BAB 2
LANDASAN TEORI

2.1 Piala Asia U-23

Pertandingan Piala Asia U-23, diadakan oleh AFC, adalah kompetisi bagi pemain di bawah 23 tahun. Ini penting karena merupakan babak kualifikasi untuk Piala Dunia U-23, di mana tim-tim dari seluruh dunia berpartisipasi. Namun, perlu dicatat bahwa Piala Asia U-23 tidak terdaftar dalam kalender FIFA. Meskipun turnamen ini memiliki pengaruh besar pada perkembangan pemain muda di Asia, turnamen ini tidak diakui oleh FIFA sebagai bagian dari kalender internasional mereka, yang biasanya terdiri dari pertandingan persahabatan dan kualifikasi Piala Dunia [4].

2.2 Analisa Sentimen

Analisis sentimen adalah suatu proses di mana data teks dipahami, diekstrak, dan diproses untuk mengidentifikasi sentimen yang terkandung dalamnya [8]. Ini merupakan bagian dari Text mining, sebuah bidang penelitian yang berkaitan dengan analisis komputasional atas sentimen, emotikon, komentar, dan segala ekspresi yang disampaikan dalam bentuk teks. Analisis sentimen dapat dikategorikan menjadi dua jenis utama, yang pertama klasifikasi dokumen berdasarkan sifat opini maupun fakta, yang kedua klasifikasi dokumen dengan kategori negatif, positif, dan netral [12].

2.3 VADER

VADER adalah alat analisis sentimen berbasis leksikon dan aturan (lexicon- and rule-based) yang secara khusus disesuaikan dengan sentimen yang diungkapkan di media sosial. VADER dibuat berdasarkan corpus berbahasa Inggris maka dari itu diperlukan terjemahan sebelum menerapkan VADER. Klasifikasi sentimen dilakukan menggunakan submodul VADER yang terdapat pada modul nltk (Natural Languange Toolkit). Pemrosesan tweet menggunakan VADER akan menghasilkan skor negatif, skor netral, skor positif [13].

2.4 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan metode yang berfungsi sebagai pembobotan kata-kata berdasarkan peran dan maknanya dalam sebuah teks. Konsepnya menggabungkan dua aspek: frekuensi kata (*term frequency*) dan kebalikan frekuensi dokumen (*inversed document frequency*). *Term frequency* mengacu pada seberapa sering kata muncul dalam suatu dokumen, sementara *inversed document frequency* mengukur seberapa umum kata tersebut di seluruh dokumen. Frekuensi kata menunjukkan pentingnya kata dalam konteks dokumen tertentu, sementara frekuensi dokumen memberikan gambaran tentang seberapa umum kata tersebut. Ini menghasilkan hubungan yang kuat antara kata dan dokumen jika kata tersebut sering muncul dalam dokumen tertentu dan jarang muncul dalam seluruh kumpulan dokumen [14]. Penghitungan TF-IDF melibatkan rumus-rumus 2.1, 2.2, dan 2.3.

Rumus menghitung TF.

$$tf(t,d) = \frac{tf}{max(tf)} \tag{2.1}$$

Rumus menghitung IDF

$$idf_t = \log(\frac{D}{df_t}) \tag{2.2}$$

Rumus menghitung TF-IDF

$$W_{t,d} = tf(t,d) \times idf_t \tag{2.3}$$

Keterangan:

tf(t,d) = Term Frequency (TF).

$max(tf)$ = Jumlah keseluruhan kata di dalam dokumen.

tf = Term dengan nilai TF (Term Frequency) tertinggi dalam dokumen.

D = Jumlah keseluruhan yang tercantum dalam dokumen.

$idf(t)$ = Bobot kemunculan term t di seluruh dokumen.

$W_{t,d}$ = Bobot term dalam suatu dokumen

df_t = Jumlah dokumen yang mengandung term t

2.5 *Support Vector Machine*

Support Vector Machine (SVM) adalah metode pembelajaran yang memanfaatkan fungsi-fungsi linier di dalam ruang fitur berdimensi untuk melakukan klasifikasi, dan dilatih menggunakan algoritma pembelajaran yang berakar pada teori optimisasi. Konsep SVM pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai serangkaian konsep utama dalam pengenalan pola [15].

SVM mampu mengkategorikan data menjadi beberapa kelompok. Di ruang dua dimensi, garis lurus digunakan sebagai pemisah, di ruang tiga dimensi menggunakan bidang datar, dan di ruang dengan dimensi lebih dari tiga menggunakan hyperplane. Dalam beberapa skenario, bentuk hyperplane di ruang dua dimensi tidak selalu berupa garis lurus. Jika garis lurus digunakan sebagai pemisah, maka SVM dikategorikan sebagai linear. Sebaliknya, jika pemisahnya bukan garis lurus, maka SVM dikategorikan sebagai non-linear, seperti polynomial dan RBF [15].

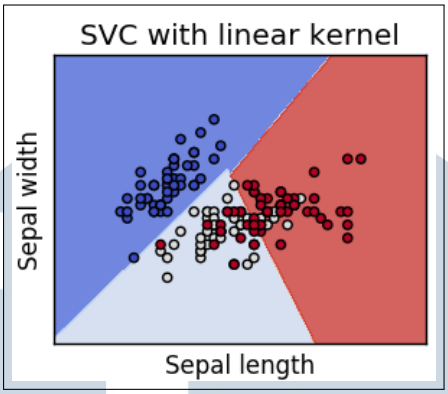
Dalam klasifikasi model dengan *Support Vector Machine (SVM)*, performa model dipengaruhi oleh beberapa parameter penting, seperti *gamma*, *cost (C)*, dan *kernel*. *Gamma* mengatur pengaruh sampel data latih. Nilai gamma rendah menunjukkan pengaruh jauh dan nilai tinggi menunjukkan pengaruh dekat.Dengan *Cost (C)* yang tepat, SVM dapat mencapai keseimbangan antara margin yang besar dan kesalahan klasifikasi yang rendah pada data latih. *Kernel* dalam SVM mentransformasikan data ke ruang dimensi lebih tinggi [16]. Beragam fungsi *kernel* tersedia untuk klasifikasi SVM, antara lain:

2.5.1 Kernel Linear

Pada Support Vector Machine (SVM), kernel linear berperan sebagai fungsi sederhana untuk menilai data yang terpisahkan secara linear [17]. Rumus matematis kernel linear ini tercantum dalam persamaan 2.4.

$$K(x, xi) = sum(x * xi) \tag{2.4}$$

Pada Penerapan *Support Vector Machine*, *xi* dan *x* mewakili data latih dan data uji secara berurutan. Garis pemisah kernel linear yang tergambar pada Gambar 2.1 berupa garis lurus.



Gambar 2.1. Kernel Linear

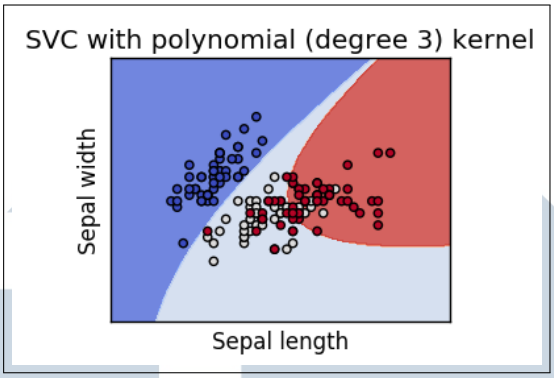
Sumber: [18]

2.5.2 Kernel Polynomial

Pada Support Vector Machine (SVM), Kernel polynomial adalah pilihan yang tepat saat garis pemisah antar kelas tidak linier. Dalam kerangka kernel polynomial, setiap vektor sampel pelatihan direpresentasikan pada ruang fitur yang sama. Penggunaan kernel polynomial dimungkinkan untuk penyelesaian permasalahan klasifikasi dalam dataset pelatihan yang sudah dinormalisasi [17]. Rumus kernel polynomial ini dapat ditemukan dalam bentuk persamaan 2.5.

$$K(x,xi) = 1 + sum(x * xi)^d \tag{2.5}$$

Pada persamaan polynomial, data pelatihan *xi*, data pengujian *x*, dan derajat (d) dicari untuk menghasilkan model yang akurat. Semakin tinggi nilai derajat, semakin tidak stabil akurasi yang dihasilkan. Ini disebabkan oleh peningkatan kelengkungan garis pemisah *hyperlane* yang digunakan. seperti terlihat pada Gambar 2.2 di mana garis pemisah menjadi semakin melengkung.



Gambar 2.2. Kernel Polynomial

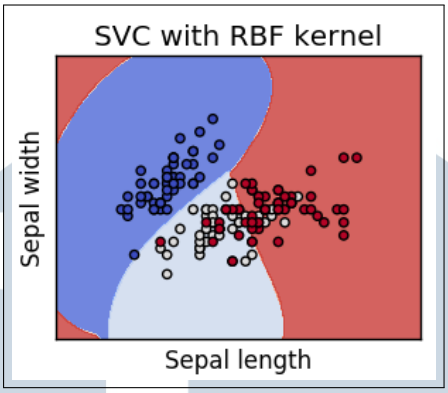
Sumber: [18]

2.5.3 Kernel RBF

Dalam Support Vector Machine (SVM), Kernel RBF atau yang dikenal sebagai kernel Gaussian, dipakai untuk mengklasifikasikan data yang tidak berbentuk linear. Kernel RBF dengan pengaturan parameter yang tepat, menunjukkan performa yang unggul dan menghasilkan tingkat error yang lebih rendah dibandingkan dengan kernel lain saat dilatih menggunakan dataset tertentu [17]. Rumus matematis kernel RBF tercantum dalam persamaan 2.6.

$$K(x,xi) = exp(-gamma * sum((x–xi^2)) \tag{2.6}$$

Pada metode Radial Basis Function (RBF), xi mewakili poin-poin dalam kumpulan data pelatihan, x mewakili data uji, dan parameter gamma mengendalikan seberapa besar pengaruh poin-poin data pelatihan terhadap prediksi. Pemisah keputusan RBF menghasilkan pola non-linear, seperti yang ditunjukkan dalam Gambar 2.3.



Gambar 2.3. Kernel RBF

Sumber: [18]

Untuk melatih model *Support Vector Machine*, kita menggunakan teknik Sequential Minimal Optimization (SMO) dengan memulai nilai α dari setiap titik data sama dengan 0. Langkah-langkah dalam pendekatan SMO adalah sebagai berikut:

1. Menghitung persamaan kernel yang dipilih
2. Menghitung matriks:

$$D_{ij} = Y_i Y_j (x_i x_j) + \lambda^2 \tag{2.7}$$

Keterangan:

D_{ij} = Elemen matriks ke ij.

Y_i = Kelas data ke-i.

Y_j = Kelas data ke-j.

α^2 = Batas teoritis yang diturunkan.

3. Menghitung nilai error

$$E_i = \sum_{j=1}^{n} a_j D_{ij} \tag{2.8}$$

Keterangan:

E_i = Nilai error data ke-i.

4. Menghitung delta a_i

$$\delta a_i = minmax[\gamma(1 - E_i) - a_i], C - a_i \tag{2.9}$$

Keterangan:

δa_i = Delta a ke-i.

γ = Gamma

C = Complexity

5. Menghitung a_i baru

$$a_i baru = a_i + \delta a_i \tag{2.10}$$

Lakukan langkah 3 hingga 5 berulang kali hingga batas iterasi tercapai

6. Menghitung nilai $w.x^+$ dan $w.x^-$ untuk menghasilkan nilai bias

$$w.x^+ = a_i Y_i K(w.x^+) w.x^- = a_i Y_i K(w.x^-) b = -1/2(w.x^+ + w.x^-) \tag{2.11}$$

Keterangan:

$w.x^+$ = Nilai kernel data x dengan data x positif yang memiliki nilai α tertinggi.

$w.x^-$ = Nilai kernel data x dengan data x negatif yang memiliki nilai α tertinggi. b = Nilai bias.

7. Menghitung nilai keputusan

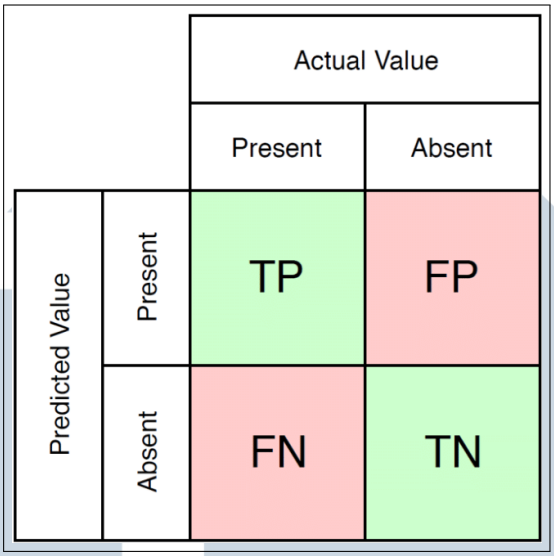
$$f(x) = \sum_{i=1}^{m} sign(a_i y_i K(x, x_i) + b) \tag{2.12}$$

Keterangan:

x = Titik data masukan SVM a_i = nilai bobot setiap titik data $K(x, x_i)$ = fungsi kernel b = nilai bias

2.6 *Confusion Matrix*

Confusion Matrix adalah alat bantu dalam machine learning untuk mengevaluasi performa klasifikasi dengan dua atau lebih kelas. Alat ini berbentuk tabel yang memuat empat kemungkinan kombinasi antara nilai prediksi dan nilai aktual. Contoh visualisasi Confusion Matrix dapat dilihat pada Gambar 2.4 [19].



Gambar 2.4. *Confusion Matrix*

Keterangan :

- TP (True Positif): Hasil prediksi menunjukkan positif dan ternyata memang benar.
- TN (True Negatif): Hasil prediksi menunjukkan negatif dan ternyata memang benar.
- FP (False Positif): Hasil prediksi menunjukkan positif tapi ternyata salah.
- FN (False Negatif): Hasil prediksi menunjukkan negatif tapi ternyata salah.

Penggunaan *Confusion Matrix* memiliki beberapa kesamaan dalam menghitung *accuracy, precision, recall,* dan *F-score*.

2.6.1 *Accuracy*

Sebagai indikator performa model klasifikasi, *Accuracy* dalam Confusion Matrix adalah sebuah metrik yang mengukur sejauh mana model klasifikasi dapat memprediksi dengan tepat. Cara menghitung *accuracy conclusion matrix* bisa ditemukan di rumus 2.13.

$$Accuracy = \frac{TP + TN}{TotalData} \tag{2.13}$$

2.6.2 *Precision*

Precision dalam konteks ini merujuk pada metrik yang mengukur sejauh mana hasil klasifikasi sesuai dengan data yang sebenarnya. Formula untuk menghitung *precision* dapat ditemukan dalam persamaan 2.14.

$$Precision = \frac{TP}{TP + FP} \tag{2.14}$$

2.6.3 *Recall*

Recall adalah istilah yang digunakan untuk menggambarkan kemampuan model dalam mengingat kembali informasi yang telah dipelajari. Rumus 2.15 mencakup perhitungan untuk *recall*.

$$Recall = \frac{TP}{TP + FN} \tag{2.15}$$

2.6.4 *F1-score*

F1-score adalah nilai tengah antara *precision* dan *recall*, yang telah diberi bobot. Rumus untuk menghitung *F1-score* dapat ditemukan pada rumus 2.16.

$$F1-Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.16}$$