

# MACHINE LEARNING

## UJIAN AKHIR SEMESTER

Nama : Adi Purnama

Nim : 20220801426

### SOAL ESSAY :

- 1) Jelaskan bagaimana memanfaatkan Machine Learning untuk Call API
- 2) Jelaskan apa yang dimaksud dengan unsupervised learning
- 3) Sebutkan beberapa tools yang di gunakan untuk mengolah machine learning

Jawab :

- 1) Memanfaatkan Machine Learning untuk Call API

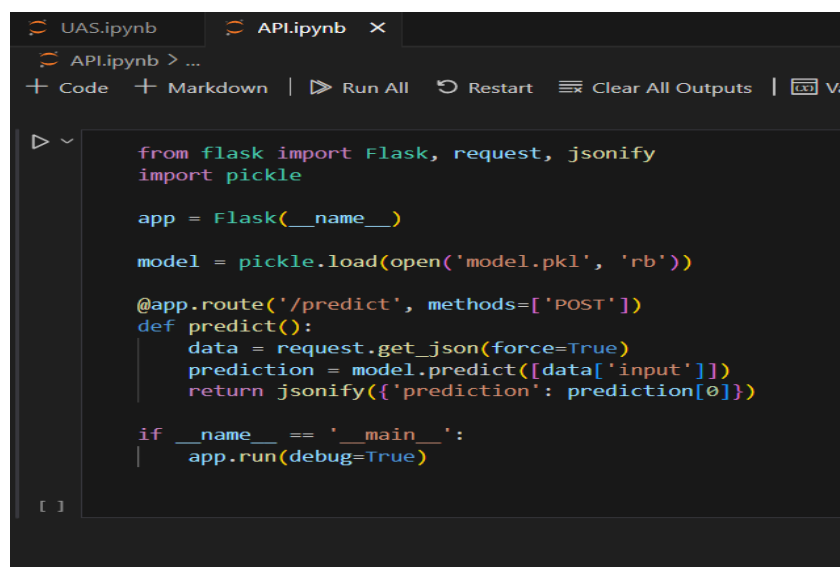
Machine Learning dapat dimanfaatkan untuk Call API dengan cara berikut:

#### 1.) **Model Deployment:**

Model Machine Learning yang telah dilatih dan dievaluasi dapat di-deploy sebagai API. Ini memungkinkan aplikasi lain untuk mengirimkan data ke model tersebut dan menerima prediksi atau hasil dari model tersebut. Berikut adalah langkah-langkah umumnya:

- **Training Model** : Melatih model menggunakan dataset yang sesuai.
- **Model Serialization** : Menyimpan model yang telah dilatih dalam format yang dapat digunakan kembali, seperti `pickle` atau `joblib` di Python.
- **API Development** : Mengembangkan API menggunakan framework seperti Flask atau FastAPI. API ini akan menerima input, memprosesnya dengan model Machine Learning, dan mengembalikan output.
- **Deployment** : Menggunakan platform seperti Docker, Heroku, atau AWS untuk meng-deploy API sehingga dapat diakses oleh aplikasi lain.

Contoh kode sederhana untuk membuat API dengan Flask:



```
from flask import Flask, request, jsonify
import pickle

app = Flask(__name__)

model = pickle.load(open('model.pkl', 'rb'))

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json(force=True)
    prediction = model.predict([data['input']])
    return jsonify({'prediction': prediction[0]})

if __name__ == '__main__':
    app.run(debug=True)
```

### **Automated Decision Making :**

API yang memanfaatkan Machine Learning dapat digunakan untuk pengambilan keputusan otomatis. Contohnya termasuk deteksi penipuan dalam transaksi keuangan, rekomendasi produk dalam e-commerce, atau analisis sentimen untuk ulasan pelanggan.

### **Scalability :**

API Machine Learning memungkinkan skalabilitas, di mana banyak request dapat diproses secara paralel. Ini sangat berguna untuk aplikasi dengan volume data besar atau permintaan prediksi yang tinggi.

## **2) Penjelasan tentang Unsupervised Learning**

### **Unsupervised Learning :**

adalah metode Machine Learning di mana model dilatih menggunakan data yang tidak berlabel. Berbeda dengan supervised learning, di mana data pelatihan berisi input-output pairs (label), unsupervised learning bekerja dengan hanya menggunakan input data tanpa label yang jelas. Tujuan utama unsupervised learning adalah untuk menemukan pola atau struktur yang tersembunyi dalam data.

### **Contoh teknik unsupervised learning:**

- **Clustering :**
  - Mengelompokkan data ke dalam beberapa kelompok berdasarkan kesamaan. Contoh algoritma clustering adalah K-Means, DBSCAN, dan Hierarchical Clustering.
- **Dimensionality Reduction :**
  - Mengurangi jumlah fitur dalam data sambil mempertahankan informasi penting. Contoh algoritma adalah Principal Component Analysis (PCA) dan t-Distributed Stochastic Neighbor Embedding (t-SNE).
- **Association Rule Learning :**
  - Mencari hubungan atau aturan dalam data. Contoh penerapannya adalah analisis keranjang belanja untuk menemukan produk yang sering dibeli bersama.

### 3) Tools untuk Mengolah Machine Learning

Beberapa tools yang sering digunakan dalam pengolahan Machine Learning meliputi:

#### 1. Libraries dan Frameworks :

- **Scikit-learn** : Pustaka Python yang menyediakan alat untuk pemodelan data dan analisis data, termasuk algoritma klasifikasi, regresi, clustering, dan reduksi dimensi.
- **TensorFlow** : Framework open-source yang digunakan untuk membangun dan melatih model Machine Learning, terutama deep learning.
- **PyTorch** : Framework deep learning yang dikembangkan oleh Facebook's AI Research lab (FAIR), yang menawarkan fleksibilitas dan kecepatan dalam penelitian dan pengembangan model.
- **NLTK** : Pustaka Python untuk Natural Language Processing yang menyediakan alat untuk pengolahan teks, tokenisasi, stemming, tagging, parsing, dan lainnya.

#### 2. Development Tools :

- **Jupyter Notebook** : Alat interaktif untuk menulis dan menjalankan kode Python, sangat berguna untuk eksplorasi data, visualisasi, dan pengembangan model Machine Learning.
- **Google Colab** : Platform gratis yang memungkinkan pengguna menjalankan Jupyter Notebook di cloud, dengan dukungan GPU untuk percepatan komputasi.

#### 3. Deployment Tools :

- **Docker** : Platform untuk mengembangkan, mengirim, dan menjalankan aplikasi dalam container, yang memastikan bahwa aplikasi berjalan secara konsisten di berbagai lingkungan.
- **Heroku** : Platform cloud yang memungkinkan deployment aplikasi web, termasuk API Machine Learning, dengan mudah.
- **AWS (Amazon Web Services)** : Menyediakan berbagai layanan untuk deployment, termasuk Amazon SageMaker untuk pembangunan, pelatihan, dan deployment model Machine Learning.

## SOAL STUDI KASUS :

Tentu! Berikut adalah penjelasan langkah demi langkah dalam bahasa Indonesia beserta hasil dari pengolahan teks yang dilakukan:

### Langkah-langkah Pengolahan Teks

#### 1. Instalasi Pustaka yang Diperlukan:

Pertama, kita perlu menginstal pustaka yang dibutuhkan menggunakan pip. Ini memastikan kita memiliki semua alat yang diperlukan untuk pengolahan teks dan visualisasi.

```
pip install wordcloud nltk matplotlib seaborn
```

#### 2. Import Pustaka dan Persiapan Data :

Mengimpor pustaka yang diperlukan untuk membuat word cloud, plotting, dan analisis sentimen. Kita juga mendefinisikan teks yang akan dianalisis.

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import re
nltk.download('vader_lexicon')

text = """
(masukkan teks cerita Snow White di sini)
"""
```

✓ 0.0s Python

#### 3. Membuat Word Cloud:

Membuat dan menampilkan word cloud dari teks.

```
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud dari Cerita Snow White")
plt.show()
```

[6] ✓ 0.0s Python

#### 4. Bar Plot Frekuensi Kata:

Tokenisasi teks, menghapus stopwords, dan memplot frekuensi kata yang paling umum.

```
basic_stopwords = set([
    'a', 'an', 'the', 'and', 'or', 'but', 'if', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with',
    'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from',
    'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there',
    'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such',
    'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don',
    'should', 'now'
])

tokens = re.findall(r'\b\w+\b', text.lower())
filtered_tokens = [word for word in tokens if word not in basic_stopwords]

word_freq = Counter(filtered_tokens)
common_words = word_freq.most_common(20)

words, frequencies = zip(*common_words)

plt.figure(figsize=(12, 6))
sns.barplot(x=list(words), y=list(frequencies))
plt.xticks(rotation=45)
plt.title("Kata yang Paling Sering Muncul dalam Cerita Snow White")
plt.xlabel("Kata")
plt.ylabel("Frekuensi")
plt.show()
```

#### 5. Analisis Sentimen :

Menganalisis sentimen teks menggunakan VADER dan memplot hasilnya.

```
sid = SentimentIntensityAnalyzer()
sentiment_scores = sid.polarity_scores(text)

labels = ['Positif', 'Netral', 'Negatif']
scores = [sentiment_scores['pos'], sentiment_scores['neu'], sentiment_scores['neg']]

plt.figure(figsize=(8, 6))
sns.barplot(x=labels, y=scores)
plt.title("Analisis Sentimen dari Cerita Snow White")
plt.ylabel("Skor")
plt.show()
```

Hasil Running

Word Cloud:

```
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Word Cloud dari Cerita Snow White")
plt.show()
```

Bar Plot Frekuensi Kata:

```
plt.figure(figsize=(12, 6))
sns.barplot(x=list(words), y=list(frequencies))
plt.xticks(rotation=45)
plt.title("Kata yang Paling Sering Muncul dalam Cerita Snow White")
plt.xlabel("Kata")
plt.ylabel("Frekuensi")
plt.show()
```

## Analisis Sentimen:



```
labels = ['Positif', 'Netral', 'Negatif']
scores = [sentiment_scores['pos'], sentiment_scores['neu'], sentiment_scores['neg']]

plt.figure(figsize=(8, 6))
sns.barplot(x=labels, y=scores)
plt.title("Analisis Sentimen dari Cerita Snow White")
plt.ylabel("Skor")
plt.show()
```

[6] ✓ 0.0s Python

## Penjelasan Langkah-Langkah

### 1. Instalasi Pustaka:

Pustaka `wordcloud`, `nltk`, `matplotlib`, dan `seaborn` diinstal untuk keperluan pengolahan dan visualisasi teks.

### 2. Import dan Persiapan Data:

Pustaka diimport, dan teks cerita dimasukkan ke dalam variabel `text`.

### 3. Pembuatan Word Cloud:

Word cloud dibuat dari teks untuk memvisualisasikan kata-kata yang paling sering muncul.

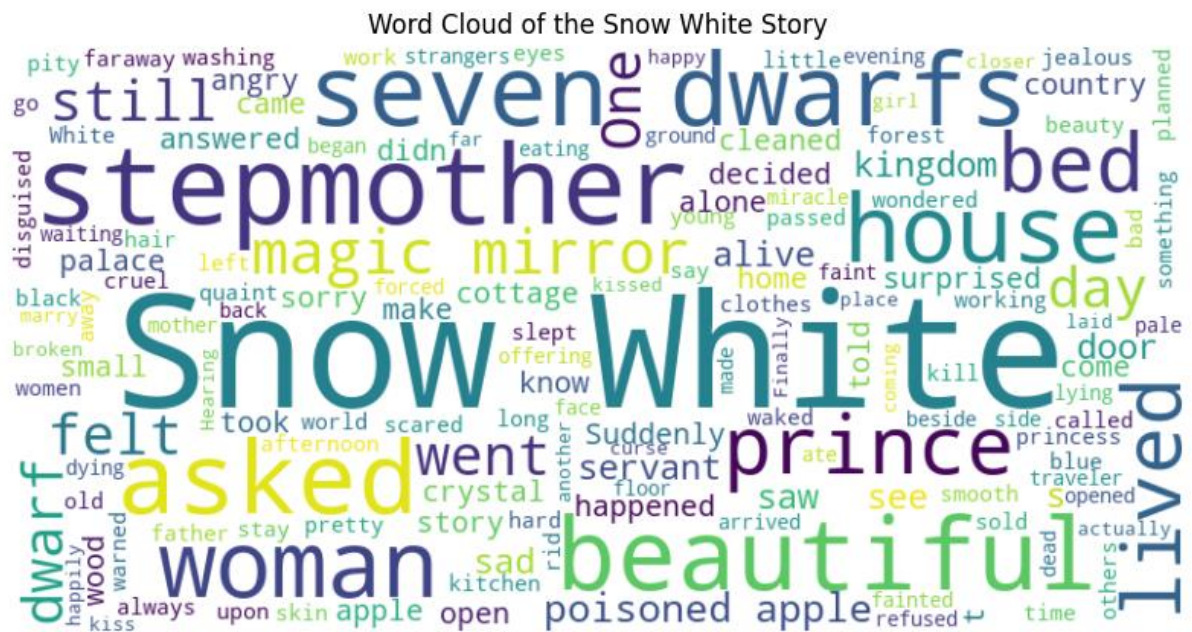
### 4. Bar Plot Frekuensi Kata:

Teks di-tokenisasi, stopwords dihapus, dan frekuensi kata yang paling umum diplot dalam bentuk bar plot.

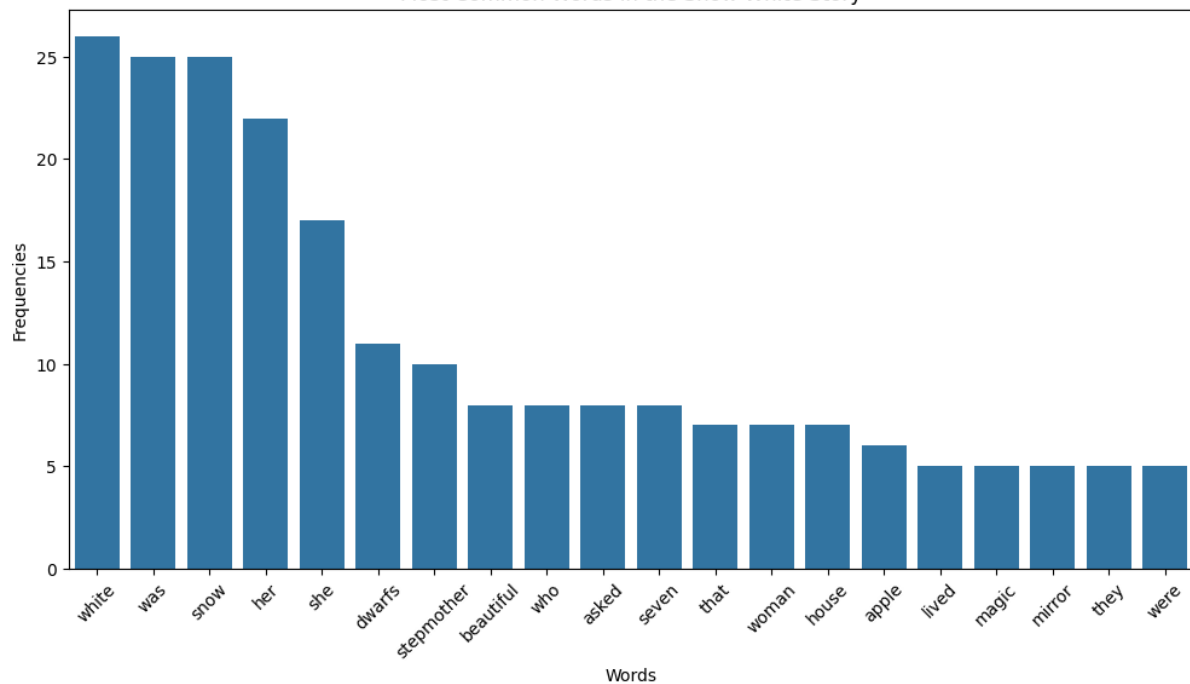
### 5. Analisis Sentimen:

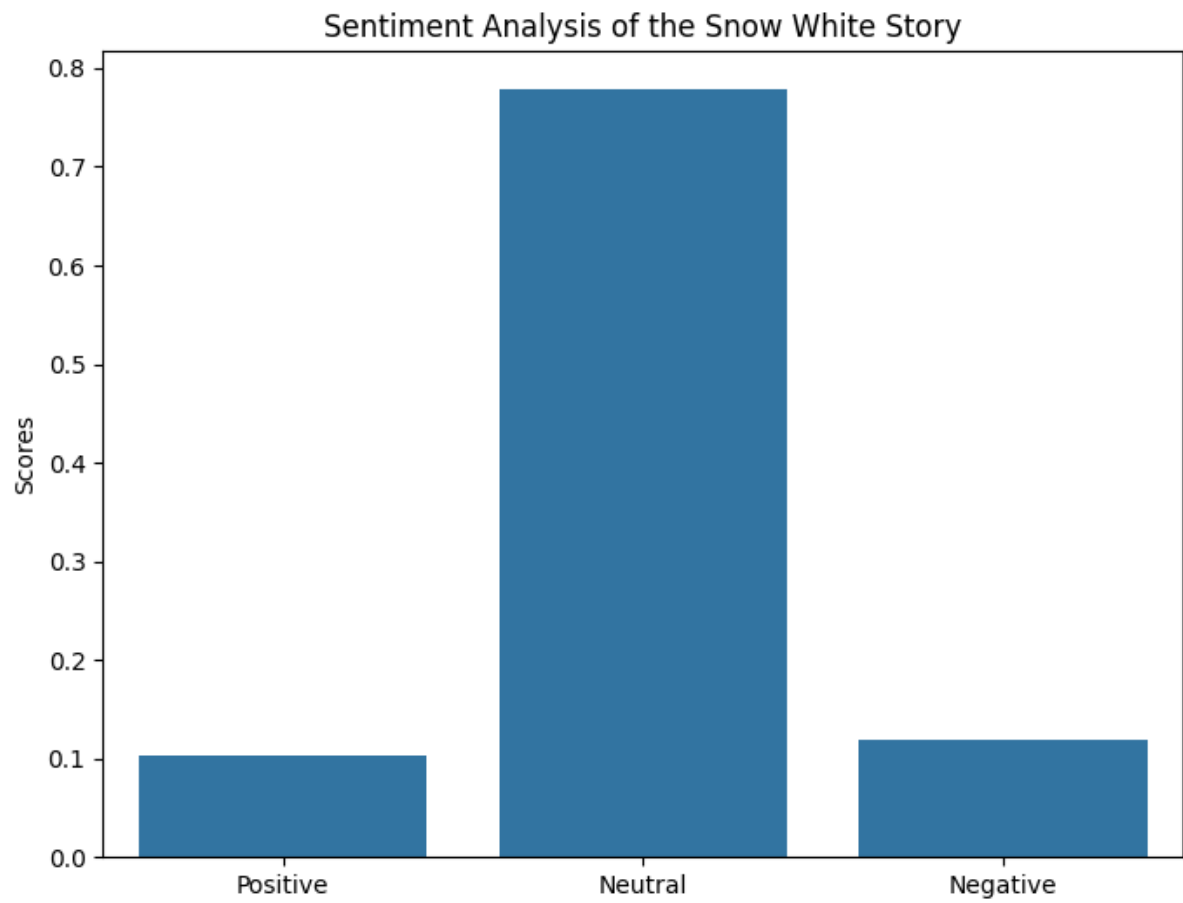
Sentimen dari teks dianalisis menggunakan VADER untuk mengidentifikasi sentimen positif, netral, dan negatif.

**Berikut adalah OutPut yang di berikan :**



### Most Common Words in the Snow White Story





[https://github.com/Poxezy/UAS-MACHINE\\_LEARNING](https://github.com/Poxezy/UAS-MACHINE_LEARNING)