BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Analisis sentimen atau *opinion mining* adalah studi komputasi pendapat, sentimen dan ekspresi emosi yang diungkapkan dalam teks. Proses yang dilakukan dalam analisis sentimen ditujukan untuk memahami, mengekstrak dan mengolah data teks menjadi suatu informasi yang bermanfaat. Sentimen analisis secara umum digunakan untuk mengenali opini yang diekspresikan pada sebuah teks (Liu, 2012).

Sentimen analisis tingkat aspek telah diterapkan dalam penelitian Ndanal dkk (2019) berbasis pembelajaran mesin pada produk Amazon. Analisis tingkat aspek dari data ulasan sangat membantu penjual dalam memahami harapan pelanggan dan membentuk kebijakan yang lebih baik. Algoritma *Support Vector Machines* digunakan untuk mengklasifikasi hasil kategorisasi pola analisis sentimen. Sistem melakukan operasi pra-pemrosesan seperti *stemming*, tokenisasi, *stop-word removal* pada dataset untuk mengekstrak informasi yang berarti dan memberikan peringkat untuk diklasifikasikan dalam hal negatif atau positif (Ndanal dkk, 2019).

Haque dkk (2018) menerapkan sentimen analisis untuk mempolarisasikan *review* produk *handphone* dan aksesorisnya serta *review* produk alat musik yang terdiri dari kurang lebih 48.500 *review* produk. Algoritma *Decision Tree, Random Forest, Stochastic Gradient Descent*, dan *Support Vector Machine* diterapkan dalam proses analisis sentimen. Simulasi yang berbeda dapat menggunakan validasi silang, rasio pengujian pelatihan, dan proses ekstraksi fitur yang berbeda untuk membandingkan jumlah data yang bervariasi sehingga mencapai hasil yang menjanjikan. Hasil dari masing-masing metode menunjukkan bahwa algoritma *Support Vector Machine* menghasilkan akurasi tertinggi (Haque dkk, 2018).

Hadwan dkk (2022) menganalisis pendapat pengguna dari enam aplikasi di sektor kesehatan. Data dikumpulkan dari ulasan 6 aplikasi seluler yang tersedia di *Google Play* dan *App Store*, yang mencakup 51 ribu ulasan. Lima model ML diterapkan untuk mengklasifikasikan opini sentiment, yaitu *Random Forest (RF), Bagging, Support Vector Machine (SVM), Logistic Regression (LR),* dan *Naïve*

Bayes (NB). Teknik pendekatan yang digunakan adalah *Bing Liu lexicon, AFINN, MPQA Subjectivity Lexicon, a bag of words (BoW), term frequency-inverse document frequency (TF-IDF)* dan *the Google pre-trained Word2Vec were integrated*. Percobaan pertama peforma ML diuji menggunakan dataset yang tidak seimbang. Percobaan lebih lanjut dilakukan dengan menggunakan dataset seimbang. Teknik penyeimbangan, teknik *SMOTE* diterapkan sebagai teknik penyeimbangan pada dataset memperoleh peningkatan yang lebih baik. Hasil eksperimen menunjukkan bahwa skor akurasi tertinggi (94,38%) diperoleh dengan menerapkan *Support Vector Machine* (SVM) menggunakan teknik *SMOTE* dengan semua fitur pendekatan bersambung (Hadwan dkk, 2022).

Penelitian yang dilakukan Obiedat dkk (2022) mengusulkan teknik evolusi *hybrid* untuk menganalisis sentimen masyarakat terhadap berbagai restoran di seluruh Yordania. Data dikumpulkan dari jejaring sosial populer, yaitu Jeeran dengan mengumpulkan lebih dari 3000 ulasan restoran dan memberi label menggunakan teknik *crowd sourcing*. Teknik *oversampling* kemudian diterapkan untuk mengatasi masalah ketidakseimbangan data dalam dataset. Pendekatan *hybrid* dengan menggabungkan algoritma *Support Vector Machine* (SVM) dengan *Particle Swarm Optimization* (PSO) dan teknik *oversampling* yang berbeda untuk menangani masalah data yang tidak seimbang. Studi ini menunjukkan bahwa pendekatan PSO-SVM yang diusulkan menghasilkan hasil terbaik dibandingkan dengan teknik klasifikasi yang berbeda dalam hal akurasi, *F-measure*, *G-mean* dan *Area Under the Curve* (AUC) untuk versi dataset yang berbeda (Obiedat dkk, 2022).

Zhang dkk (2022) merancang model prediksi status kelangsungan hidup untuk pasien osteosarkoma berdasarkan *E-CNN-SVM* dan *multisource data fusion* dengan pengurangan dimensi dan kemudian menyamakan data menggunakan metode *hybrid sampling* yang menggabungkan algoritma *SMOTE* dan algoritma *Tomek Links*. Model CNN dengan modul insentif digunakan untuk mengekstrak fitur lebih lanjut dari data untuk ekstraksi informasi karakteristik yang lebih akurat, data selanjutnya diteruskan ke model SVM untuk lebih meningkatkan stabilitas dan

kinerja klasifikasi model. Model ini telah terbukti lebih efektif dalam meningkatkan akurasi klasifikasi pasien *osteosarcoma* (Zhang dkk, 2022).

Penelitian Pratama dkk (2022) membahas sentimen dari *review* yang ditentukan berdasarkan lima aspek hotel, yaitu makanan, pelayanan, lokasi, kenyamanan dan kebersihan. Setiap *term document* diekspansi menggunakan sinonim untuk meningkatkan nilai *similarity* ke LDA dengan 100% *extended document* menggunakan *Cosine Similarity* menghasilkan nilai performansi tertinggi sebesar 0,856. Pelabelan sentimen setiap *review* berdasarkan aspek dan klasifikasi sentimen menggunakan metode *Support Vector Machine* mendapatkan nilai rata-rata 0,940 (Pratama dkk, 2022).

Penelitian Akdane dkk (2022) yang berjudul "*Application of Support Vector Machine dan Convolutional Neural Network for Sentence-Level Sentiment Analysis of Companies Products Review*" menentukan polaritas sentimen dan klasifikasi ulasan produk untuk memastikan tingkat kepuasan atau ketidakpuasan. Algoritma CNN-SVM digunakan pada analisis sentimen tingkat kalimat pada ulasan produk musik di Amazon dengan 44.463 sampel pelatihan dan 19.056 sampel digunakan untuk pengujian dan validasi model. Berdasarkan metrik kinerja *model hybrid*, CNN memiliki akurasi 85,38%, *presisi* 90,56%, *recall* 95,14%, dan AUC 0,836, sedangkan SVM memiliki akurasi 85,74%, presisi 85,62 % dan AUC 0,5 (Akdane dkk, 2022).

Das dkk (2022) dengan penelitian yang berjudul *"Application of Support Vector Machine dan Convolutional Neural Network for Sentence-Level Sentiment Analysis of Companies Products Review*" melakukan klasifikasi terhadap 6000 komentar dan pandangan tentang produk. Sentimen analisis menggunakan algoritma *KNN*, *Decision Tree*, *Support Vector Machine* (SVM), *Random Forest*, dan *Logistic Regression*. Algoritma SVM merupakan algoritma terbaik dengan akurasi 94,78% (Das dkk, 2022).

Daftar publikasi hasil penelitian terkait sentimen analisis dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Daftar Penelitian terkait

No	Penulis dan Tahun	Judul	Keterangan
1	Nhu dkk, 2022 *(International Journal of Recent Technology and Engineering 7 (6), 95–99.)*	*Churn prediction in telecommunication industry using kernel Support Vector Machines*	Riset terkait prediksi untuk mengidentifikasi pelanggan dengan potensi tinggi untuk mengakhiri kontrak atau meneruskan kontrak berlangganan pada penyedia layanan telekomunikasi. Dataset berasal dari Orange S.A Telcom Company yang dipublikasikan di *Kaggel* dengan jumlah catatan 3333 pelanggan di 51 negara bagian AS. Algoritma *Support Vector Machine* (SVM) digunakan untuk klasifikasi. *Synthetic Minority Oversampling Technique Tomek Link* (SMOTE *Tomek Links*) dan *Synthetic Minority Oversampling Technique ENN* (SMOTE ENN) digunakan untuk menangani masalah data tidak seimbang Algoritma SVM dengan kombinasi SMOTE dan *Tomek Links* menghasilkan skor F1 dan akurasi 98,88%, metode SVM dengan SMOTE-ENN menghasilkan skor F1 dan akurasi 99,01%
2	Hadwan dkk, 2022	*An Improved Sentiment Classification Approach for Measuring*	Penelitian yang melakukan klasifikasi sentimen kepuasan pelanggan terhadap aplikasi seluler menggunakan metode

	(Applied Sciences 12 (11), 5547.)	*User Satisfaction toward Governmental Services' Mobile Apps Using Machine Learning Methods with Feature Engineering and SMOTE Technique*	*machine learning* dan teknik SMOTE. Algoritma yang digunakan adalah *Random Forest* (RF), *Bagging*, *Support Vector Machine* (SVM), *Logistic Regression* (LR), dan *Naïve Bayes* (NB). Hasil skor akurasi dari masing-masing algoritma adalah RF: 93,87%, *Bagging*: 91,89%, SVM: 94,38%, LR: 92,68%, dan NB: 93,11%.
3	Bourequat dan Mourad, 2021 *(International Journal of Advances in Data and Information Systems 2 (1), 36–44)*	*Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine*	Riset terkait *text mining* pada *Twitter* mengenai rilis *iPhone* dengan metode klasifikasi sentimen menggunakan SVM. Sejumlah 2000 dataset berbahasa Inggris diproses melalui tahap *preprocessing.* Data tersebut dibagi menjadi data latih dan data uji dengan perbandingan 80:20 Hasil skor akurasi sebesar 89,21% dan matrik evaluasi *precision, recall, and F1* masing-masing menghasilkan skor 92,43%, 95,53%, dan 93,95.
4	Prasetio dkk, 2021 *(International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 202181–85)*	*Sentiment analysis on myindihome user reviews using Support Vector Machine and Naïve Bayes Classifier method*	Riset terkait sentimen analisis dengan masalah *service error* pada *myindihome* yang ditunjukkan dengan adanya ulasan negatif dari *review* pelanggan sebanyak 46,7% dari jumlah total 2.539 ulasan. Algoritma yang digunakan adalah *Support Vector Machine* (SVM) dan *Naïve Bayes Classifier* (NBC). Hasil dalam penelitian ini adalah komparasi skor akurasi

			dari algoritma *Support Vector Machine* (SVM) yang memperoleh skor akurasi 86,54% dan *Naïve Bayes Classifier* (NBC) memperoleh skor akurasi 84,69%.
5	Arivoli dan Sonali, 2021 *(prata)*	*Sentiment Analysis Using Support Vector Machine Based On Feature Selection Andsemantic Analysis*	Riset komparasi hasil sentimen analisis dengan menggunakan dua metode *Naïve Bayes* dan *Support Vector Machine classification*. Dataset yang digunakan ulasan dari film *Polarity* yang diambil dari API *Twitter*. Hasil evaluasi menggunakan *Confusion Matrix* dari masing-masing algoritma yang digunakan, hasil akurasi dikomprasikan dan diperoleh skor akurasi dari algoritma *Naïve Bayes* 76,67 dan skor akurasi algoritma *Support Vector Machine* 78,18.
6	Jonathan dkk, 2020 *(Proceedings - 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 202081–85*	*Observation Imbalanced Data Text to Predict Users Selling Products on Female Daily with SMOTE, Tomek Links, and SMOTE-Tomek Links*	Riset melakukan observasi ketidakseimbangan data untuk memprediksi pengguna *platform Female Daily* pada media sosial. Data ulasan pengguna *platform Female Daily* digunakan untuk mengevaluasi data teks yang tidak seimbang dengan teknik sentimen analisis. Algoritma yang digunakan dalam penelitian ini adalah *Support Vector Machine* (SVM) dan *Logistic Regression* (LR). Pengambilan sampel data tidak seimbang menggunakan algoritma dengan kombinasi

	Jonathan dkk, 2020 *(Proceedings - 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 202081–85)*	*Observation Imbalanced Data Text to Predict Users Selling Products on Female Daily with SMOTE, Tomek Links, and SMOTE-Tomek Links*	teknik SMOTE dan *Tomek Links*. Hasil komparasi dari algoritma *Support Vector Machine* (SVM) dengan skor 99,44 dan *Logistic Regression* (LR) dengan skor 99,17. SVM menunjukkan kategori *Precision Not Selling* tertinggi menggunakan SMOTE-*Tomek Links* 99,44% dengan peningkatan 0,18%, kategori *recall* tertinggi *Not Selling* menggunakan SMOTE dan *Tomek Links* 99,44% dengan peningkatan 0,03%, kategori penjualan presisi tertinggi menggunakan SMOTE 94,20% dengan peningkatan 5,14%, *recall* tertinggi kategori *Selling* menggunakan SMOTE-*Tomek Links* 94,43% dengan peningkatan 7,81% dan G-Mean tertinggi menggunakan SMOTE-*Tomek Links* 96,85 dengan peningkatan 4,06%.
7	Kumar dkk, 2020 *(Information and Management 56* (*2*), *172–184)*	*An Integrated Approach for Amazon Product Reviews Classification Using Sentiment Analysis*	Riset terkait klasifikasi sentimen analisis *review* produk pada situs web Amazon dengan data ulasan produk sebagai data. Algoritma yang digunakan dalam penelitian ini adalah *Naïve Bayes, Linear Support Vector Machine*, dan *Logistic Regression classifiers*. Skor akurasi dari masing-masing algoritma yang digunakan memperoleh hasil *Naïve Bayes* dengan skor akurasi 0,75, *Linear Support Vector Machine* dengan skor

			akurasi 0,77, dan *Logistic Regression classifiers* dengan skor akurasi 0,78.
8	Elreedy dan Atiya, 2019 *(Information Sciences 50532–64)*	*A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance*	Penelitian ini mengatasi masalah data tidak seimbang pada klasifikasi yang sering ditemukan pada klasifikasi *machine learning.* Penggunaan metode klasifikasi *machine learning* dan teknik SMOTE. Algoritma *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM) dengan teknik SMOTE digunakan dalam penelitian ini. Hasil tertinggi skor akurasi dari masing-masing algoritma KNN sebelum menggunakan SMOTE: 60,80% dan sesudah menggunakan SMOTE: 71,01% sedangkan SVM sebelum menggunakan SMOTE: 56,74% dan sesudah menggunakan SMOTE: 76,36%.
9	Al-Smadi dkk, 2018 *(Journal of Computational Science 27386–393)*	*Deep Recurrent neural network vs. Support Vector Machine for aspect-based sentiment analysis of Arabic hotels' reviews*	Penelitian ini menerapkan *machine learning* untuk analisis sentimen berbasis aspek (ABSA) dari ulasan Arabic Hotels. Pendekatan *deep recurcurent neural network* (RNN) dan *Support Vector Machine* (SVM) diimplementasikan. Dataset bersumber dari referensi ulasan Hotel Arab yang dianotasi menggunakan kerangka kerja ABSA yang disajikan dalam lokakarya Evaluasi Semantik 2016 (SemEval-ABSA16).

			Hasil evaluasi menunjukkan bahwa pendekatan SVM dengan skor akurasi 95,4% lebih unggul dibandingkan pendekatan RNN.
10	Singla dkk, 2017 *2017* *(International Conference on Intelligent Computing and Control (I2C2))*	*Sentiment Analysis of Customer Product Reviews Using Machine Learning*	Riset terkait sentimen analisis *review* produk HP pada pasar *online* amazon dengan menerapkan tiga metode *Naïve Bayes, Support Vector Machine* (SVM) dan *Decision Tree*. Hasil performen komparasi dengan tiga metode menunjukkan skor akurasi *Naïve Bayes*: 66,95, *Support Vector Machine*: 81,77, dan *Decision Tree*: 74,75.

2.2 Dasar Teori

2.2.1 Pasar *Online*

Pasar *online* merupakan tempat berlangsungnya *e-commerce*, yaitu aktivitas transaksi jual beli melalui media elektronik. Pasar *online* atau *online marketplace* adalah tempat pembeli dapat menemukan berbagai macam barang yang dijual oleh beberapa penjual dalam satu *platform* (Shankar dkk, 2021). Pemilik pasar *online* bertanggung jawab terhadap kelangsungan bisnis jual beli di *platform* miliknya (Guha Majumder dkk, 2022). Pemilik juga memantau transaksi uang yang terjadi antara penjual dengan pembeli, sedangkan penjual akan memastikan langsung pengiriman barang yang dibeli ke pembeli (Tong dan Chan, 2022). Hal ini membuat pembeli merasa lebih aman dengan transaksi yang dimediasi oleh *website* atau aplikasi.

Pasar *online* sebagai perantara digital yang menyatukan pembeli dan penjual untuk melakukan transaksi secara efisien (Shankar dkk, 2021). Manfaat pasar *online* yang ditawarkan penyelenggara pasar *online* untuk pelanggan, yaitu mulai dari kemudahan penggunaan aplikasi hingga komparabilitas yang lebih baik, seperti memudahkan pembeli untuk membandingkan toko satu dengan lainnya. Pembeli dapat membandingkan dengan harga yang lebih rendah namun kualitas lebih tinggi.

Umpan balik dari pelanggan dalam bentuk pengumpulan ulasan dan pengalaman yang unik (Guha Majumder dkk, 2022).

Pasar *online* untuk penjual menawarkan proposisi nilai yang bersifat ekonomis dan kuat kepada penjual. Proposisi ditawarkan pasar *online* untuk penjual dengan tujuan meramaikan jumlah penjual diantaranya pendapatan tambahan dengan sering diadakannya promo dari pasar *online* namun tidak membuat rugi penjual di pasar *online*, efisiensi waktu yang dikeluarkan untuk pengeluaran, kemudahan penggunaan, penjual tidak punya waktu atau tahu cara melakukan *SEO* atau membeli *Google Ads*. Penjual dapat merespon prospek di *thumbtack* karena lebih mudah (Akdane dkk, 2022).

2.2.2 *Web Scraping*

Web scraping adalah suatu teknik yang digunakan untuk mendapatkan informasi dari sebuah situs secara otomatis tanpa harus menyalinnya secara manual. Tujuan dari *web scraping* adalah mencari informasi tertentu dan kemudian mengumpulkannya ke dalam format berbeda. Manfaat dari *web scraping* yaitu membuat informasi yang diambil lebih fokus sehingga dapat memudahkan dalam melakukan pencarian sesuatu (Chapelle dan Eymeoud, 2022).

Kasus penggunaan utama *web scraping* termasuk pemantauan harga, intelijen harga, pemantauan berita, pembuatan prospek, dan riset pasar di antara banyak lainnya. *Web scraping* digunakan oleh orang-orang dan bisnis yang ingin memanfaatkan sejumlah besar data web yang tersedia untuk umum untuk membuat keputusan yang lebih cerdas (Wang dkk, 2022).

Proses *web scraping* yang dilakukan pertama adalah mengidentifikasi situs *web* target mengumpulkan halaman URL yang akan diekstrak datanya, membuat permintaan ke URL ini untuk mendapatkan HTML halaman, menggunakan *tool* pencari atau *sourcode* untuk menemukan data dalam HTML, kemudian menyimpan data dalam file *JSON* atau *CSV* atau format terstruktur lainnya (Mustopa dkk, 2020).

2.2.3 Analisis Sentimen

Analisis sentimen merupakan studi komputasi yang menganalisis terkait *opini*/pendapat, sentimen, dan emosi yang diekspresikan melalui teks (Li dkk, 2019). Penelitian terkait hal tersebut mulai populer pada tahun 2002 dan terus berkembang. Analisis sentimen memberikan luaran yaitu informasi yang dikategorikan menjadi nilai positif dan negatif (Das dkk, 2022). Proses untuk mendapatkan informasi tersebut, analisis sentimen menciptakan sebuah sistem yang kemudian dapat melakukan klasifikasi terhadap teks dalam suatu dokumen. Pemanfaatan analisis sentimen dilakukan untuk memeriksa pendapat terhadap suatu produk atau suatu kejadian (Borg dan Boldt, 2020).

Sistem analisis sentimen untuk analisis teks menggabungkan pemrosesan bahasa alami (NLP) dan teknik pembelajaran mesin untuk menetapkan skor sentimen berbobot ke entitas, topik, tema, dan kategori dalam kalimat atau frasa (Obiedat dkk, 2022). Analisis sentimen membantu analis data dalam perusahaan untuk mengukur opini publik, melakukan riset pasar, memantau reputasi merek dan produk, serta memahami pengalaman pelanggan (Haque dkk, 2018).

Peran utama *machine learning* dalam analisis sentimen adalah untuk meningkatkan dan mengotomatisasi fungsi analisis teks tingkat rendah yang dilakukan analis sentimen (Singla dkk, 2017). *Machine learning* membantu analis data memecahkan masalah rumit yang disebabkan oleh evolusi bahasa. Analisis sentimen secara umum digunakan untuk digunakan untuk memahami bagaimana perasaan pelanggan dan karyawan tentang subjek tertentu (Haque dkk, 2018).

Analisis sentimen untuk pelanggan dengan menganalisis tweet, ulasan *online,* dan artikel berita dalam skala besar. Analis isnis mendapatkan wawasan yang berguna tentang bagaimana perasaan pelanggan tentang merek, produk dan layanan. Perusahaan mengatasi masalah yang sedang tren sebelum menjadi viral dengan mengambil poin-poin untuk mengambil keputusan dalam sebuah diskusi (Wang dkk, 2022).

2.2.4 *Text Mining*

Text mining adalah proses mengubah teks tidak terstruktur menjadi format terstruktur untuk mengidentifikasi pola yang bermakna dan wawasan baru dengan menerapkan teknik analitik tingkat lanjut, seperti *Naïve Bayes, Support Vector Machines* (SVM) dan algoritma lainnya (Zamrodah, 2016). Teks adalah salah satu tipe data yang paling umum dalam *database* (Singla dkk, 2017). Masalah yang paling menantang dalam *text mining* adalah kompleksitas dan ambiguitas bahasa manusia. Kata yang sama yang digunakan dalam konteks yang berbeda dalam dokumen yang sama akan memiliki arti yang berbeda karena interpretasi yang berbeda (Ibrahim dan Wang, 2019).

Proses *text mining*, teks dokumen yang akan digunakan harus dipersiapkan terlebih dahulu sebelum dapat digunakan untuk proses utama (Alamoodi dkk, 2021). Proses mempersiapkan teks dokumen atau dataset mentah disebut juga dengan proses *text preprocessing*. *Text preprocessing* berfungsi untuk mengubah data teks yang tidak terstruktur atau sembarangan menjadi data yang terstruktur (Baek dkk, 2020). Proses yang dilakukan dalam tahapan *preprocessing* adalah sebagai berikut:

1. *Case Folding*

Case Folding adalah proses mengubah kata dalam bentuk huruf besar yang terdapat dalam sebuah kalimat menjadi bentuk huruf kecil. Hal ini bertujuan agar tidak terjadi *double indexing* karena dua kata yang sama dianggap berbeda disebabkan perbedaan huruf besar dan huruf kecil.

2. *Data Cleaning*

Data Cleaning atau pembersihan data dilakukan untuk memenuhi persyaratan pemodelan, yaitu melakukan segmentasi dengan menghapus beberapa simbol dan tanda baca (Singla dkk, 2017). Pembersihan data ini dilakukan dengan menghapus angka, *link url, emoticon, hashtag, mention*. Pembersihan ini dilakukan untuk mengurangi *noise* pada data.

3. Tokenisasi

Tokenisasi adalah proses pemecahan teks menjadi bagian-bagian yang lebih kecil dengan melibatkan pemrosesan awal teks untuk menghilangkan tanda baca

dan mengubah semua token menjadi huruf kecil. Proses tokenisasi ini terjadi pemotongan dokumen menjadi bagian yang lebih kecil. Keputusan pada tokenisasi ini akan memiliki pengaruh yang cukup signifikan pada analisis selanjutnya (Singla dkk, 2017).

4. Normalisasi

Normalisasi adalah mengembalikan bentuk penulisan ke bahasa yang sesuai dengan Kamus Besar Bahasa Indonesia (KKBI) serta mencocokkan pada dokumen latih maupun data uji dengan kata yang ada pada kamus bahasa tidak baku. Kata-kata yang disingkat seperti tdk, bgs, rsk dikembalikan ke bentuk normal menjadi tidak, bagus, rusak. Kamus pada normalisasi menggunakan link github "*https://raw.githubusercontent.com/nasalsabila/kamus-alay/master/colloquial-indonesian-lexicon.csv*".

5. *Filtering*

Tahap ini dilakukan penghapusan kata-kata yang sering muncul namun tidak penting atau tidak memberikan arti penting terhadap proses pengklasifikasian (Hadwan dkk, 2022). Contoh kata-kata tersebut ialah kata depan seperti "yang", "dan", "di" dan lain-lain.

6. *Stemming*

Stemming adalah proses mencari kata dasar pada suatu kata. *Stemming* memiliki dua cara, yaitu dengan menggunakan kamus dan menggunakan aturan-aturan imbuhan. *Stemmer* yang menggunakan kamus bahasa Indonesia memakai algoritma dari Nazief dan Adriani yang disimpan pada sebuah *library* bernama *library* Sastrawi. Tingkat kebenaran yang dihasilkan oleh algoritma ini lebih tinggi meskipun membutuhkan waktu komputasi yang tinggi (Singla dkk, 2017). S*temmer* yang menggunakan aturan-aturan imbuhan memakai algoritma Tala yang diadopsi dari algoritma *stemmer* bahasa inggris bernama *porter stemme*r. Penelitian menunjukkan bahwa di antara kedua cara tersebut, algoritma Nazief dan Adriani menjadi yang terbaik dalam melakukan *stemming* dikarenakan terdapat penambahan aturan untuk reduplikasi 6 dan penambahan aturan untuk awalan dan akhiran dalam meningkatkan presisi setiap kata yang di*stemming*.

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

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan salah satu metode untuk melakukan pembobotan kata dari proses ekstraksi kata dengan menerapkan perhitungan kata umum di *information retrival*. Metode pembobotan ini merupakan penggabungan antara *term frequency* dan *inverse document frequency*. *Term frequency* merupakan jumlah kemunculan sebuah *term* pada sebuah dokumen. Besarnya jumlah *term* yang muncul berbanding lurus dengan pembobotan yang diberikan. *Inverse document frequency* adalah proses untuk mengukur seberapa penting kata dalam suatu dokumen (Hunt, 2021).

Term Frequency (*TF*) menyatakan jumlah keberadaan *term* dalam suatu dokumen. Nilai $f_{t,d}$ adalah frekuensi (f) *term* (*t*) pada dokumen (*d*), misalnya suatu *term* (istilah) terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = 1 + *log* (5) =1.699. Dokumen yang tidak terdapat *term*, maka bobotnya adalah nol (0) (Hunt, 2021). Rumus *TF* logaritmik dapat dilihat pada persamaan 2.1.

$$TF = \begin{cases} 1 + log_{10} tf_{t,d}, & if \quad tf_{t,d} > 0 \\ 0, & if \quad tf_{t,d} = 0 \end{cases} \tag{2.1}$$

IDF (*Inverse Document Frequency*) merupakan sebuah perhitungan dari *term* yang didistribusikan secara luas pada koleksi dokumen. Frekuensi kata yang sering muncul pada *TF* menjadikan nilai semakin besar. Frekuensi kata yang semakin sedikit muncul pada *IDF* menjadikan nilai semakin besar. Rumus nilai IDF dapat ditentukan menggunakan persamaan 2.2.

$$IDF_t = log_{10} \frac{N}{df_t} \tag{2.2}$$

Dengan:

df_t = Jumlah dokumen yang mengandung *term* (*document frequency*)

N = Jumlah keseluruhan dokumen

Rumus bobot *term Weighting* ($W_{t,d}$) TF-IDF adalah mengalikan nilai TF dengan nilai IDF yang dapat dilihat pada persamaan 2.3.

$$W_{t,d} = tf_{t,d} \,.\, idf_t \tag{2.3}$$

Dengan:

$tf_{t,d}$ = Jumlah kemunculan *term* dalam dokumen

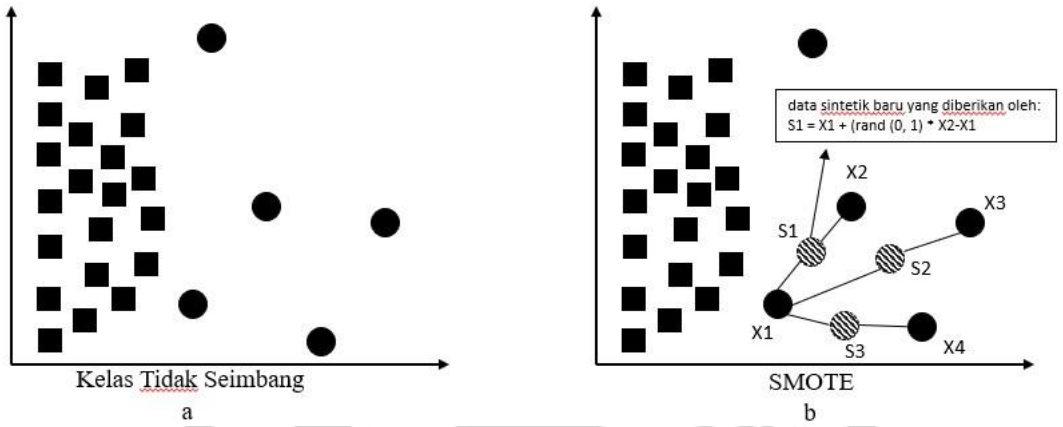
idf_t = Jumlah dokumen yang mengandung *term*

2.2.6 *Word Cloud*

Word cloud merupakan metode visualisasi dokumen teks. Word cloud merupakan representasi grafis dari sebuah dokumen yang dilakukan dengan *plotting* kata-kata yang sering mencul pada sebuah dokumen pada ruang dua dimensi. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut. Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut muncul dalam dokumen (Alamoodi dkk, 2021).

2.2.7 *Synthetic Minority Oversampling Technique* (SMOTE)

SMOTE adalah teknik *oversampling* untuk menghasilkan sampel sintetis pada kelas minoritas, yaitu jika suatu kelas dalam satu dataset memiliki banyak data yang jauh lebih sedikit dibandingkan dengan kelas lainnya. Jumlah data yang tidak seimbang yang terlalu banyak akan memengaruhi kemampuan prediksi model klasifikasi sehingga perlu untuk ditangani (Elreedy dan Atiya, 2019). Proses SMOTE dapat dilihat pada Gambar 2.1.



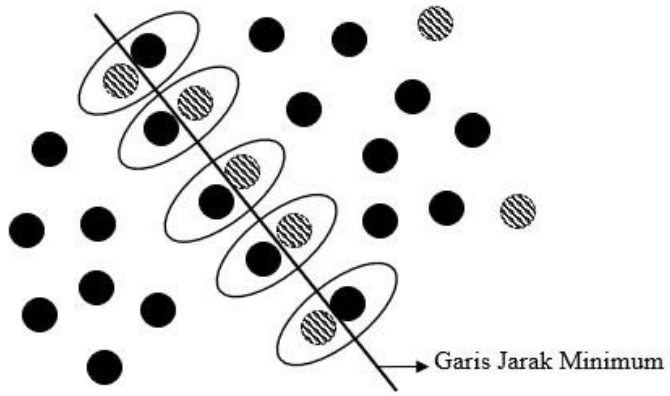
(https://towardsdatascience.com/smote-synthetic-data-augmentation-for-tabular-data-1ce28090debc)

Gambar 2. 1 Proses SMOTE

Proses SMOTE yang ditunjukkan pada Gambar 2.1 menghasilkan sampel sintetis kelas minoritas. Titik kotak adalah kelas mayoritas dan titik lingkaran adalah kelas minoritas. Kelas tidak seimbang pada Gambar a menunjukkan jumlah kelas mayoritas yang terlalu banyak dibandingkan kelas minoritas. SMOTE pada Gambar b diterapkan untuk menghasilkan data sintetik dari X1 dengan mempertimbangkan 3 tetangga terdekat (X2, X3 dan X4) untuk menghasilkan data sintetik S1, S2 dan S3. Proses SMOTE menghasilkan data sintetik baru untuk meningkatkan persentase data minoritas menjadi dua kali persentase sebelumnya.

2.2.8 *Tomek Links*

Tomek Links merupakan salah satu modifikasi dari teknik *undersampling Condensed Nearest Neighbors* (CNN) yang dikembangkan oleh Tomek (1976) (Ri dkk, 2016). *Tomek Links* digunakan untuk mengidentifikasi item data kelas mayoritas yang akan dihapus. *Tomek Links* terjadi antara dua item data yang memiliki kelas berbeda, tetapi merupakan tetangga terdekat satu sama lain (Spelmen dan Porkodi, 2018). Proses *Tomek Links* dapat dilihat pada Gambar 2.2.



(https://www.quantmetry.com/blog/classification-et-desequilibre-de-classes)

Gambar 2. 2 Proses *Tomek Links*

Proses *Tomek Links* yang ditunjukkan pada Gambar 2.2 mengidentifikasi item data kelas mayoritas yang akan dihapus. Titik lingkaran hitam adalah kelas mayoritas dan titik lingkaran terarsir adalah kelas minoritas. *Tomek Links* menghapus data kelas mayoritas yang memiliki kesamaan karakteristik dengan kelas minoritas pada garis jarak minimum sehingga data menjadi seimbang.

2.2.9 SMOTE -*Tomek Links*

Kombinasi SMOTE dan *Tomek Links* diperkenalkan pertama kali oleh Batista dkk (2003), metode ini menggabungkan kemampuan SMOTE untuk menghasilkan data sintetik untuk kelas minoritas dan kemampuan *Tomek Links* untuk menghapus data yang diidentifikasi sebagai *Tomek Links* dari kelas mayoritas, yaitu sampel data dari kelas mayoritas yang paling dekat dengan data kelas minoritas (Swana dkk, 2022). Proses dari kombinasi SMOTE dan *Tomek Links* adalah:

1. SMOTE sebagai langkah awal, dimulai dengan menambah jumlah observasi pada kelas minoritas dengan membuat objek atau observasi sintetis, yaitu objek baru yang tidak terdapat dalam dataset namun memiliki kemiripan dengan objek yang terdapat dalam dataset. Observasi sintetis dibentuk dari dua observasi, yaitu observasi pertama dipilih dari data kelas minoritas dan observasi kedua dari data kelas minoritas yang dipilih secara random dengan *k-nearest neighbor* observasi kelas minoritas yang pertama. Adanya

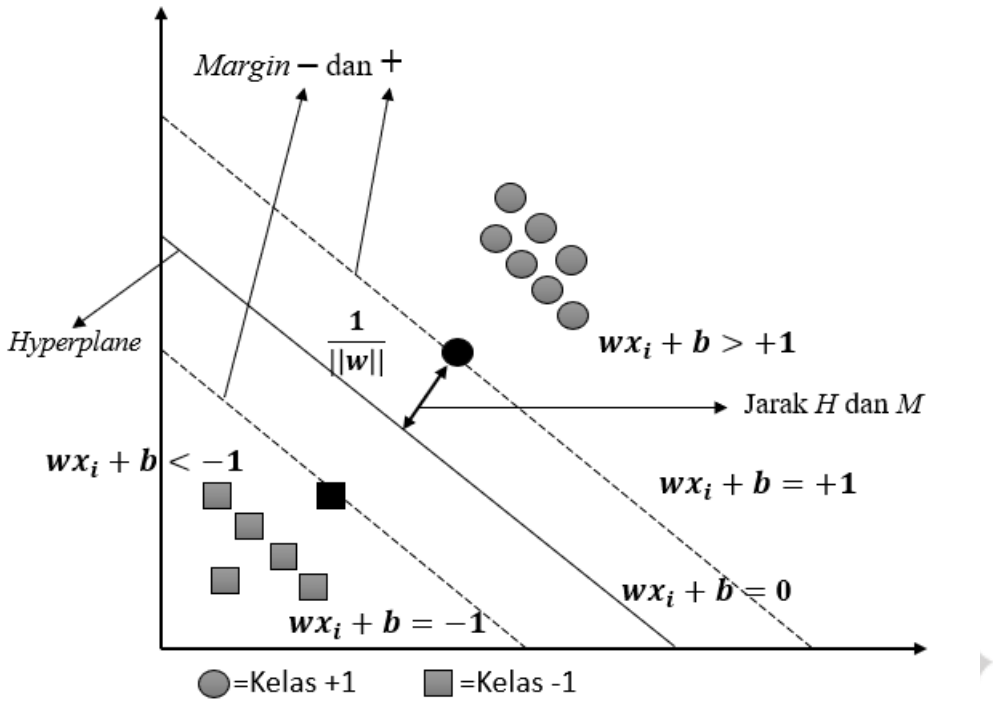
observasi sintetis tersebut maka jumlah observasi pada data kelas minoritas akan bertambah sehingga lebih seimbang dengan data kelas mayoritas.

2. Identifikasi *Tomek Links* pada data hasil SMOTE. Sepasang observasi disebut sebagai *Tomek Links* apabila kedua observasi tersebut merupakan tetangga terdekat namun memiliki kelas yang berbeda.
3. Pasangan observasi yang teridentifikasi sebagai *Tomek Links* dihapus dari dataset. Pengulangan identifikasi *Tomek Links* dilakukan hingga menghasilkan data yang bersih dari *noise*.

2.2.10 *Support Vector Machine* (SVM)

Support Vector Machine merupakan metode untuk melakukan prediksi dalam kasus klasifikasi maupun regresi (Borg dan Boldt, 2020). *Support Vector Machine* termasuk dalam *supervised learning* yang berarti model atau mesin mempelajari terlebih dahulu untuk melakukan klasifikasi dengan membagi data menjadi dataset, yaitu data *training* dan data *testing* (Lee dkk, 2022). Metode ini diperkenalkan pertama kali pada tahun 1992 oleh Vapnik. *Support Vector Machine* dapat menemukan fungsi pemisah yang bisa memisahkan dua dataset dari dua kelas yang berbeda. *Support Vector Machine* mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space* dengan memaksimalkan jarak antar kelas (AT dkk, 2016).

Hyperplane pemisah terbaik antara kedua kelas pola dapat ditemukan dengan mengukur *margin* dari *hyperplane* dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing kelas. Pola yang paling dekat ini disebut sebagai *Support Vector* (Ibrahim dan Wang, 2019). Garis solid menunjukkan *hyperplane* terbaik, yang terletak tepat pada tengah-tengah kedua kelas, sedangkan titik bulat dan kotak yang berada dalam adalah *support vector*. Usaha untuk menentukan *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM. Konsep SVM dapat dilihat pada Gambar 2.3.



(https://bsantosa.files.wordpress.com/2015/03/tutorial-svm-20158.pdf)

Gambar 2. 3 Konsep SVM

Konsep SVM yang ditunjukkan pada Gambar 2.3, *Margin* dengan garis putus-putus adalah jarak antara *hyperplane* dengan pola terdekat dari masing-masing kelas. Garis solid diantara *margin* sebagai *hyperplane* terbaik yang memisahkan kedua kelas. Titik kotak sebagai kelas -1 dan titik lingkaran sebagai kelas +1, sedangkan titik kotak hitam dan titik lingkaran hitam yang berada dekat dengan *margin* adalah *Support Vector*. Ilustrasi gambar sebagai usaha mencari *hyperplan*e terbaik yang berfungsi sebagai pemisah dua buah kelas -1 dan +1. Masalah klasifikasi dijabarkan dengan usaha menemukan *hyperplane* yang memisahkan dua kelompok. *Hyperplane* pemisah yang terbaik diantara kedua kelas ditemukan dengan cara mengukur *margin hyperplane* dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* dengan pola terdekat dari setiap kelas. Pola yang paling dekat disebut *Support Vector*. Usaha untuk mencari *hyperplane* adalah inti dari proses pembelajaran pada SVM. Persamaan garis *hyperplane* diasumsikan

kedua kelas -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* yang didefinisikan pada persamaan 2.4.

$$\vec{w}\vec{x} + b = 0 \tag{2.4}$$

Dengan:

$\vec{w}$ = bidang *hyper*

$\vec{x}$ = untuk memetakan setiap vektor masukan ke dalam ruang dimensi

b = bias

Pola $\overrightarrow{x_i}$ yang termasuk kelas -1 (sampel negatif) dapat dirumuskan sebagai pola yang memenuhi pertidaksamaan 2.5.

$$\vec{w}\vec{x} + b < -1 \tag{2.5}$$

Pola $\overrightarrow{x_i}$ yang termasuk kelas +1 (sampel positif) dirumuskan dengan pertidaksamaan 2.6.

$$\vec{w}\vec{x} + b \geq 1 \tag{2.6}$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu $\frac{1}{\|w\|}$ yang ekuivalen dengan meminimumkan $\|\vec{w}\|^2$. Hal ini dapat dirumuskan $\min_{\vec{w}} \tau(w)$ sebagai *Quadratic Programming* (QP) *problem*, yaitu mencari titik minimal persamaan 2.7 dengan memperhatikan pembatas persamaan 2.8.

$$\min_{\vec{w}} \tau(w) = \frac{1}{2}\|\vec{w}\|^2 \tag{2.7}$$

$$y_i(\overrightarrow{x_i}.\vec{w} + b) - 1 \geq 0, \forall i \tag{2.8}$$

Dengan:

y_i = korespondensi (penyampaian maksud) kelas

$\overrightarrow{x_i}$ = vektor input

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, diantaranya *Lagrange Multiplier*.

$$\mathrm{L}(\vec{w}, b, \alpha) = \frac{1}{2}|\vec{w}|^2 - \sum_{\mathrm{i}=\mathrm{l}}^{\mathrm{l}} \alpha_i \left(y_i\left((\overrightarrow{x_i}.\vec{w} + b) - 1\right)\right) \quad (2.9)$$

dengan $(i = 1,2, \dots, l)$

α_i adalah *Lagrange multipliers*, yang bernilai nol atau positif $\alpha \geq 0$. Nilai optimal dari persamaan 2.6 dapat dihitung dengan meminimalkan *L* terhadap $\vec{w}$ dan *b*, dan memaksimalkan *L* terhadap α_i . Sifat titik optimal gradien *L*=0, persamaan 2.10 dapat dimodifikasi sebagai maksimalisasi problem yang hanya mengandung saja α_i sebagaimana persamaan 2.11.

Maksimal:

$$\sum_{i=l}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=l}^{l} \alpha_i\, \alpha_j y_i y_j \overrightarrow{x_\iota \overrightarrow{x_j}} \quad (2.10)$$

Subjek:

$$\sum_{i=l}^{l} \alpha_i\, y_i = 0\,, \qquad \text{dengan} \qquad \alpha_{\mathrm{i}} \geq 0(\mathrm{i} = 1,2, \dots, \mathrm{l}) \quad (2.11)$$

Hasil dari perhitungan ini diperoleh α_i yang kebanyakan bernilai positif. Data yang berkorelasi dengan α_i yang positif inilah yang disebut sebagai *Support Vector*.

2.2.11 Ukuran Kebaikan Model

Ukuran kebaikan model digunakan untuk mengukur seberapa akurat prediksi dari suatu model klasifikasi. Cara untuk mengukur kebaikan model, salah satunya dengan menggunakan *confusion matrix*. *Confusion matrix* merupakan sebuah tabel yang terdiri atas banyaknya baris data uji yang diprediksi benar dan tidak benar oleh sebuah model klasifikasi (Lee dkk, 2022). Ilustrasi *Confusion Matrix* dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Ilustrasi *Confusion Matrix*

Nilai Aktual	**Nilai Prediksi**	
	Negatif	**Positif**
Negatif	Benar Negatif (TN)	Salah Positif (FP)
Positif	Salah Negatif (FN)	Benar Positif (TP)

Hasil dari *confusion matrix* yang digunakan dalam penelitian ini adalah nilai akurasi dan *F1-score*. Akurasi adalah banyaknya prediksi benar pada semua data yang diprediksi. Akurasi merupakan metode ukuran kebaikan model yang umum digunakan pada pemodelan klasifikasi. *F1-score* merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *F1-score* merupakan ukuran kebaikan 7 model yang bagus digunakan pada data yang tidak seimbang. Rumus perhitungan nilai akurasi, *Precission*, *Recall* dan *F1-score* berdasarkan ilustrasi *Confusion Matrix* pada Tabel 2.2 adalah sebagai berikut:

$$Akurasi = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (2.12)$$

$$Precission = \frac{TP}{TP+FP} \quad (2.13)$$

$$Recall = \frac{TP}{TP+FN} \quad (2.14)$$

$$F1-score = 2\ x\ \frac{(recall\ x\ precision)}{(recall+prescision)} \quad (2.15)$$

Dengan:

TP : Banyaknya prediksi kelas positif benar

TN : Banyaknya prediksi kelas positif salah

FP : Banyaknya kelas positif diprediksi salah

FN : Banyaknya kelas selain positif diprediksi benar

Recall : Rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif

Precision : Rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Precision* adalah tingkat ketepatan antara informasi yag diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *F*1-*score* merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan.

Ukuran kebaikan model selain menggunakan nilai akurasi juga dapat dilihat menggunakan AUC (*area under curve*). AUC menurut Wang dan Yao (2013) merupakan evaluasi model yang baik untuk data tidak seimbang. AUC adalah luas di bawah kurva ROC (*receiver operating characteristic*) yang memiliki rentang antara nol sampai dengan satu. Daftar nilai rentang AUC dan tingkatan klasifikasinya menurut Gorunescu (2011) dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Nilai rentang AUC dan tingkatan klasifikasi menurut Gorunescu (2011)

Nilai AUC	Tingkat klasifikasi
0,91 – 1,00	Sangat Baik
0,81 – 0,90	Baik
0,71 – 0,80	Cukup
0,61 – 0,70	Kurang
0,50 – 0,60	Gagal