

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 : Alex Sander (13521061)

Disusun oleh:
Kelas 01 - Kelompok 04 - Keluarga Cemara

Zaki Yudhistira Candra (13522031)
Amalia Putri (13522042)
Angelica Kierra Ninta Gurning (13522048)

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT, karena atas rahmat dan karunia-Nya, kami dapat menyelesaikan makalah ini dengan judul "Aplikasi Aljabar Vektor dalam Sistem Temu Balik Gambar". Makalah ini disusun sebagai salah satu tugas mata kuliah Aljabar Linier dan Geometri, dengan fokus pada penerapan konsep Content-Based Image Retrieval (CBIR) menggunakan parameter warna dan tekstur, yang diimplementasikan dalam sebuah website.

Dalam perkembangan teknologi informasi dan komunikasi yang semakin pesat, sistem temu balik gambar menjadi salah satu bidang penelitian yang menarik perhatian. CBIR adalah pendekatan yang memanfaatkan informasi konten dari gambar untuk melakukan pencarian dan pengindeksan. Dalam konteks ini, aljabar vektor menjadi landasan teoritis yang kuat untuk mengembangkan model matematis yang mendukung efektivitas dan efisiensi sistem temu balik gambar.

Penyusunan makalah ini tidak lepas dari bantuan berbagai pihak. Kami mengucapkan terima kasih kepada dosen beserta asisten pembimbing mata kuliah Aljabar Linier dan Geometri yang telah memberikan bimbingan dan panduan selama proses pembuatan makalah ini.

Semoga makalah ini dapat memberikan kontribusi positif dalam pemahaman dan pengembangan sistem temu balik gambar berbasis konten. Akhir kata, kami menyampaikan permohonan maaf atas segala keterbatasan dalam makalah ini, dan kami menerima dengan terbuka segala kritik dan saran yang bersifat membangun.

Bandung, 19 November 2023,

Zaki Yudhistira Candra	(13522031)
Amalia Putri	(13522042)
Angelica Kierra Ninta Gurning	(13522048)

DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR ISI.....	2
DAFTAR PEMBAGIAN TUGAS.....	3
1.1. Abstraksi.....	5
1.2. Content-Based Information Retrieval (CBIR).....	5
BAB II.....	7
2.1. CBIR dengan Parameter Warna.....	7
2.2. CBIR dengan Parameter Tekstur.....	8
2.3. Website.....	11
a. Front-End Development.....	11
b. Back-End Development.....	11
BAB III.....	13
3.1 Langkah-Langkah Pemecahan Masalah.....	13
a. CBIR dengan parameter warna.....	13
b. CBIR dengan parameter tekstur.....	14
3.2 Proses Pemetaan Masalah Menjadi Elemen-Elemen.....	14
3.3 Contoh Ilustrasi Kasus dan Penyelesaiannya.....	14
BAB IV.....	15
4.1 Implementasi Program Utama.....	15
4.2 Penjelasan Struktur Program Berdasarkan Spesifikasi.....	15
a. Content Based Image Retrieval (COLOR).....	15
b. Content Based Image Retrieval (TEXTURE).....	16
c. Flask.....	17
d. Caching.....	18
e. Download PDF.....	18
4.3 Tata Cara Penggunaan Program.....	19
a. Laman Home.....	19
b. Laman Search.....	20
c. Laman Camera.....	21
d. Laman Explore.....	22
4.4 Hasil Pengujian.....	23
4.5 Analisis Desain Solusi Algoritma.....	32
5.1. Kesimpulan.....	34
5.2. Saran.....	34
5.3. Komentar atau Tanggapan.....	34
5.4. Refleksi.....	35
5.5. Ruang Perbaikan atau Pengembangan.....	35
DAFTAR PUSTAKA.....	36
Repository.....	37
Youtube.....	37

DAFTAR PEMBAGIAN TUGAS

KEGIATAN/FITUR		PIC
Library	CBIR Warna	13522031
	CBIR Tekstur	13522042 13522048
Website	UI/UX	13522048
	Home Page	13522042 13522048
	Search Page	13522042
	Camera Page	13522042
	Explore Page	13522031 13522042
	Back-End (API)	13522031 13522042 13522048
Bonus	Camera	13522042
	Image Scraping	13522031 13522042 13522048
	Download PDF	13522048
	Caching	13522048 13522031
	Video	13522031 13522042 13522048
	Bab 1 (Deskripsi Masalah)	13522031 13522042 13522048
Laporan	Bab 2 (Teori Singkat)	13522048
	Bab 3 (Implementasi Pustaka)	13522031 13522048

	Bab 4 (Eksperimen)	13522031 13522048 13522042
	Bab 5 (Kesimpulan, Saran, Komentar, Refleksi, Ruang Perbaikan)	13522031 13522042 13522048

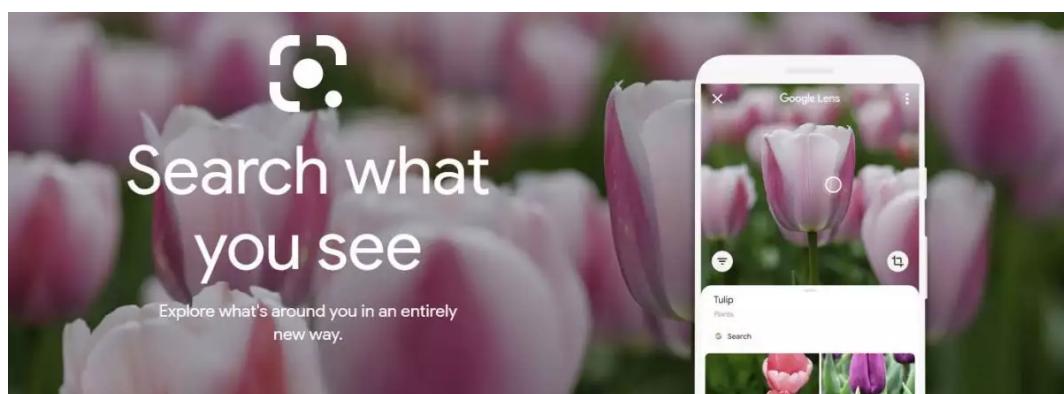
Tabel 1. Pembagian tugas dalam pembuatan Tugas Besar

BAB I

DESKRIPSI MASALAH

1.1. Abstraksi

Dalam era digital, jumlah gambar yang dihasilkan dan disimpan semakin meningkat dengan pesat, baik dalam konteks pribadi maupun profesional. Peningkatan ini mencakup berbagai jenis gambar, mulai dari foto pribadi, gambar medis, ilustrasi ilmiah, hingga gambar komersial. Terlepas dari keragaman sumber dan jenis gambar ini, sistem temu balik gambar (*image retrieval system*) menjadi sangat relevan dan penting dalam menghadapi tantangan ini. Dengan bantuan sistem temu balik gambar, pengguna dapat dengan mudah mencari, mengakses, dan mengelola koleksi gambar mereka. Sistem ini memungkinkan pengguna untuk menjelajahi informasi visual yang tersimpan di berbagai platform, baik itu dalam bentuk pencarian gambar pribadi, analisis gambar medis untuk diagnosis, pencarian ilustrasi ilmiah, hingga pencarian produk berdasarkan gambar komersial. Salah satu contoh penerapan sistem temu balik gambar yang mungkin kalian tahu adalah Google Lens.



Gambar 1. Contoh penerapan *information retrieval system* (Google Lens)

1.2. Content-Based Information Retrieval (CBIR)

Content-Based Image Retrieval (CBIR) adalah sebuah proses yang digunakan untuk mencari dan mengambil gambar berdasarkan kontennya. Proses ini dimulai dengan ekstraksi fitur-fitur penting dari gambar, seperti warna, tekstur, dan bentuk. 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.

BAB II

LANDASAN TEORI

2.1. CBIR dengan Parameter Warna

CBIR dengan parameter warna adalah melakukan Content Based Image Retrieval menggunakan warna-warna yang ada dalam suatu gambar. CBIR memanfaatkan representasi visual gambar untuk mengekstrak dan membandingkan fitur-fitur tertentu, dengan tujuan menemukan gambar yang mirip atau relevan. CBIR menggunakan karakteristik warna dari setiap pixel dalam gambar dan akan dibandingkan menggunakan histogram warna.

Histogram warna merupakan representasi grafis dari distribusi warna dalam sebuah gambar. Histogram warna akan memberikan frekuensi dari kemunculan RGB (red,green,blue) dari setiap piksel suatu gambar. Intensitas warna RGB akan dibagi dalam skala 0-255. Pada program ini Image akan diterima dan diubah menjadi matrix berisi RGB. Selanjutnya matrix akan dinormalisasi dari skala [0.255] menjadi [0,1]

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

Selanjutnya akan dicari nilai Cmax, Cmin dan Delta dari nilai RGB yang telah dinormalisasi. Delta merupakan jarak antara dua warna dalam suatu skema warna. Sedangkan Cmax adalah nilai komponen yang dominan dalam suatu skema warna. Cmin adalah nilai terkecil dari RGB dalam suatu skema warna.

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

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

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

Hasil Cmax, Cmin, Delta digunakan untuk melakukan perhitungan nilai HSV (Hue, Value,Saturation). Hue merupakan komponen yang menentukan tampilan visual warna. Hue digunakan untuk menentukan tipe warna dalam roda warna. Value adalah intensitas cahaya pada sebuah gambar. Nilai rendah value

akan menunjukkan warna yang lebih gelap sedangkan nilai yang tinggi akan menunjukkan warna yang cerah. Saturation adalah tingkat keabu-abuan suatu warna. Saturation akan membandingkan warna murni atau warna abu-abu. Saturation menentukan kemurnian suatu warna. Untuk menghitung nilai HSV dapat menggunakan perhitungan seperti berikut.

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

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

$$V = C_{max}$$

Setelah mendapatkan nilai HSV, histogram akan dibagi menjadi $n \times n$ blok untuk melakukan image processing. Langkah terakhir adalah menggunakan *cosine similarity* untuk membandingkan gambar input dengan dataset.

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

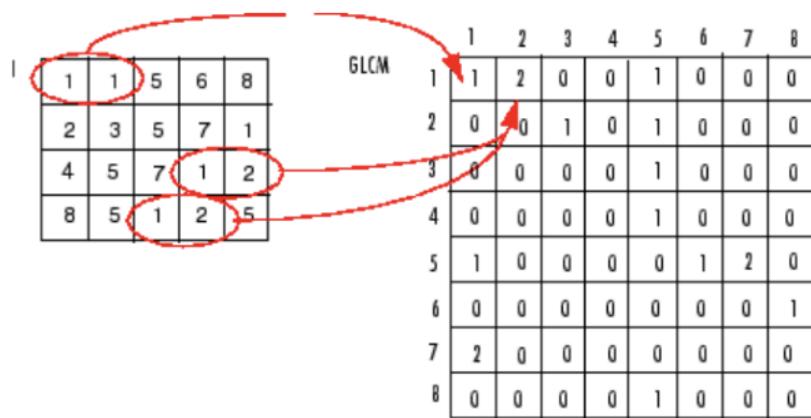
2.2. CBIR dengan Parameter Tekstur

CBIR dengan parameter tekstur adalah melakukan Content Based Image Retrieval menggunakan tekstur yang ada dalam suatu gambar. Parameter tekstur dalam CBIR mencakup karakteristik visual yang membedakan tekstur suatu objek atau area dalam gambar. CBIR dengan parameter tekstur pemrosesan pola-pola spesifik yang terdapat dalam struktur permukaan gambar. Gambar akan diekstraksi menjadi matrix dan diubah menjadi bentuk Grayscale dengan menggunakan langkah seperti berikut.

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

Karena citra grayscale berukuran 256 x 256 piksel, maka akan dibuat sebuah Gray Level Co-Occurrence Matrix. GLCM mengukur sejauh mana

pasangan intensitas piksel bersama-sama muncul dalam gambar, memberikan wawasan tentang pola distribusi intensitas di berbagai arah. Pada sistem ini akan digunakan GLCM dengan jarak 1 dan derajat 0.



Setelah didapatkan Co-Occurrence matrix , akan dibuat sebuah Symmetry Matrix dengan menjumlahkan, Co-Occurrence matrix dengan transpose-nya. Symmetry matrix tersebut akan di normalisasi menggunakan persamaan.

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

Dari normalisasi symmetry matrix, maka akan diekstraksi komponen-komponen texture. Pada sistem ini akan diekstraksi Contrast, Homogeneity, Entropy, Dissimilarity, ASM, Energy. Komponen Contrast akan mengukur perbedaan intensitas piksel yang berdekatan. Homogeneity mengukur keseragaman antara piksel-piksel yang berdekatan. Entropy mengukur tingkat ketidakpastian atau ketidakteraturan dalam distribusi intensitas piksel. Dissimilarity mengukur perbedaan antara pasangan intensitas piksel. ASM (Angular Second Moment) adalah energi kontras yang mengukur sejauh mana intensitas piksel dalam gambar tersebar dan juga keseragaman distribusi *gray level*. Terakhir adalah Energy yang merupakan distribusi energi dalam suatu intensitas piksel

1. *Contrast*

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

2. *Homogeneity*

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

3. *Entropy*

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

4. *Dissimilarity*

$$\frac{\sum_{i,j=1}^N P_{i,j} |i - j|}{N}$$

5. *ASM*

$$\frac{\sum_{i,j=1}^N P_{i,j}^2}{N}$$

6. *Energy*

$$\sqrt{ASM}$$

Setelah mendapatkan 6 komponen tersebut, akan dijadikan menjadi suatu vektor dan dicari Cosine Similarity nya menggunakan rumus seperti berikut.

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

2.3. Website

a. Front-End Development

Front-end adalah disiplin dalam pengembangan web yang bertanggung jawab untuk membangun tampilan suatu website. Front-End berhubungan dengan interaksi langsung oleh pengguna. Pengembangan front-end melibatkan pemanfaatan beragam teknologi dan bahasa pemrograman untuk menciptakan antarmuka pengguna yang responsif, menarik, dan berfungsi. Pada dasarnya, bahasa yang digunakan adalah HTML (Hypertext Markup Language) yang digunakan untuk membuat struktur dasar halaman web. Untuk membuat tampilan lebih menarik dapat digunakan CSS (Cascading Style Sheets) untuk mengatur tata letak secara visual, seperti margin, warna, font, padding dan lainnya sesuai dengan halaman web.

Pada sistem ini akan digunakan framework untuk memudahkan pengembangan web. Next.js adalah sebuah framework React yang dirancang untuk memudahkan pengembangan aplikasi web dengan menyediakan fitur-fitur seperti Server-Side Rendering (SSR), Static Site Generation (SSG), routing berbasis file system, dan banyak lagi. Dikembangkan oleh tim Vercel, Next.js memungkinkan pengembang untuk membuat aplikasi web yang responsif, efisien, dan mudah diatur.

b. Back-End Development

Back-end bertanggung jawab pada pemrosesan data, dan logika dalam bentuk sisi server. Bisa disebut backend merupakan “otak” dari suatu aplikasi. Di dalam Back-End terdapat istilah “Server-Side Logic” yang mengandung logika kode yang dijalankan di sisi server. Logika mencakup operasi-operasi yang diperlukan dan tidak ditunjukkan kepada *client* (front-end). Terdapat juga database management untuk mengelola data aplikasi. Pada Back-End juga akan menyediakan API untuk berkomunikasi dengan sisi front-end. Pada sistem ini, API digunakan untuk melakukan integrasi antar sistem NextJS (frontend) dan Flask (framework web python yang digunakan).

Pada sistem ini framework yang digunakan adalah Flask yang merupakan berbasis python. Flask dipilih karena kalkulasi CBIR menggunakan bahasa

python. Flask dirancang untuk menjadi sederhana, fleksibel, dan mudah dipahami, memungkinkan pengembang untuk membuat aplikasi web dengan cepat tanpa kompleksitas yang berlebihan.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1 Langkah-Langkah Pemecahan Masalah

Implementasi dari *Content Based Image Retrieval* pada proyek kami, berupa perangkat lunak berbasis laman yang dapat dioperasikan oleh pengguna melalui peramban masing-masing. *CBIR* memiliki dua metode yang dibedakan oleh parameter utamanya, metode pertama menggunakan warna sebagai parameter dan metode kedua menggunakan tekstur.

Pemecahan masalah kami awali dengan menentukan metode *CBIR* mana yang akan kami gunakan. Dalam kasus ini, kami menggunakan kedua metode *CBIR* sesuai dengan dasar teori di atas. Setelah metode *CBIR* telah ditentukan, kami mengimplementasikan teknik *CBIR* tersebut dalam *python* sebagai bahasa pemrograman utama. Berikut rincian dalam mengimplementasikan setiap metode *CBIR* dengan parameter-parameternya.

a. *CBIR* dengan parameter warna

1. Setiap gambar yang digunakan, harus diubah terlebih dahulu menjadi format HSV (Hue, Saturation, Value). Dibuat fungsi yang dapat mengubah gambar dengan format RGB (Red, Green, Blue) menjadi format HSV.
2. Dengan informasi HSV gambar, akan dibuat histogram sesuai dengan *channel*-nya, yaitu *hue*, *saturation*, dan *values*. Dibuat fungsi yang menghasilkan histogram dari gambar.
3. Histogram yang diperoleh dikonversi menjadi vektor yang nanti akan digunakan untuk menentukan indeks kemiripan menggunakan kaidah kemiripan cosinus.
4. Gambar dibagi menjadi 9 bagian dan setiap bagian memiliki vektor nya sendiri-sendiri.
5. Vektor gambar *database* juga akan dicari, dan akan dibandingkan dengan vektor *query* menggunakan kaidah kemiripan cosinus.

- b. CBIR dengan parameter tekstur**
1. Setiap gambar yang diterima akan diubah menjadi matrix BGR, dan akan diubah menjadi GrayScale.
 2. Selanjutnya akan dicari Co-Occurrence Matrix dengan ukuran 256 x 256, Co-Occurrence akan dicari menggunakan jarak 1 dan derajat 0.
 3. Matrix Co-Occurrence akan diubah menjadi Symmetry Matrix dengan cara menambahkan dengan transposenya.
 4. Setelah itu Symmetry Matrix akan di normalisasi dan dicari nilai Contrast, Homogeneity, Entropy, Dissimilarity, Energy, dan Angular Second Moment.
 5. Kami menambahkan Dissimilarity, Energy, dan Angular Second Moment karena ingin menambah tingkat akurasi dari pencarian berdasarkan tekstur.

3.2 Proses Pemetaan Masalah Menjadi Elemen-Elemen

Vektor dan matriks merupakan salah dua elemen dari disiplin ilmu aljabar geometri. Metode *CBIR* mengandalkan vektor untuk memperoleh indeks kemiripan antar gambar. Metode *CBIR* berparameter tekstur menggunakan matriks dalam metode perhitungannya. Indeks kemiripan gambar diperoleh dengan mencari nilai cosinus dari kedua buah vektor, vektor gambar query dan gambar *dataset*. Dalam arti lain, indeks kemiripan adalah nilai cosinus antara dua buah vektor. Nilai kosinus dapat dicari dengan melakukan operasi *dot product* vektor dan dinormalisasi atau dibagi dengan panjang kedua vektor.

3.3 Contoh Ilustrasi Kasus dan Penyelesaiannya

Seorang pengguna, Enji, adalah seorang mahasiswa IF yang memiliki galeri gambar kenangan yang terdiri dari ribuan foto yang diambilnya selama bertahun-tahun menjalani kuliah IF. Maria ingin mencari gambar-gambar tertentu dengan cepat tanpa harus menghabiskan waktu memeriksa satu per satu.

Maria dapat menggunakan sistem CBIR untuk melakukan pencarian gambar berdasarkan kemiripan konten visual dari suatu gambar. Misalnya, Maria ingin mencari gambar pemandangan kelas yang diambil pada saat mata kuliah Algeo. Dia dapat mengunggah contoh gambar sebagai *query* untuk CBIR.

BAB IV

IMPLEMENTASI DAN UJI COBA

4.1 Implementasi Program Utama

Berikut adalah implementasi program utama kami dalam bentuk *pseudocode*.

```
{START PROGRAM}

run()

input(image : .imgFormat)
input(dataset : folder)
result = []
for compare in dataset
    result ← display.append(getSimilarity(image,
compare)) {Mencari similarity dengan parameter yang ditentukan}
sort(result)
display(result)
stop()
```

4.2 Penjelasan Struktur Program Berdasarkan Spesifikasi

a. Content Based Image Retrieval (COLOR)

Tabel 4.1. Struktur program *CBIR* dengan parameter warna

Fungsi	Penjelasan
normBGRtoHSV(image matrix)	Menerima image berbentuk matrix BGR (blue,green,red) dan mengubahnya menjadi HSV (hue,saturation,value)
splitChannels(image matrix)	Mengubah matrix 3 dimensi HSV menjadi berbentuk tuple berisi 3 matrix 1 dimensi (HSV)
getHistogram(image matrix)	Membuat histogram dari HSV dan

	ditumpuk kesamping
getSimilarityIndeks(image1,image2)	Mendapatkan cosine similarity dari antara 2 matrix
get3X3Histograms(image matrix)	Membagi histogram menjadi 3x3 blok
get3x3Segments(image matrix)	Membagi menjadi blok 3x3
getVectorColor(path)	Mengembalikan matriks histogram dengan menerima <i>path</i> string
searchColor()	Mengembalikan nilai kemiripan <i>query</i> dan dataset dalam bentuk terurut <i>descending</i> (caching)
searchColorCache()	Mengembalikan nilai kemiripan CBIR warna dalam bentuk file <i>csv</i> (proses cache)
processColor(args)	Mengembalikan string <i>path</i> dan angka nilai kemiripan dalam bentuk persentase dari CBIR warna
colorParallel(base_vector, dataset_paths, parallel_processes)	Mengembalikan hasil CBIR warna dalam <i>array</i>
searchColorParallel(parallel_processes =60)	Mengembalikan nilai kemiripan secara terurut descending CBIR warna

b. Content Based Image Retrieval (TEXTURE)

Tabel 4.2. Struktur program *CBIR* dengan parameter tekstur

Fungsi	Penjelasan
getGrayScaleMatrix(image matrix)	Menerima image berbentuk matrix BGR (blue,green,red) dan mengubahnya menjadi Grayscale
getCoOccurrenceMatrix(image matrix)	Mengubah matrix grayscale dan menghasilkan matrix co-occurrence
getSymmetryMatrix(Co-occ matrix)	Menghasilkan matrix similarity dari menambahkan coOccurrence matrix dengan transposenya.
getNormalizedSymmetryMatrix(symmetry matrix)	Menormalisasi symmetry matrix

getContrast(image matrix)	Mendapatkan nilai contrast
getHomogeneity(image matrix)	Mendapatkan nilai homogeneity
getEntropy(image matrix)	Mendapatkan nilai entropy
getASM(image matrix)	Mendapatkan nilai Angular Second Moment
getEnergy(image matrix)	Mendapatkan nilai energy
getDissimilarity(image matrix)	Mendapatkan nilai dissimilarity
getVector(contrast, homogeneity, entropy, dissimilarity, asm, energy)	Mendapatkan nilai vektor dari komponen-komponen CBIR Texture
processTexture()	Melakukan proses CBIR texture dengan satu fungsi
textureParallel(base_vector, dataset_paths, parallel_processes)	Mengembalikan hasil CBIR tekstur dalam array
searchTextureParallel(parallel_processes=60)	Mengembalikan nilai kemiripan secara terurut <i>descending</i> CBIR tekstur
getVectorTexture(img)	Mengembalikan matriks histogram dengan menerima path string
extractFeature(args)	Mengambil 6 komponen: Contrast, Homogeneity, Entropy, ASM, Energy, Dissimilarity
searchTexture()	Mengembalikan nilai kemiripan query dan dataset dalam bentuk terurut <i>descending</i> (caching)
searchTextureCache()	Mengembalikan nilai kemiripan CBIR tekstur dalam bentuk file <i>csv</i> (proses cache)

c. Flask

Tabel 4.3. Struktur program *Flask*

Fungsi	Penjelasan
@app.route("api/upload") upload()	Endpoint untuk handling upload gambar dan upload dataset
@app.route("api/cbir")	Endpoint untuk melakukan

run()	perhitungan CBIR berdasarkan warna atau tekstur
@app.route("api/scrap") scrapImages()	Endpoint untuk melakukan image scraping
@app.route("api/download") downloadPDF()	Endpoint untuk melakukan download hasil perhitungan CBIR

d. Caching

Tabel 4.4. Struktur program *Caching*

Fungsi	Penjelasan
writeCacheColor()	Untuk mengembalikan nilai kemiripan berdasarkan CBIR warna pada proses caching
getCache(csv_path)	Mengembalikan data cache ke dalam tipe <i>dictionary</i>
writeCacheTexture()	Untuk mengembalikan nilai kemiripan berdasarkan CBIR tekstur pada proses caching

e. Download PDF

Tabel 4.5. Struktur program *Download PDF*

Fungsi	Penjelasan
writePDF(results)	Endpoint untuk handling upload gambar dan upload dataset

f. Web Scraping

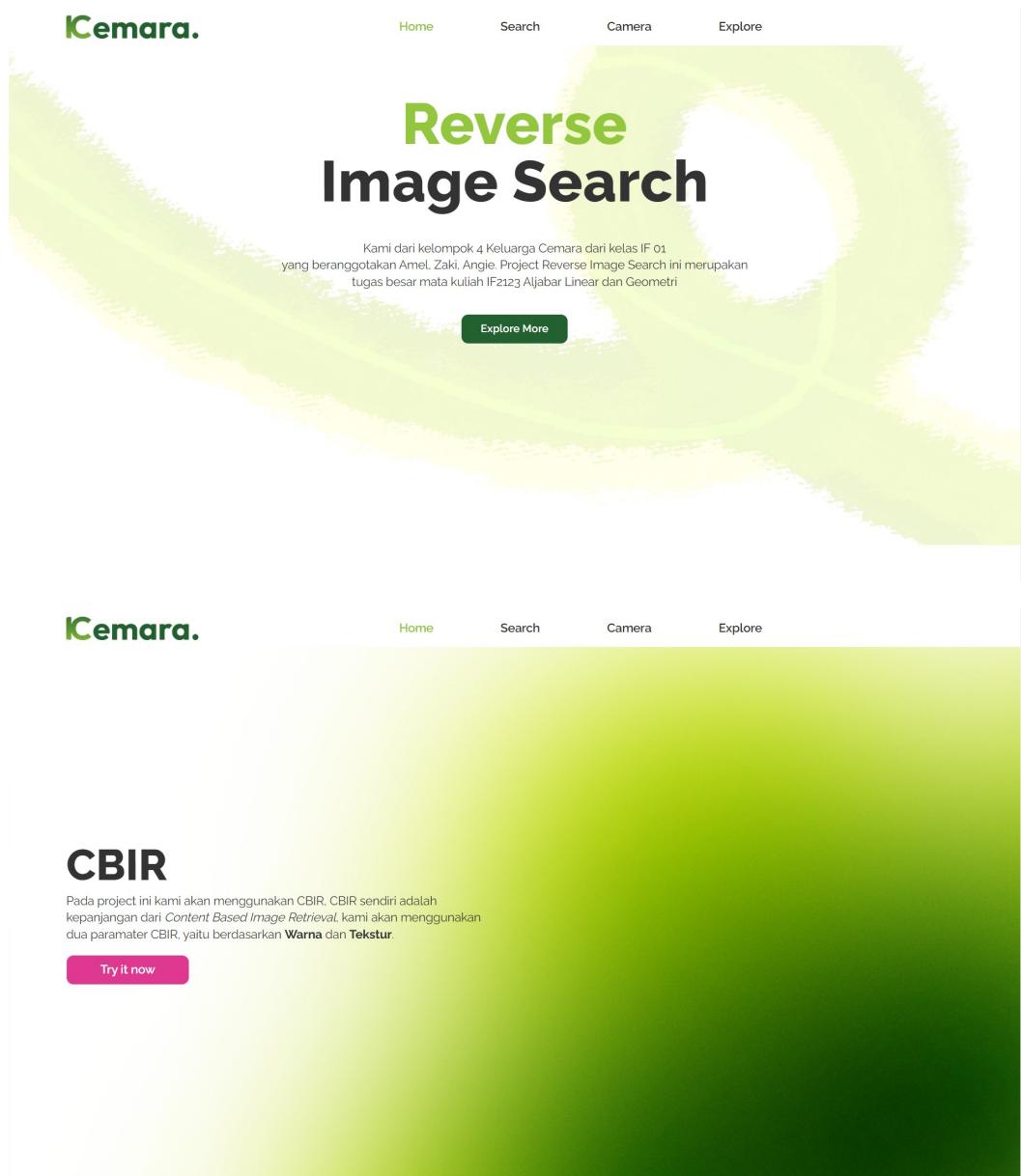
Tabel 4.6. Struktur program *Website Scraping*

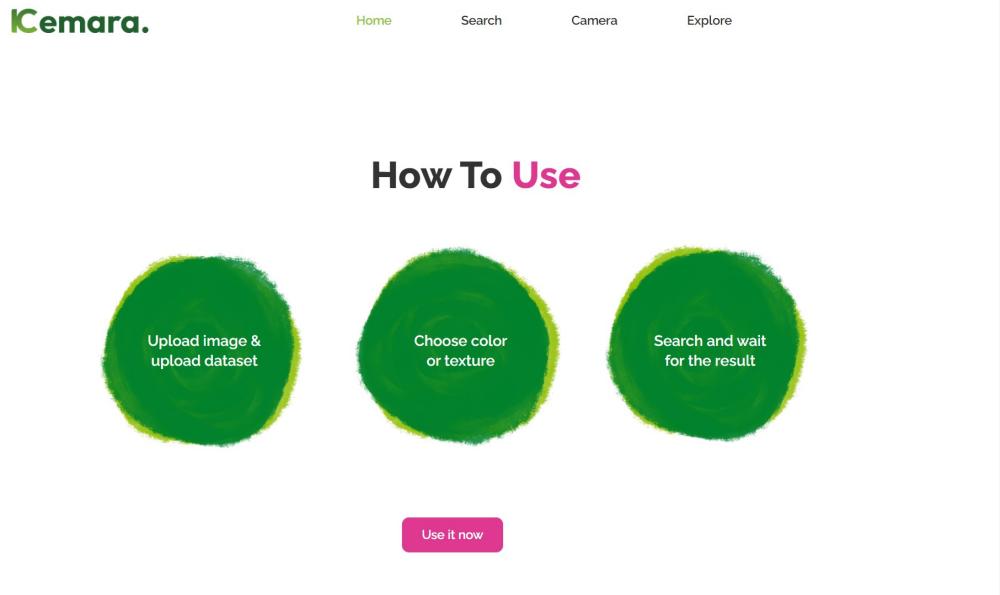
Fungsi	Penjelasan
create_folder_if_not_exists(folder_path)	Fungsi untuk membuat folder di directory apabila tidak ada
getRawHTML(url:str)	Fungsi untuk mengambil url dari sebuah website
scrapeImage(url:str,save_folder:str)	Fungsi untuk melakukan scraping image dari sebuah website

4.3 Tata Cara Penggunaan Program

a. Laman Home

Ketika memasuki aplikasi ICemara, pengguna akan berada pada laman *home*, yaitu laman utama. Pada laman tersebut, pengguna dapat membaca *overview* singkat tentang aplikasi kami, definisi singkat tentang *CBIR*, dan tata cara penggunaan aplikasi.





Gambar 4.1. Tampilan laman *home*

Terdapat *navbar* atau kolom navigasi yang berisikan 4 tombol utama yaitu, *home*, *search*, *camera*, dan *explore*.

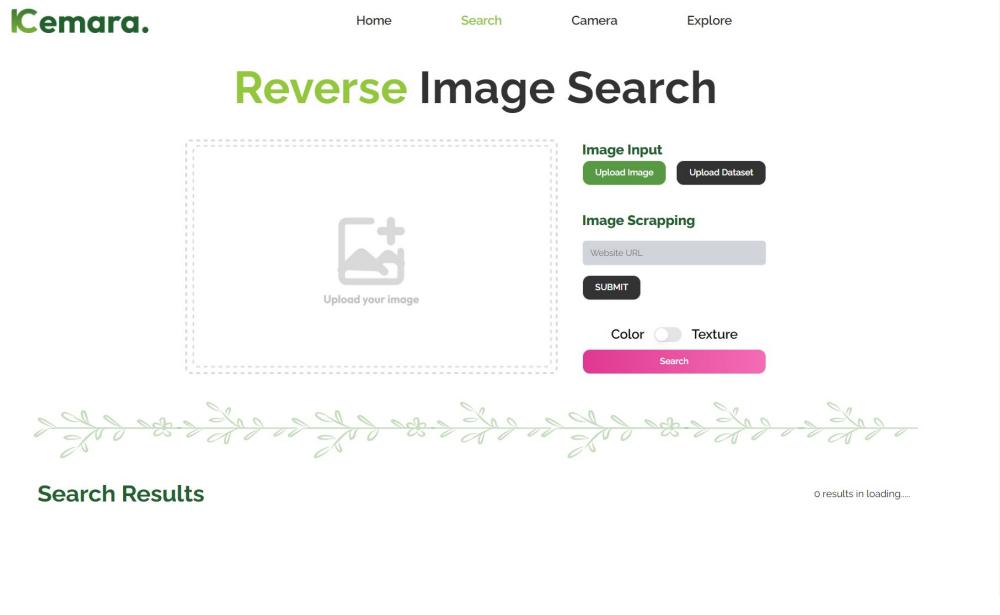


Gambar 4.2. Tampilan *Navbar*

b. Laman *Search*

Berisikan fitur utama dari ICemara antara lain, *CBIR* dengan parameter warna dan *CBIR* dengan parameter tekstur. Pengguna dapat menentukan parameter yang dikehendaki dengan menggunakan saklar yang terdapat pada laman.

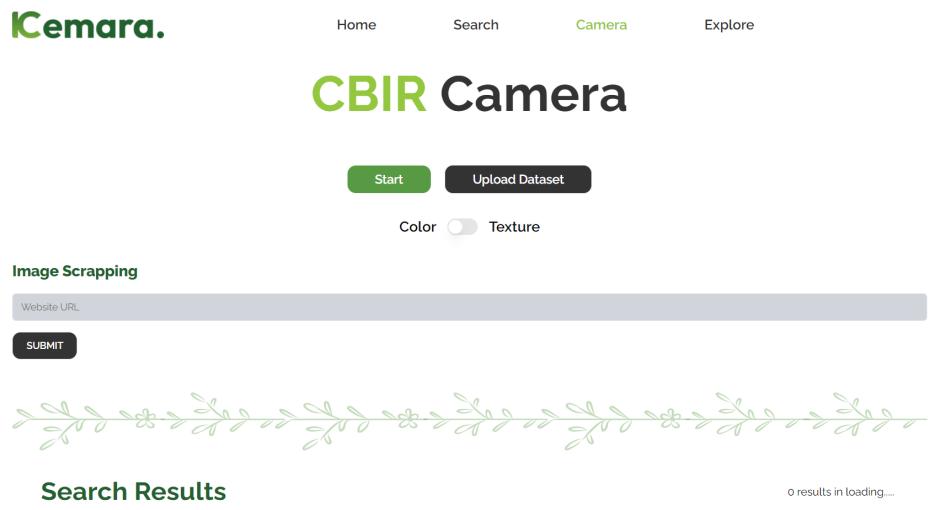
Pertama-tama, pengguna akan mengunggah foto yang akan digunakan dan selanjutnya, mengunggah *dataset* yang ingin digunakan, dapat berupa link sebuah website ataupun folder yang berisikan foto-foto. Kemudian pengguna akan mengklik tombol *search* dan menunggu hasil *CBIR* dari *query* pengguna.



Gambar 4.3. Laman *image search*

c. Laman *Camera*

Fitur *camera* memiliki fungsi yang sama dengan fitur *search*, hanya saja masukan gambar atau gambar query berasal dari kamera *device* pengguna, kamera akan mengambil gambar dalam suatu interval tertentu. Pengguna dapat menekan tombol *start*, mengupload *dataset*, dan menekan tombol *search* untuk menampilkan hasil pencarian.



Gambar 4.4 Laman *camera*

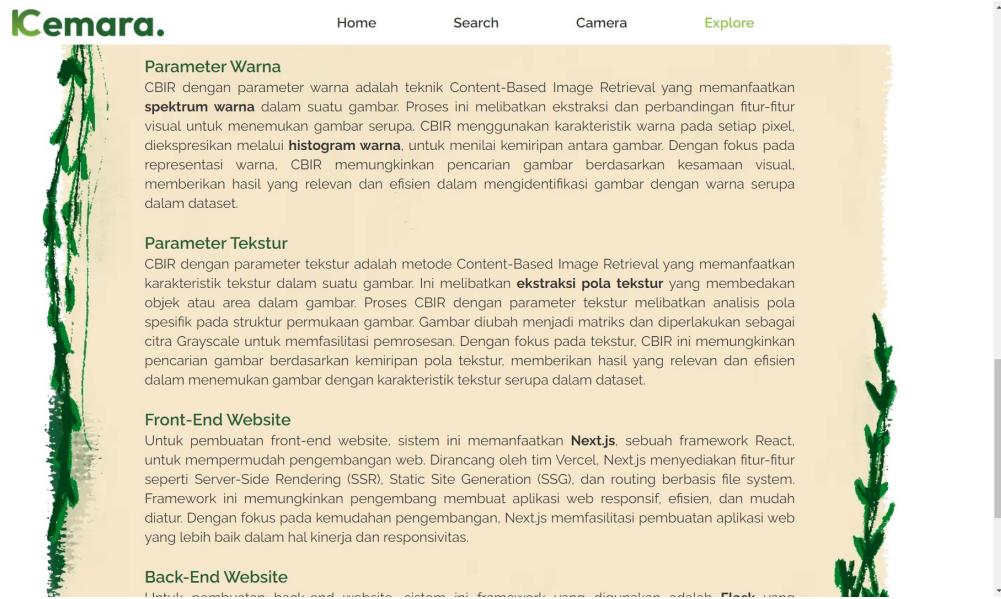
d. Laman Explore

Pada laman ini, pengguna dapat mengakses informasi tambahan terkait profil singkat *creator*, cara kerja komponen-komponen program beserta strukturnya, dan sepathah kata dari *creator*.

The screenshot shows the Cemara website with a navigation bar at the top: Home, Search, Camera, and Explore. Below the navigation is a pink header with the text "About Us". A text block says: "Kami dari kelompok 04, bernama Keluarga Cemara . Kami terdiri dari 3 orang yang semuanya dari kelas K01. Jujur banget, ini pertama kali kami ngulai website, tapi gapapa. Jadinya, kami banyak explore hal baru. So.. maap ya kalo website-nya jelek, *kami masih belajar hehe* 😊". Below this is a section titled "Meet the team" featuring three team members: Zaki Yudhistira, Amalia Putri, and Angelica Kierra, each with a photo and their student ID.

The screenshot shows the Cemara website with a navigation bar at the top: Home, Search, Camera, and Explore. Below the navigation is a pink header with the text "Explore". A text block says: "Ini adalah (•◦•) tugas besar ke-2 (•◦•) mata kuliah Aljabar Linier dan Geometri IF2123. Di website ini, pengguna dapat menjelajahi informasi visual yang tersimpan di berbagai platform, baik itu dalam bentuk pencarian gambar pribadi, analisis gambar medis untuk diagnosis, pencarian ilustrasi ilmiah hingga pencarian produk berdasarkan gambar komersial." Below this is a section titled "CBIR overview".

The screenshot shows the Cemara website with a navigation bar at the top: Home, Search, Camera, and Explore. Below the navigation is a pink header with the text "How our system works". A text block describes the Content-Based Image Retrieval (CBIR) process: "Content-Based Image Retrieval (CBIR) adalah metode pencarian gambar berdasarkan fitur-fiturnya. Proses ini melibatkan ekstraksi fitur seperti warna, tekstur, dan bentuk dari gambar, diwakili sebagai vektor numerik. Melalui algoritma pencocokan, CBIR membandingkan vektor-fitur gambar yang dicari dengan dataset, menghasilkan urutan gambar yang paling mirip secara visual. CBIR memungkinkan pengguna mengeksplorasi koleksi gambar secara efisien yang fokus pada kemiripan citra visual."

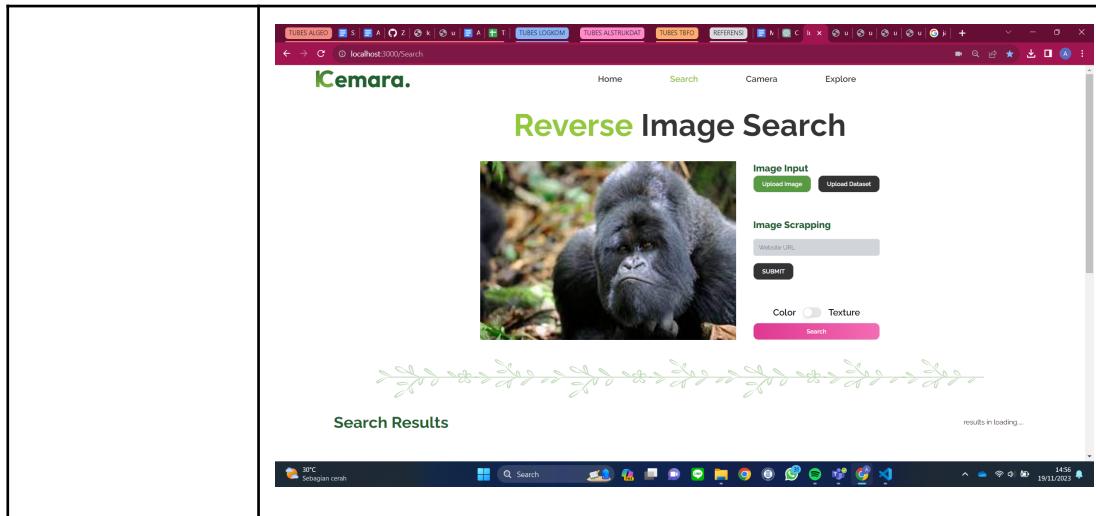


Gambar 4.5. Tampilan laman *about us*

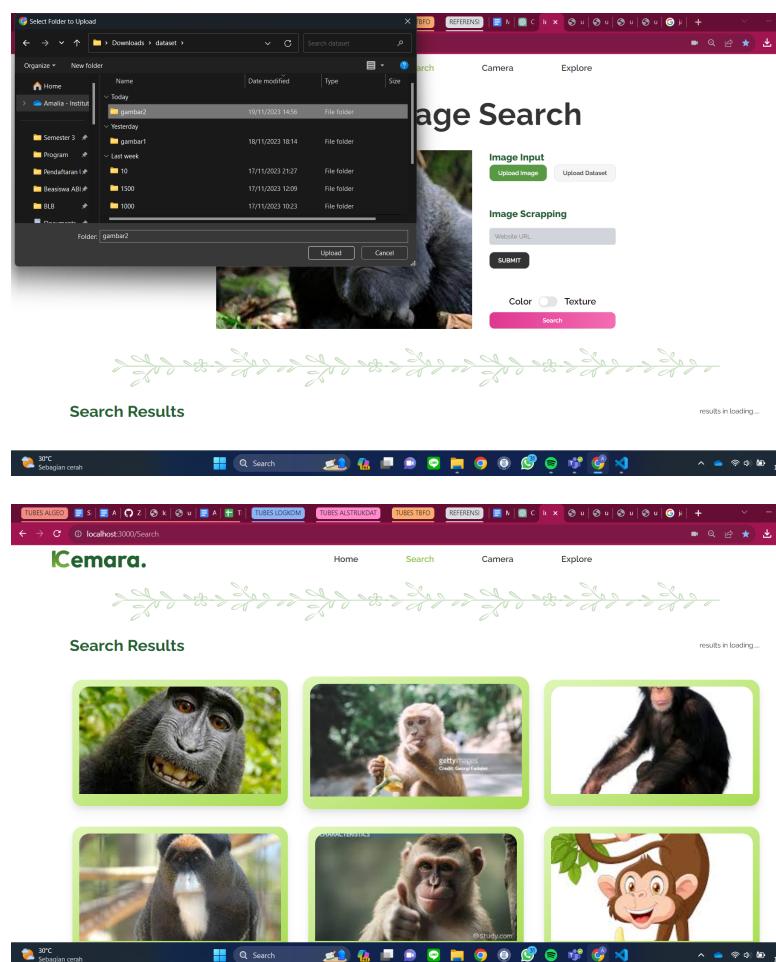
4.4 Hasil Pengujian

Tabel 4.4.1 Hasil Pengujian Komponen

Fitur	Hasil Pengujian
Upload Image	

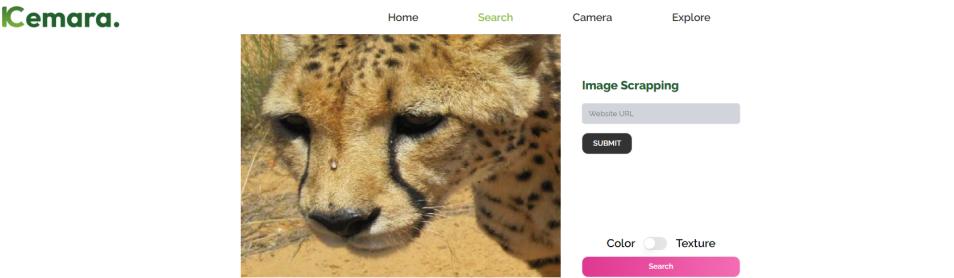


Upload Dataset



Toggle Button Warna-Tekstur	<p>a) Warna</p>  <p>b) Tekstur</p> 
Jumlah result dan waktu eksekusi	<p>500 results in 0.07 seconds</p>
Kumpulan Gambar (result) beserta tingkat kemiripannya	

Tabel 4.4.2 Hasil Pengujian CBIR

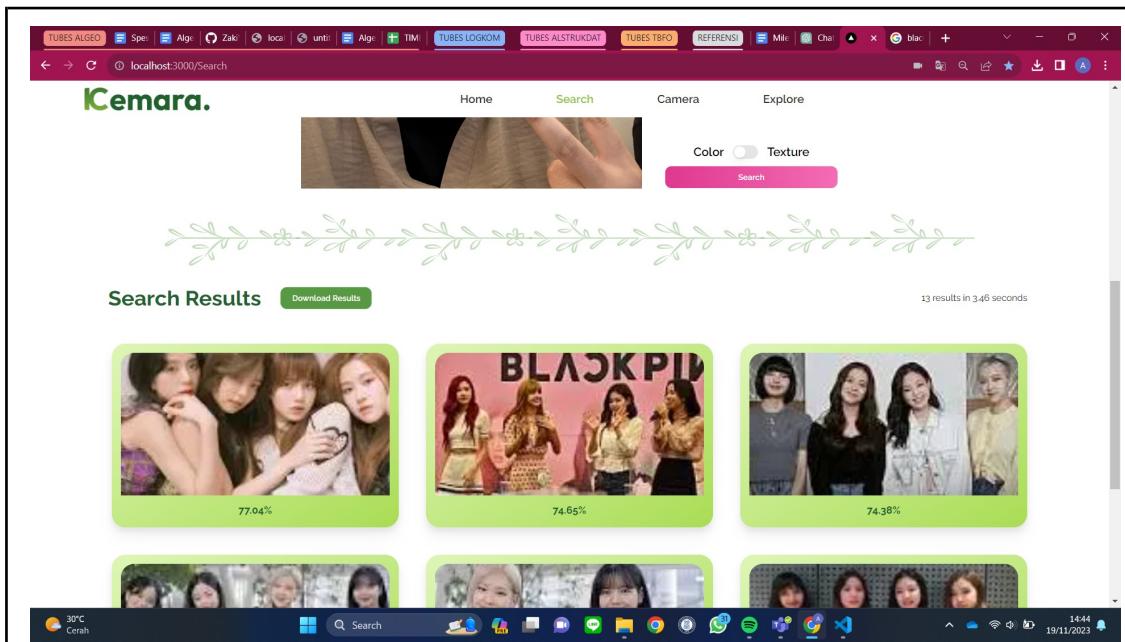
Hasil Pengujian CBIR Parameter Warna (Tanpa Caching)	
	

Hasil Pengujian CBIR Parameter Warna (Dengan Caching)

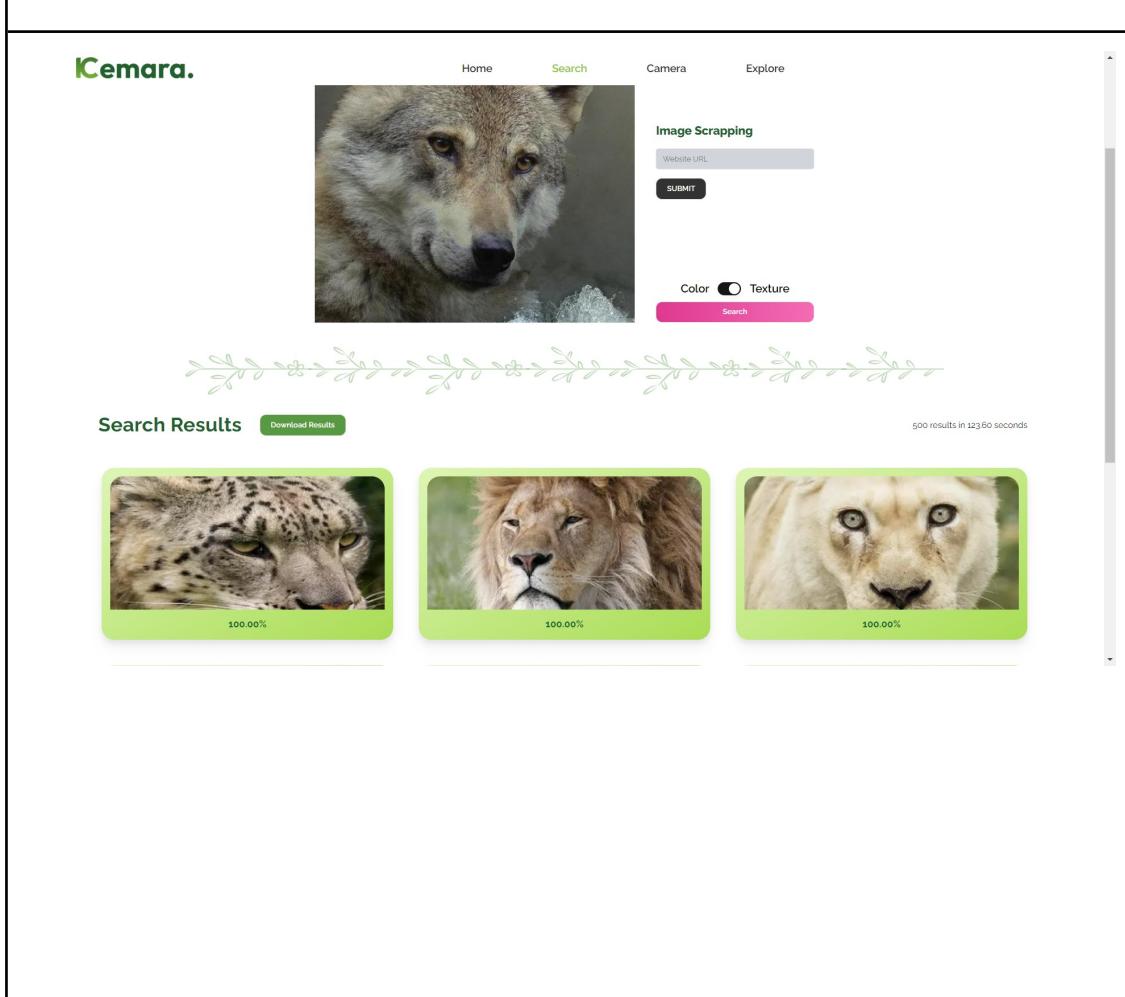
The screenshot shows the Cemara search interface. At the top, there is a navigation bar with 'Home', 'Search', 'Camera', and 'Explore' buttons. Below the navigation bar is a 'Image Input' section with 'Upload Image' and 'Upload Dataset' buttons. To the right of this is an 'Image Scrapping' section with a 'Website URL' input field and a 'SUBMIT' button. Below these sections is a decorative green vine banner. Underneath the banner, there is a 'Search Results' section with a 'Download Results' button. The results are displayed as three thumbnail images: a close-up of a wolf's face, another close-up of a wolf's face, and a close-up of a lion's face. A text overlay at the bottom right of the results area says '362 results in 0.17 seconds'.

Hasil Pengujian CBIR Parameter Warna (Dengan Image Scraping)

The screenshot shows the Cemara search interface. At the top, there is a browser-like header with various tabs and icons. Below the header is a navigation bar with 'Home', 'Search', 'Camera', and 'Explore' buttons. The main content area features an 'Image Input' section with 'Upload Image' and 'Upload Dataset' buttons. To the right is an 'Image Scrapping' section with a 'Website URL' input field containing 'https://www.google.com/search?q=black' and a 'SUBMIT' button. A success message 'Success: Image scraping was successful!' is displayed below the input field. Below these sections is a decorative green vine banner. Underneath the banner, there is a 'Search Results' section with a 'Download Results' button. The results are displayed as three thumbnail images: a close-up of a person wearing a hijab and glasses, another close-up of the same person, and a third image which appears to be a different person. A text overlay at the bottom right of the results area says 'Success: Image scraping was successful!'. The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators.



Hasil Pengujian CBIR Parameter Tekstur (Tanpa Caching)



Hasil Pengujian CBIR Parameter Tekstur (Dengan Caching)

The screenshot shows the ICemara search interface. At the top, there is a navigation bar with 'Home', 'Search' (highlighted in green), 'Camera', and 'Explore'. Below the navigation bar is a search input field containing a close-up image of a tiger's face. To the right of the input field are buttons for 'SUBMIT', 'Color' (unchecked), 'Texture' (checked), and a pink 'Search' button. A decorative green vine banner is positioned below the search bar. Underneath the banner, the text 'Search Results' is displayed next to a 'Download Results' button. On the far right, it says '500 results in 0.07 seconds'. The search results are presented in a grid of six images. Each image is a close-up of either a tiger or a jaguar, with a green overlay indicating a 100.00% match. The images are arranged in two rows of three.

Hasil Pengujian CBIR Parameter Tekstur (Image Scraping)

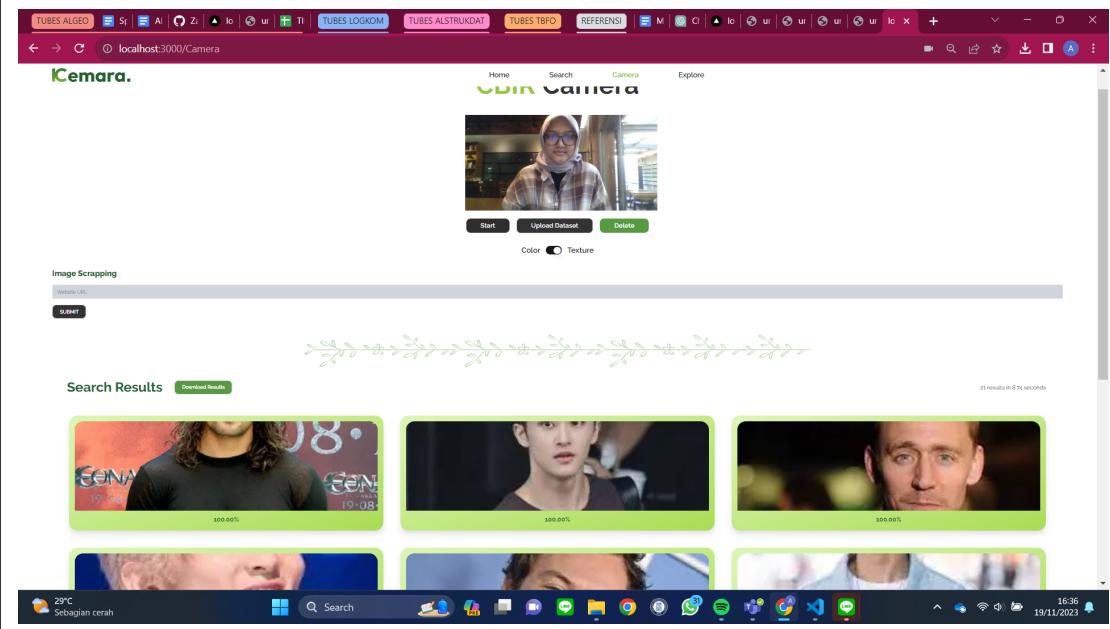
The screenshot shows the ICemara search interface. The top navigation bar includes 'Home', 'Search' (highlighted in green), 'Camera', and 'Explore'. Below the navigation bar is an 'Image Input' section with 'Upload Image' and 'Upload Dataset' buttons. The main area features a large image of a man with a mustache holding up a smartphone to take a picture of himself in a mirror. To the right of this image is a 'Image Scrapping' section. It displays a URL: 'https://www.google.com/search?q=jasonrr'. Below the URL is a 'SUBMIT' button and a message: 'Success: Image scraping was successful!'. At the bottom of the interface are 'Color' and 'Texture' buttons, and a pink 'Search' button. The bottom of the screen shows a Windows taskbar with various icons and the date/time: '14:46 19/11/2023'.

Hasil Pengujian CBIR Parameter Warna (Dengan Camera)

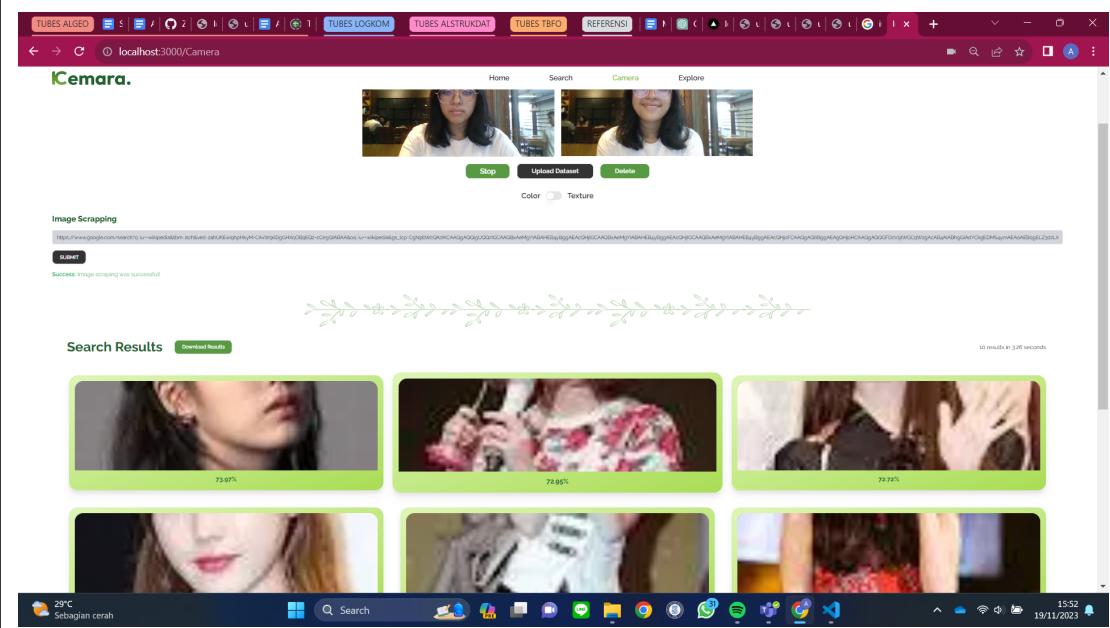
The screenshot shows the iCemara search interface. At the top, there is a navigation bar with tabs for Home, Search, Camera, and Explore. Below the navigation bar, a search bar contains the URL "localhost:3000/Search". A preview image shows a person's face. Below the preview are two toggle buttons: "Color" (which is selected) and "Texture". A pink "Search" button is located below the preview. A decorative green vine banner follows. The main section is titled "Search Results" and includes a "Download Results" button. It displays three search results, each showing a portrait of a man with a beard and a confidence score: 99.97%, 99.96%, and 99.95%. The desktop taskbar at the bottom shows various application icons and the date/time: 19/11/2023, 14:47.

The screenshot shows the iCemara search interface. At the top, there is a navigation bar with tabs for Home, Search, Camera, and Explore. Below the navigation bar, a search bar contains the URL "localhost:3000/Camera". A preview image shows a person's face. Below the preview are three buttons: "Stop", "Upload Dataset", and "Delete". A "Color" toggle button is also present. A decorative green vine banner follows. The main section is titled "Search Results" and includes a "Download results" button. It displays four search results, each showing a close-up of a monkey's face and a confidence score: 76.04%, 76.60%, 73.32%, and 73.32%. The desktop taskbar at the bottom shows various application icons and the date/time: 19/11/2023, 15:45.

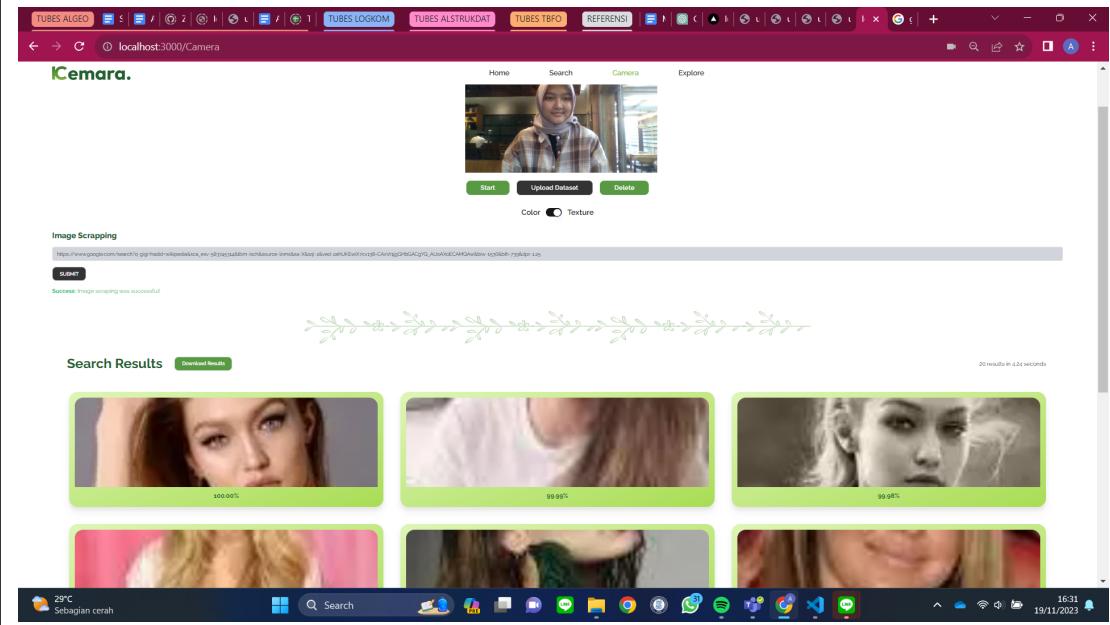
Hasil Pengujian CBIR Parameter Tekstur (Dengan Camera)



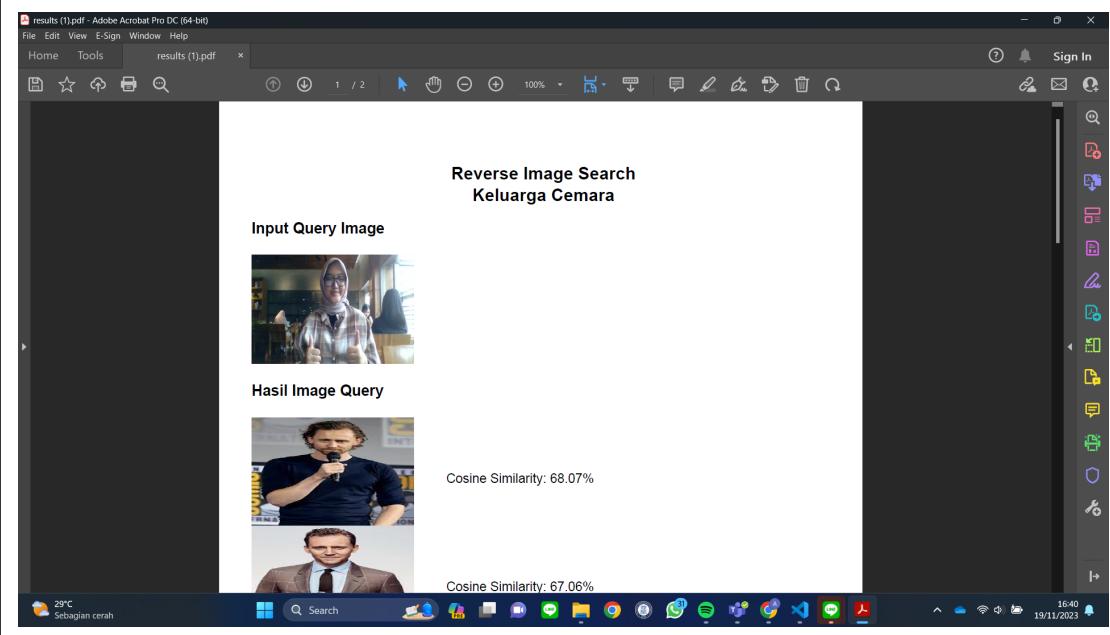
Hasil Pengujian CBIR Parameter Warna (Dengan Camera, Image Scrapping)

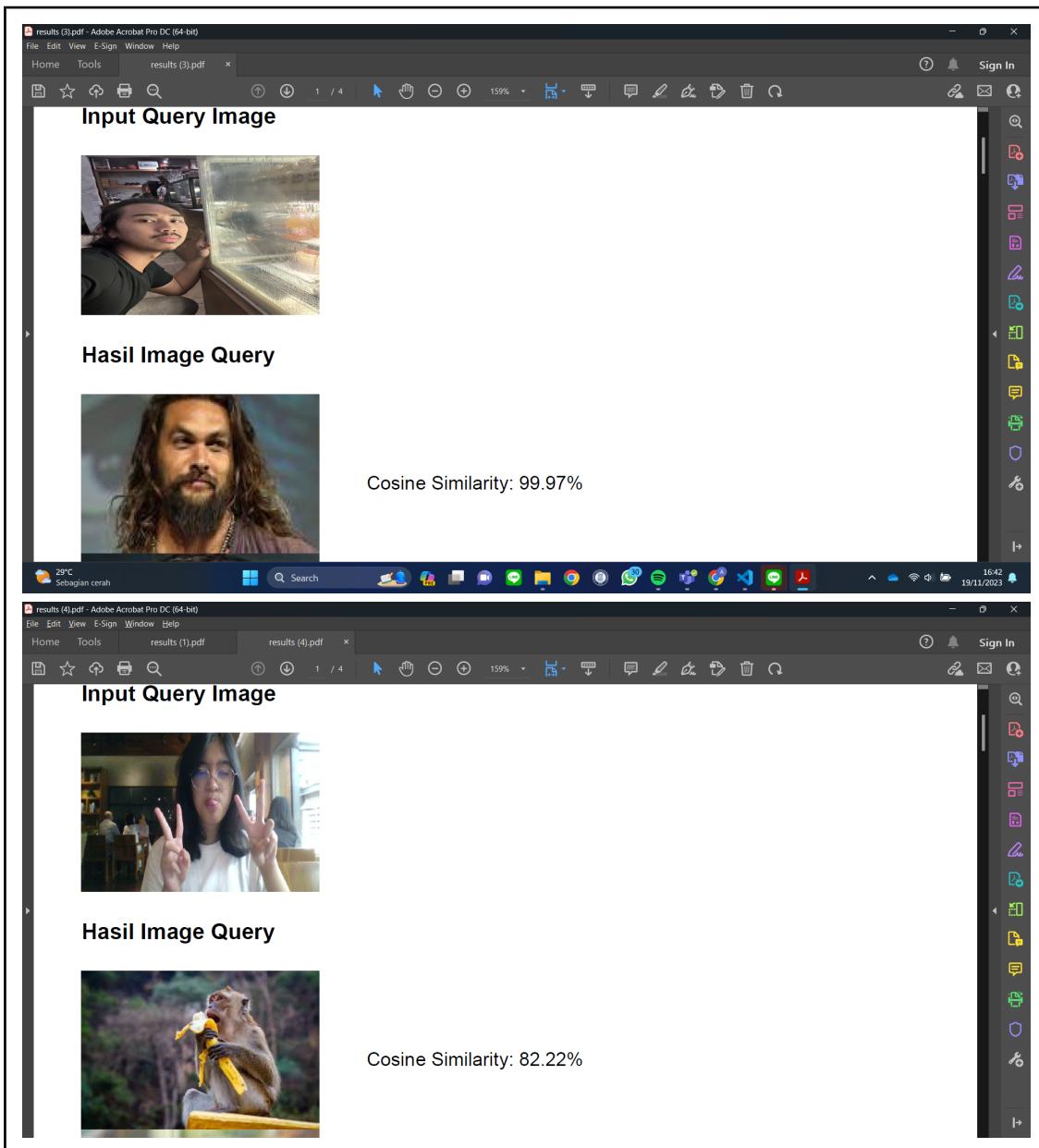


Hasil Pengujian CBIR Parameter Tekstur(Dengan Camera, Image Scrapping)



Hasil Download PDF





4.5 Analisis Desain Solusi Algoritma

Pada program CBIR warna dan tekstur yang terdistribusi pada laman *website*, hal pertama yang dilakukan pada desain solusi algoritma di program ini setelah menyusun desain UI/UX website ini, penulis mengelola algoritma CBIR warna dan tekstur dengan tahapan-tahapan yang telah dijelaskan pada bab II. Namun, dalam pembuatan algoritma dibutuhkan algoritma yang paling efektif dan efisien bagi program. Oleh karena itu, penulis berusaha menghindari penggunaan algoritma *nested loops* yang dapat menghambat waktu eksekusi pemrograman.

Selain itu, penulis menggunakan kalkulasi dengan Python yang memiliki banyak modul bawaan, seperti mengonversi *input* gambar menjadi matriks, membantu masukan dan keluaran dengan pengaturan *path*, pengelolaan *download* PDF, dan modul lainnya. Semua ini telah dilakukan untuk mempersingkat durasi eksekusi pemrograman.

Karena pengguna perlu memasukkan sebuah gambar sebagai *query* dan sebuah dataset berisi sejumlah gambar sebagai pembanding *query*, dibuatlah suatu *endpoint* untuk mengirimkan masukan tersebut ke folder lokal pengguna. Setelah itu, dalam pemrosesan CBIR warna-tekstur, dibuatlah suatu *endpoint* untuk mengirimkan nilai kemiripan berdasarkan konsep warna dan tekstur yang disaring (hanya yang bernilai $> 60\%$ saja yang muncul) dan diurutkan *descending* bersama *path* gambar tersebut dalam tipe data String.

Pada sistem kami untuk mempercepat pemrosesan data, kami menggunakan sistem caching dan juga multiprocessing. Pada saat *dataset* di unggah pertama kali, program akan menggunakan sistem multiprocessing dan saat csv sudah tertulis, program akan menggunakan sistem caching. Vector-vector dari *dataset* telah disimpan dalam bentuk csv sehingga, program hanya mengeksekusi fungsi *getSimilarityIndeks* saja.

Terdapat fitur kamera juga. Kamera akan mengambil gambar setiap interval 15 detik lalu menjadikan gambar tersebut sebagai *query*. Selanjutnya program akan melakukan pencarian CBIR berdasarkan *dataset* yang telah diupload. Kamera akan otomatis mengambil *query* baru untuk langsung dilakukan pencarian.

Selain mengunggah *dataset*, pengguna dapat memasukan *url* sebuah website. *Dataset* yang diambil merupakan gambar-gambar yang ada pada website tersebut. Gambar akan diambil menggunakan teknik ‘Image Scraping’. Jika proses *scraping* berhasil, pengguna dapat langsung menggunakan program CBIR seperti biasa.

Fitur terakhir yang ada mengunduh hasil pencarian CBIR dalam bentuk PDF. Isi dari PDF tersebut adalah gambar *query* dan hasil dataset dalam bentuk gambar dan hasil *cosine similarity*.

BAB V

KESIMPULAN, SARAN, DAN REFLEKSI

5.1. Kesimpulan

Dalam CBIR ini, perbandingan antara input gambar dan dataset dilakukan melalui transformasi gambar RGB menjadi histogram warna, memungkinkan analisis distribusi warna secara global maupun blok. Meskipun histogram warna memberikan informasi tentang frekuensi warna, namun tidak mampu mendeskripsikan objek spesifik atau posisi warna dalam gambar. Sebagai alternatif, CBIR dengan parameter tekstur menggunakan co-occurrence matrix untuk memproses tekstur dengan efisien, menghasilkan vektor berukuran lebih kecil.

Penggunaan program ini melibatkan langkah-langkah sederhana: memasukkan dataset gambar memilih opsi pencarian berdasarkan warna atau tekstur, dan menekan tombol search. Hasil pencarian disajikan dalam bentuk gambar yang diurutkan berdasarkan kemiripan, dengan persentase kemiripan ditampilkan. Dengan implementasi ini, program mampu memberikan informasi yang berguna, seperti jumlah gambar yang cocok dan waktu eksekusi program. Keseluruhan, CBIR dengan pendekatan warna dan tekstur melibatkan proses yang efektif dalam membandingkan dan mencari gambar, menawarkan solusi yang komprehensif dalam sistem temu balik gambar berbasis konten.

5.2. Saran

Saran dari penulis, sebaiknya untuk spesifikasi CBIR tekstur lebih diperjelas lagi karena akurasi tekstur belum dapat dipastikan benar jika hanya mengikuti spesifikasi yang ada sehingga menimbulkan banyak asumsi dan metode lain yang diimplementasikan pada program yang akhirnya membingungkan.

5.3. Komentar atau Tanggapan

Di kesempatan tugas besar ke-2 mata kuliah Aljabar Linier dan Geometri, kami mempelajari beberapa hal baru terkait cara kerja pemrograman terhadap konsep warna dan tekstur. Hal yang tak kalah penting juga, kami mempelajari

konsep *front-end* dan *back-end* website (*full-stack website*) yang mana sangat berkaitan dengan tujuan perkuliahan Teknik Informatika sehingga kedepannya dapat lebih siap untuk mengimplementasikan konsep ini dan menciptakan produk website yang lebih baik lagi.

Tugas besar ke-2 ini sangat menantang dan menuntut kami untuk mempelajari banyak hal baru yang sebelumnya belum pernah diajarkan di perkuliahan, serta memberikan pengalaman dan pandangan baru dalam pembuatan program layaknya *google lens* ini. Kami menjadi lebih tahu tentang cara kerja dan implementasi nyata algoritma *google lens* dalam skala kecil itu seperti apa. Semoga setelah terlaksanakannya tugas besar ke-2 ini, kami dapat lebih memahami dan mempelajarinya lagi lebih dalam, serta menambah pengetahuan dan meningkatkan keterampilan kami dalam pemrograman lebih baik lagi.

5.4. Refleksi

Dalam pembuatan tugas besar, kami belum terlalu dalam memahami tentang website sehingga di tengah pembuatan program, kami cukup kesulitan dalam mengintegrasikan *back-end* dan *front-end* website karena terkadang ide/struktur pada fitur yang ada *front-end* kurang dapat menyesuaikan *back-end* dan sebaliknya.

Saat memrogram kalkulasi tekstur, kami ragu dengan hasil dan kalkulasi kami tersebut karena tekstur tidak sejelas hasil warna yang dapat dilihat dari pewarnaan. Namun, tekstur tidak dapat diketahui dengan jelas dari kasat mata saja.

5.5. Ruang Perbaikan atau Pengembangan

Pada kedepannya sistem CBIR (*Content based image retrieval*) dapat lebih dikembangkan lagi. Contohnya pada penampilan website yang dapat dibuat secara responsif mengikuti media yang digunakan oleh pengguna, seperti handphone, tablet, dan laptop.

DAFTAR PUSTAKA

1. Beigi, Saba. “React-Webcam + TypeScript” (online).
(<https://dev.to/sababg/react-webcam-typescript-gh2>, diakses pada 19 November 2023).
2. Grinberg, Miguel. “How to create a react flask project” (online).
(<https://nextjs.org/docs>, diakses pada 19 November 2023).
3. Programming Historian. “Create APIs with python and flask” (online).
(<https://programminghistorian.org/en/lessons/creating-apis-with-python-and-flask>, diakses pada 19 November 2023).
4. “RGB to HSV color conversion” (online).
(<https://math.stackexchange.com/questions/556341/rgb-to-hsv-color-conversion-algorithm>, diakses pada 19 November 2023).
5. “Content-based image retrieval using color and texture fused features” (online).
(https://p2k.stekom.ac.id/ensiklopedia/Kaidah_Cramer, diakses pada 19 November 2023).
6. “Feature Extraction : Gray Level Co-occurrence Matrix (GLCM)” (online).
(<https://yunusmuhammad007.medium.com/feature-extraction-gray-level-co-occurrence-matrix-glcm-10c45b6d46a1>, diakses pada 19 November 2023).
7. “Next.js Documentation” (online).
(<https://blog.miguelgrinberg.com/post/how-to-create-a-react--flask-project>, diakses pada 19 November 2023).
8. “Flask Documentation” (online).
(<https://flask.palletsprojects.com/en/3.0.x/>, diakses pada 19 November 2023).
9. “Tailwindcss Documentation” (online).
(<https://tailwindcss.com/docs/installation>, diakses pada 19 November 2023).
10. “Shadcn UI Documentation” (online).
(<https://ui.shadcn.com/docs>, diakses pada 19 November 2023).

Repository

Link Repository dari Tugas Besar 02 IF 2123 Aljabar Linear dan Geometri kelompok 04 “Keluarga Cemara” adalah sebagai berikut.

<https://github.com/ZakiYudhistira/Algeo02-22035.git>

Youtube

Link video Youtube dari Tugas Besar 02 IF 2123 Aljabar Linear dan Geometri kelompok 04 “Keluarga Cemara” adalah sebagai berikut.

<https://youtu.be/uX26bgjo2tw>