**UNIVERSITAS GUNADARMA**



**PRAKTIKUM KECERDASAN ARTIFICIAL**

**MANUAL BOOK**

**“Analisis Kemiripan Teks dengan Embedding BERT Pretrained”**

**NAMA**

**NPM**

**KELAS**

**PRAKTIKUM**

**PJ**

**Muhammad Tarmidzi Bariq**

**51422161**

**3IA11**

**Kecerdasan Artifisial**

**Gilbert Jefferson Faozato Mendrofa**

**Ditulis Guna Melengkapi Sebagai Syarat Praktikum Kecerdasan Artificial**

**jenjang S1**

**LABORATORIUM INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2024**

# KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas segala karunia nikmatnya sehingga Manual Book yang berjudul Analisis Kemiripan Teks dengan Embedding BERT Pretrained ini dapat diselesaikan dengan maksimal, tanpa ada halangan yang berarti. Makalah ini disusun untuk memenuhi Praktikum Kecerdasan Artificial yang dibimbing oleh PJ yaitu Gilbert Jefferson Faozato Mendrofa

…

Depok, (Tgl Penulisan)

(Nama Praktikan)

# DAFTAR ISI

[KATA PENGANTAR 2](#_Toc150895270)

[DAFTAR ISI 3](#_Toc150895271)

[BAB I 4](#_Toc150895272)

[PENDAHULUAN 4](#_Toc150895273)

[1.1 Latar Belakang 4](#_Toc150895274)

[1.2 Tujuan 4](#_Toc150895275)

[BAB II 6](#_Toc150895276)

[PEMBAHASAN 6](#_Toc150895277)

[2.1 Artificial Intelligence 6](#_Toc150895278)  
[2.2 (Materi Terkait Penulisan) 6](#_Toc150895278)

[BAB III 11](#_Toc150895281)

[LOGIKA PROGRAM 11](#_Toc150895282)

[BAB IV 18](#_Toc150895283)

[PENUTUP 18](#_Toc150895284)

[4.1 Kesimpulan 18](#_Toc150895285)

[4.2 Saran 18](#_Toc150895286)

# **BAB I**

# PENDAHULUAN

## Latar Belakang

(Latar Belakang Penulisan)

## Tujuan

(Tujuan Pembuatan Project)

# BAB II

# PEMBAHASAN

* 1. Artificial Intelligence

Pada tahun 1950, Alan Turing, seorang pionir kecerdasan buatan (AI) dan ahli matematika asal Inggris, menjalankan sebuah percobaan yang dikenal sebagai Turing Test. Percobaan ini melibatkan penggunaan komputer yang terhubung melalui terminal di tempat yang berjauhan. Satu ujung terminal dilengkapi dengan perangkat lunak kecerdasan buatan, sementara ujung lainnya memiliki seorang operator. Menariknya, operator tersebut tidak mengetahui bahwa di ujung terminal lainnya terdapat perangkat lunak kecerdasan buatan. Selama percobaan ini, keduanya berkomunikasi, di mana terminal di ujung memberikan respons terhadap serangkaian pertanyaan yang diajukan oleh operator. Sang operator, tanpa menyadari keberadaan AI, meyakini bahwa ia sedang berinteraksi dengan operator manusia yang berada di terminal yang berbeda.

2.2 (Materi Penulisan)

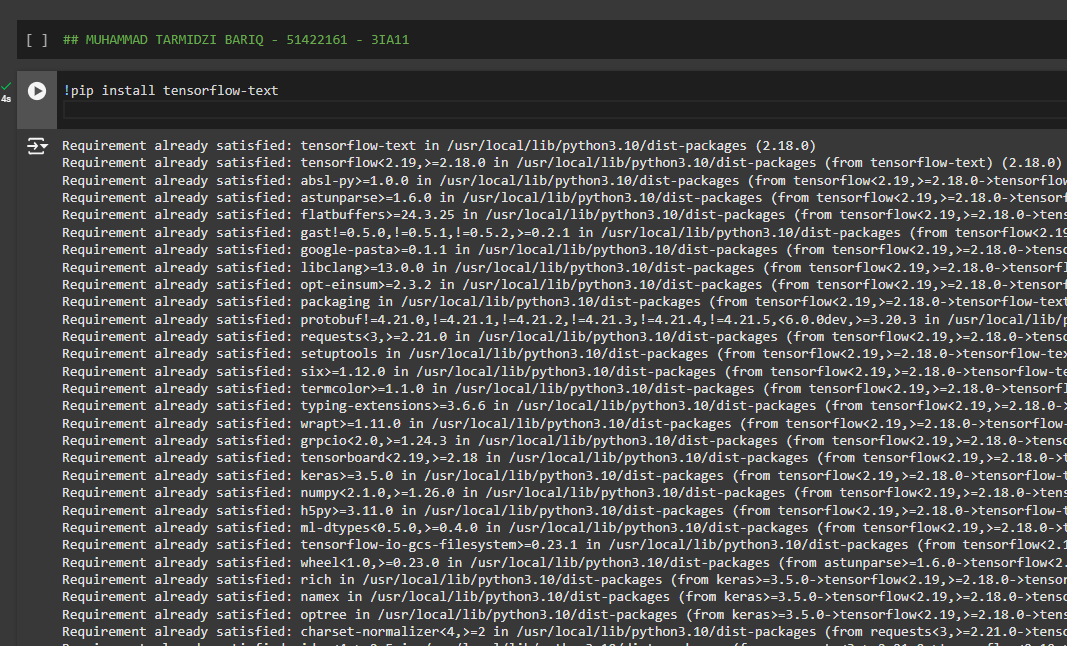
# BAB III

# LOGIKA PROGRAM

**(Screenshot Codingan dan Jelaskan sintaks program baris perbaris, Sertakan output program yang sudah dijalankan dalam screenshot)**

1. !pip install tensorflow-text

Ekstensi dari TensorFlow yang dirancang untuk membantu dalam pemrosesan teks. Paket ini digunakan untuk mengolah teks dan berinteraksi dengan model NLP berbasis TensorFlow, termasuk untuk tokenisasi dan preprocessing teks yang kompatibel dengan model seperti BERT.



1. Import modul

* import numpy as np

Digunakan untuk komputasi numerik dan manipulasi array.

* import pandas as pd

Pustaka untuk manipulasi dan analisis data dalam bentuk tabel (data frame).

* import matplotlib.pyplot as plt

Pustaka utama untuk visualisasi data dalam Python.

* import seaborn as sns

Pustaka visualisasi data berbasis matplotlib yang lebih sederhana dan estetis.

* import tensorflow as tf

Framework untuk machine learning dan deep learning.

* import tensorflow\_hub as hub

Modul untuk menggunakan model pra-latih dari TensorFlow Hub.

* import tensorflow\_text as text

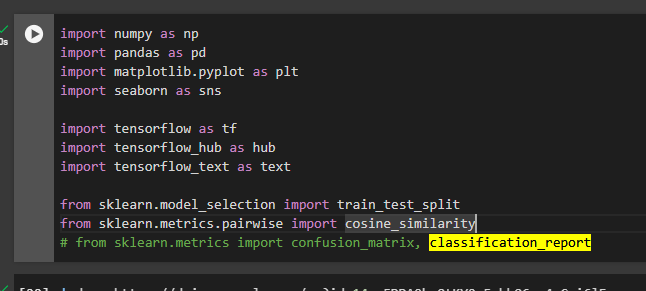
Ekstensi untuk tensorflow yang membantu dalam preprocessing teks, seperti tokenisasi, yang sering dibutuhkan saat bekerja dengan model bahasa alami seperti BERT.

* from sklearn.model\_selection import train\_test\_split

Fungsi dari scikit-learn untuk membagi dataset menjadi data latih (training) dan data uji (testing).

* from sklearn.metrics.pairwise import cosine\_similarity

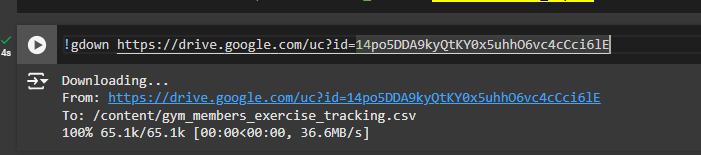
Digunakan untuk mengukur kesamaan antara dua vektor, seperti embedding teks, berdasarkan sudut di antara vektor tersebut (similarity measure).



1. upload file

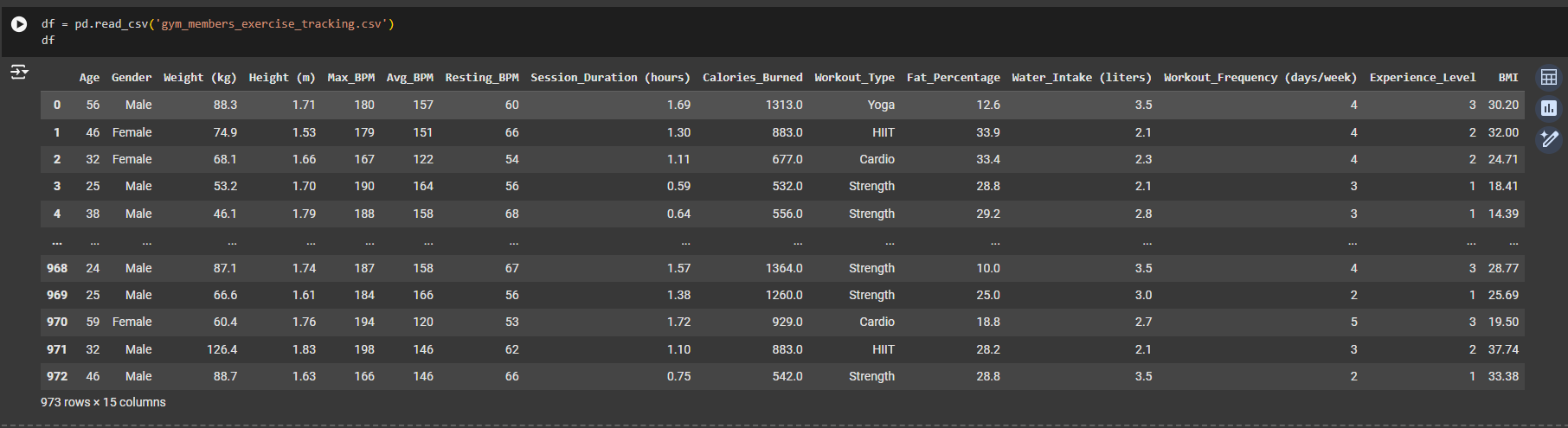
!gdown

Mengunduh file dari Google Drive langsung ke dalam lingkungan notebook,



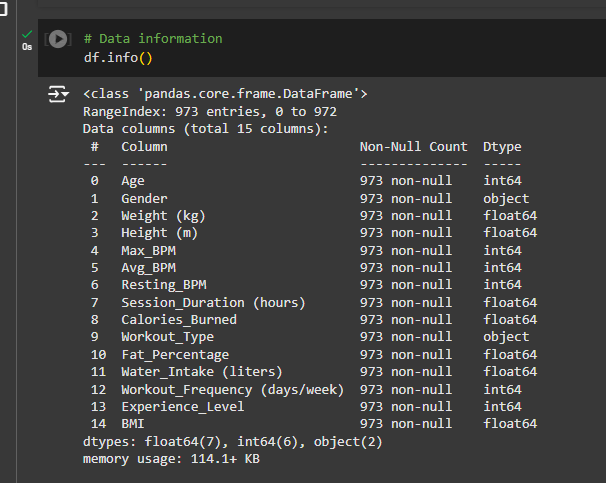
1. membaca isi file

Menggunakan pustaka pandas untuk membaca file CSV bernama gym\_members\_exercise\_tracking.csv.



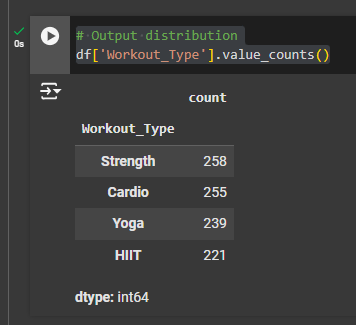
1. df.info()

Berfungsi untuk menampilkan ringkasan informasi tentang DataFrame df,



1. df['Workout\_Type'].value\_counts()

Berfungsi untuk menampilkan distribusi nilai dalam kolom Workout\_Type pada DataFrame df

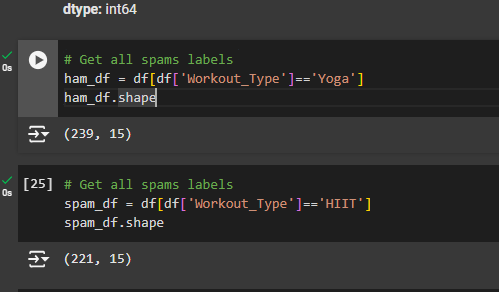


1. ham\_df = df[df['Workout\_Type']=='Yoga']

ham\_df.shape

Kode ini bertujuan untuk memfilter DataFrame df agar hanya menyertakan baris yang memiliki nilai 'Yoga' pada kolom 'Workout\_Type'.

.shape digunakan untuk mendapatkan ukuran dari DataFrame yang difilter tersebut dalam bentuk tuple, di mana nilai pertama adalah jumlah baris dan nilai kedua adalah jumlah kolom.

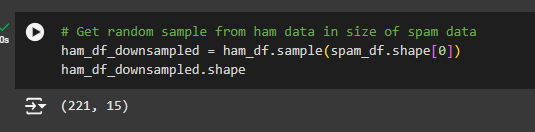


1. ham\_df\_downsampled = ham\_df.sample(spam\_df.shape[0])

Baris ini mengambil sampel acak dari *ham\_df* sebanyak jumlah baris yang ada pada *spam\_df* (nilai spam\_df.shape[0] mewakili jumlah baris). Ini dilakukan untuk menyeimbangkan data, sehingga ham\_df\_downsampled memiliki ukuran yang sama dengan spam\_df.

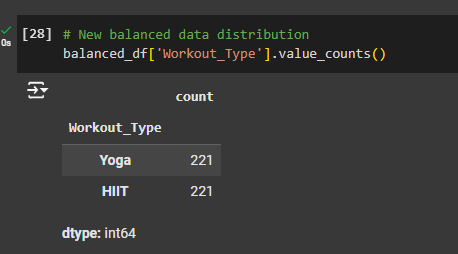
ham\_df\_downsampled.shape

Bagian ini menampilkan ukuran dari ham\_df\_downsampled. Hasil (221, 15) berarti bahwa data ham\_df\_downsampled berisi 221 baris dan 15 kolom.



1. balanced\_df['Workout\_Type'].value\_counts()

Perintah ini menghitung frekuensi dari setiap nilai unik dalam kolom Workout\_Type pada DataFrame balanced\_df



1. balanced\_df['spam'] = balanced\_df['Workout\_Type'].apply(lambda x: 1 if x == 'Yoga' else 0)

Kode ini menambahkan kolom baru bernama spam ke DataFrame balanced\_df.

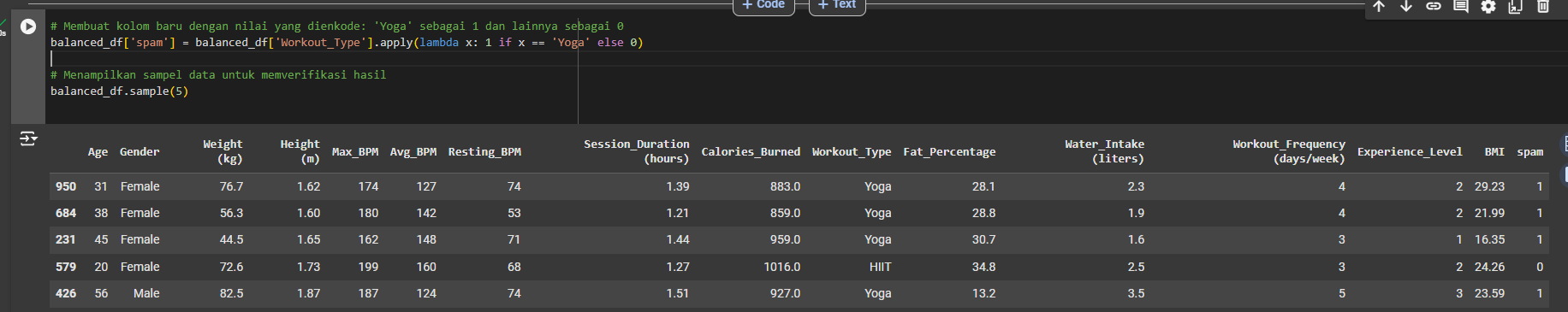
Kolom spam akan berisi nilai 1 jika Workout\_Type adalah Yoga dan 0 untuk nilai lain (dalam hal ini, HIIT).

Ini sering digunakan dalam tugas klasifikasi biner untuk memberikan label numerik (0 atau 1) bagi setiap kategori.

balanced\_df.sample(5)

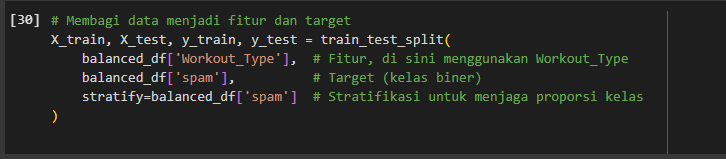
Menampilkan 5 baris sampel acak dari balanced\_df untuk memverifikasi bahwa kolom spam telah ditambahkan dengan benar

Tabel hasil menunjukkan bahwa kolom spam mencerminkan nilai 1 untuk Yoga dan 0 untuk HIIT.



1. Memisahkan Data Menjadi Fitur dan Target

* X\_train dan X\_test: Ini adalah set data fitur untuk pelatihan dan pengujian. Dalam hal ini, fitur yang digunakan adalah kolom Workout\_Type.
* y\_train dan y\_test: Ini adalah label target untuk pelatihan dan pengujian, diambil dari kolom spam yang telah dikonversi menjadi kelas biner.
* stratify=balanced\_df['spam']: Parameter ini memastikan bahwa pembagian data tetap mempertahankan proporsi kelas yang sama dalam set pelatihan dan pengujian, sehingga distribusi kelas tetap seimbang di kedua set.

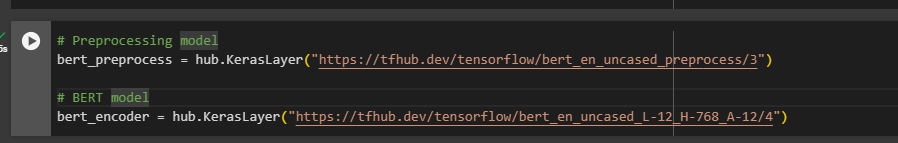


1. bert\_preprocess

Menangani tugas preprocessing seperti tokenisasi teks, normalisasi huruf kecil, dan penambahan token khusus yang dibutuhkan oleh model BERT.

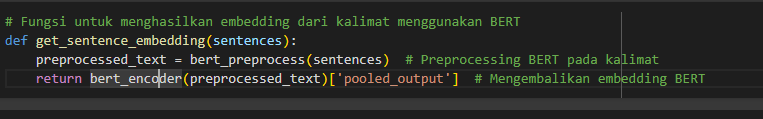
Bert\_encoder

Mengubah teks menjadi representasi vektor (embedding) yang bisa digunakan untuk analisis atau tugas-tugas NLP



1. Fungsi get\_sentence\_embedding

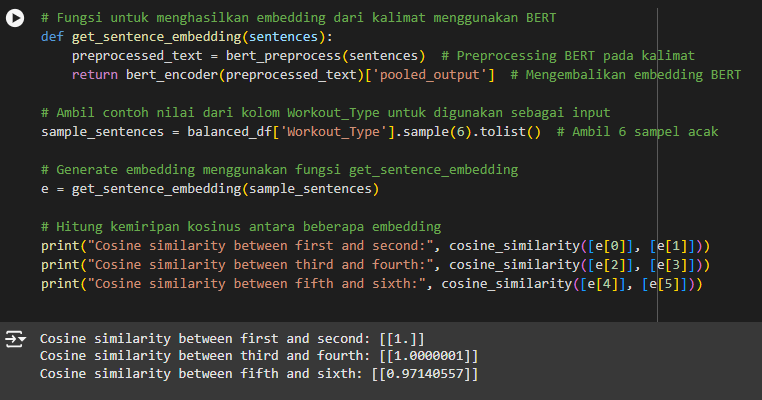
Mengambil teks (sentences), memprosesnya melalui bert\_preprocess, dan kemudian menghasilkan embedding menggunakan bert\_encoder. Hasil akhirnya adalah representasi vektor (embedding) dari kalimat tersebut



1. sample\_sentences mengambil 6 contoh acak dari kolom Workout\_Type di balanced\_df, yang menggantikan kalimat manual seperti "banana" dan "elon musk".

get\_sentence\_embedding(sample\_sentences) menghasilkan embedding untuk kalimat-kalimat yang diambil dari file CSV.

cosine\_similarity menghitung kesamaan kosinus antara embedding untuk melihat seberapa mirip kategori Workout\_Type satu dengan lainnya.



# BAB IV

# PENUTUP

* 1. Kesimpulan

(Jelaskan kesimpulan yang bisa diambil dari project)

* 1. Saran

(Saran apa yang masih bisa dikembangkan atau diperbaiki dalam pengerjaan project)