

# **Laporan Tugas Besar II**

## **IF2123 Aljabar Linier dan Geometri**



**Kelompok 34 - Para Pencari Cuan:**  
**Ivant Samuel Silaban 13523129**  
**Amira Izani 13523143**  
**Muhamad Nazih Najmudin 13523144**

**INSTITUT TEKNOLOGI BANDUNG**

**2024**

# DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>BAB I.....</b>	<b>3</b>
I. Pemrosesan Suara dan Gambar.....	3
II. Information Retrieval.....	4
<b>BAB II.....</b>	<b>4</b>
I. Album Picture Finder - Principal Component Analysis.....	4
II. Music Information Retrieval - Query by Humming.....	8
<b>BAB III.....</b>	<b>11</b>
I. Arsitektur Website.....	11
II. Frontend.....	12
III. Backend.....	13
IV. Processing Engine.....	13
V. Alur Kerja Sistem.....	13
<b>BAB IV.....</b>	<b>15</b>
I. Inisialisasi.....	15
II. Image Information Retrieval.....	16
III. Music Information Retrieval.....	18
<b>BAB V.....</b>	<b>20</b>
I. Kesimpulan.....	20
II. Saran.....	20
III. Refleksi.....	21

# BAB I

## DESKRIPSI MASALAH

### I. 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.** 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.

Pada Tugas Besar 1, telah diselesaikan pembahasan mengenai matriks dan implementasinya dalam berbagai aplikasi. Matriks sendiri merupakan salah satu komponen penting dalam aplikasi aljabar vektor. Pada Tugas Besar 2 ini, tugas yang diberikan adalah mengembangkan sebuah aplikasi serupa Shazam, yang dapat menerima input berupa lagu dan mendeteksi nama lagu tersebut beserta beberapa detail lainnya. Dalam pengerjaan tugas ini, aljabar vektor dimanfaatkan untuk membandingkan satu audio dengan audio lainnya.

Konsep Music Information Retrieval (MIR) diterapkan untuk menganalisis dan mengidentifikasi suara berdasarkan fitur-fitur tertentu. Selain itu, konsep Principal Component Analysis (PCA) juga digunakan untuk mengelompokkan kumpulan audio melalui deteksi wajah yang diasumsikan sebagai penyanyi.

## **II. Information Retrieval**

Information Retrieval merupakan konsep pengambilan informasi dari suatu data dengan memberikan data tertentu sebagai input. Dalam tugas besar ini, terdapat dua jenis Information Retrieval yang menjadi fokus utama, yaitu Image Retrieval dan Music Information Retrieval (MIR). Image Retrieval adalah konsep di mana sebuah input berupa gambar dimasukkan dengan harapan mendapatkan gambar dalam dataset yang sesuai berdasarkan informasi dan hasil perhitungan tertentu. Di sisi lain, Music Information Retrieval (MIR) merupakan konsep untuk menerima input berupa audio dan menemukan audio yang relevan di dalam dataset berdasarkan informasi serta analisis yang dilakukan.

Pada tugas besar ini, implementasi Image Retrieval akan dilakukan menggunakan metode Principal Component Analysis (PCA), sedangkan Music Information Retrieval akan diterapkan dengan memanfaatkan teknik humming.

pca

# **BAB II**

## **TEORI SINGKAT**

### **I. 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).

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  $M \cdot N$ :

## 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:

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

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

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

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).
- $\mu$ : 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:

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

- $d(q, z_i)$  : 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.

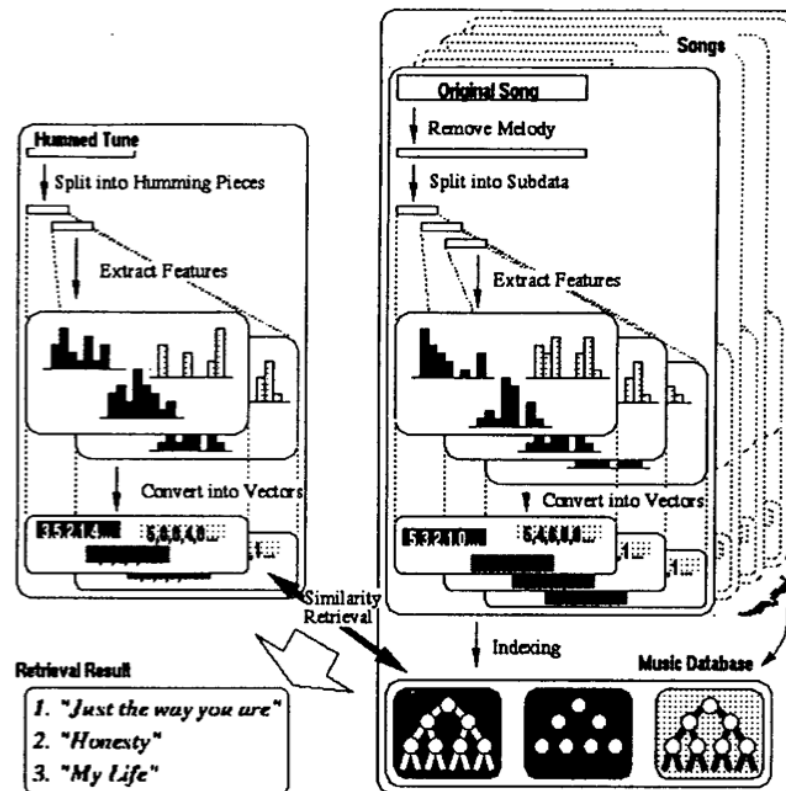
Lalu, Urutkan hasil berdasarkan jarak terkecil.

## 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.

## II. 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. Perhitungan Similaritas

Ubah setiap histogram menjadi sebuah vektor dan lakukan perhitungan kemiripan nya 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 III

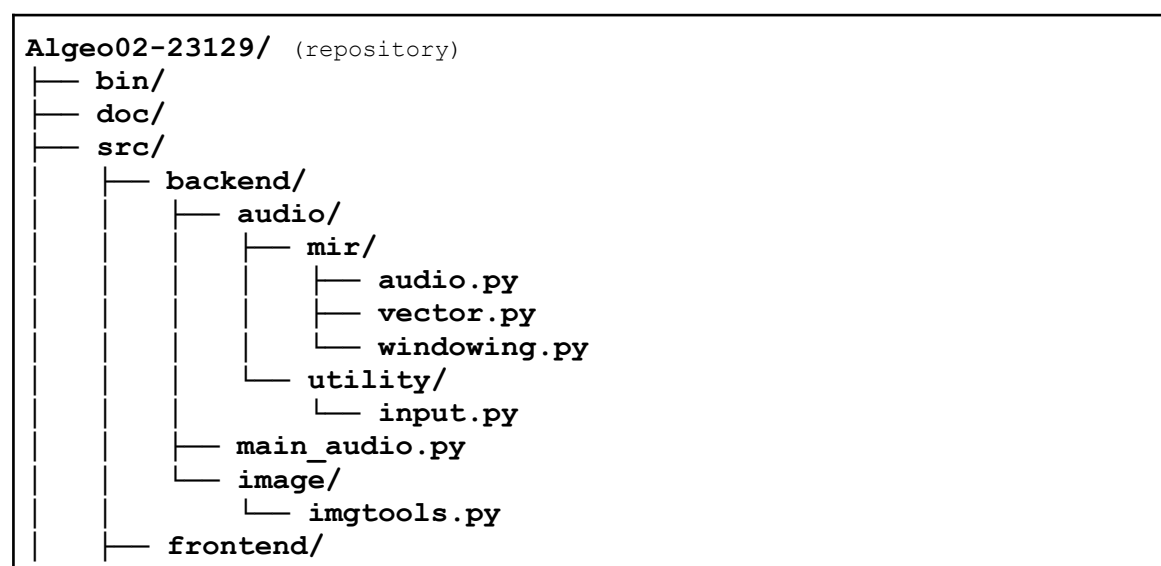
## ARSITEKTUR WEBSITE DAN PROGRAM INFORMATION RETRIEVAL

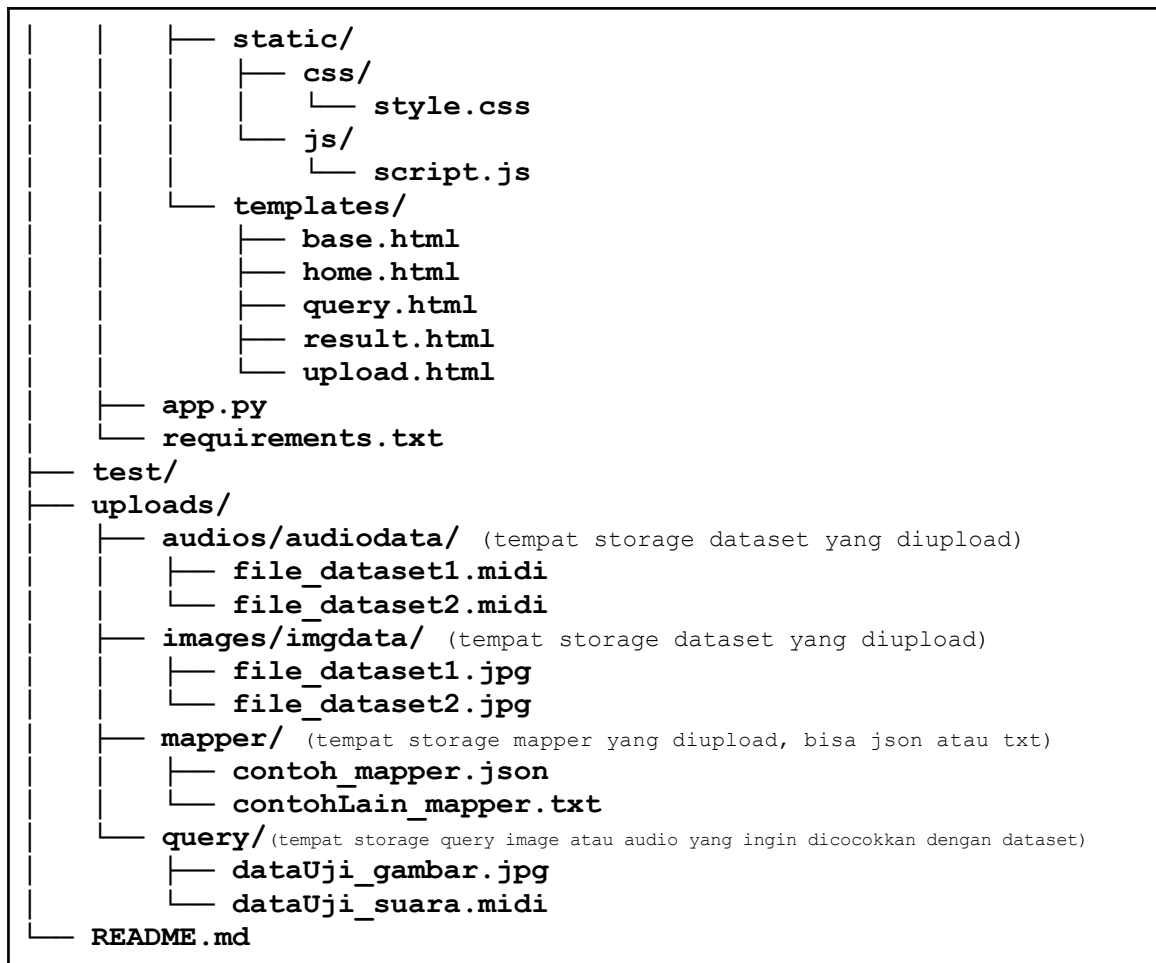
### I. Arsitektur Website

Website ini dirancang dengan arsitektur sederhana yang mencakup tiga komponen utama, yaitu Frontend, Backend, dan Processing Engine. Diagram arsitektur dapat digambarkan sebagai berikut:

- A. **Frontend:** Berfungsi sebagai antarmuka pengguna untuk mengunggah file dataset dan query, memilih jenis similaritas (image/audio), serta menampilkan hasil similaritas.
- B. **Backend:** Bertanggung jawab sebagai server untuk menangani komunikasi antara frontend dan processing engine, menerima file yang diunggah, dan memproses permintaan.
- C. **Processing Engine:** Memuat algoritma untuk pemrosesan similaritas, yaitu "Album Picture Finder - Principal Component Analysis" untuk image retrieval dan "Music Information Retrieval - Query by Humming" untuk audio retrieval.
- D. **Storage Lokal:** Semua dataset, query, dan mapper disimpan di direktori lokal tanpa menggunakan database atau cloud storage.

Struktur penempatan folder dan file dalam repository source code ditunjukkan pada diagram tree berikut:





## II. Frontend

Frontend dikembangkan menggunakan Flask dengan template engine bawaan, yaitu Jinja2. Struktur antarmuka pengguna terdiri dari beberapa halaman utama:

- A. **Home (home.html):** Halaman awal yang menyediakan opsi untuk mengunggah file dataset atau query.
- B. **Upload (upload.html):** Halaman untuk mengunggah dataset image, audio, atau file mapper.
- C. **Query (query.html):** Halaman untuk memilih jenis similaritas yang ingin dilakukan (image/audio) dan mengunggah query file.
- D. **Result (result.html):** Halaman untuk menampilkan hasil similaritas, berupa skor kemiripan atau file yang paling mirip.

Frontend juga menggunakan file statis seperti CSS (di folder static/css/style.css) untuk pengaturan tampilan dan JavaScript (di folder static/js/script.js) untuk interaktivitas tambahan.

### III. Backend

Backend dikembangkan menggunakan Flask sebagai framework utama untuk menangani logika aplikasi. Backend berfungsi sebagai penghubung antara frontend dan processing engine melalui API endpoint. Fitur utama backend meliputi:

- A. **Endpoint Upload:** Menerima file dataset image, audio, dan mapper, kemudian menyimpannya ke direktori lokal uploads/images/imgdata/, uploads/audios/audiodata/, dan uploads/mapper/.
- B. **Endpoint Query:** Menerima file query dari pengguna, menyimpannya ke direktori uploads/query/, dan memanggil processing engine untuk perhitungan similaritas.
- C. **Endpoint Result:** Mengembalikan hasil similaritas ke frontend dalam format yang dapat ditampilkan ke pengguna.

Struktur backend berada di direktori src/backend/, dengan pembagian modul berikut:

- A. **audio/:** Berisi modul untuk pemrosesan similaritas audio (MIR - Query by Humming).
- B. **image/:** Berisi modul untuk pemrosesan similaritas image (PCA - Album Picture Finder).
- C. **utility/:** Berisi modul pendukung seperti input handling.

### IV. Processing Engine

Processing Engine bertugas menjalankan algoritma utama untuk perhitungan similaritas. Algoritma yang digunakan adalah:

- A. **Album Picture Finder - Principal Component Analysis (PCA):** Algoritma ini digunakan untuk menemukan similaritas antara query image dan dataset image. Pemrosesan dilakukan pada modul imgtools.py di direktori src/backend/image/.
- B. **Music Information Retrieval - Query by Humming (MIR):** Algoritma ini digunakan untuk mencocokkan query audio dengan dataset audio berdasarkan analisis fitur audio. Pemrosesan dilakukan pada main\_audio.py yang memanggil modul audio.py, vector.py, dan windowing.py di direktori src/backend/audio/mir/.

Hasil pemrosesan dari kedua algoritma ini dikembalikan dalam bentuk skor similaritas atau file dengan tingkat kemiripan tertinggi.

### V. Alur Kerja Sistem

1. **Unggah Dataset dan Mapper:** Pengguna membuka halaman upload untuk mengunggah dataset image, audio, dan mapper ke direktori lokal masing-masing.

File disimpan di:

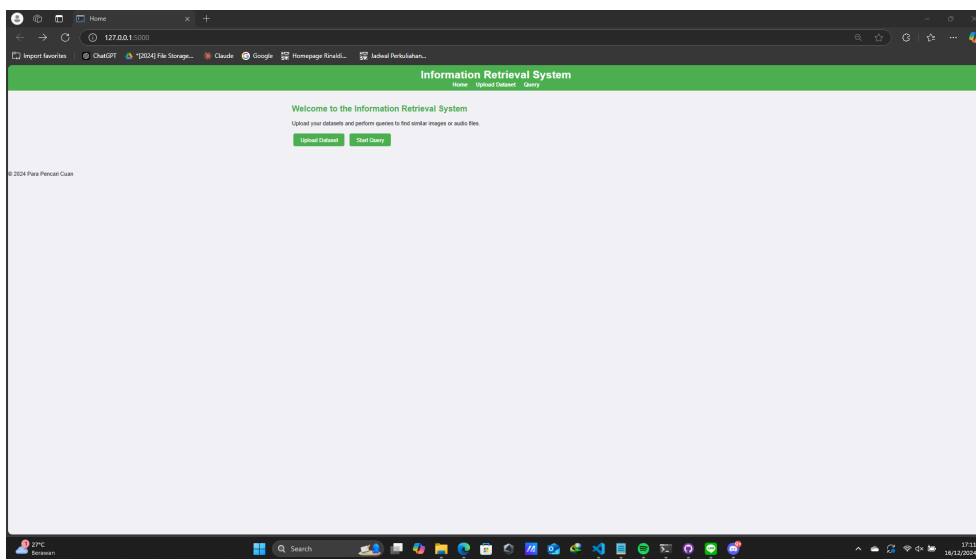
- a. uploads/images/imgdata/ untuk image dataset.
  - b. uploads/audios/audiodata/ untuk audio dataset.
  - c. uploads/mapper/ untuk file mapper.
2. **Unggah Query File:** Pengguna membuka halaman query dan memilih jenis similaritas yang diinginkan (image/audio), lalu mengunggah file query. File query disimpan di direktori uploads/query/.
  3. **Pemrosesan Similaritas:** Backend memanggil processing engine sesuai jenis similaritas yang dipilih:
    - a. **Untuk image:** Backend memanggil imgtools.py yang menggunakan algoritma PCA untuk menghitung similaritas query dengan dataset image.
    - b. **Untuk audio:** Backend memanggil main\_audio.py yang menggunakan modul MIR (audio.py, vector.py, dan windowing.py) untuk mencocokkan query dengan dataset audio.
  4. **Pengembalian Hasil:** Backend menerima hasil perhitungan dari processing engine dan mengirimkannya ke frontend. Hasil berupa skor similaritas atau daftar file yang mirip ditampilkan di halaman hasil (result.html).

# BAB IV

## EXPERIMENT

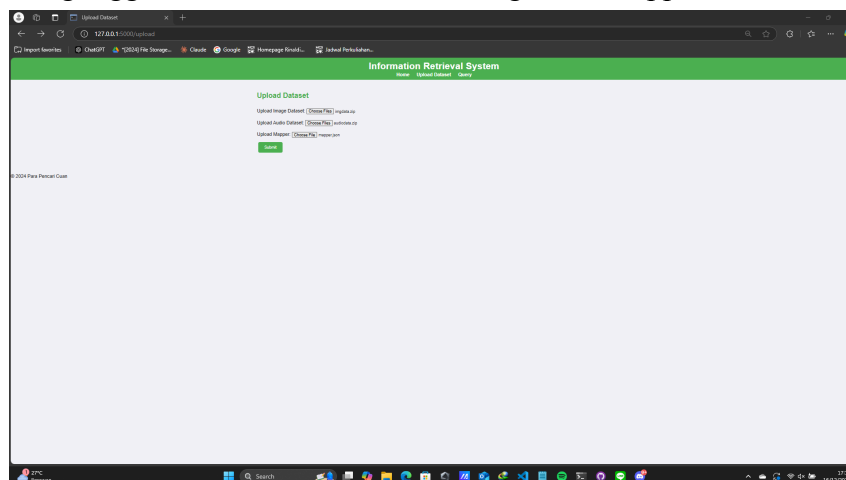
### I. Inisialisasi

Sebelum melakukan proses image information retrieval atau music information retrieval, diperlukan pengunggahan dataset gambar dan dataset audio terlebih dahulu. Dataset ini berfungsi sebagai referensi untuk membandingkan query yang akan diunggah pada tahap berikutnya. Selain mengunggah dataset gambar dan dataset audio, mapper juga diunggah untuk menghubungkan file audio dengan gambar yang relevan.



Gambar 4. Tampilan Awal

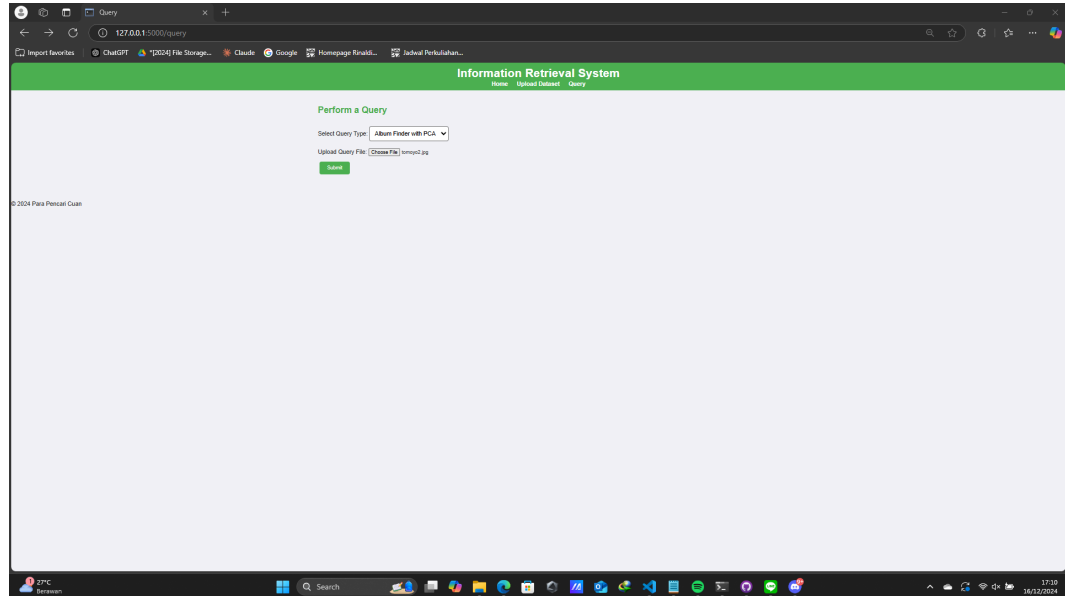
#### 1. Mengunggah dataset audio, dataset image, dan mapper



**Gambar 5.** Page mengunggah dataset audio, dataset image, dan mapper

## II. Image Information Retrieval

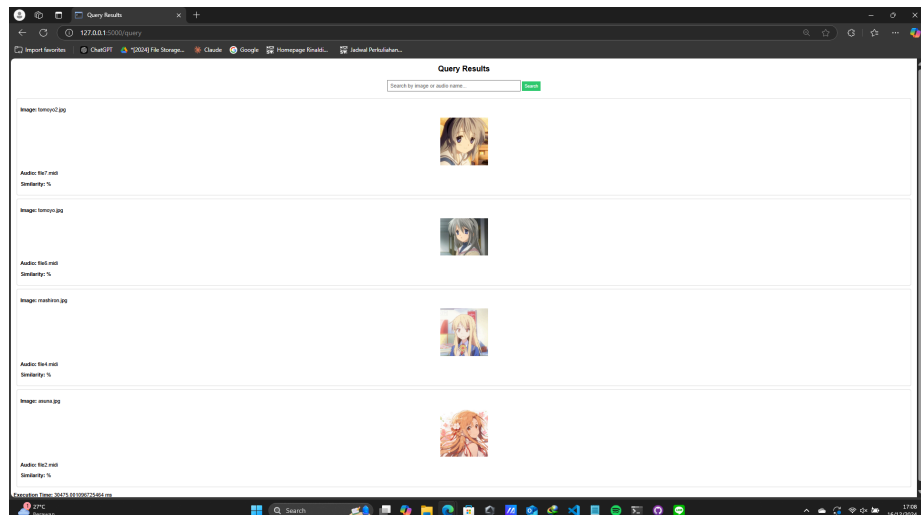
### 1. Mengunggah gambar query



**Gambar 11.** Tampilan page Query Album Finder with PCA

### 2. Hasil Pengujian

#### a. Apabila mengunggah query gambar yang ada dalam dataset



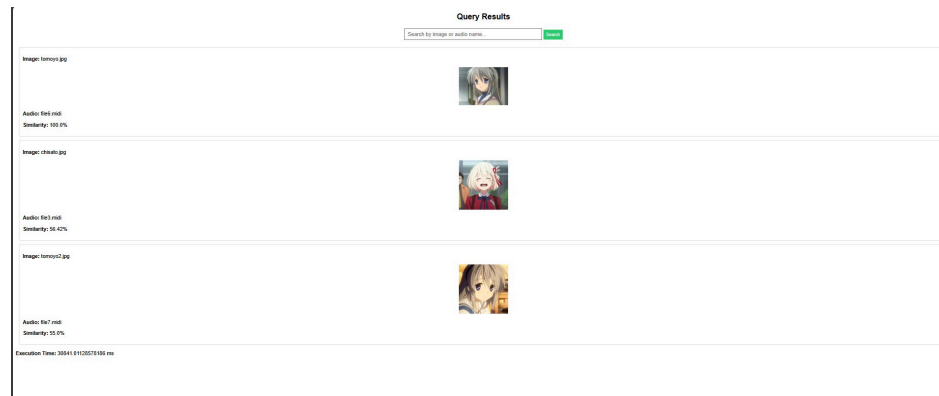
**Gambar 13.** Hasil pencarian apabila mengunggah query gambar yang ada dalam dataset

Berdasarkan uji coba ini, sistem berhasil menemukan gambar yang sesuai dengan query. Hasil ini menunjukkan bahwa metode PCA dapat bekerja dengan baik dalam mendeteksi kesamaan gambar berdasarkan representasi numeriknya. Dengan demikian, Akurasi sistem tinggi dalam pencocokan jika



query gambar memang ada di dataset. Ini membuktikan sistem dapat merepresentasikan fitur gambar.

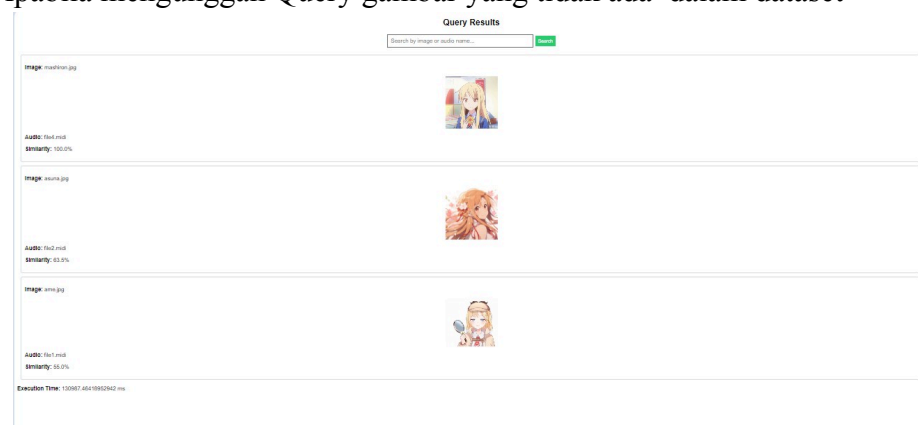
- b. Apabila mengunggah Query gambar dengan objek yang sama namun gambar berbeda dalam dataset



**Gambar 14.** Hasil pencarian apabila mengunggah Query gambar dengan objek yang sama namun gambar berbeda dalam dataset

Berdasarkan uji coba ini, sistem berhasil menemukan hasil pencarian pertama yang merupakan objek yang sama dengan query gambar meskipun gambar berbeda. Hal ini menunjukkan bahwa metode yang digunakan, seperti PCA, mampu menangkap fitur-fitur utama dari objek dalam gambar, terlepas dari perbedaan visual lainnya. Dengan demikian, sistem memiliki kemampuan yang baik dalam mendeteksi kemiripan berdasarkan karakteristik utama objek, bukan hanya kesamaan piksel secara langsung.

- c. Apabila mengunggah Query gambar yang tidak ada dalam dataset



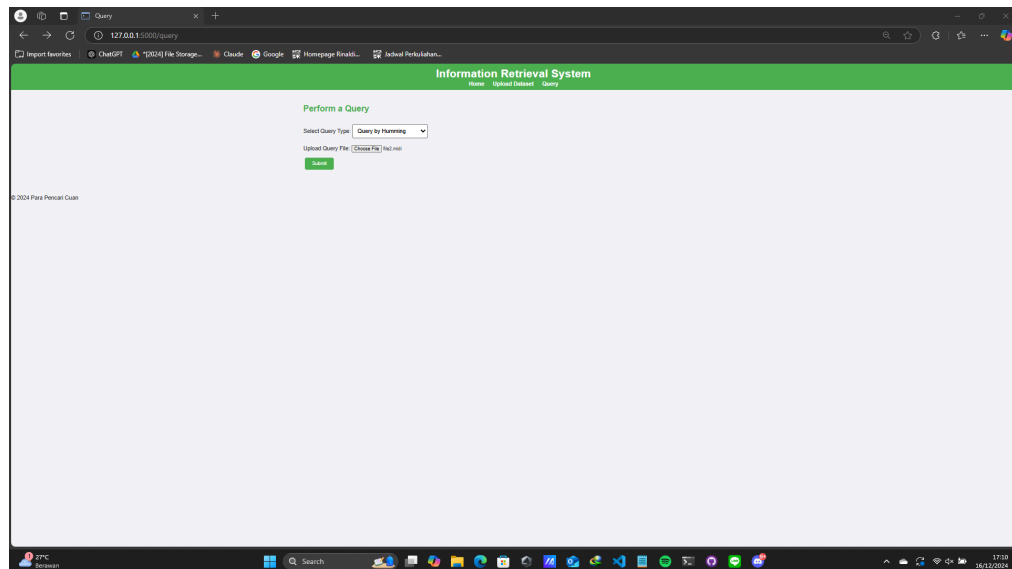
**Gambar 14.** Hasil pencarian apabila mengunggah query gambar yang tidak ada dalam dataset

Berdasarkan uji coba ini, Sistem mampu menemukan gambar yang mirip, meskipun tidak identik dengan query. Ini menunjukkan PCA juga efektif dalam menangkap hubungan visual antara gambar meskipun ada

perbedaan kecil. Dengan demikian, sistem memiliki toleransi terhadap variasi kecil dalam gambar, sehingga mendukung pencocokan berbasis fitur visual.

### III. Music Information Retrieval

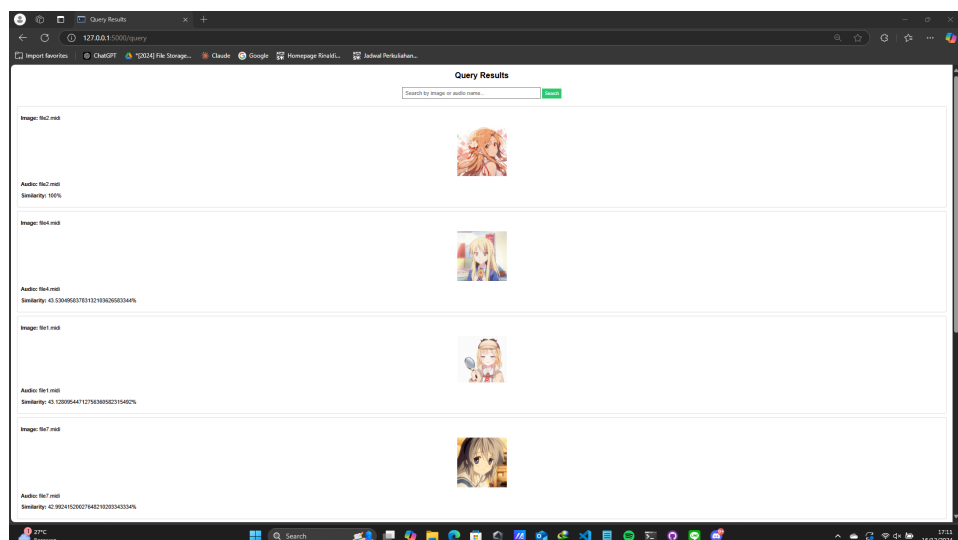
#### 1. Mengunggah audio query



**Gambar 15.** Tampilan page Query Album Finder with PCA

#### 2. Hasil pengujian

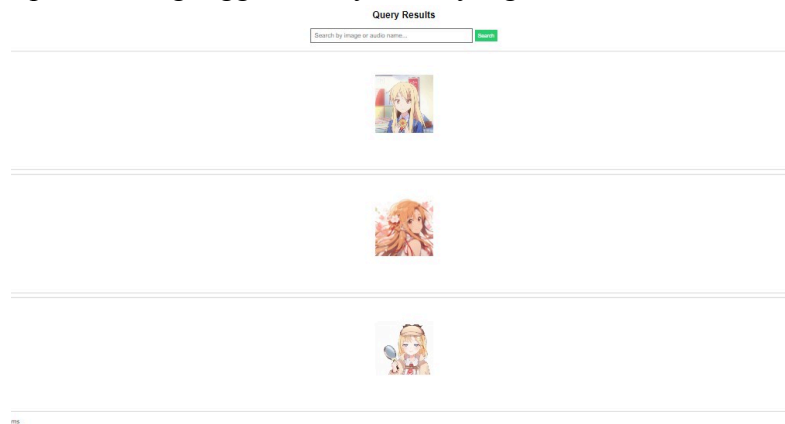
##### a. Apabila mengunggah query audio yang ada dalam dataset



**Gambar 17.** Hasil pencarian apabila mengunggah query audio yang ada dalam dataset

Berdasarkan uji coba ini, sistem berhasil menemukan audio yang sesuai dengan query. Hasil ini menunjukkan bahwa metode yang digunakan dapat bekerja dengan baik dalam mendeteksi kesamaan audio berdasarkan representasi fitur numeriknya. Dengan demikian, akurasi sistem tinggi dalam pencocokan jika query audio memang ada dalam dataset. Ini membuktikan bahwa sistem mampu merepresentasikan fitur audio dengan efektif.

- b. Apabila mengunggah Query audio yang tidak ada dalam dataset



**Gambar 18.** Hasil pencarian apabila mengunggah query audio yang tidak ada dalam dataset

Berdasarkan uji coba ini, sistem mampu menemukan hasil yang memiliki tingkat kemiripan rendah dengan query audio yang diunggah, meskipun query tersebut tidak ada dalam dataset. Ini menunjukkan bahwa metode yang digunakan tetap dapat menentukan dan mencari hasil berdasarkan representasi fitur numerik, walaupun persentase kemiripan tidak tinggi.

## **BAB V**

### **KESIMPULAN**

#### **I. Kesimpulan**

Pada pengerjaan proyek Tugas Besar 2 IF2123 Aljabar Linear dan Geometri ini, kami mengembangkan sebuah sistem untuk Image Information Retrieval dan Music Information Retrieval. Sistem ini menggunakan metode Principal Component Analysis (PCA) untuk Image Retrieval dan metode Query by Humming untuk music retrieval. Setiap mode dirancang untuk memberikan hasil optimal berdasarkan dataset dan query yang diunggah.

Dalam pengujian, sistem menunjukkan performa yang baik dalam mendeteksi kesamaan query pada dataset yang ada. Pada Image Information Retrieval, sistem mampu mengenali karakteristik utama dari objek dalam gambar, bahkan pada query gambar yang berbeda secara visual tetapi memiliki objek yang sama. Sementara itu, pada Music Information Retrieval, sistem dapat mencocokkan query audio dengan dataset berdasarkan representasi fitur numeriknya. Selain itu, diterapkan pencocok berbasis jarak (euclidean distance) untuk gambar dan cosine similarity untuk audio, yang memberikan fleksibilitas dan akurasi dalam mengidentifikasi hasil pencarian.

Secara keseluruhan, sistem ini memberikan solusi komprehensif untuk information retrieval berbasis audio dan gambar, sekaligus menunjukkan potensi penggunaan metode PCA dan humming dalam aplikasi nyata. Pengembangan lebih lanjut pada cakupan dataset dan metode representasi fitur dapat meningkatkan performa sistem di masa mendatang.

#### **II. Saran**

Berdasarkan apa yang telah dilakukan pada tugas besar ini, kami memiliki beberapa saran yang dapat diterapkan pada pengembangan selanjutnya. Saran-saran tersebut kami rangkum menjadi hal-hal berikut:

1. Pengembangan selanjutnya, dapat memanfaatkan fitur-fitur bawaan dari library yang dapat membantu pengerjaan, terutama pada pemrosesan similaritas audio.
2. Akurasi similaritas, baik pengecekan similaritas untuk image maupun audio, dapat ditingkatkan kembali.
3. Performansi dalam efisiensi penggunaan sumber daya dan waktu sebaiknya ditingkatkan. peningkatan performansi dapat dilakukan dengan menggunakan

algoritma lain yang lebih efisien, atau bahasa pemrograman lain yang dapat dieksekusi lebih cepat oleh komputer.

4. Sebaiknya dilakukan lebih banyak test case dan dataset dengan kombinasi yang beragam, agar kesalahan program dapat terlihat dan diperbaiki.
5. Jika program sudah berjalan dengan baik, sebaiknya dipublikasikan di internet agar kebermanfaatan dan ilmunya dapat dirasakan juga oleh masyarakat.

### **III. Refleksi**

Berdasarkan apa yang telah dilakukan ketika membuat program website untuk mengimplementasikan algoritma PCA dan MIR, serta melakukan testing untuk menguji program tersebut, kami mendapati beberapa kekurangan yang dapat diperbaiki untuk proyek selanjutnya yang semacam. Refleksi diantaranya adalah:

1. Keterlambatan penyampaian informasi dari pihak asisten kepada mahasiswa dalam hal aturan penggunaan library untuk menghitung SVD dan lain sebagainya.
2. Perhitungan Eigen yang diajarkan di kelas tidak dapat digunakan untuk matriks-matriks berukuran besar, dan justru menghasilkan runtime error.
3. Programmer yang mengerjakan Query by Humming tidak mahir menggunakan library eksternal seperti numpy sehingga harus membuat secara manual.
4. Performansi program Query by Humming cenderung lambat akibat algoritma windowing yang terlalu banyak. Program akan berjalan sangat lama pada dataset yang banyak dan audio yang panjang.
5. Pengetesan lebih lanjut untuk mengetes variasi input belum dapat dilakukan, karena belum memahami cara memotong atau mengedit file midi.
6. Kurangnya kemampuan dan pengalaman anggota tim dalam membuat frontend serta integrasinya dengan backend, sehingga kesulitan dalam mengeksplor dan debugging.