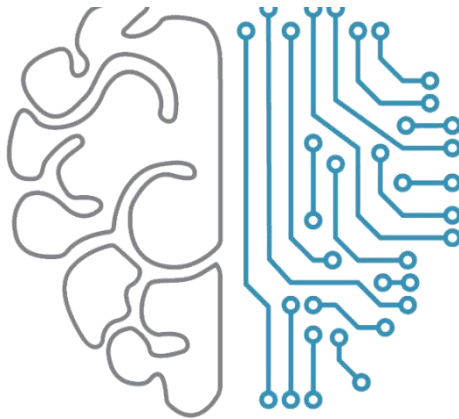


# **LAPORAN TEORI**

## **PENGOLAHAN CITRA DIGITAL**



INTELLIGENT **COMPUTING**

NAMA : Dimas Damarjati

NIM : 202231020

KELAS : B

DOSEN : Dr. Dra. Dwina Kuswardani, M.Kom

NO.PC : 13

ASISTEN : 1. Fachreza Riyanda.

2.

**INSTITUT TEKNOLOGI PLN**

**TEKNIK INFORMATIKA**

**2024**

### ➤ Teori:

Pengolahan citra digital adalah teknik mengolah citra yang bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin komputer yang dapat berupa foto maupun gambar bergerak. Dalam laporan ini digunakan library-library python seperti:

- Library OpenCV (Open Source Computer Vision) adalah library sumber terbuka yang menyediakan berbagai fungsi dan algoritma yang digunakan untuk memproses gambar, video, dan data visual.
- Library Matplotlib membuat berbagai jenis plot, grafik, dan visualisasi data dengan mudah. Matplotlib mendukung pembuatan grafik 2D dan 3D, termasuk scatter plots, line plots, bar plots, histogram, heatmap, dan lain-lain.
- Library NumPy menyediakan struktur data seperti array dan matriks, serta berbagai fungsi untuk melakukan operasi matematika dan manipulasi data.

Histogram adalah representasi grafis dari distribusi frekuensi atau probabilitas nilai-nilai data dalam satu set atau interval tertentu. Dalam pemrosesan citra, histogram digunakan untuk menganalisis distribusi intensitas piksel dalam gambar dan mencerminkan seberapa sering nilai intensitas tertentu muncul dalam gambar.

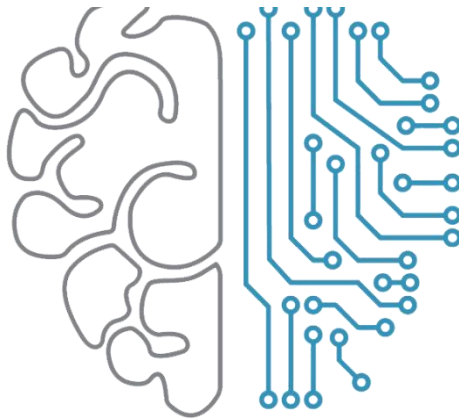
Masking merupakan pemisahan antara objek yang diamati dengan objek yang tidak diamati. Dengan cara suatu area atau objek dalam gambar "ditutupi" atau "dibatasi" oleh area yang lain, yang disebut mask. Biasanya memiliki nilai piksel 0 (hitam) atau 1 (putih), di mana nilai 1 menunjukkan area yang ingin diperhatikan atau dibiarkan, sementara nilai 0 menunjukkan area yang ingin disaring.

"threshold" (ambang batas) merupakan bagian dari segmentasi. Segmentasi Citra biasanya didasarkan pada dua properti dasar dari level keabuan yaitu diskontinuitas dan kesamaan. Prinsip diskontinuitas adalah citra dibagi berdasarkan perubahan yang besar pada tingkat keabuan sedangkan prinsip kesamaan membagi citra ke area yang mempunyai kesamaan tingkat keabuan. Threshold mengklasifikasikan piksel dalam gambar berdasarkan nilai intensitasnya. "threshold" (ambang batas) membagi gambar menjadi dua kelas atau lebih, tergantung pada nilai ambang batas yang ditentukan, yang menghasilkan gambar biner di mana piksel diberi nilai 0 atau 1 (hitam atau putih) berdasarkan kriteria tertentu.

Image segmentation membagi citra menjadi beberapa bagian, sehingga dapat memudahkan klasifikasi otomatis oleh komputer. Dibutuhkan algoritma yang tepat untuk melakukan segmentasi sehingga hasil segmentasi sesuai dengan yang diinginkan.

# **LAPORAN PRAKTIKUM**

## **PENGOLAHAN CITRA DIGITAL**



INTELLIGENT **COMPUTING**

NAMA : Dimas Damarjati

NIM : 202231020

KELAS : B

DOSEN : Dr. Dra. Dwina Kuswardani, M.Kom

NO.PC : 13

ASISTEN : 1. Fachreza Riyanda.

2.

**INSTITUT TEKNOLOGI PLN**

**TEKNIK INFORMATIKA**

**2024**

➤ Latar belakang:

Telah dilakukannya praktikum-praktikum pengolahan citra digital, sehingga dilakukanlah ujian tengah semester untuk praktikum dari mata kuliah pengolahan citra digital.

➤ Tujuan:

Membuat laporan ujian tengah semester dari praktikum mata kuliah pengolahan citra digital.

➤ Alat dan bahan:

-Jupyter Notebook

-Library OpenCV

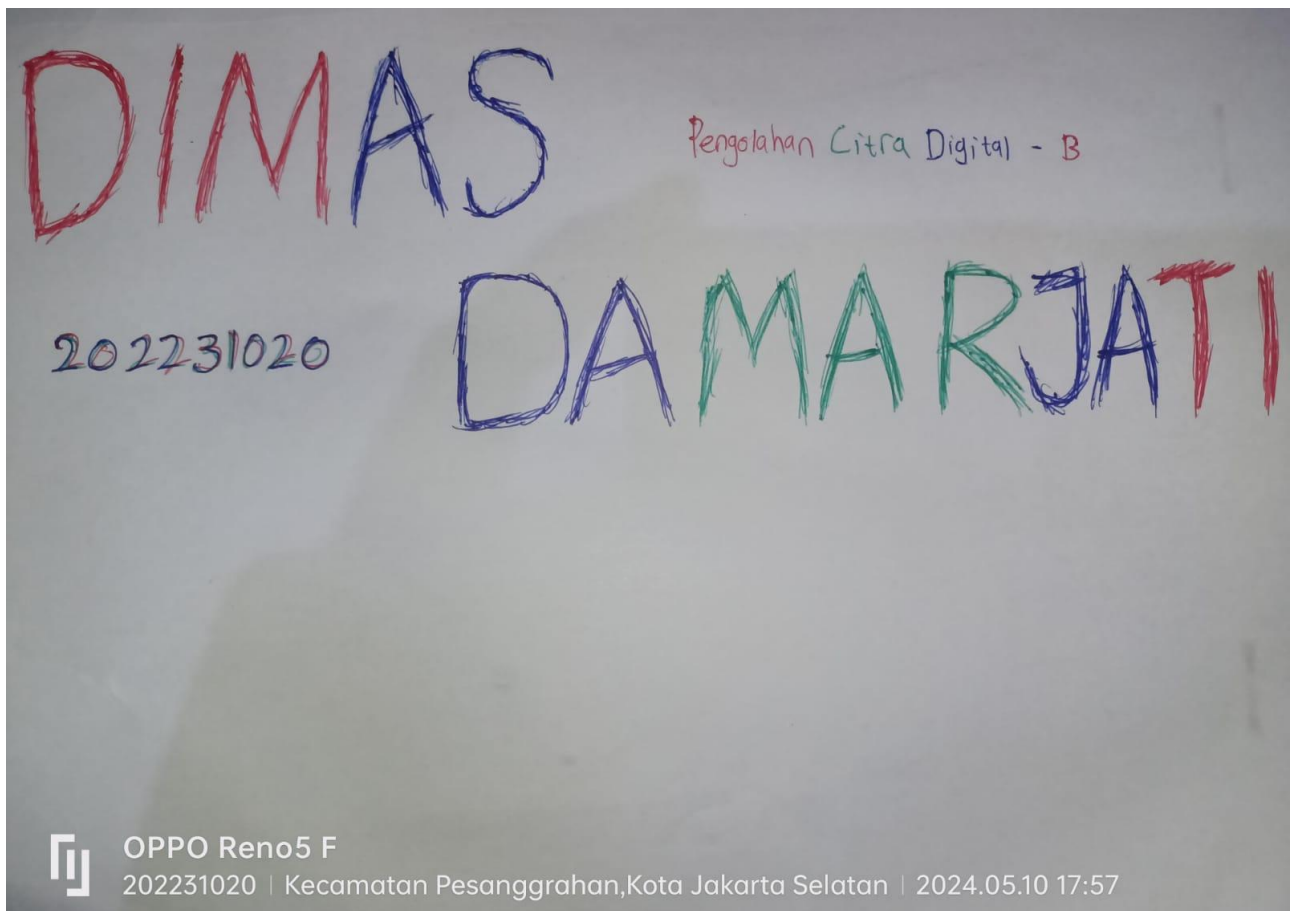
➤ Pembahasan:

➤ Arahan:

### PROJECT UTS PENGOLAHAN CITRA B

#### KETENTUAN CITRA:

bukti foto dengan membuka rincian dan tempat



## KETENTUAN PROJEK :

- Potretlah sebuah citra yang berisi Nama Lengkap masing-masing
- Nama ditulis tangan menggunakan pena dengan tinta Merah, hijau, dan biru
- Pemilihan urutan warna dibebaskan.

### o Contoh citra

Buat program dengan Bahasa pemrograman python untuk :

#### 1. DETEKSI WARNA PADA CITRA

Ketentuan :

- o Deteksi warna biru, merah dan hijau pada gambar dan tampilkan semua.
- o Analisislah apa yang terjadi pada citra dan lampirkan pada hasil analisis pada laporan.
- o **Buat histogram** untuk setiap gambar dan analisis lah histogram tersebut dan lampirkan pada laporan

#### 2. CARILAH DAN URUTKAN AMBANG BATAS TERKECIL SAMAPAI DENGAN TERBESAR

- o Carilah nilai ambang batas citra untuk dapat menampilkan kategori warna pada citra
- o Contoh hasil
- o Lampirkan nilai ambang batas yang didapat dan jelaskan alasan kenapa mendapatkan nilai ambang batas tersebut pada laporan

#### 3. Membuat LAPORAN penjelasan Proyek.

Laporan dibuat dalam bentuk file Radme.

Pembuatan laporan dapat melalui [readme.so](https://readme.so) (opsional)

Didalam laporan diharuskan terdapat bagian berikut:

- ◆ Menjelaskan tahapan cara menyelesaikan proyek secara rinci (wajib)

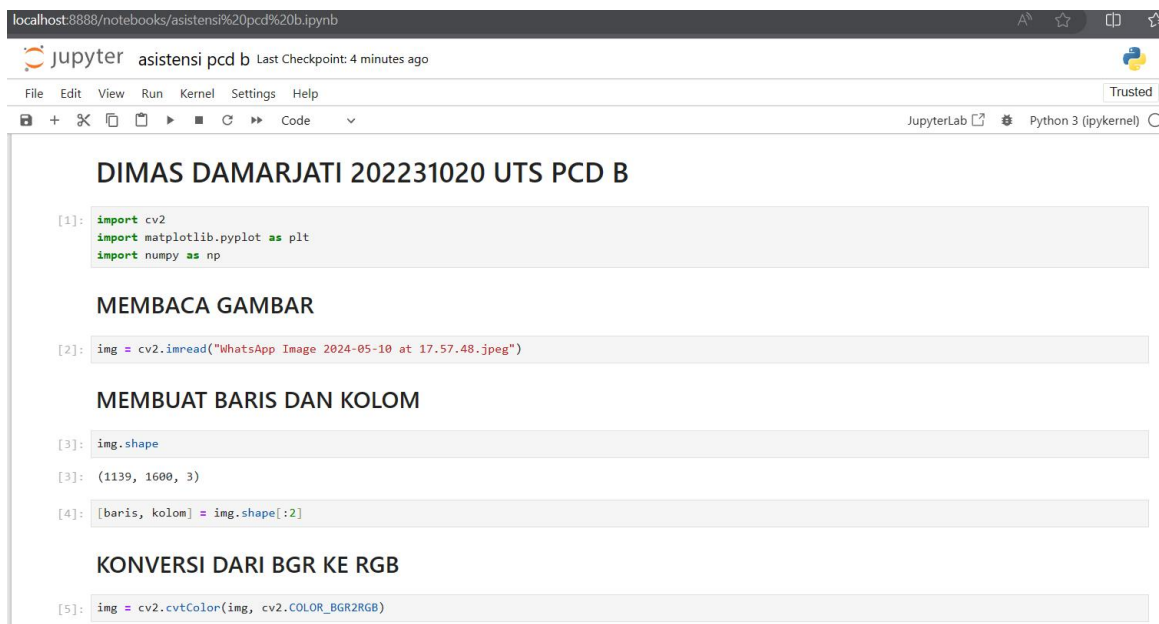
- ◆ Teori yang mendukung mengenai proyek terkait (wajib)
- ◆ Tambahkan jurnal terkait (Opsional) PENGUMPULAN PROJEK

Uploadlah File IPYNB, Citra,dan Laporan pada akun github pribadi dengan format Repository [ PCD\_UTS\_NIM\_2024\_ITPLN].

Link dari REPOSITORY dikumpulkan pada Assignment Teams.

➤ **Pengerjaan:**

## 1. DETEKSI WARNA PADA CITRA



Sebelum membuat program untuk mendeteksi warna, kita buat program untuk menggunakan library cv2 untuk pengolahan citra dan matplotlib untuk kebutuhan plot serta numpy untuk segala operasi matematik. Lalu perintah untuk membaca gambar dan menetapkan "img" sebagai gambar yang akan diolah. Lalu menetapkan baris dan kolom dari gambar tersebut. Lalu kita konversi warna gambar tersebut dari BGR ke RGB. Setelah itu kita masukkan perintah untuk mendeteksi warna merah, hijau dan biru.

## MENDETEKSI WARNA

```
[77]: lower_blue = np.array([50, 50, 100])
      upper_blue = np.array([100, 100, 255])

      lower_red = np.array([100, 50, 50])
      upper_red = np.array([255, 100, 100])

      lower_green = np.array([50, 100, 50])
      upper_green = np.array([100, 255, 100])

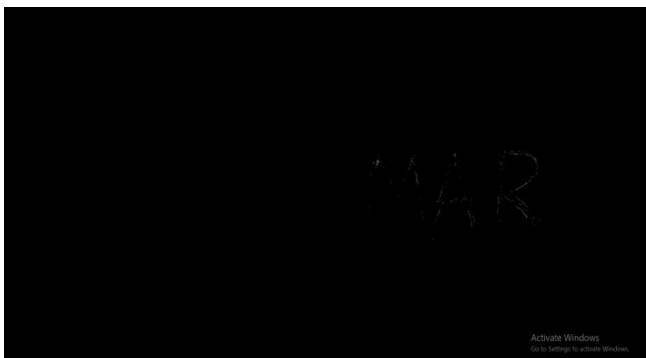
[78]: mask_blue = cv2.inRange(img, lower_blue, upper_blue)
      mask_red = cv2.inRange(img, lower_red, upper_red)
      mask_green = cv2.inRange(img, lower_green, upper_green)

[79]: color_blue = cv2.bitwise_and(img, img, mask=mask_blue)
      color_red = cv2.bitwise_and(img, img, mask=mask_red)
      color_green = cv2.bitwise_and(img, img, mask=mask_green)

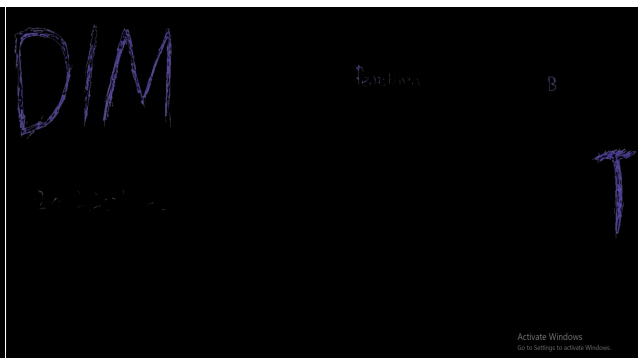
[80]: cv2.imshow('Blue Color Detected', color_blue)
      cv2.imshow('Red Color Detected', color_red)
      cv2.imshow('Green Color Detected', color_green)
      cv2.waitKey(0)
      cv2.destroyAllWindows()
```

Dalam program tersebut digunakan masking untuk memisahkan piksel-piksel yang memiliki nilai warna yang sesuai dengan rentang warna yang ditentukan. Contohnya disini `upper_blue = np.array([100, 100, 255])`. Angka tersebut dalam model RGB dimana artinya R = 100, G = 100, B = 255. Hal tersebut berarti warna merah dan hijau kelihatan sedikit dan hampir bewarna biru untuk mengambil warna yang hampir biru pada gambar dan warna biru 255 maksudnya yaitu bewarna biru murni karena skala warna 0 itu pudar warna tersebut dan semakin tinggi semakin murni warna tersebut hingga bernilai 255. Didapatkan hasil berikut:

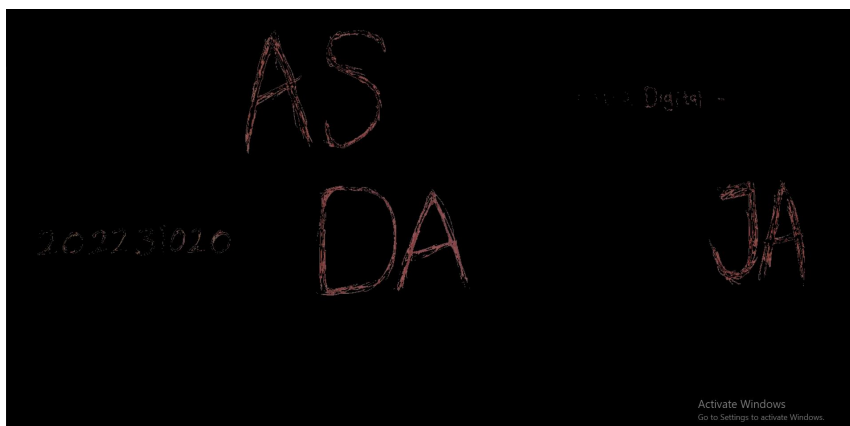
**Hijau**



**Merah**



**Biru**



Didapatkan hasil warna biru dengan visibilitas terbaik lalu merah lalu hijau. Hal ini menunjukkan bahwa warna biru yang digunakan pada gambar uji coba sangat hampir mendekati murni biru, diikuti dengan warna merah, lalu warna hijau yang kurang murni hijau pada gambar uji coba tersebut. Sekarang kita buat histogram untuk gambar-gambar tersebut.

#### HISTOGRAM

```
[139]: img1 = cv2.imread("Screenshot (1062).png")
       cv2.imwrite('Screenshot (1062).jpg', img1, [int(cv2.IMWRITE_JPEG_QUALITY), 100])

[139]: True

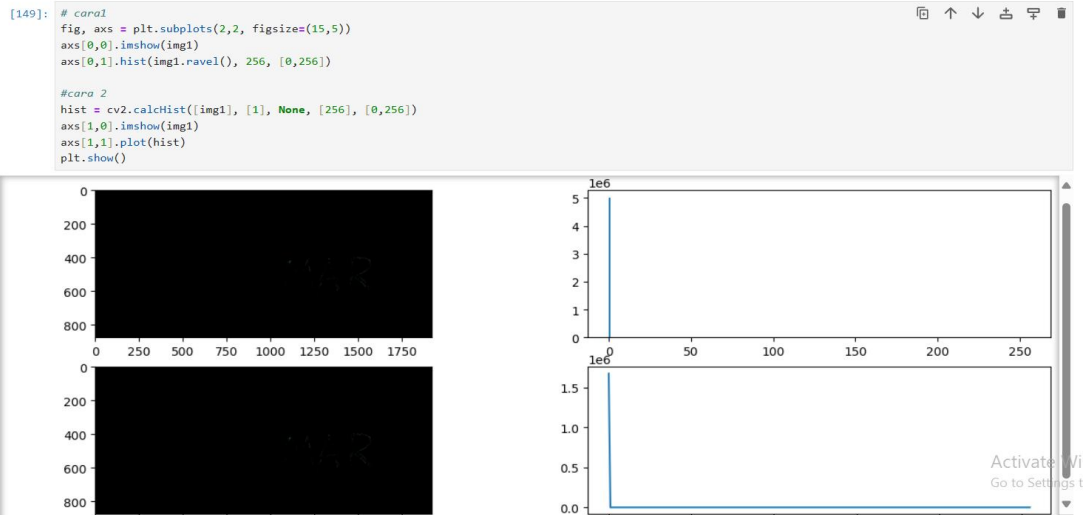
[140]: img2 = cv2.imread("Screenshot (1063).png")
       cv2.imwrite('Screenshot (1063).jpg', img2, [int(cv2.IMWRITE_JPEG_QUALITY), 100])

[140]: True

[141]: img3 = cv2.imread("Screenshot (1064).png")
       cv2.imwrite('Screenshot (1064).jpg', img3, [int(cv2.IMWRITE_JPEG_QUALITY), 100])

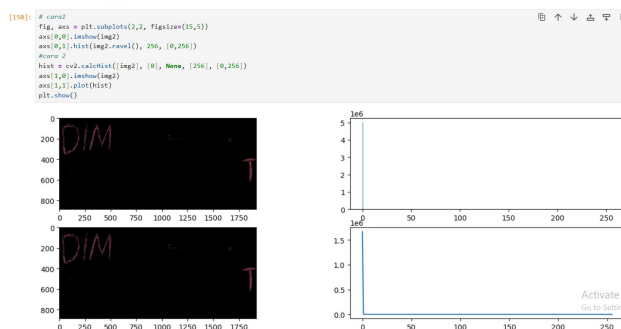
[141]: True
```

## warna hijau

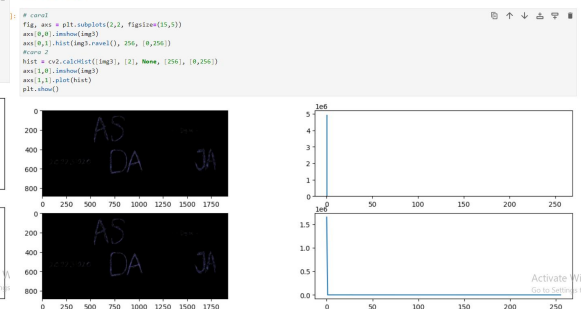


Dalam histogram warna hijau, menunjukkan warna hijau tidak intens hampir tidak ada. Cara 1 menunjukkan intensitas piksel pada seluruh gambar, sedangkan cara 2 menunjukkan intensitas piksel pada satu saluran warna. Pada cara dua cv2.calcHist menggunakan indeks dengan no 1. indeks no 1 digunakan untuk menunjukkan intensitas saluran warna hijau.

## warna merah



## warna biru



Untuk warna merah dan biru saya tidak dapat menemukan hasil kemungkinan karena kesalahan dalam penulisan program yang saya buat.



## 2. CARILAH DAN URUTKAN AMBANG BATAS TERKECIL SAMPAI DENGAN TERBESAR

nilai ambang batas citra

```
# Split the image into its color channels
blue_channel, green_channel, red_channel = cv2.split(img)

# Thresholding
_, blue_thresholded = cv2.threshold(blue_channel, 100, 255, cv2.THRESH_BINARY)
_, green_thresholded = cv2.threshold(green_channel, 150, 255, cv2.THRESH_BINARY)
_, red_thresholded = cv2.threshold(red_channel, 200, 255, cv2.THRESH_BINARY)

# Combine
combined_thresholded = cv2.merge((blue_thresholded, green_thresholded, red_thresholded))

# Display
plt.figure(figsize=(10, 5))

plt.subplot(2, 2, 1)
plt.imshow(blue_thresholded, cmap='gray')
plt.title('biru')

plt.subplot(2, 2, 2)
plt.imshow(green_thresholded, cmap='gray')
plt.title('hijau')

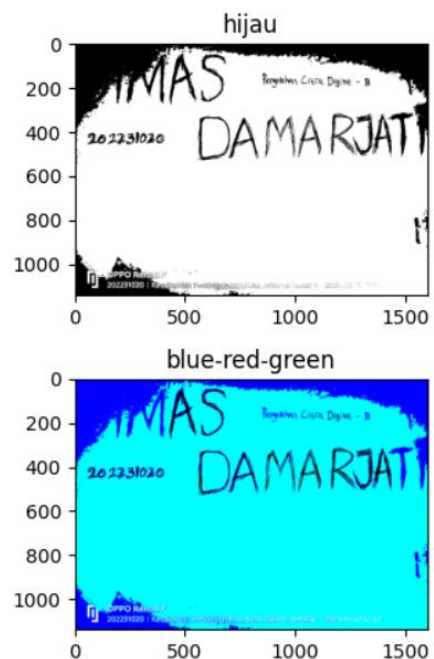
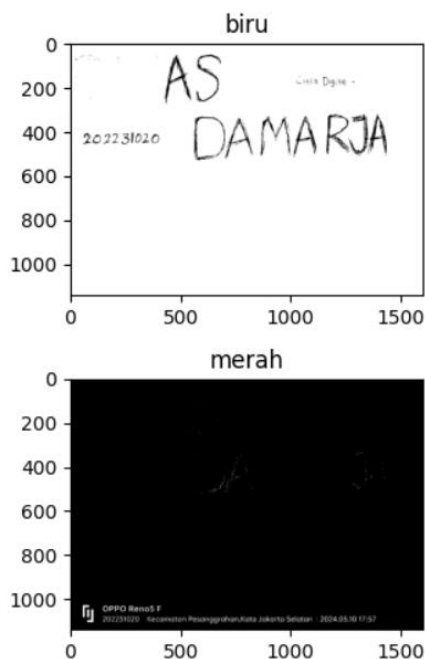
plt.subplot(2, 2, 3)
plt.imshow(red_thresholded, cmap='gray')
plt.title('merah')

plt.subplot(2, 2, 4)
plt.imshow(cv2.cvtColor(combined_thresholded, cv2.COLOR_BGR2RGB))
plt.title('blue-red-green')

plt.tight_layout()
plt.show()
```

Activate V  
Go to Settings

Pada program ini gambar dimuat menggunakan `cv2.imread()` dan dipisahkan menjadi saluran warna biru, hijau, dan merah menggunakan `cv2.split()`. Thresholding/nilai ambang batas dilakukan pada masing-masing saluran warna dengan menggunakan `cv2.threshold()`. Lalu hasil thresholding digabungkan kembali menggunakan `cv2.merge()` untuk menghasilkan gambar berwarna dengan warna yang disegmentasi berdasarkan warna yang telah dipilih. Gambar hasil ditampilkan menggunakan Matplotlib sebagai berikut:



Dalam thresholding tadi saya telah tetapkan ambang batas untuk biru 100, hijau 150, dan merah 200. Semakin tinggi ambang batas seharusnya semakin tinggi piksel yang muncul sesuai warna yang telah ditetapkan. Tetapi, hasil menunjukkan tidak sesuai harapan dengan kemungkinan kesalahan terhadap program yang telah saya buat.

➤ **Kesimpulan:**

Dalam praktikum ini OpenCV library digunakan untuk pengolahan citra dan matplotlib untuk kebutuhan plot serta numpy untuk segala operasi matematik. RGB digunakan karena penggunaanya yang memudahkan dalam pengolahan citra digital. Dalam program ini digunakan masking untuk memisahkan piksel-piksel yang memiliki nilai warna yang sesuai dengan rentang warna yang ditentukan. Pada program ini, Thresholding/ambang batas digunakan untuk menyoroti warna tertentu pada gambar uji coba. Sayangnya dengan hasil praktikum saya yang kurang memuaskan karena kesalahan dalam membuat program belum bisa didapatkan kesimpulan yang pasti untuk hasil praktikum saya kali ini.