LAPORAN TUGAS AKHIR REKAYASA ORGANISASI DAN BIG DATA

Disusun oleh :
Imam Akbar Nugraha - 1305213056

PROGRAM STUDI SARJANA SAINS DATA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2024

1. **Install transformers pinecone-client torch pandas**

```
!pip install transformers pinecone-client torch pandas
```

```
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (2.3.0+cu121)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.14.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.23.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.5.15)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.4)
Requirement already satisfied: certifi>=2019.11.17 in /usr/local/lib/python3.10/dist-packages (from pinecone-client) (2024.6.2)
Collecting pinecone-plugin-interface<0.0.8,>=0.0.7 (from pinecone-client)
  Downloading pinecone_plugin_interface-0.0.7-py3-none-any.whl (6.2 kB)
Requirement already satisfied: typing-extensions>=3.7.4 in /usr/local/lib/python3.10/dist-packages (from pinecone-client) (4.12.2)
Requirement already satisfied: urllib3>=1.26.0 in /usr/local/lib/python3.10/dist-packages (from pinecone-client) (2.0.7)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch) (1.12.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch) (2023.6.0)
Collecting nvidia-cuda-nvrtc-cu12==12.1.105 (from torch)
  Using cached nvidia_cuda_nvrtc_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (23.7 MB)
Collecting nvidia-cuda-runtime-cu12==12.1.105 (from torch)
  Using cached nvidia_cuda_runtime_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (823 kB)
Collecting nvidia-cuda-cupti-cu12==12.1.105 (from torch)
  Using cached nvidia_cuda_cupti_cu12-12.1.105-py3-none-manylinux1_x86_64.whl (14.1 MB)
Collecting nvidia-cudnn-cu12==8.9.2.26 (from torch)
  Using cached nvidia_cudnn_cu12-8.9.2.26-py3-none-manylinux1_x86_64.whl (731.7 MB)
Collecting nvidia-cublas-cu12==12.1.3.1 (from torch)
```

Baris perintah !pip install transformers pinecone-client torch pandas digunakan untuk menginstal paket transformers, pinecone-client, torch, dan pandas menggunakan pip. Paket transformers menyediakan berbagai model transformer seperti BERT, sementara pinecone-client adalah klien Python untuk layanan Pinecone yang digunakan untuk membuat dan mengelola indeks vektor. torch adalah pustaka yang mendukung komputasi numerik dan pembelajaran mesin, khususnya dalam pemrosesan model BERT. pandas digunakan untuk manipulasi dan analisis data dalam bentuk DataFrame.

2. **Import Modul yang Diperlukan**

```
import pandas as pd
import pinecone
from transformers import BertTokenizer, BertModel
import torch
import time
```

Pada bagian ini, modul yang diperlukan diimpor: pandas untuk manipulasi data, pinecone untuk berinteraksi dengan layanan Pinecone, transformers untuk menggunakan tokenizer dan model BERT, torch untuk manipulasi tensor dan komputasi model, serta time untuk mengukur waktu eksekusi. Modul-modul ini menyediakan fungsi dan kelas yang diperlukan untuk menjalankan tugas-tugas berikutnya dalam notebook.

3. Membaca Dataset

```
# Membaca data dari CSV
df = pd.read_csv('/content/spam.csv', sep=',', encoding='latin-1')
df
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

Kode df = pd.read_csv('/content/spam.csv', sep=',', encoding='latin-1') digunakan untuk membaca file CSV yang berisi data spam dan memuatnya ke dalam DataFrame df. Parameter sep=',' menunjukkan bahwa pemisah kolom adalah koma, dan encoding='latin-1' digunakan untuk memastikan bahwa karakter dalam file CSV dapat dibaca dengan benar. DataFrame df kemudian digunakan untuk menyimpan dan memanipulasi data dalam bentuk tabel.

4. Inisialisasi Tokenizer dan Model BERT

```
# Inisialisasi tokenizer dan model BERT
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased')

# Fungsi untuk mendapatkan embedding teks
def get_text_embedding(text):
    inputs = tokenizer(text, return_tensors='pt', truncation=True, padding=True, max_length=512)
    with torch.no_grad():
        outputs = model(**inputs)
    embeddings = outputs.last_hidden_state.mean(dim=1)
    return embeddings.detach().numpy().flatten().tolist()
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: 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), set it as secret in your Google Colab and restart your session.
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 or datasets.
  warnings.warn(
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 675B/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 3.97MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 6.00MB/s]
/usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning: `resume_download` is deprecated and will be removed in version 1.0.0. Downloads always resume when
  warnings.warn(
config.json: 100%  570/570 [00:00<00:00, 5.77kB/s]
model.safetensors: 100%  440M/440M [00:03<00:00, 101MB/s]
```

Tokenizer dan model BERT diinisialisasi menggunakan model pra-terlatih bert-base-uncased dari pustaka transformers. Fungsi get_text_embedding digunakan untuk mendapatkan embedding teks dengan mengubah teks menjadi token menggunakan tokenizer, kemudian memproses token tersebut dengan model BERT

untuk menghasilkan tensor embedding. Keluaran dari BERT diambil dari last_hidden_state dan dirata-rata untuk mendapatkan representasi embedding.

5. **Inisialisasi Pinecone**

```
# Inisialisasi Pinecone
from pinecone import Pinecone, ServerlessSpec

api_key = "c61bfa4d-6c77-45d7-9cde-3b525cea05de"
pc = Pinecone(api_key=api_key)
```

Pada bagian ini, klien Pinecone diinisialisasi dengan menggunakan api_key yang diberikan. ServerlessSpec digunakan untuk mengatur spesifikasi serverless untuk index yang akan dibuat. Pinecone adalah layanan yang menyediakan indeks vektor yang dapat digunakan untuk pencarian kemiripan teks berbasis embedding.

6. **Membuat Index Pinecone**

```
# Membuat index Pinecone
index_name = "spam-messages-index"
if index_name not in pc.list_indexes():
    pc.create_index(
        name=index_name,
        dimension=768,  # Replace with your model dimensions
        metric="cosine",  # Replace with your model metric
        spec=ServerlessSpec(
            cloud="aws",
            region="us-east-1"
        )
    )
```

Kode ini mendefinisikan nama index sebagai spam-messages-index dan mengecek apakah index dengan nama tersebut sudah ada di Pinecone. Jika belum ada, index dibuat dengan dimensi 768 (sesuai dengan dimensi embedding dari model BERT) dan metrik cosine similarity. ServerlessSpec digunakan untuk mengatur spesifikasi serverless dengan cloud aws dan region us-east-1.

7. **Menghubungkan ke Index**

```
# Menghubungkan ke index
index = pc.Index(index_name)
```

Baris kode index = pc.Index(index_name) digunakan untuk menghubungkan ke index spam-messages-index yang telah dibuat atau sudah ada. Ini memungkinkan kita untuk melakukan operasi pencarian dan manipulasi data pada index tersebut.

8. **Memasukkan Data ke Pinecone**

```
# Memasukkan data ke Pinecone
for i, row in df.iterrows():
    text = row['v2']  # Asumsikan kolom teks adalah 'v2'
    embedding = get_text_embedding(text)
    index.upsert([(str(i), embedding)])
```

Dalam bagian ini, setiap baris dalam DataFrame df diiterasi. Teks diambil dari kolom v2, kemudian embedding teks diperoleh menggunakan fungsi get_text_embedding. Data embedding kemudian dimasukkan ke dalam index Pinecone menggunakan metode upsert, yang meng-update atau menyisipkan data baru ke dalam index.

9. **Melakukan Query pada Index**

```
# Melakukan query pada index
query_texts = [
    "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005...",
    "WINNER!! As a valued network customer you have been selected to receivea £900 prize reward!",
    "I HAVE A DATE ON SUNDAY WITH WILL!!",
    "URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot!",
    "I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, I've cried enough today.",
    "Oh k...i'm watching here:)",
    "Fine if that's the way u feel. That's the way it's gota b"
]
```

Daftar teks query didefinisikan untuk digunakan dalam pencarian pada index Pinecone. Teks-teks ini merupakan contoh pesan spam yang akan dicari kemiripannya dalam index.

10. **Fungsi untuk Mengukur Waktu Eksekusi Query**

```
# Fungsi untuk mengukur waktu eksekusi query
def measure_query_time(query_text):
    query_embedding = get_text_embedding(query_text)
    start_time = time.time()
    query_response = index.query(vector=query_embedding, top_k=10)
    end_time = time.time()
    duration = end_time - start_time
    return duration, query_response
```

Fungsi measure_query_time didefinisikan untuk mengukur waktu yang diperlukan untuk menjalankan query pada Pinecone. Fungsi ini mengubah teks query menjadi embedding, mencatat waktu sebelum dan sesudah query, serta menghitung durasi eksekusi. Fungsi ini mengembalikan durasi eksekusi dan respons query dari Pinecone.

11. **Menjalankan Query dan Mencatat Waktu Eksekusi**

```python
# Menjalankan query dan mencatat waktu eksekusi
for query_text in query_texts:
    duration, query_response = measure_query_time(query_text)
    print(f"Query: {query_text}")
    print(f"Waktu eksekusi: {duration} detik")
    for match in query_response['matches']:
        print(f"ID: {match['id']}, Score: {match['score']}")
```

```
Query: Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005...
Waktu eksekusi: 0.729658842086792 detik
ID: 1162, Score: 0.89864713
ID: 2, Score: 0.89864713
ID: 4075, Score: 0.819228
ID: 2595, Score: 0.809554
ID: 877, Score: 0.809554
ID: 4241, Score: 0.806566954
ID: 4067, Score: 0.802371562
ID: 2611, Score: 0.794367313
ID: 2832, Score: 0.794095933
ID: 3003, Score: 0.794095933
Query: WINNER!! As a valued network customer you have been selected to receivea £900 prize reward!
Waktu eksekusi: 0.026117324829101562 detik
ID: 4034, Score: 0.885393679
ID: 5232, Score: 0.885393679
ID: 8, Score: 0.879962444
ID: 4279, Score: 0.878916323
ID: 1873, Score: 0.866761446
ID: 2495, Score: 0.848759711
ID: 3696, Score: 0.841698885
ID: 2112, Score: 0.840568244
ID: 93, Score: 0.839175761
ID: 4626, Score: 0.837771893
Query: I HAVE A DATE ON SUNDAY WITH WILL!!
Waktu eksekusi: 0.020369768142700195 detik
```

```
Query: I HAVE A DATE ON SUNDAY WITH WILL!!
Waktu eksekusi: 0.020369768142700195 detik
ID: 14, Score: 1.00100017
ID: 5155, Score: 0.78828609
ID: 259, Score: 0.780409098
ID: 1661, Score: 0.775901139
ID: 651, Score: 0.774294
ID: 4509, Score: 0.769461453
ID: 2937, Score: 0.769382179
ID: 2655, Score: 0.765554607
ID: 1878, Score: 0.762715399
ID: 3467, Score: 0.759862602
Query: URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot!
Waktu eksekusi: 0.02213001251220703 detik
ID: 93, Score: 0.844981
ID: 4626, Score: 0.842395842
ID: 187, Score: 0.840842664
ID: 1001, Score: 0.840842664
ID: 4040, Score: 0.838838458
ID: 4010, Score: 0.838838458
ID: 3966, Score: 0.83523792
ID: 4843, Score: 0.83523792
ID: 1873, Score: 0.832195938
ID: 4650, Score: 0.830685377
```

```
Query: I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, I've cried enough today.
Waktu eksekusi: 0.018512964248657227 detik
ID: 10, Score: 0.943577588
ID: 4183, Score: 0.823076427
ID: 99, Score: 0.803334355
ID: 2874, Score: 0.802716732
ID: 3665, Score: 0.799038768
ID: 3301, Score: 0.797587812
ID: 5538, Score: 0.793954194
ID: 846, Score: 0.791764677
ID: 5525, Score: 0.783107162
ID: 5246, Score: 0.780059159
Query: Oh k...i'm watching here:)
Waktu eksekusi: 0.018034934997558594 detik
ID: 16, Score: 1.00191426
ID: 399, Score: 0.845362425
ID: 4770, Score: 0.834993958
ID: 1218, Score: 0.829329491
ID: 2563, Score: 0.829329491
ID: 4386, Score: 0.82886517
ID: 3031, Score: 0.823122561
ID: 3009, Score: 0.818938792
ID: 2667, Score: 0.818586349
ID: 5321, Score: 0.818319201
Query: Fine if that's the way u feel. That's the way it's gota b
Waktu eksekusi: 0.016928672790527344 detik
ID: 18, Score: 0.90038836
ID: 4920, Score: 0.798796058
ID: 601, Score: 0.788626432
ID: 1830, Score: 0.784810364
ID: 1587, Score: 0.782213569
ID: 1784, Score: 0.782213569
ID: 4777, Score: 0.780885398
ID: 3189, Score: 0.779765904
ID: 1311, Score: 0.778784335
```


```
Query: Fine if that's the way u feel. That's the way it's gota b
Waktu eksekusi: 0.016928672790527344 detik
ID: 18, Score: 0.90038836
ID: 4920, Score: 0.798796058
ID: 601, Score: 0.788626432
ID: 1830, Score: 0.784810364
ID: 1587, Score: 0.782213569
ID: 1784, Score: 0.782213569
ID: 4777, Score: 0.780885398
ID: 3189, Score: 0.779765904
ID: 1311, Score: 0.778784335
ID: 1509, Score: 0.777925074
```


Pada bagian ini, query dijalankan untuk setiap teks dalam query_texts. Waktu eksekusi dicatat dan ditampilkan untuk setiap query. Hasil query, termasuk ID dan skor dari setiap pencocokan, juga ditampilkan. Hal ini memungkinkan untuk melihat performa dan hasil dari pencarian kemiripan teks pada index Pinecone.

12. Optimasi Query

```python
# Optimasi Query
# Pinecone tidak memiliki parameter ef_search, tetapi Anda dapat menggunakan filter atau menyesuaikan jumlah hasil
optimized_query_texts = [
    "Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005...",
    "WINNER!! As a valued network customer you have been selected to receivea £900 prize reward!",
    "I HAVE A DATE ON SUNDAY WITH WILL!!",
    "URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot!",
    "I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, I've cried enough today.",
    "Oh k...i'm watching here:)",
    "Fine if that's the way u feel. That's the way it's gota b"
]
```

Daftar teks query yang dioptimasi didefinisikan. Optimasi ini bisa mencakup penggunaan filter atau parameter tambahan dalam query Pinecone untuk meningkatkan performa atau akurasi pencarian.

13. Menjalankan Query Optimasi dan Mencatat Waktu Eksekusi

```
# Menjalankan query optimasi dan mencatat waktu eksekusi
for query_text in optimized_query_texts:
    duration, query_response = measure_query_time(query_text)
    print(f"Query (Optimasi): {query_text}")
    print(f"Waktu eksekusi setelah optimasi: {duration} detik")
    for match in query_response['matches']:
        print(f"ID: {match['id']}, Score: {match['score']}")
```

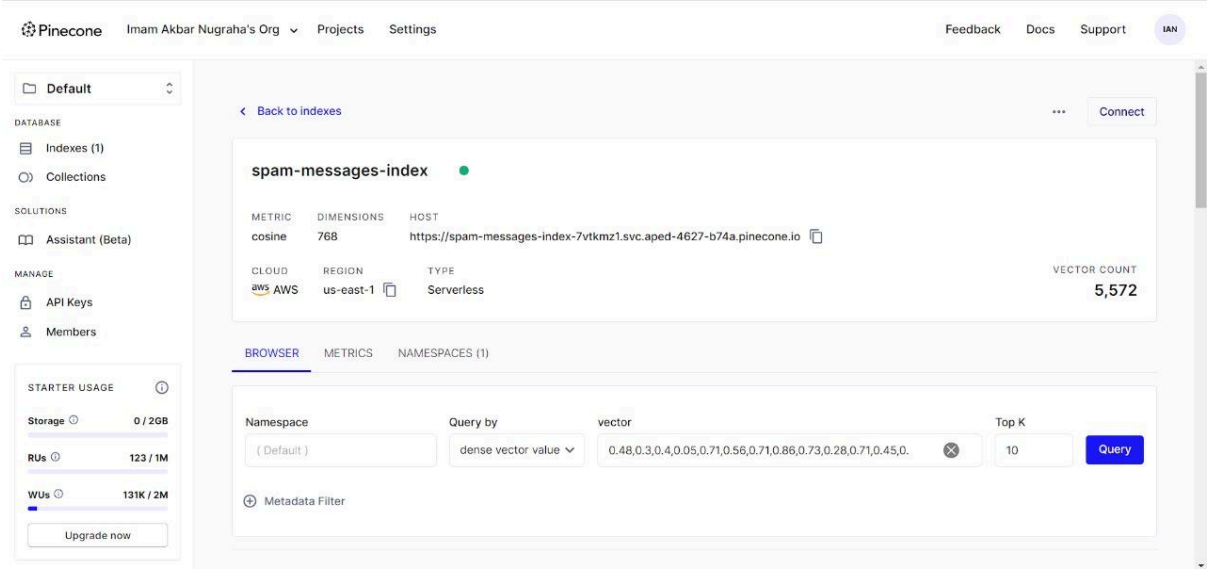
```
Query (Optimasi): Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005...
Waktu eksekusi setelah optimasi: 0.1882789134979248 detik
ID: 1162, Score: 0.89864713
ID: 2, Score: 0.89864713
ID: 4075, Score: 0.819228
ID: 2595, Score: 0.809554
ID: 877, Score: 0.809554
ID: 4241, Score: 0.806566954
ID: 4067, Score: 0.802371562
ID: 2611, Score: 0.794367313
ID: 2832, Score: 0.794095933
ID: 3003, Score: 0.794095933
Query (Optimasi): WINNER!! As a valued network customer you have been selected to receivea £900 prize reward!
Waktu eksekusi setelah optimasi: 0.025378704071044922 detik
ID: 4034, Score: 0.885393679
ID: 5232, Score: 0.885393679
ID: 8, Score: 0.879962444
ID: 4279, Score: 0.878916323
ID: 1873, Score: 0.866761446
ID: 2495, Score: 0.848759711
ID: 3696, Score: 0.841698885
ID: 2112, Score: 0.840568244
ID: 93, Score: 0.839175761
ID: 4626, Score: 0.837771893
Query (Optimasi): I HAVE A DATE ON SUNDAY WITH WILL!!
Waktu eksekusi setelah optimasi: 0.019684314727783203 detik
ID: 14, Score: 1.00100017
ID: 5155, Score: 0.78828609
ID: 259, Score: 0.780409098
ID: 1661, Score: 0.775901139
ID: 651, Score: 0.774294
ID: 4509, Score: 0.769461453
ID: 2937, Score: 0.769382179
ID: 2655, Score: 0.765554607
ID: 1878, Score: 0.762715399
ID: 3467, Score: 0.759862602
Query (Optimasi): URGENT! You have won a 1 week FREE membership in our £100,000 Prize Jackpot!
Waktu eksekusi setelah optimasi: 0.019484996795654297 detik
ID: 93, Score: 0.844981
ID: 4626, Score: 0.842395842
ID: 187, Score: 0.840842664
ID: 1001, Score: 0.840842664
ID: 4040, Score: 0.838838458
ID: 4010, Score: 0.838838458
ID: 3966, Score: 0.83523792
ID: 4843, Score: 0.83523792
ID: 1873, Score: 0.832195938
ID: 4650, Score: 0.830685377
```

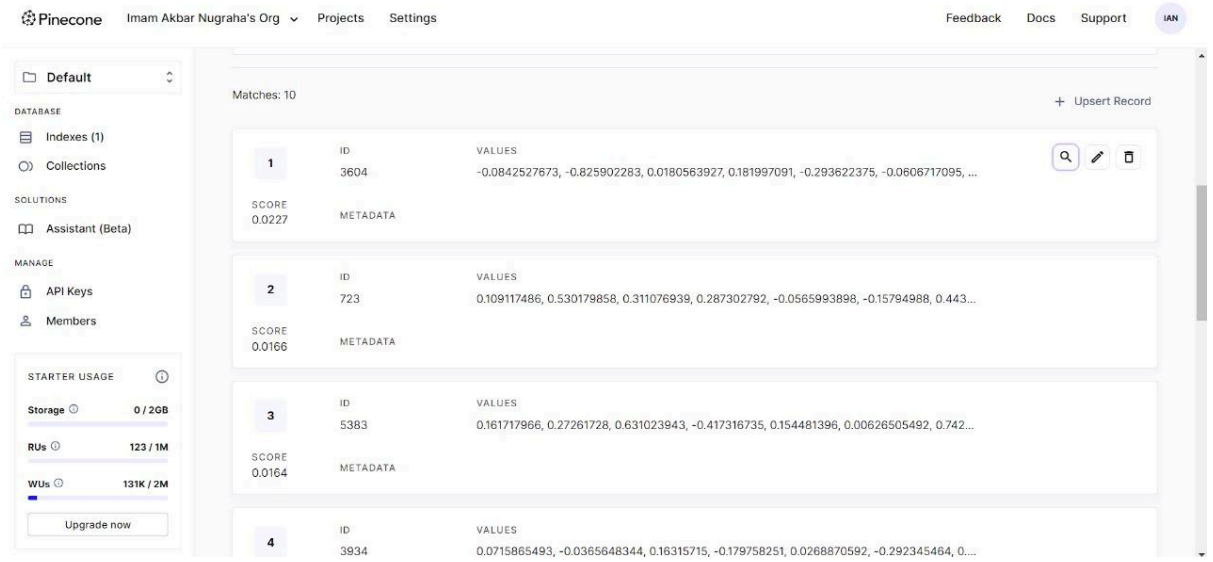
```
Query (Optimasi): I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, I've cried enough today.
Waktu eksekusi setelah optimasi: 0.022426605224609375 detik
ID: 10, Score: 0.943577588
ID: 4183, Score: 0.823076427
ID: 99, Score: 0.803334355
ID: 2874, Score: 0.802716732
ID: 3665, Score: 0.799038768
ID: 3301, Score: 0.797587812
ID: 5538, Score: 0.793954194
ID: 846, Score: 0.791764677
ID: 5525, Score: 0.783107162
ID: 5246, Score: 0.780059159
Query (Optimasi): Oh k...i'm watching here:)
Waktu eksekusi setelah optimasi: 0.020049333572387695 detik
ID: 16, Score: 1.00191426
ID: 399, Score: 0.845362425
ID: 4770, Score: 0.834993958
ID: 1218, Score: 0.829329491
ID: 2563, Score: 0.829329491
ID: 4386, Score: 0.82886517
ID: 3031, Score: 0.823122561
ID: 3009, Score: 0.818938792
ID: 2667, Score: 0.818586349
ID: 5321, Score: 0.818319201
Query (Optimasi): Fine if that's the way u feel. That's the way it's gota b
Waktu eksekusi setelah optimasi: 0.023157596588134766 detik
ID: 18, Score: 0.90038836
ID: 4920, Score: 0.798796058
ID: 601, Score: 0.788626432
ID: 1830, Score: 0.784810364
ID: 1587, Score: 0.782213569
ID: 1784, Score: 0.782213569
ID: 4777, Score: 0.780885398
ID: 3189, Score: 0.779765904
ID: 1311, Score: 0.778784335
```

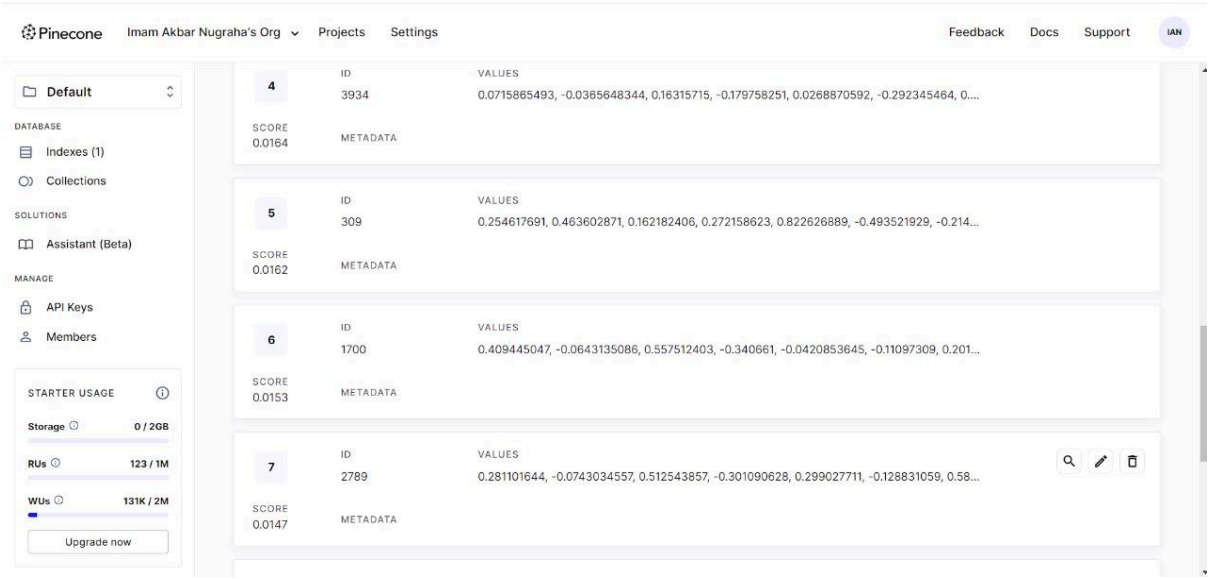
```
Query (Optimasi): Fine if that's the way u feel. That's the way it's gota b
Waktu eksekusi setelah optimasi: 0.023157596588134766 detik
ID: 18, Score: 0.90038836
ID: 4920, Score: 0.798796058
ID: 601, Score: 0.788626432
ID: 1830, Score: 0.784810364
ID: 1587, Score: 0.782213569
ID: 1784, Score: 0.782213569
ID: 4777, Score: 0.780885398
ID: 3189, Score: 0.779765904
ID: 1311, Score: 0.778784335
ID: 1509, Score: 0.777925074
```

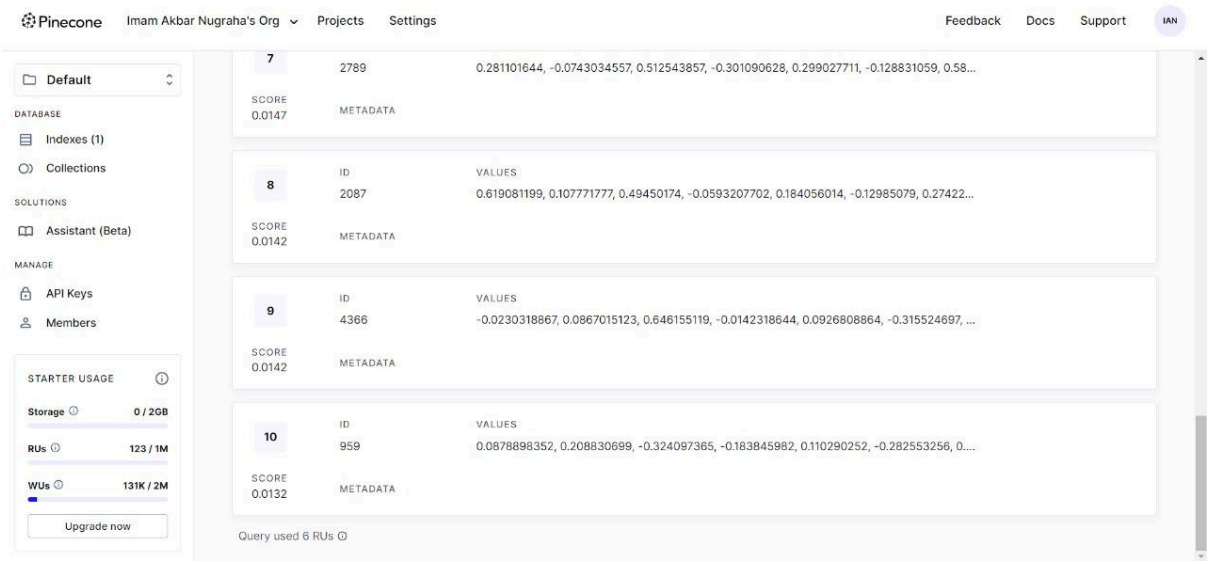
Query dijalankan pada teks yang dioptimasi, dan waktu eksekusi dicatat serta ditampilkan untuk setiap query yang dioptimasi. Hasil query yang dioptimasi, termasuk ID dan skor dari setiap pencocokan, juga ditampilkan. Hal ini memungkinkan untuk membandingkan performa dan hasil antara query standar dan query yang dioptimasi.

14. Output Membuat Index pada Database Pinecone









Tampilan dashboard Pinecone digunakan untuk mengelola indeks vektor, dengan detail indeks bernama `spam-messages-index`, yang dirancang untuk mengelola data terkait pembelian pulsa dan pengecekan kuota. Metrik yang digunakan adalah cosine untuk mengukur kesamaan antar vektor, dan dimensi indeks adalah 768, sesuai dengan panjang vektor embedding dari model seperti BERT. Indeks ini di-host di AWS, dalam region `us-east-1`, menggunakan tipe serverless untuk kemudahan skalabilitas dan manajemen infrastruktur. Pengguna dapat memasukkan vektor query langsung di antarmuka, menentukan jumlah hasil teratas (Top K) yang diinginkan, dan menjalankan pencarian dengan menekan tombol Query. Hasil pencarian ditampilkan dengan skor relevansi dan ID dari setiap hasil, menunjukkan tingkat kesamaan dengan vektor query yang dimasukkan. Setiap hasil memiliki ID unik dan skor relevansi yang menunjukkan kemiripannya dengan query, serta menampilkan sebagian dari nilai vektor terkait hasil pencarian. Tampilan ini memperlihatkan bagaimana Pinecone memudahkan manajemen dan pencarian dalam indeks vektor, memungkinkan pengguna untuk menjalankan pencarian berbasis kesamaan vektor dengan mudah dan mendapatkan hasil relevan berdasarkan metrik yang telah ditentukan. Sistem ini sangat berguna untuk berbagai aplikasi seperti pencarian teks, rekomendasi produk, dan analisis data berbasis vektor lainnya.

Link Google Collabs :
https://colab.research.google.com/drive/1z4TBI426q99Qvx4nN014rb1lcWjZnhrB?usp=sharing

Link Video Presentasi :
https://drive.google.com/drive/folders/1cJ8b_dXraru5BEWNyk_WDDzoFif5_A2Q?usp=sharing