

Nama : Muhamad Faqih Azhar

NIM : 0110221092

Kelas : 5TI03 / Senin Malam

Pembuatan Project Natural Language Processing (NLP)

Topic

Text Summarization

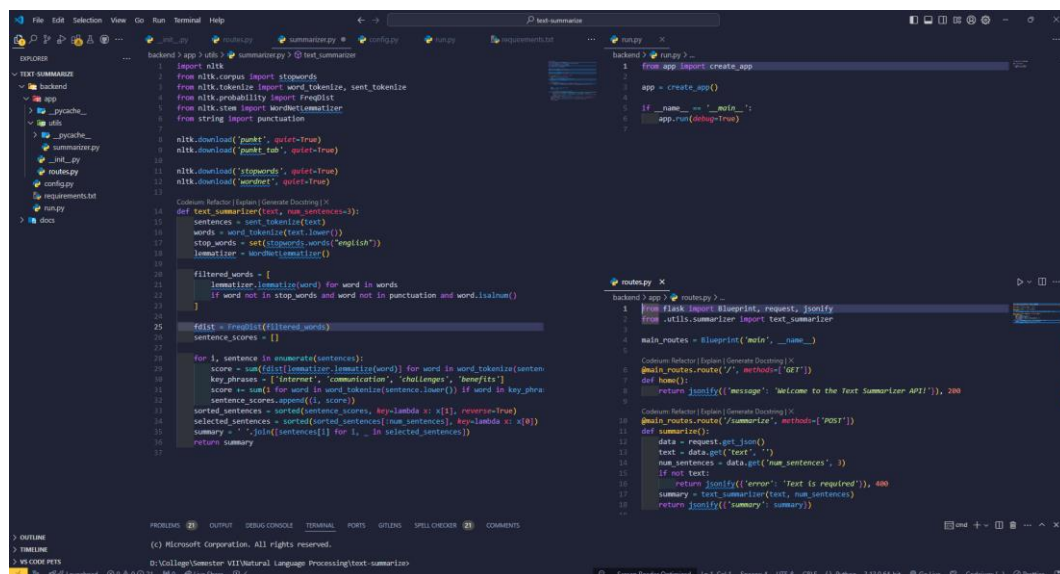
Sumber referensi:

<https://quillbot.com/summarize>

<https://www.youtube.com/watch?v=9PoKellNrBc&t=451s>

https://www.youtube.com/watch?v=SNimr_nOC7w

Laporan Progress



```
1 import nltk
2 from nltk.corpus import stopwords
3 from nltk.tokenize import word_tokenize, sent_tokenize
4 from nltk.probability import FreqDist
5 from nltk.stem import WordNetLemmatizer
6 from string import punctuation
7
8 nltk.download('punkt', quiet=True)
9 nltk.download('punkt_tab', quiet=True)
10
11 nltk.download('stopwords', quiet=True)
12 nltk.download('wordnet', quiet=True)
13
14 Custom Defacto (Bukan Generate Disting)
15 def test_summarizer(text, num_sentences=3):
16     sentences = sent_tokenize(text)
17     words = word_tokenize(text.lower())
18     stop_words = set(stopwords.words('english'))
19     lemmatizer = WordNetLemmatizer()
20
21     filtered_words = [
22         lemmatizer.lemmatize(word) for word in words
23         if word not in stop_words and word not in punctuation and word.isalnum()
24     ]
25     freq = FreqDist(filtered_words)
26     sentence_scores = []
27
28     for i, sentence in enumerate(sentences):
29         score = sum(freq[word]*lemmatizer.lemmatize(word) for word in word_tokenize(sentence.lower()))
30         key_phrases = ['internet', 'communication', 'challenges', 'benefits']
31         score += sum(1 for word in word_tokenize(sentence.lower()) if word in key_phrases)
32         sentence_scores.append((i, score))
33
34     sorted_sentences = sorted(sentence_scores, key=lambda x: x[1], reverse=True)
35     selected_sentences = sorted(sorted_sentences[num_sentences:], key=lambda x: x[0])
36     summary = ' '.join([sentences[i] for i, _ in selected_sentences])
37     return summary
38
39 # Flask App
40 from flask import Flask, request, jsonify
41 from utils.summarizer import test_summarizer
42
43 flask_app = Flask(__name__)
44 main_routes = Blueprint('main', __name__)
45
46 @main_routes.route('/', methods=['GET'])
47 def home():
48     return jsonify({'message': 'Welcome to the Text Summarizer API!'}), 200
49
50 @main_routes.route('/summarize', methods=['POST'])
51 def summarize():
52     data = request.get_json()
53     text = data.get('text')
54     num_sentences = data.get('num_sentences', 3)
55     if not text:
56         return jsonify({'error': 'Text is required'}), 400
57     summary = test_summarizer(text, num_sentences)
58     return jsonify({'summary': summary})
```

Penjelasan

Gambar di atas adalah hasil *progress* pembuatan model *text-summarization* yang sedang saya lakukan. Di kode itu saya masih membuat teks yang di ringkas dalam bentuk statis. Barulah ketika saya rasa modelnya sudah cukup sesuai seperti yang saya inginkan, akan saya buat menjadi dinamis. Disini, kenapa saya menggunakan VSCode adalah karena saya berniat untuk membuatnya menjadi *endpoint*, dan nanti

dapat di gunakan untuk tampilannya, walaupun di Google Collab juga bisa, namun saya lebih nayman menggunakan VSCode ini.

Hambatan

Sejauh ini hambatan yang saya rasa sulit terletak pada modelnya itu sendiri, dimana hasil ringkasannya menurut sya belum enak dibaca walau sudah ringkas. Jadi, saya akna mencoba mengembangkan modelnya lagi agar bisa mendekati atau bahkan melebihi harpaan saya sendiri.