NAMA : HAYA DINAH AMALIYAH

NIM : 32602000108

UAS DATA MINING

Penjelasan keyword extraction dengan metode TextRank :

1. Mengimport library

```
In [2]: import nltk
        from nltk.corpus import stopwords
        from nltk.tokenize import sent_tokenize, word_tokenize
        from nltk.probability import FreqDist
        from nltk.tokenize import RegexpTokenizer
        from nltk.stem import PorterStemmer
```

Pada bagian ini, mengimpor pustaka yang diperlukan dari NLTK untuk pemrosesan teks, termasuk tokenisasi, penghilangan stopword, dan stemming

2. Mendownload library

```
In [3]: nltk.download('punkt')
        nltk.download('stopwords')

        [nltk_data] Downloading package punkt to C:\Users\Haya
        [nltk_data]     Dinah\AppData\Roaming\nltk_data...
        [nltk_data]   Unzipping tokenizers\punkt.zip.
        [nltk_data] Downloading package stopwords to C:\Users\Haya
        [nltk_data]     Dinah\AppData\Roaming\nltk_data...
        [nltk_data]   Unzipping corpora\stopwords.zip.

Out[3]: True
```

Mendowload NLTK resource untuk menjalankan import jika belum mendownload

3. Memasukan teks

```
In [4]: text = """
        Objek wisata yang ada di Indonesia merupakan kekayaan alam yang patut untuk dibanggakan. Setiap daerah di Indonesia memiliki keun
        """
```

Masukan data teks yang akan di analisis yaitu sesuai tugas merupan abstrak dari tugas akhir mahasiswa TIF

4. Preprocessing text

```
In [5]: def preprocess_text(text):
            # Tokenize the text into sentences and words
            sentences = sent_tokenize(text)
            tokenizer = RegexpTokenizer(r'\w+')
            words = [word.lower() for sentence in sentences for word in tokenizer.tokenize(sentence)]

            # Remove stopwords and perform stemming
            stop_words = set(stopwords.words('english'))
            words = [word for word in words if word not in stop_words]
            stemmer = PorterStemmer()
            words = [stemmer.stem(word) for word in words]

            return words
```

Fungsi preprocess_text mengambil teks input, menokenya menjadi kalimat dan kata, mengubah kata-kata menjadi huruf kecil, menghapus stopwords, dan melakukan stemming menggunakan algoritma stemming Porter. Langkah prapemrosesan ini membantu mengurangi noise dan menormalkan teks.

5. Menghitung nilai kata

```
def calculate_word_scores(words, window_size=2):
    word_freq = FreqDist(words)
    word_scores = {}

    for word in set(words):
        word_scores[word] = 0

    for i, word in enumerate(words):
        for j in range(max(0, i - window_size), min(len(words), i + window_size + 1)):
            if i != j:
                word_scores[word] += word_freq[words[j]]

    return word_scores
```

Fungsi calculate_word_scores mengambil kata-kata yang telah diproses sebelumnya sebagai masukan dan menghitung skor untuk setiap kata menggunakan pendekatan kemunculan bersama. Fungsi ini menggunakan jendela geser dengan ukuran window_size di sekitar setiap kata untuk mempertimbangkan konteks dan menghitung skor berdasarkan frekuensi kemunculan bersama.

6. Menerapkan metode TextRank

```
def textrank_keywords(text, top_n=5):
    words = preprocess_text(text)
    word_scores = calculate_word_scores(words)

    # Sort the words based on their scores in descending order
    sorted_words = sorted(word_scores.items(), key=lambda x: x[1], reverse=True)

    # Get the top N keywords
    top_keywords = [word for word, score in sorted_words[:top_n]]
    return top_keywords
```

Fungsi textrank_keywords adalah bagian utama dari algoritme ekstraksi kata kunci TextRank. Fungsi ini pertama-tama melakukan praproses teks untuk mendapatkan daftar kata. Kemudian, ia menghitung skor kata menggunakan fungsi calculate_word_scores. Kata-kata tersebut kemudian diurutkan berdasarkan skornya dalam urutan menurun, dan kata kunci top_n teratas diekstraksi dan dikembalikan.

7. Mengekstrak dan printing keyword

```
In [6]: # Call the function to get the top keywords (e.g., N=5)
        top_keywords = textrank_keywords(text, top_n=5)

        # Print the top keywords
        print("Top Keywords:")
        for keyword in top_keywords:
            print(keyword)
```

```
Top Keywords:
yang
di
indonesia
ada
wisata
```

Terakhir, kode tersebut memanggil fungsi textrank_keywords dengan teks input dan top_n=5 untuk mendapatkan 5 kata kunci teratas. Kemudian mencetak kata kunci ini satu per satu, memberikan Anda hasil akhir dari ekstraksi kata kunci TextRank.