proses validasi dan evaluasi yang ketat sebelum diimplementasikan.

2.3.2 *Text mining*

Text mining merupakan proses menggali informasi dari data teks yang bersifat tidak terstruktur, dengan tujuan untuk mengidentifikasi pola, hubungan, atau wawasan baru yang berguna [23]. Teknik ini umum digunakan dalam analisis kumpulan teks yang besar untuk menemukan informasi signifikan yang tidak langsung terlihat. Dalam era digital, jumlah data teks yang terus meningkat membuat *Text mining* menjadi metode yang penting dalam berbagai bidang, termasuk bisnis, penelitian, dan pengolahan dokumen.

Proses *Text mining* terdiri dari beberapa tahapan utama yang harus dilakukan secara sistematis [24]. Tahapan pertama adalah preprocessing data teks, yang mencakup pembersihan teks dari karakter yang tidak diperlukan, pemecahan teks menjadi kata-kata individu, serta penyederhanaan kata ke bentuk dasar agar lebih mudah dianalisis. Setelah itu, dilakukan ekstraksi fitur, yaitu Proses transformasi teks menjadi representasi berbasis angka menggunakan berbagai teknik seperti frekuensi kata atau pemberian bobot pada istilah yang lebih relevan[24] .

Setelah data teks siap digunakan, tahapan berikutnya adalah analisis teks, yang bertujuan untuk menemukan informasi tersembunyi di dalamnya. Beberapa metode analisis yang umum digunakan meliputi klasifikasi teks berdasarkan kategori tertentu, pengelompokan teks yang memiliki kesamaan tema, serta identifikasi pola kata yang sering muncul dalam kumpulan teks yang besar. Hasil dari analisis ini dapat membantu dalam pengambilan keputusan dan pemahaman yang lebih mendalam terhadap data yang dianalisis.

Text mining memiliki banyak manfaat dalam berbagai sektor, terutama dalam pengolahan informasi dan pengambilan keputusan berbasis data [25]. Perusahaan dapat memanfaatkannya untuk memahami opini pelanggan, sementara akademisi dan peneliti menggunakannya untuk menemukan pola dalam literatur ilmiah atau data historis. Dengan perkembangan teknologi, metode ini terus mengalami peningkatan dan semakin banyak digunakan untuk mengolah data teks dalam berbagai skala dan kompleksitas.

2.3.3 Data Pre-Processing

2.3.3.1 *Case folding* and *Data cleaning*

Case folding dan *data cleaning* merupakan tahap awal yang fundamental dalam proses normalisasi teks[26]. Tahap ini mencakup dua proses utama: *case folding*, yaitu mengubah semua huruf kapital menjadi huruf kecil (*lowercase*) untuk menyeragamkan teks, dan *data cleaning*, yaitu membersihkan data dari elemen-elemen yang tidak relevan seperti angka, tanda baca, dan karakter khusus [6]. Proses ini sangat penting, terutama saat menangani data mentah seperti ulasan pengguna, yang sering kali mengandung variasi penulisan tidak baku, termasuk emoji atau kapitalisasi berlebihan yang dapat mengganggu proses analisis sentimen. Tujuan akhir dari kedua proses ini adalah untuk menstandardisasi dan menyederhanakan data teks, sehingga menghasilkan data yang bersih dan konsisten untuk tahap pemrosesan selanjutnya.

2.3.3.2 *Tokenization*

Tokenisasi merupakan sebuah proses fundamental dalam pengolahan teks di mana kalimat atau paragraf dipecah menjadi unit-unit yang lebih kecil yang dikenal sebagai token, yang umumnya berupa kata per kata[27]. Tujuan utama dari tokenisasi adalah untuk mempersiapkan data teks agar setiap kata dalam ulasan dapat dianalisis dan diproses secara terpisah oleh algoritma klasifikasi pada tahap selanjutnya. Dengan mengubah serangkaian kalimat

menjadi daftar kata-kata individual, proses ini menjadi landasan penting yang memungkinkan penerapan tahap-tahap berikutnya seperti normalisasi dan *stemming*. Implementasi proses tokenisasi ini dilakukan dengan memanfaatkan *library* Natural Language Toolkit (NLTK).

2.3.3.3 *Normalization*

Normalisasi merupakan sebuah proses yang bertujuan untuk menyeragamkan bentuk kata-kata tidak baku atau informal menjadi bentuknya yang baku dan formal[28]. Proses ini sangat relevan dalam analisis ulasan pengguna, karena sering kali ditemukan penggunaan kata slang atau singkatan seperti “gk”, “tdk”, dan “tp” yang merujuk pada kata “tidak” atau “tapi”. Apabila kata-kata tidak baku semacam ini tidak dinormalisasi terlebih dahulu, hal tersebut dapat mengurangi efektivitas algoritma dalam mengenali pola sentimen secara akurat. Untuk mengatasi masalah ini, normalisasi dilakukan dengan pendekatan berbasis kamus (*dictionary-based*), di mana setiap kata dicocokkan dengan daftar kata tidak baku yang telah dibuat untuk kemudian diganti dengan bentuk standarnya yang sesuai.

2.3.3.4 *Stopword removal*

Stopword removal merupakan proses penghilangan kata-kata umum yang memiliki frekuensi kemunculan tinggi namun dianggap kurang memiliki makna penting dalam menentukan sentimen sebuah kalimat, contohnya seperti kata "yang", "dan", "di", atau "ke"[29]. Tujuan utama dari penghapusan *stopword* ini adalah untuk mengurangi *noise* dan meningkatkan fokus model pada kata-kata kunci yang lebih signifikan dan relevan dalam tugas klasifikasi sentimen. Proses ini dilakukan dengan menyaring teks dan menghapus setiap kata yang termasuk dalam daftar *stopword* bahasa Indonesia, yang dalam penelitian ini disediakan oleh pustaka Natural Language Toolkit (NLTK). Tahap ini dijalankan setelah proses

normalisasi untuk memastikan bahwa kata-kata yang telah diubah ke bentuk baku juga dapat teridentifikasi dan dihapus jika termasuk dalam daftar *stopword*.

2.3.3.5 *Stemming*

Stemming merupakan sebuah proses untuk mengubah kata-kata yang memiliki imbuhan menjadi bentuk kata dasarnya (*root word*)[30]. Proses ini sangat penting dalam analisis sentimen karena bertujuan untuk menyeragamkan berbagai bentuk kata agar dianggap sebagai entitas yang sama oleh algoritma, contohnya seperti kata "mengirim", "dikirim", dan "pengiriman" yang semuanya akan diubah ke bentuk dasar "kirim". Dengan menyeragamkan kata-kata ini, *stemming* secara efektif membantu mengurangi dimensi fitur dari data teks sekaligus meningkatkan konsistensi representasi data secara keseluruhan. Untuk menangani berbagai jenis imbuhan dalam bahasa Indonesia secara akurat, proses *stemming* ini diimplementasikan dengan menggunakan pustaka Sastrawi.

2.3.3 Support Vector Machine (SVM)

SVM (Support Vector Machine) adalah algoritma klasifikasi yang banyak digunakan karena keefektifannya*, terutama dalam aplikasi* seperti analisis teks dan sentimen.[31]. *SVM* beroperasi dengan cara memisahkan data ke dalam kelas-kelas berbeda (seperti positif, negatif, atau netral) menggunakan sebuah hyperplane yang bertujuan untuk memaksimalkan jarak margin antar kelas. Algoritma ini terbukti efektif dalam mengelola data berukuran besar, yang tidak seimbang, serta bersifat kompleks, kondisi yang umum dijumpai dalam analisis sentimen.

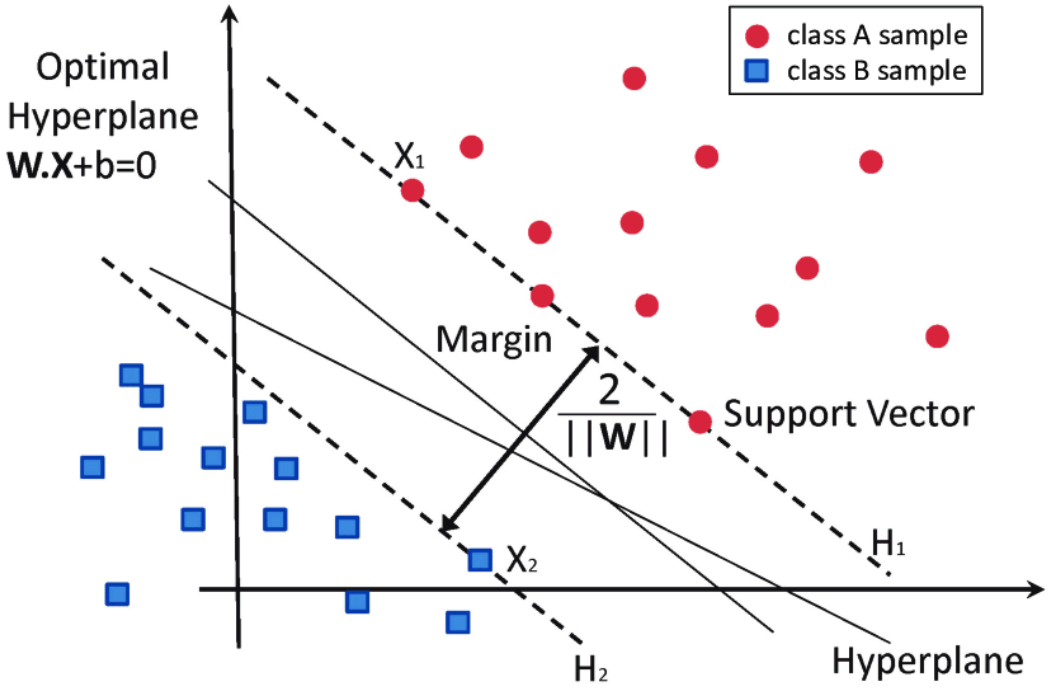
Keunggulan utama *SVM* adalah kemampuannya untuk menghasilkan klasifikasi yang akurat meskipun data memiliki banyak dimensi (fitur), seperti dalam kasus data teks yang melibatkan berbagai kata. *SVM* juga mampu mengatasi masalah *overfitting* melalui

penggunaan *hyperplane* dan margin maksimal. Meskipun *SVM* memerlukan lebih banyak sumber daya komputasi dibandingkan beberapa algoritma lain, seperti *Naive Bayes*, *SVM* cenderung memberikan hasil yang lebih baik dalam klasifikasi sentimen karena lebih baik dalam menangani data yang lebih kompleks dan tidak seimbang.

Dalam analisis sentimen pada Kredivo, SVM digunakan untuk mengkategorikan ulasan produk menjadi sentimen positif dan negatif dengan cara memisahkan ulasan berdasarkan pola teks yang terkandung di dalamnya. Algoritma ini mencari hyperplane optimal yang mampu membedakan data ke dalam kelas-kelas tersebut. SVM kemudian menentukan margin terbesar yang memisahkan sentimen positif dan negatif agar klasifikasi yang dihasilkan lebih tepat.

Prinsip kerja dasar *SVM* adalah sebagai berikut:

1. *Hyperplane*: Garis atau batas yang memisahkan data menjadi dua kelas berbeda. SVM berupaya menemukan hyperplane yang dapat memperbesar jarak (margin) antara kedua kelas tersebut.
2. *Support Vectors*: Titik data yang berada paling dekat dengan hyperplane, yang berperan penting dalam menentukan posisi dan orientasi *hyperplane*.
3. *Margin*: Jarak antara *hyperplane* dengan *support vectors* dari masing-masing kelas. *SVM* berusaha memaksimalkan margin ini guna meningkatkan akurasi dalam melakukan prediksi.



Gambar 2. 1 Hyperplane

Sumber : [7]

Gambar 2.1 menunjukkan konsep hyperplane pada metode Support Vector Machine, yaitu sebuah garis yang memisahkan dua kelompok data (Class A dan Class B) dalam ruang fitur. Garis pemisah ini secara matematis dirumuskan dalam Persamaan (2.1) berikut :

$$W \cdot X + b = 0$$

Rumus 2. 1 Hyperplane (2.1)

Keterangan:

1. W menentukan arah atau kemiringan dari garis pemisah.
2. b adalah jarak dari garis pemisah ke titik asal (pusat koordinat).

Tujuan *SVM* adalah menemukan posisi dan arah *hyperplane* ini sehingga dua kelas data terpisah dengan baik.

Rumus dasar untuk SVM dirumuskan dalam persamaan (2.2) berikut:

$$min\ w,b\ \left(\frac{1}{2}\|w\|^2\right) subject\ to\ y_i(w\ .\ x_i - b)\ \geq 1\ for\ all\ i$$

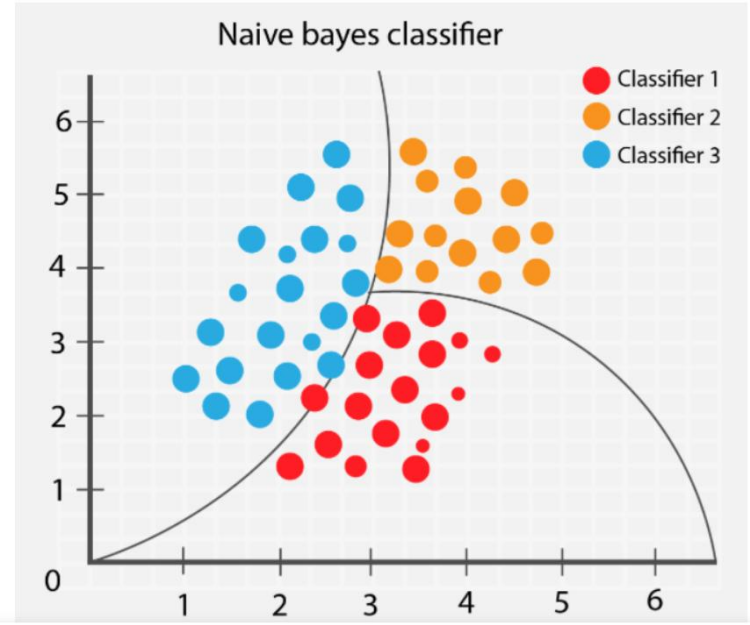
Rumus 2. 2 SVM (2.2)

Keterangan:

1. w adalah vektor bobot yang menentukan arah *hyperplane*.
2. b adalah *bias* atau *offset* dari *hyperplane*.
3. x_i adalah titik data.
4. y_i adalah label kelas (positif dan negatif).

SVM sangat cocok untuk analisis sentimen karena algoritma ini mampu menangani dataset yang besar dan tidak seimbang dengan baik, memberikan hasil klasifikasi yang sangat akurat meskipun data bersifat kompleks. Penggunaan *SVM* dalam analisis sentimen memungkinkan perusahaan untuk lebih memahami sentimen konsumen terhadap produk mereka secara lebih efektif, seperti yang diharapkan dalam penelitian ini.

2.3.3 Naïve Bayes



Gambar 2. 2 *Naive Bayes Classifier*

Sumber: [32]

Naïve Bayes Classifier pada gambar 2.2 adalah metode berbasis probabilitas yang mampu bekerja dengan data dalam jumlah besar, serta menghitung probabilitas dari kombinasi nilai dan dataset secara acak. Metode ini melibatkan dua tahap utama, yaitu pelatihan (training) dan pengujian (testing). Pada tahap pelatihan, data yang telah dilengkapi dengan label kategori digunakan untuk membangun dan mengasah model. Selanjutnya, pada tahap testing, model akan mengklasifikasikan data baru yang belum memiliki kategori. Rumus dasar Naïve bayes dirumuskan dalam persamaan (2.3) berikut:

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

Rumus 2. 3 Naïve Bayes (2.3)

2.3.4 Random Forest

Random Forest merupakan algoritma machine learning yang tergolong dalam ensemble learning, yaitu teknik yang menggabungkan beberapa model dasar (dalam hal ini decision tree) untuk menciptakan model prediksi yang lebih handal dan akurat. Algoritma ini dikembangkan oleh Leo Breiman pada tahun 2001 sebagai penyempurnaan dari metode bagging (bootstrap aggregating) dan pohon keputusan [33].

Random Forest beroperasi dengan membangun sejumlah pohon keputusan secara acak, lalu menggabungkan output dari setiap pohon untuk menentukan hasil akhir. Pada tugas klasifikasi, keputusan akhir ditentukan melalui voting mayoritas, yaitu kelas yang paling sering dipilih oleh pohon-pohon dalam hutan. Sementara itu, untuk tugas regresi, hasil akhir diperoleh dengan menghitung rata-rata dari seluruh prediksi pohon [34].

Dalam proses pelatihannya, Random Forest menggunakan dua sumber acak yaitu:

1. **Bootstrap Sampling** – setiap pohon dilatih menggunakan subset data latih yang diambil secara acak dengan pengembalian (sampling with replacement).
2. **Random Feature Selection** – pada setiap node pohon, hanya sebagian kecil fitur yang dipilih secara acak untuk dipertimbangkan dalam proses pemisahan (splitting), bukan seluruh fitur yang tersedia. Teknik ini membuat setiap pohon memiliki perbedaan yang signifikan, sehingga ketika digabungkan akan menghasilkan generalisasi model yang lebih baik.

2.3.5 Term Frequency Inverse Document Frequency (*TF-IDF*)

Dua teknik umum yang sering digunakan untuk mengubah teks menjadi format yang dapat dipahami komputer adalah TF-IDF (Term Frequency-Inverse Document Frequency). Metode TF-IDF memberikan bobot pada setiap kata dengan mempertimbangkan frekuensi kemunculannya dalam sebuah dokumen serta kelangkaannya di seluruh kumpulan dokumen [15]. Dengan kata lain, *TF-IDF* mengidentifikasi kata-kata yang mungkin penting dalam dokumen tertentu tetapi tidak umum dalam seluruh korpus. *TF-IDF* sangat berguna untuk mengurangi pengaruh kata-kata umum yang mungkin muncul di seluruh ulasan namun tidak terlalu signifikan dalam memberikan makna sentimen.

Support Vector Machine (SVM) memanfaatkan representasi teks yang diperoleh melalui *TF-IDF* untuk mengklasifikasikan teks ke dalam kategori sentimen, seperti positif dan negatif. Dengan menggunakan *TF-IDF*, *SVM* mampu meningkatkan akurasi klasifikasi, karena *TF-IDF* membantu menyoroti kata-kata yang lebih relevan untuk pengklasifikasian. Rumus *TF-IDF* dirumuskan dalam persamaan (2.4) berikut:

$$TF - IDF(t,d) = TF(t,d) \times log\left(\frac{N}{DF(t)}\right)$$

Rumus 2. 4 TF-IDF (2.4)

1. **TF(t, d)** mengacu pada banyaknya kemunculan kata t dalam dokumen d.
2. ***N*** merupakan total jumlah dokumen yang ada.
3. **DF(t)** adalah jumlah dokumen yang memuat kata t.
4. ***TF-IDF*** memberikan bobot lebih tinggi pada kata-kata yang sering muncul dalam satu dokumen namun jarang muncul di dokumen lain, sehingga membantu menyoroti kata-kata yang memiliki makna penting dalam konteks tertentu.

Dengan metode *TF-IDF*, *SVM* dapat lebih akurat dalam memproses ulasan dan melakukan klasifikasi sentimen berdasarkan relevansi kata-kata yang muncul dalam ulasan tersebut.

2.3.6 Conffusion Matrix

Confusion matrix merupakan alat evaluasi yang sangat penting untuk menilai seberapa efektif sebuah model klasifikasi dalam memprediksi hasil yang tepat. [35]. Dalam konteks analisis sentiment dengen *Support Vector Machine (SVM)*, beberapa metrik yang digunakan untuk mengevaluasi performa model dirumuskan dalam persamaan (2.5),(2.6) berikut:

1. *Precision*: Mengukur tingkat ketepatan model dalam memprediksi kelas positif. *Precision* dihitung dengan rumus:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Rumus 2. 5 *Precision* (2.5)

2. *Recall*: Mengukur kemampuan model dalam menemukan semua kasus positif yang sebenarnya. Rumus *recall* adalah:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Rumus 2. 6 *Recall* (2.6)

3. *F1-Score*: Merupakan nilai gabungan dari *precision* dan *recall* yang memberikan gambaran seimbang antara keduanya. *F1-Score* sangat berguna terutama pada data yang tidak seimbang, seperti ketika jumlah ulasan positif jauh lebih banyak dibandingkan ulasan negatif. Rumus *F1-Score* dirumuskan dalam persamaan (2.7) berikut:

$$F1 = 2 . \frac{Precision\ .\ Recall}{Precision + Recall}$$

Rumus 2. 7 *F1-Score (2.7)*

F1-Score membantu menilai keseimbangan antara kesalahan false positives dan false negatives, yang sangat penting dalam analisis sentimen di mana kedua jenis kesalahan tersebut dapat berdampak signifikan.

2.3.6 SMOTE

SMOTE (*Synthetic Minority Over-sampling Technique*) adalah sebuah metode *oversampling* yang dirancang secara khusus untuk menangani masalah data tidak seimbang (*imbalanced data*) dalam tugas klasifikasi [36]. Berbeda dengan teknik *oversampling* sederhana yang hanya menduplikasi sampel dari kelas minoritas, SMOTE bekerja dengan cara membuat sampel "sintetis" atau sampel buatan yang baru dan informatif. Prosesnya dimulai dengan memilih sebuah sampel acak dari kelas minoritas, kemudian mengidentifikasi beberapa tetangga terdekatnya (*k-nearest neighbors*). Setelah itu, sebuah sampel sintetis baru akan dibuat di suatu titik pada garis imajiner yang menghubungkan sampel asli dengan tetangga-tetangga yang telah dipilih tersebut.

Penerapan teknik seperti SMOTE menjadi sangat penting ketika dihadapkan pada dataset dengan distribusi kelas yang tidak seimbang. Kondisi ini terjadi ketika jumlah sampel pada satu kelas (kelas mayoritas) jauh lebih banyak daripada jumlah sampel pada kelas lainnya (kelas minoritas), seperti yang teridentifikasi dalam penelitian ini di

mana sentimen positif mencakup 88,5% dari data, sementara sentimen negatif hanya 11,5%. Ketidakseimbangan yang signifikan ini dapat menyebabkan model *machine learning* menjadi bias, di mana model cenderung sangat baik dalam memprediksi kelas mayoritas namun performanya sangat buruk dalam mengenali kelas minoritas.

Tujuan utama dari penerapan SMOTE adalah untuk menghasilkan dataset latih yang seimbang, di mana jumlah data pada kelas minoritas telah ditingkatkan hingga setara dengan jumlah data pada kelas mayoritas[36]. Dengan melatih model menggunakan dataset yang telah diseimbangkan ini, diharapkan model yang dihasilkan akan menjadi lebih *robust*, tidak bias, dan memiliki kemampuan yang jauh lebih baik dalam mengidentifikasi kelas minoritas. Hal ini pada akhirnya akan meningkatkan metrik evaluasi yang penting seperti *recall* dan *F1-Score*, terutama untuk kelas yang sebelumnya kurang terwakili dalam data.

2.4 Tools Penelitian

2.4.1 Python

Python adalah bahasa pemrograman yang terkenal karena kemudahannya digunakan, sifatnya yang fleksibel, serta dukungan yang luas untuk berbagai jenis aplikasi, mulai dari pengembangan web hingga analisis data dan kecerdasan buatan [37]. Python dirancang dengan filosofi yang menekankan pada keterbacaan kode, yang membuatnya mudah dipelajari dan digunakan baik oleh pemula maupun profesional. Salah satu keunggulan utama Python adalah sintaksnya yang sederhana dan mirip dengan bahasa Inggris, sehingga mempercepat proses pengembangan. Misalnya, untuk mencetak teks, Python hanya membutuhkan satu baris kode seperti print("Hello, world!"), yang akan langsung menampilkan output di layar.

Selain kemudahan penggunaannya, Python juga dilengkapi dengan ekosistem *library* dan *framework* yang sangat kaya, memungkinkan *developer* untuk menangani berbagai jenis aplikasi.

Dalam bidang data science dan analisis data, library seperti *Pandas, NumPy, dan SciPy* banyak digunakan untuk manipulasi data, perhitungan numerik, dan analisis statistik. *Python* juga unggul dalam kecerdasan buatan dan machine learning berkat library seperti *Scikit-learn*, dan *TensorFlow*,. Dengan ekosistem yang luas ini, *Python* memungkinkan pengembang untuk fokus pada pemecahan masalah inti tanpa harus membangun segalanya dari awal.

Python banyak digunakan di berbagai sektor karena keandalannya dalam otomasi, pengembangan perangkat lunak, dan web development melalui framework seperti *Django* dan *Flask*. Sebagai salah satu bahasa pemrograman paling populer di dunia, Python telah digunakan oleh perusahaan teknologi besar seperti Google, Facebook, dan Netflix, serta banyak komunitas akademik dan riset .

2.4.2 Jupyter Notebook

Jupyter Notebook adalah platform pengembangan interaktif berbasis web yang memungkinkan pengguna menulis kode, menjalankan program, melihat hasilnya, serta menyisipkan dokumentasi dalam satu antarmuka [37]. Jupyter Notebook mendukung berbagai bahasa pemrograman, tetapi yang paling umum digunakan adalah Python. Alat ini sangat cocok untuk analisis data, pembelajaran mesin, dan visualisasi, karena memungkinkan pemrogram untuk menjalankan kode selangkah demi selangkah dan melihat hasilnya secara langsung. Selain itu, Jupyter Notebook memungkinkan integrasi teks dengan kode, sehingga memudahkan pengguna untuk mencatat langkah-langkah analisis dan hasil eksperimen di satu dokumen yang dapat dibagikan atau diterbitkan.