

LAPORAN TUGAS BESAR 02
IF2123 ALJABAR LINIER DAN GEOMETRI

Kelompok 41

Malam Jumat di Sorong



Disusun Oleh:

Felix Chandra	13523012
Ahmad Ibrahim	13523089
Karol Yangqian Poetracahaya	13523093

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

2024

DAFTAR ISI

BAB 1	2
1.1. Pemrosesan Suara dan Gambar	2
1.2. Information Retrieval	3
BAB 2	4
2.1. Album Picture Finder - Principal Component Analysis	4
2.1.1. Image Processing and Loading	5
2.1.2. Data Centering (Standardization)	5
2.1.3. PCA Computation Using Singular Value Decomposition (SVD)	5
2.1.4. Similarity Computation	6
2.1.5. Retrieval and Output	7
2.2. Music Information Retrieval - Query by Humming	7
2.2.1. Pemrosesan Audio	7
2.2.2. Ekstraksi Fitur - Distribusi tone	8
2.5.1. Ekstraksi Fitur - Normalisasi	9
2.5.2. Perhitungan Similaritas	9
BAB 3	10
3.1. Arsitektur Website (Frontend)	10
3.2. Arsitektur Program Information Retrieval (Backend)	11
BAB 4	14
4.1. Halaman Utama Website	14
4.1.1. Fitur halaman utama website	14
4.2. Pencarian Menggunakan Image	15
4.2.1. Pencarian menggunakan query image yang sudah dimodifikasi	16
4.3. Pencarian Menggunakan Audio	17
4.4. Tampilan mapper pada website	19
BAB 5	20
DAFTAR PUSTAKA	22
TAUTAN REPOSITORY	24
TAUTAN VIDEO	24

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. Algoritma yang digunakan adalah Eigenvalue, Cosine Similarity, Euclidean Distance, dll.

Matriks adalah salah satu komponen yang penting dalam aplikasi aljabar vektor. Di dalam Tugas Besar 2 ini, dibuat sebuah aplikasi yang meminta input lagu dan aplikasi tersebut mendeteksi apa nama dari lagu tersebut dan beberapa detail lainnya. Dalam implementasinya digunakan aljabar vektor untuk mencari perbandingan antar satu audio dengan audio yang lain. Konsep yang digunakan adalah *Music Information Retrieval* atau MIR untuk mencari dan mengidentifikasi suara berdasarkan fitur-fitur yang dimilikinya. Konsep Principal Component Analysis (PCA) juga digunakan untuk mencari kumpulan audio melalui query gambar.

1.2. Information Retrieval

Information Retrieval adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Pada tugas besar ini, digunakan dua konsep *Information Retrieval*, yakni *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, *Image Retrieval* diimplementasikan menggunakan Principal Component Analysis dan *Music Information Retrieval* diimplementasikan menggunakan humming.

BAB 2

TEORI SINGKAT

2.1. Album Picture Finder - Principal Component Analysis

Pencarian gambar album memanfaatkan teknik Principal Component Analysis (PCA) untuk membandingkan kesesuaian gambar album dalam dataset dengan query. Sebelum melakukan PCA, nama gambar album dipetakan dengan nama audio yang bersangkutan. Dalam tugas ini, digunakan file .json untuk melakukan pemetaan.

```
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 metode analisis statistik yang digunakan untuk mereduksi dimensi data sambil mempertahankan informasi terpenting. Data yang didapatkan dari hasil PCA adalah berupa eigenvector dan proyeksi data. Berikut adalah langkah-langkah untuk melakukan pencarian gambar menggunakan PCA.

2.1.1. Image Processing and Loading

Seluruh gambar dalam dataset dilakukan pemrosesan yakni diubah menjadi *grayscale* untuk mengurangi kompleksitas gambar dengan intensitas pixel yang dinyatakan dengan persamaan berikut.

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

Gambar diubah ukurannya agar sama untuk seluruh gambar dalam dataset dan diubah menjadi 1D.

2.1.2. Data Centering (Standardization)

Data distandarisasi di sekitar nilai 0. Rata-rata dari suatu pixel pada posisi tertentu untuk setiap gambar dihitung dengan persamaan berikut.

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

di mana:

x_{ij} : nilai piksel ke-j pada gambar ke-i,

N: jumlah total gambar dalam dataset.

Kemudian, pixel pada posisi tersebut dikurangi dengan rata-rata yang sudah dihitung untuk melakukan standarisasi atau *mean centering*.

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

2.1.3. PCA Computation Using Singular Value Decomposition (SVD)

Komputasi PCA dilakukan menggunakan metode dekomposisi matriks SVD terhadap matriks kovarians dari data yang sudah distandarisasi.

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

di mana:

X' : matriks data yang sudah distandarisasi.

C = matriks kovarians

Dilakukan dekomposisi nilai singular terhadap matriks kovarians untuk mendapatkan komponen utama.

$$C = U \Sigma U^T$$

U : matriks eigenvector (komponen utama),

Σ : matriks eigenvalue (menunjukkan varian data di sepanjang komponen utama).

Sejumlah k komponen utama $k << M.N$ teratas dari hasil SVD diambil lalu sampel gambar dalam dataset diproyeksikan ke komponen utama.

$$Z = X' U_k$$

di mana:

U_k : matriks eigenvector dengan n -dimensi.

2.1.4. Similarity Computation

Kesamaan antara tiap gambar dalam dataset dengan gambar query ditentukan dengan mencari jarak euclidean. Kecocokan diurutkan dari yang paling tinggi. Gambar query diproyeksikan dalam ruang komponen utama terlebih dahulu.

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

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 piksel dari dataset (per piksel).

U_k : Matriks eigenvector dengan k dimensi utama dari PCA.

Kemudian, setiap gambar dalam dataset dicari jarak euclideananya terhadap gambar query menggunakan persamaan berikut.

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

$d(q, z_i)$ = 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.

2.1.5. Retrieval and Output

Gambar-gambar yang mirip dengan query masukan dikumpulkan dengan cara membatasi jumlah atau dengan memberikan batas jarak euclidean. Hasil pencarian gambar dapat digabungkan dengan hasil pencarian suara atau dijalankan sendiri.

2.2. Music Information Retrieval - Query by Humming

Pencarian lagu menggunakan query humming melewati beberapa langkah yaitu: Pemrosesan audio, yang merupakan proses persiapan suara yang diinput untuk tahap selanjutnya. Kemudian, data audio melewati tahap ekstraksi fitur vektor di mana data audio dikonversi menjadi representasi numerik berupa vektor fitur yang dapat dianalisis. Terakhir, vektor fitur yang dihasilkan dibandingkan dengan dataset untuk menemukan hasil yang paling sesuai dengan query yang diberikan.

2.2.1. Pemrosesan Audio

Tipe file yang digunakan dalam Music Information Retrieval menggunakan adalah MIDI. Masing-masing file MIDI diproses dengan dengan metode windowing yang membagi melodi menjadi segmen sepanjang 20-40 ketukan dengan sliding window sepanjang 4-8 ketukan. Teknik ini

memungkinkan pencocokan yang lebih fleksibel dari berbagai bagian lagu yang mungkin diingat oleh pengguna.

Proses windowing diikuti dengan normalisasi tempo dan pitch yang bertujuan untuk mengurangi variasi humming. Setiap note event dikonversi menjadi representasi numerik yang mempertimbangkan durasi dan urutan nada sehingga memungkinkan sistem untuk membandingkan potongan melodi dengan database. Normalisasi tempo menggunakan formula $NP(note) = \frac{(note-\mu)}{\sigma}$. dimana μ adalah rata rata dari pitch dan σ adalah standar deviasi dari pitch.

2.2.2. Ekstraksi Fitur - Distribusi tone

Distribusi tone terbagi atas tiga viewpoints yaitu:

2.3. Fitur Absolute Tone Based (ATB)

Setiap kemunculan nada yang berfrekuensi 0-127 dalam skala MIDI akan dicatat menjadi sebuah histogram yang menunjukkan gambaran distribusi absolut nada dalam data. Di mana histogram awalnya diset menjadi 128 bin dan setiap kemunculan frekuensi nada MIDI (1-127) akan dicatat. Terakhir histogram akan dinormalisasikan untuk mendapatkan distribusi yang terstandarisasi.

2.4. Fitur Relative Tone Based (RTB)

Perubahan antara nada-nada berturut-turut dianalisis dan dicatat dalam bentuk histogram dengan nilai berkisar dari -127 hingga +127, menggambarkan pola interval melodi secara relatif. Histogram ini, yang terdiri dari 255 bin, dibangun dengan mencatat selisih antar nada dalam data. Setelah histogram terbentuk, langkah terakhir adalah menormalisasikannya untuk menghasilkan distribusi interval yang terstandarisasi.

2.5. Fitur First Tone Based (FTB)

Setiap nada dibandingkan dengan nada pertama untuk menciptakan histogram yang merepresentasikan hubungan relatif terhadap titik referensi awal. Metode ini bertujuan untuk menangkap struktur nada yang lebih stabil terhadap perbedaan pitch. Histogram dibangun

dengan 255 bin, mencakup nilai dari -127 hingga +127. Selisih antara nada-nada dengan nada pertama dihitung, dan histogram yang dihasilkan dinormalisasi untuk mendapatkan distribusi yang seimbang dan terstandarisasi.

2.5.1. Ekstraksi Fitur - Normalisasi

Normalisasi bertujuan untuk memastikan bahwa semua nilai dalam histogram berada dalam skala probabilitas, sehingga total nilai pada histogram sama dengan 1. Rumus yang digunakan untuk normalisasi adalah $H_{norm} = \frac{H[d]}{\sum_d H[d]}$. Dengan H adalah Histogram dan d adalah bin dari histogram.

2.5.2. Perhitungan Similaritas

Setiap histogram diubah menjadi sebuah vektor dan penghitungan kemiripan, dilakukan dengan menggunakan metode **Cosine Similarity**, yaitu sebuah ukuran yang menentukan tingkat kesamaan antara dua vektor dengan cara menghitung sudut cosinus di antara keduanya dalam ruang berdimensi tinggi sehingga semakin kecil sudut cosinus antara kedua vektor maka tingkat kemiripan keduanya pun semakin tinggi. Persamaan yang digunakan untuk menghitung cosine similarity adalah

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

BAB 3

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

3.1. Arsitektur Website (Frontend)

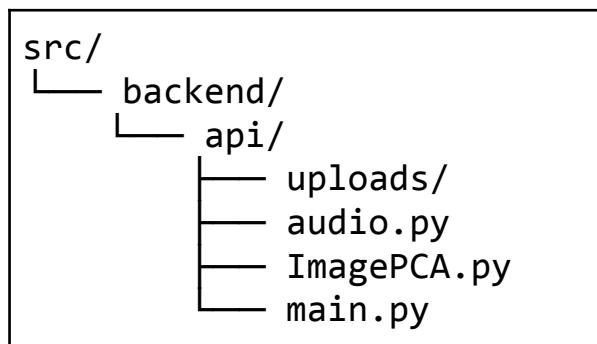
```
src/
└─frontend/
    ├── .next/
    ├── node_modules/
    ├── public/
    ├── src/
    ├── .env
    ├── .env.example
    ├── .eslintrc.json
    ├── .gitignore
    ├── bun.lockb
    ├── components.json
    ├── next-env.d.ts
    ├── next.config.js
    ├── package.json
    ├── postcss.config.mjs
    ├── tailwind.config.ts
    └── tsconfig.json
```

Frontend proyek ini dibuat menggunakan Next.js, sebuah framework berbasis React yang mempermudah pengembangan antarmuka pengguna (UI) modern. Framework ini dilengkapi fitur-fitur seperti Server-Side Rendering dan Static Site Generation, yang dapat meningkatkan performa. Dengan sistem routing berbasis file, struktur proyek menjadi lebih rapi dan mudah dikelola. Bahasa pemrograman yang digunakan adalah TypeScript.

Untuk styling, kami menggunakan Tailwind CSS, sebuah framework utility-first yang memungkinkan pengembangan antarmuka secara cepat dan efisien

dengan memanfaatkan kelas utilitas langsung di dalam komponen. Dengan pendekatan ini, kebutuhan menulis file CSS tradisional dapat diminimalkan. Selain itu, proyek ini juga menggunakan ESLint untuk menjaga kualitas dan konsistensi kode, serta Bun sebagai package manager. Bun dipilih karena kecepatannya yang lebih unggul dibandingkan npm atau Yarn, sehingga mendukung proses pengembangan yang lebih efisien.

3.2. Arsitektur Program Information Retrieval (Backend)



Program Information Retrieval menggunakan bahasa pemrograman Python dan proses retrieval dibagi atas 2 bagian yaitu audio retrieval dan image retrieval.

```
@app.post("/find_similar_images")
async def find_similar_images(query_image: UploadFile, k: int = Query(10, gt=0)):
    # **Initial State**:
    # - Direktori "query" berisi file-file lama dari proses sebelumnya (jika ada).
    # - Gambar query belum tersimpan.
    # - Dataset gambar dalam direktori "images" tersedia tetapi belum diproses.
    # - Model PCA belum dibuat atau dilatih.
    # - Tidak ada hasil kemiripan yang disimpan dalam cache.

    # **Final State**:
    # - Direktori "query" telah dibersihkan dan hanya berisi gambar query terbaru.
    # - Gambar query berhasil diproses dan dibandingkan dengan dataset gambar.
```

```

# - Model PCA telah dilatih menggunakan dataset gambar yang
# diproses.
# - Kemiripan antara gambar query dan dataset dihitung dan
# hasilnya disusun.
# - Cache diperbarui dengan daftar gambar yang paling mirip
# beserta metadatanya.
# - Waktu eksekusi untuk preprocessing, PCA fitting, dan query
# dikembalikan sebagai respons.
pass

```

```

@app.post("/find_similar_audio")
async def find_similar_audio(query_audio: UploadFile):
    # **Initial State**:
    # - Direktori "query" belum ada atau berisi file-file lama
    #   (jika ada).
    # - File audio query belum ada di sistem.
    # - Direktori pencarian audio ("uploads/audio") berisi file
    #   audio yang akan dibandingkan.
    # - Tidak ada hasil kemiripan yang disimpan dalam cache.

    # **Final State**:
    # - Direktori "query" telah dibuat (jika belum ada) dan berisi
    #   file audio query terbaru.
    # - File audio query disimpan di direktori "query".
    # - Fungsi `get_similar_audio` mencari dan menghitung
    #   kemiripan antara audio query dengan file audio dalam
    #   direktori pencarian.
    # - Cache diperbarui dengan hasil audio yang paling mirip
    #   beserta persentase kemiripan dan metadata terkait.
    # - Waktu eksekusi pencarian audio dihitung dan dikembalikan
    #   sebagai respons dalam format milidetik.

pass

```

Untuk pengambilan data gambar, library seperti PIL digunakan untuk memproses gambar, termasuk mengubah matriks RGB menjadi vektor, sedangkan NumPy digunakan untuk operasi matematis dan manipulasi data numerik. Di sisi lain, pemrosesan data audio menggunakan library seperti Mido. Selain itu, NumPy dan modul math dimanfaatkan untuk menghitung metrik seperti mean, standar deviasi,

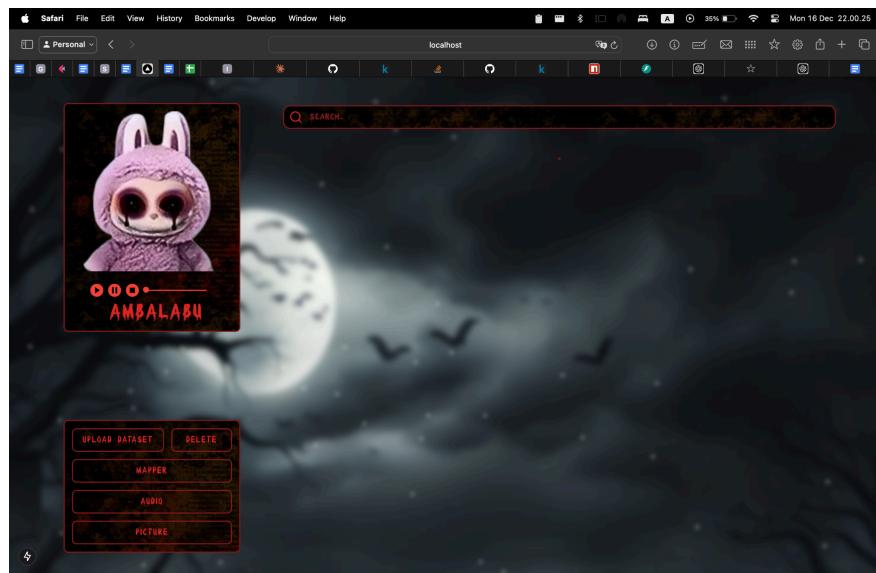
dan histogram nada yang meliputi Absolute Tone-Based (ATB), Relative Tone-Based (RTB), serta First Tone-Based (FTB).

Manajemen file dan integrasi database menjadi komponen penting dalam pengembangan website ini. Library seperti zipfile digunakan untuk menangani file terkompresi, sedangkan shutil dan os dimanfaatkan untuk membuat folder sementara guna menyimpan file yang diunduh dari database. Untuk mencatat durasi eksekusi program, digunakan library time. Pada sisi Front End, diperlukan sejumlah API baru untuk menangani berbagai fungsi seperti pencarian, mapper, lagu, gambar, dan lainnya, sehingga penggunaan API seperti FastAPI untuk koneksi dengan Back End tidak diperlukan. Selain itu, beberapa file .ts juga diperlukan, misalnya untuk memutar (play) file MIDI, yang pengintegrasianya cukup menantang.

BAB 4

EKSPERIMEN

4.1. Halaman Utama Website



Gambar 2. Tampilan halaman utama website ambalabu

4.1.1. Fitur halaman utama website

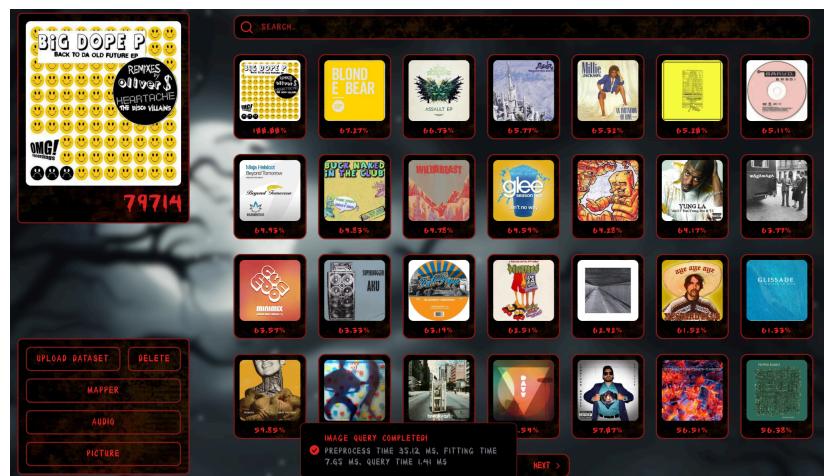


Gambar 3. Audio player

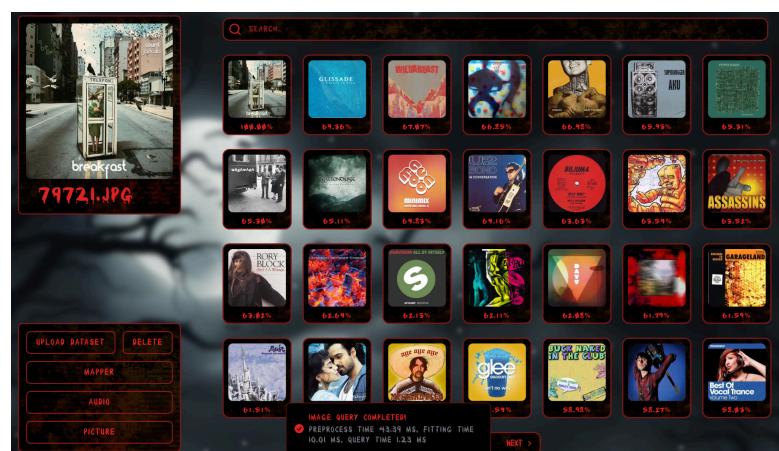
4.2. Pencarian Menggunakan Image



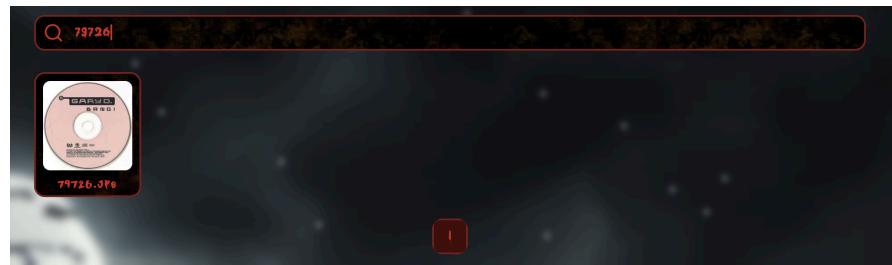
Gambar 4. Pencarian menggunakan query gambar 1



Gambar 5. Pencarian menggunakan query gambar 2



Gambar 6. Pencarian menggunakan query gambar 3

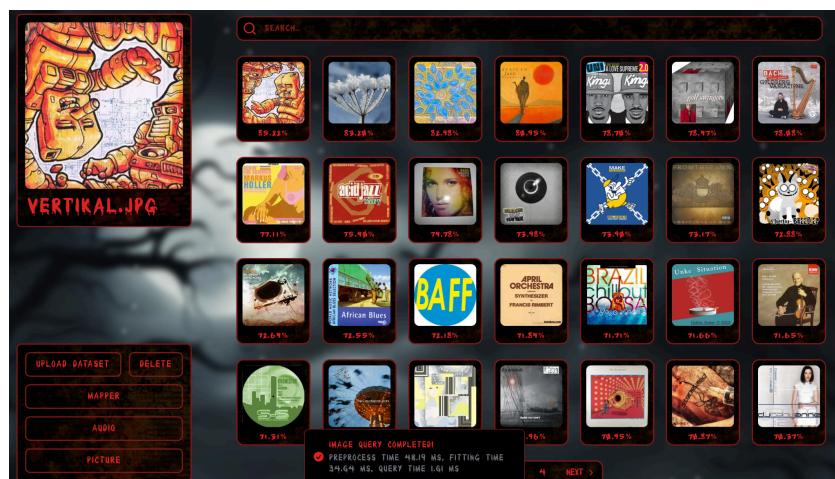


Gambar 7. Pencarian gambar menggunakan search bar

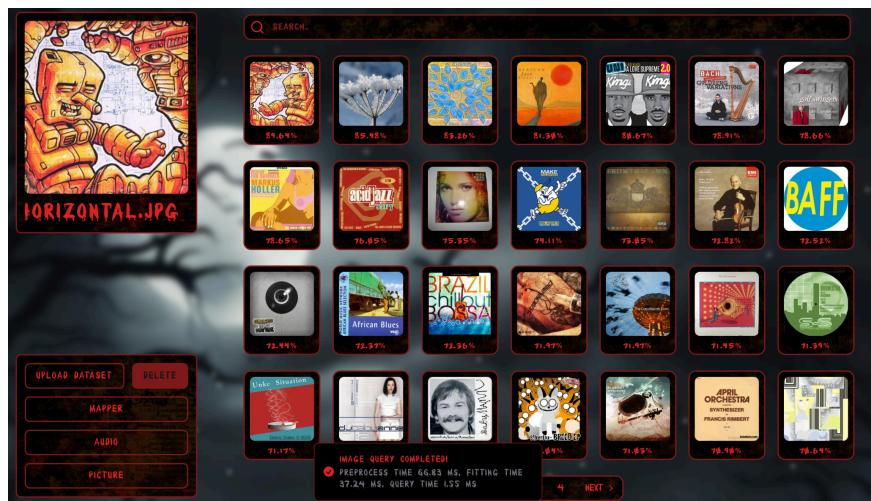
4.2.1. Pencarian menggunakan query image yang sudah dimodifikasi



Gambar 8. Pencarian menggunakan query gambar yang dicrop

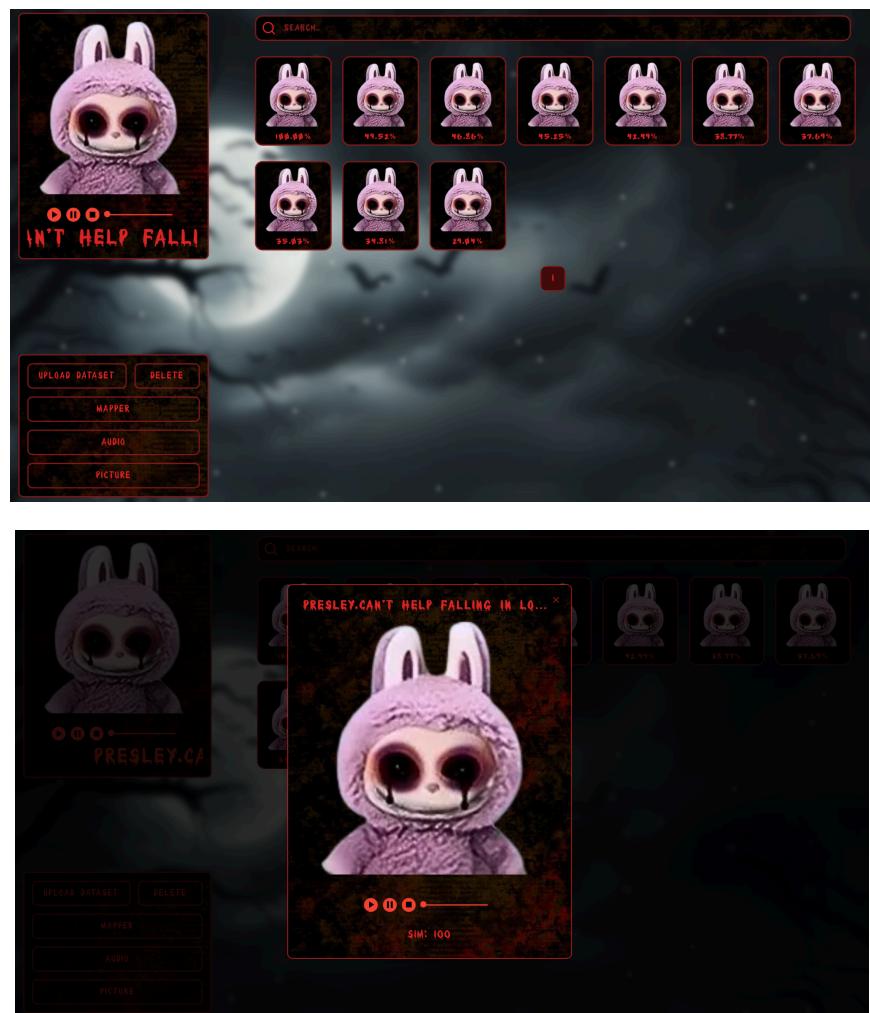


Gambar 9. Pencarian menggunakan query gambar yang diputar secara vertikal



Gambar 10. Pencarian menggunakan query gambar yang diputar secara horizontal

4.3. Pencarian Menggunakan Audio



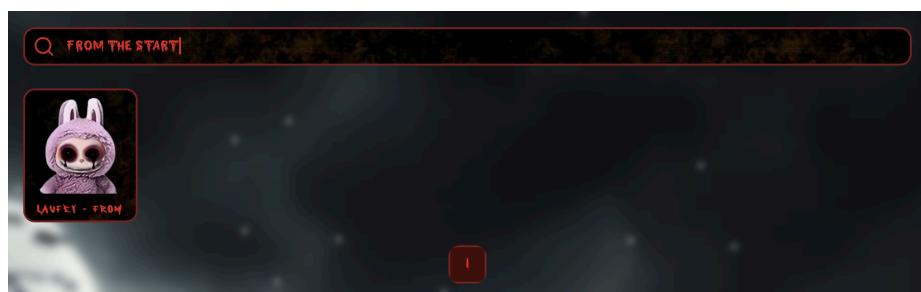
Gambar 11. Pencarian menggunakan query audio 1



Gambar 12. Pencarian menggunakan query audio 2

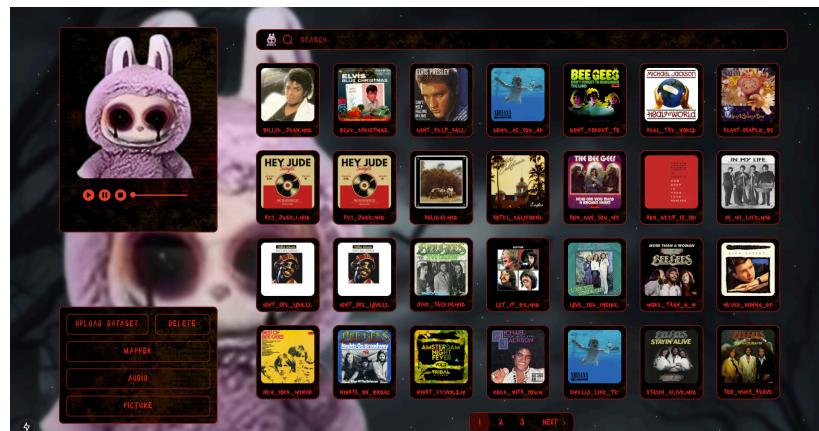


Gambar 13. Pencarian menggunakan query audio 3

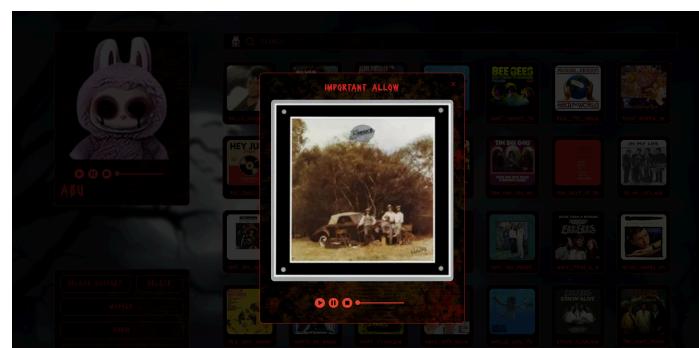
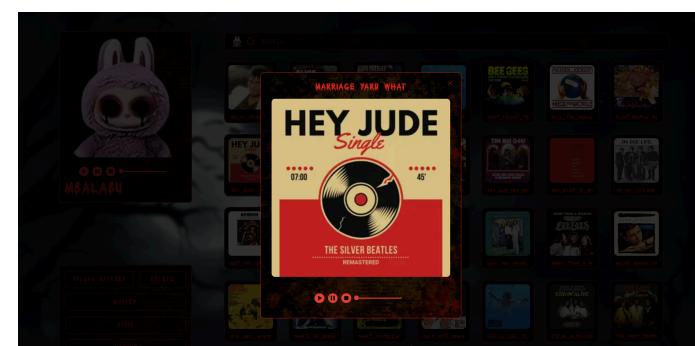


Gambar 14. Pencarian audio menggunakan search bar

4.4. Tampilan mapper pada website



Gambar 15. Tampilan mapper pada website



```
{  
    "name": "marriage yard what",  
    "audio_file": "Hey_Jude.mid",  
    "pic_name": "Hey_Jude.jpg"  
},  
{  
    "name": "important allow",  
    "audio_file": "Holiday.mid",  
    "pic_name": "Holiday.jpg"  
},
```

Gambar 16. Rincian mapper dalam website

BAB 5

KESIMPULAN, SARAN, KOMENTAR, DAN REFLEKSI

5.1. Kesimpulan

Sistem *Image Retrieval* diimplementasikan dengan beberapa tahap. Pertama, dataset gambar diproses agar menjadi *grayscale*, memiliki ukuran sama untuk seluruh gambar dalam dataset, serta menjadi 1D. Setelah melalui pemrosesan, dataset gambar distandarisasi dengan metode *mean centering*. Data yang telah *centered* didekomposisi menggunakan metode SVD untuk memperoleh matriks eigenvector (U). Gambar query yang telah diproses dan setiap data gambar dalam dataset diproyeksikan ke ruang k komponen utama lalu kemiripan setiap gambar dataset dengan query ditentukan dengan mencari jarak euclidean komponen utamanya.

Sistem *Music Information Retrieval* diimplementasikan dengan mengekstrak fitur-fitur dari sebuah audio yang meliputi fitur *Absolute Tone Based* (ATB), *Relative Tone Based* (RTB), dan *First Tone Based* (FTB). Sebelum melakukan ekstraksi fitur, file audio MIDI diproses dengan metode windowing disertai normalisasi tempo dan pitch. Fitur-fitur hasil ekstraksi yang sudah dibuat menjadi histogram dinormalisasi. Vektor histogram dari audio query dan audio dataset ditentukan kemiripannya menggunakan *cosine similarity*.

Sistem *Image Retrieval* dan *Music Information Retrieval* diintegrasikan dengan sebuah website bertema horor yang dibangun di atas framework Next.js pada frontend dan FastAPI pada backend. Pengguna dapat mengunggah dataset gambar dan audio berupa file .zip serta file mapper dengan format .json yang menghubungkan file gambar dengan audio. Setelah pengunggahan, pengguna dapat melakukan query dengan mengunggah file gambar atau audio sebagai query lalu website akan menampilkan daftar gambar beserta lagu yang telah dipetakan dengan urutan kemiripan menurun dari yang paling mirip. Website juga dapat menampilkan informasi lain seperti jarak euclidean, persentase kemiripan, nama file gambar dan audio, gambar album, waktu yang dibutuhkan untuk pemrosesan dan *querying*, dan keberhasilan query, serta memutar audio yang telah diunggah.

5.2. Saran

Setelah proses pengerjaan tugas besar 2 Aljabar Linier dan Geometri IF2123, tim pengembang memiliki beberapa saran perbaikan yang diharapkan dapat diterapkan oleh para pengembang atau peneliti lain kedepannya:

1. Melakukan eksplorasi lebih dalam mengenai pemrosesan audio untuk *Music Information Retrieval* untuk meningkatkan efisiensi sistem.
2. Mengekstrak fitur lain dalam sistem *Image Retrieval* (seperti fitur RGB, alpha channel, dll.) untuk memperoleh hasil query yang lebih akurat.
3. Website yang responsif agar sistem dapat digunakan pada gawai *smartphone*.

5.3. Komentar

Felix: Lihat belakang Dong 😊.

Hima: Ambalabu 🤪.

Karol: Jangan lihat belakang 🦸.

5.4. Refleksi

Pembuatan sebuah website untuk image dan audio retrieval menggunakan Next.js dan FastAPI melibatkan beberapa aspek penting, mulai dari pengembangan frontend dan backend, hingga implementasi teknik pemrosesan data seperti SVD (Singular Value Decomposition) dan PCA (Principal Component Analysis) untuk retrieval gambar, serta ATB (Auto-Tagged Base), RTB (Reference Time-Based), dan FTB (Feature Time-Based) untuk retrieval audio. Selain kemampuan teknis, tim pengembangan juga belajar untuk bekerja sama, mengatur waktu dengan baik, serta semangat untuk bereksplorasi terhadap teknologi baru.

DAFTAR PUSTAKA

- [1] 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," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan, 2001, pp. 404-407. [Online]. Available: <https://www.comp.nus.edu.sg/~cs4241/00799561.pdf>
- [2] R. E. Madsen, L. K. Hansen, and O. Winther, "Singular Value Decomposition and Principal Component Analysis," Technical Report IMM-2004-4000, Informatics and Mathematical Modelling, Technical University of Denmark, Feb. 2004. [Online]. Available: <https://www2.imm.dtu.dk/pubdb/doc/imm4000.pdf>
- [3] R. Munir, "Aplikasi Dot Product pada Sistem Temu-balik Informasi (Information Retrieval System)," Institut Teknologi Bandung, 2024. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-14-Aplikasi-dot-product-pada-IR-2023.pdf>
- [4] R. Munir, "Singular Value Decomposition (SVD) (Bagian 1)," Institut Teknologi Bandung, 2024. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-21-Singular-value-decomposition-Bagian1-2023.pdf>
- [5] R. Munir, "Singular Value Decomposition (SVD) (Bagian 2)," Institut Teknologi Bandung, 2024. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-22-Singular-value-decomposition-Bagian2-2023.pdf>
- [6] T. Segaran, *Programming Collective Intelligence: Building Smart Web 2.0 Applications*, Sebastopol, CA, USA: O'Reilly Media, 2007.

TAUTAN REPOSITORY

<https://github.com/aibrahim185/Algeo02-23012>

TAUTAN VIDEO

https://youtu.be/9Lb_-EInrmM