

**Tugas Besar 2 IF2123 Aljabar Linier dan Geometri**  
**Image Retrieval dan Music Information Retrieval Menggunakan PCA dan Vektor**  
**Semester I Tahun 2024/2025**

---

**I. 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.** 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).

## II. Information Retrieval

*Information Retrieval* adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Pada tugas besar ini, anda akan berkulit 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.

### 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. 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  $M \cdot N$ :

$$I = [I_1, I_2, \dots, I_{M \cdot N}]$$

### 2. Data Centering (Standardization)

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

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

di mana:

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

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

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

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

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

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

di mana:

- $X'$ : matriks data yang sudah distandarisasi.

Lalu lakukan dekomposisi nilai singular untuk mendapatkan komponen utama:

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

- $U$ : matriks eigenvector (komponen utama),
- $\Sigma$ : 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:

$$Z = X' U_k$$

di mana:

- $U_k$ : matriks eigenvector dengan  $n$ -dimensi.

### 4. Similarity Computation

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

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

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

Dimana:

- $\mathbf{q}$  = Vektor proyeksi dari gambar query ke ruang komponen utama (PCA).
- $\mathbf{q}'$ : Gambar query dalam format vektor (setelah grayscale, resize, dan flattening).
- $\mu$ : Rata-rata piksel dari dataset (per piksel).
- $\mathbf{U}_k$ : Matriks eigenvector dengan  $k$  dimensi utama dari PCA.

Kemudian, hitung jarak Euclidean antara gambar query dengan semua gambar dalam dataset:

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

- $d(\mathbf{q}, \mathbf{z}_i)$  = Jarak antara gambar query  $\mathbf{q}$  dan gambar ke- $i$  dalam ruang komponen utama.
- $\mathbf{z}_i$  = Vektor proyeksi dari gambar ke- $i$  dalam dataset ke ruang komponen utama.
- $q_j$ : Elemen ke- $j$  dari vektor proyeksi query  $\mathbf{q}$ .
- $z_{ij}$ : Elemen ke- $j$  dari vektor proyeksi gambar ke- $i$ , yaitu  $\mathbf{z}_i$ .
- $k$ : Jumlah dimensi ruang komponen utama yang dipilih.

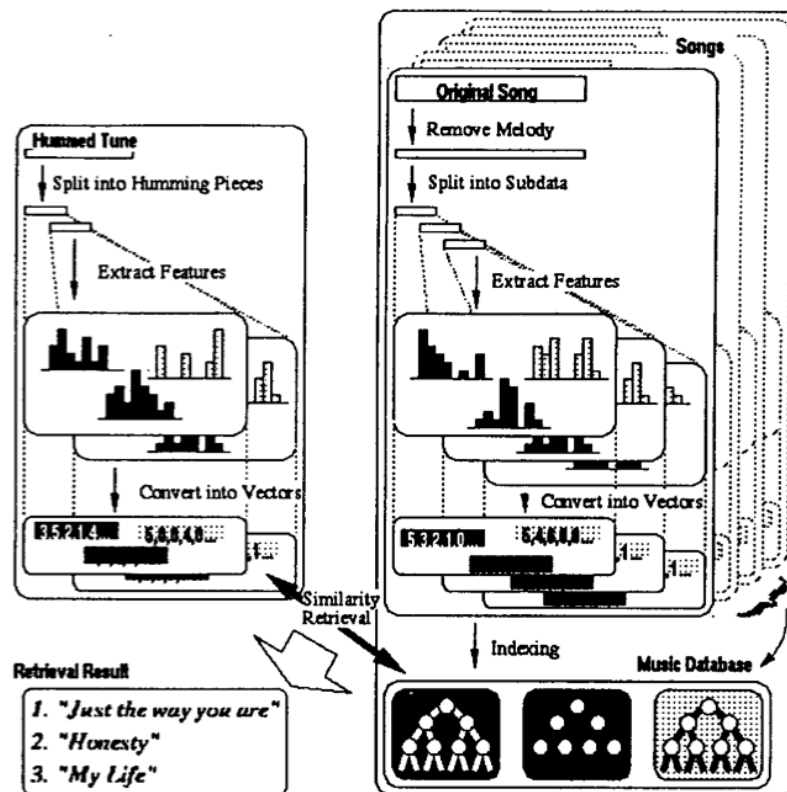
Lalu, Urutkan hasil berdasarkan jarak terkecil.

## 5. Retrieval and Output

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

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

## Music Information Retrieval – Query by Humming



**Gambar 2.** 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. 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  $\mu$  adalah rata rata dari pitch dan  $\sigma$  adalah standar deviasi dari pitch.

## 2. Ekstraksi Fitur

### 2.1. Distribusi tone

Distribusi tone diukur berdasarkan tiga viewpoints.

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

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

**Fitur First Tone Based (FTB)** fokus pada perbedaan antara setiap nada dengan nada pertama, menciptakan histogram yang



mencerminkan hubungan relatif terhadap titik referensi awal. Pendekatan ini membantu menangkap struktur relatif nada yang lebih stabil terhadap variasi pitch pengguna. Histogram dibuat dengan 255 bin, juga mencakup rentang nilai dari -127 hingga +127. Kemudian, hitung selisih antara setiap nada dalam data dengan nada pertama. Histogram yang dihasilkan kemudian dinormalisasi untuk menghasilkan distribusi yang seimbang.

## 2.2. Normalisasi

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

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

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

## 3. Penghitungan Similaritas

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

Berikut adalah formula dari cosine similarity :

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

**Gambar 3.** Cosine Similarity Formula

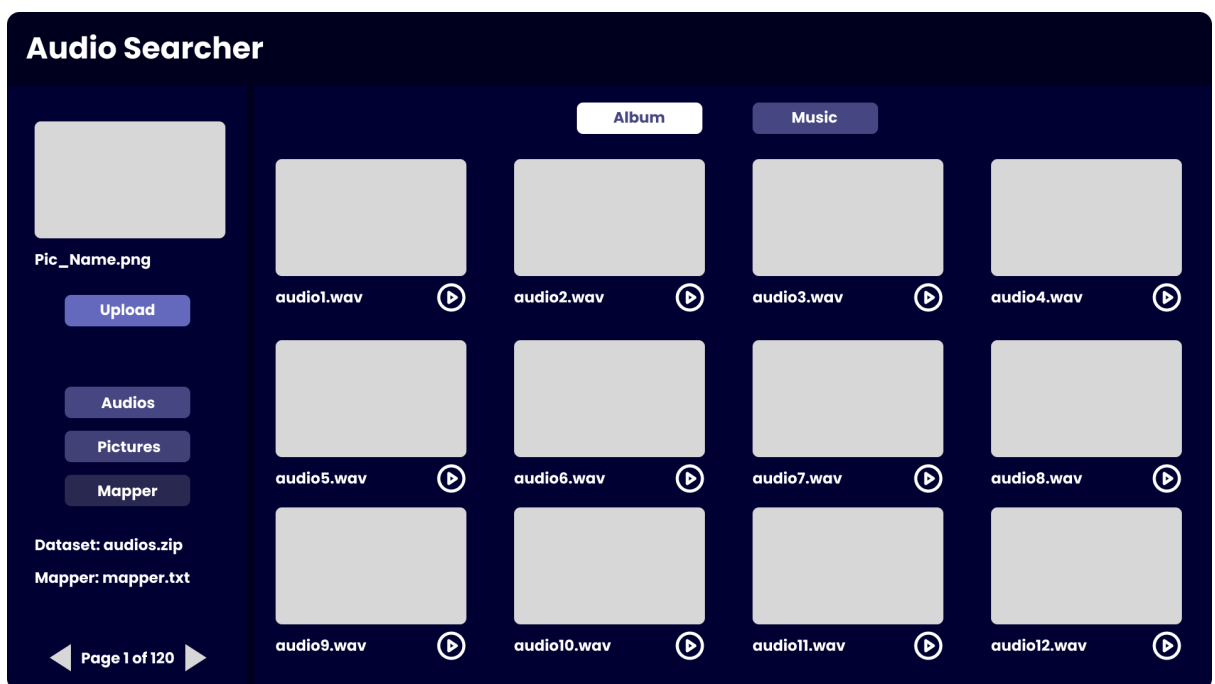
### III. Arsitektur Website

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*.
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 4.** Contoh tampak depan dari website Information Retrieval (Humming)



**Gambar 5.** 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 page 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.

### Bonus 1: Video Name Mapping

Tidak baik jika setiap video namanya langsung dari nama filenya. Oleh karena itu anda boleh membuat sebuah file .txt atau .json. Struktur dari file .txt atau .json tersebut adalah nama file video dan nama video dari file video yang bersangkutan tersebut. Dataset dari sebuah audio tidaklah sedikit. Oleh karena itu, carilah cara untuk memberi nama dari semua audio yang ada di dataset. Nama tersebut tidaklah harus nama lagu asli (Selama nama audio tersebut adalah sebuah kalimat bukan huruf saja seperti "A", "B", dll).

```
mapper.txt
-----

audio_file audio_name pic_name
audio_1.mid Billie Jean pic_1.png
audio_2.mid DJ Remix pic_2.png
audio_3.mid Gangsta Rap pic_3.png
audio_4.mid A -> (Tidak Boleh) pic_4.png
.
.
```

Nilai bonus bisa saja ditingkatkan jika anda ingin menambah detail dari sebuah audio lebih lanjut contohnya nama penyanyi, nama album, rating, dll. Silahkan gunakan imajinasi kalian untuk membuat sebuah detail audio yang menyemai aplikasi lagu seperti Spotify.

### Bonus 2: Performance Benchmark

Anda akan lebih dikenalkan terkait performa di mata kuliah IF2211 Strategi Algoritma (ini juga mata kuliah yang dipegang oleh Lab IRK). Namun sebelum mengambil mata kuliah tersebut, kami ingin membuat kalian setidaknya lebih *aware* terhadap seberapa cepat program anda dalam menyelesaikan tugasnya. Patokan dari bonus ini adalah 50 detik dengan dataset sebesar 1 GB.

### Bonus 3: Video

Silahkan berkreasi dalam penjelasan anda terkait MIR. Buatlah video penjelasan terkait program yang anda buat. Perlu diingat bahwa video bisa saja dilihat oleh dosen sehingga hindari adanya pelanggaran SARA dalam video anda.

### Bonus 4: Kreativitas Tambahan

Masih banyak kreasi yang bisa anda buat pada tugas besar ini. Pastikan kreasi terbaru yang anda buat bisa membuat asisten terpukau sehingga mereka mau mempertimbangkan untuk menambah nilai bonus kalian. Contoh bonus kreasi

yang bisa kalian buat adalah audio player yang bisa play/pause, rewind, fast-forward, dan mempercepat/memperlambat sebuah audio. Ini adalah salah satu contoh, anda boleh memikirkan bonus-bonus lainnya yang menurut kalian terkesan kreatif. Anda bahkan bisa membuat sebuah aplikasi konsep vektor dengan membuat jenis query yang baru seperti query by song name, dll (tidak harus secara audio dan gambar).

### **Bonus 5: Microphone**

1. Terdapat fitur microphone yang dapat melakukan penangkapan audio secara real-time ketika program dijalankan.
2. Dilarang menambahkan button untuk trigger pada fitur microphone. HINT: Gunakan interval waktu untuk melakukan penangkapan suara.
3. Fitur microphone merupakan fitur tambahan, sehingga fitur utama upload audio melalui website tetap harus ada.

### **Spesifikasi Tugas**

1. Buatlah sebuah website yang bisa mendeteksi dan mencari suara dan gambar dalam dataset yang paling cocok
2. Bahasa pemrograman yang digunakan pada frontend dan backend dibebaskan. Mohon dipertimbangkan bahasa-bahasa pemrograman yang kalian gunakan agar bahasa tersebut adalah yang menguntungkan kalian
3. Dataset dari sebuah audio bebas. Silahkan cari di website seperti [kaggle.com](https://www.kaggle.com). Untuk dataset yang akan diujikan, akan diumumkan nantinya.
4. Program dapat memunculkan audio dari dataset ketika sudah diunggah
5. Program dapat mencari kecocokan dari sebuah audio query atau tangkapan dari rekaman audio dengan audio yang ada di dataset berdasarkan metode query by humming
6. Untuk query by humming dapat menggunakan file berformat MIDI ataupun WAV. Silahkan pilih metode manapun yang dapat anda implementasikan.
7. Program dapat memberikan hasil audio / image yang memiliki tingkat kemiripan berkisar antara 55% - 100% (silahkan tentukan sesuai diskusi dari kelompok)
8. Program dapat mengimplementasikan *pagination* untuk menghindari adanya *infinite scrolling* ketika terdapat audio yang terlalu banyak
9. Program dapat menunjukkan jumlah audio yang didapat dari jumlah semua audio dan waktu eksekusi dari program tersebut

10. Bagian bonus hanya boleh dikerjakan apabila spesifikasi wajib dari Tugas Besar telah berhasil dipenuhi. Anda tidak diharuskan untuk mengerjakan keseluruhan bonus, tetapi semakin banyak bonus yang dikerjakan, maka akan semakin banyak tambahan nilai yang diperoleh.
11. Maksimum nilai dari tugas besar ini adalah 115. Walaupun nilai total kalian lebih tinggi karena bonus, nilai kalian akan tetap 115. Silahkan kerjakan bonus jika merasa ingin mengcover nilai spek wajib atau ingin menambah nilai saja.

### **Prosedur Pengerjaan**

1. Tugas dikerjakan secara berkelompok yang terdiri dari 3 orang. Kelompok dipilih secara mandiri dan anggota kelompok diperbolehkan lintas kelas maupun lintas kampus.
2. Daftarkan kelompok Anda dan via [tautan ini](#) sebelum hari Minggu, 24 November 2024 pukul 21.23 WIB. Peserta yang tidak terdaftar setelah batas waktu yang ditentukan akan dikelompokkan secara acak oleh tim asisten.
3. Tugas ini dikumpulkan hari Kamis, 12 Desember 2024 paling lambat pukul 23.59 WIB. Asisten akan mengumumkan jadwal demo program sesaat setelah pengumpulan selesai dilaksanakan.
4. Jika terdapat kesulitan selama mengerjakan tugas besar sehingga memerlukan bimbingan, maka dapat melakukan asistensi tugas besar kepada asisten. Oleh sebab itu, Anda juga perlu untuk melakukan pemilihan asisten melalui [tautan ini](#) sebelum hari Minggu, 12 Desember 2024 pukul 21.23 WIB.
5. Perlu untuk diketahui bahwa asisten yang dipilih juga akan menjadi asisten saat demo tugas besar, oleh sebab itu wajib untuk memilih asisten meskipun tidak melakukan asistensi.
6. Dilarang keras menyalin program dari sumber lain (buku, internet, program kakak tingkat, program kelompok lain).
7. Apabila ada pertanyaan yang ingin ditanyakan, diharapkan untuk bertanya melalui *google spreadsheet* [ini](#).

### **Laporan**

Laporan terdiri dari:

1. *Cover* : *Cover* laporan ada foto anggota kelompok (foto bertiga kalau ada, atau foto masing-masing, bebas gaya). Foto ini menggantikan logo "gajah" ganesha.
2. Bab 1 : Deskripsi masalah (dapat meng-copy paste file tugas ini).
3. Bab 2 : Teori singkat mengenai Sistem Temu Balik Suara (MIR), metode ekstraksi fitur berdasarkan humming, dan Image Retrieval dengan PCA

4. Bab 3 : Arsitektur website (frontend) dan arsitektur program Information Retrieval (backend)
5. Bab 4 : Eksperimen. Bab ini berisi hasil eksekusi program terhadap contoh-contoh kasus yang diberikan berikut analisis hasil eksekusi tersebut
6. Bab 5 : Kesimpulan, saran, komentar, dan refleksi (hasil yang dicapai, saran pengembangan, dan refleksi anda terhadap tugas ini).
7. Lampiran: Referensi, tautan *repository*, tautan video (jika ada).

Keterangan laporan dan program:

- a) Laporan ditulis dalam bahasa Indonesia yang baik dan benar, tidak perlu panjang tetapi tepat sasaran dan jelas.
- b) Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

### **Pengumpulan Tugas**

1. Program disimpan di dalam *repository github* dengan nama repository Algeo02-23XXX dengan XXX adalah tiga digit terakhir anggota dengan NIM terkecil.
2. Didalam *repository github* terdapat empat folder: bin, src, test dan doc yang masing-masing berisi:
  - Folder *src* berisi *source code* dari program anda
  - Folder *test* berisi data uji.
  - Folder *doc* berisi laporan
3. Jika file backend dan frontend terpisah, silahkan dibuat dalam folder src. Secara garis besar, struktur file src anda kemungkinan akan seperti ini
  - src
    - frontend
      - {Frontend files and folders}
    - backend
      - {Backend files and folders}
4. Pastikan menggunakan *semantic commit message* yang sesuai pada *repository* Anda. Anda dapat menggunakan [tautan ini](#) sebagai referensi.
5. Sertakan juga *README* yang dibuat sejelas mungkin. *README* minimal memuat penjelasan singkat program dan tata cara menjalankan program. Anda dapat menggunakan [template referensi](#) sebagai acuan.
6. Pastikan *repository* bersifat *private* dan telah mengundang asisten yang pembagiannya akan diumumkan kemudian.
7. Pastikan Website dapat dijalankan. Selain dijalankan saat demo, program akan dicoba dijalankan asisten diluar waktu demo untuk salah satu aspek penilaian.



Asisten pemeriksa tidak akan melakukan *setting* atau kompilasi lagi agar program dapat berjalan. Program yang tidak dapat dijalankan tidak akan diberi nilai.

8. Pengumpulan dilakukan melalui tautan berikut [Pengumpulan Tugas Besar II](#), cukup kumpulkan tautan rilis dan laporan
9. Gunakan *template* tag untuk rilis berupa "v.X"! X melambangkan berapa kali kelompok anda telah mengumpulkan tugas besar ini. Contohnya adalah jika hanya mengumpulkan sekali saja, tag rilisnya adalah v.1. Jika sudah melakukan revisi 2 kali, anda mengumpulkan tugas besar ini 3 kali sehingga tag rilisnya adalah v.3.
10. *Deadline* pengumpulan adalah Sabtu, 14 Desember 2024, pukul 23.59 WIB. Pengumpulan setelah waktu tersebut akan mendapat pengurangan nilai.
11. Segala bentuk kecurangan akan ditindaklanjuti sesuai dengan sanksi akademik yang ada.

## Penilaian

Komposisi penilaian umum adalah sebagai berikut :

1. Program : 80% + 15%
2. Laporan : 20%

## Referensi

"Music Retrieval by Humming"

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

"Audio Query by Example Using Similarity Measures Between Probability Measure Representations"

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

"Pengenalan wajah menggunakan metode principal component analysis (PCA)"

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

"Image Compression Techniques: A Closer Look at Principal Component Analysis"

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

## **Selamat Mengerjakan!**

"You cannot touch music, **But music can touch you!**"

**- Scifo -**

"Lama-lama lelah juga aku~"

**- Eka -**

"Baik, aku akan membuat kalimat penyemangat berdasarkan spesifikasi yang kamu berikan: Semangat, tugas ini adalah kesempatan untuk belajar dan mengasah keterampilan kalian! Dengan kerja keras, kerjasama tim, dan fokus, pasti kalian bisa menyelesaikannya dengan hasil terbaik!"

**- Ikhwan -**

"Fresh from the oven, chef!"

**- Ojan -**

"Keos"

**- Naufal -**

"Waduh, nambah tubes. Tenang ini belum seberapa."

**- Sutha -**