

# **APLIKASI PENGOLAHAN CITRA DIGITAL DENGAN ANTARMUKA GUI MENGUNAKAN PYTHON**

**Dosen Pengampu: Naili Suri Intizhami, S.Kom., M.Kom.**



## **TIM SUPRA:**

### **ILMU KOMPUTER 23-A**

Suci Ramadhani (231011010)

Putri Kalsum (231011009)

### **ILMU KOMPUTER 23-B**

Andi Alfiansyah (231011089)

Ryan Hidayat (231011074)

## **PROGRAM STUDI ILMU KOMPUTER**

**INSTITUT TEKNOLOGI BACHARUDDIN JUSUF HABIBIE**

**2025**

## A. DESKRIPSI APLIKASI

Aplikasi ini dirancang sebagai alat bantu pengolahan citra digital yang memudahkan pengguna dalam memanipulasi dan menganalisis gambar secara langsung dan interaktif. Program ini dibuat menggunakan bahasa pemrograman Python dengan dukungan berbagai pustaka penting seperti Tkinter untuk merancang tampilan antarmuka grafis (GUI), OpenCV (cv2) untuk fungsi pemrosesan gambar, Pillow (PIL) untuk pengelolaan gambar, serta Matplotlib untuk menyajikan histogram. Antarmuka dirancang secara intuitif, terdiri atas dua panel utama yaitu panel kiri untuk menampilkan gambar asli, dan panel kanan untuk menampilkan hasil pengolahan. Fitur-fitur utama meliputi konversi gambar ke grayscale dan biner, penyesuaian tingkat kecerahan (brightness), operasi logika NOT, deteksi tepi dengan metode Sobel (baik arah horizontal maupun vertikal), serta visualisasi histogram warna dan grayscale. Di samping itu, aplikasi ini juga menyertakan fitur morfologi berupa erosi citra dengan tiga pilihan elemen penstruktur: vertikal, horizontal, dan kotak penuh. Aplikasi ini sangat ideal sebagai sarana edukasi dan eksplorasi bagi siapapun yang ingin memahami dasar-dasar pengolahan citra digital melalui pendekatan visual dan interaktif.

### • SCREENSHOT TAMPILAN UMUM APLIKASI:



**Gambar 1: Tampilan umum aplikasi**

## B. FITUR APLIKASI

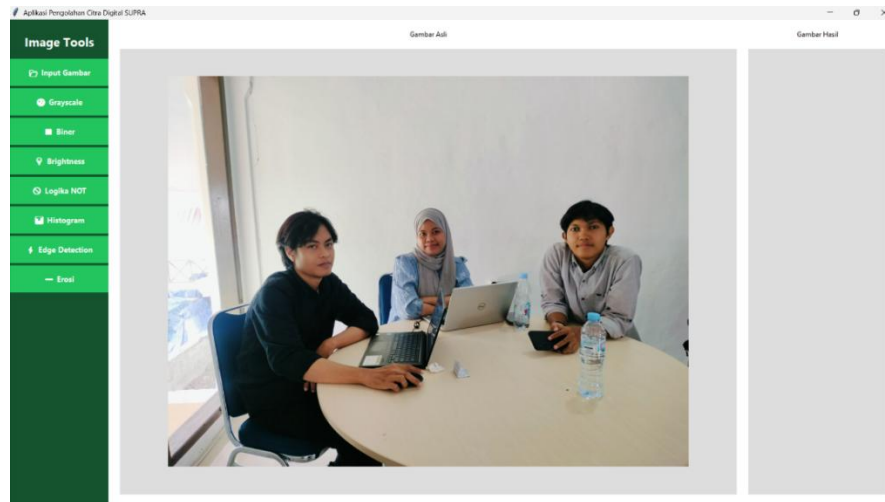
### ➤ Fitur Wajib

#### 1) Input Gambar

Fungsi ini memungkinkan pengguna untuk memilih dan memuat file gambar dari perangkat mereka ke dalam sistem aplikasi. Saat tombol ditekan, sebuah jendela dialog akan muncul untuk memilih gambar dengan ekstensi seperti .jpg, .jpeg, .png, .bmp, atau .tif. Setelah dipilih, gambar ditampilkan pada panel sebelah kiri sebagai referensi asli. Panel sebelah kanan akan menunggu hasil dari fitur pemrosesan lainnya.

Teknisnya, proses ini menggunakan `filedialog.askopenfilename()` dari Tkinter untuk memilih file, kemudian gambar dimuat menggunakan `PIL.Image.open()` dan ditampilkan dengan `ImageTk.PhotoImage`.

- **Screenshot:**

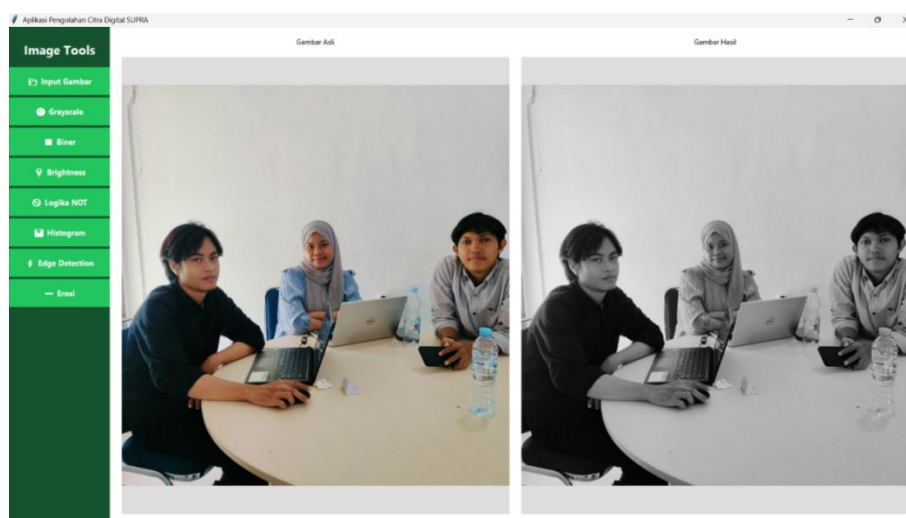


**Gambar 2: Input gambar**

## 2) Konversi ke Grayscale

Fitur ini mengonversi gambar berwarna (RGB) menjadi format hitam-putih dengan tingkat keabuan (grayscale). Setiap piksel hanya mewakili satu nilai intensitas antara 0 hingga 255. Konversi ini sangat penting karena mempermudah analisis lanjutan seperti deteksi tepi, thresholding, atau morfologi. Transformasi ke grayscale dilakukan dengan `cv2.cvtColor()` dari pustaka OpenCV, menggunakan parameter `COLOR_RGB2GRAY`. Hasilnya ditampilkan di panel sebelah kanan.

- **Screenshot:**



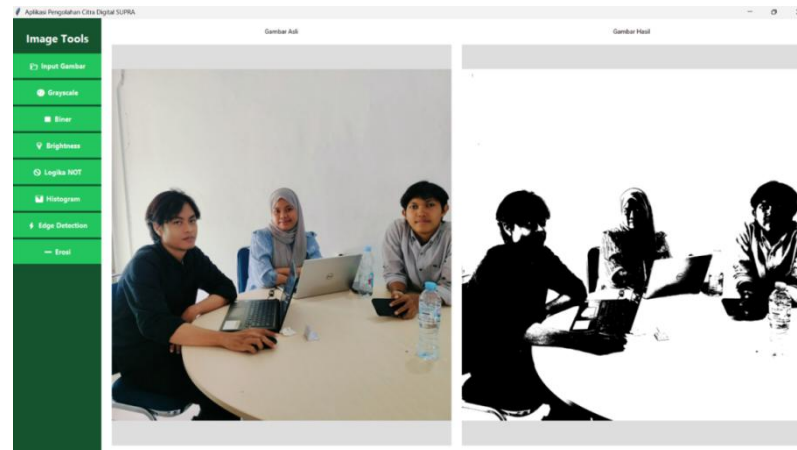
**Gambar 3: Konversi grayscale**

### 3) Konversi Citra Biner

Fungsi ini mengubah citra grayscale menjadi gambar dua warna (hitam-putih) melalui proses yang disebut thresholding. Setiap piksel akan bernilai 0 (hitam) atau 255 (putih), tergantung nilai ambangnya.

Langkah pertama adalah mengubah gambar ke grayscale, lalu menerapkan `cv2.threshold()` dengan ambang tetap 127. Piksel yang nilainya di atas ambang menjadi putih, sedangkan yang di bawah akan menjadi hitam. Fitur ini bermanfaat untuk segmentasi objek dan analisis bentuk.

- **Screenshot:**



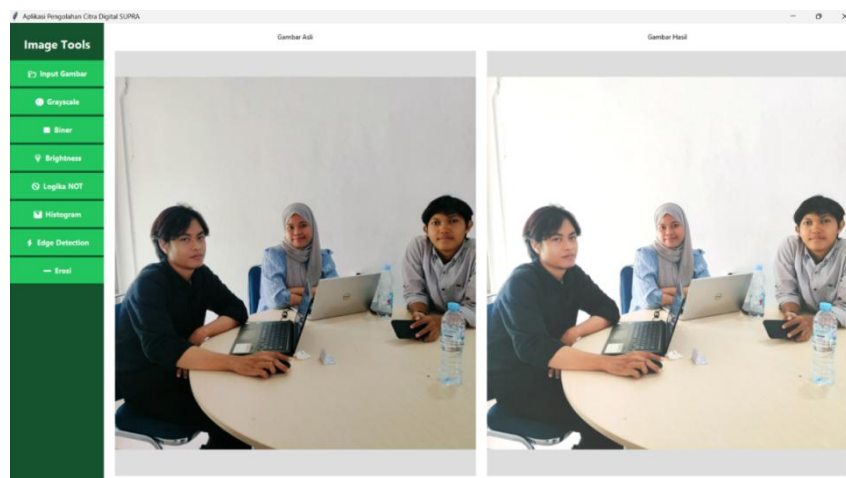
Gambar 4: Konversi citra biner

### 4) Konversi Operasi Aritmatika

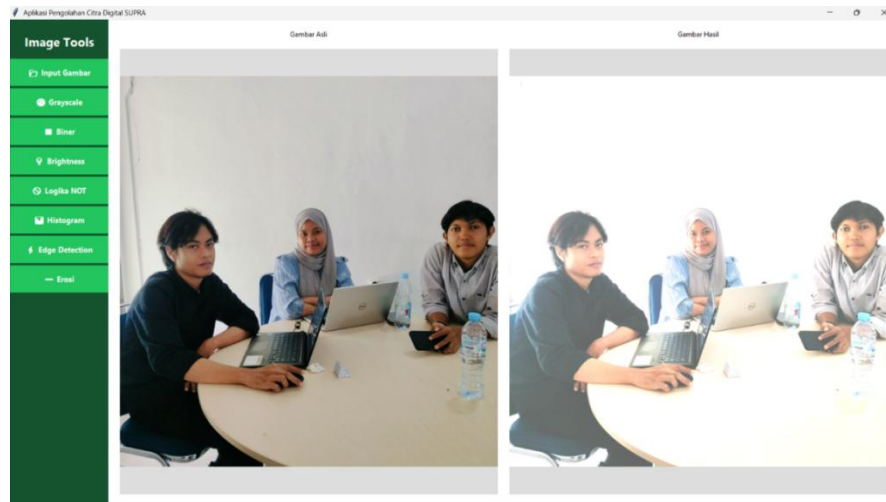
- **Brightness (Kecerahan)**

Fitur ini digunakan untuk mengatur tingkat terang pada gambar. Pengguna dapat menentukan nilai brightness secara manual, yang akan digunakan untuk mencerahkan tampilan gambar. Operasi dilakukan dengan `cv2.convertScaleAbs()`, yang menerapkan transformasi linier terhadap nilai intensitas piksel. Dalam konteks ini, nilai alpha tetap 1, dan beta adalah input dari pengguna. Semakin tinggi nilai beta, maka gambar menjadi semakin terang.

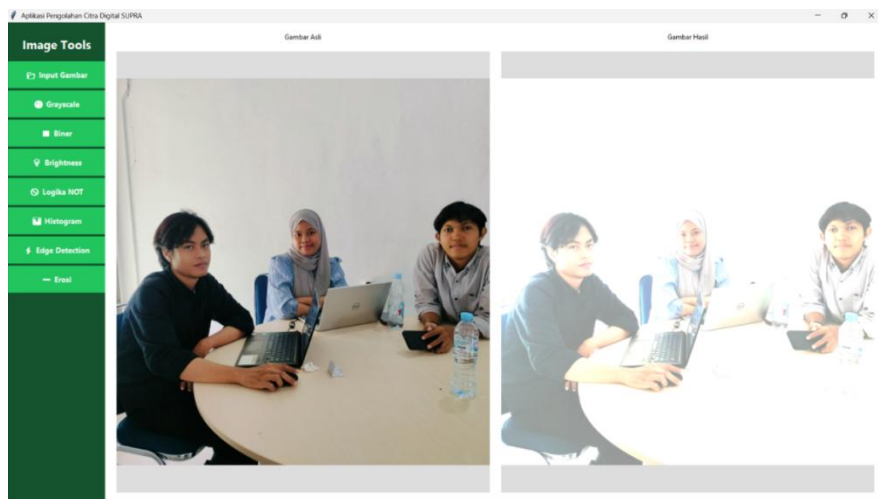
- **Screenshot:**



Gambar 5: Brightness nilai kecerahan 50



**Gambar 6: Brightness nilai kecerahan 100**



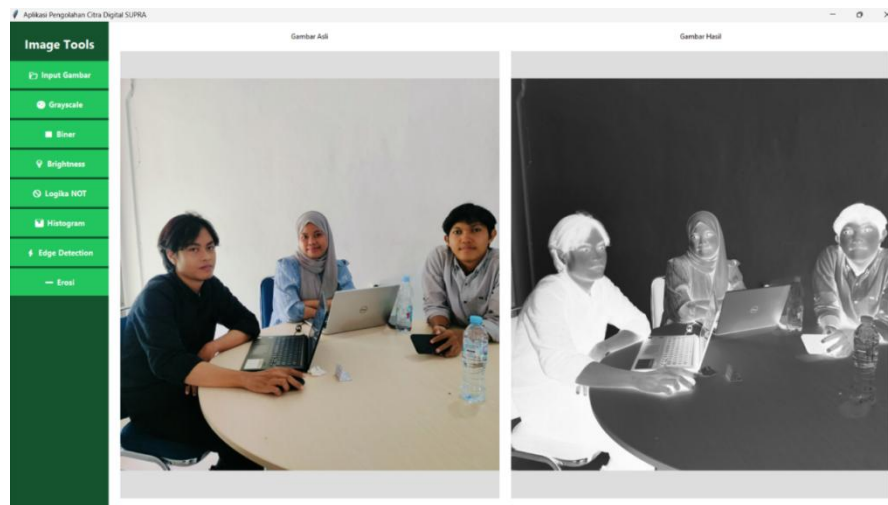
**Gambar 7: Brightness nilai kecerahan 150**

## 5. Konversi Operasi Logika

- NOT

Operasi logika NOT digunakan untuk membalik nilai intensitas pada gambar grayscale, menghasilkan efek negatif. Piksel yang awalnya terang menjadi gelap, dan sebaliknya. Secara teknis, nilai tiap piksel akan diubah dari  $x$  menjadi  $255 - x$ . Misalnya, piksel bernilai 0 akan berubah menjadi 255. Proses ini berguna untuk menonjolkan objek atau latar yang dominan terang atau gelap. Implementasinya menggunakan `cv2.bitwise_not()` dari OpenCV.

- Screenshot:



Gambar 8: Operasi logika NOT

### ➤ Fitur Optional

#### 1. Menampilkan Histogram

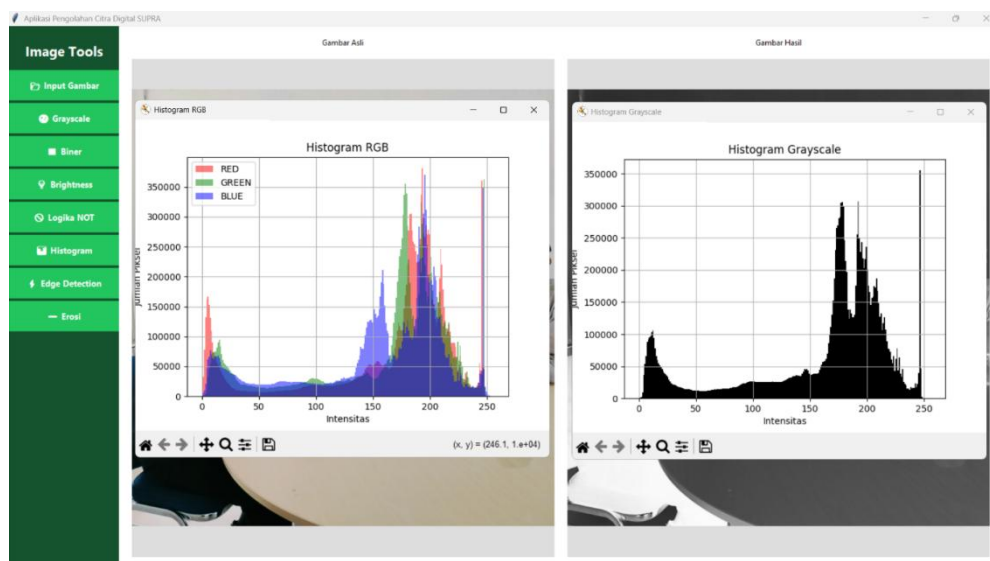
Histogram berfungsi untuk menampilkan persebaran nilai intensitas piksel dalam gambar. Grafik histogram memberikan informasi visual tentang pencahayaan, kontras, dan penyebaran warna pada gambar.

Terdapat dua tipe histogram yang dihasilkan:

- **Histogram RGB:** Menampilkan distribusi intensitas untuk kanal merah, hijau, dan biru.
- **Histogram Grayscale:** Menampilkan intensitas dalam format keabuan dari hitam ke putih.

Pembuatan histogram dilakukan menggunakan pustaka matplotlib.pyplot dan fungsi plt.hist().

- Screenshot



Gambar 9: Histogram RGB dan Histogram Grayscale

## 2. Konvolusi Edge Detection

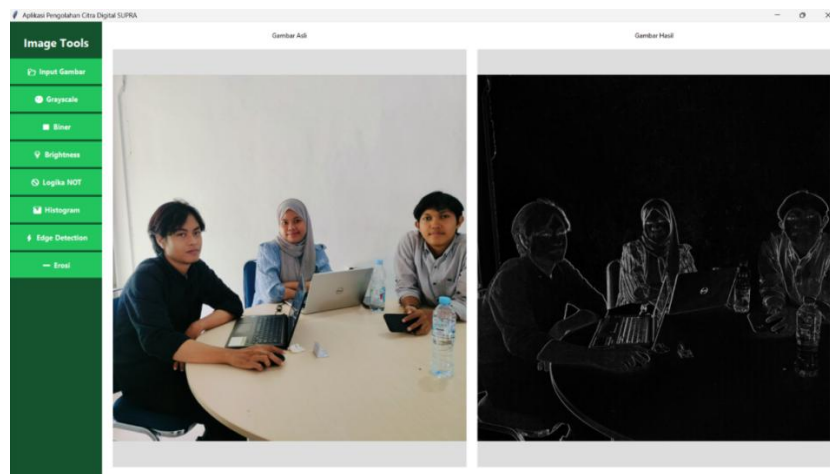
Fungsi ini mendeteksi batas-batas objek pada gambar. Deteksi tepi penting untuk segmentasi dan pengenalan bentuk.

Aplikasi menggunakan operator Sobel untuk mendeteksi perubahan intensitas:

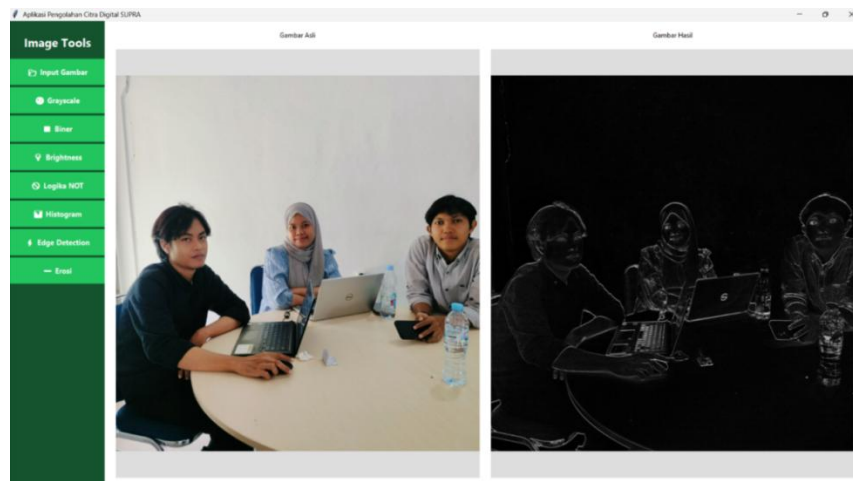
- Sobel Horizontal (SobelX): untuk mendeteksi tepi vertikal.
- Sobel Vertikal (SobelY): untuk mendeteksi tepi horizontal.

Gambar terlebih dahulu dikonversi ke grayscale, lalu Sobel diterapkan dengan `cv2.Sobel()`, dan hasilnya ditampilkan pada panel output.

- **Screenshot:**



**Gambar 10: Sobel vertical**



**Gambar 11: Sobel horizontal**

## 3. Operasi Morfologi Erosi

Fitur ini digunakan untuk menyederhanakan struktur objek putih (foreground) dalam gambar biner dengan mengurangi ukurannya atau menghilangkan noise kecil.

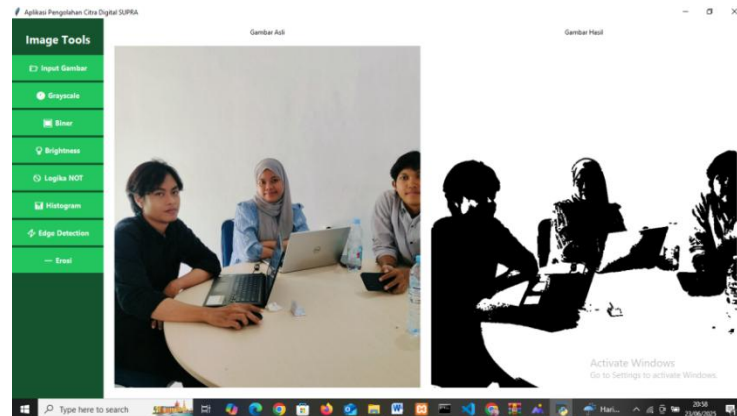
Pengguna dapat memilih jenis elemen penstruktur:



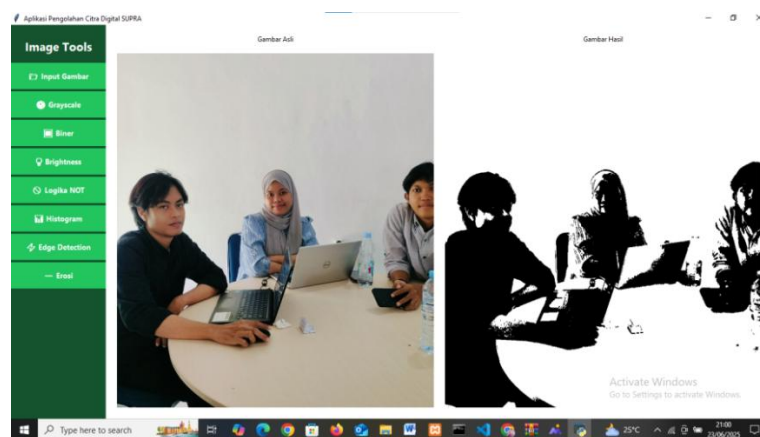
- **Vertikal:** bentuk garis tegak lurus 3 piksel,
- **Horizontal:** bentuk garis datar 3 piksel,
- **Kotak Penuh:** bentuk kotak 3x3 piksel.

Proses erosi menggunakan `cv2.erode()` dari OpenCV dengan elemen penstruktur yang didefinisikan lewat array NumPy.

- **Screenshot:**



**Gambar 12: Erosi kotak penuh**



**Gambar 13: Erosi vertikal**



**Gambar 14: Erosi horizontal**



### **C. LINK VIDEO dan LINK KODE PROGRAM**

- **Link Video:**

<https://drive.google.com/drive/folders/1gn4COootMg0G0vlrW7t3ftW9pxrsTLJT>

- **Link Github:**

[https://github.com/RyanHidayat2610/Proyek Citra Digital supra](https://github.com/RyanHidayat2610/Proyek_Citra_Digital_supra)