BAB 2
LANDASAN TEORI

2.1 Teori tentang Topik

2.1.1 Analisis Sentimen

Analisis sentimen adalah suatu proses untuk mengekstraksi dan memahami opini secara otomatis dari sebuah teks yang tidak terstruktur untuk menentukan kecenderungan opini terhadap suatu topik [9]. Analisis sentimen atau juga dikenal dengan sebagai *opinion mining* merupakan salah satu bidang studi akademis yang berfokus pada cara ekspresi sentimen, opini, sikap, dan penilaian seseorang diungkapkan dalam tulisan [13].

2.1.2 Pengungsi Rohingya

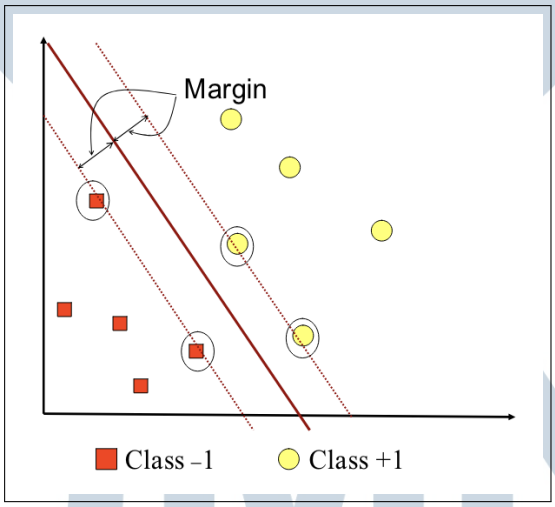
Pengungsi Rohingya merupakan salah satu pengungsi internasional yang berasal dari kelompok etnis Rohingya, yaitu etnis minoritas di Myanmar yang sebagian besar beragama Muslim. Etnis Rohingya telah menjadi sasaran diskriminasi institusional, termasuk dengan adanya undang-undang kewarganegaraan yang eksklusif, selama bertahun-tahun. Adanya kampanye militer yang diprakarsai oleh pemerintah Myanmar yang menyasar ke etnis Rohingya mengakibatkan migrasi paksa ribuan orang Rohingya ke negara-negara lain [1].

Menyikapi adanya migrasi tersebut, sejak tahun 2015, Indonesia secara konsisten terbuka dan menawarkan pengungsian sementara bagi pengungsi Rohingya yang terdampar di laut sebagai solusi terhadap krisis kemanusiaan yang sedang terjadi [2]. Pada akhir tahun 2023, terjadi lonjakan besar jumlah pengungsi Rohingya di Aceh, Indonesia yang diakibatkan karena memburuknya situasi tempat pengungsian di Bangladesh. Data dari UNHCR mencatat total populasi pengungsi Rohingya di Aceh hingga 12 Desember 2023 mencapai 1,722 orang, dengan 1,543 pengungsi datang di Aceh, Indonesia sejak 14 November 2023 [4].

2.2 Teori tentang Algoritma

2.2.1 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu algoritma dalam *supervised learning* untuk melakukan klasifikasi, artinya model belajar dari data yang telah diberi label tertentu, sehingga model dapat memetakan data dengan benar. Pada dasarnya algoritma SVM melakukan pencarian suatu garis pembatas (*hyperplane*) yang paling optimal untuk memisahkan dua atau lebih kelas tertentu dengan cara menghitung jarak (*margin*) terjauh antara *hyperplane* dan titik data terdekat dari *hyperplane* (*support vector*) [12]. Pada gambar 2.1 berikut, garis *solid* berwarna merah menunjukkan posisi *hyperplane* terbaik, dimana garis terletak di tengah-tengah kedua kelas, sedangkan titik merah dan kuning dalam lingkaran hitam adalah support vector [14].



Gambar 2.1. Hyperplane Terbaik

Sumber: [14]

Persamaan 2.1 berikut adalah persamaan dasar untuk mengukur nilai *optimal hyperplane*. Dari persamaan berikut, label data masing-masing ditandai sebagai $y_i \in \{-1,+1\}$ untuk $i = 1,2,\ldots,l$, di mana l adalah jumlah data yang ada. Diasumsikan bahwa kedua kelas -1 dan +1 dapat dipisahkan secara sempurna oleh *hyperplane* yang didefinisikan.

$$\vec{w} \cdot \vec{x} + b = 0 \tag{2.1}$$

Adapun persamaan tersebut dikembangkan berdasarkan perbedaan *output* kelas yi. Pertidaksamaan 2.2 berikut merupakan pertidaksamaan dengan *output* kelas yi = +1, dimana pola $\vec{x}_i$ termasuk dalam kelas +1 (sampel positif). Sedangkan pertidaksamaan 2.3 berikut merupakan pertidaksamaan dengan *output* kelas yi = -1, dimana pola $\vec{x}_i$ termasuk dalam kelas -1 (sampel negatif). Pertidaksamaan berikut dipakai dengan asumsi bahwa kedua kelas data terpisah secara linear, namun pada kebanyakan kasus kelas data tidak terpisah secara linear.

$$\vec{w} \cdot \vec{x}_i + b \geq 1, y_i = 1 \tag{2.2}$$

$$\vec{w} \cdot \vec{x}_i + b \leq 1, y_i = -1 \tag{2.3}$$

Untuk menentukan nilai *margin* terbesar diperlukan nilai maksimal jarak antara *hyperplane* dan *support vector*, seperti pada persamaan 2.4. Permasalahan tersebut dirumuskan kembali sebagai masalah *quadratic programming* (QP) dengan mencari titik minimum persamaan 2.5 sesuai dengan batasan atau *constrain* 2.6.

$$\max \frac{1}{2} \|\vec{w}\| \tag{2.4}$$

$$\min_{\vec{w},b} \frac{1}{2} \|\vec{w}\|^2 \tag{2.5}$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad \forall i \tag{2.6}$$

Ketika dua kelas dalam ruang *input* tidak dapat dipisahkan secara sempurna, maka menghasilkan *noise* atau kesalahan klasifikasi, sehingga batasan pada persamaan 2.6 tidak dapat dipenuhi dan optimisasi tidak dapat dilakukan. Untuk mengatasi permasalahan tersebut, digunakan teknik *soft margin*, yaitu dengan memberikan nilai toleransi kesalahan klasifikasi berupa variabel *slack*. Dalam *soft margin*, persamaan 2.5 dan batasan pada 2.6 dimodifikasi dengan menambahkan

variabel *slack*. Persamaan 2.7 dan persamaan 2.8 berikut menunjukkan persamaan untuk menentukan *soft margin* dan batasannya.

$$\min_{\vec{w},b,\xi} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.7}$$

$$y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \tag{2.8}$$

Variabel-variabel pada persamaan di atas dapat didefinisikan sebagai berikut:

- $\vec{w}$ = Garis tegak lurus yang berada diantara garis *hyperpylane* dan titik *support vector*
- $\vec{x_i}$ = Titik fitur *input* data
- b = Nilai bias
- yi = *Output* kelas dari data $\vec{x_i}$
- ξ = Nilai *slack*
- C = Parameter untuk menentukan *trade-off* antara memaksimalkan *margin* dan meminimalkan kesalahan klasifikasi

Prinsip dasar SVM adalah sebagai pengklasifikasi linear, tetapi kemudian berkembang untuk menangani masalah non linear. Ketika distribusi data yang dipetakan tidak terpisah secara linear atau acak, maka pemisahan tidak dapat dilakukan hanya dengan suatu garis linear. Hal ini menyebabkan adanya pendekatan menggunakan kernel *trick* yang bertujuan memetakan data dengan dimensi tertentu ke dimensi yang lebih tinggi sehingga memungkinkan pemisahan data non linear dengan permasaan sebagai berikut.

$$K(x_i, x_j) = \Phi(x_i)\cdot\Phi(x_j) \tag{2.9}$$

Terdapat beberapa pendekatan kernel yang dapat digunakan seperti linear, radial bias function (RBF), dan polinomial, dan sigmoid [15]. Tabel 2.1

berikut menunjukkan perbandingan persamaan yang digunakan pada masing-masing kernel.

Jenis Kernel	Definisi	Parameter
Polynomial	$K(x_i, x_j) = (x_i \cdot x_j + 1)^p$	p (derajat polinomial)
Gaussian (RBF)	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$	σ (lebar Gaussian)
Sigmoid	$K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + \beta)$	α, β (parameter kernel)

Tabel 2.1. Perbandingan berbagai jenis kernel dalam SVM

Pada klasifikasi SVM dengan *instance multiclass*, digunakan beberapa pendekatan seperti *One Against One* (OAO) dan *One Against All* (OAA). Pendekatan *One Against All* menghasilkan N *decision boundary* dengan N kelas. Sedangkan pendekatan *One Against One* (OAO) menghasilkan N(N-1)/2 *decision boundary* dengan pencarian *hyperplane* antara setiap kelas dan setiap kelas lainnya [16].

2.2.2 TF-IDF

TF-IDF merupakan metode gabungan dari Term Frequency (TF) dan Inverse Document Frequency (IDF), yaitu metode pembobotan pada fitur kata (*feature extraction*) berdasarkan perhitungan frekuensi kemunculannya yang menghasilkan nilai berbentuk numerik (vector) sehingga data dapat diproses oleh *machine learning*.

Nilai TF-IDF dihasilkan dari hasil perkalian antara TF dan IDF. Metode Term Frequency (TF) berfungsi untuk menghitung jumlah kemunculan suatu kata dalam dokumen, sedangkan metode Inverse Document Frequency (IDF) berfungsi untuk menentukan pengaruh dan seberapa penting kemunculan *term* tertentu dalam keseluruhan dokumen. Nilai TF-IDF menunjukkan bobot kata yang tinggi dihasilkan jika frekuensi kata tinggi dalam dokumen sementara frekuensi keseluruhan dokumen yang mengandung kata tersebut rendah. Adapun nilai TF, IDF, dan TF-IDF dapat diukur dengan persamaan 2.10, 2.11, dan 2.12 berikut [17].

$$tf_{t,d} = \frac{tf}{\max(tf)} \tag{2.10}$$

$$idf_t = \log\left(\frac{D}{df_t}\right) \tag{2.11}$$

$$W_{t,d} = tf_{t,d} \times idf_t \tag{2.12}$$

Variabel-variabel pada persamaan di atas dapat didefinisikan sebagai berikut:

- $tf_{t,d}$ = *Term frequency* (TF).
- tf = Jumlah kemunculan *term* dalam 1 dokumen yang sama.
- max(tf) = Total seluruh kata yang ada dalam 1 dokumen.
- idf_t = *Inverse Document Frequency* (IDF).
- D = Total dokumen secara keseluruhan (corpus).
- df_t = Jumlah dokumen yang mengandung *term* tertentu.
- $W_{t,d}$ = Bobot *term* dalam suatu dokumen.

2.2.3 Chi Square

Chi square merupakan metode statistika yang bekerja dengan melakukan pengujian perbandingan antara jumlah atau frekuensi observasi yang diperoleh dari penelitian dengan jumlah atau frekuensi yang diharapkan. Chi Square mampu mengevaluasi ketergantungan dan menentukan keterkaitan antar dua variabel. Pada perhitungan nilai chi square, semakin tinggi nilai chi square menandakan tingkat keterkaitan fitur yang tinggi (dependen) dimana menunjukkan bahwa fitur tersebut merupakan fitur yang penting dalam klasifikasi. Dengan demikian, data perhitungan chi square tiap fitur dapat diurutkan dan diseleksi seberapa banyak fitur penting yang dapat dipakai untuk proses klasifikasi [18]. Adapun nilai TF-IDF dapat diukur dengan persamaan 2.13 berikut [19].

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{2.13}$$

Variabel-variabel pada persamaan di atas dapat didefinisikan sebagai berikut:

- χ^2 adalah nilai statistik Chi-Square,
- O_i adalah frekuensi observasi untuk kategori atau sel ke-i,
- E_i adalah frekuensi yang diharapkan untuk kategori atau sel ke-i jika hipotesis nol benar.

Dalam uji chi square terdapat penggunaan tabel kontigensi yang digunakan untuk mengukur hubungan antara dua variabel. Tabel kontingensi, yang juga dikenal sebagai tabel i x j, menggambarkan elemen-elemen yang diklasifikasikan dalam i kategori yang berbeda dan sekaligus dalam j kategori yang berbeda.

2.2.4 Confussion Matrix

Confussion Matrix adalah suatu metode untuk mengevaluasi akurasi prediksi dari suatu hasil klasifikasi. Confussion matrix menyajikan perbandingan antara jumlah TP (*True Positive*), FN (*False Negative*), TN (*True Negative*), dan FP (*False Positive*) dengan rincian sebagai berikut [20]:

- TP adalah label data positif yang diprediksi dengan benar.
- FP adalah label data positif dengan prediksi salah.
- TN adalah label data negatif yang diprediksi dengan benar
- FN adalah label data negatif dengan prediksi salah.

Pada klasifikasi data dengan 3 label/*class* digunakan *multiclass* confussion matrix. Pada *multiclass* confussion matrix diperlukan penggabungan 2 kelas untuk menentukan nilai TN, FN, dan FP. Tabel 2.2, 2.3, dan 2.4 berikut menunjukkan *multiclass* confussion matrix pada masing-masing label data.

Tabel 2.2. Confusion Matrix pada Label Negatif

Class		**Predicted**		
		Negatif	Netral	Positif
Actual	Negatif	TP	FN	FN
	Netral	FP	TN	TN
	Positif	FP	TN	TN

Tabel 2.3. Confusion Matrix pada Label Netral

Class		**Predicted**		
		Negatif	Netral	Positif
Actual	Negatif	TN	FP	TN
	Netral	FN	TP	FN
	Positif	TN	FP	TN

Tabel 2.4. Confusion Matrix pada Label Positif

Class		**Predicted**		
		Negatif	Netral	Positif
Actual	Negatif	TN	TN	FP
	Netral	TN	TN	FP
	Positif	FN	FN	TP

Hasil confussion matrix dapat digunakan untuk perhitungan nilai *accuracy*, *precision*, *recall*, dan *f1-score*. *Accuracy* merupakan perhitungan untuk menggambarkan seberapa akurat model klasifikasi mengidentifikasi kelas dengan benar. *Precision* merupakan perhitungan untuk menggambarkan keberhasilan model mengidentifikasi kelas positif terhadap semua prediksi positif. *Recall* merupakan perhitungan untuk menggambarkan keberhasilan model mengidentifikasi kelas positif terhadap semua kelas yang bener-benar (*actual*) positif. *F1-score* merupakan perbandingan rata rata dari hasil *precision* dan *recall* yang telah dibobotkan. Rumus dalam menentukan *accuracy*, *precision*, *recall*, dan *f1-score* dalam confussion matrix dapat dilihat pada rumus 2.14, 2.15, 2.16, dan 2.17 berikut.

$$\text{Accuracy} = \frac{TP+TN}{\text{TotalData}} \tag{2.14}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.16}$$

$$\text{F1 Score} = 2x\frac{precissionxrecall}{precission + recall} \tag{2.17}$$