

## LAPORAN TUGAS BESAR 2

IF2123 ALJABAR LINEAR DAN GEOMETRI



### Kelompok 18 - Ada Ada Aja

Anella Utari Gunadi	13523078
Diyah Susan Nugrahani	13523080
Athian Nugraha Muarajuang	13523106

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2024

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>1</b>
<b>BAB I</b>	<b>2</b>
<b>DESKRIPSI MASALAH</b>	<b>2</b>
<b>BAB II</b>	<b>3</b>
TEORI SINGKAT	3
2.1 Album Picture Finder - Principal Component Analysis	3
1. Image Processing and Loading	3
3. PCA Computational Using Singular Value decomposition (SVD)	4
4. Similarity Computation	4
2.2 Music Information Retrieval - Query by Humming	5
1. Pemrosesan Audio	5
2. Ekstraksi Fitur	6
2.1 Distribusi Tone	6
2.2 Normalisasi	7
3. Penghitungan Similaritas	7
<b>BAB III</b>	<b>8</b>
ARSITEKTUR WEBSITE DAN PROGRAM INFORMATION RETRIEVAL	8
1. Arsitektur Website	8
2. Program Information Retrieval	9
<b>BAB IV</b>	<b>11</b>
EKSPERIMEN	11
<b>BAB V</b>	<b>16</b>
KESIMPULAN	16
5.1 Kesimpulan	16
5.2 Saran	16
5.3 Komentar	16
5.4 Refleksi	16
LAMPIRAN	17
1. Referensi :	17
2. Link repository :	17
3. Link video :	17

## BAB I

### 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 mendekripsi lagu. Manusia bisa mendekripsi suara dengan menggunakan indera pendengar dan memberikan kesimpulan akan apa jenis suara tersebut melalui respon dari otak. Sama seperti manusia, teknologi juga bisa mendekripsi 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.

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.

## BAB II

### TEORI SINGKAT

#### 2.1 Album Picture Finder - Principal Component Analysis

Setiap gambar memiliki informasi penting dalam setiap pikselnya. Untuk membandingkan suatu gambar dengan gambar yang lain dan mencari kemiripannya dapat diterapkan prinsip PCA (*Principal Component Analysis*). PCA adalah teknik analisis data yang sering digunakan untuk mencari kemiripan dari suatu gambar dengan cara mengurangi dimensi data namun tetap mempertahankan informasi pentingnya. Dalam konteks pencarian gambar, PCA digunakan untuk mengubah data gambar menjadi representasi berdimensi rendah untuk mempermudah pengolahan data. Hasil data yang diperoleh dari PCA akan berupa eigenvector dan proyeksi data, yang kemudian dicari jaraknya untuk mengklasifikasikan kemiripan gambar.

Dalam menerapkan konsep PCA perlu dilakukan beberapa tahapan. Beberapa tahapan tersebut antara lain :

##### 1. Image Processing and Loading

Sebelum dilakukan pengolahan data lebih lanjut, gambar perlu diproses terlebih dahulu. Hal pertama yang dilakukan adalah mengurangi kompleksitas gambar dan membuatnya fokus menjadi bagian terang dan gelap dengan cara mengubah gambar menjadi *grayscale*. Setiap gambar akan direpresentasikan dalam intensitas pixel saja tanpa informasi warna.

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

Setelah diubah menjadi *grayscale*, selanjutnya gambar diubah ukurannya agar sama dan konsisten untuk membuat perhitungan semakin akurat. Kemudian ubah vektor tersebut menjadi 1D agar dapat dilakukan pemrosesan data. Jika gambar memiliki dimensi M x N, maka hasilnya adalah vektor dengan panjang M.N.

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

##### 2. Data Centering (*Standardization*)

Untuk membuat perhitungan lebih akurat, perlu dilakukan standarisasi data di sekitar nilai 0. Hitung rata-rata dari 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.

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

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

### 3. PCA Computational Using Singular Value decomposition (SVD)

Untuk melakukan PCA diperlukan perhitungan SVD pada gambar yang sudah distandarisasi. Buat matriks kovarians dengan rumus sebagai berikut :

$$C = \frac{1}{N} X'^T X' \quad X': \text{matriks data yang sudah distandarisasi.}$$

Lalu lakukan dekomposisi nilai singular untuk mendapatkan komponen utama.

$U$ : matriks eigenvector (komponen utama),

$$\mathbf{C} = \mathbf{U} \Sigma \mathbf{U}^T \quad \Sigma: \text{matriks eigenvalue}$$

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:

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

$\mathbf{U}_k$ : matriks eigenvector dengan n-dimensi.

### 4. Similarity Computation

Setelah mendapatkan komponen yang dibutuhkan, selanjutnya dilakukan pencarian kesamaan gambar dengan mencari jarak euclidean dari tiap dataset dengan gambar query. Kemudian dilakukan pengurutan kecocokan dari yang paling tinggi. Representasikan gambar query dalam ruang komponen utama dengan proyeksi yang sama :

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

- $\mathbf{q}$  = 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, lalu, Urutkan hasil berdasarkan jarak terkecil..

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

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

## 2.2 Music Information Retrieval - Query by Humming

Dalam melakukan query by humming, terdapat langkah-langkah yang harus dilakukan. Yang pertama adalah memproses audio. Pada pemrosesan audio, suara yang diinput atau diterima, dipersiapkan untuk melakukan tahap selanjutnya. Langkah kedua adalah melakukan ekstraksi fitur vektor, yaitu proses konversi data audio menjadi representasi numerik berupa vektor fitur yang dapat dianalisis. Langkah terakhir adalah tahap pencarian similaritas vektor. Pada tahap ini, vektor fitur yang dihasilkan dibandingkan dengan dataset untuk menemukan hasil yang paling sesuai atau diatas nilai similaritas yang telah ditentukan.

Deskripsi lebih lanjut dari langkah-langkah melakukan query by humming adalah sebagai berikut.

### 1. Pemrosesan Audio

Pemrosesan audio dalam sistem query by humming menggunakan file MIDI berfokus pada melodi utama, biasanya terdapat di Channel 1. File MIDI diproses dengan cara membagi melodi menjadi segmen-semen kecil sepanjang 20-40 beat menggunakan metode windowing, dengan pergeseran (sliding window) sepanjang 4-8 beat. Teknik ini membantu mencocokkan bagian-bagian lagu yang mungkin diingat oleh pengguna.

Selama proses ini, tempo dan pitch dinormalisasi untuk mengurangi perbedaan saat pengguna melakukan humming. Setiap not dalam melodi diubah menjadi angka yang merepresentasikan durasi dan urutan nada. Hal ini memungkinkan sistem membandingkan potongan melodi pengguna dengan lagu-lagu yang ada di database.

Formula untuk melakukan normalisasi tempo yang dibutuhkan:

$$NP(note) = \frac{(note - \mu)}{\sigma} \quad \begin{aligned} &\text{dengan, } \mu = \text{rata rata dari pitch} \\ &\sigma = \text{standar deviasi dari pitch} \end{aligned}$$

## 2. Ekstraksi Fitur

### 2.1 Distribusi Tone

Distribusi nada diukur dengan menggunakan tiga pendekatan utama, yaitu fitur Absolute Tone Based (ATB), fitur Relative Tone Based (RTB), dan fitur First Tone Based (FTB).

Fitur Absolute Tone Based (ATB) menghitung seberapa sering setiap nada muncul dalam rentang skala MIDI (0-127). Hasil perhitungannya berupa histogram yang menunjukkan distribusi absolut nada pada data. Proses ini penting untuk menangkap karakteristik melodi secara statis dari sinyal audio. Tahapannya meliputi: pertama, membuat histogram dengan 128 bin sesuai rentang nada MIDI (0-127). Kedua, menghitung jumlah kemunculan setiap nada MIDI dalam data. Terakhir, melakukan normalisasi histogram untuk mendapatkan distribusi yang terstandarisasi.

Fitur Relative Tone Based (RTB) menganalisis perubahan nada dari satu nada ke nada berikutnya, menghasilkan histogram dengan nilai antara -127 hingga +127. Fitur ini berguna untuk memahami pola interval melodi, yang lebih relevan ketika mencocokkan humming dengan database tanpa bergantung pada pitch absolut. Prosesnya dimulai dengan membuat histogram dengan 255 bin yang mencakup rentang -127 hingga +127. Kemudian, menghitung selisih antara nada-nada berurutan dalam data. Langkah terakhir adalah menormalisasi histogram yang telah dibuat agar hasilnya konsisten dan terstandarisasi.

Fitur First Tone Based (FTB) berfokus pada perbedaan antara setiap nada dengan nada pertama sebagai titik referensi. Proses ini menghasilkan histogram yang menggambarkan hubungan relatif setiap nada terhadap nada awal. Pendekatan ini bermanfaat untuk menangkap struktur melodi yang lebih stabil terhadap variasi pitch dari pengguna. Tahapannya dimulai dengan membuat histogram berisi 255 bin, mencakup rentang nilai dari -127 hingga +127. Selanjutnya, hitung selisih antara setiap nada dalam data dengan nada pertama. Terakhir, histogram yang dihasilkan dinormalisasi untuk mendapatkan distribusi yang seimbang dan terstandarisasi.

## 2.2 Normalisasi

Normalisasi bertujuan agar semua nilai dalam histogram berada dalam skala probabilitas, sehingga distribusi data menjadi seimbang dan terstandarisasi. Formula umum yang digunakan untuk normalisasi adalah sebagai berikut:

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

Dengan, H = histogram  
d = bin dari histogram tersebut

## 3. Penghitungan Similaritas

Ubah setiap histogram menjadi vektor dan hitung kemiripannya menggunakan cosine similarity. Pada jurnal sebelumnya, metode yang digunakan adalah Euclidean Distance, namun dalam tugas ini metode yang dipakai adalah cosine similarity. Cosine similarity digunakan untuk mengukur tingkat kemiripan antara dua vektor dalam ruang berdimensi tinggi dengan menghitung sudut cosinus di antara keduanya. Semakin kecil sudut tersebut (semakin mendekati nilai 1), maka kedua vektor dianggap semakin mirip. Oleh karena itu, cosine similarity dapat digunakan sebagai alternatif metode perhitungan kemiripan. Lakukan eksplorasi dengan berbagai bobot pada setiap fitur dan tentukan kombinasi bobot terbaiknya.

Berikut adalah formula cosine similarity:

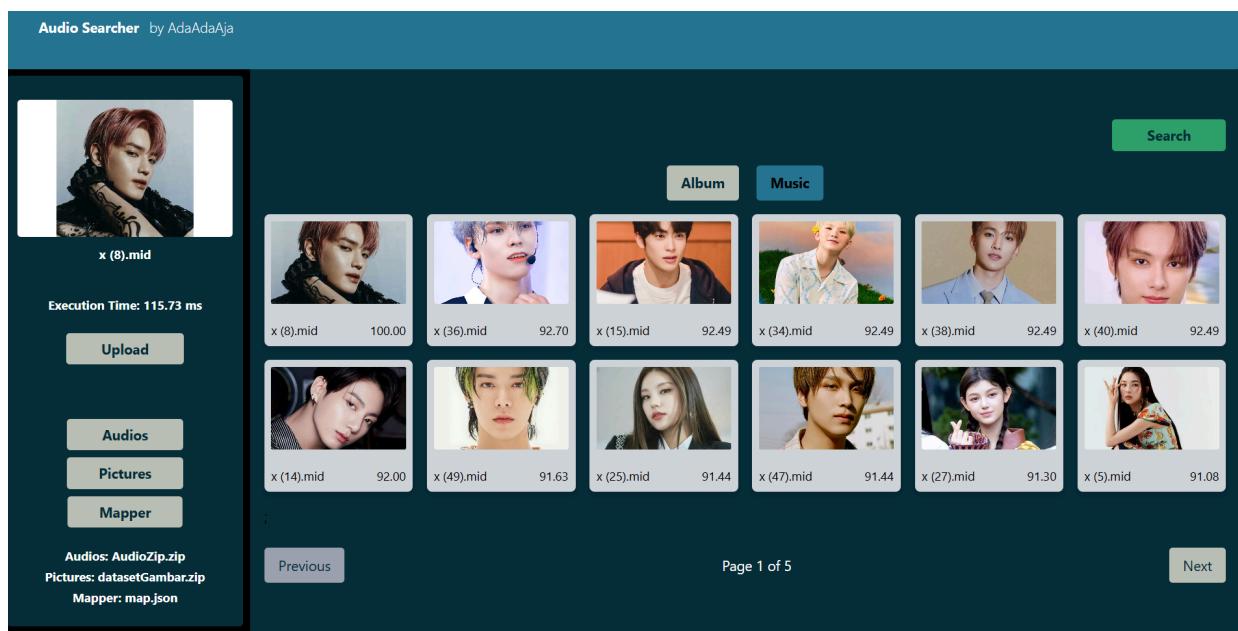
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

## BAB III

### ARSITEKTUR WEBSITE DAN PROGRAM INFORMATION RETRIEVAL

#### 1. Arsitektur Website

Pada tugas besar ini, framework yang kami gunakan untuk frontend adalah React.js, sedangkan untuk backend adalah Flask. Pada tampilan front-end, kami menggunakan tiga komponen utama, yaitu sidebar, song grid, dan navbar. Komponen sidebar digunakan untuk mengunggah file dan dataset dari pengguna, menampilkan gambar query yang ingin dicari dari masukkan pengguna, menampilkan nama file yang telah diunggah oleh pengguna, serta eksekusi waktu dari pemrosesan dataset maupun pemrosesan *audio retrieval*. Komponen song grid digunakan untuk memilih mode “Album” atau “Music”, memulai pemrosesan audio retrieval dalam button “Search” serta menghasilkan pencarian dalam grid yang berisi foto, nama file audio, serta persentase kemiripan atau *similarity*-nya yang dikemas dalam bentuk pagination. Komponen navbar digunakan untuk menampilkan navigasi dan judul website yang dibuat yaitu Audio Searcher by AdaAdaAja.



Berikut adalah penjelasan dari masing-masing komponen yang terdapat pada tampilan website Audio Searcher by AdaAdaAja.

1. Navigasi bar beserta judul website.
2. Tombol Upload adalah tombol untuk memasukkan file gambar atau audio untuk kemudian dicari kemiripannya dengan dataset yang telah diinput sebelumnya.
3. Tombol Audios adalah tombol untuk memasukkan dataset audio dengan format zip maupun *multiple files* yang berisi file berformat midi.
4. Tombol Pictures adalah tombol untuk memasukkan dataset audio dengan format zip maupun *multiple files* yang berisi file berformat jpg, jpeg, png, maupun bmp.

5. Tombol Mapper adalah tombol untuk memasukkan dataset gabungan dari audio dan picture yang telah dilakukan mapping dengan format file dapat berupa json maupun txt.
6. Kotak gambar digunakan untuk menampilkan preview dari gambar yang diinput pengguna. Jika pengguna mengunggah audio, gambar yang diinput adalah sesuai mapping yang telah diunggah atau tidak muncul apa-apa.
7. Tombol Search adalah tombol untuk memulai pencarian audio berdasarkan file (query) yang telah diunggah oleh pengguna.
8. Tombol Album dan Music adalah tombol untuk membedakan pencarian dengan menggunakan query audio by humming dan query album finder with PCA.
9. Kumpulan song card merupakan hasil dari query audio ataupun query album yang terurut dari kemiripan paling tinggi ke paling rendah
10. Tombol Previous dan Next adalah tombol untuk pagination dengan setiap halaman terdiri dari maksimal 12 song card.

## 2. Program Information Retrieval

Information Retrieval (IR) berkaitan dengan temu balik informasi berupa gambar dan audio. Pencarian gambar dilakukan dengan metode *Principal Component Analysis* (PCA) digunakan untuk mereduksi dimensi fitur gambar. Dengan bantuan *library python* pillow, data gambar diubah ke format grayscale, diproses dengan SVD (Singular Value Decomposition), dan diubah menjadi representasi vektor. Berdasarkan vektor dari gambar *query* dan dataset, dilakukan *Similarity Matching* menggunakan *Euclidean Distance*. Dengan hasil data *similarity* yang didapatkan, akan diberikan output gambar dari dataset yang paling mirip dengan gambar input, berdasarkan jarak Euclidean maksimal 250.

Kami menggunakan Flask untuk membangun backend dari website ini. Flask dipilih karena kemudahan implementasinya dalam membangun API sederhana dan efisien. Pada website ini, backend bertanggung jawab sebagai server yang menangani permintaan dari pengguna, memproses file gambar, audio, maupun dataset, serta mengembalikan hasilnya yang mencakup informasi audio dan gambar yang mirip, waktu eksekusi, dan nama file yang diunggah untuk ditampilkan di frontend. Website memiliki beberapa direktori penting yang didefinisikan di awal, antara lain:

- UPLOAD\_FOLDER yaitu direktori sementara untuk menyimpan file yang diunggah pengguna sebelum diproses.
- DATASET\_FOLDER yaitu direktori untuk menyimpan dataset gambar yang diunggah oleh pengguna.
- AUDIO\_FOLDER yaitu direktori untuk menyimpan dataset file MIDI/audio yang diunggah oleh pengguna.

- MAPPER\_FOLDER yaitu direktori untuk menyimpan file mapper berupa JSON atau TXT yang berisi mapping antara file gambar dan file audio.

Pada backend, terdapat beberapa endpoint utama yang berfungsi untuk mengunggah file, memproses data, dan mengirimkan hasilnya ke frontend. Endpoint ini mencakup:

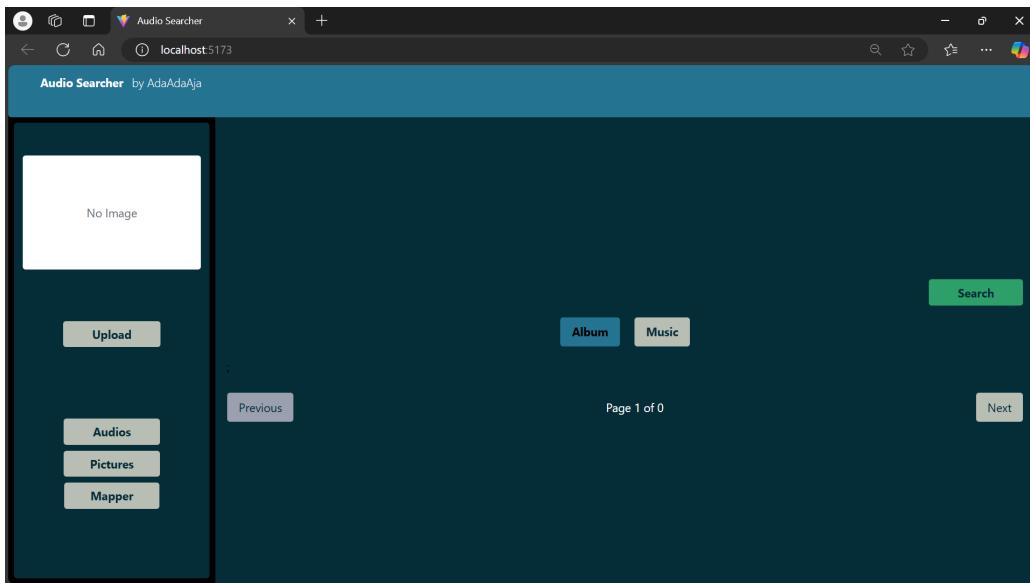
1. /upload-picture, yaitu untuk mengunggah gambar, mencocokkannya dengan dataset, dan menghubungkan hasilnya dengan file audio melalui mapper.
2. /upload-midi, yaitu untuk mengunggah file MIDI, mencocokkannya dengan dataset audio, dan mencari gambar terkait melalui mapper.
3. /upload-zip/<category>, yaitu untuk mengunggah file dataset dalam format arsip (zip ataupun rar), mengekstraknya, dan menyimpannya ke direktori sesuai kategori gambar, audio, atau mapper.
4. /dataset-image/<filename>, yaitu untuk mengirimkan file gambar dari dataset sebagai preview ke frontend.
5. /get-dataset-mapped, yaitu untuk mengambil daftar file gambar yang sudah dihubungkan dengan file audio berdasarkan file mapper.

Dalam pencarian musik (*Music Information Retrieval*), file input yang diterima dan diproses hanyalah file midi. Implementasi yang dilakukan adalah input audio diunggah sebagai MIDI, lalu input tersebut diolah dengan *library* mido dari python untuk diambil data-data notes pada file midi tersebut. Notes yang diambil diutamakan notes pada channel pertama. Namun, jika tidak memilikinya, notes akan membuat fallback notes dimana akan mengambil notes dari semua channel. Setelah itu, notes dinormalisasikan yang kemudian diekstraksi dalam bentuk histogram *Absolute Tone Based* (ATB), *Relative Tone Based* (RTB), dan *First Tone Based* (FTB). Setelah diekstraksi, histogram dinormalisasikan dan dibentuk ke dalam vektor. Dengan terbentuknya vektor, midi *query* dengan dataset bisa dicek kemiripannya dengan menggunakan *Cosine Similarity*. Program akan memberikan output audio yang paling cocok dengan *query*, dengan persentase kemiripan di atas 55%.

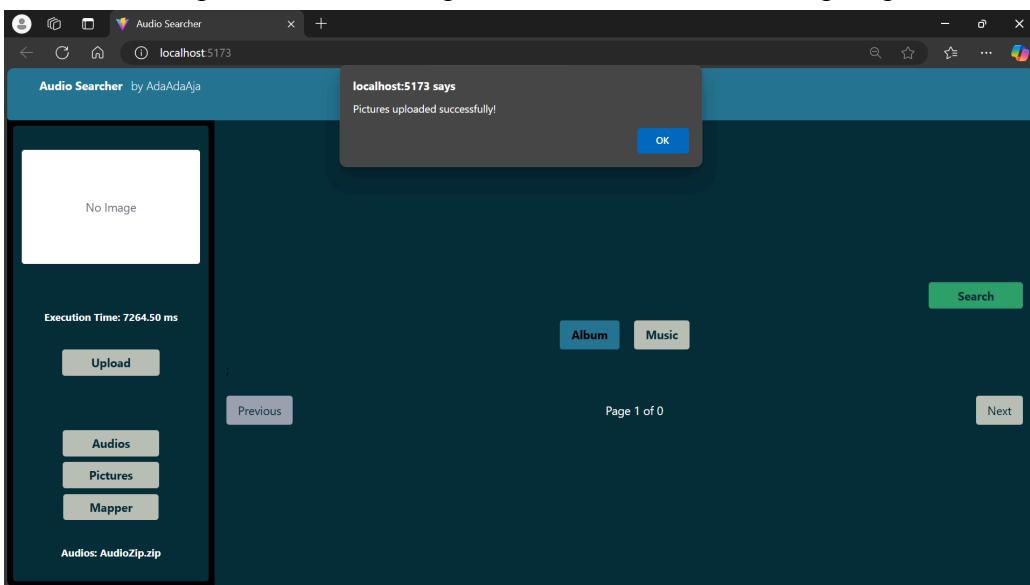
## BAB IV

### EKSPERIMEN

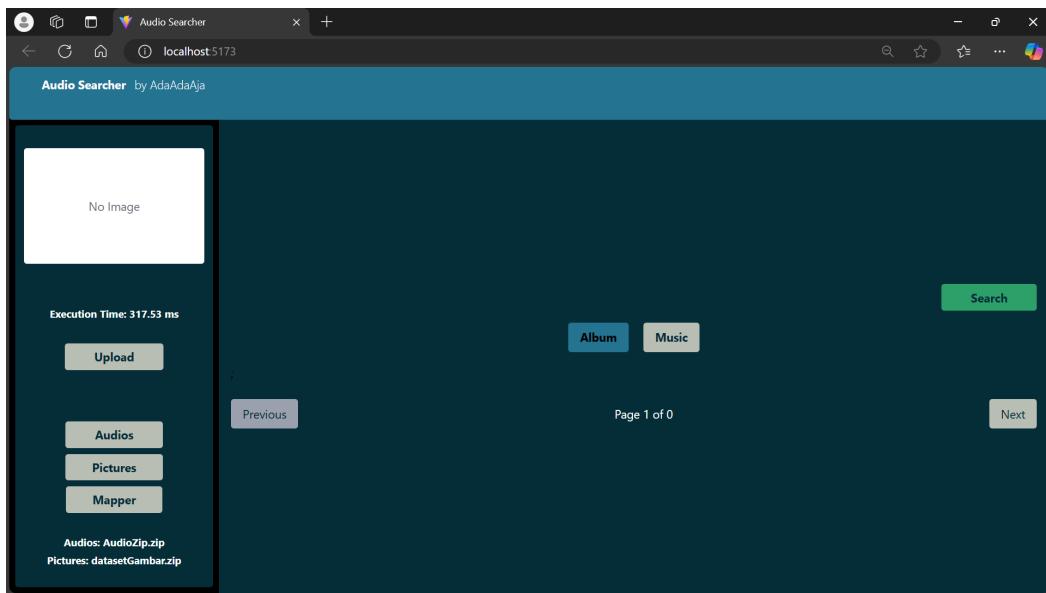
Tampilan utama website



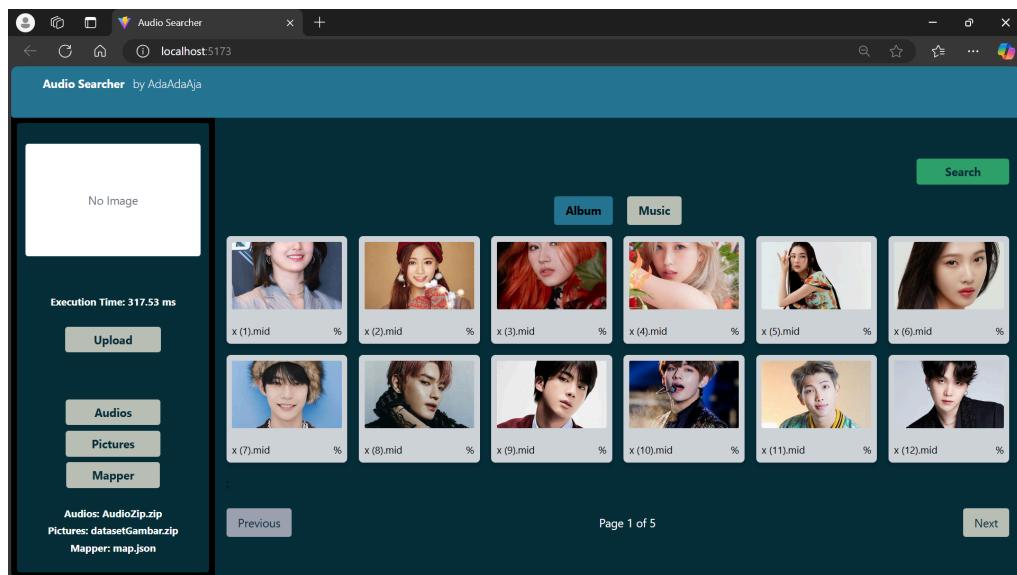
Tampilan web setelah upload dataset audio midi berupa zip



Tampilan setelah upload dataset pictures berupa zip



Tampilan setelah upload mapper



Tampilan saat dimasuki query gambar lewat tombol upload

**Kasus 1**, query pertama relatif lama karena dataset perlu diolah datanya terlebih dahulu

The screenshot shows the 'Audio Searcher' application interface. On the left, a large thumbnail of a person is labeled 'haechan.jpg'. Below it, the text 'Execution Time: 5399.72 ms' is displayed. There are three buttons: 'Upload', 'Audios', and 'Pictures'. At the bottom of this sidebar, there are links for 'Audios: AudioZip.zip', 'Pictures: datasetGambar.zip', and 'Mapper: map.json'. On the right, there are two tabs: 'Album' (selected) and 'Music'. Below these tabs is a grid of 12 smaller thumbnail images, each with a file name like 'x (47).mid' or 'x (7).mid' and a timestamp like '0.00' or '133.33'. At the bottom of the main area, there are 'Previous' and 'Next' buttons, and the text 'Page 1 of 3'.

**Kasus 2**, performa jauh lebih cepat karena data dari query pertama kali karena fitur-fitur gambar sudah diolah dan disimpan ke dalam file json

The screenshot shows the 'Audio Searcher' application interface. On the left, a large thumbnail of a person is labeled 'mark.jpg'. Below it, the text 'Execution Time: 81.83 ms' is displayed. There are three buttons: 'Upload', 'Audios', and 'Pictures'. At the bottom of this sidebar, there are links for 'Audios: AudioZip.zip', 'Pictures: datasetGambar.zip', and 'Mapper: map.json'. On the right, there are two tabs: 'Album' (selected) and 'Music'. Below these tabs is a grid of 12 smaller thumbnail images, each with a file name like 'x (50).mid' or 'x (17).mid' and a timestamp like '0.00' or '126.51'. At the bottom of the main area, there are 'Previous' and 'Next' buttons, and the text 'Page 1 of 2'.

Tampilan saat dimasuki query gambar lewat tombol upload

### Kasus 1

The screenshot shows the 'Audio Searcher' application running in a browser window. On the left, there is a large thumbnail of a person's face with the text 'x (8).mid' below it. Below the thumbnail, the text 'Execution Time: 115.73 ms' is displayed. There are three buttons: 'Upload', 'Audios', and 'Pictures'. At the bottom of this sidebar, the file paths are listed: 'Audios: AudioZip.zip', 'Pictures: datasetGambar.zip', and 'Mapper: map.json'. On the right side, there is a grid of smaller thumbnails. Two buttons at the top of the grid are 'Album' and 'Music'. The grid contains 15 items, each with a thumbnail, a file name like 'x (8).mid', a score like '100.00', and a timestamp like '92.70'. Navigation buttons 'Previous' and 'Next' are at the bottom, along with the text 'Page 1 of 5'.

### Kasus 2

This screenshot shows the same 'Audio Searcher' application interface as the previous one, but with different search results. On the left, there is a large thumbnail of a person's face with the text 'x (46).mid' below it. Below the thumbnail, the text 'Execution Time: 66.20 ms' is displayed. There are three buttons: 'Upload', 'Audios', and 'Pictures'. At the bottom of this sidebar, the file paths are listed: 'Audios: AudioZip.zip', 'Pictures: datasetGambar.zip', and 'Mapper: map.json'. On the right side, there is a grid of smaller thumbnails. Two buttons at the top of the grid are 'Album' and 'Music'. The grid contains 15 items, each with a thumbnail, a file name like 'x (46).mid', a score like '100.00', and a timestamp like '91.62'. Navigation buttons 'Previous' and 'Next' are at the bottom, along with the text 'Page 1 of 5'.

Tampilan setelah upload dataset berupa rar

The screenshot shows the 'Audio Searcher' application interface. On the left, there is a sidebar with a placeholder image for uploaded audio files, labeled 'x (46).mid'. Below it, the text 'Execution Time: 13631.17 ms' is displayed. There are three buttons: 'Upload', 'Audios', 'Pictures', and 'Mapper'. Under 'Mapper', the file 'map.json' is listed. At the bottom of the sidebar, there are three lines of text: 'Audios: AudioZip.rar', 'Pictures: datasetGambar.rar', and 'Mapper: map.json'. The main area contains a grid of 14 image thumbnails, each with a file name and a percentage value. The first row contains thumbnails for 'x (1).mid', 'x (2).mid', 'x (3).mid', 'x (4).mid', 'x (7).mid', and 'x (8).mid'. The second row contains thumbnails for 'x (9).mid', 'x (10).mid', 'x (11).mid', 'x (12).mid', 'x (13).mid', and 'x (14).mid'. Above the grid, there are two tabs: 'Album' (selected) and 'Music'. A green 'Search' button is located in the top right corner of the main content area. Navigation buttons 'Previous' and 'Next' are at the bottom of the page, along with the text 'Page 1 of 4'.

## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan

Pengerjaan tugas besar 2 pada mata kuliah Aljabar Linear dan Geometri mengajarkan kami terkait penerapan mengenai dekomposisi matriks dan operasi vektor. Dalam pencarian gambar diterapkan konsep PCA menggunakan dekomposisi SVD dengan mengambil nilai eigen vektor dan kemudian dilakukan pengolahan data untuk mencari jarak antara query dengan dataset. Sedangkan pada pencarian menggunakan audio diterapkan konsep pengolahan vektor yang nantinya akan digunakan untuk menghitung *cosine similarity* untuk membandingkan dataset dengan *query* yang dimasukkan.

Selain menerapkan materi aljabar geometri, kami juga menerapkan arsitektur website sebagai platform untuk pencarian audio dan gambar. Meskipun dalam materi mata kuliah aljabar geometri tidak pernah diajarkan mengenai arsitektur website, namun kami berusaha untuk mencari sumber belajar untuk mendukung pembuatan website yang baik dan dapat menjalankan algoritma *album finder* dan *audio searcher* dengan baik. Tugas besar ini mengajarkan banyak hal baru dan mendorong kami untuk eksplorasi dan mencoba banyak hal baru yang sangat bermanfaat dan meningkatkan *skill* kami.

#### 5.2 Saran

Saran yang dapat kami berikan adalah terkait spesifikasi yang masih ambigu dan membingungkan di beberapa bagian. Untuk kedepannya bisa diperjelas lagi terkait alur dan batasan apa yang diperbolehkan. Saran lain adalah terkait *dataset*, sebaiknya asisten memberikan contoh dataset yang banyak dan bagus sebagai referensi dalam penggerjaan tubes ini.

#### 5.3 Komentar

Menurut kami tugas besar ini memiliki persentase penerapan materi aljabar geometri yang lebih sedikit dibandingkan dengan materi terkait *web development*. Jika diperhatikan, sebagian besar waktu yang dihabiskan adalah untuk menyatukan antara algoritma yang telah dibuat dengan website yang dibuat, atau bisa dibilang waktu yang paling banyak dilakukan adalah untuk integrasi. Hal tersebut sekiranya kurang relevan dengan materi pembelajaran di kelas.

#### 5.4 Refleksi

Dalam penggerjaan tugas besar 2 aljabar geometri ini kami mengalami beberapa kendala dalam proses penggerjaan. Di tengah penggerjaan kami sempat mengganti *framework* yang awalnya ingin menggunakan django akhirnya berpindah menggunakan flask. Hal tersebut terjadi karena kurangnya riset dan kurang matangnya perencanaan awal.

## LAMPIRAN

1. Referensi :

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

<https://asmp-eurasipjournals.springeropen.com/articles/10.1155/2010/179303>

2. *Link repository* :

[https://github.com/Starath/Algeo02\\_23078](https://github.com/Starath/Algeo02_23078)

3. *Link video* :

<https://drive.google.com/file/d/1L8pDoPeAXqEQnCPDZ9hFWTDFWwDfOxXP/view?usp=drivesdk>