

TUGAS BESAR 2
IF2123 ALJABAR LINIER DAN GEOMETRI
Image Retrieval dan Music Information Retrieval Menggunakan PCA dan Vektor
Semester I Tahun 2024/2025



Dipersiapkan oleh:
Aloisius Adrian Stevan Gunawan / 13523054
Albertus Christian Poandy / 13523077
Michael Alexander Angkawijaya / 13523102

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132
2024

DAFTAR ISI

BAB I.....	3
BAB II.....	4
1. Sistem Temu Balik Suara (MIR).....	4
2. Metode ekstraksi fitur berdasarkan humming.....	4
3. Image Retrieval menggunakan PCA.....	6
BAB III.....	8
A. Arsitektur Website (Frontend).....	8
B. Arsitektur Program Information Retrieval (Backend).....	8
BAB IV.....	10
Gambar 4.2.1 Hasil query image.....	11
BAB V.....	12
A. Kesimpulan.....	12
B. Saran.....	12
C. Komentar.....	12
D. Refleksi.....	12
LAMPIRAN.....	13

BAB I

DESKRIPSI MASALAH

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. 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.

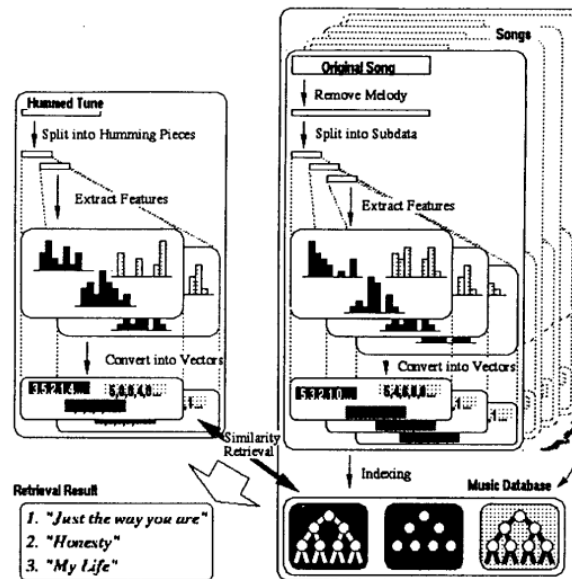
BAB II TEORI SINGKAT

1. Sistem Temu Balik Suara (MIR)

Music Information Retrieval (MIR) adalah bidang yang berfokus pada analisis, pemrosesan, dan pencarian informasi dari suatu data audio, terutama musik dan suara. MIR juga digunakan oleh berbagai aplikasi pencarian musik, identifikasi lagu, analisis genre, pengenalan pola suara, maupun tuning gitar.

2. Metode ekstraksi fitur berdasarkan *humming*

Sederhananya, metode ekstraksi fitur berdasarkan *humming* adalah proses dalam mencari lagu-lagu yang memiliki bagian yang mirip dengan nada *humming*, lalu memberi peringkat lagu-lagu tersebut sesuai dengan kedekatan kecocokannya.



Gambar 2.2 Membangun basis data musik, memproses alunan lagu, dan mengambil lagu-lagu serupa.

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 mengekstraksi seluruh track melodi selain dari instrumen *drum*. Setiap file MIDI diproses menggunakan metode windowing yang membagi melodi menjadi segmen 40 beat dengan sliding window 8 beat. Teknik ini memungkinkan pencocokan fleksibel dari berbagai bagian lagu yang mungkin diingat pengguna. Setiap note event dikonversi menjadi representasi numerik yang mempertimbangkan durasi dan urutan nada, memungkinkan sistem membandingkan potongan melodi dengan database.

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. 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. Pembobotan yang diimplementasikan adalah 20% ATB, 60% RTB, dan 20% FTB.

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 3. Cosine Similarity Formula

3. Image Retrieval menggunakan PCA

Image retrieval adalah proses dalam mencari gambar dalam sebuah dataset yang relevan dengan gambar query. Dalam hal ini, Principal Component Analysis (PCA) berperan sebagai teknik reduksi dimensi yang dapat digunakan untuk meningkatkan efisiensi fungsi. Berikut adalah teori dasar PCA dalam image retrieval:

1) Representasi Gambar

Setiap gambar dapat direpresentasikan sebagai sebuah vektor dengan fitur berdimensi tinggi. Fitur ini bisa bermacam-macam mulai dari Warna, Tekstur,

Bentuk, maupun tingkat kecerahan. Namun, pada tugas kali ini, yang perlu diperhatikan adalah *grayscale*, yang artinya adalah warna hitam-putih dari sebuah gambar. Namun, vektor berisi fitur berdimensi tinggi ini bisa menjadi masalah karena membutuhkan komputasi yang banyak, dan terkadang juga terdapat informasi yang tidak diperlukan.

2) Peran PCA

Tujuan dari PCA adalah untuk mengurangi dimensi dari data, tanpa menghilangkan informasi-informasi yang penting. Langkah-langkah PCA adalah:

- a. Normalisasi data: menstandarisasikan skala fitur pada tiap gambar
- b. Menghitung matriks kovarian: menghitung hubungan antar fitur
- c. Dekomposisi eigen: mencari eigenvector dan eigenvalue dari matriks kovarian
- d. Proyeksi ke komponen utama: memproyeksikan data ke beberapa komponen dengan eigenvalue terbesar

3) Proses Penggunaan PCA

Setelah menggunakan PCA untuk mereduksi dimensi, maka dapat dilakukan pencarian gambar yang memiliki fitur paling mendekati terhadap gambar yang ingin dicari kemiripannya. Hal ini dapat dilakukan dengan berbagai macam cara, salah satunya adalah dengan menggunakan metode *euclidean distance*. Hal ini dilakukan dengan membandingkan hasil proyeksi tiap gambar dari sebuah dataset terhadap gambar yang dicari.

4) Kelebihan dan Kekurangan PCA

PCA memiliki beberapa kelebihan, diantaranya adalah:

- a. Dimensi tereduksi: proses PCA dapat mengurangi dimensi gambar sehingga membuat proses lebih efisien
- b. Generalisasi: menghapus informasi yang tidak penting dari dimensi fitur gambar

Namun, PCA juga memiliki batasan, diantaranya adalah:

- a. Linearitas: PCA hanya bekerja untuk data yang memiliki hubungan linear
- b. Skalabilitas: Semakin besar dataset yang ingin diproses, proses PCA juga akan menjadi semakin lambat
- c. Jumlah komponen: Menentukan jumlah komponen yang tepat untuk mencari keseimbangan antara akurasi dengan efisiensi

BAB III ARSITEKTUR PROGRAM

A. Arsitektur Website (Frontend)

```
| -- app/
|   | -- album/
|   |   | -- [name]/
|   |   |   | -- page.tsx
|   |   |
|   |   | -- page.tsx
|   |
|   | -- search/
|   |   | -- result/
|   |   |   | -- page.tsx
|   |   |
|   |   | -- page.tsx
|   |
|   | -- song/
|   |   | -- page.tsx
|   |
|   | -- upload/
|   |   | -- page.tsx
|   |
|   | -- layout.tsx
|   | -- page.tsx
```

B. Arsitektur Program Information Retrieval (Backend)

```
| -- audio/
|   | -- audioprocessing.py
|
| -- image/
|   | -- image_processing.py
|
| -- uploads/
|   | -- dataset/
|   |   | -- image1.jpg
|   |   | -- image2.jpg
|   |   | -- . . .
```

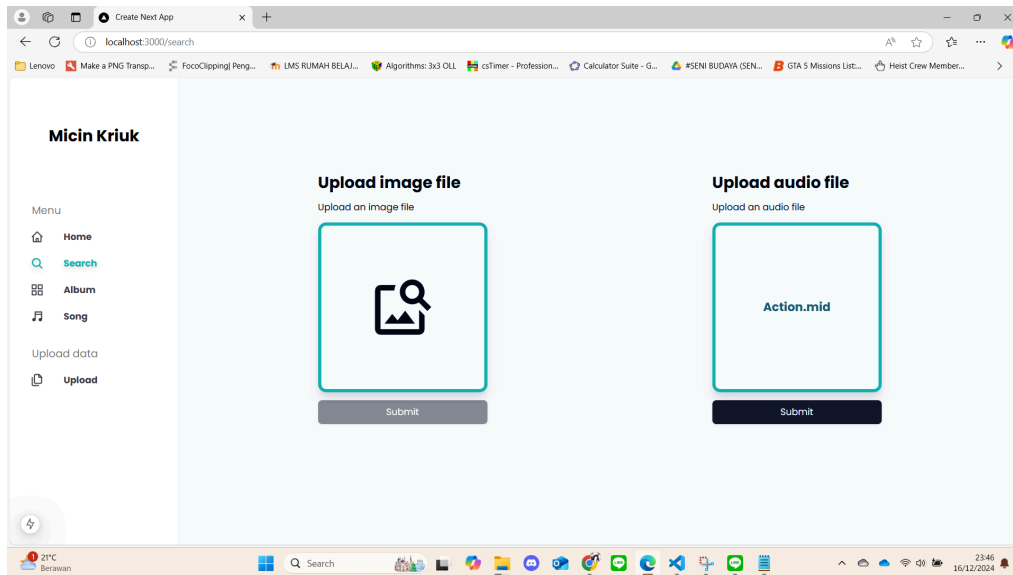


```
|      | -- audio1.mid  
|      | -- audio2.mid  
|      | -- . . . .  
|      |  
|      | -- mapper/  
|      | | -- mapper.txt  
|      |  
|      | -- query/  
|      | | -- audio_image.jpg  
|      | | -- audio_query.mid  
|      |  
|      | -- query_result/  
|      | | -- result.txt  
|  
|-- main.py  
|-- tools.py
```

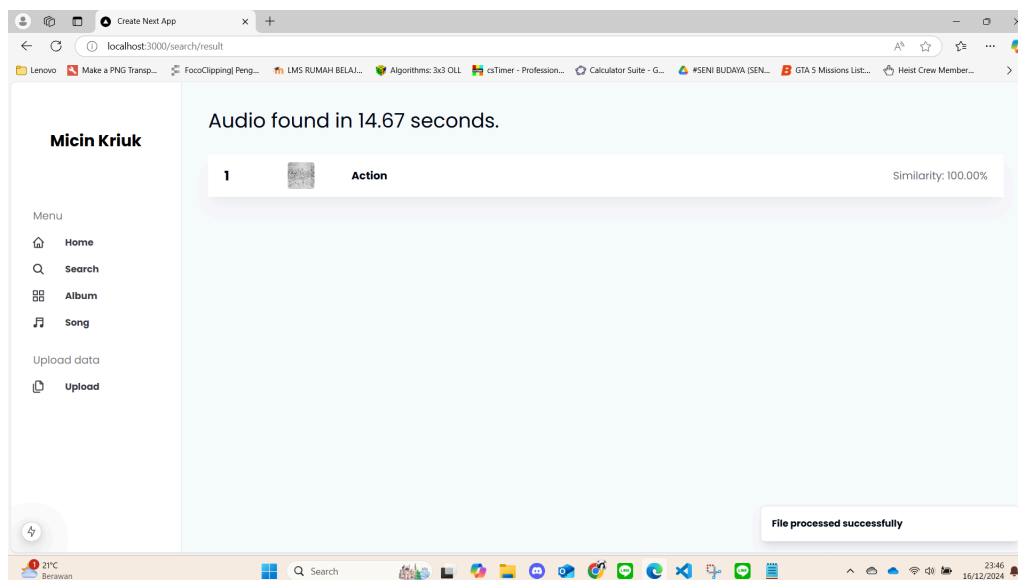
BAB IV EKSPERIMEN

1. Query Audio

Audio yang di query adalah audio lagu random bernama Action, dengan dataset sebanyak 100 lagu. Hasil memberikan tingkat kemiripan 100 persen.



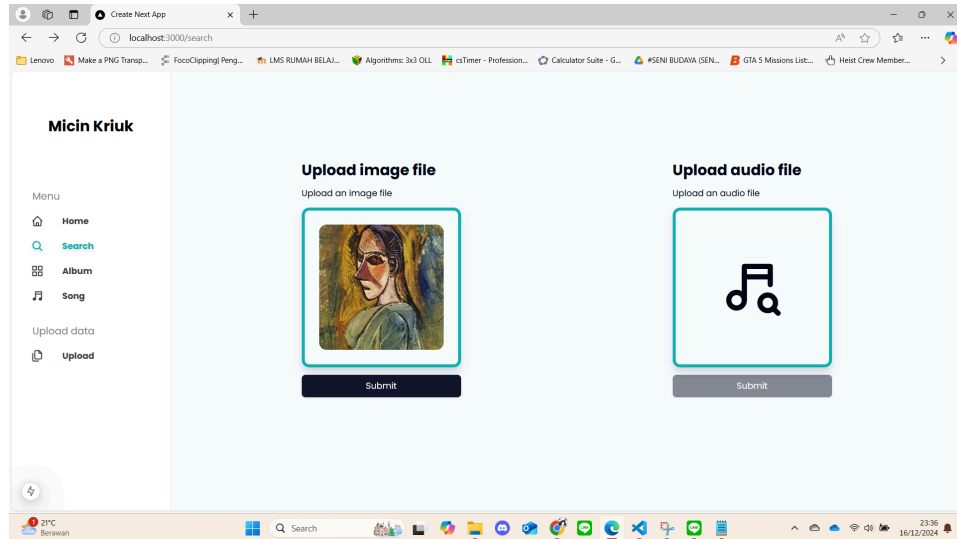
Gambar 4.2.1 Upload file query audio



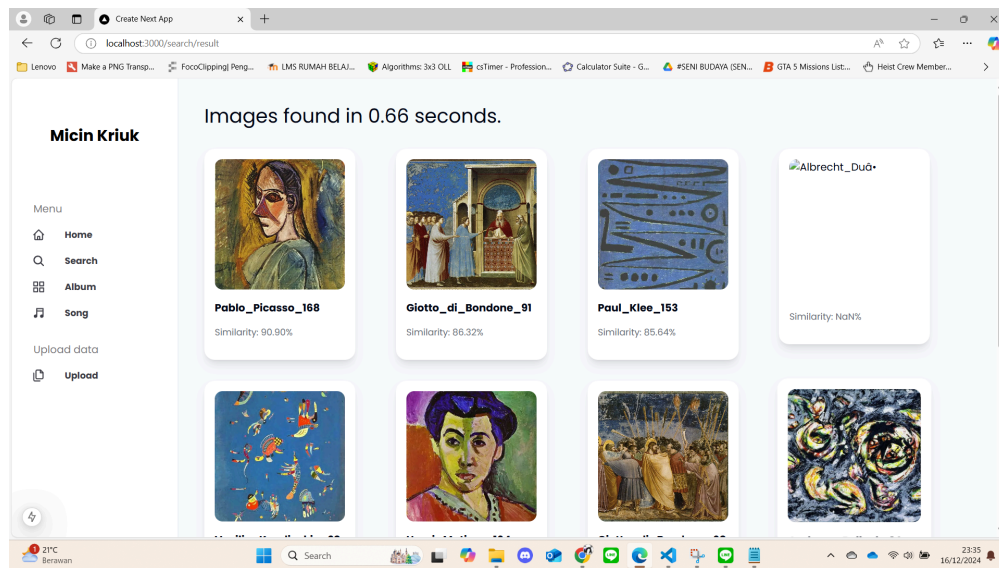
Gambar 4.2.1 Hasil query audio

2. Query Image

Image yang di query adalah crop ss dari image Pablo_Picasso_168, seperti yang terlihat pada gambar di bawah ini. Hasil memberikan tingkat kemiripan yang sesuai dengan ekspektasi , dengan tingkat kemiripan 90.90 persen dengan image asli Pablo_Picasso_168.



Gambar 4.2.3 Upload file query image



Gambar 4.2.4 Hasil query image

BAB V

KESIMPULAN

A. Kesimpulan

Mencari gambar dengan menggunakan gambar memang bisa menyita waktu berjam-jam. Maka dari itu, digunakanlah PCA untuk mempersingkat waktu dengan menyimpan fitur-fitur penting dari kumpulan gambar, sehingga pencarian bisa lebih cepat. Namun, PCA juga menghilangkan beberapa fitur, penting atau tidak. Sehingga, diperlukan keseimbangan antara efisiensi dan kualitas.

Mencari gambar dengan senandung akan memakan waktu yang sangat lama, sehingga diperlukan berbagai macam optimasi untuk mempercepat proses. Walaupun telah dilakukan optimisasi, waktu yang diperlukan untuk memproses semua datanya akan tetap memakan waktu yang sangat lama. Hal ini diakibatkan oleh sensitifnya sebuah lagu, yang mana sedikit gangguan suara bisa memberikan hasil yang berbeda.

Berdasarkan hasil dari mengerjakan tugas yang telah diberikan, dapat ditarik kesimpulan bahwa membuat kode untuk membuat perhitungan dapat menjadi sangat lama secara eksponensial. Maka dari itu, sangat diperlukan metode optimisasi agar bisa dilakukan pencarian gambar maupun audio dengan waktu yang dapat dimaklumi.

B. Saran

Dalam melakukan sebuah pencarian atau query, sangat disarankan untuk tidak melakukan proses yang dilakukan berulang kali pada data yang sama. Hasil yang telah diraih sebaiknya langsung disimpan agar bisa digunakan lagi sehingga bisa meningkatkan efisiensi dari kode.

C. Komentar

Laki laki sejati tidak berkomentar.

D. Refleksi

Dari pengerjaan, dapat ditarik kesimpulan bahwa kerja h-2 tidak efektif. ***Terbukti.***

LAMPIRAN

Link repository: <https://github.com/albertchriss/Algeo02-23054.git>

Link video:

<https://drive.google.com/drive/folders/1OOcThH0DgBZfxz0z1ku7lksXA9rCZ6IY?usp=sharing>

Referensi:

“Music Retrieval by Humming”

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