BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Birim, dkk (2022) dalam penelitiannya membandingkan pengaruh *Topic Distribution* (TP) pada *Latent Dirichlet Allocation* (LDA) dengan *Cluster Distribution* (CD) pada *Agglomerative Hierarchical Clustering* (AHC) dalam deteksi ulasan palsu pada kumpulan data amazon.com untuk mengetahui hasil dari akurasi dan prediksi dari kedua metode. Hasil menunjukkan dengan membandingkan empat algoritma klasifikasi dan yang berkinerja paling baik adalah algoritma *Random Forest* yang dimana pengaruh dari *Verified Purchase* (VP) pada model TP memiliki kinerja terbaik dengan nilai akurasi, presisi, dan skor F1 masing-masing 80,40%, 86,69%, dan 78,74%.

Penelitian identifikasi ulasan palsu yang dilakukan oleh Alsubari, dkk (2021) mengusulkan model CNN-LSTM dalam dua jenis eksperimen. Eksperimen yang pertama yaitu eksperimen dalam domain dan yang kedua eksperimen lintas domain. Untuk eksperimen dalam domain, model diterapkan pada setiap kumpulan data secara satu per satu dan untuk eksperimen lintas domain semua kumpulan data dikumpulkan dan dimasukkan ke dalam satu kerangka data dan dievaluasi seluruhnya. Penelitian ini menggunakan empat kumpulan data ulasan palsu standar dari multidomain yaitu hotel, restoran, Yelp, dan Amazon. Hasil pengujian model pada kumpulan data eksperimen dalam domain masing-masing adalah 77%, 85%, 86%, dan 87% untuk kumpulan data restoran, hotel, Yelp, dan Amazon. Dan hasil eksperimen lintas domain, model yang diusulkan memperoleh akurasi 89%.

Deteksi ulasan palsu juga dapat dilakukan dengan menggunakan model *logistic regression* dengan mempertimbangkan fitur *review centric* seperti yang diusulkan oleh Daiv, dkk (2020). Model *logistic regression* digunakan sebagai pengklasifikasi untuk klasifikasi ulasan palsu dengan memilih nilai batas dari 0,5 dan mengklasifikasikan input dengan probabilitas lebih besar dari batas sebagai tinjauan palsu. Selain itu, fitur "*verified purchase*" digunakan sebagai perbandingan dengan model tanpa menggunakan fitur "*verified purchase*". Hasil yang didapatkan

menggunakan model yang diusulkan memperoleh akurasi sebesar 82% yang lebih unggul dari model tanpa menggunakan fitur "*verified purchase*" yang hanya mencapai akurasi sebesar 63%.

Pendekatan yang diusulkan oleh Bathla, dkk (2022) yaitu melakukan ekstraksi dari ulasan dan sentimen untuk digunakan dalam deteksi ulasan palsu. Aspek yang diekstraksi dimasukkan ke dalam algoritma *Convolutional Neural Network* (CNN) untuk pembelajaran replikasi aspek. Selanjutnya, algoritma *Long Short Term Memory* (LSTM) menggunakan elemen yang direplikasi untuk deteksi ulasan palsu. Kumpulan data yang digunakan yaitu Ott dan Yelp filter untuk membandingkan kinerja dengan pendekatan terbaru. Analisis eksperimen membuktikan bahwa pendekatan yang diusulkan mengungguli pendekatan terbaru.

Pada penelitian lain yang dilakukan oleh Zhang, dkk (2022) yang mengusulkan pendekatan baru yang disebut *ImDetector* untuk mendeteksi peninjau yang curang dengan ketidakseimbangan data berdasarkan model *weighted* LDA dan *K-Nearest Neighbor* (KNN) dengan *asymmetric Kullback-Leibler* (KL) *divergence*. *Asymmetric Kullback-Leibler* (KL) *divergence* diadopsi untuk membuat ukuran kesamaan antara peninjau yang bias terhadap minoritas curang dengan menggunakan klasifikasi KNN. Eksperimen dilakukan pada kumpulan data dari Yelp.com. Hasil menunjukkan bahwa pendekatan *ImDetector* memiliki kinerja yang lebih unggul daripada Teknik lainnya dalam deteksi peninjau penipuan.

Mendeteksi ulasan yang menipu serta menggali topik dan sentimen dari suatu ulasan, Dong, dkk (2018) mengusulkan model probabilistik *Unsupervised Topic Sentiment Joint* (UTSJ) berdasarkan LDA menggunakan kumpulan data dari Yelp.com. Model ini melakukan pengambilan algoritma *Gibbs Sampling* untuk memperkirakan parameter fungsi kemungkinan maksimum secara *offline* dan mendapatkan vektor distribusi probabilitas bersama dengan topik sentimen untuk setiap ulasan. Algoritma klasifikasi *Random Forest* diterapkan dalam penelitian ini. Hasil menunjukkan model yang diusulkan lebih baik daripada model dasar seperti *character n-grams in token, n-grams, Part of Speech* (POS), *Joint Sentiment/Topic* (JST), dan LDA.

Cagnina dan Rosso (2017) mengusulkan n-gram karakter dalam token, varian yang efisien dan efektif dari model tradisional karakter n-gram yang digunakan untuk mendapatkan representasi opini berdimensi rendah untuk mendeteksi ulasan menipu. Algoritma *Support Vector Machine* juga digunakan sebagai pengklasifikasi untuk mengevaluasi model yang diusulkan pada kumpulan data ulasan hotel, dokter, dan restoran. Eksperimen dilakukan pada kasus intra dan lintas domain. Tujuan dari penelitian ini untuk mengevaluasi model yang diusulkan dalam skenario lintas domain yang realistis yang dimana ulasan yang menipu dapat ditemukan di satu domain tetapi tidak di domain lain. kesimpulan yang didapatkan dengan menggunakan karakte n-gram dalam token memungkinkan untuk mendapatkan hasil yang kompetitif dengan representasi dimensi yang rendah.

Penelitian lain yang menyaring ulasan palsu dilakukan oleh Du, dkk (2020) dengan memperkenalkan model baru yang disebut *Sentence Joint Topic Sentiment Model* (SJTSM). SJTSM menggabungkan struktur kalimat ulasan dan informasi label sentiment kata berdasarkan model LDA untuk mengekstrak fitur ulasan. Model yang diusulkan menggunakan algoritma *Gibbs* untuk memperkirakan parameter kemungkinan maksimum dan mengambil vector distribusi topik sentiment sebagai fitur tinjauan. Kemudian sistem pengklasifikasi ganda diterapkan yang mencakup algoritma *Decision Tree*, *Naïve Bayes*, dan *Support Vector Machine* (SVM) yang mengambil vektor fitur ulasan yang diekstraksi sebagai inputnya. Hasil menunjukkan SJTSM mampu menghasilkan kinerja terbaik di setiap kumpulan data yang diterapkan.

Identifikasi ulasan palsu yang dilakukan oleh Wang, dkk (2018) mengusulkan dua jenis fitur semantik baru yaitu fitur *readability* dan fitur topik. Selain itu, satu set fitur perilaku juga diusulkan dalam penelitian ini. Eksperimen dilakukan pada data kehidupan nyata dari Yelp. Algoritma pembelajaran mesin juga diterapkan untuk mengklasifikasi kumpulan data yang diusulkan. Hasil menunjukkan bahwa meskipun fitur perilaku meraih akurasi deteksi yang sangat tinggi, fitur semantik juga bekerja dengan baik pada kumpulan data yang diusulkan. Diantara fitur semantik, fitur *readability,* fitur topik, dan fitur n-gram memberikan hasil yang kurang lebih sama. Untuk fitur n-gram, vektor fitur dibangun

menggunakan nilai *count* dan Tf-idf dari konten ulasan. Secara keseluruhan algoritma *Logistic Regression* menunjukkan kinerja terbaik dengan akurasi mencapai 97.2% di semua fitur.

Rahayu, dkk (2022) menggunakan KNN untuk pengklasifikasian teks karena bersifat *self learning*, sehingga dapat mempelajari struktur data yang ada dan mengkategorikan data nya sendiri melalui tetangga terdekat. Data diambil dari *website female daily* yang merupakan data ulasan produk *sunscreen brand* "Wardah", "Azarine", dan "Skin Aqua". Nilai k yang ditetapkan untuk parameter KNN pada penelitian ini yaitu k=1 dan k=3. Algoritma *Particle Swarm Optimization* (PSO) digunakan dalam penelitian ini untuk meningkatkan akurasi hasil klasifikasi dan memperoleh hasil terbaik 92.80% pada produk *sunscreen brand* "Azarine".

Penelitian yang dilakukan Aburomman dan Reaz (2016) menggunakan SVM dan KNN dengan algoritma optimasi PSO untuk membuat kombinasi klasifikasi dengan akurasi yang lebih baik untuk deteksi intrusi. Nilai parameter k yang ditetapkan untuk KNN antara lain k=[1,3,5,7,9,11]. Hasil yang diperoleh dari penggunaan PSO berhasil meningkatkan nilai akurasi dengan akurasi tertinggi 92.897%.

2.2 Dasar Teori

2.2.1 Ulasan Palsu

Keberhasilan atau kegagalan bisnis *e-commerce* sangat dipengaruhi oleh ulasan produk *online*. Dalam pembelian suatu produk atau jasa, pembeli biasanya akan mencari tau informasi penggunaan produk atau jasa tersebut dari ulasan online yang diposting oleh pelanggan sebelumnya untuk mendapatkan rekomendasi detail produk dan membuat keputusan pembelian. Namun juga terdapat ulasan palsu yang dapat menghambat berkembangnya produk *e-bussiness* yang ditulis oleh orang yang sengaja menipu. Ulasan ini dapat menyebabkan kerugian finansial bagi pelaku bisnis *e-commerce* dan dapat menyesatkan konsumen untuk mengambil keputusan yang salah. Dengan demikian, pengembangan sistem pendeteksi ulasan palsu diperlukan untuk bisnis *e-commerce* (Alsubari dkk, 2021).

Motivasi seseorang menulis ulasan palsu pada suatu produk *e-bussiness* karena ulasan tersebut dapat mempengaruhi keputusan pembelian, reputasi produk, dan ingin mendapatkan keuntungan. Contoh kalimat yang merupakan kalimat dari ulasan palsu seperti:

"Kamar yang tidak rapi dan gelap. Pemandangan yang buruk, suasana yang sangat bising, tempat tidur yang tidak nyaman, dll. Semua staf yang melayani dengan buruk dan tidak ramah. Untuk uang dan kualitas yang juga sangat buruk serta lokasi yang tidak strategis."

Kalimat di atas sangat jelas bahwa *spammer* mengubah polaritas aspek seperti kamar, tempat tidur, staf, dan lokasi. Sentimen ruangan disimpulkan dari *tag* seperti indah, bersih, dan cerah kemudian polaritas ruangan diubah dengan menggunakan kata-kata seperti tidak rapi dan gelap dalam ulasan palsu (Bathla dkk, 2022).

2.2.2 *E-Commerce* Amazon

Perdagangan elektronik atau yang biasa dikenal sebagai *e-commerce* adalah transaksi barang atau jasa melalui internet. Transformasi digital dari model bisnis tradisional memungkinkan pembeli untuk menjelajahi dan membeli produk secara *online* hanya dengan di dalam rumah atau dimanapun mereka berada. Platform *e-commerce* menawarkan beberapa keunggulan seperti proses pembelian yang lebih cepat, pengurangan biaya, fleksibilitas bagi pelanggan, perbandingan produk dan harga, respon yang lebih cepat terhadap permintaan pembeli atau pasar, dan menyediakan beberapa mode pembayaran. Selain itu, *e-commerce* juga sangat penting bagi perekonomian dan bahkan sangat penting di masa pandemi yang dimana pemerintah memberlakukan peraturan untuk tetap tinggal di rumah untuk mengurangi jumlah keramaian di tempat umum. Karena situasi tersebut, menyebabkan meningkatnya permintaan untuk belanja secara *online* seperti halnya belanja kebutuhan sehari-hari (Rodrigues dkk, 2022).

Salah satu *e-commerce* yang paling terkenal saat ini yaitu Amazon. Amazon didirikan pada tahun 1994 oleh Jeffrey P. Bezos. Sungai Amazon merupakan sungai terbesar di dunia yang menjadi inspirasi dari nama *e-commerce* ini. Sama halnya dengan sungai Amazon, Bezos berambisi agar perusahaan

Amazon bisa menjadi perusahaan terbesar di dunia. Setelah satu tahun pengembangan perangkat lunak dengan tim yang terdiri dari 10 karyawan, maka pada bulan Juli 1995 situs web Amazon resmi diluncurkan (Wells dkk, 2018).

Amazon telah memperluas bisnisnya dengan mencakup barang-barang ritel lainnya seperti kosmetik dan peralatan elektronik. Karena keberhasilannya, Amazon berhasil mendapatkan peringkat yang tinggi diantara toko ritel *online* lainnya yang terkenal di dunia. Mayoritas pendapatan Amazon berasal dari penjualan produk ritel, layanan langganan ritel seperti Amazon *Prime* atau AWA dan mesin pembaca buku elektronik yaitu Amazon *Kindle*. Pada tahun 2022, pendapatan Amazon diprediksi mencapai 356 miliar dolar AS dan jumlah pelanggan Amazon meningkat dari yang hanya 200.000 pelanggan pada tahun 2016 menjadi lebih dari satu juta pelanggan pada tahun 2017 (Tangmanee dan Jongtavornvitaya, 2022).

2.2.3 Prapengolahan Teks

Prapengolahan teks merupakan prosedur yang penting dalam *Natural Language Processing* (NLP). Tujuan dari prapengolahan teks untuk dapat merekonstruksi teks menjadi bentuk yang lebih mudah dicerna untuk algoritma pembelajaran mesin (Qorib dkk, 2022). Berikut merupakan langkah-langkah dari prosedur prapengolahan teks yang antara lain sebagai berikut:

1. *Case folding*

Case folding adalah metode untuk mengubah semua ukuran huruf dalam kumpulan data. Hal ini dilakukan untuk mempermudah proses analisis kumpulan data dan mengurangi jumlah memori yang digunakan. Seperti contoh pengubahan kalimat “Hutan Kalimantan sedang terjadi KEBAKARAN BESAR!” menjadi “hutan kalimantan sedang terjadi kebakaran besar!” (Mustaqim dkk, 2019).

2. Penghapusan tanda baca

Pada proses ini, semua tanda baca seperti tanda seru, tanda tanya, tanda titik dua, koma, dan lain-lain dihapus dari kalimat. Semua tanda baca tersebut tidak

memiliki arti apapun dan tidak akan berpengaruh pada proses analisis teks (Birim dkk, 2022).

3. *Stemming*

Stemming merupakan teknik mereduksi kata agar menjadi bentuk dasar, batang kata, atau bentuk akarnya. Seperti contoh kata melihat, terlihat, dan dilihat yang direduksi menjadi kata lihat. Dengan kata lain, proses *stemming* melakukan pengurangan infleksi dalam kata-kata ke bentuk akarnya. Hal ini dilakukan karena beberapa kata mungkin tidak *valid* dalam suatu bahasa tertentu (Qorib dkk, 2023).

4. *Lemmatization*

Berbeda dengan *stemming, lemmatization* adalah proses mereduksi kata yang diinfleksikan dengan benar dan memastikan bahwa kata dasar tersebut memiliki bahasa sesuai kamus. *Lemmatization* didasarkan pada kosa kata dan bentuk kata aslinya(Qorib dkk, 2023).

5. Tokenisasi

Tokenisasi adalah proses membagi teks menjadi kata-kata individual yang disebut dengan token. Seperti contoh dari kalimat berikut “saya sedang mencari meja murah” diubah menjadi [“saya”, “sedang”, “mencari”, “meja”, “murah”] setelah dilakukan proses tokenisasi (Birim dkk, 2022).

6. *Stopword removal*

Stopword adalah kata-kata yang sangat umum dengan fungsi struktural utama yang berulang kali digunakan dalam teks. Kata-kata tersebut memiliki sedikit makna dan fungsinya hanya sebagai sintaksis yang tidak menunjukkan pokok bahasan dan tidak memiliki pengaruh apapun pada analisis tekstual. Kata-kata tersebut dapat dihapus dari teks untuk membantu mengidentifikasi kata-kata yang bermakna (Alami dkk, 2021).

2.2.4 Analisis Sentimen

Analisis sentimen didefinisikan sebagai tugas untuk menemukan pendapat penulis tentang suatu entitas tertentu. Proses dasar analisis sentimen mengandung dua tugas yaitu pengenalan emosi yang mengekstraksi sekumpulan label emosi dan deteksi polaritas yang mengklasifikasikan label emosi menjadi positif atau negatif. Klasifikasi pada analisis sentimen dapat dilakukan pada seluruh dokumen (untuk sentimen tingkat dokumen), pada kalimat individual (untuk sentimen tingkat kalimat), dan pada aspek entitas tertentu (untuk sentimen berbasis aspek) (Zhang dkk, 2021). Dalam analisis sentimen, ada dua pendekatan yang dapat digunakan dalam mengekstrak sentimen dari ulasan yang diberikan dan mengklasifikasikannya sebagai positif, negatif, dan netral. Pertama yaitu pendekatan secara *machine learning* yang dimana secara otomatis mengklasifikasikan ulasan yang membutuhkan data pelatihan (Bonta dkk, 2019). Kedua yaitu pendekatan berbasis leksikon yang dimana membutuhkan kamus yang berisi leksikon yang telah ditentukan sebelumnya dan informasi tentang polaritas kata-kata yang berhubungan dengan sentimen (Kim dan Lim, 2021).

Ada banyak Teknik yang dapat diterapkan pada teks alami untuk menentukan sentimen seperti ekstraksi fitur, studi emotikon, tokenisasi, dll. Selama proses analisis sentimen, kata-kata positif dan negatif diekstraksi dari teks dan diberi skor dari kamus kata (Ahuja dan Dubey, 2017). Analisis sentimen dapat mengubah data teks yang luas dan tidak terstruktur menjadi persepsi yang terstruktur dan dapat diukur. Jumlah kata positif, negatif, dan netral dalam setiap dokumen dihitung untuk menentukan skor sentimen dari dokumen teks. Untuk menghitung skor sentimen teks, setiap kata positif dihitung sebagai +1, negatif dihitung sebagai -1, dan netral dihitung sebagai 0 (Qorib dkk, 2022).

2.2.5 Analisis Sentimen Berbasis Leksikon

Menurut Keshavarz dan Abadeh (2017), Leksikon sentimen adalah kumpulan kata dan frasa yang diasosiasikan dengan skor numerik yang menunjukkan sentimen atau orientasi emosional dari kata-kata tersebut. Analisis sentimen leksikon dapat dibuat melalui anotasi manusia atau dengan menggunakan

mekanisme yang otomatis. Leksikon dapat membedakan antara tujuan umum atau khusus domain. Pada tujuan umum, yaitu digunakan untuk berbagai aplikasi karena dibuat untuk mencakup leksikon yang luas dan umum. Pada tujuan khusus domain, yaitu khusus untuk domain aplikasi tertentu (Consoli dkk, 2022). Sentimen analisis berbasis leksikon mengeksploitasi sumber daya leksikal yang disebut leksikon terpolarisasi, yaitu daftar kata yang dianotasi dengan skor polaritas negatif atau positif. Skor tersebut mencerminkan kekuatan dan intensitas sentimen yang diungkapkan oleh kata tersebut. Polaritas akhir dari suatu dokumen atau kalimat diperoleh dengan menjumlahkan nilai polaritas dari setiap kata penyusun teks (Polignano dkk, 2022).

Menurut Hernandez, dkk (2023) Ada dua langkah utama yang dapat digunakan dalam pendekatan leksikon. Langkah-langkah tersebut antara lain:

1. Langkah pertama dalam pendekatan leksikon melibatkan penentuan daftar kata kunci yang relevan dan penting untuk analisis dokumen. Daftar kata kunci ini disebut juga sebagai kamus yang harus menyertakan istilah umum yang digunakan dalam dokumen dan menunjukkan konten dan maknanya. Untuk membuat kamus, yang harus diperhatikan yaitu dalam meninjau dokumen dan mengidentifikasi kata kunci yang paling penting dan relevan. Proses ini dapat melibatkan pra-pemrosesan dokumen untuk menghapus informasi yang tidak relevan, seperti penghapusan *stopword* dan tanda baca, *stemming* dan *lemmatization* untuk mengurangi variasi kata. Setelah kata kunci yang relevan diidentifikasi, maka kata kunci tersebut dapat digunakan untuk membuat kamus yang akan digunakan pada proses selanjutnya.
2. Langkah kedua yaitu mewakili dokumen dalam hal frekuensi kata kunci dalam kamus. Pada Langkah ini, frekuensi setiap kata kunci dalam kamus dihitung dan digunakan sebagai fitur untuk dokumen tersebut. Hal ini memungkinkan representasi konten dokumen yang efisien dalam bentuk numerik yang dapat memudahkan untuk dianalisis dan diklasifikasi.

Keuntungan utama dalam menggunakan pendekatan berbasis leksikon adalah lebih mudah dipahami dan mudah dimodifikasi oleh manusia. Dengan menggunakan teknik ini, orientasi semantik dapat ditangkap dan diberi label

sebagai positif, negatif, dan netral (Neogi dkk, 2021). Sebagian besar leksikon sentimen telah dibangun untuk sentimen berbahasa Inggris. Jika leksikon sentimen tidak tersedia untuk bahasa tertentu, maka teks dapat diterjemahkan terlebih dahulu. Namun, biasanya terjemahan teks dapat menghasilkan kinerja metode analisis sentimen yang lebih buruk (Consoli dkk, 2022).

2.2.6 Analisis Sentimen *Textblob*

Textblob adalah analisis sentimen leksikon yang populer berbasis model pustaka yang tersedia dalam Phyton yang menyediakan pemrosesan teks yang disederhanakan. Banyak ilmuwan data yang menggunakan *Textblob* karena bisa menjadi pustaka yang lebih cepat dan ringkas. API (*Application Programming Interface*) yang sederhana dari *Textblob* memfasilitasi banyak pemrosesan teks umum dan tugas dari *Natural Language Processing* (NLP) seperti terjemahan bahasa, POS (*Parts of Speech*) *tagging*, tokenisasi,ekstraksi frasa, klasifikasi, analisis sentimen, dll. *Textblob* memungkinkan analisis sentimen teks yang tepat dan memungkinkan terjemahan teks dari satu bahasa ke bahasa lain (Aljedaani dkk, 2022).

Ketika melakukan analisis sentimen dengan *Textblob,* ada dua skor yang diperoleh untuk setiap dokumen yang dianalisis yaitu subjektivitas dan polaritas. Skor subjektivitas adalah nilai dalam rentang [0,1] dimana 0 diartikan sebagai sangat objektif dan 1 diartikan sebagai sangat subjektif. Skor polaritas adalah nilai dalam rentang [-1,1] dan mewakili keseluruhan sentimen dokumen (Batista dkk, 2021). Subjektivitas memberikan informasi jumlah opini pribadi dan informasi faktual yang ada dalam ulasan. Ketika subjektivitasnya tinggi menandakan bahwa ada lebih banyak pendapat pribadi dalam ulasan. *Textblob* memiliki satu parameter lain yaitu intensitas yang diperlukan untuk menghitung subjektivitas. Intensitas menentukan apakah sebuah kata memiliki pengaruh pada kata berikutnya. Berdasarkan nilai polaritas, skor sentimen akan diberikan dan perhitungannya dihitung sedemikian rupa sehingga jika skornya kurang dari 0 maka sentimen akan dikembalikan sebagai negatif dan sentimen akan dikembalikan sebagai positif jika

polaritas lebih besar dari 0. Dalam kasus lainnya, skor akan menjadi 0 dan sentimen dikembalikan sebagai netral (Neogi dkk, 2021).

Textblob menggunakan algoritma *Pattern Analizer* sebagai standar untuk menghitung polaritas dan subjektivitas. Algoritma *Pattern Analizer* didasarkan pada *Pattern library* yang ditulis dengan *Python* yang terbagi menjadi beberapa modul untuk penambangan data (Google, Twitter, dan Wikipedia API), *Natural Language Processing* (NLP) (POS *taggers,* n-gram *search*, analisis sentimen, dan *World Net*), pembelajaran mesin dengan SVM, *network analysis,* dan visualisasi. Pada algoritma *Pattern Analyzer*, kelas *_text.py* bertanggung jawab untuk menghitung sentimen. Properti sentimen yang diimplementasikan dalam modul *textblob.en.sentiments()* mengembalikan *tuple* dalam bentuk sentimen (polaritas dan subjektivitas). Untuk menghitung polaritas $(\bar{P})$ dan subjektivitas $(\bar{S})$ dari satu kata, algoritma *Pattern Analyzer* menerapkan konsep rata-rata aritmatika sederhana yang dimana jumlah polaritas atau subjektivitas dari kata $(\sum_{i=1}^{n} X_i)$ dibagi menjadi beberapa kali nilai (n) yang merupakan kata yang muncul dalam kumpulan data leksikal. Berikut merupakan rumus perhitungan polaritas dan subjektivitas suatu kata dalam kumpulan data leksikal pada persamaan 2.5 dan 2.6 (Junior dkk, 2021).

$$\text{Perhitungan polaritas} = (\bar{P}(x) = \frac{1}{n}\sum_{i=1}^{n} X_i) \quad (2.5)$$

$$\text{Perhitungan subjektivitas} = (\bar{S}(x) = \frac{1}{n}\sum_{i=1}^{n} X_i) \quad (2.6)$$

2.2.7 *CountVectorizer*

CountVectorizer (CV) biasa digunakan untuk perhitungan numerik fitur kelas dan metode ekstraksi fitur teks. Untuk setiap teks pelatihan hanya mempertimbangkan setiap kata dalam frekuensi pelatihan dalam teks (Yang dkk, 2020). Seluruh proses vektorisasi dapat dibagi menjadi dua Langkah. Pertama, CV membuat kamus kata dengan menghitung jumlah kata dalam *dataset*. Melalui kamus kata, dapat ditemukan kata dan frekuensi yang sesuai. Kedua, CV dapat mengubah kalimat menjadi vektor. Vektor yang dihasilkan dari CV disejajarkan karena panjang vektor adalah panjang dari kamus kata (Zheng dkk, 2020). Dalam

Teknik CV, baik menghitung kemunculan token maupun proses tokenisasi keduanya sama-sama dilakukan. CV memiliki banyak parameter untuk menyempurnakan jenis fitur. Untuk dapat membangun fitur, ada tiga parameter yang dapat digunakan yaitu *unigram* (*min* df=1), *bigram* (*min* df=2), dan *trigram* (*min* df=3). Setiap vektor (*term*) dalam dokumen merepresentasikan nama fitur individu dan kemunculannya digambarkan melalui matriks agar lebih mudah dipahami (Kaur dkk, 2020).

Countvectorizer mengubah dokumen *d* menjadi vektor numerik $d = \{u_1, u_2, \dots, u_i, \dots, u_T\}$ dimana (u_i) adalah bobot kata dengan angka *i* pada dokumen *d*. Fitur *i* dari dokumen merupakan jumlah waktu kemunculan kata *i* pada CV yang dimana u_i terdiri dari frekuensi kemunculan setiap kata *i* pada dokumen *d* (Castorena dkk, 2021). Contoh dari perhitungan matriks CV yaitu misalkan kita memiliki tiga dokumen berbeda yang berisikan kalimat seperti "*book is good*", "*book is average*", dan "*book is nice*". Maka, matriks yang dihasilkan yaitu matriks dengan ukuran 3x5 karena ada 3 dokumen dan 5 fitur berbeda seperti *book, is, good, average,* dan *nice*. Unsur-unsur matriks direpresentasikan pada Tabel 2.1.

Tabel 2.1 Skema matriks *CountVectorizer*

	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5
Kalimat 1	1	1	1	0	0
Kalimat 2	1	1	0	1	0
Kalimat 3	1	1	0	0	1

Pada Tabel 2.1 dapat dilihat jika setiap nilai 1 dalam satu baris sesuai dengan keberadaan fitur dan 0 mewakili tidak adanya fitur dari dokumen tertentu (Tripathy dkk, 2018).

2.2.8 *K-Nearest Neighbor*

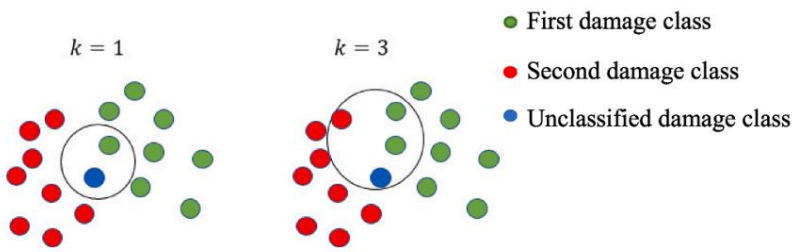
Algoritma *K-Nearest Neighbor* (KNN) merupakan algoritma klasifikasi yang tidak membuat asumsi apapun pada distribusi data yang mendasari dan bekerja berdasarkan kesamaan fitur. Dalam algoritma KNN, titik data

diklasifikasikan berdasarkan suara mayoritas tetangganya, dengan titik data dialokasikan ke kelas yang paling umum diantara K tetangga terdekatnya. Selain detail teknis, KNN juga menghitung batas keputusan untuk lebih dari dua kelas untuk mengklasifikasikan poin baru. Jumlah tetangga K adalah *hyperparameter* yang menyesuaikan kompleksitas pengklasifikasi ini. Nilai K yang kecil melatih kembali wilayah prediksi yang diberikan dan mengharuskan pengklasifikasi menjadi buta terhadap distribusi umum. Oleh karena itu, K yang bernilai kecil menyebabkan bias yang rendah dan varians yang tinggi. Sedangkan untuk K yang bernilai besar menghasilkan varians yang lebih rendah tetapi menyebabkan bias yang besar karena memberikan batas keputusan yang halus (Kiani dkk, 2019).

Kinerja dari algoritma KNN sangat bergantung pada metrik jarak yang digunakan untuk menentukan tetangga paling dekat. Sebagian besar algoritma KNN menggunakan metrik *Euclidean* dan *Manhattan* untuk menghitung jarak tetangga terdekat. Namun dalam analisis teks, *Cosine Similarity* merupakan matrik yang banyak digunakan pada penelitian sebelumnya. *Cosine Similarity* memiliki keunggulan karena tidak bergantung pada ukuran vektor fitur ketika teks diproses sebagai kumpulan kata (Cunningham dan Delany, 2021). Pada persamaan 2.7 menunjukkan θ merupakan sudut yang dibentuk oleh vektor A dan B yang merupakan representasi dari *Cosine Similarity*. A dan B mewakili vektor fitur, sedangkan A_i^2 dan B_i^2 masing-masing menunjukkan panjangnya (Han dkk, 2019).

$$Similarity = cos\theta = \frac{\sum_{i=1}^{p} A_i \times B_i}{\sqrt{\sum_{i=1}^{p} A_i^2} \times \sqrt{\sum_{i=1}^{p} B_i^2}} \quad (2.7)$$

Pada Gambar 2.1 menunjukkan secara sederhana ilustrasi cara kerja algoritma KNN. Algoritma KNN dapat menetapkan kelas kerusakan ke kelas yang tidak diklasifikasikan (biru) berdasarkan suara mayoritas dari K tetangga terdekat di set pelatihan (kelas kerusakan hijau dan merah). Pada contoh ditunjukkan dengan menetapkan nilai k=1 dan k=3 (Ghiasi dkk, 2022).



Gambar 2.1 Skema klasifikasi algoritma KNN 2 dimensi menggunakan k=1 dan k=3

2.2.9 *Particle Swarm Optimization*

Particle Swarm Optimization (PSO) merupakan algoritma *Swarm Intelligence* yang diperkenalkan pada tahun 1995. Algoritma ini terinspirasi dari perilaku sosial seperti kawanan burung, kawanan ikan dan khususnya dalam teori kawanan (Fitas dkk, 2022). Algoritma PSO diterapkan secara luas untuk berbagai bidang seperti penjadwalan tugas, sistem fuzzy, sistem kontrol dan tenaga, dan klasifikasi. Pada algoritma PSO, populasi dianggap sebagai kawanan (*swarm)* . Dalam kawanan, setiap individu direpresentasikan sebagai sebuah partikel. Sebuah *swarm* berisikan sejumlah partikel (n) dan setiap partikel mengimplikasikan solusi kandidat pada ruang pencarian dimensi d. Setiap partikel dikaitkan dengan kecepatan tertentu. Dalam populasi, partikel ke-i direpresentasikan menggunakan posisi P_i dengan (p_{i1}, p_{i2},…, p_{id}) dan kecepatan V_i dengan (v_{i1},v_{i2},…,v_{id}). Setiap partikel bergerak dalam ruang pencarian untuk mendapatkan solusi optimal. Pergerakan setiap partikel diarahkan oleh posisi *pbest* dan posisi *gbest*. Setiap kandidat solusi dianggap sebagai *pbest* (*pbest*$_1$, *pbest*$_2$, *pbest*$_{id}$). Posisi terbaik seluruh kawanan dilambangkan dengan *gbest* seperti (*gbest*$_{i1}$, *gbest*$_{i2}$, *gbest*$_{id}$). Nilai *fitness* yang digunakan untuk mengevaluasi posisi terbaik partikel. Posisi saat ini, kecepatan partikel ke-i ditingkatkan menggunakan persamaan 2.8 dan 2.9.

$$V_{(t+1)} = W \times (V_t + C_1 \times rand(0,1) \times pBest_t) - (currentvalue_t + C_2 \times rand(0,1)) \quad (2.8)$$

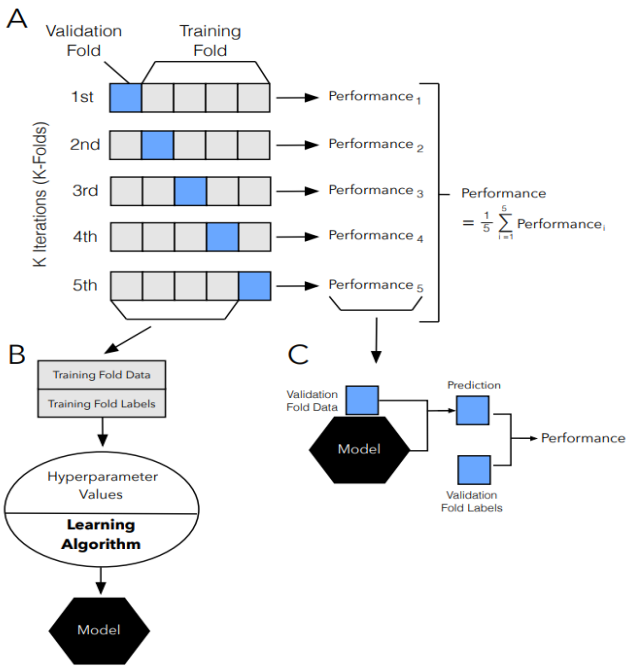
$$currentvalue_{(t+1)} = currentvalue_t + V_{(t+1)} \quad (2.9)$$

V(t+1) mewakili kecepatan partikel sebelumnya dan V(t) menunjukkan kecepatan pembaruan partikel. C1 dan C2 ada sebagai konstanta. Faktor W menunjukkan bobot inersia yang berkisar antara [0.0,1.0] untuk mengendalikan dampak kecepatan sebelumnya. Nilai saat ini (t+1) dan nilai saat ini (t) diperbarui, posisi partikel sebelumnya, masing-masing pada persamaan 2.8 dan 2.9 (Rajamohana dan Umamaheswari, 2018).

2.2.10 *Cross validation*

Cross validation merupakan metode yang paling populer untuk pemilihan parameter model dan *tuning* dalam statistik dan pembelajaran mesin karena kesederhanaan konseptual dan penerapannya yang luas. Ide dasar dari *cross validation* adalah menyesuaikan dan mengevaluasi setiap model kandidat pada kumpulan data yang terpisah sehingga evaluasi kinerja tidak bias (Lei, 2017). *Cross validation* membagi data menjadi set pelatihan dan pengujian untuk memperkirakan akurasi generalisasi pengklasifikasi untuk kumpulan data tertentu. Metode ini telah diperluas dengan berbagai teknik untuk menggabungkan pemilihan fitur dan penyetelan parameter. Beberapa metode *cross validation* yang biasa digunakan oleh peneliti yaitu *k-fold cross validation, cross validation leave-one-out, repeated double cross validation,* dan *nested cross validation* (Parvandeh dkk, 2020).

Dari beberapa metode *cross validation* yang biasa digunakan dalam penelitian *machine learning*, *k-fold cross validation* adalah yang paling banyak digunakan. Pada *k-fold cross validation*, iterasi diterapkan pada kumpulan data sebanyak k kali. Pada setiap putaran, data dibagi menjadi k bagian. Satu bagian digunakan untuk validasi, dan sisa k-1 bagian digabungkan menjadi subset pelatihan untuk evaluasi model seperti yang ditunjukkan pada Gambar 2.2 yang mengilustrasikan proses 5-*fold cross validation* (Raschka, 2018).



Gambar 2.2 Ilustrasi prosedur dari *k-fold cross validation*

2.2.11 *Confusion Matrix*

Dalam *machine learning*, *confusion matrix* merupakan alat evaluasi visual. Pada *confusion matrix* terdapat kolom dan baris yang dimana pada kolom mewakili hasil kelas prediksi dan pada baris yang menunjukkan hasil kelas yang sebenarnya. *Confusion matrix* dapat menyebutkan semua kemungkinan kasus pada masalah klasifikasi (Xu dkk, 2020). Dalam kasus masalah klasifikasi biner dimana *confusion matrix* memiliki dimensi 2x2 yang salah satu labelnya dianggap "positif" dan label lainnya dianggap "negatif" seperti yang dapat dilihat pada Gambar 2.3. Elemen matriks dikarakterisasi berdasarkan label prediksi (positif,negatif) dan hasil perbandingan prediksi dengan label kelas sebenarnya (*true*,*false*) yaitu *true positives* (TP), *true negatives* (TN), *false positives* (FP), dan *false negatives* (FN). Dalam konteks ini, *confusion matrix* menunjukkan bagaimana model klasifikasi "*confused*" ketika membuat prediksi dan *confusion matrix* dapat menunjukkan kesalahan yang dibuat oleh algoritma klasifikasi. *Confusion matrix* dapat diterapkan pada sekumpulan metrik kinerja dari masalah klasifikasi untuk menilai suatu algoritma atau membandingkan performa algoritma yang berbeda (Markoulidakis dkk, 2021).

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Gambar 2.3 *Confusion matrix*

Beberapa ukuran kinerja yang banyak digunakan untuk mengevaluasi algoritma klasifikasi yang diterapkan pada *confusion matrix* yaitu akurasi, presisi, *recall*, dan F1-*score*. Berikut merupakan persamaan dari keempat metrik kinerja yang dapat dilihat masing-masing pada persamaan 2.10, 2.11, 2.12, dan 2.13 (Alsubari dkk, 2021).

$$\text{Akurasi} = \frac{TP+TN}{FP+FN+TP+TN} \times 100 \quad (2.10)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \times 100 \quad (2.11)$$

$$\textit{Recall} = \frac{TP}{TP+FN} \times 100 \quad (2.12)$$

$$\text{F1-}\textit{score} = 2 * \frac{precision \times sensitivity}{precision+sensitivity} \times 100 \quad (2.13)$$