```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
!ls /content/drive/MyDrive/Cosinus-Similarity
```

```
16595-37036-1-PB.pdf  cosim-230411100197-1.pdf  source2.txt                  source.tx
17167-37507-1-PB.pdf  feature_extraction.py     source3.txt                  stopwords
17342-37530-1-PB.pdf  mySource.txt              source4.txt                  term-frec
17526-37540-1-PB.pdf  README.md                 source5.txt                  tf-idf.cs
17838-37545-1-PB.pdf  source1.txt               source_raw_for_extraction.txt  Untitled.
```

```
!cp -r /content/drive/MyDrive/Cosinus-Similarity/* /content/
```

```
pip install numpy==1.26.4 -U scikit-learn Sastrawi
```

```
Requirement already satisfied: numpy==1.26.4 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (
Collecting Sastrawi
  Downloading Sastrawi-1.0.1-py2.py3-none-any.whl.metadata (909 bytes)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-pa
Downloading Sastrawi-1.0.1-py2.py3-none-any.whl (209 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.7/209.7 kB 5.5 MB/s eta 0:00:00
Installing collected packages: Sastrawi
Successfully installed Sastrawi-1.0.1
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import re
```

```
REGEX = re.compile(r"\s")
def tokenize(text):
    return [tok.strip().lower() for tok in REGEX.split(text)]

def stopwords(text):
    reg = re.compile(r"\n")
    return reg.split(text)
```

```
file = open("source1.txt","r");
raw1 = file.read()
```

```
file = open("source2.txt","r");
raw2 = file.read()

file = open("source3.txt","r");
raw3 = file.read()

file = open("source4.txt","r");
raw4 = file.read()

file = open("source5.txt","r");
raw5 = file.read()
```

```
# menghilangkan tanda baca
tandabaca = [".",",","-","%"]
for td in tandabaca:
    raw1=raw1.replace(td,"")
    raw2=raw2.replace(td,"")
    raw3=raw3.replace(td,"")
    raw4=raw4.replace(td,"")
    raw5=raw5.replace(td,"")
```

```
# menghilangkan stop words
file = open("stopwords.txt","r");
st = file.read()
stopwords = stopwords(st)

for word in stopwords:
    raw1=raw1.replace(" "+word+" "," ")
    raw2=raw2.replace(" "+word+" "," ")
    raw3=raw3.replace(" "+word+" "," ")
    raw4=raw4.replace(" "+word+" "," ")
    raw5=raw5.replace(" "+word+" "," ")
```

```
# stemming
factory = StemmerFactory()
stemmer = factory.create_stemmer()


hasilstem1 = stemmer.stem(raw1)
hasilstem2 = stemmer.stem(raw2)
hasilstem3 = stemmer.stem(raw3)
hasilstem4 = stemmer.stem(raw4)
hasilstem5 = stemmer.stem(raw5)
```

```
#tokenization
train_set = [hasilstem1,hasilstem2,hasilstem3,hasilstem4,hasilstem5]
```

```
count_vectorizer = CountVectorizer(tokenizer=tokenize)
data = count_vectorizer.fit_transform(train_set).toarray()
vocab = count_vectorizer.get_feature_names_out()
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/feature_extraction/text.py:517: UserWarr
  warnings.warn(
```

```
print("Jumlah Term FREQUENCY============================")
print(data)
```

```
Jumlah Term FREQUENCY============================
[[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 1]
 [0 1 1 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [1 0 1 ... 0 0 0]]
```

```
print("VECTOR FITUR============================")
print(vocab)
```

```
VECTOR FITUR============================
['1' '16' '2' '2019' '20192021' '2020' '2021' '2022' '22' '244' '2440'
 '2767' '33' '400' '88' '92' 'agung' 'ajar' 'akademik' 'akibat' 'alami'
 'alhamdulillah' 'ambil' 'anak' 'anakanak' 'angka' 'apa' 'area' 'asing'
 'aspek' 'asupan' 'badan' 'bahan' 'bahaya' 'bakar' 'balita' 'bangga'
 'bangku' 'bbm' 'beberapa' 'beda' 'bensin' 'bersih' 'bisa' 'bkkbn' 'buka'
 'buruk' 'campur' 'corporate' 'covid19' 'dampak' 'dari' 'dasar' 'data'
 'degree' 'dengar' 'departemen' 'didik' 'double' 'dua' 'duduk' 'efek'
 'endap' 'fakta' 'faktor' 'fungsi' 'gelar' 'gizi' 'hantu' 'harga' 'hasil'
 'hasto' 'heboh' 'heppy' 'hukum' 'indonesia' 'infeksi' 'institusi'
 'internasional' 'istilah' 'jakarta' 'jaksa' 'jenis' 'jenjang' 'jual'
 'jurus' 'juta' 'kampus' 'kandung' 'karat' 'kebal' 'kejakgung' 'keluarga'
 'kembang' 'kencana' 'kendara' 'kepala' 'khawatir' 'kilang' 'kok'
 'kompascom' 'kondisi' 'konsumen' 'korupsi' 'kotor' 'kronis' 'kualitas'
 'kuliah' 'kurang' 'kurun' 'kutip' 'label' 'lahir' 'laman' 'lemah' 'lho'
 'mahasiswa' 'major' 'maksimal' 'malang' 'mampu' 'masa' 'masalah'
 'masyaralat' 'mati' 'meni' 'mentah' 'mental' 'mesin' 'mesti' 'milik'
 'minyak' 'miskin' 'modus' 'nahkirakira' 'nasional' 'negeri' 'ni' 'niaga'
 'nikah' 'oplos' 'otak' 'pada' 'pandemi' 'panjang' 'pasang' 'pasar'
 'pasuk' 'patra' 'pendek' 'performa' 'perintah' 'periode' 'persen'
 'persentase' 'persentasi' 'persero' 'pertalite' 'pertama' 'pertamax'
 'pertamina' 'pertatec' 'picu' 'postur' 'potensi' 'premium' 'produk'
 'program' 'pt' 'pusat' 'rakernas' 'rencana' 'republikacoid' 'resiko'
 'ribu' 'ron' 'rupa' 'rusa' 'saat' 'sama' 'sambut' 'sebab' 'secretary'
 'sekitar' 'selasa' 'serius' 'sidi' 'sistem' 'sosioekonomi' 'ssgbi'
 'statistik' 'status' 'studi' 'stunting' 'survei' 'syukur' 'tahun'
 'tangan' 'tangki' 'tapi' 'tiap' 'tingkat' 'tubuh' 'tumbuh' 'turun'
 'turut' 'universitas' 'usaha' 'usut' 'virtual' 'vs' 'wardoyo' 'warning'
 'wulansari']
```

```
print("JUMLAH VECTOR FITUR==============================")
print(len(vocab))
```

```
JUMLAH VECTOR FITUR==============================
214
```

```
tfidf = TfidfVectorizer().fit_transform(train_set)
pairwise_similarity = tfidf * tfidf.T
```

```
print("Jumlah Term FREQUENCY-Inverse Document Frequency============================")
print(tfidf)
```

```
Jumlah Term FREQUENCY-Inverse Document Frequency============================
  (0, 155)      0.08143434842508683
  (0, 157)      0.16286869685017366
  (0, 174)      0.10093577586565702
  (0, 197)      0.10093577586565702
  (0, 177)      0.08143434842508683
  (0, 170)      0.10093577586565702
  (0, 78)       0.06759786545853118
  (0, 37)       0.10093577586565702
  (0, 121)      0.10093577586565702
  (0, 70)       0.10093577586565702
  (0, 101)      0.20187155173131405
  (0, 129)      0.24430304527526048
  (0, 124)      0.20187155173131405
  (0, 21)       0.08143434842508683
  (0, 205)      0.10093577586565702
  (0, 166)      0.4037431034626281
  (0, 158)      0.3257373937003473
  (0, 154)      0.10093577586565702
  (0, 146)      0.16286869685017366
  (0, 136)      0.16286869685017366
  (0, 206)      0.10093577586565702
  (0, 164)      0.10093577586565702
  (0, 96)       0.10093577586565702
  (0, 131)      0.10093577586565702
  (0, 138)      0.20187155173131405
  :     :
  (4, 26)       0.04921212032900189
  (4, 77)       0.04921212032900189
  (4, 53)       0.04921212032900189
  (4, 27)       0.04921212032900189
  (4, 113)      0.04921212032900189
  (4, 20)       0.14763636098700567
  (4, 83)       0.14763636098700567
  (4, 132)      0.04921212032900189
  (4, 135)      0.04921212032900189
  (4, 108)      0.04921212032900189
  (4, 111)      0.04921212032900189
  (4, 204)      0.09842424065800379
  (4, 117)      0.04921212032900189
  (4, 190)      0.04921212032900189
  (4, 75)       0.04921212032900189
```

```
(4, 81)       0.04921212032900189
(4, 55)       0.04921212032900189
(4, 25)       0.04921212032900189
(4, 54)       0.04921212032900189
(4, 64)       0.04921212032900189
(4, 150)      0.04921212032900189
(4, 16)       0.04921212032900189
(4, 198)      0.04921212032900189
(4, 85)       0.04921212032900189
(4, 76)       0.04921212032900189
```

```
print("Jumlah COSINE-SIMILARITY============================")
print(pairwise_similarity)
```

```
Jumlah COSINE-SIMILARITY============================
  (0, 3)        0.027849404319431885
  (0, 2)        0.0033137880667345075
  (0, 4)        0.011927700903354021
  (0, 1)        0.2054787218211708
  (0, 0)        1.0
  (1, 4)        0.005272786736020421
  (1, 2)        0.011764223817362907
  (1, 3)        0.06812478933504837
  (1, 1)        0.9999999999999997
  (1, 0)        0.2054787218211708
  (2, 3)        0.21408219079755897
  (2, 1)        0.011764223817362907
  (2, 4)        0.006305214120519257
  (2, 2)        1.0000000000000004
  (2, 0)        0.0033137880667345075
  (3, 2)        0.21408219079755897
  (3, 1)        0.06812478933504837
  (3, 3)        0.9999999999999982
  (3, 0)        0.027849404319431885
  (4, 1)        0.005272786736020421
  (4, 2)        0.006305214120519257
  (4, 4)        0.9999999999999991
  (4, 0)        0.011927700903354021
```

Kesimpulan Perbandingan Kesamaan source1.txt hingga source5.txt

Berdasarkan analisis cosine similarity dan term frequency-inverse document frequency (TF-IDF), teridentifikasi bahwa kelima dokumen terbagi ke dalam tiga kelompok tematik yang berbeda. Dokumen 1 (S1) dan Dokumen 2 (S2) membentuk kelompok pertama dengan fokus pada topik bahan bakar kendaraan, khususnya perbandingan "Pertalite dan Pertamax". Kedua dokumen ini memiliki kemiripan kata kunci seperti pertalite, pertamax, bensin, mesin, karat, dan tangki, yang mencerminkan pembahasan teknis seputar karakteristik bahan bakar, dampak pencampuran, serta risiko kerusakan mesin. Nilai cosine similarity antara S1 dan S2 sebesar 0.2055 menunjukkan bahwa keduanya berbagi konteks umum meski tidak sepenuhnya identik. S1 lebih menitikberatkan

pada perbandingan produk, sementara S2 membahas efek teknis pencampuran bahan bakar terhadap tangki kendaraan.

Kelompok kedua terdiri dari Dokumen 3 (S3) dan Dokumen 4 (S4), yang sama-sama membahas isu "stunting" (gagal tumbuh) pada anak di Indonesia. Keduanya memiliki kata kunci dominan seperti stunting, anak, angka, persen, tahun, pandemi, dan gizi, dengan nilai cosine similarity tertinggi antardokumen (0.2141). S3 fokus pada pencapaian penurunan angka stunting sebesar 33% dalam kurun 2019-2021, sementara S4 mengulas dampak pandemi COVID-19 terhadap peningkatan risiko stunting akibat keterbatasan akses gizi dan layanan kesehatan. Meski berbeda sudut pandang, keduanya saling melengkapi dalam membahas faktor penyebab dan upaya penanganan stunting.

Dokumen 5 (S5) berdiri sendiri sebagai kelompok ketiga dengan topik yang sama sekali berbeda, yakni perbandingan istilah akademik double degree dan double major. Kata kunci unik seperti double degree, major, universitas, akademik, dan jenjang menunjukkan fokus pada sistem pendidikan tinggi. Nilai cosine similarity S5 dengan dokumen lain mendekati 0, menegaskan ketidakterkaitan tematiknya dengan kelompok bahan bakar (S1-S2) atau kesehatan anak (S3-S4).

Secara keseluruhan, analisis ini mengungkap dua klaster tematik utama (bahan bakar dan stunting) serta satu dokumen outlier (pendidikan). Kemiripan tertinggi berada di dalam klaster masing-masing, sementara perbedaan antarklaster sangat signifikan, seperti terlihat dari rendahnya nilai similarity antara S1/S2 dengan S3/S4 (sekitar 0.03-0.07). Hal ini menunjukkan bahwa metode cosine similarity efektif mengelompokkan dokumen berdasarkan kesamaan kata kunci dan konteks, meskipun terdapat variasi dalam sudut pembahasan di dalam klaster yang sama. Dokumen-dokumen dalam satu klaster juga mencerminkan dinamika topik yang relevan dengan isu aktual, seperti kebijakan energi (S1-S2) dan kesehatan masyarakat (S3-S4), sementara S5 merepresentasikan niche informasi di bidang pendidikan tinggi