BAB 2
LANDASAN TEORI

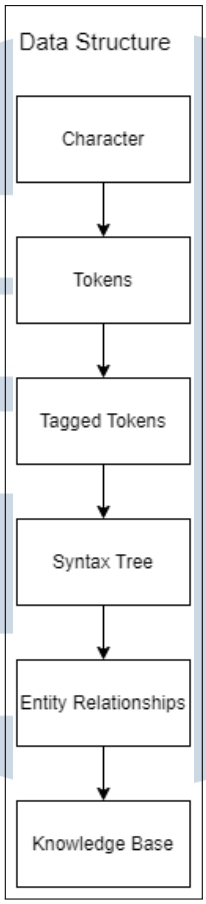
2.1 Machine Learning

Machine learning atau pembelajaran mesin adalah sebuah teknik yang dimana digunakan untuk melakukan interferensi pada data dengan pendekatan matematis. *Machine Learning* ini digunakan untuk membuat model - model untuk mereflesikan pola data sehingga data tersebut dapat digunakan untuk mengklasifikasikan atau memprediksi data untuk membuat atau mendukung dalam pengambilan keputusan [10]. Dari semua machine learning ini terdapat juga beberapa pembelajarannya, pembelajaran machine learning adalah *supervised learning, unsupervised learning* dan *reinforcement learning*. *Supervised learning* adalah teknik machine learning yang dimana pembelajarannya menggunakan dataset yang dimana merupakan data labeled response, untuk *Supervised learning* terdapat beberapa algoritma yang digunakan yaitu K-*Nearest Neighbour* (KNN), Naive Bayes, dan regresi. Sedangkan, *unsupervised learning* adalah teknik machine learning yang menarik dari dataset sehingga mendapatkan kesimpulan dari dataset tersebut. Algoritma yang digunakan dalam *unsupervised learning* adalah *Fuzzy, K-Means,* dan *Self Organizing Map* [11]. Terdapat juga *reinforcement learning* yang merupakan bagian dari *artificiall intelligence* yang dimana merupakan pengambilan keputusan untuk menyelesaikan masalah, selain itu juga *reinforcement learning* ini terus berlatih dengan *trial* dan *error* yang dilakukan dalam pembelajaran sehingga dapat memilih keputusan yang terbaik dalam masalah yang diberikan [12].

2.2 Natural Language Processing

Natural language processing adalah teknologi yang menggunakan ilmu komputer dalam hal linguistik komputasional sehingga digunakan untuk mengkaji interaksi antara bahasa manusia dengan bahasa komputer [13]. Terdapat juga beberapa tugas yang dapat dilakukan dalam penerapan *natural languange processing* yaitu sentiman teks, pengenalan suara dan tanggapan dalam pertanyaan. Dari hal ini terdapat juga pemahaman tentang *natural languange understanding* yang dimana merupakan sub-bidang yang merupakan reprentrasi dari bahasa alami ke bahasa formal [14]. Contoh beberapa aplikasi nlp yang ada di Indonesia yaitu penggunaan google translate, Chatbot Tokopedia, Deteksi Spam Email, dan Voice

Recognition.



Gambar 2.1. NLP Pipeline

sumber : [14]

Pada Gambar 2.1 dapat terlihat 6 proses terkait *data structure* untuk NLP Pipeline. Pada proses pertama yaitu perubahan character ke tokens dengan menggunakan algorithm *reguler expression* yang selanjutkan dilakukan perubahan token ke tagged tokens dengan menggunakan algorithma *Part of Speech tagger* atau *Finite State Automaton*. Selain itu juga terdapat pos tags yang langsung dilakukan secara otomatis dengan pipeline tersebut. Setelah mencampai syntax tree maka dilanjutkan ke *entity relationship* dan terakhir adalah *knowledge base*. Pada 2 tahap layer terakhir ini maka akan dilakukan pengumpulan data tentang domain tertentu sehingga dapat mengumpulkan informasi dan menentukan kesimpulan. Kesimpulan - kesimpulan ini dapat digunakan untuk membuat pilihan yang logis. Selain itu juga tanpa 2 layer yang dibawah yaitu *entity relationship* dan *knowledge base* juga dapat dilakukan pembuat keputusan tetapi apabila keputusan itu dilanjutkan maka akan didapatkan hasil perilaku yang mirip dengan manusia.

2.3 Text Preprocessing

text preprocessing adalah sebuah langkah yang penting untuk melakukan proses dalam *text mining*, *natural language processing*, dan *information retrieval*. Selain itu pembangunan *text prerprocessing* ini berguna untuk mengambil data yang penting dari data yang tidak terstruktur. Selain itu juga terdapat 4 tahap dalam *text prerpocessing*, yaitu *case folding*, tokenisasi, *stopword removal*, dan *stemming* [15].

1. Case Folding
   Case folding adalah proses dalam *text preprocessing* yang mengubah kata huruf besar menjadi huruf kecil. Selain itu juga pada proses *case folding* ini akan juga dihapuskan tanda baca atau angka - angka yang tidak diperlukan dalam membaca teks. Salah satu contoh case folding adalah kata ”APAKAH” menjadi kata yaitu ”apakah” [16].

2. Tokenisasi
   Tokenisasi adalah proses dalam *text prerprocessing* yang mengubah kalimat - kalimat menjadi kata sendiri atau menjadikan kalimat tersebut menjadi beberapa token. Selain itu pada bagian ini dilakukan dengan cara pembagian berdasarkan spasi sehingga mendapatkan token dan juga pada tahap ini dilakukan pembuangan tanda baca [16].

3. Stopword Removal
   Stopword removal adalah proses dalam *text preprocessing* yang menghapus kata - kata umum atau kata - kata yang sering muncul. Kata - kata yang sering muncul ini salah satunya yang dapat dihapus adalah kata ”dan” [16].

4. Stemming
   Stemming adalah proses dalam *text preprocessing* yang menghilangkan imbuhan kata sehingga kata - kata tersebut dapat kembali ke kata dasarnya. Salah satu contoh dari kata imbuhan yang kembali ke dasarnya adalah kata ”besokan” menjadi kata ”besok” [16].

5. Text Normalization
   Text normalization adalah proses dalam *text preprocessing* yang mengubah teks menjadi format tertentu seperti menghilangkan angka dari teks sehingga memenuhi tujuan dari penelitian. Salah satu contohnya adalah ”vol 5” sehingga menjadi ”vol”[17].

2.4 N-Gram

N-Gram adalah urutan dari n unit yang dimana umumnya berupa karakter tunggal atau *string* yang dipisahkan oleh spasi. N-gram adalah potongan n-kata yang diambil dari teks [18]. Selain itu juga terdapat N-gram yang digunakan dalam penelitian ini yaitu unigram atau satu gram yang dimana token terdiri dari satu kata dan bigram atau dua gram yang merupakan token yang terdiri dari dua kata [19]. Untuk pengambilan kata - kata ini menggunakan blank yang dimana merupakan spasi pada kata. Berikut ini adalah contoh tersebut:

Teks : Hari Ini
Unigram : Hari, ini
Bigram : Hari Ini

Selain itu juga dari n-gram ini dapat digunakan untuk membandingkan string yang satu dengan string yang lainnya dalam hal penelitian ini yaitu perbandingan dengan menggunakan algoritma *cosine similarity*.

2.5 Jarak

Pada algoritma yang akan digunakan yaitu algoritma *cosine similarity* ini akan dilakukan pengukuran jarak antara dua vektor. Vektor - vektor ini merupakan dokumen - dokumen yang akan dimasukkan datanya ke dalam algoritma *cosine similarity*. Maka dari itu jarak dalam algoritma *cosine similarity* ini memiliki arti kesamaan atau kemiripan dari dokumen yang pertama dengan dokumen selanjutnya. Dari kemiripan itu maka dapat diketahui beberapa tingkat akurasi dari kesamaan suatu dokumen [20].

2.6 Streamlit

Streamlit adalah sebuah kerangka web yang ditunjukkan untuk penyebaran model dan visualisasi pada python. Streamlit ini digunakan karena banyak sekali widget dan juga dari widget - widget yang digunakan ini merupakan widget yang mudah dipahami sehingga pengguna dapat menggunakanya dengan ramah. Selain itu juga streamlit ini tidak perlu memahami *front-end* secara mendalam karena penggunaanya yang gampang dan mudah sehingga dapat dioperasikan secara gampang [21].

2.7 Cosine Similarity

cosine similarity adalah suatu ukuran kemiripan yang berguna untuk mencari informasi dan mengukur sudut vektor. Selain itu kemiripan ini biasa diukur bedasarkan kemiripan antara kata dengan kata lain [22]. *cosine similarity* ini nantinya akan digunakan dalam *natural languange processing* sehingga dapat mengecek kesalahan ejaan kata pada kata keterangan waktu yang akan diteliti.

Berikut ini adalah rumus algoritma *cosine similarity*.

$$\cos \alpha = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{2.1}$$

Keterangan:

A = Vektor A, yang akan dibandingkan kemiripannya

B = Vektor B, yang akan dibandingkan kemiripannya

A • B = dot product antara vektor A dan vektor B

| A | = panjang vektor A

| B | = panjang vektor B

| A | | B | = cross product antara | A | dan | B | [23]

2.8 Confusion Matrix

Confusion Matrix adalah metode yang digunakan untuk pengukuran kinerja dari klasifikasi. Hasil klasifikasi ini dibandingkan dari hasil klasifikasi sistem dengan klasifikasi yang seharusnya, Terdapat empat klasifikasi untuk representasi hasil yaitu *True Positive* (TP), *True Negative* (TN), *False Positif* (FP) dan *False negative* (FN). Nilai tersebut yaitu *True Negative* adalah nilai negatif yang dideteksi benar sedangkan *False Positive* adalah nilai negatif namun terdeteksi bahwa nilainya positif. Terdapat juga beberapa nilai lainnya yaitu *True Positive* adalah nilai positif yang dideteksi bahwa nilainya juga positif sedangkan False Negatif adalah nilai negatif yang terdeteksi bahwa nilainya negatif [24].

Tabel 2.1. Table confusion Matrix.

	Predicted Negative	Predicted Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Berikut ini adalah rumus *confusion matrix* yang dimana untuk mengukur performa dari data tersebut.

1. Accuracy adalah total dari keseluruhan data yang sering benar klasifikasinya. Berikut ini adalah rumusnya [25].

$$Accuracy = \frac{TP + TN}{Total} \tag{2.2}$$

2. Precision adalah data yang diprediksi secara positif sehingga mengetahui berapa kali data tersebut benar. Berikut ini adalah rumusnya [25].

$$Precision = \frac{TP}{FP + TP} \tag{2.3}$$

3. Recall adalah kelas yang datanya positif dan seberapa sering data tersebut positif. Berikut ini adalah rumusnya [25].

$$Recall = \frac{TP}{FN + TP} \tag{2.4}$$

4. F1-Score adalah rata - rata dari *precision* dan *recall*. Berikut ini adalah rumusnya [25].

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \tag{2.5}$$