

LAPORAN TUGAS BESAR 02

IF 2123 ALJABAR LINEAR DAN GEOMETRI

Kelompok EDP445



Disusun oleh:

Ahmad Wafi Idzharulhaqq	13523131
Arlow Emmanuel Hergara	13523161
Filbert Engyo	13523163

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA (STEI)
INSTITUT TEKNOLOGI BANDUNG
2024

DAFTAR ISI

BAB I	
DESKRIPSI MASALAH	2
1.1. Latar Belakang	2
1.2. Spesifikasi Kebutuhan	3
BAB II	
LANDASAN TEORI	4
2.1. Information Retrieval	4
2.1.1. Album Picture Finder - Principal Component Analysis	4
1. Image Processing and Loading	4
2. Data Centering (Standardization)	5
3. PCA Computation Using Singular Value Decomposition (SVD)	5
4. Similarity Computation	6
5. Retrieval and Output	7
2.1.2. Music Information Retrieval - Query by Humming	7
1. Pemrosesan Audio	8
2. Ekstraksi Fitur	8
2.2. Dasar Pengembangan Website	9
BAB III	
ARSITEKTUR WEBSITE	11
BAB IV	
IMPLEMENTASI DAN UJI COBA	12
BAB V	
PENUTUP	18
5.1 Simpulan	18
5.2 Saran dan Masukan	19
DAFTAR PUSTAKA	19
LAMPIRAN	20

BAB I

DESKRIPSI MASALAH

1.1. Latar Belakang

Dalam kehidupan sehari-hari, manusia mengandalkan indera pendengaran dan penglihatan untuk mengenali, memahami, serta menginterpretasikan dunia di sekitarnya. Suara dan gambar menjadi media utama dalam menyampaikan informasi, memicu emosi, hingga menciptakan karya seni. Namun, seiring dengan perkembangan teknologi, tantangan muncul ketika manusia ingin mengenali atau mencari informasi spesifik dari kumpulan data suara atau gambar secara otomatis dan akurat.

Salah satu permasalahan yang sering dihadapi adalah bagaimana mencocokkan suatu potongan suara, seperti melodi atau rekaman pendek, dengan database besar yang berisi lagu-lagu atau audio lainnya. Dalam konteks ini, diperlukan teknologi yang mampu menganalisis dan mengidentifikasi ciri khas dari audio, seperti nada, ritme, atau pola frekuensi, sehingga aplikasi dapat memberikan hasil yang sesuai dengan input pengguna. Proses ini dikenal dengan istilah *Music Information Retrieval* (MIR). Contoh nyata penerapannya adalah aplikasi Shazam, yang mampu mendeteksi judul lagu hanya dari beberapa detik cuplikan audio.

Selain itu, pengenalan dan pencarian informasi visual juga menjadi tantangan tersendiri. Sering kali, manusia lebih mengingat gambar, seperti sampul album musik, dibandingkan nama lagu atau penyanyinya. Dalam kasus ini, diperlukan sistem yang mampu mengolah data gambar untuk mengenali pola atau ciri khas visual yang relevan dengan database. Teknologi ini dikenal dengan istilah *Image Retrieval*, yang memungkinkan pencarian data gambar berdasarkan input visual tertentu.

Namun, pengembangan kedua jenis sistem ini memerlukan pendekatan yang berbeda tetapi saling melengkapi. Pada MIR, algoritma perlu bekerja dengan fitur audio seperti spektrum frekuensi atau pola humming, sedangkan pada *Image Retrieval*, teknologi harus mampu merepresentasikan gambar dalam bentuk numerik untuk dianalisis. Kedua pendekatan ini dapat dikombinasikan untuk memberikan solusi yang lebih komprehensif, seperti mencocokkan gambar sampul album dengan lagu terkait.

Untuk menyelesaikan masalah ini, diperlukan penerapan teknik komputasi seperti *Principal Component Analysis* (PCA) untuk mengekstraksi fitur utama dari data gambar serta algoritma perbandingan vektor untuk mendeteksi kemiripan antar data. Dengan mengintegrasikan metode ini, diharapkan dapat dibangun sebuah aplikasi yang tidak hanya mampu mengenali lagu melalui potongan audio, tetapi juga mengidentifikasi lagu berdasarkan gambar albumnya, memberikan pengalaman pencarian yang lebih intuitif bagi pengguna.

1.2. Spesifikasi Kebutuhan

1. Membuat 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

BAB II

LANDASAN TEORI

2.1. *Information Retrieval*

Information Retrieval adalah proses pencarian informasi dari suatu data dengan memberikan input tertentu. Dalam tugas besar ini, Anda akan berfokus pada dua jenis *Information Retrieval*, yaitu *Image Retrieval* dan *Music Information Retrieval* (MIR). *Image Retrieval* adalah proses memasukkan sebuah input gambar untuk menemukan gambar yang sesuai dalam database berdasarkan perhitungan tertentu. Sementara itu, *Music Information Retrieval* (MIR) adalah proses pencarian data audio dengan menggunakan input berupa potongan audio untuk menemukan audio yang relevan dalam database. Implementasi *Image Retrieval* dapat memanfaatkan metode *Principal Component Analysis* (PCA), sedangkan *Music Information Retrieval* dengan metode *query by humming*.

2.1.1. *Album Picture Finder - Principal Component Analysis*

Principal Component Analysis (PCA) adalah teknik statistik yang digunakan untuk mereduksi dimensi data dengan tetap mempertahankan sebanyak mungkin informasi yang ada. PCA mengubah data berdimensi tinggi menjadi beberapa dimensi yang lebih kecil, disebut principal components, tanpa kehilangan esensi atau pola utama dalam data tersebut. Hasil data yang didapatkan dari PCA ini akan berupa eigenvector dan proyeksi data.

Langkah-langkah untuk melakukan pencarian gambar menggunakan PCA adalah sebagai berikut:

1. *Image Processing and Loading*

Lakukan pemrosesan seluruh gambar yang ada pada dataset. Ubah gambar menjadi grayscale untuk mengurangi kompleksitas gambar dan membuat fokus menjadi bagian terang dan gelap gambar. Yang berarti setiap gambar direpresentasikan dalam intensitas pixel saja tanpa informasi warna.

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

Selanjutnya, gambar akan diubah besarnya sehingga ukurannya sama untuk seluruh gambar. Ukuran seluruh gambar harus konsisten untuk membuat perhitungan semakin akurat.

Lalu ubah vektor *grayscale* pada gambar menjadi 1D untuk membuat dimensi vektor dapat dilakukan pemrosesan data. Jika gambar memiliki dimensi $M \times N$, maka hasilnya adalah vektor dengan panjang $M \cdot N$:

$$I = [I1, I2, ..., IM \cdot N]$$

2. *Data Centering (Standardization)*

Pada step ini lakukan standarisasi data di sekitar nilai 0. Hitung rata rata dari setiap gambar untuk suatu piksel.

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

di mana:

- x_{ij} : nilai piksel ke-j pada gambar ke-i,
- N: jumlah total gambar dalam dataset.

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

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

3. *PCA Computation Using Singular Value Decomposition (SVD)*

Lakukan perhitungan SVD pada gambar-gambar yang sudah distandarisasi. Buat matriks kovarians dari data yang sudah distandarisasi:

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

di mana:

- X' : matriks data yang sudah distandarisasi.

Lalu lakukan dekomposisi nilai singular untuk mendapatkan komponen utama:

- U: matriks eigenvector (komponen utama),

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

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

Lalu ambil n jumlah component utama teratas dari hasil SVD dan lakukan proyeksikan gambar ke komponen utama.

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

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

di mana:

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

4. *Similarity Computation*

Lakukan pencarian kesamaan dengan mencari jarak euclidean dari tiap dataset dengan gambar query. Lalu lakukan pengurutan kecocokan dari yang paling tinggi.

Pertama-tama, representasikan gambar query dalam ruang komponen utama dengan proyeksi yang sama:

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

Dimana:

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

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

- $d(\mathbf{q}, \mathbf{z}_i)$ = 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.

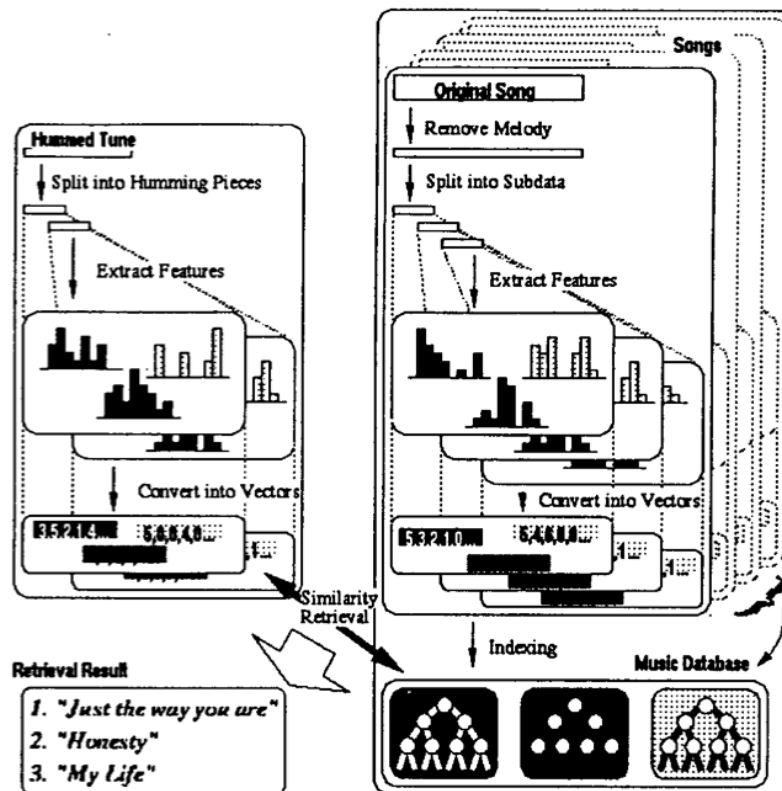
Lalu, Urutkan hasil berdasarkan jarak terkecil.

5. *Retrieval and Output*

Kumpulkan gambar-gambar yang mirip dengan query masukan dengan cara melakukan limitasi jumlah atau dengan memberikan batas jarak euclidean.

Hasil dari pencarian gambar dapat digabungkan dengan hasil pencarian suara ataupun dijalankan sendiri.

2.1.2. *Music Information Retrieval - Query by Humming*



Pada *query by humming* akan dilakukan beberapa langkah utama. Pertama, dilakukan pemrosesan audio, di mana suara yang diinput direkam atau diterima dan dipersiapkan untuk tahap berikutnya. Selanjutnya, data audio tersebut melalui tahap ekstraksi fitur vektor, yaitu proses konversi data audio menjadi representasi numerik berupa vektor fitur yang dapat dianalisis. Terakhir, pada tahap pencarian similaritas vektor, vektor fitur yang dihasilkan dibandingkan dengan dataset untuk menemukan hasil yang paling sesuai atau diatas nilai kemiripan/similaritas yang telah ditentukan.

1. Pemrosesan Audio

Pemrosesan audio dalam sistem query by humming menggunakan file MIDI dengan fokus pada track melodi utama, umumnya di Channel 1. Setiap file MIDI diproses menggunakan metode windowing yang membagi melodi menjadi segmen 20-40 beat dengan sliding window 4-8 beat. Teknik ini memungkinkan pencocokan fleksibel dari berbagai bagian lagu yang mungkin diingat pengguna.

Proses windowing disertai normalisasi tempo dan pitch untuk mengurangi variasi humming. Setiap note event dikonversi menjadi representasi numerik yang mempertimbangkan durasi dan urutan nada, memungkinkan sistem membandingkan potongan melodi dengan database.

Berikut adalah formula untuk melakukan normalisasi tempo yang dibutuhkan :

$$NP(note) = \frac{(note - \mu)}{\sigma}$$

Dimana μ adalah rata rata dari pitch dan σ adalah standar deviasi dari pitch.

2. Ekstraksi Fitur

2.1. Distribusi tone

Distribusi tone diukur berdasarkan tiga viewpoints.

Fitur Absolute Tone Based (ATB) menghitung frekuensi kemunculan setiap nada berdasarkan skala MIDI (0-127). Histogram yang dihasilkan memberikan gambaran distribusi absolut nada dalam data. Hal ini penting untuk menangkap karakteristik statis melodi dalam sinyal audio. Langkah pertama adalah membuat histogram dengan 128 bin, sesuai dengan rentang nada MIDI dari 0 hingga 127. Kemudian, hitung frekuensi kemunculan masing-masing nada MIDI dalam data. Setelah itu, normalisasi histogram untuk mendapatkan distribusi yang terstandarisasi.

Fitur Relative Tone Based (RTB) menganalisis perubahan antara nada yang berurutan, menghasilkan histogram dengan nilai dari -127 hingga +127. RTB berguna untuk memahami pola interval melodi, yang lebih relevan dalam mencocokkan humming dengan dataset yang tidak bergantung pada pitch absolut. Dimulai dengan membangun histogram yang memiliki 255 bin dengan rentang nilai dari -127 hingga +127. Selanjutnya, hitung selisih antara nada-nada yang berurutan dalam data. Terakhir, lakukan normalisasi pada histogram yang telah dibuat.

Fitur First Tone Based (FTB) fokus pada perbedaan antara setiap nada dengan nada pertama, menciptakan histogram yang mencerminkan hubungan relatif terhadap titik referensi awal. Pendekatan ini membantu menangkap struktur relatif nada yang lebih stabil terhadap variasi pitch pengguna. Histogram dibuat

dengan 255 bin, juga mencakup rentang nilai dari -127 hingga +127. Kemudian, hitung selisih antara setiap nada dalam data dengan nada pertama. Histogram yang dihasilkan kemudian dinormalisasi untuk menghasilkan distribusi yang seimbang.

2.2. Normalisasi

Normalisasi memastikan bahwa semua nilai dalam histogram berada dalam skala probabilitas. Berikut adalah formula umum dari normalisasi yang digunakan:

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

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

3. Penghitungan Similaritas

Ubah setiap histogram menjadi sebuah vektor dan lakukan perhitungan kemiripannya menggunakan cosine similarity. Pada jurnal terkait, metode yang digunakan adalah euclidean distance, tetapi pada tugas kali ini metode perhitungan similaritas yang akan digunakan adalah cosine similarity. *Cosine Similarity* adalah ukuran untuk menentukan seberapa mirip dua vektor dalam ruang berdimensi tinggi, dengan menghitung sudut cosinus di antara keduanya. Semakin kecil sudutnya (semakin dekat ke 1 hasilnya), semakin mirip kedua vektor tersebut. Sehingga cosine similarity bisa dijadikan salah satu metode lain dalam perhitungan similaritas. Silahkan lakukan eksplorasi eksperimen dengan pembobotan berbeda untuk setiap fitur dan tentukan bobot terbaiknya.

Berikut adalah formula dari *cosine similarity* :

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

2.2. Dasar Pengembangan Website

1. Perencanaan (Planning)

Pada tahap ini, dilakukan perencanaan awal untuk menentukan tujuan utama website, target audiens, dan fungsi yang diinginkan. Perencanaan melibatkan pembuatan strategi,

wireframe, dan roadmap pengembangan agar setiap langkah memiliki arah yang jelas. Tahap ini juga mencakup diskusi dengan tim atau pemangku kepentingan untuk menyelaraskan visi proyek.

2. Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan untuk mengidentifikasi semua kebutuhan teknis dan fungsional website. Kebutuhan ini mencakup fitur utama, teknologi yang diperlukan, kapasitas server, serta kebutuhan pengguna akhir. Dengan analisis yang baik, pengembang dapat merancang sistem yang sesuai dengan kebutuhan klien dan audiens.

3. Desain (*Design*)

Proses desain mencakup pembuatan antarmuka pengguna (UI) dan pengalaman pengguna (UX) untuk memastikan website menarik dan mudah digunakan. Pada tahap ini, biasanya dibuat prototipe atau mockup yang menampilkan struktur tata letak, skema warna, font, dan elemen visual lainnya. Desain yang baik akan memengaruhi bagaimana pengguna berinteraksi dengan website secara keseluruhan.

4. Pengembangan dan Implementasi

Tahap ini adalah proses pengkodean website berdasarkan desain yang telah disepakati. Pengembangan melibatkan pembuatan sisi *front-end* untuk tampilan antarmuka pengguna dan *back-end* untuk logika bisnis serta pengelolaan data. Setelah website selesai dikembangkan, implementasi dilakukan dengan mengunggahnya ke server dan mengatur domain agar dapat diakses publik.

5. Tahap Pemeliharaan (*Maintenance*)

Setelah website diluncurkan, pemeliharaan dilakukan secara berkala untuk memastikan kinerja tetap optimal. Pemeliharaan mencakup pembaruan sistem, perbaikan bug, peningkatan fitur, serta pemantauan keamanan. Dengan pemeliharaan yang baik, website dapat terus memenuhi kebutuhan pengguna dan berkembang sesuai teknologi terbaru.

BAB III

ARSITEKTUR WEBSITE

Website yang kami rancang ini berfokus pada information retrieval khususnya untuk aspek gambar yang dalam hal ini thumbnail video klip dan audio. Untuk bagian *Front-End*, website ini menggunakan TypeScript sebagai bahasanya dengan Tech Stack VITE, React.js, Tailwind CSS untuk mengelola tampilan website, Framer Motion untuk animasi-animasi, dan Lottie Player untuk menunjukkan animasinya.

Untuk bagian *Back-End*, proses retrieval menggunakan bahasa Python dengan perincian berupa untuk *image retrieval* mengandalkan library numpy dalam mempermudah beberapa penghitungan matematis dan juga pillow untuk memproses gambar. Sedangkan untuk music information retrieval menggunakan *library* mido yaitu untuk mengekstrak file berformat midi sehingga mendapatkan komponen yang diolah selanjutnya, yang juga diolah dengan memanfaatkan numpy untuk penghitungan.

Secara garis besar, desain antarmuka yang dibuat mengambil referensi dari beberapa aplikasi streaming musik yang sudah ada saat seperti spotify, dll namun ditambahkan beberapa komponen untuk menyokong fitur sesuai spesifikasi yaitu image dan music information retrieval yang mirip seperti shahzam. Berdasarkan seluruh kemampuan dan keterbatasan yang kami miliki, kami mencoba untuk menghadirkan suatu website yang fungsional dengan tampilan yang minimalis namun elegan sehingga mempermudah pengguna untuk menggunakannya selagi tetap tidak merasa terganggu akan desain yang terlalu berlebihan.

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1. Image Processing


Untuk Image Retrieval bisa menghasilkan hasil yang sesuai maka dibuatlah 2 tahapan yang dibentuk dalam 2 kelas utama yaitu ImagePreprocessor dan ImageProcessor. Kelas ImagePreprocessor berfungsi untuk proses awal pengolahan gambar seperti perubahan gambar menjadi grayscale, lalu resize gambar dengan ukuran yang sesuai, lalu konversi bentuk grayscale menjadi vektor 1 dimensi yang panjangnya sesuai dengan ukuran dimensi gambar yang dikalikan, dilanjutkan dengan standardisasi melalui pengurangan piksel dengan rata-rata yang dihitung dengan persamaan. Selanjutnya untuk kelas ImageProcessor akan mengolah hasil dari ImagePreprocessor yang berawal dari tahap PCA yang menggunakan operasi dekomposisi matriks yaitu dengan metode *Singular Value Decomposition* (SVD) hingga mendapat bentuk matriks eigenvector dengan dimensi tertentu, kemudian untuk similarity computation dicarilah jarak euclidean antara gambar yang di query dengan seluruh gambar pada dataset dengan melakukan selisih dengan rata-rata sudah didapat sebelumnya, setelah itu jarak euclidean diurutkan dari yang terkecil sebagai data dalam dataset yang paling mirip dengan data query.

4.2. Audio Processing


Implementasi pemrosesan audio melibatkan dua kelas utama, yaitu AudioPreprocessor dan AudioProcessor. Kelas AudioPreprocessor berfungsi untuk memproses dan menyiapkan *raw file* menjadi format yang dapat diolah dalam fungsi-fungsi processor. Sedangkan, kelas AudioProcessor berisi program utama dalam pengolahan data hingga mencapai bentuk final. Beberapa langkah yang dilakukan diantaranya adalah normalisasi pitch, window sliding, perhitungan histogram dari hasil ekstraksi fitur berupa Absolute Tone Based (ATB), Relative Tone Based (RTB), dan First Tone Based (FTB). Setiap data histogram dari *humming query* dan *dataset audio* kemudian dibandingkan menggunakan *cosine similarity*. Nilai maksimal dari rata-rata hasil perbandingan ketiga fitur kemudian diambil untuk menjadi nilai *similarity* lagu tersebut relatif terhadap *query* pengguna. Sistem ini menggunakan NumPy untuk perhitungan numerik, os untuk membaca dan mencari folder/file, json untuk membaca file mapper, dan mido untuk melakukan ekstraksi data dari file dengan format midi.

1. Tampilan Pemrosesan Oleh Audio


EDP445
Searching took 0.322 seconds




There's Nothing Holding Me Back
Play
Similarity: 0.9962421700356027




Symphony
Play
Similarity: 0.9880205478661437



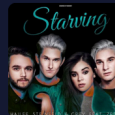
Lost Boy
Play
Similarity: 0.9559559500142548




Paris
Play
Similarity: 0.9455046505682443




Look What You Made Me Do
Play
Similarity: 0.9453833341711287




Starving
Play
Similarity: 0.9420895423785985




Something Just Like This
Play
Similarity: 0.9409740653185809



Dance Monkey
Play
Similarity: 0.6613220261795074



Se7orita
Play
Similarity: 0.6604395484698959




Blinding Lights
Play
Similarity: 0.6378010772911255


Prev Play Next

0:00 0:00


EDP445
Searching took 0.294 seconds



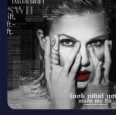
Blinding Lights
Play
Similarity: 0.9786486424658049



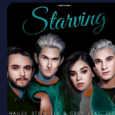
Dance Monkey
Play
Similarity: 0.9513337846858092




Something Just Like This
Play
Similarity: 0.8441237722021406



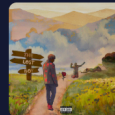
Look What You Made Me Do
Play
Similarity: 0.8354611918623652



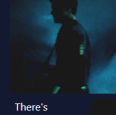
Starving
Play
Similarity: 0.6568420813469243



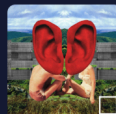
Se7orita
Play
Similarity: 0.6563262465879268




Lost Boy
Play
Similarity: 0.6532495214168055



There's Nothing Holding Me Back
Play
Similarity: 0.633226867126421



Symphony
Play
Similarity: 0.6167485164576736



Paris
Play
Similarity: 0.6164256600786528

Prev Play Next

0:00 0:00

2. Tampilan Hasil Pemrosesan berdasarkan Album

EDP445

Searching took 0.037 seconds

Look What You Made Me Do

Play

Similarity: 7.041140594545452e-13

Something Just Like This

Play

Similarity: 2043.562331740054

Paris

Play

Similarity: 2298.690601575253

There's Nothing Holding Me Back

Play

Similarity: 2758.515309629046

Lost Boy

Play

Similarity: 2783.628551283969

Starving

Play

Similarity: 2954.259142662283

Set7orita

Play

Similarity: 3019.8832166834704

Symphony

Play

Similarity: 3159.7978393840144

Dance Monkey

Play

Similarity: 3241.1920989804776

Blinding Lights

Play

Similarity: 3291.2132316502048

Prev

Next

0:00

Play

0:00

3. Tampilan Halaman Awal

EDP445

Search

Lost Boy

Play

There's Nothing Holding Me Back

Play

Set7orita

Play

Look What You Made Me Do

Play

Paris

Play

Something Just Like This

Play

Blinding Lights

Play

Dance Monkey

Play

Starving

Play

Symphony

Play

Prev

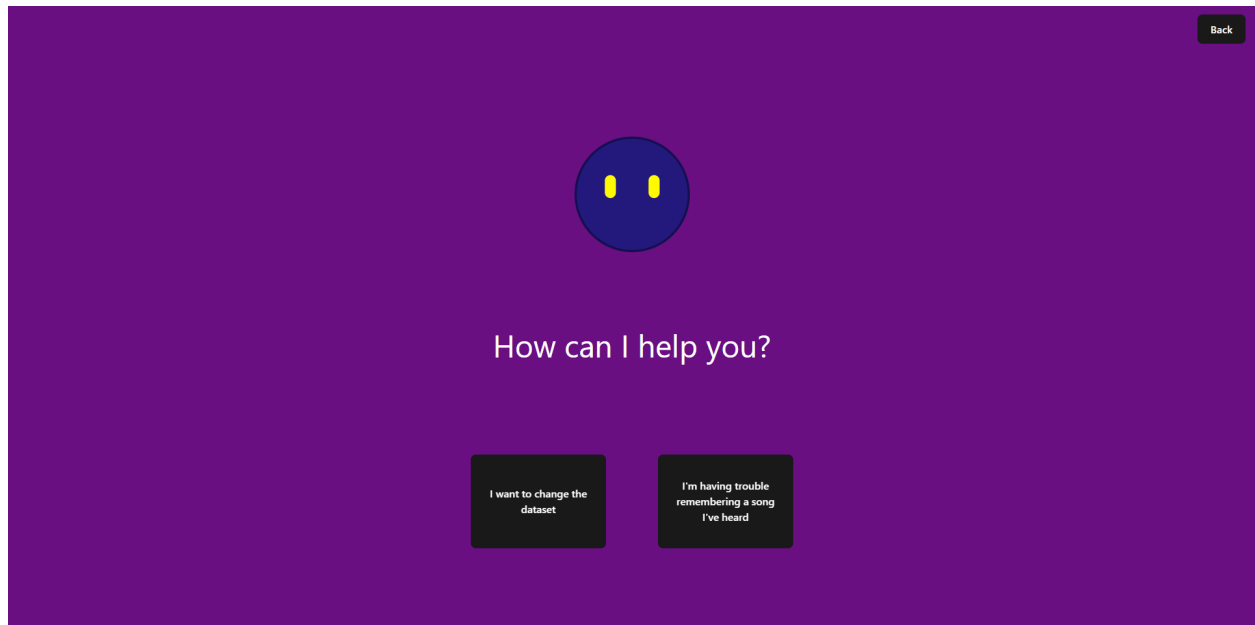
Next

0:00

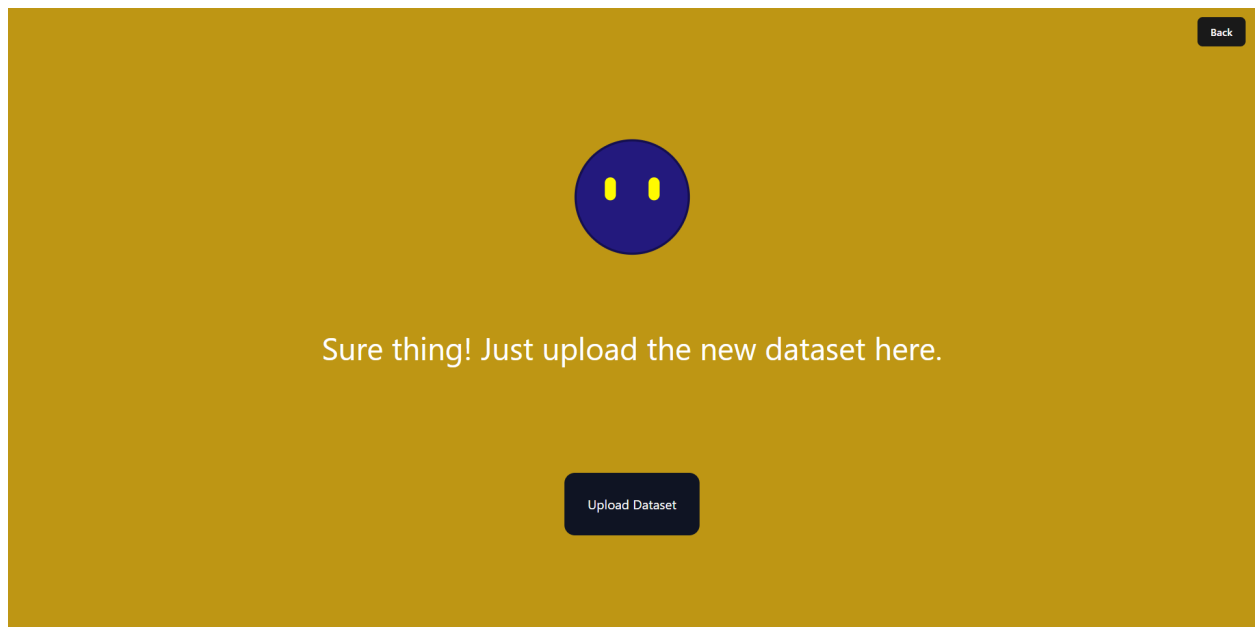
Play

0:00

4. Tampilan Pilihan untuk Mencari Lagu atau Mengunggah Dataset Baru




5. Tampilan untuk Mengunggah Dataset



6. Tampilan Pilihan untuk Memilih Bentuk *Query* yang ingin Diunggah

[Back](#)




Let me help with that. Tell me what you remember about the song you're thinking of.

I remember its album cover

I remember a part of it

7. Tampilan untuk Mengunggah Bentuk *Query* Gambar

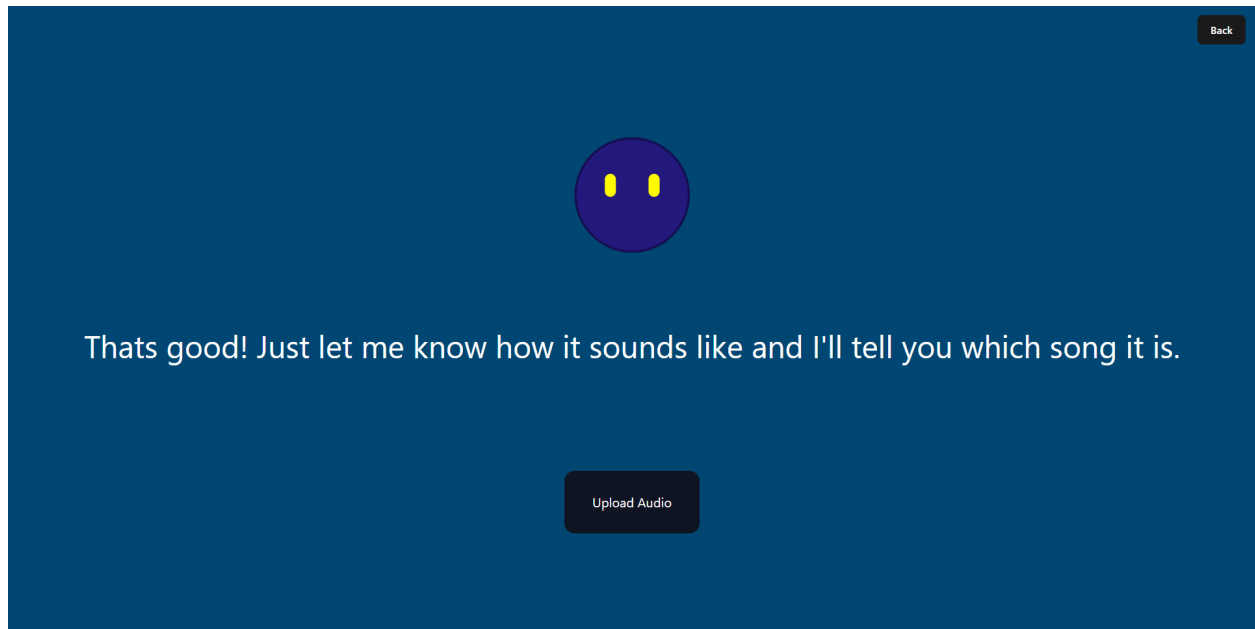
[Back](#)



I can work with that! Just give me a picture of it and I'll see which song it matches with.

Upload Image

8. Tampilan untuk Menungguah Bentuk Query Suara



BAB V

PENUTUP

5.1 Simpulan

Pada tugas besar ini, telah berhasil dibuat aplikasi berbasis web yang memiliki fitur utama *Image Retrieval* dan *Music Information Retrieval* (MIR) dengan memanfaatkan ilmu dan teknologi komputasi yang memproses data numerik secara rinci dan matematis sesuai teori yang diajarkan. Pemanfaatan ilmu *Principal Component Analysis* (PCA) dengan *Singular Value Decomposition* (SVD) dalam mencari similaritas antar gambar dan *Query by Humming* untuk identifikasi audio merupakan bentuk nyata pemanfaatan konsep-konsep aljabar linear yang telah diajarkan seperti nilai eigen, dekomposisi matriks, *cosine similarity*, dan vektor di ruang euclidean. Semuanya disajikan dalam suatu website dengan desain yang minimalis dan elegan selagi tetap memenuhi spesifikasi yang dibutuhkan dalam pengolahan dataset beserta query yang dimasukkan pengguna.

5.2 Saran dan Masukan

Dalam pelaksanaan tugas besar ini, kami menemui beberapa rintangan yang dapat menjadi masukan bagi pengembangan selanjutnya. Kurangnya efisiensi manajemen waktu dan kurangnya eksplorasi membuat penyuguhan fitur pada pengembangan saat ini belum mampu meraih puncaknya. Selain itu, pemilihan bahasa pemrograman terasa mampu dapat lebih dioptimalkan supaya mendapat waktu eksekusi yang lebih baik.

DAFTAR PUSTAKA

Music retrieval by humming-using similarity retrieval over high dimensional feature vector space. (1999). IEEE Conference Publication | IEEE Xplore.

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

Helén, M., & Virtanen, T. (2010). Audio Query by Example Using Similarity Measures between Probability Density Functions of Features. *EURASIP Journal on Audio Speech and Music Processing*, 2010, 1–12. <https://doi.org/10.1155/2010/179303>

Pamungkas, A. (2023, August 23). *Pengenalan wajah menggunakan metode Principal Component Analysis (PCA)*. Pemrograman Matlab.

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

Riswanto, U. (2023, March 2). 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>

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

[Mido - MIDI Objects for Python — Mido 1.3.4.dev6+ga0158ff documentation](#)

LAMPIRAN

Tautan Link Spesifikasi Tugas Besar:

<https://docs.google.com/document/d/13qCaJJtmTyNFSrAvUT6BRVtRclOYFUHCVGsXvg3DU00/edit?tab=t.0>

Tautan Link Repository Github: <https://github.com/filbertengyo/Algeo02-23131>