

LAPORAN TUGAS BESAR 2
IF2123 ALJABAR LINIER DAN GEOMETRI
Image Retrieval dan Music Information Retrieval Menggunakan
PCA dan Vektor



Disusun oleh:

Kelas K2 - Kelompok 56 - BARmusic

Nama	NIM
Adhimas Aryo Bimo	13523052
Muh. Rusmin Nurwandi	13523068
Aria Judhistira	13523112

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.....	4
BAB 1 : Deskripsi Masalah.....	5
1.1. Pemrosesan Suara dan Gambar.....	5
1.2. Information Retrieval.....	6
Album Picture Finder - Principal Component Analysis.....	6
1. Image Processing and Loading.....	8
2. Data Centering (Standardization).....	9
3. PCA Computation Using Singular Value Decomposition (SVD).....	9
4. Similarity Computation.....	10
5. Retrieval and Output.....	11
Music Information Retrieval - Query by Humming.....	12
1. Pemrosesan Audio.....	12
2. Ekstraksi Fitur.....	13
3. Penghitungan Similaritas.....	14
1.3. Arsitektur Website.....	15
BAB 2: Teori Singkat.....	19
1.1. Sistem Temu Balik Suara (MIR).....	19
1.2. Metode Audio Retrieval by Humming.....	19
1.3. Image Retrieval dengan PCA.....	20
BAB 3 : Arsitektur Website (Frontend dan Backend).....	22
3.1 Frontend.....	22
3.2 Backend.....	25
BAB 4: Eksperimen.....	28
4.1 Eksperimen Temu Balik Gambar.....	28
4.1.1. Temu Balik Gambar yang Ada di Dataset.....	28
4.1.2. Temu Balik Gambar yang Ada di Dataset (Hitam-Putih).....	28
4.1.3. Temu Balik Gambar yang Ada di Dataset (Rotasi).....	29
4.1.4. Temu Balik Gambar yang Tidak ada di Dataset.....	30
4.1.5. Temu Balik Gambar yang Tidak ada di Dataset (Hitam-Putih).....	31
4.1.6. Temu Balik Gambar yang Tidak ada di Dataset (Rotasi).....	32
4.2 Eksperimen Temu Balik Audio.....	33
4.2.1. Temu Balik Audio yang Ada di Dataset.....	33
4.2.2. Temu Balik Audio yang Tidak ada di Dataset.....	33
4.2.3. Temu Balik Audio dengan Microphone.....	34
BAB 5 : Kesimpulan, Saran, Komentar, dan Refleksi.....	35
5.1. Kesimpulan.....	35
5.2. Saran.....	35
5.3. Komentar.....	35
5.4. Refleksi.....	36
Lampiran.....	37

DAFTAR GAMBAR

Gambar 1. Shazam sebagai aplikasi audio retrieval system.....	4
Gambar 2. Tahapan pada MIR dengan metode Query By Humming.....	11
Gambar 3. Cosine Similarity Formula.....	14
Gambar 4. Contoh tampak depan dari website Information Retrieval (Humming).....	15
Gambar 5. Contoh tampak depan dari website Information Retrieval (Album).....	15
Gambar 3.1 Struktur Folder Frontend.....	21
Gambar 3.2 Tampilan Website Query Album.....	22
Gambar 3.3 Tampilan Website Query Lagu.....	23
Gambar 3.4 Tampilan Temu Balik Gambar.....	23
Gambar 3.5 Struktur Folder Backend.....	24
Gambar 4.1 Hasil Percobaan Temu Balik Gambar A.....	27
Gambar 4.2 Hasil Percobaan Temu Balik Gambar B.....	28
Gambar 4.3 Percobaan Gambar C.....	28
Gambar 4.4 Percobaan Gambar D.....	29
Gambar 4.5 Percobaan Gambar E.....	30
Gambar 4.6 Percobaan Gambar F.....	31
Gambar 4.7 Percobaan Audio A.....	32
Gambar 4.8 Percobaan Audio B.....	32
Gambar 4.9 Percobaan Audio C.....	33

BAB 1 : 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, 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. Information Retrieval

Information Retrieval adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Pada tugas besar ini, anda akan berikutik 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.

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.

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

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:

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

- 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:

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

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:

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

Dimana:

- q : Vektor proyeksi dari gambar query ke ruang komponen utama (PCA).
- q' : Gambar query dalam format vektor (setelah grayscale, resize, dan flattening).
- μ : 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.

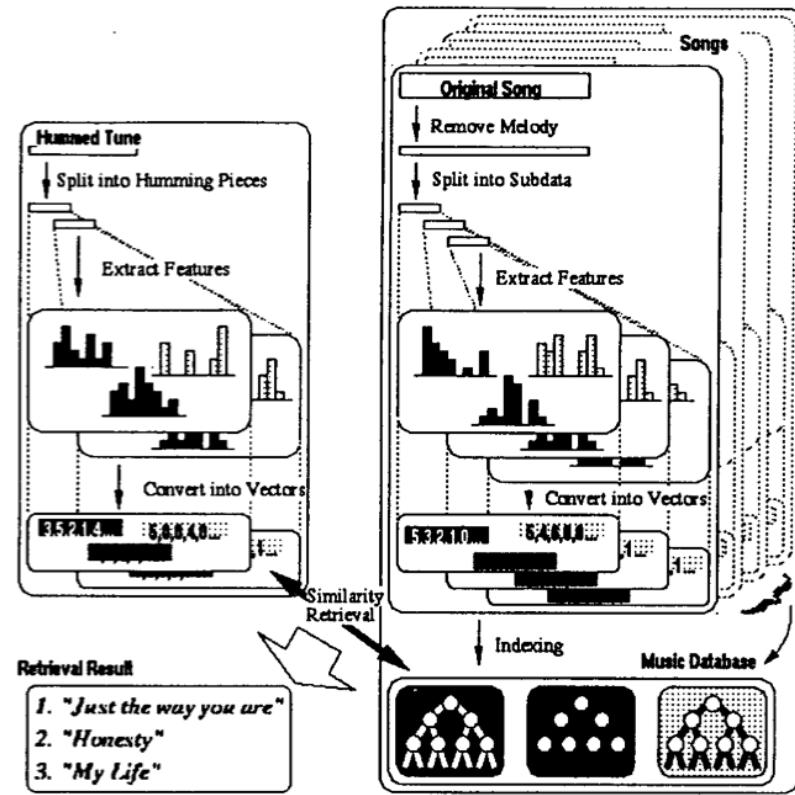
Lalu, Urutkan hasil berdasarkan jarak terkecil.

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 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:

$$H_{norm} = \frac{H[d]}{\sum_d^{127} H[d]}$$

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 :

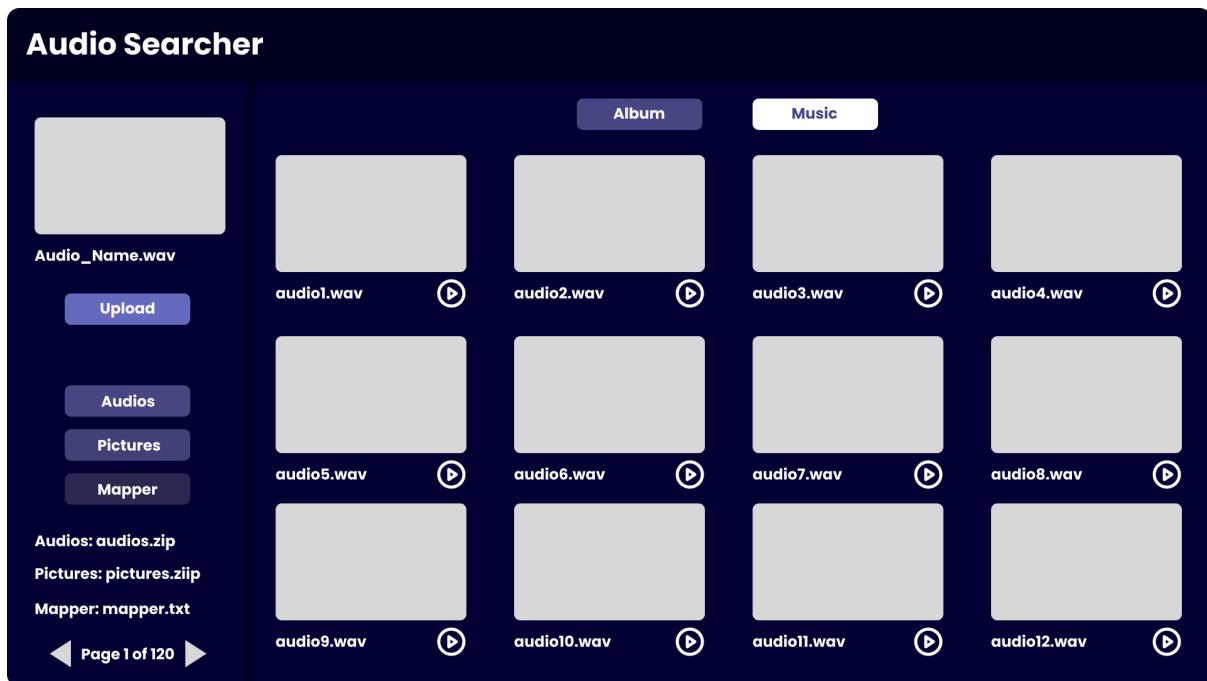
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Gambar 3. Cosine Similarity Formula

1.3. 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 4. Contoh tampak depan dari website Information Retrieval (Humming)



Gambar 5. 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 2: Teori Singkat

1.1. Sistem Temu Balik Suara (MIR)

Sistem Temu Balik Suara atau *Music Information Retrieval* (MIR) merupakan sebuah metode untuk mendapatkan informasi dari sebuah input suara / audio. Metode ini memungkinkan untuk mendapatkan audio pada sebuah dataset menggunakan input audio yang lain. Dalam pendekatan MIR, sebuah input suara akan dianalisis untuk menemukan hubungan atau kemiripan dengan data lain yang tersimpan. Proses untuk melakukan Sistem Temu Balik ini melibatkan beberapa teknik seperti ekstraksi fitur audio, pencocokan pola, dan penerapan algoritma *machine learning* untuk memaksimalkan akurasi dari pencarian.

Penerapan sistem MIR sangat luas di berbagai bidang. Salah satu pengaplikasian yang paling umum dari MIR adalah pada aplikasi Shazam. Aplikasi tersebut dapat mendeteksi sebuah lagu hanya berdasarkan potongan input audio kecil. Aplikasi ini juga dapat mengelompokkan lagu berdasarkan genre atau suasana. Tidak hanya itu, dalam bidang medis, MIR dapat digunakan untuk terapi berbasis musik.

1.2. Metode Audio Retrieval by Humming

Metode yang digunakan untuk mengambil dan memproses audio pada Tugas Besar 2 ini adalah metode pengambilan audio dengan *humming*. Metode ini, seperti namanya, memungkinkan pengguna untuk mencari file audio dengan *query* berupa *humming* atau menyenandungkan melodi dari audio yang diinginkan. Proses ekstraksi file audio dapat dibagi menjadi tiga tahap utama, yakni pemrosesan audio, ekstraksi fitur, dan perhitungan kemiripan audio.

Tahap pemrosesan audio bertujuan untuk mengolah data pada dataset dan dari *query* supaya dapat dibandingkan dengan lebih mudah. Pemrosesan dilakukan pada track melodi utama pada file MIDI. Melodi utama tersebut akan dipecah menjadi bagian-bagian kecil menggunakan metode *windowing*. Setiap bagian berisi 20-40 ketukan dengan *sliding window* sebesar 4-8 ketukan. Setelah itu, dilakukan normalisasi data untuk membuatnya lebih konsisten. Setiap nada dikonversi menjadi representasi numerik untuk memudahkan proses perhitungan kemiripan.

Tahap selanjutnya adalah ekstraksi fitur yang mengubah data melodi menjadi representasi berupa histogram vektor. Representasi histogram berdasarkan tiga fitur, yakni distribusi *Absolute Tone Based (ATB)*, distribusi *Relative Tone Based (RTB)*, dan distribusi *First Tone Based (FTB)*. Setiap distribusi menganalisis melodi berdasarkan skala MIDI yang berbeda-beda. Setelah didapatkan, nilai-nilai histogram akan dinormalisasi supaya setiap nilai berada dalam skala probabilitas.

Tahap terakhir adalah perhitungan kemiripan antara *query* dengan dataset menggunakan metode *cosine similarity*. Metode ini melibatkan perhitungan sudut cosinus antara dua vektor. Sudut yang semakin kecil (atau dalam sudut cosinus semakin mendekati 1) berarti kedua audio semakin mirip.

1.3. Image Retrieval dengan PCA

Pengambilan dan pemrosesan gambar pada Tugas Besar 2 ini menggunakan metode PCA atau Principal Component Analysis. PCA sendiri adalah teknik statistika yang digunakan untuk mengurangi dimensi data sembari mempertahankan sebanyak variasi/informasi mungkin. Hal ini dicapai dengan mengetahui arah atau komponen data yang paling bervariasi dan memproyeksikan data ke arah tersebut.

Sebelum dilakukan PCA, data perlu diekstrak dulu dari gambar. Hal ini dicapai dengan mengubah gambar menjadi grayscale terlebih dahulu dengan mengubah nilai-nilai RGB-nya. Ukuran gambar kemudian diatur supaya seragam dan vektor *grayscale* dijadikan 1 dimensi supaya dimuat dalam satu vektor keseluruhan.

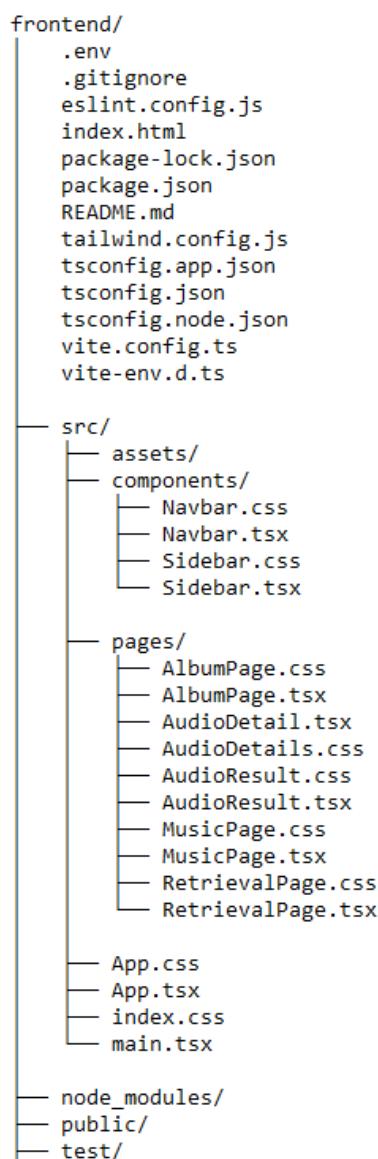
Tahap selanjutnya adalah menstandarisasi data di sekitar nilai 0 dengan mengurangi masing-masing data dengan rata-rata keseluruhan. Setelah data distandarisasi, PCA dilakukan pada data dengan membuat matriks kovarians dari data tersebut. Setelah itu, dilakukan dekomposisi SVD untuk mendapatkan matriks berisi vektor-vektor eigen yang adalah komponen utama dan matriks berisi nilai-nilai eigen yang adalah varian data di sepanjang komponen utama. Kemudian, diambil n komponen utama teratas yang akan diproyeksikan. Jumlah n akan menjadi jumlah dimensi data setelah dikurangi.

Data yang diproyeksikan akan menjadi basis data utama untuk dibandingkan dengan data hasil *query* yang melalui proses yang sama dengan data lainnya. Perhitungan kemiripan gambar menggunakan perhitungan jarak Euclidean antara data *query* dengan data pada dataset. Jarak terkecil menunjukkan gambar yang paling mirip dengan gambar *query*.

BAB 3 : Arsitektur Website (Frontend dan Backend)

3.1 Frontend

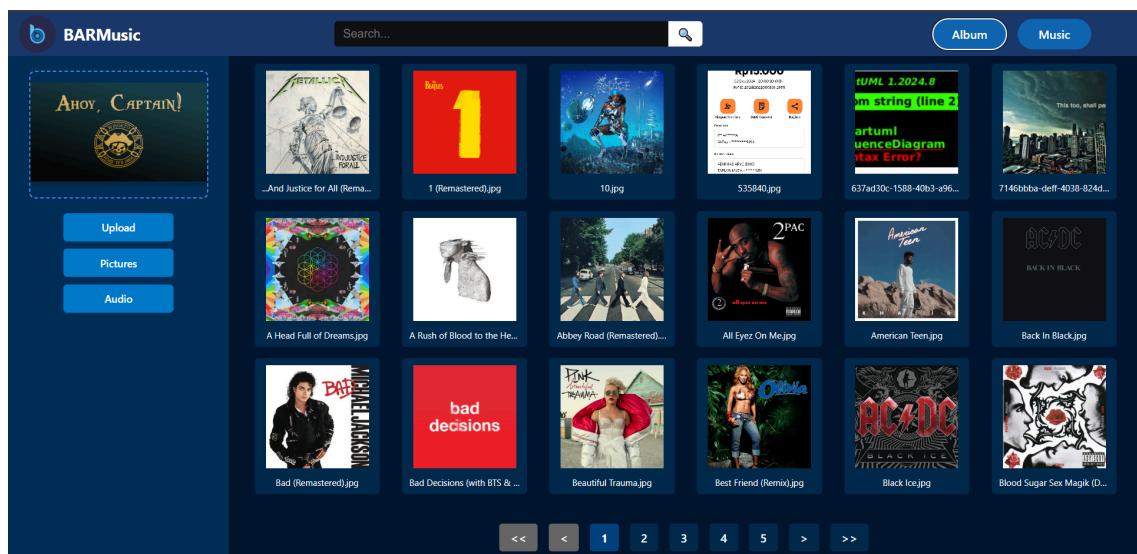
Pada arsitektur website yang kami miliki. Kelompok kami menggunakan *framework* Node.js sebagai *framework* utama kamu dalam membuat website. Dalam Node.js, kami menggunakan TypeScript dan CSS untuk melakukan styling dan juga membuat laman yang diperlukan. Berikut merupakan *folder structure* dari komponen *frontend*:



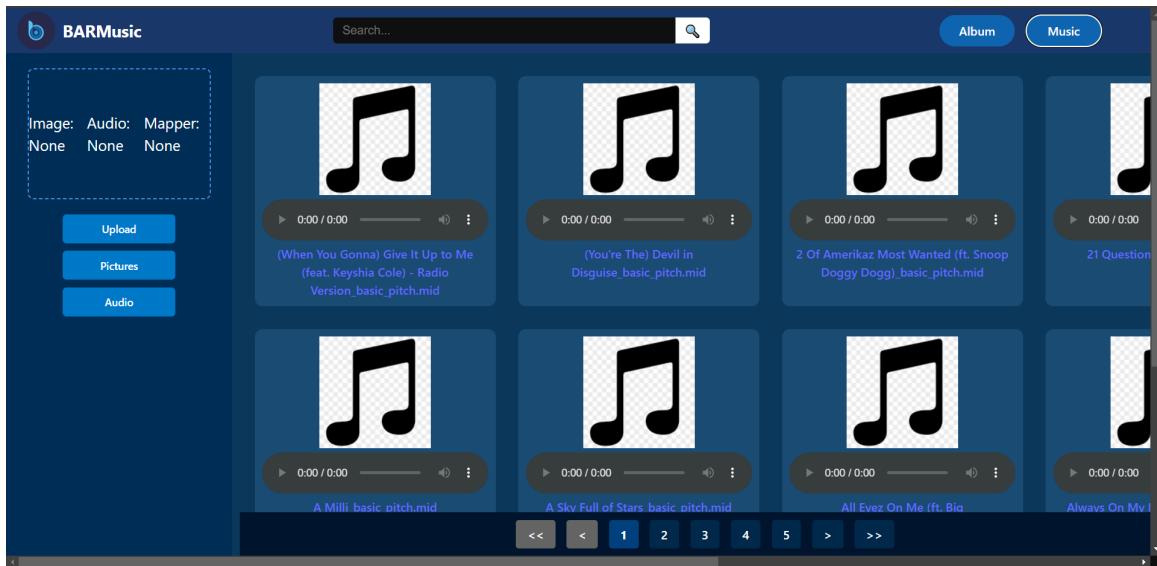
Gambar 3.1 Struktur Folder *Frontend*

Semua komponen *page* akan dimasukkan ke dalam folder *page* sesuai dengan utilities yang dibutuhkan dalam website, yakni dapat menampilkan gambar (AlbumPage), menampilkan lagu (MusicPage). Komponen seperti *navigation bar* dan *side bar* terdapat pada folder components. Komponen *navigation bar* digunakan dalam melakukan *query* untuk AlbumPage dan MusicPage, sedangkan *side bar* digunakan untuk melakukan *upload* dataset dan juga *upload* gambar/audio yang ingin dilakukan pencarian oleh pengguna.

AudioResult dan RetrievalPage merupakan laman yang digunakan untuk menampilkan *output* yang didapatkan oleh pengguna ketika mencari berdasarkan *input* pengguna. Pada laman tersebut akan ditampilkan hasil dengan tingkat kesamaan diatas 70% di dataset. Jika tidak ada data dengan tingkat tersebut akan ditampilkan 1 terbaik yang paling mendekati input pengguna. Ditampilkan juga *run time* untuk *benchmarking* efisiensi algoritma.

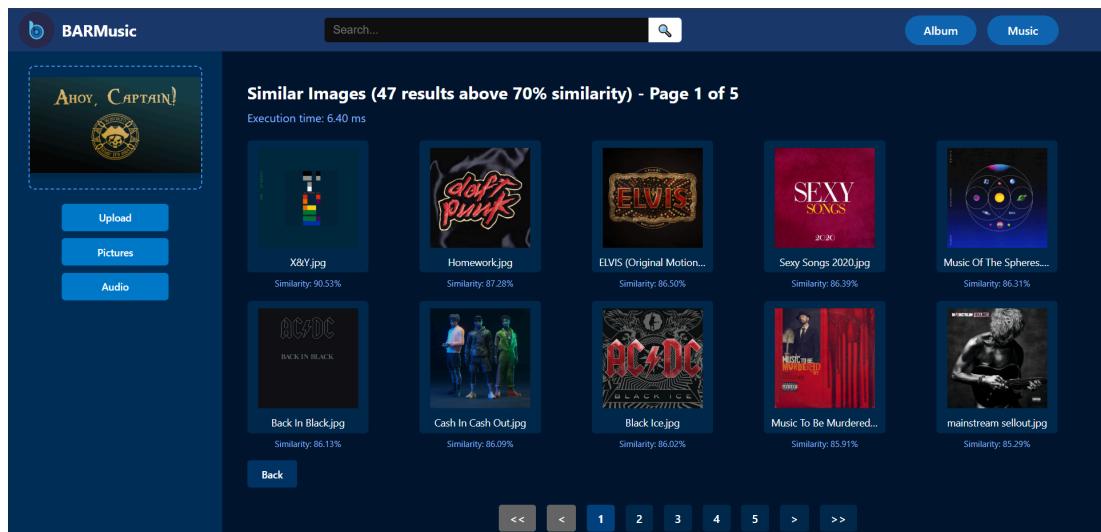


Gambar 3.2 Tampilan Website *Query Album*



Gambar 3.3 Tampilan Website *Query Lagu*

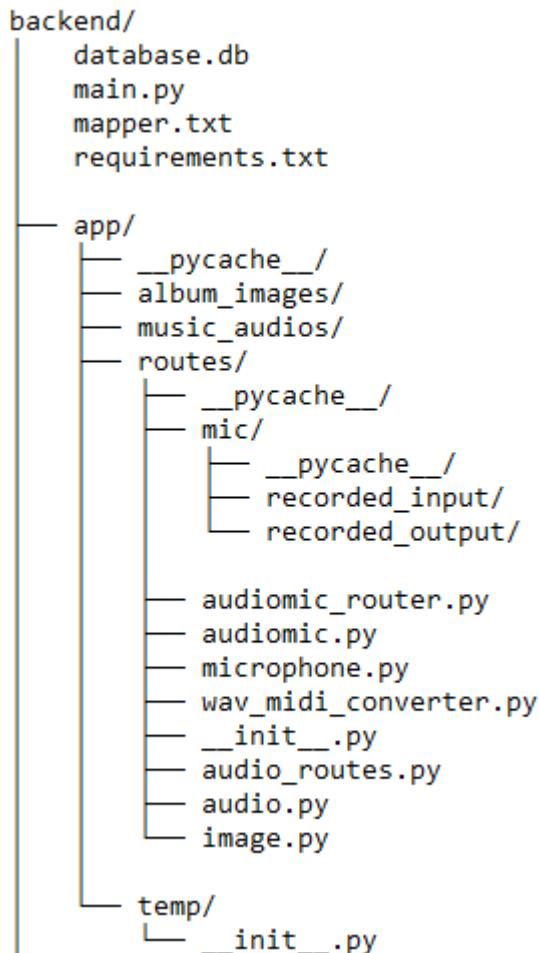
Untuk melakukan pencarian (Temu Balik Audio dan Temu Balik Gambar), pengguna dapat menekan tombol *Pictures* untuk mencari berdasarkan gambar dan tombol *Audio* untuk melakukan pencarian berdasarkan suara. Berikut adalah contoh ketika pengguna melakukan temu balik dalam website kami



Gambar 3.4 Tampilan Temu Balik Gambar

3.2 Backend

Arsitektur Backend yang kita gunakan untuk website ini adalah menggunakan bahasa python dengan *framework* FastAPI. Kami menggunakan FastAPI agar routes yang dipakai bisa lebih cepat dalam memproses pencarian gambar dan audio. Berikut merupakan *folder structure* dari komponen Backend:



Gambar 3.5 Struktur Folder Backend

3.2.1 Temu Balik Gambar

Proses pemrosesan dan pencocokan gambar dilakukan pada file `image.py` pada folder “routes”. Sesuai dengan spesifikasi, pencarian gambar dilakukan menggunakan teknik Principal Component Analysis. Alur pemrosesan gambar adalah sebagai berikut. Pertama, gambar dijadikan *grayscale* melalui fungsi `convert_grayscale` dengan mengonversikan gambar menjadi sebuah array NumPy dan menyesuaikan nilai RGB-nya. Gambar kemudian di-*resize* pada fungsi `resize_image` agar seragam dan

di-*flatten* pada fungsi *flatten_image* menjadi array satu dimensi. Setiap gambar pada database akan mengikuti prosedur ini hingga terbentuk suatu array NumPy dua dimensi berisi nilai-nilai setiap gambar.

Setelah pemrosesan awal ini, dilakukan standarisasi data pada fungsi *center_data* di sekitar nilai 0 dengan mengurangi setiap elemen gambar dengan rata-rata setiap piksel. Selanjutnya adalah perhitungan PCA pada fungsi *compute_PCA* menggunakan SVD. Dari dekomposisi tersebut, didapatkan hasil proyeksi dan komponen utama. Hasil proyeksi digunakan untuk menghitung jarak Euclidean pada fungsi *euclidean_distance*. Setelah itu, nilai kemiripan gambar disortir dengan rumus berikut.

$$e = \frac{D}{D_{\max}}$$

Dengan D adalah jarak Euclidean gambar dengan gambar *query* dan D_{\max} adalah jarak Euclidean terbesar pada dataset.

Penyambungan pemrosesan gambar dengan website terletak pada API *image-search* menggunakan FastAPI. API tersebut menerima file gambar melalui metode POST dan menyimpan file di server untuk sementara. Gambar kemudian diproses bersamaan dengan gambar-gambar pada dataset dan dibandingkan untuk mendapatkan nilai kemiripan setiap gambar dataset. Hasil pencarian mencakupi gambar-gambar yang paling mirip, nilai kemiripan masing-masing gambar, dan lama waktu pemrosesan dalam format JSON. Terakhir, tempat penyimpanan file sementara dihapuskan.

3.2.2 Temu Balik Suara

Pemrosesan temu balik suara menggunakan metode *query by humming* dilakukan pada file *audio.py*. Proses dimulai dengan mengambil melodi dari file MIDI dengan mencatat waktu dan not musik yang dimainkan. Fungsi *extract_melody* membatasi jumlah not yang diekstrak pada 1000 not untuk membuat proses lebih efisien. Setelah itu, dilakukan normalisasi melodi dan *windowing* pada fungsi *windowing* untuk membagi melodi menjadi segmen-segmen yang lebih kecil.

Proses berikutnya adalah mengekstrak nilai histogram berdasarkan fitur-fitur *Absolute Tone Based* (ATB), *Relative Tone Based* (RTB), dan *First Tone Based* (FTB). Setiap nilai histogram dinormalisasi dan dikonkatenasi menjadi suatu vektor. Vektor

tersebut akan digunakan untuk menghitung kemiripan cosinus (*cosine similarity*) dengan vektor audio *query*. Selain sebagai nilai kemiripan, nilai kemiripan cosinus digunakan untuk mengurutkan dataset audio.

Fungsi `search_similar_audio` pada route FastAPI menerima file audio yang diunggah dan menyimpannya dalam file sementara. Setelah itu, fungsi memroses file audio tersebut dan mencari kesamaan dengan file audio yang ada dalam dataset. Hasil pencarian berupa daftar file audio yang mirip beserta skor kesamaan dihitung dan dikembalikan dalam respons JSON, bersama dengan waktu eksekusi pencarian. Terakhir, tempat penyimpanan file sementara dihapuskan.

3.2.3 Microphone

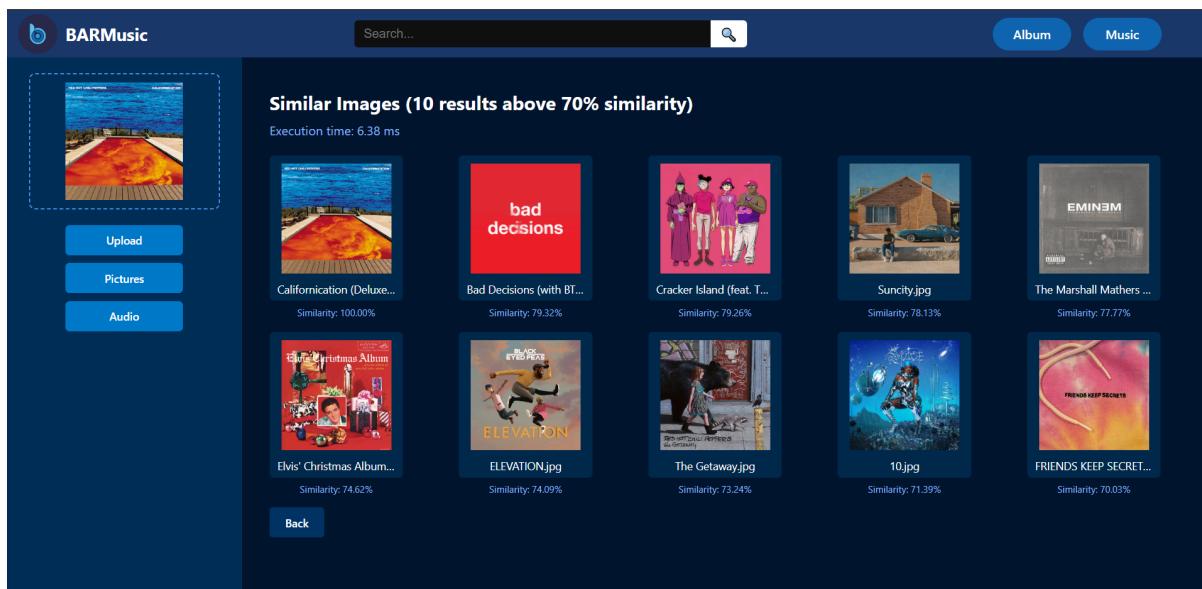
Pemrosesan audio dari microphone menggunakan interval waktu untuk mengontrol kapan rekaman audio berhenti. Ide utama algoritma untuk pemrosesan audio dengan microphone adalah untuk menangkap audio dalam jangka waktu interval yang ditentukan (20 detik), mengonversikannya menjadi format .mid, dan memrosesnya. Berdasarkan beberapa kali uji coba, kami memutuskan untuk menggunakan algoritma yang sedikit berbeda dengan algoritma temu balik suara, yakni tanpa adanya windowing untuk mengoptimisasi pencarian dan pencocokan audio.

Input microphone dapat diterima program dengan memanfaatkan library `sounddevice`. Setelah itu, audio disimpan dalam format .wav pada suatu folder. Audio yang tertangkap tersebut kemudian diproses pada file `wav_midi_converter.py` dan menyimpannya dalam folder lain. Karena algoritma pemrosesan berbeda dengan pencarian audio melalui upload file, API untuk pemrosesan dan penyambungan dipisahkan.

BAB 4: Eksperimen

4.1 Eksperimen Temu Balik Gambar

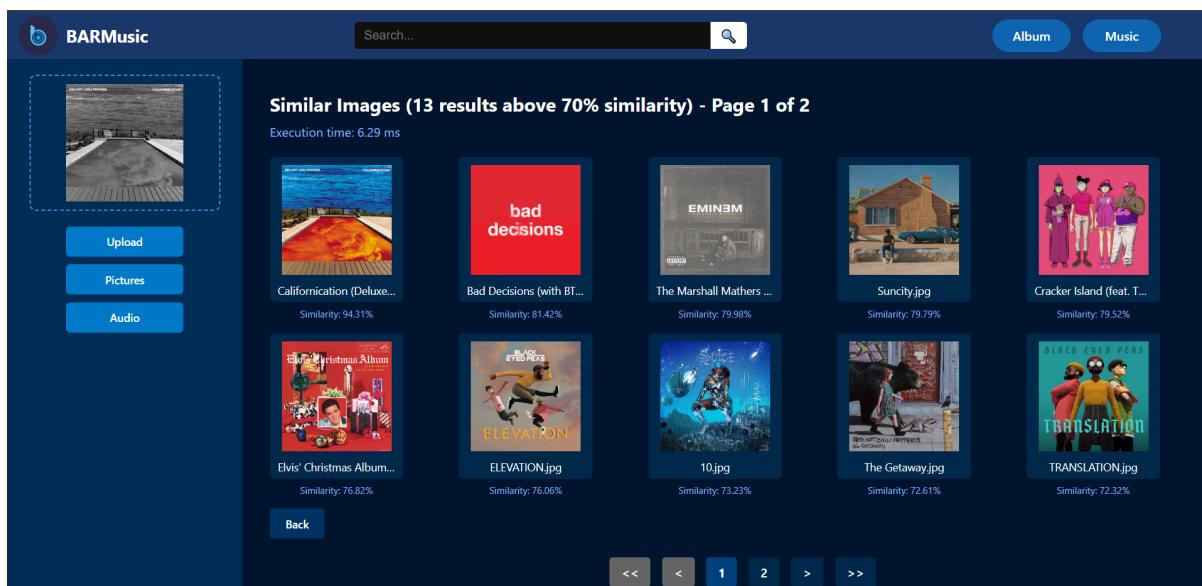
4.1.1. Temu Balik Gambar yang Ada di Dataset



Gambar 4.1 Hasil Percobaan Temu Balik Gambar A

Pada percobaan diatas dihasilkan akurasi sebesar 100%. Hal ini terjadi karena terdapat data yang sama pada dataset tersebut.

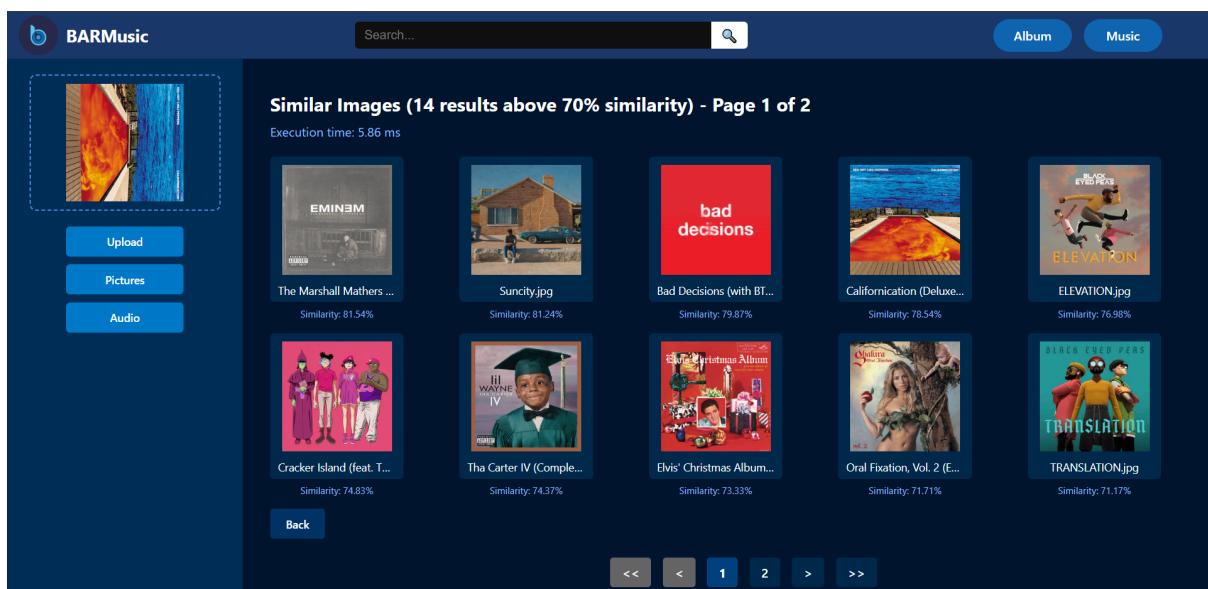
4.1.2. Temu Balik Gambar yang Ada di Dataset (Hitam-Putih)



Gambar 4.2 Hasil Percobaan Temu Balik Gambar B

Pada percobaan ini, kami menggunakan gambar yang sama, namun diubah warnanya sehingga hanya memiliki dua warna, yakni hitam dan putih (*grayscale images*). Dalam eksperimen ini dihasilkan gambar yang memiliki kemiripan 94,31%. Hasil eksperimen tetap mendapatkan akurasi yang tinggi karena melakukan *image processing* untuk *grayscale image* yang ada di dataset

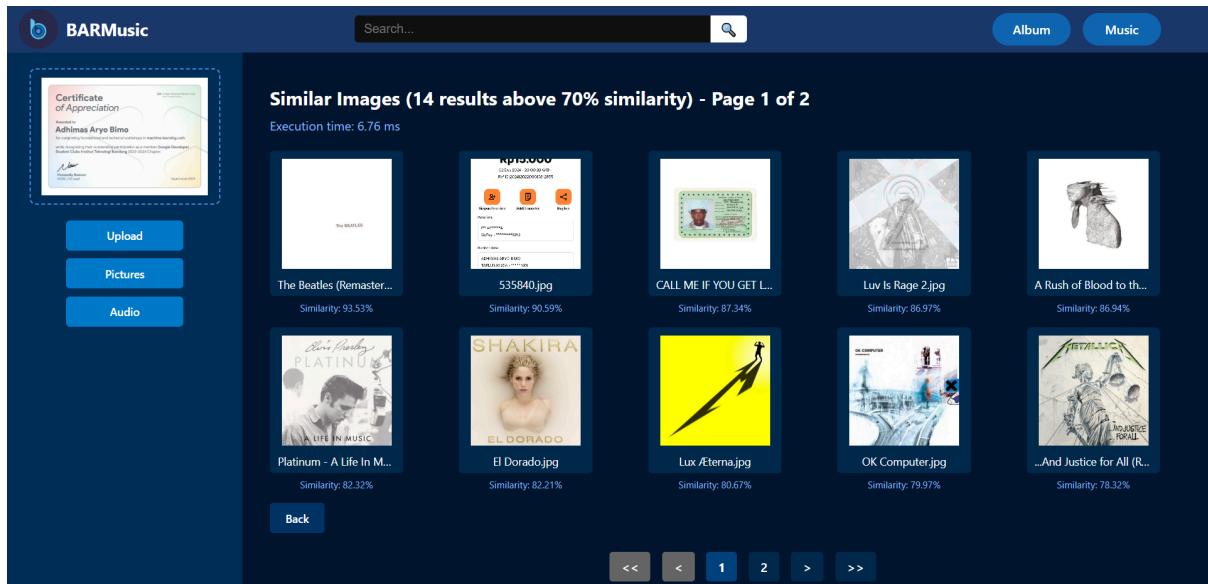
4.1.3. Temu Balik Gambar yang Ada di Dataset (Rotasi)



Gambar 4.3 Percobaan Gambar C

Presentasi kemiripan menggunakan gambar yang sama namun dirotasi memiliki nilai yang cukup jauh dibanding dengan dua percobaan sebelumnya, yakni menghasilkan nilai 78,45%. Hal ini cukup wajar karena dengan rotasi mengakibatkan perpindahan pixel warna tiap gambar. Vektor yang merepresentasikan warna tiap pixel digambar akan berbeda dengan vektor pada gambar sesungguhnya.

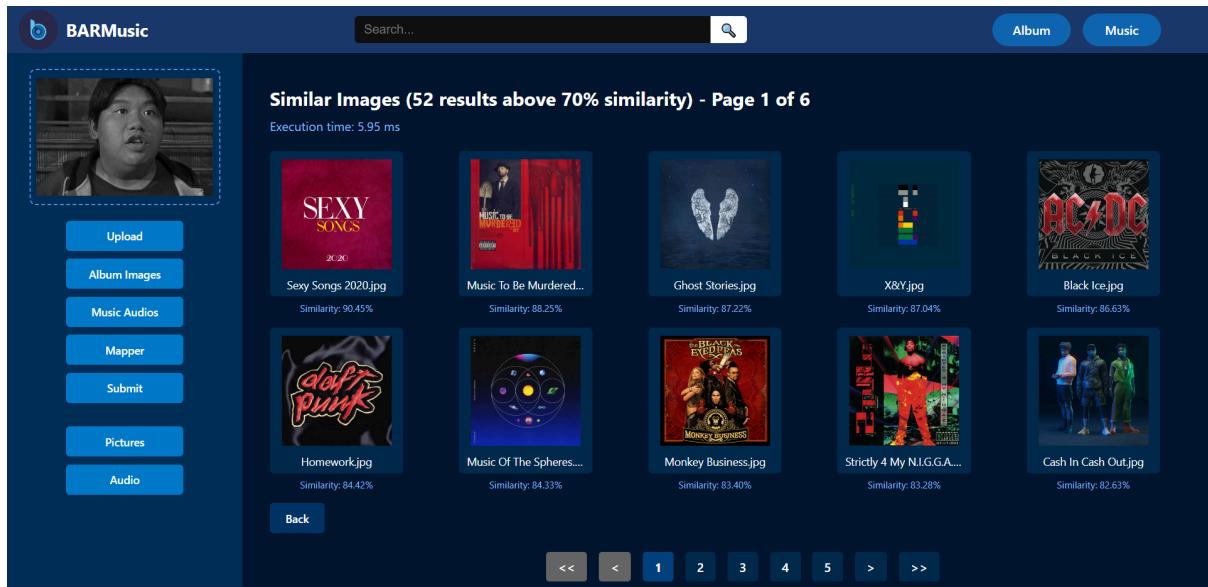
4.1.4. Temu Balik Gambar yang Tidak ada di Dataset



Gambar 4.4 Percobaan Gambar D

Pada percobaan ini dihasilkan kemiripan dengan nilai yang cukup tinggi yakni 93.5%. Gambar yang dimasukkan dominan berwarna putih. Walaupun warna tidak terlalu berpengaruh dalam proses pencarian gambar karena telah melakukan *image processing*, warna tersebut berubah representasinya menjadi terang dan gelap, bukan lagi merepresentasikan warna sesungguhnya. Representasi terang gelap pada gambar yang dimasukkan mirip dengan gambar yang memiliki kemiripan hingga 93.5% karena kedua gambar tersebut memiliki representasi terang gelap yang mirip.

4.1.5. Temu Balik Gambar yang Tidak ada di Dataset (Hitam-Putih)



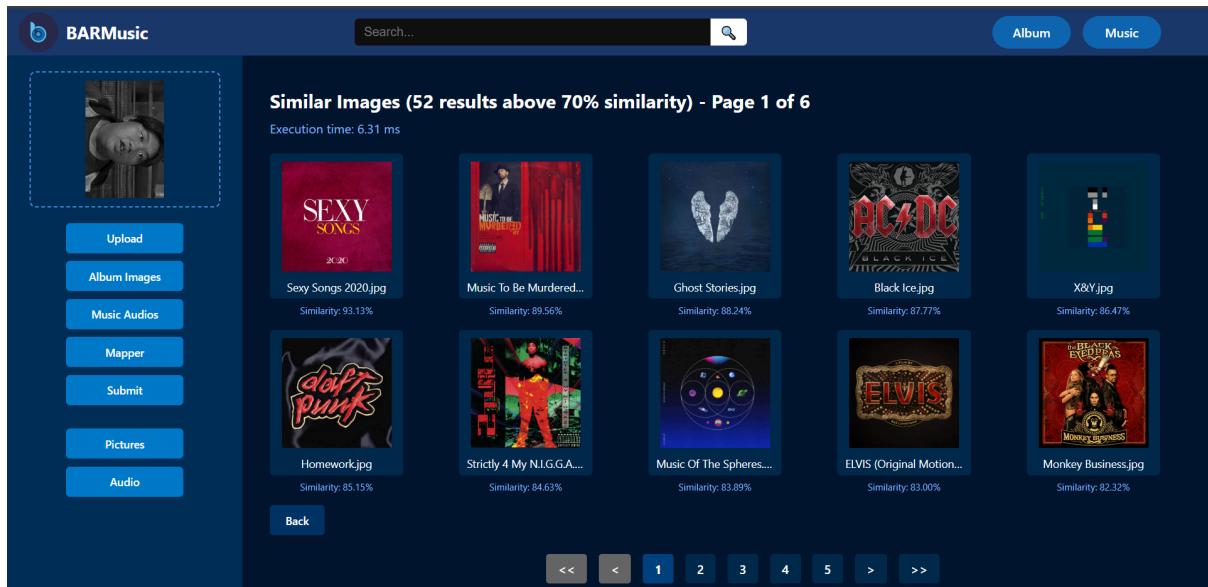
Gambar 4.5 Percobaan Gambar E

Pada percobaan ini, gambar yang dimasukkan oleh pengguna memiliki warna hitam putih dan menghasilkan kemiripan yang cukup tinggi, yakni hingga 90%. Padahal jika kita lihat secara sekilas gambar tersebut sangatlah jauh berbeda dengan gambar yang lain. Hal ini karena proses perhitungan untuk mengetahui kemiripan sebuah gambar menggunakan perhitungan euclidean (*Euclidean Distance*). Dan dalam mengukur persentase kemiripannya, kami menggunakan rumus sebagai berikut:

$$\text{similarity score \%} = 1 - \frac{D}{D_{max}}$$

Dengan D adalah nilai euclidean dari gambar dan D_{max} adalah nilai euclidean dari nilai maksimum di dataset. Nilai D_{max} bisa menjadi rancu jika terdapat nilai euclidean yang sangat jauh dari data yang dimasukkan ke dalam perhitungan. Sehingga hasil kemiripan bisa menjadi bias tergantung dari dataset yang dimiliki oleh pengguna.

4.1.6. Temu Balik Gambar yang Tidak ada di Dataset (Rotasi)

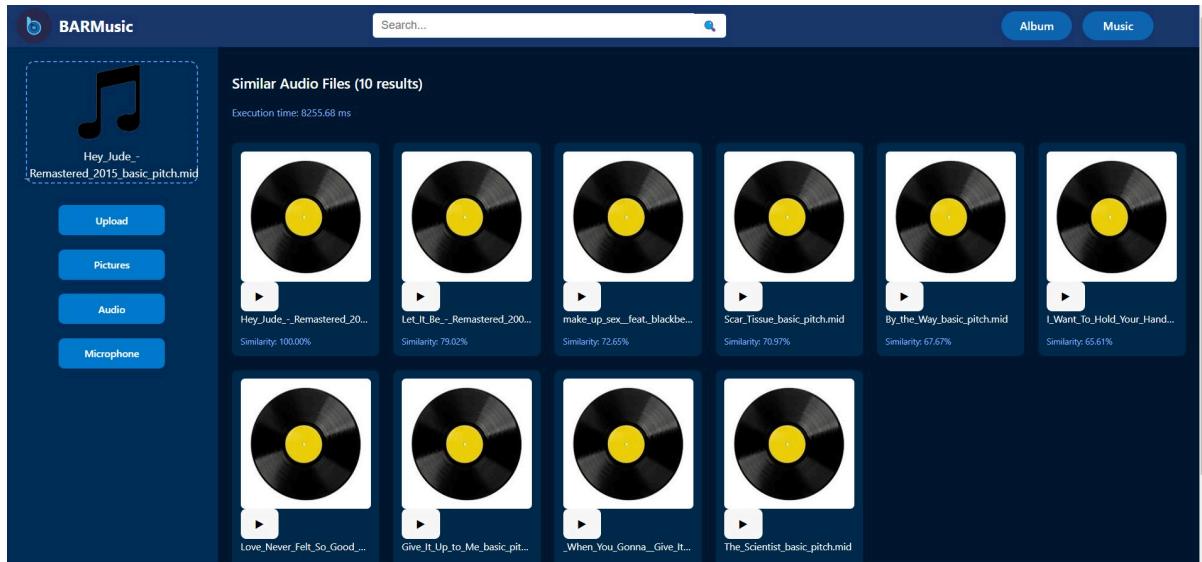


Gambar 4.6 Percobaan Gambar F

Pada percobaan ini, nilai kemiripan menggunakan gambar yang sama pada percobaan E namun dirotasi memiliki nilai kemiripan yang meningkat dibanding pada percobaan E. Hal ini memvalidasi bahwa nilai kemiripan suatu gambar sangat berpengaruh dengan gambar yang ada di dataset. Jika terdapat gambar yang sangat berbeda dengan gambar yang dimasukkan oleh pengguna, nilai Dmax akan sangat besar dan perbandingan nilai D dengan Dmax akan semakin besar juga sehingga menghasilkan nilai kemiripan yang bias. Dengan kata lain percobaan F memvalidasi percobaan E.

4.2 Eksperimen Temu Balik Audio

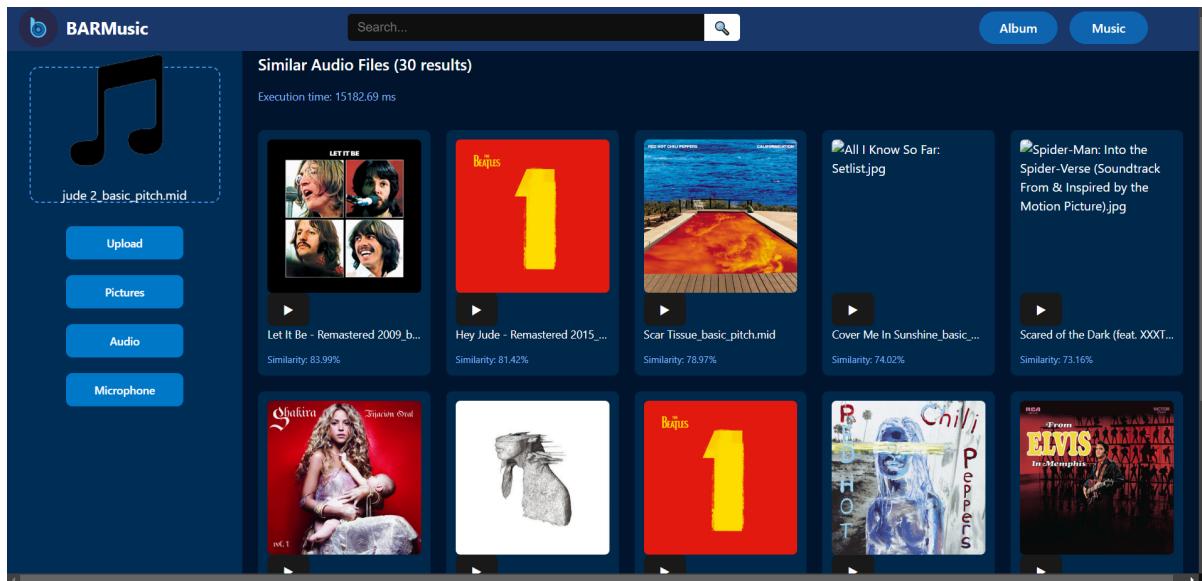
4.2.1. Temu Balik Audio yang Ada di Dataset



Gambar 4.7 Percobaan Audio A

Pada percobaan ini dihasilkan kemiripan audio sebesar 100%. Hal ini karena data yang dimasukkan oleh pengguna ada di dataset

4.2.2. Temu Balik Audio yang Tidak ada di Dataset

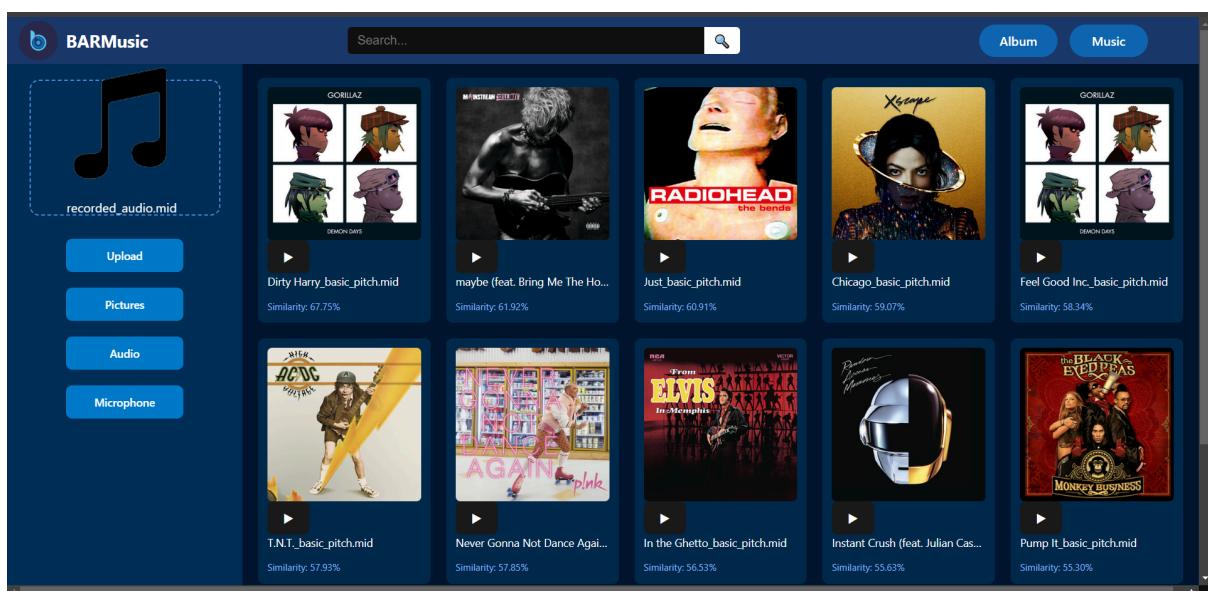


Gambar 4.8 Percobaan Audio B

Pada percobaan ini digunakan sebuah data yang tidak ada di dalam dataset. Data yang dimasukkan adalah sebuah video youtube yang bertajuk Paul McCartney - Hey Jude Live at

Hyde Park yang telah di convert dari MP4 ke WAV menjadi MIDI. Dalam percobaan ini dihasilkan similarity yang cukup menarik. Alih-alih mirip dengan lagu aslinya, yakni Hey Jude, suara ini justru lebih mirip dengan lagu Let It Be yang dinyanyikan oleh penyanyi yang sama, yakni The Beatles. Hal ini bisa terjadi karena *converting* dari WAV ke MIDI menghilangkan banyak informasi penting dari lagunya, terutama library yang dipakai dalam *converting* WAV ke MIDI yang tidak terlalu baik jika file WAV yang digunakan memiliki banyak *Noise*.

4.2.3. Temu Balik Audio dengan Microphone



Gambar 4.9 Percobaan Audio C

Pada percobaan ini, pengguna menyanyikan lagu Viva La Vida - Coldplay, namun keluaran yang dihasilkan sangat jauh dari harapan pengguna. Hal ini terjadi karena beberapa faktor, diantaranya microphone pengguna yang tidak terlalu baik sehingga menimbulkan *noise*. Hal ini mengakibatkan pengubahan format dari WAV ke MIDI berlangsung kurang baik. Hal tersebut menyebabkan sulitnya pencarian lagu menggunakan Microphone.

BAB 5 : Kesimpulan, Saran, Komentar, dan Refleksi

5.1. Kesimpulan

Algoritma PCA dan Query by Humming dapat digunakan sebagai metode untuk melakukan *Information Retrieval*. Walaupun algoritma ini merupakan algoritma yang cukup lawas dan memiliki beberapa kelemahan, algoritma ini masih mampu untuk melakukan pencarian dengan baik dan efisien dalam percobaan yang kami lakukan.

Algoritma PCA memiliki beberapa kekurangan, yakni nilai Dmax yang cenderung bisa karena bergantung pada dataset yang digunakan dan sulitnya algoritma ini dalam mencari gambar yang dirotasikan. Untuk Algoritma Query by Humming, ekstraksi WAV ke dalam format MIDI menghilangkan beberapa informasi yang kaya di WAV. Hal ini dapat memengaruhi data yang diolah dalam melakukan pencarian.

Untuk mendapatkan *information retrieval* yang lebih baik, bisa menggunakan metode yang lebih *advance* dibandingkan dengan metode ini, yakni menggunakan *machine learning*.

5.2. Saran

Pada tugas besar ini, kami merasa bahwa tugas ini cenderung lebih berat pada pengaplikasian pengembangan web dibanding pengaplikasian ilmu aljabar linier dan geometri itu sendiri. Kami berharap untuk tugas-tugas selanjutnya pengaplikasian aljabar linear dan geometri bisa lebih ditekankan seperti pada tugas besar sebelumnya.

5.3. Komentar

Tugas besar ini awalnya terasa berat karena *deadline* yang cukup singkat, yakni hanya 2 minggu saja, dan tugas besar lain yang menanti. Beruntung asisten berhati baik hingga dapat menambah tenggat tugas besar kali ini, terima kasih asisten :).

5.4. Refleksi

Pembagian tugas merupakan rancangan utama dalam mengeksekusi tugas besar. Dengan pembagian tugas yang baik, kita bisa mengeksekusi tugas besar dengan baik juga. Dan dalam kelompok kami, kami telah membagi tugas dengan baik. Rusmin yang bertanggung jawab atas *frontend*, Aria yang bertanggung jawab atas algortima Query by Humming dan Image retrieval, dan Bimo yang bertanggung jawab atas dataset dan converting data. Walaupun pengaplikasian materi Algeo porporsinya tidak sebanyak itu, namun kami tetap menjadi lebih paham mengenai pengaplikasian PCA dan juga Query by Humming dalam *Information Retrieval*.

Lampiran

Link Video Tubes 2

[https://drive.google.com/drive/folders/1bHiSWCJfS6E7iMWKeURPJ4uyfhX
MXhiZ](https://drive.google.com/drive/folders/1bHiSWCJfS6E7iMWKeURPJ4uyfhXMXhiZ)

Link Repository Github

<https://github.com/ryonlunar/Algeo02-23052>