

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
IMPLEMENTASI.....	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

IMPLEMENTASI

3.1 Operasi Biner AND

A. Gambar Operasi Biner AND



B. Code Operasi Biner AND

```
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     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);
19     end
20 end
21
22 figure;
23 subplot(2,2,1);
24 imshow(Biner_A);
25 title('Citra Biner Logik 1');
26 subplot(2,2,2);
27 imshow(Biner_B);
28 title('Citra Biner Logik 2');
29 subplot(2,2,3);
30 imshow(C);
31 title('Hasil Operasi AND');
```

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\operasi_biner.m
operasi_biner.m + 
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(2,2,1);
24 imshow(Biner_A);
25 title('Citra Biner Logik 1');
26 subplot(2,2,2);
27 imshow(Biner_B);
28 title('Citra Biner Logik 2');
29 subplot(2,2,3);
30 imshow(C);
31 title('Hasil Operasi AND');
```

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	if size(A,3) == 3 \n citra_A = rgb2gray(A); \n end	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	if size(B,3) == 3 \n citra_B = rgb2gray(B); \n end	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	Biner_A = imbinarize(A);	Mengubah Citra A (asli) menjadi Citra Biner (Biner_A).
12	Biner_B = imbinarize(B);	Mengubah Citra B (asli) menjadi Citra Biner (Biner_B).
13	[r1, c1] = size(A);	Mendapatkan dimensi (baris, kolom) dari Citra A asli (r1 = baris, c1 = kolom).
14	C = false(r1, c1);	Membuat matriks hasil C dengan ukuran yang sama, diinisialisasi dengan nilai logika false (0/Hitam).
16	for x = 1:r1	Memulai <i>loop</i> baris (dari 1 hingga r1).
17	for y = 1:c1	Memulai <i>loop</i> kolom di dalam <i>loop</i> baris (dari 1 hingga c1).
18	C(x,y) = Biner_A(x,y) & Biner_B(x,y);	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

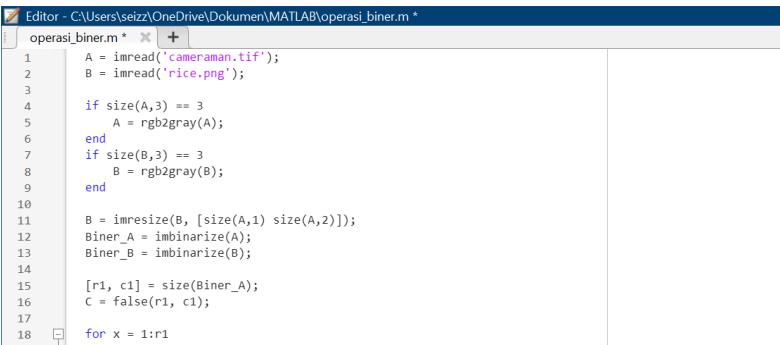
Baris	Kode	Deskripsi
		\$Biner_A\$ dan \$Biner_B\$. Tanda & adalah operator AND.
22	figure;	Membuat jendela <i>figure</i> baru untuk menampilkan hasil.
23	subplot(2,3,1);	Mengaktifkan sub-plot posisi 1.
24	imshow(Biner_A);	Menampilkan Biner_A.
25	title('Citra Biner Logik 1');	Memberi judul.
26	subplot(2,3,2);	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
    for y = 1:c1
        C(x,y) = Biner_A(x,y) | Biner_B(x,y); % OR
    end
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

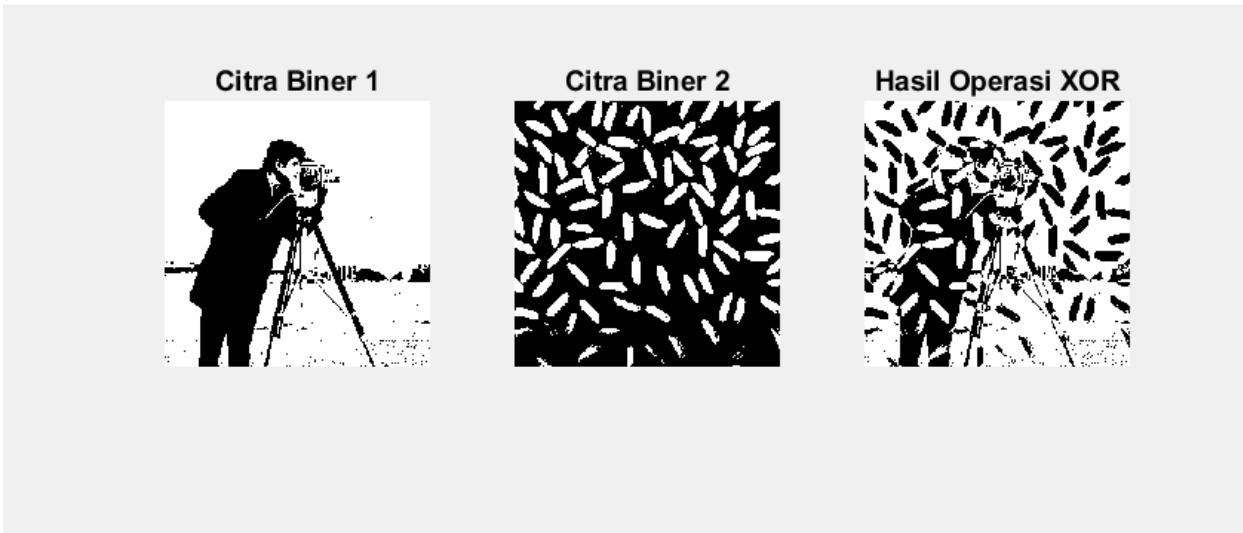
Baris	Kode	Deskripsi
1	A = imread('cameraman.tif');	Membaca citra pertama (Citra A) dari file cameraman.tif.
2	B = imread('rice.png');	Membaca citra kedua (Citra B) dari file rice.png.
4	if size(A,3) == 3	Memeriksa apakah Citra A adalah citra berwarna (memiliki 3 dimensi: R, G, B).
5	A = rgb2gray(A);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

Baris	Kode	Deskripsi
7	if size(B,3) == 3	Memeriksa apakah Citra B adalah citra berwarna.
8	B = rgb2gray(B);	Jika berwarna, Citra B diubah menjadi citra grayscale.
11	B = imresize(B, [size(A,1) size(A,2)]);	Mengubah ukuran (resize) Citra B agar dimensinya sama persis dengan Citra A (size(A,1) adalah baris, size(A,2) adalah kolom).
12	Biner_A = imbinarize(A);	Mengubah Citra A (grayscale) menjadi Citra Biner (Biner_A).
13	Biner_B = imbinarize(B);	Mengubah Citra B (grayscale, setelah di-resize) menjadi Citra Biner (Biner_B).
15	[r1, c1] = size(Biner_A);	Mendapatkan dimensi Citra Biner A (r1 = baris, c1 = kolom).
16	C = false(r1, c1);	Membuat matriks hasil C dengan ukuran yang sama, diinisialisasi dengan nilai logika false (0).
17	for x = 1:r1	Memulai <i>loop</i> baris (dari 1 hingga r1).
18	for y = 1:c1	Memulai <i>loop</i> kolom di dalam <i>loop</i> baris (dari 1 hingga c1).
19	C(x,y) = Biner_A(x,y) Biner_B(x,y); \% OR	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\seiz\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     for y = 1:c1
20         C(x,y) = xor(Biner_A(x,y), Biner_B(x,y)); % XOR
21     end
22
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 XOR');
28
```

C. Analisis Code

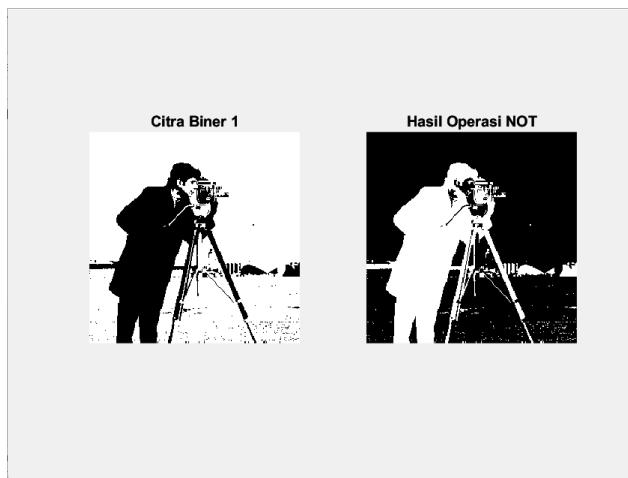
Baris	Perintah	Deskripsi/Fungsi
1-2	A = imread('cameraman.tif'); B = imread('rice.png');	Memuat Citra: Membaca dua citra (cameraman.tif dan rice.png) ke dalam variabel A dan B.

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\seizz\OneDrive\Dokumen\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 subplot(1,2,1); imshow(Biner_A); title('Citra Biner 1');
20 subplot(1,2,2); imshow(C); title('Hasil Operasi NOT');
```

C. Analisis Code

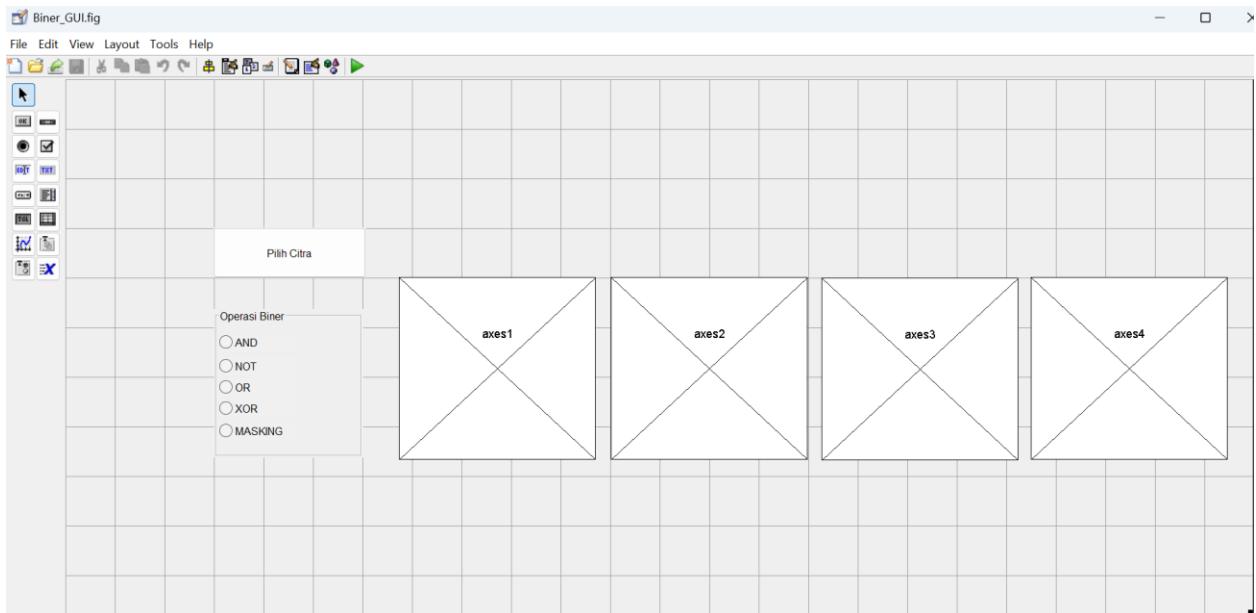
Baris	Perintah	Deskripsi/Fungsi
1	A = imread('cameraman.tif');	Memuat Citra: Membaca citra (cameraman.tif) ke dalam variabel A.
3-5	if size(A,3) == 3 ... A = rgb2gray(A); end	Konversi ke Grayscale: Mengecek apakah citra A adalah citra RGB (3 dimensi). Jika ya, citra dikonversi menjadi Grayscale (keabuan)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

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

BAB IV

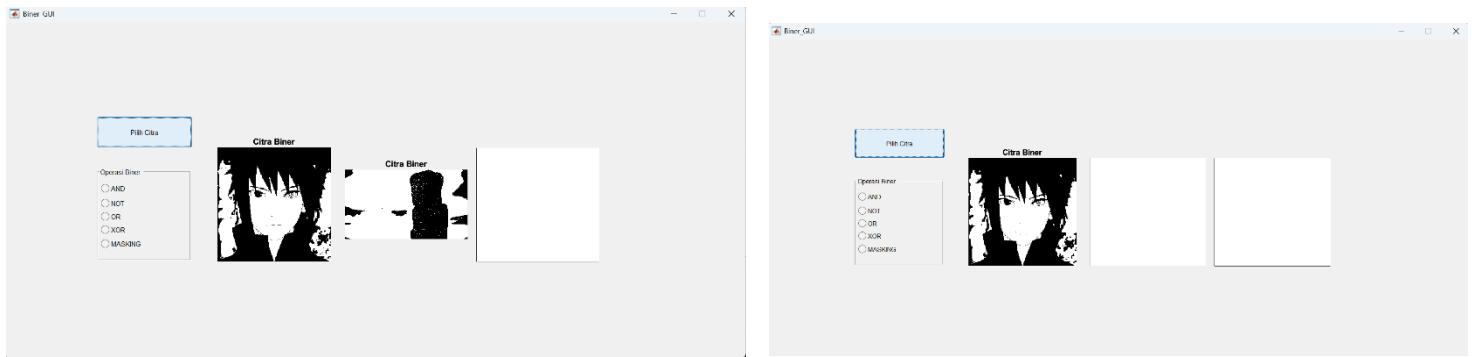
APLIKASI OPERASI BINER

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,[])
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
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
107 guidata(hObject,handles);
108 set(handles.axes4, 'Visible', 'off');
109 else
110 return
111 end

```

C. Analisis Code

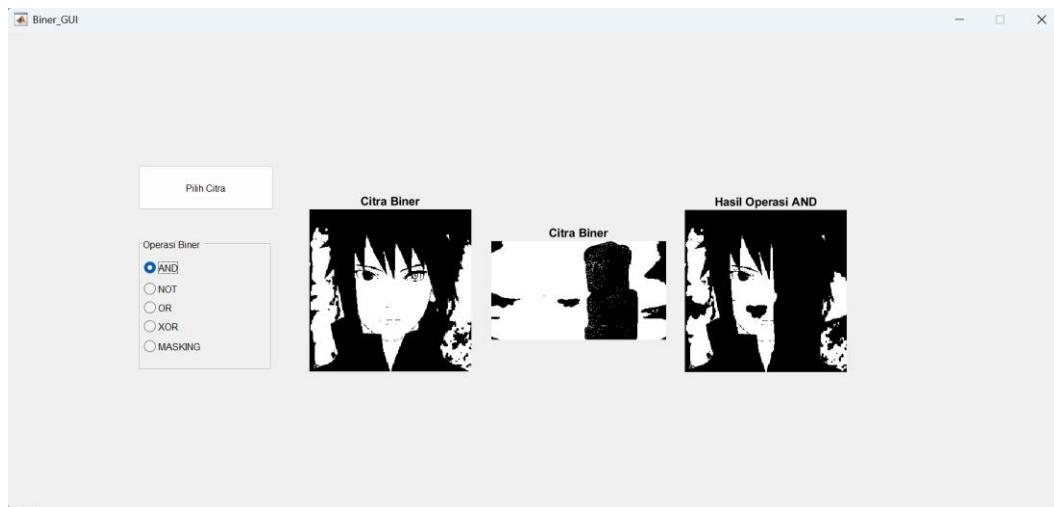
Baris	Kode	Deskripsi
77	function pushbutton1_Callback(hObject, eventdata, handles)	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat tombol (pushbutton) diklik.
78	guidata(hObject, handles);	Menyimpan struktur handles yang diperbarui.
79	movegui(handles.figure1, 'center');	Memindahkan jendela GUI (figure1) ke posisi tengah layar.
81-83	if ~isfield(handles, 'uploadCount') handles.uploadCount = 0; 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 path disimpan di variabel.

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

Baris	Kode	Deskripsi
		handles.axes2, dan beri judul 'Citra Biner'.
107	guidata(hObject, handles);	Menyimpan struktur handles yang sudah diperbarui (termasuk uploadCount, data1, data2).
108	set(handles.axes4, 'Visible', 'off');	Mengatur sumbu (axes4) menjadi tidak terlihat.
109- 111	else \n return \n end	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

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m Biner_GUI.m untitled.m operasi_biner.asv +
114 function radiobutton1_Callback(hObject, eventdata, handles)
115 set(handles radiobutton2, 'Value', 0);
116 set(handles radiobutton3, 'Value', 0);
117 set(handles radiobutton4, 'Value', 0);
118 set(handles radiobutton5, 'Value', 0);
119
120 set(handles axes4, 'Visible', 'off');
121 cla(handles axes4);
122 title(handles axes4, '');
123
124 % Cek apakah data sudah ada
125 if ~isfield(handles, 'data1') || ~isfield(handles, 'data2')
126     warning('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');
127     return;
128 end
129
130 % Ambil citra dari handles
131 citra1 = handles.data1; % Citra biner upload 1
132 citra2 = handles.data2; % Citra biner upload 2
133
134 % --- Resize otomatis agar ukuran sama ---
135 ukuran1 = size(citra1);
136 citra2_resized = imresize(citra2, ukuran1(1:2)); % resize citra2 mengikuti citra1
137
138 % Operasi AND
139 hasil = citra1 & citra2_resized;
140
141 % Simpan hasil
142 handles.hasilAND = hasil;
143
144 % Tampilkan hasil di axes4
145 axes(handles.axes4);
146 imshow(hasil);
147 title('Hasil Operasi AND');
148 guidata(hObject, handles);
149
```

C. Analisis Code

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

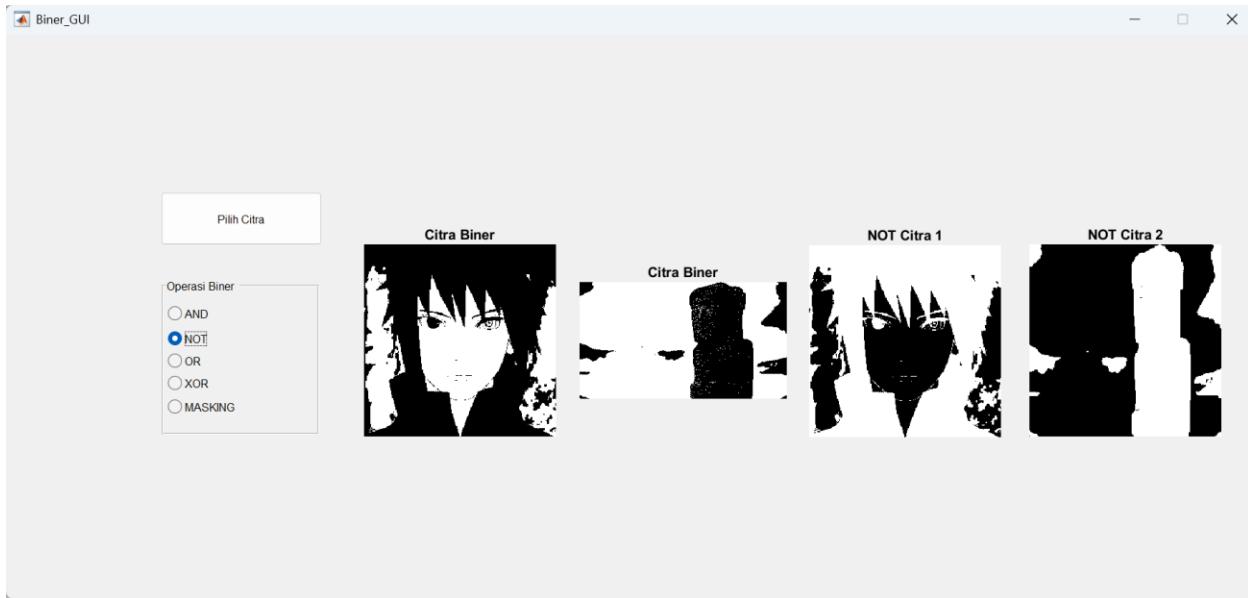
Baris	Kode	Deskripsi
		0 (tidak terpilih). Ini memastikan hanya satu operasi yang aktif.
120	set(handles.axes4, 'Visible', 'off');	Mengatur sumbu (axes4) menjadi tidak terlihat.
121	cla(handles.axes4);	Membersihkan sumbu (axes4).
122	title(handles.axes4, "");	Menghapus judul dari sumbu (axes4).
125	if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

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

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

4.4 Operasi Biner NOT

A. Gambar Operasi Biner NOT



B. Gambar Code Operasi Biner NOT

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m * Biner_GUI.m * untitled.m * operasi_biner.asv * + 
165
166 citra1 = handles.data1;
167 citra2 = handles.data2;
168 ukuran1 = size(citra1);
169 citra2_resized = imresize(citra2, ukuran1(1:2));
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
```

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m *
operasi_biner.m * Biner_GUI.m * untitled.m * operasi_biner.asv * + 
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 warnlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');
163 return;
164 end
```

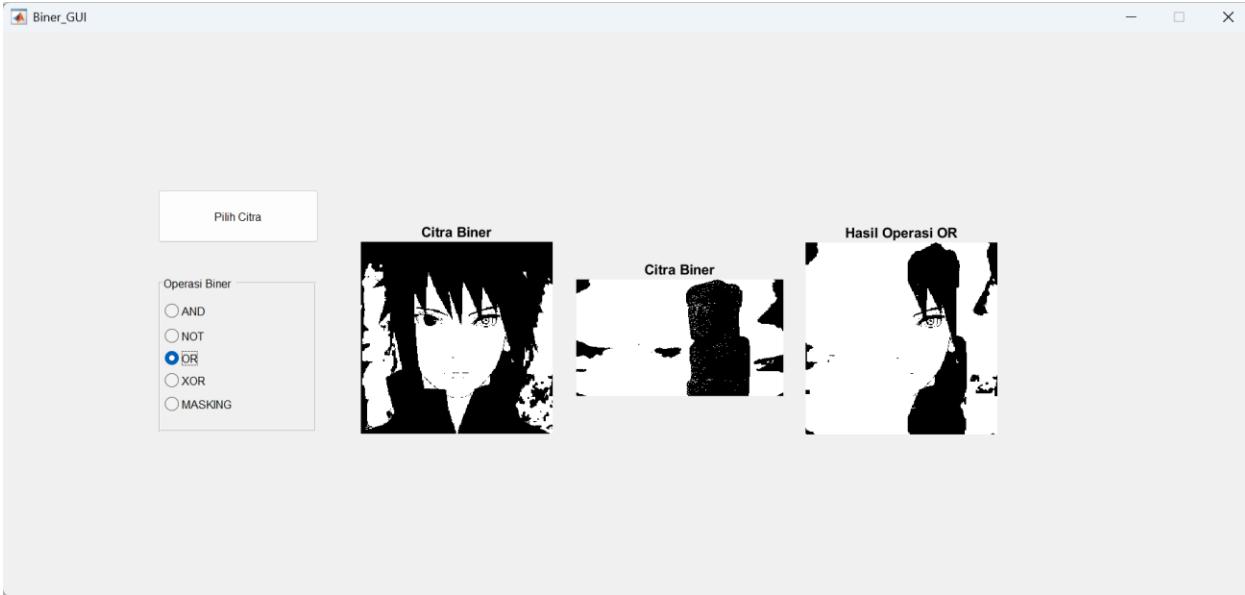
C. Analisis Code

Baris	Kode	Deskripsi
155	function radiobutton2_Callback(hObject, eventdata, handles)	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 2 diklik (diperkirakan ini adalah tombol untuk operasi NOT).
156- 159	set(handles radiobutton1, 'Value', 0); (dan seterusnya)	Mengatur nilai (Value) dari radiobutton lain (1, 3, 4, 5) menjadi 0 (tidak terpilih).
161	if ~isfield(handles, 'data1') ~isfield(handles, 'data2')	Memeriksa apakah kedua citra (data1 dan data2) sudah ada di struktur handles.
162- 164	warndlg('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');// return; end	Jika salah satu atau kedua citra belum ada, tampilkan pesan peringatan dan keluar (return) dari fungsi.
166	citra1 = handles.data1;	Mengambil citra biner pertama yang diunggah.
167	citra2 = handles.data2;	Mengambil citra biner kedua yang diunggah.
168	ukuran1 = size(citra1);	Mendapatkan dimensi dari citra1.
169	citra2_resized = imresize(citra2, ukuran1(1:2));	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\Biner_GUI.m
operasi_biner.m Biner_GUI.m untitled.m operasi_biner.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 ukuran2 = size(citra2);
209 citra2_resized = imresize(citra2, ukuran1(1:2));
210
211 % --- Operasi OR ---
212 hasil = citra1 | citra2_resized;
213
214 % Simpan hasil jika dibutuhkan
215 handles.hasilOR = hasil;
216
217 % Tampilkan hasil di axes4
218 axes(handles.axes4);
219 imshow(hasil);
220 title('Hasil Operasi OR');
221 guidata(hObject, handles);
222

Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m Biner_GUI.m untitled.m operasi_biner.asv + 
183 % --- Executes on button press in radiobutton3.
184 function radiobutton3_Callback(hObject, eventdata, handles)
185 set(handles.radioButton1,'value',0);
186 set(handles.radioButton2,'value',0);
187 set(handles.radioButton4,'value',0);
188 set(handles.radioButton5,'value',0);
189
190 set(handles.axes4, 'Visible', 'off');
191 cla(handles.axes4);
192 title(handles.axes4, '');
193
194
195
196 % Cek apakah data sudah ada
197 if ~isfield(handles, 'data1') || ~isfield(handles, 'data2')
198 warndig('Silakan upload gambar untuk axes1 dan axes2 terlebih dahulu.');
199 return;
200 end
```

C. Analisis Code

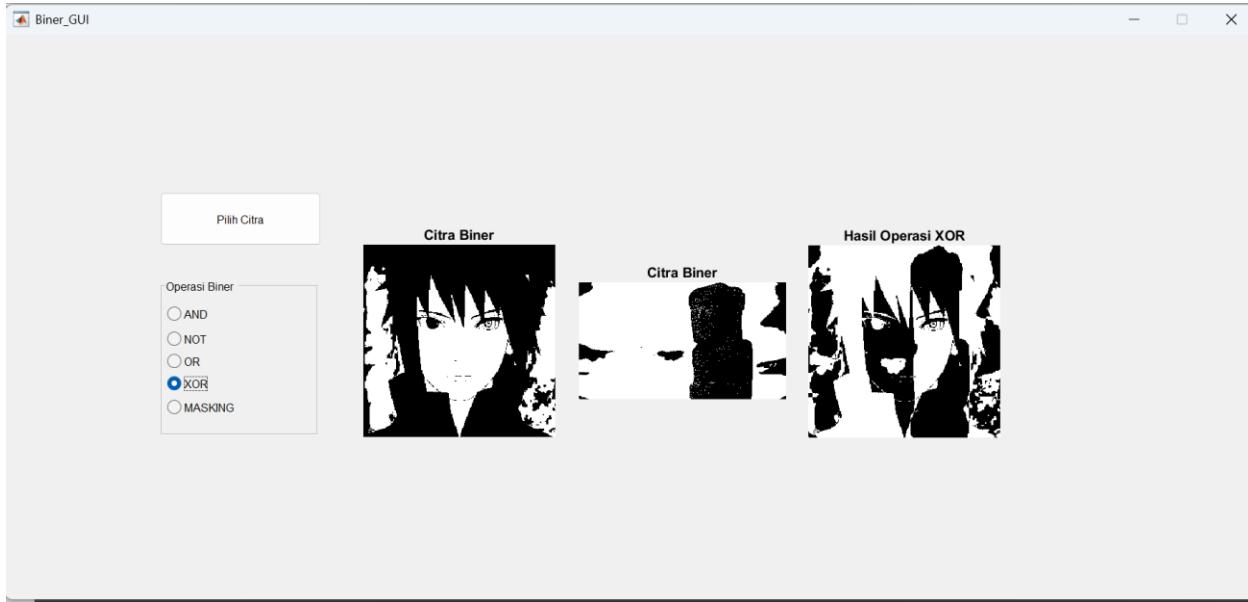
Baris	Kode	Deskripsi
185	function radiobutton3_Callback(hObject, eventdata, handles)	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 3 diklik (diperkirakan ini adalah tombol untuk operasi OR).

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

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

4.6 Operasi Biner XOR

A. Gambar Operasi Biner XOR



B. Gambar Code Operasi Biner XOR

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m  Biner_GUI.m  untitled.m  operasi_biner.asv +
```

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

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m  Biner_GUI.m  untitled.m  operasi_biner.asv +
```

```
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.');
234 return;
235 end
```

C. Analisis Code

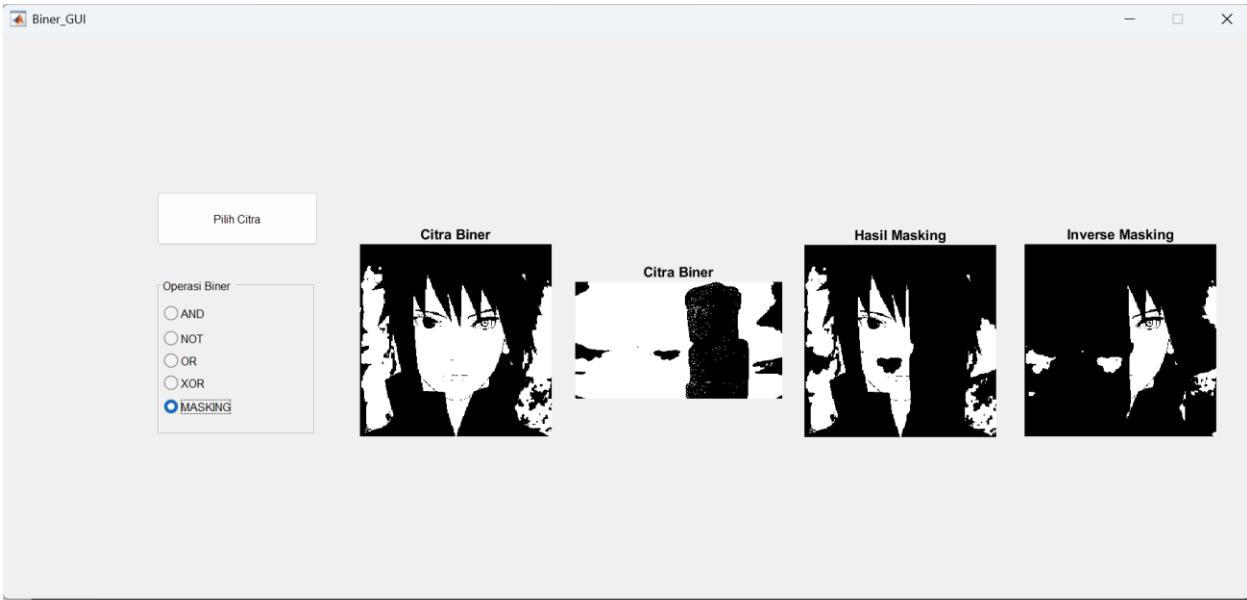
Baris	Kode	Deskripsi
224	function radiobutton4_Callback(hObject, eventdata, handles)	Mendefinisikan fungsi <i>callback</i> yang dijalankan saat Radiobutton 4 diklik (diperkirakan ini adalah tombol untuk operasi XOR).

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

4.7 Operasi Biner Masking

A. Gambar Operasi Biner Masking



B. Gambar Code Operasi Biner MASKING

The image displays two MATLAB code editors side-by-side. The left editor shows the file 'operasi_biner.m' with code for image processing and masking operations. The right editor shows the file 'Biner_GUI.m' with code for defining a callback function for a radio button.

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m
operasi_biner.m * Biner_GUI.m * untitled.m * operasi_biner.asv * +
267 % --- Executes on button press in radiobutton5.
268 function radiobutton5_Callback(hObject, eventdata, handles)
269 %
270 % hObject - handle to radiobutton5 (see GCBO)
271 % eventdata - event data passed from the callback
272 % handles - structure with handles and user data (see GUIDATA)
273 %
274 % Get the current value of the radiobutton
275 % value = get(hObject, 'Value');
276 %
277 % Set the value of the other radiobuttons to 0
278 % set(handles.radiobutton1, 'Value', 0);
279 % set(handles.radiobutton2, 'Value', 0);
280 % set(handles.radiobutton3, 'Value', 0);
281 % set(handles.radiobutton4, 'Value', 0);
282 %
283 % Set the value of the radiobutton5 to 1
284 % set(hObject, 'Value', 1);
285 %
286 %
287 % Call the function to process the image
288 % operasi_biner(handles);
289 %
290 % Close the figure
291 % close(gcf);
```

```
Editor - C:\Users\seizz\OneDrive\Dokumen\MATLAB\Biner_GUI.m *
operasi_biner.m * Biner_GUI.m * * * * *
254 % --- Executes on button press in radiobutton5.
255 function radiobutton5_Callback(hObject, eventdata, handles)
256 %
257 % hObject - handle to radiobutton5 (see GCBO)
258 % eventdata - event data passed from the callback
259 % handles - structure with handles and user data (see GUIDATA)
260 %
261 % Get the current value of the radiobutton
262 % value = get(hObject, 'Value');
263 %
264 % Set the value of the other radiobuttons to 0
265 % set(handles.radiobutton1, 'Value', 0);
266 % set(handles.radiobutton2, 'Value', 0);
267 % set(handles.radiobutton3, 'Value', 0);
268 % set(handles.radiobutton4, 'Value', 0);
269 %
270 % Set the value of the radiobutton5 to 1
271 % set(hObject, 'Value', 1);
272 %
273 % Call the function to process the image
274 % operasi_biner(handles);
275 %
276 % Close the figure
277 % close(gcf);
```

C. Analisis Code

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

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

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