

# **LAPORAN TUGAS BESAR 2 IF2123 ALJABAR LINIER DAN GEOMETRI**

## **APLIKASI ALJABAR VEKTOR DALAM SISTEM TEMU BALIK GAMBAR**



**Dosen Pengampu : Dr. Ir. Rinaldi, M.T.  
Asisten Pembimbing : Mohammad Rifqi Farhansyah (13521166)**

**Disusun oleh :  
Kelompok 44 – S43W79N**

**Erdianti Wiga Putri Andini (13522053)  
Imanuel Sebastian Girsang (13522058)  
Salsabiila (13522062)**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2023**

## KATA PENGANTAR

Dalam kesempatan yang mulia ini, dengan penuh rasa syukur dan semangat, kami ingin menyampaikan sebuah laporan yang mengupas secara mendalam materi mata kuliah yang sangat penting dalam dunia matematika dan pemrograman, yaitu "Aljabar Linier dan Geometri."

Mata kuliah ini merupakan landasan utama dalam pemahaman konsep-konsep dasar Aljabar Linier dan Geometri yang memainkan peran sentral dalam berbagai cabang ilmu matematika hingga ilmu komputer. Dengan pemahaman yang kuat terhadap Aljabar Linier dan Geometri, kita dapat membuka pintu menuju berbagai pengetahuan yang lebih dalam dan pemecahan masalah yang lebih kompleks.

Laporan ini bertujuan untuk memberikan pandangan komprehensif tentang konsep-konsep dasar Aljabar Linier dan Geometri, serta bagaimana penerapannya dalam dunia nyata. Kami berharap makalah ini dapat memberikan wawasan yang bermanfaat dan mendalam bagi para pembaca, baik mereka yang baru mengenal mata kuliah ini maupun mereka yang ingin memperdalam pemahaman mereka.

Kami ingin mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan inspirasi dalam penyusunan makalah ini. Semoga laporan ini dapat menjadi panduan yang bermanfaat dalam perjalanan Anda dalam memahami dunia Aljabar Linier dan Geometri.

Akhir kata, kami berharap makalah ini dapat bermanfaat dan memotivasi pembaca untuk terus mengeksplorasi serta mendalami dunia yang menarik dari Aljabar Linier dan Geometri.

Bandung, 19 November 2023

Erdianti Wiga Putri Andini	(13522053)
Immanuel Sebastian Girsang	(13522058)
Salsabiila	(13522062)

## DAFTAR PEMBAGIAN TUGAS

	KEGIATAN	PIC
Program	CBIR dengan Parameter Warna	13522053
	CBIR dengan Parameter Tekstur	13522062
	Utility Program	13522053 13522062
	Read image	13522053
	Read dataset	13522053
	Caching	13522058
	Web Scraping	13522058
	Saving to PDF	13522058
	Camera	13522058
Website	Parallel Processing	13522058
	Main page	13522058
	About us page	13522058
	How to use page	13522053 13522058 13522062
	Deskripsi program	13522053 13522062
	About project page (konsep singkat search engine)	13522053 13522058
	Connect front-end dan back-end	13522058
	Camera page	13522058

Laporan	Kata Pengantar dan Daftar Isi	13522053
	Bab 1 (Deskripsi Masalah)	13522053
	Bab 2 (Landasan Teori)	13522053 13522062
	Bab 3 (Analisis Pemecahan Masalah)	13522053 13522058 13522062
	Bab 4 (Implementasi dan Uji Coba)	13522053 13522058 13522062
	Bab 5 (Kesimpulan dan Saran)	13522053 13522062
	Daftar Pustaka	13522053 13522062
	Lampiran Catatan Asistensi	13522053 13522058 13522062
	Dokumentasi	13522053 13522058 13522062
	Take Video	13522053 13522058 13522062
Video	Editing Video	13522053 13522062

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>DAFTAR PEMBAGIAN TUGAS.....</b>	<b>3</b>
<b>DAFTAR ISI.....</b>	<b>5</b>
<b>BAB I.....</b>	<b>7</b>
1.1. Abstraksi.....	7
1.2. Content-Based Information Retrieval (CBIR).....	8
<b>BAB II.....</b>	<b>9</b>
2.1. CBIR dengan Parameter Warna.....	9
2.2. CBIR dengan Parameter Tekstur.....	12
2.3. Pengembangan Website.....	14
2.3.1. Pengembangan Web Front-End.....	15
2.3.2. Pengembangan Web Back-End.....	16
2.3.3. Pengembangan Web Full-Stack.....	17
2.4. Programming Language.....	17
2.4.1. Python.....	17
2.4.2. NextJS.....	18
2.4.3. Flask API.....	19
<b>BAB III.....</b>	<b>21</b>
3.1. Citra.....	21
3.2. Analisis Citra Menggunakan Strategi Warna.....	21
3.3. Analisis Citra Menggunakan Strategi Tekstur.....	23
3.4. Pemrosesan Program Berbasis Website.....	26
<b>BAB IV.....</b>	<b>28</b>
4.1. Implementasi Fungsi Program.....	28

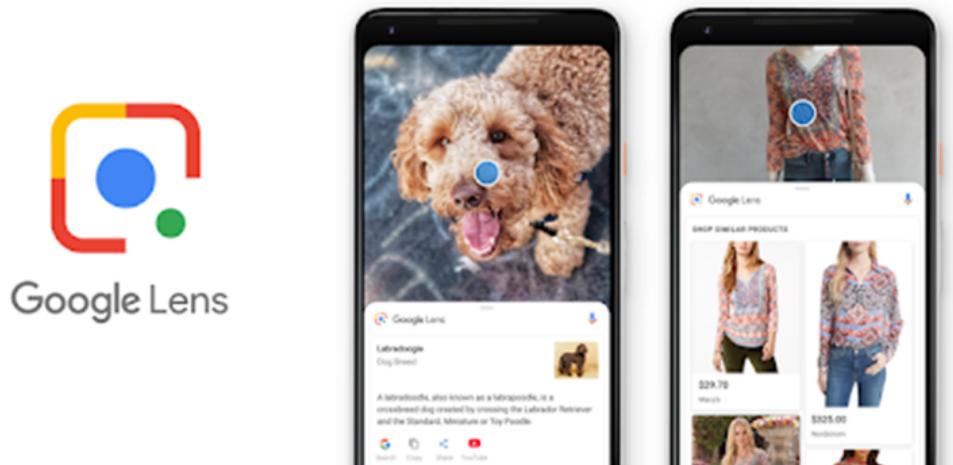
4.1.1. Utility Program (util.py).....	28
4.1.2. CBIR dengan Parameter Warna (colorbased.py).....	29
4.1.3. CBIR dengan Parameter Tekstur (texturebased.py).....	31
4.2. Struktur Website.....	33
4.2.1. Halaman Utama (Home).....	33
4.2.2. Halaman How to Use.....	33
4.2.3. Halaman About This Project.....	33
4.2.4. Halaman About Us.....	34
4.2.5. Halaman Kamera.....	34
4.2.6. Fitur Web Scraping.....	34
4.3. Tata Cara Penggunaan Program.....	34
4.4. Hasil Pengujian.....	35
4.5. Analisis Solusi Algoritma.....	44
<b>BAB V.....</b>	<b>45</b>
5.1. Kesimpulan.....	45
5.2. Saran.....	45
5.3. Komentar dan Tanggapan.....	45
5.4. Refleksi terhadap Tugas Besar.....	46
5.5. Ruang Perbaikan atau Pengembangan.....	46
<b>DAFTAR PUSTAKA.....</b>	<b>47</b>
<b>LAST RELEASE REPOSITORY.....</b>	<b>50</b>
<b>YOUTUBE VIDEO.....</b>	<b>51</b>
<b>LAMPIRAN CATATAN ASISTENSI.....</b>	<b>53</b>
<b>DOKUMENTASI SELAMA TIMELINE TUBES.....</b>	<b>54</b>

## BAB I

### DESKRIPSI MASALAH

#### 1.1. Abstraksi

Dalam era digital, jumlah gambar yang dihasilkan dan disimpan meningkat pesat, baik untuk keperluan pribadi maupun profesional. Peningkatan ini mencakup berbagai jenis gambar, mulai dari foto pribadi, gambar medis, ilustrasi ilmiah, hingga gambar komersial. Meskipun sumber dan jenis gambar bervariasi, sistem temu balik gambar (*image retrieval system*) menjadi sangat relevan dan penting untuk mengatasi tantangan ini. Dengan sistem temu balik gambar (*image retrieval system*), pengguna dapat dengan mudah mencari, mengakses, dan mengelola koleksi gambar mereka. Sistem ini memungkinkan pengguna untuk mengeksplorasi informasi visual di berbagai platform, termasuk pencarian gambar pribadi, analisis gambar medis untuk diagnosis, pencarian ilustrasi ilmiah, hingga pencarian produk berdasarkan gambar komersial. Salah satu contoh aplikasi sistem temu balik gambar yang mungkin dikenal adalah Google Lens.



**Gambar 1.** Contoh Penerapan *Information Retrieval System* (Google Lens)

Dalam Tugas Besar 2 ini, kami diminta untuk mengimplementasikan sistem temu balik gambar yang telah dijelaskan sebelumnya dengan menggunakan Aljabar Vektor

dalam bentuk sebuah situs web. Pendekatan ini sangat penting dalam pemrosesan data dan pencarian informasi. Dalam konteks ini, aljabar vektor digunakan untuk menggambarkan dan menganalisis data melalui pendekatan klasifikasi berbasis konten (*Content-Based Image Retrieval* atau CBIR), di mana sistem temu balik gambar bekerja dengan mengidentifikasi gambar berdasarkan konten visual seperti warna dan tekstur.

## **1.2. Content-Based Information Retrieval (CBIR)**

Content Based Image Retrieval System (CBIR) merupakan proses yang digunakan untuk mencari dan mengambil gambar berdasarkan kontennya. Metode pencarian citra ini dengan melakukan perbandingan antara citra query dengan citra yang ada di database berdasarkan informasi yang ada pada citra tersebut (Query by Example). Metode CBIR yang sering digunakan adalah pencarian berdasarkan kemiripan warna, bentuk, dan tekstur. Setelah fitur-fitur tersebut diekstraksi, mereka diwakili dalam bentuk vektor atau deskripsi numerik yang dapat dibandingkan dengan gambar lain. Kemudian, CBIR menggunakan algoritma pencocokan untuk membandingkan vektor-fitur dari gambar yang dicari dengan vektor-fitur gambar dalam dataset. Hasil dari pencocokan ini digunakan untuk mengurutkan gambar-gambar dalam dataset dan menampilkan gambar yang paling mirip dengan gambar yang dicari. Proses CBIR membantu pengguna dalam mengakses dan mengeksplorasi koleksi gambar dengan cara yang lebih efisien, karena tidak memerlukan pencarian berdasarkan teks atau kata kunci, melainkan berdasarkan kesamaan nilai citra visual antara gambar-gambar tersebut. Beberapa pokok persoalan yang terjadi dalam pembangunan CBIR dapat diuraikan menjadi 3, yaitu:

1. Pemahaman image yang dibutuhkan oleh user dan pencarian informasi.
2. Pengidentifikasi cara yang sesuai dalam image content atau karakteristik dari image.
3. Proses ekstraksi fitur dari image.

Implementasi yang digunakan pada Tugas Besar 2 IF2123 kali ini adalah CBIR dengan parameter warna dan parameter tekstur. Penjelasan lebih lanjut tertera pada bab selanjutnya.

## BAB II

### LANDASAN TEORI

#### 2.1. CBIR dengan Parameter Warna

Fitur yang paling penting di ekstrak dari suatu citra yaitu warna. Warna adalah properti citra hasil tergantung pada refleksi cahaya ke mata dan pengolahan informasi di otak. Citra warna sehari-hari dapat dibedakan antara obyek, tempat dan waktu pengambilan citra. Biasanya warna dapat didefinisikan dalam tiga dimensi ruang warna yaitu RGB (Red, Green, dan Blue), atau HSV (Hue, Saturation, dan Value), atau HSB (Hue, Saturation, Brightness). Format citra yang banyak digunakan yaitu JPEG, BMP, JPG, GIF, dengan menggunakan ruang warna RGB untuk menyimpan informasi. Ruang warna RGB didefinisikan dengan bentuk kubus dengan merah, hijau, biru dan background. Dengan demikian, bila ketiga koordinat diatur pada sumbu koordinat xyz nilainya 0, warna yanghasilkan adalah hitam. Bila ketiga sumbu koordinat di set dengan nilai 1, maka warna yang dihasilkan putih.

Pada CBIR kali ini akan dibandingkan *input* dari sebuah *image* dengan *image* yang dimiliki oleh dataset, hal ini dilakukan dengan cara mengubah *image* yang berbentuk RGB menjadi sebuah metode histogram warna yang lebih umum.

Histogram warna adalah frekuensi dari berbagai warna yang ada pada ruang warna tertentu hal ini dilakukan untuk melakukan pendistribusian warna dari *image*. Histogram warna tidak bisa mendeteksi sebuah objek yang spesifik yang terdapat pada *image* dan tidak bisa mendeskripsikan posisi dari warna yang didistribusikan.

Pembentukan ruang warna perlu dilakukan dalam rangka pembagian nilai citra menjadi beberapa *range* yang lebih kecil. Hal itu dilakukan untuk membuat sebuah histogram warna yang setiap interval tiap *range* dianggap sebagai *bin*. Histogram warna dapat dihitung dengan menghitung piksel yang menyatakan nilai warna pada setiap interval. Fitur warna mencakup histogram warna global dan histogram warna blok.

Pada perhitungan histogram, warna global HSV lebih dipilih karena warna tersebut dapat digunakan pada kertas (*background* berwarna putih) yang lebih umum untuk digunakan. Maka dari itu, perlu dilakukan konversi warna dari RGB ke HSV

dengan prosedur sebagai berikut.

- Nilai dari RGB harus dinormalisasi dengan mengubah nilai *range* [0, 255] menjadi [0, 1]

$$R' = \frac{R}{255} \quad G' = \frac{G}{255} \quad B' = \frac{B}{255}$$

**Gambar 2.** Normalisasi Nilai RGB pada Gambar

- Mencari  $C_{max}$ ,  $C_{min}$ , dan  $\Delta$

$$C_{max} = \max(R', G', B')$$

$$C_{min} = \min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

**Gambar 3.** Rumus CMax, CMin, dan  $\Delta$

- Selanjutnya gunakan hasil perhitungan di atas untuk mendapatkan nilai HSV

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left( \frac{G' - B'}{\Delta} \bmod 6 \right) & , C' \text{ max} = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right) & , C' \text{ max} = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right) & , C' \text{ max} = B' \end{cases}$$

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

$$V = C_{max}$$

**Gambar 4.** Rumus H, S, V

Nilai H pada perhitungan tersebut akan berkisar antara 0 - 360, sedangkan nilai S dan V akan berkisar antara 0 - 1. Maka dari itu, untuk menentukan kombinasi, nilai-nilai tersebut dipetakan sesuai *range*-nya. Pemetaan tersebut terdiri sebagai berikut.

$$H = \begin{cases} 0 & h \in [316, 360] \\ 1 & h \in [1, 25] \\ 2 & h \in [26, 40] \\ 3 & h \in [41, 120] \\ 4 & h \in [121, 190] \\ 5 & h \in [191, 270] \\ 6 & h \in [271, 295] \\ 7 & h \in [295, 315] \end{cases}$$

$$S = \begin{cases} 0 & s \in [0, 0.2) \\ 1 & s \in [0.2, 0.7) \\ 2 & s \in [0.7, 1] \end{cases}$$

$$V = \begin{cases} 0 & v \in [0, 0.2) \\ 1 & v \in [0.2, 0.7) \\ 2 & v \in [0.7, 1] . \end{cases}$$

**Gambar 5.** Pemetaan Nilai H, S, V

Setelah mendapatkan nilai HSV lakukanlah perbandingan antara *image* dari input dengan dataset dengan menggunakan *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 6.** Rumus *Cosine Similarity*

Dengan  $A$  dan  $B$  adalah vektor dan  $n$  adalah jumlah dimensi dari vektor. Tingkat kemiripan dihitung dari seberapa besar hasil dari *Cosine Similarity*.

Untuk melakukan pencarian histogram, dilakukan pencarian nilai H, S, dan V pada masing-masing pixel *image*. Lalu frekuensi kemunculan setiap kombinasi warna akan disimpan ke dalam suatu bins berukuran 72 dengan spesifikasi per elemen menunjukkan frekuensi kemunculan kombinasi HSV, misalnya [000, 001, 002, 010, 011, 012, dst sampai 722]. Tiap elemen bins tersebut menunjukkan jumlah kemunculan kombinasi HSV pada suatu gambar, seperti [293, 122, 493, dst]. Hal ini menunjukkan bahwa kemunculan kombinasi 000 pada gambar tersebut adalah sebanyak 293, 001 sebanyak 122, 002 sebanyak 493, dst. Kemudian data frekuensi kemunculan kombinasi

nilai HSV akan direpresentasikan sebagai satu vektor. Vektor antara *image* satu dan *image* lain nantinya akan dikalkulasi menggunakan *Cosine Similarity* untuk mencari persen kemiripan dari dua buah *image* atau lebih berdasarkan warnanya.

## 2.2. CBIR dengan Parameter Tekstur

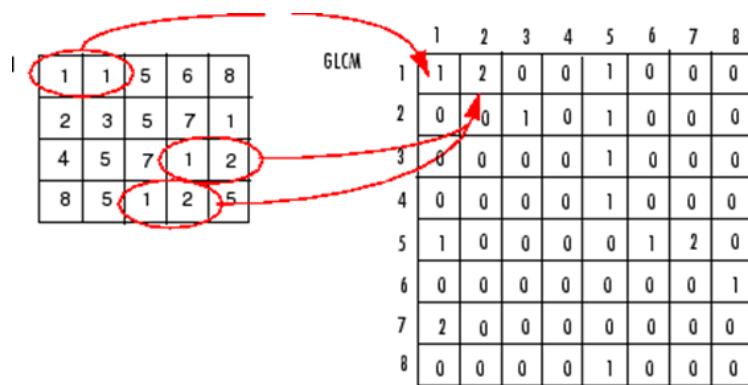
CBIR dengan perbandingan tekstur dilakukan menggunakan suatu matriks yang dinamakan *co-occurrence matrix*. Matriks ini digunakan karena dapat melakukan pemrosesan yang lebih mudah dan cepat. Vektor yang dihasilkan juga mempunyai ukuran yang lebih kecil. Misalkan terdapat suatu gambar  $I$  dengan  $n \times m$  piksel dan suatu parameter offset ( $\Delta x, \Delta y$ ), Maka dapat dirumuskan matriksnya sebagai berikut:

Dengan menggunakan nilai  $i$  dan  $j$  sebagai nilai intensitas dari gambar dan  $p$  serta  $q$  sebagai posisi dari gambar, maka *offset*  $\Delta x$  dan  $\Delta y$  bergantung pada arah  $\theta$  dan jarak yang digunakan melalui persamaan di bawah ini.

$$C_{\Delta x, \Delta y} (i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{jika } I(p, q) = i \text{ dan } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{jika lainnya} \end{cases}$$

**Gambar 7.** Rumus Pencarian Nilai Matrix Occurrence

Melalui persamaan tersebut, digunakan nilai  $\theta$  adalah  $0^\circ, 45^\circ, 90^\circ$ , dan  $135^\circ$ .



**Gambar 8.** Cara Pembuatan Matrix Occurrence

Setelah didapat *co-occurrence matrix*, buatlah *symmetric matrix* dengan menjumlahkan *co-occurrence matrix* dengan hasil transpose-nya. Lalu cari *matrix normalization* dengan persamaan.

$$\text{MatrixNorm} = \frac{\text{MatrixOcc}}{\sum \text{MatrixOcc}}$$

**Gambar 9.** Rumus Normalisasi Matrix Occurrence

Langkah-langkah dalam CBIR dengan parameter tekstur adalah sebagai berikut :

1. Konversi warna gambar menjadi *grayscale*, ini dilakukan karena warna tidaklah penting dalam penentuan tekstur. Oleh karena itu, warna RGB dapat diubah menjadi suatu warna *grayscale*  $Y$  dengan rumus:

$$Y = 0.29 \times R + 0.587 \times G + 0.114 \times B$$

**Gambar 10.** Rumus Pengubahan RGB menjadi *Grayscale*

2. Lakukan kuantifikasi dari nilai *grayscale*. Karena citra *grayscale* berukuran 256 piksel, maka matriks yang berkoresponden akan berukuran  $256 \times 256$ . Berdasarkan penglihatan manusia, tingkat kemiripan dari gambar dinilai berdasarkan kekasaran tekstur dari gambar tersebut.
3. Dari *co-occurrence matrix* bisa diperoleh 3 komponen ekstraksi tekstur, yaitu *contrast*, *entropy* dan *homogeneity*. Persamaan yang dapat digunakan untuk mendapatkan nilai 3 komponen tersebut antara lain :

*Contrast*:

$$\sum_{i,j=0}^{\text{dimensi}-1} P_{i,j} (i - j)^2$$

*Homogeneity* :

$$\sum_{i,j=0}^{\text{dimensi}-1} \frac{P_{i,j}}{1 + (i - j)^2}$$

*Entropy* :

$$-\left( \sum_{i,j=0}^{\text{dimensi}-1} P_{i,j} \times \log P_{i,j} \right)$$

**Gambar 11.** Rumus Contrast, Homogeneity, dan Entropy

**Keterangan :**  $P$  merupakan matriks *co-occurrence*

Dari ketiga komponen tersebut, dibuatlah sebuah vektor yang akan digunakan dalam proses pengukuran tingkat kemiripan.

4. Ukurlah kemiripan dari kedua gambar dengan menggunakan Teorema *Cosine Similarity*, yaitu:

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

**Gambar 12.** Rumus *Cosine Similarity*

Di sini  $A$  dan  $B$  adalah dua vektor dari dua gambar. Semakin besar hasil *Cosine Similarity* kedua vektor maka tingkat kemiripannya semakin tinggi.

### 2.3. Pengembangan Website

Pengembangan situs web adalah istilah yang merujuk pada praktik membangun, membuat, dan merawat situs web. Berdasarkan definisi dari sumber Techterms, ini mencakup aspek seperti desain web, penerbitan web, pemrograman web, dan manajemen basis data. Terkait dengan fungsi utamanya, pengembangan web tidak hanya berfokus pada desain situs web, melainkan terutama berkaitan dengan bagian pemrograman dan pengkodean, yang merupakan alasan utama berfungsinya situs web.

Dalam proses pengembangan web, beberapa tools atau alat yang umumnya digunakan meliputi GitHub untuk manajemen versi, Sass untuk pengolahan gaya dan tata letak, CodePen sebagai platform kolaboratif untuk pengembangan dan demonstrasi kode, AngularJS dan TypeScript untuk pengembangan aplikasi web dinamis, Sketch untuk desain antarmuka pengguna, jQuery sebagai perpustakaan JavaScript yang serbaguna, Sublime Text sebagai editor teks canggih, Bootstrap untuk pembangunan

situs web responsif, Grunt sebagai task runner, Chrome DevTools untuk debug dan profil web, NPM untuk manajemen paket JavaScript, dan Visual Studio Code sebagai editor kode sumber lintas platform. Tools ini membantu mempermudah dan meningkatkan efisiensi dalam proses pengembangan web.

Jenis pengembangan web terkait prosesnya, secara luas dapat didefinisikan menjadi tiga jenis, yaitu sebagai berikut:

### 2.3.1. Pengembangan Web *Front-End*

Tipe pengembangan web ini berkontribusi dalam mengubah informasi dan data yang tersedia menjadi antarmuka pengguna grafis menggunakan CSS, HTML, dan JavaScript. Hal ini bertujuan agar pengguna dapat berinteraksi dengan antarmuka yang disediakan dengan mudah. Tipe ini melibatkan semua elemen yang dapat secara langsung diamati dan dirasakan, termasuk warna, teks, gambar, tombol, dan unsur visual lainnya.



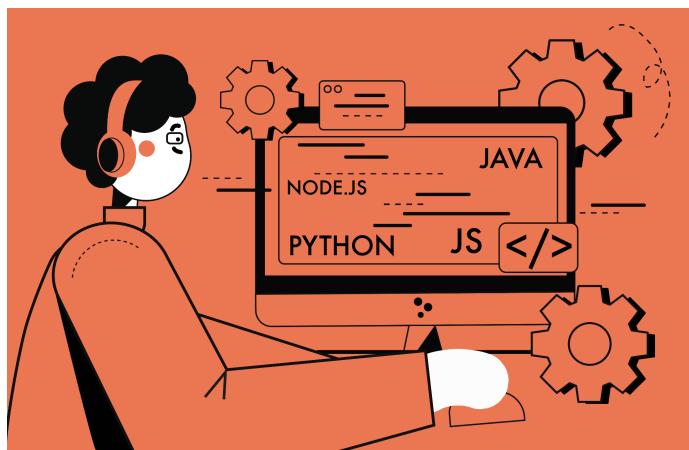
**Gambar 13.** Ilustrasi *Front-End*

Dalam hal tujuan dan fungsi utamanya, front-end developer memiliki beberapa tanggung jawab. Ini mencakup memberikan prioritas kepada *User Experience* (UX), mewujudkan konsep menggunakan HTML, CSS, dan JavaScript, serta menghasilkan serta merawat *User Interface* (UI) dari situs web dan aplikasi web. Selain itu, mereka bertanggung jawab untuk menciptakan alat-alat yang meningkatkan interaksi pengguna dengan situs di berbagai browser, menerapkan desain yang responsif untuk situs seluler, menjaga manajemen alur kerja perangkat lunak, mengikuti praktik terbaik dalam *Search Engine Optimization* (SEO), dan melakukan pengujian fungsi situs serta memperbaiki

bug yang muncul.

### 2.3.2. Pengembangan Web *Back-End*

Komponen back-end dari setiap halaman web tidak dapat terlihat atau diakses oleh pengguna. Meskipun tak terlihat, bagian ini merupakan tulang punggung atau inti dari suatu situs web. Meskipun tidak terlihat oleh pengguna, namun bagian ini memiliki peran krusial dalam memastikan fungsi keseluruhan situs web. Fungsi utamanya adalah menyimpan dan mengelola data serta menjamin kinerja dan fungsi yang tepat dari semua elemen yang terlihat di bagian depan atau sisi klien (client side). Bagian back-end berinteraksi dengan bagian depan halaman web dengan mengirim dan menerima data, yang nantinya ditampilkan secara langsung di situs web. Sebagai contoh, ketika pengguna menginput data, mengisi formulir, atau melakukan pembelian, browser mengirimkan permintaan tersebut ke bagian back-end atau sisi server, yang kemudian merespons dengan mengirimkan data sebagai kode front-end untuk halaman web, sehingga dapat dipahami, ditafsirkan, dan ditampilkan kepada pengguna.



**Gambar 14.** Ilustrasi *Back-End*

Perlu dicatat bahwa dalam pengembangan back-end, terdapat dua pendekatan utama, yaitu Object Oriented Programming (OOP) dan Functional Programming. OOP, sebagai jenis pertama, memusatkan pada penciptaan objek dan mengeksekusi pernyataan dalam urutan tertentu. Beberapa bahasa pemrograman OOP yang umum digunakan antara lain Java, .NET, Python, dan

Ruby. Sementara itu, Functional Programming, sebagai jenis kedua, menggunakan pendekatan yang lebih berbasis "action atau aksi" dan memungkinkan eksekusi pernyataan dalam urutan apa pun. Biasanya digunakan dalam ilmu data atau data science, bahasa pemrograman fungsional melibatkan SQL, F#, dan R sebagai contoh yang populer.

### **2.3.3. Pengembangan Web *Full-Stack***

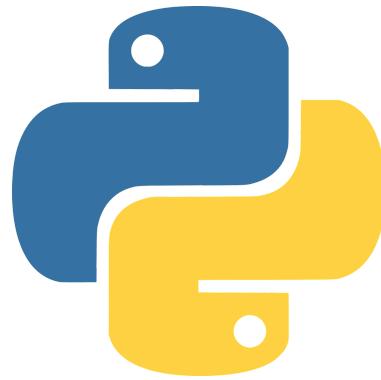
Full stack development merupakan pengembangan yang menggabungkan front-end dan back-end halaman web. Proses ini melibatkan pembuatan grafis dan desain halaman web, serta pengelolaan database untuk menyusun dan menyimpan data. Lebih spesifiknya, *full stack* developer bisa bekerja dengan javascript, PHP, java, database (backend) dan juga bisa mengkonversi desain ke dalam kode pemrograman seperti HTML, CSS, XML (front end).

## **2.4. *Programming Language***

Pada Tugas Besar kali ini, kami menerapkan *website* dengan *back-end* dan juga *front-end*. Untuk masing-masing bagian, bahasa yang digunakan berbeda. Kami menggunakan bahasa Python untuk *back-end*, NextJS untuk *front-end*, dan untuk menghubungkan kedua komponen tersebut, kami menggunakan Flask API. Penjelasan lebih lanjut tertera sebagai berikut.

### **2.4.1. Python**

Python adalah suatu bahasa pemrograman komputer yang umumnya digunakan untuk mengembangkan situs web, perangkat lunak/aplikasi, otomatisasi tugas, dan analisis data. Bahasa pemrograman ini termasuk dalam kategori tujuan umum, yang berarti dapat digunakan untuk membuat berbagai jenis program tanpa terbatas pada masalah tertentu. Karena sifatnya yang serbaguna dan kemudahan penggunaannya, Python menjadi salah satu bahasa pemrograman yang paling populer.



**Gambar 15.** Python *Programming Language*

Python sering dipakai untuk mengembangkan bagian belakang (*back-end*) dari suatu situs atau aplikasi, yang merupakan bagian yang tidak terlihat oleh pengguna. Peran Python dalam pengembangan web mencakup pengelolaan pengiriman data dari dan ke server, pemrosesan data, interaksi dengan server, pengaturan URL, dan penjaminan keamanan situs. Selain itu, Python juga menyediakan beberapa kerangka kerja (*framework*) untuk pengembangan web seperti Flask dan Django.

#### 2.4.2. NextJS

Next.js adalah suatu kerangka kerja yang sudah siap digunakan untuk membangun proyek dengan mudah dan langsung siap digunakan secara produksi. Ini berarti bahwa ketika proyek Next.js diatur untuk pertama kalinya, proyek tersebut dapat segera dijalankan tanpa perlu beralih dari mode pengembangan ke mode produksi terlebih dahulu. Selain itu, Next.js juga merupakan kerangka kerja full-stack, yang berarti bahwa Next.js tidak hanya berfungsi sebagai kerangka kerja front-end untuk membangun tampilan situs web, tetapi juga kerangka kerja back-end untuk menangani proses rendering dan manajemen basis data.



**Gambar 16.** Next.js *Framework*

Next.js menonjol dengan keunggulan-keunggulan yang tidak bisa dianggap remeh. Pertama, dalam hal optimasi SEO, Next.js memastikan pembentukan HTML saat browser memuat halaman, memudahkan proses crawling Google, dan meningkatkan peringkat website serta potensi traffic. Kedua, pengaturan proyek yang sederhana memudahkan bahkan pemula untuk mengatur proyek dengan mudah menggunakan Node.js tanpa perlu menyusun plugin atau library tambahan. Selain itu, proses setup dapat dilakukan di berbagai platform dan dapat diintegrasikan dengan layanan hosting yang mendukung Node.js. Ketiga, kinerja unggul Next.js terlihat dari penerapan code splitting dan navigasi sisi klien, mengurangi waktu loading secara signifikan dan memungkinkan navigasi tanpa jeda loading saat berpindah halaman. Terakhir, proses deploy proyek menjadi mudah berkat integrasi dengan Vercel, platform deployment berbasis cloud dengan dukungan first-class terhadap Next.js, memberikan kebebasan penggunaan SSG atau SSR, meningkatkan kecepatan loading, dan merampingkan beban kerja server utama. Dengan keunggulan-keunggulan ini, Next.js menjadi pilihan yang cocok untuk berbagai jenis website, mulai dari toko online, portal berita, blog, media sosial, hingga layanan streaming.

#### **2.4.3. Flask API**

Flask adalah sebuah framework dan library Python yang berfungsi sebagai alat bantu dalam pengembangan web development. Dengan menyediakan berbagai fitur terbaik melalui tools dan pustakanya, Flask memungkinkan pengguna untuk membangun web development secara efisien. Keunggulan Flask terletak pada kemampuannya menggunakan Python tanpa ketergantungan pada library pihak ketiga, membuatnya menjadi pilihan utama bersamaan dengan bahasa pemrograman Python.

Berbeda dengan beberapa bahasa pemrograman lain yang membutuhkan library tambahan untuk pertumbuhannya, Flask memungkinkan pengguna untuk menggunakan Python dengan lebih ringan dan mudah. Sebagai mikro framework, Flask mengurangi ketergantungan pada library dan extensi eksternal,

memungkinkan pembuatan core dengan sintaks yang sangat sederhana. Keunikan Flask terletak pada kategorinya sebagai mikro framework, yang tidak hanya mendukung kebutuhan dasar seperti deteksi error dan keamanan sistem, tetapi juga memungkinkan pengguna untuk menambahkan ekstensi sesuai kebutuhan.

Flask ini memiliki beberapa kelebihan. Pertama, kelebihan performa dengan desain modular yang super simpel menjadikan Flask sebagai framework web development yang sangat ringan. Kedua, keberlanjutan dalam pembelajaran, karena menggunakan bahasa yang ramah untuk pemula, membuat proses memahami Python dengan Flask lebih mudah. Ketiga, dukungan terhadap extensi pihak ketiga memungkinkan pengguna membangun web development yang kompleks dengan penambahan library, modul, dan plugin tambahan. Keempat, sifat open source dan dukungan sistem dari Flask memungkinkan penggunaan secara gratis, bahkan untuk kebutuhan komersial. Kelima, fleksibilitasnya memungkinkan pengembangan front end dan back end dalam satu framework. Keenam, berbagai fitur bawaan menarik, seperti pengembangan server, dukungan integrasi pada pengujian unit, Secure Cookies, dan ReSTful Request dispatching. Flask dapat digunakan untuk berbagai keperluan, seperti membuat website, aplikasi web, sistem informasi, game berbasis web, dan toko online, menawarkan berbagai fitur dan kemungkinan eksplorasi dalam pengembangan berbagai proyek.

## **BAB III**

### **ANALISIS PEMECAHAN MASALAH**

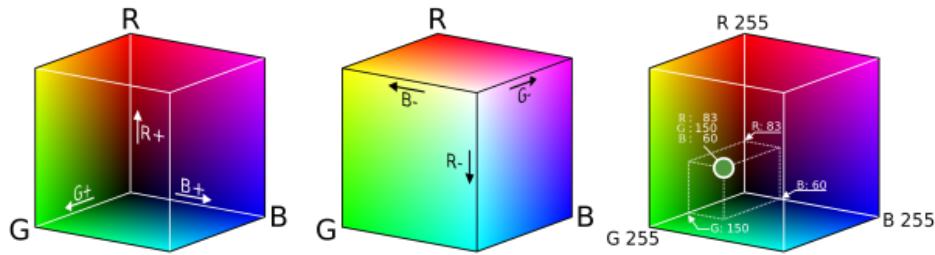
#### **3.1. Citra**

Citra merupakan salah satu bentuk media informasi yang terbentuk dari titik-titik, garis, dan bidang, dengan satu atau lebih warna. Kategorinya mencakup gambar dan video, dimana pada gambar hanya terdapat satu citra, sedangkan pada video, serangkaian citra berganti seiring waktu, menciptakan gerakan. Citra digital adalah representasi citra yang telah digitalisasi menjadi struktur data, dikenal sebagai *raw image*. *Raw image* adalah citra yang baru saja digitalisasi dengan struktur data sederhana yang mencakup informasi ukuran citra dan intensitas data untuk setiap pixel. Citra digital direpresentasikan dalam matriks  $M \times N$ , dengan citra berwarna umumnya menggunakan model warna RGB, di mana setiap pixel memiliki informasi warna merah, hijau, dan biru.

Namun pada citra tidak hanya terdapat unsur warna saja, terdapat pula unsur lain. Unsur pertama adalah tekstur yang menggambarkan sifat permukaan pada gambar, seperti halus, kasar, licin, atau kusam. Suatu gambar dapat dianalisis dari unsur-unsurnya pula. Pada Tugas Besar kali ini, analisis gambar dilakukan pada aspek warna dan tekstur. Penjelasan lebih lanjut tertera pada subbab berikut.

#### **3.2. Analisis Citra Menggunakan Strategi Warna**

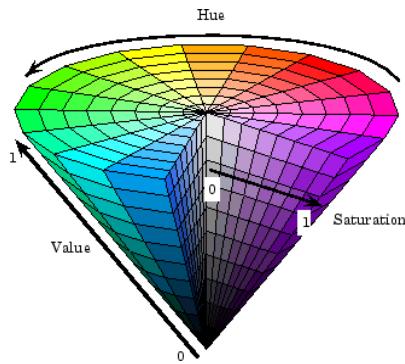
Seperti yang sudah dijelaskan sebelumnya, warna pada citra biasanya direpresentasikan dengan model warna RGB. Pada model warna RGB terdapat 3 matriks yaitu matriks warna red, warna green, dan warna blue. Pada setiap pixel warna memiliki rentang nilai intensitas mulai dari 0 sampai dengan 255. Rentang intensitas ini bernilai 8bit.



**Gambar 17.** Ruang Model Warna RGB

Pada Tubes ini, analisis gambar dilakukan setelah mengkonversi nilai RGB menjadi HSV yaitu Hue, Saturation, dan Value. Hue merupakan atribut penting dalam model warna HSV yang digunakan untuk menunjukkan warna sebenarnya, seperti merah, oranye, kuning, dan sebagainya. Hue berperan dalam membedakan tingkat kemerahan, kehijauan, kekuningan, kebiruan, dan sebagainya, berdasarkan gelombang cahaya. Rentang nilai untuk Hue adalah dari 0 hingga  $2\pi$ , menciptakan representasi lingkaran sehingga jika kita melacak dari warna merah, kita akan kembali ke warna merah. Dalam model HSV terdapat rumus khusus untuk menghitung nilai Hue, yang berbeda untuk mempermudah implementasi pada komputer (tertera pada bab 2).

Atribut selanjutnya adalah saturation yang menunjukkan sejauh mana keputihan warna pada suatu hue. Sebagai contoh, jika tingkat saturation warna merah tinggi atau bernilai 1, itu berarti tidak ada kandungan warna putih pada hue tersebut. Sebaliknya, jika saturation memiliki nilai 0, maka warna tersebut tidak memiliki hue atau sepenuhnya berwarna putih. Dalam model HSV terdapat rumus khusus untuk menghitung nilai saturation yang telah dipaparkan pada bab 2. Sedangkan value pada model HSV merupakan nilai kadar warna yang berada pada warna tersebut. Mirip seperti saturation, value ini memiliki rentang dari 0 hingga 1. Apabila nilai value ini sebesar 0. Maka warna tersebut akan menjadi hitam. Sedangkan apabila nilai value bernilai 1 maka tingkat kandungan warna hitam pada warna tersebut menjadi hilang.



**Gambar 18.** Ruang Model Warna HSV

Cara kami menganalisis gambar berdasarkan nilai HSV ini, kami menghitung frekuensi kemunculan kombinasi HSV pada tiap pixel pada gambar. Frekuensi tersebut akan direpresentasikan sebagai suatu vektor yang berukuran 72. Lalu untuk mencari kemiripan antara gambar satu dengan gambar lain, akan dihitung nilai *cosine similarity* antara vektor HSV gambar satu dengan vektor HSV gambar lainnya.

### 3.3. Analisis Citra Menggunakan Strategi Tekstur

Tekstur merupakan salah satu karakteristik penting dalam melakukan analisis dan identifikasi citra. Komponen tekstur mengandung informasi mengenai susunan struktur permukaan sebuah citra dan terdapat 14 fitur yang dapat diekstraksi dari sekumpulan data gambar di sekitar area yang dianalisis. Salah satu cara untuk melakukan ekstraksi fitur-fitur tersebut adalah dengan menggunakan Gray Level Co-occurrence Matrix (GLCM) atau Gray-Tone Spatial-Dependence Matrices yang pertama kali dikenalkan penggunaanya oleh Haralick dkk. GLCM merupakan histogram dua dimensi dari tingkat keabuan (*gray level*) antara sepasang piksel, yang dipisahkan oleh hubungan spasial yang tetap. Dengan kata lain, GLCM adalah tabulasi kemunculan kombinasi perbedaan tingkat keabuan pada citra. Matriks GLCM dapat dikalkulasikan menggunakan vektor perpindahan dengan radius  $\delta$  dan orientasi  $\theta$ .

Dalam penggeraan tugas besar ini, sebelum dilakukan kalkulasi GLCM, citra masukkan akan dikonversi menjadi citra *grayscale* dengan rentang tingkat keabuan piksel antara 0 hingga 255 dengan rumus berikut:

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

Dalam kalkulasi GLCM terdapat variasi  $\delta$  dengan rentang nilai 1, 2, hingga 10 dan  $\theta$  yang memiliki 8 opsi, yaitu  $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ$  atau  $315^\circ$ , karena setiap pikselnya dikelilingi oleh 8 piksel lainnya. Namun dengan mengacu pada definisi GLCM sendiri, pasangan *co-occurrence* yang didapatkan dengan orientasi sudut  $\theta$  dan  $(\theta + 180)$  adalah sama sehingga hanya terdapat 4 variasi GLCM berdasarkan orientasinya. Dari hasil percobaan yang telah dilakukan, didapatkan  $\delta$  dengan nilai 1 dan  $\theta$  dengan nilai  $90^\circ$  sebagai variasi paling optimal dengan mempertimbangkan waktu pemrosesan program dan akurasi.

Matriks GLCM hasil kalkulasi akan diubah menjadi matriks simetris dengan menjumlahkan matriks dengan matriks hasil transposenya sendiri dan selanjutnya dinormalisasi dengan membagi nilai *co-occurrence* piksel dengan jumlah seluruh nilai *co-occurrence*. Fitur tekstur kemudian baru dapat diekstraksi dari GLCM yang simetris dan telah dinormalisasi dengan berdasar pada statistik yang menyimpulkan relasi frekuensi distribusi seberapa seringnya suatu tingkat keabuan muncul pada hubungan spasial yang spesifik dengan tingkat keabuan lain pada citra. Terdapat 6 tekstur yang diekstraksi untuk analisis citra pada tugas besar ini, yaitu *energy*, *dissimilarity*, *entropy*, *contrast*, *homogeneity*, dan *correlation*.

*Energy* biasa juga disebut sebagai *Uniformity* atau *Angular Second Moment* (ASM) mengukur keseragaman tekstural yaitu pengulangan kemunculan pasangan piksel. Nilai dari *energy* dapat dihitung dengan menggunakan rumus berikut:

$$\text{Energy} = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j}}$$

*Dissimilarity* mengukur jarak antara sepasang piksel dalam area yang sedang dianalisis dan nilainya dapat dihitung dengan menggunakan rumus berikut:

$$\text{Dissimilarity} = \sum_{i,j=0}^{N-1} P_{i,j} |i - j|$$

*Entropy* mengukur ketidakteraturan atau kompleksitas dari sebuah citra dan nilainya dapat dihitung dengan menggunakan rumus berikut:

$$\text{Entropy} = \sum_{i,j=0}^{N-1} P_{i,j} (-\ln(P_{i,j}))$$

*Contrast* mengukur perbedaan intensitas antara suatu piksel dengan piksel ketetanggaannya yang kontinu secara keseluruhan. Nilai *contrast* dapat dihitung dengan menggunakan rumus berikut:

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P_{i,j} (i - j)^2$$

*Homogeneity* juga biasa disebut dengan *Inverse Difference Moment* mengukur homogenitas atau seberapa dekat distribusi elemen GLCM dengan diagonalnya. Nilai *homogeneity* dapat dihitung dengan rumus berikut:

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} P_{i,j} \left( \frac{1}{1 + (i - j)^2} \right)$$

*Correlation* mengukur korelasi linear tingkat keabuan pada citra, dengan kata lain mengukur korelasi suatu piksel dengan piksel ketetanggaannya secara keseluruhan. Nilai *correlation* dapat dihitung dengan menggunakan rumus berikut:

$$\text{Correlation} = \sum_{i,j=0}^{N-1} P_{i,j} \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \right]$$

Dengan keterangan:

$$\mu_i = \sum_{i,j=0}^{N-1} i (P_{i,j})$$

$$\mu_j = \sum_{i,j=0}^{N-1} j (P_{i,j})$$

$$\sigma_i = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j} (i - \mu_i^2)}$$

$$\sigma_j = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j} (j - \mu_j^2)}$$

Untuk mencari tingkat kemiripan suatu citra dengan citra lainnya, 6 fitur tekstur yang telah diekstraksi akan dibuat ke dalam bentuk vektor untuk masing-masing citra dan dihitung besar nilainya dalam persentase dengan menggunakan *cosine similarity* antara vektor fitur tekstur satu citra dengan yang lainnya.

### 3.4. Pemrosesan Program Berbasis Website

Pengimplementasian program sistem temu balik gambar ini diluncurkan melalui sebuah *website*. Kami membuat website yang dapat menerima input sebuah gambar dan juga dataset image sehingga dapat dilakukan perbandingan antara kedua hal tersebut. Untuk menyambungkan program *front-end* dan *back-end*, kami membuat sebuah API proses image yang menerima array of image dan juga sebuah image untuk dibandingkan. Dengan kata lain, hasil upload pada web akan diproses melalui API ini untuk diteruskan ke program back-end sehingga program dapat dijalankan. Program kami akan mengembalikan hasil perbandingan berupa besar nilai persentase dan wujud gambar yang direpresentasikan. Hasil ini ditampilkan pada *website* dengan diurutkan berdasarkan persen kemiripan tinggi ke rendah.

Pembuatan *website* ini menggunakan dua tipe yaitu *front-end* dan *back-end*. Kami menggunakan bahasa pemrograman python untuk bagian *back-end*, sedangkan untuk *front-end* kami menggunakan *framework* Next.js. Alasan kami menggunakan bahasa python adalah karena python memiliki cukup banyak variasi library yang dapat digunakan untuk program ini seperti numpy, cv2, matplotlib, csv, multiprocessing, os, pathlib, dll. Untuk menyambungkan antara *front-end* dan *back-end*, kami menggunakan Flask API karena termasuk API (Application Programming Interface) karena ringan, cukup mudah dipahami, fleksibel, mudah diimplementasikan, bekerja dengan baik untuk

pengembangan prototipe, serta banyak ekstensi yang tersedia seperti flask cors. Kami juga menggunakan beberapa library lain seperti werkzeug.utils, urllib.parse, requests, bs4, dan juga reportlab. Fitur-fitur pada program kami dapat berjalan lebih efisien karena adanya pemanfaatan library-library tadi.

Pembagian halaman pada *website* kami bertujuan untuk mempermudah pengguna dalam mengakses program kami. Terdapat 5 *page* pada *website* kami yaitu Home, About Us, About Project, How to Use, dan Camera. Halaman-halaman ini memiliki informasi dan cara kerja yang berbeda-beda. Kami menggunakan sistem scrolling pada tiap page untuk memudahkan akses pengguna dan juga mengutamakan kenyamanan pengguna ketika mengakses *website*.

## BAB IV

### IMPLEMENTASI DAN UJI COBA

#### 4.1. Implementasi Fungsi Program

##### 4.1.1. Utility Program (util.py)

Fungsi	Parameter Input	Parameter Output	Penjelasan
get_img_path ()	1. file_name : nama file gambar yang diupload	String path image yang disambungkan dari web	Fungsi yang mengembalikan path image yang diupload dari web
read_img ()	1. file_name : nama file gambar yang diupload	Matrix BGR dari file yang diupload.	Fungsi yang membaca gambar yang telah diupload pada web.
get_dataset_path ()	-	String path dataset yang disambungkan dari web	Fungsi yang mengembalikan path image yang diupload dari web
read_dataset ()	1. data_path : path dataset yang telah dikonstruksi sebelumnya	1. img : array of matrix images di dalam dataset tersebut 2. img_paths : array of path images di	Fungsi yang membaca dataset yang telah diupload pada web dan menyimpan pathnya juga.

		dalam dataset tersebut	
dot_product_vector ()	<ol style="list-style-type: none"> <li>1. vector 1 : vektor yang berupa array</li> <li>2. vector 2 : vektor yang berupa array</li> </ol>	Hasil perhitungan dot product dari kedua vektor.	Fungsi yang memproses perkalian dot kedua vektor dan mengembalikan suatu nilai hasil perhitungan dot tersebut.
vector_length ()	<ol style="list-style-type: none"> <li>1. vector : vektor yang berupa array</li> </ol>	Hasil perhitungan panjang vektor.	Fungsi yang menghitung panjang suatu vektor.
cosine_similarity ()	<ol style="list-style-type: none"> <li>1. vector_img1: vektor yang berupa array</li> <li>2. vector_img2: vektor yang berupa array</li> </ol>	Hasil perhitungan <i>cosine similarity</i> antara kedua vektor image.	Fungsi yang menghitung persen kemiripan antara kedua vektor gambar.

#### 4.1.2. CBIR dengan Parameter Warna (colorbased.py)

Fungsi	Parameter Input	Parameter Output	Penjelasan
hsv_histogram ()	<ol style="list-style-type: none"> <li>1. img : gambar yang sudah dibaca dan tersimpan sebagai matriks</li> </ol>	<ol style="list-style-type: none"> <li>1. frequency_vector : array yang merepresentasikan vektor frekuensi</li> </ol>	Fungsi yang menghitung nilai Hue, Saturation, dan Value pada setiap pixel gambar yang diinput berdasarkan

		kemunculan tiap kombinasi HSV	perhitungan pada spek dan paper. Fungsi ini juga menghitung frekuensi kemunculan kombinasi HSV lalu disimpan pada array vektor berukuran 72.
parallel_hsv_histogram ()	1. args :	Vektor frekuensi kemunculan kombinasi HSV dari suatu dataset.	Fungsi yang menghitung frekuensi kemunculan kombinasi HSV pada image-image yang terletak pada dataset dengan cara multiprocessing,
parallel_check_similarity ()	1. img : satu gambar yang diupload dan sudah dibaca serta disimpan sebagai matriks 2. imgs : array of matrix images yang terdapat pada dataset	1. sorted_res : array yang berisi gambar-gambar yang memiliki kemiripan >60% dengan img yang diupload, disimpan secara descending berdasarkan persennya dan	Fungsi yang menghitung <i>cosine similarity</i> tiap gambar pada dataset dengan image yang diupload berdasarkan tingkat kemiripan warna dengan menggunakan multiprocessing.

		juga menyimpan indeks gambar pada dataset.	
--	--	--	--

#### 4.1.3. CBIR dengan Parameter Tekstur (texturebased.py)

Fungsi	Parameter Input	Parameter Output	Penjelasan
createTextureVect ()	1. img : gambar yang diupload dan sudah dibaca serta disimpan sebagai matriks	Array yang berisi komponen vektor properti GLCM	Fungsi yang mengembalikan vektor properti GLCM (Gray-Level Co-Occurrence Matrix)
energy ()	1. glcm : Matriks GLCM yang telah dinormalisasi	Value energy dari GLCM	Fungsi yang menghitung value energy dari GLCM yang telah dinormalisasi
dissimilarity ()	1. glcm : Matriks GLCM yang telah dinormalisasi	Value dissimilarity dari GLCM	Fungsi yang menghitung value dissimilarity dari GLCM yang telah dinormalisasi
entropy ()	1. glcm :	Value entropy dari	Fungsi yang

	Matriks GLCM yang telah dinormalisasi	GLCM	menghitung value entropy dari GLCM yang telah dinormalisasi
contrast ()	1. glcm : Matriks GLCM yang telah dinormalisasi	Value contrast dari GLCM	Fungsi yang menghitung value contrast dari GLCM yang telah dinormalisasi
homogeneity ()	1. glcm : Matriks GLCM yang telah dinormalisasi	Value homogeneity dari GLCM	Fungsi yang menghitung value homogeneity dari GLCM yang telah dinormalisasi
correlation ()	1. glcm : Matriks GLCM yang telah dinormalisasi	Value correlation dari GLCM	Fungsi yang menghitung value correlation dari GLCM yang telah dinormalisasi
checkTextureSimilarity ()	1. img : satu gambar yang diupload dan sudah dibaca serta disimpan sebagai matriks	1. sorted_res : array yang berisi gambar-gambar yang memiliki kemiripan >60% dengan img yang diupload,	Fungsi yang menghitung <i>cosine similarity</i> tiap gambar pada dataset dengan image yang diupload berdasarkan tingkat kemiripan tekstur dengan menggunakan multiprocessing.

	2. imgs : array of matrix images yang terdapat pada dataset	disimpan secara descending berdasarkan persennya dan juga menyimpan indeks gambar pada dataset.	
--	--	---	--

## 4.2. Struktur Website

### 4.2.1. Halaman Utama (Home)

Halaman ini berisi program utama kami yaitu Content Based Image Retrieval untuk mencari persentase kemiripan suatu gambar dengan gambar-gambar pada dataset. Terdapat dua tipe CBIR yang kami sediakan yaitu CBIR warna dan CBIR tekstur. Untuk masukan dataset dapat menggunakan dua cara yaitu dengan upload dari files dan *web scraping*.

### 4.2.2. Halaman How to Use

Halaman ini berisi cara penggunaan website dan program utama, *Content-Based Image Retrieval* (CBIR). Petunjuk penggunaan tertera baik dalam Bahasa Indonesia maupun Bahasa Inggris.

### 4.2.3. Halaman About This Project

Halaman ini berisi penjelasan singkat mengenai konsep dasar yang digunakan dalam *Content-Based Image Retrieval* (CBIR) dengan parameter warna dan tekstur.

#### **4.2.4. Halaman About Us**

Halaman ini berisi identitas pembuat program, yaitu kami bertiga. Terdapat informasi mengenai nama lengkap, NIM, dan juga kesan pesan kami mengenai tugas besar ini.

#### **4.2.5. Halaman Kamera**

Halaman ini berisi fitur pencarian gambar menggunakan tangkapan gambar *real time* dengan kamera sebagai masukan. Program kamera yang kami gunakan memanfaatkan fitur webcam dari React. Kamera ini akan aktif selama 10 detik di saat awal untuk menangkap gambar *real time* lalu dikomparasi dengan dataset yang diupload.

#### **4.2.6. Fitur *Web Scraping***

Fitur ini memungkinkan pengguna untuk mencari dataset dari suatu *website*. Jadi pengguna tidak selalu mengupload dataset dari file lokal mereka.

### **4.3. Tata Cara Penggunaan Program**

Berikut merupakan tata cara penggunaan *website* program dalam Bahasa Indonesia:

1. Navigasikan *website* menggunakan tombol pada bagian kiri atas; 'Home,' 'How to Use,' and 'About Us.'
2. Untuk membandingkan gambar, tekan tombol 'Home' dan pilih metode yang diinginkan; 'Color' atau 'Tekstur.'
3. Unggah gambar Anda pada kotak sebelah kiri ('Upload your image!') dan *dataset* (dalam bentuk folder) Anda pada kotak sebelah kanan ('Upload your dataset!').
4. Tekan tombol 'Search!' di bagian bawah untuk memulai proses perbandingan; tunggu hingga proses selesai untuk waktu lama proses muncul di layar.
5. Gambar dataset yang memiliki kemiripan akan muncul secara urut. Untuk menavigasikan hasil perbandingan, gunakan tombol dengan angka.

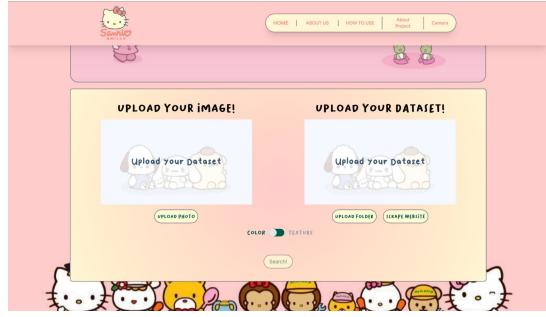
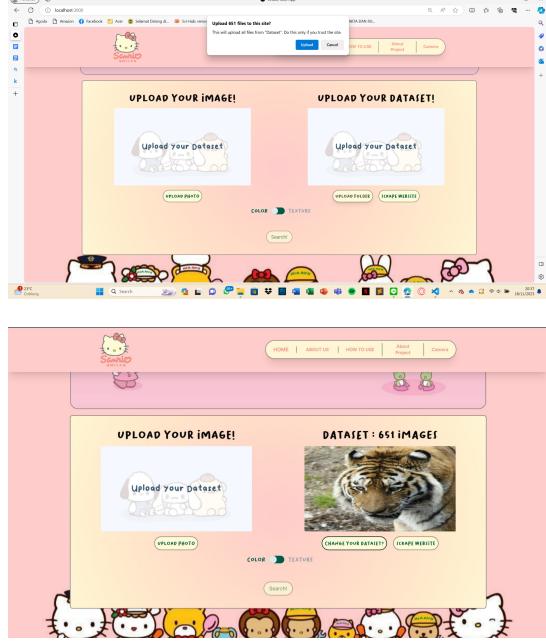
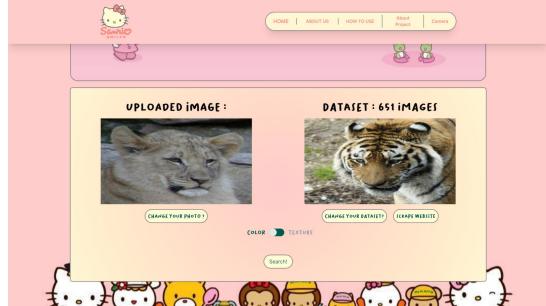
6. Untuk mengganti gambar atau *dataset* yang telah diunggah, masing-masing tekan tombol 'Change your image?' atau 'Change your dataset?.'
7. Untuk mengakses laman *web* lain cnya, tekan tombol 'How to Use' untuk instruksi penggunaan dan tombol 'About Us' untuk informasi tentang kami.

Berikut merupakan tata cara penggunaan *website* program dalam Bahasa Inggris:

1. Navigate using the top-left buttons; 'Home,' 'How to Use,' and 'About Us.'
2. To compare images, click 'Home' and choose a method; 'Color' or 'Texture.'
3. Upload your image in the left box ('Upload your image!') and your dataset (as a folder) in the right box ('Upload your dataset!').
4. Click 'Search!' at the bottom to initiate the comparison; wait for the process completion time to display.
5. Similar dataset images will appear, ordered by similarity. Navigate through pages using numbered buttons.
6. Change your uploaded image or dataset by clicking 'Change your image?' or 'Change your dataset?' respectively.
7. Explore additional pages or find instructions on the 'How to Use' page and information about us on the 'About Us' page.

#### 4.4. Hasil Pengujian

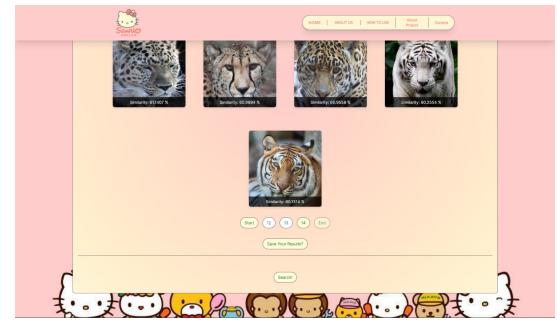
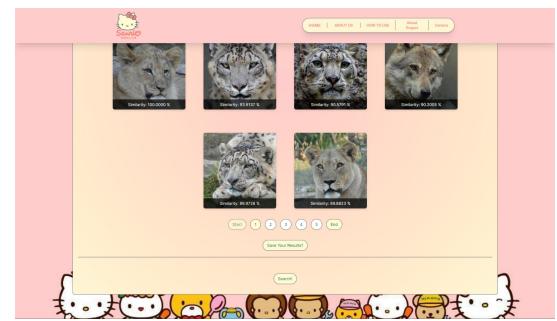
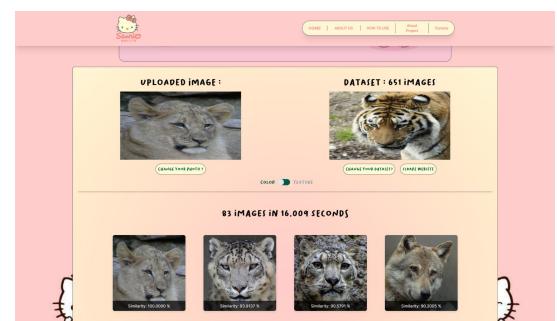
Pengujian yang Dilakukan	Hasil Pengujian
Halaman utama <i>website</i> yang berisi deskripsi program	

<p>Halaman utama <i>website</i> sebelum dilakukan <i>upload</i> dataset dan image</p>	
<p>Pengujian tombol <i>upload</i> dataset, terdapat <i>confirmation message</i> pada bagian atas halaman <i>website</i>. Setelah dikonfirmasi, akan muncul gambar paling awal dari dataset tersebut pada kotak kanan.</p>	
<p>Pengujian tombol <i>upload image</i></p>	

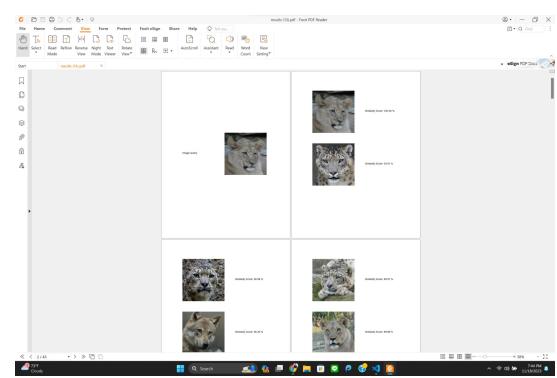
Pengujian tombol *search* untuk memulai proses perbandingan berdasarkan warna, tertera waktu *real time* pengujian



Halaman hasil *query image* dari dataset yang memiliki kemiripan warna sebesar > 60% dengan *image* yang diupload, tertera pula waktu yang dibutuhkan untuk melakukan rangkaian proses perbandingan. Tertera gambar pertama memiliki tingkat kemiripan warna sebesar 100%, sedangkan gambar terakhir memiliki tingkat kemiripan warna sebesar 60.1%.



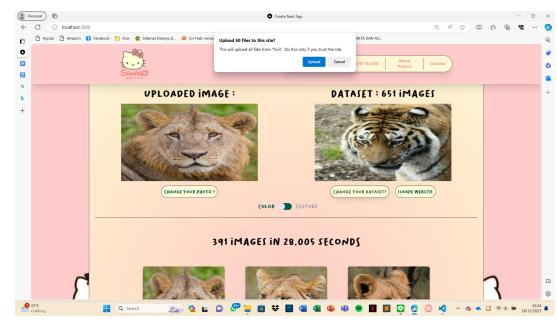
Hasil penulisan ke dalam suatu file PDF setelah menekan tombol ‘Save Your Results’

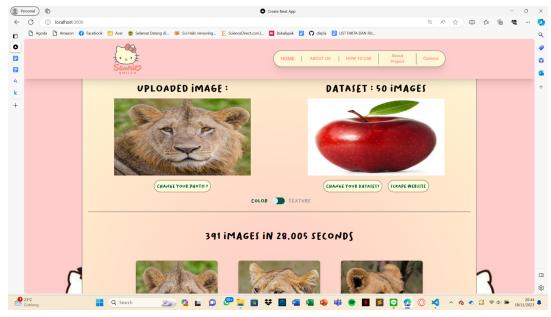
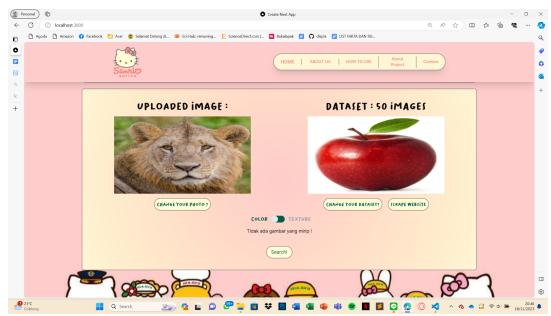
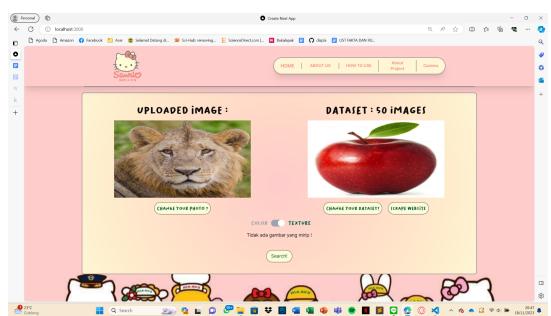
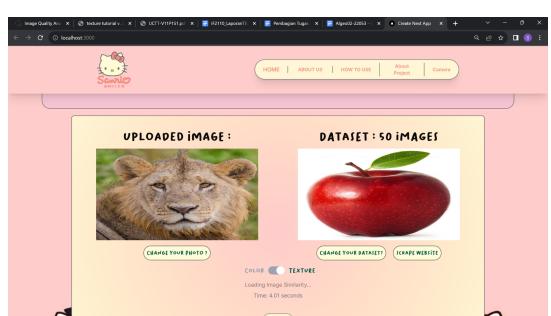


Penggantian gambar yang diupload serta hasil pengujian pada gambar lain dengan dataset yang sama. Karena penggunaan *caching* pada dataset yang pernah diakses, maka waktu yang diperlukan untuk membandingkan image lain dengan dataset yang sama. Waktu yang dibutuhkan lebih cepat dibandingkan yang pertama.

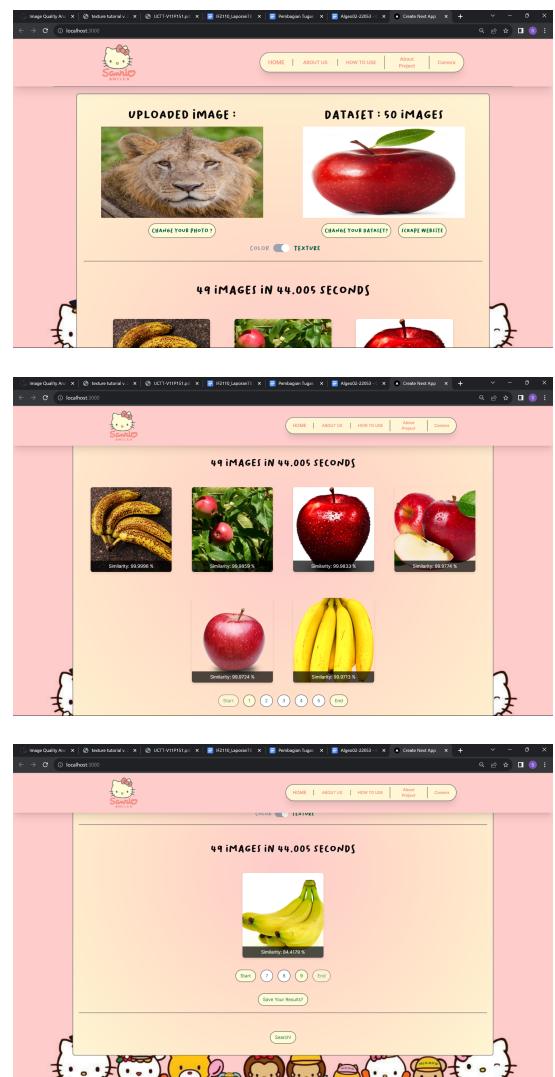


Penggantian dataset yang lain

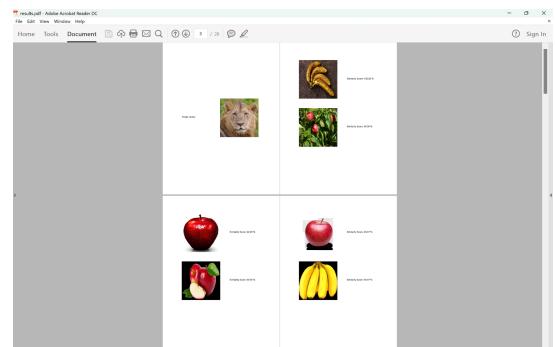


	
Kasus dimana tingkat kemiripan gambar-gambar pada dataset dengan gambar yang diupload tidak ada yang bernilai $> 60\%$	
Penggantian tipe CBIR dari color menjadi texture	
Pengujian tombol <i>search</i> untuk memulai proses perbandingan tekstur, tertera waktu <i>real time</i> pengujian	

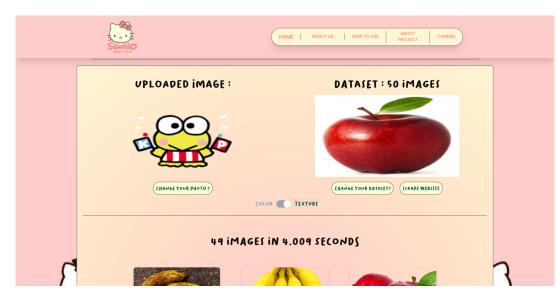
Halaman hasil *query image* dari dataset yang memiliki kemiripan tekstur sebesar > 60% dengan *image* yang diupload, tertera pula waktu yang dibutuhkan untuk melakukan rangkaian proses perbandingan. Tertera gambar pertama memiliki tingkat kemiripan warna sebesar 99.9%, sedangkan gambar terakhir memiliki tingkat kemiripan warna sebesar 84.4%.



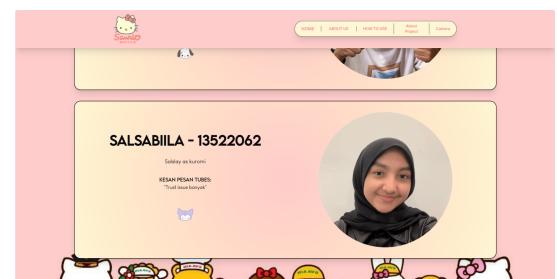
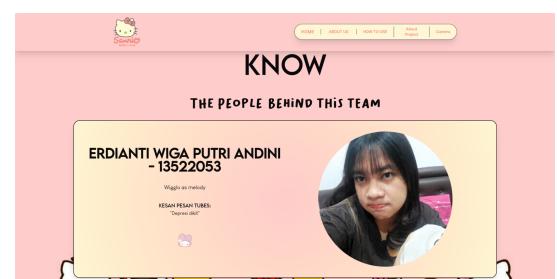
Hasil penulisan ke dalam suatu file PDF setelah menekan tombol ‘Save Your Results’



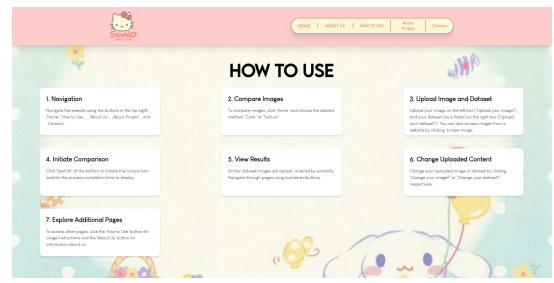
Penggantian gambar yang diupload serta hasil pengujian pada gambar lain dengan dataset yang sama. Karena penggunaan *caching* pada dataset yang pernah diakses, maka waktu yang diperlukan untuk membandingkan image lain dengan dataset yang sama. Waktu yang dibutuhkan lebih cepat dibandingkan yang pertama.



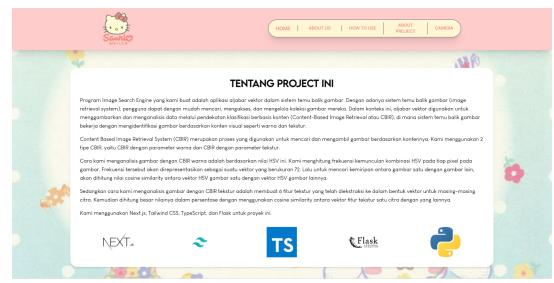
Halaman About Us yang berupa *page scrolling*, berisi identitas dan pesan kesan dari anggota kelompok



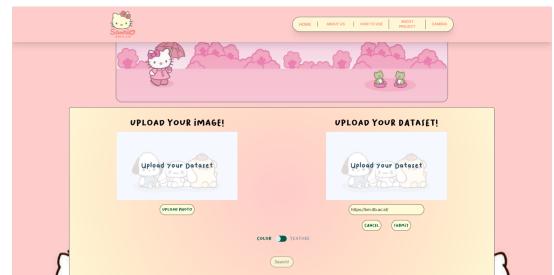
Halaman How to Use yang berisi penjelasan cara penggunaan website

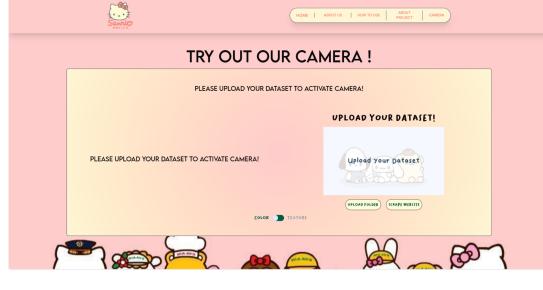
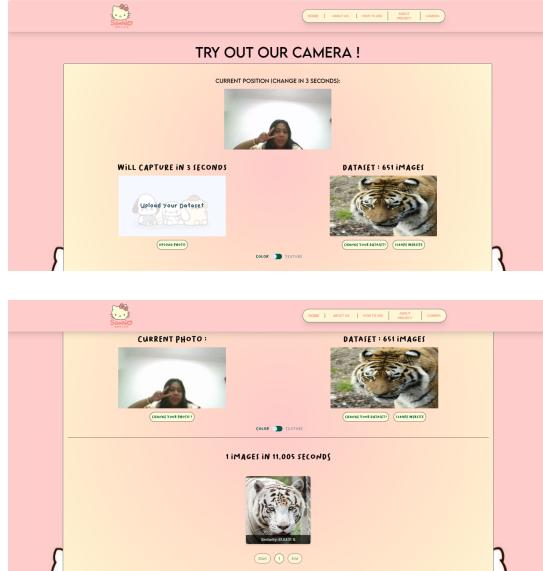


Halaman penjelasan konsep singkat *search engine* yang dibuat



Penggunaan fitur *web scraping* sebagai masukan dataset serta hasil perbandingannya



	
<p>Tampilan awal pada halaman fitur kamera. Diharuskan mengupload sebuah dataset terlebih dahulu untuk mengaktifkan kamera.</p>	
<p>Setelah dataset diupload, kamera akan otomatis aktif dan menangkap gambar dalam rentang waktu 10 detik dan otomatis melakukan perbandingan antara tangkapan gambar dengan dataset. Untuk tampilan waktu dan hasil masih sama seperti sebelumnya.</p>	

#### **4.5. Analisis Solusi Algoritma**

Berdasarkan uji coba dua tipe perhitungan CBIR yang kami lakukan, didapat hasil bahwa hampir 80% perhitungan berdasarkan fitur warna lebih baik daripada fitur tekstur. Hal ini dapat terjadi dikarenakan fitur warna mengekstrak warna per pixel image lalu mengkonversikannya menjadi nilai HSV. Perhitungan yang dilakukan adalah berdasarkan frekuensi kemunculan tiap kombinasi nilai HSV sehingga lebih akurat. Menurut kami, yang membuat fitur tekstur tidak terlalu baik adalah karena fitur ini mengekstrak suatu gambar menjadi *grayscale* terlebih dahulu dan kami menganggap bahwa pengkonversian ini setidaknya menimbulkan sedikit perbedaan. Terlebih dengan perhitungan manual GLCM beserta banyaknya ketersediaan variasi sudut dan jarak. Selain itu, fitur warna bekerja lebih cepat dibandingkan fitur tekstur karena perhitungan pada fitur warna relatif lebih sedikit walaupun vektornya berukuran lebih panjang.

*Parallel processing* yang kami terapkan juga berpengaruh tinggi pada kecepatan program. Dari awalnya proses perbandingan bisa memakan waktu sampai 0.5 detik per image, bisa dipercepat hingga 0.1 detik per image. Proses *caching* yang kami terapkan pada sebuah dataset juga mempercepat kerja program perbandingan suatu image berbeda dengan dataset yang sama. *Caching* ini menyimpan vektor tiap image pada dataset sehingga saat dataset tersebut digunakan kembali, program tidak perlu mengekstrak nilai vektor lagi, hanya langsung melakukan perbandingan antargambar.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Dalam sistem temu balik gambar, Content Based Image Retrieval (CBIR) dapat dilakukan dengan menggunakan parameter warna maupun tekstur. Pengujian dengan parameter warna menggunakan perhitungan nilai Hue, Saturation, dan Value (HSV) dari suatu gambar, sedangkan pengujian dengan parameter tekstur menggunakan perhitungan Gray Level Co-occurrence Matrix (GLCM) dari citra yang telah diubah ke citra *grayscale*.

Berdasarkan hasil percobaan yang telah dilakukan, terlihat bahwa metode CBIR dengan parameter warna membutuhkan waktu proses citra yang lebih sedikit dibandingkan dengan parameter tekstur. Hal ini disebabkan oleh penggunaan *for loop* pada kalkulasi GLCM dan perhitungan untuk ekstraksi fitur tekstur lebih banyak, seperti *energy*, *entropy*, *contrast*, *homogeneity*, *correlation*, dan *dissimilarity*. Sedangkan pada CBIR parameter warna, perhitungan hanya bersifat *increment* pada frekuensi kemunculan kombinasi nilai HSV. Maka dari itu, walaupun ukuran vektor CBIR warna lebih panjang yaitu 72 dibandingkan SBIR tekstur dengan panjang 6, CBIR tekstur membutuhkan waktu lebih lama untuk memproses tiap perhitungan.

#### **5.2. Saran**

Memperbaiki penggunaan *for loop* pada perhitungan GLCM untuk mempercepat lama *image processing*. Selain itu, dari segi desain UI/UX *website*, masih terdapat banyak aspek yang dapat diperbaiki untuk tugas maupun proyek selanjutnya.

#### **5.3. Komentar dan Tanggapan**

Tugas besar ini telah memberikan kesempatan bagi kami untuk mengeksplorasi lebih banyak mengenai CBIR serta pengembangan *website*. Melalui tugas besar ini juga kami dapat menjadi lebih *bonding* satu sama lain dan meningkatkan kerja sama tim serta komunikasi antarsesama.

#### **5.4. Refleksi terhadap Tugas Besar**

Kami berhasil menerapkan prinsip-prinsip dasar CBIR dengan baik dalam tugas besar ini. Analisis warna dan tekstur menggunakan metode CBIR terbukti dapat membandingkan suatu gambar dengan gambar lainnya. Ekstraksi fitur program telah berhasil diimplementasikan dengan cukup baik, hanya terdapat kekurangan-kekurangan minor saja. Tugas besar ini memperkaya pemahaman kami tentang pengolahan citra, temu balik informasi, serta bahkan pemrograman *website*. Kolaborasi dan komunikasi kelompok juga berjalan dengan lancar beserta distribusi tugas dan tanggung jawab yang dilakukan dengan efisien.

#### **5.5. Ruang Perbaikan atau Pengembangan**

Terdapat beberapa hal yang dapat diperbaiki dalam program yang telah kami buat. Perlu ditingkatkannya efisiensi dan waktu dalam melakukan *image processing*, yaitu dalam menghitung nilai HSV untuk parameter warna dan kalkulasi GLCM untuk parameter tekstur. Selain itu, *readability* dari *website* dapat diperbaiki sehingga menghasilkan pengalaman pengguna yang lebih baik dalam penggunaan program kami.

## DAFTAR PUSTAKA

- Ramadjanti, N. (2006). *Content Based Image Retrieval Berdasarkan Ciri Tekstur Menggunakan Wavelet*. Seminar Nasional Aplikasi Teknologi Informasi (SNATI) -. <https://journal.uii.ac.id/Snati/article/view/1530>
- Hastuti Ida, dkk. (2009). *Content Based Image Retrieval Berdasarkan Fitur Bentuk Menggunakan Metode Gradient Vector Flow Snake*. Seminar Nasional Informatika 2009 -. <https://media.neliti.com/media/publications/171268-ID-none.pdf>
- Halim, A. (2017, November 27). *APLIKASI CONTENT BASED IMAGE RETRIEVAL DENGAN ALGORITMA SOBEL'S EDGE DETECTION*. Halim | Seminar Nasional Informatika (SNIf). <https://e-jurnal.potensi-utama.ac.id/ojs/index.php/SNIf/article/view/237>
- Siagian, P. (2012, October 3). *Klasifikasi Citra Content-Based Image Retrieval dengan metode Shape Base Threshold*. Siagian | Seminar Nasional Teknologi Informasi Komunikasi Dan Industri. <https://ejurnal.uin-suska.ac.id/index.php/SNTIKI/article/view/2846>
- Yue, J., Li, Z., Liu, L., & Fu, Z. (2011). Content-based image retrieval using color and texture fused features. *Mathematical and Computer Modelling*, 54(3–4), 1121–1127. <https://doi.org/10.1016/j.mcm.2010.11.044>
- Rifqi Mulyawan (19 November 2023) *Web Development: Apa itu Pengembangan Website? Tujuan dan Fungsi, Jenis, Macam, Cara serta Bedanya dengan Web Design!*. Diambil dari <https://rifqimulyawan.com/blog/pengertian-web-development/>.
- Rifqi Mulyawan (19 November 2023) *Memahami Pengertian Front-End Development: Apa itu? Tujuan dan Fungsi, Jenis Pekerjaan, Macam-Macam Bahasa serta Perbedaannya dengan Back-End Development!*. Diambil dari <https://rifqimulyawan.com/blog/pengertian-front-end-development/>.

Rifqi Mulyawan (19 November 2023) *Penjelasan Pengertian Back-End Development: Apa itu? Tujuan, Jenis dan Bagian, Macam-Macam Bahasa Pemrograman serta Apa Saja Keterampilannya!. Diambil dari <https://rifqimulyawan.com/blog/pengertian-back-end-development/>.*

Intern, D. (2020, June 16). *Apa itu Full Stack Developer? Keahlian-keahlian yang harus dikuasai.*  
<https://www.dicoding.com/blog/apa-itu-full-stack-developer-keahlian-keahlian-yang-harus-dikuasai/>

Intern, D. (2023, May 29). *Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya.* Dicoding Blog.  
<https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>

Muhammad, A. (2023, July 14). *Next.js: Pengertian, Cara Kerja, dan Tutorial Project Next.js.* Niagahoster Blog. <https://www.niagahoster.co.id/blog/next-js-adalah/>

Alexandromeo. (2023, September 19). *Apa itu Flask? Pengertian, Kelebihan, dan Kegunaannya (Lengkap).* Makinrajin - Digital Marketing Agency Terbaik di Indonesia.  
<https://makinrajin.com/blog/flask-adalah/>

Istilah, P. D. (2023, September 25). *Pengertian Gambar, Jenis, dan Fungsinya dalam Kehidupan. Kumparan.*  
<https://kumparan.com/pengertian-dan-istilah/pengertian-gambar-jenis-dan-fungsinya-dalam-kehidupan-21FrJwGCOfd/1>

6 Unsur-unsur dalam menggambar. (n.d.). KoSingkat.  
<https://www.kosingkat.id/2020/02/5-unsur-unsur-dalam-menggambar.html>

Annas, Trian. (2019). *Perbandingan Model Warna RGB, HSL dan HSV Sebagai Fitur dalam Prediksi Cuaca pada Citra Langit menggunakan K-Means.*

Gadkari, Dhanashree. (2004). *Image Quality Analysis Using GLCM*.  
[https://stars.library.ucf.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1186&co\\_ntext=etd](https://stars.library.ucf.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1186&co_ntext=etd)

Aborisade, dkk. (2014). *Comparative Analysis of Textural Features Derived from GLCM for Ultrasound Liver Image Classification*.  
<https://ijcttjournal.org/Volume11/number-6/IJCTT-V11P151.pdf>

Hall-Beyer, Mryka. (2005). *GLCM TEXTURE: A TUTORIAL*.  
<http://lapi.fi-p.unam.mx/wp-content/uploads/Tutorial-GLCM.pdf>

## **LAST RELEASE REPOSITORY**

Repository dari Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri kelompok 44 “S43W79N” tertera pada tautan berikut.

<https://github.com/wigaandini/Algeo02-22053.git>

## **YOUTUBE VIDEO**

Pengerjaan bonus video tertera pada link berikut.

<https://youtu.be/YOp30OjEUek?si=o7xNCKRhyMP5eYe>

## LAMPIRAN CATATAN ASISTENSI

Kami melakukan asistensi pada Rabu, 8 November 2023. Hasil pembahasan asistensi kami tertera sebagai berikut.

TOPIK	PEMBAHASAN
CBIR dengan Parameter Warna	Perhitungan HSV image dilakukan setelah membagi image menjadi 9 blok. Lalu dilakukan perhitungan rata-rata nilai HSV per blok. Setiap nilai HSV per blok tersebut dicari nilai <i>cosine similarity</i> nya dengan setiap HSV blok gambar lain. Hasil <i>cosine similarity</i> antara 9 blok tersebut akan dirata-rata kembali sehingga didapat persen kemiripan antara kedua gambar. Gambar perlu di crop hingga dapat dibagi menjadi 9 blok dengan ukuran yang merata.
CBIR dengan Parameter Tekstur	Hasil 99.99% yang banyak pada cosine similarity untuk parameter tekstur adalah normal mengingat yang dilihat adalah tingkat kemiripan tekturnya, jadi mungkin saja foto muka orang memiliki kemiripan tekstur yang sangat tinggi dengan ilustrasi kiamat. Fitur tekstur yang diekstraksi boleh di luar apa yang tertera di spek dan variasi jarak dan angle dari GLCM bebas untuk dipilih dengan alasan yang sesuai. Gambar tidak boleh di resize ke 256 x 256.
Caching	Penjelasan mengenai tata cara caching

## DOKUMENTASI SELAMA *TIMELINE* TUBES

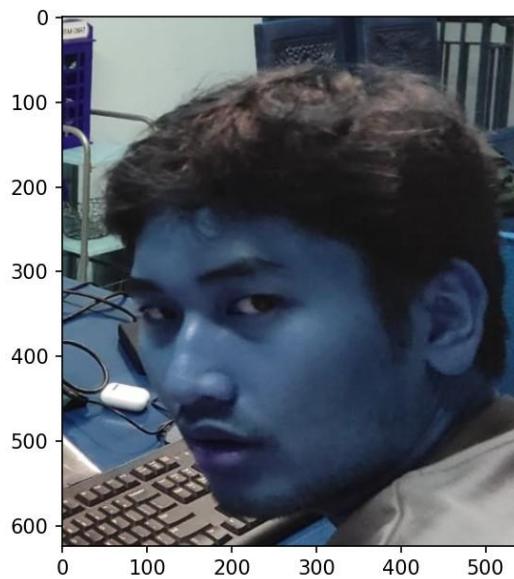
DOKUMENTASI	BACK STORY
	Hari pertama nubes offline bareng openingnya family din din chinese food dulu baru otw bangbayang
	Mba wigglo bela belain nubes padahal dia lagi bersusah payah sekuat tenaga ngejagain orang yang lagi sakit lemah lesu tidak berdaya di Rumah Sakit Santo Borromeus kasian x



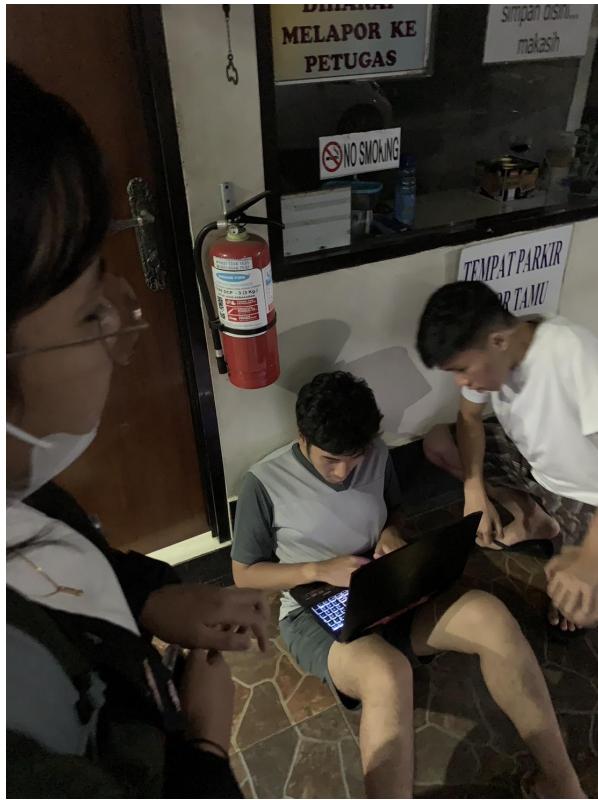
Nubes algeo disponsori martabak dan jullie ganteng x baik x keren x religius x <3



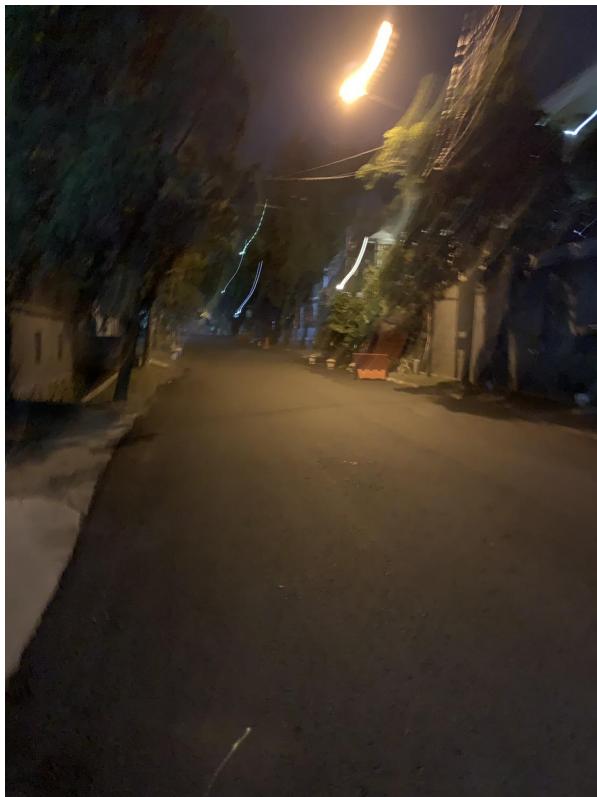
Close up martabak, kata wiga katanya naomi martabak mertua gak enak tapi ternyata updated verdict nya rasanya oke



Pengcarry handal algeo alias nuvatar dijadiin  
rgb saur kewl



Nubes sampai dikick sama yang jagain  
bangbayang eh pas nunggu gocar ini dua  
orang masih penasaran jadilah nuel ngemper  
pake baju tidur yang pas sampai kamar dia  
gak ganti jorok x



Balik jalan kaki ke kos jam 2 pagi tiap ada orang yang lewat n suspicious dikit langsung sport jantung kecil. Harus call biar ga takut jalan sendiri



Nuel yang bolak-balik naik tangga karena:

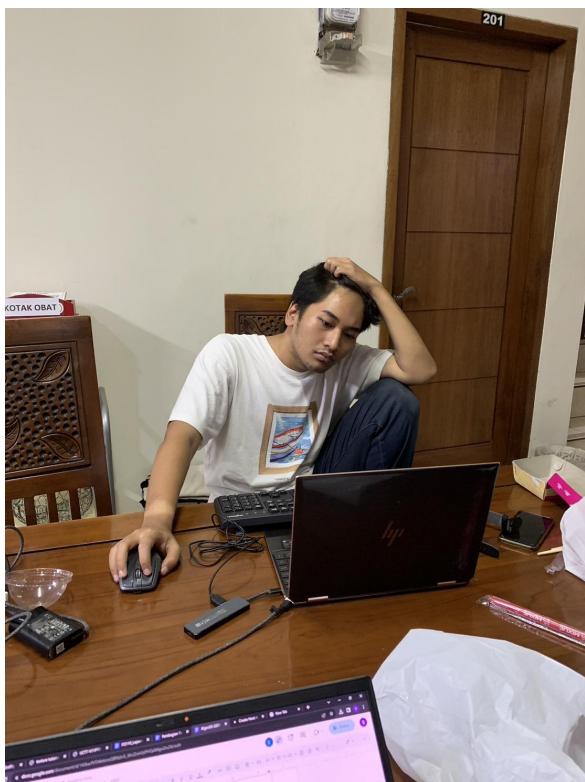
- Pake celana pendek
- Udah ganti celana tapi masih pake baju gelap
- Balik ketiga baru bener



Orang-orang yang fotoin gifnya nuel



Sebby jobdescnya nubes sisanya emotional support tapi boong kak semuanya kerja kok pls ;-;



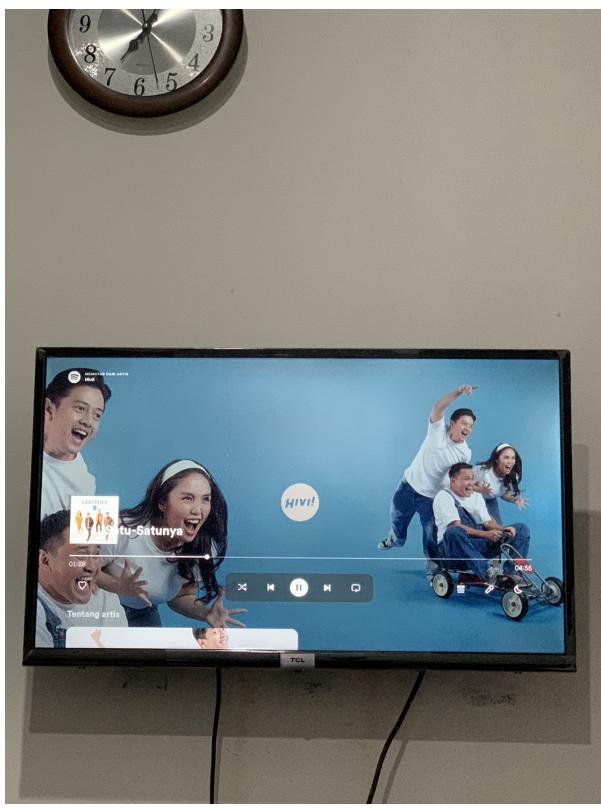
Kasian si sebby sebby ini semangat terus ya  
seb jangan depresi



Detik-detik sebelum kejadian drama video



Asupan self reward menamatkan tubes 2



Big thanks buat lagu kematian Tubes 2 Algeo

