

IF2123 Aljabar Linear dan Geometri

Image Retrieval dan Music Information Retrieval Menggunakan PCA dan Vektor

Laporan Tugas Besar 2

Disusun untuk memenuhi tugas mata kuliah Aljabar Linear dan Geometri
pada Semester 1 (satu) Tahun Akademik 2024/2025



Oleh:

Rafizan Muhammad Syawalazmi 13523034

Abdullah Farhan 13523042

Muhammad Zahran Ramadhan Ardiana 13523104

Kelompok 16 / Tolong dibawa jas hujannya

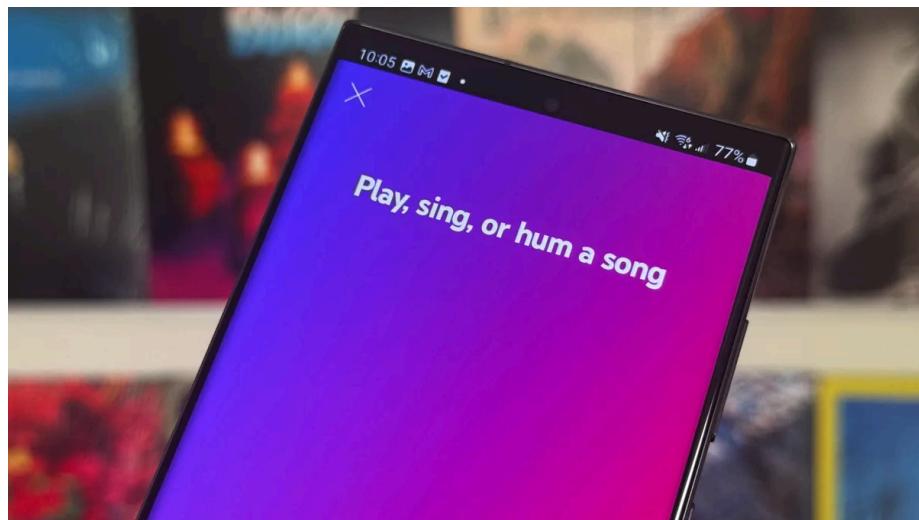
**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2023**

DAFTAR ISI

| | |
|--|-----------|
| DAFTAR ISI..... | 2 |
| BAB 1 DESKRIPSI MASALAH..... | 3 |
| BAB 2 LANDASAN TEORI..... | 7 |
| 2.1 Images Retrieval..... | 7 |
| 2.2 Music Information Retrieval..... | 11 |
| 2.3 Website..... | 14 |
| BAB 3 ARSITEKTUR WEBSITE..... | 15 |
| 3.1 Overview..... | 15 |
| 3.2 Struktur Direktori..... | 15 |
| 3.3 Frontend..... | 17 |
| 3.4 Backend..... | 19 |
| 3.4.1 Image Searching and Uploading Backend..... | 19 |
| 3.4.2 Music Information Retrieval Backend..... | 20 |
| BAB 4 UJI COBA..... | 21 |
| 4.1 Tata Cara Penggunaan Program..... | 21 |
| BAB 5..... | 28 |
| 5.1 Kesimpulan..... | 28 |
| 5.2 Saran..... | 28 |
| 5.3 Komentar..... | 29 |
| 5.4 Refleksi..... | 30 |
| DAFTAR PUSTAKA..... | 31 |
| LAMPIRAN..... | 32 |

BAB 1 DESKRIPSI MASALAH

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 pendekripsi 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 dan Youtube humming search.



Gambar 1. Contoh penerapan *audio-based information retrieval system* (ABIR) di Youtube humming search

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.

Dalam Tugas Besar 2 Aljabar Linear dan Geometri, semua mahasiswa yang terdaftar di IF2123 diminta untuk membuat semacam aplikasi Shazam yaitu sebuah aplikasi yang meminta input lagu dan aplikasi tersebut mendekripsi 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).

SPESIFIKASI TUGAS

1. Buatlah sebuah website yang bisa mendeteksi dan mencari suara dan gambar dalam dataset yang paling cocok.
2. Bahasa pemrograman yang digunakan pada frontend dan backend dibebaskan. Mohon dipertimbangkan bahasa-bahasa pemrograman yang kalian gunakan agar bahasa tersebut adalah yang menguntungkan kalian.
3. Dataset dari sebuah audio bebas. Silahkan cari di website seperti kaggle.com. Untuk dataset yang akan diujikan, akan diumumkan nantinya.
4. Program Dapat memunculkan audio dari dataset ketika sudah diunggah.
5. Program dapat mencari kecocokan dari sebuah audio query atau tangkapan dari rekaman audio dengan audio yang ada di dataset berdasarkan metode query by humming.
6. Untuk query by humming dapat menggunakan file berformat MIDI ataupun WAV. Silahkan pilih metode manapun yang dapat anda implementasikan.
7. Program dapat memberikan hasil audio / image yang memiliki tingkat kemiripan berkisar antara 55% - 100% (silahkan tentukan sesuai diskusi dari kelompok).
8. Program dapat mengimplementasikan pagination untuk menghindari adanya infinite scrolling ketika terdapat audio yang terlalu banyak.
9. Program dapat menunjukkan jumlah audio yang didapat dari jumlah semua audio dan waktu eksekusi dari program tersebut.

BONUS

Bagian bonus hanya boleh dikerjakan apabila spesifikasi wajib dari Tugas Besar telah berhasil dipenuhi. Anda tidak diharuskan untuk mengerjakan keseluruhan bonus, tetapi semakin banyak bonus yang dikerjakan, maka akan semakin banyak tambahan nilai yang diperoleh.

- **Bonus 1: Video Name Mapping**

Tidak baik jika setiap video namanya langsung dari nama filenya. Oleh karena itu anda boleh membuat sebuah file .txt atau .json. Struktur dari file .txt atau .json tersebut adalah nama file video dan nama video dari file video yang bersangkutan tersebut. Dataset dari sebuah audio tidaklah sedikit. Oleh karena itu, carilah cara untuk memberi nama dari semua audio yang ada di dataset. Nama tersebut tidaklah harus nama lagu asli (Selama nama audio tersebut adalah sebuah kalimat bukan huruf saja seperti “A”, “B”, dll).

```
Mapper.txt
-----
audio_file audio_name pic_name
audio_1.mid Billie Jean pic_1.png
audio_2.mid DJ Remix pic_2.png
audio_3.mid Gangsta Rap pic_3.png
audio_4.mid A-> (Tidak Boleh) pic_4.png
.
.
```

Nilai bonus bisa saja ditingkatkan jika anda ingin menambah detail dari sebuah audio lebih lanjut contohnya nama penyanyi, nama album, rating, dll. Silahkan gunakan imajinasi kalian untuk membuat sebuah detail audio yang menyamai aplikasi lagu seperti Spotify.

- **Bonus 2: Performance Benchmark**

Anda akan lebih dikenalkan terkait performa di mata kuliah IF2211 Strategi Algoritma (ini juga mata kuliah yang dipegang oleh Lab IRK). Namun sebelum mengambil mata kuliah tersebut, kami ingin membuat kalian setidaknya lebih aware terhadap seberapa cepat program anda dalam menyelesaikan tugasnya. Patokan dari bonus ini adalah 50 detik dengan dataset sebesar 1 GB.

- **Bonus 3: Video**

Silahkan berkreasi dalam penjelasan anda terkait MIR. Buatlah video penjelasan terkait program yang anda buat. Perlu diingat bahwa video bisa saja dilihat oleh dosen sehingga hindari adanya pelanggaran SARA dalam video anda.

- **Bonus 4: Kreativitas Tambahan**

Masih banyak kreasi yang bisa anda buat pada tugas besar ini. Pastikan kreasi terbaru yang anda buat bisa membuat asisten terpukau sehingga mereka mau mempertimbangkan untuk menambah nilai bonus kalian. Contoh bonus kreasi yang bisa kalian buat adalah audio player yang bisa play/pause, rewind, fast-forward, dan mempercepat/memperlambat sebuah audio. Ini adalah salah satu contoh, anda boleh memikirkan bonus-bonus lainnya yang menurut kalian terkesan kreatif. Anda bahkan bisa membuat sebuah aplikasi konsep vektor dengan membuat jenis query yang baru seperti query by song name, dll (tidak harus secara audio dan gambar).

- **Bonus 5: Microphone**

1. Terdapat fitur microphone yang dapat melakukan penangkapan audio secara real-time ketika program dijalankan.

2. Dilarang menambahkan button untuk trigger pada fitur microphone. HINT: Gunakan interval waktu untuk melakukan penangkapan suara.
3. Fitur microphone merupakan fitur tambahan, sehingga fitur utama upload audio melalui website tetap harus ada.

BAB 2 LANDASAN TEORI

2.1 Images Retrieval

Images retrieval adalah suatu metode yang digunakan untuk mencari dan mengambil citra digital dari basis data berdasarkan kriteria tertentu. Proses ini dapat dilakukan menggunakan Content-Based Image Retrieval (CBIR), yang berfokus pada karakteristik visual citra seperti warna, tekstur, dan bentuk. Tahap awal dalam CBIR melibatkan ekstraksi fitur kunci dari citra, di mana fitur-fitur ini direpresentasikan dalam bentuk vektor numerik. Vektor tersebut memungkinkan perbandingan antar citra dalam dataset untuk mengidentifikasi citra yang paling relevan berdasarkan tingkat kemiripannya.

Dalam penerapan Principal Component Analysis (PCA) pada sistem pencarian citra, metode ini digunakan untuk mereduksi dimensi vektor fitur dari citra. Reduksi dimensi sangat penting karena fitur citra yang diekstraksi seringkali memiliki jumlah komponen yang besar, yang menyebabkan perhitungan menjadi lebih kompleks dan memakan waktu. PCA membantu menyederhanakan data tanpa menghilangkan informasi penting, sehingga proses perbandingan citra menjadi lebih efisien. Pencarian dengan menggunakan citra memberikan kemudahan bagi pengguna untuk menjelajahi koleksi citra tanpa bergantung pada pencarian berbasis teks, melainkan melalui kesamaan visual antar citra. Tahapan utama dalam penerapan PCA pada images retrieval adalah:

1. Image Processing and Loading

Tahap awal dimulai dengan mengubah seluruh gambar pada dataset ke dalam format grayscale untuk menyederhanakan informasi visual. Grayscale hanya mempertimbangkan intensitas pixel berdasarkan rumus:

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

di mana $R(x, y)$, $G(x, y)$, $B(x, y)$ adalah nilai intensitas warna merah, hijau, dan biru pada koordinat pixel (x,y) . Sehingga yang awalnya suatu gambar memiliki *array* 3D dengan nilai di suatu pixel (x,y) berbentuk sebuah *array of RGB*: [R,G,B] dengan rentang nilai masing-masing [0, 255] menjadi *array* 2D yang suatu pixelnya hanya mengandung 1 nilai saja dengan rentang yang sama.



Figure 1



Gambar 2.1 Gambar hasil penyederhanaan intensitas warna

Selanjutnya, semua gambar diubah ukurannya menjadi seragam untuk memastikan konsistensi dalam perhitungan. Gambar tersebut kemudian direpresentasikan sebagai vektor satu dimensi (1D) atau menjadi *flatten*. Jika gambar memiliki ukuran $M \times N$, maka hasilnya adalah vektor dengan panjang $M \cdot N$:

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

2. Data Centering (Standardization)

Langkah ini bertujuan untuk memusatkan data di sekitar nilai nol dengan menghitung rata-rata pixel untuk setiap posisi pixel tertentu di seluruh dataset. Rata-rata ini dihitung menggunakan:

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

di mana x_{ij} adalah nilai pixel ke- j dari gambar ke- i , dan N adalah jumlah gambar dalam dataset. Setiap nilai pixel kemudian distandarisasi dengan mengurangkan rata-rata yang telah dihitung:

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

3. PCA Computation Using Singular Value Decomposition (SVD)

Setelah standarisasi, PCA diterapkan dengan menghitung matriks kovarian dari data yang sudah distandarisasi. Matriks kovarian dapat dihitung sebagai:

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

di mana X' adalah matriks data yang sudah distandarisasi. Matriks kovarian ini menggambarkan relasi antara pixel dalam gambar tersebut. Jika dalam suatu dataset ukuran gambar adalah $M \times N$ maka ukuran matriks kovarian yang terbentuk akan menjadi $M \cdot N \times M \cdot N$. Selanjutnya, Singular Value Decomposition (SVD) digunakan untuk mendekomposisi matriks kovarian:

$$X' = U \Sigma V^T$$

U : Matriks eigenvector yang mewakili komponen utama.

Σ : Matriks diagonal yang berisi nilai eigen (eigenvalue), yang menunjukkan variasi data sepanjang komponen utama.

Dari hasil SVD, komponen utama dipilih berdasarkan nilai eigen terbesar, yang menunjukkan kontribusi variasi tertinggi dalam data. Jika k adalah jumlah komponen utama yang dipilih, maka proyeksi data ke ruang komponen utama adalah:

$$Z = X' U_k$$

di mana U_k adalah matriks eigenvector dengan dimensi k . Sehingga jika matriks U berukuran $N \times N$ maka U_k akan berukuran $N \times k$

4. Similarity Computation

Similarity Computation adalah tahap krusial dalam image retrieval menggunakan PCA. Tujuannya adalah mengukur tingkat kemiripan antara gambar query (input) dengan gambar-gambar yang ada di dataset. Pada tahap ini, perhitungan dilakukan di ruang komponen utama hasil reduksi dimensi PCA. Salah satu metode yang umum digunakan adalah Jarak Euclidean (Euclidean Distance), yang mengukur seberapa "jauh" dua vektor dalam ruang n-dimensi. Setelah PCA diaplikasikan, setiap gambar dalam dataset direpresentasikan sebagai vektor proyeksi dalam ruang komponen utama. Gambar query (q) juga harus direpresentasikan dengan cara yang sama agar dapat dibandingkan.

Vektor proyeksi untuk gambar query diperoleh menggunakan rumus:

$$q = (q' - \mu) U_k$$

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 pixel dari dataset

U_k : Matriks eigenvector dengan k dimensi utama dari PCA.

Vektor proyeksi q dari gambar query ini memiliki dimensi yang sama dengan vektor proyeksi dari gambar-gambar dalam dataset.

Untuk mengukur kemiripan antara gambar query dan gambar dalam dataset, digunakan **jarak Euclidean**. Jarak ini didefinisikan sebagai ukuran linear dari "perbedaan" antara dua vektor dalam ruang komponen utama. Rumus jarak Euclidean adalah:

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

$d(q, z_i)$: Jarak Euclidean antara gambar query q dan gambar ke- i di dataset z_i

q_j : Elemen ke- j dari vektor proyeksi gambar query.

z_{ij} : Elemen ke- j dari vektor proyeksi gambar ke- i di dataset.

k : Jumlah komponen utama yang dipilih (dimensi ruang PCA)

5. Retrieval and Output

Semakin kecil nilai $d(q, z_i)$, semakin mirip gambar query q dengan gambar z_i . Gambar-gambar dalam dataset akan diurutkan berdasarkan jarak d , dari yang paling kecil hingga yang terbesar dan akan dilakukan limitasi jumlah dengan memberikan batas persentase kemiripan sehingga gambar dengan jarak terkecil akan menjadi hasil retrieval yang paling relevan.

Referensi:

- Smith, J. R., & Chang, S. F. (1997). "VisualSEEK: A Fully Automated Content-Based Image Query System." Dalam *Proceedings of ICIP 1997*. Artikel ini memperkenalkan sistem CBIR berbasis pola spasial, yang memanfaatkan fitur visual seperti warna dan tekstur untuk pencarian otomatis.

- Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). "Image Retrieval: Ideas, Influences, and Trends of the New Age." *ACM Computing Surveys*. Artikel ini membahas tantangan utama dalam CBIR, termasuk semantic gap, dan solusi seperti penggunaan feedback relevansi untuk meningkatkan akurasi hasil pencarian.
- Zhou, X. S., & Huang, T. S. (2002). "Relevance Feedback in Image Retrieval: A Comprehensive Review." *ACM Multimedia Systems*. Artikel ini mengulas pentingnya feedback pengguna dalam proses CBIR untuk menyempurnakan hasil pencarian berdasarkan kebutuhan spesifik pengguna.

2.2 Music Information Retrieval

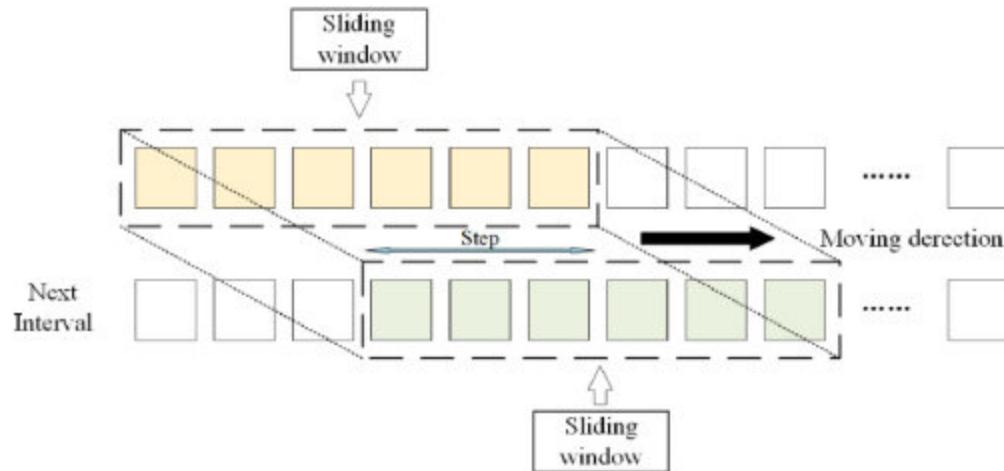
Music information retrieval adalah sebuah metode untuk mencari lagu menggunakan audio hasil bernyanyi atau bersenandung berdasarkan fitur-fitur yang dimilikinya. MIR dapat mencari lagu yang memiliki segmen yang mirip dan mengurutkannya sesuai dengan kemiripan tersebut.

MIR melakukan perbandingan menggunakan melodi dalam sebuah musik. Hal ini karena manusia akan lebih mengenal bagian melodi dari suatu musik dibanding dengan suara lainnya. Pertama, MIR akan mengekstrak data nada pada semua musik di dalam database dan musik yang ingin dicari. Kemudian, data nada dibagi menjadi banyak sub data menggunakan metode *sliding window*. Sub-sub data ini selanjutnya diubah menjadi vektor fitur numerik agar dapat dianalisis.

Similaritas didapatkan dari membandingkan vektor fitur dari musik yang ingin dicari dengan setiap musik di dalam database. Nilai akhir dikalkulasi dengan penjumlahan berbeban dari perbandingan masing-masing vektor fitur pada setiap segmen musik dan diambil yang paling tinggi.

1. Pemrosesan Audio

Pemrosesan audio diawali dengan mengekstrak setiap nada dalam audio berformat MIDI. Audio berformat lainnya seperti MP3 dan WAV perlu dikonversi terlebih dahulu menjadi MIDI. Pemrosesan nada dibagi menjadi segmen lagu lebih kecil menggunakan metode *sliding window* dengan besar *window* 20 *beat* dan pergeseran 4 *beat*. Hal ini untuk memastikan pengguna lebih fleksibel dan tidak harus konsisten dalam memberikan audio yang ingin dicari.



Gambar 2.2 Gambar Visualisasi Metode Sliding Window

Segmen-semen ini untuk sekarang tidak perlu dinormalisasikan karena tidak sesuai untuk digunakan pada ekstraksi fitur nanti.

2. Ekstraksi Fitur

Masing-masing segmen selanjutnya akan diekstraksi fiturnya menjadi vektor-numerik. Terdapat 3 pendekatan dalam mengekstraksi fitur dari melodi.

- Fitur Absolute Tone Based (ATB)**, merupakan fitur menghitung frekuensi kemunculan nada dalam skala MIDI 0–127. ATB berupa vektor histogram yang *bins*-nya berjumlah 128. Nilai-nilainya diambil dari frekuensi seluruh nada.
- Fitur Relative Tone Based (RTB)**, merupakan fitur menghitung perubahan nada antara nada bersebelahan. RTB berupa vektor histogram yang *bins*-nya berjumlah 255 dengan representasi nilai dari -127 sampai dengan 127. Nilai-nilainya diambil dari selisih nada dengan nada sebelumnya.
- Fitur First Tone Based (FTB)**, merupakan fitur menghitung perbedaan nada relatif terhadap nada pertama. FTB berupa vektor histogram yang *bins*-nya berjumlah 255 dengan representasi nilai dari -127 sampai dengan 127. Nilai-nilainya diambil dari selisih nada dengan nada pertama.

Setelah didapatkan vektor-vektor fitur di atas, masing-masing histogram dinormalisasikan menggunakan rumus:

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

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

Dengan melakukan normalisasi ini, hasil vektor histogram akan berada dalam skala probabilitas.

3. Penghitungan Similaritas

Penghitungan similaritas antar lagu menggunakan formula *cosine similarity*. *Cosine similarity* merupakan formula untuk menghitung kedekatan dari dua buah vektor pada ruang berdimensi tinggi dengan menghitung besar sudutnya. Semakin kecil sudutnya, semakin *similar* kedua buah vektor.

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

Gambar 2.3 Gambar Formula *Cosine Similarity*

Pada aplikasi ini, setiap fitur dihitung *similarity*-nya dengan fitur yang sama. Hasil skor similaritas dari kedua vektor akan berupa penjumlahan ketiga fitur dengan formula:

$$S = 0.2 * A + 0.4 * B + 0.4 * C$$

Dengan S adalah skor similaritas, A, B, dan C adalah hasil *cosine similarity* dari fitur ATB, RTB, dan FTB berturut-turut.

Nilai similaritas antar kedua lagu didapatkan dari nilai skor similaritas antar segmen-semen dari kedua lagu paling tinggi.

2.3 Website

Website adalah kumpulan halaman digital yang dapat diakses melalui internet, menyajikan teks, gambar, audio, atau video. Website berfungsi sebagai medium digital untuk berbagi informasi, menyampaikan pesan, atau menyediakan layanan, baik bagi individu maupun organisasi. Dalam konteks tertentu, website juga dapat menjadi alat pemasaran, platform bisnis, atau sarana interaksi dengan pengguna. Struktur dasarnya terdiri dari halaman-halaman yang saling terhubung dan diakses melalui peramban web. Kontennya mencakup elemen seperti teks informatif, grafik, hingga aplikasi web interaktif. Seiring kemajuan teknologi, website kini tidak hanya menjadi sumber informasi, tetapi juga medium dinamis yang mendukung interaksi dua arah antara pengguna dan penyedia konten.

Pengembangan Website Pencarian Lagu Berbasis CBIR dan MIR

Website untuk pencarian lagu berbasis audio atau gambar cover album memanfaatkan teknologi Content-Based Image Retrieval (CBIR) dan Music Information Retrieval (MIR). MIR dan CBIR memungkinkan pengguna mencari lagu dengan input berupa audio humming atau visual (seperti cover album) menggunakan fitur-fitur utama dari konten tersebut.

Dalam CBIR, fitur penting seperti pola visual diekstraksi dan direpresentasikan sebagai vektor di ruang fitur menggunakan teknik seperti transformasi Fourier atau ekstraksi histogram warna. Pendekatan ini memanfaatkan konsep Aljabar Linear, seperti transformasi matriks dan proyeksi, untuk mengidentifikasi kemiripan antara data input pengguna dan data dalam basis data.

Pencarian Berbasis MIR

MIR memanfaatkan analisis struktur musik dan metadata untuk menemukan lagu berdasarkan pola musik tertentu.

- Fitur yang diekstraksi adalah pitch/nada..
- Teknologi ini mendukung pencarian berbasis humming atau melodi sederhana dengan mencocokkan pola input dengan database lagu menggunakan algoritma cosine similarity.

Pencarian Berbasis CBIR

CBIR digunakan untuk mengenali lagu dari gambar cover album.

- Sistem mengekstraksi fitur visual, seperti histogram warna, atau bentuk geometris.
- Representasi fitur ini dianalisis di ruang fitur menggunakan teknik seperti Principal Component Analysis (PCA) untuk mempermudah pencocokan.

BAB 3 ARSITEKTUR WEBSITE

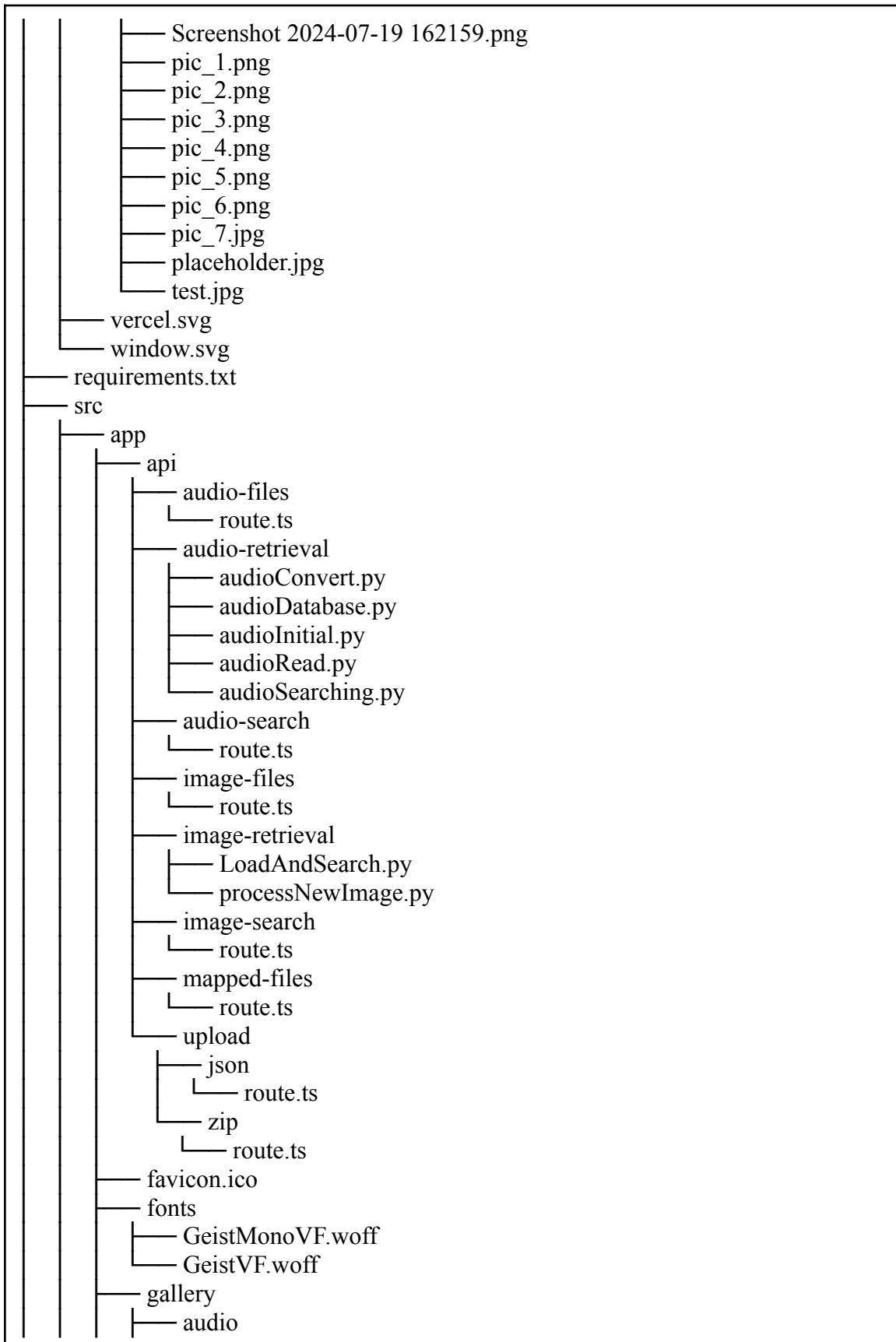
3.1 Overview

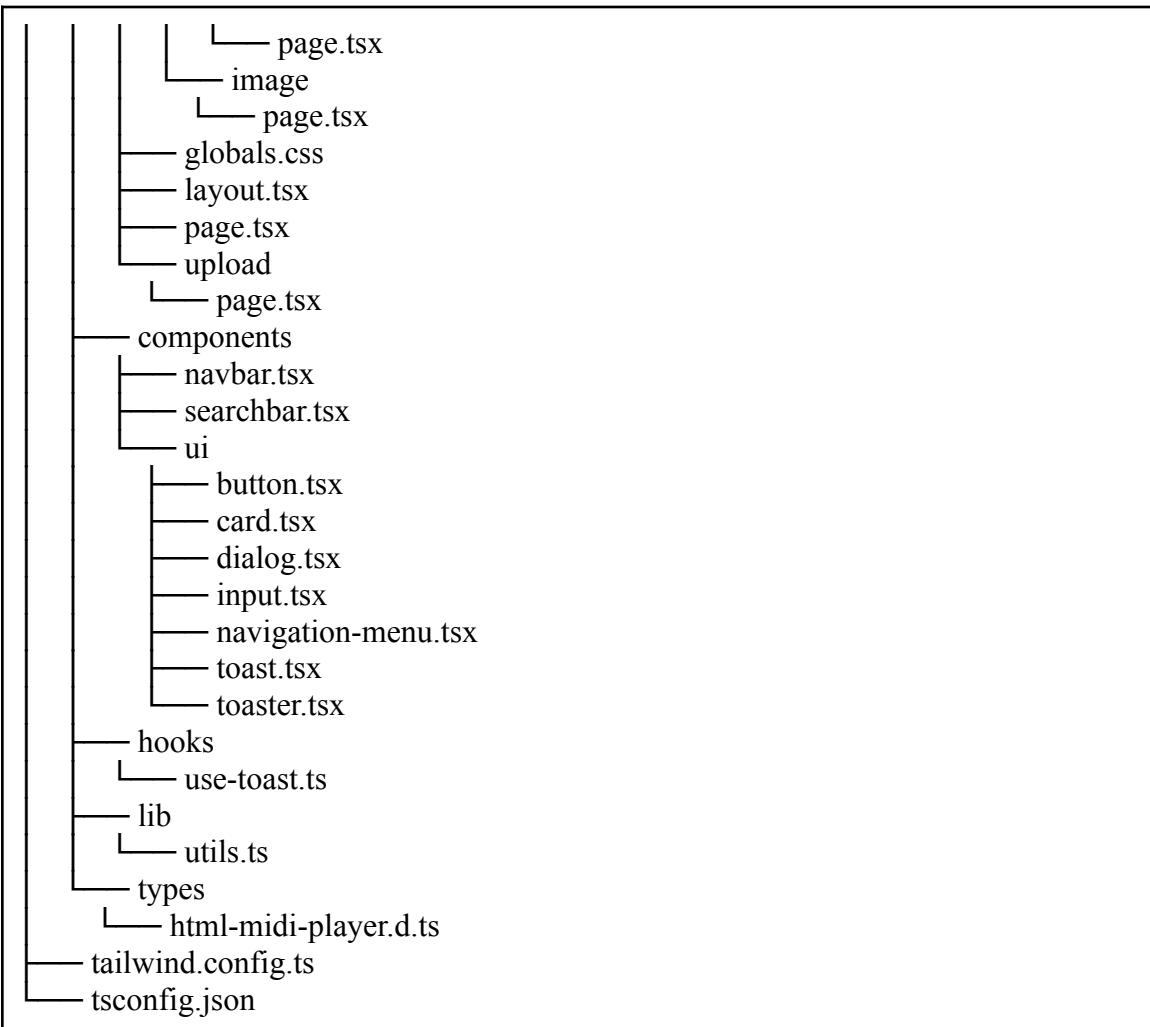
Dalam tugas besar kali ini, kami membangun aplikasi menggunakan Next.js sebagai framework utama dan dengan bantuan tailwind untuk pengembangan frontend. Aplikasi ini memiliki struktur yang menggabungkan frontend dan backend dalam satu proyek, dengan folder **src/app/api** berfungsi sebagai backend, sedangkan folder **src/app/, src/app/gallery/image, dan src/app/gallery/audio** berisi halaman-halaman dari website kami.

Dalam prakteknya, kami mengembangkan backend menggunakan Python untuk menangani logika pemrosesan data dan interaksi dengan sistem. Backend kemudian akan dieksekusi melalui API yang disediakan oleh aplikasi Next.js. Ketika frontend mengirimkan permintaan (request) ke API, API tersebut akan mengeksekusi kode Python yang relevan dan mengembalikan hasilnya dalam bentuk JSON. Data JSON ini kemudian diterima oleh frontend untuk ditampilkan kepada pengguna.

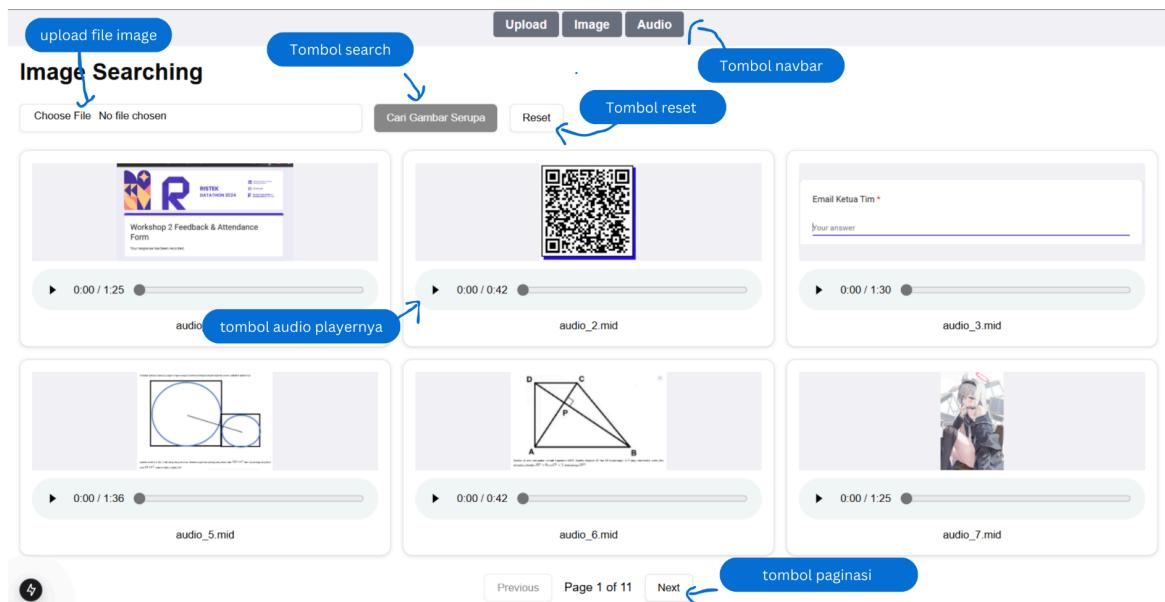
3.2 Struktur Direktori

```
Algeo02-23034
├── .eslintrc.json
├── .gitignore
├── README.md
├── components.json
└── database
    ├── audioDataset.pkl
    ├── imgDataset.txt
    └── imgName.txt
├── next.config.ts
├── package-lock.json
├── package.json
├── postcss.config.mjs
└── public
    ├── file.svg
    ├── globe.svg
    ├── mapper.json
    ├── next.svg
    ├── query
    │   └── audio
    │       ├── input.mp3
    │       └── input_basic_pitch.mid
    └── uploads
        └── image
            └── Aoi.jpg
```

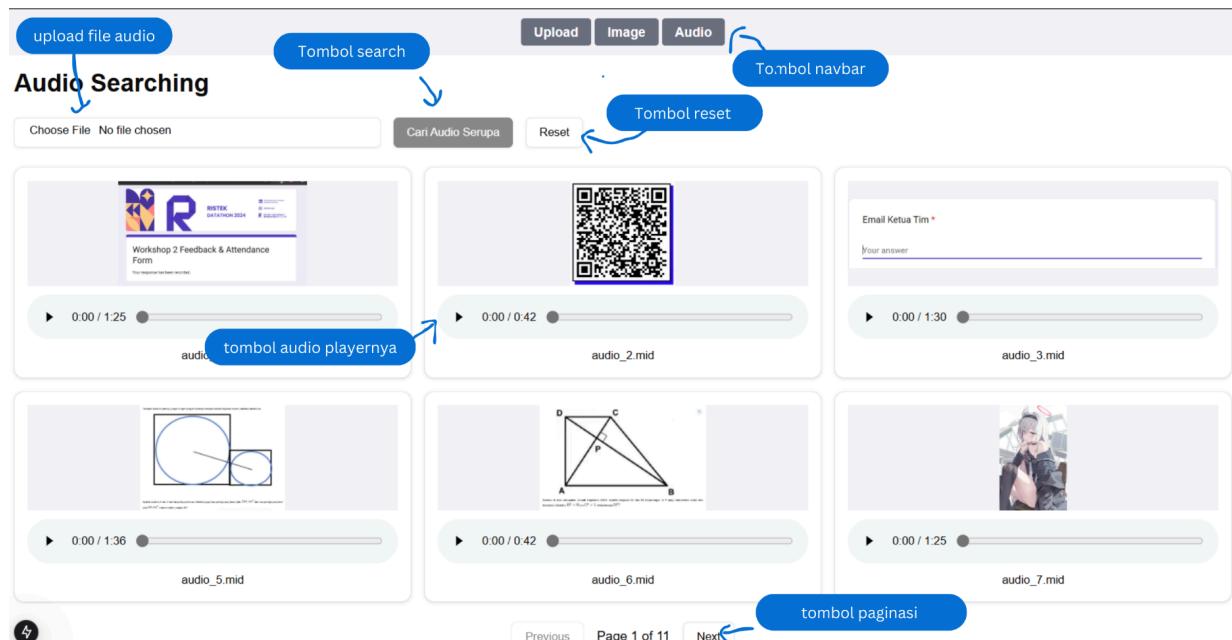




3.3 Frontend



Gambar 3.1 Tampilan UI untuk pencarian image



Gambar 3.2 Tampilan UI untuk pencarian audio

3.4 Backend

3.4.1 Image Searching and Uploading Backend

Melakukan pencarian berdasar gambar

Pada langkah pertama, program menerima input data dari pengguna melalui antarmuka web yang interaktif dan mudah digunakan. Pengguna dapat mengunggah gambar query untuk menemukan gambar dengan kemiripan tertentu di dalam data set. Input gambar ini akan diterima pada halaman page.tsx, yang kemudian memanggil API api/image-files untuk melanjutkan proses pencarian ke backend. API tersebut akan meneruskan nama atau path file gambar yang diunggah ke program Python utama (LoadAndSearch.py), yang bertugas melakukan pencarian.

Setelah pengguna mengunggah gambar, gambar yang diterima akan dikonversi menjadi format digital yang dapat diproses, seperti array matriks piksel dengan ukuran yang sudah ditentukan untuk dioperasikan. Validasi format file dilakukan saat di interface website dimana hanya bisa format yang sesuai saja yang bisa diinput. Gambar yang valid akan di-normalisasi, seperti konversi ke skala abu-abu dan di-*flatten* menjadi array 1D.

Setelah query gambar diubah, program akan membandingkan gambar yang diunggah oleh pengguna dengan setiap gambar dalam basis data menggunakan metode PCA sesuai pada bab kajian teori. Gambar yang memiliki kemiripan tertinggi dengan gambar yang diunggah oleh pengguna akan ditampilkan kepada pengguna sebagai hasil pencarian

Upload gambar untuk disimpan ke database

Fitur upload gambar (ZIP) memungkinkan pengguna untuk mengunggah kumpulan gambar melalui antarmuka web. Pada halaman utama, pengguna dapat memilih file ZIP yang berisi gambar-gambar yang ingin ditambahkan ke dalam database.

Pengguna mengunggah file ZIP dari halaman page.tsx. Setelah file dipilih, halaman ini memanggil endpoint API upload/zip/route.ts, yang bertugas menangani upload dan menyimpan sementara file gambar dari ZIP tersebut ke dalam folder temp di server. Folder temp ini berfungsi sebagai lokasi sementara sebelum gambar diproses lebih lanjut. Setelah gambar tersimpan di folder temp, sistem langsung memanggil backend untuk memulai pengolahan. Backend menjalankan skrip Python utama bernama processNewImage.py, yang memiliki fungsi inti processImageFolder. Fungsi ini bertugas mengekstraksi semua file gambar dalam folder temp dibaca dan diubah ke format grayscale untuk menyederhanakan data lalu di-flatten atau konversi menjadi array 1D.

Array flatten dari masing-masing gambar ditambahkan ke database yang sudah berisi kumpulan array dari gambar-gambar sebelumnya. Database ini dirancang untuk mempercepat proses pencarian gambar di masa mendatang. Setelah seluruh gambar dalam folder temp selesai diproses dan ditambahkan ke database, folder temp secara otomatis dihapus. Langkah ini dilakukan untuk menjaga efisiensi penggunaan ruang penyimpanan di server.

3.4.2 Music Information Retrieval Backend

Melakukan pencarian berdasar musik

Pada langkah awal, program menerima data dari pengguna melalui antarmuka web yang dirancang interaktif dan mudah digunakan. Pengguna dapat mengunggah file audio query untuk mencari musik dengan kemiripan tertentu dalam dataset. File audio yang diunggah akan diterima pada halaman page.tsx, yang kemudian memanggil API api/audio-files untuk meneruskan proses pencarian ke backend. API tersebut akan mengirimkan nama atau path file audio yang diunggah ke program Python audioInitial.py, yang bertugas mengonversi file audio non-MIDI menjadi file audio berformat MIDI. Setelah itu, API akan kembali meneruskan nama atau path file tersebut ke program Python utama, audioSearching.py.

Setelah audio diterima, file tersebut akan dikonversi menjadi vektor fitur yang telah dinormalisasi dan siap untuk dianalisis. Program kemudian membandingkan vektor query audio dengan semua data audio dalam basis data menggunakan metode cosine similarity, sebagaimana dijelaskan dalam kajian teori. Hasil pencarian berupa audio dengan tingkat kemiripan tertinggi akan ditampilkan kepada pengguna.

Upload gambar untuk disimpan ke database

Fitur unggah gambar (ZIP) memungkinkan pengguna untuk mengunggah kumpulan gambar melalui antarmuka web. Pada halaman utama, pengguna dapat memilih file ZIP yang berisi audio yang akan ditambahkan ke database.

Pengguna mengunggah file ZIP melalui halaman page.tsx, yang kemudian memanggil endpoint API upload/zip/route.ts. Endpoint ini bertanggung jawab untuk menangani proses unggah dan menyimpan file audio dari ZIP ke folder audio di server. Setelah file tersimpan, sistem secara otomatis memanggil backend untuk memulai proses pengolahan. Backend menjalankan script Python utama, audioDatabase.py, yang menggunakan fungsi utama bernama process_database. Fungsi ini bertugas mengekstraksi semua file audio dari folder, mengonversinya

menjadi vektor fitur pada setiap segmen audio, dan mengubah file non-MIDI menjadi format MIDI.

Vektor dari setiap gambar disimpan dalam file .pkl menggunakan library Python. Database ini dirancang untuk mempercepat proses pencarian gambar di masa mendatang.

BAB 4 UJI COBA

4.1 Tata Cara Penggunaan Program

Untuk menggunakan aplikasi, pengguna harus memastikan node.js, *python dependencies* dan npm (Node Package Manager) telah diinstall dalam komputer. Untuk melakukan *set-up* pada komputer, pengguna dapat mengikuti langkah-langkah yang sudah tersedia pada README (<https://github.com/Farhanabd05/Algeo02-23034/blob/master/README.md>) yang ada di repository github kelompok kami. Berikut adalah tatacara penggunaan program yang telah kami buat.

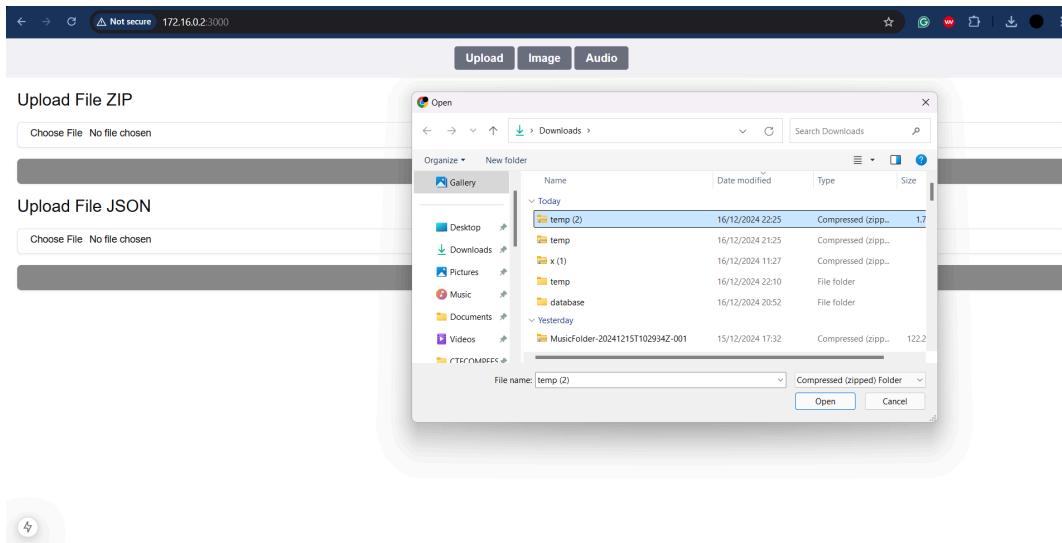
Pengguna dapat mengunggah gambar/audio yang ingin dicari dari suatu dataset. Proses penginputan gambar/audio dapat dilakukan dengan memilih opsi upload pada interface.

Setelah memilih opsi pencarian, baik pencarian image ataupun audio, pengguna dapat mengunggah file yang ingin di *query* dan menekan tombol "search". Program akan mulai memproses gambar yang telah diunggah. Proses ini mencakup pencarian gambar-gambar atau audio-audio dari dataset yang memiliki kemiripan dengan gambar/audio yang diunggah oleh pengguna.

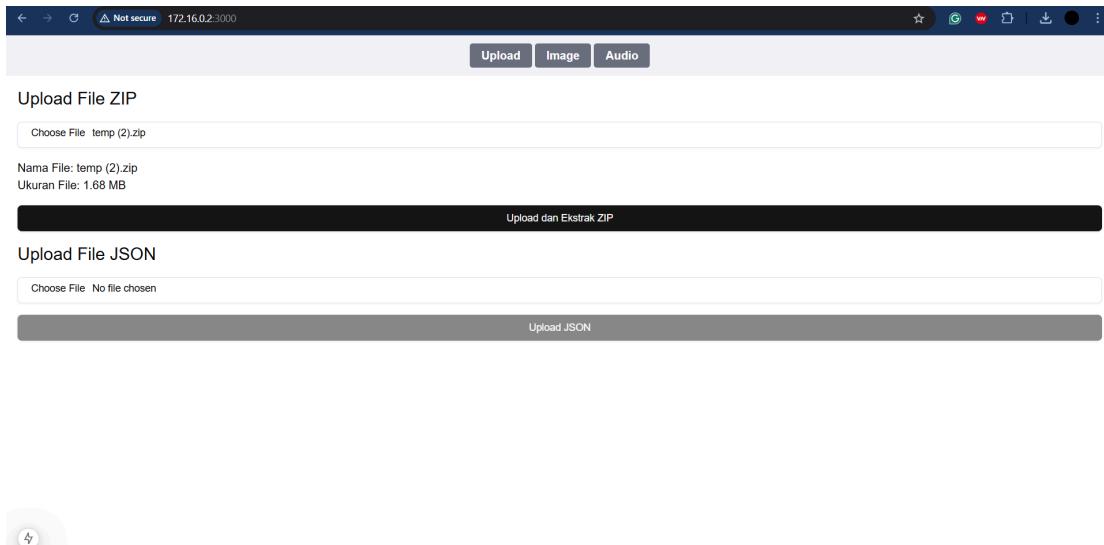
Hasil pencarian akan ditampilkan sebagai kumpulan gambar/audio yang mirip, diurutkan berdasarkan tingkat kemiripan dari yang paling tinggi hingga yang paling rendah. Setiap gambar/audio yang muncul akan disertai dengan persentase kemiripannya terhadap gambar/aduio yang diunggah oleh pengguna.

Program juga akan menyajikan waktu eksekusi programnya. Dengan demikian, pengguna dapat dengan mudah mengevaluasi hasil pencarian dan menemukan gambar/audio yang paling relevan sesuai dengan preferensi mereka.

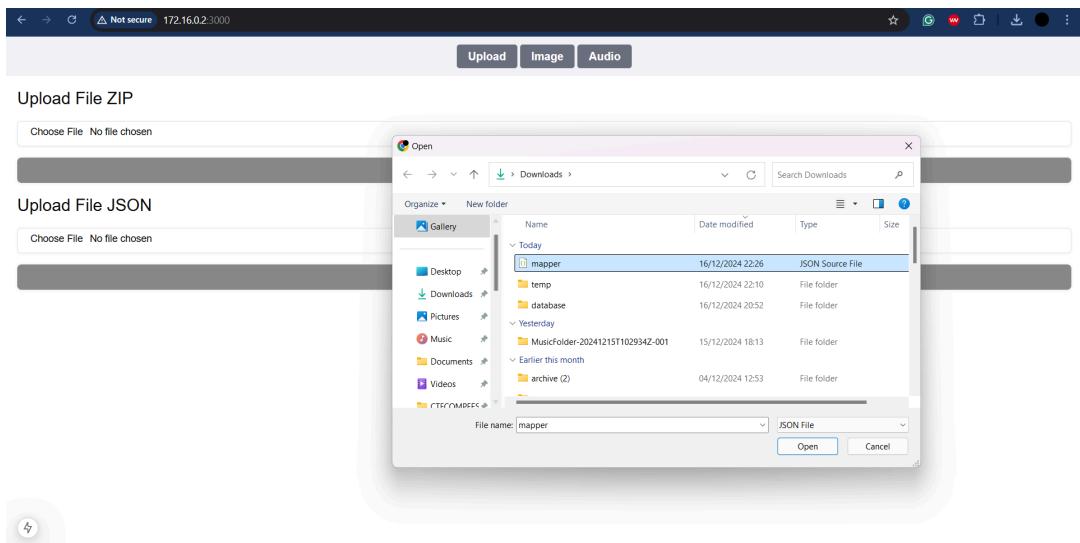
4.4 Hasil Pengujian



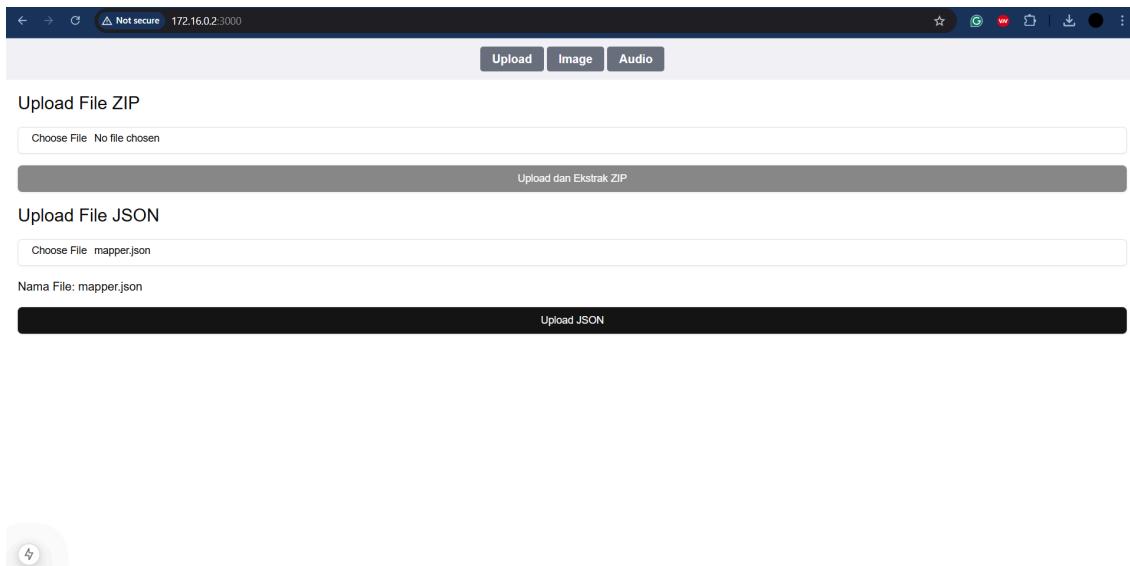
Gambar 4.1 Uji memilih file zip untuk di upload



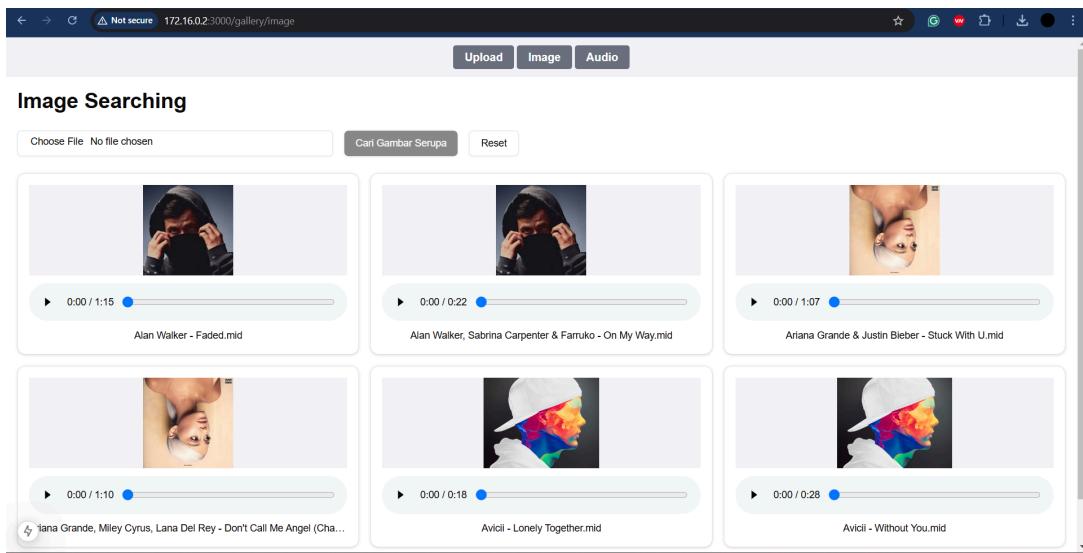
Gambar 4.2 Hasil upload zip



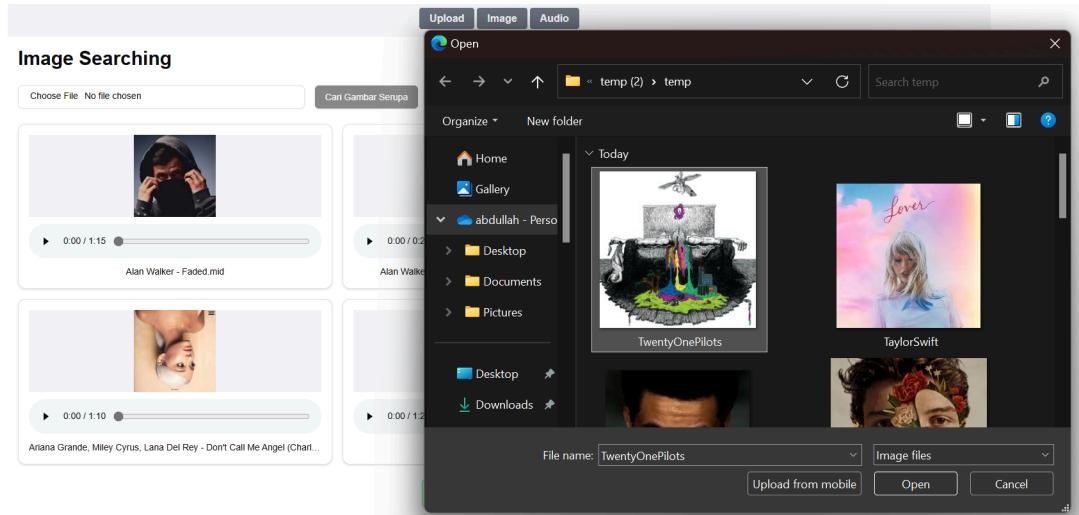
Gambar 4.3 Uji memilih file map untuk di upload



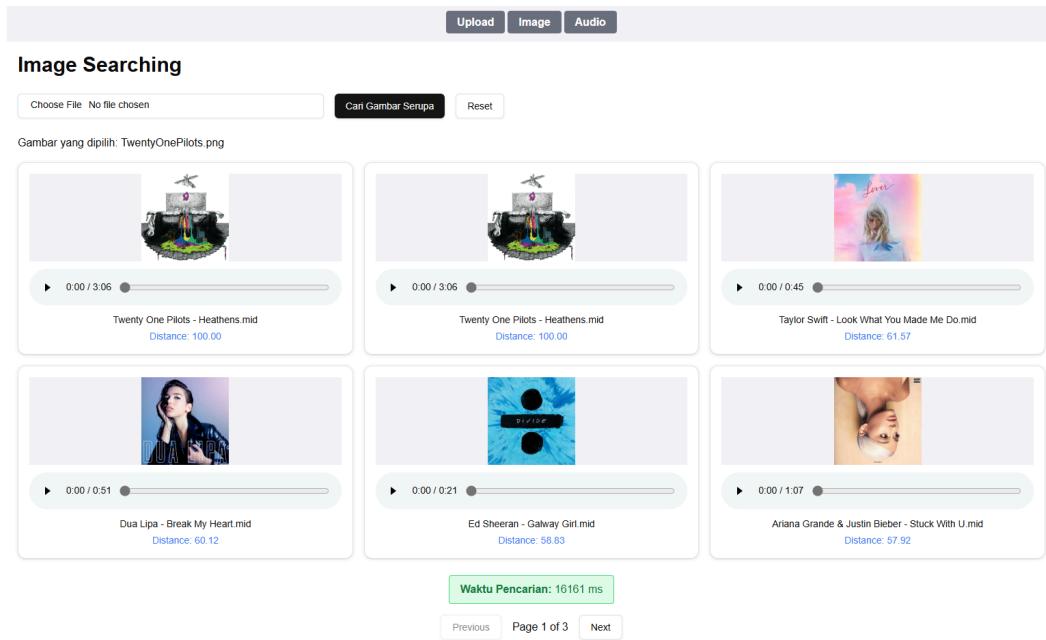
Gambar 4.4 Hasil upload map



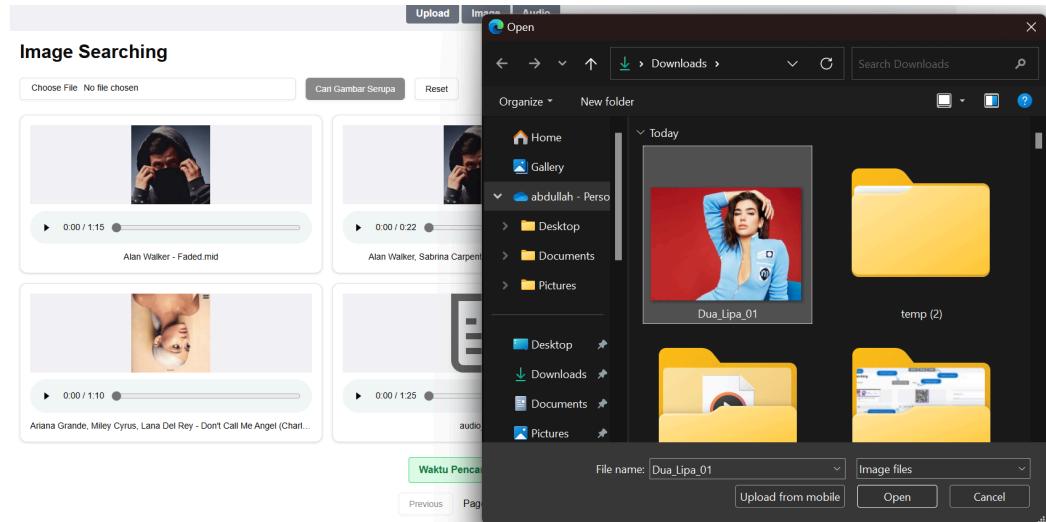
Gambar 4.5 Tampilan UI di Home



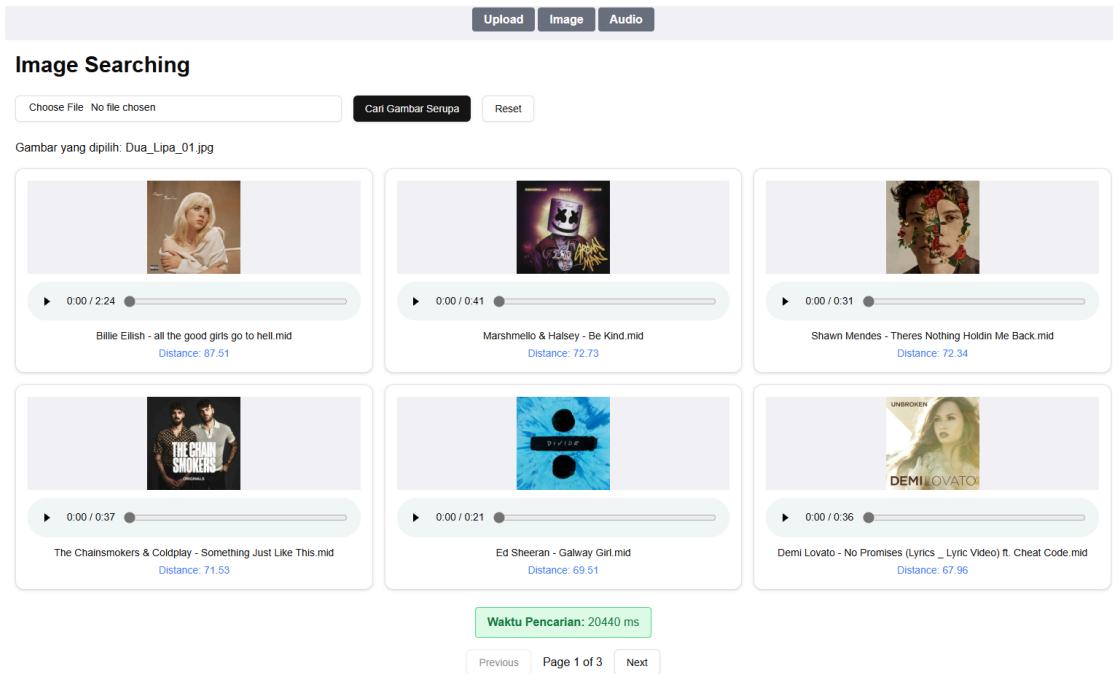
Gambar 4.6 Uji memilih file gambar untuk di upload



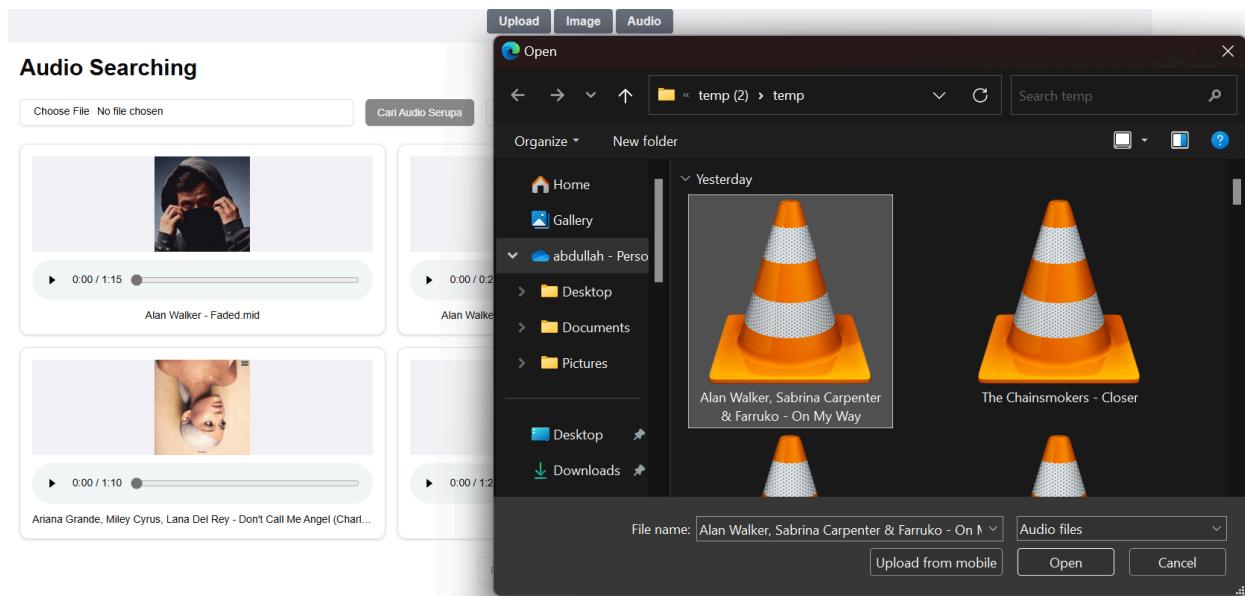
Gambar 4.7 Tampilan setelah berhasil melakukan image searching



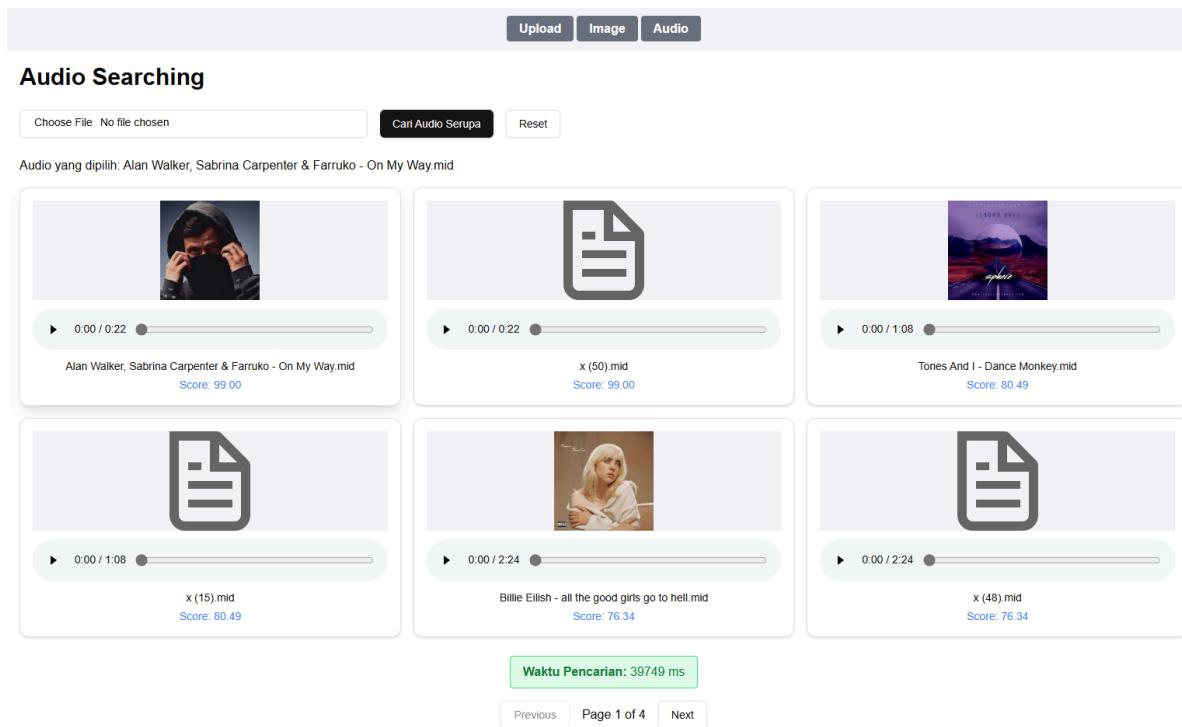
Gambar 4.8 Uji memilih file gambar untuk di upload yang di luar dataset



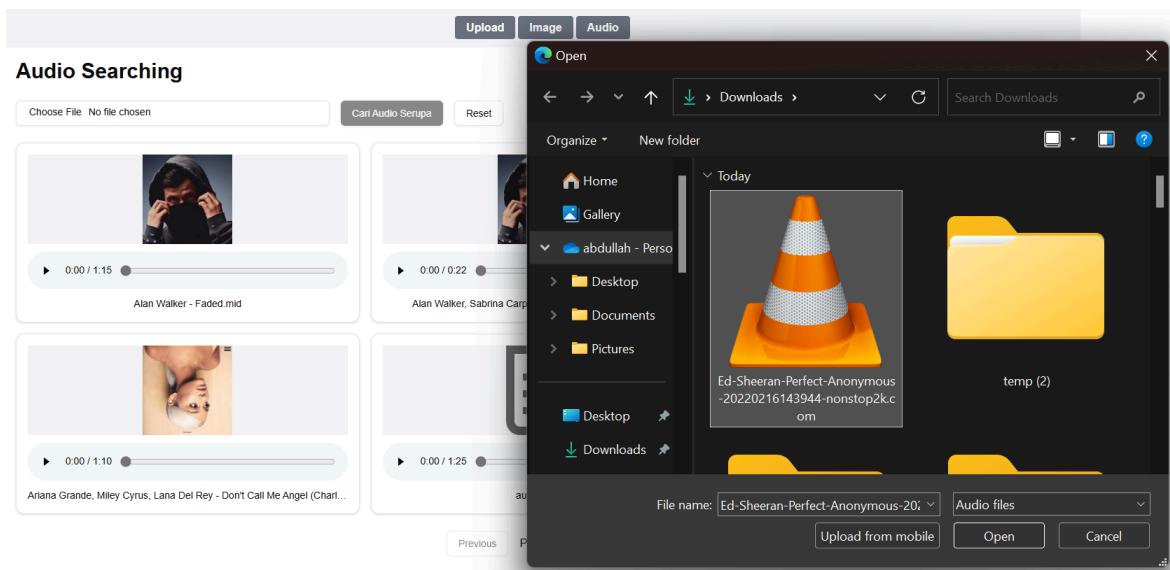
Gambar 4.9 Tampilan setelah berhasil melakukan image searching yang diluar dataset



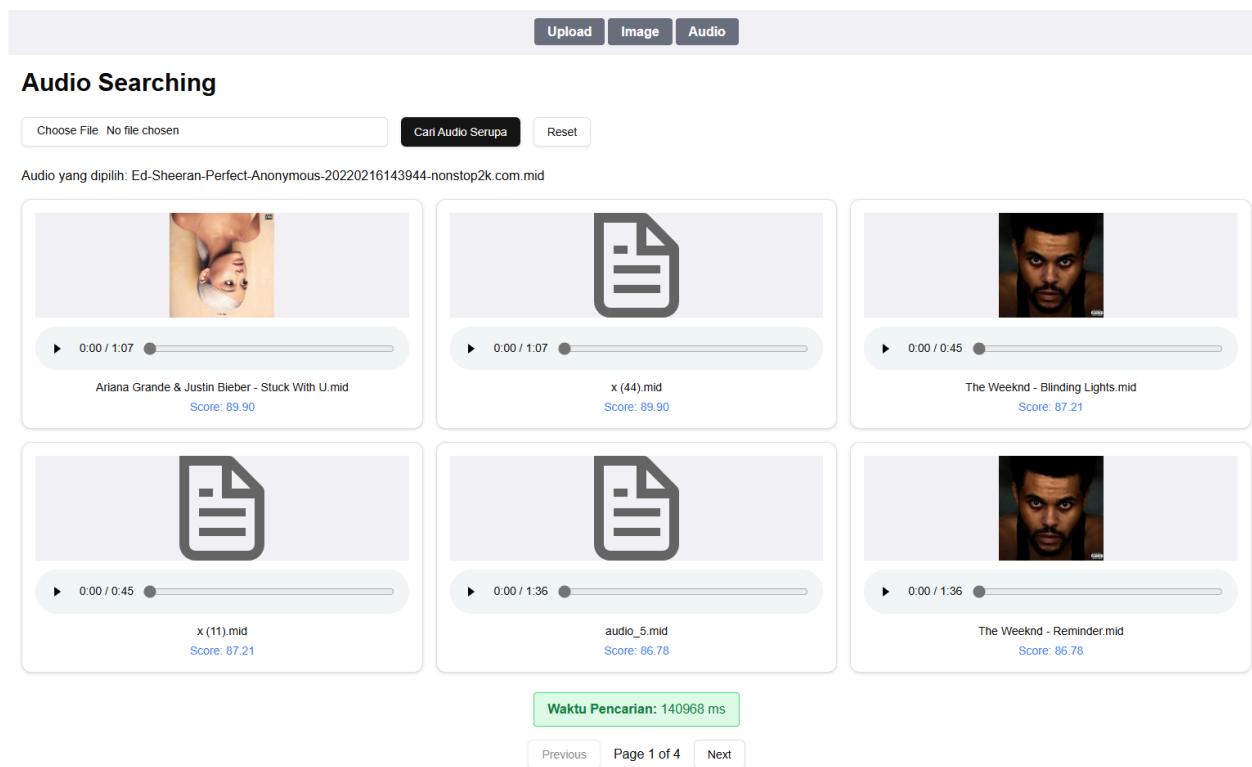
Gambar 4.10 Tampilan saat akan menguji untuk searching audio



Gambar 4.11 Tampilan setelah berhasil melakukan audio searching



Gambar 4.12 Tampilan saat akan menguji untuk searching audio yang tidak berada di dalam dataset



Gambar 4.13 Tampilan setelah berhasil melakukan audio searching di luar dataset

BAB 5

5.1 Kesimpulan

Dalam proyek Tugas Besar 2 IF2123 Aljabar Linier dan Geometri ini, kami berhasil mengimplementasikan aplikasi Aljabar Vektor untuk Sistem Temu Balik Gambar. Program kami menggunakan salah satu metode perbandingan gambar yaitu CBIR warna dan tekstur dengan ekstraksi fitur, dekomposisi matriks, dan normalisasi. Hasilnya, program mampu membandingkan dua gambar dengan memberikan persentase kemiripan yang dihitung menggunakan *euclidean distance* dan menampilkan hasilnya secara interaktif di web. Fitur utama program mencakup upload gambar, upload dataset, penggunaan kamera, dan *image-scraping*. Program ini memberikan pengalaman pencarian gambar berbasis konten yang efisien dan informatif, menciptakan solusi kuat bagi pengguna yang mencari gambar dengan kemiripan visual.

Dalam pencarian cosine similarity pada audio, kami menggunakan tiga metode ekstraksi fitur: Absolute Tone Based (ATB), Relative Tone Based (RTB), dan First Tone Based (FTB). Setiap histogram dikonversi menjadi vektor, lalu dihitung kemiripannya menggunakan cosine similarity. Berbeda dari jurnal terkait yang menggunakan euclidean distance, cosine similarity dipilih untuk mengukur seberapa mirip dua vektor dalam ruang berdimensi tinggi berdasarkan sudut cosinus di antaranya. Semakin mendekati 1, semakin mirip kedua vektor tersebut. Eksperimen dapat dilakukan dengan berbagai pembobotan untuk setiap fitur guna menentukan bobot terbaik. Hasilnya, program mampu membandingkan gambar query dengan dataset, menghitung persentase kemiripan menggunakan cosine similarity, dan menampilkan hasil secara interaktif di web.

5.2 Saran

Pelaksanaan Tugas Besar 2 IF2123 Aljabar Linier dan Geometri di Semester I Tahun 2024/2025 merupakan pengalaman yang sangat berharga bagi kami. Dari pengalaman ini, kami ingin berbagi beberapa saran kepada pembaca yang mungkin akan menghadapi tugas serupa di masa depan:

1. Efektivitas Kerja Sama Tim

Kerja sama tim sangat berperan dalam keberhasilan tugas ini. Penting untuk memahami peran masing-masing anggota, menjaga komunikasi yang teratur terkait progres, dan membagi tugas dengan jelas. Untuk mendukung kolaborasi, alat seperti VSCode Live Share dapat digunakan untuk pengembangan program secara bersamaan, sementara Google Docs efektif untuk menyusun laporan bersama. Selain itu, penggunaan GitHub sangat disarankan untuk mengelola kode sumber secara asinkron, menghindari konflik, dan menjaga versi proyek dengan baik.

2. Pemanfaatan Library yang Mempermudah

Penggunaan library Python tertentu menjadi elemen penting dalam tugas ini karena dapat menyederhanakan berbagai perhitungan yang kompleks. Penting untuk memahami fungsi-fungsi utama yang disediakan oleh library seperti NumPy atau Matplotlib, sehingga implementasi dalam program dapat dilakukan dengan lebih efisien. Mengacu pada dokumentasi resmi dan sumber daya belajar online akan sangat membantu memahami fitur-fitur ini.

3. Peningkatan Pemahaman Pengembangan Website

Tugas ini juga menuntut pengembangan website sebagai bagian dari penyajian solusi. Namun, bagi sebagian mahasiswa, pengembangan website mungkin menjadi tantangan karena belum diajarkan secara formal. Oleh karena itu, disarankan untuk belajar dasar-dasar teknologi web, seperti HTML, CSS, dan JavaScript, serta kerangka kerja yang relevan, untuk mempermudah pengerjaan bagian ini di masa depan.

Melalui penguasaan teknis dan kolaborasi yang efektif, pelaksanaan tugas besar ini tidak hanya membantu memahami konsep aljabar linier secara mendalam tetapi juga memberikan pengalaman berharga dalam manajemen proyek dan pemrograman. Semoga saran-saran ini membantu pembaca dalam menyiapkan diri untuk menangani tugas serupa di masa depan.

5.3 Komentar

Tugas Besar 2 Aljabar Linier dan Geometri di Semester 3 Tahun 2024/2025 memberikan pengalaman yang sangat berharga bagi kami. Walaupun menghadapi berbagai tantangan selama proses pengerjaan, kami merasa terbantu oleh dukungan dari para asisten yang selalu siap memberikan panduan serta teman-teman kelompok yang berkontribusi dalam menciptakan suasana kerja sama yang positif.

Kesulitan yang dihadapi selama tugas ini justru menjadi pemicu semangat kami untuk mengeksplorasi konsep-konsep baru dalam aljabar linier dan geometri. Proses pembelajaran ini tidak hanya memperluas wawasan kami, tetapi juga mengubah cara pandang kami terhadap aplikasi yang menggunakan teknologi pencarian, seperti Google Lens atau YouTube Humming Search. Kami tidak hanya belajar menggunakannya, tetapi juga memahami konsep dan prinsip kerja di balik teknologi tersebut. Pengalaman ini memperkaya pemahaman kami tentang bagaimana teori yang diajarkan di kelas dapat diterapkan dalam pengembangan aplikasi nyata, menjadikan pembelajaran ini semakin bermakna dan relevan dengan dunia teknologi.

“Tubesnya sangat mengapilkasikan materinya, haha (b00ng?)” - Muzaraar

5.4 Refleksi

Tugas Besar 2 IF2123 Aljabar Linier dan Geometri di Semester I Tahun 2024/2025 menjadi pengalaman yang tidak hanya menantang secara akademis, tetapi juga memperluas cara pandang kami terhadap penerapan teknologi modern. Melalui tugas ini, kami mendalamai bagaimana konsep aljabar linier diterapkan dalam sistem temu balik multimedia, seperti pencarian berbasis audio (MIR) atau gambar (CBIR), terutama menggunakan metode seperti PCA. Proyek ini memberikan wawasan baru tentang bagaimana fondasi matematika dapat diintegrasikan dengan teknologi untuk menciptakan aplikasi yang relevan dalam kehidupan sehari-hari.

Penerapan PCA dalam sistem CBIR membuka pemahaman kami tentang bagaimana matriks digunakan untuk merepresentasikan data visual. Misalnya, PCA mengurangi dimensi data gambar tanpa kehilangan informasi penting, memungkinkan pencarian yang lebih efisien dalam basis data yang besar. Kami menyadari bahwa teknologi ini tidak jauh berbeda dari yang kita temui di aplikasi seperti Google Lens atau sistem pencarian gambar lainnya. Di sisi lain, konsep MIR yang digunakan untuk pencarian berbasis audio memberikan gambaran tentang bagaimana fitur suara diekstraksi dan dibandingkan untuk menemukan kecocokan, seperti pada YouTube Humming Search.

Tugas ini juga memperlihatkan bagaimana aljabar linier berperan dalam merepresentasikan fitur, baik dari gambar maupun audio, dalam ruang vektor. Misalnya, metode PCA membantu menyederhanakan data sehingga pola dan hubungan antar fitur lebih mudah dikenali. Ini mencerminkan bagaimana aplikasi teknologi modern, seperti pengenalan objek dalam gambar atau pencarian lagu berdasarkan humming, bergantung pada konsep-konsep dasar matematika yang selama ini mungkin hanya terlihat teoritis.

Proses penggerjaan tugas ini juga menekankan pentingnya kerja sama tim yang solid. Setiap anggota kelompok memiliki peran unik yang harus diintegrasikan untuk menghasilkan hasil yang optimal. Seperti halnya dalam sistem MIR atau CBIR, di mana setiap komponen algoritma harus bekerja bersama untuk mencapai hasil yang akurat, kerja tim kami menjadi cerminan dari sinergi yang diperlukan dalam pengembangan teknologi.

Kesimpulannya, Tugas Besar ini tidak hanya memperkaya pemahaman kami tentang teori aljabar linier, tetapi juga memberikan pengalaman nyata tentang bagaimana konsep-konsep tersebut diterapkan dalam teknologi pencarian konten. Dengan memahami prinsip-prinsip seperti PCA dalam CBIR atau proses pencocokan audio dalam MIR, kami tidak hanya menjadi pengguna teknologi tetapi juga memperoleh apresiasi lebih dalam terhadap inovasi di balik layar. Pengalaman ini memotivasi kami untuk terus belajar dan mengeksplorasi potensi aljabar linier dalam menciptakan solusi teknologi yang lebih canggih di masa depan.

DAFTAR PUSTAKA

Rinaldi Munir. "Aljabar dan Geometri untuk Informatika 2024/2025."

informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2024-2025/algeo24-25.htm

N. Kosugi, Y. Nishihara, S. Kon'ya, M. Yamamuro and K. Kushima, "Music retrieval by humming-using similarity retrieval over high dimensional feature vector space," 1999 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM 1999). Conference Proceedings (Cat. No.99CH36368), Victoria, BC, Canada, 1999, pp. 404-407, doi: 10.1109/PACRIM.1999.799561.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=799561>

Riswanto, U. (2023,). *Image Compression Techniques: A Closer Look at Principal Component Analysis*. Medium.

<https://ujangriswanto08.medium.com/image-compression-techniques-a-closer-look-at-principal-component-analysis-67cf7a29fdb9>

LAMPIRAN

1. **Link Repository Github :** <https://github.com/Farhanabd05/Algeo02-23034.git>