

# **Laporan Tugas Besar 2**

## **IF2123 Aljabar Linear dan Geometri 2024/2025**



Disusun oleh:

Kelompok 10 - Bos Pablo

Varel Tiara	(13523008)
Yonatan Edward Njoto	(13523036)
Benedict Presley	(13523067)

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

2024

## DAFTAR PUSTAKA

<b>DAFTAR PUSTAKA</b>	<b>2</b>
<b>DAFTAR GAMBAR</b>	<b>3</b>
<b>BAB I</b>	<b>5</b>
<b>DESKRIPSI MASALAH</b>	<b>5</b>
1.1. Pemrosesan Suara dan Gambar	5
1.2. Information Retrieval	6
<b>BAB II</b>	<b>16</b>
<b>TEORI SINGKAT</b>	<b>16</b>
2.1 Sistem Temu Balik Suara (MIR)	16
2.2 Metode Ekstraksi Fitur Berdasarkan Humming	17
2.3 Image Retrieval dengan PCA	18
<b>BAB III</b>	<b>19</b>
<b>ARSITEKTUR WEBSITE (FRONTEND) DAN ARSITEKTUR PROGRAM INFORMATION RETRIEVAL (BACKEND)</b>	<b>19</b>
3.1 Arsitektur Website (FrontEnd)	19
3.2.1 Home	19
3.2.2 Albums	19
3.2.3 Songs	22
3.2 Arsitektur Program Information Retrieval (BackEnd)	23
3.2.1 Go	23
3.2.2 Server Midi Convert	23
<b>BAB IV</b>	<b>25</b>
<b>EKSPERIMEN</b>	<b>25</b>
4.1 Pemrosesan Gambar	25
4.2 Pemrosesan Suara	26
<b>BAB V</b>	<b>28</b>
<b>KESIMPULAN, SARAN, KOMENTAR, DAN REFLEKSI</b>	<b>28</b>
5.1 Kesimpulan	28
5.2 Saran	28
5.3 Komentar	29
5.4 Refleksi	29
<b>BAB VI</b>	<b>30</b>
<b>LAMPIRAN</b>	<b>30</b>
6.1 Daftar Referensi	30

6.2 Tautan Repository	30
6.3 Tautan Video	31

## **DAFTAR GAMBAR**

Gambar 1. Shazam sebagai aplikasi audio retrieval system	5
Gambar 2. Tahapan pada MIR dengan metode Query By Humming	12
Gambar 3. Cosine Similarity Formula	15
Gambar 4. Arsitektur Website Home	19
Gambar 5. Arsitektur Website Albums Halaman 1	20
Gambar 6. Arsitektur Website Albums Halaman 2	20
Gambar 7. Arsitektur Website Albums Detail Belum Terhubung dengan Song	21
Gambar 8. Arsitektur Website Albums Detail Sudah Terhubung dengan Song	21
Gambar 9. Arsitektur Website Songs	22
Gambar 10. Go	23
Gambar 11. Python	24
Gambar 12. Pemrosesan Gambar Ditemukan	25
Gambar 13. Pemrosesan Gambar Tidak Ditemukan	26
Gambar 14. Pemrosesan Suara Lagu Ditemukan dengan Metode Upload	27
Gambar 15. Pemrosesan Suara Lagu Tidak Ditemukan dengan Metode Upload	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 pendeksi 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.** 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, dan lain-lain.

Matriks adalah salah satu komponen yang penting dalam aplikasi aljabar vektor. Di dalam Tugas Besar 2 ini, kami 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, kami menggunakan aljabar vektor untuk mencari perbandingan antar satu audio dengan audio yang lain. Kami menggunakan konsep yang bernama *Music Information Retrieval* atau MIR untuk mencari dan mengidentifikasi suara berdasarkan fitur-fitur yang dimilikinya. Tidak hanya itu, kami juga akan menggunakan konsep Principal Component Analysis (PCA) untuk mencari kumpulan audio melalui deteksi wajah berbagai orang.

## 1.2. Information Retrieval

*Information Retrieval* adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Pada tugas besar ini, anda akan berkutik 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, kami mengimplementasikan *Image Retrieval* dengan menggunakan Principal Component Analysis dan *Music Information Retrieval* dengan menggunakan humming.

### 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.

```
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.

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

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.

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

## 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:

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

di mana:

- $X'$ : matriks data yang sudah distandarisasi.

Lalu lakukan dekomposisi nilai singular untuk mendapatkan komponen utama:

$$\mathbf{C} = \mathbf{U}\Sigma\mathbf{U}^T$$

- $\mathbf{U}$ : matriks eigenvector (komponen utama),
- $\Sigma$ : 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:

$$\mathbf{Z} = \mathbf{X}'\mathbf{U}_k$$

di mana:

- $\mathbf{U}_k$ : 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:

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

Dimana:

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

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

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

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

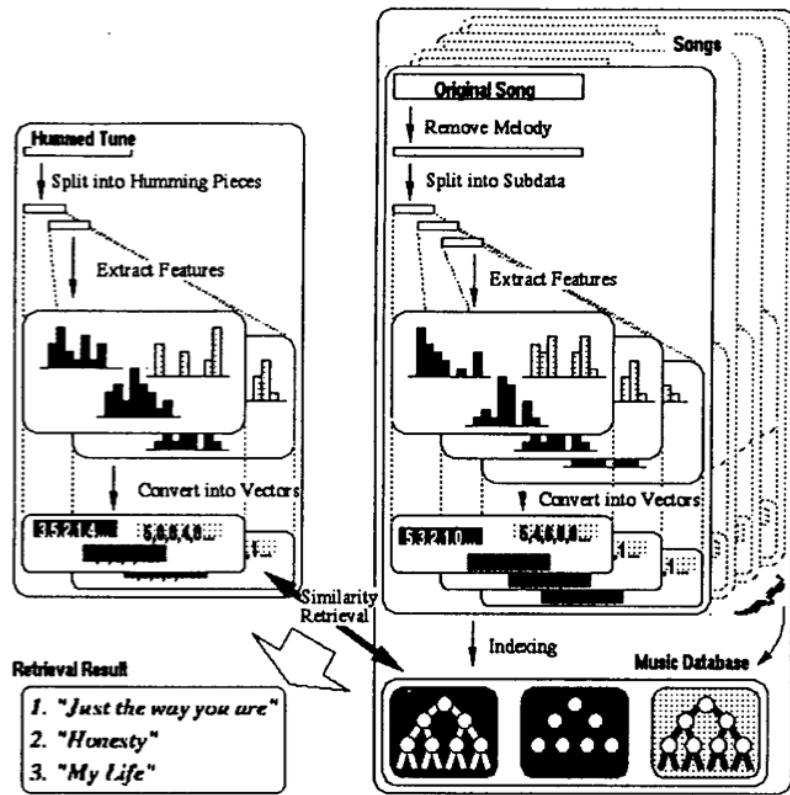
Lalu, Urutkan hasil berdasarkan jarak terkecil.

## 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.

## Music Information Retrieval - Query by Humming



**Gambar 2.** 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 :

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

Dimana  $\mu$  adalah rata rata dari pitch dan  $\sigma$  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:

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

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 3.** Cosine Similarity Formula

## **BAB II**

### **TEORI SINGKAT**

#### **2.1 Sistem Temu Balik Suara (MIR)**

Music Information Retrieval (MIR) adalah bidang interdisipliner yang menggabungkan teknologi audio dengan analisis data untuk mengakses, menganalisis, dan memanfaatkan informasi musik dari berbagai sumber data. Contoh populer dari sistem MIR adalah aplikasi seperti Shazam, yang mampu mengenali lagu hanya dari cuplikan audio. Dalam sistem MIR, algoritma digunakan untuk menganalisis fitur-fitur audio seperti melodi, ritme, dan harmoni. Salah satu pendekatan dalam MIR adalah **Query by Humming (QBH)**, yang memungkinkan pengguna mencari lagu berdasarkan penginputan melodi dalam bentuk humming. Proses ini terdiri dari tiga tahap utama:

1. **Pemrosesan Audio:** Menggunakan file MIDI dengan fokus pada track melodi utama, lalu setiap file MIDI akan diproses menggunakan metode windowing dan disertai dengan normalisasi tempo, dan pitch untuk mengurangi variasi humming. Setiap note event akan dikonversi menjadi representasi numerik yang mempertimbangkan durasi dan urutan nada, yang memungkinkan sistem membandingkan potongan melodi dengan database.
2. **Ekstraksi Fitur:**
  - a. Distribusi Tone  
Melibatkan distribusi nada dengan tiga pendekatan utama—Absolute Tone Based (ATB), Relative Tone Based (RTB), dan First Tone Based (FTB).
  - b. Normalisasi  
Normalisasi memastikan bahwa semua nilai dalam histogram berada dalam skala probabilitas.
3. **Penghitungan Similaritas:** Histogram diubah menjadi vektor, lalu dihitung tingkat kemiripannya menggunakan *cosine similarity*. Metode ini menentukan

kemiripan dua vektor dalam ruang berdimensi tinggi dengan menghitung sudut cosinus di antara keduanya. Semakin kecil sudutnya (nilai mendekati 1), semakin mirip vektor tersebut. Eksperimen dapat dilakukan dengan memberikan bobot berbeda pada setiap fitur untuk menemukan bobot terbaik.

## 2.2 Metode Ekstraksi Fitur Berdasarkan Humming

Ekstraksi fitur dalam sistem query by humming merupakan inti dari MIR. Tiga pendekatan utama yang digunakan meliputi:

1. Fitur Absolute Tone Based (ATB): Menghitung frekuensi kemunculan nada berdasarkan skala MIDI (0-127) menggunakan histogram dengan 128 bin. Fitur ini menangkap karakteristik statis melodi dalam data audio, di mana histogram distandarisasi melalui normalisasi.
2. Fitur Relative Tone Based (RTB): Menganalisis perubahan antara nada berurutan menggunakan histogram dengan 255 bin (rentang -127 hingga +127). Fitur ini menangkap pola interval melodi yang relevan untuk mencocokkan humming dengan dataset tanpa bergantung pada pitch absolut. Histogram dinormalisasi setelah perhitungan selisih nada.
3. Fitur First Tone Based (FTB): Memfokuskan pada hubungan setiap nada dengan nada pertama menggunakan histogram 255 bin (rentang -127 hingga +127). Fitur ini mencerminkan struktur melodi relatif terhadap titik referensi awal, membantu menstabilkan variasi pitch pengguna. Histogram hasil dihitung dan dinormalisasi untuk keseimbangan distribusi.

Penggunaan metode seperti cosine similarity membantu mempermudah pencocokan melodi antara input dan dataset. Dalam mencocokkan 2 buah lagu yang memiliki panjang berbeda, digunakan metode *sliding window* untuk memastikan banyaknya elemen pada interval yang diperiksa berjumlah sama.

## 2.3 Image Retrieval dengan PCA

Image Retrieval menggunakan Principal Component Analysis (PCA) untuk mencari gambar berdasarkan kemiripan fitur. PCA digunakan untuk mereduksi dimensi data gambar, dengan fokus pada elemen penting atau komponen utama yang mewakili informasi paling signifikan dalam dataset. Proses ini mengubah data berdimensi tinggi menjadi beberapa dimensi lebih kecil (*principal components*). Hasil PCA berupa eigenvector dan proyeksi data yang memudahkan perhitungan jarak antara gambar query dan dataset. Langkah-langkah utama dalam PCA meliputi:

1. **Image Processing and Loading:** Gambar diproses dengan mengubahnya ke grayscale untuk menyederhanakan analisis, lalu ukurannya diseragamkan. Selanjutnya, gambar grayscale dikonversi menjadi vektor 1D untuk mempermudah pemrosesan data. Standarisasi dilakukan untuk menempatkan data di sekitar nilai nol.
2. **Data Centering (Standardization):** Standarisasi dilakukan untuk menempatkan data di sekitar nilai nol
3. **PCA Computation Using Singular Value Decomposition (SVD):** Menggunakan metode Singular Value Decomposition (SVD) untuk mendapatkan eigenvalue dan eigenvector.
4. **Similarity Computation:** Gambar query direpresentasikan dalam ruang komponen utama (PCA) dengan proyeksi yang sama. Kemudian, jarak Euclidean dihitung antara gambar query dan setiap gambar dalam dataset untuk mengukur kesamaan. Hasil pencocokan diurutkan berdasarkan jarak terkecil, sehingga gambar yang paling mirip dengan query berada di urutan teratas.
5. **Retrieval and Output:** Mengumpulkan gambar-gambar yang mirip dengan query masukan dengan cara melakukan limitasi jumlah atau dengan memberikan batas jarak euclidean.

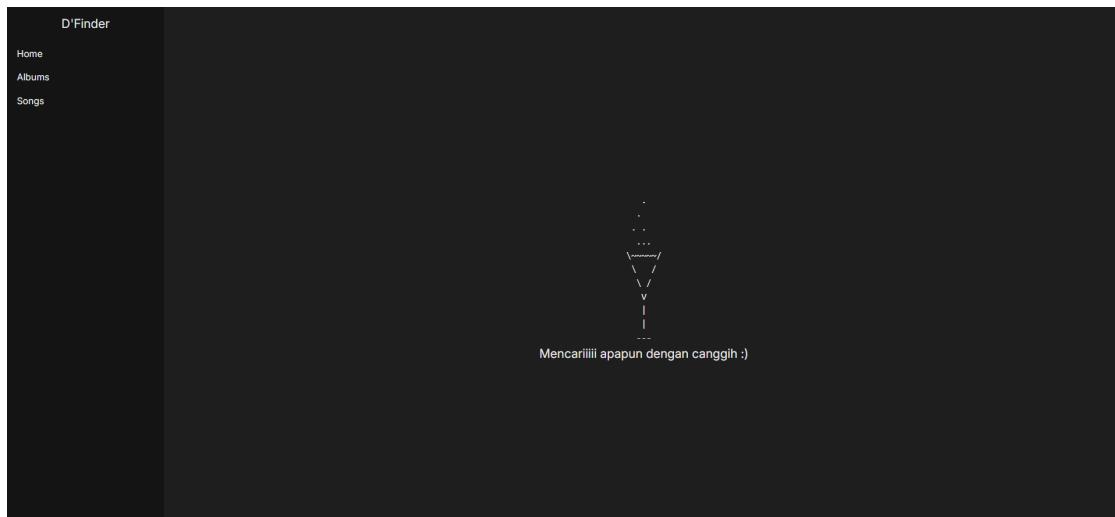
## BAB III

# ARSITEKTUR WEBSITE (FRONTEND) DAN ARSITEKTUR PROGRAM INFORMATION RETRIEVAL (BACKEND)

### 3.1 Arsitektur Website (FrontEnd)

Kami menggunakan remix pada front end, yang merupakan salah satu framework dari react. Kami menggunakan remix untuk memudahkan kami dalam mendesain dan juga interaksi dengan back end.

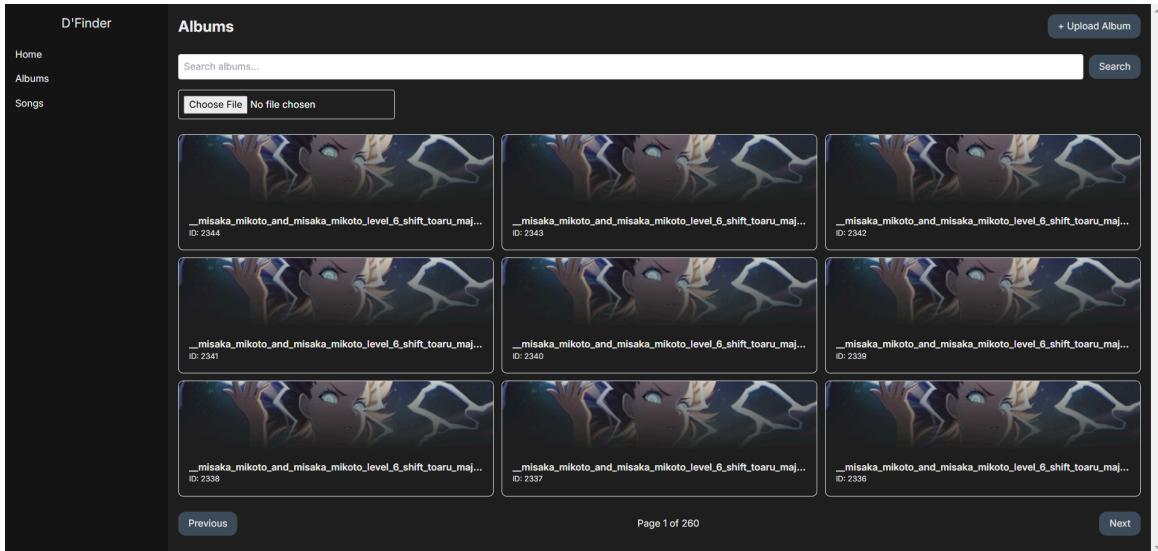
#### 3.2.1 Home



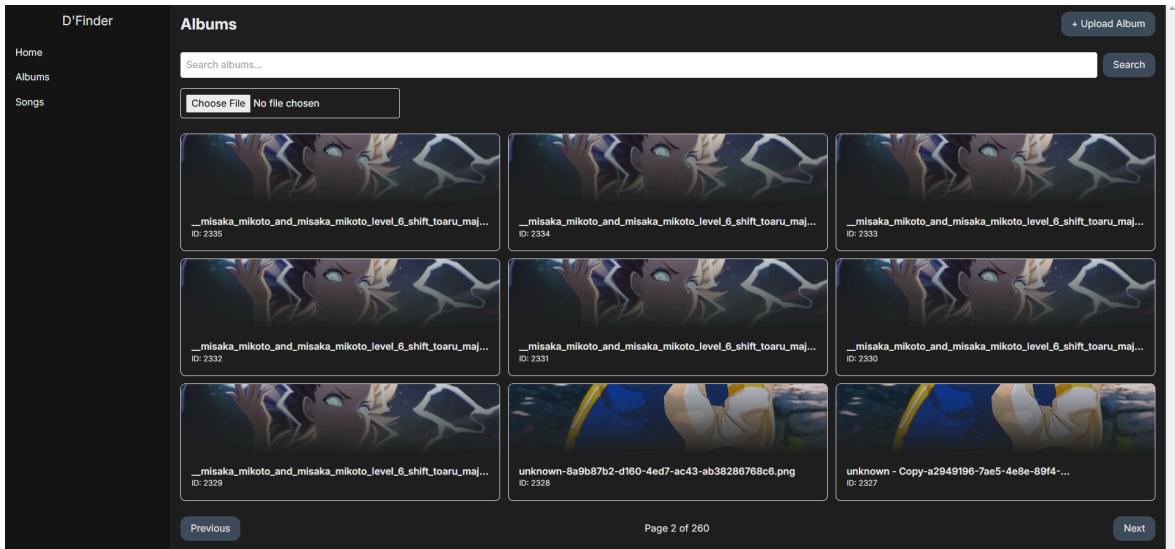
**Gambar 4.** Arsitektur Website Home

#### 3.2.2 Albums

Halaman **Albums** dari website **D'Finder** berfungsi untuk mengelola gambar dalam bentuk album. Terdapat kolom pencarian untuk mencari album, tombol **Upload Album** untuk menambahkan album baru, dan fitur **Choose File** untuk mencocokkan gambar dengan database. Daftar album ditampilkan dalam format grid dengan gambar thumbnail dan nama file unik, serta navigasi **Previous** dan **Next** untuk berpindah antar halaman.

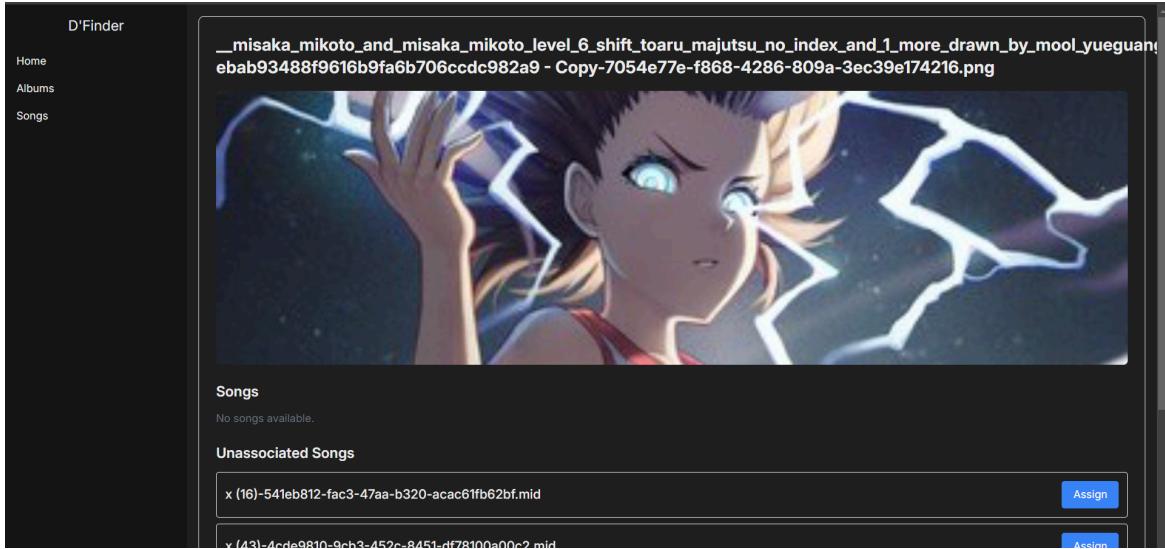


**Gambar 5.** Arsitektur Website Albums Halaman 1

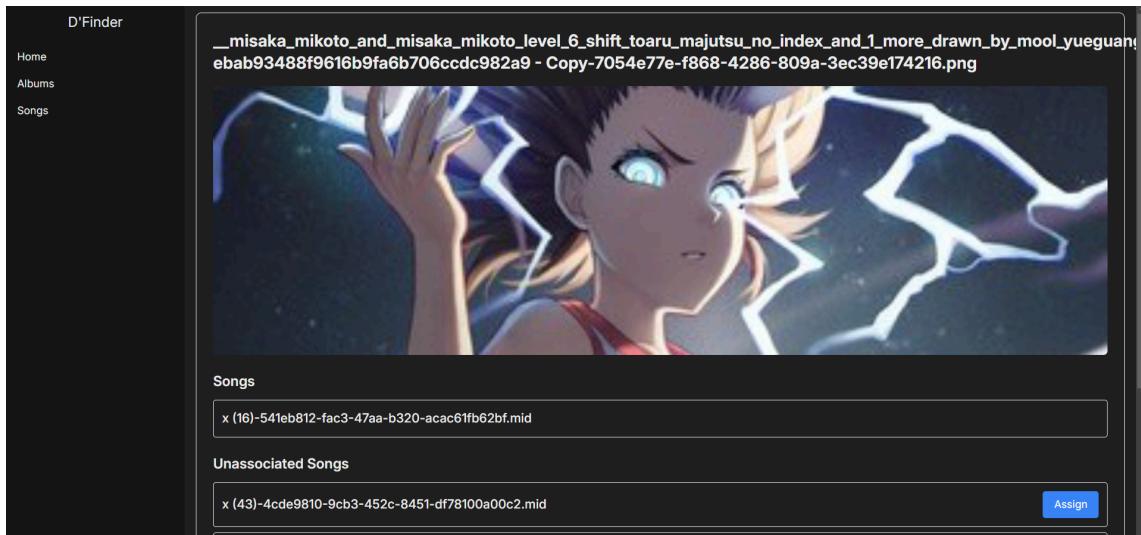


**Gambar 6.** Arsitektur Website Albums Halaman 2

Halaman Detail Album berfungsi untuk melihat detail gambar. Terdapat tombol **Assign** untuk menghubungkan suatu lagu dengan gambar tersebut.



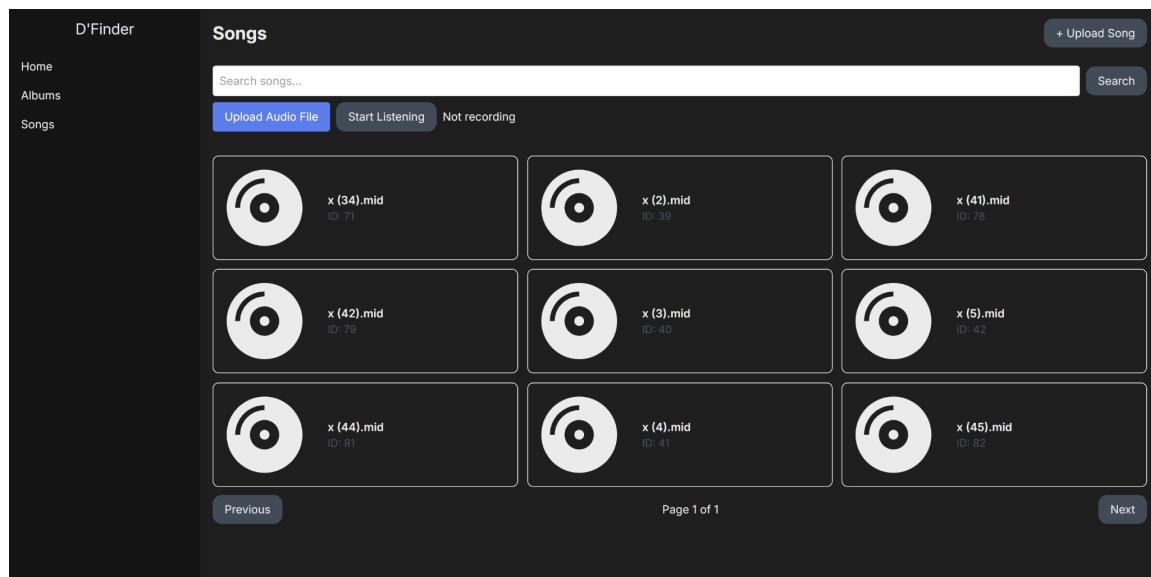
**Gambar 7.** Arsitektur Website Albums Detail Belum Terhubung dengan Song



**Gambar 8.** Arsitektur Website Albums Detail Sudah Terhubung dengan Song

### 3.2.3 Songs

Halaman **Songs** dari website **D'Finder** berfungsi untuk mengelola lagu dalam bentuk album. Terdapat kolom pencarian untuk mencari album, tombol **Upload Song** untuk menambahkan lagu baru, tombol **Search by Humming**, dan **Upload Audio File** untuk mencocokkan audio dengan database. Daftar lagu ditampilkan dalam format grid dengan icon disk dan nama file unik, serta navigasi **Previous** dan **Next** untuk berpindah antar halaman.



**Gambar 9.** Arsitektur Website Songs

### **3.2 Arsitektur Program Information Retrieval (BackEnd)**

Kami menggunakan kombinasi dua bahasa pada backend untuk mengatasi keterbatasan yang ada pada Go, yang dioptimalkan untuk mencapai performa tinggi. Bagian Go menggunakan framework Gin, yang disesuaikan dengan pola Model-Controller-Router (MCR) dan dilengkapi dengan folder tambahan untuk mendukung utilitas. Server konversi MIDI dibuat dengan Python menggunakan file main.py, yang dirancang agar fungsionalitasnya tetap sederhana dan efisien.

#### **3.2.1 Go**

Go dipilih karena kemampuannya dalam menangani beban perhitungan yang berat secara paralel dengan kecepatan yang sangat tinggi, yang sangat berguna untuk operasi backend yang memerlukan skalabilitas dan performa optimal.



**Gambar 10.** Go

#### **3.2.2 Server Midi Convert**

Server konversi MIDI, yang dibangun dengan FastAPI dari Python, memiliki fokus pada konversi file audio menjadi format MIDI, serta pembacaan file MIDI tersebut. Kami memilih Python untuk server ini karena fleksibilitasnya dalam pengolahan data audio dan MIDI, serta banyaknya pustaka yang tersedia untuk mendukung berbagai jenis konversi dan manipulasi file audio. Dengan menggunakan file main.py, kami menjaga fungsionalitas server tetap sederhana dan efisien, menghindari kompleksitas berlebih sementara tetap mempertahankan performa yang cukup baik dalam proses konversi dan pembacaan data. Server ini memungkinkan proses

konversi audio ke MIDI berjalan dengan lancar, menyediakan solusi praktis bagi kebutuhan konversi format audio dalam aplikasi kami."



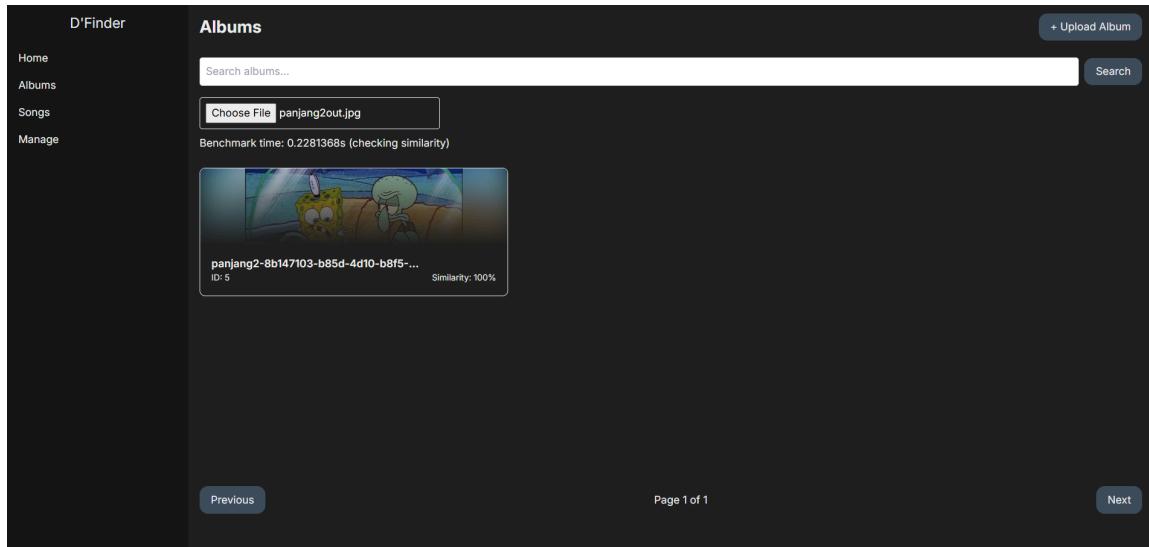
**Gambar 11.** Python

## BAB IV

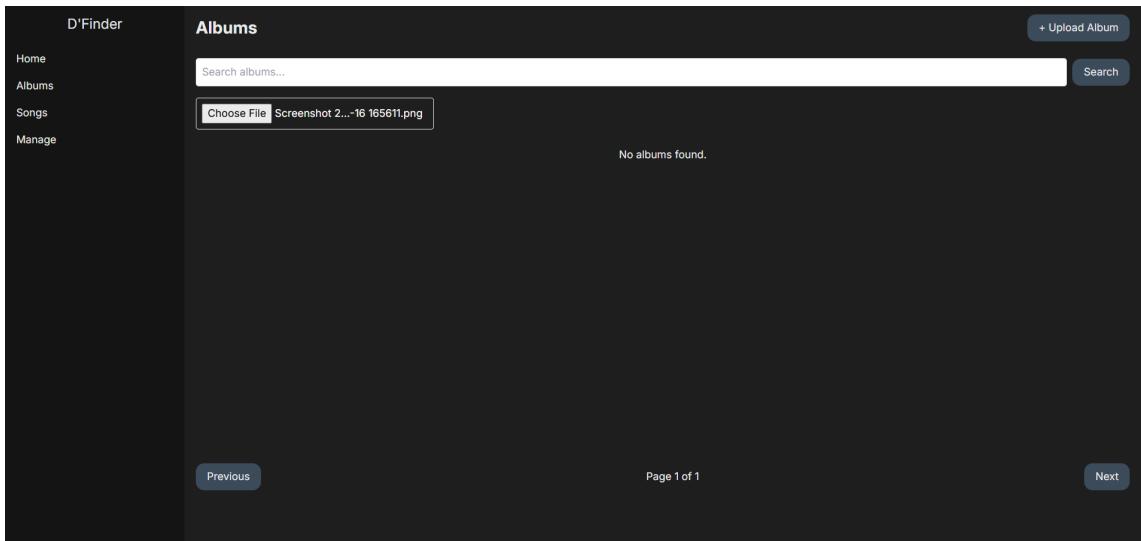
# EKSPERIMEN

### 4.1 Pemrosesan Gambar

Hasil pemrosesan gambar akan menampilkan beberapa gambar yang memiliki kemiripan dengan gambar yang diberikan. Jika tidak ada gambar yang mirip, sistem akan mengeluarkan pesan bahwa tidak terdapat gambar yang sesuai.



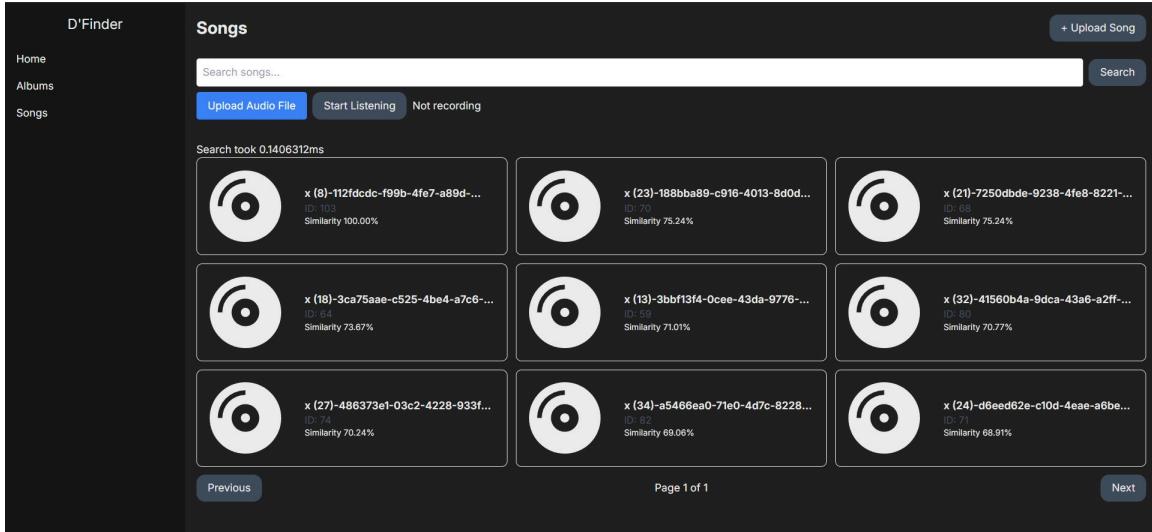
**Gambar 12.** Pemrosesan Gambar Ditemukan



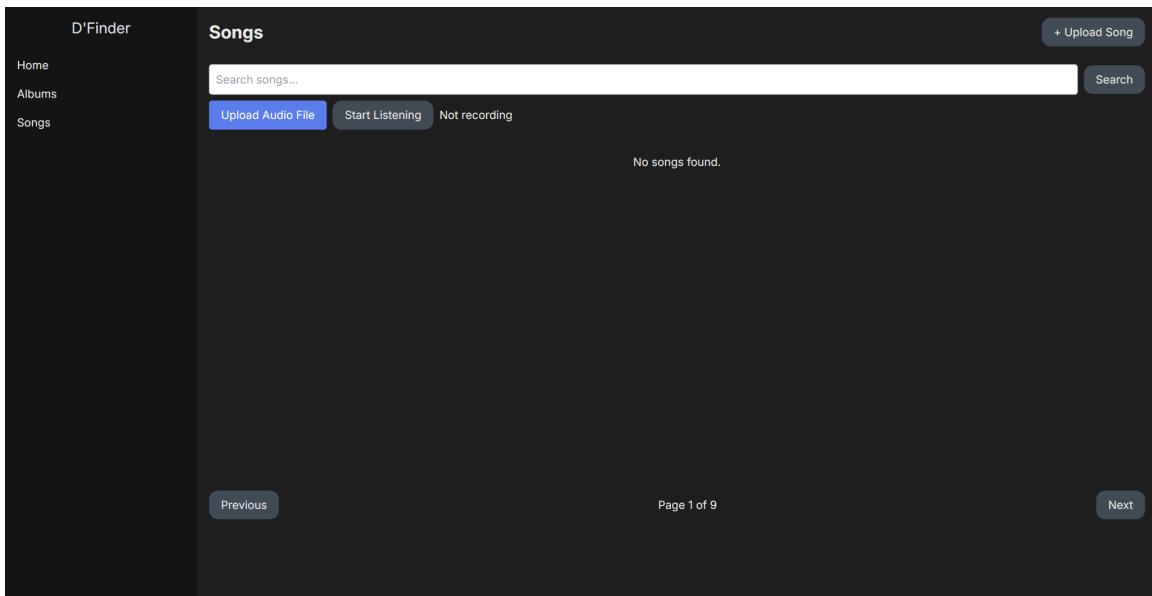
**Gambar 13.** Pemrosesan Gambar Tidak Ditemukan

## 4.2 Pemrosesan Suara

Pada fitur pemrosesan suara, terdapat lagu-lagu yang sudah tersimpan dalam database. Pencocokan suara dapat dilakukan dengan meng-*upload* file suara secara langsung atau menggunakan microphone. Berikut disertakan juga hasil setelah melakukan pencocokan.



**Gambar 14.** Pemrosesan Suara Lagu Ditemukan dengan Metode Upload



**Gambar 15.** Pemrosesan Suara Lagu Tidak Ditemukan dengan Metode Upload

## **BAB V**

### **KESIMPULAN, SARAN, KOMENTAR, DAN REFLEKSI**

#### **5.1 Kesimpulan**

Melalui penggeraan tugas besar ini, kami telah berhasil mengimplementasikan berbagai konsep dalam pemrosesan suara dan gambar dengan memanfaatkan aljabar vektor dan teknik statistik seperti Principal Component Analysis (PCA). Dalam tugas ini, kami mengembangkan dua aplikasi utama: Image Retrieval menggunakan PCA dan Music Information Retrieval melalui metode Query by Humming. Untuk Image Retrieval, kami mengubah gambar menjadi representasi grayscale, melakukan standarisasi data, dan menghitung kesamaan menggunakan jarak Euclidean dalam ruang komponen utama. Sementara itu, untuk Music Information Retrieval, kami berhasil mengonversi audio menjadi vektor fitur, menormalisasi tempo dan pitch, serta mengukur kemiripan antar audio menggunakan Cosine Similarity.

Penerapan PCA pada Image Retrieval memungkinkan kami untuk mengurangi dimensi data gambar tanpa kehilangan informasi penting, sedangkan metode Query by Humming pada Music Information Retrieval memungkinkan pencocokan melodi berdasarkan input humming, yang sangat berguna untuk aplikasi seperti Shazam. Dalam pengembangan aplikasi ini, kami menghadapi tantangan dalam pemrosesan data, tetapi berhasil mengatasinya dengan pendekatan yang tepat dan eksperimen untuk memilih metode yang paling sesuai.

#### **5.2 Saran**

Saran dari penulis, di kesempatan selanjutnya, aplikasi **Image Retrieval** dan **Music Information Retrieval** dapat dikembangkan lebih lanjut dengan fokus pada peningkatan pengalaman pengguna (UI/UX). Beberapa aspek yang dapat diperbaiki mencakup penyempurnaan antarmuka agar lebih intuitif dan mudah digunakan, serta penambahan fitur interaktif untuk membuat aplikasi menjadi lebih user-friendly.

### 5.3 Komentar

Nama Anggota	Komentar
Varel Tiara (13523008)	Gacor mantap keren bagus cool gila takjub kelas
Yonatan Edward Njoto (13523036)	Kalau boleh batasannya dikurangi hehe
Benedict Presley (13523067)	Dari segi algoritma tubesnya kurang menarik. Bukan berarti tidak sulit dalam membuat prediksi yang akurat namun terlalu terbatas dan tidak terlalu menunjukkan kekuatan dari materi-materi algeo seperti tubes sebelumnya. Tetapi saya mengucapkan terima kasih karena algoritma yang diperlukan tidak terlalu susah sehingga tubes ini tidak terlalu chaos.

### 5.4 Refleksi

Pengerjaan tugas besar ini memberikan pengalaman yang berharga dalam mengaplikasikan teori-teori aljabar linear dan pemrosesan sinyal ke dalam pengembangan aplikasi nyata. Kami belajar untuk mengolah data audio dan gambar dengan cara yang efisien, serta menghadapi tantangan teknis yang membutuhkan pemahaman mendalam tentang teknik statistik dan algoritma pengolahan data. Pekerjaan ini juga mengajarkan pentingnya pemilihan metode yang tepat untuk setiap jenis data yang dihadapi, serta bagaimana eksperimen dan pengujian dapat mengarah pada solusi yang lebih optimal. Tugas ini meningkatkan kemampuan kami dalam memahami dan menerapkan konsep-konsep canggih dalam pengolahan informasi, yang tentunya akan berguna dalam proyek-proyek teknologi yang lebih kompleks di masa depan.

**Untuk Tuhan, Bangsa, dan Almamater. Hidup Informatika!**

## **BAB VI**

### **LAMPIRAN**

#### 6.1 Daftar Referensi

- Munir, Rinaldi. 2023. “Singular Value Decomposition (SVD) (Bagian 1)” (<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf>, diakses 6 Desember 2024)
- Munir, Rinaldi. 2023. “Singular Value Decomposition (SVD) (Bagian 2)” (<https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-22-Singular-value-decomposition-Bagian2-2023.pdf>, diakses 6 Desember 2024)
- Pamungkas, Adi. 2023. “Pengenalan Wajah Menggunakan Metode Principal Component Analysis (PCA)”. Pemrograman Matlab. (<https://pemrogramannmatlab.com/2023/08/10/pengenalan-wajah-menggunakan-metode-principal-component-analysis-pca/>, diakses 7 Desember 2024)
- Riswanto, Ujang. 2023. “Image Compression Techniques: A Closer Look at Principal Component Analysis”. (<https://ujangriswanto08.medium.com/image-compression-techniques-a-closer-look-at-principal-component-analysis-67cf7a29fdb9>, diakses 7 Desember 2024)
- Müller, Meinard. 2015. “Fundamentals of Music Processing Audio, Analysis, Algorithms, Applications”. Springer International Publishing. (diakses 9 Desember 2024)

#### 6.2 Tautan Repository

Link repository dari Tugas Besar 2 IF2123 Aljabar Linear Dan Geometri kelompok Bos Pablo 10 adalah sebagai berikut:

<https://github.com/yonatan-nyo/Algeo02-23008.git>

### 6.3 Tautan Video

Link video dari Tugas Besar 2 IF2123 Aljabar Linear Dan Geometri kelompok Bos Pablo 10 adalah sebagai berikut:

<https://youtu.be/hGb0FzL3xtw?si=xXEF02dJaQcq5EHl>