Nama : Muhamad Faqih Azhar

NIM : 0110221092

Kelas : 5TI03 / Senin Malam

Repository

https://github.com/FaqihAzh/nlp-task/tree/main/week-4

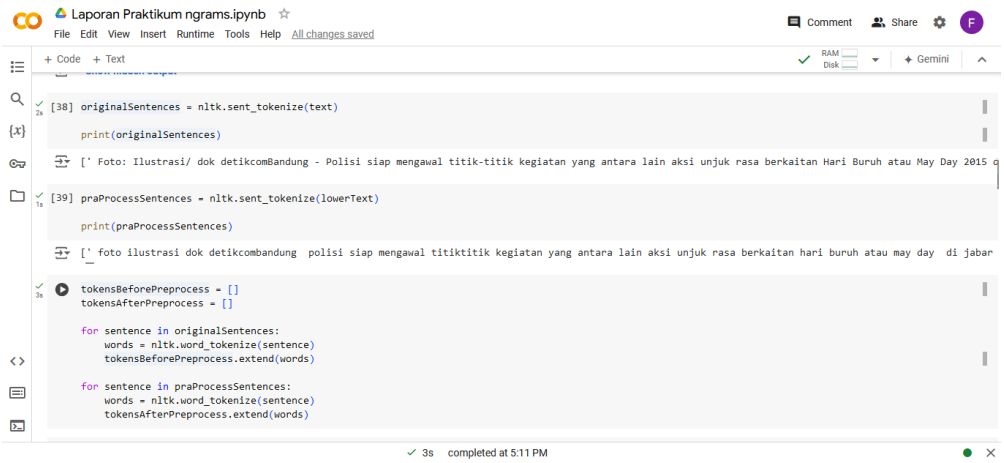
Pertanyaan

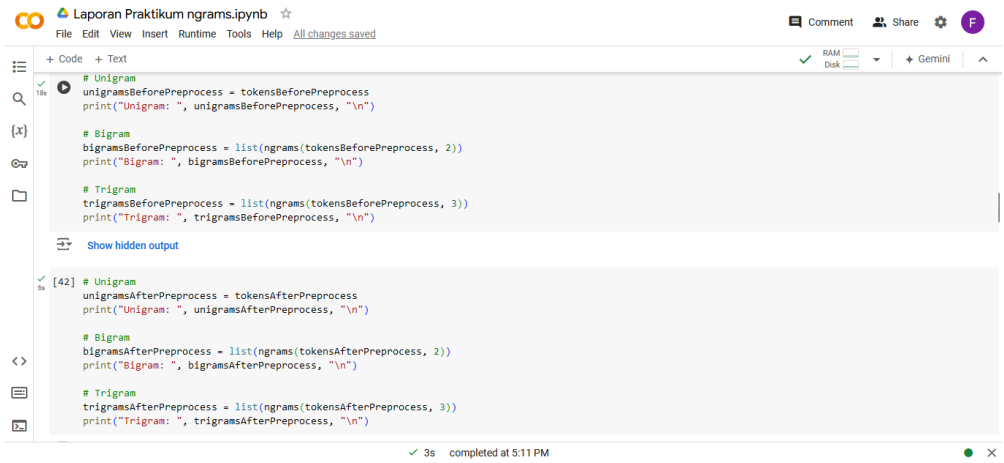
1. Lakukan pra-proses terhadap data teksnya terlebih dahulu
   a. Hapus simbol, angka, dan token yang tidak umum,
   b. Ubah semua teks menjadi lowercase

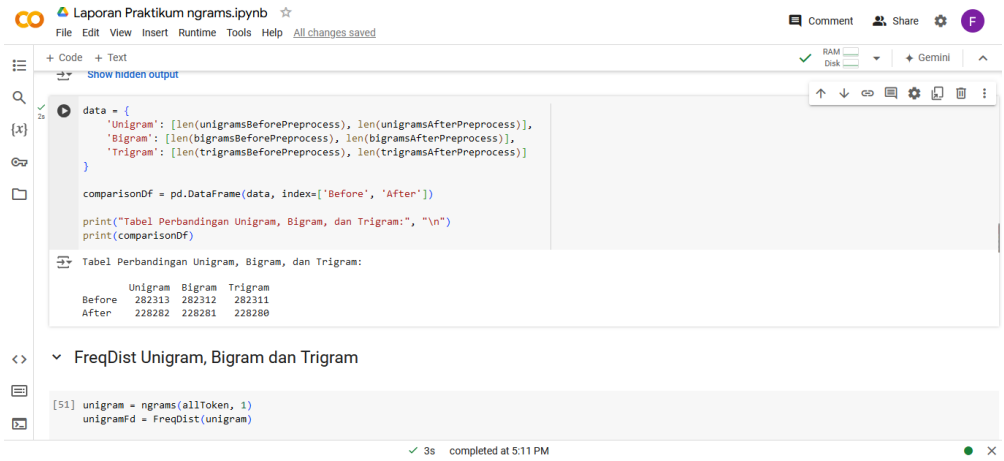
   Jawab:

```python
from bs4 import BeautifulSoup as BS
import re

with open('artikel1k.txt', 'r') as artikelFile:
  artikelData = artikelFile.read()

soup = BS(artikelData, 'html.parser')

allText = soup.findAll("div", {"class": "detail_text"})
text = " ".join([x.getText() for x in allText])

# Lakukan pra-proses terhadap data teksnya terlebih dahulu
# Hapus simbol, angka, dan token yang tidak umum
cleanedText = re.sub(r'\s+', ' ', text) # Menghilangkan spasi/jarak berlebih pada text
cleanedText = re.sub(r'[^a-zA-Z\s]', '', text) # Menghilangkan semua karakter simbol kecuali huruf dan spasi pada text

# Ubah semua teks menjadi lowercase
lowerText = cleanedText.lower()
```

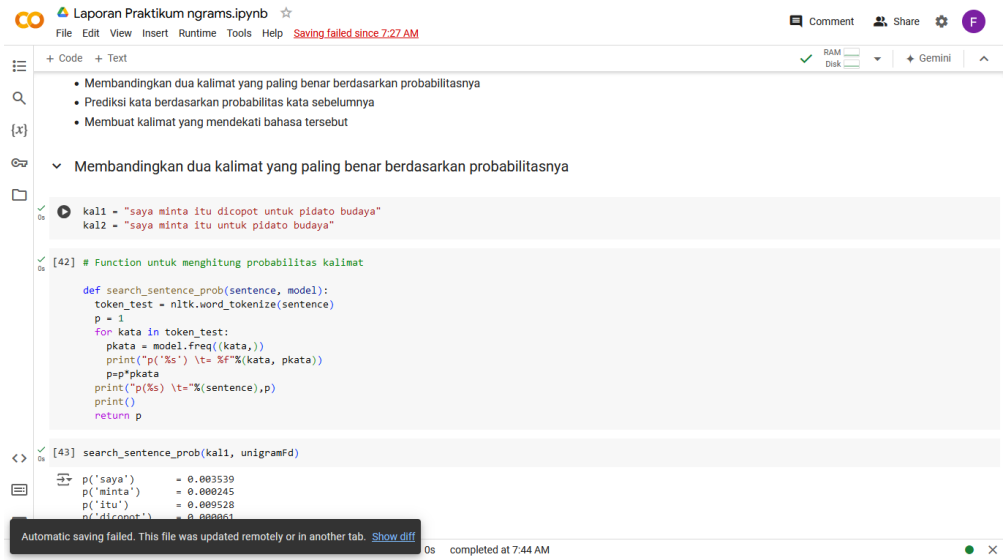
2. Hitung berapa banyak token unigram, bigram, dan trigramnya sekarang kemudian bandingkan dengan kondisi awal yang sebelum pra-proses!

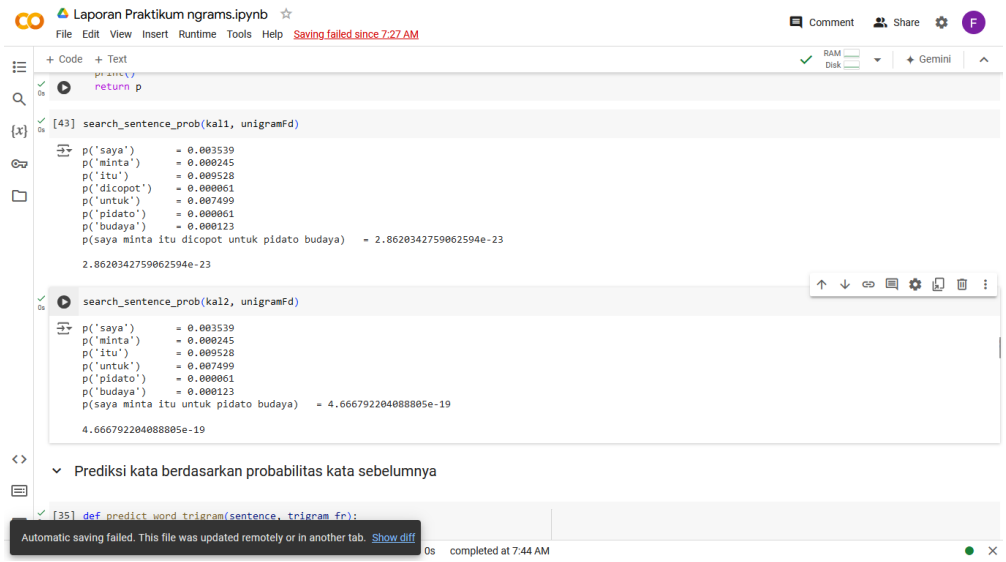




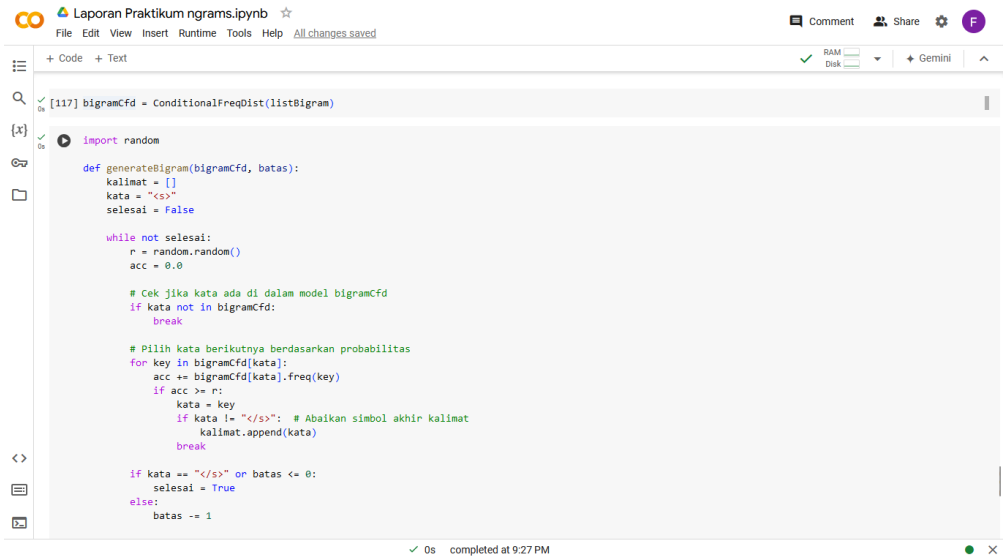


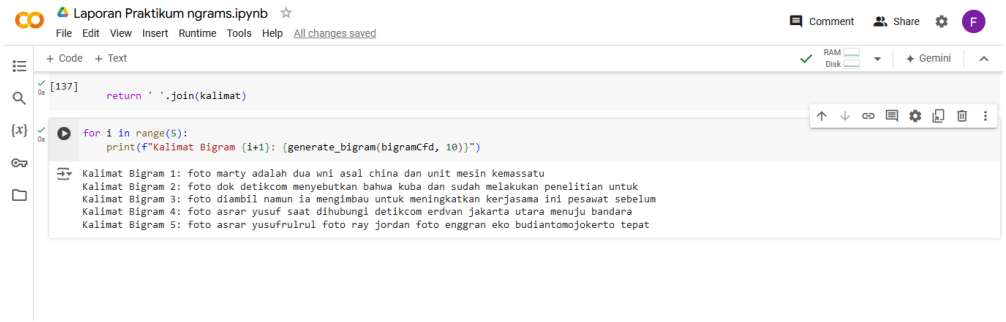
3. Hitung ulang peluang dari kedua kalimat yang diberikan (kal1 dan kal2), mana yang lebih mungkin untuk terjadi?



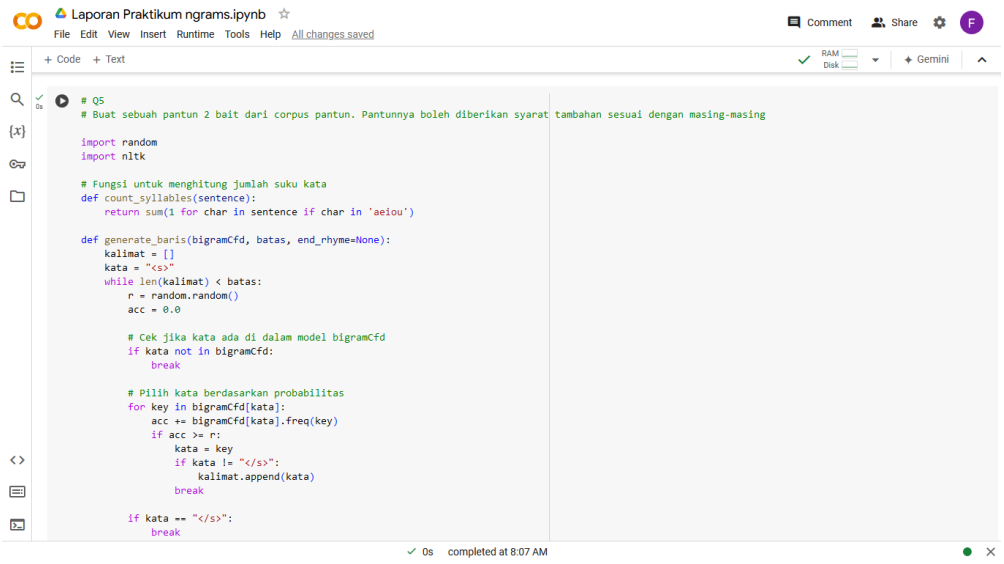


4. *Generate* 5 buah kalimat dari corpus berita yang diberikan! Diperbolehkan memanfaatkan bigram, trigram, atau ngram di atasnya.





5. Buat sebuah pantun 2 bait dari corpus pantun dan diperbolehkan memberikan syarat tambahan sesuai dengan masing-masing!

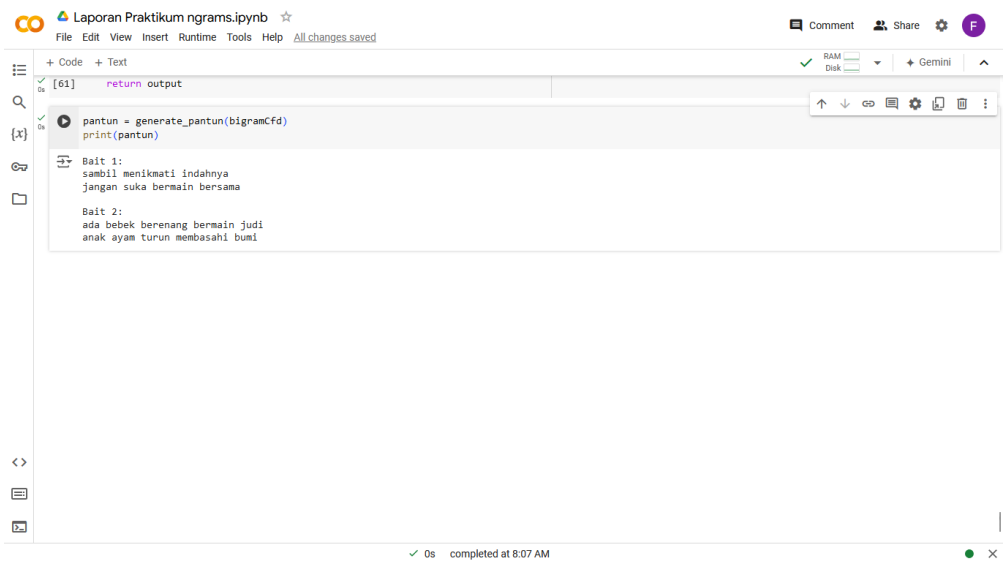


```
# Q5
# Buat sebuah pantun 2 bait dari corpus pantun. Pantunnya boleh diberikan syarat tambahan sesuai dengan masing-masing

import random
import nltk

# Fungsi untuk menghitung jumlah suku kata
def count_syllables(sentence):
    return sum(1 for char in sentence if char in 'aeiou')

def generate_baris(bigramCfd, batas, end_rhyme=None):
    kalimat = []
    kata = "<s>"
    while len(kalimat) < batas:
        r = random.random()
        acc = 0.0

        # Cek jika kata ada di dalam model bigramCfd
        if kata not in bigramCfd:
            break

        # Pilih kata berdasarkan probabilitas
        for key in bigramCfd[kata]:
            acc += bigramCfd[kata].freq(key)
            if acc >= r:
                kata = key
                if kata != "</s>":
                    kalimat.append(kata)
                break

        if kata == "</s>":
            break
```

0s completed at 8:07 AM

```
[61]        if kata == "</s>":
                break

        # Lanju jika ada kalimat yang dihasilkan
        if len(kalimat) == 0:
            # Jika kosong, ulang
            return generate_baris(bigramCfd, batas, end_rhyme)
        sentence = ' '.join(kalimat)

        # Cek jumlah suku kata
        if not (8 <= count_syllables(sentence) <= 12):
            # Jika tidak sesuai jumlah suku kata ulan
            return generate_baris(bigramCfd, batas, end_rhyme)

        # Cek rima
        if end_rhyme:
            if sentence.split()[-1][-len(end_rhyme):] != end_rhyme:
                return generate_baris(bigramCfd, batas, end_rhyme)  # Ga sesuai ulang

        return sentence

    # Fungsi buat pantun 2 bait
    def generate_pantun(bigramCfd):
        bait1 = [generate_baris(bigramCfd, 6, end_rhyme='a') for _ in range(2)]
        bait2 = [generate_baris(bigramCfd, 6, end_rhyme='i') for _ in range(2)]

        output = "Bait 1:\n" + "\n".join(bait1) + "\n\n" + "Bait 2:\n" + "\n".join(bait2)
        return output

pantun = generate_pantun(bigramCfd)
print(pantun)
```

0s completed at 8:07 AM


Laporan Praktikum ngrams.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
F
+ Code + Text
RAM
Disk
Gemini
[61] return output
pantun = generate_pantun(bigramCfd)
print(pantun)
Bait 1:
sambil menikmati indahnya
jangan suka bermain bersama
Bait 2:
ada bebek berenang bermain judi
anak ayam turun membasahi bumi
0s completed at 8:07 AM