BAB 2
LANDASAN TEORI

2.1 Tweet Harvest (Twitter Crawler)

Tweet Harvest merupakan sebuah *command-line tool* yang menggunakan Playwright untuk mendapatkan *tweets* dari hasil pencarian Twitter berdasarkan kata kunci dan rentang tanggal yang ditentukan. *Tweets* yang berhasil didapatkan kemudian akan disimpan dalam *file* berbentuk CSV. Untuk menggunakan Tweet Harvest dibutuhkan *authorization token* yang bisa didapatkan dengan *login* ke akun Twitter di browser anda kemudian mengekstrak *authorization token*.

2.2 Analisis Sentimen

Analisis sentimen merupakan sebuah studi yang bertujuan untuk menganalisis opini, perasaan, dan emosi yang terkandung dalam suatu dokumen atau dataset. Fokus utama dari analisis sentimen adalah untuk mengklasifikasikan karakteristik teks yang terdapat dalam sebuah kalimat atau opini menjadi dua kategori utama, yaitu positif dan negatif. Pertumbuhan dan dampak yang signifikan dari analisis sentimen telah mendorong perkembangan penelitian dan aplikasi berbasis analisis sentimen [9].

2.3 TF-IDF (Term Frequency-Inverse Document Frequency)

Term Frequency-Inverse Document Frequency yang biasa dikenal sebagai TF-IDF merupakan suatu metode algoritma yang memberikan bobot terhadap teks. Konsep ini menggabungkan frekuensi kata dalam dokumen (TF) dan nilai invers dari dokumen yang mengandung kata tersebut (IDF). Bobot dari kata tersebut dihasilkan dengan mengkalikan nilai TF dan IDF [10]. Persamaan TF-IDF dapat dilihat pada Persamaan 2.1.

$$TF-IDF(d,t) = TD(d,t) * IDF(t) \tag{2.1}$$

Dimana:

$$TF(d,t) = \frac{\text{jumlah kata t pada dokumen d}}{\text{total kata pada dokumen d}} \tag{2.2}$$

$$IDF(t) = log \frac{\text{total dokumen}}{\text{jumlah dokumen yang mengandung kata t}} \quad (2.3)$$

Keterangan:

t = kata

d = dokumen

2.4 Naive Bayes Classifier

Naive Bayes merupakan metode pengklasifikasian yang didasari oleh sebuah metode untuk memprediksi peluang dimasa depan berdasarkan pengalaman di masa sebelumnya yang pertama kali dikemukakan oleh Thomas Bayes dan kemudian dikenal sebagai Teorema Bayes [11]. Algoritma mengunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas [12]. *Naive Bayes* menggunakan metode klasifikasi statistik untuk memprediksi probabilitas keanggotaan suatu kelas. Metode klasifikasi ini didasarkan pada teorema Bayes yang memiliki kemampuan klasifikasi serupa dengan *decision tree* dan *neural network*. Implementasi klasifikasi ini telah terbukti memiliki akurasi tinggi dan kinerja yang cepat ketika digunakan dalam *database* dengan *dataset* yang besar [13].

2.4.1 Multinomial Naive Bayes

Multinomial adalah salah satu implementasi dari algoritma *naive Bayes* yang digunakan untuk data yang memiliki distribusi multinomial. Ini adalah salah satu variasi klasik dari *naive Bayes* yang sering digunakan dalam klasifikasi teks, di mana data sering direpresentasikan sebagai vektor kata yang menghitung jumlah kemunculan kata-kata. Distribusi multinomial mengacu pada vektor x untuk setiap kelas y, di mana jumlah fitur dalam klasifikasi teks dan ukuran kosa kata digunakan untuk menggambarkan probabilitas P(x—y) dari fitur i yang muncul dalam sampel dari kelas y. [14]. Multinomial Naive Bayes digunakan untuk mengklasifikasi kategori dokumen, dokumen dapat bertema olahraga, politik, teknologi, atau lain-lain berdasarkan frekuensi kata-kata yang muncul dalam dokumen [15]. Untuk menghitung frekuensi relatif dalam metode multinomial ini, kita dapat menggunakan Rumus 2.4.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_{yi} + \alpha n} \tag{2.4}$$

2.4.2 Gaussian Naive Bayes

Gaussian Naive Bayes digunakan untuk menghitung probabilitas suatu data kontinu terhadap kelas yang spesifik. Perhitungan dilakukan dengan menggunakan rumus Densitas Gauss dan ditandai dengan dua parameter, yaitu rata-rata dan standar deviasi [16]. Rumus Algoritma Gaussian dapat dilihat pada Rumus 2.5.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}) \tag{2.5}$$

2.4.3 Bernoulli Naive Bayes

Bernoulli merupakan implementasi dari pelatihan dan algoritma klasifikasi *Naive Bayes* yang digunakan untuk data yang mengikuti distribusi Bernoulli multivariat. Dalam distribusi ini, beberapa fitur diasumsikan sebagai variabel biner. Oleh karena itu, kelas Bernoulli membutuhkan representasi sampel sebagai vektor fitur biner. Jika diberikan jenis data yang berbeda, turunan Bernoulli dapat mengonversinya menjadi input biner. [14]. Aturan keputusan untuk Bernoulli naive Bayes didasarkan pada Rumus 2.6.

$$P(x_I|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_y) \tag{2.6}$$

2.5 Support Vector Machine

Support Vector Machine (SVM) adalah algoritma machine learning yang menggunakan fungsi *hyperplane* untuk memisahkan data ke dalam daerah-daerah yang mewakili masing-masing kelas. *Hyperplane* adalah fungsi yang digunakan untuk memisahkan antara kelas-kelas yang ada. Dalam proses prediksi kelas suatu data, SVM akan memberikan label berdasarkan daerah kelas mana data tersebut berada [10]. SVM berusaha untuk mendapatkan *hyperplane* terbaik untuk memisahkan kedua kelas dan memaksimalkan margin antara kelas-kelas tersebut. Pencarian *hyperplane* terbaik sebagai pemisah antar kelas merupakan inti dari metode SVM.

Terdapat beberapa parameter pada klasifikasi model menggunakan *Support Vector Machine* yang berguna untuk meningkatkan kinerja dari pemodelan yaitu *gamma*, *cost* (C), dan kernel. Parameter *gamma* merupakan parameter yang menentukan seberapa jauh pengaruh dari sampel *dataset* yang dilatih, nilai rendah pada parameter *gamma* berarti jauh sedangkan nilai tinggi berarti dekat. Parameter *cost* (C) merupakan parameter yang digunakan sebagai pengoptimalan metode SVM untuk menghindari kesalahan dalam klasifikasi pada data yang dilatih. Kernel Dalam algoritma *support vector machine* berguna untuk mentranformasikan data ke ruang dimensi tinggi [17].

2.5.1 Kernel Linear

Kernel linear adalah fungsi kernel yang sangat sederhana dalam penggunaannya. Fungsi ini ideal digunakan ketika data yang akan dianalisis dapat dipisahkan secara linier dalam ruang fitur. Kelebihan penggunaan kernel linear terutama terlihat pada data dengan banyak fitur, karena memetakan data ke ruang dimensi yang lebih tinggi tidak memberikan peningkatan kinerja yang signifikan [18]. Persamaan fungsi kernel linear dapat dilihat pada Persamaan 2.7.

$$K(x,xi) = sum(x * xi) \tag{2.7}$$

2.5.2 Kernel Polynomial

Fungsi *kernel polynomial* digunakan saat data tidak dapat dipisahkan secara linear. Fungsi ini merupakan bentuk yang lebih umum dari fungsi kernel linear. Dalam pembelajaran mesin, fungsi *kernel polynomial* digunakan dalam algoritma seperti *Support Vector Machines* (SVM) untuk mengukur kemiripan antara vektor sampel pelatihan dalam ruang fitur. Fungsi *kernel polynomial* terutama efektif dalam menyelesaikan masalah klasifikasi pada dataset *training* yang telah dilakukan normalisasi.[18]. Persamaan fungsi *kernel polynomial* dapat dilihat pada Persamaan 2.8.

$$K(x,xi) = 1 + sum(x * xi)^d \tag{2.8}$$

Kernel polynomial memiliki parameter derajat (d) yang digunakan untuk mencari nilai optimal pada setiap dataset yang digunakan. Parameter d menunjukkan derajat dari fungsi kernel polynomial, dengan nilai default d = 2.

Ketika nilai d semakin besar, sistem dapat menghasilkan akurasi yang fluktuatif dan kurang stabil. Hal ini disebabkan oleh tingginya nilai parameter d yang menghasilkan garis hyperplane yang semakin melengkung.

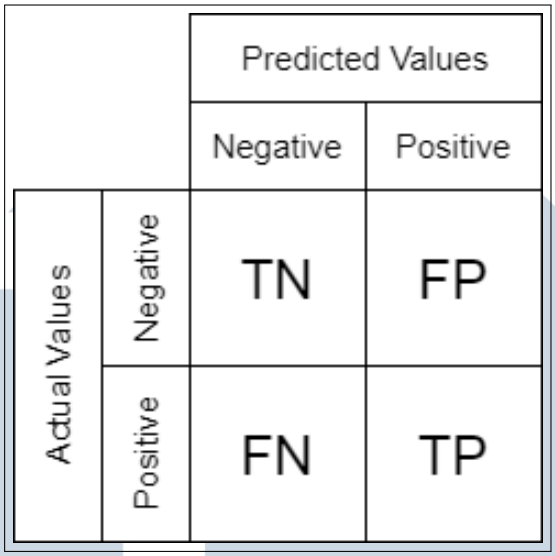
2.5.3 Kenel RBF

Fungsi kernel RBF (Radial Basis Function) digunakan dalam klasifikasi data yang tidak dapat dipisahkan secara linear. Fungsi kernel ini, juga dikenal sebagai kernel Gaussian, merupakan salah satu konsep kernel yang paling umum digunakan dalam pemecahan masalah tersebut. Kelebihan dari kernel RBF adalah kinerjanya yang baik dengan parameter yang tepat, serta menghasilkan model pelatihan dengan tingkat kesalahan yang rendah dibandingkan dengan kernel lainnya [18]. Persamaan fungsi kernel RBF dapat dilihat pada Persamaan 2.9.

$$K(x,xi) = exp(-gamma * sum((x–xi^2)) \tag{2.9}$$

2.6 Confusion Matrix

Confusion Matrix adalah metode yang digunakan untuk menghitung akurasi pada konsep dalam data mining. Evaluasi menggunakan *Confusion Matrix* menghasilkan nilai akurasi (accuracy), presisi (precision), dan recall. *Accuracy* dalam klasifikasi dalam data mining adalah persentase kebenaran dalam mengklasifikasikan data yang telah diuji. *Precision* adalah proporsi kasus yang diprediksi positif yang sebenarnya juga positif. *Recall* adalah proporsi kasus positif yang diprediksi dengan benar[19]. Confusion Matrix memiliki tabel dari empat kombinasi berbeda dari hasil nilai prediksi dan nilai aktual, yaitu *true positive*, *false positive*, *true negative* dan *false negative* [20]. *Confusion matrix* dapat dilihat pada Gambar 2.1



Gambar 2.1. Confusion Matrix

Sumber: [20]

Keterangan :

- TP (*True Positive*): Prediksi yang dibuat positif dan benar.
  Contoh: Diprediksi bahwa seorang wanita sedang hamil dan kenyataannya memang benar wanita tersebut hamil.
- TN (*True Negative*): Prediksi yang dibuat negatif dan benar
  Contoh: Diprediksi bahwa seorang pria tidak hamil dan kenyataannya memang benar bahwa pria tersebut tidak hamil.
- FP (*False Positive*): Prediksi yang dibuat positif dan salah
  Contoh: Diprediksi bahwa seorang pria sedang hamil, tetapi kenyataannya pria tersebut tidak hamil.
- FN (*False Negative*): Prediksi yang dibuat negatif dan salah
  Contoh: Diprediksi bahwa seorang wanita tidak hamil, tetapi kenyataannya wanita tersebut sedang hamil.

Terdapat beberapa rumus dalam perhitungannya yang digunakan untuk menghitung *accuracy, precision, recall dan F-score*.

2.6.1 Accuracy

Accuracy merupakan perbandingan antara jumlah data yang terklasifikasi dengan benar dan jumlah total data yang ada. Rumus *accuracy* dapat dilihat pada

Rumus 2.10.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

2.6.2 Precision

Precision merupakan perbandingan antara jumlah prediksi positif yang benar dengan jumlah keseluruhan hasil prediksi positif. Rumus *precision* dapat dilihat pada Rumus 2.11.

$$Precision = \frac{TP}{TP + FP} \tag{2.11}$$

2.6.3 Recall

Recall merupakan perbandingan antara jumlah prediksi positif yang benar dengan jumlah keseluruhan data yang sebenarnya positif. Rumus *recall* dapat dilihat pada Rumus 2.12.

$$Recall = \frac{TP}{TP + FN} \tag{2.12}$$

2.6.4 F1-Score

F1-Score merupakan perbandingan rata-rata presisi dan recall yang dibobotkan. Rumus *F1-Score* dapat dilihat pada Rumus 2.13.

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.13}$$