BAB 2
LANDASAN TEORI

2.1 *Tinder*

Tinder merupakan sebuah aplikasi kencan *online* atau bisa juga disebut sebagai aplikasi pencarian jodoh[2]. Aplikasi perjodohan ini pertama kali diluncurkan pada tahun 2012 di Los Angeles, California, Amerika Serikat oleh sekelompok orang yang bernama Justin Mateen, Sean Rad, Jonathan Badeen, dan beberapa teman lainnya[8]. Pada tahun 2018 Aplikasi *Tinder* secara resmi diluncurkan di Indonesia[3]. Menurut data.tempo.co aplikasi *Tinder* menjadi aplikasi kencan *online* yang banyak digemari di Indonesia dan menduduki *survey* tertinggi yaitu sebanyak 57,6% responden sebagai penggunanya dibanding dengan aplikasi atau media sosial kencan *online* lainnya[2].

Cara menggunakan aplikasi *Tinder* yaitu dimana seseorang akan memasang foto terbaiknya untuk menarik perhatian lawan jenis, sehingga pengguna yang merasa tertarik akan menggeser ke kanan agar dapat *match* dengan orang tersebut[9]. Istilah *match* ini dipergunakan apabila penggunanya dengan pengguna lain sama-sama saling menyukai. Barulah setelahnya kedua pengguna tersebut dapat berkomunikasi dan saling berkirim pesan di aplikasi *Tinder*[2]. Pada aplikasi *Tinder* terdapat 2 jenis yang bisa digunakan oleh penggunanya, yaitu *Tinder* yang tidak berbayar (akun *standard*) dan *Tinder* yang berbayar (akun *premium*). Pada aplikasi *Tinder* yang tidak berbayar, para pengguna hanya bisa menggunakan lokasi sesuai dengan GPS pada telepon selulernya, sehingga orang yang dapat ditemui di aplikasi *Tinder* hanya orang yang berada di sekitarnya. Sedangkan pada aplikasi *Tinder* yang berbayar pengguna dapat menggunakan fasilitas yang lebih optimal. Pengguna dapat melihat siapa saja lawan jenis yang menyukai profilnya, dapat menyukai profil

lawan jenisnya tanpa batas, mendapatkan profil-profil unggulan dari *Tinder*, menghilangkan iklan, dapat mengembalikan/*rewind* profil yang sudah terlewat, dan dapat merubah lokasi sesuai keinginannya[3].

2.2 Google Play

Salah satu tempat untuk mengunduh ratusan ribu aplikasi *android* adalah *Google Play Store*. *Google Play Store* adalah pasar *platform android* yang penting untuk pendistribusian aplikasi *mobile*. *Google Play Store* memungkinkan pengguna untuk mengunduh dan menggunakan aplikasi-aplikasi pihak ketiga secara bebas. Pada *Google Play Store*, aplikasi-aplikasi *android* dibagi menjadi kategori-kategori yang unik. Dengan adanya kategori-kategori tersebut, pengguna bisa dengan mudah mencari aplikasi yang dibutuhkannya[10].

Aksesibilitas *Google Play Store* yang dapat diakses di seluruh dunia membuat pengembang aplikasi berlomba–lomba membuat beragam aplikasi yang dapat diunduh pengguna. Dalam pengembangan aplikasi, pengembang perlu memprediksi aplikasi di pasar secara akurat yang sangat penting dalam menunjukkan penilaian pengguna yang memengaruhi keberhasilan suatu aplikasi. *Rating* diberikan oleh pengguna untuk menilai apakah aplikasi tersebut bagus atau tidak. Semakin tinggi *rating* yang diberikan oleh pengguna, berarti pengguna menyukai aplikasi tersebut dan dapat menjadi tolak ukur bagi pengguna lain untuk *mendownload* aplikasi tersebut[11]. Pada penelitian ini akan diambil data ulasan para pengguna mengenai aplikasi *Tinder*.

2.3 *Web Scraping*

Web scraping adalah proses pengambilan sebuah dokumen semi-terstruktur dari internet, umumnya berupa halaman-halaman *web* dalam bahasa *markup* seperti HTML atau XHTML dan menganalisis dokumen

tersebut untuk diambil data tertentu yang dibutuhkan untuk kepentingan. *Web scraping* hanya fokus pada cara memperoleh data dan mengekstrak data dalam ukuran yang bervariasi[12].

2.4 Analisis Sentimen

Analisis sentimen adalah salah satu bagian dari *text mining*. Analisis sentimen adalah sub-bagian dari *Natural Language Processing* (NLP) yang berfokus pada penentuan *text's feelings*. Analisis sentimen dikenal dengan istilah *opinion mining*, yaitu proses memahami, mengekstrak, dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen dalam sebuah kalimat opini[13].

Analisis sentimen merupakan sebuah proses untuk menganalisis atau mengidentifikasi sebuah opini seseorang yang menunjukkan sikap terhadap suatu topik atau produk tertentu masuk ke dalam kategori positif, negatif, atau netral. Opini sentimen memiliki karakteristik yang menunjukkan bahwa hal tersebut bersifat subjektif[14]. Pemeriksaan opini diperlukan agar dapat melakukan peringkasan terhadap suatu pendapat.

Analisis sentimen terbagi menjadi tiga jenis, yang terdiri atas *document level sentiment analysis, sentence level sentiment analysis*, dan *aspect-level sentiment analysis*[15]. *Document level sentiment analysis* merupakan bidang analisis sentimen dengan masukan dokumen pembahasan tentang satu tema yang dianggap sebagai satu unit masukan. *Sentence level sentiment analysis* disisi lain merupakan bidang analisis sentimen dengan masukan berupa satu kalimat sebagai satu unit masukan. Level yang lebih detail dari analisis sentimen ini adalah *aspect-level sentiment analysis* dengan melakukan klasifikasi sentimen atas aspek tertentu dari entitas yang dibahas, seperti sentimen atas spesifikasi dari produk yang lebih detail dari sekedar sentimen atas produk[16].

2.4.1 Analisis Sentimen Berbasis Aspek

Analisis sentimen berbasis aspek atau *Aspect Based Sentiment Analysis* (ABSA) adalah salah satu perkembangan dari analisis sentimen yang mengacu pada sebuah kalimat. Ada dua tugas utama di dalam analisis sentimen berbasis aspek yaitu aspek ekstraksi dan klasifikasi aspek. Dalam ekstraksi aspek, sentimen ditentukan oleh jenis aspek yang dibahas. Sedangkan pada aspek sentimen klasifikasi, sentimen diklasifikasi sebagai positif, negatif, atau netral untuk aspek itu. Analisis sentimen berbasis aspek mampu mengungkap aspek atau atribut produk mana yang diekspresikan oleh orang tersebut[7]. Untuk mencapai tujuan analisis sentimen berbasis aspek terdapat dua tugas utama yaitu *Aspect exraction* dan *Aspect sentiment classification*.

1. *Aspect extraction*

   Pada tahap ini akan mengidentifikasi aspek yang sudah ditentukan sebelumnya. Contohnya adalah untuk kalimat “langganan aplikasi ini murah banget”. Dari kalimat tersebut dapat mengekstrak kata “murah” menjadi entitas aspek yang diwakilkan oleh aplikasi yaitu aspek “harga”.

2. *Aspect sentiment classification*

   Pada tahap ini akan menentukan polaritas sentimen kepada aspek yang telah diekstraksi dengan nilai polaritas “positif”, “negatif”, atau “netral”. Tujuan dari tahap ini adalah untuk mengidentifikasi nilai sentimen dari aspek yang telah diekstraksi sebelumnya. Pada contoh kalimat “fiturnya canggih aku bisa ngobrol sama orang-orang yang jauh dari tempatku”. Setimen dari aspek “fitur aplikasi” adalah positif.

2.5 Klasifikasi Multikelas dan Multilabel

Dalam klasifikasi, tujuannya adalah untuk memprediksi label kelas yang merupakan pilihan dari daftar kemungkinan yang telah ditentukan sebelumnya. Contohnya adalah mengklasifikasikan iris menjadi salah satu dari tiga kemungkinan spesies. Klasifikasi kadang-kadang dipisahkan menjadi klasifikasi biner, yang merupakan kasus khusus untuk membedakan antara tepat dua kelas, dan klasifikasi *multiclass*, yaitu klasifikasi antara lebih dari dua kelas. Klasifikasi biner mencoba menjawab pertanyaan ya/tidak. Mengklasifikasikan email sebagai *spam* atau bukan *spam* adalah contoh masalah klasifikasi biner. Dalam tugas klasifikasi biner ini, pertanyaan ya/tidak yang diajukan adalah “Apakah ini email spam?”[17].

2.5.1 Klasifikasi Multikelas

Dalam bidang yang luas dari pembelajaran mesin, fokus utamanya adalah memprediksi suatu hasil menggunakan data yang tersedia. Tugas prediksi ini disebut "masalah klasifikasi" ketika hasilnya mewakili berbagai kelas, sedangkan disebut "masalah regresi" ketika hasilnya adalah pengukuran numerik. Terkait dengan klasifikasi, pengaturan yang paling umum melibatkan hanya dua kelas, meskipun bisa ada lebih dari dua. Tugas klasifikasi dalam pembelajaran mesin yang melibatkan lebih dari dua kelas dikenal dengan istilah "klasifikasi multi-kelas"[18].

Dari sudut pandang algoritmik, tugas prediksi diatasi menggunakan teknik matematika mutakhir. Ada banyak solusi berbeda, namun masing-masing memiliki faktor umum: mereka menggunakan data yang tersedia (variabel X) untuk memperoleh prediksi terbaik $\hat{Y}$ dari variabel hasil Y. Dalam klasifikasi multi-kelas, kita dapat menganggap variabel respon Y dan prediksi $\hat{Y}$ sebagai dua variabel acak diskret: mereka

mengambil nilai dalam {1, ⋯, K} dan setiap angka mewakili kelas yang berbeda[18].

Algoritma menghasilkan probabilitas bahwa suatu unit tertentu termasuk dalam satu kelas yang mungkin, kemudian aturan klasifikasi digunakan untuk menetapkan satu kelas pada setiap individu. Aturannya umumnya sangat sederhana, aturan yang paling umum adalah menetapkan unit ke kelas dengan probabilitas tertinggi[18].

Model klasifikasi memberikan kita probabilitas keanggotaan dalam suatu kelas tertentu untuk setiap unit yang mungkin. Berdasarkan probabilitas yang diberikan oleh model, dalam masalah klasifikasi dua kelas, biasanya diterapkan ambang batas untuk memutuskan kelas mana yang harus diprediksi untuk setiap unit. Sementara dalam kasus multi-kelas, ada berbagai kemungkinan; di antaranya, nilai probabilitas tertinggi dan *softmax* adalah teknik yang paling sering digunakan[18].

2.5.2 Klasifikasi Multilabel

Klasifikasi *single-label* berkaitan dengan pembelajaran dari sekumpulan contoh yang dikaitkan dengan satu label l dari sekumpulan label L yang terpisah, di mana |L| > 1. Jika |L| = 2, maka masalah pembelajaran ini disebut masalah klasifikasi biner, sedangkan jika |L| > 2, maka disebut masalah klasifikasi multi-kelas[19].

Dalam klasifikasi multi-label, contoh-contoh dikaitkan dengan sekumpulan label Y ⊆ L[19]. Klasifikasi multi-label adalah salah satu tugas paling menantang dalam klasifikasi karena tidak seperti kasus multi-kelas di mana sampel hanya termasuk dalam satu kelas, keluaran multi-label dapat menunjukkan lebih dari satu label secara bersamaan[20]. Ketika mempertimbangkan klasifikasi multi-label, perlu dicatat bahwa ini adalah generalisasi dari klasifikasi multi-kelas.

Tidak ada batasan jumlah kelas yang dapat diberikan pada suatu *instance* dalam masalah multi-label[21].

Pada masa lalu, klasifikasi multi-label terutama dimotivasi oleh tugas-tugas pengkategorian teks dan diagnosis medis. Dokumen teks biasanya termasuk dalam lebih dari satu kelas konseptual. Misalnya, sebuah artikel surat kabar yang membahas reaksi gereja kristen terhadap perilisan film Da Vinci Code dapat diklasifikasikan ke dalam kedua kategori *Society/Religion* dan *Arts/Movies*. Demikian pula dalam diagnosis medis, seorang pasien dapat menderita diabetes dan kanker prostat secara bersamaan[19].

2.6 *Text Mining*

Text mining adalah proses menemukan informasi dalam kumpulan teks besar dan mengidentifikasi pola dan hubungan yang terdapat dalam data tekstual. *Text mining* merupakan ilmu yang bersifat interdisiplin, yang di mana memerlukan pengetahuan di bidang *data mining*, *natural language processing*, *machine learning*, dan pencarian informasi. *Text mining* sangat erat kaitannya dengan *data mining* karena sama-sama mencari hubungan yang menarik pada suatu data, namun *text mining* cenderung lebih sulit dilakukan karena sumber data yang digunakan bukan angka melainkan tulisan atau teks[7].

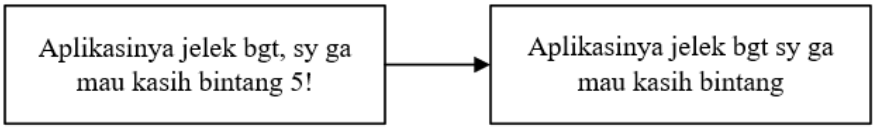
2.7 *Text Preprocessing*

Dalam proses ekstraksi informasi dengan sumber informasi yang tidak terstruktur memberikan kesulitan dalam proses komputerisasi secara otomatis, maka dari itu dibutuhkannya suatu proses yang akan merubah dokumen yang memiliki format tidak terstruktur menjadi terstruktur. Proses ini biasa disebut *Text Preprocessing*[22]. Dalam prosesnya mengubah dokumen tidak terstruktur menjadi terstruktur dengan memberikan data sebuah nilai-nilai

numerik. Setelah data menjadi terstruktur dan memiliki nilai-nilai numerik maka dapat diolah lebih lanjut. Ada beberapa tahap *preprocessing* yang dilakukan pada penelitian ini, yaitu:

2.7.1 *Cleaning*

Proses pembersihan data bertujuan untuk menemukan dan memperbaiki atau menghapus data yang tidak valid atau tidak berguna dari suatu kumpulan data. *Cleaning* adalah proses dimana sistem akan menghilangkan seluruh angka, tanda baca ataupun simbol yang ada dalam kalimat seperti “?”, “@”, “,”, dst.[23]. Setelah melalui proses *cleaning* data, data akan siap untuk digunakan dalam tahap selanjutnya dari data *preprocessing*, yaitu pengolahan dan pengintegrasian data. Namun dalam beberapa kasus, *cleaning* data dapat dilakukan secara terus menerus sesuai dengan kebutuhan analisis atau pemodelan yang dilakukan. Proses *cleaning* dapat dilihat pada Gambar 2.1.

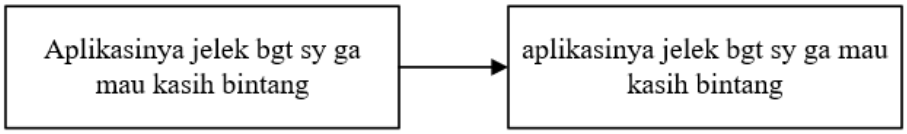


Gambar 2.1 Ilustrasi Proses *Cleaning*

2.7.2 *Case Folding*

Case folding adalah proses pada *text preprocessing* yang dilakukan untuk mengubah semua huruf pada teks ke dalam huruf kecil atau huruf besar yang sama, sehingga memudahkan dalam proses pembandingan dan analisis teks[24]. Proses *case folding* juga dapat digunakan untuk mengatasi masalah penulisan yang tidak konsisten dalam data, seperti penggunaan huruf besar dan kecil yang tidak sesuai standar atau ada kesalahan ejaan. Dengan mengubah semua karakter menjadi huruf kecil, konsistensi dalam data akan diperbaiki sehingga lebih mudah

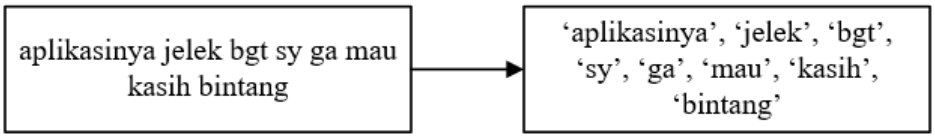
untuk diproses dan dianalisis. Proses *case folding* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Ilustrasi Gambar *Case Folding*

2.7.3 *Tokenizing*

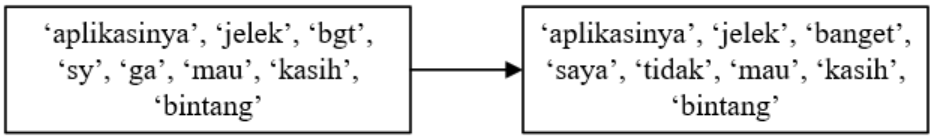
Tokenizing merupakan suatu tahapan untuk memecah suatu kumpulan teks menjadi sebuah kata. Analisis kata-kata dalam teks menjadi penting karena dengan melakukan tokenisasi, makna dari teks dapat dengan mudah ditentukan [16]. Dalam pemrosesan bahasa alami, *tokenizing* sering digunakan untuk memecah teks menjadi unit-unit yang lebih kecil dan mudah diolah oleh mesin. Beberapa algoritma *tokenizing* yang umum digunakan adalah pemotongan karakter, pemotongan kata, dan pemotongan frase. Proses *tokenizing* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi Proses *Tokenizing*

2.7.4 *Normalization*

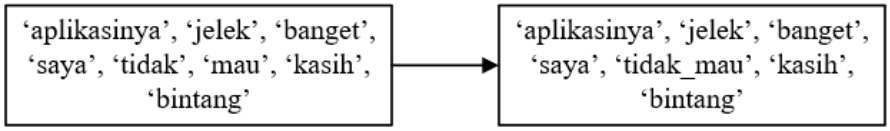
Normalization adalah proses perbaikan kata-kata yang salah eja atau disingkat dalam bentuk tertentu. Tahap ini bertujuan untuk memperkecil dimensi kata yang memiliki arti yang sama tetapi memiliki ejaan yang salah atau disingkat dalam bentuk tertentu[22]. Proses *normalization* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi Proses *Normalization*

2.7.5 *Convert Negation*

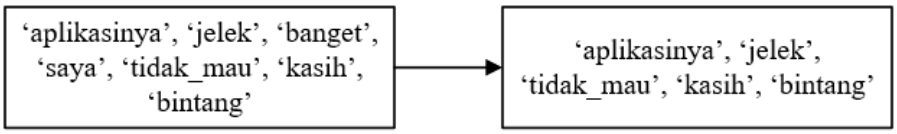
Convert negation merupakan proses mengubah kata negasi dan menggabungkannya dengan kata setelahnya agar menjadi satu kesatuan kata, misalnya “tidak” dan “mau” menjadi “tidak_mau”[25]. Untuk melakukan proses *convert negation*, tentunya perlu digunakan daftar kata negasi untuk dapat mengidentifikasi kata negasi pada data komentar. Terdapat enam kata penanda negasi lazim yang digunakan dalam bahasa Indonesia yaitu tidak, bukan, jangan, tanpa, belum dan kurang[26]. Proses *convert negation* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Ilustrasi Proses *Convert Negation*

2.7.6 *Stopword Removal*

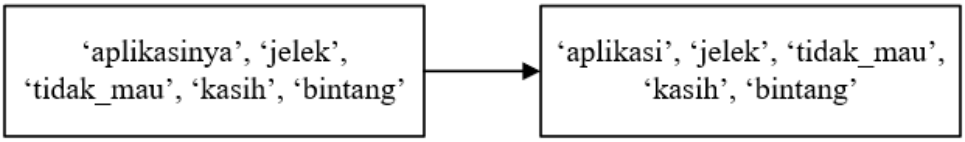
Kata yang tidak memiliki makna atau tidak efektif akan dihilangkan dengan menggunakan *stopwords*[22]. *Stopwords* biasanya diabaikan dalam proses pemrosesan teks karena dianggap tidak berguna dan hanya menambah ukuran file tanpa memberikan informasi yang bermanfaat. Namun, ada juga kasus di mana *stopwords* perlu dipertahankan karena memiliki makna penting dalam konteks tertentu. Proses *stopword removal* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Ilustrasi Proses *Stopword Removal*

2.7.7 *Stemming*

Stemming merupakan proses melakukan reduksi kata atau token ke dasar bentuk kata[27]. Tujuannya adalah untuk mengurangi kata-kata yang berbeda menjadi bentuk kata dasar yang sama, sehingga lebih mudah untuk mengelompokkan atau menganalisis kata-kata tersebut. Proses ini sering digunakan dalam *Natural Language Processing* (NLP) dan *Information Retrieval* (IR). Proses *stemming* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Ilustrasi Proses *Stemming*

2.8 Pembobotan TF-IDF

TF-IDF (*term frequency-inverse document frequency*) adalah metode yang digunakan untuk mengevaluasi pentingnya sebuah kata dalam suatu dokumen terhadap koleksi dokumen lain. Fitur TF-IDF untuk optimasi dalam analisis sentimen. Dengan menggabungkan ekstraksi fitur TF-IDF dan algoritma *stochastic gradient descent*, analisis sentimen dapat mengklasifikasikan teks dalam bahasa Indonesia dengan tepat menurut sentimen positif dan negatif[27]. Metode TF-IDF memperhitungkan dua faktor penting:

a. *Term Frequency* (TF):

Term Frequency menyatakan nilai frekuensi *term* yang sering muncul pada sebuah dokumen. Semakin besar jumlah kemunculan suatu *term*

dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. $f(t_d, d_t)$ mendefinisikan jumlah kemunculan *term d* pada sebuah *t*.

b. *Inverse Document Frequency* (IDF):

Mengukur seberapa penting suatu kata dalam konteks koleksi dokumen yang lebih besar. Kata-kata yang muncul lebih jarang di seluruh koleksi dokumen cenderung memiliki IDF yang lebih tinggi. IDF dihitung dengan membagi jumlah total dokumen dalam koleksi dengan jumlah dokumen yang mengandung kata tersebut. Hasilnya kemudian diambil logaritma untuk memperhalus skala.

Dalam metode TF-IDF, nilai TF dan IDF dikalikan bersama-sama untuk menghasilkan bobot kata (*term weight*) untuk setiap kata dalam sebuah dokumen. Bobot ini mencerminkan tingkat pentingnya kata dalam dokumen tersebut dibandingkan dengan koleksi dokumen yang lebih besar[28]. Berikut merupakan rumus TF-IDF.

$$\mathrm{TF}(t_d, d_t) = \mathrm{f}(t_d, d_t) \quad (2.1)$$

$$IDF_t = \log 10\ (\frac{D}{df_t}) \quad (2.2)$$

$$W_{d,t} = TF_{d,t} \times IDF_{d,t} \quad (2.3)$$

Keterangan:

TF = banyaknya kata yang dicari pada sebuah dokumen

$\mathrm{f}(t_d, d_t)$ = jumlah kemunculan banyaknya term pada dokumen sama

D = total dokumen

df_t = jumlah dokumen yang mengandung term t

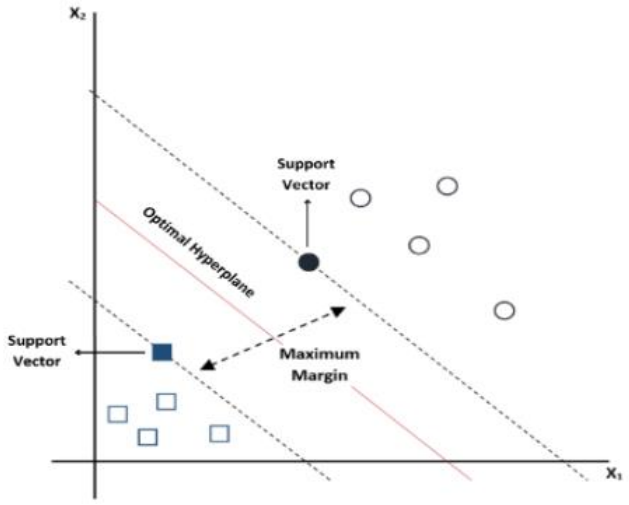
IDF = *Inversed Dokument Frequency*

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

2.9 *Support Vector Machine*

Metode *Support Vector Machine* (SVM) merupakan sistem pembelajaran yang menggunakan ruang hipotesis yang berupa fungsi-fungsi linear di dalam sebuah fitur yang memiliki dimensi tinggi dan dilatih dengan menggunakan algoritma pembelajaran berdasarkan teori optimasi. *Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, dan Vapnik pertama kali pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. SVM memecahkan masalah klasifikasi dengan mencari fungsi pemisah (*hyperplane*) margin maksimum yang dapat memisahkan dua kelas secara optimal. *Hyperplane* dengan margin yang lebih besar lebih akurat dalam mengklasifikasikan data dibandingkan yang lebih kecil, hal ini dikenal dengan istilah *Maximum Marginal Hyperplane*[29]. Konsep *hyperplane* disajikan pada Gambar 2.8.



Gambar 2.8 Konsep *Hyperplane* pada SVM

Margin adalah dua kali jarak antara *hyperplane* dan *support vector*, dimana *support vector* adalah titik yang berada paling dekat dengan *hyperplane*. *Support vector* juga dapat dikatakan objek-objek data terluar yang

paling dekat dengan *hyperplane*. *Support vector* inilah yang nantinya akan diperhitungkan oleh SVM. Berdasarkan dari karakteristiknya, metode SVM dibagi menjadi dua, yaitu SVM Linier dan SVM Non-Linier. SVM linier merupakan data yang dipisahkan secara linier, yaitu memisahkan kedua *class* pada *hyperplane* dengan *soft margin*. Sedangkan SVM Non-Linier yaitu menerapkan fungsi dari *kernel trick* terhadap ruang berdimensi tinggi[30].

Support Vector Machine (SVM) memiliki beberapa tahap dalam pengerjaannya, pada tahap awal yaitu pendefinisan persamaan suatu *hyperplane* pemisah. *Hyperplane* adalah sebuah garis lurus atau bidang mendatar yang memisahkan kelas-kelas.

Persamaan (2.4) *Hyperplane*

$$f(x) = \vec{w} \,.\, \vec{x} + b \tag{2.4}$$

dimana w merupakan suatu bobot vektor, yaitu $\{w_1, w_2, \ldots, w_n\}$ n adalah jumlah atribut dan b merupakan suatu skalar yang disebut dengan bias. Jika berdasarkan pada atribut A1, A2 dengan permisalan tupel pelatihan X = (x_1, x_2) , x1 dan x2 merupakan nilai dari atribut A1 dan A2.

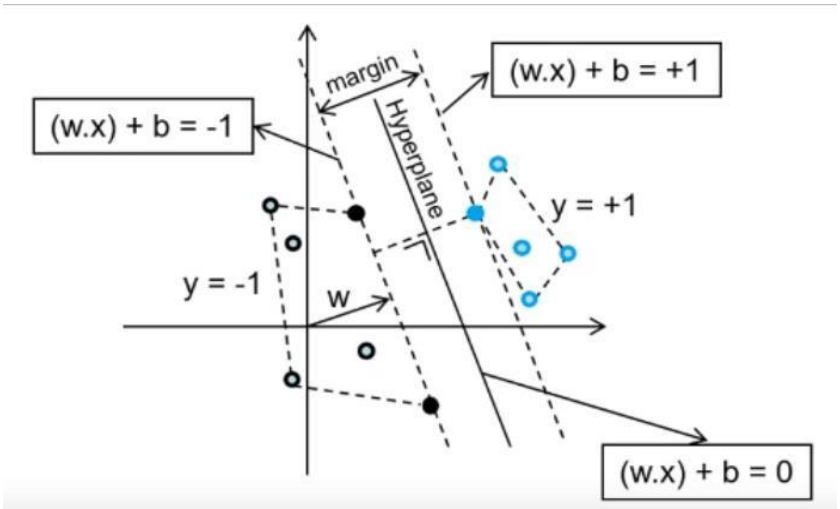
Sehingga diperoleh persamaan -1 (sample negatif) memenuhi pertidaksamaan (2.5)

$$\vec{w} \,.\, \vec{x} + b \leq -1 \tag{2.5}$$

Dan kelas +1 *pattern* yang memenuhi pertidaksamaan (2.6) :

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

Pada Gambar 2.9 merupakan ilustrasi *Hyperplane*.



Gambar 2.9 Ilustrasi *Hyperplane*

Dengan mengetahui garis normal dan posisi terhadap bidang yang relatif terhadap titik koordinat maka kita akan menentukan nilai jarak terdekat terhadap margin. Hal tersebut menjadikan sebagai pembatas atau *support* untuk menentukan perbedaan terhadap nilai sentimen -1 atau negatif, 0 atau netral dan 1 atau positif[29]. Nilai jarak margin dapat dioptimalkan dengan menggunakan *hyperplane* dan titik terdekatnya dengan $1/|w|$. Untuk mendapatkan titik tersebut perlu adanya persamaan *Quadratic Programming* (QP) *Problem*, dengan mencari persamaan terhadap nilai titik minimum pada persamaan 2.7 dan mendapatkan nilai *constraint* dengan persamaan 2.8.

$$\min\ r(w) = \frac{1}{2}||w||^2 \tag{2.7}$$

$$y_i\ (w * x_i + b) - 1 \geq 0, (i = 1, \dots, n) \tag{2.8}$$

Permasalahan ini ini dapat dipecahkan dengan berbagi teknik komputasi. Lebih mudah diselesaikan dengan mengubah persamaan (2.7) ke dalam fungsi *Lagrangian* pada persamaan (2.9) berikut:

$$Lp = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i y_i\ (xi\ .\ w^{\mathrm{r}} + b)\ \text{-}1 \tag{2.9}$$

α_i adalah *Lagrange Multiplier* yang berkorespondensi dengan x_i. Nilai α_i adalah nol atau positif. Untuk meminimalkan *Lagrangian*, Persamaan (2.9) harus diturunkan terhadap w dan b, dan diset dengan nilai nol untuk syarat optimasi di atas:

Syarat 1 :

$$\frac{\partial \mathrm{Lp}}{\partial \mathrm{w}} = 0 \Rightarrow \mathrm{w} = \sum_{i=1}^{n} a_i y_i x_i \tag{2.10}$$

Syarat 2 :

$$\frac{\partial \mathrm{Lp}}{\partial \mathrm{w}} = 0 \Rightarrow \mathrm{w} = \sum_{i=1}^{n} a_i y_i = 0 \tag{2.11}$$

N adalah jumlah data yang menjadi *support vector*.

Karena *Lagrange Multiplier* (α) tidak diketahui nilainya, persamaan di atas tidak dapat diselesaikan secara langsung untuk mendapatkan w dan b. Untuk menyelesaikan masalah tersebut, modifikasi Persamaan 2.9 di atas menjadi kasus memaksimalkan dengan syarat optimal untuk dualitasnya menggunakan konstrain Karush-Kuhn-Tucker (KKT) sebagai berikut[30]:
Syarart 1 :

$$\alpha_i[y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1] = 0 \tag{2.12}$$

Syarat 2 :

$$\alpha_i > 0, i = 1,2, \dots, N \tag{2.13}$$

Dengan menerapkan konstrain pada Persamaan (2.12) dan (2.13) maka dipastikan bahwa nilai *Lagrange Multiplier* sama banyaknya dengan data latih, meskipun sebenarnya banyak dari data latih yang *Lagrange Multiplier* sama dengan nol (karena hanya beberapa saja yang akan menjadi *support vector*) ketika menerapkan syarat pertama. Konstrain di atas menyatakan bahwa *Lagrange Multiplier* α_i harus nol kecuali untuk data latih x_i yang

memenuhi persamaan:

$$y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) = 0 \quad (2.14)$$

Data latih tersebut, dengan $\alpha_i > 0$, terletak pada *hyperplane* b_{i1} atau b_{i2}, dan disebut *support vector*. Data latih yang tidak terletak di *hyperplane* tersebut mempunyai $\alpha_i = 0$,. Persamaan 2.16 dan 2.17 juga menyarankan parameter *w* dan *b* yang mendefinisikan *hyperplane* hanya tergantung *support vector*.

Masalah optimasi di atas masih sulit diselesaikan karena banyaknya parameter (*w*, *b* dan α_i). Untuk menyederhanakannya, persamaan optimasi 2.9 diatas harus ditransformasi ke dalam fungsi *Lagrange Multiplier* itu sendiri (disebut dualitas masalah).

Persamaan *Lagrange Multiplier* 2.15 dapat dijabarkan menjadi:

$$Lp = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} a_i y_i(\boldsymbol{w}.\boldsymbol{x_i}) - \text{b}\sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i \quad (2.15)$$

Syarat optimal (2.11) ada dalam suku ketiga di ruas kanan dalam persamaan (2.15), dan memaksa suku ini menjadi sama dengan nol. Dengan mengganti *w* dari syarat (2.10), dan suku $\|w\|^2 = \boldsymbol{w_i}.\boldsymbol{w_j}$, maka persamaan di atas akan berubah menjadi dualitas *Lagrange Multiplier* berupa Ld dan didapatkan:

Maksimalkan:

$$L\text{d} = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j}^{n} a_i a_j y_i y_j \boldsymbol{x_i}.\boldsymbol{x_j} \quad (2.16)$$

$\boldsymbol{x_i}.\boldsymbol{x_j}$ merupakan *dot-product* dua data dalam data latih.

Syarat 1 :

$$\sum_{i=1}^{n} a_i y_i = 0 \quad (2.17)$$

Syarat 2 :

$$a_i > 0, \text{ i} = 1,2, \dots, N \quad (2.18)$$

Untuk set data yang besar, masalah dualitas optimasi tersebut (2.16, 2.17, 2.18) dapat diselesaikan dengan metode numerik seperti *Quadratic Programming*. Sekali α_i didapatkan, persamaan (2.10) dan (2.11) bisa digunakan untuk mendapatkan solusi layak untuk *w* dan *b*.

Hyperplane (batas keputusan) didapatkan dengan formula:

$$(\sum_{i=1}^{n} a_i y_i \boldsymbol{x_i}.\boldsymbol{z}) + \mathrm{b} = 0 \qquad (2.19)$$

N adalah jumlah data yang menjadi *support vector*, x_i merupakan *support vector, z* merupakan data uji yang akan diprediksi kelasnya, dan $\boldsymbol{x_i}.\boldsymbol{z}$ merupakan *inner-product* antara x_i dan *z*. Untuk nilai *b* didapatkan dari persamaan (2.12) pada *support vector*. Karena α_i dihitung dengan teknik metode numerik dan mempunyai *error* numerik, nilai yang dihitung untuk *b* bisa jadi tidak sama. Hal ini disebabkan oleh *support vector* yang digunakan dalam persamaan (2.12), biasanya diambil nilai rata-rata dari *b* yang didapat untuk menjadi parameter *hyperplane*. Untuk persamaan (2.12) dalam mendapat dapat *b* dapat disederhanakan menjadi:

$$b_i = 1 - y_i(\boldsymbol{w}.\boldsymbol{x_i}) \qquad (2.20)$$

Penjelasan di atas berdasarkan asumsi bahwa kedua kelas dapat terpisah secara sempurna oleh *hyperplane*. Akan tetapi, pada umumnya kedua kelas tersebut tidak dapat terpisah secara sempurna. Hal ini menyebabkan proses optimalisasi tidak dapat diselesaikan karena tidak ada *w* dan *b* yang memenuhi pertidaksamaan 2.8. Untuk itu pertidaksamaan tersebut dimodifikasi dengan memasukan variabel slack ξ_i ($\xi_i \geq 0$), Menjadi :

$$y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) \geq 1 - \xi_i \qquad (2.21)$$

Demikian juga untuk masalah persamaan(2.7):

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i \quad (2.22)$$

Parameter *C* berguna untuk mengontrol *trade-off* antara margin dan *error* klasifikasi. Semakin besar nilai *C* maka semakin besar pula pelanggaran yang dikenakan untuk tiap klasifikasi[31].

Metode untuk mengoptimalisasi *hyperplane* SVM umumnya dipakai untuk menyelesaikan *Quadratic Programming* dengan konstrain yang ditetapkan. Beberapa pilihan metode yang bisa digunakan adalah *chunking*, metode dekomposisi, dan *Sequential Minimal Optimization* (SMO).

SVM sebenarnya adalah *hyperplane* linear yang hanya bekerja pada data yang dapat dipisahkan secara linear. Untuk data yang distribusi kelasnya tidak linear biasanya digunakan pendekatan kernel pada fitur data dari awal set data. Kernel dapat di definisikan sebagai suatu fungsi yang memetakan fitur data dari dimensi awal (rendah) ke fitur lain yang berdimensi lain yang lebih tinggi (*feature space*)[30].

Berikut ini adalah beberapa fungsi kernel yang umum digunakan yaitu:

a) Kernel linier

   $K(x_i, x) = x_i^T, x$

b) Polynomial

   $K(x_i, x) = (\gamma. x_i^T + r)^p, \gamma > 0$

c) Radial basis function (RBF)

   $K(x_i, x) = \exp(-y|x_i - x|^2), \gamma > 0$

d) Sigmoid kernel

   $K(x_i, x) = \tanh(\gamma. x_i^T + r)$

Keterangan:

x adalah pasangan dua data dari semua bagian data latih. Parameter $\gamma > 0$, merupakan konstanta. $|x_i - x|^2$ merupakan kuadrat jarak antara vektor x_i dan x.

Untuk fungsi kernel yang digunakan pada penelitian ini adalah fungsi kernel RBF, karena RBF biasanya digunakan untuk mengklasifikasi *multiclass* dengan kompleksitas yang tinggi serta memiliki kinerja yang paling baik dibandingkan kernel linier pada parameter tertentu maupun kernel polynomial dan dapat beroperasi dalam ruang berdimensi tinggi.

2.9.1 SVM *Multiclass* dengan "*One-Against-All*"

SVM saat pertama kali diperkenalkan oleh Vapnik hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). SVM hanya dapat melakukan klasifikasi biner (dua kelas), sementara masalah di dunia nyata umumnya mempunyai banyak kelas seperti pengenalan karakter, pengenalan wajah atau diagnosis pasien, dimana data masukan terbagi menjadi lebih dari dua kelas. Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas terus dilakukan. Ada 3 pendekatan SVM multikelas yaitu *one-against-all (OAA), one-against-one* (OAO), dan *error correcting output code* (ECOC)[31]. Tetapi yang akan dijelaskan pada penelitian ini adalah SVM multikelas *one-against-all* (OAA).

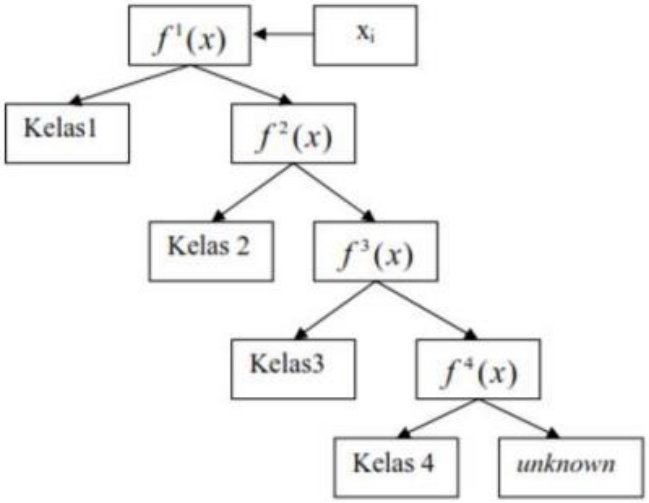
Dengan metode ini, dibangun k buah model SVM biner (k adalah jumlah kelas). Setiap model klasifikasi ke-i dilatih dengan menggunakan seluruh data, untuk mencari solusi permasalahan[31]. Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM biner dapat dilihat pada Tabel 2.1.

Tabel 2.1 Contoh 4 SVM Biner dengan Metode *one-against-all*

Yi = +1	**Yi = -1**	**Hipotesis**
Kelas 1	Bukan kelas 1	$f^1(x) = (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x) = (w^2)x + b^2$

Kelas 3	Bukan kelas 3	$f^3(x) = (w^3)x + b^3$
Kelas 4	Bukan kelas 4	$f^4(x) = (w^4)x + b^4$

Untuk ilustrasi dari metode *one-against-all* dapat dilihat pada Gambar 2.10.



Gambar 2.10 Klasifikasi dengan Metode *One-Against-All*

Penelitian lebih lanjut untuk megembangkan SVM sehingga dapat melakukan klasifikasi lebih dari dua kelas yaitu multikelas SVM. Dalam klasifikasi kasus multikelas SVM, *hyperplane* yang terbentuk adalah lebih dari satu, yang umum digunakan untuk mengimplementasikan multikelas SVM adalah pendekatan metode *one-against-all* (OAA).

Konsep pada OAA yaitu dimisalkan pada kasus empat kelas, kelas 1,2,3 dan 4. Bila akan diujikan ρ(1), semua data dalam kelas 1 diberi label +1 dan data dari kelas 2,3, dan 4 diberi label -1. Pada ρ(2), semua data dalam kelas 2 diberi label +1 dan data dari kelas 1,3, dan 4

diberi label -1. Pada ρ(3), semua data dalam kelas 3 diberi label +1 dan data dari kelas 1,2, dan 4 diberi label -1. Begitu juga untuk ρ(4), semua data dalam kelas 4 diberi label +1 dan data dari kelas 1,2, dan 3 diberi label -1. Lalu dicari *hyperplane* untuk masing-masing kelas di atas. Kemudian kelas dari suatu data baru x ditentukan berdasarkan nilai terbesar dari *hyperplane*.

2.10 *Confussion Matrix*

Pada bidang klasifikasi, ukuran akurasi dari suatu model klasifikasi merupakan hal yang penting untuk diperhatikan. Nilai akurasi dapat menggambarkan bagus tidaknya suatu model klasifikasi. Dalam penelitian ini dilakukan pengujian akurasi dengan teknik *cross-validation*, dimana dataset akan dibagi menjadi 2 bagian yaitu *training set* (data latih) dan *testing set* (data uji). *Training set* digunakan untuk melatih model, sedangkan *testing set* digunakan untuk mengevaluasi performa dari model. Teknik *cross-validation* dengan sejumlah perulangan (*epoch*) dilakukan untuk menghindari terjadinya *overfitting* dan *overlapping* pada data uji. Data uji kemudian diproses dalam pembuatan *confusion matrix*.

Confusion Matrix adalah sebuah matriks yang memuat data klasifikasi yang dilakukan oleh sistem klasifikasi baik secara aktual maupun prediktif. Dengan mengevaluasi data pada matriks akan diketahui bagaimana performa suatu model. *Support Vector Machine* (SVM) adalah algoritma klasifikasi yang digunakan untuk membuat garis pemisah (*hyperplane*) antara dua atau lebih kelas. Proses klasifikasi yang digunakan berdasarkan model pembelajaran dari SVM dan mengevaluasi dengan *confusion matrix* disajikan pada Tabel 2.2.

Tabel 2.2 *Confussion Matrix*

Komentar	**Prediksi Data**

	Negatif	**Positif**
Negatif	TN (*True Negative*)	FN (*False Negative*)
Positif	FP (*False Positive)*	TP (*True Positive*)

Keterangan:

True Negative (TN) : Mendefinisikan data negatif dan diprediksi negatif

False Positive (FP) : Mendefinisikan data positif dan diprediksi negatif

False Negative (FN) : Mendefinisikan data negatif dan diprediksi positif

True Positive (TP) : Mendefinisikan data positif dan diprediksi positif

Confuison matrix adalah tabel yang digunakan untuk mengevaluasi kinerja algoritme klasifikasi. Nilai *Accuracy*, *precision*, *recall*, dan *f-measure* dapat dihitung setelah mengetahui nilai-nilai dalam *confusion matriks*[32].

1. *Accuracy* mengukur seberapa baik hasil klasifikasi yang dilakukan oleh suatu model sesuai dengan nilai yang sebenarnya. Ini menunjukkan persentase dari klasifikasi yang benar yang dilakukan oleh model.

$$Accuracy = \frac{Jumlah\ aspek\ atau\ kategori\ yang\ benar\ dideteksi}{Jumlah\ aspek\ atau\ kategori\ yang\ beranotasi} \quad (2.23)$$

2. *Precision* mengukur seberapa baik model dalam mengidentifikasi sampel positif yang sebenarnya. Ini mengukur rasio dari sampel positif yang benar terhadap jumlah sampel positif yang diidentifikasi oleh model.

$$Precision = \frac{TP}{TP+FP} \quad (2.24)$$

3. *Recall* mengukur seberapa baik model dalam menemukan sampel yang sebenarnya positif. Ini mengukur rasio dari sampel positif yang benar yang ditemukan oleh model terhadap jumlah sampel positif yang sebenarnya.

$$Recall = \frac{TP}{TP+FN} \quad (2.25)$$

4. Nilai *f-measure* didapatkan dengan memadukan nilai *recall* dan *precision*.

$$f - measure = 2\ x\ \frac{precision\ x\ recall}{precision + recall} \quad (2.26)$$

2.11 Penelitian-penelitian Terkait

Kajian literatur adalah satu penelusuran dan penelitian kepustakaan dengan membaca berbagai buku, jurnal, dan terbitan-terbitan lain yang berkaitan dengan topik penelitian, untuk menghasilkan satu tulisan berkenaan dengan satu topik atau isu tertentu. Satu projek penelitian menghasilkan satu laporan bagi satu badan, kantor, atau perusahaan tertentu, atau untuk kepentingan peningkatan pengetahuan pribadi tentang satu hal tertentu, atau untuk diterbitkan dalam sebuah jurnal, atau untuk kepentingan mencapai satu ijazah (skripsi, tesis, dan disertasi) – tentulah menggunakan sejumlah literatur untuk bahan rujukan atau referensi[33]. Berikut adalah kajian literatur yang menjadi referensi pada tugas skripsi ini.

Tabel 2.3 Penelitian-penelitian Terkait

Review literatur pertama[6]	
Judul Artikel	Komparasi Metode K-NN, *Support Vector Machine*, Dan *Random Forest* Pada *E-Commerce* Shopee
Penulis	Sri Watmah, Suryanto, Martias
Judul Jurnal	INSANtek – Jurnal Inovasi dan Sains Teknik Elektro
Tahun Penerbitan	2021
Masalah Utama yang diangkat	Seberapa kepuasan pengguna akun *Shopee* berdasarkan *review* pada *Google Playstore* menggunakan metode K-NN, *Support Vector Machine*, dan *Random Forest*.
Metode Ekstraksi	Menggunakan TF-IDF

Metode Klasifikasi	K-NN, *Support Vector Machine*, dan *Random Forest*
Hasil Penelitian dan Kesimpulan	Hasil: Dari hasil studi yang dilakukan dengan menggunakan 265 data ulasan pada *Google Playstore* dengan menggunakan tiga metode klasifikasi yaitu *k-Nearest Neighbor*, *Support Vector Machine* dan *Random Forest* menunjukan bahwa metode K-NN mempunyai nilai akurasi 89,0%, presisi 89,7% dan recall 87,5%. Pada metode klasifikasi *Random Forest* menunjukan nilai akurasi 83,0%, presisi 85,7% dan *recall* 81,4%. Untuk metode SVM menunjukan nilai akurasi 89,4%, presisi 89,5% dan *recall* 89,7%. Kesimpulan: Klasifikasi terbaik pada studi ini adalah metode SVM dengan nilai 89,4% presisi 89,5% dan *recall* 89,7%.
Review literatur kedua[34]	
Judul Artikel	Analisis Sentimen Ulasan Aplikasi MOLA Pada *Google Play Store* Menggunakan Algoritma *Support Vector Machine*
Penulis	Muhammad Diki Hendriyanto, Azhari Ali Ridha, Ultach Enri
Judul Jurnal	*Journal of Information Technology and Computer Science* (INTECOMS)
Tahun Penerbitan	2022

Masalah Utama yang diangkat	Seberapa kepuasan pengguna aplikasi MOLA berdasarkan *review* pada *Google Playstore* menggunakan metode *Support Vector Machine*.
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: Sentimen pengguna terhadap aplikasi MOLA menghasilkan 312 ulasan positif dan 208 ulasan negatif. Pada hasil visualisasi kata-kata yang sering muncul pada ulasan positif yaitu “bagus”, “mola”, “mantap”, dan “gratis”. Sedangkan kata-kata yang sering muncul pada ulasan negatif yaitu “aplikasi”, “mola”, “eror”, “macet”. Kesimpulan: Pada hasil evaluasi kinerja empat kernel algoritma *Support Vector Machine* dalam menganalisis sentimen ulasan aplikasi MOLA dengan tiga skenario split data diperoleh hasil terbaik pada skenario 1 dengan rasio perbandingan data *training* 90% dan data *testing* 10% dengan hasil *accuracy* 92,31%, *precision* 96,3%, *recall* 89,66%, dan *f1- score* 92,86%
Review literatur ketiga[35]	
Judul Artikel	Analisis Sentimen berbasis Aspek Terhadap Data Ulasan Rumah Makan Menggunakan *Metode Support Vector Machine* (SVM)

Penulis	Salsabila Rahma Yustihan, Putra Pandu Adikara, Indriati
Judul Jurnal	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Tahun Penerbitan	2021
Masalah Utama yang diangkat	Ingin mengetahui kepuasan pelanggan terhadap beberapa aspek yang terkandung dalam suatu ulasan rumah makan.
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Metode Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: Metode SVM dalam mengklasifikasikan aspek dengan menggunakan pendekatan *macro averaging* menghasilkan *precision* sebesar 0,94, *recall* sebesar 0,6, *accuracy* sebesar 0,88, dan *f-measure* sebesar 0,73. Sedangkan hasil evaluasi yang diperoleh oleh metode SVM dalam mengklasifikasikan polaritas sentimen menghasilkan *precision* sebesar 0,86, *recall* sebesar 0,98, *accuracy* sebesar 0,86, dan *f-measure* sebesar 0,92. Kesimpulan: Hasil evaluasi dengan menggunakan pendekatan *macro averaging* menunjukkan bahwa sistem sudah cukup tepat dalam mengidentifikasi kelas yang bukan termasuk aspeknya dengan benar namun sistem belum cukup berhasil untuk

	mengidentifikasi semua kelas yang termasuk aspeknya dengan benar dikarenakan jumlah data yang tidak seimbang saat proses *training* pada setiap kelas aspek data. Hasil evaluasi dalam mengklasifikasikan polaritas sentimen menunjukkan bahwa sistem sudah cukup mampu dalam mengidentifikasi kelas yang mempunyai sentimen positif dengan benar, namun belum berhasil mengidentifikasi kelas yang mempunyai sentimen negatif dikarenakan jumlah data *training* yang digunakan saat proses pelatihan klasifikasi sentimen tidak seimbang dimana data positif lebih banyak dibandingkan dengan data negatif
Review literatur keempat[36]	
Judul Artikel	Implementasi *Support Vector Machine* untuk Analisis Sentimen Terhadap Pengaruh Program Promosi *Event* Belanja pada *Marketplace*
Penulis	Gientry Rachma Ditami, Eva Faja Ripanti, Herry Sujaini
Judul Jurnal	JEPIN (Jurnal Edukasi dan Penelitian Informatika)
Tahun Penerbitan	2022
Masalah Utama yang diangkat	Mengukur kepuasan pengguna aplikasi Shopee dan Tokopedia terhadap program promosi *event* belanja.

Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Metode Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: Dari hasil analisis pengujian model skenario 3 memang memiliki selisih nilai akurasi model *training* dan *testing* tertinggi sebesar +5.18%. Namun apabila dilihat dari segi model yang *good fit* maka Skenario 2 (*dataset Tokopedia*-parameter *grid search*) dan skenario 6 (dataset gabungan *Tokopedia* dan *Shopee*-parameter *grid search*) merupakan model yang baik karena memiliki nilai akurasi yang hampir sama dengan nilai selisih antara model *training* dan *testing* terkecil yaitu sebesar 0.7% dan 2.27%. Kesimpulan: Berdasarkan hasil perbandingan model yang dilakukan didapatkan bahwa penggunaan *grid search* untuk mencari parameter terbaik dapat meningkatkan nilai akurasi pada dataset Tokopedia dengan kenaikan nilai akurasi sebesar 1.44% dan dataset Shopee dengan kenaikan nilai akurasi sebesar 0.54%. Namun terjadi penurunan nilai akurasi saat menggunakan dataset gabungan Tokopedia dan Shopee sebesar -1.25%.
Review literatur kelima[5]	

Judul Artikel	Analisis Sentimen Aplikasi Gojek Menggunakan *Support Vector Machine* Dan *K-Nearest Neighbor*
Penulis	M.Nurul Muttaqin, Iqbal Kharisudin
Judul Jurnal	UNNES *Journal of Mathematics*
Tahun Penerbitan	2021
Masalah Utama yang diangkat	Seberapa kepuasan pengguna aplikasi Gojek berdasarkan *review* pada *Google Playstore* menggunakan metode *Support Vector Machine* dan *K-Nearest Neighbor*.
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine* dan *K-Nearest Neighbor*
Hasil Penelitian dan Kesimpulan	Hasil: Hasil pengujian pada metode SVM diperoleh hasil bahwa kernel linear dengan parameter C=1 memperoleh nilai terbaik dengan akurasi, presisi, dan *recall* berturutturut sebesar 87,98%, 88,55%, dan 95,43%. Hasil pengujian pada Metode KNN dengan nilai K=22 sebagai nilai K terbaiknya memperoleh nilai akurasi, presisi, dan *recall* berturut-turut sebesar 82,14%, 82,28%, dan 95,43%. Kesimpulan: Dapat disimpulkan bahwa metode *Support Vector Machine* (SVM) melakukan klasifikasi secara lebih baik dibandingkan *K-Nearest Neighbor* (KNN)

	pada ulasan pengguna aplikasi Gojek di *Google Playstore*.
Review literatur keenam[37]	
Judul Artikel	*Sentiment Analysis For Customer Review: Case Study of Traveloka*
Penulis	Ziedhan Alifio Dieksona, Muhammad Rivyyan Bagas Prakosoa, Muhammad Savio Qalby Putra, Muhammad Shaden Al Fadel Syaputra, Said Achmad, Rhio Sutoyo
Judul Jurnal	7th *International Conference on Computer Science and Computational Intelligence* 2022
Tahun Penerbitan	2022
Masalah Utama yang diangkat	Mengukur kepuasaan pengguna aplikasi Travekoka berdasarkan *tweet* pada Twitter menggunakan metode *Support Vector Machine, Logistic Regression*, dan *Naïve Bayes*
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine, Logistic Regression*, dan *Naïve Bayes*
Hasil Penelitian dan Kesimpulan	Hasil: Dari dataset diperoleh total 133.227 kata di dalamnya. Terdapat sekitar 690 *tweet* berkategori positif, dan 510 *tweet* berkategori negatif. Kesimpulan: Menggunakan model SVM diperoleh

	akurasi tertinggi dari dua model lainnya yaitu 84,5%. Yang terendah adalah regresi logistik, yang akurasinya memperoleh 82,50%, dan metode *Naive Bayes* sebesar 82,91%.
Review literatur ketujuh[38]	
Judul Artikel	*An Implementation Of Support Vector Machine On Sentiment Classification Of Movie Reviews*
Penulis	I M Yulietha, S A Faraby, Adiwijaya, dan W C Widyaningtyas
Judul Jurnal	*International Conference on Data and Information Science*
Tahun Penerbitan	2020
Masalah Utama yang diangkat	Mengklasifikasi sentimen pada dokumen ulasan film.
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: Pada pengujian *linear separable* dan *non-linear separable*, F1-*Score* yang lebih baik adalah 84,9% menggunakan *linear separable*. Dalam hal ini digunakan kernel linier, kernel RBF, dan kernel polinomial di SVM. Hasilnya adalah kernel linier memiliki hasil F1-*Score* terbaik yaitu 84,9%. Kesimpulan: Berdasarkan penelitian diperoleh kesimpulan bahwa semakin banyak data yang dijadikan data latih maka hasil F1-

	Score semakin baik untuk mengklasifikasikan. Dalam hal ini hasil terbaiknya adalah 85,6% dengan 90% data sebagai data pelatihan dan 10% data sebagai pengujian data.
Review literatur kedelapan[39]	
Judul Artikel	*Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method*
Penulis	Susanti Fransiska, Rianto, Acep Irham Gufroni
Judul Jurnal	*Scientific Journal of Informatics*
Tahun Penerbitan	2020
Masalah Utama yang diangkat	Seberapa kepuasan pengguna *provider by.U* berdasarkan *review* pada *Google Playstore* menggunakan metode *Support Vector Machine*.
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: Analisis data menghasilkan ulasan yang cenderung positif meskipun perbandingan angka dengan ulasan negatif tidak jauh berbeda sebanyak 54,6% ulasan dikelompokkan menjadi ulasan positif dan 45,4% ulasan dikelompokkan menjadi ulasan positif ulasan negatif.

	Kesimpulan: Penelitian ini mengungkapkan bahwa metode TF-IDF dan SVM dapat diterapkan pada proses klasifikasi dengan cukup baik hasil pengukuran, namun angka yang diperoleh tidak lebih baik dari penelitian sebelumnya, hal ini disebabkan adanya perbedaan pada dataset, proses pelabelan, tahapan *preprocessing* dan penggunaan fitur. Selain itu, pengaruh TF-IDF sebagai ekstraksi fitur pada pengukuran performa model tidak terlalu bagus, namun lebih baik menggunakan TF-IDF pembobotan.
Review literatur kesembilan[40]	
Judul Artikel	*Sentiment Analysis of Beauty Product E-Commerce Using Support Vector Machine Method*
Penulis	Muhammad Rio Pratama, Faza Abdillah Gunawan S, Rafdi Reyhan Zhafari, Rendy, Helena Nurramdhani Irmanda
Judul Jurnal	JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)
Tahun Penerbitan	2022
Masalah Utama yang diangkat	Ingin mengklasifikasikan *review* produk kecantikan *e-commerce* menggunakan metode *Support Vector Machine* untuk membuat model untuk mengkategorikan ulasan produk kecantikan dan menganalisis keakuratannya.

Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: Nilai-nilai presisi, *recall*, dan spesifisitas pada kelas positif yang lebih tinggi dibandingkan kelas negatif sebesar 80,05%, 81%, dan 80,77%. Namun, kelas negatif memiliki nilai yang cukup pada nilai presisi, *recall*, dan spesifisitas yang tinggi dianggap baik dan dapat diterima. Nilai tertinggi dari presisi, *recall*, dan spesifisitas dipengaruhi oleh jumlah data sebanyak 15.000. Kesimpulan: Berdasarkan hasil kelas positif dan negatif penelitian klasifikasi dengan dataset 50.000 *record* terdiri dari 35.000 data latih dan 15.000 data uji maka dapat disimpulkan bahwa *support vector machine* dapat mengklasifikasikan kelas ulasan dengan akurat nilai 80,06%.
Review literatur kesepuluh[41]	
Judul Artikel	*Sentiment Analysis of Community Opinion on Online Store in Indonesia on Twitter using Support Vector Machine Algorithm (SVM)*
Penulis	H Syahputra
Judul Jurnal	*Journal of Physics: Conference Series*
Tahun Penerbitan	2021

Masalah Utama yang diangkat	Masalah dari penelitian ini adalah ingin melihat opini masyarakat terhadap toko *online* di Indonesia melalui *Twitter* menggunakan *Support Vector Machine.*
Metode Ekstraksi	Menggunakan TF-IDF
Metode Klasifikasi	*Support Vector Machine*
Hasil Penelitian dan Kesimpulan	Hasil: *Support Vector Machine* dapat mengklasifikasikan *tweet* ke dalam Shopee, Tokopedia, Bukalapak, dan JDid *Online Shop* yang diperoleh dari Twitter menjadi tiga kelas yaitu negatif, netral, dan positif dengan akurasi tertinggi diatas 75%. Pada algoritma *Multiclass (One Vs Rest) Support Vector Machine*, kernel yang memiliki kualitas terbaik akurasinya adalah kernel *Sigmoid* dengan akurasi 82% pada dataset toko *online* Shopee, 94,7% pada dataset Tokopedia dan 75,3% untuk dataset Bukalapak, sedangkan pada dataset *The Linear kernel* Jdid memberikan akurasi yang lebih baik yaitu 78%. Kesimpulan: Kesalahan klarifikasi bisa terjadi karena *overfiting*, yaitu model sangat menyesuaikan data pelatihan baik (akurasinya bisa mencapai 100%) sehingga model tidak bisa menggeneralisasi dengan baik terhadap data pengujian. Lalu di dalam *tweet* tersebut

	terdapat sebuah kata yang mempunyai bobot lebih besar di kelas yang tidak seharusnya, yaitu mengakibatkan klasifikasi data yang salah.