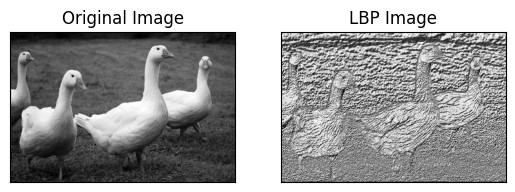
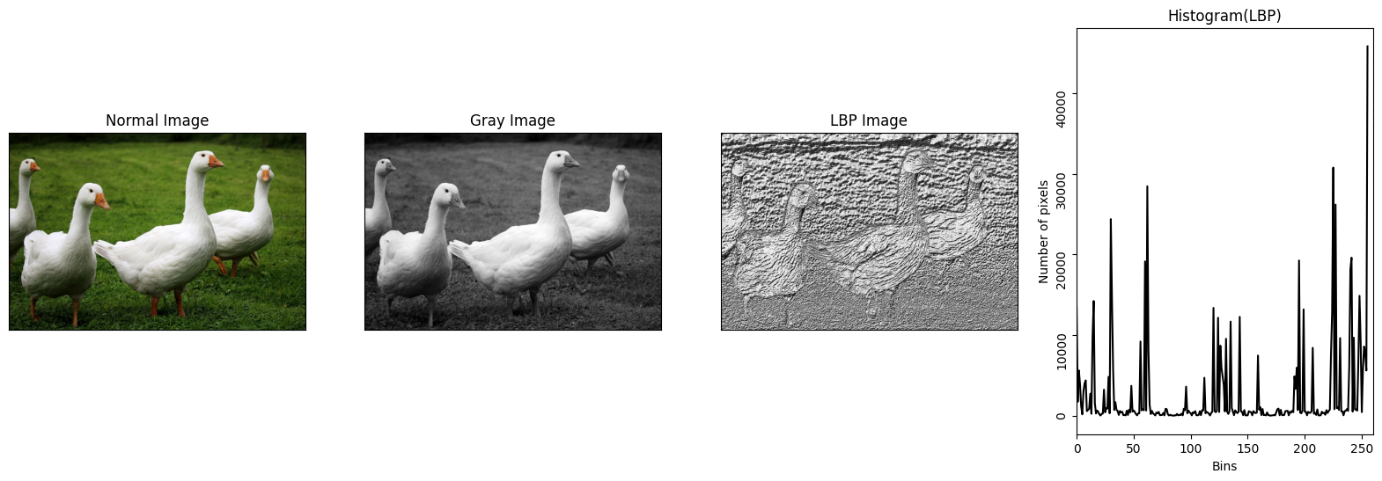
Nama : Ismail Nurhapid Jamal

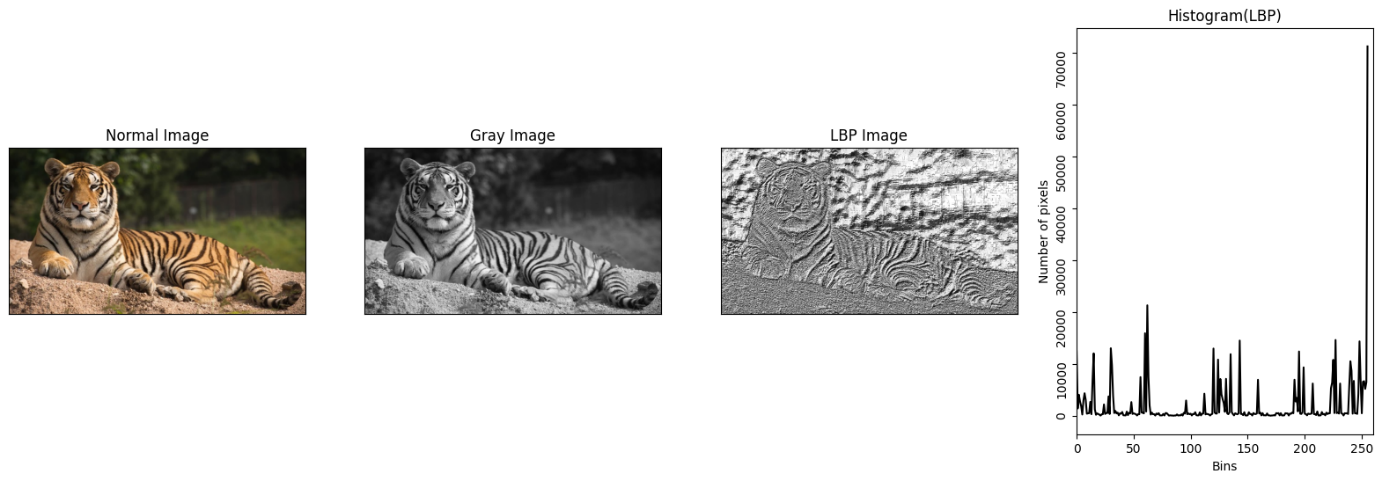
NIM : 1207070060

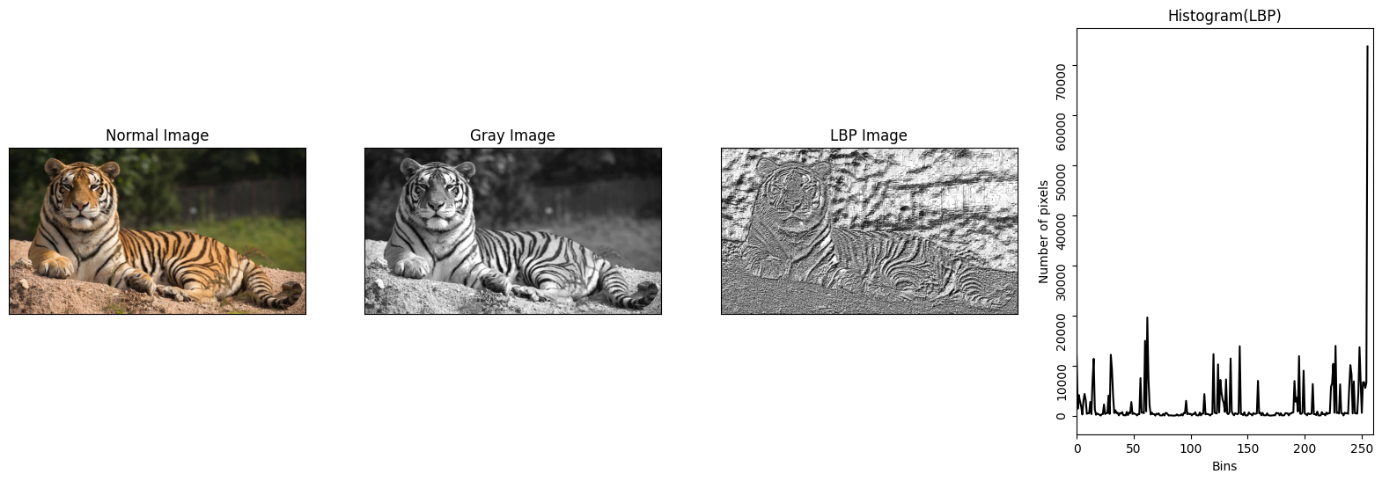
**PENGOLAHAN CITRA DIGITAL TSEB**

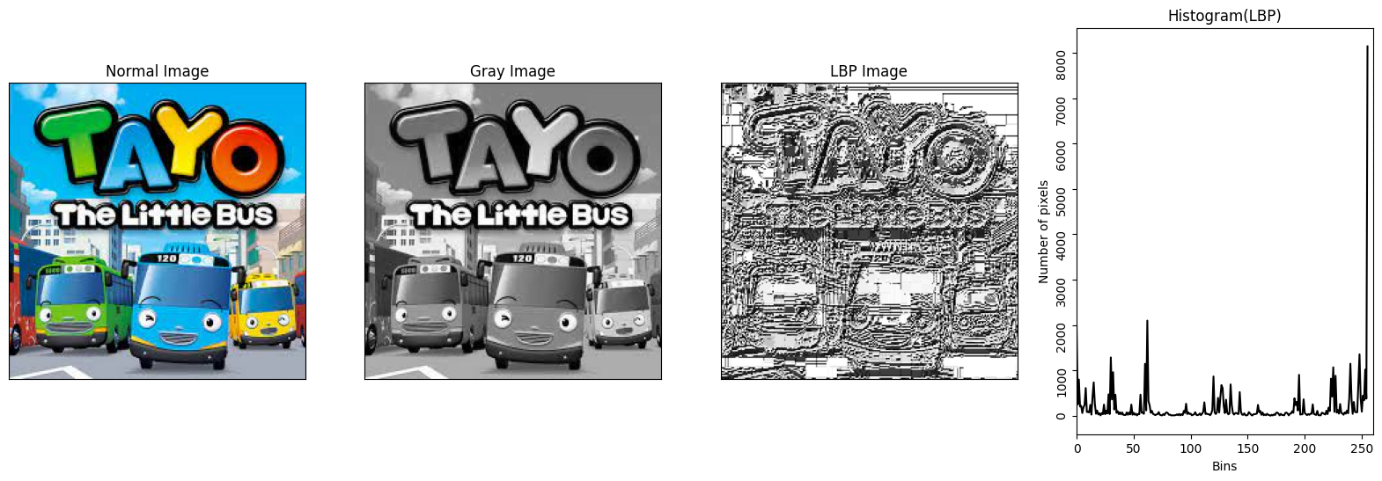
1. EKSTRASI FITUR LBP









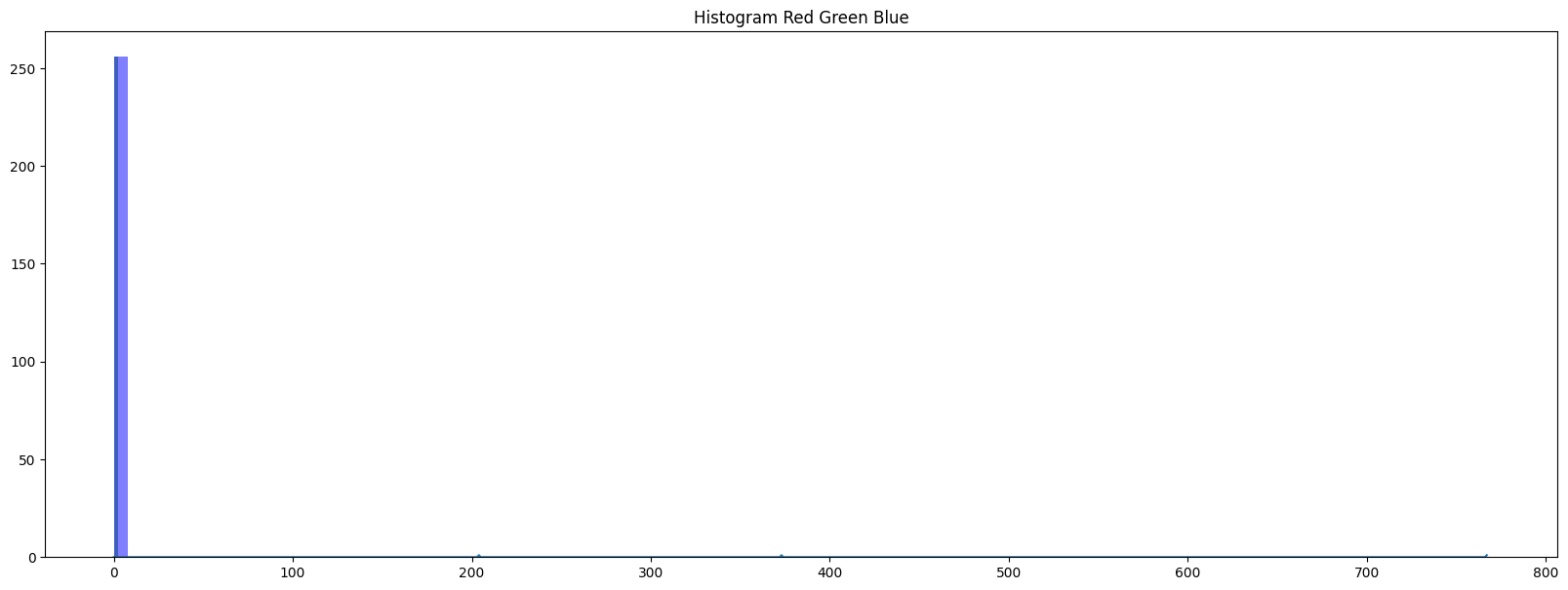


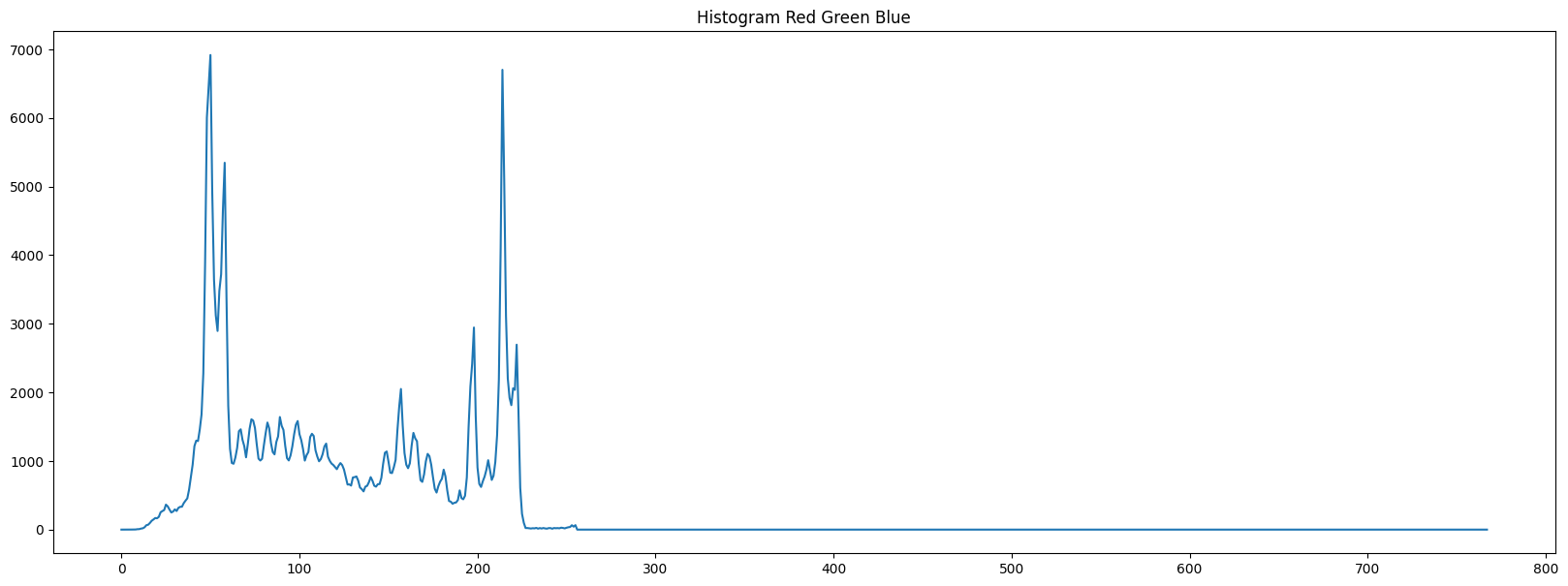
Berdasarkan program yang diberikan, terdapat beberapa perbedaan antara gambar-gambar yang dihasilkan dalam setiap pemanggilan fungsi `lbp` dengan argumen yang berbeda. Adapun perbedaannya:

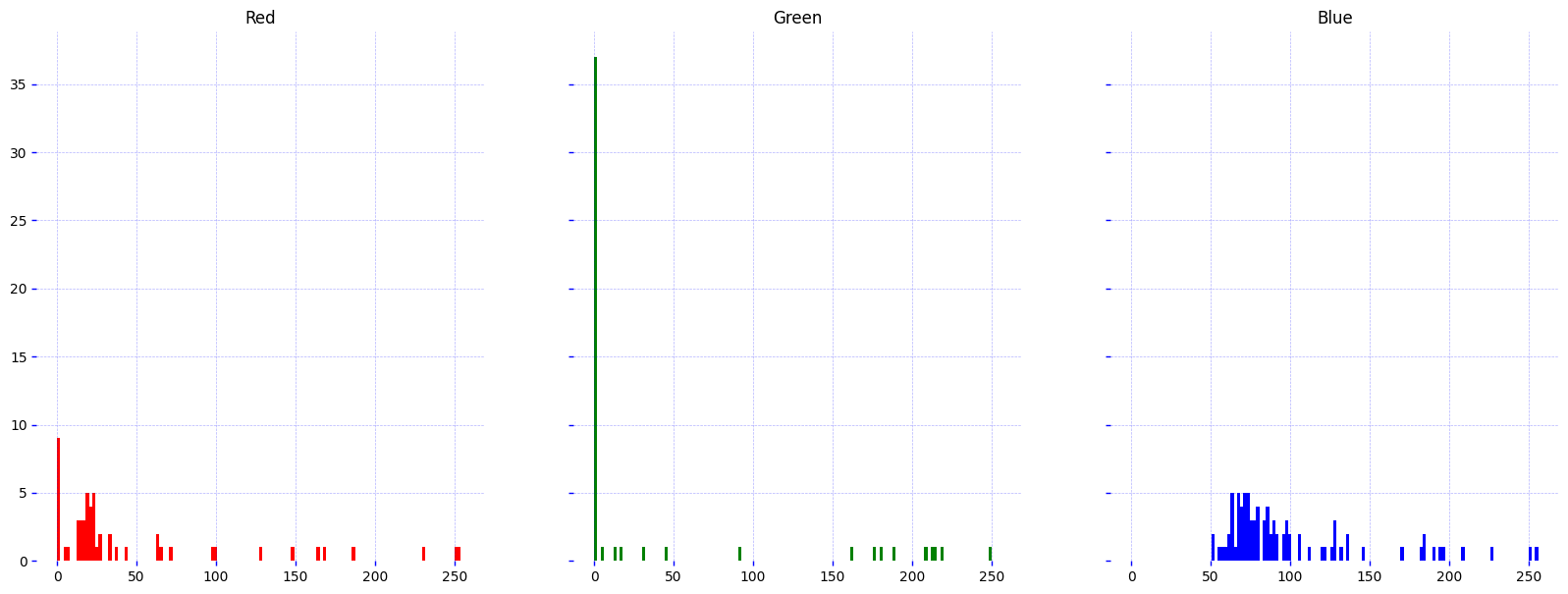
* Gambar pertama yang ditampilkan adalah gambar asli atau normal (`img\_bgr`).
* Gambar kedua adalah gambar keabuan (`img\_gray`) yang dihasilkan dari konversi gambar asli ke mode keabuan.
* Gambar ketiga adalah gambar LBP (`img\_lbp`) yang dihasilkan dari pengolahan LBP pada gambar keabuan.
* Gambar keempat adalah histogram LBP (`hist\_lbp`) yang menggambarkan distribusi piksel-piksel LBP dalam gambar.

Dengan demikian, perbedaan antara gambar-gambar yang dihasilkan terletak pada sumber gambar yang digunakan, tingkat kecerahan gambar (pada pemanggilan kedua), dan ukuran gambar (pada pemanggilan ketiga). Hal ini akan mempengaruhi nilai piksel, tekstur, dan distribusi LBP dalam gambar. Sebagai hasilnya, gambar-gambar tersebut akan memiliki tampilan yang berbeda dalam bentuk visual dan histogram LBP yang dihasilkan.

1. EKSTRASI FITUR BERDASARKAN WARNA





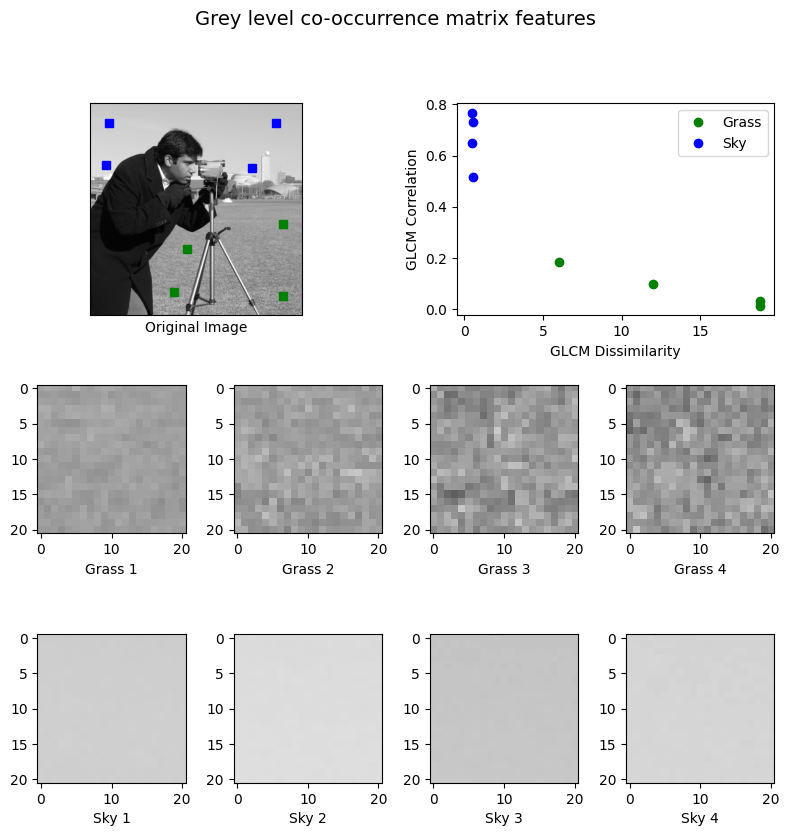


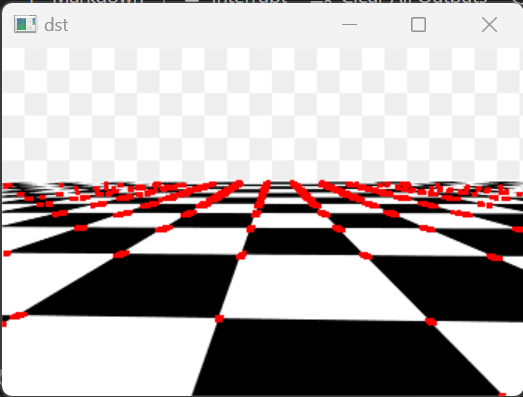
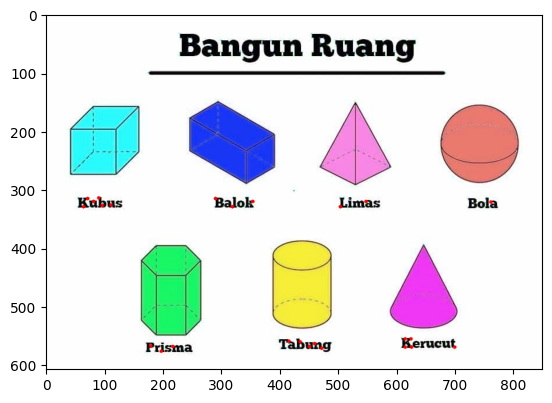
Analisis : Dalam program tersebut, terdapat beberapa proses analisis histogram dan perbandingan gambar yang dihasilkan. Berikut adalah analisis perbandingan histogram yang dihasilkan:

1. Histogram Red Green Blue (dengan menggunakan `plt.hist`):
   * Histogram ini menampilkan distribusi frekuensi nilai piksel pada kanal merah (red), hijau (green), dan biru (blue) dari gambar.
   * Setiap kanal memiliki histogram terpisah yang ditampilkan dalam warna yang sesuai.
   * Masing-masing histogram menunjukkan jumlah piksel dengan nilai tertentu pada rentang 0 hingga 256.
   * Histogram ini memberikan gambaran visual tentang distribusi intensitas piksel pada setiap kanal warna dalam gambar.
2. Histogram Red Green Blue (dengan menggunakan `cv2.calcHist` dan `plt.plot`):
   * Histogram ini juga menampilkan distribusi frekuensi nilai piksel pada kanal merah (red), hijau (green), dan biru (blue) dari gambar.
   * Menggunakan fungsi `cv2.calcHist` untuk menghitung histogram pada gambar.
   * Histogram ini direpresentasikan dengan grafik garis yang menunjukkan jumlah piksel dengan nilai tertentu pada rentang 0 hingga 768.
   * Memberikan gambaran visual tentang distribusi intensitas piksel pada setiap kanal warna dalam gambar.
3. Histogram dan Gambar Perbandingan Red, Green, dan Blue:
   * Histogram dan gambar ini menunjukkan distribusi frekuensi nilai piksel pada kanal merah (red), hijau (green), dan biru (blue) secara terpisah.
   * Terdapat tiga grafik histogram yang mewakili masing-masing kanal warna.
   * Grafik histogram menampilkan jumlah piksel dengan nilai tertentu pada rentang 0 hingga 256.
   * Grafik histogram ini memberikan pemahaman yang lebih rinci tentang distribusi intensitas piksel pada setiap kanal warna dalam gambar.

Histogram ini disertai dengan gambar histogram yang ditampilkan dengan menggunakan fungsi `plt.hist`.

1. EKSTRASI FITUR GLCM



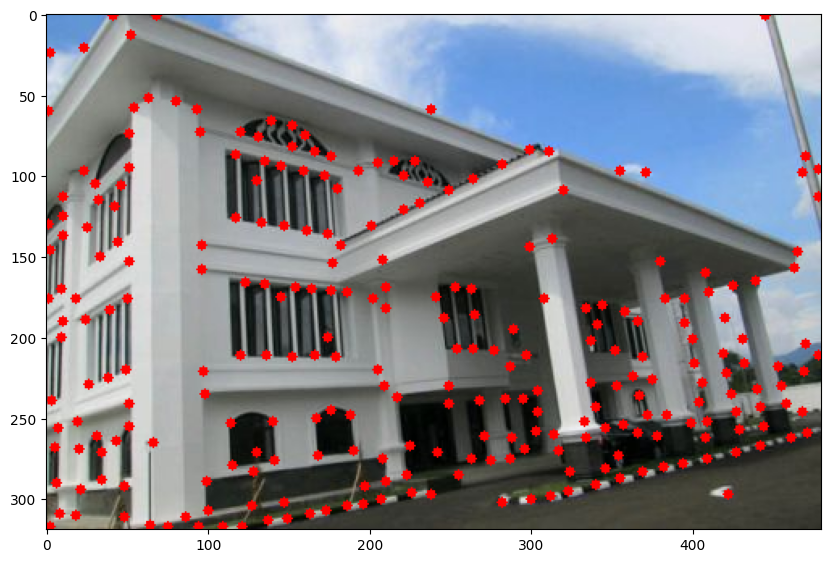


Analisis : Berdasarkan gambar yang dihasilkan, kita dapat melihat perbedaan antara patch rumput dan patch langit dalam hal fitur GLCM dissimilarity dan correlation.

1. Scatter Plot:
   * Pada scatter plot, terlihat bahwa patch rumput (ditandai dengan warna hijau) cenderung memiliki nilai dissimilarity yang lebih rendah daripada patch langit (ditandai dengan warna biru). Hal ini menunjukkan bahwa patch rumput memiliki intensitas piksel yang lebih serupa di sekitar pasangan piksel yang dianalisis, sedangkan patch langit memiliki variasi intensitas piksel yang lebih tinggi.
   * Terkait dengan correlation, terlihat bahwa patch rumput dan patch langit cenderung memiliki tingkat korelasi yang berbeda. Patch rumput cenderung memiliki nilai correlation yang lebih rendah, menunjukkan bahwa intensitas pikselnya cenderung tidak berkorelasi secara linier. Sementara itu, patch langit cenderung memiliki nilai correlation yang lebih tinggi, menunjukkan adanya korelasi linier antara intensitas pikselnya.
2. Gambar Patch:
   * Pada gambar-gambar patch rumput, kita dapat melihat bahwa teksturnya cenderung lebih homogen dengan sedikit variasi intensitas. Hal ini sesuai dengan nilai dissimilarity yang rendah yang ditemukan dalam analisis GLCM.
   * Di sisi lain, pada gambar-gambar patch langit, terlihat variasi intensitas yang lebih tinggi dan tekstur yang lebih kompleks dibandingkan dengan patch rumput. Hal ini sesuai dengan nilai dissimilarity yang lebih tinggi yang ditemukan dalam analisis GLCM.

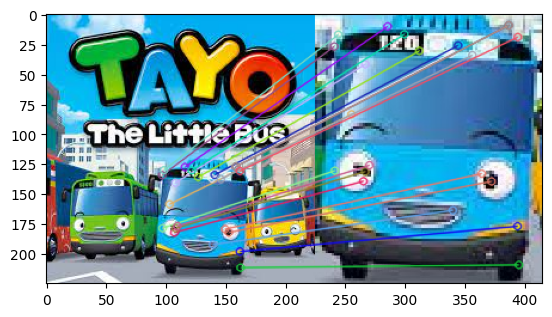
Dengan demikian, perbandingan dari gambar yang dihasilkan menunjukkan bahwa terdapat perbedaan dalam pola dan tekstur antara patch rumput dan patch langit. Patch rumput memiliki tekstur yang lebih homogen dan intensitas piksel yang lebih serupa, sedangkan patch langit memiliki variasi intensitas yang lebih tinggi dan tekstur yang lebih kompleks.

1. MENGGUNAKAN CORNER DETECTOR



Analisis : Program di atas menggunakan pustaka OpenCV (cv2) dan Matplotlib untuk melakukan deteksi pojok menggunakan metode Good Features to Track (GFTT). Pojok-pojok tersebut dipilih berdasarkan tingkat kecocokan mereka dengan pola lokal di sekitarnya. Semakin banyak pojok yang terdeteksi, semakin kompleks pola dan tekstur dalam gambar. Dengan menggunakan analisis deteksi pojok ini, program tersebut memvisualisasikan titik-titik pojok yang berhasil terdeteksi dalam gambar.

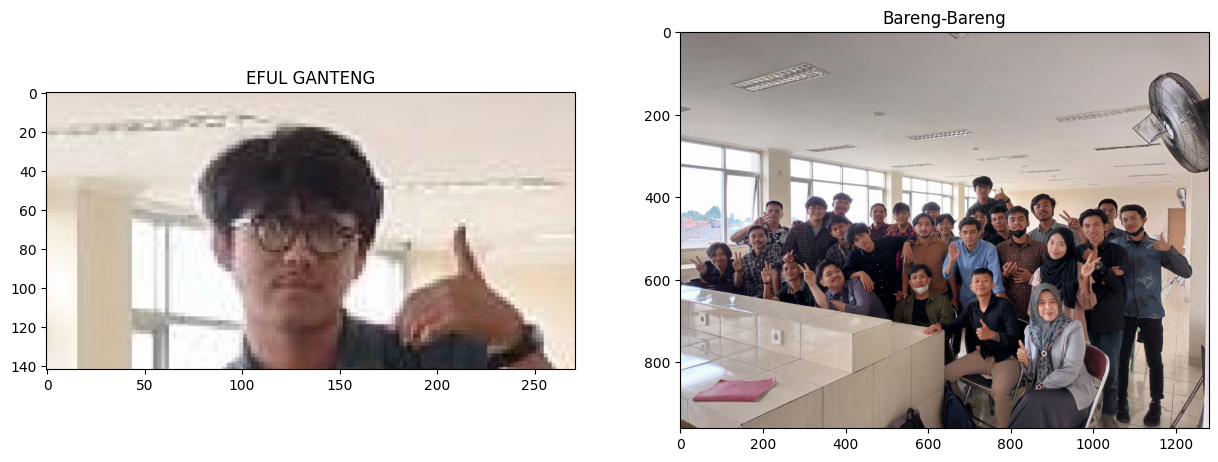
1. FEATURE DETECTION AND MATCHING

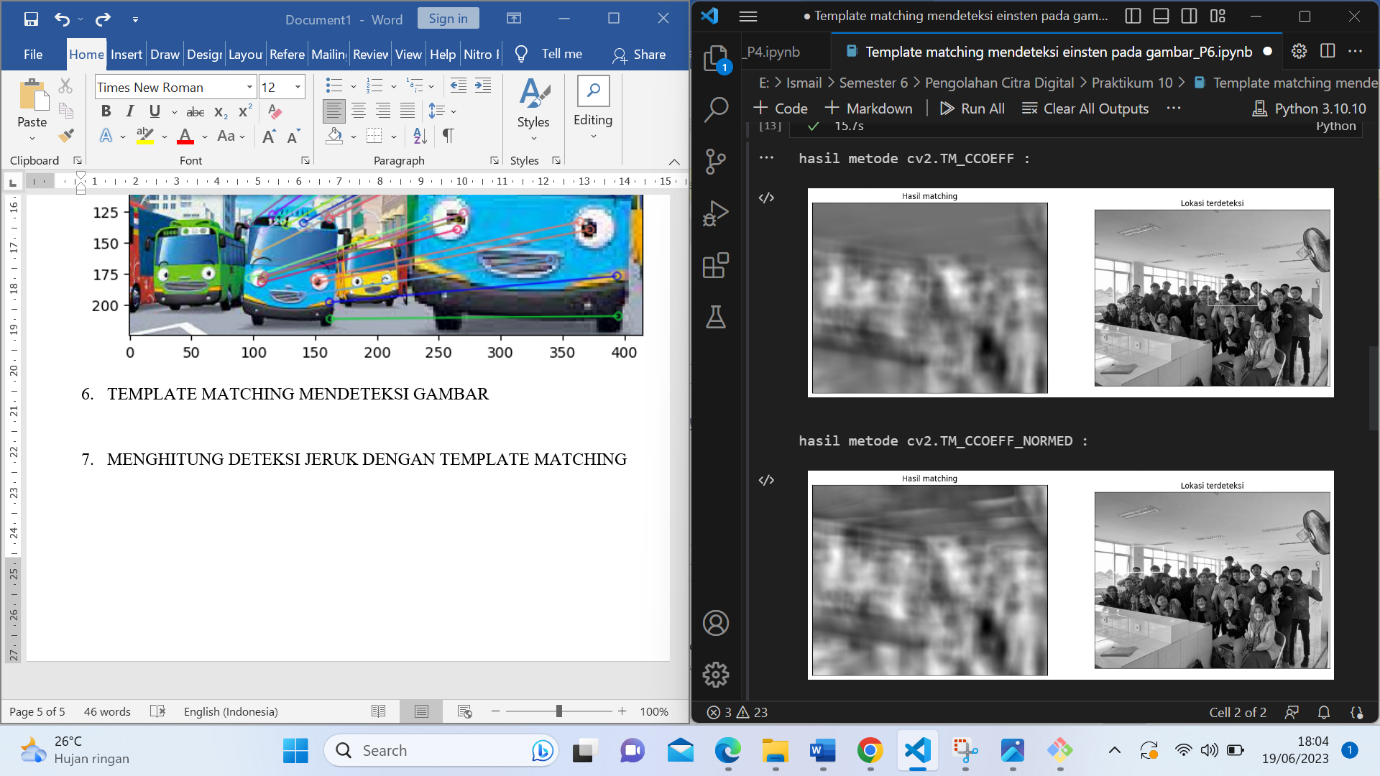
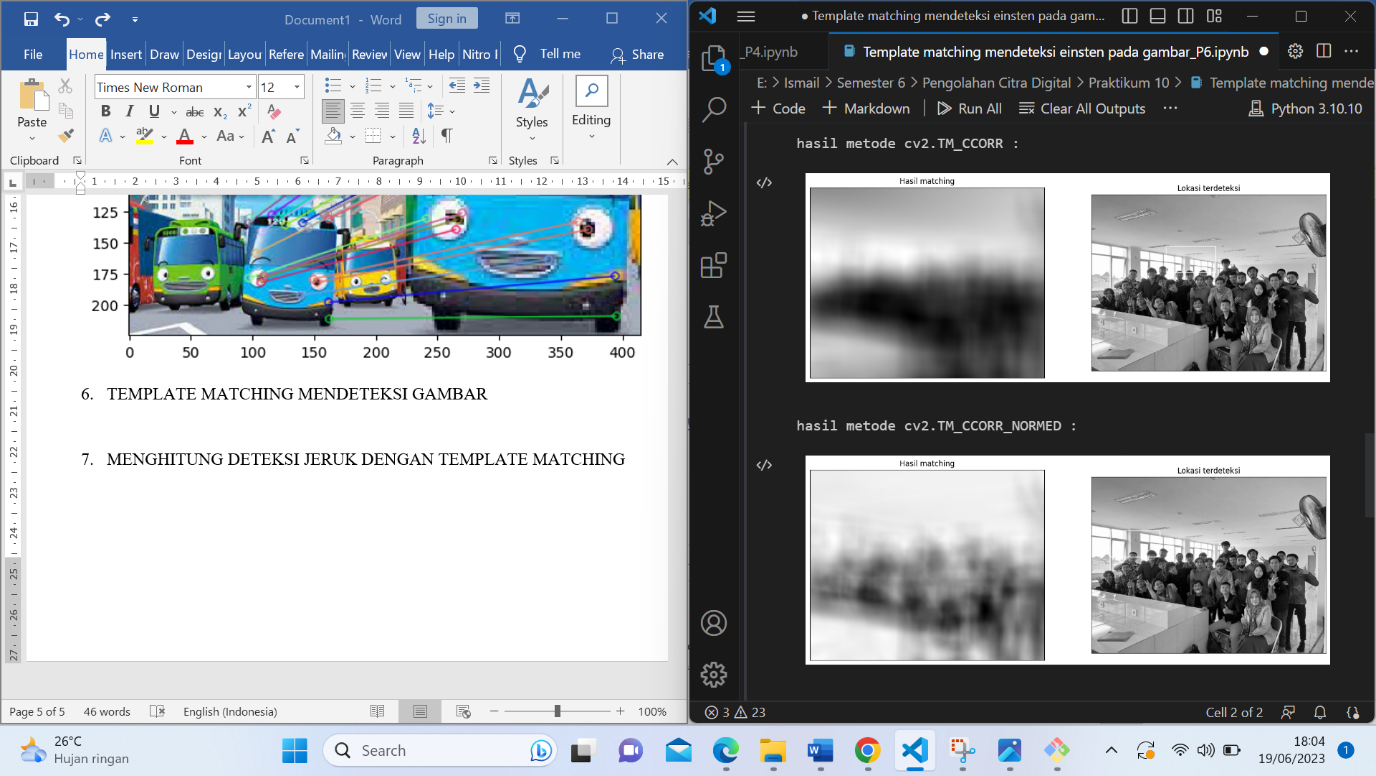


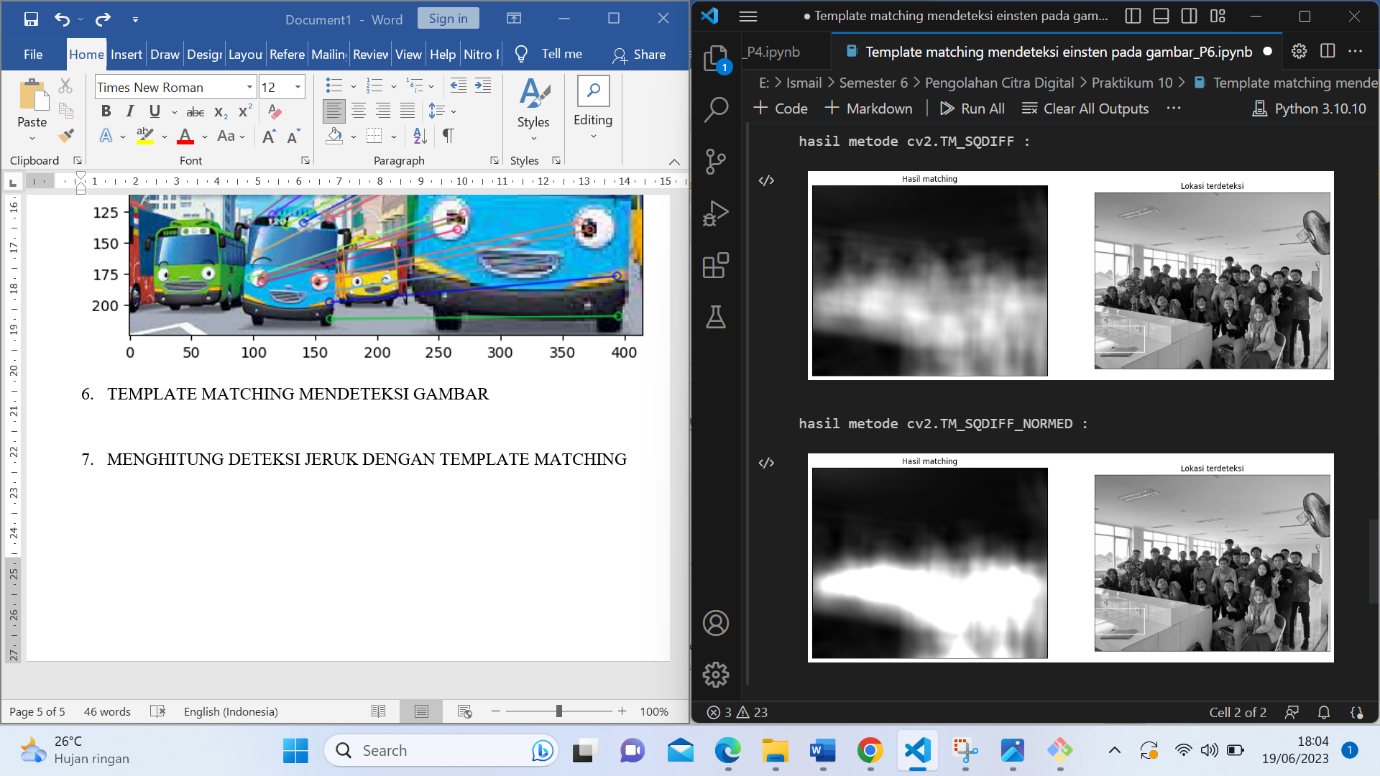
Analisis : Berdasarkan gambar yang dihasilkan oleh program tersebut, kita dapat melihat hasil pencocokan fitur antara dua gambar menggunakan metode SIFT (Scale-Invariant Feature Transform) dan BruteForce Matcher.

Dengan menggunakan analisis pencocokan fitur ini, program tersebut memvisualisasikan fitur-fitur yang cocok antara dua gambar. Garis-garis yang menghubungkan fitur-fitur yang cocok memberikan informasi tentang kesamaan dan kecocokan fitur-fitur tersebut. Semakin banyak dan semakin baik fitur-fitur yang cocok, semakin mirip kedua gambar tersebut dalam konteks fitur-fitur yang telah dicocokkan.

1. TEMPLATE MATCHING MENDETEKSI GAMBAR



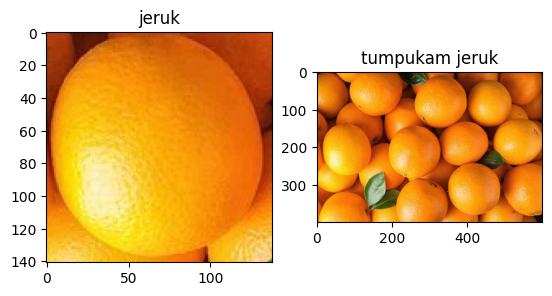


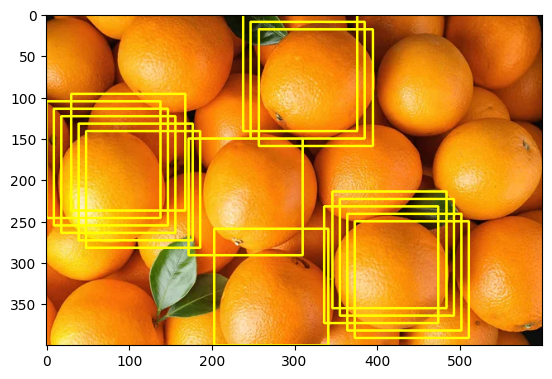


Analisis : Berdasarkan gambar yang dihasilkan oleh program tersebut, kita dapat melihat hasil pencocokan fitur antara dua gambar menggunakan metode SIFT (Scale-Invariant Feature Transform) dan BruteForce Matcher.

Dengan menggunakan analisis pencocokan fitur ini, program tersebut memvisualisasikan fitur-fitur yang cocok antara dua gambar. Garis-garis yang menghubungkan fitur-fitur yang cocok memberikan informasi tentang kesamaan dan kecocokan fitur-fitur tersebut. Semakin banyak dan semakin baik fitur-fitur yang cocok, semakin mirip kedua gambar tersebut dalam konteks fitur-fitur yang telah dicocokkan.

1. MENGHITUNG DETEKSI JERUK DENGAN TEMPLATE MATCHING





Analisis : Berdasarkan gambar yang dihasilkan oleh program tersebut, kita dapat melihat hasil pencocokan template pada gambar utama menggunakan metode Template Matching dengan berbagai metode yang berbeda.

Dari gambar yang dihasilkan, kita dapat memperoleh informasi tentang sejauh mana template cocok dengan citra utama berdasarkan tingkat kesamaan dan lokasi terdeteksi. Metode dengan hasil yang lebih baik akan menghasilkan gambar hasil pencocokan yang lebih baik dengan persegi panjang yang tepat mengelilingi lokasi yang cocok.