

LAPORAN PRAKTIKUM
IMAGE PROCESSING OPERASI BINER



Disusun Oleh :

Name : Sheiza Fakhru Rasyid

Npm : 092024090463

PRODI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA

2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum yang berjudul Operasi Biner pada Pengolahan Citra Digital (XOR, OR, dan NOT) dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu syarat untuk memenuhi kegiatan praktikum dalam mata kuliah Pengolahan Citra Digital.

Dalam praktikum ini, penulis mempelajari dan menerapkan berbagai jenis operasi biner pada citra digital, seperti OR, XOR, dan NOT, yang bertujuan untuk memahami bagaimana proses manipulasi citra dilakukan melalui operasi logika. Melalui kegiatan praktikum ini, penulis memperoleh wawasan mengenai cara kerja operasi logika pada piksel, serta bagaimana operasi tersebut dapat digunakan untuk segmentasi, pemfilteran, maupun deteksi objek dalam pengolahan citra.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis berharap adanya kritik dan saran yang bersifat membangun guna perbaikan laporan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta semua pihak yang telah membantu terlaksananya praktikum dan penyusunan laporan ini.

LEMBAR PENGESAHAN

Name : Sheiza Fakhru Rasyid

Npm : 092024090463

Depok, 29 November 2025

Disusun Oleh :

Dosen Pengampu

Mahasiswa

Dian Nugraha, S.ST., MIT

Sheiza Fakhru Rasyid

DAFTAR ISI

BAB I	1
PENDAHULUAN	1
1.1 Pendahuluan	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	1
BAB II	2
LANDASAN TEORI	2
2.1 Citra Greyscale	2
2.2 Biner	2
2.3 Operasi Biner	3
BAB III.....	4
IMPELEMENTASI.....	4
3.1 Operasi Biner AND	4
A. Gambar Operasi Biner AND	4
B. Code Operasi Biner AND	4
C. Analisis Code	4
3.2 Operasi Biner OR	6
A. Gambar Operasi Biner OR.....	6
B. Gambar Code Operasi Biner OR.....	7
C. Analisis Code	7
3.3 Operasi Biner XOR	10
A. Gambar Operasi Biner XOR	10
B. Gambar Code Operasi Biner	10

C. Analisis Code.....	10
3.4 Operasi Biner NOT.....	13
A. Gambar Operasi Biner NOT	13
B. Gambar Code Operasi Biner NOT.....	13
C. Analisis Code	13
BAB IV.....	15
APLIKASI OPERASI BINER	15
4.1 GUI APLIKASI.....	15
4.2 Upload Citra.....	15
A. Gambar Upload Citra.....	15
B. Gambar Code Upload Citra.....	16
C. Analisis Code	16
4.3 Operasi Biner AND	19
A. Gambar Operasi Biner AND	19
B. Gambar Code Operasi Biner	19
C. Analisis Code	19
4.4 Operasi Biner NOT.....	22
A. Gambar Operasi Biner NOT	22
B. Gambar Code Operasi Biner NOT.....	22
C. Analisis Code	23
4.5 Operasi Biner OR	25
A. Gambar Operasi Biner OR.....	25
B. Gambar Code Operasi Biner OR	25
C. Analisis Code	25
4.6 Operasi Biner XOR	28

A. Gambar Operasi Biner XOR	28
B. Gambar Code Operasi Biner XOR	28
C. Analisis Code	28
4.7 Operasi Biner Masking	31
A. Gambar Operasi Biner Masking	31
B. Gambar Code Operasi Biner MASKING	31
C. Analisis Code	31
DAFTAR PUSTAKA	34

BAB I

PENDAHULUAN

1.1 Pendahuluan

Pengolahan citra digital merupakan bidang ilmu yang mempelajari teknik-teknik untuk memanipulasi, menganalisis, dan meningkatkan citra digital. Salah satu konsep dasar dalam pengolahan citra adalah citra biner, yaitu citra yang hanya memiliki dua tingkat intensitas piksel: 0 (hitam) dan 1 (putih). Citra biner sering digunakan untuk menyederhanakan informasi citra, misalnya untuk segmentasi objek, deteksi tepi, dan analisis bentuk.

Operasi biner pada citra merupakan proses penerapan logika Boolean terhadap piksel-piksel citra biner. Operasi ini meliputi AND, OR, XOR, dan NOT, yang dapat digunakan untuk berbagai tujuan, seperti menggabungkan dua citra, mengekstraksi fitur tertentu, atau membalik citra.

- AND digunakan untuk mengekstraksi piksel yang bernilai 1 pada kedua citra sekaligus.
- OR digunakan untuk menghasilkan piksel bernilai 1 jika salah satu dari kedua citra bernilai 1.
- XOR menekankan perbedaan antara dua citra dengan menampilkan piksel bernilai 1 hanya jika piksel berbeda.
- NOT membalik nilai piksel dari 0 menjadi 1 dan sebaliknya.

Penggunaan operasi biner sangat penting dalam aplikasi seperti pengenalan pola, deteksi objek, dan pemrosesan dokumen. Dengan memahami operasi logika biner, kita dapat memanipulasi citra secara efisien untuk mengekstraksi informasi yang relevan dari citra digital.

1.2 Rumusan Masalah

1. Bagaimana mengimplementasikan operasi biner pada citra biner?
2. Bagaimana cara kerja operasi biner?
3. Bagaimana implementasi Operasi Biner menggunakan matlab?

1.3 Tujuan

1. Mengimplementasikan operasi biner pada citra biner
2. Mengetahui cara kerja operasi biner pada citra biner menggunakan matlab

BAB II

LANDASAN TEORI

2.1 Citra Greyscale

Citra grayscale adalah citra digital yang hanya memiliki satu komponen intensitas warna, yaitu tingkat keabuan (grayscale).

Berbeda dengan citra berwarna (RGB) yang menyimpan tiga komponen warna (Red, Green, Blue), citra grayscale hanya menyimpan nilai intensitas dari 0 sampai 255, di mana:

- 0 = hitam
- 255 = putih
- Nilai di antara keduanya = tingkat abu-abu

2.2 Biner

Citra biner adalah citra digital yang hanya memiliki dua nilai piksel, yaitu 0 dan 1, atau dalam skala intensitas sering direpresentasikan sebagai hitam (0) dan putih (255). Citra ini diperoleh dari proses *thresholding* terhadap citra grayscale, sehingga setiap piksel diklasifikasikan menjadi objek atau latar belakang.

Citra biner banyak digunakan dalam:

- Deteksi objek
- Segmentasi citra
- OCR (Optical Character Recognition)
- Analisis bentuk (*shape analysis*)
- Deteksi tepi sederhana

2.3 Operasi Biner

Operasi biner adalah operasi dalam matematika atau komputasi yang dilakukan antara dua buah operand.

Disebut *biner* karena melibatkan dua input untuk menghasilkan satu output.

Contoh operasi biner dalam citra:

1. AND

Piksel bernilai putih (1) hanya jika kedua piksel pada posisi yang sama bernilai putih.

2. OR

Piksel bernilai putih jika salah satu dari dua piksel bernilai putih.

3. XOR

Piksel bernilai putih jika kedua piksel berbeda.

4. NOT (walaupun ini operasi unary, sering dibahas bersama)

Membalik nilai piksel (0 jadi 1, 1 jadi 0).

5. Subtraction (Pengurangan Citra)

Mengurangi intensitas piksel citra A dengan citra B.

6. Addition (Penjumlahan Citra)

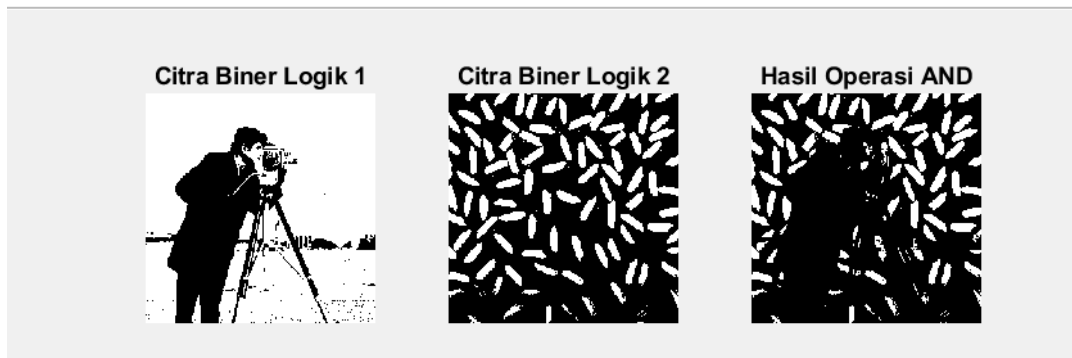
Menjumlahkan intensitas piksel citra A dan B.

BAB III

IMPELEMENTASI

3.1 Operasi Biner AND

A. Gambar Operasi Biner AND



B. Code Operasi Biner AND

```

1  A = imread('cameraman.tif');
2  B = imread('rice.png');
3
4  if size(A,3) == 3
5      citra_A = rgb2gray(A);
6  end
7  if size(B,3) == 3
8      citra_B = rgb2gray(B);
9  end
10
11  Biner_A = imbinarize(A);
12  Biner_B = imbinarize(B);
13  [r1, c1] = size(A);
14  C = false(r1, c1);
15
16  for x = 1:r1
17      for y = 1:c1
18          C(x,y) = Biner_A(x,y) & Biner_B(x,y);

```

```

15
16  for x = 1:r1
17      for y = 1:c1
18          C(x,y) = Biner_A(x,y) & Biner_B(x,y);
19      end
20  end
21
22  figure;
23  subplot(2,3,1);
24  imshow(Biner_A);
25  title('Citra Biner Logik 1');
26  subplot(2,3,2);
27  imshow(Biner_B);
28  title('Citra Biner Logik 2');
29  subplot(2,3,3);
30  imshow(C);
31  title('Hasil Operasi AND');
32

```

C. Analisis Code

Baris	Kode	Deskripsi
1	A = imread('cameraman.tif');	Membaca citra pertama (Citra A) dari file cameraman.tif.
2	B = imread('rice.png');	Membaca citra kedua (Citra B) dari file rice.png.

Baris	Kode	Deskripsi
4-6	<pre>if size(A,3) == 3 \n citra_A = rgb2gray(A); \n end</pre>	Memeriksa apakah Citra A berwarna. Jika ya, Citra A dikonversi ke grayscale dan disimpan di citra_A. (Catatan: Variabel A asli tampaknya tidak diperbarui, yang mungkin menjadi <i>bug</i> jika A berwarna dan kode selanjutnya menggunakan A).
7-9	<pre>if size(B,3) == 3 \n citra_B = rgb2gray(B); \n end</pre>	Memeriksa apakah Citra B berwarna. Jika ya, Citra B dikonversi ke grayscale dan disimpan di citra_B. (Catatan: Variabel B asli tampaknya tidak diperbarui).
11	<pre>Biner_A = imbinarize(A);</pre>	Mengubah Citra A (asli) menjadi Citra Biner (Biner_A).
12	<pre>Biner_B = imbinarize(B);</pre>	Mengubah Citra B (asli) menjadi Citra Biner (Biner_B).
13	<pre>[r1, c1] = size(A);</pre>	Mendapatkan dimensi (baris, kolom) dari Citra A asli (r1 = baris, c1 = kolom).
14	<pre>C = false(r1, c1);</pre>	Membuat matriks hasil C dengan ukuran yang sama, diinisialisasi dengan nilai logika false (0/Hitam).
16	<pre>for x = 1:r1</pre>	Memulai <i>loop</i> baris (dari 1 hingga r1).
17	<pre>for y = 1:c1</pre>	Memulai <i>loop</i> kolom di dalam <i>loop</i> baris (dari 1 hingga c1).
18	<pre>C(x,y) = Biner_A(x,y) & Biner_B(x,y);</pre>	Operasi Logika AND: Nilai <i>pixel</i> pada posisi \$(x,y)\$ di Citra Hasil \$C\$ diisi dengan hasil AND antara <i>pixel</i> yang bersesuaian pada

Baris	Kode	Deskripsi
		\$Biner_A\$ dan \$Biner_B\$. Tanda & adalah operator AND.
22	figure;	Membuat jendela <i>figure</i> baru untuk menampilkan hasil.
23	subplot(2,3,1);	Mengaktifkan sub-plot posisi 1.
24	imshow(Biner_A);	Menampilkan Biner_A.
25	title('Citra Biner Logik 1');	Memberi judul.
26	subplot(2,3,2);	Mengaktifkan sub-plot posisi 2.
27	imshow(Biner_B);	Menampilkan Biner_B.
28	title('Citra Biner Logik 2');	Memberi judul.
29	subplot(2,3,3);	Mengaktifkan sub-plot posisi 3.
30	imshow(C);	Menampilkan Citra Hasil \$C\$.
31	title('Hasil Operasi AND');	Memberi judul pada hasil.

3.2 Operasi Biner OR

A. Gambar Operasi Biner OR



B. Gambar Code Operasi Biner OR

```

Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\operasi_biner.m *
operasi_biner.m *
1  A = imread('cameraman.tif');
2  B = imread('rice.png');
3
4  if size(A,3) == 3
5      A = rgb2gray(A);
6  end
7  if size(B,3) == 3
8      B = rgb2gray(B);
9  end
10
11  B = imresize(B, [size(A,1) size(A,2)]);
12  Biner_A = imbinarize(A);
13  Biner_B = imbinarize(B);
14
15  [r1, c1] = size(Biner_A);
16  C = false(r1, c1);
17
18  for x = 1:r1

```

```

17
18  for x = 1:r1
19      for y = 1:c1
20          C(x,y) = Biner_A(x,y) | Biner_B(x,y); % OR
21      end
22  end
23
24  figure;
25  subplot(2,3,1); imshow(Biner_A); title('Citra Biner 1');
26  subplot(2,3,2); imshow(Biner_B); title('Citra Biner 2');
27  subplot(2,3,3); imshow(C); title('Hasil Operasi OR');
28

```

C. Analisis Code

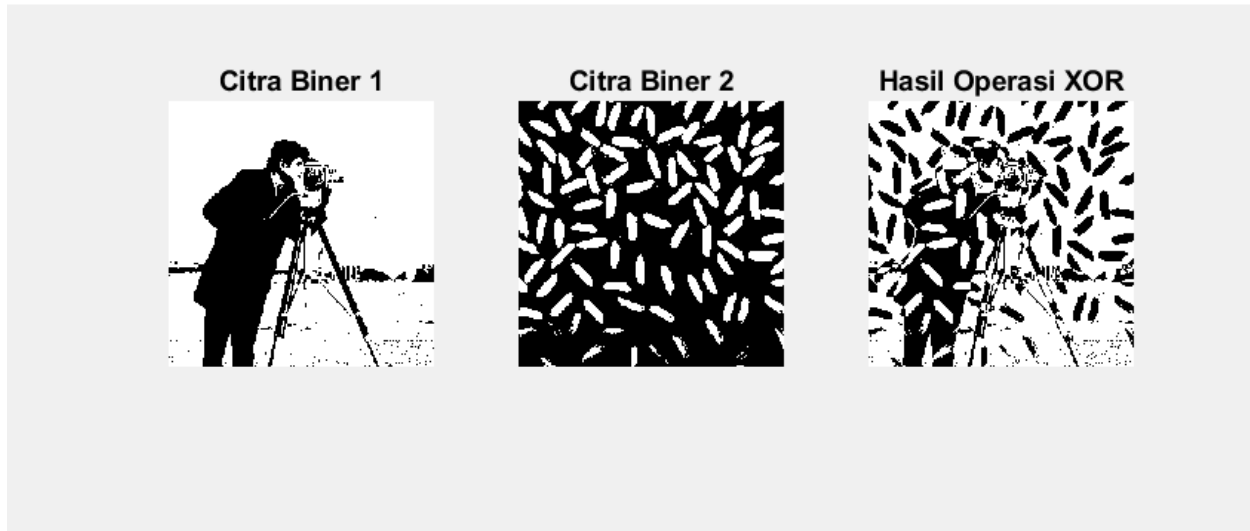
Baris	Kode	Deskripsi
1	<code>A = imread('cameraman.tif');</code>	Membaca citra pertama (Citra A) dari file cameraman.tif.
2	<code>B = imread('rice.png');</code>	Membaca citra kedua (Citra B) dari file rice.png.
4	<code>if size(A,3) == 3</code>	Memeriksa apakah Citra A adalah citra berwarna (memiliki 3 dimensi: R, G, B).
5	<code>A = rgb2gray(A);</code>	Jika berwarna, Citra A diubah menjadi citra grayscale.

Baris	Kode	Deskripsi
7	<code>if size(B,3) == 3</code>	Memeriksa apakah Citra B adalah citra berwarna.
8	<code>B = rgb2gray(B);</code>	Jika berwarna, Citra B diubah menjadi citra grayscale.
11	<code>B = imresize(B, [size(A,1) size(A,2)]);</code>	Mengubah ukuran (resize) Citra B agar dimensinya sama persis dengan Citra A (size(A,1) adalah baris, size(A,2) adalah kolom).
12	<code>Biner_A = imbinarize(A);</code>	Mengubah Citra A (grayscale) menjadi Citra Biner (Biner_A).
13	<code>Biner_B = imbinarize(B);</code>	Mengubah Citra B (grayscale, setelah di-resize) menjadi Citra Biner (Biner_B).
15	<code>[r1, c1] = size(Biner_A);</code>	Mendapatkan dimensi Citra Biner A (r1 = baris, c1 = kolom).
16	<code>C = false(r1, c1);</code>	Membuat matriks hasil C dengan ukuran yang sama, diinisialisasi dengan nilai logika false (0).
17	<code>for x = 1:r1</code>	Memulai <i>loop</i> baris (dari 1 hingga r1).
18	<code>for y = 1:c1</code>	Memulai <i>loop</i> kolom di dalam <i>loop</i> baris (dari 1 hingga c1).
19	<code>C(x,y) = Biner_A(x,y) Biner_B(x,y); \% OR</code>	Operasi Logika OR: Nilai <i>pixel</i> pada posisi \$(x,y)\$ di Citra Hasil \$C\$ diisi dengan hasil OR antara <i>pixel</i> yang bersesuaian pada

Baris	Kode	Deskripsi
		\$Biner_A\$ dan \$Biner_B\$. Tanda adalah operator OR.
24	figure;	Membuat jendela <i>figure</i> baru untuk menampilkan hasil.
25	subplot(2,3,1); imshow(Biner_A); title('Citra Biner 1');	Menampilkan Biner_A di sub-plot posisi 1.
26	subplot(2,3,2); imshow(Biner_B); title('Citra Biner 2');	Menampilkan Biner_B di sub-plot posisi 2.
27	subplot(2,3,3); imshow(C); title('Hasil Operasi OR');	Menampilkan Citra Hasil \$C\$ di sub-plot posisi 3.

3.3 Operasi Biner XOR

A. Gambar Operasi Biner XOR



B. Gambar Code Operasi Biner

```

Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\operasi_biner.m *
operasi_biner.m *
1  A = imread('cameraman.tif');
2  B = imread('rice.png');
3
4  if size(A,3) == 3
5      A = rgb2gray(A);
6  end
7  if size(B,3) == 3
8      B = rgb2gray(B);
9  end
10
11  B = imresize(B, [size(A,1) size(A,2)]);
12  Biner_A = imbinarize(A);
13  Biner_B = imbinarize(B);
14
15  [r1, c1] = size(Biner_A);
16  C = false(r1, c1);
17
18  for x = 1:r1
19
20      for y = 1:c1
21          C(x,y) = xor(Biner_A(x,y), Biner_B(x,y)); % XOR
22      end
23  end
24
25  figure;
26  subplot(2,3,1); imshow(Biner_A); title('Citra Biner 1');
27  subplot(2,3,2); imshow(Biner_B); title('Citra Biner 2');
28  subplot(2,3,3); imshow(C); title('Hasil Operasi XOR');

```

C. Analisis Code

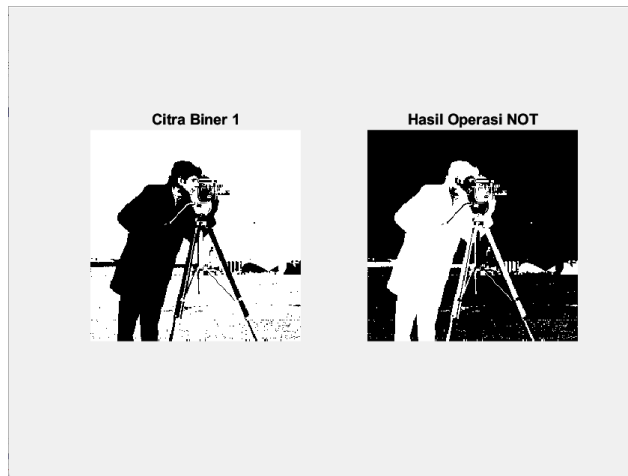
Baris	Perintah	Deskripsi/Fungsi
1-2	<pre> A = imread('cameraman.tif'); \$newline\$ B = imread('rice.png'); </pre>	Memuat Citra: Membaca dua citra (cameraman.tif dan rice.png) ke dalam variabel A dan B.

Baris	Perintah	Deskripsi/Fungsi
4-6	if size(A,3) == 3 ... A = rgb2gray(A); end	Konversi ke Grayscale (A): Mengecek apakah citra A memiliki 3 dimensi (citra RGB). Jika ya, citra dikonversi dari RGB menjadi Grayscale (citra keabuan) menggunakan rgb2gray.
7-9	if size(B,3) == 3 ... B = rgb2gray(B); end	Konversi ke Grayscale (B): Mengecek apakah citra B memiliki 3 dimensi (citra RGB). Jika ya, citra dikonversi dari RGB menjadi Grayscale.
11	B = imresize(B, [size(A,1) size(A,2)]);	Penyesuaian Ukuran: Mengubah ukuran citra B agar sama persis dengan ukuran citra A (jumlah baris size(A,1) dan kolom size(A,2)). Ini penting untuk operasi biner.
12-13	Biner_A = imbinarize(A); Biner_B = imbinarize(B);	Binarisasi: Mengkonversi citra Grayscale A dan B menjadi citra Biner (hanya berisi nilai 0 dan 1/false dan true) menggunakan imbinarize.
15	[r1, c1] = size(Biner_A);	Dimensi Citra: Mendapatkan jumlah baris (r1) dan kolom (c1) dari citra biner (yang ukurannya sudah disamakan).
16	C = false(r1, c1);	Inisialisasi Hasil: Membuat matriks hasil C berukuran r1 x c1 yang diinisialisasi dengan nilai false (0) untuk menyimpan hasil operasi XOR.

Baris	Perintah	Deskripsi/Fungsi
18-22	for x = 1:r1 ... for y = 1:c1 ... end end	Looping Piksel: Membuat loop bersarang (nested loop) untuk mengiterasi melalui setiap baris (x) dan kolom (y) dari kedua citra biner.
20	C(x,y) = xor(Biner_A(x,y), Biner_B(x,y));	Operasi XOR: Melakukan operasi XOR antara piksel pada koordinat \$(x,y)\$ dari Biner_A dan Biner_B. Hasilnya (true/1 jika piksel berbeda, false/0 jika piksel sama) disimpan dalam matriks hasil C.
24	figure;	Jendela Gambar: Membuat jendela gambar baru untuk menampilkan citra.
25-27	subplot(2,3,1); imshow(Biner_A); title(...) \$\newline\$ subplot(2,3,2); imshow(Biner_B); title(...) \$\newline\$ subplot(2,3,3); imshow(C); title(...)	Menampilkan Hasil: Menggunakan subplot untuk menampilkan tiga citra (Citra Biner 1, Citra Biner 2, dan Hasil Operasi XOR) dalam satu jendela gambar dengan tata letak 2 baris dan 3 kolom.

3.4 Operasi Biner NOT

A. Gambar Operasi Biner NOT



B. Gambar Code Operasi Biner NOT

```

Editor - C:\Users\seiza\OneDrive\Documents\MATLAB\operasi_biner.m
operasi_biner.m
1  A = imread('cameraman.tif');
2
3  if size(A,3) == 3
4      A = rgb2gray(A);
5  end
6
7  Biner_A = imbinarize(A);
8
9  [r1, c1] = size(Biner_A);
10 C = false(r1, c1);
11
12 for x = 1:r1
13     for y = 1:c1
14         C(x,y) = ~Biner_A(x,y);
15     end
16 end
17
18 figure;
19
20 figure;
21 subplot(1,2,1); imshow(Biner_A); title('Citra Biner 1');
22 subplot(1,2,2); imshow(C); title('Hasil Operasi NOT');

```

C. Analisis Code

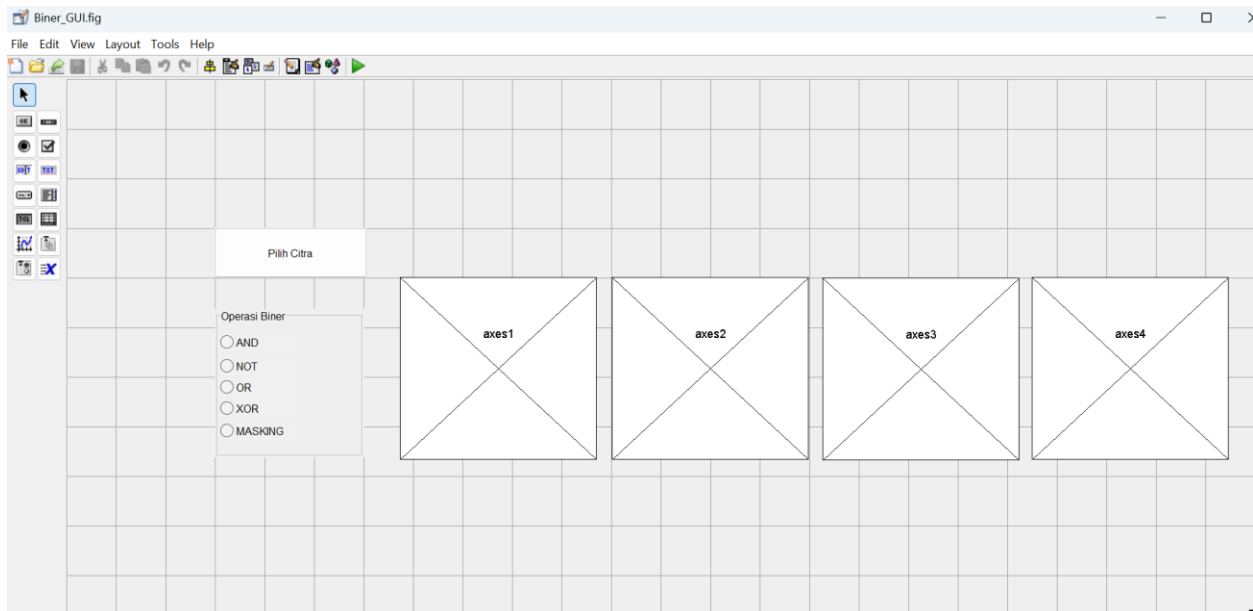
Baris	Perintah	Deskripsi/Fungsi
1	<code>A = imread('cameraman.tif');</code>	Memuat Citra: Membaca citra (cameraman.tif) ke dalam variabel A.
3-5	<code>if size(A,3) == 3 ... A = rgb2gray(A); end</code>	Konversi ke Grayscale: Mengecek apakah citra A adalah citra RGB (3 dimensi). Jika ya, citra dikonversi menjadi Grayscale (keabuan) menggunakan rgb2gray.

Baris	Perintah	Deskripsi/Fungsi
7	<code>Biner_A = imbinarize(A);</code>	Binarisasi: Mengkonversi citra Grayscale A menjadi citra Biner (hanya berisi nilai 0 dan 1/false dan true) menggunakan imbinarize.
9	<code>[r1, c1] = size(Biner_A);</code>	Dimensi Citra: Mendapatkan jumlah baris (r1) dan kolom (c1) dari citra biner.
10	<code>C = false(r1, c1);</code>	Inisialisasi Hasil: Membuat matriks hasil C berukuran r1 x c1 yang diinisialisasi dengan nilai false (0) untuk menyimpan hasil operasi NOT.
12-16	<code>for x = 1:r1 ... for y = 1:c1 ... end end</code>	Looping Pixel: Membuat loop bersarang (nested loop) untuk mengiterasi melalui setiap baris (x) dan kolom (y) dari citra biner.
14	<code>C(x,y) = ~Biner_A(x,y);</code>	Operasi NOT: Melakukan operasi NOT (Negasi) pada pixel pada koordinat \$(x,y)\$ dari Biner_A. Operator ~ (tilde) membalikkan nilai pixel: 0 menjadi 1 dan 1 menjadi 0. Hasilnya disimpan dalam matriks C.
18	<code>figure;</code>	Jendela Gambar: Membuat jendela gambar baru untuk menampilkan citra.
19	<code>subplot(1,2,1); imshow(Biner_A); title('Citra Biner 1');</code>	Menampilkan Citra Asli: Menampilkan citra biner asli (Biner_A) di posisi pertama (1 baris, 2 kolom, posisi 1).
20	<code>subplot(1,2,2); imshow(C); title('Hasil Operasi NOT');</code>	Menampilkan Hasil: Menampilkan citra hasil operasi NOT (C) di posisi kedua (1 baris, 2 kolom, posisi 2).

BAB IV

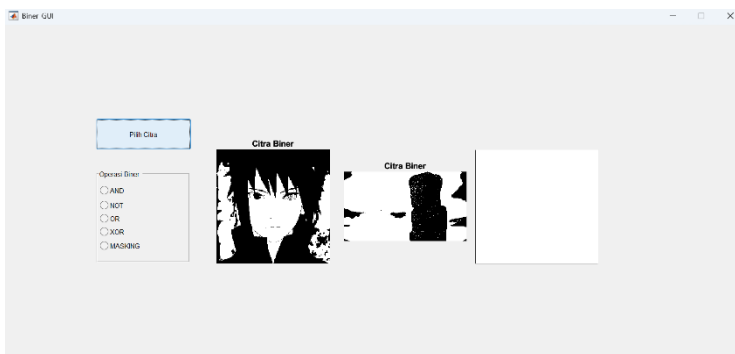
APLIKASI OPERASI BINER

4.1 GUI APLIKASI



4.2 Upload Citra

A. Gambar Upload Citra



B. Gambar Code Upload Citra

```

Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m *
operasi_biner.m  Biner_GUI.m *  untitled.m  operasi_biner.asv  +
77 function pushbutton1_Callback(hObject, eventdata, handles)
78 guidata(hObject, handles);
79 movegui(handles.figure1, 'center');
80
81 if ~isfield(handles, 'uploadCount')
82     handles.uploadCount = 0;
83 end
84
85 [nama_file, nama_path] = uigetfile({'*.jpg'; '*.bmp'; '*.png'; '*.tif'}, 'Buka Citra');
86
87 if ~isequal(nama_file, 0)
88     data = imread(fullfile(nama_path, nama_file));
89     citra_grey = rgb2gray(data);
90     citra_biner = imbinarize(citra_grey);
91
92     handles.uploadCount = handles.uploadCount + 1;
93

```

```

Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m  Biner_GUI.m  untitled.m  operasi_biner.asv  +
94 if handles.uploadCount == 1
95     handles.data1 = citra_biner;
96     axes(handles.axes1);
97     imshow(citra_biner);
98     title('Citra Biner');
99 elseif handles.uploadCount == 2
100     handles.data2 = citra_biner;
101     axes(handles.axes2);
102     imshow(citra_biner);
103     title('Citra Biner');
104 else
105     end
106 guidata(hObject, handles);
107 set(handles.axes4, 'Visible', 'off');
108 else
109     return
110 end
111

```

C. Analisis Code

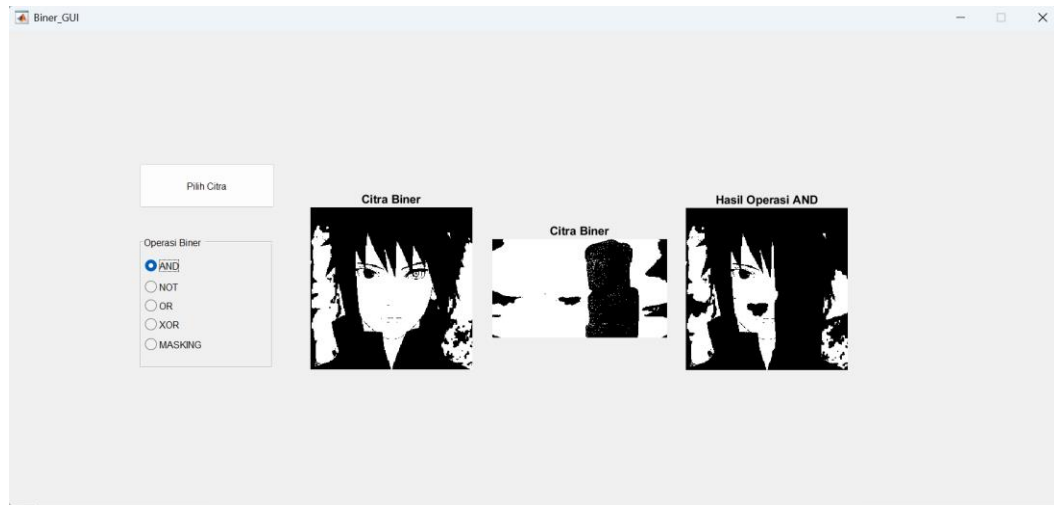
Baris	Kode	Deskripsi
77	function pushbutton1_Callback(hObject, eventdata, handles)	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat tombol (pushbutton) diklik.
78	guidata(hObject, handles);	Menyimpan struktur handles yang diperbarui.
79	movegui(handles.figure1, 'center');	Memindahkan jendela GUI (figure1) ke posisi tengah layar.
81-83	if ~isfield(handles, 'uploadCount') \n handles.uploadCount = 0; \n end	Memeriksa apakah <i>field</i> uploadCount sudah ada di struktur handles. Jika belum ada, ia diinisialisasi menjadi 0.
85	[nama_file, nama_path] = uigetfile({'*.jpg'; '*.bmp'; '*.png'; '*.tif'}, 'Buka Citra');	Membuka dialog "Buka File" untuk memilih file citra dengan ekstensi yang diizinkan (.jpg, .bmp, .png, .tif). Nama file dan <i>path</i> disimpan di variabel.

Baris	Kode	Deskripsi
87	<code>if ~isequal(nama_file,0)</code>	Memeriksa apakah pengguna tidak membatalkan pemilihan file (nilai <code>nama_file</code> akan menjadi 0 jika dibatalkan).
88	<code>data = imread(fullfile(nama_path,nama_file));</code>	Membaca file citra yang dipilih ke dalam variabel <code>data</code> .
89	<code>citra_grey = rgb2gray(data);</code>	Mengubah citra <code>data</code> (jika berwarna/RGB) menjadi citra <i>grayscale</i> .
90	<code>citra_biner = imbinarize(citra_grey);</code>	Mengubah citra <i>grayscale</i> menjadi citra biner (hanya berisi nilai 0 atau 1).
92	<code>handles.uploadCount = handles.uploadCount + 1;</code>	Menambah nilai penghitung unggahan (<i>upload count</i>).
94- 106	<code>if handles.uploadCount == 1 ... elseif handles.uploadCount == 2 ... else end</code>	Logika tampilan citra berdasarkan berapa kali tombol diklik/citra diunggah:
94- 98	<code>if handles.uploadCount == 1</code>	Jika ini adalah unggahan pertama: Simpan <code>citra_biner</code> ke <code>handles.data1</code> , tampilkan di <code>handles.axes1</code> , dan beri judul 'Citra Biner'.
99- 103	<code>elseif handles.uploadCount == 2</code>	Jika ini adalah unggahan kedua: Simpan <code>citra_biner</code> ke <code>handles.data2</code> , tampilkan di

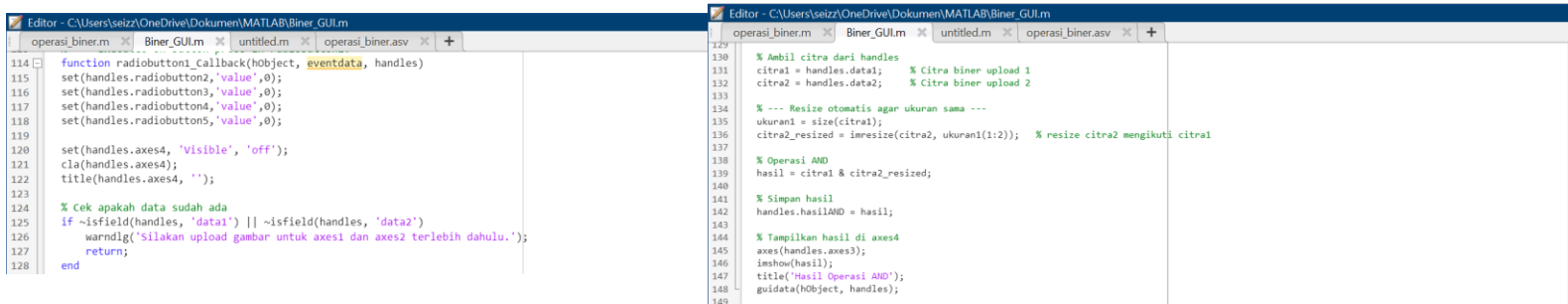
Baris	Kode	Deskripsi
		handles.axes2, dan beri judul 'Citra Biner'.
107	<code>guidata(hObject, handles);</code>	Menyimpan struktur handles yang sudah diperbarui (termasuk uploadCount, data1, data2).
108	<code>set(handles.axes4, 'Visible', 'off');</code>	Mengatur sumbu (axes4) menjadi tidak terlihat.
109-111	<code>else \n return \n end</code>	Jika pengguna membatalkan pemilihan file (Baris 87), fungsi akan keluar (return).

4.3 Operasi Biner AND

A. Gambar Operasi Biner AND



B. Gambar Code Operasi Biner



C. Analisis Code

Baris	Kode	Deskripsi
114	<code>function radiobutton1_Callback(hObject, eventdata, handles)</code>	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 1 diklik (diperkirakan ini adalah tombol untuk operasi AND).
115- 118	<code>set(handles.radiobutton2, 'Value', 0); (dan seterusnya)</code>	Mengatur nilai (Value) dari radiobutton lain (2, 3, 4, 5) menjadi

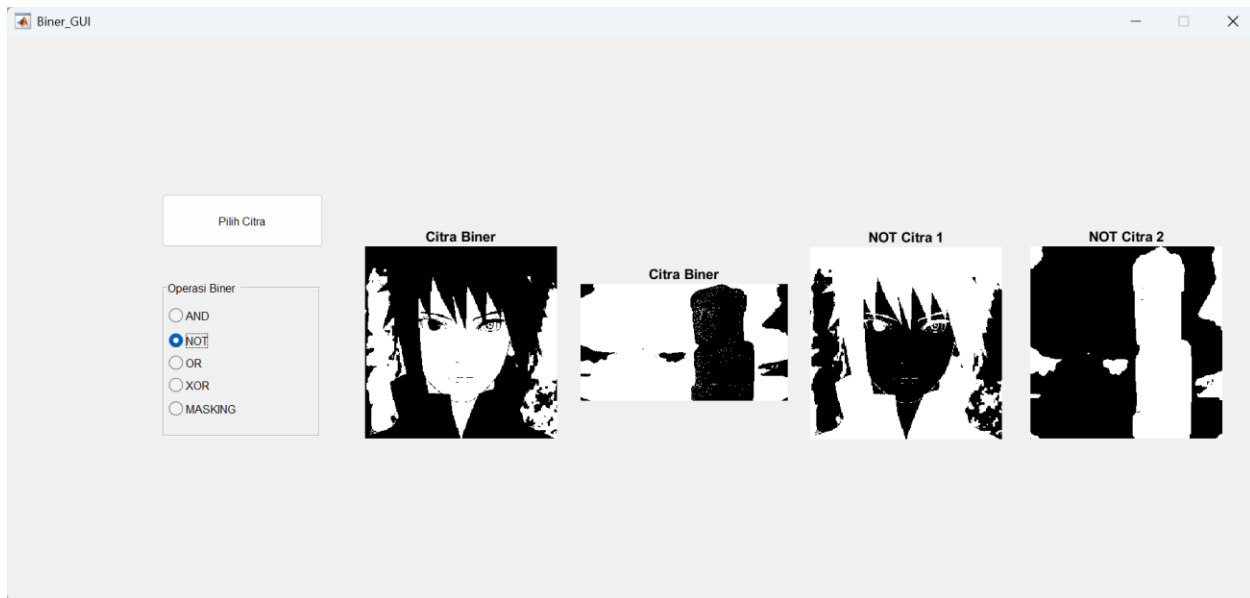
Baris	Kode	Deskripsi
		0 (tidak terpilih). Ini memastikan hanya satu operasi yang aktif.
120	<code>set(handles.axes4, 'Visible', 'off');</code>	Mengatur sumbu (axes4) menjadi tidak terlihat.
121	<code>cla(handles.axes4);</code>	Membersihkan sumbu (axes4).
122	<code>title(handles.axes4, "");</code>	Menghapus judul dari sumbu (axes4).
125	<code>if ~isfield(handles, 'data1') ~isfield(handles, 'data2')</code>	Memeriksa apakah kedua citra (data1 dan data2) sudah ada di struktur handles.
126- 128	<code>warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');</code> <code>return; end</code>	Jika salah satu atau kedua citra belum ada, tampilkan pesan peringatan dan keluar (return) dari fungsi.
130	<code>\% Ambil citra dari handles</code>	Komentar: Bagian ini mengambil data citra.
131	<code>citra1 = handles.data1;</code>	Mengambil citra biner pertama yang diunggah.
132	<code>citra2 = handles.data2;</code>	Mengambil citra biner kedua yang diunggah.
134	<code>\% --- Resize otomatis agar ukuran sama ---</code>	Komentar: Bagian ini memastikan kedua citra memiliki dimensi yang sama.

Baris	Kode	Deskripsi
135	<code>ukuran1 = size(citra1);</code>	Mendapatkan dimensi (baris, kolom) dari citra1.
136	<code>citra2_resized = imresize(citra2, ukuran1(1:2));</code>	Mengubah ukuran citra2 agar sesuai dengan dimensi baris dan kolom dari citra1.
138	<code>\% Operasi AND</code>	Komentar: Bagian ini melakukan operasi logika inti.
139	<code>hasil = citra1 & citra2_resized;</code>	Melakukan operasi Logika AND (element-wise) antara citra1 dan citra2 yang sudah diubah ukurannya.
141	<code>\% Simpan hasil</code>	Komentar: Menyimpan hasil operasi.
142	<code>handles.hasilAND = hasil;</code>	Menyimpan hasil citra AND ke <i>field</i> baru handles.hasilAND.
144	<code>\% Tampilkan hasil di axes4</code>	Komentar: Menampilkan hasil operasi.
145	<code>axes(handles.axes3);</code>	Mengaktifkan axes3 sebagai <i>current axes</i> . (Perhatikan: Komentar B144 menyebut axes4, tetapi kodenya menggunakan axes3).
146	<code>imshow(hasil);</code>	Menampilkan citra hasil operasi AND.
147	<code>title('Hasil Operasi AND');</code>	Memberi judul pada <i>axis</i> yang menampilkan hasil.

Baris	Kode	Deskripsi
148	<code>guidata(hObject, handles);</code>	Menyimpan struktur handles yang sudah diperbarui (termasuk <code>handles.hasilAND</code>).

4.4 Operasi Biner NOT

A. Gambar Operasi Biner NOT



B. Gambar Code Operasi Biner NOT

```

165 citra1 = handles.data1;
166 citra2 = handles.data2;
167 ukuran1 = size(citra1);
168 citra2_resized = imresize(citra2, ukuran1(1:2));
169
170 hasilNOT1 = ~citra1;
171 hasilNOT2 = ~citra2_resized;
172 handles.hasilNOT1 = hasilNOT1;
173 handles.hasilNOT2 = hasilNOT2;
174
175 axes(handles.axes3);
176 imshow(hasilNOT1);
177 title('NOT Citra 1');
178 axes(handles.axes4);
179 imshow(hasilNOT2);
180 title('NOT Citra 2');
181 guidata(hObject, handles);
182

```

```

152
153
154 % --- Executes on button press in radiobutton2.
155 function radiobutton2_Callback(hObject, eventdata, handles)
156 set(handles.radiobutton1,'value',0);
157 set(handles.radiobutton3,'value',0);
158 set(handles.radiobutton4,'value',0);
159 set(handles.radiobutton5,'value',0);
160
161 if ~isfield(handles, 'data1') || ~isfield(handles, 'data2')
162     warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');
```

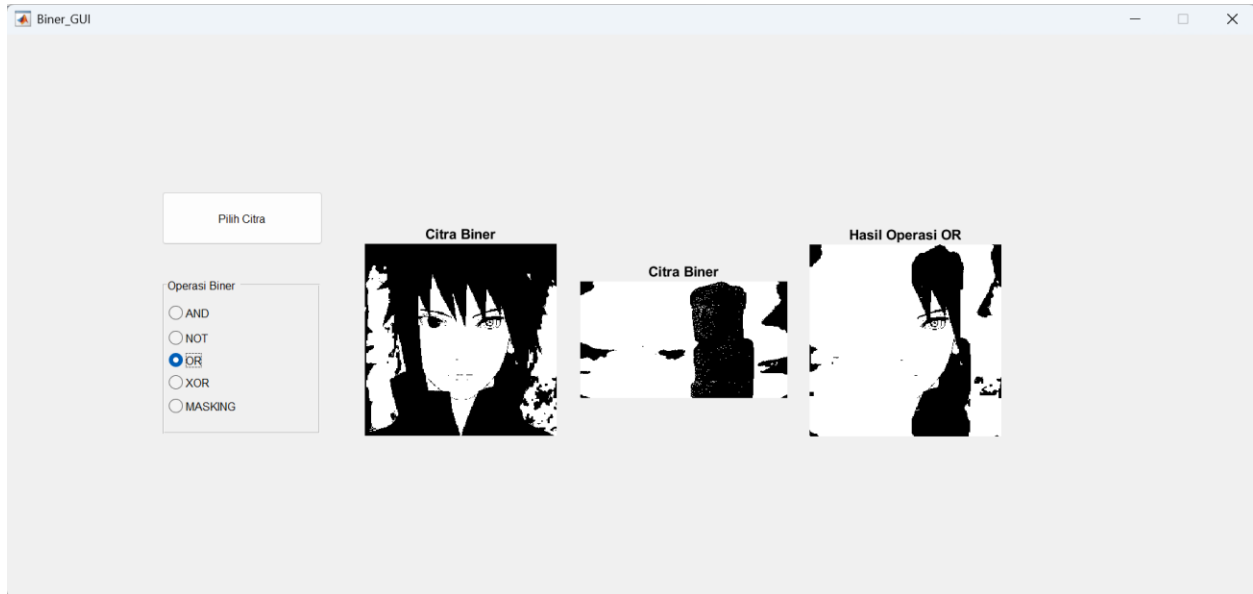
C. Analisis Code

Baris	Kode	Deskripsi
155	<code>function radiobutton2_Callback(hObject, eventdata, handles)</code>	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 2 diklik (diperkirakan ini adalah tombol untuk operasi NOT).
156- 159	<code>set(handles.radiobutton1, 'Value', 0); (dan seterusnya)</code>	Mengatur nilai (Value) dari radiobutton lain (1, 3, 4, 5) menjadi 0 (tidak terpilih).
161	<code>if ~isfield(handles, 'data1') ~isfield(handles, 'data2')</code>	Memeriksa apakah kedua citra (data1 dan data2) sudah ada di struktur handles.
162- 164	<code>warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');</code> <code>return; end</code>	Jika salah satu atau kedua citra belum ada, tampilkan pesan peringatan dan keluar (return) dari fungsi.
166	<code>citra1 = handles.data1;</code>	Mengambil citra biner pertama yang diunggah.
167	<code>citra2 = handles.data2;</code>	Mengambil citra biner kedua yang diunggah.
168	<code>ukuran1 = size(citra1);</code>	Mendapatkan dimensi dari citra1.
169	<code>citra2_resized = imresize(citra2, ukuran1(1:2));</code>	Mengubah ukuran citra2 agar memiliki dimensi yang sama dengan citra1.

Baris	Kode	Deskripsi
170	hasilNOT1 = ~citra1;	Melakukan Operasi Logika NOT pada citra1. Tanda ~ (tilde) membalikkan nilai <i>pixel</i> .
171	hasilNOT2 = ~citra2_resized;	Melakukan Operasi Logika NOT pada citra2 yang sudah diubah ukurannya.
172	handles.hasilNOT1 = hasilNOT1;	Menyimpan hasil NOT dari citra 1.
173	handles.hasilNOT2 = hasilNOT2;	Menyimpan hasil NOT dari citra 2.
175	axes(handles.axes3);	Mengaktifkan axes3 untuk menampilkan hasil pertama.
176	imshow(hasilNOT1);	Menampilkan hasil operasi NOT pada citra 1.
177	title('NOT Citra 1');	Memberi judul pada axes3.
178	axes(handles.axes4);	Mengaktifkan axes4 untuk menampilkan hasil kedua.
179	imshow(hasilNOT2);	Menampilkan hasil operasi NOT pada citra 2.
180	title('NOT Citra 2');	Memberi judul pada axes4.
181	guidata(hObject, handles);	Menyimpan struktur handles yang sudah diperbarui (termasuk handles.hasilNOT1 dan handles.hasilNOT2).

4.5 Operasi Biner OR

A. Gambar Operasi Biner OR



B. Gambar Code Operasi Biner OR

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Binier_GUI.m
operasi_binier.m  Binier_GUI.m  untitled.m  operasi_binier.asv  +
202 % Ambil citra biner dari handles
203 citra1 = handles.data1; % Citra biner upload 1
204 citra2 = handles.data2; % Citra biner upload 2
205
206 % --- Resize otomatis supaya ukuran sama ---
207 ukuran1 = size(citra1);
208 citra2_resized = imresize(citra2, ukuran1(1:2));
209
210 % --- Operasi OR ---
211 hasil = citra1 | citra2_resized;
212
213 % Simpan hasil jika dibutuhkan
214 handles.hasilOR = hasil;
215
216 % Tampilkan hasil di axes4
217 axes(handles.axes4);
218 imshow(hasil);
219 title('Hasil Operasi OR');
220 guidata(hObject, handles);
221
```

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Binier_GUI.m
operasi_binier.m  Binier_GUI.m  untitled.m  operasi_binier.asv  +
183
184 % --- Executes on button press in radiobutton3.
185 function radiobutton3_Callback(hObject, eventdata, handles)
186 set(handles.radiobutton1,'value',0);
187 set(handles.radiobutton2,'value',0);
188 set(handles.radiobutton4,'value',0);
189 set(handles.radiobutton5,'value',0);
190
191 set(handles.axes4, 'Visible', 'off');
192 cla(handles.axes4);
193 title(handles.axes4, '');
194
195
196 % Cek apakah data sudah ada
197 if ~isfield(handles, 'data1') || ~isfield(handles, 'data2')
198     warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');
```

C. Analisis Code

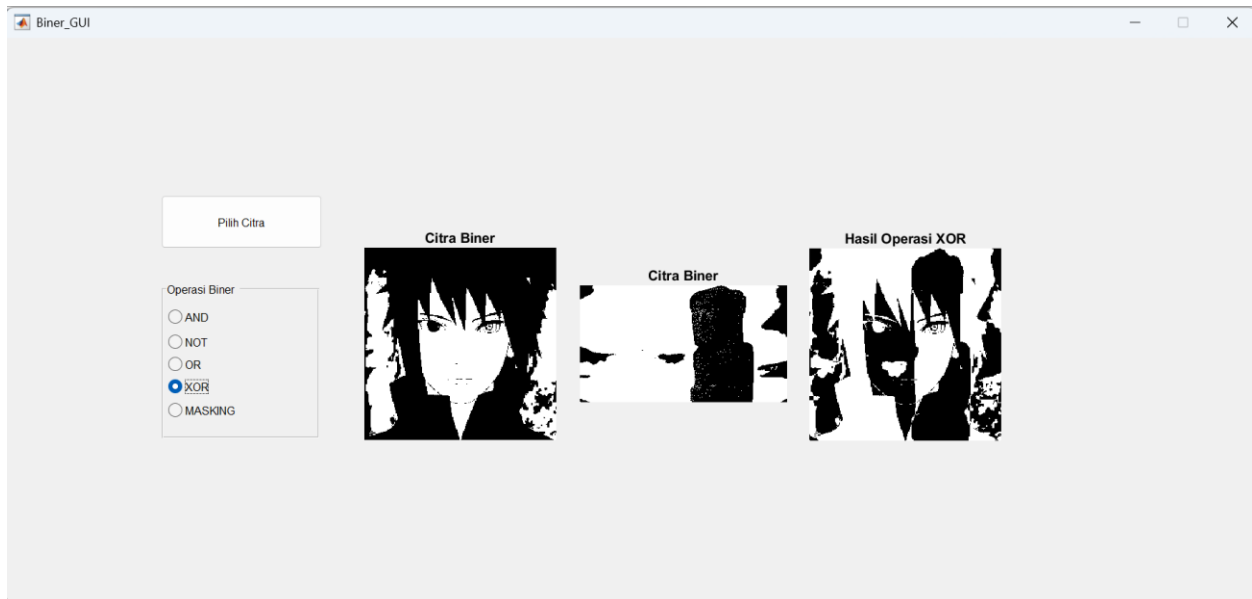
Baris	Kode	Deskripsi
185	<pre>function radiobutton3_Callback(hObject, eventdata, handles)</pre>	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 3 diklik (diperkirakan ini adalah tombol untuk operasi OR).

Baris	Kode	Deskripsi
186-189	<code>set(handles radiobutton1, 'value', 0);</code> (dan seterusnya)	Mengatur nilai (Value) dari radiobutton lain (1, 2, 4, 5) menjadi 0 (tidak terpilih), memastikan hanya operasi OR yang aktif.
191	<code>set(handles.axes4, 'Visible', 'off');</code>	Mengatur sumbu (axes4) menjadi tidak terlihat.
192	<code>cla(handles.axes4);</code>	Membersihkan sumbu (axes4).
193	<code>title(handles.axes4, "");</code>	Menghapus judul dari sumbu (axes4).
197	<code>if ~isfield(handles, 'data1') ~isfield(handles, 'data2')</code>	Memeriksa apakah kedua citra (data1 dan data2) sudah ada di struktur handles.
198-200	<code>warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');</code> <code>return; end</code>	Jika citra belum lengkap, tampilkan pesan peringatan dan keluar (return) dari fungsi.
202	<code>% Ambil citra dari handles</code>	Komentar: Bagian ini mengambil data citra.
203	<code>citra1 = handles.data1;</code>	Mengambil citra biner pertama.
204	<code>citra2 = handles.data2;</code>	Mengambil citra biner kedua.
206	<code>% --- Resize otomatis supaya ukuran sama ---</code>	Komentar: Memastikan dimensi kedua citra sama.
207	<code>ukuran1 = size(citra1);</code>	Mendapatkan dimensi dari citra1.

Baris	Kode	Deskripsi
208	<code>citra2_resized = imresize(citra2, ukuran1(1:2));</code>	Mengubah ukuran citra2 agar sesuai dengan dimensi citra1.
210	<code>\% --- Operasi OR ---</code>	Komentar: Bagian ini melakukan operasi logika inti.
211	<code>hasil = citra1 citra2_resized;</code>	Melakukan operasi Logika OR (element-wise) antara citra1 dan citra2 yang sudah diubah ukurannya. Tanda adalah operator OR di MATLAB.
213	<code>\% Simpan hasil jika dibutuhkan</code>	Komentar: Menyimpan hasil operasi.
214	<code>handles.hasilOR = hasil;</code>	Menyimpan citra hasil OR ke <i>field</i> baru handles.hasilOR.
216	<code>\% Tampilkan hasil di axes4</code>	Komentar: Menampilkan hasil operasi (Baris 217 menggunakan axes3).
217	<code>axes(handles.axes3);</code>	Mengaktifkan axes3 sebagai <i>current axes</i> .
218	<code>imshow(hasil);</code>	Menampilkan citra hasil operasi OR.
219	<code>title('Hasil Operasi OR');</code>	Memberi judul pada <i>axis</i> yang menampilkan hasil.
220	<code>guidata(hObject, handles);</code>	Menyimpan struktur handles yang sudah diperbarui.

4.6 Operasi Biner XOR

A. Gambar Operasi Biner XOR



B. Gambar Code Operasi Biner XOR

```

236
237 citra1 = handles.data1; % Citra biner upload pertama
238 citra2 = handles.data2; % Citra biner upload kedua
239
240 ukuran1 = size(citra1);
241 citra2_resized = imresize(citra2, ukuran1(1:2));
242
243 hasil = xor(citra1, citra2_resized);
244
245 handles.hasilXOR = hasil;
246
247 axes(handles.axes3);
248 imshow(hasil);
249 title('Hasil Operasi XOR');
250 set(handles.axes4, 'Visible', 'off');
251 cla(handles.axes4);
252 title(handles.axes4, '');
253 guidata(hObject, handles);
254

```

```

223 % --- Executes on button press in radiobutton4.
224 function radiobutton4_Callback(hObject, eventdata, handles)
225 set(handles.radiobutton1,'value',0);
226 set(handles.radiobutton2,'value',0);
227 set(handles.radiobutton3,'value',0);
228 set(handles.radiobutton5,'value',0);
229
230
231 % Pastikan data sudah diupload
232 if ~isfield(handles, 'data1') || ~isfield(handles, 'data2')
233     warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');
```

C. Analisis Code

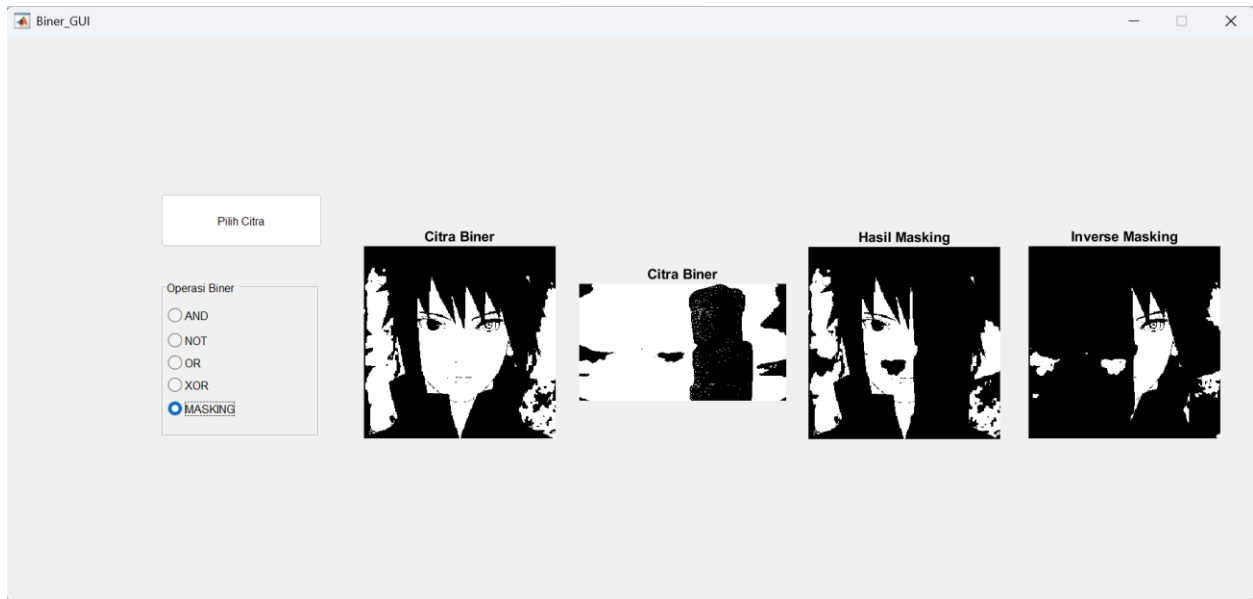
Baris	Kode	Deskripsi
224	function radiobutton4_Callback(hObject, eventdata, handles)	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 4 diklik (diperkirakan ini adalah tombol untuk operasi XOR).

Baris	Kode	Deskripsi
225-228	set(handles.radiobutton1, 'value', 0); (dan seterusnya)	Mengatur nilai (Value) dari radiobutton lain (1, 2, 3, 5) menjadi 0 (tidak terpilih) , memastikan hanya operasi XOR yang aktif.
231	\% Pastikan data sudah diupload	Komentar: Memeriksa ketersediaan data.
232	if ~isfield(handles, 'data1') ~isfield(handles, 'data2')	Memeriksa apakah kedua citra (data1 dan data2) sudah ada di struktur handles.
233-235	warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu. '); return; end	Jika citra belum lengkap, tampilkan pesan peringatan dan keluar (return) dari fungsi.
237	citra1 = handles.data1;	Mengambil citra biner pertama.
238	citra2 = handles.data2;	Mengambil citra biner kedua.
240	ukuran1 = size(citra1);	Mendapatkan dimensi dari citra1.
241	citra2_resized = imresize(citra2, ukuran1(1:2));	Mengubah ukuran citra2 agar memiliki dimensi yang sama dengan citra1.
243	hasil = xor(citra1, citra2_resized);	Melakukan Operasi Logika XOR (Exclusive OR) antara citra1 dan citra2 yang sudah diubah ukurannya. Fungsi xor pada MATLAB melakukan operasi ini secara <i>element-wise</i> .

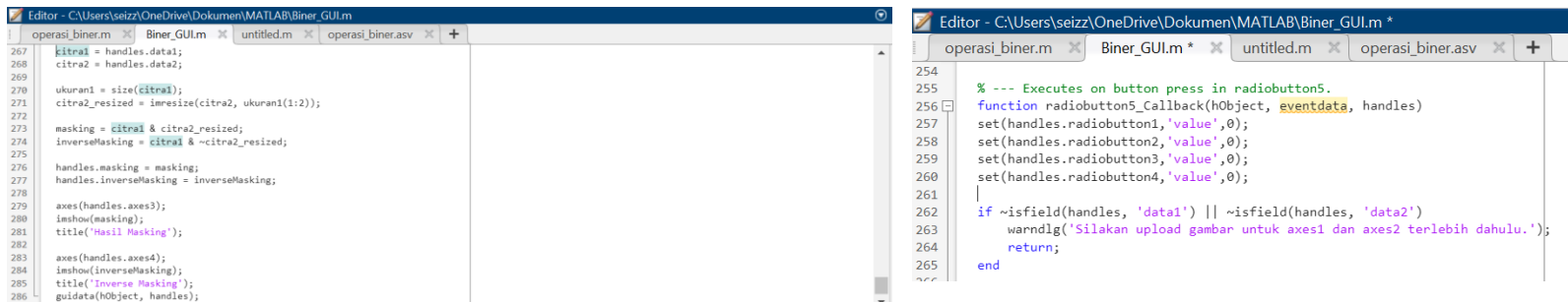
Baris	Kode	Deskripsi
245	<code>handles.hasilXOR = hasil;</code>	Menyimpan citra hasil XOR ke <i>field</i> baru <code>handles.hasilXOR</code> .
247	<code>axes(handles.axes3);</code>	Mengaktifkan axes3 sebagai <i>current axes</i> .
248	<code>imshow(hasil);</code>	Menampilkan citra hasil operasi XOR.
249	<code>title('Hasil Operasi XOR');</code>	Memberi judul pada <code>axes3</code> .
250	<code>set(handles.axes4, 'Visible', 'off');</code>	Mengatur sumbu (<code>axes4</code>) menjadi tidak terlihat .
251	<code>cla(handles.axes4);</code>	Membersihkan sumbu (<code>axes4</code>).
252	<code>title(handles.axes4, "");</code>	Menghapus judul dari sumbu (<code>axes4</code>).
253	<code>guidata(hObject, handles);</code>	Menyimpan struktur <code>handles</code> yang sudah diperbarui.

4.7 Operasi Biner Masking

A. Gambar Operasi Biner Masking



B. Gambar Code Operasi Biner MASKING



C. Analisis Code

Baris	Kode	Deskripsi
256	<pre>function radiobutton5_Callback(hObject, eventdata, handles)</pre>	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 5 diklik (diperkirakan ini adalah tombol untuk operasi Masking).

Baris	Kode	Deskripsi
257-260	<code>set(handles.radiobutton1, 'value', 0);</code> (dan seterusnya)	Mengatur nilai (Value) dari radiobutton lain (1, 2, 3, 4) menjadi 0 (tidak terpilih).
262	<code>if ~isfield(handles, 'data1') ~isfield(handles, 'data2')</code>	Memeriksa apakah kedua citra (data1 dan data2) sudah ada di struktur handles.
263-265	<code>warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');</code> <code>return; end</code>	Jika citra belum lengkap, tampilkan pesan peringatan dan keluar (return) dari fungsi.
267	<code>citra1 = handles.data1;</code>	Mengambil citra biner pertama (diasumsikan sebagai Citra Utama/Data).
268	<code>citra2 = handles.data2;</code>	Mengambil citra biner kedua (diasumsikan sebagai Masker).
270	<code>ukuran1 = size(citra1);</code>	Mendapatkan dimensi dari citra1.
271	<code>citra2_resized = imresize(citra2, ukuran1(1:2));</code>	Mengubah ukuran citra2 agar memiliki dimensi yang sama dengan citra1.
273	<code>masking = citra1 & citra2_resized;</code>	Melakukan operasi Masking. Ini adalah Logika AND antara Citra Utama (citra1) dan Masker (citra2_resized).
274	<code>inverseMasking = citra1 & ~citra2_resized;</code>	Melakukan operasi Inverse Masking. Ini adalah Logika AND antara Citra Utama (citra1) dan

Baris	Kode	Deskripsi
		Komplemen/NOT dari Masker (~citra2_resized).
276	<code>handles.masking = masking;</code>	Menyimpan hasil operasi Masking.
277	<code>handles.inverseMasking = inverseMasking;</code>	Menyimpan hasil operasi Inverse Masking.
279	<code>axes(handles.axes3);</code>	Mengaktifkan axes3 untuk menampilkan hasil Masking.
280	<code>imshow(masking);</code>	Menampilkan citra hasil Masking.
281	<code>title('Hasil Masking');</code>	Memberi judul pada axes3.
283	<code>axes(handles.axes4);</code>	Mengaktifkan axes4 untuk menampilkan hasil Inverse Masking.
284	<code>imshow(inverseMasking);</code>	Menampilkan citra hasil Inverse Masking.
285	<code>title('Inverse Masking');</code>	Memberi judul pada axes4.
286	<code>guidata(hObject, handles);</code>	Menyimpan struktur handles yang sudah diperbarui.

DAFTAR PUSTAKA

- Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed.). Pearson.
- Jain, A. K. (1989). *Fundamentals of digital image processing*. Prentice-Hall.
- Pratt, W. K. (2007). *Digital image processing: PIKS scientific inside* (4th ed.). Wiley-Interscience.
- Russ, J. C. (2015). *The image processing handbook* (7th ed.). CRC Press.
- Szeliski, R. (2022). *Computer vision: Algorithms and applications* (2nd ed.). Springer.
- Umbaugh, S. E. (2010). *Digital image processing and analysis: Human and computer vision applications with CVIPtools* (3rd ed.). CRC Press.
- Sonka, M., Hlavac, V., & Boyle, R. (2014). *Image processing, analysis, and machine vision* (4th ed.). Cengage Learning.