

**ANALISIS HISTOGRAM BERDASARKAN BGR
UNTUK PENCARIAN CITRA**

Untuk Memenuhi Tugas Mata Kuliah Visi Komputer

Dosen Pengampu : Viska Mutiawani, B.IT, M.IT



Di Susun Oleh :

- | | |
|------------------------------------|------------------------|
| 1. Muhammad Ilhaam Ghiffari | (2108107010005) |
| 2. Aulia Muzhaffar | (2108107010033) |
| 3. Rachmat Fajar | (2108107010076) |
| 4. Rahmat Azrima | (2108107010077) |
| 5. Alifan Naufally Atha | (2108107010078) |
| 6. Muhammad Ghufraan | (2108107010080) |

**PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU
PENGETAHUAN ALAM UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH 2023**

BAB I

1. LATAR BELAKANG

Dalam pemrosesan citra, histogram adalah alat yang penting untuk menganalisis dan memahami distribusi intensitas piksel dalam sebuah gambar. Histogram digunakan dalam berbagai konteks, termasuk pelacakan objek dalam citra dengan algoritma CamShift, ekstraksi fitur seperti Histogram of Oriented Gradients (HOG) dan Scale-Invariant Feature Transform (SIFT), serta dalam pembuatan deskriptor untuk teknik bag-of-visual-words pada pencarian citra dan machine learning.

Histogram merupakan representasi statistik dari intensitas warna dalam gambar, dan bergantung pada model warna yang digunakan. Dalam kasus ruang warna RGB (Red, Green, Blue), setiap piksel dalam gambar direpresentasikan oleh tiga komponen/channel warna: merah (Red), hijau (Green), dan biru (Blue). Setiap komponen ini memiliki nilai dalam kisaran 0 hingga 255, di mana 0 mewakili ketiadaan warna dan 255 mewakili warna penuh. Misalnya, jika kita membuat histogram dengan 256 bin (satu bin untuk setiap nilai 0-255), kita akan menghitung berapa kali setiap nilai intensitas piksel muncul dalam gambar. Dengan menggunakan 256 bin ini, kita dapat melihat dengan jelas distribusi intensitas piksel.

Dalam konteks visualisasi komputer, model warna BGR (Blue, Green, Red) adalah salah satu sistem yang paling umum digunakan untuk merepresentasikan warna dalam citra digital. Model warna BGR adalah model warna berbasis RGB, di mana setiap piksel dalam gambar direpresentasikan oleh tiga komponen warna utama: Biru (Blue), Hijau (Green), dan Merah (Red). Setiap komponen ini memiliki nilai dalam kisaran 0 hingga 255.

Komponen Biru (Blue) Nilai dari 0 hingga 255 menggambarkan intensitas warna biru pada piksel tersebut. Nilai 0 menunjukkan ketiadaan warna biru, sementara nilai 255 menunjukkan warna biru penuh atau kecerahan maksimum. Komponen Hijau (Green) Seperti komponen biru, nilai dari 0 hingga 255 merepresentasikan tingkat kecerahan warna hijau pada piksel. Nilai 0 adalah ketiadaan warna hijau, sedangkan nilai 255 adalah hijau penuh. Komponen Merah (Red) Komponen merah juga memiliki rentang nilai yang sama, yaitu 0 hingga 255. Nilai 0 mengindikasikan ketiadaan warna merah, sementara nilai 255 adalah merah penuh.

Pada laporan kali ini kami membahas terkait proyek kami yaitu dengan judul analisis histogram berdasarkan bgr untuk pencarian citra. Yang mana kami memanfaatkan histogram dalam melakukan pencarian nilai warna berdasarkan pengelompokkan BGR. Nilai yang akan dimasukkan pada model yang di masukkan akan di analisis berdasarkan histogram lalu, akan di cari pada Kumpulan gambar dalam storage berdasarkan histogramnya dan akan di rangkingkan yang paling mirip dengan nilai bgr dengan penghitungan yang menggunakan histogram.

Menyelidiki persepsi warna pada orang dengan gangguan penglihatan warna melalui analisis wilayah warna yang membingungkan dalam gambar dan mengusulkan metode baru yang mengoreksi jumlah minimum wilayah warna menjadi warna optimal (J. Choi, 2019).

BAB II

1. TEORI VISI KOMPUTER

- Histogram

Histogram adalah representasi grafis yang menunjukkan distribusi frekuensi intensitas piksel dalam sebuah gambar. Dalam konteks pemrosesan citra, histogram memungkinkan kita untuk memahami bagaimana intensitas piksel didistribusikan di seluruh gambar.

Penggunaan dalam Algoritma CamShift CamShift (Continuously Adaptive Mean-Shift) adalah algoritma pelacakan objek yang memanfaatkan histogram untuk mencari objek dalam citra berdasarkan distribusi warnanya.

Ekstraksi Fitur Histogram sering digunakan dalam teknik ekstraksi fitur seperti HOG (Histogram of Oriented Gradients) dan SIFT (Scale-Invariant Feature Transform) yang membantu dalam deteksi dan pencocokan objek dalam gambar.

Deskriptor untuk Bag-of-Visual-Words Dalam teknik bag-of-visual-words, histogram digunakan sebagai deskriptor untuk mengklasifikasikan dan mencari gambar berdasarkan fitur visualnya.

- BGR atau RGB

Walaupun RGB (Red, Green, Blue) adalah model warna yang umum digunakan, dalam pemrosesan citra, khususnya dengan perangkat keras dan perangkat lunak tertentu, model warna BGR (Blue, Green, Red) lebih sering digunakan. Seperti namanya, urutan warna dalam model ini dimulai dengan biru, diikuti oleh hijau, dan merah.

Komponen Warna Setiap komponen warna (Biru, Hijau, dan Merah) memiliki rentang nilai dari 0 hingga 255. Nilai 0 menunjukkan ketiadaan warna, sementara nilai 255 menunjukkan intensitas warna maksimum.

Representasi Piksel Setiap piksel dalam gambar direpresentasikan oleh kombinasi tiga komponen warna ini.

Proyek ini berfokus pada analisis histogram berdasarkan komponen BGR untuk mencari gambar yang memiliki distribusi warna serupa. Dengan membandingkan histogram dari gambar target dengan histogram dari kumpulan gambar dalam storage, lalu dapat menemukan gambar yang paling mirip dengan gambar target berdasarkan distribusi warnanya.

Warna suatu benda tersusun oleh dua komponen warna, yaitu domain frekuensi dan domain spasial. Oleh karena itu, setiap warna memiliki panjang gelombang dan bentuk dalam jumlah frekuensi. Sensor warna TCS 3200 menghasilkan sinyal

keluaran dengan frekuensi berbeda berdasarkan warna yang dipindai melalui fotodioda sensor. Mengenali suatu benda berupa bentuk, warna dan variabel lain dari benda tersebut. Ekstraksi fitur diperoleh dari pola data dan histogram (Andrizal, 2018).

Salah satu cabang ilmu pengetahuan di bidang teknologi informasi yang berkembang pesat adalah computer vision. Salah satu peran penting computer vision dalam kehidupan sehari-hari adalah pemanfaatan computer vision. Hal ini dapat diterapkan dalam hal pengenalan wajah, deteksi objek, dan dapat diterapkan pada pengelompokan gambar berdasarkan urutan kemiripan gambar, kemampuan computer vision diterapkan untuk memudahkan pekerjaan manusia dalam memilih dari beberapa gambar untuk menemukan yang paling banyak. gambar serupa. Dalam penelitian ini dijelaskan proses pencarian kemiripan suatu gambar dengan gambar lainnya melalui beberapa tahapan alur penelitian, metode yang digunakan adalah dengan menggunakan nilai RGB yang telah dikonversi ke dalam skala abu-abu, kemudian dihitung jarak jarak euludian untuk menentukan nilai kedekatan suatu gambar sambil menghitung kinerja akurasi algoritma menggunakan matriks konfusi (Wahyu Wijaya Widiyanto, 2019).

- Metode ChiSquare

Uji Chi-square (Chi-squared) dalam konteks visi komputer dan pemrosesan gambar seringkali digunakan dalam perbandingan histogram untuk mengukur kesamaan antara dua histogram. Dalam konteks ini, histogram mewakili distribusi warna dalam gambar. Dengan membandingkan histogram dari dua gambar, kita dapat menilai sejauh mana kedua gambar tersebut serupa dalam hal distribusi warna mereka.

Metode Chi-squared digunakan untuk mengukur perbedaan antara histogram yang diamati dari sebuah gambar dan histogram target atau referensi. Semakin kecil nilai Chi-squared antara dua histogram, semakin serupa kedua gambar tersebut dalam hal distribusi warna.

Formula untuk perhitungan jarak Chi-squared antara dua histogram (H_1) dan (H_2) didefinisikan sebagai:

$$d(H_1, H_2) = \sum \frac{(H_1(i) - H_2(i))^2}{(H_1(i) + H_2(i))}$$

di mana:

- ($H_1(i)$) adalah nilai bin ke-i dari histogram (H_1)
- ($H_2(i)$) adalah nilai bin ke-i dari histogram (H_2)
- Penjumlahan dilakukan di seluruh bin histogram

Dalam aplikasi pencocokan gambar atau pencarian gambar berdasarkan konten (content-based image retrieval), metode Chi-squared dapat digunakan untuk membandingkan histogram warna gambar kueri dengan histogram warna dari setiap gambar dalam database. Gambar yang memiliki jarak Chi-squared terkecil dengan gambar kueri akan dianggap sebagai gambar yang paling mirip.

Penting untuk diingat bahwa metode ini hanya membandingkan distribusi warna dan tidak mempertimbangkan informasi spasial atau struktural dari gambar. Sebagai hasilnya, dua gambar dengan distribusi warna yang sama tetapi dengan konten yang berbeda mungkin dianggap serupa oleh metode ini.

- DropBox(penggunaan storage online)

Library Dropbox di Python adalah alat yang kuat yang memungkinkan pengembang untuk mengintegrasikan layanan penyimpanan awan Dropbox ke dalam aplikasi Python mereka. Dropbox adalah platform penyimpanan berbasis awan yang sangat populer yang memungkinkan pengguna untuk menyimpan, mengelola, dan berbagi file secara online. Dengan library Dropbox ini, pengembang dapat mengakses berbagai fitur yang ditawarkan oleh Dropbox, seperti mengunggah dan mengunduh file, mengelola berkas dan folder, berbagi file dengan pengguna lain, dan banyak lagi, semuanya melalui kode Python mereka sendiri. Penggunaan library ini memungkinkan integrasi yang kuat antara aplikasi Python dengan Dropbox, sehingga pengguna dapat dengan mudah mengelola dan berinteraksi dengan data mereka yang ada di Dropbox, membuat aplikasi yang lebih bermanfaat dan terhubung dengan layanan penyimpanan awan yang terpercaya. Library Dropbox ini juga menyediakan berbagai alat untuk otentikasi pengguna, manajemen izin, serta pengelolaan revisi file, menjadikannya pilihan yang kuat untuk pengembang yang ingin mengintegrasikan Dropbox ke dalam proyek mereka.

- ThinkIR

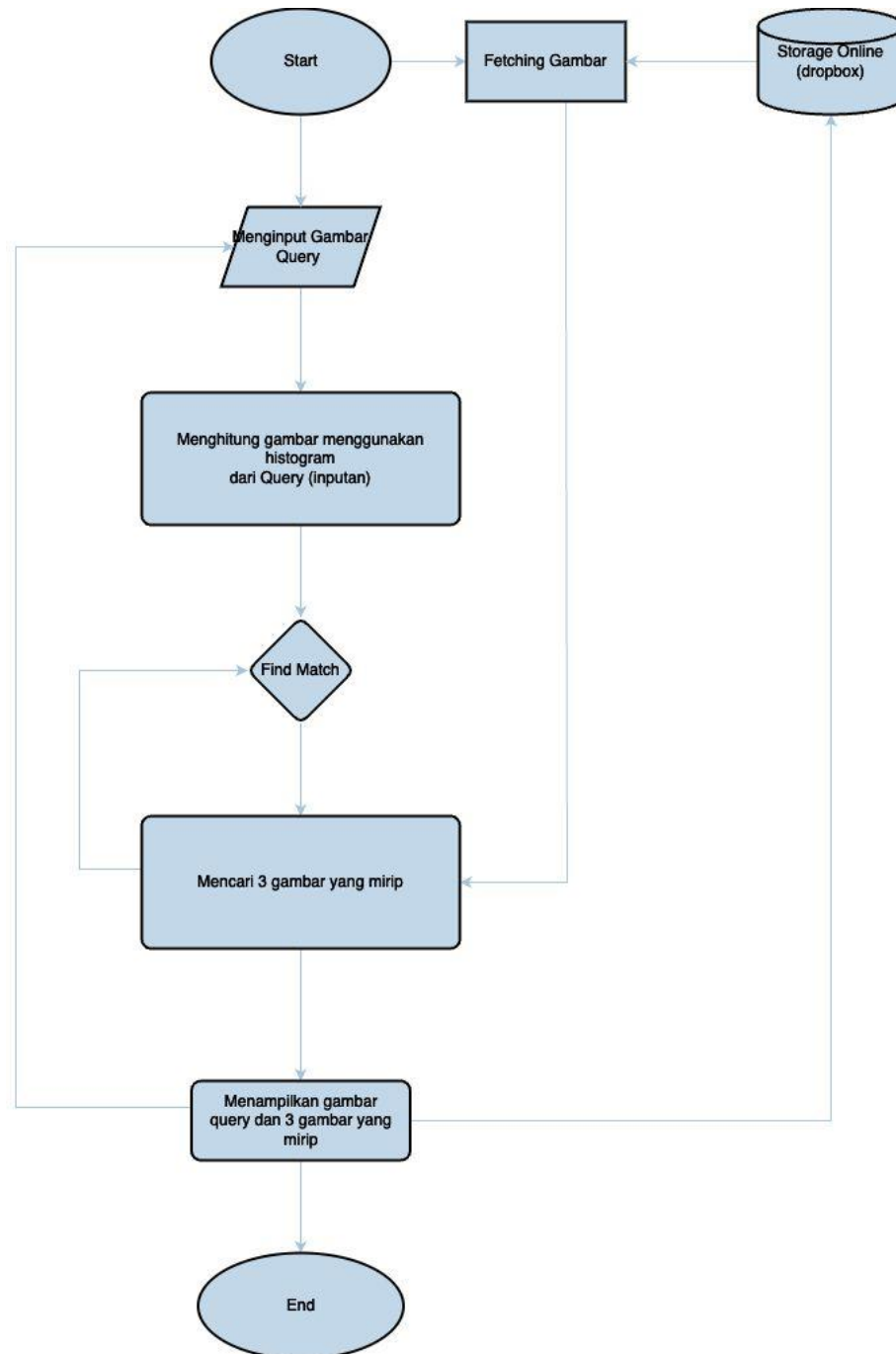
ThinkIR adalah pustaka Python yang dibuat untuk memfasilitasi pemikiran dan inferensi berbasis teks dalam bahasa alami. Pustaka ini dirancang untuk membantu pengembang dalam memproses dan menganalisis teks dengan cara yang lebih efisien, terutama dalam konteks pengolahan bahasa alami (Natural Language Processing atau NLP). ThinkIR menyediakan alat-alat untuk melakukan berbagai tugas seperti pemrosesan teks, analisis sentimen, ekstraksi entitas, dan pemodelan bahasa.

Salah satu fitur utama ThinkIR adalah integrasinya dengan spaCy, pustaka NLP yang kuat dalam bahasa Python. Ini memungkinkan pengguna untuk mengakses berbagai fitur pemrosesan teks yang ada di spaCy dengan lebih mudah. Dengan menggunakan ThinkIR, pengguna dapat melakukan tokenisasi, pemisahan kata, ekstraksi entitas, dan analisis sentimen dengan sedikit kode.

Selain itu, ThinkIR juga dapat digunakan untuk pemodelan bahasa dengan pendekatan yang lebih mudah dipahami dan diterapkan. Dengan kata lain, ThinkIR adalah alat yang sangat berguna bagi pengembang yang ingin menjalankan analisis teks dalam bahasa alami dengan mudah dan efisien dalam lingkungan Python. Ini dapat membantu mempercepat pengembangan aplikasi NLP dan membantu dalam pemahaman dan pengambilan keputusan berbasis teks.

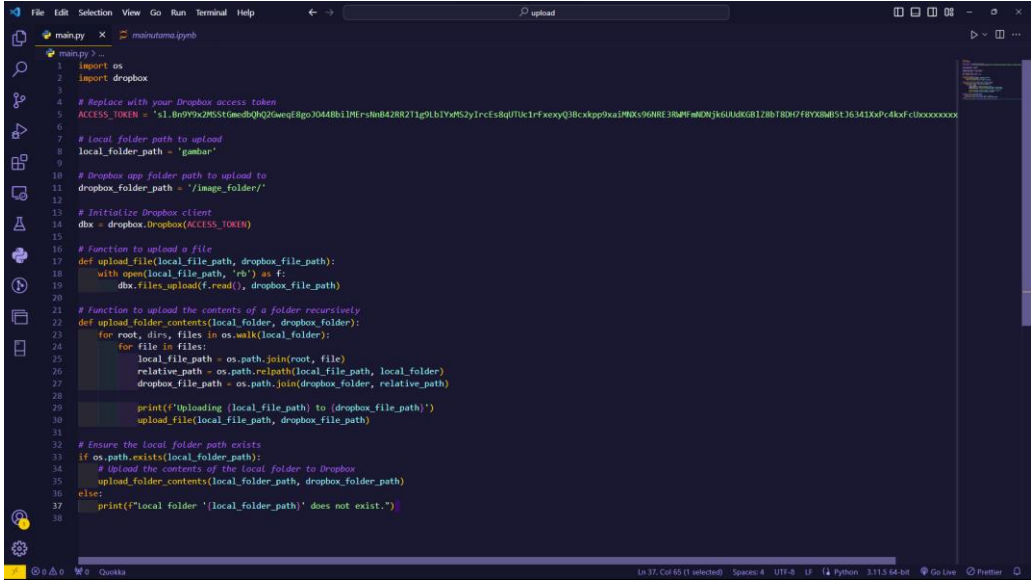
BAB III

1. LANGKAH ATAU ALUR



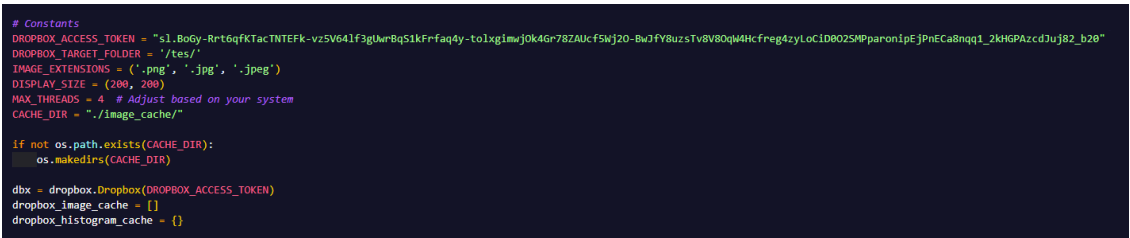
BAB IV

1. KODE PROGRAM



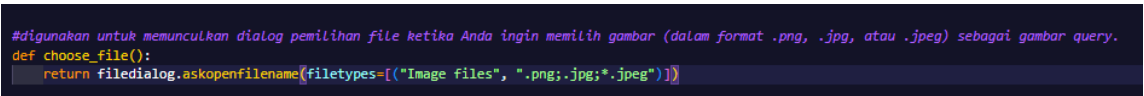
Interpretasi :

Jadi pada gambar diatas merupakan kode program yang bertujuan mengupload sebuah gambar dari folder local ke folder di sebuah storage online atau dropbox. Setelah itu, ada dua fungsi utama. Fungsi `upload_file` mengunggah satu file dari lokasi lokal ke Dropbox. Fungsi `upload_folder_contents` menelusuri semua file dalam folder lokal dan mengunggahnya satu per satu ke folder Dropbox yang ditentukan. Jika folder lokal tidak ada, kode akan mencetak pesan kesalahan. Jadi, kode ini essentially mempermudah kita dalam melakukan backup file ke Dropbox.



Interpretasi :

Pada gambar diatas berfungsi untuk mengatur konstanta dan variabel awal untuk koneksi Dropbox dan cache gambar. Ia juga membuat folder cache jika belum ada, dan menginisialisasi klien Dropbox untuk akses selanjutnya.



Interpretasi :

Pada gambar diatas berfungsi untuk memunculkan dialog pemilihan file ketika ingin memilih gambar dalam format .png, .jpg, atau .jpeg sebagai gambar query.

```
#Mengambil gambar dari folder Dropbox dan menghitung histogram untuk setiap gambar, kemudian menyimpannya dalam cache untuk digunakan nanti.
def fetch_files_from_dropbox(force_refresh=False, limit=50):
    global dropbox_image_cache, dropbox_histogram_cache
    if not force_refresh and dropbox_image_cache:
        return dropbox_image_cache, dropbox_histogram_cache

    file_data = []
    hist_data = {}
    cursor = None

    try:
        while True:
            if cursor:
                response = dbx.files_list_folder_continue(cursor)
            else:
                response = dbx.files_list_folder(DROPBOX_TARGET_FOLDER, limit=limit)

            for entry in response.entries:
                if isinstance(entry, dropbox.files.FileMetadata) and entry.name.endswith(IMAGE_EXTENSIONS):
                    cached_image_path = os.path.join(CACHE_DIR, entry.name)

                    if not os.path.exists(cached_image_path) or force_refresh:
                        metadata, res = dbx.files_download(path=entry.path_lower)
                        with open(cached_image_path, "wb") as f:
                            f.write(res.content)

                    file_data.append(cached_image_path)

                    if cached_image_path not in dropbox_histogram_cache:
                        image = cv2.imread(cached_image_path)
                        hist = compute_histogram(image)
                        hist_data[cached_image_path] = hist

            if not response.has_more:
                break

            cursor = response.cursor

    dropbox_image_cache = file_data
    dropbox_histogram_cache.update(hist_data)
    except Exception as e:
        status_var.set(f"Error fetching images: {str(e)}")

    return dropbox_image_cache, dropbox_histogram_cache
```

Interpretasi :

Pada gambar diatas berasal dari folder dropbox dan menghitung untuk setiap gambar, kemudian menyimpan dalam cache untuk digunakan nanti. Fungsi fetch_files_from_dropbox mengunduh gambar dari folder Dropbox yang ditentukan dan menghitung histogram mereka. Jika cache gambar sudah ada dan force_refresh tidak diaktifkan, fungsi ini akan mengembalikan data dari cache. Fungsi ini menggunakan pagination dengan variabel cursor untuk menangani banyak file. Semua gambar dan histogram yang diunduh disimpan dalam variabel global dropbox_image_cache dan dropbox_histogram_cache.

```
def main():
    # Inisialisasi variabel global
    global dropbox_image_cache, dropbox_histogram_cache

    # Mengambil gambar dari Dropbox dan menghitung histogram
    dropbox_image_cache, dropbox_histogram_cache = fetch_files_from_dropbox()

    # Menampilkan dialog pemilihan file
    file_path = askopenfilename()

    # Menghitung histogram dari gambar yang dipilih
    image = cv2.imread(file_path)
    hist = compute_histogram(image)

    # Menampilkan histogram
    show_hist(hist)
```

Interpretasi :

Pada gambar diatas bertujuan untuk menghitung histogram dari gambar.

```

#Mencari gambar yang mirip dengan gambar query berdasarkan perbandingan histogram.
def find_matches(query_image_path):
    query_image = cv2.imread(query_image_path)
    query_hist = compute_histogram(query_image)

    scores = {}
    file_path_list, hist_data = fetch_files_from_dropbox()

    #digunakan untuk menghitung skor kesamaan antara histogram dari gambar query
    #dengan histogram gambar lainnya yang diambil dari Dropbox. Skor ini mengukur sejauh mana histogram-histogram tersebut mirip
    def compute_score(file_path):
        other_hist = hist_data[file_path]
        score = cv2.compareHist(query_hist, other_hist, cv2.HISTCMP_CORREL)
        return file_path, score

    with ThreadPoolExecutor(max_workers=MAX_THREADS) as executor:
        results = executor.map(compute_score, file_path_list)
        for file_path, score in results:
            scores[file_path] = score

    top_matches = [item[0] for item in sorted(scores.items(), key=lambda x: x[1], reverse=True)[:3]]

    return top_matches

```

Interpretasi :

Pada gambar diatas bertujuan untuk mencari gambar yang mirip dengan gambar query berdasarkan perbandingan histogramnya. Lalu akan menghitung skor kesamaan antara histogram dari gambar query. Selanjutnya dengan histogram gambar lainnya yang diambil dari dropbox, skor ini mengukur sejauh mana histogram tersebut mirip.

```

#Digunakan untuk menampilkan gambar dan histogramnya dalam GUI.
def display_image_and_histogram(img_source, row, col, frame, title):
    global image_references
    img_np = cv2.imread(img_source)
    img_pil = Image.fromarray(cv2.cvtColor(img_np, cv2.COLOR_BGR2RGB)).resize(DISPLAY_SIZE)
    ttk.Label(frame, text=title, font=("Arial", 12, "bold")).grid(row=0, column=col, padx=10, pady=10)
    img_photo = ImageTk.PhotoImage(img_pil)
    ttk.Label(frame, image=img_photo).grid(row=1, column=col, padx=10, pady=10)
    image_references.append(img_photo)

    # Save the image to a file
    img_save_path = f"saved_image_{col}.png"
    cv2.imwrite(img_save_path, img_np)

    # Display smaller histogram below the image
    hist_image = Image.open(get_histogram_image(img_np))
    hist_photo = ImageTk.PhotoImage(hist_image)
    ttk.Label(frame, image=hist_photo).grid(row=2, column=col, padx=10, pady=10)
    image_references.append(hist_photo)

    return img_save_path # the simpan gambar

```

Interpretasi :

Pada gambar diatas bertujuan untuk menampilkan gambar dan histogram dalam GUI. Fungsi display_image_and_histogram menampilkan gambar dan histogramnya di GUI Tkinter. Ia mengambil path gambar, posisi baris dan kolom di grid, frame tempat elemen ditempatkan, dan judul. Gambar dibaca dan diubah ke format PIL untuk ditampilkan. Histogram juga di-render sebagai gambar kecil dan ditempatkan di bawah gambar utama. Kedua gambar tersebut disimpan dalam list global image_references untuk mencegah garbage collection.

```

#Tkinter GUI: Ini adalah bagian yang menginisialisasi antarmuka pengguna Tkinter.
# Ini mencakup tombol untuk memilih gambar query dan area di mana gambar dan histogram akan ditampilkan.
root = Tk()
root.title("Image Matching App")
root.geometry("900x750")

style = ttk.Style(root)
style.configure("TFrame", background="#e1e1e1")
style.configure("TButton", padding=(10, 5), font=("Arial", 12))
style.configure("TLabel", background="#e1e1e1", font=("Arial", 10))

ttk.Button(root, text="Select Query Image", command=process_and_display).pack(pady=20)

result_frame = ttk.Frame(root)
result_frame.pack(pady=20, padx=20, fill="both", expand=True)
print(result_frame)

status_var = StringVar()
status_var.set(f"Fetching images from Dropbox folder: {DROPTBOX_TARGET_FOLDER}")
status_bar = ttk.Label(root, textvariable=status_var, relief="sunken", anchor="w")
status_bar.pack(side=BOTTOM, fill="x")

image_references = []

root.mainloop()

```

Interpretasi :

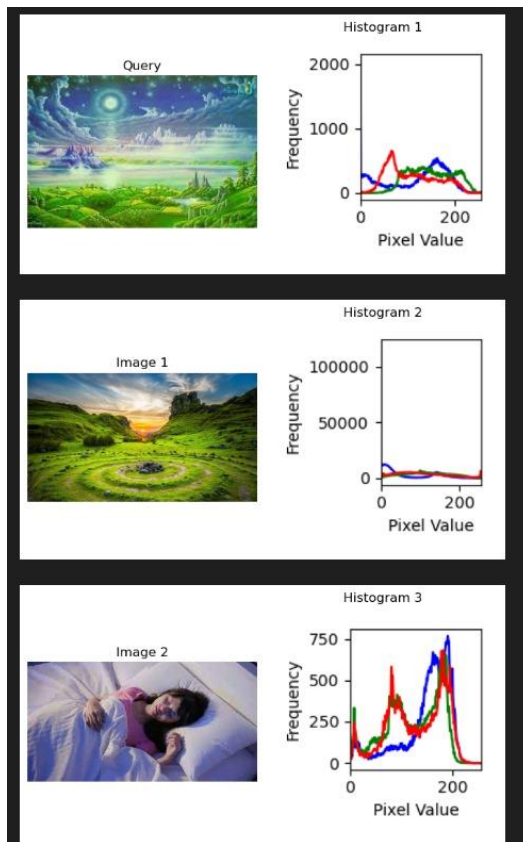
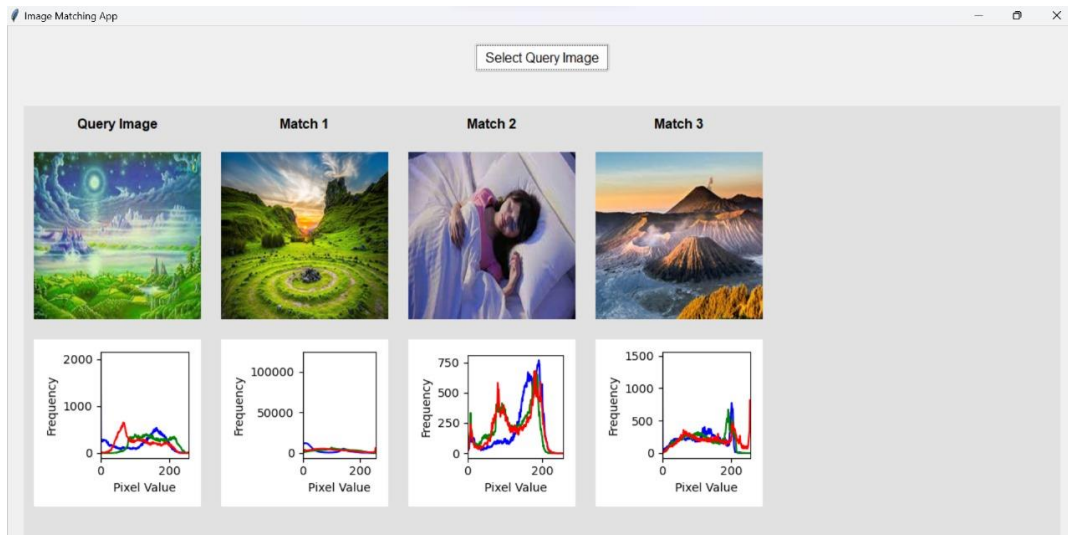
Pada gambar diatas berfungsi untuk menginisialisasi antarmuka pengguna tkinter. Ini mencakup tombol untuk memilih gambar query dan area dimana gambar dan histogram akan ditampilkan.

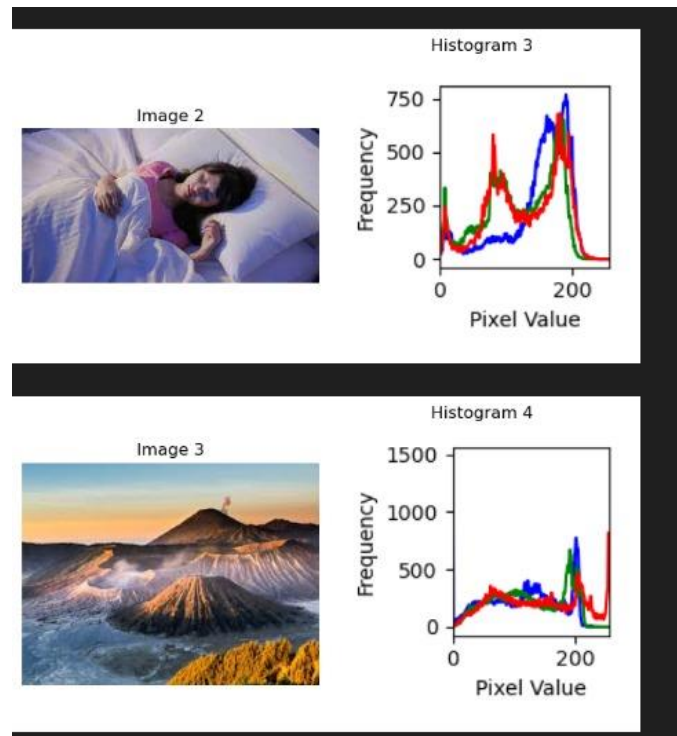
Bagian penting dari kode program ini yaitu :

- Constants: Ini adalah bagian yang mendefinisikan konstanta seperti token akses Dropbox, folder target Dropbox, ekstensi gambar yang diizinkan, ukuran tampilan, dan lain-lain.
- fetch_files_from_dropbox: Ini adalah fungsi yang mengambil gambar dari folder Dropbox yang telah ditentukan. Fungsi ini juga menghitung histogram untuk setiap gambar dan menyimpannya dalam cache untuk digunakan nanti.
- compute_histogram: Fungsi ini menghitung histogram dari gambar yang diberikan menggunakan OpenCV.
- find_matches: Fungsi ini mencari gambar yang mirip dengan gambar query berdasarkan perbandingan histogram. Ini adalah inti dari aplikasi ini.
- plot_histogram: Fungsi ini digunakan untuk menampilkan plot histogram.
- get_histogram_image: Fungsi ini menghasilkan gambar histogram yang lebih kecil dari histogram gambar yang diberikan.
- display_image_and_histogram: Fungsi ini digunakan untuk menampilkan gambar dan histogramnya dalam GUI.
- process_and_display: Fungsi ini dipanggil saat pengguna memilih gambar query. Ini akan menampilkan gambar query dan gambar yang mirip dalam GUI.
- Tkinter GUI: Ini adalah bagian yang menginisialisasi antarmuka pengguna Tkinter. Ini mencakup tombol untuk memilih gambar query dan area di mana gambar dan histogram akan ditampilkan.
- Fungsi process_and_display digunakan untuk memproses gambar query yang dipilih pengguna dan menampilkan gambar yang mirip di GUI.

Jadi dalam menguji source code nya kita menggunakan 80 data untuk testing yang terdapat pada storage online. Jadi saat user mencoba query input gambar, user bebas untuk menginput gambar apapun. Ditahap ini kita sudah mencoba mengumpulkan 3000 kumpulan data dan hasil waktu yang didapatkan adalah tidak berhasil (Not responded). Dan pada akhirnya kita menggunakan 80 data saja sebagai sample untuk menjalankan Project ini.

2. TAMPILAN HASIL KODE PROGRAM



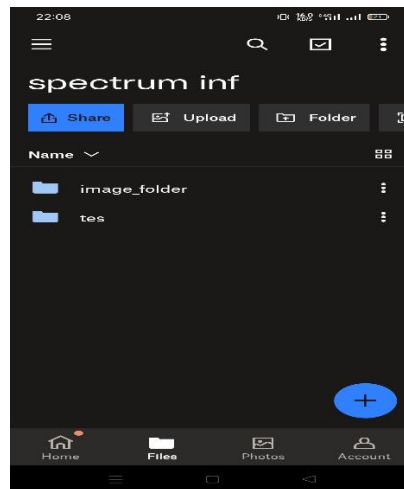


Interpretasi :

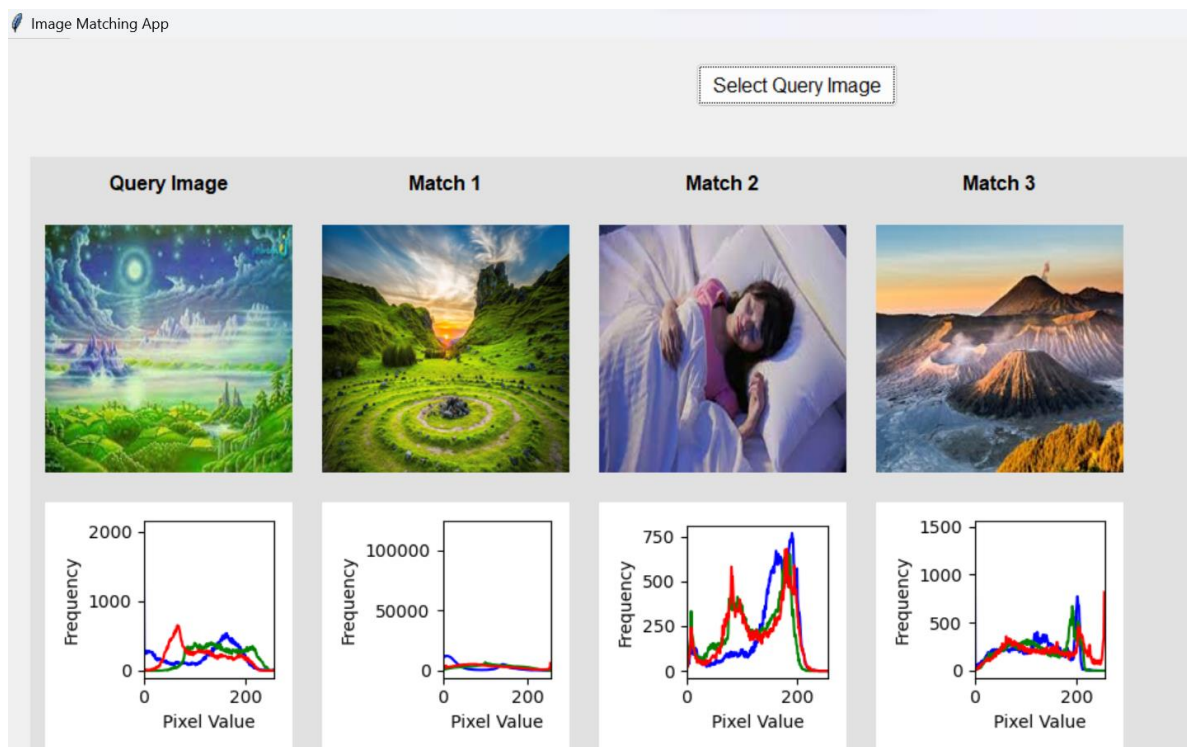
Dari hasil query yang kita masukkan, kami mendapatkan informasi tentang kemiripan warna terdekat dalam bentuk histogram. Gambar yang dibandingkan tersimpan secara online di Dropbox, dan akses menggunakan token. Hasil perangkingan tersebut mengidentifikasi tiga gambar terdekat berdasarkan kemiripan warna, dengan gambar utama yang disebutkan dalam judul query. Histogram ini menunjukkan frekuensi warna, baik dalam format BGR maupun RGB, dengan sumbu x mewakili nilai piksel dalam gambar dan sumbu y menunjukkan frekuensi kemunculan warna tersebut pada gambar.

3. EVALUASI

- Perbandingan penggunaan multi trade
Pada project kali ini kami menggunakan multithreads untuk mempercepat proses pada project kami. Kami menggunakan nilai = 4 . Sebelumnya kami menggunakan multithreads dengan nilai = 5 tapi kebetulan di karenakan device yang kurang support hasil semakin menjadi buruk.
- Waktu pemrosesan penggunaan database'test' dan 'image_folder'
Kami memiliki 2 databases yang mana database test di gunakan dalam pelaksanaan persentasi, pada databases test memiliki 80 gambar, yang nanti akan di bandingkan oleh gambar query, memeiliki durasi 30 detik. Sedangkan pada database image_folder memiliki total gambar hingga 300, namun dikarenakan device yang kurang support pada gui akan menampilkan not responding. (tambah gambar)



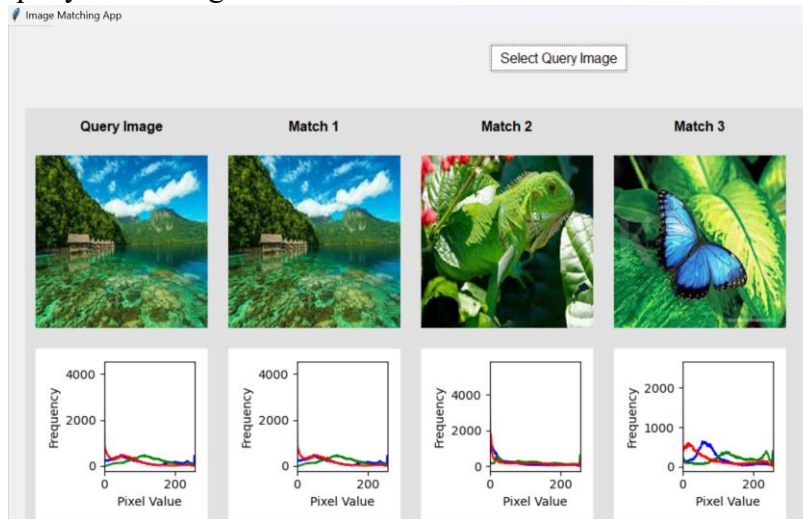
- Perbandingan dalam keakuratan kemiripan antara menjadikan gambar query dari storage online dan gambar yang di luar storage online
 - gambar yang di luar storage online



Interpretasi:

Pada gambar ini, gambar query merupakan gambar yang tidak ada di dalam storage online, jadi pada perangkat akan mengeluarkan hasil termirip berdasarkan nilai histogram nya.

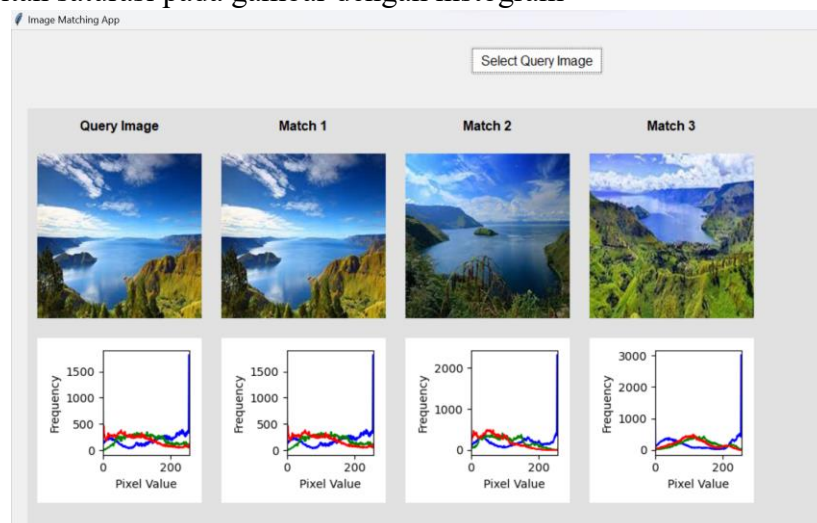
- query dari storage online



Interpretasi:

Pada gambar di atas query image terdapat juga di dalam storage online, jadi Ketika search maka ranking hasil perangkingan yang paling mirip akan memunculkan gambar itu sendiri contohnya seperti pada gambar diatas, pada ranking 1 merupakan gambar itu sendiri dengan tingkat keakuratan 100%.

- Keterkaitan saturasi pada gambar dengan histogram



Interpretasi:

Pada project kami histogram menghitung berdasarkan BGR, bukan saturasi pada sebuah gambar, pada contoh gambar di atas terlihat pada query dan match 2, match 1 merupakan keakuratan 100% gambar itu sendiri, dan match kedua merupakan gambar yang sama seperti yang di query hanya saja bedanya gambar query lebih cerah di banding yang match 2.

BAB V

KESIMPULAN

Dalam dunia teknologi informasi dan komunikasi, kemampuan untuk mencari dan menganalisis citra berdasarkan warna menjadi semakin penting. Terlebih lagi, ada kebutuhan untuk memastikan teknologi ini dapat diakses oleh semua individu, termasuk mereka yang memiliki gangguan penglihatan warna. Oleh karena itu, diperlukan metode yang tidak hanya efisien tetapi juga inklusif.

Tujuan dari proyek ini adalah untuk mengembangkan sistem pencarian citra berdasarkan komponen warna dengan menggunakan analisis histogram pada komponen BGR. Sistem ini dirancang untuk menemukan gambar yang memiliki distribusi warna serupa dengan gambar target. Selain itu, proyek ini bertujuan untuk meningkatkan kualitas visual bagi individu dengan gangguan penglihatan warna melalui koreksi warna yang tepat berdasarkan analisis area gambar yang membingungkan.

Teknologi visi komputer, khususnya analisis histogram berdasarkan komponen BGR, menawarkan solusi yang efektif untuk mencari gambar berdasarkan distribusinya. Dengan membandingkan histogram dari gambar target dengan histogram dari kumpulan gambar dalam storage, kita dapat menemukan gambar yang paling mirip dengan gambar target. Namun, keunggulan utama dari pendekatan ini adalah kemampuannya untuk mempertimbangkan individu dengan gangguan penglihatan warna. Melalui analisis wilayah warna yang membingungkan dan koreksi warna yang tepat, teknologi ini berpotensi meningkatkan pengalaman visual bagi mereka, menunjukkan bagaimana inovasi dalam visi komputer dapat berdampak positif pada masyarakat yang lebih luas.