

**Laporan Tugas Besar 02**  
**IF2123 ALJABAR LINEAR & GEOMETRI**



Disusun oleh:

*Kelompok 31 - skibiditify*

Muhammad Raihaan Perdana (13523124)

Ardell Aghna Mahendra (13523151)

Muhammad Rizain Firdaus (13523164)

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

**2024**

## Daftar Isi

<b>Daftar Isi</b>	<b>2</b>
<b>BAB I DESKRIPSI MASALAH</b>	<b>3</b>
1.1. Latar Belakang	3
1.2. Tujuan	4
<b>BAB II TEORI SINGKAT</b>	<b>5</b>
2.1. Sistem Temu Balik Suara (MIR)	5
2.2. Metode Ekstraksi Fitur berdasarkan Humming	5
2.3. Image Retrieval dengan PCA	6
<b>BAB III ARSITEKTUR WEBSITE dan ARSITEKTUR PROGRAM INFORMATION RETRIEVAL</b>	<b>8</b>
3.1. Arsitektur Website Secara Keseluruhan	8
3.2. Rincian Struktur Program	11
3.1.1. Struktur Folder dan File Rancangan Program	11
3.2.2. Backend	13
3.2.3. Frontend	15
<b>BAB IV EKSPERIMEN</b>	<b>16</b>
4.1. Pencocokan Audio Berdasarkan Query by Humming	16
4.2. Pagination pada Menu Music	17
4.3. Fitur Pencarian pada Menu Music	18
4.4. Image Retrieval dengan PCA	19
4.5. Tampilan About the Website	21
4.6. Tampilan Home Page	21
<b>BAB V PENUTUP</b>	<b>23</b>
5.1. Kesimpulan	23
5.2. Saran	23
5.3. Komentar	23
5.4. Refleksi	23
<b>LAMPIRAN</b>	<b>24</b>
Link Repository	24
Link Video	24
<b>DAFTAR REFERENSI</b>	<b>25</b>

# **BAB I**

## **DESKRIPSI MASALAH**

### **1.1. Latar Belakang**

Suara selalu menjadi hal yang paling penting dalam kehidupan manusia. Manusia berbicara mengeluarkan suara dan mendengarkan suatu suara untuk diserap 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 pendeteksi 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.



Gambar 1. Shazam sebagai aplikasi audio retrieval system

Di dalam Tugas Besar 2 ini, kami diminta untuk membuat semacam aplikasi Shazam yaitu sebuah aplikasi yang meminta input lagu dan aplikasi tersebut mendeteksi apa nama dari lagu tersebut dan beberapa detail lainnya. Pada tugas besar ini, kami akan menggunakan aljabar vektor untuk mencari perbandingan antar satu audio dengan audio yang lain. kami akan menggunakan konsep yang bernama Music Information Retrieval atau MIR untuk mencari dan mengidentifikasi suara

berdasarkan fitur-fitur yang dimilikinya. Tidak hanya itu, kami juga akan menggunakan konsep Principal Component Analysis (PCA) untuk mencari kumpulan audio melalui deteksi wajah berbagai orang.

## 1.2. Tujuan

1. Mengaplikasikan aljabar vektor untuk membandingkan antara satu audio dengan audio lainnya.
2. Mengaplikasikan konsep Music Information Retrieval (MIR) untuk mengidentifikasi audio dan detail lainnya berdasarkan fitur-fitur dari masing-masing audio.
3. Mengaplikasikan Principal Component Analysis (PCA) untuk menganalisis audio berdasarkan wajah terdeteksi.

## **BAB II**

### **TEORI SINGKAT**

#### **2.1. Sistem Temu Balik Suara (MIR)**

Sistem Temu Balik Suara atau Music Information Retrieval atau MIR adalah konsep untuk menerima input suara atau musik (audio) dan mengembalikan audio yang relevan dari basis data berdasarkan perhitungan, analisis, dan informasi yang diinginkan.

#### **2.2. Metode Ekstraksi Fitur berdasarkan Humming**

Humming adalah aktivitas seseorang yaitu menyanyikan lagu atau melodi dengan menutup bibir dan mengeluarkan suara dari hidung atau dengan mulut tertutup. Humming biasanya dilakukan oleh seseorang sebelum ia bernyanyi. Dalam konteks pemrosesan audio, humming biasanya merujuk pada tindakan seseorang menyanyikan sebuah melodi atau nada tanpa kata-kata dan dari suara tersebut akan diproses untuk mencari sebuah lagu. Langkah pertama yaitu pemrosesan audio yang mana suara yang diinput akan direkam atau diterima dan dipersiapkan untuk tahap berikutnya. Pemrosesan audio menggunakan file MIDI yang kemudian diproses menggunakan metode windowing yang disertai dengan normalisasi tempo dan pitch yang mana membagi melodi menjadi segmen 20-40 beat dengan sliding window 4-8 beat dan mengurangi jumlah variasi melodi.

Selanjutnya, data audio dikonversi menjadi representasi numerik berupa vektor fitur melalui tiga pendekatan utama, yaitu Absolute Tone Based (ATB), Relative Tone Based (RTB), dan First Tone Based (FTB). ATB adalah menghitung frekuensi kemunculan setiap nada berdasarkan skala MIDI dan histogram yang dihasilkan mencerminkan distribusi nada dalam data. RTB menganalisis perubahan antara nada yang berurutan dan menghasilkan histogram yang merepresentasikan pola interval melodi yang lebih relevan dalam mencocokkan humming dengan dataset. FTB memfokuskan pada hubungan setiap nada terhadap nada pertama, menciptakan histogram yang menangkap struktur relatif melodi dengan stabilitas terhadap variasi pitch pengguna. Tahap terakhir adalah pencarian similaritas, di mana histogram yang dinormalisasi diubah menjadi vektor fitur dan dibandingkan dengan dataset menggunakan metode seperti cosine similarity. Cosine similarity menghitung sudut cosinus antara vektor dalam ruang berdimensi tinggi untuk menentukan tingkat kemiripan, yang mana apabila nilainya mendekati 1, maka menunjukkan kecocokan yang tinggi.

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

### 2.3. Image Retrieval dengan PCA

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.

Pencarian gambar menggunakan PCA dimulai dengan pemrosesan awal, di mana semua gambar diubah menjadi grayscale untuk fokus pada intensitas piksel tanpa informasi warna.

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

Gambar juga disesuaikan ukurannya agar konsisten, lalu direpresentasikan sebagai vektor 1D. Jika gambar memiliki dimensi  $M \times N$ , maka hasilnya adalah vektor dengan panjang  $M \cdot N$ . Selanjutnya, data distandarisasi dengan mengurangi rata-rata intensitas piksel untuk setiap dimensi agar terpusat di sekitar nol. Setelah itu, Principal Component Analysis (PCA) diterapkan dengan menghitung matriks kovarians.

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

Selanjutnya lakukan dekomposisi Singular Value Decomposition (SVD) untuk memperoleh komponen utama berupa eigenvector dan eigenvalue.

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

Data kemudian diproyeksikan ke komponen utama untuk mengurangi dimensi. Gambar query juga diproyeksikan ke ruang komponen utama, lalu dihitung jarak Euclidean antara gambar query dan gambar-gambar dalam dataset untuk menilai kesamaan.

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

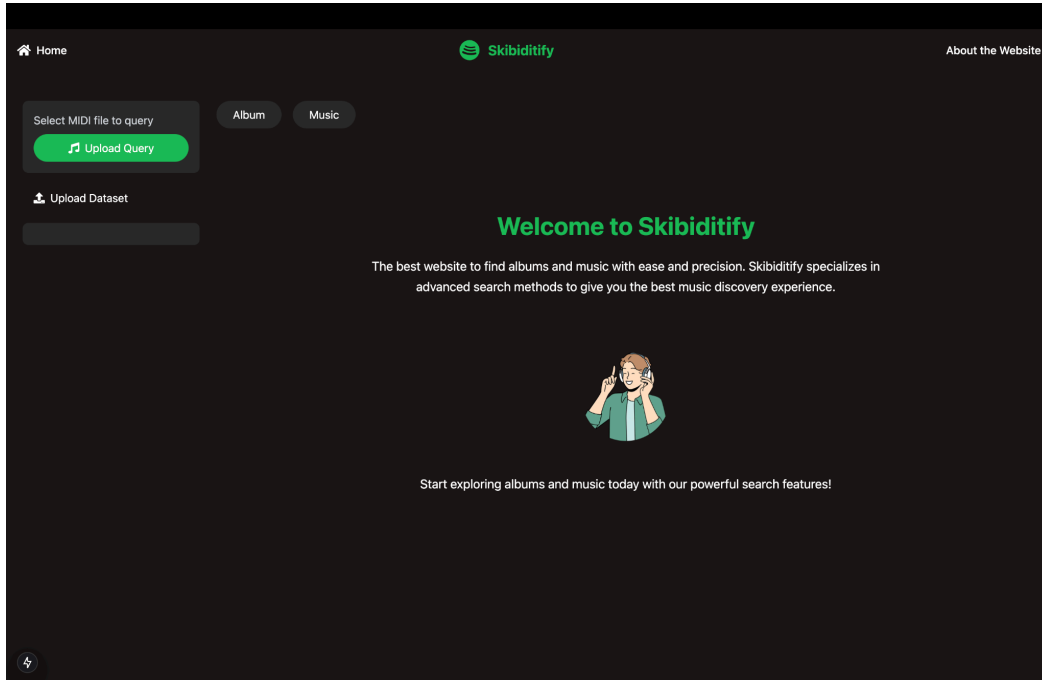
Pada akhirnya, gambar-gambar dengan jarak terkecil dipilih sebagai hasil pencarian. Pendekatan ini memungkinkan pengelolaan dataset gambar secara efisien dan mendukung integrasi dengan pencarian berbasis suara.

# BAB III

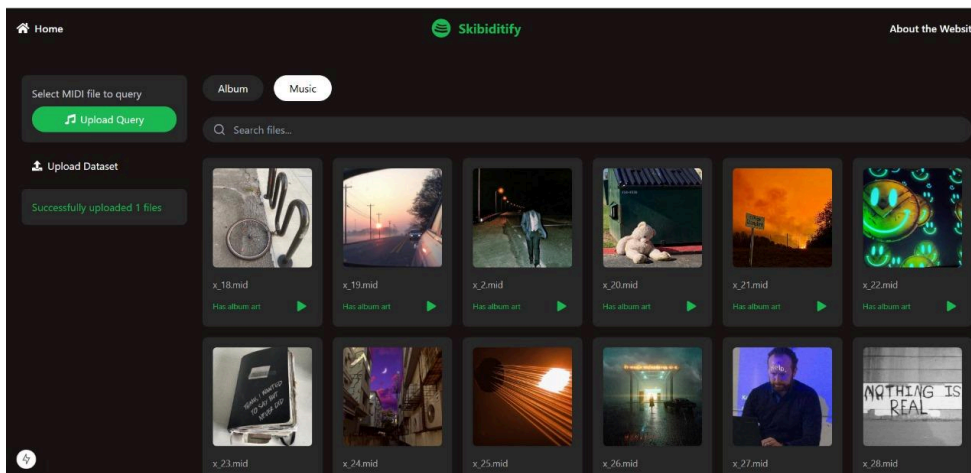
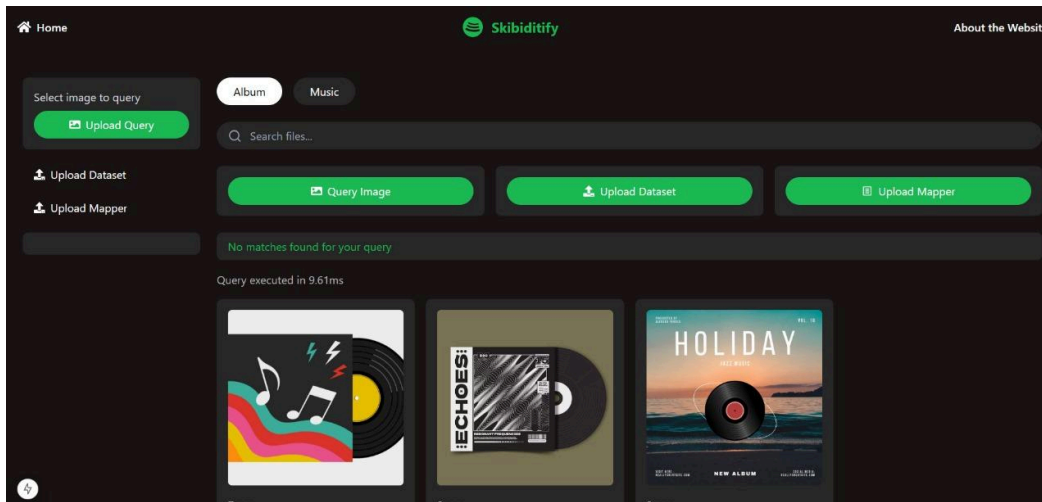
## ARSITEKTUR WEBSITE dan ARSITEKTUR PROGRAM

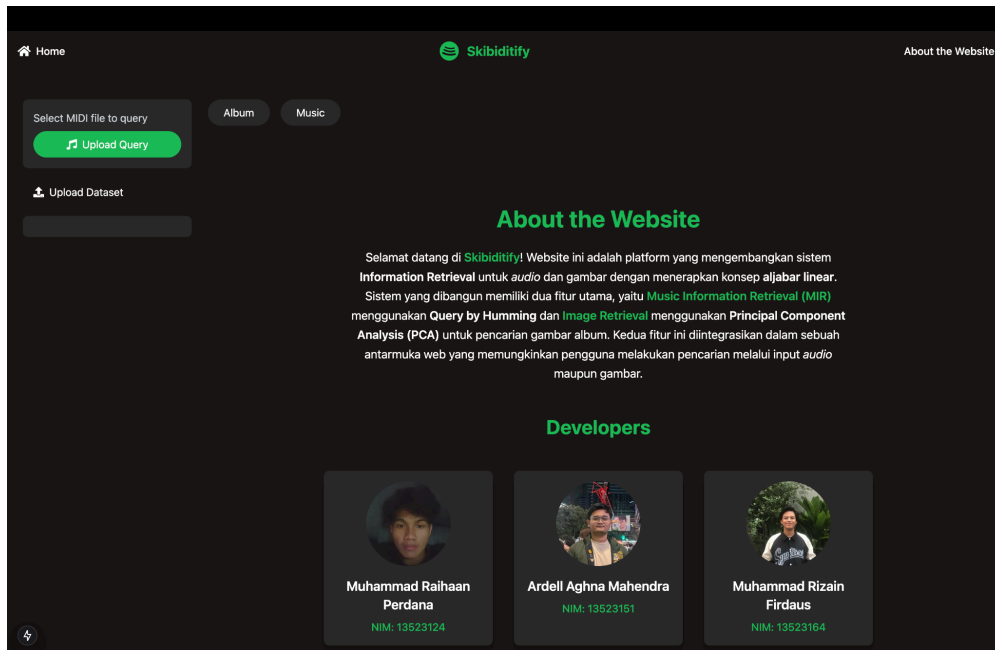
### INFORMATION RETRIEVAL

#### 3.1. Arsitektur Website Secara Keseluruhan









Adapun rincian dari arsitektur website yang dibuat berdasarkan gambar di atas yakni:

- Tombol Home: sebagai tombol navigasi yang mengarahkan ke bagian home page dari website.
- Tombol Album: sebagai tombol navigasi yang mengarahkan ke bagian section dari tampilan dataset gambar yang telah diupload
- Tombol Music: sebagai tombol navigasi yang mengarahkan ke bagian section dari tampilan dataset musik yang telah diupload
- Bagian sidebar dengan tombol upload query, upload dataset, dan upload mapper yang digunakan untuk dapat mengupload query (data yang ingin diuji untuk menghasilkan similaritas), dataset untuk menampilkan dataset di bagian section masing-masing tampilan, dan mapper untuk menghubungkan antar dataset.
- Search bar: untuk dapat melakukan fungsi search pada dataset yang telah diupload.
- Pagination (Previous, Next, dan Jumlah Page): untuk dapat melakukan fungsi navigasi
- Tombol About The Website: sebagai tombol navigasi yang mengarahkan ke bagian mengenai penjelasan website disertai data para developers dari website tersebut.

## 3.2. Rincian Struktur Program

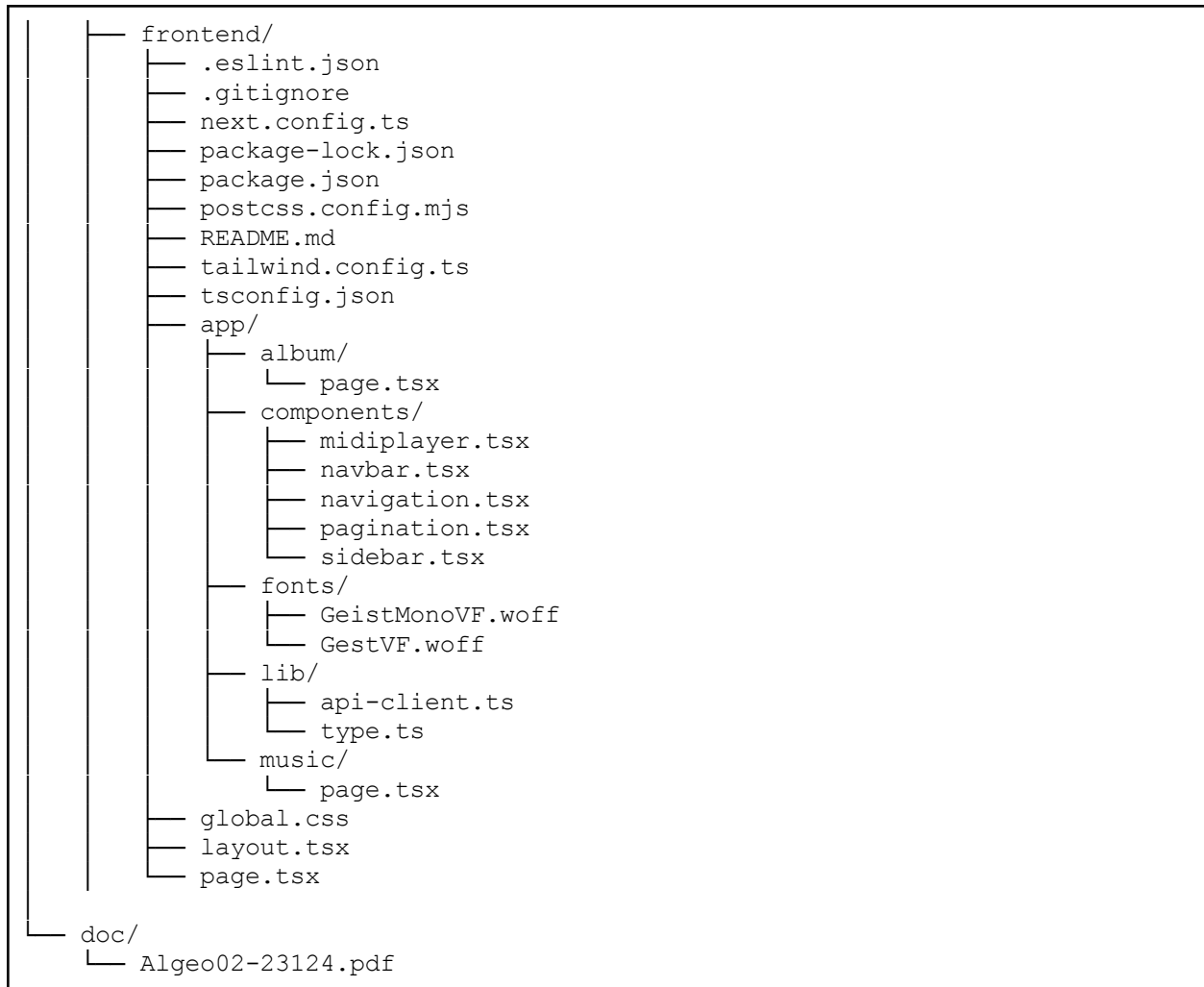
### 3.1.1. Struktur Folder dan File Rancangan Program

Berdasarkan pembangunan website music information retrieval yang dibuat dengan sebuah struktur frontend dan backend, para developer merancang struktur website dengan dibagi dalam sebuah folder struktur backend dan frontend. Untuk struktur dari pembangunan website yang developer rancang yakni dengan menerapkan struktur seperti berikut:

```
Algeo02-23124/
├── README.md
├── bin/
│   └── README.md
├── test/
│   ├── x (1).mid
│   ├── x (2).mid
│   ├── x (3).mid
│   ├── x (4).mid
│   ├── x (5).mid
│   ├── x (6).mid
│   ├── x (7).mid
│   ├── x (8).mid
│   ├── x (9).mid
│   ├── x (10).mid
│   ├── x (11).mid
│   ├── x (12).mid
│   ├── x (13).mid
│   ├── x (14).mid
│   ├── x (15).mid
│   ├── x (16).mid
│   ├── x (17).mid
│   ├── x (18).mid
│   ├── x (19).mid
│   ├── x (20).mid
│   ├── x (21).mid
│   ├── x (22).mid
│   ├── x (23).mid
│   ├── x (24).mid
│   ├── x (25).mid
│   ├── x (26).mid
│   ├── x (27).mid
│   ├── x (28).mid
│   ├── x (29).mid
│   └── x (30).mid
```

```
├── x (31).mid
├── x (32).mid
├── x (33).mid
├── x (34).mid
├── x (35).mid
├── x (36).mid
├── x (37).mid
├── x (38).mid
├── x (39).mid
├── x (40).mid
├── x (41).mid
├── x (42).mid
├── x (43).mid
├── x (44).mid
├── x (45).mid
├── x (46).mid
├── x (47).mid
├── x (48).mid
├── x (49).mid
└── x (50).mid
```

```
├── src/
│   ├── backend/
│   │   ├── app/
│   │   │   ├── routes/
│   │   │   │   ├── __init__.py
│   │   │   │   ├── audiou_routes.py
│   │   │   │   └── image_routes.py
│   │   │   ├── services/
│   │   │   │   ├── __init__.py
│   │   │   │   ├── audio_services.py
│   │   │   │   └── image_services.py
│   │   │   ├── utils/
│   │   │   │   ├── audio/
│   │   │   │   │   ├── dataset_scanner.py
│   │   │   │   │   ├── feature_extraction.py
│   │   │   │   │   ├── file_handler.py
│   │   │   │   │   ├── tempo_normalizer.py
│   │   │   │   │   └── window_processor.py
│   │   │   │   └── image/
│   │   │   │       ├── feature_extraction.py
│   │   │   │       ├── grayscale_resizer.py
│   │   │   │       └── pca_processor.py
│   │   │   ├── config.py
│   │   │   └── __init__.py
│   │   ├── .gitignore
│   │   ├── requirements.txt
│   │   └── run.py
```



### 3.2.2. Backend

Folder backend adalah bagian dari proyek yang mengatur aplikasi backend menggunakan Python. Berikut adalah penjelasan dari setiap folder dan file dalam direktori ini:

1. Routes

Folder ini berisi file-file yang mendefinisikan rute atau endpoint API yang dapat diakses oleh klien. Dua file utama di sini adalah:

- audio\_routes: Mengatur rute-rute terkait dengan audio, seperti pengambilan atau pemrosesan data audio.
- image\_routes: Mengatur rute-rute terkait dengan gambar, termasuk pengolahan gambar atau pengambilan gambar.

## 2. Services

Folder ini berisi file-file yang menangani logika bisnis dan pemrosesan data backend. Layanan ini mengelola dan memproses data untuk menghasilkan respons yang dibutuhkan oleh aplikasi frontend. Ada dua file utama di dalam folder ini:

- `audio_services`: Berfungsi untuk memproses data terkait audio, seperti ekstraksi fitur dari file audio atau pengolahan data terkait musik.
- `image_services`: Berfungsi untuk memproses data gambar, termasuk pengolahan atau analisis gambar.

## 3. Utils

Folder ini berisi utilitas atau fungsi tambahan yang digunakan di berbagai bagian aplikasi backend. Fungsi-fungsi ini mendukung proses pemrosesan data dalam aplikasi secara keseluruhan. Di dalam folder ini terdapat dua subfolder:

- Folder `audio`: Berisi file-file yang membantu dalam pemrosesan data audio. Beberapa file di dalamnya adalah:
  - `dataset_scanner`: Memindai dataset dan mengelola file-file audio yang ada.
  - `feature_extraction`: Mengekstraksi fitur dari file MIDI, seperti fitur tonality atau tempo.
  - `file_handler`: Mengelola dan memproses file audio yang diterima oleh aplikasi.
  - `tempo_normalizer`: Menormalkan tempo dalam data audio.
  - `window_processor`: Memproses data audio dalam bentuk jendela atau segmen untuk analisis lebih lanjut.
- Folder `image`: Berisi file yang berhubungan dengan pemrosesan gambar, seperti:
  - `feature_extraction`: Mengekstraksi fitur-fitur gambar yang dapat digunakan dalam aplikasi.
  - `grayscale_resizer`: Mengubah ukuran gambar dalam format grayscale.
  - `pca_processor`: Memproses gambar menggunakan teknik Principal Component Analysis (PCA).

## 4. config

File ini berisi pengaturan atau konfigurasi untuk aplikasi backend, seperti konfigurasi koneksi ke database, pengaturan API, atau pengaturan lain yang diperlukan oleh aplikasi.

## 5. requirements

Berisi daftar dependensi atau pustaka Python yang diperlukan untuk menjalankan aplikasi. Ini termasuk pustaka seperti `pretty_midi`, `numpy`, dan pustaka lainnya yang digunakan dalam proyek.

## 6. run

File ini digunakan untuk menjalankan aplikasi backend. Biasanya berisi instruksi untuk memulai server atau menjalankan aplikasi, sehingga Anda dapat mengakses API atau fitur lainnya.

### 3.2.3. Frontend

Folder frontend ini berisikan aplikasi frontend yang menggunakan framework Next.js dengan diselingi bahasa typescript dan juga tailwind css sebagai stylingnya. Adapun komponen-komponen yang ada di folder ini yakni:

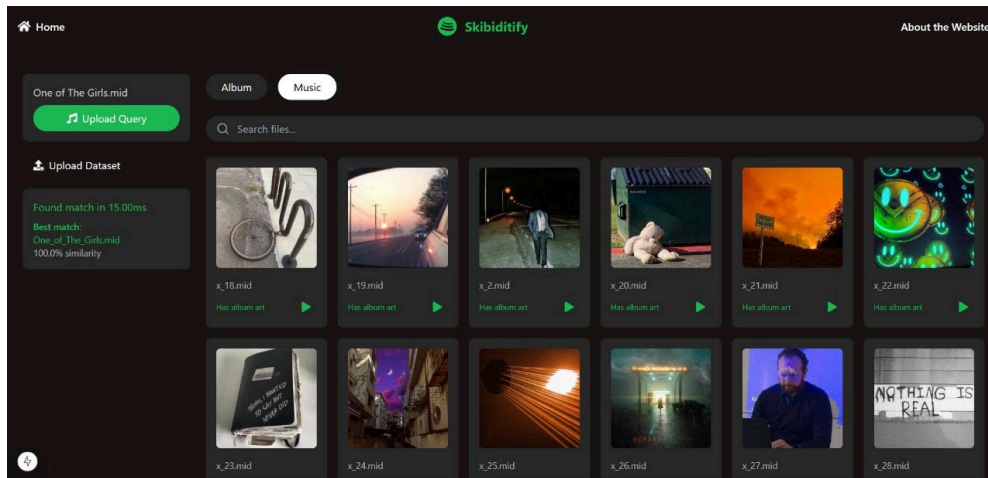
- Folder app: Berisi halaman dan komponen utama aplikasi frontend.
  - album/: Berisi halaman untuk menampilkan album atau koleksi musik.
  - components/: Folder ini berisi komponen UI seperti pemutar musik (midiplayer.tsx), navbar, sidebar, pagination, dll.
  - lib/: Berisi file utilitas dan klien API (api-client.ts).
  - music/: Halaman atau komponen yang berhubungan dengan musik.
- global.css: File CSS global untuk mengatur gaya umum aplikasi.
- layout.tsx dan page.tsx: File untuk mengatur layout umum dan konten utama aplikasi.
- Folder doc : folder ini berisi file dokumentasi untuk proyek, seperti penjelasan umum atau panduan penggunaan. File README .md adalah file dokumentasi utama.

## BAB IV

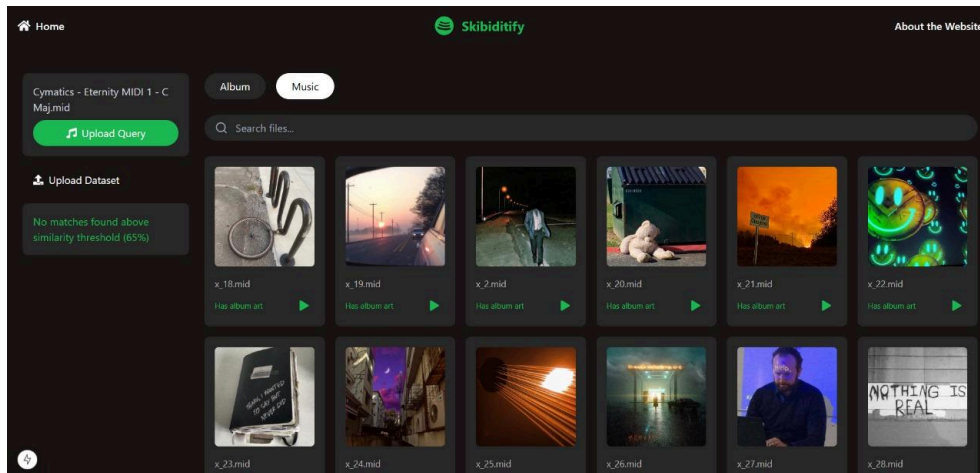
### EKSPERIMEN

#### 4.1. Pencocokan Audio Berdasarkan Query by Humming

- Program akan menampilkan audio yang cocok dengan audio yang diupload pada sidebar.

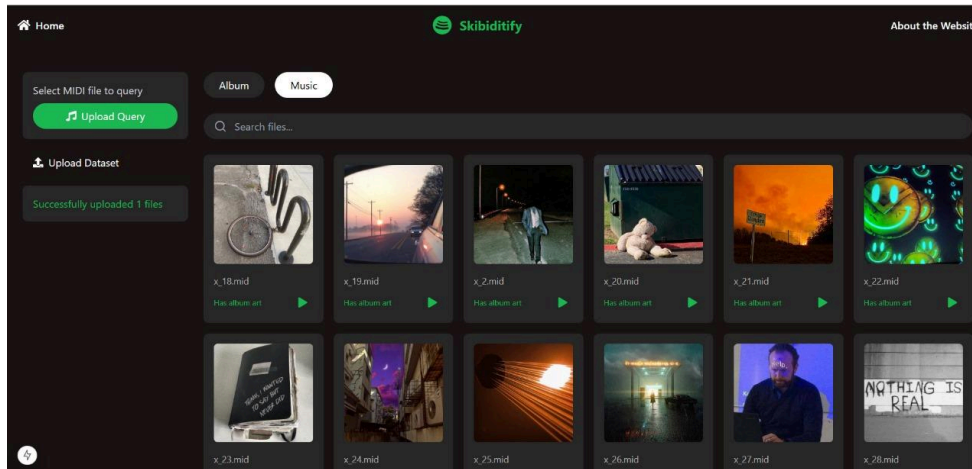


- Program tidak akan menampilkan audio yang cocok di sidebar jika persentase kecocokan audio yang diupload di bawah 65%.



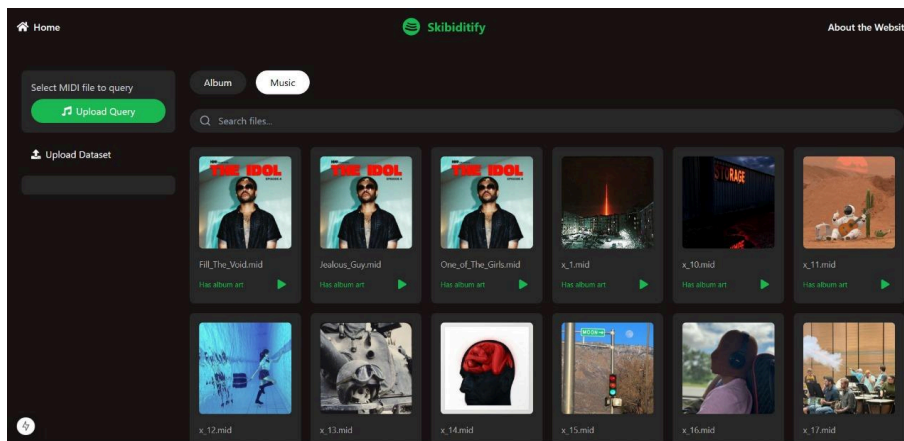
- Program akan menampilkan pesan successfully uploaded ketika berhasil mengupload dataset di sidebar.



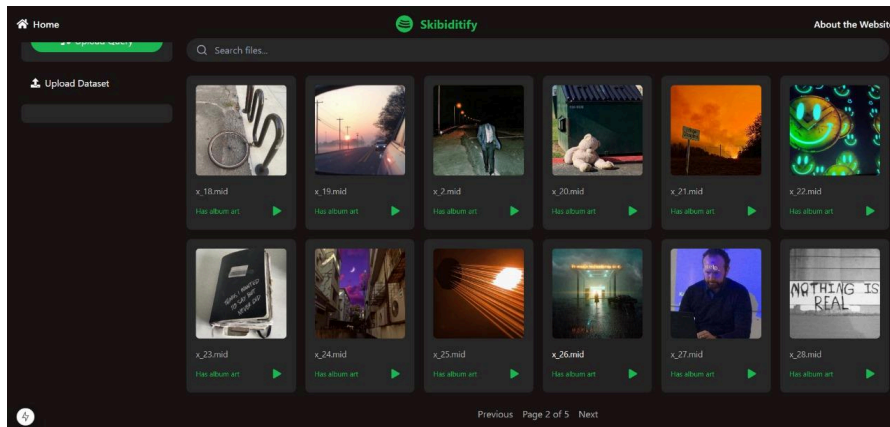


## 4.2. Pagination pada Menu Music

- Tampilan utama menu music page 1.

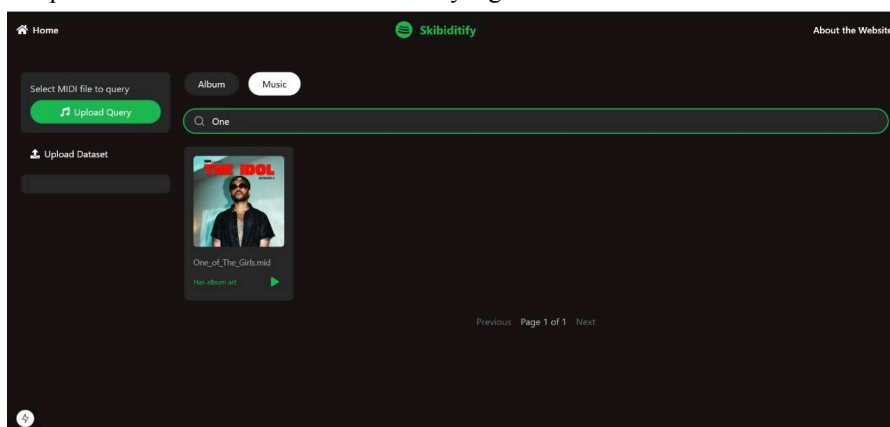


- Tampilan utama menu music page 2.

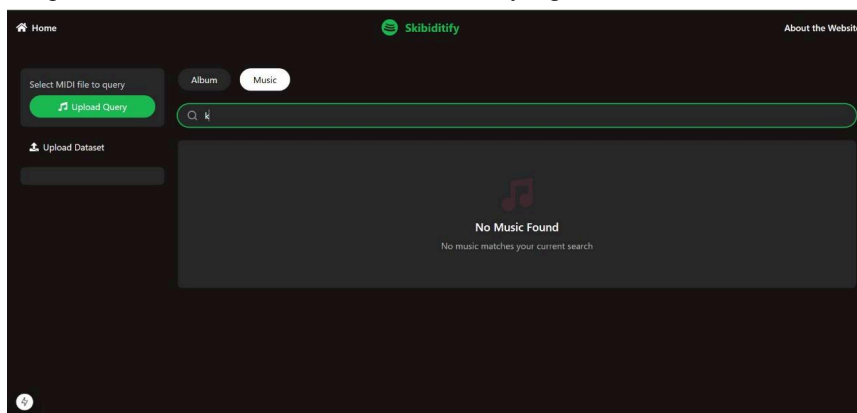


### 4.3. Fitur Pencarian pada Menu Music

- Tampilan website ketika ada audio/music yang dicari.

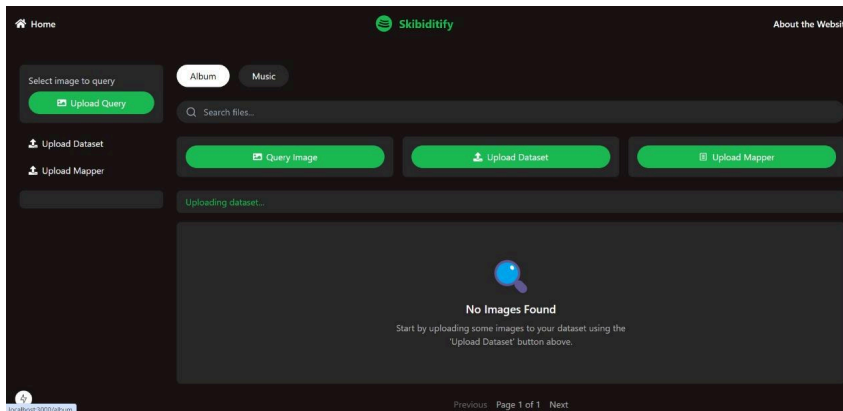


- Tampilan website ketika tidak ada audio/music yang dicari.

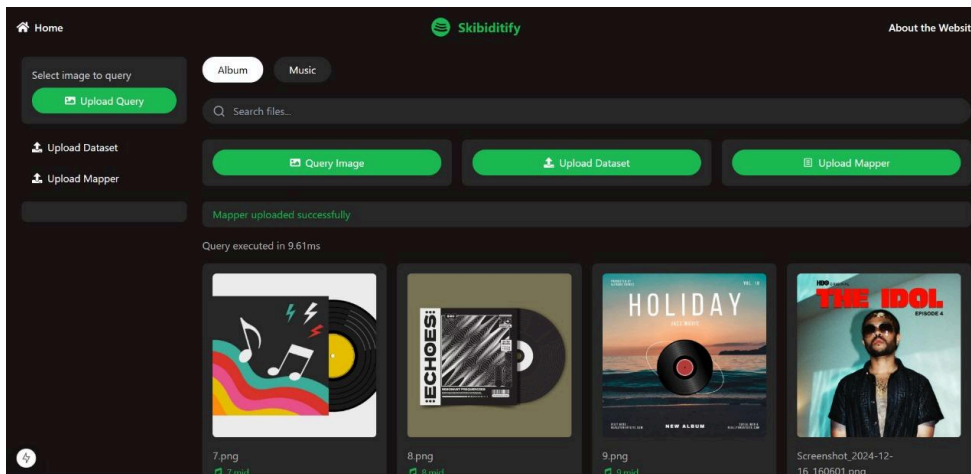


## 4.4. Image Retrieval dengan PCA

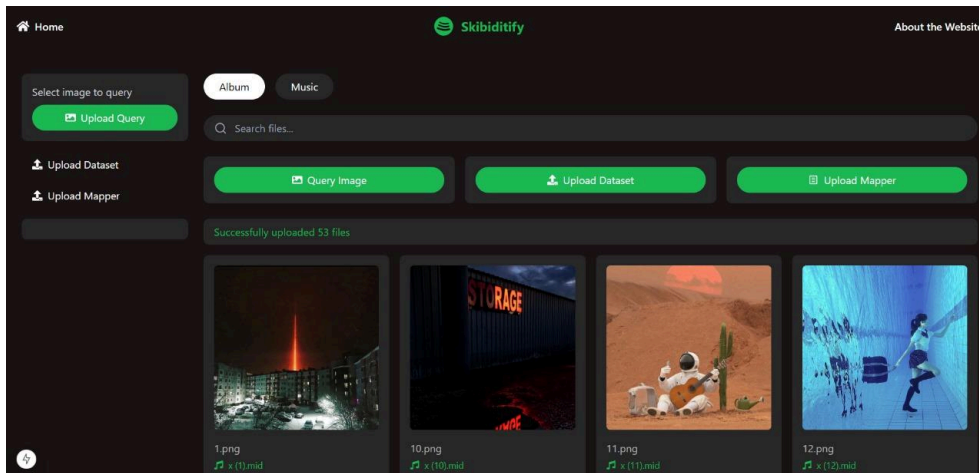
- Tampilan menu album ketika tidak ada file didalamnya.



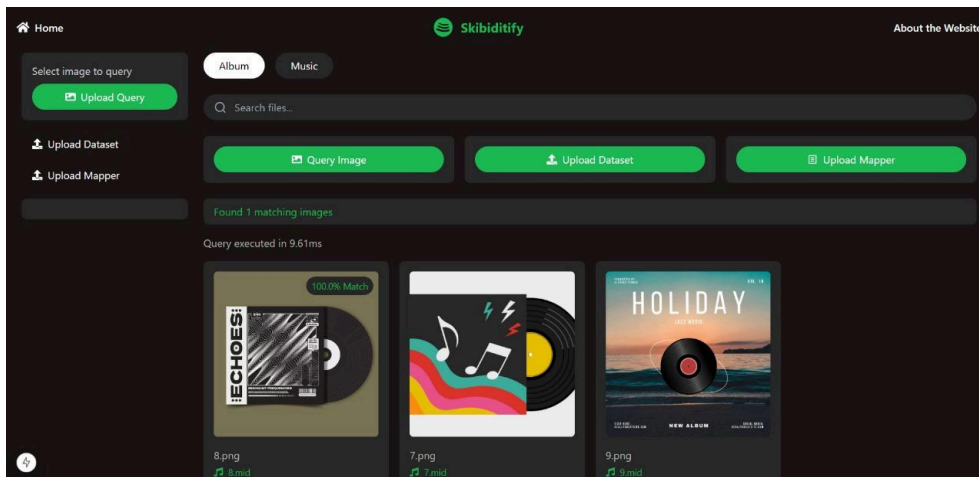
- Tampilan ketika mapper berhasil di upload.



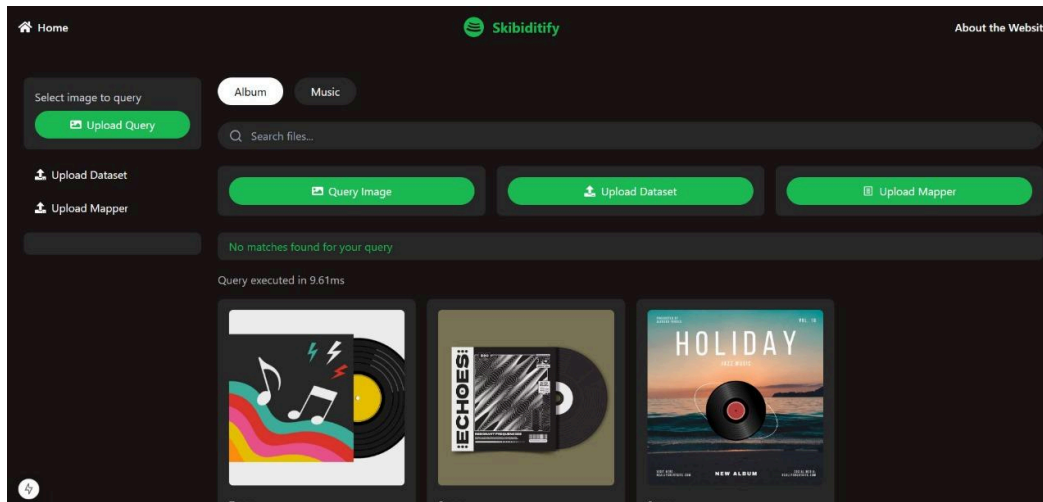
- Tampilan ketika berhasil upload dataset



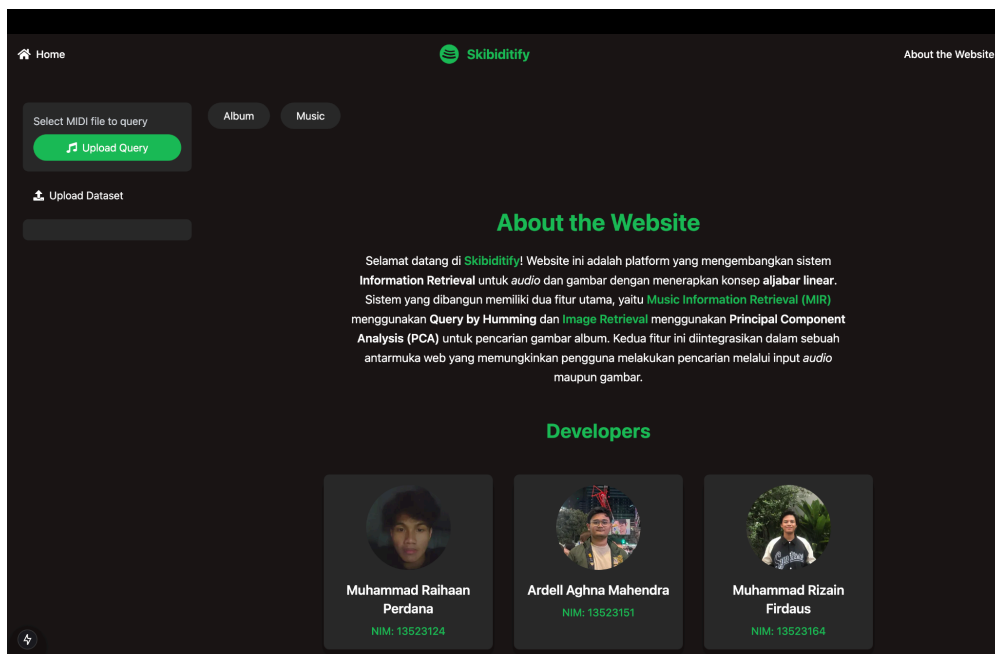
- Tampilan ketika album yang diupload terdapat kecocokan dengan album yang berada di website.



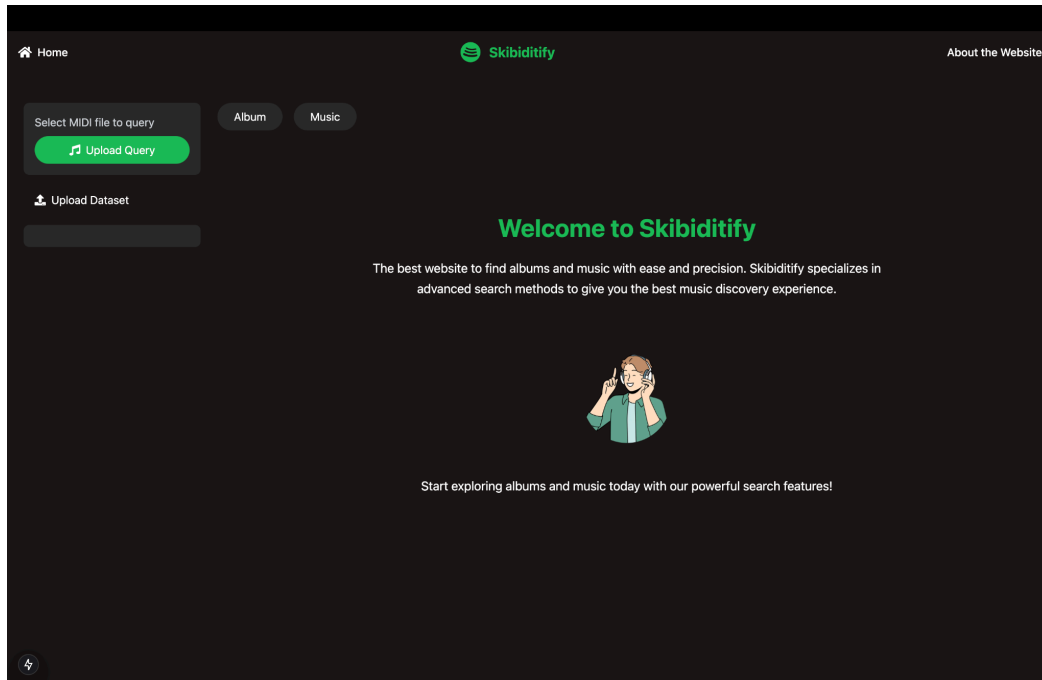
- Tampilan ketika album yang diupload tidak terdapat kecocokan dengan album yang berada di website.



## 4.5. Tampilan About the Website



## 4.6. Tampilan Home Page



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Pada Tugas Besar 2 mata kuliah IF2123 Aljabar Linear dan Geometri, kami telah berhasil mengembangkan sistem Information Retrieval untuk audio dan gambar dengan menerapkan konsep aljabar linear. Sistem yang dibangun memiliki dua fitur utama, yaitu Music Information Retrieval (MIR) menggunakan Query by Humming dan Image Retrieval menggunakan Principal Component Analysis (PCA) untuk pencarian gambar album. Kedua fitur ini diintegrasikan dalam sebuah antarmuka web yang memungkinkan pengguna melakukan pencarian melalui input audio maupun gambar. Melalui pengerjaan tugas ini, wawasan kami terhadap dunia pemrograman semakin bertambah dan juga menambah ilmu serta kemampuan kami dalam menggunakan bahasa pemrograman python, serta mendapatkan pemahaman dalam mengaplikasikan ilmu dari mata kuliah IF2123 Aljabar Linear dan Geometri dalam pemrosesan sinyal digital dan pengolahan citra. Tugas besar ini juga memberikan wawasan kepada kami sebagai pengguna shazam yang mana jadi mengetahui bagaimana aplikasi tersebut bekerja.

#### **5.2. Saran**

Tidak ada.

#### **5.3. Komentar**

Akhirnya tugas besar ini selesai juga di tengah-tengah ke chaos-an perkuliahan karena banyaknya tugas dan selama ngerjain juga lumayan seru karena ternyata membuat website lumayan asik. Ternyata kita juga bisa bikin Shazam × Google lens lite version.

#### **5.4. Refleksi**

Tugas besar ini memberikan banyak pengalaman baru dan berkesan bagi kami. Pengalaman baru dalam pengembangan sistem information retrieval yaitu pengalaman dalam mengintegrasikan pemrosesan audio dan gambar dengan antarmuka web, serta pengalaman bekerja sama secara tim dengan orang baru. Kami juga belajar banyak tentang komunikasi efektif antar anggota kelompok dan manajemen waktu untuk memenuhi deadline tugas besar ini.

## **LAMPIRAN**

### **Link Repository**

Link Repository kelompok kami: <https://github.com/inRiza/Algeo02-23124.git>

### **Link Video**

Link video kami: [https://youtu.be/p\\_oxlAVveUk?si=s-IRVgmfDLEu7pJ6](https://youtu.be/p_oxlAVveUk?si=s-IRVgmfDLEu7pJ6)



## DAFTAR REFERENSI

Calling, V. J. (n.d.). *Speaker Recognition Audio Dataset*. Di akses pada tanggal 16 Desember 2024, dari

<https://www.kaggle.com/datasets/vjcalling/speaker-recognition-audio-dataset/code>

Rahim, M. (n.d.). *Introduction to PCA - Image Compression Example*. Di akses pada tanggal 8 Desember 2024, dari

<https://www.kaggle.com/code/mirzarahim/introduction-to-pca-image-compression-example>

Mueller, M., Kurth, F., & Clausen, M. (2005). Audio matching via chroma-based statistical features. *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 288–295. Di akses pada tanggal 9

Desember 2024, dari <https://www.ismir.net/>