

## ↳ Responsi NLP

Nama : Dzaky Athallah S.

NIM : 2211110040

```
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
!pip install nltk
```

```
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) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
```

```
import nltk
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
```

```
df = pd.read_csv('/content/dataset_mentalhealth.csv')
```

```
# Membersihkan teks
def clean_text(text):
    text = re.sub(r'^a-zA-Z0-9\s', '', text).lower()
    return text

# Tokenisasi dan menghapus stopwords
def remove_stopwords(text):
    stop_words = set(stopwords.words('indonesian'))
    words = text.split()
    words = [word for word in words if word.isalnum() and word not in stop_words]
    return ' '.join(words)

# Membersihkan dan tokenisasi pertanyaan
df['Questions_Bersih'] = df['Questions'].apply(clean_text)
df['Questions_Bersih'] = df['Questions_Bersih'].apply(remove_stopwords)

# Membersihkan dan tokenisasi jawaban
df['Answers_Bersih'] = df['Jawaban'].apply(clean_text)
df['Answers_Bersih'] = df['Answers_Bersih'].apply(remove_stopwords)
```

```
df
```

	Question_ID	Questions	Jawaban	Questions_Bersih	Answers_Bersih
0	1590140	Apa yang dimaksud dengan penyakit mental?	Penyakit mental adalah kondisi kesehatan yang ...	penyakit mental	penyakit mental kondisi kesehatan mengganggu p...
1	2110618	Siapa yang terpengaruh oleh penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruhi...	terpengaruh penyakit mental	penyakit mental mempengaruhi 1 5 orang dewasa ...
2	6361820	Apa penyebab penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruhi...	penyebab penyakit mental	penyakit mental mempengaruhi 1 5 orang dewasa ...
3	9434130	Apa sajakah tanda-tanda peringatan penyakit me...	Gejala gangguan kesehatan mental bervariasi te...	sajakah tandatanda peringatan penyakit mental	gejala gangguan kesehatan mental bervariasi te...

```
# Menggabungkan teks dari kolom pertanyaan dan jawaban
df['combined_text'] = df['Questions_Bersih'] + ' ' + df['Answers_Bersih']
```

df

	Question_ID	Questions	Jawaban	Questions_Bersih	Answers_Bersih	coml
0	1590140	Apa yang dimaksud dengan penyakit mental?	Penyakit mental adalah kondisi kesehatan yang ...	penyakit mental	penyakit mental kondisi kesehatan mengganggu p...	pen
1	2110618	Siapa yang terpengaruh oleh penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruhi...	terpengaruh penyakit mental	penyakit mental mempengaruhi 1 5 orang dewasa ...	1 pen
2	6361820	Apa penyebab penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruhi...	penyebab penyakit mental	penyakit mental mempengaruhi 1 5 orang dewasa ...	pen
3	9434130	Apa sajakah tanda-tanda peringatan penyakit me...	Gejala gangguan kesehatan mental bervariasi te...	sajakah tandatanda peringatan penyakit mental	gejala gangguan kesehatan mental bervariasi te...	pen
4	7657263	Apakah penderita penyakit jiwa bisa sembuh?	Ketika penyembuhan dari penyakit mental, ident...	penderita penyakit jiwa sembuh	penyembuhan penyakit mental identifikasi pengos...	p pe
...	...	...	...	...	...	...
		Bagaimana	Menyortir jika		menyortir minum	

Double-click (or enter) to edit

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(df['combined_text'])
```

tfidf\_matrix

```
<98x2107 sparse matrix of type '<class 'numpy.float64'>'
  with 7296 stored elements in Compressed Sparse Row format>
```

Double-click (or enter) to edit

```
# Cosine Similarity
from sklearn.metrics.pairwise import cosine_similarity

def get_answer(question, tfidf_matrix):

    question_vector = tfidf_vectorizer.transform([question])
    similarities = cosine_similarity(question_vector, tfidf_matrix).flatten()
    answer_index = similarities.argmax()
    return df['Jawaban'][answer_index]

sample_question = "Penyakit mental yang delusi"
answer = get_answer(sample_question, tfidf_matrix)
print("Jawaban:", answer)
```

Jawaban: 'Kesehatan mental' dan 'penyakit mental' semakin banyak digunakan seolah-olah memaksudkan hal yang sama, tetapi tidak. Set  
Ketika kita berbicara tentang kesehatan mental, kita berbicara tentang kesejahteraan mental kita: emosi kita, pikiran dan perasaan  
Penyakit mental adalah penyakit yang memengaruhi cara orang berpikir, merasakan, berperilaku, atau berinteraksi dengan orang lain.  
Kesehatan tidak seperti sakelar hidup/mati. Ada berbagai tingkat kesehatan. Orang-orang beralih pada kontinum mulai dari kesehatan  
Sama seperti seseorang yang merasa tidak sehat mungkin tidak memiliki penyakit serius, orang mungkin memiliki kesehatan mental yang  
Sama seperti itu mungkin untuk memiliki kesehatan mental yang buruk tetapi tidak ada penyakit mental, sangat mungkin untuk memiliki  
Dengan dukungan dan alat yang tepat, siapa pun dapat hidup dengan baik - betapun mereka mendefinisikan dengan baik - dan menemukan

```
import pandas as pd
# Test Pertanyaan
test_questions = [
    "Apakah terapi kognitif efektif untuk mengatasi stres?",
    "Obat apa yang dapat mengurangi stress?",
    "Apakah makanan tertentu dapat memengaruhi kesehatan mental?",
    "Bagaimana cara terhindar dari penyakit mental?",
    "Apakah olahraga dapat membantu meningkatkan kesehatan mental?",
    "Apa itu ADHD dan bagaimana cara diagnosisnya?",
    "Bagaimana cara memanaskan motor?",
    "Siapa kaprodi dari prodi sains data?",
    "Apa yang dimaksud dengan pajak?",
    "Bagaimana cara menabung yang baik?"]

# Fungsi untuk melakukan uji dan analisis hasilnya
def perform_test(questions, tfidf_matrix):
    results = []

    for idx, test_question in enumerate(questions):
        predicted_answer = get_answer(test_question, tfidf_matrix)

        result_dict = {
            "Pertanyaan": test_question,
            "Jawaban Prediksi": predicted_answer
        }

        results.append(result_dict)

        print(f"Pertanyaan {idx + 1}: {test_question}")
        print(f"Jawaban Prediksi: {predicted_answer}\n")

    return pd.DataFrame(results)

# Simpan hasil uji ke Excel
#results_df = perform_test(test_questions, tfidf_matrix)
#results_df.to_excel("TF-IDF Cosine Similarity.xlsx.xlsx", index=False)
```

## ▼ Evaluasi

```
def get_answer(question, tfidf_matrix, threshold=0.15): # Menghasilkan vektor TF-IDF untuk pertanyaan
    question_vector = tfidf_vectorizer.transform([question]) # Menghitung cosine similarity antara vektor pertanyaan dan vektor jawaban
    similarities = cosine_similarity(question_vector, tfidf_matrix).flatten() # Mendapatkan indeks jawaban dengan similarity tertinggi
    answer_index = similarities.argmax() # Jika similarity di bawah threshold, anggap pertanyaan kurang relevan
    if similarities[answer_index] < threshold:
        return "Pertanyaan kurang relevan..."
    # Mengembalikan jawaban dari indeks yang ditemukan
    return df['Jawaban'][answer_index]
```

```

test_questions = [
    "Apakah terapi kognitif efektif untuk mengatasi stres?",
    "Apa itu ADHD dan bagaimana cara diagnosisnya?",
    "Apakah makanan tertentu dapat memengaruhi kesehatan mental?",
    "Bagaimana cara mengenali tanda-tanda psikosis?",
    "Apakah olahraga dapat membantu meningkatkan kesehatan mental?",
    "Berapa banyak planet di tata surya kita?",
    "Bagaimana cara merawat tanaman hias dalam ruangan?",
    "Siapa presiden pertama Amerika Serikat?",
    "Apa yang dimaksud dengan hukum permintaan dan penawaran?",
    "Bagaimana cara mengatasi rambut bercabang?"]

import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity

def perform_test(questions, tfidf_matrix):
    results = []

    for idx, test_question in enumerate(questions):
        predicted_answer = get_answer(test_question, tfidf_matrix)
        question_vector = tfidf_vectorizer.transform([test_question])
        similarities = cosine_similarity(question_vector, tfidf_matrix).flatten()
        similarity_index = similarities.max()
        answer_index = similarities.argmax()

        result_dict = {
            "Pertanyaan": test_question,
            "Jawaban Prediksi": predicted_answer,
            "Similarity Index": similarity_index
        }

        results.append(result_dict)

    print(f"Pertanyaan {idx + 1}: {test_question}")
    print(f"Jawaban Prediksi: {predicted_answer}")
    print(f"Similarity Index: {similarity_index}")
    print()

    return pd.DataFrame(results)

# Perform the test and export the results to Excel
#results_df = perform_test(test_questions, tfidf_matrix)
#results_df.to_excel("TF-IDF Cosine Similarity dengan Score.xlsx", index=False)

```

## ▼ IndoBert

```

df = pd.read_csv('/content/dataset_mentalhealth.csv')

from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("indolem/indobert-base-uncased")
sentence = "Ini contoh kalimat untuk di-tokenize."

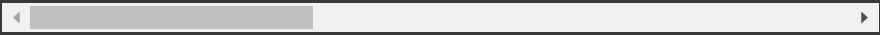
tokens = tokenizer(sentence, return_tensors="pt")

print("Original Sentence:", sentence)
print("Tokenized Output:", tokens)

```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens)
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models.
warnings.warn(

tokenizer_config.json: 100% 42.0/42.0 [00:00<00:00, 1.52kB/s]
config.json: 100% 1.01k/1.01k [00:00<00:00, 37.8kB/s]
vocab.txt: 100% 234k/234k [00:00<00:00, 4.26MB/s]
added_tokens.json: 100% 2.00/2.00 [00:00<00:00, 107B/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 6.36kB/s]
```



```

import re
import string
import torch
import numpy as np
import pandas as pd
from transformers import AutoTokenizer, AutoModel
from sklearn.metrics.pairwise import cosine_similarity

class QASystem:
    def __init__(self, df):
        self.df = df
        self.tokenizer = AutoTokenizer.from_pretrained("indolem/indobert-base-uncased")
        self.model = AutoModel.from_pretrained("indolem/indobert-base-uncased")

    def clean_text(self, text):
        text = text.translate(str.maketrans('', '', string.punctuation))
        text = re.sub(r'\s+', ' ', text).strip()
        return text.lower()

    def vectorize_text(self, text):
        inputs = self.tokenizer(
            text,
            return_tensors="pt",
            max_length=512,
            truncation=True
        )
        with torch.no_grad():
            outputs = self.model(**inputs)
        return outputs['pooler_output'].numpy()

    def preprocess_dataframe(self):
        self.df['cleaned_Jawaban'] = self.df['Jawaban'].apply(self.clean_text)
        self.df['Vectorized_Jawaban'] = self.df['cleaned_Jawaban'].apply(self.vectorize_text)

    def vectorize_user_question(self, user_question):
        user_question = self.clean_text(user_question)
        user_question_embedding = self.vectorize_text(user_question)
        return user_question_embedding

    def answer_question(self, user_question):
        user_question_embedding = self.vectorize_user_question(user_question)

        similarities = cosine_similarity(
            user_question_embedding, np.vstack(self.df['Vectorized_Jawaban'])
        )

        most_similar_index = np.argmax(similarities)
        similarity_score = similarities[0, most_similar_index]

        if similarity_score < 0.15:
            return "Pertanyaan kurang relevan...", similarity_score

        return self.df.loc[most_similar_index, 'Jawaban'], similarity_score

    def answer_user_input(self, user_input):
        answer_result, similarity_score = self.answer_question(user_input)
        print(f"Pertanyaan: {user_input}")
        print(f"Jawaban Prediksi: {answer_result}")
        print(f"Similarity Index: {similarity_score}")
        print("")

qas_model = QASystem(df)
qas_model.preprocess_dataframe()

```

pytorch\_model.bin: 445M/445M [00:05<00:00,

400% 40.7MB/s

```

test_questions = [
    "Apa gejala dari depresi?",
    "Bagaimana cara mengatasi depresi?",
    "Bagaimana cara mengatasi gangguan kecemasan?",
    "Apa itu ADHD dan bagaimana cara diagnosisnya?",
    "Bagaimana mengatasi stres berlebihan?",
    "Apakah ada makanan yang menaikkan mood?",
    "Apa saja metode terapi yang digunakan untuk skizofrenia?",
    "Apa yang dimaksud dengan self-harm dalam konteks kesehatan mental?",
    "Bagaimana cara mengenali tanda-tanda gangguan mental?",
    "Apakah olahraga dapat membantu meningkatkan kesehatan mental?",
    "Berapa banyak dosen di prodi sains data?",
    "Apakah resep dari obat untuk penyakit mental ??",
    "Bagaimana cara merawat ikan hias?",
    "Apa saja jenis kamera yang direkomendasikan untuk fotografi profesional?",
    "Siapa presiden kedua Indonesia?",
    "Apa itu teori relativitas  $E = MC^2$ ?",
    "Bagaimana cara membuat aplikasi?",
    "Apa yang dimaksud dengan hukum permintaan dan penawaran?",
    "Obat apa yang harus dibeli untuk menenangkan pasien?",
    "Bagaimana cara mengatasi rambut bercabang?"
]

# List to store results
user_results = []

# Answer user questions and store results
for user_question in test_questions:
    result = qas_model.answer_user_input(user_question)

    if result is not None:
        answer_result, similarity_score = result
        user_result = {
            "Pertanyaan Pengguna": user_question,
            "Jawaban": answer_result,
            "Similarity Index": similarity_score
        }
        user_results.append(user_result)
    else:
        print(f"Unable to answer question: {user_question}")

# Create DataFrame and export to Excel
user_results_df = pd.DataFrame(user_results)
user_results_df.to_excel("user_results.xlsx", index=False)

```

Pertanyaan: Apa gejala dari depresi?  
 Jawaban Prediksi: Jika Anda atau seseorang yang Anda kenal dalam krisis, perawatan rawat inap dapat membantu. Perawatan rawat inap dapat membantu.  
 Similarity Index: 0.5485141277313232

Unable to answer question: Apa gejala dari depresi?  
 Pertanyaan: Bagaimana cara mengatasi depresi?  
 Jawaban Prediksi: Mirip dengan Petunjuk Kemajuan Medis atau Kekuasaan Layanan Kesehatan, Petunjuk Kemajuan Psikiatri adalah dokumen yang memberikan informasi tentang perawatan dan pengobatan.  
 Similarity Index: 0.5272756218910217

Unable to answer question: Bagaimana cara mengatasi depresi?  
 Pertanyaan: Bagaimana cara mengatasi gangguan kecemasan?  
 Jawaban Prediksi: Merasa nyaman dengan profesional yang Anda atau anak Anda bekerja sama sangat penting untuk keberhasilan perawatan.  
 Similarity Index: 0.5147053003311157

Unable to answer question: Bagaimana cara mengatasi gangguan kecemasan?  
 Pertanyaan: Apa itu ADHD dan bagaimana cara diagnosisnya?  
 Jawaban Prediksi: Berbagai jenis terapi lebih efektif berdasarkan sifat kondisi kesehatan mental dan/atau gejala dan orang yang mengalami kondisi tersebut.  
 Similarity Index: 0.5354523658752441

Unable to answer question: Apa itu ADHD dan bagaimana cara diagnosisnya?  
 Pertanyaan: Bagaimana mengatasi stres berlebihan?  
 Jawaban Prediksi: Seringkali lebih realistis dan bermanfaat untuk mengetahui apa yang membantu dengan masalah yang Anda hadapi. Berbicara dengan profesional kesehatan mental dapat membantu.  
 Similarity Index: 0.5482479333877563

Unable to answer question: Bagaimana mengatasi stres berlebihan?  
 Pertanyaan: Apakah ada makanan yang menaikkan mood?  
 Jawaban Prediksi: Jika Anda atau seseorang yang Anda kenal dalam krisis, perawatan rawat inap dapat membantu. Perawatan rawat inap dapat membantu.  
 Similarity Index: 0.5330169200897217

Unable to answer question: Apakah ada makanan yang menaikkan mood?  
 Pertanyaan: Apa saja metode terapi yang digunakan untuk skizofrenia?  
 Jawaban Prediksi: Jika Anda atau seseorang yang Anda kenal dalam krisis, perawatan rawat inap dapat membantu. Perawatan rawat inap dapat membantu.  
 Similarity Index: 0.6563464403152466

Unable to answer question: Apa saja metode terapi yang digunakan untuk skizofrenia?  
 Pertanyaan: Apa yang dimaksud dengan self-harm dalam konteks kesehatan mental?  
 Jawaban Prediksi: Jika Anda atau seseorang yang Anda kenal dalam krisis, perawatan rawat inap dapat membantu. Perawatan rawat inap dapat membantu.  
 Similarity Index: 0.5893571376800537

Unable to answer question: Apa yang dimaksud dengan self-harm dalam konteks kesehatan mental?  
 Pertanyaan: Bagaimana cara mengenali tanda-tanda gangguan mental?

Jawaban Prediksi: Kita semua memiliki kesehatan mental yang terdiri dari keyakinan, pikiran, perasaan, dan perilaku kita.  
Similarity Index: 0.6235824823379517

Unable to answer question: Bagaimana cara mengenali tanda-tanda gangguan mental?

Pertanyaan: Apakah olahraga dapat membantu meningkatkan kesehatan mental?

Jawaban Prediksi: Kita semua memiliki kesehatan mental yang terdiri dari keyakinan, pikiran, perasaan, dan perilaku kita.  
Similarity Index: 0.6021199226379395

Unable to answer question: Apakah olahraga dapat membantu meningkatkan kesehatan mental?

Pertanyaan: Berapa banyak dosen di prodi sains data?

Jawaban Prediksi: Ada banyak jenis profesional kesehatan mental. Menemukan yang tepat untuk Anda mungkin memerlukan penelitian.  
Similarity Index: 0.6396421790122986

Unable to answer question: Berapa banyak dosen di prodi sains data?

Pertanyaan: Apakah resep dari obat untuk penyakit mental ??

Jawaban Prediksi: Kita semua memiliki kesehatan mental yang terdiri dari keyakinan, pikiran, perasaan, dan perilaku kita.

## ✓ IndoBert QA

```
from transformers import pipeline
import pandas as pd

df = pd.read_csv('/content/dataset_mentalhealth.csv')
qa_pipeline = pipeline("question-answering", model="Rifky/Indobert-QA", tokenizer="Rifky/Indobert-QA")

def answer_question(row):
    question = row['Questions']
    context = row['Jawaban']
    inputs = {'question': question, 'context': context}
    answer = qa_pipeline(inputs)
    return answer['answer']

df['Answer'] = df.apply(answer_question, axis=1)
print(df[['Question_ID', 'Questions', 'Jawaban', 'Answer']])
```



model.safetensors: 440M/440M [00:03<00:00,  
100% 128MB/s]

```
from transformers import pipeline
import pandas as pd

class InteractiveQAS:
    def __init__(self, model_name="Rifky/Indobert-QA"):
        self.qa_pipeline = pipeline("question-answering", model=model_name)
        self.df = pd.DataFrame()

    def load_data(self, data_frame):
        self.df = data_frame

    def answer_question(self, question):
        context = " ".join(self.df['Jawaban'].tolist())

        inputs = {'question': question, 'context': context}
        answer = self.qa_pipeline(inputs)

        return answer['answer']

df
```

	Question_ID	Questions	Jawaban	Answer
0	1590140	Apa yang dimaksud dengan penyakit mental?	Penyakit mental adalah kondisi kesehatan yang ...	adalah kondisi kesehatan yang mengganggu pikir...
1	2110618	Siapa yang terpengaruh oleh penyakit mental?	Diperkirakan bahwa penyakit mental mempengaruh...	1 dari 5 orang dewasa di Amerika
		Apa penyebab	Diperkirakan bahwa	Ini dapat