

*Syayid Al Aziz*

---

# ***Repository Tugas Pencarian dan Penambahan Web***

To blah, blah, and blah.

---

## *Table of contents*

---

<b>Tentang</b>	<b>v</b>
<b>Tentang</b>	<b>v</b>
<b>Crawl PTA Trunojoyo</b>	<b>vii</b>
<b>Crawling Data</b>	<b>ix</b>
<b>Topic Modeling PTA</b>	<b>xi</b>
<b>Instalisasi</b>	<b>xiii</b>
<b>Data</b>	<b>xv</b>
<b>Preprocessing Data</b>	<b>xvii</b>
0.1 Tokenisasi . . . . .	xvii
0.2 Process Punctuation . . . . .	xviii
0.3 Stopword . . . . .	xix
0.4 Steeming . . . . .	xx
<b>Feature Extraction</b>	<b>xxi</b>
0.5 Local Weighting . . . . .	xxii
0.6 TF-IDF . . . . .	xxiv
0.7 LDA Topic . . . . .	xxiv
<b>Clustering</b>	<b>xxvii</b>
0.8 Clustering TF-IDF . . . . .	xxvii
0.9 Clustering LDA Topic . . . . .	xxviii
<b>Classification</b>	<b>xxxi</b>
0.10 TF-IDF . . . . .	xxxii
0.11 LDA Topic . . . . .	xxxv
<b>Crawl Berita CNN</b>	<b>xli</b>
<b>Crawling Data</b>	<b>xliii</b>
<b>Ringkasan Berita</b>	<b>xlvi</b>

<b>Instalasi</b>	<b>xlix</b>
<b>Data</b>	<b>li</b>
<b>Preprocessing</b>	<b>liii</b>
<b>Ekstraksi Fitur</b>	<b>lv</b>
0.12 TF-IDF . . . . .	lv
<b>Membentuk Graph</b>	<b>lix</b>
0.13 Cosine Similarity . . . . .	lix
0.14 Graph . . . . .	lxi
<b>Matriks Sentralitas</b>	<b>lxv</b>
0.15 Closeness Centrality . . . . .	lxv
0.16 Page Rank . . . . .	lxvii
0.17 Eigen Vector . . . . .	lxx
<b>Evaluasi</b>	<b>lxxiii</b>
0.18 Closeness Centrality . . . . .	lxxiii
0.19 Page Rank . . . . .	lxxiv
0.20 Eigen Vector . . . . .	lxxiv

---

## *Tentang*

---

Berikut merupakan repository yang saya buat berisikan tugas pada mata kuliah Pencarian dan Penambangan Web, dalam repository ini berisikan topic modeling pada tugas akhir Universitas Trunojoyo Madura dan ringkasan berita pada yang datanya diambil pada website CNN Indonesia, tidak hanya itu terdapat juga beberapa langkah-langkah mulai dari pengambilan data (crawling) dan tahapan utama hingga selesai dari kedua pekerjaan ini yaitu topic modeling dan ringkasan dokumen.



# 0

## *Crawl PTA Trunojoyo*

```
!pip install requests
!pip install beautifulsoup4
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.31.0)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (3.6)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (2.0.7)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (2023.7.22)

Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (4.11.2)

Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (2.5)





# 0

## *Crawling Data*

```
def ptaa():
    data = {"penulis": [], "judul": [], "pembimbing_pertama": [], "pembimbing_kedua": [], "abst

    for i in range(1, 4):
        url = "https://pta.trunojoyo.ac.id/c_search/byprod/10/{}".format(i)
        r = requests.get(url)
        request = r.content
        soup = BeautifulSoup(request, "html.parser")
        journals = soup.select('li[data-cat="#luxury"]')

        for jurnal in journals:
            response = requests.get(jurnal.select_one('a.gray.button')['href'])
            soup1 = BeautifulSoup(response.content, "html.parser")

            isi = soup1.select_one('div#content_journal')

            judul = isi.select_one('a.title').text

            penulis = isi.select_one('span:contains("Penulis")').text.split(' : ')[1]
            # penulis = penulis_span.find_next('span').text.split(' : ')

            pembimbing_pertama = isi.select_one('span:contains("Dosen Pembimbing I")').text.spl
            # pembimbing_pertama = pembimbing_pertama_span.find_next('span')

            pembimbing_kedua = isi.select_one('span:contains("Dosen Pembimbing II")').text.spli
            # pembimbing_kedua = pembimbing_kedua_span.find_next('span')

            abstrak = isi.select_one('p[align="justify"]').text
            if abstrak == '':
                abstrak = ' '.join(isi.find('p').findNext('p').stripped_strings).capitalize()

            data["penulis"].append(penulis)
            data["judul"].append(judul)
            data["pembimbing_pertama"].append(pembimbing_pertama)
```

```

        data["pembimbing_kedua"].append(pembimbing_kedua)
        data["abstrak"].append(abstrak)

    df = pd.DataFrame(data)
    df.to_csv("pta.csv", index=False)

    return df

```

```
pta()
```

	penulis	judul
0	A.Ubaidillah S.Kom	PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE \...
1	M. Basith Ardianto,	APLIKASI KONTROL DAN MONITORING JARINGAN KOMPU...
2	Akhmad Suyandi, S.Kom	RANCANG BANGUN APLIKASI PROXY SERVER UNTUK\r\n...
3	Heri Supriyanto	SISTEM PENDUKUNG KEPUTUSAN OPTIMASI PENJADWALA...
4	Septian Rahman Hakim	SISTEM AUGMENTED REALITY ANIMASI BENDA BERGERA...
5	Adi Chandra Laksono	Gerak Pekerja Pada Game Real Time Strategy Men...
6	NURRACHMAT	RANCANG BANGUN GAME PERAWATAN SAPI KARAPAN MEN...
7	Muhammad Choirur Rozi	EKSTRAKSI FITUR BERBASIS TWO DIMENSIONAL LINEA...
8	M Khoiril Anwar	IMPLEMENTASI ALGORITMA PRIM DAN DEPTH FIRST ...
9	MALIKUL HAMZAH	Perancangan Sistem Informasi Badan Kepegawaian...
10	Norman	PEMANFAATAN TOGAF ADM UNTUK PERANCANGAN SISTEM...
11	Robiatul Adawiyah, S.Kom	APLIKASI METODE FUZZY ANALYTIC NETWORK PROCESS...
12	Desy Mariana S. Kom	SISTEM PENDUKUNG KEPUTUSAN REKOMENDASI MENU DI...
13	Lia Fransiska	RANCANG BANGUN APLIKASI PEMILIHAN TEKNIK REKAY...
14	Erwina Safitri	DETEKSI COREPOINT SIDIK JARI MENGGUNAKAN METOD...

# 0

## *Topic Modeling PTA*



# 0

## *Instalasi*

```
!pip install nltk
!pip install Sastrawi
!pip install gensim
```

Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)  
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)  
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)  
Requirement already satisfied: regex<=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)  
Requirement already satisfied: Sastrawi in /usr/local/lib/python3.10/dist-packages (1.0.1)  
Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (4.3.2)  
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from gensim)  
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from gensim)  
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim)

```
import pandas as pd
import nltk
import gensim
import re
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.corpus import stopwords
from gensim import corpora
from gensim.models import LdaModel
from gensim.models.coherencemodel import CoherenceModel
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import ConfusionMatrixDisplay, confusion_matrix, accuracy_score, precision
```

```
nltk.download("punkt")  
nltk.download("stopwords")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...  
[nltk_data]   Package punkt is already up-to-date!  
[nltk_data] Downloading package stopwords to /root/nltk_data...  
[nltk_data]   Package stopwords is already up-to-date!
```

```
True
```

# 0

## Data

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

csv_path = '/content/drive/My Drive/Study Of Informatika/Pencarian dan Penambangan Web/Tugas 1/
df = pd.read_csv(csv_path)
df
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive")

	penulis	judul
0	A.Ubaidillah S.Kom	PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE T...
1	M. Basith Ardianto,	APLIKASI KONTROL DAN MONITORING JARINGAN KOMPU...
2	Akhmad Suyandi, S.Kom	RANCANG BANGUN APLIKASI PROXY SERVER UNTUK\nEN...
3	Heri Supriyanto	SISTEM PENDUKUNG KEPUTUSAN OPTIMASI PENJADWALA...
4	Septian Rahman Hakim	SISTEM AUGMENTED REALITY ANIMASI BENDA BERGERA...
...	...	...
826	Rachmad Agung Pambudi	PENERAPAN ALGORITMA LONG-SHORT TERM MEMORY UNT...
827	Nadila Hidayanti	SISTEM PENCARIAN TEKS AL-QURAN TERJEMAHAN BERB...
828	Afni Sakinah	KLASIFIKASI KOMPLEKSITAS VISUAL CITRA SAMPAH M...
829	Friska Fatmawatiningrum	IDENTIFIKASI BINER ATRIBUT PEJALAN KAKI MENGGU...
830	Dian Wibowo	DETEKSI OBJEK MANUSIA BERBASIS ONE STAGE DETEC...

```
count_komputasi = (df['label-topic'] == 'komputasi').sum()
count_rpl = (df['label-topic'] == 'rpl').sum()

print('komputasi = ', count_komputasi, 'rpl = ', count_rpl)
```

komputasi = 548 rpl = 283





# 0

## *Preprocessing Data*

Text preprocessing merupakan tahapan dalam Natural Language Processing (NLP) yang bertujuan untuk membersihkan dan menyiapkan teks mentah menjadi data siap digunakan pada proses yang lebih lanjut. Text preprocessing ini menjadi sangat krusial karena membantu mengatasi berbagai masalah seperti data berantakan, tanda baca yang berlebihan, serta variasi dalam bentuk kata

```
df=df.astype(str)
df["abstrak"] = df["abstrak"].apply(lambda x: x.lower())

abstrak_column = df["abstrak"]
abstrak_column
```

```
0      sistem informasi akademik (siakad) merupaka...
1      berjalannya koneksi jaringan komputer dengan l...
2      web server adalah sebuah perangkat lunak serve...
3      penjadwalan kuliah di perguruan tinggi me...
4      seiring perkembangan teknologi yang ada diduni...
...
826     investasi saham selama ini memiliki resiko ker...
827     information retrieval (ir) merupakan pengambil...
828     klasifikasi citra merupakan proses pengelompok...
829     identifikasi atribut pejalan kaki merupakan sa...
830     topik deteksi objek telah menarik perhatian ya...
Name: abstrak, Length: 831, dtype: object
```

### 0.1 Tokenisasi

Tokenizing adalah proses yang mengubah teks berkelanjutan menjadi unit-unit yang lebih kecil dan disebut dengan token. Token ini biasanya adalah kata, frasa, atau tanda baca yang memisahkan kata-kata dalam teks, dengan ini akan memudahkan untuk melakukan analisis terhadap teks dan akan

membantu menyaring kata-kata yang tidak diinginkan pada pemrosesan teks lebih lanjut

```
def process_tokenize(text):
    text = text.split()
    return text

tokenize_abstrak = abstrak_column.apply(process_tokenize)
tokenize_abstrak

# token = pd.DataFrame(df, columns = tokenize_abstrak)

0      [sistem, informasi, akademik, (siakad), merupa...
1      [berjalannya, koneksi, jaringan, komputer, den...
2      [web, server, adalah, sebuah, perangkat, lunak...
3      [penjadwalan, kuliah, di, perguruan, tinggi, m...
4      [seiring, perkembangan, teknologi, yang, ada, ...
      ...
826     [investasi, saham, selama, ini, memiliki, resi...
827     [information, retrieval, (ir), merupakan, peng...
828     [klasifikasi, citra, merupakan, proses, pengel...
829     [identifikasi, atribut, pejalan, kaki, merupak...
830     [topik, deteksi, objek, telah, menarik, perhat...
Name: abstrak, Length: 831, dtype: object
```

## 0.2 Process Punctuation

Proses punctuation ini digunakan untuk menghapus karakter yang tidak digunakan pada teks, seperti tanda baca dan angka. Beberapa karakter ini harus dihilangkan karena karakter tersebut tidak akan mempengaruhi hasil pada saat melakukan klasifikasi topic

```
def process_punctuation(tokens):
    cleaned_tokens = [re.sub(r'[.,()&=:~]', '', token) for token in tokens]
    cleaned_tokens = [re.sub(r'\d+', '', token) for token in cleaned_tokens]

    return cleaned_tokens

punctuation_abstrak = tokenize_abstrak.apply(process_punctuation)

# data = pd.DataFrame(df, columns=['punctuation_abstrak'])
```

```
# data
punctuation_abstrak

0      [sistem, informasi, akademik, siacad, merupaka...
1      [berjalannya, koneksi, jaringan, komputer, den...
2      [web, server, adalah, sebuah, perangkat, lunak...
3      [penjadwalan, kuliah, di, perguruan, tinggi, m...
4      [seiring, perkembangan, teknologi, yang, ada, ...
      ...
826     [investasi, saham, selama, ini, memiliki, resi...
827     [information, retrieval, ir, merupakan, pengam...
828     [klasifikasi, citra, merupakan, proses, pengel...
829     [identifikasi, atribut, pejalan, kaki, merupak...
830     [topik, deteksi, objek, telah, menarik, perhat...
Name: abstrak, Length: 831, dtype: object
```

---

### 0.3 Stopword

Remove Stopword adalah tahap yang melibatkan penghapusan kata-kata umum yang dianggap tidak memberikan nilai signifikan dalam analisis teks. Kata-kata semacam ini disebut stop words karena mereka sering muncul dalam teks bahasa alami tanpa memberikan informasi penting tentang isi atau makna teks. Contoh kata-kata umum ini termasuk “dan”, “atau”, “di”, “dari”, “yang”, “itu”, dan sebagainya. Penghapusan stop words bertujuan untuk mengurangi ukuran teks, mempercepat pemrosesan, dan meningkatkan relevansi informasi yang diambil dari teks, sehingga hanya kata kunci yang membentuk topik yang akan diekstraksi.

```
def process_stopword_token(tokens):
    stop_words = set(stopwords.words("indonesian"))
    custom_stop_words = ['masingmasing', 'tiap', 'satunya', 'intinya', 'seiring']
    stop_words.update(custom_stop_words)
    filtered_tokens = [token for token in tokens if token.lower() not in stop_words]
    return " ".join(filtered_tokens)

stopword_abstrak = punctuation_abstrak.apply(process_stopword_token)
stopword_abstrak

0      sistem informasi akademik siacad sistem inform...
1      berjalannya koneksi jaringan komputer lancar g...
2      web server perangkat lunak server berfungsi me...
```

```

3    penjadwalan kuliah perguruan kompleks permasal...
4    perkembangan teknologi didunia muncul teknolog...
...
826  investasi saham memiliki resiko kerugian perge...
827  information retrieval ir pengambilan informasi...
828  klasifikasi citra proses pengelompokan piksel ...
829  identifikasi atribut pejalan kaki salah peneli...
830  topik deteksi objek menarik perhatian perkemba...
Name: abstrak, Length: 831, dtype: object

```

## 0.4 Steeming

Stemming merupakan proses untuk mengurangi kata-kata ke bentuk dasarnya atau akar kata. Tujuannya adalah untuk mengidentifikasi kata-kata yang memiliki akar yang sama, meskipun mereka mungkin memiliki akhiran atau imbuhan yang berbeda

```

def process_stemming(text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    return stemmer.stem(text)

steeming_abstrak = stopword_abstrak.apply(process_stemming)
steeming_abstrak

```

```

0    sistem informasi akademik siacad sistem inform...
1    jalan koneksi jaring komputer lancar ganggu ha...
2    web server perangkat lunak server fungsi terim...
3    jadwal kuliah guru kompleks masalah variabel t...
4    kembang teknologi dunia muncul teknologi augme...
...
826  investasi saham milik resiko rugi gera harga s...
827  information retrieval ir ambil informasi simpa...
828  klasifikasi citra proses kelompok piksel citra...
829  identifikasi atribut pejal kaki salah teliti k...
830  topik deteksi objek tarik perhati kembang tekn...
Name: abstrak, Length: 831, dtype: object

```

```

df['stopword-abstrak'] = stopword_abstrak
df['steeming-abstrak'] = steeming_abstrak
df.to_csv('/content/drive/My Drive/Study Of Informatika/Pencarian dan Penambangan Web/Tugas 1/d

```

# 0

## *Feature Extraction*

Feature extraction ini digunakan untuk mengubah keseluruhan teks dalam dokumen menjadi angka numerik dengan menggunakan algoritma TF-IDF dan LDA Topic

```
csv_preprocessing = '/content/drive/My Drive/Study Of Informatika/Pencarian dan Penambangan Web  
data = pd.read_csv(csv_preprocessing)  
data
```

	penulis	judul
0	A.Ubaidillah S.Kom	PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE T...
1	M. Basith Ardianto,	APLIKASI KONTROL DAN MONITORING JARINGAN KOMPU...
2	Akhmad Suyandi, S.Kom	RANCANG BANGUN APLIKASI PROXY SERVER UNTUK\ nEN...
3	Heri Supriyanto	SISTEM PENDUKUNG KEPUTUSAN OPTIMASI PENJADWALA...
4	Septian Rahman Hakim	SISTEM AUGMENTED REALITY ANIMASI BENDA BERGERA...
...	...	...
826	Rachmad Agung Pambudi	PENERAPAN ALGORITMA LONG-SHORT TERM MEMORY UNT...
827	Nadila Hidayanti	SISTEM PENCARIAN TEKS AL-QURAN TERJEMAHAN BERB...
828	Afni Sakinah	KLASIFIKASI KOMPLEKSITAS VISUAL CITRA SAMPAH M...
829	Friska Fatmawatiningrum	IDENTIFIKASI BINER ATRIBUT PEJALAN KAKI MENGGU...
830	Dian Wibowo	DETEKSI OBJEK MANUSIA BERBASIS ONE STAGE DETEC...

```
get_steeming_abstrak = df["steeming-abstrak"]  
get_stopword_abstrak = df["stopword-abstrak"]
```

## 0.5 Local Weighting

```
countvectorizer = CountVectorizer(analyzer= 'word')
term_matrix = countvectorizer.fit_transform(get_steeming_abstrak)
count_tokens = countvectorizer.get_feature_names_out()
df_countvect = pd.DataFrame(data = term_matrix.toarray(),columns = count_tokens)
print('Term Frequency\n')
df_countvect

# Transpose DataFrame
# df_countvect_transposed = df_countvect.T

# Print transposed DataFrame
# print('Term Frequency Transposed\n')
# df_countvect_transposed
```

Term Frequency

	aalysis	aam	ab	abad	abadi	abai	abdi	ability	abjad	absah	...	zara	zat	zcz	zf	zona
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
826	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
827	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
828	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
829	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
830	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

```
countvectorizer = CountVectorizer(analyzer='word')

log_matrix = countvectorizer.fit_transform(get_steeming_abstrak)
count_tokens = countvectorizer.get_feature_names_out()

count_log_matrix = np.log1p(log_matrix)

df_log_countvect = pd.DataFrame(data=count_log_matrix.toarray(), columns=count_tokens)
```

```
print('Log Frequency\n')
df_log_countvect
```

Log Frequency

	aalysis	aam	ab	abad	abadi	abai	abdi	ability	abjad	absah	...	zara	zat	zcz	zf	zon
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
826	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
827	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
828	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
829	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
830	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

```
countvectorizer = CountVectorizer(analyzer='word', binary=True)

binary_matrix = countvectorizer.fit_transform(get_steeming_abstrak)
df_binary = pd.DataFrame(binary_matrix.toarray(), columns=countvectorizer.get_feature_names_out())

print('Binary\n')
df_binary
```

Binary

	aalysis	aam	ab	abad	abadi	abai	abdi	ability	abjad	absah	...	zara	zat	zcz	zf	zon
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
826	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
827	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
828	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
829	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
830	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0

TF-IDF adalah teknik yang digunakan untuk mengukur pentingnya kata-kata dalam suatu dokumen dalam konteks korpus dokumen yang lebih besar. Ini dapat digunakan untuk mengidentifikasi kata-kata kunci atau fitur penting dalam analisis teks dan Natural Language Processing

	aalysis	aam	ab	abad	abadi	abai	abdi	ability	abjad	absah	...	zat	zcz	zf	zona	zo
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
826	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
827	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
828	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
829	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
830	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

[illegible]



```

        random_state=100,
        passes=10,
        per_word_topics=True)

print(lda_model.print_topics())
doc_lda = lda_model[corpus]

[(0, '0.021*"citra" + 0.016*"metode" + 0.013*"hasil" + 0.013*"penelitian" + 0.011*"proses" + 0.010*"ga

topic_proportions_list = []

for index, doc in enumerate(corpus):
    topic_prop = lda_model.get_document_topics(doc)
    proportions = {f'Topic {i+1}': 0.0 for i in range(num_topics)}

    for topic in topic_prop:
        proportions[f'Topic {topic[0] + 1}'] = topic[1]
    topic_proportions_list.append(proportions)

topic_proportions = pd.DataFrame(topic_proportions_list)
topic_proportions_df = pd.DataFrame(topic_proportions_list)
topic_proportions_df.insert(0, 'judul', df['judul'])
topic_proportions_df.insert(4, 'label-topic', df['label-topic'])
topic_proportions_df

```

	judul	Topic 1	Topic 2	Topic 3
0	PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE T...	0.000000	0.831309	0.163000
1	APLIKASI KONTROL DAN MONITORING JARINGAN KOMPU...	0.000000	0.000000	0.993000
2	RANCANG BANGUN APLIKASI PROXY SERVER UNTUK\hEN...	0.542891	0.000000	0.453000
3	SISTEM PENDUKUNG KEPUTUSAN OPTIMASI PENJADWALA...	0.000000	0.000000	0.988000
4	SISTEM AUGMENTED REALITY ANIMASI BENDA BERGERA...	0.000000	0.000000	0.989000
...	...	...	...	...
826	PENERAPAN ALGORITMA LONG-SHORT TERM MEMORY UNT...	0.239679	0.000000	0.757000
827	SISTEM PENCARIAN TEKS AL-QURAN TERJEMAHAN BERB...	0.568052	0.000000	0.427000
828	KLASIFIKASI KOMPLEKSITAS VISUAL CITRA SAMPAH M...	0.995387	0.000000	0.000000
829	IDENTIFIKASI BINER ATRIBUT PEJALAN KAKI MENGGU...	0.994435	0.000000	0.000000
830	DETEKSI OBJEK MANUSIA BERBASIS ONE STAGE DETEC...	0.991774	0.000000	0.000000



# 0

## *Clustering*

Untuk melakukan clustering akan menggunakan hasil fitur yang didapatkan pada ekstraksi kalimat di tahapan sebelumnya dengan menggunakan TF-IDF dan LDA Topic. Kedua algoritma tersebut memiliki cara kerja yang berbeda maka dengan melakukan perbandingan nilai skor yang dihasilkan pada saat clustering dokumen maka nanti dapat diambil kesimpulan dari hasil ekstraksi fitur yang terbaik ketika menggunakan kedua algoritma tersebut.

```
def kmeans_clustering(data):
    scaler = StandardScaler()
    scaled_data = scaler.fit_transform(data)
    num_clusters = 3
    kmeans = KMeans(n_clusters=num_clusters, random_state=42)
    clusters = kmeans.fit_predict(scaled_data)
    return clusters
```

### 0.8 Clustering TF-IDF

```
tfidf_clusters = kmeans_clustering(tfidf_df.values)
silhouette_tfidf = silhouette_score(tfidf_df.values, tfidf_clusters)

tfidf_clusters_df = pd.DataFrame({'cluster':tfidf_clusters})
tfidf_clusters_df.insert(0, 'judul', data['judul'])
tfidf_clusters_df.insert(2, 'label-topic', data['label-topic'])

tfidf_clusters_df
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default v
warnings.warn(
```

	judul	cluster	label-topic
0	PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE T...	1	rpl
1	APLIKASI KONTROL DAN MONITORING JARINGAN KOMPU...	1	rpl
2	RANCANG BANGUN APLIKASI PROXY SERVER UNTUK\	1	rpl
3	SISTEM PENDUKUNG KEPUTUSAN OPTIMASI PENJADWALA...	1	komputasi
4	SISTEM AUGMENTED REALITY ANIMASI BENDA BERGERA...	1	komputasi
...	...	...	...
826	PENERAPAN ALGORITMA LONG-SHORT TERM MEMORY UNT...	1	komputasi
827	SISTEM Pencarian teks al-Quran terjemahan berb...	1	komputasi
828	KLASIFIKASI Kompleksitas visual citra sampah m...	0	komputasi
829	Identifikasi biner atribut pejalan kaki menggu...	1	komputasi
830	Deteksi objek manusia berbasis one stage detec...	1	komputasi

```
print('Silhouette TF-IDF', silhouette_tfidf)
```

Silhouette TF-IDF -0.005708632045599099

## 0.9 Clustering LDA Topic

```
topic_proportions_clusters = kmeans_clustering(topic_proportions.values)
silhouette_topic_proportions = silhouette_score(topic_proportions.values, topic_proportions_clusters)

topic_proportions_df = pd.DataFrame({'cluster':topic_proportions_clusters})
topic_proportions_df.insert(0, 'judul', data['judul'])
topic_proportions_df.insert(2, 'label-topic', data['label-topic'])

topic_proportions_df
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default v
warnings.warn(
```

	judul	cluster	label-topic
0	PERANCANGAN DAN IMPLEMENTASI SISTEM DATABASE T...	2	rpl
1	APLIKASI KONTROL DAN MONITORING JARINGAN KOMPU...	0	rpl
2	RANCANG BANGUN APLIKASI PROXY SERVER UNTUK\	1	rpl
3	SISTEM PENDUKUNG KEPUTUSAN OPTIMASI PENJADWALA...	0	komputasi
4	SISTEM AUGMENTED REALITY ANIMASI BENDA BERGERA...	0	komputasi
...	...	...	...
826	PENERAPAN ALGORITMA LONG-SHORT TERM MEMORY UNT...	0	komputasi

	judul	cluster	label-topic
827	SISTEM PENCARIAN TEKS AL-QURAN TERJEMAHAN BERB...	1	komputasi
828	KLASIFIKASI KOMPLEKSITAS VISUAL CITRA SAMPAH M...	1	komputasi
829	IDENTIFIKASI BINER ATRIBUT PEJALAN KAKI MENGGU...	1	komputasi
830	DETEKSI OBJEK MANUSIA BERBASIS ONE STAGE DETEC...	1	komputasi

```
print('Silhouette LDA', silhouette_topic_proportions)
```

Silhouette LDA 0.7976753714213479



# 0

## *Classification*

Sama halnya pada clustering, proses klasifikasi kali ini akan membandingkan antara kedua algoritma ekstraksi fitur yaitu TF-IDF dan LDA Topic sehingga nanti akan dihasilkan perbandingan antara akurasi terbaik ketika menggunakan LDA atau TF-IDF

### 0.10 TF-IDF

```
X = tfidf_df
y = data['label-topic']
y = y.replace({'komputasi': 1, 'rpl': 0})

def train_knn_classifier(k):

    # train model knn
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    knn_classifier = KNeighborsClassifier(n_neighbors=k)
    knn_classifier.fit(X_train, y_train)

    y_pred = knn_classifier.predict(X_test)

    # confusion matrix
    cm = confusion_matrix(y_test, y_pred)
    disp = ConfusionMatrixDisplay(confusion_matrix=cm)

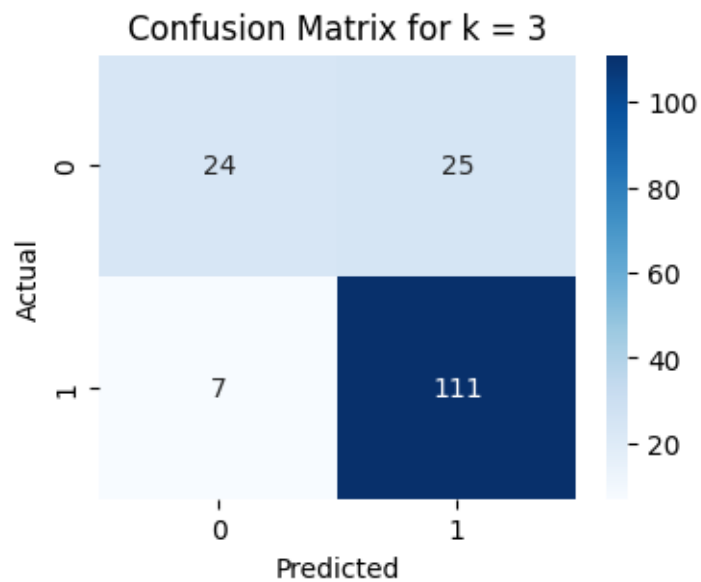
    plt.figure(figsize=(4, 3))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title(f'Confusion Matrix for k = {k}')
```

```
plt.show()

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(
    '\nHasil Evaluasi Nilai K adalah',k,
    '\nAccuracy =', accuracy,
    '\nPrecision =', precision,
    '\nRecall =', recall,
    '\nF1 Score =', f1
)

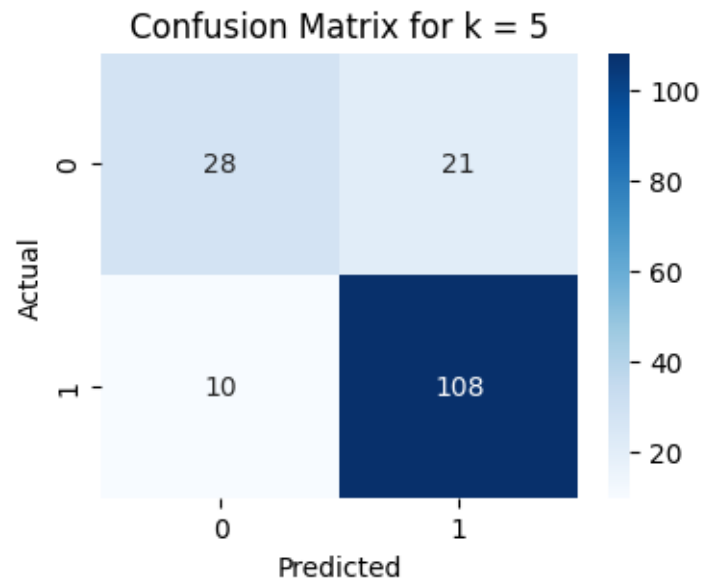
train_knn_classifier(3)
```



```
Hasil Evaluasi Nilai K adalah 3
Accuracy = 0.8083832335329342
Precision = 0.8161764705882353
Recall = 0.940677966101695
F1 Score = 0.8740157480314961
```



```
train_knn_classifier(5)
```



Hasil Evaluasi Nilai K adalah 5

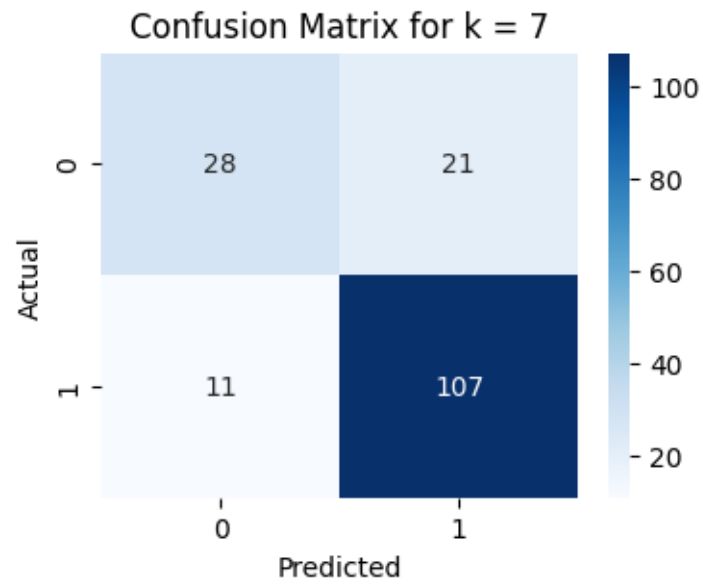
Accuracy = 0.8143712574850299

Precision = 0.8372093023255814

Recall = 0.9152542372881356

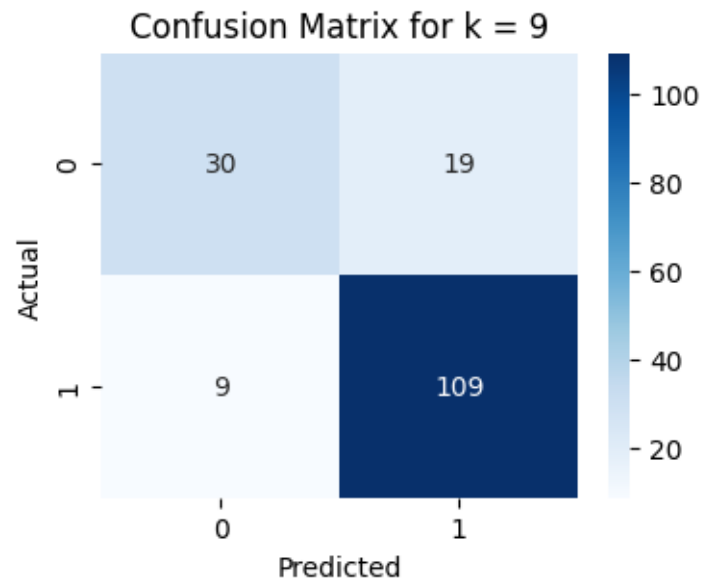
F1 Score = 0.8744939271255062

```
train_knn_classifier(7)
```



Hasil Evaluasi Nilai K adalah 7  
Accuracy = 0.8083832335329342  
Precision = 0.8359375  
Recall = 0.9067796610169492  
F1 Score = 0.8699186991869919

```
train_knn_classifier(9)
```



Hasil Evaluasi Nilai K adalah 9

Accuracy = 0.8323353293413174

Precision = 0.8515625

Recall = 0.923728813559322

F1 Score = 0.8861788617886179

## 0.11 LDA Topic

```
X = topic_proportions
y = data['label-topic']
y = y.replace({'komputasi': 1, 'rpl': 0})

def train_knn_lda_classifier(k):

    # train model knn
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    knn_classifier = KNeighborsClassifier(n_neighbors=k)
    knn_classifier.fit(X_train, y_train)
```

```
y_pred = knn_classifier.predict(X_test)

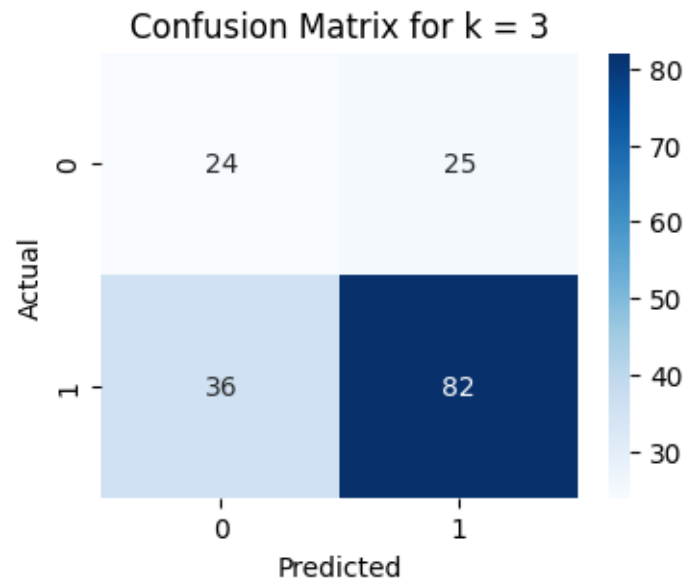
# confusion matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)

plt.figure(figsize=(4, 3))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title(f'Confusion Matrix for k = {k}')
plt.show()

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

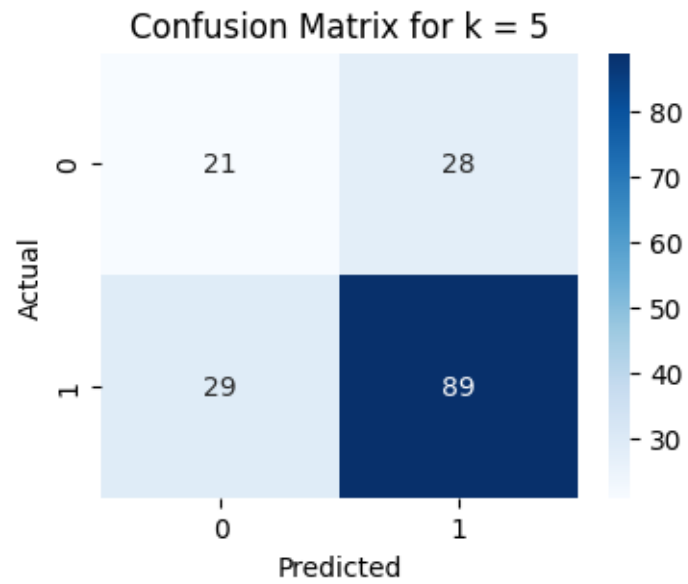
print(
    '\nHasil Evaluasi Nilai K adalah', k,
    '\nAccuracy =', accuracy,
    '\nPrecision =', precision,
    '\nRecall =', recall,
    '\nF1 Score =', f1
)

train_knn_lda_classifier(3)
```



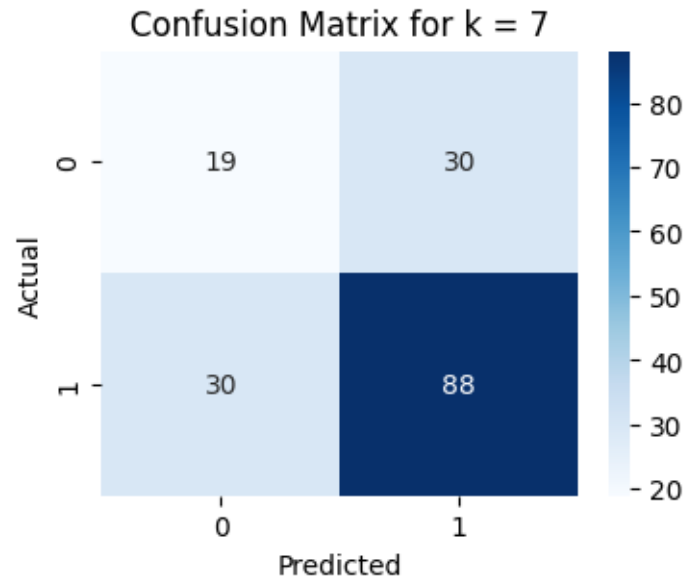
Hasil Evaluasi Nilai K adalah 3  
Accuracy = 0.6347305389221557  
Precision = 0.7663551401869159  
Recall = 0.6949152542372882  
F1 Score = 0.7288888888888889

```
train_knn_lda_classifier(5)
```



Hasil Evaluasi Nilai K adalah 5  
Accuracy = 0.6586826347305389  
Precision = 0.7606837606837606  
Recall = 0.7542372881355932  
F1 Score = 0.7574468085106383

```
train_knn_lda_classifier(7)
```

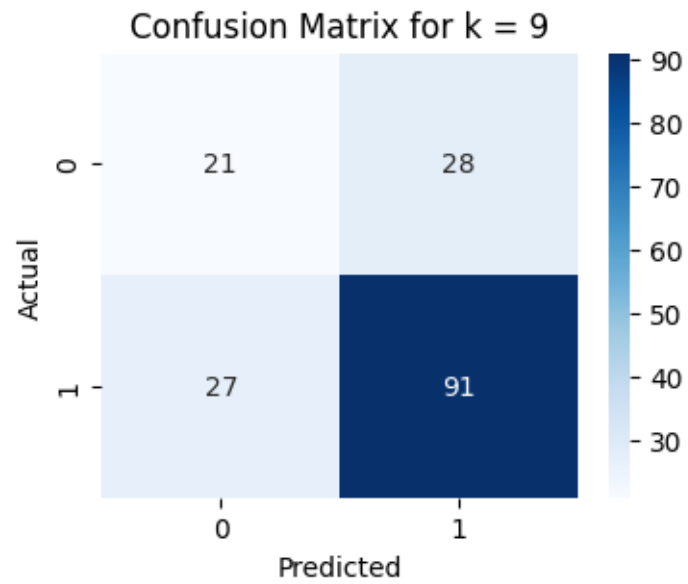


Hasil Evaluasi Nilai K adalah 7  
Accuracy = 0.6407185628742516  
Precision = 0.7457627118644068  
Recall = 0.7457627118644068  
F1 Score = 0.7457627118644068

```
train_knn_lda_classifier(9)
```

xl

*Classification*



Hasil Evaluasi Nilai K adalah 9  
Accuracy = 0.6706586826347305  
Precision = 0.7647058823529411  
Recall = 0.7711864406779662  
F1 Score = 0.7679324894514766



# 0

## *Crawl Berita CNN*

```
!pip install requests
!pip install beautifulsoup4
!pip install tqdm
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from re
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from re
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (4.11.2)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beauti
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (4.66.1)
```

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
from tqdm.auto import tqdm
```



# 0

## *Crawling Data*

Teknik crawling data adalah proses pengumpulan informasi dari berbagai sumber di internet secara otomatis. Crawling data dilakukan oleh program yang disebut “web crawler”. Web crawler pada proses ini menggunakan beautifulsoup untuk melakukan ekstraksi halaman konten tersebut agar dapat mengambil isi-isi konten pada berita di website CNN Indonesia. Terdapat juga library request yang digunakan untuk meminta respon pada situs yang akan dilakukan crawling data.

```
from os import replace
def cnnnews(page):

    data = {'judul': [], 'berita': []}
    for i in tqdm(range(1, page+1)):
        url = f"https://www.cnnindonesia.com/nasional/indeks/3/{i}"
        r = requests.get(url)
        request = r.content
        soup = BeautifulSoup(request, 'html.parser')
        soup = soup.find('div', {'class': 'flex flex-col gap-5'})
        news = soup.findAll('article', {'class': 'flex-grow'})
        # news = soup.findAll('a', {'aria-label': 'link description'})

        for new in tqdm(news):
            a_element = new.find('a', {'aria-label': 'link description'})['href']
            detail_request = requests.get(a_element)
            detail_soup = BeautifulSoup(detail_request.content, 'html.parser')

            judul = detail_soup.find('h1', {'class': 'leading-9'})
            berita = detail_soup.find('div', {'class': 'detail-text'})

            if judul and berita:
                judul = judul.text
                berita = berita.text
                noise = detail_soup.find('strong').text

            berita = berita.replace("ADVERTISEMENT", "").replace("SCROLL TO CONTINUE WITH CONTENT", "")
```

```
        berita = ' '.join(berita.split())
        data["judul"].append(judul)
        data["berita"].append(berita)

    df = pd.DataFrame(data)
    df.to_csv("berita-cnn.csv", index=False)

    return df

cnnnews(5)
```

0%| | 0/5 [00:00<?, ?it/s]  
0%| | 0/10 [00:00<?, ?it/s]  
0%| | 0/10 [00:00<?, ?it/s]  
0%| | 0/10 [00:00<?, ?it/s]  
0%| | 0/10 [00:00<?, ?it/s]  
0%| | 0/10 [00:00<?, ?it/s]

	judul	berita
0	Dirjen Kemendagri Safrizal ZA Jadi Pj Gubernu...	Menteri Dalam Negeri (Mendagri) Tito Karn...
1	Ganjar Temui Gus Mus di Rembang, Bahas Polemi...	Bakal capres PDIP Ganjar Pranowo mengunj...
2	Nomor Urut Capres-Cawapres Diundi KPU Besok	Komisi Pemilihan Umum (KPU) RI bakal me...
3	Utut Adianto Pimpin Panja Netralitas TNI di P...	Komisi I DPR sudah menyepakati pembentuk...
4	Istri Cak Nur Curhat ke Gus Mus: Nepotisme Di...	Sejumlah tokoh bangsa yang mengatassnamak...
5	Eks Stafsus SBY Velix Wanggai Jadi Pj Gubernu...	Presiden Joko Widodo menunjuk mantan Sta...
6	KPU Resmi Tetapkan Prabowo, Ganjar, Anies seb...	Komisi Pemilihan Umum (KPU) resmi menet...
7	Eks Kadis PUPR Papua Didakwa Terima Suap & Gr...	Kepala Dinas Pekerjaan Umum dan Perumah...
8	Fatia Maulidiyanti Dituntut 3,5 Tahun Penjara...	Aktivitis Hak Asasi Manusia (HAM) Fatia M...
9	KPK Panggil Keponakan SYL Istri Kapolrestabes...	Komisi Pemberantasan Korupsi (KPK) mema...
10	Haris Azhar Dituntut 4 Tahun Penjara Kasus Lo...	Aktivitis Hak Asasi Manusia (HAM) Haris A...
11	KPK Dalam Dugaan Kartu Anggota Kasino SYL	Komisi Pemberantasan Korupsi (KPK) tenga...
12	Pelecehan Anggota BEM UNY Hoaks, Polisi Tangk...	Polisi memastikan dugaan kasus pelecehan m...
13	Dinkes DKI: Kasus Cacar Monyet di Jakarta Cap...	Dinas Kesehatan (Dinkes) DKI Jakarta melap...
14	Pemkab Ungkap Susun 6 Dimensi pada Perencanaa...	Kabupaten Klaten menyatakan memiliki enar...
15	Sri Mulyani Sebut Petugas Kesehatan Merupakan...	Pemerintah Kabupaten Klaten menggelar Up...
16	Klaten Ambil Bagian pada Evaluasi Tahap II Sm...	Kabupaten Klaten turut ambil bagian sebaga...
17	Puncak Peringatan HKN Ke-59, Bupati Klaten Ap...	Bupati Klaten, Jawa Tengah, Sri Mulyani, m...
18	Nakes Klaten Diminta Jadi Pelopor Budaya Hidu...	Bupati Klaten, Sri Mulyani, meminta tenaga...
19	FOTO: Suhartoyo Resmi Jadi Ketua MK Gantikan ...	Hakim konstitusi Suhartoyo resmi dilantik sel...
20	Fit and Proper Test Panglima TNI Singgung Isu...	Calon Panglima TNI Jenderal Agus Subiyant...
21	Kapolda Metro Sebut Tersangka Kasus Pemerasan...	Kapolda Metro Jaya Irjen Karyoto menyebut...

	judul	berita
22	Irjen Karyoto soal Firli: Kita Lihat Saja Bes...	Kapolda Metro Jaya Irjen Karyoto angkat su
23	Suhartoyo Janji Segera Bentuk MKMK Permanen	Ketua Mahkamah Konstitusi (MK) Suhartoyo
24	Mahfud Singgung Menteri Jokowi Ditangkap Koru...	Menko Polhukam Mahfud MD menyinggung
25	Mahfud Respons Dugaan Saling Sandera KPK-Pold...	Menko Polhukam Mahfud MD angkat suara s
26	Agus Subiyanto di DPR: Jika Ingin Damai, Bers...	Kepala Staf Angkatan Darat (KSAD) sekalig
27	OTT KPK di Sorong Terkait Pengondisian Temuan...	Komisi Pemberantasan Korupsi (KPK) meng
28	KPK Total Tangkap 5 Orang Terkait OTT di Sorong	Komisi Pemberantasan Korupsi (KPK) total
29	Anwar Usman Absen di Pelantikan Ketua MK, Izi...	Hakim Konstitusi Anwar Usman tidak hadir
30	KPK Total Tangkap 5 Orang Terkait OTT di Sorong	Komisi Pemberantasan Korupsi (KPK) total
31	Anwar Usman Absen di Pelantikan Ketua MK, Izi...	Hakim Konstitusi Anwar Usman tidak hadir
32	Komisi I DPR Sepakati Agus Subiyanto Jadi Pan...	Komisi I DPR secara resmi menyepakati KSA
33	Jelang Penetapan Capres-Cawapres, Jalan Depan...	Jalan Imam Bonjol depan kantor Komisi Pem
34	Massa Atribut Serba Hitam Demo di KPU Jelang ...	Demonstrasi terjadi di depan kantor Komisi I
35	PAN Solid Dukung Prabowo-Gibran dan Menang Pi...	Wakil Bendahara Umum Partai Amanat Nasi
36	LHKPN Ketua MK Suhartoyo, Punya Harta Rp14,7 ...	Ketua Mahkamah Konstitusi (MK) terpilih p
37	Fit & Proper Test, Agus Janji Ingatkan Prajur...	Calon Panglima TNI Jenderal Agus Subiyant
38	Yasonna soal Wamenkumham Jadi Tersangka: Sila...	Menteri Hukum dan Hak Asasi Manusia (Me
39	Ketua MK Suhartoyo Menangis Saat Pidato Soal ...	Hakim Mahkamah Konstitusi (MK) Suhartoy
40	Bahlil Heran Gibran Dipersoalkan: Banyak Ment...	Ketua Dewan Pembina Relawan Pengusaha N
41	Gibran: Laporkan Saja ke Bawaslu Jika Ada Kec...	Wali Kota Solo sekaligus bakal cawapres Gibr
42	Bupati Dhito: Batik Kediri Siap Masuk Kancan ...	Bupati Kediri Hanindhito Himawan Pramana
43	Sempat Macet Parah, Jalan Mampang Prapatan Ar...	Kemacetan sempat terjadi di Jalan Mampang
44	Pejabat Sorong dan Pegawai BPK Terjaring OTT ...	Komisi Pemberantasan Korupsi (KPK) melak
45	Fit and Proper Test Calon Panglima TNI, Jende...	Kepala Staf Angkatan Darat (KSAD) sekalig
46	VIDEO: Momen Suhartoyo Resmi Dilantik Jadi Ke...	Mahkamah Konstitusi resmi melantik Suharto
47	Wali Kota Semarang Ingin Masyarakat Tak Berga...	Festival pangan pendamping beras bertajuk I
48	Panglima TNI-Kapolri ke Rumah Agus Subiyanto ...	Panglima TNI Laksamana Yudo Margono dan
49	Suhartoyo Resmi Jadi Ketua MK Gantikan Anwar ...	Hakim konstitusi Suhartoyo resmi dilantik sel



# 0

## *Ringkasan Berita*





# 0

## *Instalasi*

```
!pip install rouge
```

Requirement already satisfied: rouge in /usr/local/lib/python3.10/dist-packages (1.0.1)

Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from rouge) (1.16.0)

```
import pandas as pd
import nltk
import networkx as nx
import matplotlib.pyplot as plt
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from nltk.corpus import stopwords
from rouge import Rouge
```

```
nltk.download('punkt')
nltk.download("stopwords")
```

[nltk\_data] Downloading package punkt to /root/nltk\_data...

[nltk\_data] Package punkt is already up-to-date!

[nltk\_data] Downloading package stopwords to /root/nltk\_data...

[nltk\_data] Package stopwords is already up-to-date!

True



# 0

## Data

Data yang digunakan menggunakan hasil crawling data pada website CNN Indonesia dalam kategori berita nasional.

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

csv_path = '/content/drive/My Drive/Task/Pencarian dan Penambangan Web/Tugas 2/data/berita-cnn.csv'
df = pd.read_csv(csv_path)
df
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive")

	judul	berita
0	Dirjen Kemendagri Safrizal ZA Jadi Pj Gubernu...	Menteri Dalam Negeri (Mendagri) Tito Karn...
1	Ganjar Temui Gus Mus di Rembang, Bahas Polemi...	Bakal capres PDIP Ganjar Pranowo mengunj...
2	Nomor Urut Capres-Cawapres Diundi KPU Besok	Komisi Pemilihan Umum (KPU) RI bakal me...
3	Utut Adianto Pimpin Panja Netralitas TNI di P...	Komisi I DPR sudah menyepakati pembentuk...
4	Istri Cak Nur Curhat ke Gus Mus: Nepotisme Di...	Sejumlah tokoh bangsa yang mengatasnamak...
5	Eks Stafsus SBY Velix Wanggai Jadi Pj Gubernu...	Presiden Joko Widodo menunjuk mantan Sta...
6	KPU Resmi Tetapkan Prabowo, Ganjar, Anies seb...	Komisi Pemilihan Umum (KPU) resmi menet...
7	Eks Kadis PUPR Papua Didakwa Terima Suap & Gr...	Kepala Dinas Pekerjaan Umum dan Perumah...
8	Fatia Maulidiyanti Dituntut 3,5 Tahun Penjara...	Aktivitis Hak Asasi Manusia (HAM) Fatia M...
9	KPK Panggil Keponakan SYL Istri Kapolrestabes...	Komisi Pemberantasan Korupsi (KPK) mema...
10	Haris Azhar Dituntut 4 Tahun Penjara Kasus Lo...	Aktivitis Hak Asasi Manusia (HAM) Haris A...
11	KPK Dalam Dugaan Kartu Anggota Kasino SYL	Komisi Pemberantasan Korupsi (KPK) tenga...
12	Pelecehan Anggota BEM UNY Hoaks, Polisi Tangk...	Polisi memastikan dugaan kasus pelecehan m...
13	Dinkes DKI: Kasus Cacar Monyet di Jakarta Cap...	Dinas Kesehatan (Dinkes) DKI Jakarta melap...
14	Pemkab Ungkap Susun 6 Dimensi pada Perencanaa...	Kabupaten Klaten menyatakan memiliki enar...
15	Sri Mulyani Sebut Petugas Kesehatan Merupakan...	Pemerintah Kabupaten Klaten menggelar Up...
16	Klaten Ambil Bagian pada Evaluasi Tahap II Sm...	Kabupaten Klaten turut ambil bagian sebaga...
17	Puncak Peringatan HKN Ke-59, Bupati Klaten Ap...	Bupati Klaten, Jawa Tengah, Sri Mulyani, m...
18	Nakes Klaten Diminta Jadi Pelopor Budaya Hidu...	Bupati Klaten, Sri Mulyani, meminta tenaga...
19	FOTO: Suhartoyo Resmi Jadi Ketua MK Gantikan ...	Hakim konstitusi Suhartoyo resmi dilantik sel...
20	Fit and Proper Test Panglima TNI Singgung Isu...	Calon Panglima TNI Jenderal Agus Subiyant...
21	Kapolda Metro Sebut Tersangka Kasus Pemerasan...	Kapolda Metro Jaya Irjen Karyoto menyebut...

	judul	berita
22	Irjen Karyoto soal Firli: Kita Lihat Saja Bes...	Kapolda Metro Jaya Irjen Karyoto angkat su
23	Suhartoyo Janji Segera Bentuk MKMK Permanen	Ketua Mahkamah Konstitusi (MK) Suhartoyo
24	Mahfud Singgung Menteri Jokowi Ditangkap Koru...	Menko Polhukam Mahfud MD menyinggung
25	Mahfud Respons Dugaan Saling Sandera KPK-Pold...	Menko Polhukam Mahfud MD angkat suara s
26	Agus Subiyanto di DPR: Jika Ingin Damai, Bers...	Kepala Staf Angkatan Darat (KSAD) sekalig
27	OTT KPK di Sorong Terkait Pengondisian Temuan...	Komisi Pemberantasan Korupsi (KPK) meng
28	KPK Total Tangkap 5 Orang Terkait OTT di Sorong	Komisi Pemberantasan Korupsi (KPK) total
29	Anwar Usman Absen di Pelantikan Ketua MK, Izi...	Hakim Konstitusi Anwar Usman tidak hadir
30	KPK Total Tangkap 5 Orang Terkait OTT di Sorong	Komisi Pemberantasan Korupsi (KPK) total
31	Anwar Usman Absen di Pelantikan Ketua MK, Izi...	Hakim Konstitusi Anwar Usman tidak hadir
32	Komisi I DPR Sepakati Agus Subiyanto Jadi Pan...	Komisi I DPR secara resmi menyepakati KSA
33	Jelang Penetapan Capres-Cawapres, Jalan Depan...	Jalan Imam Bonjol depan kantor Komisi Pem
34	Massa Atribut Serba Hitam Demo di KPU Jelang ...	Demonstrasi terjadi di depan kantor Komisi I
35	PAN Solid Dukung Prabowo-Gibran dan Menang Pi...	Wakil Bendahara Umum Partai Amanat Nasi
36	LHKPN Ketua MK Suhartoyo, Punya Harta Rp14,7 ...	Ketua Mahkamah Konstitusi (MK) terpilih p
37	Fit & Proper Test, Agus Janji Ingatkan Prajur...	Calon Panglima TNI Jenderal Agus Subiyant
38	Yasonna soal Wamenkumham Jadi Tersangka: Sila...	Menteri Hukum dan Hak Asasi Manusia (Me
39	Ketua MK Suhartoyo Menangis Saat Pidato Soal ...	Hakim Mahkamah Konstitusi (MK) Suhartoy
40	Bahlil Heran Gibran Dipersoalkan: Banyak Ment...	Ketua Dewan Pembina Relawan Pengusaha N
41	Gibran: Laporkan Saja ke Bawaslu Jika Ada Kec...	Wali Kota Solo sekaligus bakal cawapres Gibr
42	Bupati Dhito: Batik Kediri Siap Masuk Kancan ...	Bupati Kediri Hanindhito Himawan Pramana
43	Sempat Macet Parah, Jalan Mampang Prapatan Ar...	Kemacetan sempat terjadi di Jalan Mampang
44	Pejabat Sorong dan Pegawai BPK Terjaring OTT ...	Komisi Pemberantasan Korupsi (KPK) melak
45	Fit and Proper Test Calon Panglima TNI, Jende...	Kepala Staf Angkatan Darat (KSAD) sekalig
46	VIDEO: Momen Suhartoyo Resmi Dilantik Jadi Ke...	Mahkamah Konstitusi resmi melantik Suharto
47	Wali Kota Semarang Ingin Masyarakat Tak Berga...	Festival pangan pendamping beras bertajuk I
48	Panglima TNI-Kapolri ke Rumah Agus Subiyanto ...	Panglima TNI Laksamana Yudo Margono dan
49	Suhartoyo Resmi Jadi Ketua MK Gantikan Anwar ...	Hakim konstitusi Suhartoyo resmi dilantik sel

Untuk meringkas suatu dokumen, maka kita hanya memerlukan satu sampel berita yang akan digunakan dengan menggunakan berita pertama

```
berita = df['berita'].iloc[0]
```

# 0

## *Preprocessing*

Pada analisis ringkasan dokumen kali ini akan menggunakan 2 metode pengujian, pengujian pertama akan dilakukan langkah ringkasan berita tanpa menggunakan preprocessing dan tahapan pengujian kedua berita yang diringkas akan menggunakan tahapan preprocessing. Tahapan preprocessing ini antara lain yaitu menghapus angka, simbol dan stopwords pada berita.

```
def preprocessing(text):
    text = re.sub(r'\d+', '', text)
    text = re.sub(r'^\w\s.', '', text)
    text = text.lower()

    stop_words = set(stopwords.words('indonesian'))
    words = text.split()
    filtered_words = [word for word in words if word.lower() not in stop_words]

    preprocessing_text = ' '.join(filtered_words)

    return preprocessing_text

kalimat_preprocessing = preprocessing(berita)
```



# 0

## Ekstraksi Fitur

Ekstraksi fitur pada tahapan ini menggunakan TF-IDF untuk membentuk vektor pada setiap kalimatnya, sedangkan fitur yang akan digunakan pada vektor TF-IDF ini meliputi *term* pada keseluruhan dokumen.

```
kalimat = nltk.sent_tokenize(berita) #memecah dokumen berdasarkan kalimatnya tanpa preprocessing

kalimat_preprocessing = nltk.sent_tokenize(kalimat_preprocessing) #memecah dokumen berdasarkan
```

### 0.12 TF-IDF

*Term Frequency-Inverse Document Frequency* (TF-IDF) adalah vektor yang digunakan untuk mengevaluasi pentingnya kata-kata dalam sebuah dokumen. Nilai frekuensi kemunculan kata dalam setiap dokumen ini menunjukkan seberapa penting kata tersebut dalam dokumen. Berikut merupakan rumus untuk menghitung TF-IDF:

$$W_{d,t} = tf_{t,d} \cdot idf_{t,d}$$

Keterangan =  $W_{d,t}$  = Nilai *Term Frequency* untuk *term* (t) dalam dokumen (d).  $tf_{t,d}$  = Frekuensi kemunculan *term* (t) dalam dokumen (d).  $idf_{t,d}$  = Inverse Document Frequency Nilai kebalikan frekuensi dokumen *term* (t) dalam dokumen (d). Pada dasarnya TF-IDF adalah gabungan dari *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) sehingga sebelum kita membentuk nilai TFIDF, maka kita harus menghitung kedua nilai tersebut. *Term Frequency* (TF) merupakan perhitungan yang digunakan untuk menentukan seberapa sering kata-kata muncul dalam sebuah dokumen.

$$tf = \frac{tf}{\max(tf)}$$

Keterangan =  $tf$  = banyaknya kata yang dicari dalam dokumen  $\max(tf)$  = jumlah kemunculan term terbanyak pada dokumen yang sama

Sedangkan *Inverse Document Frequency* (IDF) menilai kata-kata yang sering

muncul sebagai kurang signifikan karena kemunculannya dalam banyak dokumen. Semakin rendah nilai IDF, maka kata tersebut akan dianggap kurang berarti dan sebaliknya, semakin tinggi nilai IDF maka kata tersebut akan dianggap lebih relevan atau penting dalam dokumen tersebut.

$$idf_t = \frac{1}{\max(df_t)}$$

Keterangan =  $D$  = total dokumen  $df(t)$  = jumlah dokumen yang mengandung *term* ( $t$ )

#### TF-IDF Tanpa Preprocessing

```
tfidf_vectorizer = TfidfVectorizer()
tfidf = tfidf_vectorizer.fit_transform(kalimat)

terms = tfidf_vectorizer.get_feature_names_out()
tfidf = pd.DataFrame(data=tfidf.toarray(), columns=terms)

tfidf
```

	11	13	19	2024	adalah	administrasi	adwil	akan	awasi	b
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
1	0.269719	0.269719	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
3	0.000000	0.000000	0.000000	0.000000	0.162701	0.000000	0.000000	0.207738	0.000000	0
4	0.255375	0.255375	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
9	0.000000	0.000000	0.000000	0.312276	0.000000	0.000000	0.000000	0.000000	0.000000	0
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
11	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
12	0.000000	0.000000	0.000000	0.000000	0.194693	0.248586	0.248586	0.000000	0.248586	0
13	0.000000	0.000000	0.279549	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
14	0.000000	0.000000	0.000000	0.000000	0.279706	0.000000	0.000000	0.000000	0.000000	0
15	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0

#### TF-IDF Menggunakan Preprocessing

```
tfidf_vectorizer = TfidfVectorizer()
tfidf_preprocessing = tfidf_vectorizer.fit_transform(kalimat_preprocessing)
```



```

terms = tfidf_vectorizer.get_feature_names_out()
tfidf_preprocessing = pd.DataFrame(data=tfidf_preprocessing.toarray(), columns=terms)

tfidf_preprocessing

```

	administrasi	adwil	awasi	bangka	bapakbapak	belitung	berharap	berjalan	berperan
0	0.000000	0.000000	0.000000	0.294052	0.0000	0.294052	0.000000	0.000000	0.000000
1	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.308065	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.380838	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.260917	0.0000	0.260917	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000	0.000000	0.2477	0.000000	0.216287	0.000000	0.000000
12	0.261989	0.261989	0.261989	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.358922
14	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
15	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.000000	0.000000	0.000000



# 0

## Membentuk Graph

Dalam meringkas dokumen ini diperlukan untuk membentuk graph sebagai gambaran antara kedekatan pada masing-masing kalimat, sehingga dalam ringkasan dokumen yang dihasilkan akan memunculkan kalimat-kalimat penting dan memiliki kedekatan di setiap dokumennya.

### 0.13 Cosine Similarity

*Cosine similarity* digunakan untuk mengukur seberapa mirip dua vektor dalam ruang berdimensi banyak. Hasil dari *cosine similarity* ini akan menentukan apakah vektor tersebut menuju ke arah yang sama. Semakin kecil sudut antara dua vektor, maka semakin mirip satu sama lain sedangkan begitu juga sebaliknya, semakin besar nilai *cosine similarity* maka vektor tersebut dianggap jauh kemiripannya. Dalam ringkasan dokumen ini penting untuk menghitung nilai *cosine similarity* untuk mengetahui hubungan kesamaan antara kalimat satu dengan kalimat lainnya. Vektor yang digunakan untuk menghitung nilai *cosine similarity* ini adalah hasil dari TF-IDF pada langkah sebelumnya. Berikut merupakan rumus yang digunakan untuk menghitung nilai *cosine similarity*.

$$\text{similarity}(A, B) = \frac{A \cdot B}{|A||B|}$$

Keterangan =  $A \cdot B$  = Vector dot product dari A dan B dihitung dengan  $\sum_{i=1}^n x_i y_i$   $|A|$  = Panjang vektor A dihitung dengan  $\sum_{i=1}^n x_i^2$   $|B|$  = Panjang vektor B dihitung dengan  $\sum_{i=1}^n y_i^2$

Cosine Similarity Tanpa Preprocessing

```
cosine = cosine_similarity(tfidf, tfidf)
```

```
similarity = pd.DataFrame(cosine, columns=range(len(kalimat)), index=range(len(kalimat)))  
similarity
```

lx

Membentuk Graph

	0	1	2	3	4	5	6	7	8	9
0	1.000000	0.000000	0.183156	0.031750	0.077184	0.106877	0.000000	0.234579	0.000000	0.034
1	0.000000	1.000000	0.000000	0.000000	0.206638	0.000000	0.041888	0.035372	0.048745	0.000
2	0.183156	0.000000	1.000000	0.000000	0.034132	0.112296	0.060918	0.276659	0.200600	0.096
3	0.031750	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.058014	0.000
4	0.077184	0.206638	0.034132	0.000000	1.000000	0.032889	0.000000	0.000000	0.000000	0.033
5	0.106877	0.000000	0.112296	0.000000	0.032889	1.000000	0.000000	0.187934	0.000000	0.083
6	0.000000	0.041888	0.060918	0.000000	0.000000	0.000000	1.000000	0.036043	0.049670	0.060
7	0.234579	0.035372	0.276659	0.000000	0.000000	0.187934	0.036043	1.000000	0.210924	0.000
8	0.000000	0.048745	0.200600	0.058014	0.000000	0.000000	0.049670	0.210924	1.000000	0.000
9	0.034685	0.000000	0.096883	0.000000	0.033780	0.083440	0.060291	0.000000	0.000000	1.000
10	0.116650	0.000000	0.032666	0.029594	0.041664	0.083680	0.000000	0.027517	0.000000	0.000
11	0.067043	0.000000	0.041731	0.075541	0.059209	0.101223	0.171626	0.017577	0.099860	0.091
12	0.077952	0.058545	0.033793	0.031677	0.000000	0.032563	0.000000	0.068995	0.000000	0.000
13	0.042725	0.000000	0.000000	0.044278	0.041611	0.000000	0.000000	0.000000	0.000000	0.000
14	0.000000	0.047528	0.057351	0.045509	0.000000	0.000000	0.048430	0.089207	0.222667	0.000
15	0.000000	0.000000	0.000000	0.000000	0.061096	0.042397	0.000000	0.055524	0.000000	0.043
16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000

Cosine Similarity Menggunakan Preprocessing

```
cosine_preprocessing = cosine_similarity(tfidf_preprocessing, tfidf_preprocessing)
```

```
similarity_preprocessing = pd.DataFrame(cosine_preprocessing, columns=range(len(kalimat)), index=range(len(kalimat)))
similarity_preprocessing
```

	0	1	2	3	4	5	6	7	8	9
0	1.000000	0.000000	0.152651	0.060469	0.047436	0.131893	0.000000	0.295745	0.000000	0.043
1	0.000000	1.000000	0.000000	0.000000	0.105852	0.000000	0.000000	0.000000	0.000000	0.000
2	0.152651	0.000000	1.000000	0.000000	0.054547	0.151663	0.000000	0.381721	0.316919	0.050
3	0.060469	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.126832	0.000
4	0.047436	0.105852	0.054547	0.000000	1.000000	0.047130	0.000000	0.000000	0.000000	0.049
5	0.131893	0.000000	0.151663	0.000000	0.047130	1.000000	0.000000	0.156279	0.000000	0.043
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000
7	0.295745	0.000000	0.381721	0.000000	0.000000	0.156279	0.000000	1.000000	0.244551	0.000
8	0.000000	0.000000	0.316919	0.126832	0.000000	0.000000	0.000000	0.244551	1.000000	0.000
9	0.043678	0.000000	0.050225	0.000000	0.049395	0.043396	0.000000	0.000000	0.000000	1.000
10	0.112841	0.000000	0.054610	0.068464	0.000000	0.047184	0.000000	0.042140	0.000000	0.000
11	0.104161	0.000000	0.070955	0.111109	0.034891	0.061307	0.097365	0.027376	0.000000	0.098
12	0.092131	0.072819	0.043710	0.000000	0.000000	0.037766	0.000000	0.081749	0.000000	0.000
13	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
14	0.000000	0.000000	0.087843	0.000000	0.000000	0.000000	0.000000	0.067784	0.106957	0.000

	0	1	2	3	4	5	6	7	8	9
15	0.000000	0.000000	0.000000	0.000000	0.083649	0.000000	0.000000	0.000000	0.000000	0.000000
16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

## 0.14 Graph

Hasil dari nilai *cosine similarity* ini akan dapat digunakan untuk membentuk graph dengan menggunakan modul `nx.graph`. Graph ini akan menggambarkan ilustrasi dari kedekatan setiap kalimatnya dalam berita tersebut. Dalam proses penggambaran graph tersebut diperlukan ambang batas (*threshold*) yang digunakan untuk memberikan batasan agar keseluruhan kalimatnya tidak dihubungkan menggunakan garis (*edge*). Nilai ambang batas (*threshold*) yang digunakan adalah 0.1

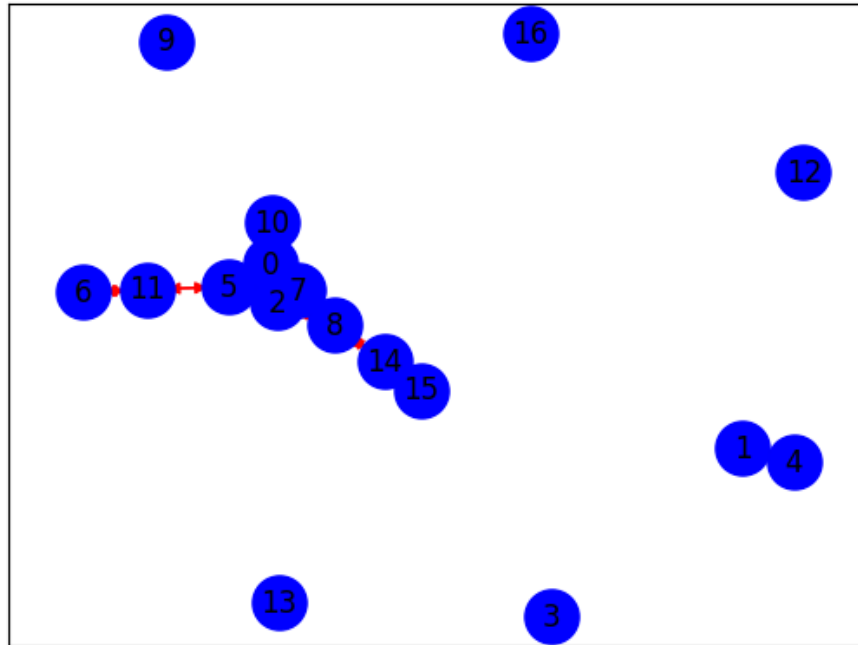
Graph Tanpa Preprocessing

```
G = nx.DiGraph()
for i in range(len(cosine)):
    G.add_node(i)

for i in range(len(cosine)):
    for j in range(len(cosine)):
        similarity = cosine[i][j]
        if similarity > 0.1 and i != j:
            G.add_edge(i, j)

pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos, node_size=500, node_color='b')
nx.draw_networkx_edges(G, pos, edge_color='red', arrows=True)
nx.draw_networkx_labels(G, pos)

plt.show()
```



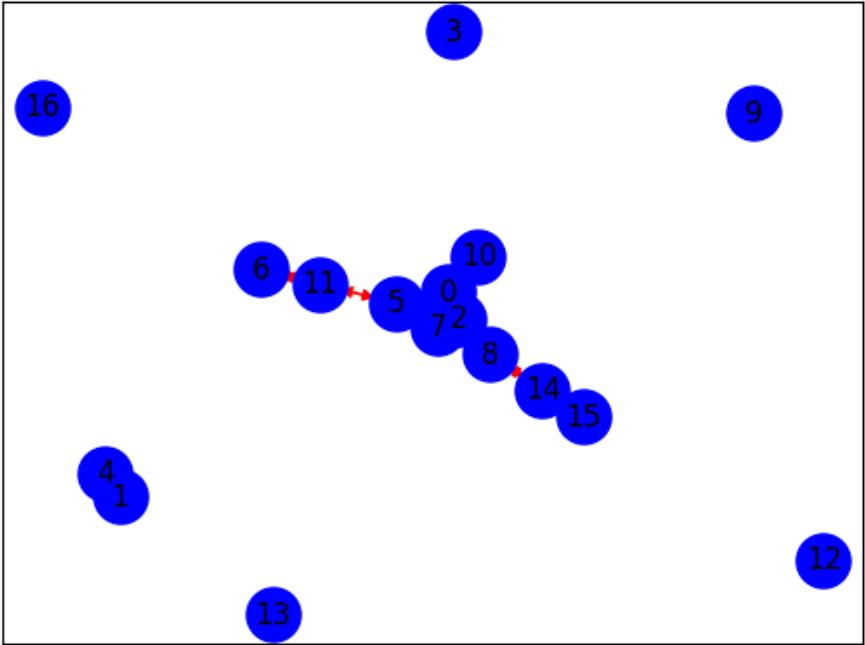
### Graph Menggunakan Preprocessing

```
G_preprocessing = nx.DiGraph()
for i in range(len(cosine_preprocessing)):
    G_preprocessing.add_node(i)

for i in range(len(cosine_preprocessing)):
    for j in range(len(cosine_preprocessing)):
        similarity_preprocessing = cosine_preprocessing[i][j]
        if similarity_preprocessing > 0.1 and i != j:
            G_preprocessing.add_edge(i, j)

pos = nx.spring_layout(G)
nx.draw_networkx_nodes(G, pos, node_size=500, node_color='b')
nx.draw_networkx_edges(G, pos, edge_color='red', arrows=True)
nx.draw_networkx_labels(G, pos)

plt.show()
```







# 0

## *Matriks Sentralitas*

Matriks Sentralitas adalah matriks yang digunakan untuk merepresentasikan ukuran sentralitas dari setiap *node* dalam jaringan. Sentralitas adalah konsep dalam analisis jaringan yang mencoba mengukur sejauh mana suatu *node* berada di pusat jaringan atau sejauh mana suatu *node* penting dalam graph. Beberapa matriks ini akan digunakan untuk membangun ringkasan dokumen yang dibuat oleh sistem, matriks ini didapatkan dari bentuk graph yang telah terbentuk pada langkah sebelumnya.

### 0.15 Closeness Centrality

*Closeness similarity* adalah ukuran sejauh mana nilai kedekatan antara pasangan node dalam suatu jaringan serupa. Dengan ini kita dapat mengukur kesamaan struktural antara node-node dalam graf berdasarkan nilai kedekatan mereka. Closeness similarity dirumuskan sebagai berikut.

$$CC(i) = \frac{N - 1}{\sum_j d(i, j)}$$

Keterangan = N = nomor dari masing-masing node  $d(i, j)$  = d adalah panjang jalur terpendek antara node i dan j dalam jaringan

Closeness Centrality Tanpa Preprocessing

```
closeness= nx.closeness centrality(G)

sorted_closeness = sorted(closeness.items(), key=lambda x: x[1], reverse=True)
print("Closeness Centrality:")
for node, closeness in sorted_closeness:
    print(f"Node {node}: {closeness:.4f}")
```

Closeness Centrality:

Node 2: 0.3164

Node 7: 0.3164

Node 5: 0.2978  
 Node 0: 0.2812  
 Node 8: 0.2664  
 Node 11: 0.2201  
 Node 14: 0.2025  
 Node 10: 0.1947  
 Node 6: 0.1633  
 Node 15: 0.1534  
 Node 1: 0.0625  
 Node 4: 0.0625  
 Node 3: 0.0000  
 Node 9: 0.0000  
 Node 12: 0.0000  
 Node 13: 0.0000  
 Node 16: 0.0000

```
ringkasan_closeness = ""
print("Tiga Node Tertinggi Closeness Centrality:")
for node, closeness in sorted_closeness[:3]:
    top_sentence = kalimat[node]
    ringkasan_closeness += top_sentence + " "
    print(f"Node {node}: Closeness Centrality = {closeness:.4f}")
    print(f"Kalimat: {top_sentence}\n")
```

Tiga Node Tertinggi Closeness Centrality:

Node 2: Closeness Centrality = 0.3164

Kalimat: Pada saat bersamaan, Tito juga melantik Velix Vernando Wanggai sebagai Pj Gubernur Papua Pegunungan

Node 7: Closeness Centrality = 0.3164

Kalimat: Eks Stafsus SBY Velix Wanggai Jadi Pj Gubernur Papua Pegunungan Dia meminta Safrizal menangani

Node 5: Closeness Centrality = 0.2978

Kalimat: Tito meminta dua pj gubernur menangani kemiskinan ekstrem, inflasi, stunting, infrastruktur, dan

Closeness Centrality Menggunakan Preprocessing

```
closeness_preprocessing = nx.closeness centrality(G_preprocessing)

sorted_closeness_preprocessing = sorted(closeness_preprocessing.items(), key=lambda x: x[1], reverse=True)
print("Closeness Centrality:")
for node, closeness in sorted_closeness_preprocessing:
    print(f"Node {node}: {closeness:.4f}")
```

Closeness Centrality:

Node 2: 0.3375

```

Node 7: 0.3375
Node 8: 0.3375
Node 0: 0.3164
Node 3: 0.2664
Node 5: 0.2664
Node 11: 0.2664
Node 14: 0.2411
Node 10: 0.2109
Node 15: 0.1746
Node 1: 0.0625
Node 4: 0.0625
Node 6: 0.0000
Node 9: 0.0000
Node 12: 0.0000
Node 13: 0.0000
Node 16: 0.0000

```

```

ringkasan_closeness_preprocessing = ""
print("Tiga Node Tertinggi Closeness Centrality Menggunakan Preprocessing:")
for node, closeness_preprocessing in sorted_closeness_preprocessing[:3]:
    top_sentence = kalimat[node]
    ringkasan_closeness_preprocessing += top_sentence + " "
    print(f"Node {node}: Closeness Centrality = {closeness_preprocessing:.4f}")
    print(f"Kalimat: {top_sentence}\n")

```

Tiga Node Tertinggi Closeness Centrality Menggunakan Preprocessing:

Node 2: Closeness Centrality = 0.3375

Kalimat: Pada saat bersamaan, Tito juga melantik Velix Vernando Wanggai sebagai Pj Gubernur Papua Pegunungan

Node 7: Closeness Centrality = 0.3375

Kalimat: Eks Stafsus SBY Velix Wanggai Jadi Pj Gubernur Papua Pegunungan Dia meminta Safrizal menangani

Node 8: Closeness Centrality = 0.3375

Kalimat: Sementara untuk Velix, tugas keamanan di Papua Pegunungan menjadi prioritas.

---

## 0.16 Page Rank

Dalam konteks ini, dokumen dianggap sebagai “halaman” yang terhubung oleh hubungan yang merefleksikan keterkaitan atau relevansinya. Dengan menerapkan konsep PageRank, dokumen yang dianggap lebih “penting” atau relevan dapat diberikan skor lebih tinggi. Penggunaan faktor damping, serupa dengan dalam algoritma PageRank, dapat membantu mengontrol sejauh mana

pengaruh satu dokumen terhadap yang lain. Dengan memberikan skor pada dokumen berdasarkan hubungan mereka dalam jaringan informasi, kita dapat menghasilkan ringkasan yang mencerminkan tingkat relevansi dan pentingnya masing-masing dokumen dalam konteks keseluruhan.

$$S(V_i) = (1 - d) + d * \sum \frac{1}{Out(V_j)} S(V_j)$$

Keterangan =  $d$  = faktor redaman, jika tidak ada sambungan keluar  $in(V_i)$  = tautan masuk dari  $i$ , yang merupakan satu set  $out(V_j)$  = tautan keluar dari  $j$ , yang merupakan satu set  $|out(V_j)|$  = jumlah tautan keluar

Page Rank Tanpa Preprocessing

```
pagerank = nx.pagerank(G)

sorted_pagerank= sorted(pagerank.items(), key=lambda x: x[1], reverse=True)
print("Page Rank :")
for node, pagerank in sorted_pagerank:
    print(f"Node {node}: {pagerank:.4f}")
```

```
Page Rank :
Node 5: 0.1103
Node 0: 0.1102
Node 2: 0.1061
Node 7: 0.1061
Node 8: 0.0879
Node 1: 0.0784
Node 4: 0.0784
Node 14: 0.0731
Node 11: 0.0708
Node 15: 0.0428
Node 6: 0.0418
Node 10: 0.0352
Node 3: 0.0118
Node 9: 0.0118
Node 12: 0.0118
Node 13: 0.0118
Node 16: 0.0118
```

```
ringkasan_pagerank = ""
print("Tiga Node Tertinggi Page Rank :")
for node, pagerank in sorted_pagerank[:3]:
    top_sentence = kalimat[node]
    ringkasan_pagerank += top_sentence + " "
```

```
print(f"Node {node}: Page Rank = {pagerank:.4f}")
print(f"Kalimat: {top_sentence}\n")
```

Tiga Node Tertinggi Page Rank :

Node 5: Page Rank = 0.1103

Kalimat: Tito meminta dua pj gubernur menangani kemiskinan ekstrem, inflasi, stunting, infrastruktur, dan

Node 0: Page Rank = 0.1102

Kalimat: Menteri Dalam Negeri (Mendagri) Tito Karnavian menunjuk Safrizal ZA sebagai Penjabat (Pj) Gubernur

Node 2: Page Rank = 0.1061

Kalimat: Pada saat bersamaan, Tito juga melantik Velix Vernando Wanggai sebagai Pj Gubernur Papua Pegunungan

Page Rank Menggunakan Preprocessing

```
pagerank_preprocessing = nx.pagerank(G_preprocessing)

sorted_pagerank_preprocessing= sorted(pagerank_preprocessing.items(), key=lambda x: x[1], reverse=True)
print("Page Rank :")
for node, pagerank_preprocessing in sorted_pagerank_preprocessing:
    print(f"Node {node}: {pagerank_preprocessing:.4f}")
```

Page Rank :

Node 0: 0.1303

Node 8: 0.1102

Node 2: 0.1003

Node 7: 0.1003

Node 1: 0.0784

Node 4: 0.0784

Node 5: 0.0766

Node 14: 0.0707

Node 3: 0.0605

Node 11: 0.0596

Node 15: 0.0418

Node 10: 0.0339

Node 6: 0.0118

Node 9: 0.0118

Node 12: 0.0118

Node 13: 0.0118

Node 16: 0.0118

```
ringkasan_pagerank_preprocessing = ""
print("Tiga Node Tertinggi Page Rank Menggunakan Preprocessing:")
for node, pagerank_preprocessing in sorted_pagerank_preprocessing[:3]:
```

```

top_sentence = kalimat[node]
ringkasan_pagerank_preprocessing += top_sentence + " "
print(f"Node {node}: Page Rank = {pagerank_preprocessing:.4f}")
print(f"Kalimat: {top_sentence}\n")

```

Tiga Node Tertinggi Page Rank Menggunakan Preprocessing:

Node 0: Page Rank = 0.1303

Kalimat: Menteri Dalam Negeri (Mendagri) Tito Karnavian menunjuk Safrizal ZA sebagai Penjabat (Pj) Gubernur

Node 8: Page Rank = 0.1102

Kalimat: Sementara untuk Velix, tugas keamanan di Papua Pegunungan menjadi prioritas.

Node 2: Page Rank = 0.1003

Kalimat: Pada saat bersamaan, Tito juga melantik Velix Vernando Wanggai sebagai Pj Gubernur Papua Pegunungan

## 0.17 Eigen Vector

Dalam ringkasan dokumen, eigenvector tidak digunakan secara langsung untuk menghitung ringkasan. Namun, Anda dapat memanfaatkan konsep pengukuran penting dari eigenvector untuk memberikan bobot atau skor pada elemen-elemen dalam dokumen yang mungkin memiliki kepentingan lebih besar dalam rangka membuat ringkasan. Perhitungan eigenvector yang didapatkan dari matriks korelasi untuk menentukan bobot relatif setiap dokumen atau kata. Berikut merupakan rumus untuk menghitung nilai eigenvector.

$$\det(\lambda I - A)$$

Keterangan =  $\det()$  = Merupakan fungsi determinan, yang menghasilkan nilai determinan dari suatu matriks.  $\lambda$  = Merupakan simbol yang mewakili eigenvalue yang sedang dicari.  $I$  = Merupakan matriks identitas yang sesuai dengan dimensi matriks.  $A$  = Merupakan matriks yang eigenvalues-nya ingin dicari.

Eigen Vector Tanpa Preprocessing

```

eigenvector = nx.eigenvector_centrality(G)

sorted_eigenvector= sorted(eigenvector.items(), key=lambda x: x[1], reverse=True)
print("Eigen Vector :")
for node, eigenvector in sorted_eigenvector:
    print(f"Node {node}: {eigenvector:.4f}")

```

Eigen Vector :

Node 2: 0.4823  
 Node 7: 0.4823  
 Node 5: 0.4453  
 Node 0: 0.4426  
 Node 8: 0.3053  
 Node 11: 0.1398  
 Node 10: 0.1274  
 Node 14: 0.0958  
 Node 6: 0.0402  
 Node 15: 0.0276  
 Node 1: 0.0000  
 Node 4: 0.0000  
 Node 3: 0.0000  
 Node 9: 0.0000  
 Node 12: 0.0000  
 Node 13: 0.0000  
 Node 16: 0.0000

```
ringkasan_eigenvector = ""
print("Tiga Node Tertinggi Eigen Vector:")
for node, eigenvector in sorted_eigenvector[:3]:
    top_sentence = kalimat[node]
    ringkasan_eigenvector += top_sentence + " "
    print(f"Node {node}: Page Rank = {eigenvector:.4f}")
    print(f"Kalimat: {top_sentence}\n")
```

Tiga Node Tertinggi Eigen Vector:

Node 2: Page Rank = 0.4823

Kalimat: Pada saat bersamaan, Tito juga melantik Velix Vernando Wanggai sebagai Pj Gubernur Papua Pegunungan

Node 7: Page Rank = 0.4823

Kalimat: Eks Stafsus SBY Velix Wanggai Jadi Pj Gubernur Papua Pegunungan Dia meminta Safrizal menangani

Node 5: Page Rank = 0.4453

Kalimat: Tito meminta dua pj gubernur menangani kemiskinan ekstrem, inflasi, stunting, infrastruktur, dan

```
eigenvector_preprocessing = nx.eigenvector_centrality(G_preprocessing)

sorted_eigenvector_preprocessing= sorted(eigenvector_preprocessing.items(), key=lambda x: x[1],
print("Eigen Vector :")
for node, eigenvector_preprocessing in sorted_eigenvector_preprocessing:
    print(f"Node {node}: {eigenvector_preprocessing:.4f}")
```

Eigen Vector :

Node 2: 0.4692  
Node 7: 0.4692  
Node 0: 0.4619  
Node 5: 0.3956  
Node 8: 0.3344  
Node 11: 0.1708  
Node 3: 0.1427  
Node 10: 0.1305  
Node 14: 0.1027  
Node 15: 0.0290  
Node 1: 0.0000  
Node 4: 0.0000  
Node 6: 0.0000  
Node 9: 0.0000  
Node 12: 0.0000  
Node 13: 0.0000  
Node 16: 0.0000

```
ringkasan_eigenvector_preprocessing = ""  
print("Tiga Node Tertinggi Eigen Vector Menggunakan Preprocessing:")  
for node, eigenvector_preprocessing in sorted_eigenvector_preprocessing[:3]:  
    top_sentence = kalimat[node]  
    ringkasan_eigenvector_preprocessing += top_sentence + " "  
    print(f"Node {node}: Page Rank = {eigenvector_preprocessing:.4f}")  
    print(f"Kalimat: {top_sentence}\n")
```

Tiga Node Tertinggi Eigen Vector Menggunakan Preprocessing:

Node 2: Page Rank = 0.4692

Kalimat: Pada saat bersamaan, Tito juga melantik Velix Vernando Wanggai sebagai Pj Gubernur Papua Pegunungan

Node 7: Page Rank = 0.4692

Kalimat: Eks Stafsus SBY Velix Wanggai Jadi Pj Gubernur Papua Pegunungan Dia meminta Safrizal menanganikan

Node 0: Page Rank = 0.4619

Kalimat: Menteri Dalam Negeri (Mendagri) Tito Karnavian menunjuk Safrizal ZA sebagai Penjabat (Pj) Gubernur



# 0

## *Evaluasi*

ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) adalah sekelompok metrik evaluasi otomatis yang umum digunakan untuk mengukur kualitas ringkasan teks. ROUGE memiliki beberapa varian, dan di antaranya, ROUGE-1, ROUGE-2, dan ROUGE-L merupakan metrik yang sering digunakan untuk mengukur sejauh mana kedekatan ringkasan sistem dengan ringkasan referensi.

```
#referensi ringkasan yang dibuat secara manual
referensi = ['Menteri Dalam Negeri Tito Karnavian melantik Safrizal ZA sebagai Penjabat Gubernur

print(referensi)
```

```
['Menteri Dalam Negeri Tito Karnavian melantik Safrizal ZA sebagai Penjabat Gubernur Bangka Belitung da
```

```
def rouge(referensi, hasil_ringkasan):
    rouge = Rouge()
    scores = rouge.get_scores(hasil_ringkasan, referensi)
    print (scores)
```

### 0.18 Closeness Centrality

```
rouge(referensi[0], ringkasan_closeness)
```

```
[{'rouge-1': {'r': 0.3333333333333333, 'p': 0.5641025641025641, 'f': 0.41904761437823135}, 'rouge-2':
```

```
rouge(referensi[0], ringkasan_closeness_preprocessing)
```

```
[{'rouge-1': {'r': 0.22727272727272727, 'p': 0.42857142857142855, 'f': 0.2970296984413293}, 'rouge-2':
```

Berikut adalah hasil simpulan berdasarkan evaluasi ROUGE pada *Closeness Centrality*:

Berdasarkan evaluasi ROUGE *closeness centrality*, ditemukan bahwa tanpa menggunakan preprocessing, sistem ringkasan cenderung mencapai hasil yang lebih baik. Secara khusus, pada metrik ROUGE-1, sistem tanpa preprocessing memiliki recall sekitar 0.33, precision sekitar 0.56, dan F1-Score sekitar 0.42. Sebaliknya, ketika menggunakan preprocessing, nilai recall turun menjadi sekitar 0.23, precision sekitar 0.43, dan F1-Score sekitar 0.30.

---

## 0.19 Page Rank

```
rouge(referensi[0], ringkasan_pagerank)
```

```
[{'rouge-1': {'r': 0.4090909090909091, 'p': 0.675, 'f': 0.5094339575649698}, 'rouge-2': {'r': 0.24324324324324326, 'p': 0.5, 'f': 0.36363636363636365}]
```

```
rouge(referensi[0], ringkasan_pagerank_preprocessing)
```

```
[{'rouge-1': {'r': 0.30303030303030304, 'p': 0.5714285714285714, 'f': 0.39603959943142836}, 'rouge-2': {'r': 0.18181818181818182, 'p': 0.4, 'f': 0.2857142857142857}}]
```

Berdasarkan hasil evaluasi ROUGE pada *Page Rank* untuk ringkasan dengan dan tanpa preprocessing menggunakan metode *Page Rank*, dapat disimpulkan bahwa tanpa preprocessing sistem mampu mencapai hasil yang lebih baik dalam sebagian besar metrik evaluasi. Pada metrik ROUGE-1, sistem tanpa preprocessing memiliki nilai recall sekitar 0.41, precision sekitar 0.675, dan F1-Score sekitar 0.51. Sementara itu, dengan preprocessing, nilai recall turun menjadi sekitar 0.30, precision sekitar 0.571, dan F1-Score sekitar 0.40

---

## 0.20 Eigen Vector

```
rouge(referensi[0], ringkasan_eigenvector)
```

```
[{'rouge-1': {'r': 0.3333333333333333, 'p': 0.5641025641025641, 'f': 0.41904761437823135}, 'rouge-2': {'r': 0.18181818181818182, 'p': 0.4, 'f': 0.2857142857142857}}]
```

```
rouge(referensi[0], ringkasan_eigenvector_preprocessing)
```

```
[{'rouge-1': {'r': 0.30303030303030304, 'p': 0.5405405405405406, 'f': 0.3883495099594685}, 'rouge-2': {'r': 0.18181818181818182, 'p': 0.4, 'f': 0.2857142857142857}}]
```

Berdasarkan hasil evaluasi ROUGE untuk ringkasan dengan dan tanpa preprocessing menggunakan metode eigenvector centrality, dapat ditarik beberapa kesimpulan. Tanpa preprocessing, sistem cenderung mencapai hasil yang sedikit lebih baik pada sebagian besar metrik evaluasi. Pada metrik ROUGE-1, sistem tanpa preprocessing memiliki nilai recall sekitar 0.33, precision sekitar 0.56, dan F1-Score sekitar 0.42. Sementara itu, dengan preprocessing, nilai recall dan F1-Score tetap stabil, sedangkan precision mengalami penurunan menjadi sekitar 0.54.

