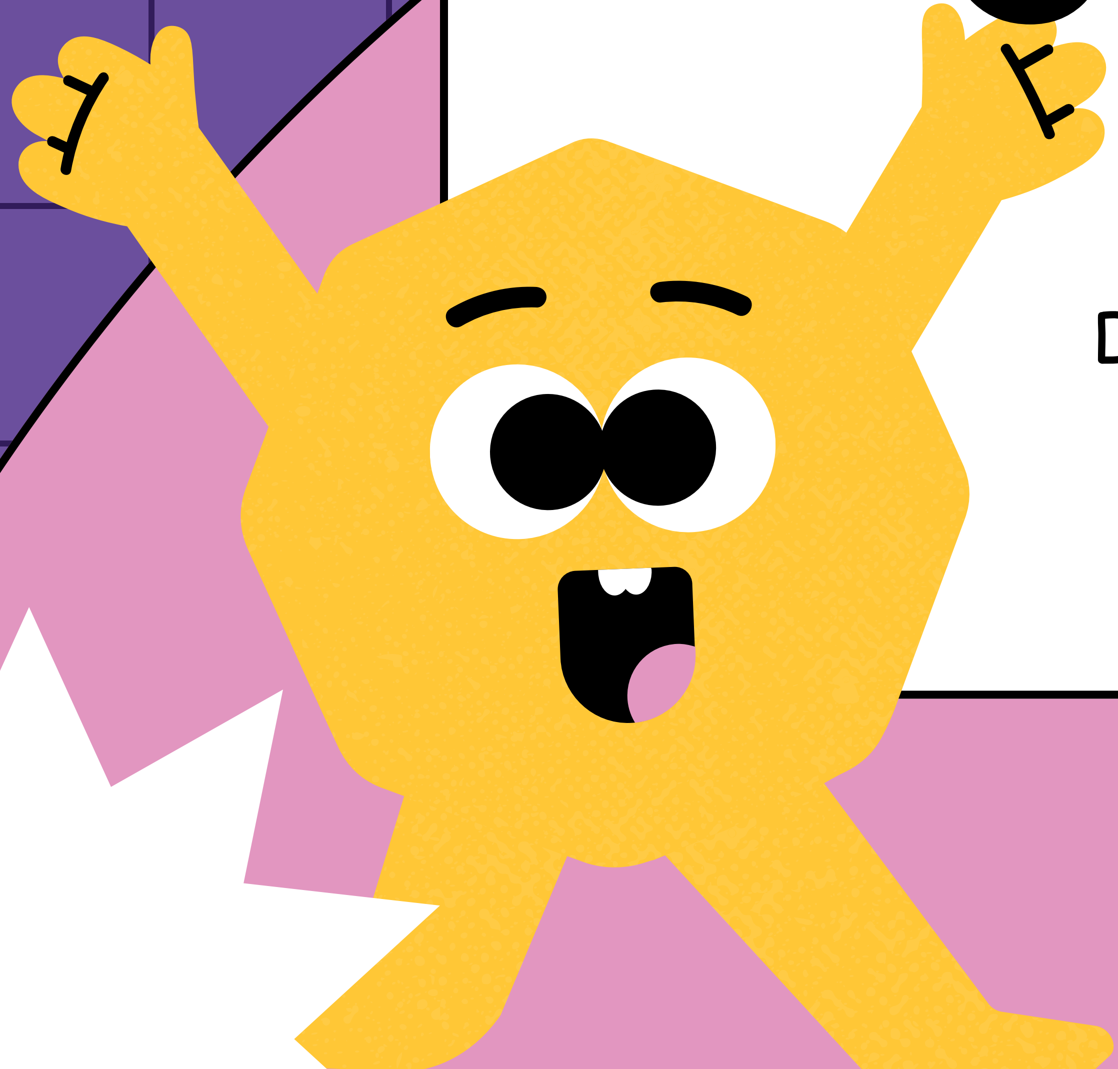
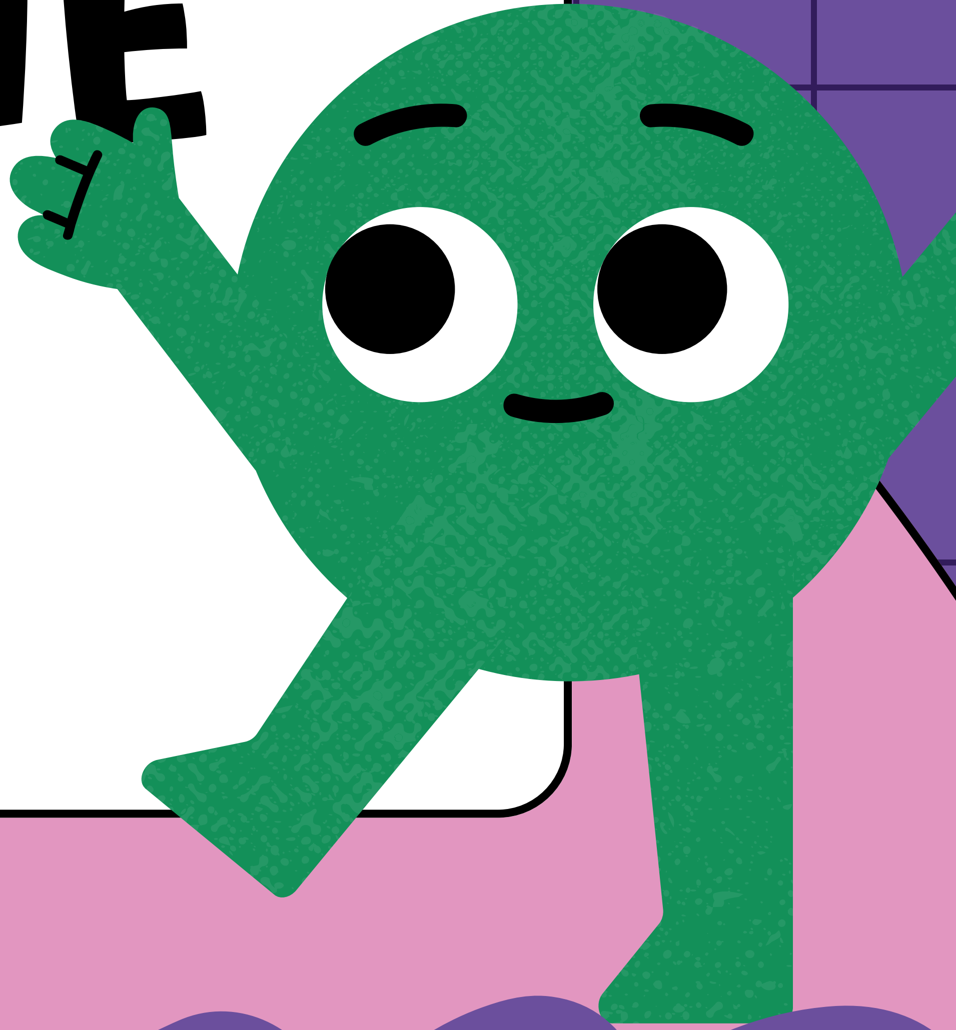
LAPORAN PRAKTIKUM TEXT MAINING

FUN!

SEARCH ENGINE



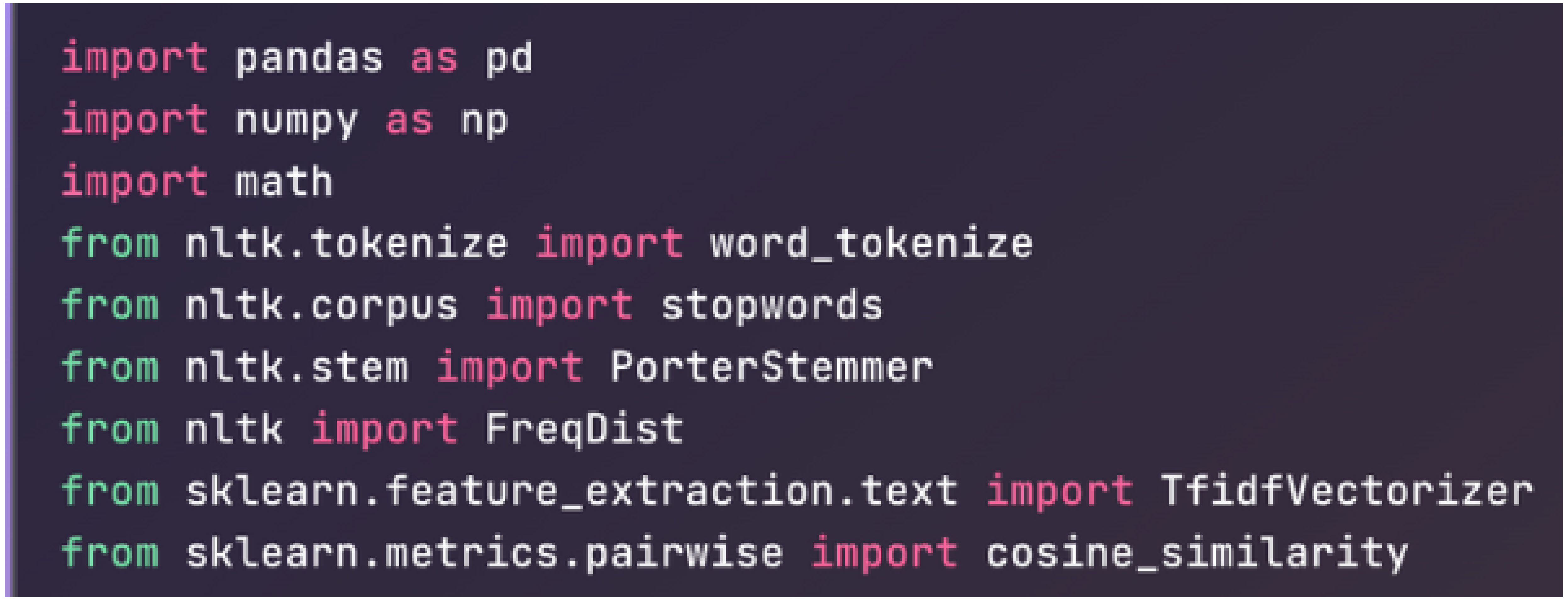


Dosen Pengampu: Dr. Tita Karlita, S.Kom., M.Kom.

Bayu Kurniawan / 3322600019

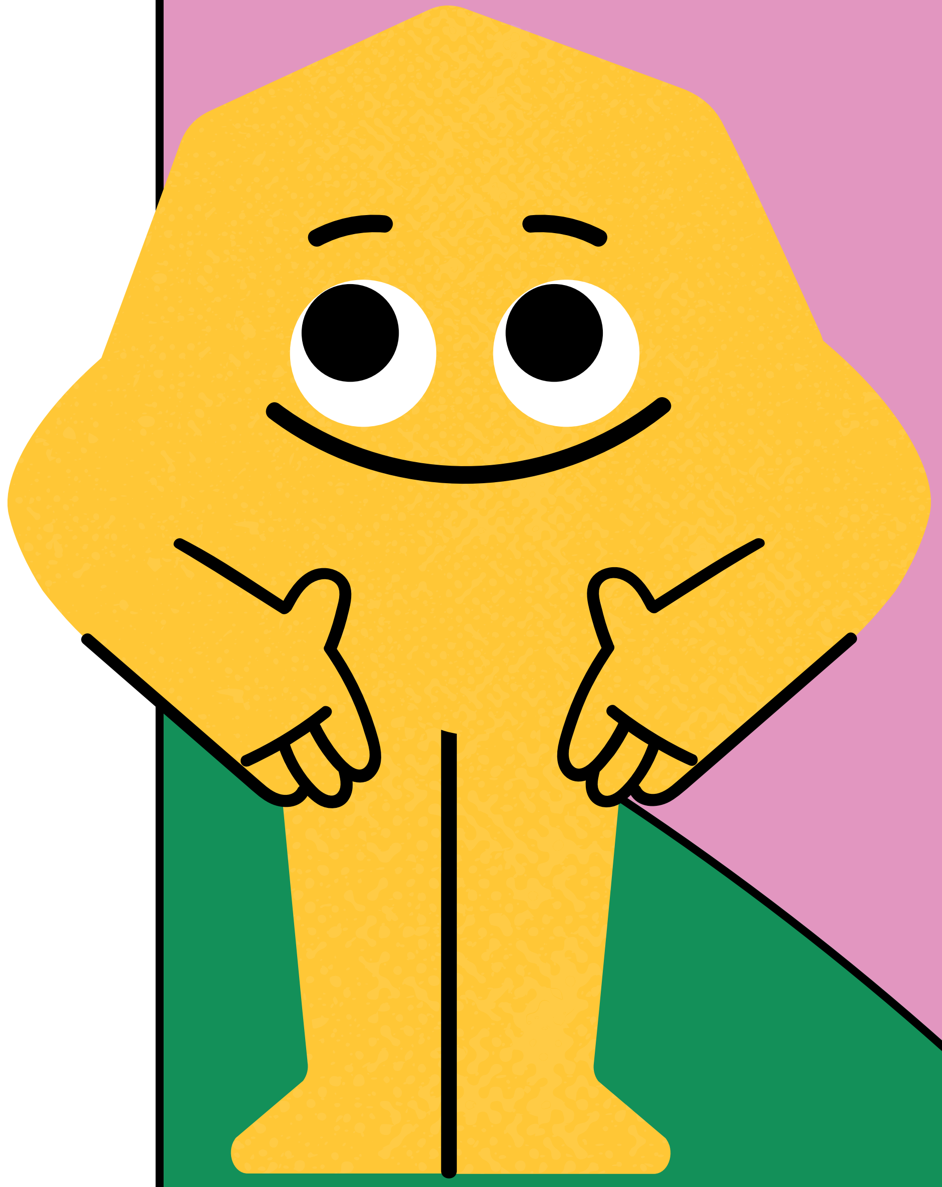
SEARCH ENGINE

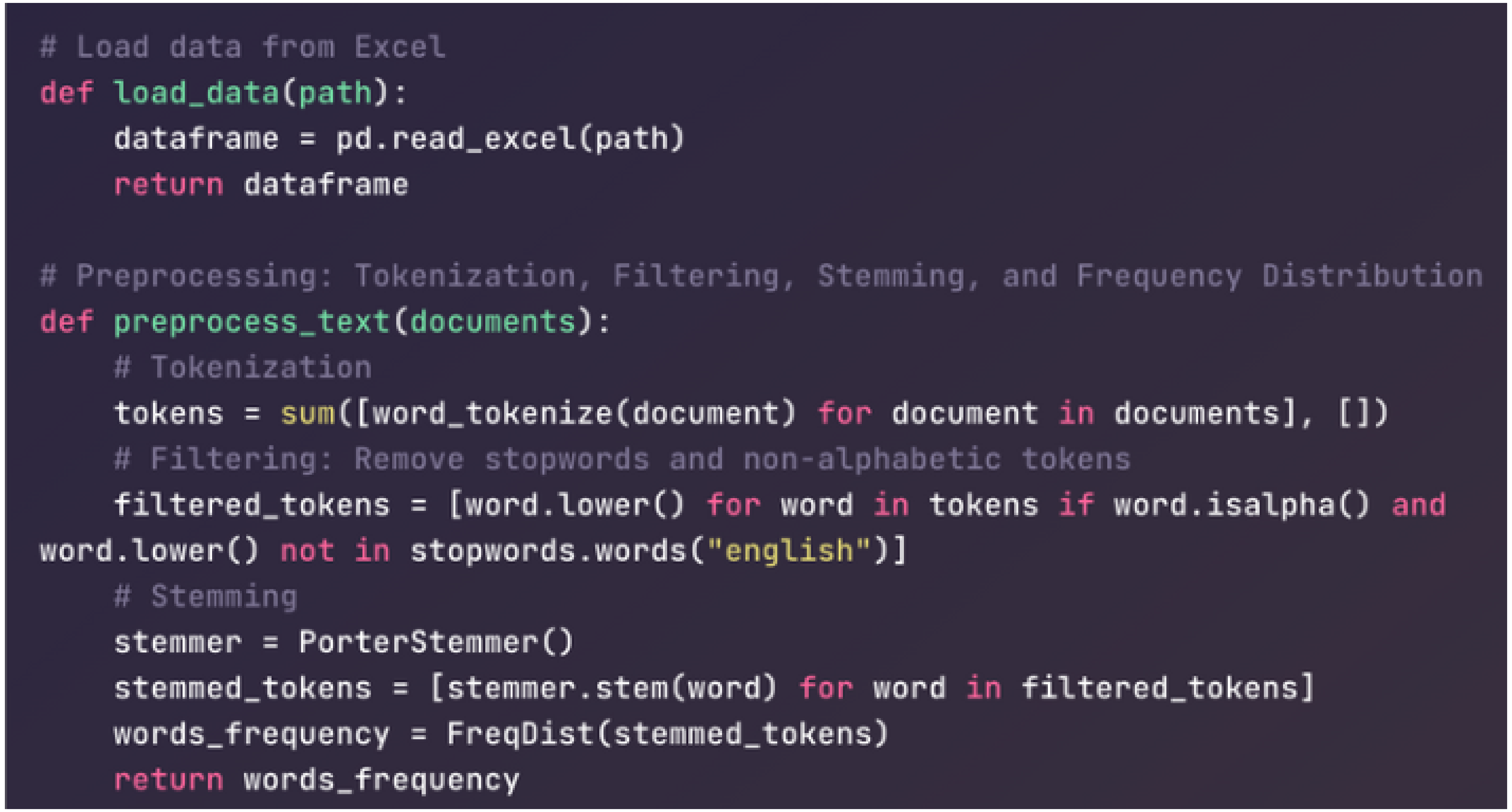
Search engine pada dasarnya meupakan program yang dirancang untuk mencari item dalam database yang sesuai dengan kueri yang diberikan oleh pengguna. Search engine dapat dibuat untuk berjalan secara lokal atau menjadi aplikasi berbasis web, dan item yang dicari dapat berupa apa saja, seperti halaman web, gambar, video, dan lain-lain. Namun konsep pencarian informasi dan data mining yang diperlukan untuk melakukannya pada dasarnya sama.



```
import pandas as pd
import numpy as np
import math
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk import FreqDist
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

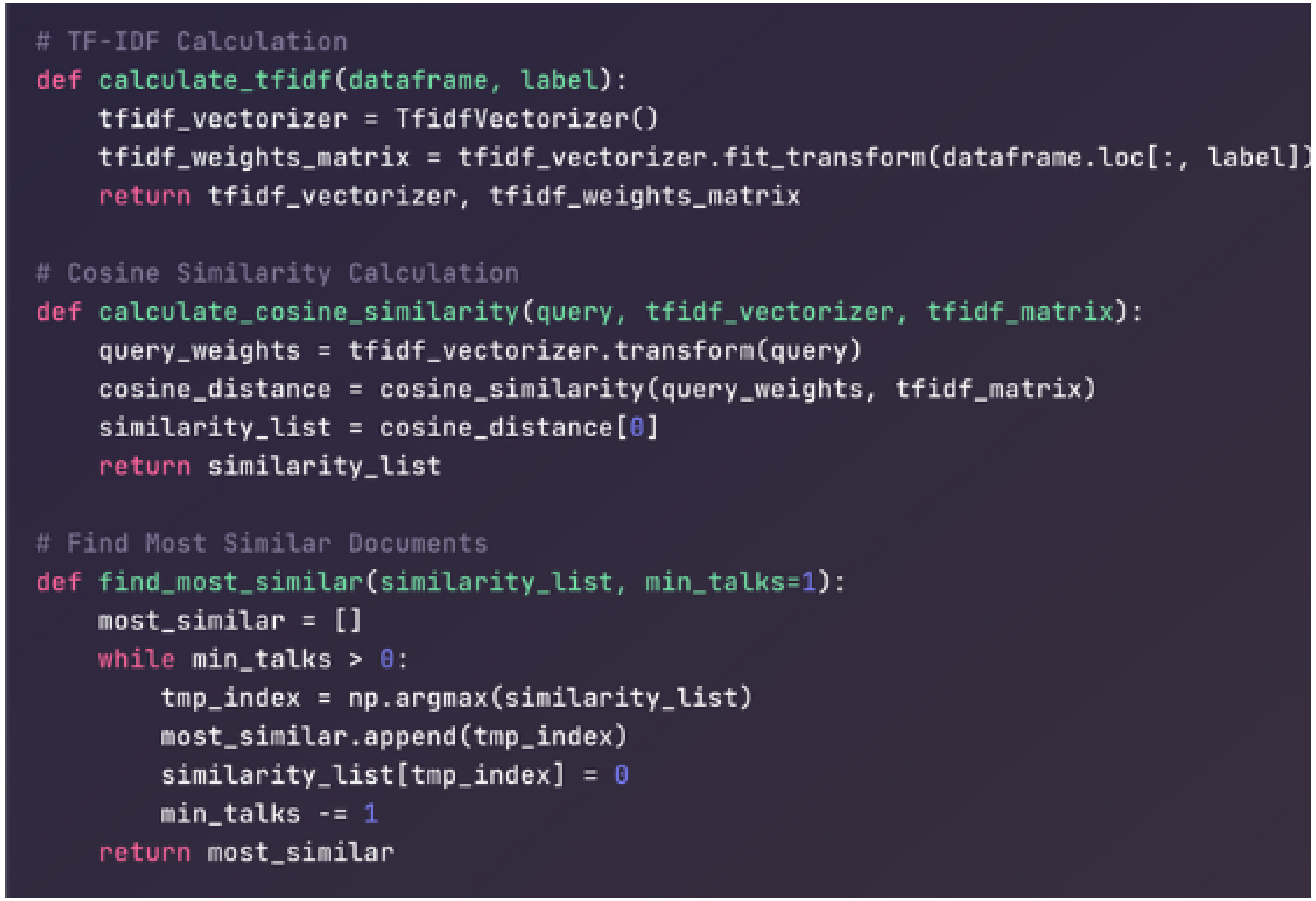
import library yang akan digunakan





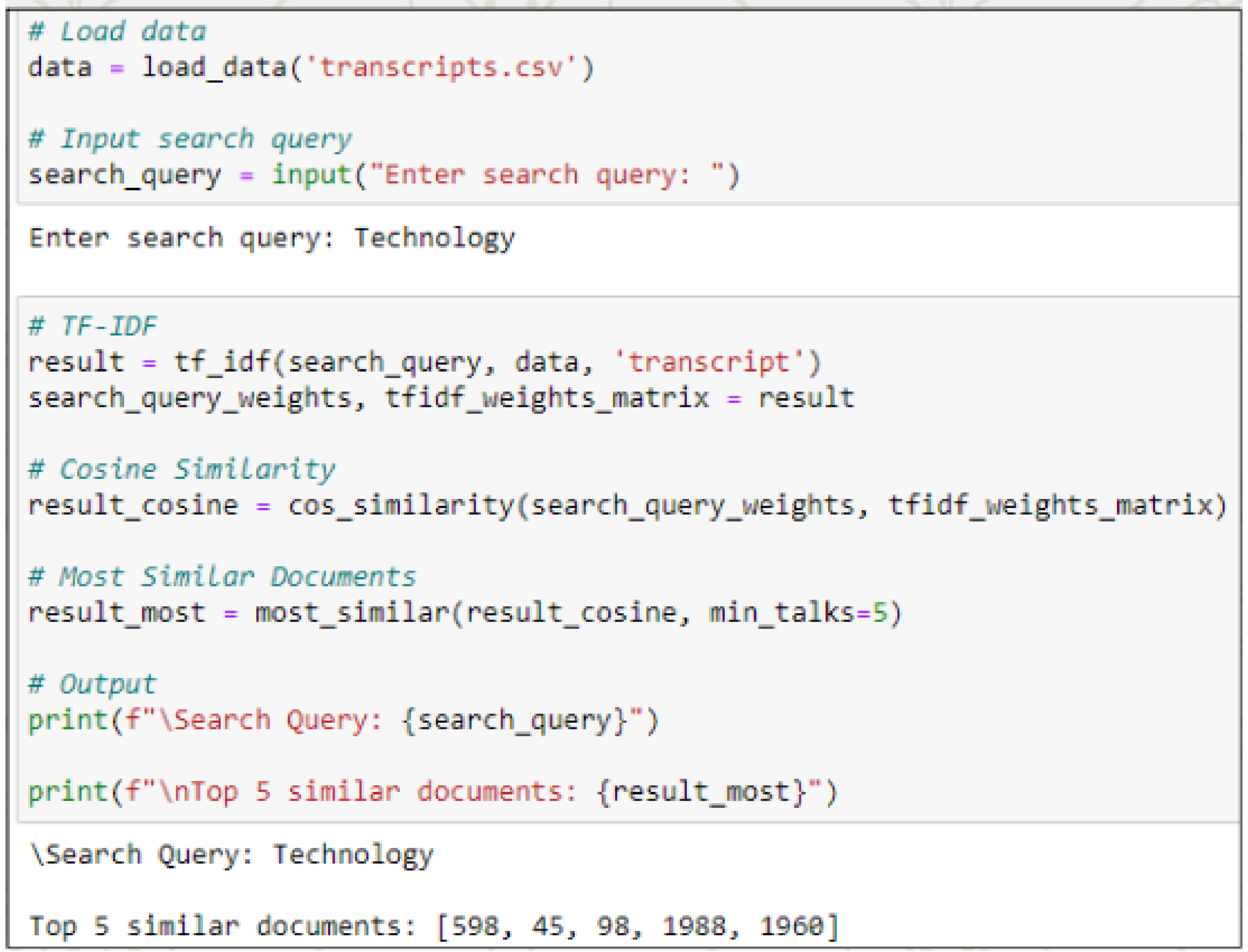
```python
# Load data from Excel
def load_data(path):
    dataframe = pd.read_excel(path)
    return dataframe

# Preprocessing: Tokenization, Filtering, Stemming, and Frequency Distribution
def preprocess_text(documents):
    # Tokenization
    tokens = sum([word_tokenize(document) for document in documents], [])
    # Filtering: Remove stopwords and non-alphabetic tokens
    filtered_tokens = [word.lower() for word in tokens if word.isalpha() and
word.lower() not in stopwords.words("english")]
    # Stemming
    stemmer = PorterStemmer()
    stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]
    words_frequency = FreqDist(stemmed_tokens)
    return words_frequency
```

- Fungsi pertama untuk membaca data berupa excel
- Fungsi kedua untuk mempreprocessing text yang nanti akan dimasukkan dengan mentokenizing, filtering, dan stemming.



```python
# TF-IDF Calculation
def calculate_tfidf(dataframe, label):
    tfidf_vectorizer = TfidfVectorizer()
    tfidf_weights_matrix = tfidf_vectorizer.fit_transform(dataframe.loc[:, label])
    return tfidf_vectorizer, tfidf_weights_matrix

# Cosine Similarity Calculation
def calculate_cosine_similarity(query, tfidf_vectorizer, tfidf_matrix):
    query_weights = tfidf_vectorizer.transform(query)
    cosine_distance = cosine_similarity(query_weights, tfidf_matrix)
    similarity_list = cosine_distance[0]
    return similarity_list

# Find Most Similar Documents
def find_most_similar(similarity_list, min_talks=1):
    most_similar = []
    while min_talks > 0:
        tmp_index = np.argmax(similarity_list)
        most_similar.append(tmp_index)
        similarity_list[tmp_index] = 0
        min_talks -= 1
    return most_similar
```

Fungsi calculate_tf idf menghitung TF-IDF menggunakan TfidfVectorizer dari scikit-learn. Pertama, fungsi calculate_tfidf mengonversi teks ke dalam matriks fitur TF-IDF dan mengembalikan vektorisasi TF-IDF beserta matriks bobotnya. Kemudian, calculate_cosine_similarity menghitung kemiripan kosinus antara query dan dokumen dengan matriks bobot TF-IDF yang sama. Terakhir, find_most_similar mengidentifikasi dokumen yang paling mirip dengan query berdasarkan kemiripan kosinus, memungkinkan pemilihan beberapa dokumen yang paling mirip.



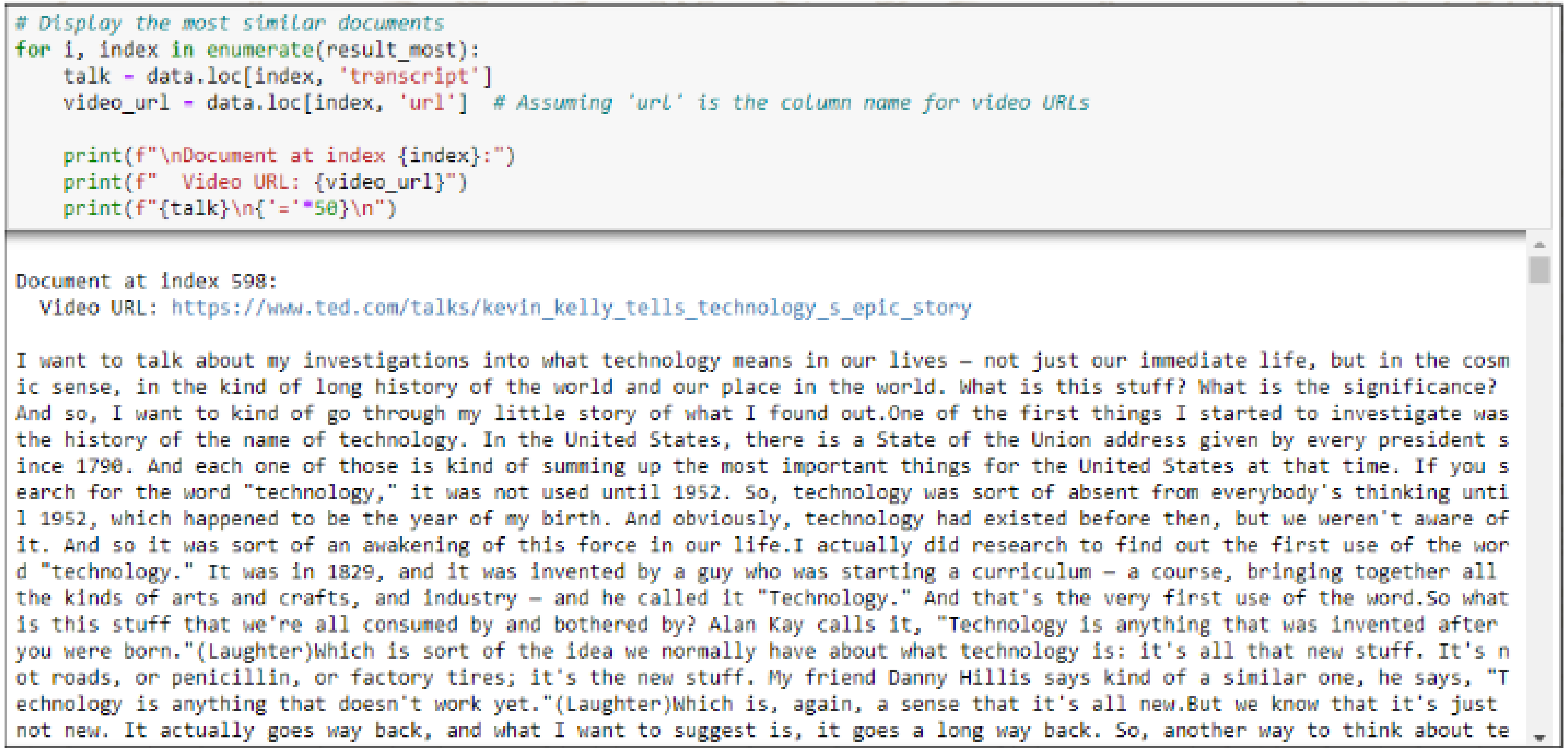
```
# Load data
data = load_data('transcripts.csv')

# Input search query
search_query = input("Enter search query: ")
```

```
Enter search query: Technology
```

```
# TF-IDF
result = tf_idf(search_query, data, 'transcript')
search_query_weights, tfidf_weights_matrix = result

# Cosine Similarity
result_cosine = cos_similarity(search_query_weights, tfidf_weights_matrix)

# Most Similar Documents
result_most = most_similar(result_cosine, min_talks=5)

# Output
print(f"\Search Query: {search_query}")

print(f"\nTop 5 similar documents: {result_most}")
```

```
\Search Query: Technology

Top 5 similar documents: [598, 45, 98, 1988, 1960]
```

Pertama, data dari file CSV 'transcripts.csv' dimuat menggunakan fungsi `load_data’. Kemudian, pengguna diminta untuk memasukkan kueri pencarian melalui input. Kueri tersebut kemudian diolah menggunakan fungsi `tf_idf` untuk menghasilkan vektor bobot TF-IDF dan matriks bobot TF-IDF dari dokumen. Selanjutnya, menggunakan fungsi `cos_similarity`, dihitung kemiripan kosinus antara kueri pencarian dan setiap dokumen dalam DataFrame. Hasil kemiripan digunakan oleh fungsi `most_similar` untuk mengidentifikasi dan mencetak lima dokumen paling mirip dengan kueri pencarian



```python
# Display the most similar documents
for i, index in enumerate(result_most):
    talk = data.loc[index, 'transcript']
    video_url = data.loc[index, 'url']  # Assuming 'url' is the column name for video URLs

    print(f"\nDocument at index {index}:")
    print(f"  Video URL: {video_url}")
    print(f"{talk}\n{'='*50}\n")
```

```
Document at index 598:
  Video URL: https://www.ted.com/talks/kevin_kelly_tells_technology_s_epic_story

I want to talk about my investigations into what technology means in our lives — not just our immediate life, but in the cosm
ic sense, in the kind of long history of the world and our place in the world. What is this stuff? What is the significance?
And so, I want to kind of go through my little story of what I found out.One of the first things I started to investigate was
the history of the name of technology. In the United States, there is a State of the Union address given by every president s
ince 1790. And each one of those is kind of summing up the most important things for the United States at that time. If you s
earch for the word "technology," it was not used until 1952. So, technology was sort of absent from everybody's thinking unti
l 1952, which happened to be the year of my birth. And obviously, technology had existed before then, but we weren't aware of
it. And so it was sort of an awakening of this force in our life.I actually did research to find out the first use of the wor
d "technology." It was in 1829, and it was invented by a guy who was starting a curriculum — a course, bringing together all
the kinds of arts and crafts, and industry — and he called it "Technology." And that's the very first use of the word.So what
is this stuff that we're all consumed by and bothered by? Alan Kay calls it, "Technology is anything that was invented after
you were born."(Laughter)Which is sort of the idea we normally have about what technology is: it's all that new stuff. It's n
ot roads, or penicillin, or factory tires; it's the new stuff. My friend Danny Hillis says kind of a similar one, he says, "T
echnology is anything that doesn't work yet."(Laughter)Which is, again, a sense that it's all new.But we know that it's just
not new. It actually goes way back, and what I want to suggest is, it goes a long way back. So, another way to think about te
```

Langkah terakhir menampilkan lima dokumen paling mirip dengan kueri pencarian. Melalui loop, indeks dari lima dokumen tersebut digunakan untuk mengakses teks transkrip dan URL video terkait dari DataFrame. Setiap dokumen ditampilkan dalam format yang jelas, termasuk URL video, teks transkrip, dan pemisah garis horizontal untuk membedakan antar-dokumen.

TERIMA KASIH

Semoga dapat ilmu yang bermanfaat dari presentasi ini. Semoga beruntung !