BAB II

LANDASAN TEORI

2.1 Data Mining

Menurut (Aji Ghassa, dkk., 2022), "*Data mining* dikenal sebagai *Knowledge Discovery in Database (KDD)*. *Data mining* dijelaskan sebagai suatu proses menemukan relasi atau pola yang bermakna dengan mengamati data yang banyak, dimana data tersebut disimpan dalam penyimpanan dan mengelolanya dengan menggunakan teknik dan metode statistik kumpulan data dari ukuran data yang sangat tinggi dan bervariasi."

2.2 Python

Menurut (Dianati Duei Putri, dkk., 2022), "*Python* dijelaskan sebagai bahasa pemrograman tingkat tinggi yang mendukung pemrograman berorientasi objek. Dalam bahasa pemrograman *Python*, terdapat berbagai perpustakaan dan kerangka kerja yang digunakan untuk melakukan analisis data. *Python* berbeda dengan bahasa pemrograman lain terutama dalam penulisan sintaksisnya."

2.3 Jupyter Lab

Menurut (Vindua Raditia, dkk., 2023), "*Jupyter Lab* dikenal sebagai *Jupyter*, merupakan evolusi dari *IPython (Interactive Python)*. *Jupyter Lab* sebagai *editor* untuk aplikasi web yang berjalan di server lokal. *Jupyter Lab* dapat difungsikan untuk melakukan *coding* dengan menggunakan bahasa *Python*."

2.4 Sentiment Analysis

Menurut (Cholid Fadilah Hasri, dkk., 2022), "*Sentiment Analysis (opinion mining)* dijelaskan sebagai bidang ilmu yang mengungkapkan pendapat atau emosi dari suatu teks atau sentimen. Salah satu topik dalam *Sentiment Analysis* yang sering dipelajari ialah klasifikasi sentimen, berfokus pada pengelompokan pendapat dari opini yang membahas isu-isu menarik."

2.5 CRISP-DM

Menurut (Rhini Fatmasari, dkk., 2022), "*CRISP-DM (Cross-Industry Standard Process for Data Mining)* dijelaskan sebagai standar dari proses untuk penambangan data, metode ini banyak dipakai karena penerapanya yang efektif, serta memiliki langkah-langkah yang mudah diterapkan, selain itu juga sebagai strategi pemecahan masalah umum dari bisnis ataupun penelitian. *CRISP-DM* terdiri dari enam tahap, yaitu :

1. *Business Understanding*, dijelaskan sebagai tahap pertama. Tahap ini diperlukan pemahaman tentang objek bisnis yang diteliti, seperti permasalahan yang terjadi dan informasi akhir yang ingin didapatkan.
2. *Data Understanding*, dijelaskan sebagai tahap kedua. Bertujuan untuk mendapatkan, memverifikasi, serta memperbaiki data agar tidak terdapat data yang buruk atau tidak jelas, sehingga nantinya dapat diolah dan menghasilkan informasi yang baik dan jelas.
3. *Data Preparation*, dijelaskan sebagai tahap ketiga. Tahap ini dilakukan manipulasi data untuk memperbaiki setiap masalah yang ditemukan dalam data, diperlukan pemahaman dan algoritma yang tepat agar mendapatkan hasil yang tepat pula.

4. *Modeling*, dijelaskan sebagai tahap keempat. Tahap ini data yang telah dilakukan perbaikan dan manipulasi dengan algoritma tertentu, selanjutnya diterapkan metode perhitungan atau statistika dengan suatu model *Machine Learning.*
5. *Evaluation*, dijelaskan sebagai tahap kelima. Tahap ini menampilkan hasil evaluasi dari klasifikasi yang dilakukan, bertujuan untuk mengetahui kelebihan dan kekurangan dari model yang digunakan pada data.
6. *Deployment*, dijelaskan sebagai tahap keenam. Tahap ini mengimplementasikan analisis yang telah dilakukan dari tahap pertama hingga tahap kelima dari *CRISP-DM* yang telah dilakukan. Bertujuan menampilkan semua proses agar bisa dilihat oleh pengguna dalam sebuah aplikasi atau web."

2.6 Text Mining

Menurut (Ibnu Afdhal, dkk., 2022), "*Text Mining* dijelaskan sebagai alur menambang data atau dokumen yang berbentuk teks. Dari teks atau dokumen tersebut dicari hubungannya, sehingga dapat ditemukan informasi atau kata-kata yang dapat diungkap keterkaitannya."

2.7 Text Preprocessing

Menurut (Ibnu Afdhal, dkk., 2022), "*Text Preprocessing* dijelaskan sebagai alur dalam membersihkan data sebelum diolah. Terdapat 5 alur penting, yaitu:

1. *Cleaning*, membuang teks yang berisi *noise* (angka, tanda baca, emoji, spasi ganda dan baris enter).

2. *Case Folding*, melakukan penyeragaman teks menjadi bentuk huruf kecil (*lowercase*).
3. *Tokenizing*, melakukan pemisahan atau pemecahan kata pada kalimat.
4. *Stopword Removal*, melakukan penghilangan kata yang termasuk kedalam bentuk *stopword*. *Stopword* dijelaskan sebagai kata yang seringkali muncul namun dianggap tidak bermakna.
5. *Stemming*, dilakukan untuk mendapatkan kata dasar dengan memotong imbuhan yang menyatu pada kata."

2.8 Pelabelan Data (*Labeling*)

Menurut (Brata Mas Pintoko, dkk., 2018), "*Labeling* dilakukan dengan cara manual oleh peneliti. Peneliti melakukan *labeling* manual dengan menentukan data dengan kelas positif atau kelas negatif. Kelas positif terlihat dari isi ulasan yang berkonotasi positif, seperti dukungan dan pernyataan setuju. Kelas negatif terlihat dari isi ulasan yang berkonotasi negatif, seperti cibiran dan cemoohan. Sehingga, pelabelan data atau *labeling* (kelas positif dan kelas negatif) dilakukan dengan cara manual oleh 3 *reviewer* berdasarkan ulasan."

2.9 Pembobotan Kata (*TF-IDF*)

Menurut (Hana Chyntia Morana, dkk., 2022), "Dijelaskan bahwa pembobotan kata (*term*) bertujuan untuk memberikan bobot pada setiap kata (*term*) yang terdapat pada teks."

Menurut (Ibnu Afdhal, dkk., 2022), "Data harus berbentuk numerik (angka) supaya dapat masuk kedalam proses klasifikasi. Data tersebut bisa diubah bentuknya menjadi numerik menggunakan pembobotan kata (*TF-IDF)*. Nilai *TF-*

IDF dari sebuah kata merupakan kombinasi dari nilai TF dan nilai IDF dalam perhitungan bobot. *TF (term frequency)* dijelaskan sebagai nilai frekuensi kata dalam sebuah dokumen, sedangkan *IDF (inverse document frequency)* merupakan nilai kebalikan dari kata dalam dokumen."

2.10 Random Forest

Menurut (Hana Chyntia Morana, dkk., 2022), "*Random Forest* dijelaskan sebagai algoritma klasifikasi yang menghasilkan lebih dari satu pohon keputusan *(decision tree)* menggunakan *subset* sampel dan variabel data latih yang dipilih secara *random* yang dapat meningkatkan akurasi. Alur dalam membuat pohon keputusan, yaitu :

1. Menentukan jumlah dari *decision tree*, sebaiknya jumlahnya ganjil untuk menghindari jumlah prediksi sama.
2. Menghitung *entropy*.
3. Menghitung *information gain* pada setiap *spliting*.
4. Memilih *root* dari nilai *information gain* tertinggi.
5. Mengulang alur ke-2 hingga ke-4 sampai k-*tree* (seluruh pohon)."

Menurut (Farahdiva Assyifa Andrin, 2021), "Pohon keputusan dijelaskan sebagai struktur yang membagi kumpulan data besar menjadi kumpulan data yang lebih kecil sehingga algoritma dapat diterapkan. Dalam pohon keputusan, untuk mengetahui propertinya, kita harus mempusatkannya pada nilai *gain* tertinggi dari atribut itu."

Berikut ini cara membuat pohon keputusan.

$$Gain\ (S, A) = \ Entropy(S) - \sum_{i=1}^{n} * Entropy(Si)$$

Keterangan :

S : himpunan kasus

A : atribut

N : jumlah partisi atribut *A*

|*Si*| : jumlah kasus pada partisi ke-*i*

|*S*| : jumlah kasus dalam *S*

Berikut rumus untuk mengetahui nilai *entropy* :

$$Entropy(S) = \sum_{i=1}^{n} -pi * log2pi$$

Keterangan :

S : himpunan kasus

A : fitur

N : jumlah partisi *S*

Pi : proporsi dari *Si* terhadap *S*

Pada umumnya pohon keputusan dibangun seperti berikut :

1. Memilih atribut sebagai akar
2. Membuat cabang untuk tiap-tiap nilai
3. Bagi kasus dalam cabang

4. Mengulang proses untuk setiap cabang sehingga semua kasus pada cabang memilih kelas yang sama.

2.11 Confusion Matrix

Menurut (Ibnu Afdhal, dkk., 2022), "*Confusion matrix* dijelaskan sebagai matriks yang berisi informasi dari nilai-nilai yang diprediksi oleh model, biasa digunakan untuk menghitung *accuracy*, *recall*, *precision*, serta *F1-score*. *Confusion matrix* divisualkan dengan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan.

Tabel 2. 1 *Confusion Matrix*

Actual Label	*Predicted Label*	
	Positive (+)	*Negative (-)*
Positive (+)	*True Positives (TP)*	*False Negatives (FN)*
Negative (-)	*False Positives (FP)*	*True Negatives (TN)*

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

Berikut adalah keterangan dari tabel *confusion matrix* :

1. *True Positives (TP)*, yaitu jumlah data positif yang diklasifikasikan sebagai nilai positif oleh model.
2. *False Positives (FP)*, yaitu jumlah data negatif yang diklasifikasikan sebagai nilai positif oleh model.

3. *False Negatives (FN)*, yaitu jumlah data positif yang diklasifikasikan sebagai nilai negatif oleh model.
4. *True Negatives (TN)*, yaitu jumlah data negatif yang diklasifikasikan sebagai nilai negatif oleh model. “

Menurut (Muhammad Yusril Aldean, dkk., 2022), “*Precision* dijelaskan sebagai rasio antara *True Positive* dan *Predicted Positive*, sedangkan *Recall* dijelaskan sebagai rasio antara *True Positives* dan *Actual Positives*. Berikut cara menghitung nilai *Precision, Recall,* dan *F1 Score*.”

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

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

$$F1\ Score = 2\ .\ \frac{Precision\ .\ Recall}{Precision\ .Recall} = 2\ .\ \frac{TP}{TP + FN + FP}$$

2.12 Streamlit

Dikutip dari (*Streamlit Docs*, 2023), “*Streamlit* dijelaskan sebagai *web framework* yang berfungsi dalam pengembangan *web* dalam bidang ilmu data, dengan menggunakan bahasa Python.”

Selanjutnya, masih dikutip dari (*Streamlit Docs*, 2023), cara memasang *streamlit* adalah sebagai berikut :

1. *Install Anaconda*
2. Buat *environment* baru untuk *Streamlit*:

- Buka *Anaconda Navigator*, Ikuti langkah-langkah yang disediakan oleh *Anaconda* untuk mengatur dan mengelola *environment* menggunakan *Anaconda Navigator*.
- Pilih *icon* "▶" pada *environment*. Kemudian pilih "*Open terminal*"
- Pada *terminal* yang muncul, ketik "*pip install streamlit*"
- Kemudian cek apakah sudah terpasang dan bekerja dengan baik, ketik "*streamlit hello*". jika berhasil maka *Streamlit Hello App* akan muncul.

3. Gunakan pada *environment* baru:
   - Pada *Anaconda Navigator*, buka *terminal* di *environment* yang digunakan.
   - Pada *terminal* yang terbuka, gunakan *Streamlit* dengan mengetikkan "*streamlit run myfile.py*"

2.13 Cloud Database (Deta Space)

Dikutip dari (*Deta Space*, 2023), "*Deta Base* dijelaskan sebagai penyimpanan *Cloud Database* dimana penggunaannya menggunakan Koleksi *(Collection)* seperti yang digunakan pada metode *NoSQL* dengan memasangkan *Key-Value*. "

Selanjutnya, masih dikutip dari (*Deta Space*, 2023), berikut adalah cara menggunakan Deta Base dalam pengembangan web :

1. *Install Deta library* melalui *terminal*, ketik "*pip install deta*"
2. *Import Deta*, kemudian lakukan inisiasi menggunakan *Data Key* sebagai kunci atau biasa disebut dengan nama "*DETA_PROJECT_KEY*" dalam *environment*.

3. Setelah melakukan autentikasi menggunakan *Data Key*, selanjutnya lakukan *instance subclass* bernama "*Base*" dengan nama basis data yang dipilih. *Deta Base* otomatis dibuat dan sudah siap digunakan.

TEKNOLOGI INFORMASI

UNIVERSITAS DARMA PERSADA