Laporan Tugas Besar 2

Image Retrieval dan Music Information Retrieval

Disusun untuk memenuhi tugas mata kuliah IF2123 Aljabar Linear dan Geometri
pada Semester 1 (satu) Tahun Akademik 2023/2024

Kelompok 21 (atur atur ajalah):

Razi Rachman Widyadhana	13523004
Guntara Hambali	13523114
Reza Ahmad Syarif	13523114

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

2024

Daftar Isi

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.

Dengan menggunakan konsep yang bernama Music Information Retrieval atau MIR, dapat dicari dan diidentifikasi suara berdasarkan fitur-fitur yang dimilikinya. Selan itu, dengan menggunakan konsep Principal Component Analysis (PCA), dapat dicari kumpulan audio melalui deteksi wajah berbagai orang (Dengan asumsi mereka sebagai seorang penyanyi).

Bab II
Teori Singkat

2.1. Information Retrieval

Information Retrieval adalah konsep meminta informasi dari sebuah data dengan memasukkan data tertentu. Terdapat 2 jenis Information Retrieval yang ada pada tugas besar in, yaitu 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.

2.1.1 Image Retrieval - Principal Component Analysis

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

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

a. Pemuatan dan Pemrosesan Gambar

Gambar diubah menjadi *grayscale* untuk mengurangi kompleksitas gambar dan membuat fokus menjadi bagian terang dan gelap gambar. Setiap gambar direpresentasikan dalam intensitas piksel saja tanpa informasi warna.

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

Kemudian, gambar diubah sehingga ukurannya sama untuk seluruh data gambar. Ukuran seluruh gambar harus konsisten untuk membuat perhitungan semakin akurat.

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

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

b. Pemusatan Data (Standardization)

Rata-rata dari setiap gambar untuk suatu piksel

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

dalam hal ini:

x_{ij}: nilai piksel ke-j pada gambar ke-i

N: jumlah total gambar dalam dataset

Standarisasi seluruh data piksel tiap gambar dikurangi dengan rata-rata yang telah dihitung

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

c. Komputasi PCA dengan Singlular Value Decomposition (SVD)

Matriks kovarians dari data yang sudah distandarsasi

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

dalam hal ini:

X': matriks data yang sudah distandarisasi

Dekomposisi nilai singular untuk mendapatkan kompoen utama

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

dalam hal ini:

U: matriks eigenvector (komponen utama)

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

Dari hasil SVD, diambil n jumlah komponen teratas. Pilih k-komponen utama teratas ($k \ll M \cdot N$) dan proyeksikan data:

$$Z = X' U_k$$

dalam hal ini:

U_k: matriks eigenvector dengan n-dimensi.

d. Perhitungan Similaritas

Gambar *query* direpresentasikan dalam ruang komponen utama dengan proyeksi yang sama

$$q = (q' - \mu)\ U_k$$

dalam hal ini:

q: Vektor proyeksi dari gambar *query* ke ruang komponen utama

q': Gambar *query* dalam format vektor (setelah processing)

μ: Rata-rata piksel dari *dataset* (per piksel)

U_k: matriks eigenvector dengan k dimensi utama dari PCA

Terakhir, diurut jarak terkecil dari jarak Euclidean antara gambar *query* dengan semua gambar dalam *dataset*

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

dalam hal ini:

$d(\boldsymbol{q}, \boldsymbol{z}_i)$: Jarak antara gambar *query* $\boldsymbol{q}$ dan gambar ke-i

$\boldsymbol{z}_i$: vektor proyeksi dari gambar ke-i dalam dataset ke ruang komponen utama

$\boldsymbol{q}_j$: Elemen ke-j dari vektor proyeksi *query* $\boldsymbol{q}$

z_{ij}: Elemen ke-j dari vektor proyeksi gambar ke-i, yaitu z

k: Jumlah dimensi ruang komponen utama yang dipilih $\boldsymbol{q}$

e. Temu-Balik and Keluaran

Gambar-gambar yang mirip dengan q*query* masukan yang telah dibatasi dengan memberikan batas jarak euclidean dikumpulkan dan ditampilkan.

2.1.2 Music Information Retrieval - Query by Humming

Langkah-langkah untuk melakukan pencarian audio menggunakan Query by Humming adalah sebagai berikut:

a. Pemrosesan Audio

Dengan menggunakan file MIDI yang berfokus 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*.

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*.

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

dalam hal ini:

μ: Rata-rata dari *pitch*.

σ: Standar deviasi dari *pitch*

b. Ekstraksi Fitur

Distribusi *tone* diukur berdasarkan tiga *viewpoints*:

- Absolute Tone Based (ATB)

Perhitungan frekuensi kemunculan setiap nada berdasarkan skala MIDI (0-127). Histogram yang dihasilkan memberikan gambaran distribusi absolut nada dalam data.

- Relative Tone Based (RTB)

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.

- First Tone Based (FTB)

Perbedaan antara setiap nada dengan nada pertama, menciptakan histogram yang mencerminkan hubungan relatif terhadap titik referensi awal. Pendekatan yang membantu menangkap struktur relatif nada yang lebih stabil terhadap variasi *pitch* pengguna. Histrogram dibuat dengan 255 bin dari selisih antara setiap nada dengan nada pertama, mencakup rentang nilai -127 hingga +127.

c. Normalisasi

Semua nilai dalam histogram dipastikan berada dalam skala probabilitas

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

dalam hal ini:

H: Histogram

d: Bin dari histogram

d. Perhitungan Similaritas

Setiap histogram diubah menjadi sebuah vektor dan dihitung 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.

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

2.1.3 Polyphonic/Monophonic Audio Converter to MIDI

Langkah-langkah untuk melakukan konversi Audio menjadi MIDI bergantung pada tipe audio yang ingin dikonversi:

a. Polyphonic Audio

Apabila audio yang ingin dikonversi bertipe *polyphonic*, diperlukan langkah awal, yaitu mengekstrak vocal/melodi utama dari audio tersebut untuk mendapatkan audio bertipe *monophonic*.

Disebabkan kompleksitas yang terlalu tinggi, digunakan model Pembelajaran Mesin dari Demucs. Demucs adalah model pemisahan sumber musik yang canggih, yang saat ini mampu memisahkan drum, bass, dan vokal dari iringan lainnya. Demucs didasarkan pada arsitektur konvolusi U-Net yang terinspirasi oleh Wave-U-Net.

b. Monophonic Audio

Apabila audio yang ingin dikonversi bertipe *monophonic*. atau audio *polyphonic* telah selesai diekstrak vokal/melodi utamanya, dilakukan *pitch detection* dengan menggunakan algoritma SWIPE.

SWIPE adalah algoritma pelacakan nada yang menganalisis spektrum sinyal menggunakan jendela berbentuk gigi gergaji. Algoritma ini sangat efektif dalam menangani struktur harmonik yang kompleks dan menawarkan ketahanan yang baik terhadap noise. Algoritma ini mencari komponen spektral yang terkait secara harmonis untuk memperkirakan frekuensi fundamental.

Dengan menambah *Median Filtering*, yaitu pembatasan rentang nada yang mungkin dari median, menghasilkan waktu proses yang cepat membuatnya unggul daripada model Pembelajaran Mesin yang langsung memprediksi menjadi MIDI.

2.2. Website

Website adalah halaman-halaman yang saling terkait dan dapat diakses dengan menggunakan peramban. Konten dalam *website* dapat melibatkan berbagai elemen, seperti teks informatif, grafik, formulir interaktif, dan *Website* yang lebih kompleks. Seiring dengan perkembangan teknologi, website menjadi medium dinamis yang memfasilitasi interaksi dua arah antara pengguna dan penyedia konten.

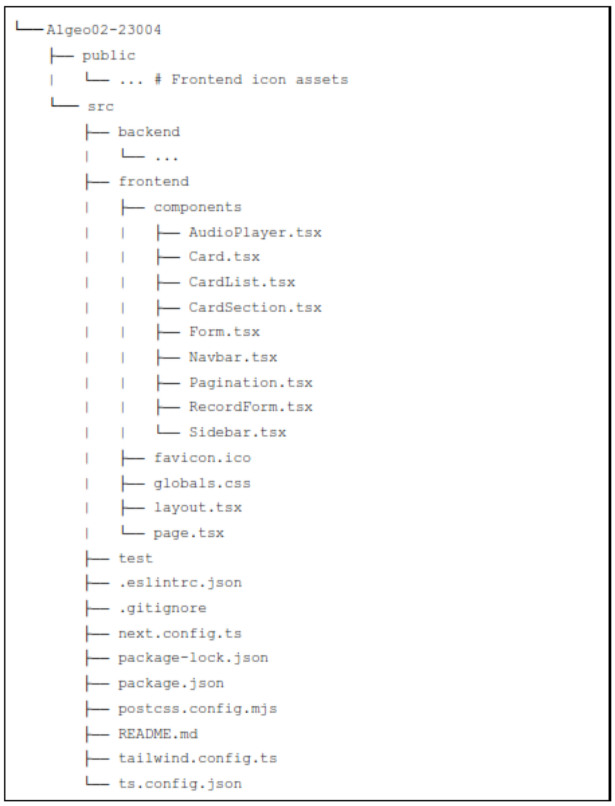
Bab III
Arsitektur Website

3.1. Arsitektur Tampilan (Frontend)

Kakas yang digunakan pada arsitektur *Frontend* adalah Next.js, TypeScript, dan Tailwind CSS. NextJS adalah framework berbasis React yang memudahkan pembuatan aplikasi *web* yang siap produksi dan SEO-*friendly* tanpa memerlukan banyak konfigurasi manual. Dengan fitur Server-Side Rendering (SSR) dan Static Site Generation (SSG) bawaan, Next.js mampu mengoptimalkan performa aplikasi dan SEO secara signifikan.

Framework ini juga menyediakan *file-based routing* yang intuitif, sehingga struktur aplikasi menjadi lebih terorganisir dan mudah dipelihara. Fitur *hot reloading* memungkinkan pengembang melihat perubahan secara *real-time* tanpa perlu me-*refresh*. TypeScript yang kuat, optimasi gambar otomatis, dan API *routes* yang memudahkan pembuatan endpoint sederhana tanpa perlu server terpisah. Tailwind CSS juga mempercepat proses *styling* dengan pendekatan *utility-first* yang memungkinkan pengembang membuat desain yang konsisten dan responsif tanpa meninggalkan HTML.

Berikut adalah struktur directory *client-side/frontend* dari **Melodia**

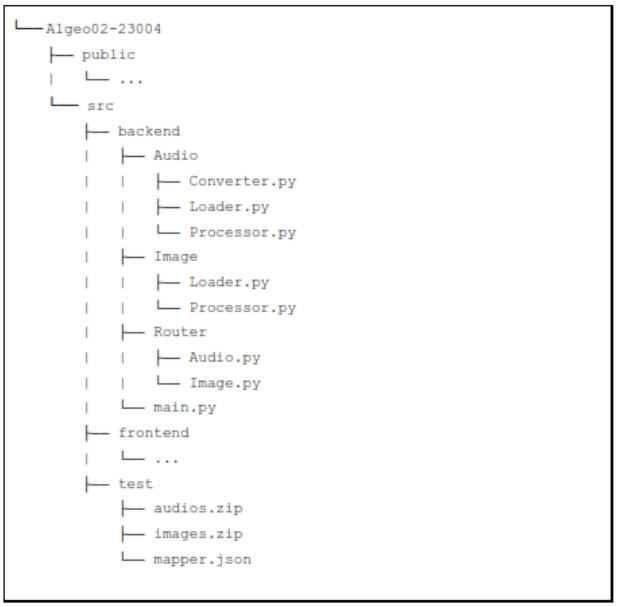


Gambar 2: Arsitektur Frontend Melodia

3.2. Arsitektur Program Information Retrieval (Backend)

Kakas yang digunakan pada arsitektur *Frontend* adalah FastAPI dari Python. FastAPI, dengan fondasi Python-nya, membuka akses ke ekosistem *library* komputasi dan audio yang sangat kuat. Untuk pemrosesan audio, tersedia *library* seperti librosa, pretty-midi, dan mido untuk analisis musik dan audio. Dalam hal komputasi, Python menyediakan *library-library Overpower* seperti NumPy dan SciPy untuk operasi array dan matriks yang dioptimalkan. Library-library ini dapat diintegrasikan dengan mulus ke dalam aplikasi FastAPI untuk membuat endpoint-endpoint yang menangani pemrosesan audio secara efisien.

Berikut adalah struktur directory *server-side/backend* dari **Melodia**



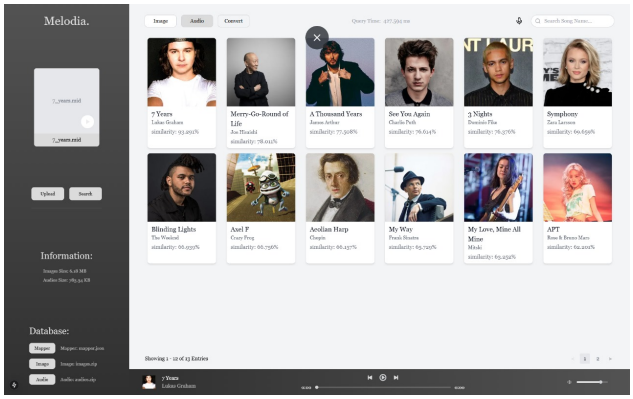
Gambar 3: Arsitektur Frontend Melodia

3.3. Interaksi Frontend-Backend

Dalam arsitektur aplikasi *web* modern, Next.js dan FastAPI berinteraksi melalui HTTP dengan format JSON. Frontend Next.js akan mengirimkan *request* ke endpoint-endpoint yang telah didefinisikan di FastAPI, biasanya menggunakan *fetch* API atau *library* seperti Axios. FastAPI akan memproses request tersebut, melakukan operasi yang diperlukan (seperti *query database* atau bisnis *logic*), kemudian mengembalikan response dalam format JSON yang akan di-*render* oleh Next.js. Untuk mengoptimalkan performa, Next.js dapat memanfaatkan fitur SSR untuk mengambil data dari FastAPI di sisi *server* sebelum halaman di-*render*, sementara untuk interaksi dinamis dapat menggunakan *client-side fetching*. Kedua framework ini dapat dikonfigurasi untuk menangani CORS (Cross-Origin Resource Sharing) yang memungkinkan komunikasi aman antar domain yang berbeda.

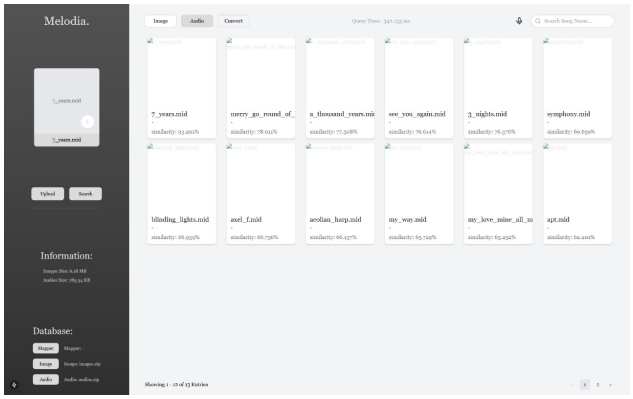
Bab IV
Eksperimen

4.1. Image Retrieval



Gambar 4: Eksperimen Query Images

4.2. Music Information Retrieval



Gambar 5: Eksperimen Query Audio

Bab V
Penutup

5.1. Kesimpulan

Dalam proyek Tugas Besar 2 IF2123 Aljabar Linier dan Geometri ini, kami berhasil mengimplementasikan aplikasi Aljabar Vektor untuk *Information Retrieval* pada *Image Retrieval* dan *Music Information Retrieval*. Program kami menggunakan metode pencarian gambar melalui PCA dan pencarian audio melalui fitur MIDI. Hasilnya, program mampu mencari query gambar melalui persentase Euclidean Distance dan mampu mencari query audio dengan memberikan persentase kemiripan yang dihitung menggunakan cosine similarity dan menampilkan hasilnya secara interaktif di web.

Fitur utama program mencakup upload mapper, upload dataset, upload gambar/audio query, penggunaan audio-player, penggunaan mikrofon dan Audio-To-MIDI *converter*. Program ini memberikan pengalaman pencarian gambar dan audio berbasis konten yang efisien dan informatif, menciptakan solusi kuat bagi pengguna yang mencari mencari audio dengan *humming* dan mencari gambar dengan kemiripan visual.

5.2. Saran

Pelaksanaan Tugas Besar 2 IF2123 Aljabar Linier dan Geometri di Semester I Tahun 2024/2025 merupakan pengalaman yang sangat berharga bagi kami. Dari pengalaman ini, kami ingin berbagi beberapa saran kepada pembaca yang mungkin akan menghadapi tugas serupa di masa depan:

Definisi .1: Razi

- Pemahaman Bahasa Pemrograman Berbasis Web:

Tugas ini menuntut penulisan pemrograman web, yang mungkin belum familiar bagi sebagian mahasiswa. Saya sangat menyarankan untuk meluangkan waktu yang cukup guna mempelajari keahlian ini dengan baik, terutama jika Anda belum memiliki pengalaman sebelumnya. Karena tugas ini melibatkan banyak fitur *frontend* dan juga *backend* yang perlu dikuasai dengan rinci dengan berbagai *passing variables* dan validasi.

Definisi .2: Guntara

- Kerja Sama Tim:

Efektivitas dalam kerja sama tim memiliki peran penting dalam menyelesaikan tugas ini. Kolaborasi secara real-time melalui alat seperti VSCode untuk pembuatan program dan kolaborasi langsung pada Google Docs untuk pembuatan laporan bisa sangat membantu. Selain itu, dalam pengembangan source code, konflik antara anggota tim dapat terjadi. Oleh karena itu, menggunakan alat pengelola versi seperti Github sangat direkomendasikan agar mempermudah manajemen proyek secara asinkron.

Definisi .3: Reza

- Pemahaman Fungsi dari Library yang Mempermudah:

Tugas ini mungkin melibatkan penggunaan library Python tertentu untuk menyelesaikan permasalahan tertentu. Penting untuk memahami dengan baik fungsi-fungsi yang disediakan oleh library tersebut agar dapat memanfaatkannya secara efektif dalam pengembangan program. Sumber daya online dan dokumentasi resmi library dapat menjadi panduan yang berguna dalam memahami cara menggunakan fitur-fitur tersebut.

Semoga saran-saran ini membantu pembaca dalam menyiapkan diri untuk menangani tugas serupa di masa depan.

5.3. Komentar

Komentar 1

Tingkat kesulitan yang naik signifikan dari tahun-tahun sebelumnya

Komentar 2

Inkonsistensi dalam konfirmasi lingkup library-library yang dapat digunakan

Komentar 3

Referensi untuk Music Information tidak begitu jelas dipergunakannya dan cenderung kurang lengkap dibanding Image Retrieval

5.4. Refleksi

Ruang perbaikan dan pengembangan dalam mengerjakan tugas dapat difokuskan pada beberapa aspek yang dapat ditingkatkan. Pertama-tama, pengaturan waktu menjadi kunci utama. Kami menyadari bahwa perencanaan waktu yang lebih baik dapat meningkatkan efisiensi pengerjaan. Menerapkan strategi manajemen waktu, seperti membuat jadwal yang terstruktur dan menetapkan tenggat waktu internal untuk setiap tahap pekerjaan, dapat membantu menghindari tekanan waktu yang tidak perlu.

Selain itu, ketelitian membaca spesifikasi dari awal menjadi aspek yang perlu diperhatikan. Memahami secara menyeluruh tentang apa yang diminta dalam tugas, termasuk detail-detail kecil, dapat mengurangi risiko kesalahan dan memastikan pekerjaan berjalan sesuai dengan harapan. Menempatkan perhatian ekstra pada spesifikasi tugas dapat meminimalkan revisi dan penyesuaian yang mungkin diperlukan.

Komunikasi tim yang baik juga dapat dioptimalkan. Membuat saluran komunikasi yang jelas dan terbuka di antara anggota tim dapat menghindari kebingungan dan memastikan bahwa semua anggota memiliki pemahaman yang sama tentang tujuan dan tanggung jawab masing-masing. Diskusi reguler dan pembahasan mengenai kemajuan proyek dapat meningkatkan keterlibatan semua anggota tim.

Dalam konteks pengembangan web, penambahan fitur-fitur yang mendukung dapat memberikan nilai tambah. Memperhatikan kebutuhan pengguna dan mengidentifikasi area yang dapat diperbaiki atau ditingkatkan dalam hal fungsionalitas dapat meningkatkan kualitas tugas. Pemikiran kreatif dalam mengimplementasikan fitur-fitur baru yang relevan dengan tujuan proyek dapat membuat proyek lebih bermanfaat dan menarik.

Dalam konteks pengolahan gambar, dapat melakukan eksplorasi lebih jauh lagi mengenai efisiensi kinerja program. Dengan memahami cara agar kinerja program lebih efisien, program pengolahan gambar akan menjadi lebih cepat dan lebih baik lagi. Hal tersebut sangat diperlukan agar program kami bisa menjadi program yang layak untuk dipublikasikan.

Terakhir, evaluasi diri secara berkala dapat menjadi langkah penting. Menerima umpan balik, baik dari anggota tim maupun asisten, dan memanfaatkannya sebagai dasar untuk perbaikan lebih lanjut adalah cara efektif untuk terus berkembang. Selalu terbuka terhadap saran dan berkomitmen untuk belajar dari setiap pengalaman dapat membantu memperbaiki kinerja secara berkelanjutan.

Lampiran

Rinaldi Munir 2024. "Aljabar dan Geometri untuk Informatika 2024/2025." informatika.stei.itb.ac.id/ rinaldi.munir/AljabarGeometri/2024-2025/algeo24-25.html

Disha Garg. 2007. "SWIPE' pitch estimator". University of Florida. Diakses pada 7 Desember 2024. https://github.com/dishagarg/SWIPE

François. 2024. "Record Audio in JS and upload as wav or mp3 file to your backend". Medium. Diakses 15 Desember 2024. https://franzeus.medium.com/record-audio-in-js-and

Tautan

Repository Release: https://github.com/zirachw/Algeo02-23004

Video: https://linktr.ee/Zirach