BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Ada beberapa penelitian yang membahas tentang analisis sentimen dengan berbagai metode yang digunakan sebagai acuan pembuatan penelitian ini. Di antaranya adalah penelitian yang dilakukan oleh Himawan & Eliyani (2021) yang membahas tentang perbandingan tiga algoritma *machine learning* yaitu algoritma *Random Forest Classifier*, *Naïve Bayes*, dan *Support Vector Machine* untuk melakukan penelitian berupa analisis sentimen terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi dengan variabel nilai negatif, netral, dan positif. Penelitian ini menghasilkan hasil akurasi algoritma *Random Forest Classifier* sebesar 75,81%, algoritma *Naïve Bayes* dengan hasil akurasi 75,22%, dan algoritma *Support Vector Machine* 77,58%.

Selanjutnya adalah penelitian yang dilakukan oleh Simorangkir dan Lhaksamana (2018) mengenai analisis sentimen untuk Mobile Legends dan Arena of Valor di media sosial Twitter. Dengan menggunakan metode Naïve Bayes Classifier, peneliti menentukan klasifikasi *tweets* yang memiliki sentimen negatif dan positif. Hasilnya Mobile Legends memiliki 33 *tweet* positif dan 44 *tweet* negatif. Hasil akurasi, *error*, *recall* dan *precision* yang didapat untuk Mobile Legends masing-masing sebesar 88.89%, 19,18%, 96,97%, dan 69,57%. Sementara Arena of Valor memiliki 54 tweet positif dan 151 tweet negatif. Hasil nilai akurasi, *error*, *recall* dan *precision* untuk Arena of Valor masing-masing sebesar 39,02%, 60,98%, 88,89% dan 28,74%.

Penelitian berikutnya adalah penelitian yang dilakukan oleh Septian, Fahrudin, dan Nugroho (2019) mengenai analisis sentimen pada Twitter tentang polemik persepakbolaan di Indonesia. Penelitian ini menggunakan pembobotan TF-IDF dan klasifikasi dengan metode *K-Nearest Neighbor* pada 2000 data *tweet* berbahasa Indonesia yang memiliki kata kunci "@pssi". Dari seluruh data tersebut, didapatkan hasil akurasi optimal pada nilai k=23 sejumlah 79,99%.

Pravina, Cholissodin, Adikara (2019) melakukan penelitian berupa analisis sentimen pada opini masyarakat di Twitter tentang maskapai penerbangan menggunakan metode *Support Vector Machine*. Sentimen analisis ini menerapkan fitur *Lexicon Based* untuk menerima opini berbahasa lain selain Bahasa Indonesia. Dengan menggunakan parameter C bernilai 10 dan learning rate bernilai 0,03 dan menggunakan Lexicon Based Features dengan iterasi sebanyak 50 kali, dapat dihasilkan *accuracy* sebesar 40%, *precision* 40%, 100% *recall*, dan *f-measure* sebesar 57,14%.

Kemudian Haranto dan Sari (2019) juga melakukan penelitian tentang implementsi *Support Vector Machine* untuk analisis sentimen tentang opini masyarakat terhadap pelayanan Telkom dan Biznet pada medias sosial Twitter. Penelitian ini menggunakan dataset sebanyak 500 *tweet* yang berasal dari *crawling* data Twitter, dan terdapat 250 *tweet* yang dijadikan *dataset* pada masing-masing objek. Penelitian ini menghasilkan nilai *accuracy* 79,6%, *precision* 76,5%, *recall* 72,8% , dan *F1-score* 74,6% untuk Telkom, serta *accuracy* 83,2%, *precision* 78,8%, *recall* 71,6%, dan *F1-score* 75% untuk Biznet.

Penelitian terakhir adalah penelitian yang dilakukan oleh Tuhuteru dan Iriani (2018) membahas tentang analisis sentimen mengenai kinerja PLN cabang Ambon menggunakan dua metode *machine learning* yaitu *Support Vector Machine* dan *Naïve Bayes Classifier*. Hasil perbandingan metode klasifikasi analisis sentimen pada kasus ini menunjukkan metode SVM lebih baik daripada NBC, dengan tingkat akurasi sebesar 81.67%. Sedangkan metode klasifikasi NBC hanya memiliki nilai akurasi sebesar 67.20%.

Tabel 2.1 Tinjauan Pustaka

No	Nama Peneliti	Judul Penelitian	Metode	Hasil
1	Himawan & Eliyani (2021)	Perbandingan Akurasi Analisis Sentimen Tweet terhadap Pemerintah Provinsi DKI Jakarta di Masa Pandemi	*Random Forest Classifier*, *Naïve Bayes* , *Support Vector Machine*.	*Linear SVM* memiliki akurasi terbaik dengan hasil 77,58%, *Random Forset Classifier* dengan hasil 75,81%, dan *Multinomial Naive Bayes* sebesar 75,22%..
2	Simorangkir dan Lhaksmana (2018)	Analisis Sentimen pada Twitter untuk Games Online Mobile Legends dan Arena of Valor dengan Metode Naïve Bayes Classifier	*Naïve Bayes Classifier*	Dapat memprediksi polarisasi sentimen *Mobile Legends* dengan nilai hasil akurasi, error, recall dan precision yang didapat masing-masing sebesar 88.89%, 19,18%, 96,97%, dan 69,57%. Sedangkan *Arena of Valor* memiliki nilai

No	Nama Peneliti	Judul Penelitian	Metode	Hasil
				akurasi, eror, recall dan precision masing-masing sebesar 39,02%, 60,98%, 88,89% dan 28,74%.
3	Septian, Fahrudin, dan Nugroho (2019)	Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor	*K-Nearest Neighbor*	Dari range nilai k=1 hingga k=30 yang merupakan bilangan ganjil, didapatkan akurasi optimal pada k=23 dengan akurasi sebesar 79,99% dan *error rate* sebesar 20,01%.
4	Pravina, Cholissodin, Adikara (2019)	Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM)	*Support Vector Machine*	Didapatkan nilai parameter *learning rate* (*gamma*) sebesar 0,03 dan nilai C sebesar 10 sebagai nilai parameter paling optimal. Didapatkan tingkat akurasi paling baik sebesar 40%, *precision* sebesar 40%, *recall* sebesar 100%, dan *f-measure* sebesar 57,14%.
5	Haranto dan Sari (2019)	Implementasi Support Vector Machine untuk Analisis Sentimen Pengguna Twitter	*Support Vector Machine*	Menghasilkan nilai *accuracy* 79,6%, *precision* 76,5%, *recall* 72,8% , dan *F1-score* 74,6% untuk Telkom,

No	Nama Peneliti	Judul Penelitian	Metode	Hasil
		Terhadap Pelayanan Telkom dan Biznet.		serta *accuracy* 83,2%, *precision* 78,8%, *recall* 71,6%, dan *F1-score* 75% untuk Biznet.
6	Tuhuteru dan Iriani (2018)	Analisis Sentimen Perusahaan Listrik Negara Cabang Ambon Menggunakan Metode Support Vector Machine dan Naive Bayes Classifier	*Support Vector Machine* dan *Naïve Bayes Classifier*	Pada penelitian ini, SVM memiliki tingkat akurasi sebesar 81.67%. Sedangkan metode klasifikasi NBC hanya memiliki nilai akurasi sebesar 67.20%.

2.2 Dasar Teori

2.2.1 Twitter

Twitter adalah sebuah situs jejaring media sosial *micro-blogging* gratis yang dikembangkan pada Maret 2006 oleh Jack Dorsey, Noah Glass, Biz Stone, dan Evan Williams dan dapat digunakan oleh khayalak umum semenjak Juli 2006 (Paramastri dan Gumilar, 2019). Media sosial ini memungkinkan *user* untuk menulis pesan singkat yang disebut dengan *tweet*. *Tweet* tersebut bisa berupa teks, video, foto, atau link. Pesan-pesan *tweet* yang ditulis oleh pengguna Twitter tersebut akan ditampilkan di laman *profile*, ditampilkan ke para *followers*, dan juga bisa dicari dengan fitur *search*. (help.twitter.com, 2022).

Laporan terbaru We Are Social di tahun 2022 mengungkapkan suatu fakta bahwa Indonesia adalah salah satu negara dengan pengguna Twitter terbanyak di dunia. Menurut laporan tersebut, jumlah pengguna Twitter di Indonesia pada tahun

2022 mencapai jumlah 18,45 juta pengguna atau setara dengan 4,23% dari total seluruh pengguna Twitter di dunia yang mencapai angka 436 juta (dataindonesia.id, 2022).

2.2.2 Snscrape

Snscrape adalah sebuah *scraper* yang bisa digunakan untuk mengambil data dari *social networking services* (SNS) atau biasa disebut dengan media sosial. Tool ini bisa digunakan untuk mengambil data-data seperti profil pengguna, tagar, pencarian, dan lain-lain pada media sosial seperti Facebook, Instagram, Reddit, Telegram, Twitter, dan lain-lain. Tool ini memerlukan bahasa pemrograman Python versi 3.8 atau lebih supaya bisa diinstall dan digunakan. (JustAnotherArchivist, 2023).

2.2.3 Python

Python adalah bahasa pemrograman *high-level*, interpretatif, multiguna, berorientasi objek dengan semantik dinamis. Sintaks Python yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan *code reuse*. *Interpreter* Python dan pustaka standar yang luas tersedia dalam bentuk *source* atau biner tanpa biaya untuk semua platform dan dapat didistribusikan secara bebas. (Python Software Foundation, 2022)

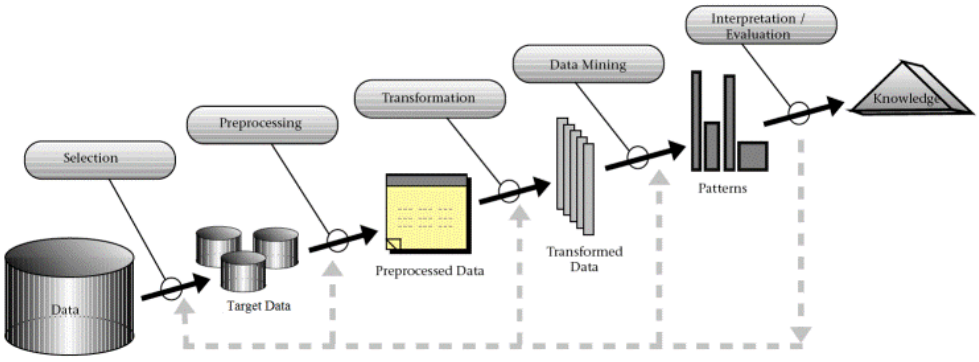
Python membuat penulisan program menjadi padat (*compact*) dan mudah dibaca. Program yang ditulis di Python pada dasarnya memerlukan kode yang lebih sedikit daripada program yang ditulis dengan bahasa C, C++, atau Java karena:

1. Tipe data *high-level* yang dapat melakukan operasi kompleks dalam satu *statement*.
2. Pengelompokan *statement* dilakukan dengan *indentation* (tulisan sedikit menjorok ke kanan), bukan dengan *brackets*.
3. Tidak memerlukan deklarasi variabel atau argumen.

2.2.4 Data Mining

Data mining adalah adalah gabungan dari beberapa ilmu komputer yang didefinisikan sebagai proses penemuan pola-pola baru dari kumpulan data yang sangat besar, meliputi metode-metode yang merupakan bagian dari *artificial intelligence*, *machine learning*, *statistics*, dan *database systems*. Data mining bertujuan untuk mengekstrak pengetahuan dari kumpulan data supaya didapatkan strukur yang dapat dipahami oleh manusia (Suyanto, 2017).

Dalam penerapannya, data mining merupakan salah satu bagian dari sebuah proses yang dinamakan *Knowledge Discovery in Database* (KDD) yaitu proses ektraksi *non trivial* dari implisit suatu informasi yang sebelumnya tidak diketahui tetapi terdapat potensi informasi yang dihasilkan dari data yang ada (Ependi & Putra, 2019). Grafik KDD ditunjukkan di Gambar 2.1.



Gambar 2.1 Tahapan *Knowledge Discovery Database* (Ependi & Putra, 2019)

Adapun proses *Knowledge Discovery Database* adalah sebagai berikut:

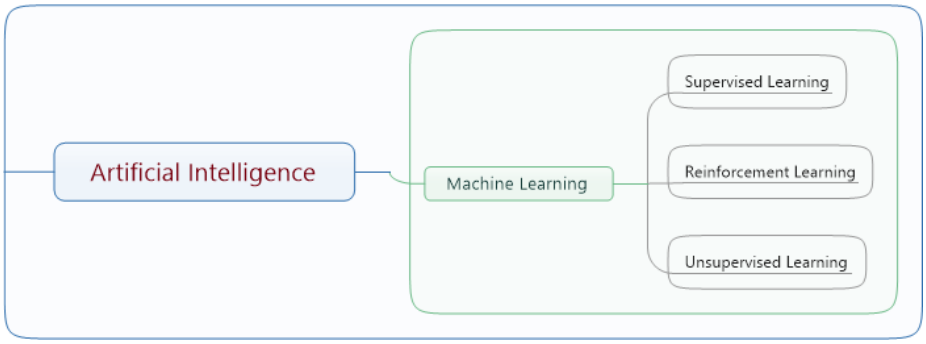
1. *Data Selection*: proses pengambilan data - data yang relevan untuk kemudian dimasukkan ke proses analisis.
2. *Preprocessing*: proses cleaning data yang mencakup beberapa proses seperti membuang duplikasi data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan pada data.
3. *Data transformation*: proses transformasi dan konsolidasi data ke dalam bentuk yang sesuai untuk ditambang supaya bisa menghasilkan sebuah kesimpulan atau penggabungan.
4. *Data mining*: proses awal yaitu penerapan metode pengkajian untuk mengekstraksi pola data.
5. *Pattern evaluation*: proses mengindentifikasi pola unik yang mewakili basis pengetahuan berdasarkan ukuran tertentu.
6. *Knowledge Presentation*: proses teknik visualisasi dan presentasi yang digunakan untuk menampilkan pengetahuan atau hasil kepada pengguna.

2.2.5 Machine Learning

Machine learning merupakan subbidang dari bidang keilmuan *artificial intelligence*, dengan pemrograman untuk memberikan kecerdasan kepada komputer yang pemahaman maupun kemampuannya dapat ditingkatkan melalui pengalaman secara otomatis. *Machine learning* dapat dilakukan jika ada data yang tersedia sebagai *input* untuk kemudian dilakukan analisis terhadap kumpulan *big data* untuk menemukan pola tertentu. Di dalam *machine learning* dikenal istilah *data training*

dan *data testing*. Proses *data training* digunakan untuk melatih algoritma yang digunakan, sedangkan *data testing* digunakan untuk mengetahui performa dari algoritma *machine learning* yang telah dilatih yaitu ketika diterapkan pada *dataset* baru yang belum pernah diberikan dalam proses *training* (Retnoningsih & Pramudita, 2020).

Machine learning dibagi menjadi tiga kategori yaitu: *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Roihan, et al., 2020). Grafik relasi antara *artificial intelligence* dan *machine learning* ditunjukkan dalam Gambar 2.2.



Gambar 2.2 Skema *Artificial Intelligence* dan *Machine Learning* (Roihan, et al., 2020)

Supervised learning adalah metode klasifikasi yang memberikan label untuk kumpulan data untuk kemudian diklasifikasikan ke dalam kelas. Sementara pada *unsupervised learning* tidak dibutuhkan pemberian label dalam kumpulan data dan hasilnya tidak mengidentifikasi contoh di kelas yang telah ditentukan. Sedangkan *reinforcement learning* bekerja di dalam lingkungan yang dinamis yang memiliki konsep yaitu harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai (Roihan, et al., 2020).

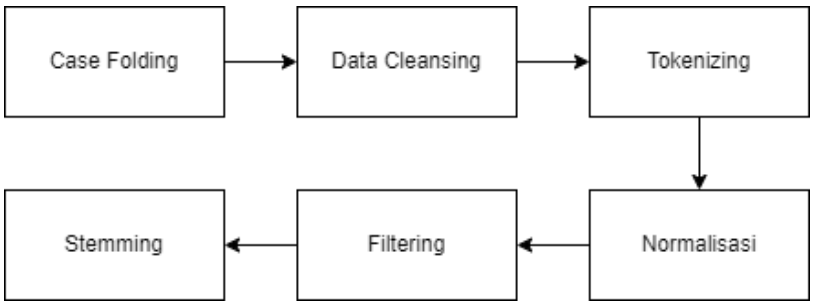
2.2.6 Analisis Sentimen

Analisis sentimen adalah metode komputasional untuk mengekstraksi dan menganalisis sentimen pada suatu entitas dan atribut yang dimiliki (Himawan & Eliyani, 2021). Analisis sentimen dilakukan untuk melihat pendapat terhadap sebuah masalah atau dapat juga digunakan untuk identifikasi kecenderungan dari suatu permasalahan. Tugas dasar dalam analisis sentimen adalah mengelompokkan polaritas dari teks yang ada dalam dokumen, kalimat, atau pendapat. Polaritas mempunyai arti apakah teks yang ada dalam dokumen, kalimat, atau pendapat memiliki aspek positif, netral, atau negatif (Simorangkir & Lhaksmana, 2018).

2.2.7 Text Preprocessing

Text Preprocessing atau praproses teks adalah suatu proses yang digunakan untuk melakukan transformasi teks dari yang awalnya berbentuk data tidak terstruktur yang memiliki banyak *noise* menjadi data yang terstruktur sehingga proses analisis sentimen dapat menjadi lebih mudah untuk dilakukan (Husada & Paramita, 2021).

Teks yang berada dalam internet sering mengandung banyak *noise* dan hal mengganggu lainnya seperti tag HTML, *script*, dan iklan. Dengan *preprocessing*, maka *noise* dalam teks dapat dikurangi sehingga hal tersebut bisa meningkatkan performa dan mempercepat proses klasifikasi supaya dapat dengan efektif membantu dalam proses analisis sentimen secara *real-time* (Indrayuni, 2019). *Preprocessing* meliputi beberapa proses yang bisa dilihat pada Gambar 2.3.



Gambar 2.3 *Text Preprocessing*

Penjelasan tahap *preprocessing* (Septian, et al., 2018):

1. *Case Folding*: proses konversi semua teks dalam suatu dokumen menjadi bentuk yang seragam. Dengan kata lain, *case folding* berfungsi untuk membuat seluruh huruf teks dalam dokumen menjadi huruf kecil.
2. *Data Cleansing*: proses pembersihan pada dokumen yang berisi angka, url (http://) ,username (@) ,tanda pagar (#) ,delimiter seperti koma (,) dan titik (.) dan tanda baca lainnya.
3. *Tokenizing*: Proses pemotongan pada dokumen atau kalimat menjadi kata-kata yang disebut denagan token.
4. *Normalisasi*: Proses normalisasi terhadap setiap kata dalam dokumen yang tidak baku menjadi kata yang baku dan siap diolah. Kata tidak baku tersebut artinya adalah kata-kata yang tidak sesuai dengan Kamus Besar Bahasa Indonesia (KBBI).
5. *Filtering*: Proses ini juga bisa disebut dengan *stopword removal*. Proses ini dilakukan untuk menghapus kata-kata yang frekuensi kemunculannya tinggi tetapi tidak berpengaruh terhadap proses analisis data seperti ‘yang’, ‘dan’,’ke’, ‘di’, dan lain-lain.

6. *Stemming*: Proses ini merupakan proses untuk mengubah semua kata-kata pada dokumen menjadi kata dasar dengan menghilangkan semua kata imbuhan. Kata imbuhan yang dihilangkan terdiri dari awalan (prefix), akhiran (suffix), sisipan (infix), dan gabungan awalan-akhiran (confix).

2.2.8 Pelabelan Data

Pelabelan data adalah suatu proses untuk menentukan suatu kalimat opini termasuk ke dalam kelas sentimen positif atau sentimen negatif kemudian diberi label sesuai sentimennya. Umumnya, proses pelabelan data membagi kelas menjadi tiga kelas yaitu positif, netral, dan negatif. Skor > 0 akan diklasifikasikan ke dalam kelas sentimen positif, skor < 0 maka akan diklasifikasikan ke dalam kelas negatif, dan skor = 0 akan diklasifikasikan ke dalam kelas netral (Mubaroroh et al., 2022).

Proses pelabelan data bisa dilakukan menggunakan *library* Python bernama TextBlob. TextBlob adalah sebuah paket *open-source* pada Python yang berguna untuk melakukan tugas – tugas dasar *Natural Language Processing* seperti tokenisasi, klasifikasi, pelabelan, terjemahan, sentimen analisis, dan lain – lain. (Suanpang et al., 2021). Berikut cara kerja TextBlob:

1. Model yang ada pada TextBlob dilatih dengan memasukkan teks untuk kemudian didapatkan nilai sentimen dalam bentuk polaritas dan subjektivitas.
2. TextBlob memberikan nilai polaritas pada teks masukan. Nilai teks masukan ada di *range* [-1.0, 1.0] dimana skor -1 merupakan teks yang mengandung sentimen negatif dan skor 1.0 merupakan teks yang mengandung sentimen positif.

3. TextBlob juga mendeteksi objektivitas dan subjektivitas pada sebuah teks yang memiliki *range* nilai [0.0, 1.0] dimana nilai 0.0 adalah teks yang sangat objektif, sementara 1.0 adalah teks yang sangat subjektif. Subjektivitas mengukur banyaknya pendapat pribadi dan informasi faktual yang terkandung dalam sebuah teks. Subjektivitas yang tinggi berarti teks tersebut mengandung pendapat pribadi, subjektivitas yang rendah berarti teks tersebut mengandung informasi yang faktual.

2.2.9 Ekstraksi Fitur

Ekstraksi fitur adalah sebuah proses untuk mencari nilai fitur yang terkandung dalam dokumen yang dapat digunakan untuk analisis sentimen atau *opinion mining* (Prihatini, 2017). Proses ini adalah sebuah proses penting pada klasifikasi teks yang digunakan untuk mengubah format tekstual yang tidak terstruktur menjadi format tekstual terstruktur sehingga selanjutnya dapat diproses oleh algoritma *machine learning* untuk diklasifikasikan ke dalam kelas yang telah ditentukan (Budiman, et al., 2020). Ekstraksi fitur bisa dilakukan dengan salah satu *library Python* menggunakan *CountVectorizer*. *CountVectorizer* berfungsi untuk menghitung frekuensi kata dalam suatu dokumen dan juga dapat mengubah fitur teks menjadi representasi *vector* (Munawar & Silitonga, 2019).

N-gram juga diimplementasikan sebagai metode ekstraksi fitur di dalam penelitian ini. Proses ini mengambil sejumlah n karakter sebagai suatu dan menghitung berapa banyak kata itu muncul dan probabilitas dari *n-gram* tersebut. Dengan kata lain, metode ini berguna untuk mengambil potongan – potongan

karakter dari kata atau kalimat sebanyak jumlah karakter pada kata tersebut (Nugroho, 2018).

Dari penjelasan tersebut, dapat dituliskan algoritma atau cara kerja ekstraksi fitur dalam penelitian ini yaitu:

1. Mengubah fitur teks menjadi representasi vektor dengan *CountVectorizer*. Output dari proses tersebut adalah berupa data dengan tipe *Document Term Matrix* (DTM). DTM adalah suatu matrix yang menggambarkan frekuensi kemunculan kata atau istilah dalam suatu dokumen. Baris pada DTM mempresentasikan dokumen teks dan kolom mempresentasikan istilah teks (Ellina et al., 2022).
2. Contoh sederhana output *CountVectorizer* di Tabel 2.2.

   Tabel 2.2 Contoh DTM

	Pernikahan	**Di**	**Usia**	**Sangat**	**Muda**
DTM-1: Pernikahan Usia Muda	1	0	1	0	1
DTM-2: Pernikahan di Usia Sangat Muda	1	1	1	1	1

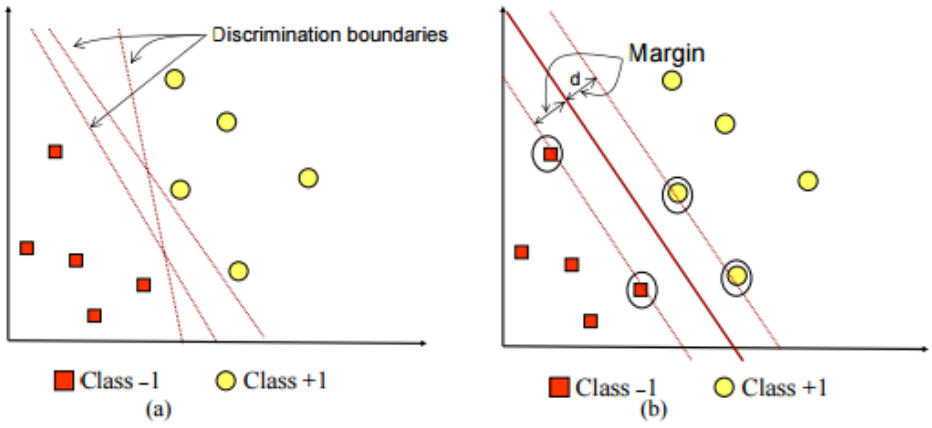
3. Selanjutnya ada proses penerapan *n-gram*. *N-gram* memiliki tiga jenis pemecahan kata yaitu *unigram*, *bigram*, dan *trigram*. *Unigram* adalah pemisahan kata pada teks dengan n=1, *bigram* adalah pemisahan kata pada teks dengan n=2, *trigram* adalah pemisahan kata pada teks dengan n=3 (Anjani dan Fauzan, 2021). Ilustrasi penerapan *n-gram* ada di Gambar 2.3.

   Tabel 2.3 Contoh Penerapan *n-gram*

Unigram	‘di’, ’desa’, ‘saya’, ‘banyak’, ‘yang’, ‘nikah’, ’muda’
Bigram	‘di desa’, ‘desa saya’, ‘saya banyak’, ‘banyak yang’, ‘yang nikah’, ‘nikah muda’
Trigram	‘di desa saya’, ‘desa saya banyak’, ‘saya banyak yang’, ‘banyak yang nikah’, ‘ yang nikah muda’

2.2.10 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode *machine learning* yang bekerja dengan prinsip *Structural Risk Minimization* (SRM) yang termasuk dalam kategori *supervised learning*. Dalam prosesnya, metode SVM memiliki tujuan yaitu untuk menemukan *hyperplane* paling optimal yang yang berfungsi untuk memisahkan dua buah kelas. Tingkat akurasi pada model yang dihasilkan oleh proses klasifikasi dengan SVM sangat bergantung terhadap fungsi kernel dan parameter yang digunakan. (Parapat, et al., 2018).



Gambar 2.4 Support Vector Machine (Parapat, et al., 2018)

Pada ilustrasi Gambar 2.4, ada dua kelas yang dipisahkan oleh garis *hyperplane* yaitu kelas positif yang bernilai +1 (lingkaran kuning) dan kelas negatif

yang bernilai -1 (kotak merah). Garis solid yang terdapat pada tengah-tengah kedua kelas adalah *hyperplane* terbaik, dan objek merah dan kuning yang berada dalam lingkaran hitam disebut dengan *support vector*.

Pada algoritma *Support Vector* Machine, data ke-i pada dataset diwakilkan dengan variabel x_i, sementara kelas pada dataset diwakilkan dengan variabel y_i. Data x_i yang termasuk dalam kelas +1 dirumuskan dengan persamaan (1), sedangkan data x_i yang termasuk dalam kelas -1 dirumuskan dengan persamaan (2) (Parapat et al., 2018).

$$x_i.w + b \geq 1, y_i = 1 \quad (1)$$

$$x_i.w + b \leq 1, y_i = -1 \quad (2)$$

Keterangan:

x_i = data ke -i

w = nilai bobot *support vector* yang tegak lurus dengan *hyperplane*

b = nilai bias

y_i = kelas data ke-i

Berikut tahap – tahap perhitungan klasifikasi menggunakan SVM:

1. Meminimalkan nilai margin (Zalyhaty et al., 2020). Tahap ini dapat dilakukan dengan menggunakan persamaan berikut:

   $$\frac{1}{2}\,||w||^2 = \frac{1}{2}\,(w_1^2 + w_2^2) \quad (3)$$

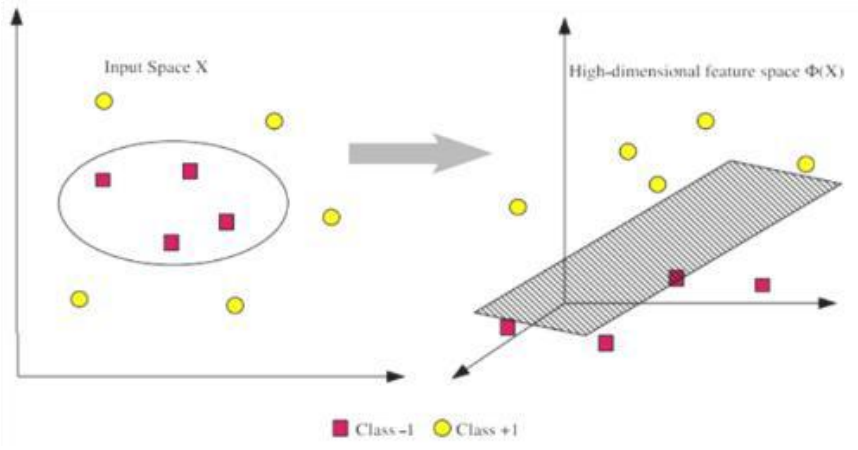
   dengan syarat: $y_i(w_i.x_i + b) \geq 1,\ i = 1,2,3,\dots,n$ (4)

2. Setelah meminimalkan nilai margin, maka dapat ditemukan nilai w (bobot) dan nilai b (bias), lalu bisa dicari persamaan *hyperplane*.

3. Menghitung *margin hyperplane* dalam proses menemukan titik maksimal. Persamaan (5) adalah rumus untuk memperoleh garis *hyperplane* pada SVM (Husada & Paramita, 2021).

$$w_i . x_i + b = 0 \quad (5)$$

Prinsip kerja algoritma *Support Vector Machine* pada dasarnya adalah suatu algoritma yang digunakan untuk klasifikasi data *linear*, sehingga dalam proses klasifikasi seringkali ditemukan kondisi dimana SVM tidak bekerja dengan baik dalam melakukan klasifikasi pada data *non-linear*. Masalah tersebut bisa diatasi dengan menggunakan *kernel trick*. *Kernel trick* digunakan untuk memetakan data *non-linear* berdimensi rendah ke dalam ruang dimensi yang lebih tinggi sehingga membuat data terpisah secara *linear* lalu dapat terbentuk *hyperplane* yang optimal. Proses klasifikasi dengan SVM dapat dilakukan dengan memilih salah satu di antara 4 kernel yang tersedia yaitu *linear*, *polymonial*, *RBF*, dan *sigmoid* (Husada & Paramita, 2021). Ilustrasi *kernel trick* dapat dilihat di Gambar 2.5.



Gambar 2.5 Pemetaan *Input Space* Berdimensi Dua dengan Pemetaan ke Dimensi Tinggi (Rahutomo et al., 2018)

2.2.11 Evaluasi Performansi

Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya *confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Karsito dan Susanti, 2019). Tabel 2.4 menggambarkan contoh *confusion matrix*.

Tabel 2.4 *Confusion Matrix*

	Kelas Prediksi	
	Positif	Negatif
Positif	True Positive	True Negative
Negatif	False Positive	False Negative

Keterangan isi tabel:

1. *True Positive* (TP), yaitu data asli positif dan data klasifikasi positif.
2. *True Negative* (TN), yaitu data asli negatif dan data klasifikasi negatif.
3. *False Positive* (FP), yaitu data asli negatif dan data klasifikasi positif.
4. *False Negative* (FN), yaitu data asli positif dan data klasifikasi negatif.

Setelah itu dapat dilakukan perhitungan untuk menghasilkan accuracy, precision, recall, dan f1-score. *Accuracy* adalah perbandingan kasus yang diidentifikasi benar dengan jumlah semua data. *Precision* adalah rasio prediksi benar positif dibandingkan dengan hasil prediksi positif secara keseluruhan. *Recall* adalah rasio benar positif dibandingkan dengan seluruh data positif. *F1-Score*

adalah parameter perbandingan rata-rata *precision* dan *recall* yang dibobotkan (Hidayat, Ardiansyah, & Setyanto, 2021). Berikut rumusnya:

$$\text{Akurasi} = \frac{TP+TN}{TP+FN+FN+TN} \text{ x } 100\ \% \quad (6)$$

$$\textit{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\textit{Recall} = \frac{TP}{TP+FN} \quad (8)$$

$$\textit{F1-Score} = \frac{2*Precision*Recall}{Precision+Recall} \quad (9)$$