

LAPORAN TUGAS BESAR 2
IF2123 ALJABAR LINEAR DAN GEOMETRI

Image Retrieval dan Music Information Retrieval
Menggunakan PCA dan Vektor



Disusun oleh :

Dzaky Aurelia Fawwaz 13523065

Carlo Angkisan 13523091

Ferdin Arsenarendra Purtadi 13523117

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESHA 10, BANDUNG 40132

2024

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
1.1 Tujuan.....	3
1.2 Spesifikasi Program.....	3
BAB II.....	5
2.1 Sistem Temu Balik Suara (MIR).....	5
2.2 Metode Ekstraksi Fitur Berdasarkan Humming.....	5
2.3 Image Retrieval dengan PCA.....	7
BAB III.....	9
3.1 Arsitektur Website (Frontend).....	9
3.2 Arsitektur Program Information Retrieval (Backend).....	10
3.2.1 Overview.....	10
3.2.2 Komponen Utama Arsitektur.....	11
3.2.2.1 Input Layer.....	11
3.2.2.2 Preprocessing Layer.....	12
3.2.2.3 Feature Layer.....	12
3.2.2.4 Processing Layer.....	12
3.2.2.5 Matching Layer.....	13
3.2.2.6 Metadata Layer.....	13
3.2.2.7 Output Layer.....	13
3.2.3. Alur Proses.....	14
BAB IV.....	15
4.1 Hasil Eksperimen Album Picture Finder - Principal Component Analysis.....	15
4.2 Analisis Hasil Eksperimen Album Picture Finder - Principal Component Analysis.....	17
4.3 Hasil Eksperimen Music Retrieval Query by Humming.....	18
4.4 Analisis Hasil Eksperimen Music Retrieval Query by Humming.....	21
BAB V.....	23
5.1 Kesimpulan.....	23
5.2 Saran.....	23
5.3 Refleksi.....	23
5.4 Komentar.....	24
LAMPIRAN.....	25

BAB I

DESKRIPSI MASALAH

1.1 Tujuan

Tugas besar ini dibuat untuk memenuhi tugas besar 2 mata kuliah IF2123 Aljabar Linear dan Geometri. Tujuan dari tugas besar ini adalah sebagai berikut.

1. Mengimplementasikan Sistem Temu Balik Informasi (Information Retrieval)
Membuat aplikasi yang dapat melakukan pencarian gambar berdasarkan input gambar (Image Retrieval menggunakan PCA) dan pencarian suara berdasarkan input audio atau humming (Music Information Retrieval menggunakan fitur melodi).
2. Mengembangkan Algoritma Ekstraksi Fitur
Menggunakan metode Principal Component Analysis (PCA) untuk mereduksi dimensi data gambar dan mempercepat proses pencocokan berdasarkan fitur visual serta menerapkan metode ekstraksi fitur audio seperti Absolute Tone Based (ATB), Relative Tone Based (RTB), dan First Tone Based (FTB) untuk menggambarkan karakteristik melodi dari input humming.
3. Mengukur Similaritas Data
Menerapkan metode seperti Cosine Similarity untuk menghitung tingkat kemiripan antara input dengan data dalam dataset, baik untuk gambar maupun audio dengan menggunakan pembobotan pada fitur audio (ATB, RTB, FTB) untuk meningkatkan akurasi pencocokan humming dengan data dalam dataset.
4. Meningkatkan Pemahaman Terhadap Aljabar Linear dan Geometri
Mengaplikasikan konsep-konsep aljabar linear seperti dekomposisi nilai singular (Singular Value Decomposition/SVD) untuk PCA dan analisis vektor dalam sistem temu balik informasi.

1.2 Spesifikasi Program

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](https://www.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
10. 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.
11. Maksimum nilai dari tugas besar ini adalah 115. Biarpun nilai total kalian lebih tinggi karena bonus, nilai kalian akan tetap 115. Silahkan kerjakan bonus jika merasa ingin mengcover nilai spek wajib atau ingin menambah nilai saja.

BAB II

TEORI SINGKAT

2.1 Sistem Temu Balik Suara (MIR)

Music Information Retrieval (MIR) adalah bidang yang bertujuan untuk mencari, menganalisis, dan mengelompokkan informasi dari data audio. Dalam sistem MIR, input berupa audio (lagu, humming, atau suara lainnya) diproses untuk mencocokkan informasi dalam dataset berdasarkan fitur audio yang diekstraksi. Komponen utama MIR meliputi Input Audio dan Processing.

Proses MIR:

1. Ekstraksi fitur audio
 - a. Fitur seperti ATB, RTB, dan FTB digunakan untuk menggambarkan karakteristik audio.
 - b. Histogram fitur distandarkan menjadi distribusi probabilitas.
2. Perhitungan Similaritas

Similaritas antara input dan dataset dihitung menggunakan metrik seperti Cosine Similarity :

$$\text{CosSim}(q, z_i) = \frac{\sum_{j=1}^k q_j z_{ij}}{\sqrt{\sum_{j=1}^k q_j^2} \cdot \sqrt{\sum_{j=1}^k z_{ij}^2}}$$

di mana q adalah fitur input, dan z adalah fitur dataset.

3. Output

Hasil pencocokan diurutkan berdasarkan nilai similaritas tertinggi. Audio dengan tingkat kemiripan terbesar ditampilkan sebagai hasil.

2.2 Metode Ekstraksi Fitur Berdasarkan Humming

Humming adalah proses bersenandung melodi tanpa lirik atau kata-kata. Dalam konteks sistem temu balik suara, humming digunakan sebagai input untuk mencari lagu atau melodi yang mirip dalam dataset. Proses ini membutuhkan langkah-langkah spesifik untuk mengekstraksi fitur-fitur penting dari humming.

Pemrosesan Audio Humming :

1. Parsing input
 - Input berupa file MIDI yang mengandung informasi notasi melodi.
 - File diproses untuk mengidentifikasi nada-nada utama, mengabaikan suara drum atau elemen yang tidak relevan

2. Segmentasi

Humming dipisah menjadi beberapa segmen menggunakan sliding window. Tiap window memiliki ukuran tetap (misalnya 40 beat) dengan overlap antar window (misalnya 8 beat). Teknik ini memungkinkan fleksibilitas dalam mencocokkan humming dengan database

3. Normalisasi tempo

Tempo dinormalisasi untuk mengurangi pengaruh variasi kecepatan humming oleh pengguna. Durasi tiap nada disesuaikan menggunakan formula:

$$NP(note) = (note - \mu)/\sigma$$

di mana μ adalah rata-rata durasi, dan σ adalah standar deviasi durasi

Setelah normalisasi, fitur-fitur utama diekstraksi untuk mendeskripsikan humming:

1. Absolute Tone Based (ATB)

- Histogram distribusi nada absolut dihitung untuk menggambarkan frekuensi nada-nada tertentu (dalam rentang 0-127).
- Normalisasi dilakukan untuk memastikan histogram menjadi distribusi probabilitas:

$$H_{\text{norm}}[d] = \frac{H[d]}{\sum_{d=0}^{127} H[d]}$$

di mana $H[d]$ adalah frekuensi nada pada bin d .

2. Relative Tone Based (RTB)

- Histogram interval antar nada berurutan dihitung, dengan nilai rentang antara -127 hingga +127.
- Fitur ini mencerminkan pola interval melodi dan tidak bergantung pada pitch absolut

3. First Tone Based (FTB)

Histogram perbedaan nada terhadap nada pertama dihitung, mencerminkan struktur relatif humming terhadap titik referensi awal.

2.3 Image Retrieval dengan PCA

Image Retrieval adalah proses mencari dan menemukan gambar dalam data set berdasarkan input gambar query. Salah satu pendekatan yang digunakan adalah **Principal Component Analysis (PCA)**, sebuah metode statistik untuk mereduksi dimensi data sambil mempertahankan karakteristik utama. PCA sangat efektif untuk mengolah data berdimensi tinggi seperti gambar.

Proses PCA untuk Image Retrieval mencakup langkah-langkah berikut:

1. Pemrosesan Gambar
 - a. Konversi ke grayscale
Gambar berwarna diubah menjadi grayscale untuk menyederhanakan analisis.
Formula:
$$I(x,y) = 0.2989 \cdot R(x,y) + 0.5870 \cdot G(x,y) + 0.1140 \cdot B(x,y)$$
di mana $R(x,y)$, $G(x,y)$, dan $B(x,y)$ adalah nilai intensitas warna merah, hijau, dan biru pada piksel (x,y) .
 - b. Resizing
Semua gambar diubah ukurannya menjadi dimensi tetap (misalnya 64×64) agar dimensi data konsisten.
 - c. Flattening
Gambar direpresentasikan sebagai vektor 1D. Jika dimensi gambar adalah $M \times N$, hasil flattening akan berupa vektor dengan panjang $M \cdot N$.
$$I = [I_1, I_2, \dots, I_{M \cdot N}]$$

2. Standardisasi Data

Data gambar dinormalisasi agar terpusat di sekitar nol, dengan langkah:

- a. Hitung rata-rata piksel untuk seluruh data set

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

di mana: x_{ij} adalah nilai piksel ke- j pada gambar ke- i dan N adalah jumlah total gambar dalam dataset.

- b. Kurangi nilai rata-rata dari setiap piksel:

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

3. Perhitungan PCA

- a. Matriks Kovarians

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

Dimana X' adalah matriks data yang sudah distandarisasi.

b. Dekomposisi SVD

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

Dimana \mathbf{U} adalah matriks eigenvector (komponen utama), Σ adalah matriks eigenvalue (menunjukkan varian data di sepanjang komponen utama).

c. Reduksi Dimensi

Pilih k -komponen utama teratas ($k \ll M \cdot N$) dan proyeksikan data:

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

4. Representasi Query

Gambar query diproses dengan langkah yang sama seperti dataset:

- Grayscale, resizing, dan flattening.
- Standardisasi menggunakan rata-rata dataset.
- Proyeksikan ke ruang komponen utama:

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

5. Perhitungan Similaritas

Ukur kesamaan antara gambar query dan gambar dataset menggunakan metrik jarak, seperti:

- Jarak Euclidean

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

- Cos Similarity

Semakin kecil jarak atau semakin besar cosine similarity, semakin mirip gambar tersebut.

$$\text{CosSim}(q, z_i) = \frac{\sum_{j=1}^k q_j z_{ij}}{\sqrt{\sum_{j=1}^k q_j^2} \cdot \sqrt{\sum_{j=1}^k z_{ij}^2}}$$

di mana \mathbf{q} adalah vektor fitur query, dan \mathbf{z} adalah vektor fitur dalam dataset.

BAB III

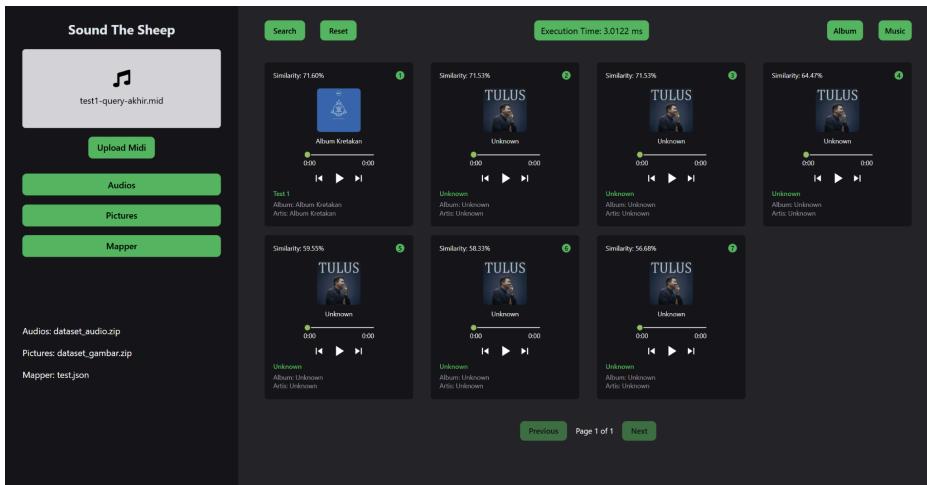
ARSITEKTUR WEBSITE DAN PROGRAM

3.1 Arsitektur Website (Frontend)

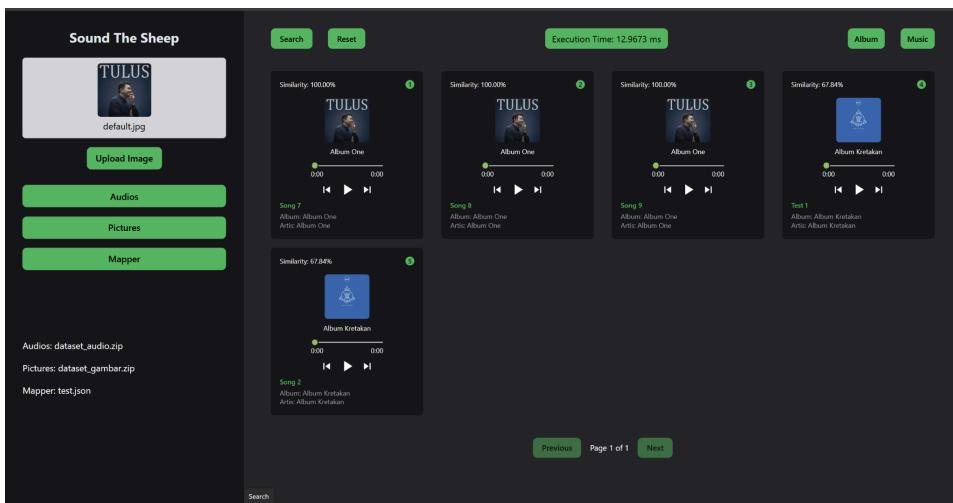
Arsitektur frontend website ini menggunakan framework Next.js sebagai fondasi utama pengembangan aplikasi web dengan menggunakan bahasa typescript. Styling antarmuka dilakukan menggunakan Tailwind CSS yaitu sebuah framework berbasis utilitas yang memungkinkan pengembangan desain secara cepat, konsisten, dan modular dengan kelas-kelas bawaan yang dapat dikustomisasi. Kombinasi Next.js dan Tailwind CSS mendukung pengembangan komponen UI yang efisien. Berikut arsitektur antarmuka website yang telah dibuat.

Arsitektur frontend meliputi dua mode yaitu :

1. Mode Music



2. Mode Album



Berikut komponen yang terdapat pada website Sound the Sheep :

- a. Judul Website : Sound The Sheep
- b. Tombol untuk mengunggah audio ketika dalam moda query audio by humming dan mengunggah gambar ketika dalam moda query album finder with PCA
- c. Tombol untuk mengunggah dataset gambar album dan audio.
- d. Tombol untuk memilih antara query audio by humming dan album finder by PCA
- e. Tombol untuk mengunggah mapper
- f. Daftar lagu beserta album gambar yang bersangkutan yang ada di dataset
- g. Pagination
- h. Tombol search untuk melakukan pencarian
- i. Tombol reset untuk mereset pencarian
- j. Persentase kemiripan pada setiap audio atau album yang ada pada dataset
- k. Waktu dari eksekusi program (dalam waktu ms).

3.2 Arsitektur Program Information Retrieval (Backend)

3.2.1 Overview

Berikut adalah gambaran arsitektur backend secara keseluruhan :

```

└── backend
    ├── album_finder
    │   └── album_finder.py
    ├── music_retriever
    │   ├── audioprocessing
    │   │   └── __init__.py
    │   │   └── audio_normaliz...
    │   │   └── extractor.py
    │   │   └── loader.py
    │   │   └── similarity.py
    │   └── audiotypes
    │       └── __init__.py
    │       └── audio_types.py
    │       └── __init__.py
    └── music_retriever.py

```

Arsitektur backend ini dirancang untuk mendukung aplikasi pencarian musik berbasis fitur audio dan gambar terkait. Sistem ini terdiri dari berbagai modul yang saling terintegrasi untuk mendukung proses pencarian, pengelolaan metadata, dan pengolahan data audio maupun gambar. Berikut adalah detail lengkap arsitekturnya:

3.2.2 Komponen Utama Arsitektur

3.2.2.1 Input Layer

1. Audio Loader

- Fungsi Utama: Memuat file audio (khususnya format MIDI) dan mengekstraksi data notasi.
- Komponen Utama:
 - load_audio_file(file_path: str)
 - load_midi_file(file_path: str)
- Output: Data not berupa objek Note (pitch, durasi, waktu mulai).

2. Image Loader

- Fungsi Utama: Memuat gambar dari dataset untuk diproses lebih lanjut.
- Komponen Utama:
 - load_and_preprocess_images(directory_path: str, image_size=(64, 64))
- Output: Dataset gambar dalam bentuk array yang sudah diproses.

3.2.2.2 Preprocessing Layer

1. Audio Normalizer

- a. Fungsi Utama: Melakukan normalisasi data audio, seperti durasi, tempo, dan membagi data menjadi jendela kecil.
 - b. Komponen Utama:
 - normalize_tempo(notes: List[Note])
 - apply_windowing(notes: List[Note], window_size: int, slide_size: int)
 - c. Output: Data audio yang dinormalisasi dan dibagi ke dalam jendela waktu tertentu.
2. **Image Preprocessor**
- a. Fungsi Utama: Standarisasi dataset gambar dalam hal ukuran dan intensitas pixel.
 - b. Komponen Utama:
 - standardize_images(image_data)
 - c. Output: Dataset gambar yang sudah distandarisasi.
3. **Extractor**
- a. Fungsi Utama: Mengekstrak fitur audio seperti ATB, RTB, dan FTB.
 - b. Komponen Utama:
 - extract_atb(notes: List[Note])
 - extract_rtb(notes: List[Note])
 - extract_ftb(notes: List[Note])
 - c. Output: Objek AudioFeatures berisi data fitur audio yang terestraksi.

3.2.2.3 Feature Layer

1. **Feature Repository**
- a. Fungsi Utama: Menyimpan fitur-fitur audio dan gambar dalam struktur data untuk pencocokan selanjutnya.
 - b. Komponen Utama:
 - Dictionary dataset_features: Dict[str, AudioFeatures]
 - c. Output: Database fitur audio dan metadata terkait.
2. **PCA Transformation**
- a. Fungsi Utama: Mengurangi dimensi data gambar menggunakan Principal Component Analysis (PCA).
 - b. Komponen Utama:
 - apply_pca(image_data, components_count: int)
 - c. Output: Dataset gambar dalam dimensi yang lebih kecil.

3.2.2.4 Processing Layer

1. **Similarity Calculator**
- a. Fungsi Utama: Menghitung kemiripan antara fitur audio query dan dataset menggunakan cosine similarity.
 - b. Komponen Utama:
 - cosine_similarity(v1: List[float], v2: List[float])

- calculate_weighted_similarity(query: AudioFeatures, dataset: AudioFeatures)
 - c. Output: Skor kemiripan (0-1).
- 2. **Query Processor**
 - a. Fungsi Utama: Memproses file query audio dan mempersiapkan data untuk pencocokan.
 - b. Komponen Utama:
 - process_query_file(root_dir: str, file_name: str)
 - c. Output: Objek AudioFeatures dari query.

3.2.2.5 Matching Layer

- 1. **Search Engine**
 - a. Fungsi Utama: Mencocokkan fitur query dengan dataset dan menghasilkan daftar hasil terurut.
 - b. Komponen Utama:
 - find_matches(query_features: AudioFeatures)
 - c. Output: List objek SearchResult.

3.2.2.6 Metadata Layer

- 1. **Metadata Mapper**
 - a. Fungsi Utama: Menghubungkan file audio dengan metadata seperti judul lagu, nama album, artis, dan gambar.
 - b. Komponen Utama:
 - set_mapping(mapping: Dict[str, Dict[str, Any]])
 - c. Output: Struktur data mapping metadata.
- 2. **File Info Provider**
 - a. Fungsi Utama: Mengambil informasi file berdasarkan nama file.
 - b. Komponen Utama:
 - get_file_info(audio_name: str)
 - c. Output: Metadata file seperti judul, gambar, nama album, dan artis.

3.2.2.7 Output Layer

- 1. **Search Result Formatter**
 - a. Fungsi Utama: Menyusun hasil pencarian dalam format terstruktur.
 - b. Komponen Utama:
 - rank_results(similarities: List[float])
 - c. Output: Daftar hasil pencarian yang terurut berdasarkan skor kemiripan.

3.2.3. Alur Proses

1. **Memuat Dataset**

File audio dan gambar dimuat melalui AudioLoader dan Image Loader. Data diolah menjadi fitur menggunakan Extractor dan PCA.

2. **Menyimpan Fitur**

Fitur yang telah diekstraksi disimpan dalam Feature Repository untuk digunakan dalam pencocokan.

3. **Memproses Query**

File query diproses menggunakan pipeline preprocessing yang sama untuk mendapatkan fitur query.

4. **Mencari Kemiripan**

Similarity Calculator membandingkan fitur query dengan dataset untuk menghasilkan skor kemiripan.

5. **Menghasilkan Hasil Pencarian**

Search Engine menyusun hasil pencarian dan menambahkan metadata menggunakan Metadata Mapper.

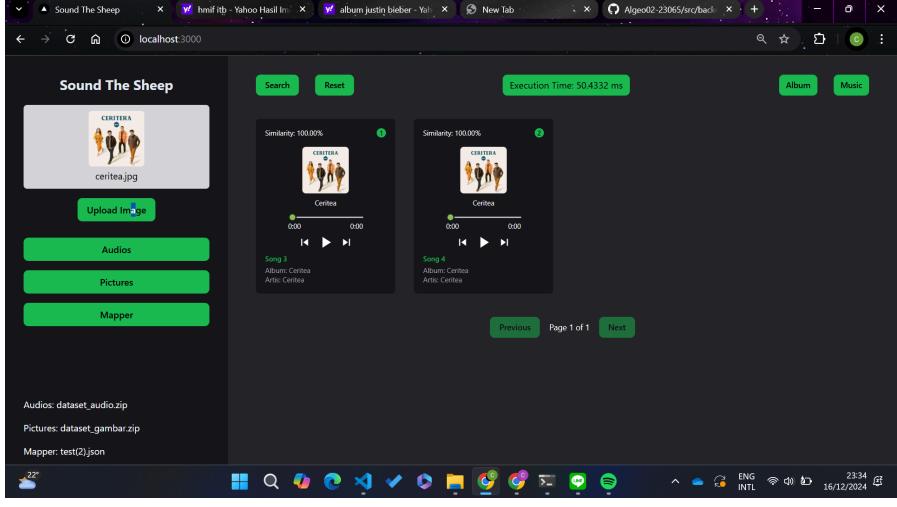
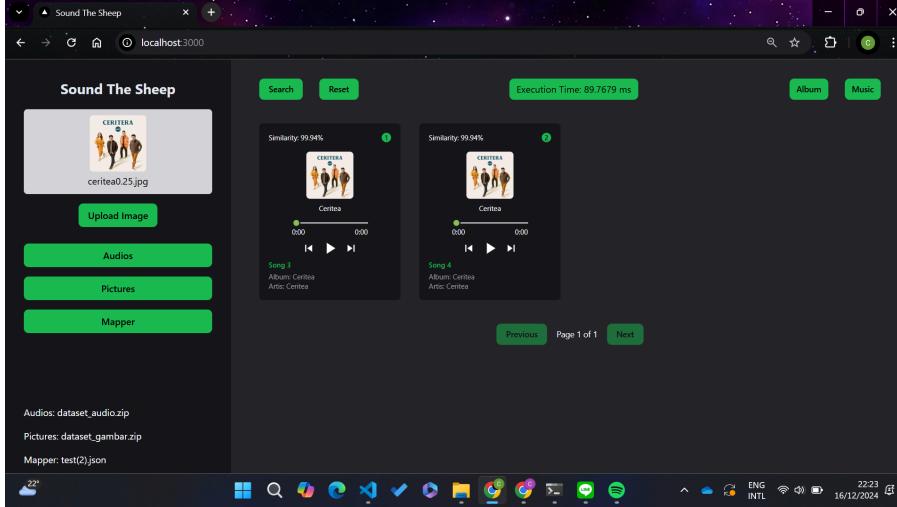
6. **Mengembalikan Hasil**

Sistem mengembalikan daftar hasil dalam format terstruktur melalui Search Result Formatter.

BAB IV

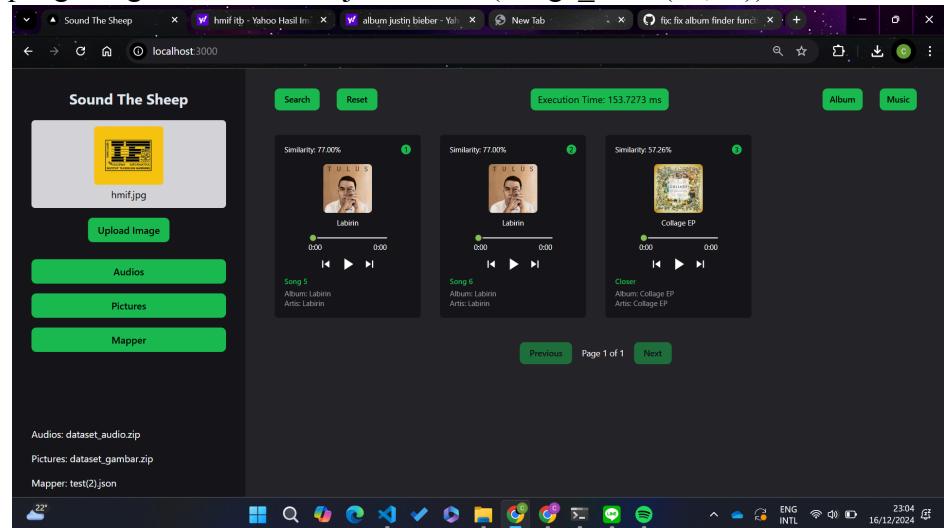
EKSPERIMENT

4.1 Hasil Eksperimen Album Picture Finder - Principal Component Analysis

Test Case 1	<p>Pengujian dengan salah satu <i>album picture</i> yang ada dalam dataset pengurangan dimensi menjadi 32x32 (image_size=(32,32))</p>  <p>Hasil Similaritas : 100 % Waktu Eksekusi : 50,43 ms</p>
Test Case 2	<p>Pengujian dengan salah satu <i>album picture</i> yang sama namun dengan ukuran yang berbeda pengurangan dimensi menjadi 32x32 (image_size=(32,32))</p>  <p>Hasil Similaritas : 99,94% Waktu Eksekusi : 89,76 ms</p>

Test Case 3

Pengujian dengan *album picture* yang tidak ada dalam dataset dengan pengurangan dimensi menjadi 32x32 (image_size=(32,32))

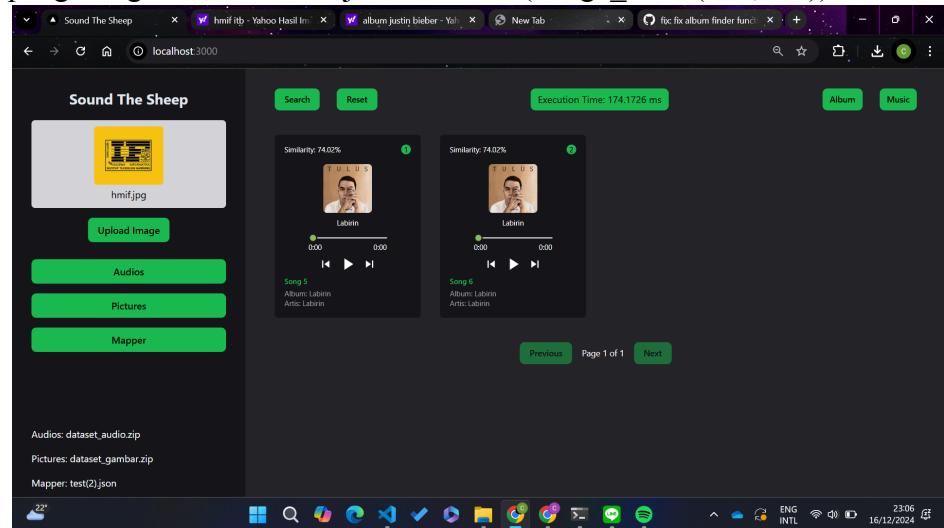


Hasil Similaritas : 77%

Waktu eksekusi : 153,72 ms

Test Case 4

Pengujian dengan *album picture* yang tidak ada dalam dataset dengan pengurangan dimensi menjadi 128x128 (image_size=(128,128))

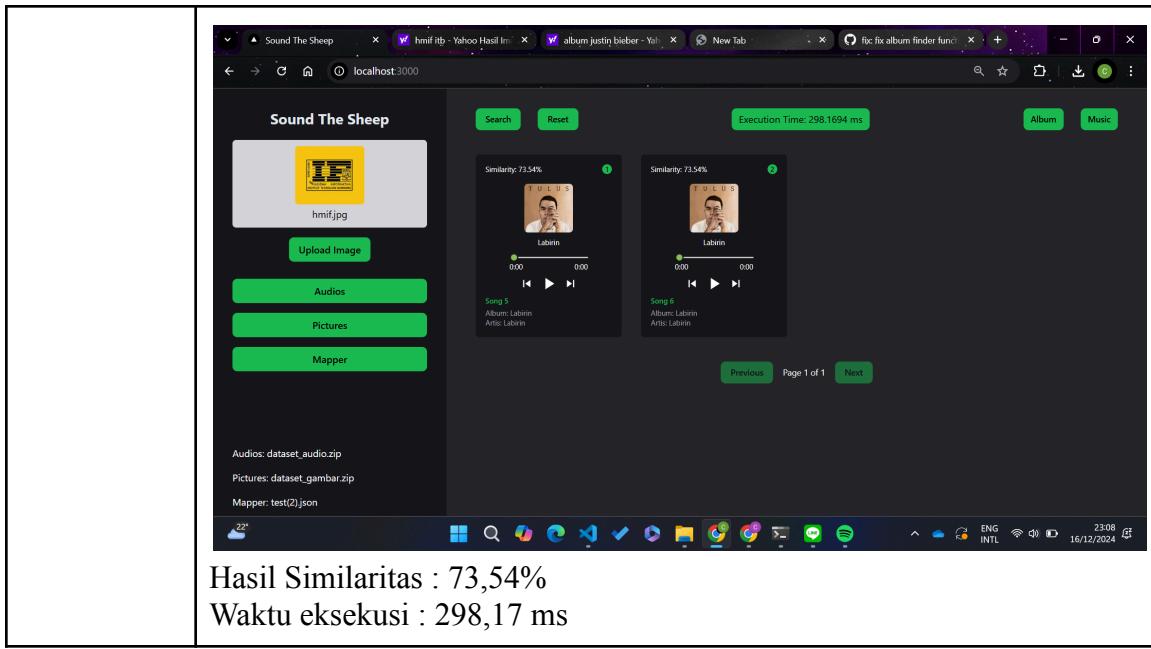


Hasil Similaritas : 74,02%

Waktu Eksekusi : 174,17 ms

Test Case 5

Pengujian dengan *album picture* yang tidak ada dalam dataset dengan pengurangan dimensi menjadi 256x256 (image_size=(256,256))



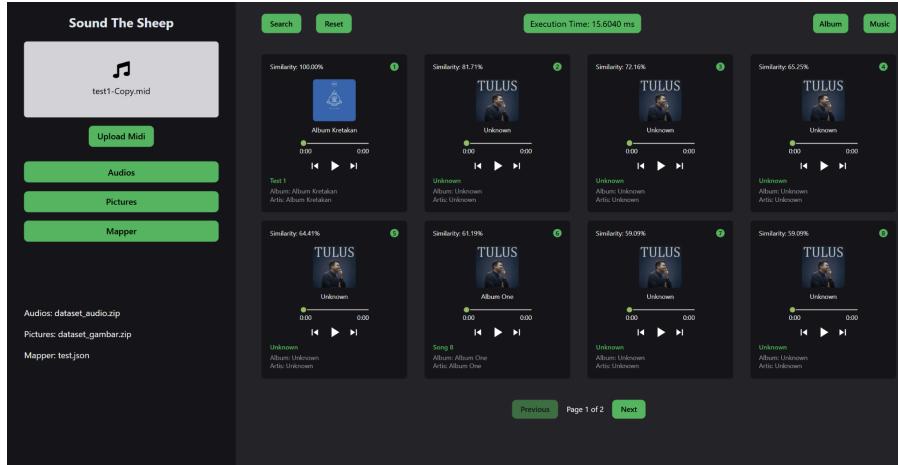
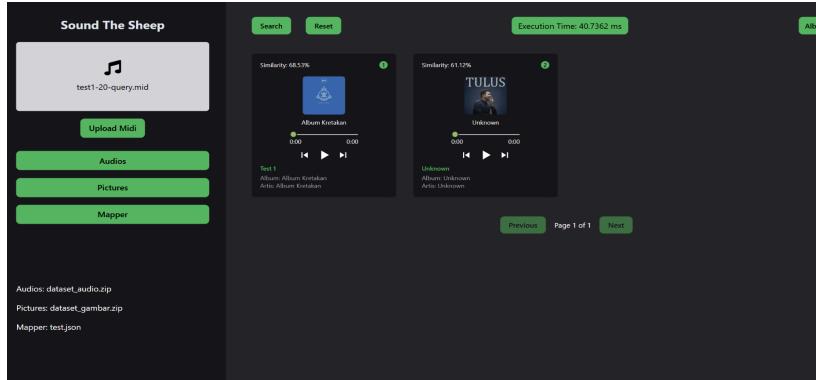
4.2 Analisis Hasil Eksperimen Album Picture Finder - Principal Component Analysis

Berdasarkan analisis implementasi dan hasil pengujian fitur pencarian gambar album menggunakan PCA, dapat disimpulkan beberapa hal penting. Implementasi yang dilakukan mencakup beberapa tahap utama yakni preprocessing gambar, implementasi inti PCA, dan perhitungan similaritas. Pada tahap preprocessing, gambar dikonversi menjadi grayscale dan diubah ukurannya menjadi dimensi standar (32x32, 128x128, atau 256x256), kemudian dilakukan flattening dari matriks 2D menjadi vektor 1D, serta standarisasi data dengan mengurangkan nilai rata-rata gambar. Implementasi inti PCA menggunakan Singular Value Decomposition (SVD) untuk menghitung komponen utama, melakukan reduksi dimensi melalui pemilihan komponen, dan transformasi gambar ke dalam ruang PCA. Untuk perhitungan similaritas, digunakan jarak Euclidean yang dinormalisasi menjadi skor similaritas dengan rentang 55 hingga 100%.

Hasil pengujian menunjukkan performa yang sangat baik untuk beberapa skenario. Pada pengujian dengan gambar yang sama (Test Case 1), diperoleh similaritas 100% yang menunjukkan recognition sempurna untuk gambar yang identik. Untuk pengujian dengan gambar yang sama namun ukuran berbeda (Test Case 2), diperoleh similaritas 99.94% yang menunjukkan bahwa sistem memiliki ketahanan terhadap variasi ukuran gambar. Pada pengujian dengan gambar yang berbeda menggunakan resolusi 32x32 (Test Case 3), diperoleh similaritas 77% dengan waktu eksekusi 153,72 milidetik. Sementara itu, pengujian dengan resolusi yang lebih tinggi (128x128 dan 256x256) pada Test Case 4 menghasilkan similaritas 74,02 dengan waktu eksekusi 174,17 milidetik dan Test Case 5 menghasilkan similaritas 73,54 dengan waktu eksekusi 298,17 milidetik. Dari beberapa percobaan tersebut didapatkan kesimpulan bahwa pengurangan dimensi gambar memiliki efek trade-off antara akurasi dan efisiensi. Resolusi rendah meningkatkan nilai similaritas

tetapi mengurangi kemampuan algoritma untuk membedakan gambar dengan detail tinggi. Di sisi lain, resolusi tinggi memberikan hasil yang lebih akurat namun memerlukan waktu komputasi yang lebih lama.

4.3 Hasil Eksperimen Music Retrieval Query by Humming

Test Case 1	<p>Query audio merupakan copy file dari file audio yang dituju</p>  <p>Hasil Similaritas : 100 % Waktu eksekusi : 15,6040 ms</p>
Test Case 2	<p>Query audio merupakan 20 detik pertama dari file audio yang dituju</p>  <p>Hasil Similaritas : 68,53 % Waktu eksekusi : 40,7362 ms</p>
Test Case 3	<p>Query audio merupakan potongan sebagian awal lagu dan akhir lagu</p>

	<p>The screenshot shows the Sound The Sheep application interface. On the left, there's a file upload section with 'test1-akhir-awal-note-dihilangkan.mid' selected, and buttons for 'Upload Midi', 'Audios', 'Pictures', and 'Mapper'. Below this are links to 'dataset_audio.zip', 'dataset_gambar.zip', and 'test.json'. On the right, the search results are displayed with execution time '8.0230 ms'. There are four results shown, each with a similarity score (71.97%, 68.21%, 68.23%, 64.59%), album ('Album Kretakan' or 'Unknown'), artist ('Artist Kretakan' or 'Unknown'), and a preview player.</p> <p>Hasil Similaritas : 71,97 % Waktu eksekusi : 8,0230 ms</p>
Test Case 4	<p>Query audio merupakan potongan 20 detik terakhir lagu</p> <p>The screenshot shows the Sound The Sheep application interface. On the left, there's a file upload section with 'test1-query-akhir.mid' selected, and buttons for 'Upload Midi', 'Audios', 'Pictures', and 'Mapper'. Below this are links to 'dataset_audio.zip', 'dataset_gambar.zip', and 'test.json'. On the right, the search results are displayed with execution time '16.9055 ms'. There are seven results shown, each with a similarity score (71.67%, 71.51%, 71.51%, 71.51%, 59.31%, 58.37%, 56.68%), album ('Album Kretakan' or 'Unknown'), artist ('Artist Kretakan' or 'Unknown'), and a preview player.</p> <p>Hasil Similaritas : 71,60 % Waktu eksekusi : 16.9055 ms</p>
Test Case 5	<p>Query audio merupakan salah lagu pada dataset yang temponya diperlambat 2 kali dari tempo semula</p>

	<p>The screenshot shows the Sound The Sheep application interface. On the left, there's a file upload section with 'test1-tempo-60.mid' selected, and buttons for 'Upload Midi', 'Audios', 'Pictures', and 'Mapper'. Below this are links to 'Audios: dataset_audio.zip', 'Pictures: dataset_gambar.zip', and 'Mapper: test.json'. The main area displays search results for 'test1-tempo-60.mid' against a dataset of songs by 'TULUS'. The results show similarity scores: 71.60%, 71.53%, 71.53%, 64.41%, 59.55%, 58.33%, and 56.69%. Each result includes a preview player, album cover, and artist information ('Unknown'). At the bottom, there are 'Previous', 'Page 1 of 1', and 'Next' buttons.</p>
	<p>Hasil Similaritas : 71,60 % Waktu eksekusi : 16.9055 ms</p>
Test Case 6	<p>Query audio merupakan salah lagu pada dataset yang temponya dipercepatan 1,5 kali dari tempo semula.</p> <p>The screenshot shows the Sound The Sheep application interface. On the left, there's a file upload section with 'test1-tempo-180-full.mid' selected, and buttons for 'Upload Midi', 'Audios', 'Pictures', and 'Mapper'. Below this are links to 'Audios: dataset_audio.zip', 'Pictures: dataset_gambar.zip', and 'Mapper: test.json'. The main area displays search results for 'test1-tempo-180-full.mid' against a dataset of songs by 'TULUS'. The results show similarity scores: 70.88%, 77.85%, 77.85%, 73.14%, 69.24%, 64.25%, 62.01%, and 58.08%. Each result includes a preview player, album cover, and artist information ('Unknown'). At the bottom, there are 'Previous', 'Page 1 of 3', and 'Next' buttons.</p>
	<p>Hasil Similaritas : 78,88 % Waktu eksekusi : 21,00999 ms</p>

Test case 7	<p>Query audio merupakan potongan bagian awal dan akhir lagu pada dataset yang temponya diperlambat 2 kali dari tempo semula</p> <p>Hasil Similaritas : 65,16 % Waktu eksekusi : 17,0875 ms</p>
Test case 8	<p>Query audio merupakan potongan bagian akhir lagu yang temponya dipercepat 1,25 kali dari tempo semula</p> <p>Hasil Similaritas : 71,60 % Waktu Eksekusi : 11,2402 ms</p>

4.4 Analisis Hasil Eksperimen Music Retrieval Query by Humming

Untuk query dengan variasi temporal (Test Case 2-4), sistem menunjukkan konsistensi yang baik dengan similaritas berkisar antara 68-72%. Sistem dapat menangkap frase musical dalam potongan lagu dengan baik, baik itu bagian awal, kombinasi awal-akhir, maupun bagian akhir. Sistem juga menunjukkan ketahanan terhadap variasi tempo (Test Case 5-6), dimana query dengan tempo yang dimodifikasi masih dapat dikenali dengan hasil percobaan dengan dataset lagu yang ada serta menggunakan window size 30 beat dan slide size 6 beat. Pada Test Case 1 dengan waktu

eksekusi 15,604 ms, sistem mencapai akurasi sempurna (100%) ketika query merupakan salinan identik dari file target, memvalidasi bahwa parameter windowing ini efektif untuk pencocokan dasar. Window size 30 beat terbukti cukup untuk menangkap karakteristik melodi, sementara slide size 6 beat memberikan overlap 24 beat (80%) yang memadai untuk mendeteksi kesamaan antar segmen. Dari hasil tersebut dapat terlihat juga bahwa proses load audio berfungsi dengan baik dalam menangani dataset MIDI, dimana setiap file berhasil diproses melalui tahapan ekstraksi notes, normalisasi tempo, windowing, dan ekstraksi fitur ATB, RTB, dan FTB tanpa mengalami kegagalan.

Untuk query dengan variasi temporal (Test Case 2-4), sistem menunjukkan konsistensi yang baik dengan similaritas berkisar antara 68-72%. Sistem mampu menangkap frase musical dalam potongan lagu dengan baik, baik itu bagian awal, kombinasi awal-akhir, maupun bagian akhir. Sistem juga menunjukkan ketahanan terhadap variasi tempo (Test Case 5-6), dimana query dengan tempo yang dimodifikasi masih dapat dikenali dengan similaritas 71-79%. Ini menunjukkan bahwa kombinasi window size yang lebih kecil dengan pembobotan fitur ATB (0.3), RTB (0.4), dan FTB (0.3) efektif dalam menangkap pola melodi meski ada perubahan kecepatan.

Namun, sistem menunjukkan limitasi saat menghadapi kombinasi multiple variasi. Pada Test Case 7, dimana query merupakan kombinasi potongan lagu dan perlambatan tempo, sistem menghasilkan similaritas 65.16% dan lagu target berada di peringkat 5. Hal ini mengindikasikan bahwa window size 30 beat mungkin terlalu pendek untuk menangkap konteks global saat menghadapi variasi kompleks, dan overlap 24 beat bisa menghasilkan fragmentasi berlebih saat query memiliki potongan terpisah. Meski demikian, secara keseluruhan sistem menunjukkan performa yang solid dengan similaritas konsisten di atas 65% untuk sebagian besar kasus pengujian.

BAB V

PENUTUP

5.1 Kesimpulan

Dalam mata kuliah IF2123 Aljabar Linear dan Geometri, telah dipelajari berbagai konsep fundamental seperti vektor, nilai eigen dan vektor eigen, serta metode Singular Value Decomposition (SVD). Konsep-konsep tersebut kemudian diimplementasikan dalam penyelesaian permasalahan Image Retrieval dan Music Information Retrieval dengan menggunakan metode Principal Component Analysis (PCA) dan analisis berbasis vektor.

Melalui program yang telah dikembangkan, dilakukan pencarian informasi audio dengan memanfaatkan gambar sampul album serta pencarian audio berdasarkan fitur-fiturnya. Berdasarkan pelaksanaan tugas besar ini, dapat disimpulkan bahwa materi aljabar vektor memiliki peranan yang signifikan dan relevan dalam berbagai aspek kehidupan sehari-hari, khususnya dalam pengolahan data dan informasi digital.

5.2 Saran

Dalam pelaksanaan tugas besar ini, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan program di masa depan:

1. Sejak awal, sebaiknya dilakukan kesepakatan terkait paradigma pemrograman yang digunakan agar tidak menggunakan dua paradigma pemrograman yang berbeda yaitu prosedural dan berorientasi objek.
2. Untuk meningkatkan kenyamanan pengguna dalam menggunakan program, sebaiknya website dibuat dengan tampilan responsif.

5.3 Refleksi

Melalui tugas besar ini, kami menyadari pentingnya koordinasi dan komunikasi yang efektif dalam pembagian tanggung jawab serta berbagai aspek lainnya. Kerjasama yang solid antar anggota tim menjadi faktor kunci dalam mencapai tujuan bersama, sehingga setiap individu dapat memberikan kontribusi terbaik sesuai dengan keahlian dan peran yang telah ditetapkan.

Kami juga memahami bahwa manajemen waktu yang baik sangatlah penting, terutama di tengah jadwal akademik yang padat. Dengan pengelolaan waktu yang efisien, kami berhasil menghindari penundaan pekerjaan dan memastikan setiap tahap penggerjaan berjalan optimal. Hal ini memungkinkan kami menjalani proses belajar dengan lebih menyenangkan dan produktif, sehingga setiap langkah dalam penggerjaan tugas besar ini dapat dinikmati dengan maksimal.

5.4 Komentar

- Fawwaz: “Keossss apalagi banyak tubes2 lain, jadi manajemen waktu yg baik sangat diperlukan , Alhamdulillah dari tubes ini banyak pembelajaran baik dari sisi teknikal ataupun nonteknikal yang dapat diambil“
- Carlo : “KCW. Keos, Cape, Wow”
- Ferdin : “Sangat memberikan sensasi baru dalam mengoding, bertemu dengan sepuh dan kingg membuatku ingin terus belajar”

LAMPIRAN

1. Referensi

Slide Kuliah IF2123 2024/2025 :

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

“Music Retrieval by Humming”

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

“Audio Query by Example Using Similarity Measures Between Probability Measure Representations”

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

“Pengenalan wajah menggunakan metode principal component analysis (PCA)”

<https://pemrogramanmatlab.com/2023/08/10/pengenalan-wajah-menggunakan-metode-principal-component-analysis-pca/>

“Image Compression Techniques: A Closer Look at Principal Component Analysis”

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

2. Tautan Repository

Berikut tautan repository github kelompok kami untuk Tugas Besar 2 IF2123 Aljabar Linear dan Geometri :

<https://github.com/carllix/Algeo02-23065>

3. Tautan Video

Berikut tautan video kelompok kami untuk Tugas Besar 1 IF2123 Aljabar Linear dan Geometri :

<https://youtu.be/29KblYEYD10?feature=shared>

4. Tautan Data Test

Berikut tautan data test kelompok kami untuk testing Tugas Besar 1 IF2123 Aljabar Linear dan Geometri :

<https://drive.google.com/drive/folders/14o6nZolFuFSfGLh7Y4FMspw9l7-l2XIA?usp=sharing>