LAPORAN TUGAS BESAR 2 IF2123 ALJABAR LINEAR DAN GEOMETRI

IMAGE RETRIEVAL DAN MUSIC INFORMATION RETRIEVAL MENGGUNAKAN PCA DAN VEKTOR

Dosen Pengampu : Ir. Rila Mandala, M.Eng., Ph.D.

Asisten Pembimbing : Ahmad Naufal Ramadan (13522005)

Disusun oleh:

Kelompok 2 - Broken Home

Nathaniel Jonathan Rusli (13523013)

Maheswara Bayu Kaindra (13523015)

Jessica Allen (13523059)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

KATA PENGANTAR

Dengan penuh rasa syukur, kami menyusun dan mempersembahkan makalah ini yang membahas topik menarik dan relevan dalam dunia matematika dan teknologi, yaitu "*Image Retrieval* dan *Music Information Retrieval* Menggunakan PCA dan Vektor". Makalah ini disusun sebagai bagian dari Tugas Besar 2 IF2123 Aljabar Linear dan Geometri yang bertujuan untuk menggali penerapan teori matematika dalam memecahkan permasalahan di bidang pemrosesan data visual dan audio.

Makalah ini tidak hanya membahas konsep-konsep dasar yang esensial, tetapi juga memperkenalkan metode analitik dan teknik komputasi yang berperan penting dalam pengolahan data yang semakin kompleks. Pemahaman mendalam tentang teknik reduksi dimensi seperti Principal Component Analysis (PCA) dan pendekatan berbasis vektor sangat penting di era modern ini, di mana data menjadi salah satu sumber daya utama dalam berbagai bidang, seperti kecerdasan buatan, sistem rekomendasi, pembelajaran mesin, dan teknologi multimedia.

Melalui makalah ini, kami berharap dapat memberikan penjelasan yang komprehensif dan mendalam tentang topik yang dibahas, sekaligus menunjukkan relevansi praktis dari konsep-konsep ini dalam berbagai aplikasi nyata. Kami berharap makalah ini tidak hanya membantu pembaca memahami teori dasar, tetapi juga menginspirasi pembaca untuk mengaplikasikan teknik-teknik ini dalam memecahkan tantangan dunia nyata yang serupa.

Kami mengucapkan terima kasih kepada dosen pembimbing dan semua pihak yang telah memberikan bimbingan, saran, dan dukungan selama proses penyusunan makalah ini. Semoga makalah ini dapat memberikan manfaat yang berarti dan menjadi langkah awal yang baik bagi pembaca yang ingin mendalami lebih jauh bidang yang menarik ini.

Akhir kata, kami berharap makalah ini dapat menjadi sumber motivasi bagi pembaca untuk terus mengeksplorasi peluang dan tantangan baru di bidang matematika dan teknologi komputasi.

Bandung, 16 Desember 2024

Nathaniel Jonathan Rusli (13523013)
Maheswara Bayu Kaindra (13523015)
Jessica Allen (13523059)

DAFTAR ISI

KATA PENGANTAR.....2
DAFTAR ISI.....3
DAFTAR TABEL.....5
DAFTAR GAMBAR.....6
DAFTAR PEMBAGIAN TUGAS.....8
BAB I DESKRIPSI MASALAH.....9
1.1 Pemrosesan Suara dan Gambar.....9
1.2 Information Retrieval.....10
1.2.1 Album Picture Finder - Principal Component Analysis.....10
1.2.2 Music Information Retrieval - Query by Humming.....15
1.3 Arsitektur Website.....18
BAB II TEORI SINGKAT.....21
2.1 Sistem Temu Balik Suara (MIR).....21
2.2.2 Skema umum Query by Humming.....21
2.2 Metode Ekstraksi Fitur Berdasarkan Humming (QBH).....22
2.2.1 Konsep Dasar Query by Humming (QBH).....22
2.2.2 Jenis-Jenis Metode Ekstraksi Fitur berdasarkan konsep QBH.....22
2.3 Image Retrieval dengan PCA.....24
2.3.1 Konsep Dasar Principal Component Analysis (PCA).....24
2.3.2 Keunggulan dan Keterbatasan PCA dalam Image Retrieval.....24
2.3.3 Penerapan PCA dalam Image Retrieval.....25
BAB III IMPLEMENTASI PUSTAKA DAN PROGRAM.....27
3.1 Arsitektur Website (Frontend).....27
3.1.1 Home Page.....28
3.1.2 FinderPage.....31
3.2 Arsitektur Program Information Retrieval (Backend).....34
3.2.1 Arsitektur Backend Album Picture Finder.....34
3.2.2 Arsitektur Backend Music Information Retrieval.....36
3.2.2.1 MIDI File Processing (process_midi.py).....37
3.2.2.2 Absolute Tone Based (absolute_tone_based.py).....37
3.2.2.3 Relative Tone Based (relative_tone_based.py).....38
3.2.2.4 First Tone Based (first_tone_based.py).....38
3.2.2.5 Calculating Similarities (cosine_similarity.py).....38
3.2.2.6 Database Sorting (find_most_similar.py).....38
3.2.3 Arsitektur Backend API.....38
3.2.3.1 Album Query API.....39

3.2.3.2 Music Query API..... 39
BAB IV EKSPERIMEN..... 41
4.1 Eksperimen…..... 41
BAB V KESIMPULAN, SARAN, DAN REFLEKSI.....50
5.1 Kesimpulan..... 50
5.2 Saran..... 51
5.3 Refleksi..... 51
DAFTAR PUSTAKA..... 52
LAMPIRAN.....53

DAFTAR TABEL

Tabel 1. Pembagian Tugas.....8
Tabel 2. Eksperimen.....41

DAFTAR GAMBAR

Gambar 1.1.1. Shazam sebagai aplikasi audio retrieval system....8
Gambar 1.2.2.1. Tahapan pada MIR dengan metode Query By Humming.... 14
Gambar 1.2.2.2.1. Cosine Similarity Formula.... 17
Gambar 1.3.1. Contoh tampak depan dari website Information Retrieval (Humming)....18
Gambar 1.3.2. Contoh tampak depan dari website Information Retrieval (Album).... 18
Gambar 3.1.1 Arsitektur Website Frontend....27
Gambar 3.1.1 Home Page....27
Gambar 3.1.1.1 Header....27
Gambar 3.1.1.2 About....28
Gambar 3.1.1.2 Finder Button....28
Gambar 3.1.1.3.1 Developer Information (1)....29
Gambar 3.1.1.3.2 Developer Information (2)....29
Gambar 3.1.2 Finder Page....30
Gambar 3.1.2.1 Upload Form....30
Gambar 3.1.2.2 Image Dataset Upload Form....31
Gambar 3.1.2.3 Audio Dataset Upload Form....31
Gambar 3.1.2.3 Mapper Upload Form.... 31
Gambar 3.1.2.4 Album and Music Sort Button....32
Gambar 3.1.2.5 Result....32
Gambar 3.1.2.6 Pagination....33
Gambar 3.1.2.7 Total Output dan Processing Time.... 33
Gambar 3.2.1.1 Gambar Folder Structure Backend Album Picture Finder.... 34
Gambar 3.2.2.1 Struktur Backend MIR.... 36
Gambar 4.1.1 Tangkapan Layar Awal Sebelum Upload Image Dataset.... 40
Gambar 4.1.2 Menekan Choose File....41
Gambar 4.1.3 Memilih File Dataset.... 41
Gambar 4.1.4 Menekan tombol Pictures....42
Gambar 4.1.5 Tangkapan Layar Setelah Upload....42
Gambar 4.1.6 Tangkapan Layar Sebelum Upload Audio Dataset.... 43
Gambar 4.1.7 Memilih file Audio File....43
Gambar 4.1.8 Memilih Tombol Audio....44
Gambar 4.1.9 Tangkapan Layar Setelah Upload Audio Dataset....44
Gambar 4.1.10 Tangkapan Layar Sebelum Upload Audio Dataset.... 45
Gambar 4.1.11 Menekan Choose File.... 46
Gambar 4.1.12 Memilih File Mapper....46
Gambar 4.1.13 Tangkapan Layar Sesudah Upload Mapper....46

Gambar 4.1.14 Memilih File Image Query..... 47
Gambar 4.1.15 Tangkapan Layar Hasil Pencarian Image Query..... 47
Gambar 4.1.16 Memilih File Audio Query..... 48
Gambar 4.1.17 Tangkapan Layar Hasil Pencarian Audio Query..... 48

DAFTAR PEMBAGIAN TUGAS

Tabel 1. Pembagian Tugas

Kegiatan		PIC
1.	Image Retrieval and API	Nathaniel Jonathan Rusli
2.	Music Information Retrieval	Maheswara Bayu Kaindra
3.	Website Designing and Implementation	Jessica Allen

BAB I
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 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.

Gambar 1.1.1. 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. Tahun ke tahun, *image processing* selalu menjadi fokus utama dari tugas besar 2

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

Anda sudah melewati Tugas Besar 1 yaitu tentang matriks dan implementasi terhadap berbagai hal. Matriks adalah salah satu komponen yang penting dalam aplikasi aljabar vektor. Di dalam Tugas Besar 2 ini, anda 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, anda akan menggunakan aljabar vektor untuk mencari perbandingan antar satu audio dengan audio yang lain. Anda akan menggunakan konsep yang bernama *Music Information Retrieval* atau MIR untuk mencari dan mengidentifikasi suara berdasarkan fitur-fitur yang dimilikinya. Tidak hanya itu, anda juga akan menggunakan konsep Principal Component Analysis (PCA) untuk mencari kumpulan audio melalui deteksi wajah berbagai orang (anggap saja mereka sebagai seorang penyanyi).

1.2 Information Retrieval

Information Retrieval adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Pada tugas besar ini, anda akan berkutik dengan 2 jenis Information Retrieval. *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, kalian akan mengimplementasikan *Image Retrieval* dengan menggunakan Principal Component Analysis dan *Music Information Retrieval* dengan menggunakan humming.

1.2.1 Album Picture Finder - Principal Component Analysis

Setiap audio pastinya memiliki gambar albumnya sendiri masing-masing. Terkadang kita lebih mengingat suatu visual dibandingkan nama dari lagu itu sendiri. Untuk memudahkan pengguna yang hanya memiliki gambar dari suatu album, maka dibutuhkan album finder dengan menggunakan teknik Principal Component Analysis (PCA).

Sebelum implementasi PCA, jangan lupa untuk memasukkan pemetaan nama audio dengan nama gambar yang bersangkutan, gunakan file .txt atau .json untuk melakukan

pemetaan audio dengan gambar yang bersangkutan. Gunakan kreativitas anda untuk memetakan audio dengan gambar tersebut.

```
mapper.txt
------------------------------------------------------------

audio_file pic_name
audio_1.mid pic_1.png
audio_2.mid pic_2.png
audio_3.mid pic_3.png
audio_4.mid pic_4.png
.
.
```

```
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 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:

1.2.1.1 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 $M \cdot N$:

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

1.2.1.2 Data Centering (Standardization)

Pada step ini lakukan standarisasi data di sekitar nilai 0. Hitung rata rata dari

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

setiap gambar untuk suatu piksel.

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.

$$xij' = xij - \mu j$$

1.2.1.3 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:

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

- U: matriks eigenvector (komponen utama),
- 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:

- Uk: matriks eigenvector dengan n-dimensi.

1.2.1.4 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:

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

di mana:

- 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, hitung jarak Euclidean antara gambar query dengan semua gambar dalam

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

dataset:

- $d(q,z_i)$ = Jarak antara gambar query q dan gambar ke-i dalam ruang komponen utama.
- zi = 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.

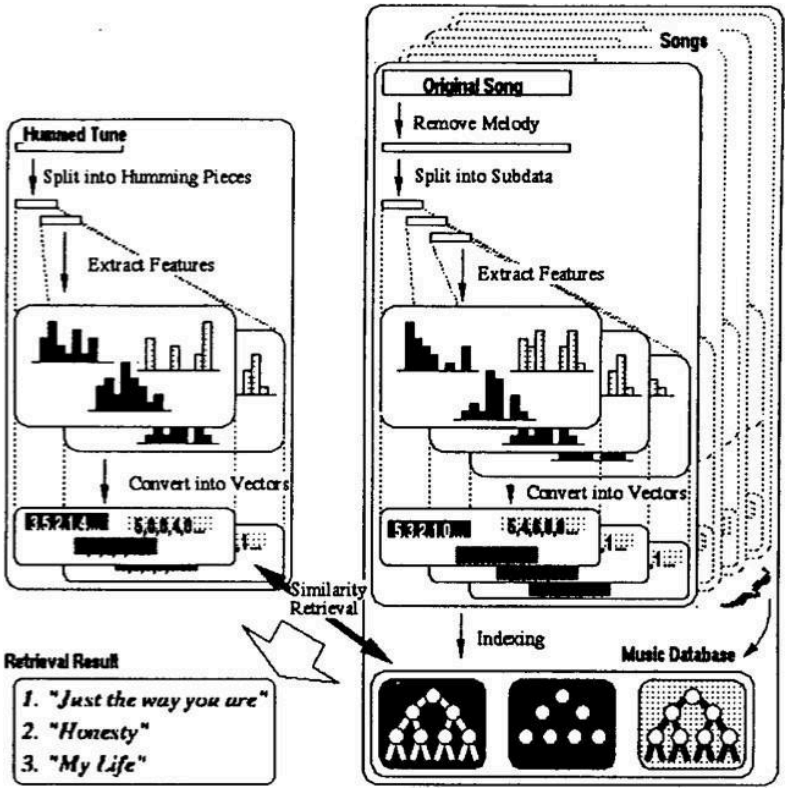
  Lalu, Urutkan hasil berdasarkan jarak terkecil.

1.2.1.5 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.

1.2.2 Music Information Retrieval - Query by Humming



Gambar 1.2.2.1. Tahapan pada MIR dengan metode 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.

1.2.2.1 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.

1.2.2.2 Ekstraksi Fitur

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.

Normalisasi

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

$$Hnorm = \frac{H[d]}{\sum_{d}^{127} H[d]}$$

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

Penghitungan Similaritas

Ubah setiap histogram menjadi sebuah vektor dan lakukan perhitungan kemiripan nya 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 :

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

Gambar 1.2.2.2.1. Cosine Similarity Formula

1.3 Arsitektur Website

Untuk tugas besar kali ini, Anda akan diminta untuk membuat website. Oleh karena itu, kami akan memberikan gambaran singkat terkait arsitektur website yang akan dibuat.

1. *Audio query*, berisi file audio yang akan digunakan untuk melakukan pencocokan suara
2. *Voice recording module*, tombol untuk memproseskan suara dari microphone pengguna dan disimpan untuk diproses
3. Kumpulan suara yang disimpan dalam bentuk dataset, bisa dilakukan dengan cara mengunggah *multiple audio*, folder, .zip, .rar, dan file kompresi lainnya ke dalam website anda. Setelah memasukkan kumpulan suara tersebut, diharapkan semua audio yang ada dalam dataset dimunculkan di halaman website. Agar pengguna tidak melakukan scrolling yang berlebihan, silahkan gunakan *pagination*.a
4. Bila mengerjakan bonus PCA, anda harus memetakan file lagu ke file gambar yang bersangkutan, gambar tersebut merepresentasikan album dari lagu tersebut. Pemetaan dapat disimpan dalam file .txt dan .json.

Gambar 1.3.1. Contoh tampak depan dari website Information Retrieval (Humming)

Gambar 1.3.2. Contoh tampak depan dari website Information Retrieval (Album)

Anda tidak harus membuat website dengan tampak depan yang sama persis dengan gambar diatas. Namun, yang terpenting adalah komponen ada gambar diimplementasikan

semua dalam website yang anda buat. Secara garis besar, berikut adalah komponen-komponen yang perlu anda buat dalam website pada tugas besar ini.

1. Judul Website
2. Tombol untuk mengunggah audio ketika dalam moda query audio by humming dan mengunggah gambar ketika dalam moda query album finder with PCA
3. Tombol untuk merekam dan mengunggah rekaman audio tersebut (bonus)
4. Tombol untuk mengunggah dataset gambar wajah dan audio
5. Tombol untuk memilih antara query audio by humming dan album finder by PCA
6. Tombol untuk mengunggah mapper
7. Daftar lagu beserta album gambar yang bersangkutan yang ada di dataset
8. Pagination agar lagu yang muncul tidak banyak. Jumlah lagu dalam satu pag9. dibebaskan asalkan tidak membuat pengguna harus scrolling terlalu lama
9. Tombol search untuk melakukan pencarian
10. Persentase kemiripan pada setiap audio atau album yang ada pada dataset
11. Waktu dari eksekusi program (disarankan dalam waktu ms)

Secara umum, berikut adalah cara-cara dari bagaimana sebuah website Information Retrieval bisa digunakan.

1. Pengguna terlebih dahulu mengunggah dataset pada tombol audio dan picture yang tertera di website.
2. Pengguna mengunggah pemetaan file lagu dengan judulnya dan gambar album yang bersangkutan (Bersifat opsional dan bonus pada bagian judul)
3. Pengguna melakukan query dengan dua cara yaitu mengunggah file audio atau gambar
4. Pengguna memilih akan melakukan pencarian dengan metode album finder with PCA atau query by humming
5. Program akan menampilkan kumpulan audio atau gambar yang mirip, diurutkan dari yang memiliki kemiripan paling tinggi ke yang paling rendah. Setiap audio atau gambar yang muncul diberi persentase kemiripannya.
6. Terdapat informasi terkait jumlah audio atau gambar yang muncul, dan waktu eksekusi programnya.

BAB II
TEORI SINGKAT

2.1 Sistem Temu Balik Suara (MIR)

Music Information Retrieval merupakan suatu disiplin atau penerapan yang berfokus mengenai ekstraksi, analisis, dan penerimaan informasi suara dari berbagai format (MIDI, wav, dan sebagainya). MIR menggabungkan implementasi pemrosesan sinyal, *machine learning*, dan *data science* untuk menganalisis data yang berkaitan dengan musik atau suara.

Menurut Downie (2003), *Music Information Retrieval* adalah upaya penelitian multidisiplin yang berupaya mengembangkan skema, mekanisme, dan alat pencarian berbasis konten yang inovatif untuk data musik. Tujuan dari MIR adalah untuk memungkinkan pengguna berinteraksi dengan konten musik melalui segala bentuk data suara.

2.2.2 Skema umum *Query by Humming*

Secara umum, query by humming memiliki beberapa langkah utama, yaitu [1]pemrosesan audio, [2]ekstraksi (dibahas pada 2.2 - Metode Ekstraksi Fitur Berdasarkan Humming), Normalisasi, dan perhitungan similaritas.

2.2.2.1 Pemrosesan Audio

Pemrosesan audio merupakan tahapan pertama pada MIR, yaitu dengan numerasi suatu file (pada kasus ini MIDI) menjadi suatu list yang berisi nada-nada yang muncul pada file audio tersebut. Pada kasus ini list tersebut berisi tuple yang terdiri atas [1]*pitch* (nada saat ini) dan [2]*time signature* (kapan kemunculan nada tersebut dinormalisasi dalam detik).

2.2.2.2 Ekstraksi

Dibahas pada 2.2 - Metode Ekstraksi Fitur Berdasarkan Humming.

2.2.2.3 Normalisasi

Normalisasi merupakan salah satu tahap yang penting untuk memastikan bahwa data yang sudah diekstrak (seperti *pitch interval* dan *absolute tones*) dicatat dalam rentang yang bisa dikomparasi dan bersifat standar. Proses normalisasi memudahkan perhitungan similaritas berdasarkan tempo yang terstandarisasi.

2.2.2.3 Perhitungan Similaritas

Perhitungan similaritas merupakan tahapan terakhir dari MIR khususnya *Query by Humming*, yaitu mencari kemiripan antara file query dengan file-file audio di dataset. Perhitungan similaritas dapat dilakukan dengan berbagai metode, namun untuk konteks ini digunakan metode *cosine similarity*, di mana semua histogram dibentuk menjadi vektor multidimensi. Dua vektor yang berbeda dihitung nilai cosinusnya, sehingga dapat ditentukan apakah kedua vektor memiliki sudut yang kecil. Semakin kecil sudut antara dua buah vektor, semakin mirip kedua file audio tersebut.

2.2 Metode Ekstraksi Fitur Berdasarkan Humming (QBH)

2.2.1 Konsep Dasar *Query by Humming* (QBH)

Metode ekstraksi fitur berdasarkan Humming merupakan salah satu implementasi *Music Information Retrieval* yang digunakan untuk rekognisi suatu data audio berdasarkan input suara. Metode yang seringkali disebut sebagai *Query by Humming* memungkinkan pengguna untuk mencari data musik tertentu berdasarkan input (*humming*) sebagian atau seluruh melodi dari suatu data musik.

Menurut Kosugi dkk. dalam *A Practical Query-by-Humming System for a Large Music Database*, sistem QBH bertujuan untuk mengidentifikasi kemiripan antara input melodi dari pengguna dengan file audio dalam database, mengatasi tantangan seperti perbedaan pitch, tempo, dan tingkat intensitas suara file input.

2.2.2 Jenis-Jenis Metode Ekstraksi Fitur berdasarkan konsep QBH

Pada konteks proyek ini, terdapat tiga jenis metode ekstraksi fitur berdasarkan konsep QBH, yaitu *Absolute Tone Based*, *Relative Tone Based*, dan *First Tone Based*.

2.2.2.1 Absolute Tone Based

Absolute Tone Based merupakan salah satu metode untuk mengekstraksi nada-nada audio berdasarkan seringnya kemunculan suatu nada dalam suatu file audio. Pengimplementasian *absolute tone based* dapat berupa histogram frekuensi kemunculan nada. Di mana terdapat suatu skala MIDI (rentang 0-127) untuk menghasilkan representasi statistik berupa histogram dari nada-nada dalam data musik.

Absolute Tone Based memiliki beberapa kelebihan, yaitu mudah diimplementasikan, memiliki skalabilitas yang jelas, dan independen terhadap melodi dinamis. Di samping

kelebihan-kelebihan tersebut, *absolute tone based* tidak relevan untuk lagu-lagu yang memiliki pitch dengan dinamika tinggi, karena akan terdapat nada-nada hilang atau ternormalisasi menjadi nol) akibat penyetaraan tangga nada. Selain itu, *Absolute Tone Based* seringkali merekam nada-nada yang seharusnya tidak ada (misalnya hasil *noise*) pada histogram, sehingga mengurangi akurasi perhitungan.

2.2.2.2 Relative Tone Based

Menurut Kosugi (2000), *Relative Tone Based* merupakan salah satu metode QBH untuk menganalisis interval antar nada berdekatan sebagai karakteristik utama untuk membandingkan melodi, yang memungkinkan sistem mengabaikan perbedaan key atau tempo.

Relative Tone Based memiliki beberapa kelebihan dan kekurangan menurut pandangan berbagai pihak. Menurut Pardo dan Birmingham, Pendekatan berbasis interval pitch relatif lebih relevan dalam mencocokkan input humming dengan database, karena interval antar nada bersifat invarian terhadap pitch tertentu (tangga nada atau *key* tidak mempengaruhi perhitungan).

Di samping itu, perhitungan dengan *relative tone based* hanya akan efektif apabila audio yang diproses bersifat dinamis (memiliki interval nada yang tinggi). Audio yang memiliki sedikit perubahan pitch akan menghasilkan grafik yang tidak signifikan sehingga cenderung sulit untuk dibandingkan. Selain itu, kesalahan dalam proses *pitch tracking* akan sangat mempengaruhi histogram dan perhitungan similaritas.

2.2.2.3 First Tone Based

Pendekatan *First Tone Based* menggunakan pitch referensi (yaitu pitch pertama) untuk menangkap pola interval yang invarian terhadap perbedaan key, sehingga cocok untuk sistem pencarian melodi berbasis *humming* (Pardo, Birmingham, 2002). Pendekatan *First Tone Based* cukup relevan dan sering digunakan karena seringkali nada awal memberikan konteks dalam proses pencocokan melodi. Meski sama-sama menggunakan selisih *pitch*, *first tone based* cenderung lebih stabil dibanding *relative tone based* (pola interval lebih stabil dibandingkan *absolute pitch*).

First tone based sederhana, efisien, dan hemat memori; namun sangat bergantung pada nada awal, sehingga apabila nada awal tidak dinyanyikan dengan benar (atau tidak tepat), FTB akan gagal memberikan hasil pencocokan yang akurat. Selain itu, FTB tidak memperhitungkan hubungan antar-nada, sehingga kurang cocok untuk perhitungan melodi kompleks.

2.3 Image Retrieval dengan PCA

2.3.1 Konsep Dasar Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah teknik statistik yang sering digunakan di berbagai bidang, termasuk psikologi, untuk membantu pengembangan tes objektif serta pengukuran kepribadian dan kecerdasan. PCA secara linear mengubah sekumpulan variabel asli menjadi kumpulan variabel baru yang lebih kecil dan saling tidak berkorelasi, namun tetap dapat merepresentasikan informasi dari variabel asli (Dunteman, 1989).

Menurut Tabachnick (2001), PCA adalah teknik statistik yang diterapkan pada sekumpulan variabel untuk mengidentifikasi hubungan antar variabel. Teknik ini bertujuan menemukan variabel-variabel yang berkorelasi satu sama lain tetapi tetap independen terhadap subset variabel lainnya. Subset tersebut biasanya merupakan kombinasi dari beberapa variabel yang digabungkan ke dalam suatu faktor. Faktor ini mencerminkan proses mendasar yang menyebabkan adanya korelasi antar variabel.

Tujuan utama PCA adalah menjelaskan variasi dalam sekumpulan variabel yang diamati berdasarkan sejumlah dimensi yang lebih kecil. PCA menyederhanakan variabel yang banyak menjadi sedikit tanpa kehilangan informasi signifikan. Secara khusus, tujuan PCA meliputi:

1. Merangkum pola korelasi antar variabel yang diamati.
2. Mereduksi jumlah besar variabel menjadi hanya beberapa faktor utama.
3. Memberikan definisi operasional (misalnya dalam bentuk persamaan regresi) untuk dimensi utama berdasarkan variabel yang diamati.
4. Menguji teori yang mendasari kumpulan variabel tersebut (Tabachnick, 2001).

2.3.2 Keunggulan dan Keterbatasan PCA dalam *Image Retrieval*

Teknik Principal Component Analysis (PCA) merupakan teknik yang memiliki banyak keunggulan, namun bukan merupakan teknik yang sempurna sehingga juga memiliki keterbatasan. Menurut Jolliffe dan Cadima (2016), PCA memiliki berbagai keunggulan seperti berikut:

1. Reduksi Dimensi yang Efisien: PCA mampu mengurangi dimensi data tanpa kehilangan informasi yang signifikan, terutama ketika sejumlah kecil komponen utama (principal components) mampu menjelaskan sebagian besar variansi dalam data.

2. Penghapusan Redundansi: PCA menciptakan variabel baru (principal components) yang tidak berkorelasi (orthogonal), sehingga menghilangkan redundansi yang ada di antara variabel asli.
3. Penanganan Noise: Dengan berfokus pada variansi terbesar dalam data, PCA dapat secara efektif mengurangi pengaruh noise dan variasi kecil yang mungkin tidak relevan untuk analisis.
4. Kemudahan Implementasi: PCA memiliki algoritma yang sederhana dan dapat diimplementasikan dengan cepat menggunakan teknik linear algebra, seperti dekomposisi nilai singular (SVD).

Di sisi lain, menurutnya PCA memiliki berbagai keterbatasan seperti berikut:

1. Hubungan Non-Linear Tidak Ditangkap: PCA hanya efektif untuk data yang hubungan antar variabelnya linear. Jika terdapat hubungan non-linear, PCA gagal menangkap informasi penting tersebut.
2. Sensitivitas terhadap Skala: PCA sangat sensitif terhadap skala variabel. Variabel dengan skala besar cenderung mendominasi hasil PCA, sehingga data harus dinormalisasi terlebih dahulu.
3. Kesulitan Interpretasi: komponen utama yang dihasilkan PCA adalah kombinasi linear dari variabel asli, yang sering kali sulit untuk diinterpretasikan secara langsung.
4. Ketergantungan pada Variansi: PCA mengasumsikan bahwa variansi yang besar adalah sumber informasi yang penting. Dalam beberapa kasus, variansi kecil mungkin lebih relevan, tergantung pada konteks analisis.
5. Kehilangan Informasi Potensial: meskipun PCA menjaga sebagian besar variansi data, beberapa informasi penting dari variabel asli dapat hilang, terutama jika data tidak dapat direpresentasikan dengan baik dalam dimensi yang lebih rendah.

2.3.3 Penerapan PCA dalam *Image Retrieval*

Dalam penelitian yang dilakukan oleh Hafiane dan Atif (2008) berjudul *"Content-Based Image Retrieval Using PCA and Spatial Information"*, Principal Component Analysis (PCA) digunakan sebagai teknik utama untuk ekstraksi fitur dalam sistem Content-Based Image Retrieval (CBIR). PCA berperan dalam mereduksi dimensi data gambar yang besar dengan tetap mempertahankan informasi penting yang relevan untuk pencarian. Dengan PCA, setiap gambar

direpresentasikan dalam bentuk vektor komponen utama yang tidak berkorelasi, yang kemudian digunakan untuk menghitung kemiripan antara gambar query dan gambar dalam dataset.

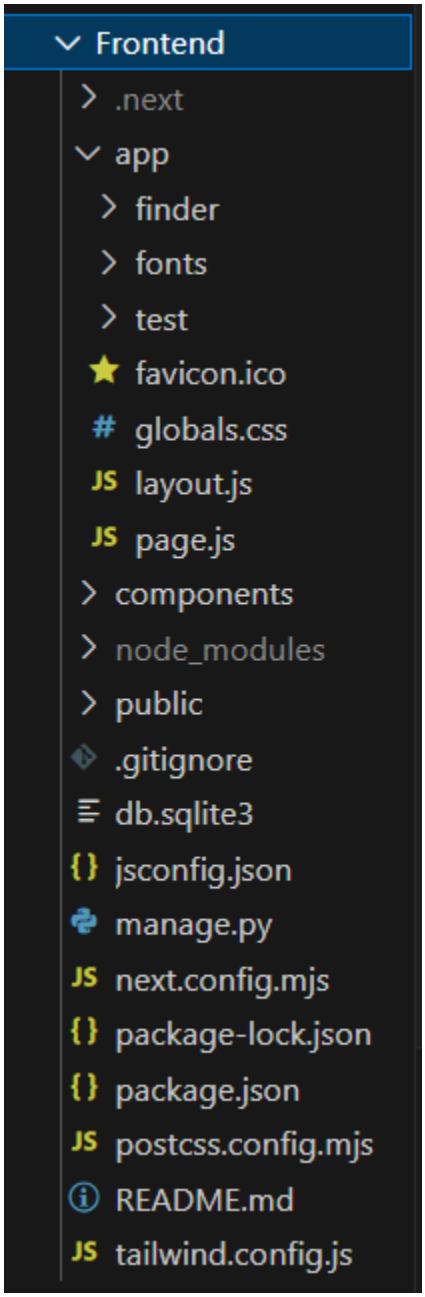
Selain itu, penelitian ini menunjukkan bahwa penggabungan PCA dengan informasi spasial dari gambar mampu meningkatkan akurasi retrieval secara signifikan. Informasi spasial berfungsi untuk mempertahankan hubungan geometris antarfitur dalam gambar, sehingga gambar-gambar dengan struktur visual serupa lebih mudah dikenali. Kombinasi ini membantu mengatasi keterbatasan PCA yang hanya menangkap hubungan linear dengan menambahkan konteks geometris pada representasi data.

Hasil eksperimen menunjukkan bahwa metode ini tidak hanya meningkatkan performa pencarian dalam hal akurasi, tetapi juga efisiensi komputasi karena jumlah dimensi yang dianalisis berkurang secara signifikan. Penelitian ini menjadi salah satu bukti bahwa PCA, ketika dikombinasikan dengan teknik lain seperti analisis spasial, dapat meningkatkan performa sistem CBIR pada dataset gambar dengan kompleksitas tinggi.

BAB III

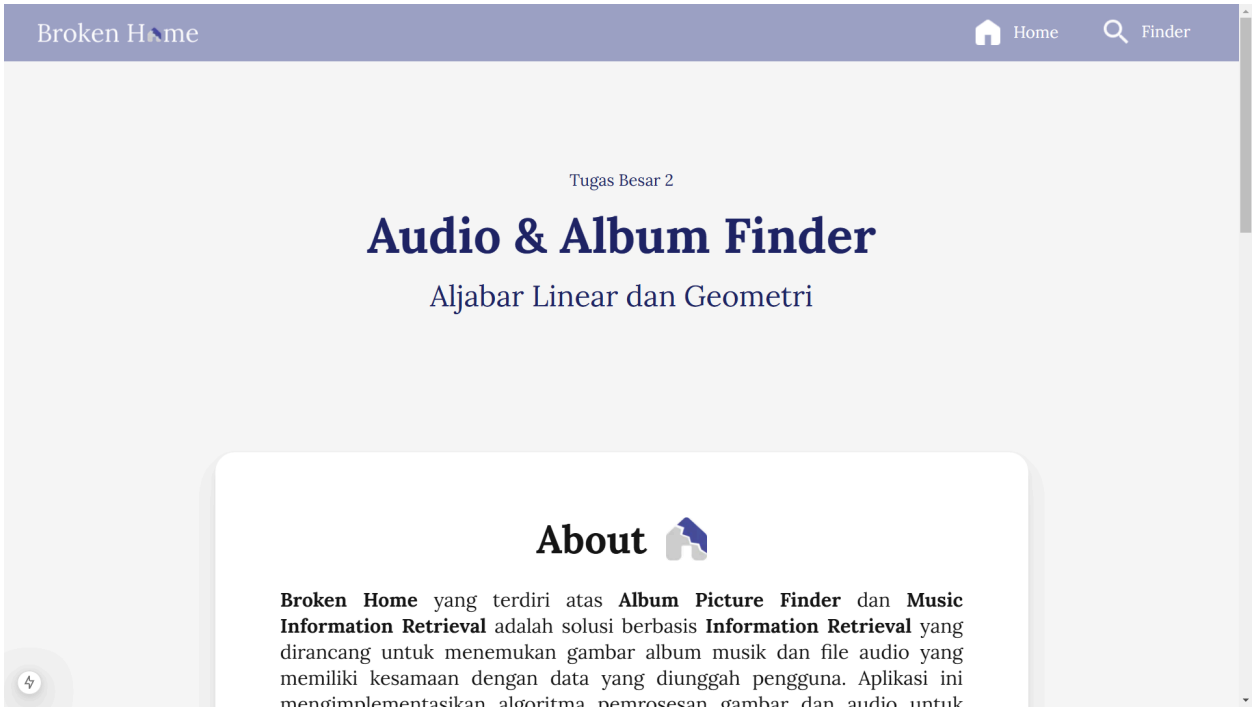
IMPLEMENTASI PUSTAKA DAN PROGRAM

3.1 Arsitektur Website (Frontend)



Gambar 3.1.1 Arsitektur Website Frontend

3.1.1 Home Page



Gambar 3.1.1 Home Page

Home page merupakan halaman pertama yang dimunculkan ketika pengguna membuka website. Halaman ini berguna untuk memberikan pengenalan kepada pengguna atas apa yang ada di dalam website. Disertai dengan navbar, yang memudahkan navigasi pengguna dalam website.

3.1.1.1 Header

Gambar 3.1.1.1 Header

Bagian ini terletak di bagian atas halaman dengan latar belakang biru tua. Berisi judul besar "Audio & Album Finder", sub judul "Tugas Besar 2", dan deskripsi singkat "Aljabar Linear

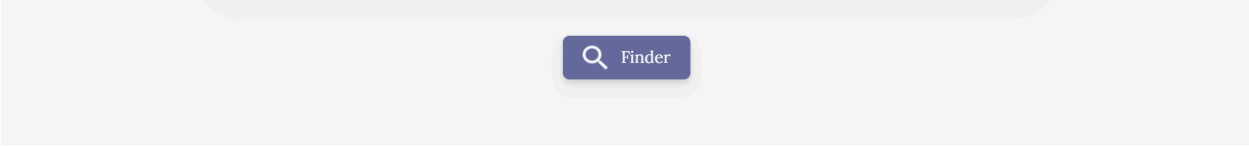
dan Geometri". Header berfungsi untuk memperkenalkan aplikasi dan memberikan konteks proyek (tugas besar untuk mata kuliah tertentu) dan memastikan pengguna memahami tujuan utama aplikasi sejak awal.

3.1.1.2 About

Gambar 3.1.1.2 About

Berisi penjelasan singkat tentang aplikasi. Teks yang ada di dalamnya bertujuan untuk menjelaskan bahwa aplikasi ini adalah proyek kolaborasi dari tim "Broken Home". About berfungsi untuk memberikan pengantar ringkas tentang aplikasi kepada pengguna. Menjelaskan tujuan atau fungsionalitas utama website, yaitu membantu pengguna .

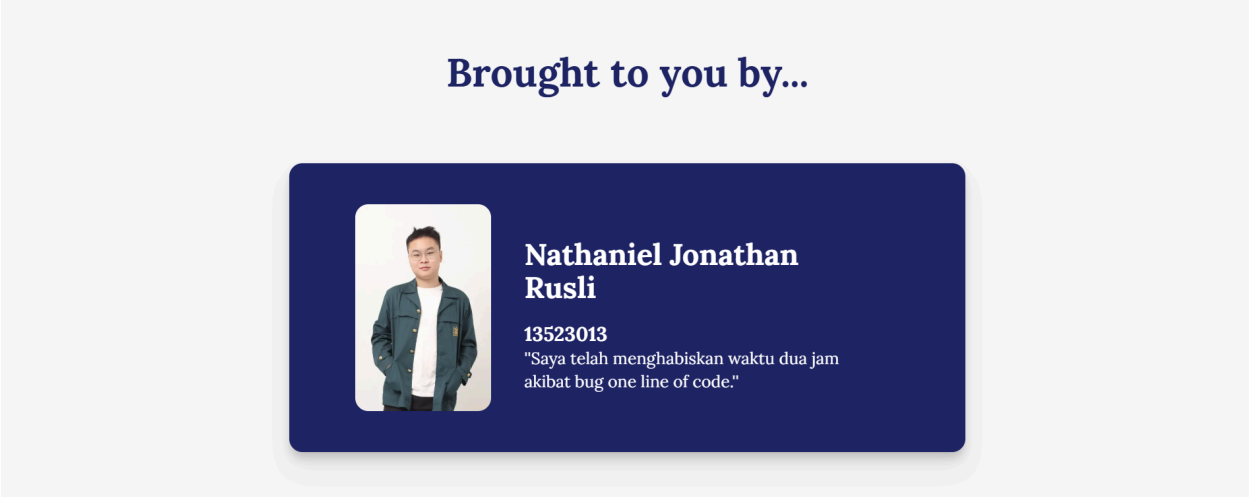
3.1.1.2 Finder Button



Gambar 3.1.1.2 Finder Button

Finer Button yang terletak di bawah About merupakan shortcut untuk menggunakan fitur utama website, yaitu Finder Page. Ketika menekan button ini, pengguna akan diarahkan menuju ke page tersebut.

3.1.1.3 Developer Information

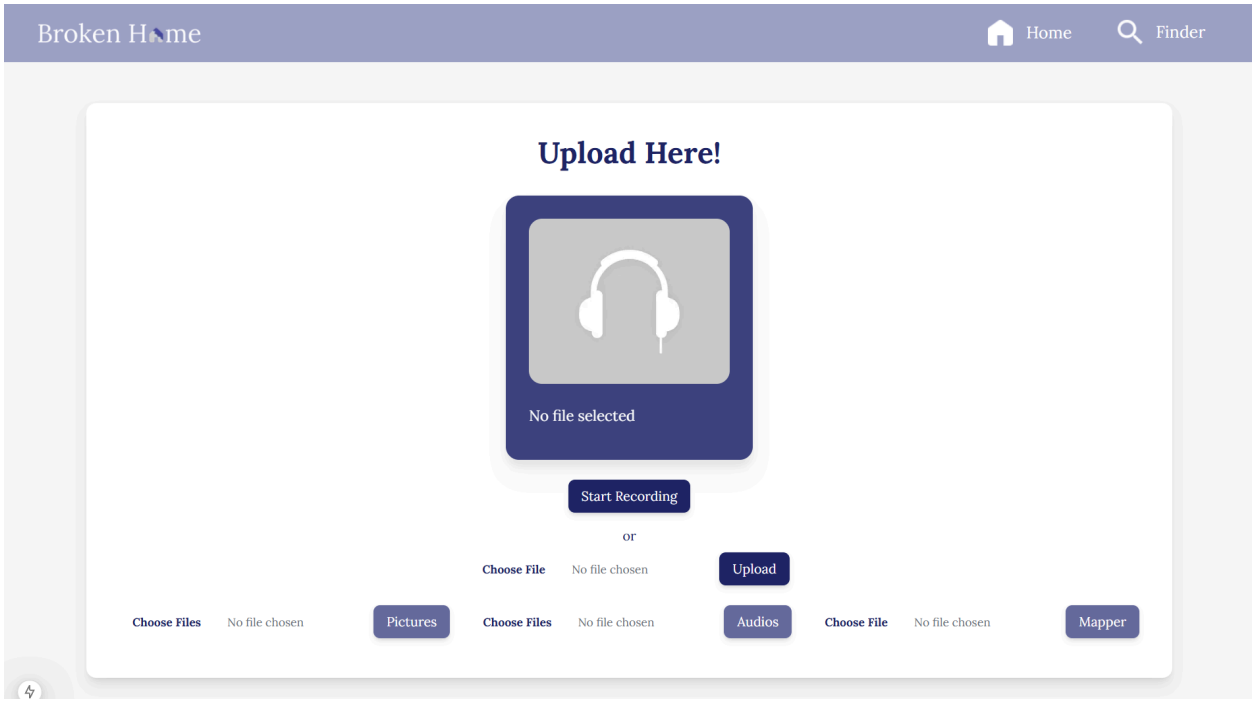


Gambar 3.1.1.3.1 Developer Information (1)

Gambar 3.1.1.3.2 Developer Information (2)

Bagian ini berisi informasi tentang anggota tim pengembang website. Setiap anggota tim ditampilkan dengan komponen HomeCard yang berisi nama anggota, NIM anggota, serta pesan dan kesan singkat dari anggota mengenai pengerjaan tugas besar. Bagian ini berfungsi untuk memberikan apresiasi atas pengembang website, serta meningkatkan personalisasi website.

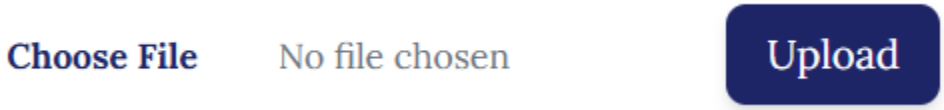
3.1.2 FinderPage



Gambar 3.1.2 Finder Page

Finder page merupakan page yang berisi fitur utama dari website Broken Home. Finder page berisi finder berdasarkan album picture dan berdasarkan file midi. Disertai dengan navbar, yang memudahkan navigasi pengguna dalam website.

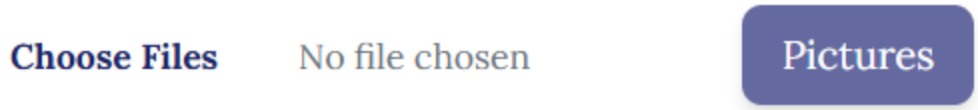
3.1.2.1 Upload Form



Gambar 3.1.2.1 Upload Form

Form ini digunakan untuk mengunggah file query yang ingin diidentifikasi. Pengguna dapat mengunggah file picture maupun midi.

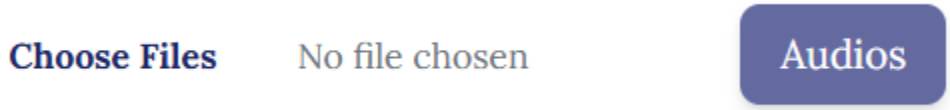
3.1.2.2 Image Dataset Upload Form



Gambar 3.1.2.2 Image Dataset Upload Form

Form ini digunakan untuk mengunggah kumpulan data gambar yang akan menjadi acuan atau basis pencarian atau pengurutan. Form ini akan memungkinkan pengguna untuk menambahkan dataset gambar untuk pencarian berbasis gambar.

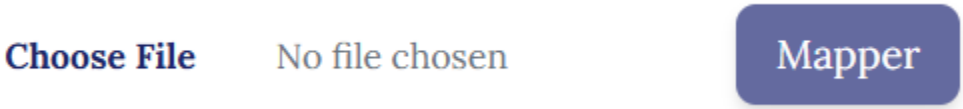
3.1.2.3 Audio Dataset Upload Form



Gambar 3.1.2.3 Audio Dataset Upload Form

Form ini digunakan untuk mengunggah dataset audio, yang berupa file midi. Fungsi dari form ini adalah untuk mengunggah koleksi audio sebagai basis pencarian untuk fitur yang melibatkan analisis suara atau humming.

3.1.2.3 Mapper Upload Form



Gambar 3.1.2.3 Mapper Upload Form

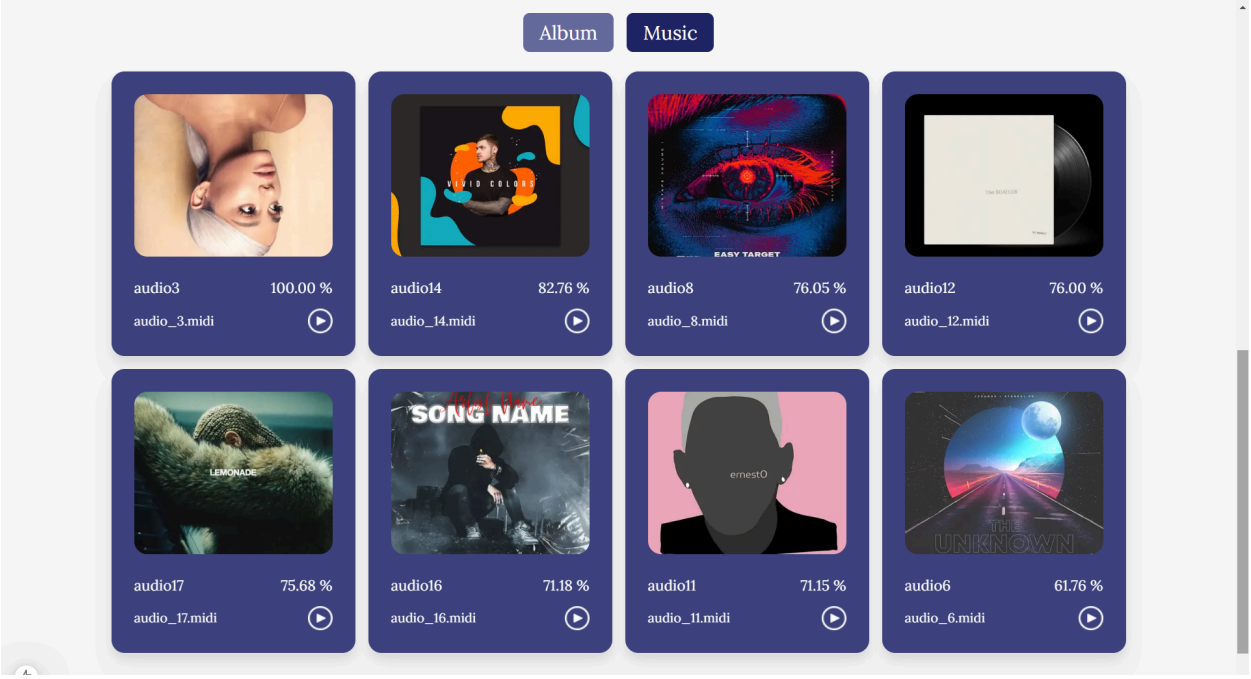
Form ini digunakan untuk mengupload mapper file yang digunakan untuk mapping nama file audio dengan file picture, serta dengan nama lagu. Form ini berfungsi untuk memungkinkan pengguna mengupload file yang akan menjadi patokan penamaan hasil pencarian.

3.1.2.4 Album and Music Sort Button

Gambar 3.1.2.4 Album and Music Sort Button

Kedua tombol ini berfungsi untuk membedakan pencarian berdasarkan gambar atau berdasarkan audio. Jika berdasarkan audio, maka jika Album yang terpilih, tidak akan memunculkan hasil apa-apa, sedangkan jika Music yang terpilih, maka akan dimunculkan hasil search berdasarkan audio. Begitu juga sebaliknya.

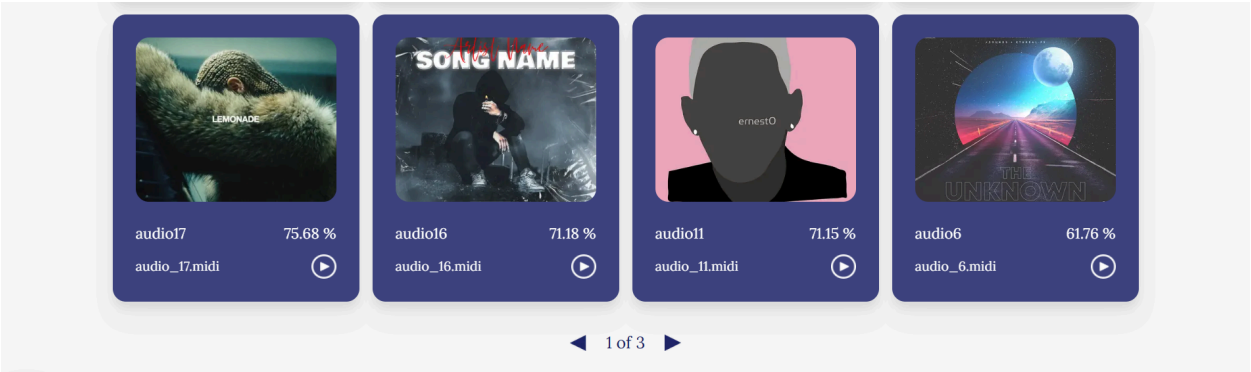
3.1.2.5 Result



Gambar 3.1.2.5 Result

Bagian ini menampilkan hasil pencarian dalam bentuk Card, yang berisi nama lagu, nama file audio, serta persentase kemiripan. Pengguna juga dapat play MIDI file sesuai dengan nama audio file.

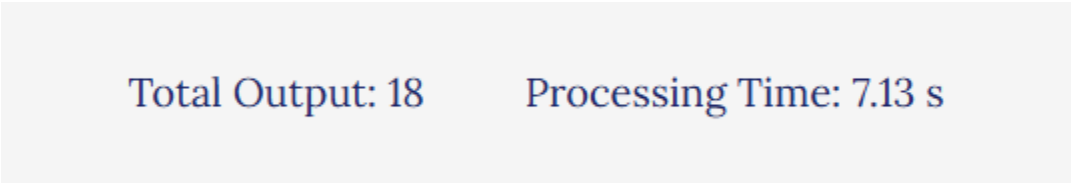
3.1.2.6 Pagination



Gambar 3.1.2.6 Pagination

Bagian ini menampilkan kontrol navigasi untuk memunculkan lebih banyak data hasil pencarian jika jumlah hasil melebihi kapasitas satu halaman, dalam kasus ini, ditampilkan sebanyak 8 hasil dalam satu halaman. Kontrol ini dapat menelusuri halaman sebelum maupun sesudahnya dan ditempatkan di bawah hasil pencarian.

3.1.2.7 Total Output dan Processing Time



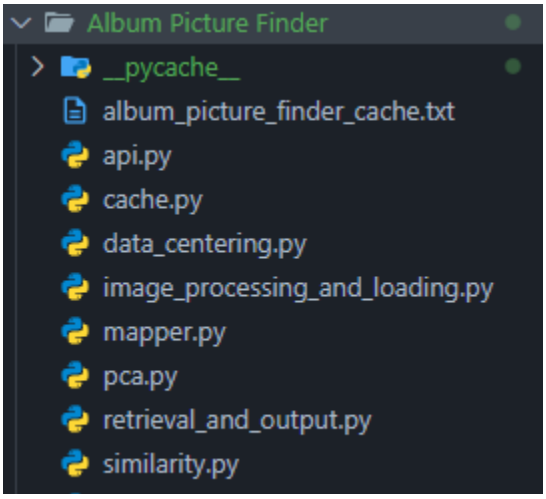
Gambar 3.1.2.7 Total Output dan Processing Time

Bagian ini menampilkan total output hasil pencarian yang diberikan oleh website, serta berapa lama processing time yang diperlukan untuk melakukan pencarian. Bagian ini berfungsi untuk memperjelas hasil yang diberikan dari backend dan terletak di bawah pagination.

3.2 Arsitektur Program Information Retrieval (*Backend*)

3.2.1 Arsitektur *Backend* Album Picture Finder

Seluruh tahap pemrosesan *image retrieval* dipisah per modul berdasarkan masing-masing tahapan mulai dari *image processing* hingga *retrieval and output*. Selain itu, terdapat juga modul untuk sistem *cache*, *mapping* antara hasil gambar dengan *file* audio serta nama audio bersangkutan, dan API untuk menghubungkan *backend* dengan *frontend*.



Gambar 3.2.1.1 Gambar *Folder Structure Backend* Album Picture Finder

3.2.1.1 *Image Processing and Loading* (image_processing_and_loading.py)

Modul *image processing and loading* memuat *function* seperti load_and_process_image yang berfungsi untuk memilih seluruh *file* gambar pada *dataset*, lalu memroses masing-masing gambar dan diubah menjadi array 1 dimensi dari nilai-nilai piksel *grayscale*-nya. Seluruh *grayscale value* dari masing-masing gambar akan ditampung pada sebuah array 2 dimensi dan dikembalikan.

3.2.1.2 *Data Centering or Standardization* (data_centering.py)

Modul *data centering* memuat function standardize_data yang menerima *grayscale value* seluruh gambar yang telah diproses sebelumnya berupa array 2 dimensi. Matriks atau array 2 dimensi tersebut pun kemudian akan diproses dengan menghitung rata-rata piksel masing-masing gambar lalu seluruh piksel gambar tersebut akan dikurang nilai rata-rata. Nilai piksel gambar yang telah diproses atau distandarisasi tersebut kemudian dikembalikan.

3.2.1.3 Principal Component Analysis (pca.py)

Modul Principal Component Analysis memuat berbagai *function* untuk membantu proses komputasi dan kalkulasi nilai *eigenvector* dari SVD hingga proyeksi hasil *n components* terhadap masing-masing *image*. Pertama, terdapat *function* perform_svd yang akan mengembalikan seluruh matriks hasil dekomposisi termasuk *eigenvector matrix*. Terdapat juga *function* get_principal_components yang akan mengambil sejumlah n komponen utama dari hasil *eigenvector matrix*. Terakhir, terdapat *function* project_images yang akan memproyeksikan serta mengembalikan seluruh n komponen yang tepat pada masing-masing gambar.

3.2.1.4 *Similarity Calculation* (similarity.py)

Modul *similarity calculation* memuat fungsi calculate_euclidean_distance yang dijadikan basis untuk perhitungan *similarity*. Jarak masing-masing gambar terhadap *query image* akan dikembalikan untuk diproses lebih lanjut pada proses selanjutnya.

3.2.1.5 *Retrieval and Output* (retrieval_and_output.py)

Modul *retrieval and output* menyediakan sebuah *pipeline* proses dari awal (*retrieval*) hingga *output* hasil *similarity*. Terdapat fungsi preprocess_query_image sebagai *pipeline* pemrosesan *query image* (gambar yang ingin dibandingkan). Lalu, terdapat fungsi output_similarity yang akan menghitung jarak seluruh gambar pada *dataset* dengan *query image*, mengurutkannya berdasarkan yang paling mirip, dan mengembalikan jumlah gambar teratas sesuai *input* top_n_images.

3.2.1.6 *Caching System* (cache.py)

Modul *caching system* dibuat untuk mendukung pemrosesan seluruh gambar pada *dataset* sehingga bila *dataset* dan variabel lainnya tidak berubah, hasil komputasi PCA pun akan digunakan ulang, menghemat biaya waktu dan komputasi. Proses *caching* ini didukung oleh fungsi-fungsi pembantu seperti calculate_dataset_hash untuk membandingkan sama tidaknya *dataset* dan variabel, load_cache untuk memuat data *cache*, save_cache dan check_cache untuk menyimpan ulang *cache* yang berubah, serta preprocess_database_image yang menjadi *pipeline* pemrosesan seluruh image *dataset* dari *loading* hingga *output similarity*.

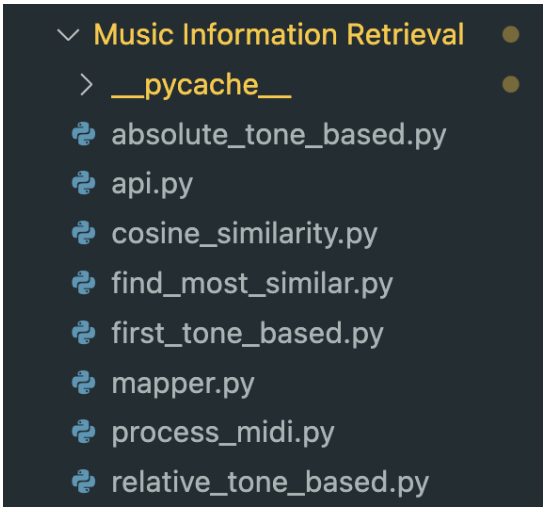
3.2.1.7 *Result Mapping* (mapper.py)

Modul *result mapping* merupakan modul pembantu dalam menghubungkan hasil *image* yang paling *similar* dengan nama *file audio* serta nama *audio* yang bersangkutan. Fungsi pembantu load_mapper akan memuat *file mapper* (mapper.txt) pada *directory* yang sesuai, lalu membaca dan memvalidasi setiap baris pada mapper.txt. Informasi setiap baris yang valid akan diproses dan disimpan pada sebuah variabel bernama mapper.

3.2.2 Arsitektur *Backend Music Information Retrieval*

Secara struktural, arsitektur backend *Music Information Retrieval* didekomposisi secara temporal, yaitu dibagi berdasarkan proses-proses *Music Information Retrieval*. Apabila dilihat secara state, perhitungan similaritas dimulai dari pemrosesan file-file .mid atau .midi menjadi list

of tuple yang terdiri atas pitch dan waktu kemunculan pitch. Kemudian, pitch dinormalisasi dan mengalami ketiga metode *query by humming* dan disimpan dalam bentuk histogram. Histogram yang ternormalisasi akan dibentuk menjadi vektor-vektor yang akan dihitung cosinus sudutnya untuk mendapatkan similaritas. Berikut merupakan struktur backend menurut proses tersebut.



Gambar 3.2.2.1 Struktur Backend MIR

3.2.2.1 *MIDI File Processing* (process_midi.py)

Modul *MIDI File Processing* mencakup fungsi-fungsi yang mendukung segala perhitungan dari awal hingga akhir (fungsi untuk melakukan normalisasi) seperti transpose_to_c, normalize_length, clamp, dan normalize_histogram_cumulative. Di luar itu, process_midi.py memiliki fungsi utama untuk memproses MIDI file menjadi array of tuple melalui fungsi process_midi. Fungsi process midi memeriksa channel 0 (channel pertama) dari file MIDI dan mengembalikan *error message* apabila tidak ditemukan nada pada channel 0. Apabila file MIDI memenuhi syarat, setiap nada dan time stamp dari nada tersebut (dalam satuan detik) akan ditambahkan ke dalam list yang nantinya akan dikembalikan bersamaan dengan rata-rata pitch dan standar deviasi pitch.

3.2.2.2 *Absolute Tone Based* (absolute_tone_based.py)

Modul *Absolute Tone Based* mencakup satu fungsi sesuai tujuannya, yaitu melakukan perhitungan histogram dengan metode *Absolute Tone Based* melalui fungsi absolute_tone_based. Fungsi ini memanggil process_midi dan melanjutkan proses hingga mengembalikan histogram hasil perhitungan. Dalam pemrosesan list of tuple, digunakan fungsi clamp untuk mencegah nada

keluar dari range karena normalisasi nada (pada kasus ini digunakan tangga nada C sebagai referensi). Fungsi akan mengembalikan histogram frekuensi kemunculan nada.

3.2.2.3 *Relative Tone Based* (relative_tone_based.py)

Modul *Absolute Tone Based* mencakup fungsi absolute_tone_based yang bekerja sesuai fungsinya untuk membentuk histogram berdasarkan selisih antar dua nada bersebelahan (dalam kasus ini elemen nada yang bersebelahan dari array of tuple). Fungsi ini mengembalikan histogram frekuensi kemunculan selisih nada (dengan skala -127 sampai 127).

3.2.2.4 *First Tone Based* (first_tone_based.py)

Modul *First Tone Based* mencakup fungsi first_tone_based yang bekerja sesuai fungsinya untuk menghasilkan histogram berdasarkan selisih setiap nada dengan nada pertama, sama seperti modul-modul sebelumnya, first_tone_based juga memanggil process_midi. Fungsi akan mengembalikan histogram yang berisi frekuensi kemunculan selisih setiap nada dalam file terhadap nada pertama (dengan skala -127 sampai 127).

3.2.2.5 *Calculating Similarities* (cosine_similarity.py)

Modul *Calculating Similarities* mencakup fungsi cosine_similarity_custom yang menerima input dua buah vektor dan mengkalkulasi nilai kosinus kedua vektor tersebut melalui *dot product* dari kedua vektor dibagi dengan jumlah panjang kedua vektor. Semakin dekat nilai kosinus dengan 1, semakin mirip kedua vektor tersebut.

3.2.2.6 *Database Sorting* (find_most_similar.py)

Modul *Database Sorting* merupakan modul yang mengimplementasi lebih lanjut modul sebelumnya, yaitu membandingkan setiap file dengan file query berdasarkan similaritas dan menyimpannya ke dalam list of tuple yang terdiri atas rank, filename, dan similarity. Fungsi ini akan dipanggil di dalam API dan akan melakukan pengurutan berdasarkan file input dari pengguna.

3.2.3 Arsitektur *Backend* API

Seluruh API *backend* dari aplikasi diletakkan pada *directory* API pada sebuah file bernama api.py, seperti API untuk *query* berdasarkan *album image file* dan *music file*, API untuk memudahkan proses mengunggah *album dataset* dan *music dataset* serta *mapper file*.

3.2.3.1 *Album Query API*

Album query API dengan *route* image diimplementasikan untuk menunjang persoalan *output* hasil *similarity* berdasarkan *input* file gambar. Walaupun jenis *file* sudah divalidasi pada sisi *client*, jenis *file* akan divalidasi ulang pada sisi server. *Query image* akan di-*preprocess* seperti gambar-gambar yang terdapat pada database, lalu akan dibandingkan satu per satu serta dicari gambar yang paling mirip berdasarkan *euclidean distance*. API pun akhirnya akan mengembalikan *field* rank, pic_name, audio_fie, audio_name, distance, dan similarity_percentage yang sudah melalui proses *mapping*.

3.2.3.2 *Music Query API*

Music query API dengan *route* music diimplementasikan untuk menunjang persoalan *output* hasil *similarity* berdasarkan *input* file audio, serupa dengan API sebelumnya. Perbedaannya adalah *similarity* antara *query audio* dengan *file* audio dapat langsung dihitung tanpa melalui proses *preprocessing*. API akan mengembalikan *field* rank, pic_name, audio_file, audio_name, similarity_percentage yang telah melalui proses *mapping*.

3.2.3.3 *Album Dataset* dan *Music Dataset* API

Album dataset API dengan *route* upload-images dan *Music dataset* API dengan *route* upload-music diimplementasikan untuk mempermudah proses *upload* dataset *image* maupun *audio* yang akan digunakan dalam proses *image retrieval* dan *music information retrieval*. Kedua API memiliki fungsi serupa dan akan dipanggil bergantung pada file *query* yang diunggah pada *frontend*. Seperti API lainnya, *dataset* API menggunakan penerapan *application* dari FastAPI. Dengan metode 'POST', API akan menerima *multiple* files dan *compressable files* yang melalui proses validasi jenis *file*. Jenis *file* di luar yang diperbolehkan akan menghasilkan *error message* dan file dengan jenis yang valid akan melalui pemrosesan lebih lanjut menggunakan *helper functions*. *Compressable files* yang akan di-*extract* dan dialokasikan pada directory untuk menyimpan *dataset*.

3.2.3.4 *Mapper* API

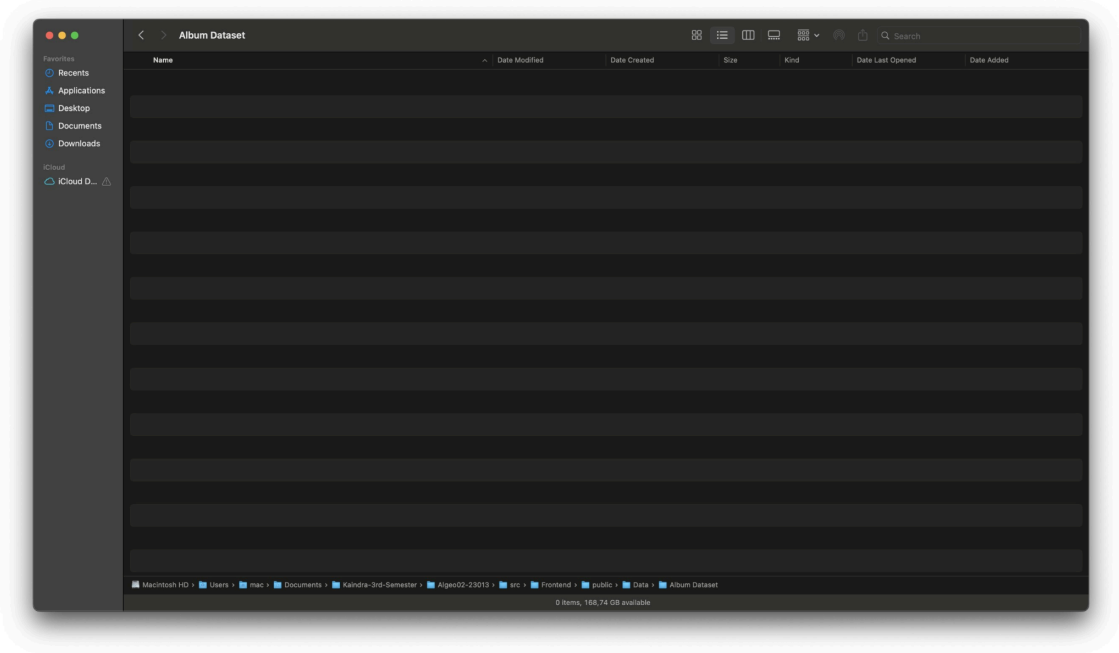
Mapper API diimplementasikan dengan *route* uploader-mapper untuk mempermudah proses *upload* penghubung antara *file image* dan *audio* yang berkorelasi menggunakan sebuah *file* mapper berformat txt. Dengan menggunakan *application* FastAPI bermetode 'POST', API akan menerima input file yang semula akan divalidasi berformat txt dengan bantuan fungsi

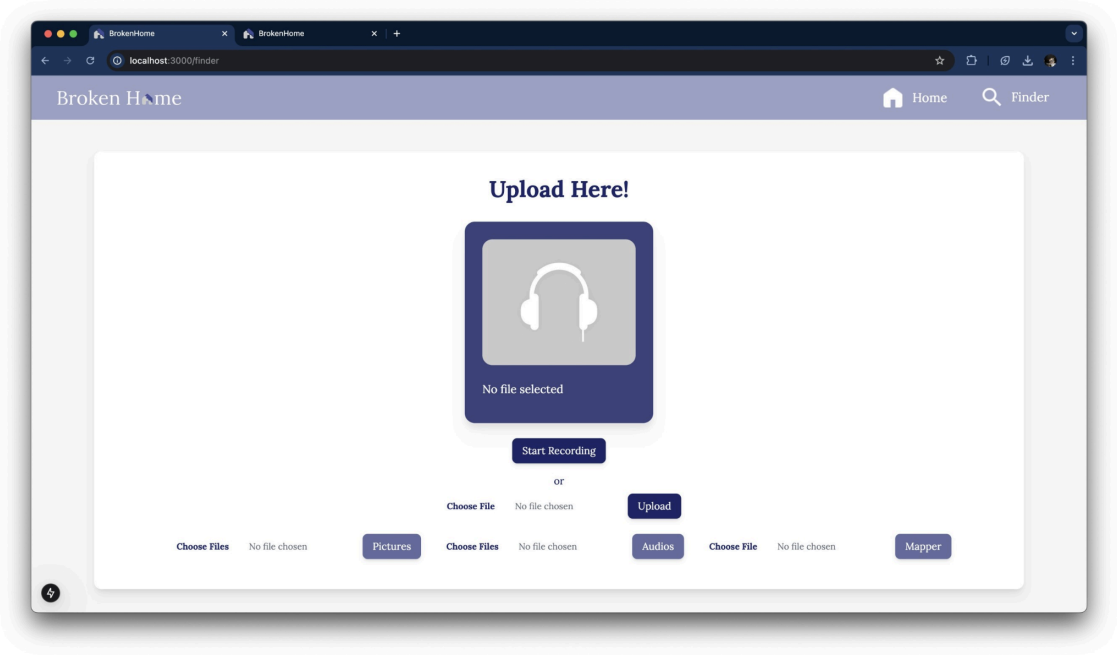
validate_mapper_format. Bila *input file* valid, maka file akan diproses lebih lanjut dan disimpan pada *directory* yang sesuai untuk menyimpan file mapper.txt.

BAB IV
EKSPERIMEN

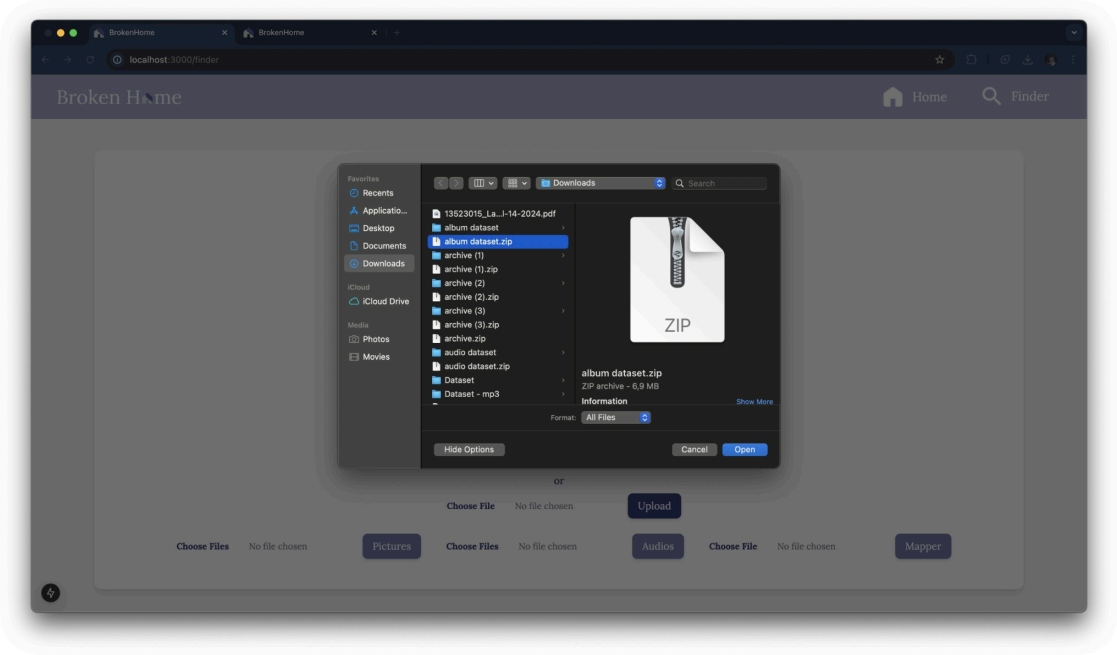
4.1 Eksperimen

Tabel 2. Eksperimen

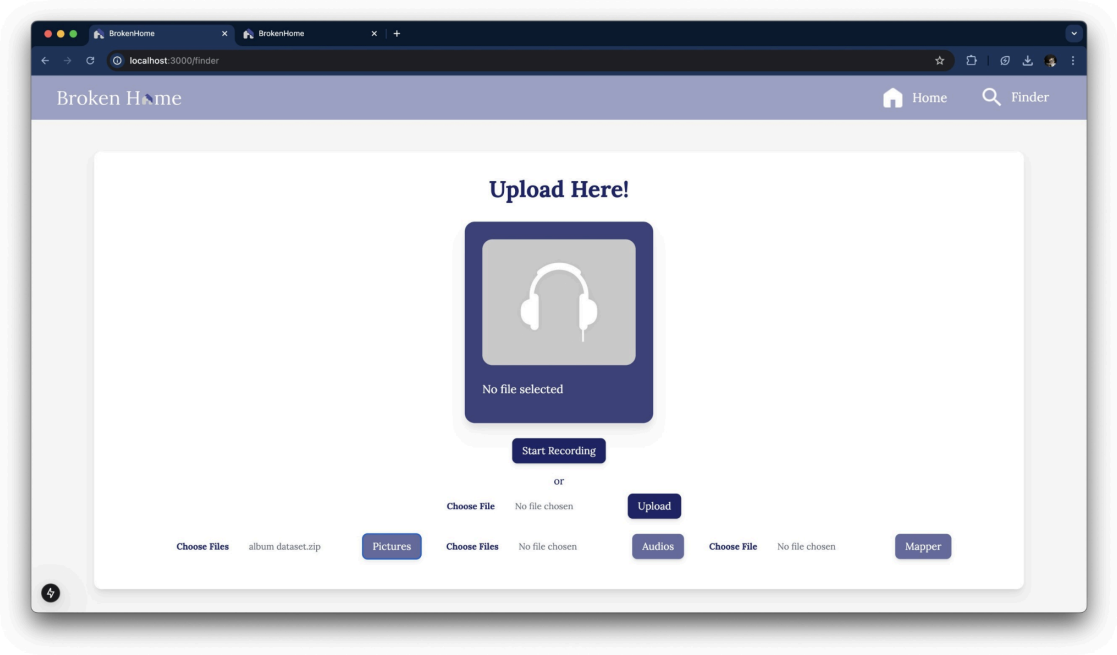
No.	Soal dan Hasil Eksperimen
a.	Mengupload Dataset Album Sebelum:  **Gambar 4.1.1 Tangkapan Layar Awal Sebelum Upload Image Dataset** Upload:



Gambar 4.1.2 Menekan Choose File

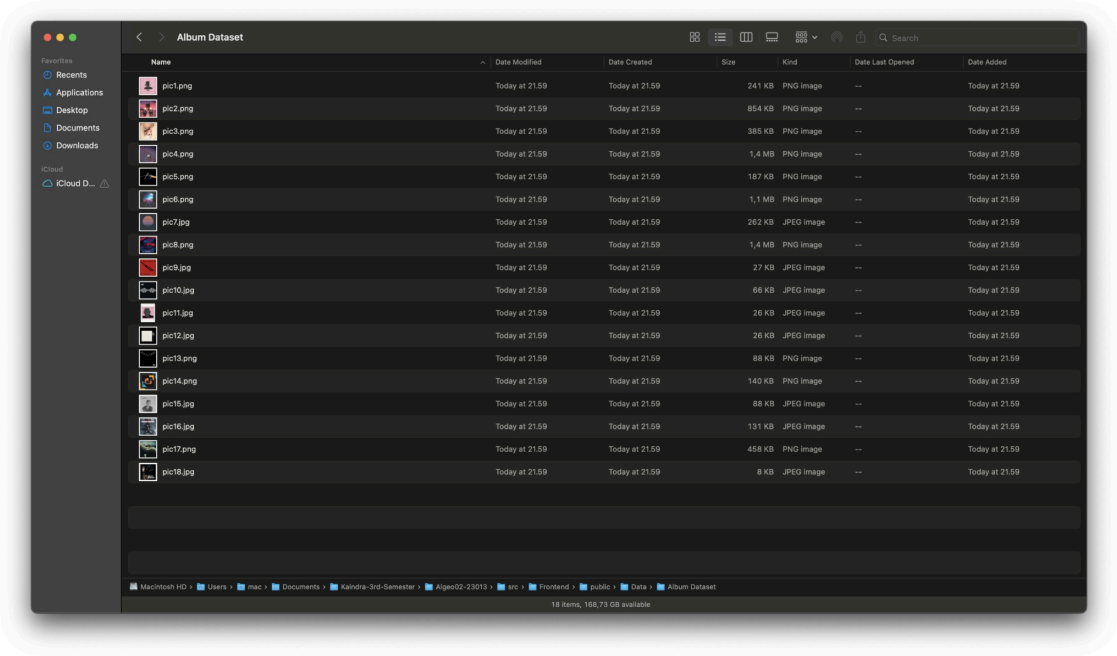


Gambar 4.1.3 Memilih File Dataset



Gambar 4.1.4 Menekan tombol Pictures

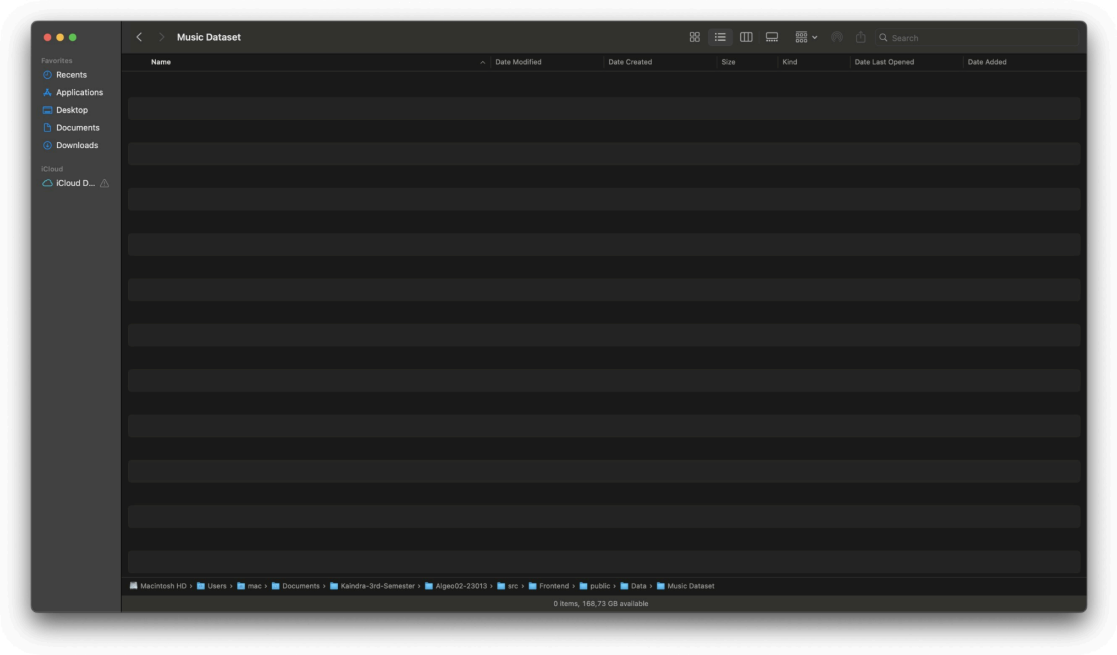
Sesudah:



Gambar 4.1.5 Tangkapan Layar Setelah Upload

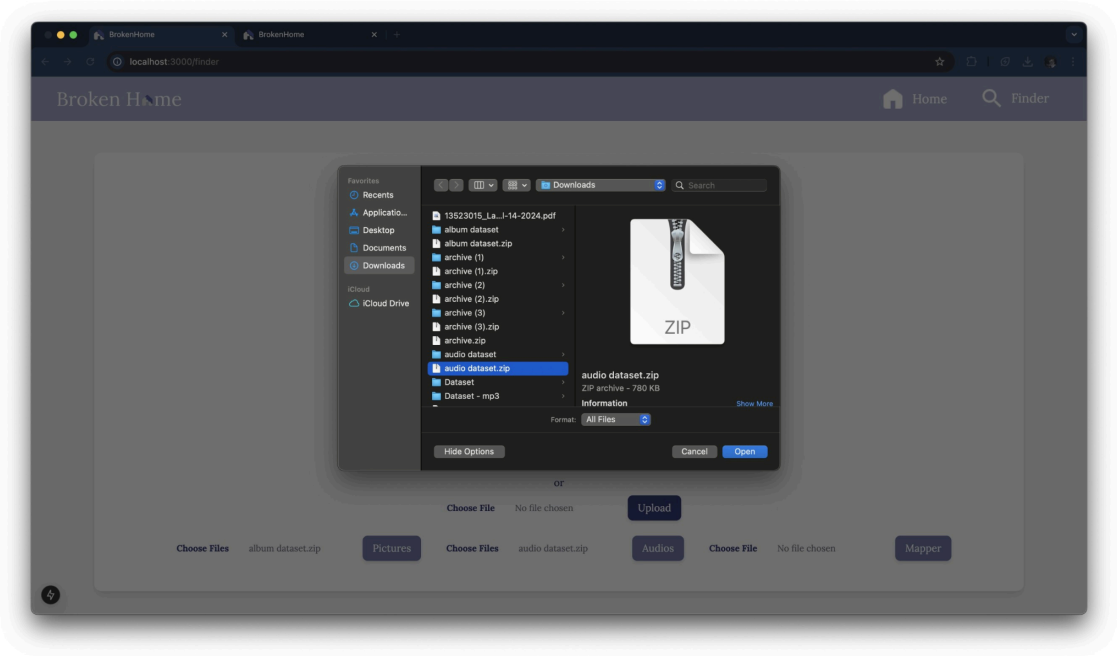
b. Mengupload Dataset Audio

Sebelum:

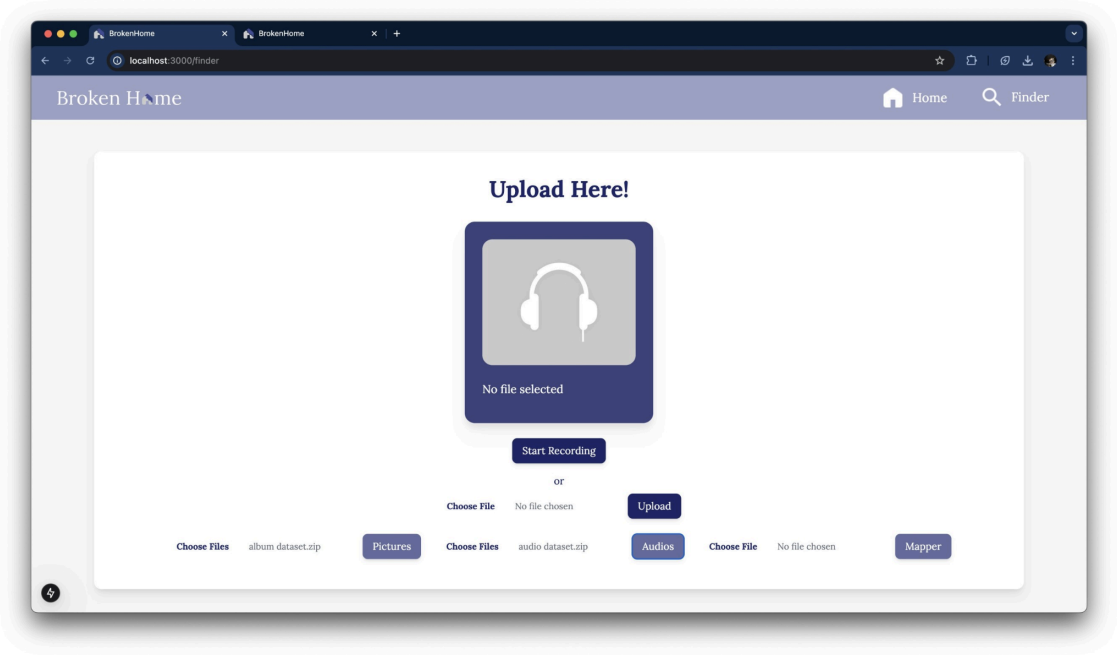


Gambar 4.1.6 Tangkapan Layar Sebelum Upload Audio Dataset

Upload:

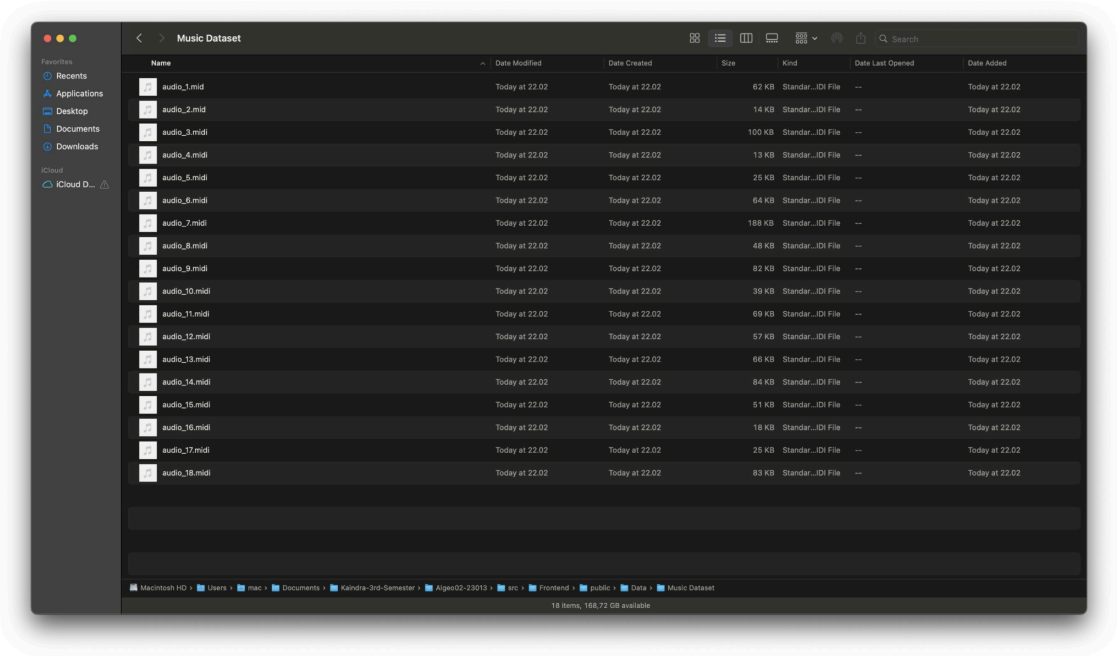


Gambar 4.1.7 Memilih file Audio File



Gambar 4.1.8 Memilih Tombol Audio

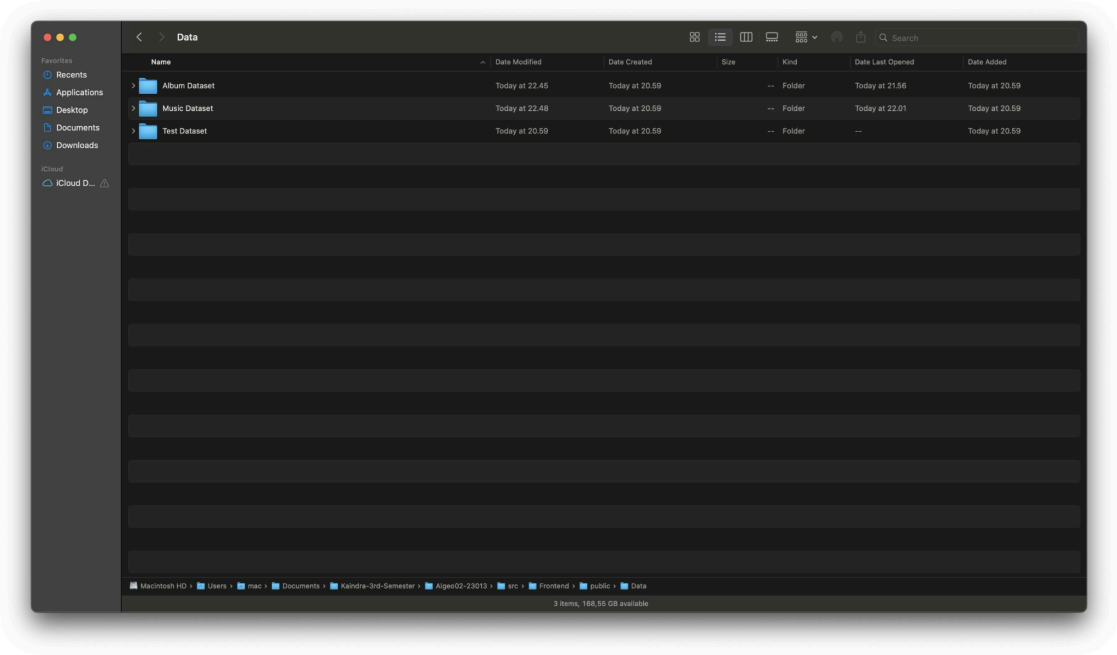
Sesudah:



Gambar 4.1.9 Tangkapan Layar Setelah Upload Audio Dataset

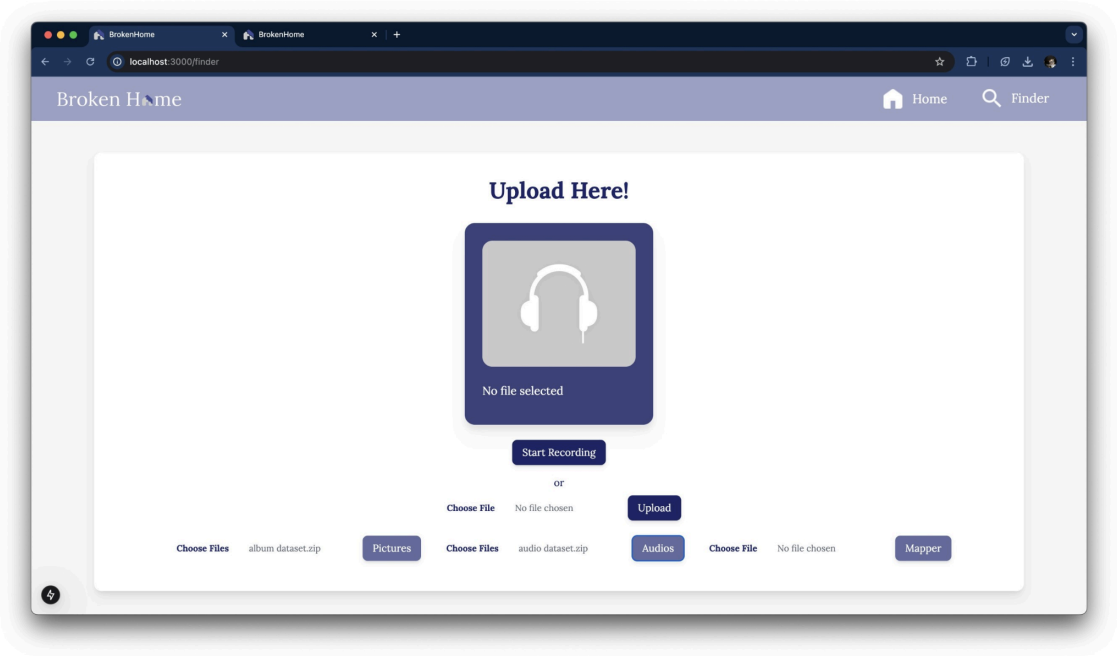
c. Mengupload Mapper

Sebelum:

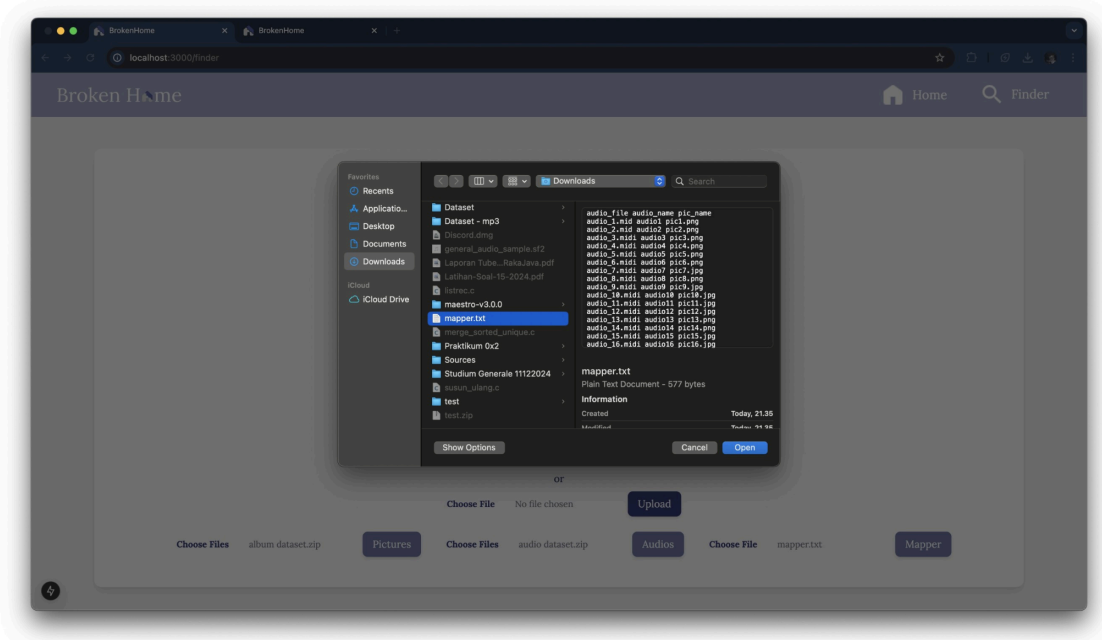


Gambar 4.1.10 Tangkapan Layar Sebelum Upload Audio Dataset

Upload:

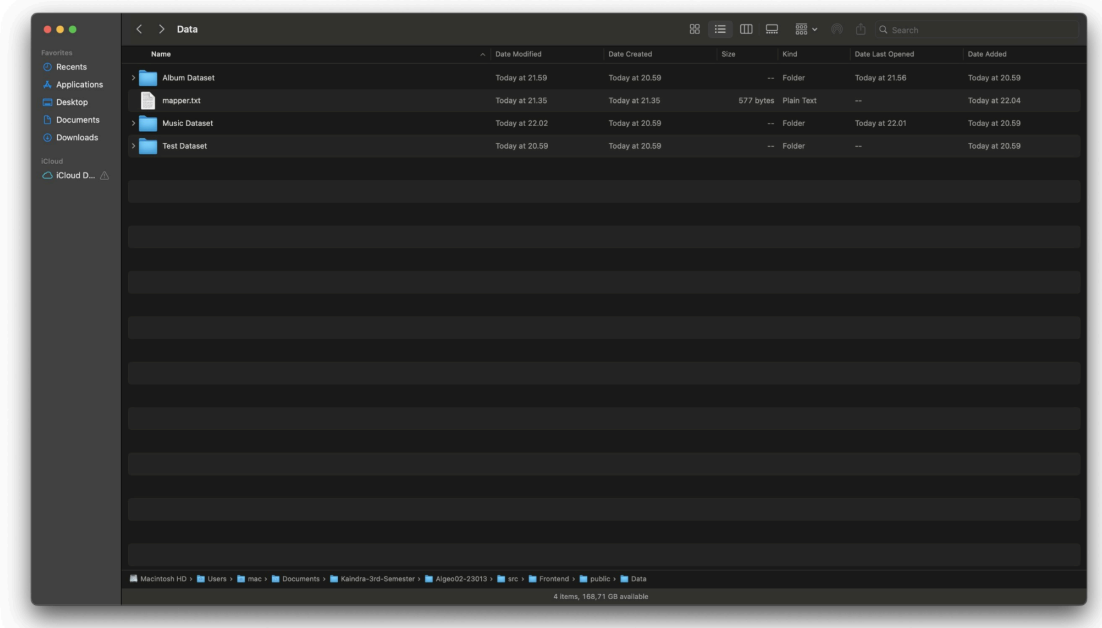


Gambar 4.1.11 Menekan Choose File



Gambar 4.1.12 Memilih File Mapper

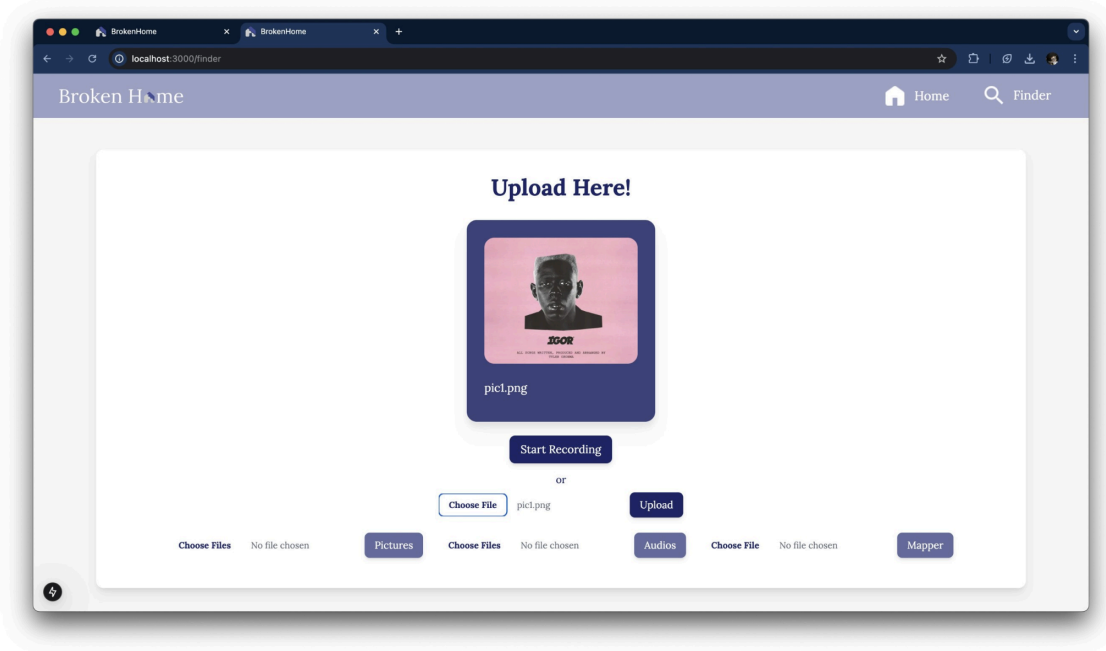
Sesudah:



Gambar 4.1.13 Tangkapan Layar Sesudah Upload Mapper

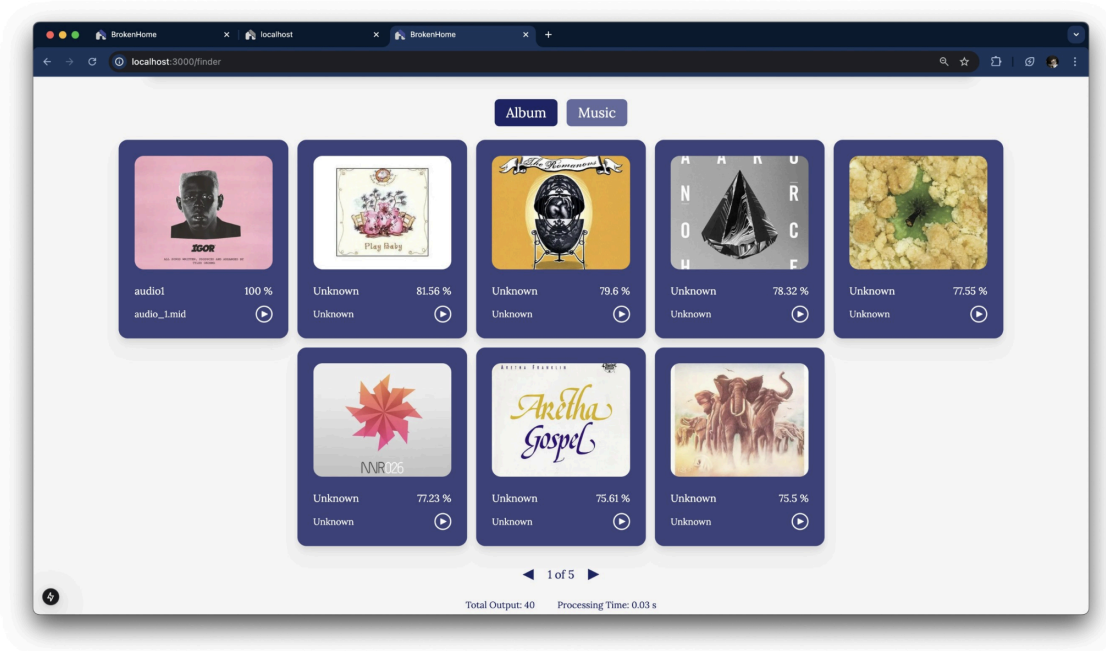
d.	Mengupload Query Image dan Memunculkan Hasil Pencarian

Upload:



Gambar 4.1.14 Memilih File Image Query

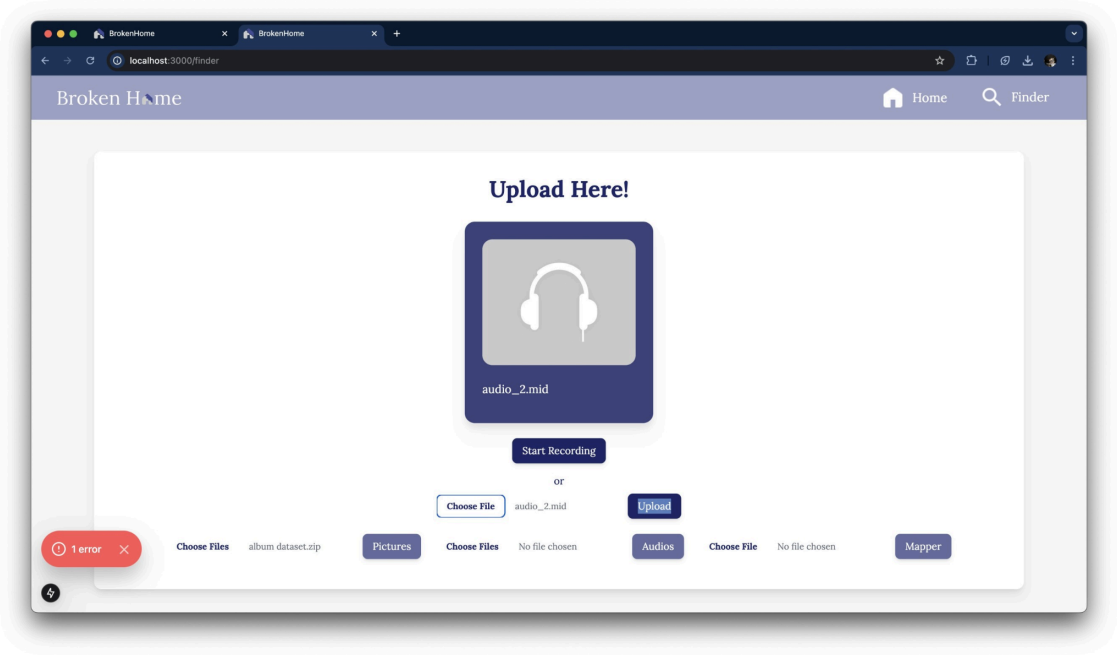
Sesudah:



Gambar 4.1.15 Tangkapan Layar Hasil Pencarian Image Query

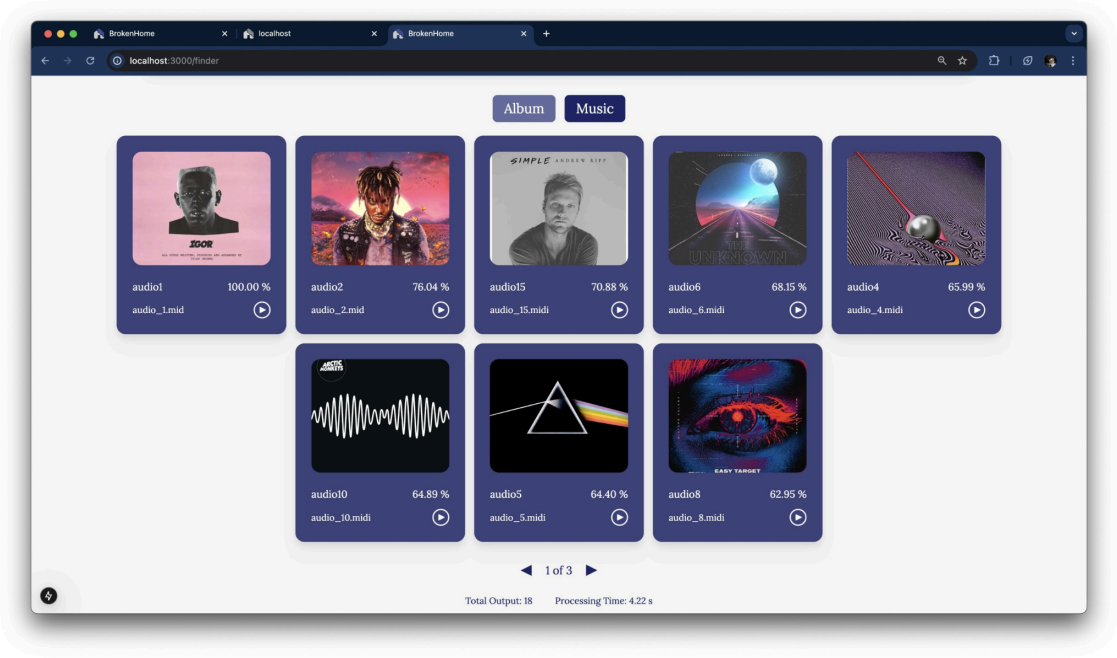
e. Mengupload Query Audio dan Memunculkan Hasil Pencarian

Upload:



Gambar 4.1.16 Memilih File Audio Query

Sesudah:



Gambar 4.1.17 Tangkapan Layar Hasil Pencarian Audio Query

BAB V
KESIMPULAN, SARAN, DAN REFLEKSI

5.1 Kesimpulan

Principal Component Analysis (PCA) dan representasi vektor merupakan pendekatan matematis yang sangat penting dan memiliki aplikasi luas dalam berbagai bidang, terutama dalam pengolahan data multidimensi seperti *Image Retrieval* dan *Music Information Retrieval*. PCA berfungsi sebagai teknik reduksi dimensi yang mampu menyederhanakan data dengan tetap mempertahankan informasi yang signifikan. Hal ini sangat relevan dalam mengatasi permasalahan data berdimensi tinggi, di mana kompleksitas data dapat diminimalkan tanpa kehilangan esensi utama.

Dalam *Image Retrieval*, PCA digunakan untuk mengekstraksi fitur utama dari gambar-gambar dalam dataset, sehingga memungkinkan pencarian gambar berdasarkan kemiripan fitur dengan efisiensi yang tinggi. Representasi ini dilakukan melalui proyeksi gambar ke ruang komponen utama, yang memungkinkan perbandingan antargambar dilakukan dengan metrik jarak seperti jarak Euclidean. Penerapan PCA pada *image retrieval* telah terbukti meningkatkan efisiensi dan akurasi dalam menemukan gambar yang relevan.

Di sisi lain, dalam *Music Information Retrieval*, PCA dan vektor digunakan untuk merepresentasikan file audio dalam ruang fitur yang lebih sederhana. Teknik ini memungkinkan analisis kemiripan antar file audio dengan lebih cepat, misalnya untuk menemukan musik yang mirip atau melakukan pengelompokan musik berdasarkan karakteristik tertentu. Dengan memanfaatkan teknik ini, analisis data audio menjadi lebih terstruktur dan efisien.

Secara keseluruhan, penerapan PCA dan vektor tidak hanya terbatas pada pengolahan citra atau data audio, tetapi juga meluas ke bidang lainnya seperti pembelajaran mesin, statistik, dan analisis data. Dengan kemampuan untuk mereduksi dimensi dan mengidentifikasi pola utama dalam data, PCA dan representasi vektor memainkan peran yang sangat penting dalam pengembangan teknologi berbasis data. Pemahaman yang mendalam tentang metode ini memberikan kemampuan untuk menyelesaikan berbagai permasalahan praktis dan membuka peluang untuk inovasi di berbagai bidang. Oleh karena itu, PCA dan vektor menjadi fondasi penting dalam pengolahan data modern.

5.2 Saran

Penulis menyarankan pemberi tugas untuk melakukan integrasi dan koordinasi yang lebih baik antara spesifikasi dan format laporan terutama pada bagian *Music Information Retrieval*. Pada spesifikasi, terdapat bagian “Music Information Retrieval - Query by Humming” sedangkan pada format laporan terdapat “Metode Ekstraksi Fitur Berdasarkan Humming” sebagai poin yang berbeda dari “Sistem Temu Balik Suara”, padahal ekstraksi merupakan salah satu tahapan dari *Query by Humming*, yang juga merupakan bagian dari *Music Information Retrieval*. Sebaiknya struktur disusun sebagai Sistem Temu Balik Suara - Query by Humming (menjadi satu poin).

Dengan ini, penulis dapat mengembangkan topik secara terstruktur mulai dari definisi MIR dan QBH, tahapan pemrosesan dengan menggunakan QBH (mencakup ekstraksi), dan dekomposisi ekstraksi menjadi beberapa metode.

5.3 Refleksi

Dalam proses penyelesaian tugas besar ini, kami sebagai tim mengalami berbagai dinamika, mulai dari tantangan teknis hingga perbedaan pengalaman individu. Setiap anggota tim memiliki pandangan dan pengalaman unik terhadap perannya masing-masing, yang memberikan pelajaran berharga bagi kami semua. Salah satu anggota tim merasakan "kejutan" yang tidak terduga saat harus bekerja dengan FastAPI. Awalnya, teknologi ini tampak menjanjikan dan sederhana, namun seiring perjalanan, muncul berbagai tantangan yang membuat proses pengembangannya jauh lebih kompleks dari perkiraan awal. Hal ini memberikan pelajaran penting bahwa memilih teknologi yang baru dikuasai untuk proyek besar membutuhkan kesiapan dan riset mendalam.

Anggota lain mengakui bahwa mereka sempat meremehkan tugas yang diberikan. Pada akhirnya, rasa penyesalan muncul ketika menghadapi kesulitan yang memakan waktu lebih lama dari yang diperkirakan. Pengalaman ini menjadi pengingat bahwa tanggung jawab apa pun, kecil atau besar, seharusnya dipersiapkan dengan baik dan dikerjakan dengan dedikasi penuh. Di sisi lain, ada anggota tim yang merasa "biasa saja" terhadap tanggung jawabnya. Ia menilai bahwa pekerjaan yang diberikan sesuai ekspektasi dan tantangan yang dihadapi tidak terlalu memberatkan. Hal ini menunjukkan pentingnya komunikasi dalam pembagian tugas agar beban kerja antar anggota lebih merata dan adil.

Terakhir, salah satu anggota tim merasa sangat beruntung karena mendapatkan pekerjaan yang relatif lebih mudah dibandingkan rekan-rekan lainnya. Namun, refleksi ini juga menjadi pengingat bahwa setiap anggota harus saling mendukung, sehingga keberuntungan yang dirasakan oleh satu anggota dapat diimbangi dengan kontribusi tambahan untuk membantu anggota lain yang mengalami kesulitan.

Dari pengalaman ini, kami menyadari bahwa pembagian tugas yang lebih seimbang, persiapan yang matang, serta komunikasi yang lebih baik adalah kunci keberhasilan dalam proyek tim. Selain itu, kami belajar untuk lebih menghargai setiap tantangan, tidak meremehkan pekerjaan, dan senantiasa mendukung satu sama lain dalam menghadapi kesulitan.

DAFTAR PUSTAKA

Birmingham, W., Pardo, B., (2002). *Query by Humming : How Good Can It Get?*. ISMIN Proceedings. Paris, France.

Downie, J. (2003). *Music Information Retrieval*. Annual Review of Information Science and Technology. Medford, NJ, USA.

Dunteman, H. George ,(1989). "*Principal Component Analysis"*. Sage Publications., Newbury Park London New Delhi. (Research Triangle Institute).

Hafiane, A., & Atif, J. (2008). Content-Based Image Retrieval Using PCA and Spatial Information. *Proceedings of the International Conference on Computer Vision and Graphics 5337:* 238-246.

Kosugi, N., Nishihara, Y., Sakata, T. (2000). *A Practical Query-By-Humming System For a Large Music Database*. ACM Multimedia

Jolliffe, I.T., & Cadima, J. (2016). Principal Component Analysis: A Review and Recent Developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374(2065).*

Tabachnick B 7 Fidell L.S (2001). "*Using Multivariate Statistics,4rd ed"*. Boston: Allyn & Bacon.

LAMPIRAN

a. Repository : https://github.com/0xNathaniel/Algeo02-23013

b. Video : https://youtu.be/L70nuod6fIk