

Laporan Tugas Besar I IF2123
Aljabar Linier dan Geometri
Sistem Persamaan Linier, Determinan, dan Aplikasinya

Oleh:



Kelompok 53 - Hawk 2: uhh

Joel Hotlan Haris Siahaan 13523025

Samuel Gerrard 13523064
Hamonangan G.

Muhammad Iqbal Haidar 13522111

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132

2024

Daftar Isi

Daftar Isi.....	2
Daftar Gambar.....	3
BAB I	
DESKRIPSI MASALAH.....	4
1.1 Pemrosesan Suara dan Gambar.....	4
1.2 Pemrosesan Suara dan Gambar.....	5
1.3 Album Picture Finder - Principal Component Analysis.....	5
1.4 Music Information Retrieval - Query by Humming.....	10
1.5 Arsitektur Website.....	13
BAB II	
TEORI SINGKAT.....	16
2.1 Sistem Temu Balik Suara (MIR).....	16
2.2 Metode Ekstraksi Fitur Berdasarkan Humming.....	16
2.3 Image Retrieval dengan PCA.....	17
BAB III	
ARSITEKTUR FRONTEND DAN BACKEND.....	19
3.1 Frontend.....	19
3.1.1. Judul Website.....	19
3.1.2. Komponen Pencarian dan Pengunggahan Data.....	19
3.1.5. Pagination.....	20
3.1.6. Hasil Pencarian.....	21
3.2 Backend.....	21
BAB IV	
EKSPERIMEN.....	23
4.1 Eksperimen 1.....	23
4.2 Eksperimen 2.....	24
4.3 Eksperimen 3.....	26
BAB V	
KESIMPULAN.....	28
5.1 Kesimpulan.....	28
5.2 Saran.....	28
5.3 Komentar.....	28
LAMPIRAN.....	29

Daftar Gambar

Gambar 1.1 Shazam sebagai aplikasi audio retrieval system	05
Gambar 1.2 Rata-Rata Pixel	08
Gambar 1.3 Pixel Standardization	08
Gambar 1.4 Matriks Kovarian	09
Gambar 1.5 Singular Value Decomposition	09
Gambar 1.6 Proyeksi Gambar ke Komponen Utama	09
Gambar 1.7 Representasi Query Dalam Komponen Utama	10
Gambar 1.8 Euclidean Distance	10
Gambar 1.9 Tahapan pada MIR dengan metode Query By Humming	11
Gambar 1.10 Normalisasi Tempo	12
Gambar 1.11 Normalisasi Histogram	13
Gambar 1.12 Cosine Similarity Formula	13
Gambar 1.13 Contoh tampak depan dari website Information Retrieval (Humming)	14
Gambar 1.14 Contoh tampak depan dari website Information Retrieval (Album)	15
Gambar 4.1 Dataset Gambar dan Audio Berhasil Upload Eksperimen 1	23
Gambar 4.2 Gambar Query Eksperimen 1	23
Gambar 4.3 Hasil Pencarian Image Retrieval Eksperimen 1	24
Gambar 4.4 Hasil Pencarian Music Retrieval Eksperimen 1	24
Gambar 4.5 Dataset Gambar dan Audio Berhasil Upload Eksperimen 2	25
Gambar 4.6 Gambar Query Eksperimen (Greyscale) 2	25
Gambar 4.7 Hasil Pencarian Image Retrieval Eksperimen 2	25
Gambar 4.8 Hasil Pencarian Music Retrieval Eksperimen 2	26
Gambar 4.9 Dataset Gambar dan Audio Berhasil Upload Eksperimen 3	26
Gambar 4.10 Gambar Query Eksperimen (Noise) 3	27
Gambar 4.11 Hasil Pencarian Image Retrieval Eksperimen 3	27
Gambar 4.12 Hasil Pencarian Music Retrieval Eksperimen 3	27

BAB I

DESKRIPSI MASALAH

1.1 Pemrosesan Suara dan Gambar

Suara selalu menjadi hal yang paling penting dalam kehidupan manusia. Manusia berbicara mengeluarkan suara dan mendengarkan suatu suara untuk diresap ke otak dan mencari informasi dari suara tersebut. Suara juga bisa dijadikan orang-orang di dunia ini sebuah media untuk membuat karya seni. Contohnya adalah alat mendeteksi lagu. Manusia bisa mendeteksi suara dengan menggunakan indera pendengar dan memberikan kesimpulan akan apa jenis suara tersebut melalui respon dari otak. Sama seperti manusia, teknologi juga bisa mendeteksi suara dan memberikan jawaban mereka melalui algoritma-algoritma yang beragam bahkan bisa melebihi kapabilitas manusia. Dengan menggunakan algoritma apapun, konsep dari pendeteksi dan interpretasi suara itu bisa juga disebut dengan sistem temu balik suara atau bisa disebut juga dengan *audio retrieval system*. Banyak aplikasi yang menggunakan konsep sistem temu balik contohnya adalah Shazam.



Gambar 1.1 Shazam sebagai aplikasi *audio retrieval system*

Selain suara, manusia juga memiliki penglihatan sebagai salah satu inderanya dan bisa melihat warna dan gambar yang bermacam-macam. Teknologi komputasi juga memiliki kapabilitas yang sama dan bisa melihat gambar sama seperti kita, tetapi teknologi seperti ini juga bisa merepresentasikan gambar tersebut sebagai beragam-ragam angka yang bisa disebut juga fitur. Tahun ke tahun, *image processing* selalu menjadi fokus utama dari tugas

besar 2 Algeo. Algoritma yang digunakan adalah Eigenvalue, Cosine Similarity, Euclidean Distance, dll.

Anda sudah melewati Tugas Besar 1 yaitu tentang matriks dan implementasi terhadap berbagai hal. Matriks adalah salah satu komponen yang penting dalam aplikasi aljabar vektor. Di dalam Tugas Besar 2 ini, anda diminta untuk membuat semacam aplikasi Shazam yaitu sebuah aplikasi yang meminta input lagu dan aplikasi tersebut mendeteksi apa nama dari lagu tersebut dan beberapa detail lainnya. Pada tugas besar ini, anda akan menggunakan aljabar vektor untuk mencari perbandingan antar satu audio dengan audio yang lain. Anda akan menggunakan konsep yang bernama *Music Information Retrieval* atau MIR untuk mencari dan mengidentifikasi suara berdasarkan fitur-fitur yang dimilikinya. Tidak hanya itu, anda juga akan menggunakan konsep Principal Component Analysis (PCA) untuk mencari kumpulan audio melalui deteksi wajah berbagai orang (anggap saja mereka sebagai seorang penyanyi).

1.2 Pemrosesan Suara dan Gambar

Information Retrieval adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Pada tugas besar ini, anda akan berkitik dengan 2 jenis *Information Retrieval*. *Image Retrieval* dan *Music Information Retrieval*. *Image Retrieval* adalah konsep untuk memasukkan sebuah input gambar dan berharap mendapatkan gambar yang ada di data sesuai dengan informasi dan perhitungan yang diinginkan. Sedangkan *Music Information Retrieval* (MIR) adalah konsep untuk memasukkan sebuah input audio dan berharap mendapatkan audio yang ada di data sesuai dengan informasi dan perhitungan yang diinginkan. Pada tugas besar kali ini, kalian akan mengimplementasikan *Image Retrieval* dengan menggunakan Principal Component Analysis dan *Music Information Retrieval* dengan menggunakan humming.

1.3 Album Picture Finder - Principal Component Analysis

Setiap audio pastinya memiliki gambar albumnya sendiri masing-masing. Terkadang kita lebih mengingat suatu visual dibandingkan nama dari lagu itu sendiri. Untuk memudahkan pengguna yang hanya memiliki gambar dari suatu album, maka dibutuhkan album finder dengan menggunakan teknik Principal Component Analysis (PCA).

Sebelum implementasi PCA, jangan lupa untuk memasukkan pemetaan nama audio dengan nama gambar yang bersangkutan, gunakan file .txt atau .json untuk melakukan pemetaan audio dengan gambar yang bersangkutan. Gunakan kreativitas anda untuk memetakan audio dengan gambar tersebut.

`mapper.txt`

```
-----  
audio_file pic_name  
audio_1.mid pic_1.png  
audio_2.mid pic_2.png  
audio_3.mid pic_3.png  
audio_4.mid pic_4.png  
.  
.
```

```
mapper.json
```

```
-----  
[  
  {  
    "audio_file": audio_1.mid,  
    "pic_name": pic_1.png  
  },  
  {  
    "audio_file": audio_2.mid,  
    "pic_name": pic_2.png  
  },  
  {  
    "audio_file": audio_3.mid,  
    "pic_name": pic_3.png  
  },  
  {  
    "audio_file": audio_4.mid,  
    "pic_name": pic_4.png  
  }  
]
```

Principal Component Analysis (PCA) adalah teknik statistik yang digunakan untuk mereduksi dimensi data dengan tetap mempertahankan sebanyak mungkin informasi yang ada. PCA mengubah data berdimensi tinggi menjadi beberapa dimensi yang lebih kecil, disebut principal components, tanpa kehilangan esensi atau pola utama dalam data tersebut. Hasil data yang didapatkan dari PCA ini akan berupa eigenvector dan proyeksi data.

Langkah-langkah untuk melakukan pencarian gambar menggunakan PCA adalah sebagai berikut:

1. Image Processing and Loading

Lakukan pemrosesan seluruh gambar yang ada pada dataset. Ubah gambar menjadi grayscale untuk mengurangi kompleksitas gambar dan membuat fokus menjadi

bagian terang dan gelap gambar. Yang berarti setiap gambar direpresentasikan dalam intensitas pixel saja tanpa informasi warna.

$$I(x,y) = 0.2989 \cdot R(x,y) + 0.5870 \cdot G(x,y) + 0.1140 \cdot B(x,y)$$

Selanjutnya, gambar akan diubah besarnya sehingga ukurannya sama untuk seluruh gambar. Ukuran seluruh gambar harus konsisten untuk membuat perhitungan semakin akurat.

Lalu ubah vektor grayscale pada gambar menjadi 1D untuk membuat dimensi vektor dapat dilakukan pemrosesan data. Jika gambar memiliki dimensi $M \times N$, maka hasilnya adalah vektor dengan panjang $M \cdot N$:

$$I = [I_1, I_2, \dots, I_{M \cdot N}]$$

2. Data Centering (Standardization)

Pada step ini lakukan standarisasi data di sekitar nilai 0. Hitung rata rata dari setiap gambar untuk suatu piksel.

$$\mu_j = \frac{1}{N} \sum_{i=1}^N x_{ij}$$

Gambar 1.2 Rata-Rata Pixel

di mana:

- x_{ij} : nilai piksel ke- j pada gambar ke- i ,
- N : jumlah total gambar dalam dataset.

Lalu kurangi piksel tersebut dengan rata-rata yang sudah dihitung untuk melakukan standarisasi.

$$x_{ij}' = x_{ij} - \mu_j$$

Gambar 1.3 Pixel Standardization

3. PCA Computation Using Singular Value Decomposition (SVD)

Lakukan perhitungan SVD pada gambar-gambar yang sudah distandarisasi. Buat matriks kovarians dari data yang sudah distandarisasi:

$$C = \frac{1}{N} X'^T X'$$

Gambar 1.4 Matriks Kovarian

di mana:

- X' : matriks data yang sudah distandarisasi. Lalu lakukan dekomposisi nilai singular untuk mendapatkan komponen utama:

$$C = U \Sigma U^T$$

Gambar 1.5 Singular Value Decomposition

- U : matriks eigenvector (komponen utama),
- Σ : matriks eigenvalue (menunjukkan varian data di sepanjang komponen utama).

Lalu ambil n jumlah component utama teratas dari hasil SVD dan lakukan proyeksikan gambar ke komponen utama.

Pilih k -komponen utama teratas ($k \ll M \cdot N$) dan proyeksikan data:

$$Z = X' U_k$$

Gambar 1.6 Proyeksi Gambar ke Komponen Utama

di mana:

- U_k : matriks eigenvector dengan n -dimensi.

4. Similarity Computation

Lakukan pencarian kesamaan dengan mencari jarak euclidean dari tiap dataset dengan gambar query. Lalu lakukan pengurutan kecocokan dari yang paling tinggi.

Pertama-tama, representasikan gambar query dalam ruang komponen utama dengan proyeksi yang sama:

$$\mathbf{q} = (\mathbf{q}' - \mu) \mathbf{U}_k$$

Gambar 1.7 Representasi Query Dalam Komponen Utama

Dimana:

- \mathbf{q} = Vektor proyeksi dari gambar query ke ruang komponen utama (PCA).
- \mathbf{q}' : Gambar query dalam format vektor (setelah grayscale, resize, dan flattening).
- μ : Rata-rata piksel dari dataset (per piksel).
- \mathbf{U}_k : Matriks eigenvector dengan k dimensi utama dari PCA.

Kemudian, hitung jarak Euclidean antara gambar query dengan semua gambar dalam dataset:

$$d(\mathbf{q}, \mathbf{z}_i) = \sqrt{\sum_{j=1}^k (q_j - z_{ij})^2}$$

Gambar 1.8 Euclidean Distance

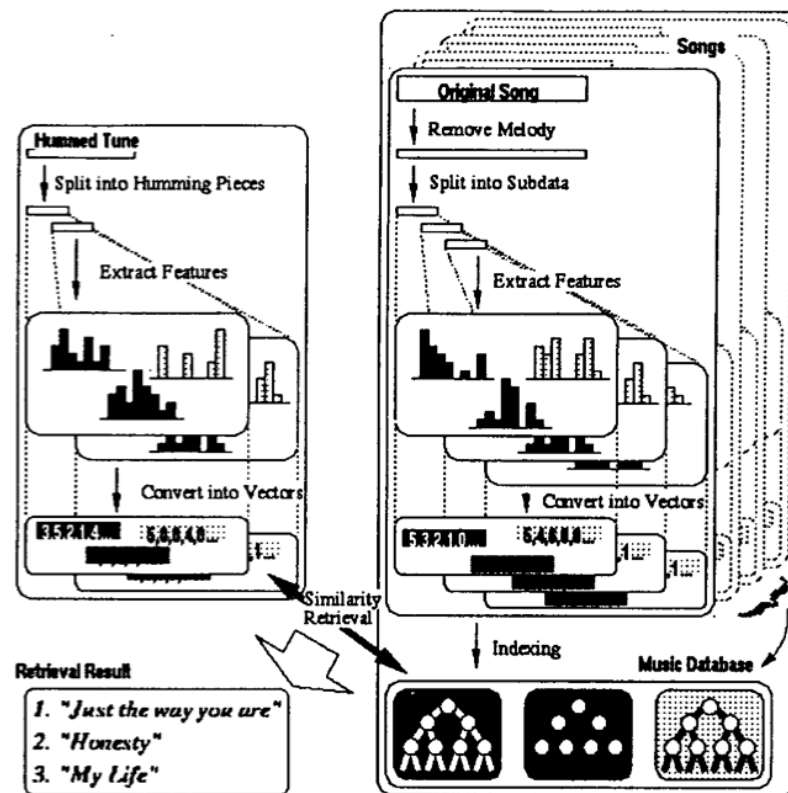
- $d(\mathbf{q}, \mathbf{z}_i)$ = Jarak antara gambar query \mathbf{q} dan gambar ke- i dalam ruang komponen utama.
- \mathbf{z}_i = Vektor proyeksi dari gambar ke- i dalam dataset ke ruang komponen utama.
- q_j : Elemen ke- j dari vektor proyeksi query \mathbf{q} .
- z_{ij} : Elemen ke- j dari vektor proyeksi gambar ke- i , yaitu \mathbf{z}_i .
- k : Jumlah dimensi ruang komponen utama yang dipilih.

Lalu, Urutkan hasil berdasarkan jarak terkecil.

5. Retrieval and Output

Kumpulkan gambar-gambar yang mirip dengan query masukan dengan cara melakukan limitasi jumlah atau dengan memberikan batas jarak euclidean. Hasil dari pencarian gambar dapat digabungkan dengan hasil pencarian suara ataupun dijalankan sendiri.

1.4 Music Information Retrieval - Query by Humming



Gambar 1.9 Tahapan pada MIR dengan metode Query By Humming

Pada query by humming akan dilakukan beberapa langkah utama. Pertama, dilakukan pemrosesan audio, di mana suara yang diinput direkam atau diterima dan dipersiapkan untuk tahap berikutnya. Selanjutnya, data audio tersebut melalui tahap ekstraksi fitur vektor, yaitu proses konversi data audio menjadi representasi numerik berupa vektor fitur yang dapat dianalisis. Terakhir, pada tahap pencarian similaritas vektor, vektor fitur yang dihasilkan dibandingkan dengan dataset untuk menemukan hasil yang paling sesuai atau diatas nilai kemiripan/similaritas yang telah ditentukan.

1. Pemrosesan Audio

Pemrosesan audio dalam sistem query by humming menggunakan file MIDI dengan fokus pada track melodi utama, umumnya di Channel 1. Setiap file MIDI diproses menggunakan metode windowing yang membagi melodi menjadi segmen 20-40 beat dengan sliding window 4-8 beat. Teknik ini memungkinkan pencocokan fleksibel dari berbagai bagian lagu yang mungkin diingat pengguna.

Proses windowing disertai normalisasi tempo dan pitch untuk mengurangi variasi humming. Setiap note event dikonversi menjadi representasi numerik yang mempertimbangkan durasi dan urutan nada, memungkinkan sistem membandingkan potongan melodi dengan database.

Berikut adalah formula untuk melakukan normalisasi tempo yang dibutuhkan :

$$NP(note) = \frac{(note - \mu)}{\sigma}$$

Gambar 1.10 Normalisasi Tempo

Dimana μ adalah rata rata dari pitch dan σ adalah standar deviasi dari pitch.

2. Ekstraksi Fitur

2.1. Distribusi tone

Distribusi tone diukur berdasarkan tiga viewpoints.

Fitur Absolute Tone Based (ATB) menghitung frekuensi kemunculan setiap nada berdasarkan skala MIDI (0-127). Histogram yang dihasilkan memberikan gambaran distribusi absolut nada dalam data. Hal ini penting untuk menangkap karakteristik statis melodi dalam sinyal audio. Langkah pertama adalah membuat histogram dengan 128 bin, sesuai dengan rentang nada MIDI dari 0 hingga 127. Kemudian, hitung frekuensi kemunculan masing-masing nada MIDI dalam data. Setelah itu, normalisasi histogram untuk mendapatkan distribusi yang terstandarisasi.

Fitur Relative Tone Based (RTB) menganalisis perubahan antara nada yang berurutan, menghasilkan histogram dengan nilai dari -127 hingga +127. RTB berguna untuk memahami pola interval melodi, yang lebih relevan dalam mencocokkan humming dengan dataset yang tidak bergantung pada pitch absolut. Dimulai dengan membangun histogram yang memiliki 255 bin dengan rentang nilai dari -127 hingga +127. Selanjutnya, hitung selisih antara nada-nada yang berurutan dalam data. Terakhir, lakukan normalisasi pada histogram yang telah dibuat.

Fitur First Tone Based (FTB) fokus pada perbedaan antara setiap nada dengan nada pertama, menciptakan histogram yang mencerminkan hubungan relatif terhadap titik referensi awal. Pendekatan ini membantu menangkap struktur relatif nada yang lebih stabil terhadap variasi pitch pengguna. Histogram dibuat

dengan 255 bin, juga mencakup rentang nilai dari -127 hingga +127. Kemudian, hitung selisih antara setiap nada dalam data dengan nada pertama. Histogram yang dihasilkan kemudian dinormalisasi untuk menghasilkan distribusi yang seimbang.

2.2. Normalisasi

Normalisasi memastikan bahwa semua nilai dalam histogram berada dalam skala probabilitas. Berikut adalah formula umum dari normalisasi yang digunakan:

$$H_{norm} = \frac{H[d]}{\sum_d H[d]}$$

Gambar 1.11 Normalisasi Histogram

Dimana H adalah Histogram dan d adalah bin dari histogram tersebut.

3. Penghitungan Similaritas

Ubah setiap histogram menjadi sebuah vektor dan lakukan perhitungan kemiripannya menggunakan cosine similarity. Pada jurnal terkait, metode yang digunakan adalah euclidean distance, tetapi pada tugas kali ini metode perhitungan similaritas yang akan digunakan adalah cosine similarity. Cosine Similarity adalah ukuran untuk menentukan seberapa mirip dua vektor dalam ruang berdimensi tinggi, dengan menghitung sudut cosinus di antara keduanya. Semakin kecil sudutnya (semakin dekat ke 1 hasilnya), semakin mirip kedua vektor tersebut. Sehingga cosine similarity bisa dijadikan salah satu metode lain dalam perhitungan similaritas. Silahkan lakukan eksplorasi eksperimen dengan pembobotan berbeda untuk setiap fitur dan tentukan bobot terbaiknya.

Berikut adalah formula dari cosine similarity :

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Gambar 1.12 Cosine Similarity Formula

1.5 Arsitektur Website

Untuk tugas besar kali ini, Anda akan diminta untuk membuat website. Oleh karena itu, kami akan memberikan gambaran singkat terkait arsitektur website yang akan dibuat.

1. *Audio query*, berisi file audio yang akan digunakan untuk melakukan pencocokan suara
2. *Voice recording module*, tombol untuk memproseskan suara dari microphone pengguna dan disimpan untuk diproses
3. Kumpulan suara yang disimpan dalam bentuk dataset, bisa dilakukan dengan cara mengunggah *multiple audio*, folder, .zip, .rar, dan file kompresi lainnya ke dalam website anda. Setelah memasukkan kumpulan suara tersebut, diharapkan semua audio yang ada dalam dataset dimunculkan di halaman website. Agar pengguna tidak melakukan scrolling yang berlebihan, silahkan gunakan *pagination*.
4. Bila mengerjakan bonus PCA, anda harus memetakan file lagu ke file gambar yang bersangkutan, gambar tersebut merepresentasikan album dari lagu tersebut. Pemetaan dapat disimpan dalam file .txt dan .json.



Gambar 1.13 Contoh tampak depan dari website Information Retrieval (Humming)



Gambar 1.14 Contoh tampak depan dari website Information Retrieval (Album)

Anda tidak harus membuat website dengan tampak depan yang sama persis dengan gambar diatas. Namun, yang terpenting adalah komponen ada gambar diimplementasikan semua dalam website yang anda buat. Secara garis besar, berikut adalah komponen-komponen yang perlu anda buat dalam website pada tugas besar ini.

1. Judul Website
2. Tombol untuk mengunggah audio ketika dalam moda query audio by humming dan mengunggah gambar ketika dalam moda query album finder with PCA
3. Tombol untuk merekam dan mengunggah rekaman audio tersebut (bonus)
4. Tombol untuk mengunggah dataset gambar wajah dan audio
5. Tombol untuk memilih antara query audio by humming dan album finder by PCA
6. Tombol untuk mengunggah mapper
7. Daftar lagu beserta album gambar yang bersangkutan yang ada di dataset
8. Pagination agar lagu yang muncul tidak banyak. Jumlah lagu dalam satu page dibebaskan asalkan tidak membuat pengguna harus scrolling terlalu lama
9. Tombol search untuk melakukan pencarian
10. Persentase kemiripan pada setiap audio atau album yang ada pada dataset
11. Waktu dari eksekusi program (disarankan dalam waktu ms)

Secara umum, berikut adalah cara-cara dari bagaimana sebuah website Information Retrieval bisa digunakan.

1. Pengguna terlebih dahulu mengunggah dataset pada tombol audio dan picture yang tertera di website.
2. Pengguna mengunggah pemetaan file lagu dengan judulnya dan gambar album yang bersangkutan (Bersifat opsional dan bonus pada bagian judul)
3. Pengguna melakukan query dengan dua cara yaitu mengunggah file audio atau gambar
4. Pengguna memilih akan melakukan pencarian dengan metode album finder with PCA atau query by humming
5. Program akan menampilkan kumpulan audio atau gambar yang mirip, diurutkan dari yang memiliki kemiripan paling tinggi ke yang paling rendah. Setiap audio atau gambar yang muncul diberi persentase kemiripannya.
6. Terdapat informasi terkait jumlah audio atau gambar yang muncul, dan waktu eksekusi programnya.

BAB II

TEORI SINGKAT

2.1 Sistem Temu Balik Suara (MIR)

Music Information Retrieval (MIR) adalah cabang ilmu multidisiplin yang menggabungkan ilmu musik, ilmu komputer, dan ilmu sains data untuk mengembangkan metode atau teknik yang memungkinkan pengambilan, analisis, dan pengelolaan informasi terkait sebuah audio. Tujuan utama MIR adalah untuk memungkinkan komputer memahami dan memproses musik secara otomatis mirip seperti computer vision yang membuat komputer bisa memproses objek grafis.

Music Information Retrieval (MIR) memiliki berbagai aplikasi yang signifikan dalam industri musik dan teknologi. Salah satu aplikasi utamanya adalah sistem rekomendasi musik, yang memungkinkan platform streaming seperti Spotify dan Apple Music untuk menyarankan lagu atau artis berdasarkan preferensi dan pola mendengarkan pengguna. Selain itu, MIR digunakan dalam teknologi pengenalan lagu seperti Shazam, yang dapat mengidentifikasi sebuah lagu hanya dari cuplikan audio singkat.

Secara keseluruhan, Music Information Retrieval (MIR) memainkan peran penting dalam mengubah cara kita berinteraksi dengan musik dan mengelola data musik secara digital. Melalui kontribusinya yang multidimensi, MIR terus mengembangkan dunia teknologi musik untuk masa depan.

2.2 Metode Ekstraksi Fitur Berdasarkan Humming

Pada prosesnya, sebuah kumpulan file audio akan diekstrak fiturnya terlebih dahulu seperti Absolute Tone, Relative Tone, First Tone nya kemudian fitur-fitur ini akan dikaitkan dengan nama lagu/file audio tersebut dan disimpan dalam sebuah database. Database ini akan digunakan sebagai data acuan dalam mencari lagu yang karakteristiknya mirip dengan query.

Pada program ini, ekstraksi berfokus pada track dan channel yang mengandung melodi utama. Umumnya melodi utama berada pada track 1 channel 0, namun apabila audio menyimpan melodi utama di track/channel lain maka program akan mencari track/channel yang sekiranya dijadikan sebagai melodi utama pada audio tersebut dengan cara melakukan pembobotan terhadap setiap track dan channel kemudian dipilih skor tertinggi.

Ekstraksi menggunakan teknik sliding window, dengan panjang window 30 beat dan sliding tiap 6 beat. Pemilihan teknik ini supaya proses pencocokan lebih fleksibel sebab durasi query tidak harus spesifik namun tetap dapat memberikan hasil yang cukup akurat dan performa yang masih dapat diterima. Apabila terdapat not yang berjalan secara bersamaan, maka program hanya akan meninjau not tertinggi dalam interval tersebut untuk menjaga akurasi data fitur. Tidak lupa setiap fitur dilakukan normalisasi supaya tidak ada outlier seperti distorsi sehingga data yang didapatkan lebih reliable. Rumus normalisasi yang digunakan adalah;

$$H_{norm} = \frac{H[d]}{\sum_d H[d]}$$

Kemudian akan dimasukkan query sebuah audio yang akan dicari kemiripannya dengan audio yang telah diekstraksi dan disimpan dalam database. Query audio ini juga akan melewati proses ekstraksi fitur-fitur untuk mendapatkan karakteristiknya. Program ini menggunakan metode cosine similarity untuk menghitung nilai kesamaan antara audio dalam database dengan query. Nilai cosine similarity merupakan representasi sudut antara dua vektor berdimensi tinggi, dalam hal ini vektor berdimensi tinggi adalah fitur dalam bentuk histogram. Semakin kecil sudut antara/nilai cos mendekati 1, maka kedua audio tersebut semakin mirip karakteristiknya. Metode cosine similarity dihitung dengan rumus;

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

2.3 Image Retrieval dengan PCA

Image retrieval menggunakan Principal Component Analysis (PCA) adalah metode pencarian gambar berdasarkan kemiripan visual dengan cara mengoptimalkan representasi data melalui reduksi dimensi. Teknik ini memungkinkan kita untuk menangkap pola-pola utama dalam dataset gambar, tanpa memproses semua detail yang kurang relevan. Prosesnya dimulai dengan mengubah setiap gambar menjadi format grayscale untuk menyederhanakan informasi visual menjadi intensitas cahaya.

Gambar-gambar ini kemudian distandarkan dalam ukuran yang seragam dan diubah menjadi vektor satu dimensi untuk mempermudah analisis matematis.

$$I(x,y) = 0.2989 \cdot R(x,y) + 0.5870 \cdot G(x,y) + 0.1140 \cdot B(x,y)$$

$$I = [I_1, I_2, \dots, I_{M \cdot N}]$$

PCA bekerja dengan memusatkan data gambar di sekitar nol melalui proses standarisasi, di mana setiap nilai piksel dikurangi dengan rata-ratanya. Selanjutnya, PCA menggunakan pendekatan matematika seperti Singular Value Decomposition (SVD) untuk menghasilkan komponen-komponen utama, yaitu dimensi baru yang mencerminkan variasi terbesar dalam dataset. Hanya beberapa komponen utama teratas yang dipilih, karena komponen ini merepresentasikan karakteristik paling signifikan dari gambar.

$$C = \frac{1}{N} X'^T X'$$

$$C = U \Sigma U^T$$

Setelah ruang dimensi rendah ini terbentuk, gambar query juga diproyeksikan ke ruang yang sama, sehingga memudahkan perbandingan dengan gambar-gambar lain. Kemiripan antara gambar query dan dataset dihitung menggunakan jarak Euclidean, di mana gambar dengan jarak terkecil dianggap paling mirip. Hasilnya adalah daftar gambar yang diurutkan berdasarkan tingkat kesamaannya dengan query. Pendekatan ini tidak hanya efisien secara komputasi tetapi juga mampu menjaga informasi penting dari dataset, sehingga sangat cocok untuk aplikasi yang membutuhkan pengolahan data visual dalam skala besar.

$$d(\mathbf{q}, \mathbf{z}_i) = \sqrt{\sum_{j=1}^k (q_j - z_{ij})^2}$$

BAB III

ARSITEKTUR FRONTEND DAN BACKEND

3.1 Frontend

Arsitektur umum website

3.1.1. Judul Website

- **Deskripsi:** Bagian atas website menampilkan judul utama untuk memberikan identitas aplikasi kepada pengguna.
- **Fungsi:** Menyampaikan tujuan aplikasi dengan jelas kepada pengguna.

3.1.2. Komponen Pencarian dan Pengunggahan Data

3.1.2.1. Audio Query (File Audio untuk Pencocokan Suara)

- **Deskripsi:** Komponen ini memungkinkan pengguna untuk mengunggah file audio yang akan digunakan untuk mencocokkan suara dengan dataset yang ada.
- **Fungsi:**
 - Menjalankan pencarian berbasis suara.
 - File audio akan diproses oleh backend untuk mencari file audio dari database yang mirip.

3.1.2.2. Image Query (File Image Pencocokan Gambar)

- **Deskripsi:** Komponen ini memungkinkan pengguna untuk mengunggah file image yang akan digunakan untuk mencocokkan gambar dengan dataset yang ada.
- **Fungsi:**
 - Menjalankan pencarian berbasis gambar.
 - File image akan diproses oleh backend untuk mencari file image dari datase yang mirip.

3.1.2.3. Pengunggahan Dataset Audio

- **Deskripsi:** Tombol untuk mengunggah kumpulan suara dalam berbagai format seperti multiple audio files, folder, atau file kompresi (.zip, .rar).
- **Fungsi:**
 - Mengelola dataset audio yang digunakan oleh sistem.
 - Secara otomatis menampilkan daftar audio yang diunggah di halaman website.

3.1.2.4. Pengunggahan Dataset Gambar

- **Deskripsi:** Tombol untuk mengunggah kumpulan gambar album yang digunakan dalam mode **Album Finder by PCA**.
- **Fungsi:**
 - Menambahkan dataset gambar untuk pencarian album berdasarkan PCA.

3.1.2.5. Pengunggahan Mapper

- **Deskripsi:** Tombol untuk mengunggah file mapper (mapper.json).
- **Fungsi:**
 - Menyelaraskan hubungan antara file audio dan gambar album sehingga sistem dapat menampilkan hasil pencarian yang terhubung.

3.1.3. Pemilihan Mode Pencarian

- **Deskripsi:** Tombol ini memberikan opsi kepada pengguna untuk memilih mode pencarian:
 1. **Query Audio by Humming:** Pencarian berbasis suara.
 2. **Album Finder by PCA:** Pencarian berbasis gambar album menggunakan analisis PCA.
- **Fungsi:** Mempermudah pengguna untuk beralih antar mode sesuai kebutuhan.

3.1.4. Dataset Audio dan Gambar Album

- **Deskripsi:** Berisi daftar lagu dan gambar album yang tersedia di dataset.
- **Fungsi:** Sebagai data yang dibandingkan dengan query audio dan image.

3.1.5. Pagination

- **Deskripsi:** Pagination digunakan untuk membatasi jumlah elemen (lagu/audio) yang ditampilkan pada satu halaman.
- **Fungsi:**
 - Mengurangi kepadatan halaman.
 - Menampilkan maksimal **12 item per halaman** untuk menampilkan database lagu/gambar album, dan **8 item per halaman** untuk menampilkan hasil retrieval.

3.1.6. Hasil Pencarian

- **Deskripsi:** Setelah pengguna mengunggah file audio atau gambar album untuk pencarian, hasilnya ditampilkan dalam bentuk pagination.
- **Fungsi:**
 - a. Menampilkan daftar lagu yang cocok dengan gambar album terkait (untuk mode PCA) beserta similarity dan execution time.
 - b. Menampilkan file audio yang relevan berdasarkan pencarian humming beserta gambar album, nilai similarity, dan execution time.

3.2 Backend

Struktur folder backend adalah sebagai berikut:

```

├── backend/
│   ├── main.py          # FastAPI endpoints
│   ├── process.py       # Mathematical functions for image and audio retrieval
│   ├── datasetimage/    # Store image database
│   ├── datasetaudio/    # Store audio database
│   ├── mapper.json      # Store mapper
│   ├── result.json      # Store result after computation
│   └── .gitignore       # Ignore some files or folders inside backend

```

3.2.1. main.py

Program main pada backend ini berisikan endpoints dari FastAPI. File ini berisikan fungsi-fungsi yang dijalankan setelah menerima request dari server API. Fungsi-fungsi tersebut terdiri upload_image, upload_audio, search_image,

search_audio, dan upload. upload_image dan upload_audio menghandle image query dan audio query yang diupload oleh user. Upload menghandle file .zip maupun .rar dari audio/image dataset serta .json untuk mapper. Search_audio dan search_image memanggil fungsi dari process.py untuk menghitung similiaritas query dengan dataset.

3.2.2. process.py

Program ini berisikan perhitungan matematis yang diperlukan untuk mencari kemiripan antara file query dengan file yang terdapat di dataset. Perhitungan tersebut berisikan perhitungan MIR untuk Audio Retrieval dan perhitungan PCA dengan Singular Value Decomposition untuk Image Retrieval.

3.2.3. Flow Query

3.2.3.1. Upload

Ketika user menekan tombol upload, website akan memberi request kepada server API untuk mengirimkan file yang diupload ke bagian backend. Ketika backend telah menerima file yang diberikan, file tersebut akan dibaca dan disimpan pada datasetaudio untuk file zip berisi audio, datasetimage untuk file zip image. Untuk query, backend akan menyimpan file query gambar menjadi query_image dan audio menjadi query_audio dalam bentuk file midi. Jika diterima file wav, maka akan diconvert menjadi file midi.

3.2.3.2. Search

Ketika user menekan tombol search, website akan memberi request pada server API untuk menjalankan fungsi search_image ataupun search_audio. Kedua fungsi ini adalah fungsi yang berisi perhitungan matematis. Setelah fungsi tersebut dijalankan, hasil dari pencariannya adalah array of objek dengan atribut 'nama', 'sim' sebagai kemiripan dan di akhir adalah objek executiontime.

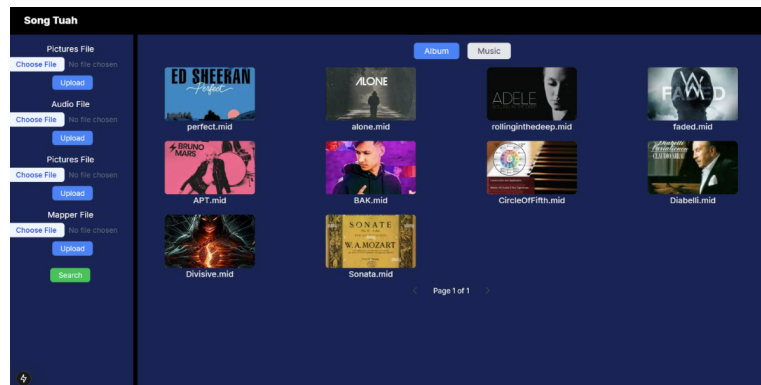
3.2.3.3. Fetch Data

Untuk menampilkan data yang dimiliki dari backend, website harus memberi request untuk mendapatkan data dari datasetimage, datasetaudio, dan juga data mapper.json. Website akan melakukan fetch data secara dinamik dari backend untuk menjadi objek yang ditampilkan di User Interface.

BAB IV EKSPERIMEN

4.1 Eksperimen 1

Pada eksperimen ini kami menggunakan dataset sebesar 10 mb untuk audio yang terdiri atas 5 lagu .mid dan 5 lagu .wav. Untuk ukuran dataset gambar kami yakni sebesar 1.36 mb yang terdiri atas 10 file .jpg. Tidak lupa, kami juga membuat file mappernya untuk mengaitkan file audio dengan gambar.

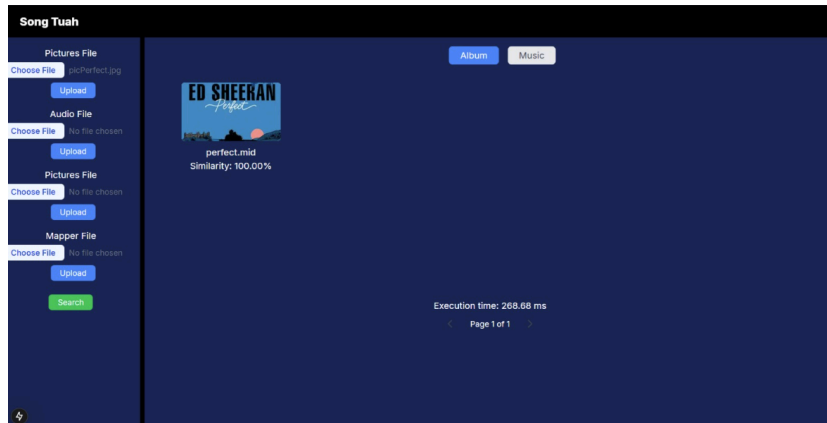


Gambar 4.1 Dataset Gambar dan Audio Berhasil Upload Eksperimen 1

Program dapat melakukan fungsi image retrieval dengan menerima masukan berupa gambar yang nantinya akan dicocokkan dengan dataset gambar yang sudah ada sebelumnya. Kemudian program akan menampilkan gambar yang kiranya memiliki tingkat kesamaan yang tinggi serta juga menampilkan lama waktu eksekusi pencarian tersebut.

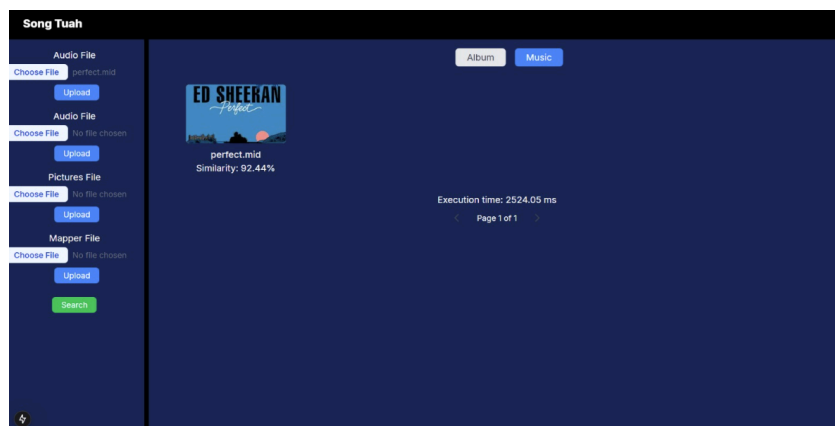


Gambar 4.2 Gambar Query Eksperimen 1



Gambar 4.3 Hasil Pencarian Image Retrieval Eksperimen 1

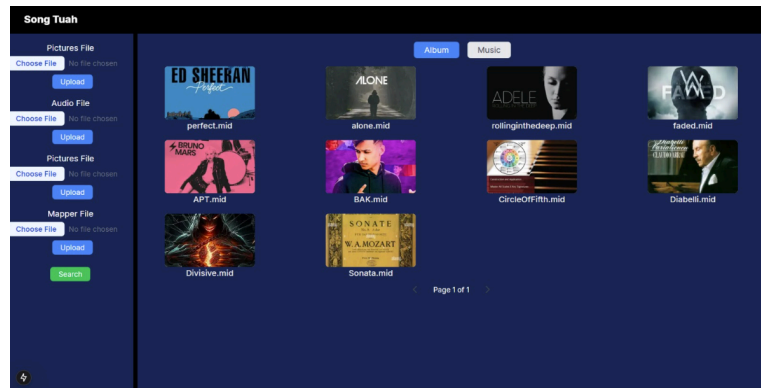
Program juga dapat melakukan fungsi music retrieval dengan menerima masukan berupa file audio (.mid/.wav) yang nantinya akan dicocokkan dengan dataset audio yang sudah ada sebelumnya. Kemudian program akan menampilkan audio yang kiranya memiliki tingkat kesamaan yang tinggi serta juga menampilkan lama waktu eksekusi pencarian tersebut. Pada eksperimen ini kami menggunakan query audio “perfect.mid”.



Gambar 4.4 Hasil Pencarian Music Retrieval Eksperimen 1

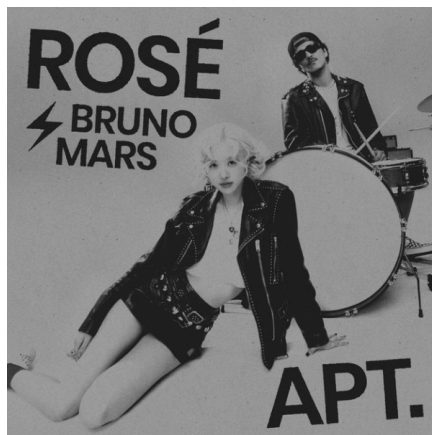
4.2 Eksperimen 2

Pada eksperimen ini kami menggunakan dataset sebesar 10 mb untuk audio yang terdiri atas 5 lagu .mid dan 5 lagu .wav. Untuk ukuran dataset gambar kami yakni sebesar 1.36 mb yang terdiri atas 10 file .jpg. Tidak lupa, kami juga membuat file mappernya untuk mengaitkan file audio dengan gambar.

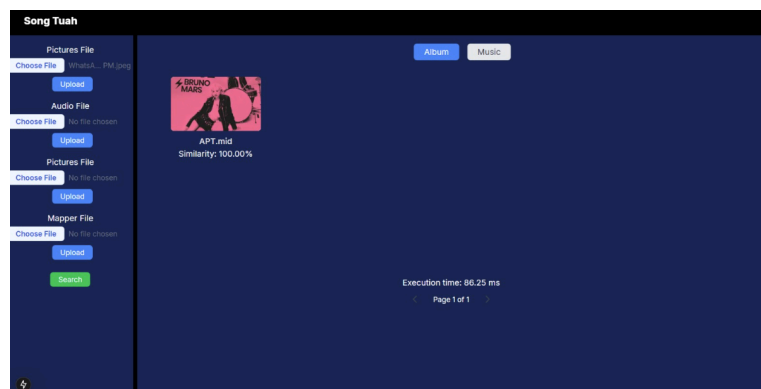


Gambar 4.5 Dataset Gambar dan Audio Berhasil Upload Eksperimen 2

Program dapat melakukan fungsi image retrieval dengan menerima masukan berupa gambar yang nantinya akan dicocokkan dengan dataset gambar yang sudah ada sebelumnya. Kemudian program akan menampilkan gambar yang kiranya memiliki tingkat kesamaan yang tinggi serta juga menampilkan lama waktu eksekusi pencarian tersebut.

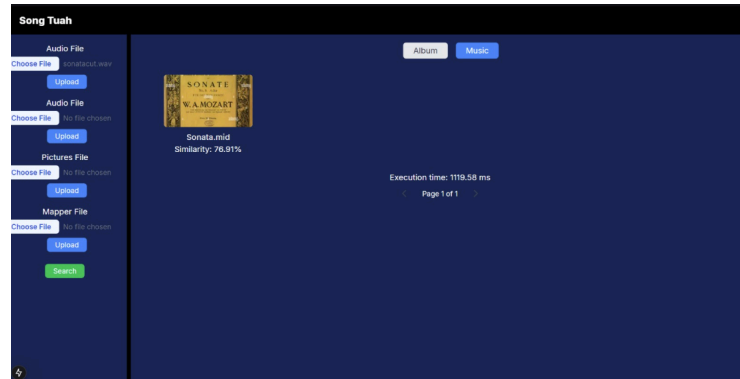


Gambar 4.6 Gambar Query (Greyscale) Eksperimen 2



Gambar 4.7 Hasil Pencarian Image Retrieval Eksperimen 2

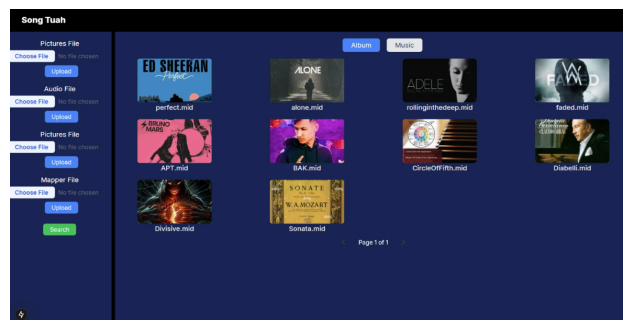
Program juga dapat melakukan fungsi music retrieval dengan menerima masukan berupa file audio (.mid/.wav) yang nantinya akan dicocokkan dengan dataset audio yang sudah ada sebelumnya. Kemudian program akan menampilkan audio yang kiranya memiliki tingkat kesamaan yang tinggi serta juga menampilkan lama waktu eksekusi pencarian tersebut. Pada eksperimen ini kami menggunakan query audio “sonatacut.wav” yang merupakan potongan dari “Sonata.wav” berdurasi 5 detik pada bagian tengah .



Gambar 4.8 Hasil Pencarian Music Retrieval Eksperimen 2

4.3 Eksperimen 3

Pada eksperimen ini kami menggunakan dataset sebesar 10 mb untuk audio yang terdiri atas 5 lagu .mid dan 5 lagu .wav. Untuk ukuran dataset gambar kami yakni sebesar 1.36 mb yang terdiri atas 10 file .jpg. Tidak lupa, kami juga membuat file mappernya untuk mengaitkan file audio dengan gambar.

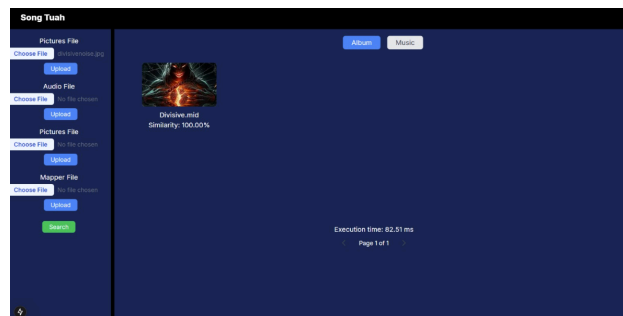


Gambar 4.9 Dataset Gambar dan Audio Berhasil Upload Eksperimen 3

Program dapat melakukan fungsi image retrieval dengan menerima masukan berupa gambar yang nantinya akan dicocokkan dengan dataset gambar yang sudah ada sebelumnya. Kemudian program akan menampilkan gambar yang kiranya memiliki tingkat kesamaan yang tinggi serta juga menampilkan lama waktu eksekusi pencarian tersebut.

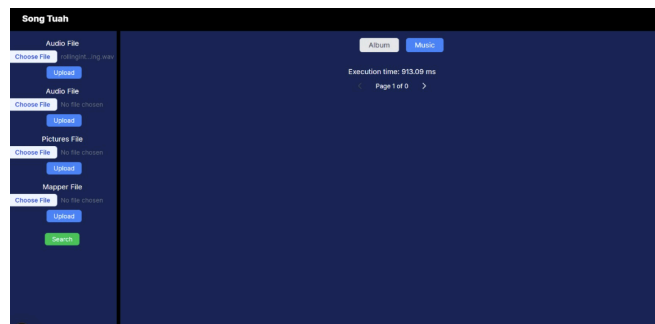


Gambar 4.10 Gambar Query (Noise) Eksperimen 3



Gambar 4.11 Hasil Pencarian Image Retrieval Eksperimen 3

Program juga dapat melakukan fungsi music retrieval dengan menerima masukan berupa file audio (.mid/.wav) yang nantinya akan dicocokkan dengan dataset audio yang sudah ada sebelumnya. Kemudian program akan menampilkan audio yang kiranya memiliki tingkat kesamaan yang tinggi serta juga menampilkan lama waktu eksekusi pencarian tersebut. Pada eksperimen ini kami menggunakan query audio “rollinginthedeep-humming.wav” yang merupakan humming dari “rollinginthedeep.mid” berdurasi 18 detik. Namun tidak ada hasil kesamaan, hal ini dapat disebabkan oleh beberapa faktor seperti noise, miss konversi wav to mid, atau bahkan humming yang tidak mirip. Bisa juga karena program dibatasi hanya menampilkan lagu dengan tingkat kesamaan lebih dari 70%.



Gambar 4.12 Hasil Pencarian Music Retrieval Eksperimen 3

BAB V

KESIMPULAN

5.1 Kesimpulan

Tugas besar ini merupakan implementasi dari materi perkuliahan IF2123 Aljabar Linier dan Geometri yang diterapkan dalam sebuah program/website. Program ini mengimplementasikan beberapa materi dasar yang berkaitan dengan perkuliahan seperti Vektor dan Singular Value Decomposition (SVD) untuk penghitungan similarity serta fitur Image Retrieval.

Secara keseluruhan, tugas ini merupakan puncak dari pembelajaran yang telah dilakukan selama perkuliahan, menggabungkan berbagai konsep teoritis dan teknik pemrograman menjadi suatu hal yang nyata dan dapat digunakan secara riil, sekaligus memberikan pemahaman yang lebih mendalam tentang penerapan materi telah diajarkan dalam pemecahan masalah di dunia nyata.

5.2 Saran

Saran untuk pengembangan selanjutnya antara lain,

- a. Optimalisasi proses ekstraksi fitur dan pencarian similarity pada Music Information Retrieval agar lebih cepat dan akurat
- b. Perbaiki User Interface
- c. Keakuratan image retrieval ditingkatkan
- d. Fetch data yang lebih dinamik.

5.3 Komentar

Tugas besar ini memberikan kami pengetahuan tidak hanya seputar materi perkuliahan, tetapi juga di luar perkuliahan seperti pembuatan website dan pengetahuan seputar pemrosesan audio dan video yang selama ini dilakukan oleh komputer. Setelah mengerjakan tugas besar ini, kami merasakan kemampuan dan pengetahuan kami bertambah dari yang sebelumnya.

5.4 Refleksi

Tugas besar ini memberikan pengalaman berharga mengenai manajemen waktu dan bekerja sama dalam tim. Kami dilatih untuk bisa membagi waktu, berkomunikasi efektif, serta meningkatkan disiplin membaca dan mencari tahu mengenai suatu hal. Kami juga memperoleh pemahaman mendalam tentang bagaimana materi perkuliahan diimplementasikan di dunia nyata.

LAMPIRAN

Referensi

R. Munir, “Ruang Vektor Umum (bagian 1),” IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-15-Ruang-vektor-umum-Bagian1-2023.pdf>. [Diakses: 13 Desember 2024].

R. Munir, “Ruang Vektor Umum (bagian 2),” IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-16-Ruang-vektor-umum-Bagian2-2023.pdf>. [Diakses: 13 Desember 2024].

R. Munir, “Ruang Vektor Umum (bagian 3),” IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-17-Ruang-vektor-umum-Bagian3-2023.pdf>. [Diakses: 13 Desember 2024].

R. Munir, “Ruang Vektor Umum (bagian 4),” IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-18-Ruang-vektor-umum-Bagian4-2023.pdf>. [Diakses: 13 Desember 2024].

R. Munir, “Singular Value Decomposition (SVD) (Bagian 1),” IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf>. [Diakses: 13 Desember 2024].

R. Munir, “Singular Value Decomposition (SVD) (Bagian 1),” IF2123Aljabar Linier dan Geometri, 2023. [Online]. Tersedia: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-22-Singular-value-decomposition-Bagian2-2023.pdf>. [Diakses: 13 Desember 2024].

Repository

<https://github.com/Henshou/Algeo02-23025>