

## **Visualisasi Operasi Aritmatika Pada Pengolahan Citra Digital**

Indo Intan, ST. Aminah Dinayati Ghani, Nur Salman

STMIK Dipanegara Makassar

Jalan Perintis Kemerdekaan KM. 9 Makassar, Telp. (0411)587194 – Fax. (0411) 588284

e-mail: [indo.intan@dipanegara.ac.id](mailto:indo.intan@dipanegara.ac.id), [dinayati.amy@dipanegara.ac.id](mailto:dinayati.amy@dipanegara.ac.id), [nursalman@dipanegara.ac.id](mailto:nursalman@dipanegara.ac.id)

---

### **Abstrak**

*Pengolahan Citra merupakan mata kuliah yang banyak dikembangkan untuk tugas akhir sehingga memerlukan eksplorasi terhadap metode dasarnya bagi dosen dan mahasiswa. Studi ini bertujuan untuk membuat aplikasi sederhana yang bisa memvisualisasikan operasi aritmatika pada pengolahan citra. Metode yang digunakan yaitu operasi aritmatika dan logika untuk menganalisis citra. Metode pengujian aplikasi menggunakan black box testing dengan menguji fungsionalitas semua tombol yang digunakan. Hasil yang diperoleh bahwa aplikasi sudah bisa menjalankan fungsinya sesuai yang diharapkan meskipun masih butuh pengembangan lanjut. Bagi mahasiswa, mereka sudah memiliki kemampuan coding dalam pengolahan citra.*

**Kata Kunci : aplikasi, visualisasi, operasi aritmatika, pengolahan citra.**

### **Abstract**

*Image processing is the most developed subject to last project so need exploration of basic method for lecturers and students. This study aims to make a simple application can visualitate arithmetic operation on image processing. Used method are arithmetics operation and logic to analyze image. The Application testing used black box testing to test function of used buttons. Provided Output that application can to do functionalization like as been expected although need advanced developed. For students, they get coding competention in image processing.*

**Keywords : application, visualization, arithmetics operation, image processing.**

## **1. PENDAHULUAN**

Mata Kuliah Pengolahan Citra merupakan salah satu mata kuliah program studi Teknik Informatika yang disajikan pada semester kelima. Mata Kuliah ini juga disajikan untuk konsentrasi peminatan Sistem Cerdas dan Multimedia. Pesertanya mencapai sekitar 150 orang atau empat kelas.

Tujuan mata kuliah ini agar mahasiswa bisa memahami proses pengolahan citra, teknik yang digunakan, serta implementasinya pada bidang lainnya. Mata kuliah tersebut sangat dinamis memiliki keterkaitan dengan mata kuliah lainnya seiring dengan perkembangan teknologi dewasa ini. Di antara mata kuliah tersebut adalah pengenalan pola, keamanan komputer dan jaringan, mikrokontroler, jaringan syaraf tiruan, multimedia dan komputer grafis, serta data mining.

Konten mata kuliah pengolahan citra memiliki peluang dikembangkan sebagai tugas akhir karena di dalamnya terdapat banyak metode pengekstraksi fitur. Metode tersebut berkorelasi dengan tujuan pengolahan citra yaitu untuk memperbaiki kualitas citra sehingga bisa digunakan sebagai output maupun sebagai input dari obyek sesuai citra kebutuhan. Penerapan pengolahan citra digital antara lain dalam bidang biomedis misalnya citra sel darah, citra penyakit; biometric misalnya wajah, sidik jari; fotografi, desain visual, image retrieval, dan sebagainya. Mengingat urgensi mata kuliah Pengolahan Citra, maka pemahaman mahasiswa terhadap mata kuliah tersebut memerlukan variasi teknik yang menarik melalui visualisasi. Pada mata kuliah tersebut, setelah diperkenalkan konsep mendasarnya, selanjutnya dielaborasi melalui penggunaan operasi aritmatika dan operasi geometri dilanjutkan dengan teknik lainnya.

Beberapa permasalahan dalam penyajian mata kuliah ini, antara lain: 1) mata kuliah ini tidak didukung oleh praktikum secara tersendiri sehingga tidak terdapat sesi praktek di laboratorium; 2)

beberapa materi acuan masih menggunakan teori matematika yang tentu saja agak sedikit menjemukan peserta mata kuliah. Mereka sering gagal fokus saat penjelasan pada bagian tersebut. Dosen berusaha memberikan contoh-contoh numerik melalui proses perhitungannya sehingga secara motorik mengurangi kejenuhan. Masalahnya adalah meski mereka lancar dalam perhitungan secara numerik, akan tetapi mahasiswa belum bisa menginterpretasikan angka tersebut ke dalam bentuk citra. Sehingga ketika secara visual diperlihatkan gambar atau citra kemudian meminta mereka mengaitkannya dengan citra input maupun output, sebagian besarnya bahkan seluruhnya tidak mampu memberikan penjelasan sesuai konsep materi secara ilmiah.

Tujuan penelitian ini yakni untuk membuat aplikasi sederhana menggunakan operasi aritmatika pada pengolahan citra digital sehingga memudahkan analisis secara visual. Tujuan akhir yang diharapkan agar mahasiswa mampu mengelaborasi kemampuan *coding* mereka dari mata kuliah sebelumnya dengan berbagai operasi dan teknik pengolahan citra.

Penulis mengambil operasi aritmatika sebagai operasi dasar untuk memvisualisasikannya, karena operasi ini akan menstimulus dan membuka wawasan mahasiswa pada teknik pengembangan selanjutnya.

Visualisasi merupakan bagian penting agar memudahkan pencerapan dan pemahaman mahasiswa terhadap konten materi. Visualisasi yang dimaksudkan di sini adalah memberikan penyajian materi berupa tampilan berdasarkan citra (gambar atau foto) menggunakan proses citra melalui penggunaan tombol-tombol sesuai fungsi operasi aritmatika.

Penelitian terkait dalam bidang pengolahan citra tentu saja sangat bervariasi, baik menggunakan pengekstraksi fitur maupun pengklasifikasi antara lain yang dilakukan oleh penulis tentang obyek kanker tiroid menggunakan pengekstraksi fitur deteksi tepi dan pengklasifikasi ANFIS [1]; pengekstraksi fitur GMRF dan pengklasifikasi SOM Kohonen [2]; biometrik wajah, menggunakan penjumlahan pengekstraksi fitur PCA dan LDA serta pengklasifikasi Euclidean Distance [3]; biometrik multi modal, menggabungkan wajah dan telapak tangan menggunakan pengekstraksi fitur Gabor transformation dan Minutiae serta pengklasifikasi penjumlahan citra [4]; forensic, mengambil citra gigi dan DNA dengan pengekstraksi PCA, pengklasifikasi PSO dan ANN [5]; pertanian, untuk meneliti kualitas beras [6] dan identifikasi dengan inventarisasi rotasi tanam melalui pembacaan GIS menggunakan kombinasi pengekstraksi fitur PCA+NDVI+TCT [7]; perikanan dan industri, meneliti jenis ikan berdasarkan ukuran tubuh menggunakan ekstraksi fitur deteksi tepi [8]; mendeteksi kendaraan bergerak menggunakan webcam [9] mendeteksi kelainan paru-paru menggunakan metode deteksi tepi Canny dan HPH [10], psikologi tentang ekspresi wajah menggunakan otot frontalis pada citra 3D menggunakan pengekstraksi fitur GLCM dan pengklasifikasi Canberra Distance [11].

Penelitian tersebut merupakan pengembangan implementatif dari penggunaan operasi aritmatika citra. Penelitian yang dilakukan penulis berpusat pada penggunaan operasi aritmatika, sebagai proses dasar pengolahan citra. Mengacu pada penelitian ini, maka penulis akan mengeksplorasi penelitian lanjut yang mengembangkan operasi dasar hingga taraf implementasi pengekstraksi fitur. Akhirnya, visualisasi pengolahan citra menjadi implementatif.

## 1.1 Pengolahan Citra

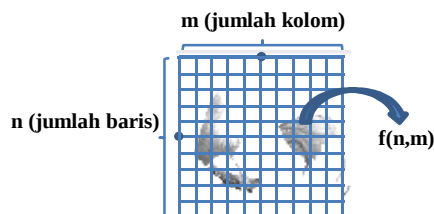
Pengolahan citra merupakan proses memanipulasi dan menganalisis citra dua dimensi dengan bantuan komputer sehingga menghasilkan citra dengan kualitas yang lebih baik [2][9]. Proses dasar pengolahan citra yaitu: a) *sampling*, proses digitasi pada koordinat  $x,y$ ; b) *kuantisasi*, adalah membagi skala keabuan menjadi sejumlah level keabuan hingga 8 bit [1], kuantisasi dilakukan dengan greyscale mengubah citra warna menjadi citra keabuan 256 level keabuan; Resolusi citra merupakan tingkat detail suatu citra [12]. Citra terdiri atas: 1) citra warna (24 bit) memiliki 3 kanal warna, *Red*, *Green*, dan *Blue* [12]; 2) citra greyscale adalah citra yang hanya memiliki satu nilai kanal yaitu 8 bit dengan 256 kombinasi keabuan [1].

## 1.2 Operasi Aritmatika Citra

Merupakan proses pengolahan citra dengan memanfaatkan operator aritmatika atau operator logika terhadap dua atau lebih citra input. Setiap piksel citra output tergantung pada piksel citra input atau vektornya [12]. Operasi aritmatika, antara lain: 1) *Penjumlahan*, proses menjumlahkan nilai piksel pada dua buah atau lebih citra; 2) *Pengurangan*, mengurangi nilai piksel pada dua buah citra atau lebih; 3) *Perkalian*, mengalikankan nilai piksel pada dua buah citra atau lebih; 4) *Pembagian*, membagi nilai piksel pada dua buah citra atau lebih; 5) *Logika AND* (tabel 1.1), proses mengalikan antara 2 buah citra atau lebih, jika semua inputnya 0 maka outputnya 0, jika semua inputnya 1 maka outputnya 1; 6)

*Logika OR* (tabel 1.1), proses menjumlahkan antara nilai piksel pada dua buah citra atau lebih, jika salah satu inputnya 1 maka outputnya 1, jika semua inputnya 0 maka outputnya 0; 7) *Logika XOR* (tabel 1.1), proses menjumlahkan antara nilai piksel pada dua buah citra atau lebih, jika jumlah masukannya ganjil, maka bernilai 1, jika genap maka berlogika 0; 8) *Logika NOT* (tabel 1.1), melakukan inversi logika terhadap inputnya, jika input 0 maka output 1, jika input 1 maka output 0 [12].

Sebuah citra ditulis dalam bentuk  $f(n, m)$  sebagai pemetaan citra dua dimensi, yang mana  $n$  sebagai jumlah baris, dan  $m$  sebagai jumlah kolom seperti ditunjukkan gambar 1.1.



Gambar 1.1 Pemetaan Piksel Citra [2]

Operasi aritmatika penjumlahan menggunakan persamaan 1.1, pengurangan (persamaan 1.2), perkalian (persamaan 1.3), pembagian (persamaan 1.4) berikut:

$$f(n, m) = f_1(n, m) + f_2(n, m) \quad (1.1)$$

$$f(n, m) = f_1(n, m) - f_2(n, m) \quad (1.2)$$

$$f(n, m) = f_1(n, m) * f_2(n, m) \quad (1.3)$$

$$f(n, m) = f_1(n, m) / f_2(n, m) \quad (1.4)$$

Tabel 1.1 Tabel Kebenaran Logika AND, OR, XOR, dan NOT.

| A | B | Y AND B |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 0       |
| 1 | 0 | 0       |
| 1 | 1 | 1       |

| A | B | Y OR B |
|---|---|--------|
| 0 | 0 | 0      |
| 0 | 1 | 1      |
| 1 | 0 | 1      |
| 1 | 1 | 1      |

| A | B | Y XOR B |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 1       |
| 1 | 0 | 1       |
| 1 | 1 | 0       |

| A | Y= NOT A |
|---|----------|
| 0 | 1        |
| 1 | 0        |

### 1.3 Netbeans dan MySQL

Netbeans adalah aplikasi *Integrated Development Environment* (IDE) yang berbasis Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi java untuk mengembangkan aplikasi desktop yang dapat berjalan pada berbagai macam platform. MySQL adalah sebuah implementasi dari system manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis [13].

## 2. METODE PENELITIAN

Penelitian ini menggunakan flowchart untuk menggambarkan aliran proses operasi aritmatika. Tahapan proses pengolahan citra yaitu: penginputan citra, sampling, greyscale, operasi aritmatika.

Metode penelitian yang dilakukan dalam penelitian ini yaitu menggunakan operasi aritmatika, antara lain operasi dan logika: penjumlahan, pengurangan, perkalian, pembagian, operasi logika AND, OR, NOT, dan XOR. Bahan penelitian berupa citra berformat JPEG yang merupakan input dari aplikasinya. Aplikasi menggunakan Java Netbeans sebagai bahasa pemrogramannya dengan menggunakan MySQL sebagai databasenya.

Analisis dan perbandingan yang dilakukan terdiri atas pengamatan terhadap konversi citra secara visual dan vektor citra yang sudah dikonversi menjadi bilangan numerik (bentuk matriks).

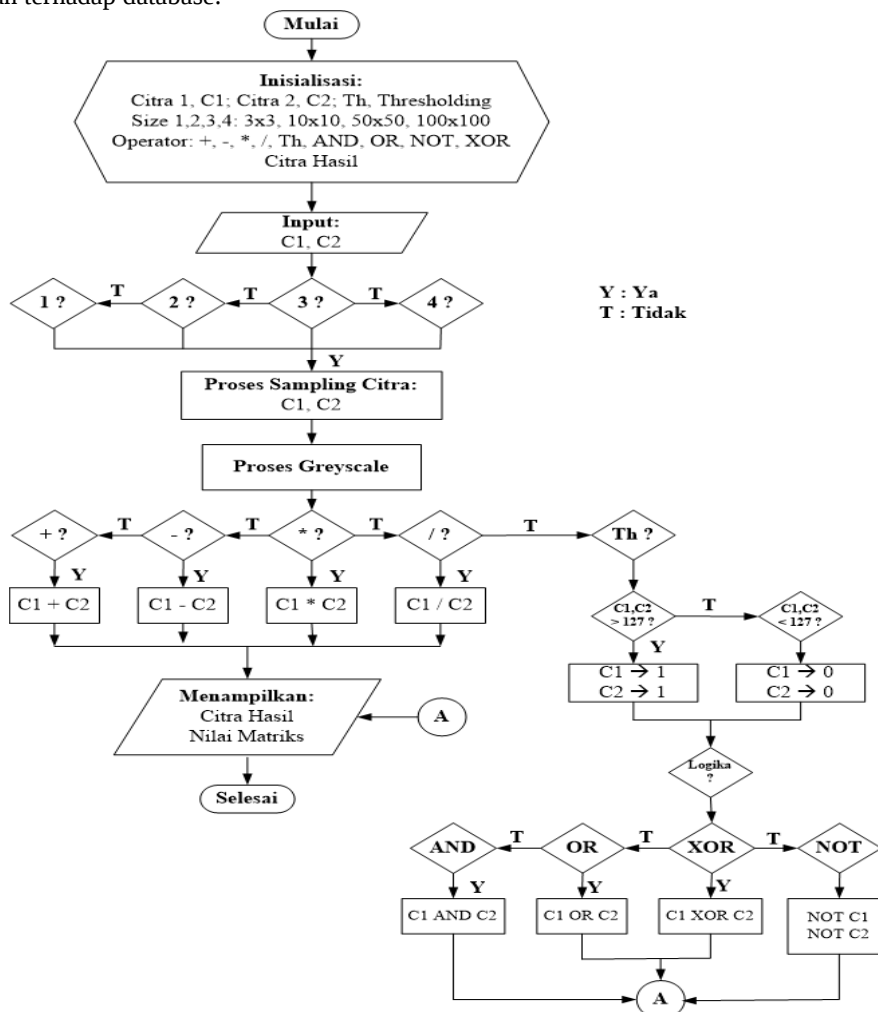
Berdasarkan gambar 2.1, citra input yang digunakan berformat JPG atau JPEG yang merupakan citra hasil kompresi. Langkah awal, melakukan proses inisialisasi terhadap atribut data atau variabel. Kemudian dilakukan proses pencarian citra pada storage sehingga file citra terbuka (input) pada aplikasinya. Citra input terdiri atas 2, yaitu Citra 1 ( $C_1$ ) dan Citra 2 20x20( $C_2$ ). Kedua citra ini merupakan

citra warna dalam bentuk true colours. Selanjutnya melakukan pemilihan sampling citra sesuai size (ukuran) 10x10, 20x20, 50x50, dan 100x100 piksel yang awalnya berukuran 2050x1500 piksel. Pada saat memilih size 4x4 piksel, maka citra warna diubah menjadi citra keabuan melalui proses greyscale.  $C_1$ ,  $C_2$  dan  $C$  Hasil akan ditampilkan dalam bentuk citra keabuan yang memiliki level keabuan 0 s.d. 255.

Citra tersebut kemudian diproses menggunakan teknik sampling. Sampling digunakan untuk mengambil piksel citra sesuai size (ukuran) 10x10, 20x20, 50x50, 100x100 piksel yang awalnya berukuran 2050x1500 piksel. Teknik ini digunakan untuk mengecilkan ukuran citra sehingga memudahkan analisis. Setelah citra melalui proses greyscale, selanjutnya melakukan operasi aritmatika melalui beberapa operasi pilihan. Operasi tersebut di antaranya: penjumlahan, pengurangan, perkalian, pembagian, dan thresholding (binerisasi). Operasi ini memungkinkan hanya satu tasking (penugasan) pada waktu yang sama, jika penjumlahan yang dikerjakan, maka teknik lainnya seperti pengurangan menunggu untuk dipilih. Demikian dilakukan secara bergantian sehingga semua proses yang disajikan bisa dilaksanakan sesuai kebutuhan.

Proses logika harus melalui proses thresholding untuk menjalankan operasi AND, OR, NOT, dan XOR. Proses thresholding merupakan kelanjutan dari proses greyscale, yaitu mengembangkan nilai greyscale pada kondisi 0 s.d 127 sebagai nilai 0, sedangkan nilai piksel > 128 bernilai 1 untuk kedua citra input.

Semua citra tersebut divisualisasikan dalam bentuk gambar dan nilai vektor berbentuk matriks. Jika salah satu atau semua proses yang ada pada pilihan dijalankan, maka akan ditampilkan. Vektor 0 akan berwarna hitam, sedangkan vektor 1 akan berwarna putih. Operasi logika hanya akan menampilkan warna hitam atau putih, tidak ada keabuan. Setiap nilai greyscale akan ditampilkan pada interface aplikasi, kemudian hasil aritmatika akan disimpan ke database dan diimpor ke Excel untuk kemudahan pengaksesan terhadap database.



Gambar 2.1. Diagram Alir Operasi Aritmatika

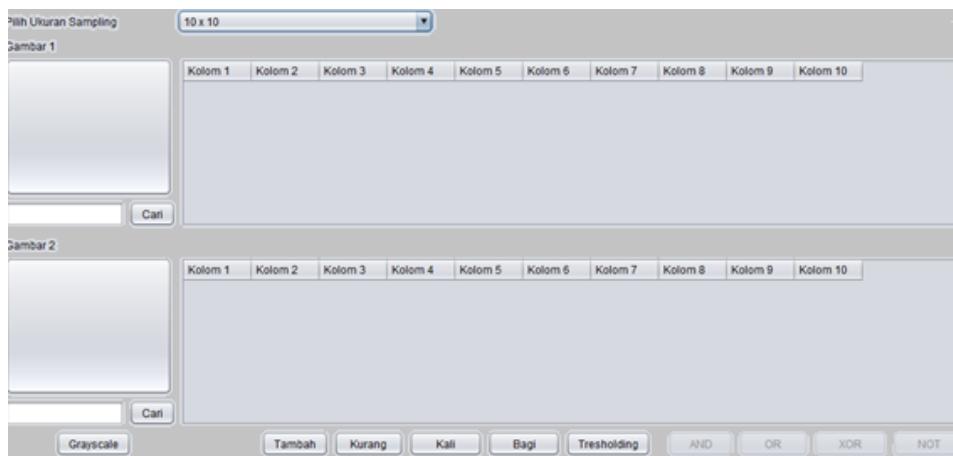
### 3. HASIL DAN PEMBAHASAN

Hasil dari pembuatan aplikasi ini dipaparkan dalam beberapa tahapan, yaitu:

#### 3.1. Hasil Perancangan Aplikasi

Hasil perancangan aplikasi meliputi tampilan awal aplikasi, tampilan coding aplikasi, dan pengujian perangkat lunak.

##### 3.1.1 Tampilan Awal Aplikasi



**Gambar 3.1 Interface Operasi Aritmatika**

Gambar 3.1 menunjukkan tampilan awal aplikasi yang terdiri atas beberapa item: yaitu ukuran sampling 10x10, 20x20, 50x50, dan 100x100. Bingkai citra terdiri atas citra input (citra 1 dan citra 2) dan citra hasil (hasil) untuk menempatkan citra yang dipilih dari *storage device* sekaligus menampilkan nama file-nya serta hasil operasi kedua citra tersebut pada Hasil. Nilai citra merupakan vektor citra sesuai ukuran sampling yang terpilih di antara keempat pilihan pada *combo box*. Nilai citra memiliki kolom x baris sama dengan sampling terpilih. Tombol Cari untuk melakukan proses pencarian citra, sampling sesuai ukuran terpilih dan menampilkan nilai citra (vektor) pada database berdasarkan citra input dan citra hasil. Citra Hasil akan muncul setelah salah satu tombol operasi diklik.

##### 3.1.2 Tampilan Coding Aplikasi

Bagian yang penting dalam pembuatan aplikasi adalah proses coding yang akan menentukan apakah logika aplikasi tersebut valid (benar) dan sesuai dengan fungsi yang diharapkan.

1. *Membuka Citra*, melakukan pencarian file pada device storage kemudian membukanya pada bingkai (frame) aplikasi. Berikut codingnya.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    NilaiDatabaseCitra1 = "";
    JFileChooser chooser = new JFileChooser();
    chooser.setSelectionMode(JFileChooser.FILES_ONLY);
    chooser.setFileFilter(new FileFilter() {
        @Override
        public boolean accept(File f) {
            if (f.isDirectory()) {
                return true;
            } else {
                return
                f.getName().toLowerCase().endsWith(".jpg") || f.getName().toLowerCase()
            }
        }
        @Override
        public String getDescription() {
            return "All Image Support";
        }
    });
    int res = chooser.showOpenDialog(ProsesInputDalamDatabase.this);
    if (res == JFileChooser.APPROVE_OPTION) {
```

2. *Sampling*, proses pemilihan size citra: 10x10, 20x20, 50x50, dan 100x100.

```
File file = chooser.getSelectedFile();
JResizeImage ukuran = new JResizeImage();
Image img = ukuran.rescale(ImageIO.read(file), Ukuran,Ukuran);
```

3. *Greyscale*, merupakan bagian dari kuantisasi untuk mengubah citra warna menjadi keabuan pada level 8 bit atau 0-255 level.

```
MatrixC2 = new Matrix(s.getWidth(),s.getHeight());
for (int i = 0; i < s.getWidth(); i++) {
    for (int j = 0; j < s.getHeight(); j++){
        int p = s.getRGB(i, j);
        int r = (p & 0xff0000)>>16;
        int g = (p & 0x00ff00)>>8;
        int b = (p & 0x0000ff);

        double gry = (r+g+b)/3;
        MatrixC2.set(i, j, gry);
        raster.setSample(i, j, 0, gry);
        NilaiDatabaseCitra2 += gry + ",";
    }
    NilaiDatabaseCitra2 += "\n";
}
```

4. *Penjumlahan*, menjumlahkan vektor citra input, jika vektornya > 255, maka diflipping (dipotong) tetap pada level keabuan 255 kemudian hasilnya disimpan dalam database.

```
BufferedImage g = new BufferedImage(MatrixC1.getColumnDimension(),
MatrixC2.getColumnDimension(), BufferedImage.TYPE_BYTE_GRAY);
WritableRaster raster = g.getRaster();
MatrixH = MatrixC1.plus(MatrixC2);
for (int i = 0; i < MatrixC1.getRowDimension(); i++){
    for (int j = 0; j < MatrixC1.getRowDimension(); j++){
        double Hasil = MatrixH.get(i, j);
        if (Hasil > 255){
            Hasil = 255;
        }
        raster.setSample(i, j, 0, Hasil);
        NilaiDatabaseCitraHasil += Hasil + ",";
    }
    NilaiDatabaseCitraHasil += "\n";
}
```

5. *Pengurangan*, mengurangi vektor citra input kemudian hasilnya disimpan ke dalam database.

```
BufferedImage g = new BufferedImage(MatrixC1.getColumnDimension(),
MatrixC2.getColumnDimension(), BufferedImage.TYPE_BYTE_GRAY);
WritableRaster raster = g.getRaster();

MatrixH = MatrixC1.minus(MatrixC2);
for (int i = 0; i < MatrixC1.getRowDimension(); i++){
    for (int j = 0; j < MatrixC1.getRowDimension(); j++){
        double Hasil = MatrixH.get(i, j);
        if (Hasil < 0){
            Hasil = 0;
        }
        raster.setSample(i, j, 0, Hasil);
        NilaiDatabaseCitraHasil += Hasil + ",";
    }
    NilaiDatabaseCitraHasil += "\n";
}
```

6. *Perkalian*, mengalikan vektor citra input kemudian hasilnya disimpan dalam database sebagaimana coding sebelumnya.

```
MatrixH = MatrixC1.arrayTimes(MatrixC2);
for (int i = 0; i < MatrixC1.getRowDimension(); i++){
    for (int j = 0; j < MatrixC1.getRowDimension(); j++){
        double Hasil = MatrixH.get(i, j);
        if (Hasil > 255){
            Hasil = 255;
        }
    }
}
```

7. *Pembagian*, mengalikan vektor citra input kemudian hasilnya disimpan dalam database sebagaimana coding sebelumnya.

```

MatrixH = MatrixC1.arrayRightDivide(MatrixC2);
for (int i = 0; i < MatrixC1.getRowDimension(); i++) {
    for (int j = 0; j < MatrixC1.getRowDimension(); j++) {
        int Hasil = (int) MatrixH.get(i, j);
        if (Hasil > 255) {
            Hasil = 255;
        }
    }
}

```

8. *Thresholding* (Binerisasi), mengkonversi vektor citra ke biner (0 dan 1), jika vektor citra < 127 maka berlogika "0", jika vektor > 127, maka berlogika "1".

```

String[] BarisC1 = ThresholdingCitra1.split(":");
String[] BarisC2 = ThresholdingCitra2.split(":");
int[][] DataC1 = new int[Ukuran][Ukuran];
int[][] DataC2 = new int[Ukuran][Ukuran];
for (int i = 0; i < BarisC1.length; i++) {
    String[] NilaiC1 = BarisC1[i].split(",");
    String[] NilaiC2 = BarisC2[i].split(",");
    for (int j = 0; j < NilaiC1.length; j++) {
        DataC1[i][j] = Integer.valueOf(NilaiC1[j].toString());
        DataC2[i][j] = Integer.valueOf(NilaiC2[j].toString());
    }
}

raster.setSample(i, j, 0, 255);
ThresholdingCitraHasil += 1 + " ";
}else{
    raster.setSample(i, j, 0, 0);
    ThresholdingCitraHasil += 0 + " ";
}
ThresholdingCitraHasil += ":";
}

```

9. *Logika AND*, melakukan operasi AND antara vektor citra input.

```

DataC1[i][j] = Integer.valueOf(NilaiC1[j].toString());
DataC2[i][j] = Integer.valueOf(NilaiC2[j].toString());

if (DataC1[i][j] == 1 && DataC2[i][j] == 1) {
    raster.setSample(i, j, 0, 255);
    ThresholdingCitraHasil += 1 + " ";
}else{
    raster.setSample(i, j, 0, 0);
    ThresholdingCitraHasil += 0 + " ";
}
}

```

10. *Logika OR*, melakukan operasi OR antara vektor citra input.

```

DataC1[i][j] = Integer.valueOf(NilaiC1[j].toString());
DataC2[i][j] = Integer.valueOf(NilaiC2[j].toString());

if (DataC1[i][j] == 1 || DataC2[i][j] == 1) {
    raster.setSample(i, j, 0, 255);
    ThresholdingCitraHasil += 1 + " ";
}else{
    raster.setSample(i, j, 0, 0);
    ThresholdingCitraHasil += 0 + " ";
}
}

```

11. *Logika XOR*, melakukan operasi XOR antara kedua vektor citra input.

```

if (DataC1[i][j] != DataC2[i][j]) {
    raster.setSample(i, j, 0, 255);
    ThresholdingCitraHasil += 1 + " ";
}else{
    raster.setSample(i, j, 0, 0);
    ThresholdingCitraHasil += 0 + " ";
}
}
ThresholdingCitraHasil += ":";
}

```

12. *Logika NOT*, melakukan operasi NOT pada setiap citra input atau hasil.

```

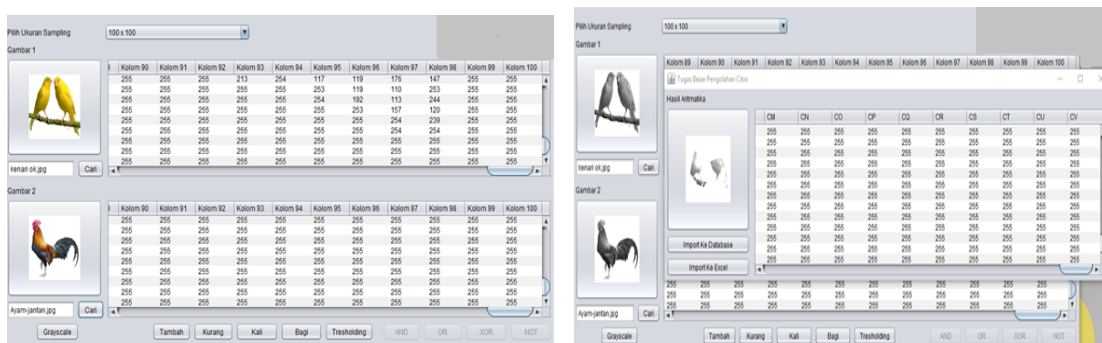
if (DataC1[i][j] == 0) {
    rasterC1.setSample(i, j, 0, 255);
    TresholdingCitra1 += 1 + ",";
} else {
    rasterC1.setSample(i, j, 0, 0);
    TresholdingCitra1 += 0 + ",";
}

if (DataC2[i][j] == 0) {
    rasterC2.setSample(i, j, 0, 255);
    TresholdingCitra2 += 1 + ",";
} else {
    rasterC2.setSample(i, j, 0, 0);
    TresholdingCitra2 += 0 + ",";
}
}
TresholdingCitra1 += "-";
TresholdingCitra2 += "-";

```

### 3.1.3 Tampilan Data Aplikasi

Interface operasi aritmatika yang mencakup semua proses pengolahan citra dasar yang meliputi sampling, greyscale, dan operasi aritmatika. Hal ini ditunjukkan pada gambar 3.2. Pilihan ukuran sampling 100x100 akan mengubah size citra semula menjadi 100x100 piksel. Tombol cari akan menampilkan citra input, yaitu citra kenari.jpg dan citra ayam-jantan.jpg. Kedua citra akan menempati bingkainya. Tombol greyscale akan menampilkan vektor citra atau nilai matriks citra input pada database masing-masing. Bila salah satu tombol operasi dilakukan (misalnya tombol Tambah) maka kedua citra input dan citra hasil akan diubah visualisasinya menjadi citra keabuan, selanjutnya nilai vektornya juga akan ditampilkan pada database saat mengklik tombol Tambah seperti ditunjukkan gambar 3.3.



**Gambar 3.2 Interface Operasi Aritmatika**

Citra Input, citra 1 (Gambar 3.3) dan citra 2 (Gambar 3.4) merupakan masukan bagi aplikasi sekaligus obyek pengolahan datanya.



**Gambar 3.3 Citra 1 (Citra Asli)**

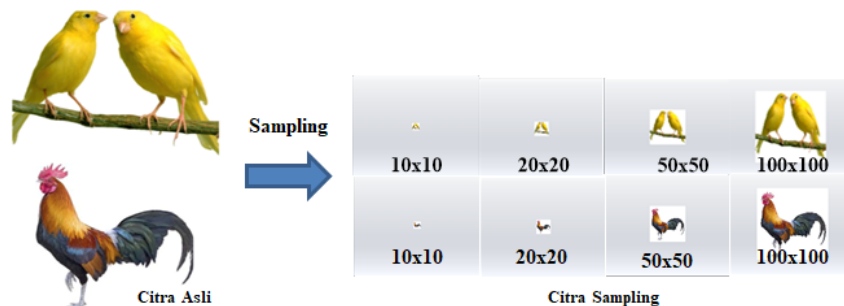


**Gambar 3.4 Citra 2 (Citra Asli)**

Citra 1 sebagai file kenari.jpg memiliki size (baris x kolom) 860 x 1197 piksel dan citra 2 (ayamjantan.jpg) memiliki size 298 x 300 piksel. Size kedua citra akan dikonversi sesuai dengan pemilihan sampling yang dilakukan saat mengakses aplikasi.

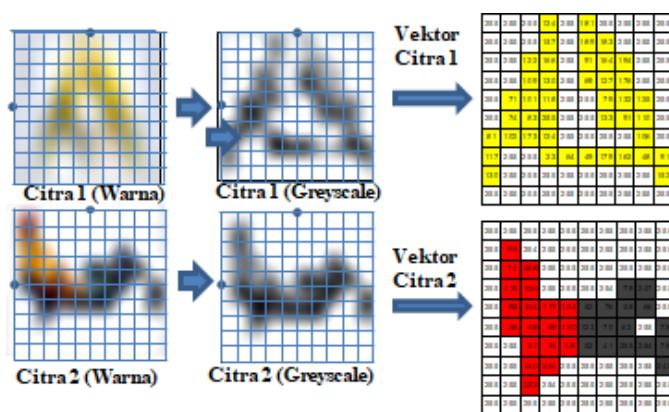
Gambar 3.5 menunjukkan sampling citra dilakukan secara berturut-turut pada citra 10x10, 20x20, 50x50 dan 100x100 memiliki 100, 400, 2.500, dan 10.000 piksel sehingga sizanya jauh lebih kecil dari ukuran aslinya. Perbandingan size citra asli dengan ukuran sampling berturut-turut berkisar 10 ribu,

2500 ribu, 411 ribu dan 102 ribu kali dari citra aslinya. Tujuan sampling dalam hal ini untuk memudahkan analisis dan memudahkan visualisasi vektor citra pada database.



Gambar 3.5 Sampling Citra Input

Gambar 3.6 merupakan citra input yang mengalami greyscale (keabuan) dengan sampling 10x10 piksel. Skala keabuan 8 bit yang memiliki 255 level. Pada gambar tersebut, secara visual gambar mengalami blur karena resolusinya rendah kemudian vektor citranya menunjukkan nilai representasi numerik atau matriks seperti ditunjukkan pada pemetaan nilai piksel citranya. Citra greyscale dibagi menjadi 100 kotak *square* kemudian nilai ekstraksi citra diperlihatkan dengan citra warna keabuan. Nilai tersebut sebagai nilai obyek kenari ( $C_1$ ) dan ayam-jantan ( $C_2$ ).

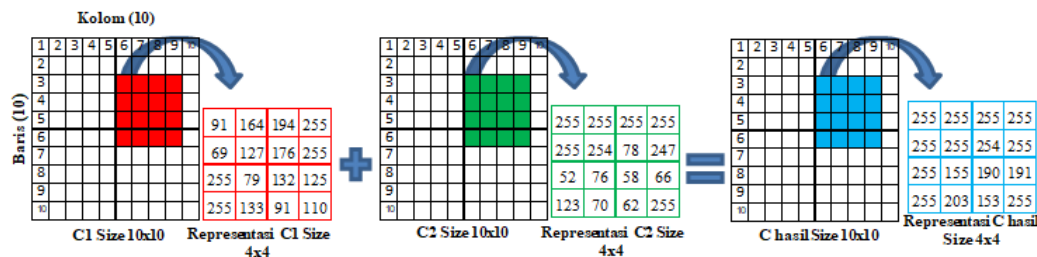


Gambar 3.6 Greyscale Citra Sampling 10x10 piksel



Gambar 3.7 Proses Penjumlahan Citra

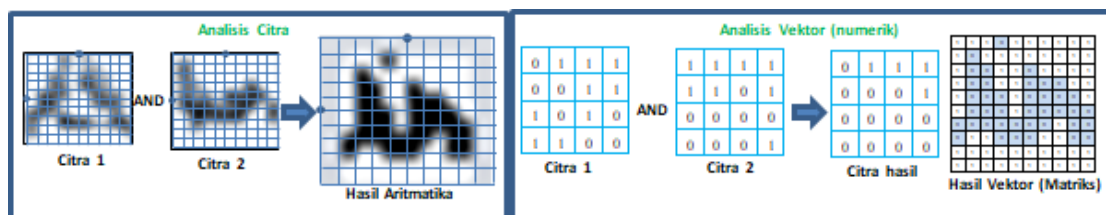
Gambar 3.7 menunjukkan operasi penjumlahan antara citra 1 dan citra 2 menghasilkan citra hasil yang lebih terang secara visual. Sedangkan secara numerik, hasil penjumlahan vektor akan mengakibatkan bertambahnya nilai vektor (matriks) sehingga secara visual hal ini menunjukkan citra hasil semakin terang dibandingkan kedua citra inputnya.



Gambar 3.8 Sampling 4x4 Pixel pada Operasi Penjumlahan

Untuk mengurangi banyaknya angka sebagai analisis perhitungan, maka vektor citra 10x10 disampling menjadi size 4x4 piksel seperti pada gambar 3.8. Operasi penjumlahan antara vektor Citra 1 ( $C_1$ ) dan vektor  $C_2$  akan menghasilkan  $C$  hasil. Misalnya piksel  $fc_1(0,0)$  yaitu  $C_1 = 91$ ,  $fc_2(0,0)$  yaitu  $C_2 = 255$ , maka  $fc_{\text{hasil}}(0,0)$  yaitu  $C_{\text{Hasil}} = 346$ ; hasil penjumlahan  $C_{\text{Hasil}} > 255$ , maka diflipping (dipotong) menjadi 255 (nilai tertinggi level keabuan). Sedangkan penjumlahan piksel yang menghasilkan vektor  $\leq 255$  maka hasilnya dituliskan sebagai nilai pada range 0 – 255, misalnya piksel  $fc_1(2,1)$  yaitu  $C_1 = 79$ ,  $C_2 = 76$ , maka  $C_{\text{Hasil}} = 155$ .

Operasi aritmatika secara lengkap terangkum pada tabel 3.1. Tabel 3.1 menjelaskan operasi aritmatika meliputi: greyscale dua citra input, penjumlahan, pengurangan, perkalian, pembagian, thresholding citra input, logika AND, OR, XOR, dan NOT. Setiap operasi ditandai dengan citra visual hasil operasinya, kemudian nilai vektornya ditunjukkan dalam pemetaan size 10x10 piksel = 100 kotak piksel. Peta piksel tersebut menunjukkan skala keabuannya yang merupakan nilai numerik dari citra hasilnya.



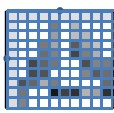
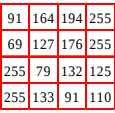
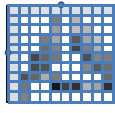
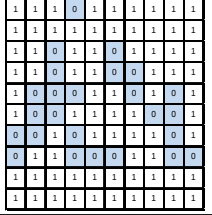
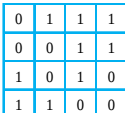
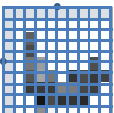
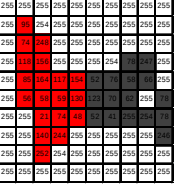
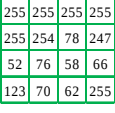
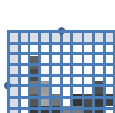
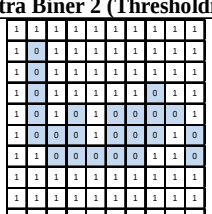
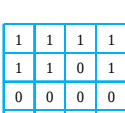
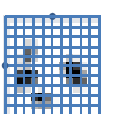
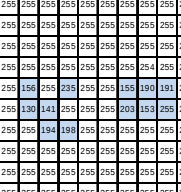
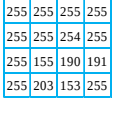
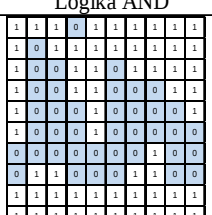
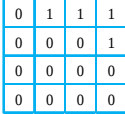
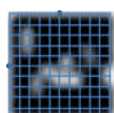
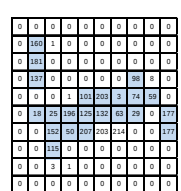
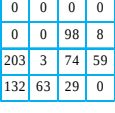
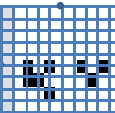
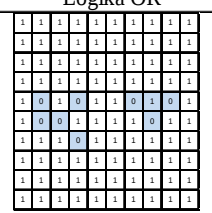
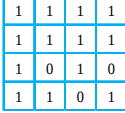
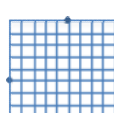
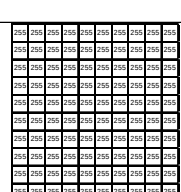
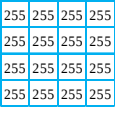
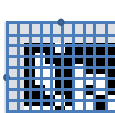
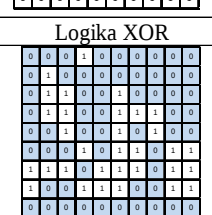
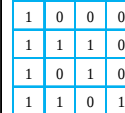
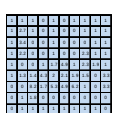
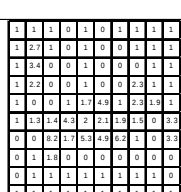
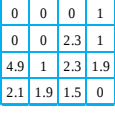
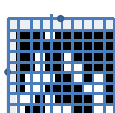
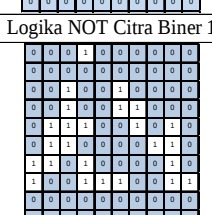
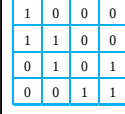
Gambar 3.9 Analisis Operasi Aritmatika pada Citra

Pada gambar 3.9, analisis terhadap operasi aritmatika terdiri dari dua, yaitu analisis terhadap citra dan analisis terhadap vektor. Pada bagian *analisis citra*, yang menjadi obyek pemrosesan secara visual adalah berupa citra input yang dioperasikan menggunakan logika AND. Hasilnya ditunjukkan pada gambar 3.9 yang mana obyek citra hasilnya berwarna hitam dengan pola tertentu yang dikelilingi warna putih. Sedangkan bagian *analisis vektor*, yang menjadi pemrosesan adalah nilai numerik greyscale. Citra input dioperasikan menggunakan logika AND, yang mana jika salah satu inputnya bernilai 0, maka outputnya bernilai 0; outputnya akan bernilai 1 jika kedua inputnya berlogika 1. Hasil pemetaan vektor lengkap 10x10 menunjukkan sebaran logika 0 dan logika 1. Pola hasil vektor sama dengan pola citra hasil. Representasi 0 pada peta menunjukkan warna hitam pada citra, 1 menunjukkan warna putih. Secara visual, terdapat kesamaan pola antara citra hasil dengan hasil vektor.

Untuk melihat secara jelas perbandingan analisis citra dengan vektornya ditunjukkan melalui tabel 3.1 untuk semua operasi aritmatika. Nilai vektor dengan size 4x4 hanya sebagai minimalisasi dari ukuran citra yang sebenarnya sehingga bisa menghemat jumlah halaman analisis.

Beberapa kesimpulan yang bisa diambil berdasarkan hasil pengolahan data yaitu: 1) citra dengan nilai piksel kecil akan mengalami blur (kekaburan) karena resolusinya rendah akibat jumlah piksel kecil yang terjadi pada size 10x10 dan 20x20, sedangkan size 50x50 piksel dan 100x100 memiliki resolusi yang cukup. 2) Citra hasil akan menampilkan citra yang lebih terang pada operasi penjumlahan dan perkalian, sedangkan citra yang lebih gelap pada operasi pengurangan dan pembagian. Pada operasi penjumlahan terjadi peningkatan nilai vektor yang berarti nilainya semakin mendekati nilai piksel maksimum (255= warna putih), begitu pula dengan perkalian, pelipatgandaan nilai vektor akan memotong nilai vektor menjadi 255, sehingga secara keseluruhan nilai vektor bernilai 255. Secara visualisasi, citra hasilnya akan tampil lebih terang bahkan keseluruhannya berwarna putih (Tabel 3.1, No.5). Sebaliknya, pada pengurangan citra, akan semakin mendekatkan citra hasil pada nilai vektor terendah yaitu 0 (0 = warna hitam). Terlebih lagi kalau nilai tersebut merupakan hasil pembagian, akan

membuat citra hasilnya bernilai 0 atau berwarna hitam keseluruhan (Tabel 3.1, No.6). 3) operasi logika akan menyederhanakan level warna keabuan hanya pada warna hitam dan putih. Pada proses ini terjadi bipolarisasi nilai, yaitu  $>128$ , bernilai 1 dan di bawahnya bernilai 0. 4) analisis numerik pada citra telah memetakan nilai obyek pada citra terpolarisasi pada nilai vektornya dan jika diberikan arsiran atau pewarnaan luasan, maka akan terlihat kesamaan pola citra hasil dengan pola nilai numerik pada bidang pemetaan dua dimensi.

| Operasi Aritmatika |                                                                                     |                                                                                     |                                                                                     |    |                                                                                     |                                                                                      |                                                                                       |
|--------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| No                 | Citra Hasil                                                                         | Vektor 10x10                                                                        | Representasi Size 4x4                                                               | No | Citra Hasil                                                                         | Vektor 10x10                                                                         | Representasi Size 4x4                                                                 |
| 1                  | Input Citra 1                                                                       |                                                                                     |                                                                                     | 7  | Citra Biner 1 (Thresholding)                                                        |                                                                                      |                                                                                       |
|                    |    |    |    |    |    |    |    |
| 2                  | Input Citra 2                                                                       |                                                                                     |                                                                                     | 8  | Citra Biner 2 (Thresholding)                                                        |                                                                                      |                                                                                       |
|                    |    |    |    |    |    |    |    |
| 3                  | Penjumlahan                                                                         |                                                                                     |                                                                                     | 9  | Logika AND                                                                          |                                                                                      |                                                                                       |
|                    |  |  |  |    |  |  |  |
| 4                  | Pengurangan                                                                         |                                                                                     |                                                                                     | 10 | Logika OR                                                                           |                                                                                      |                                                                                       |
|                    |  |  |  |    |  |  |  |
| 5                  | Perkalian                                                                           |                                                                                     |                                                                                     | 11 | Logika XOR                                                                          |                                                                                      |                                                                                       |
|                    |  |  |  |    |  |  |  |
| 6                  | Pembagian                                                                           |                                                                                     |                                                                                     | 12 | Logika NOT Citra Biner 1                                                            |                                                                                      |                                                                                       |
|                    |  |  |  |    |  |  |  |

Tabel 3.1 Operasi Aritmatika

### 3.2 Hasil Pengujian Sistem

Aplikasi pengolahan citra digital untuk memudahkan visualisasi operasi aritmatika sehingga mahasiswa bisa melihat secara visual terhadap input dan output operasi tersebut. Untuk menguji validitas fungsional dari tombol-tombol, maka dilakukan pengujian *black box*. Adapun hasil pengujian *test case*-nya seperti pada tabel 3.2. Tabel 3.2 menunjukkan bahwa terdapat 15 fungsi tombol yang diuji, dan semuanya berfungsi sesuai dengan yang diharapkan.

**Tabel 3.2 Hasil Pengujian Black Box**

| No | Test Case                       | Pengujian Black Box                                                                                                                  |                                                                                     | Keterangan |
|----|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------|
|    |                                 | Fungsi                                                                                                                               | Hasil                                                                               |            |
| 1  | Menu Utama                      | Untuk menampilkan menu utama dengan semua tombolnya.                                                                                 | Bisa mengakses menu utama dan menjalankannya.                                       | Valid      |
| 2. | Tombol Sampling                 | Untuk melakukan pemilihan sampling pada combo box: 10x10, 20x20, 50x50, dan 100x100.                                                 | Bisa menjalankan pemilihan sampling.                                                | Valid      |
| 3  | Tombol Cari                     | Untuk melakukan proses pencarian file citra 1 dan citra 2 secara serial pada storage device untuk ditempatkan pada bingkai aplikasi. | Berhasil mengambil file citra secara serial dan menampilkan pada bingkai.           | Valid      |
| 4  | Tombol Greyscale                | Untuk mengkonversi citra 1 dan citra 2 menjadi citra greyscale.                                                                      | Berhasil mengubah citra warna menjadi citra keabuan pada range nilai 0 – 255 level. | Valid      |
| 5  | Tombol Tambah                   | Untuk membuka form Hasil Aritmatika penjumlahan (vektor citra hasil).                                                                | Berhasil menampilkan nilai vektor citra dan citra hasilnya.                         | Valid      |
| 6  | Tombol Export data ke database. | Untuk mengirim hasil operasi aritmatika ke database.                                                                                 | Berhasil mengirimkan nilai vektor ke database.                                      | Valid      |
| 7  | Tombol Import ke Excel          | Untuk mengirim hasil aritmatika dari database ke Excel.                                                                              | Berhasil mengirimkan hasil operasi aritmatika dari database ke Excel.               | Valid      |
| 8  | Tombol Kurang                   | Untuk membuka form Hasil pengurangan (vektor citra hasil).                                                                           | Berhasil menampilkan nilai vektor citra.                                            | Valid      |
| 9  | Tombol Kali                     | Untuk membuka form Hasil perkalian (vektor citra hasil).                                                                             | Berhasil menampilkan citra hasil perkalian dan nilai vektornya.                     | Valid      |
| 10 | Tombol Bagi                     | Untuk membuka form Hasil pembagian (vektor citra hasil).                                                                             | Berhasil menampilkan citra hasil pembagian dan nilai vektornya.                     | Valid      |
| 11 | Tombol Thresholding             | Untuk melakukan thresholding pada Citra 1 dan Citra 2.                                                                               | Berhasil menampilkan citra 1 dan citra 2 serta menampilkan nilai vektornya.         | Valid      |
| 12 | Tombol AND                      | Untuk membuka form hasil logika AND.                                                                                                 | Berhasil menampilkan citra hasil logika AND dan nilai vektornya.                    | Valid      |
| 13 | Tombol OR                       | Untuk membuka form hasil logika OR.                                                                                                  | Berhasil menampilkan citra hasil logika OR dan nilai vektornya.                     | Valid      |
| 14 | Tombol XOR                      | Untuk membuka form hasil logika XOR.                                                                                                 | Berhasil menampilkan citra hasil logika XOR dan nilai                               | Valid      |

| No | Test Case  | Pengujian <i>Black Box</i>           |                                                                  | Keterangan |
|----|------------|--------------------------------------|------------------------------------------------------------------|------------|
|    |            | Fungsi                               | Hasil                                                            |            |
|    |            |                                      | vektornya.                                                       |            |
| 15 | Tombol NOT | Untuk membuka form hasil logika NOT. | Berhasil menampilkan citra hasil logika NOT dan nilai vektornya. | Valid      |

#### 4 KESIMPULAN

Aplikasi sederhana menggunakan operasi aritmatika sudah dibuat, berfungsi sebagai media untuk menganalisis pengolahan citra menggunakan operasi aritmatika dengan teknik pengamatan visual dan analisis numerik. Aplikasi sudah melalui tahap pengujian *black box* dan semua *test-case*-nya valid dan sesuai dengan yang diharapkan.

Aplikasi ini menjadi acuan pengembangan operasi pengolahan citra lainnya. Mahasiswa mampu untuk mengelaborasi kemampuan coding mereka untuk membuat dan mengembangkan aplikasi sederhana berbasis pengolahan citra.

Aplikasi ini masih sangat sederhana, belum bisa menampilkan pola sekaligus memberikan luasan pada titik operasi citra. Perlu dilakukan pengembangan terhadap aplikasi ini agar mahasiswa dan dosen mampu memvisualisasikan metode yang rumit menjadi metode yang mudah dimengerti. Pengolahan tidak hanya sekedar teori hitungan semata secara manual. Akan tetapi dapat mengasah kemampuan dosen dan mahasiswa untuk mengeksplorasi penelitian tugas akhir.

#### DAFTAR PUSTAKA

- [1] Intan, I, Pattern Recognition of Thyroid Carcinoma Histopatology with Adaptive Neural Fuzzy Inference System, in Proc. 3th Macassar International Conference Electrical Engineering and Informatics, 2012, 2(1) pp. 157-159.
- [2] Intan, I., Salman, N., Klasifikasi Karsinoma Tiroid Menggunakan Gaussian Markov Random Field dan Klasifikasi Self Organizing Map Kohonen, Semnas Teknomedia 2016, 4(1), hal.4-4-13.
- [3] Intan, I, Combining of feature extraction for real-time facial authentication system, 2017 5th International Conference on Cyber and IT Service Management (CITSM), 2017, pp. 1-6. doi: 10.1109/CITSM.2017.8089223
- [4] Oladimeji, I.W., Folasade, I., Adeleye, F.S. American International Journal of Research in Science, Technology, Engineering & Mathematics, 20(1), SeptemberNovember, 2017, pp. 66-72
- [5] Alzoubiady, L.M., Saleh, I.A. Personal Identification based on Hybrid Artificial Intelligence Technique using Serial Mode Architecture. International Journal of Computer Applications, 2014, 87(18), pp. 0975-8887.
- [6] Bakti, V.K., Dairoh, Huda, M., Segmentasi dan Perbaikan Citra untuk Proses Pengukuran Dimensi Beras, Jurnal Infotel, 2016, 8(1), hal. 88-93.
- [7] Murdiyati, S.R., Danuedoro, P., Jatmiko, R.H., Integrasi Transformasi Spektral Citra Landsat Etm+ dan Sig Untuk Pemetaan Pola Rotasi Tanam Lahan Sawah Kabupaten dan Kota Semarang serta Daerah Sekitarnya di Jawa Tengah, Majalah Geografi Indonesia, 24(2), 2010, hal.121-141.
- [8] Ogunlana, S.O., Olabode, O., Oluwadare, S.A.A., Iwasokun, G.B., Fish Classification Using Support Vektor Machine, African Journal. Computing ICT, 2015, 8(2).
- [9] Putri, A.R., Pengolahan Citra dengan Menggunakan Web Cam pada Kendaraan Bergerak di Jalan Raya, Jurnal Ilmiah Pendidikan Informatika, 1(1), 2016, hal. 1-6.
- [10] Desviana, Safitri, R., Syahreza, R., Identifikasi Citra CT Scan Tumor Paru Dengan Memvariasikan Nilai Tepi Kuat Dan Lemah Berbasis Algoritma Deteksi Tepi Canny, J. Aceh Phys. Soc, 7(2), 2018, pp.61-66.
- [11] Wibowo, H., Purnomo, M.H., Yuniarno, E.M., Deteksi Gerak Otot Frontalis Berbasis Citra 3 Dimensi Menggunakan Gray Level Co-Occurrence Matrix, Kinetik, 2016, 1(2), Hal. 55-62.
- [12] Putra, D., Pengolahan Citra Digital, Penerbit Andi, 2010.
- [13] Wahana Komputer, Membangun Sistem Informasi dengan Java Netbeans dan MySQL, 2015, Wahana Komputer dan Penerbit Andi.