



**PERBANDINGAN CITRA RGB DAN *GRAYSCALE* PADA
PENGKODEAN *IMAGE* DENGAN ALGORITMA 3D
*PLAYFAIR***

SKRIPSI

Oleh

**Farokhi Abdiansyah
NIM 161810101057**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020**



**PERBANDINGAN CITRA RGB DAN *GRAYSCALE* PADA
PENGKODEAN *IMAGE* DENGAN ALGORITMA 3D
*PLAYFAIR***

SKRIPSI

disusun guna memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Oleh

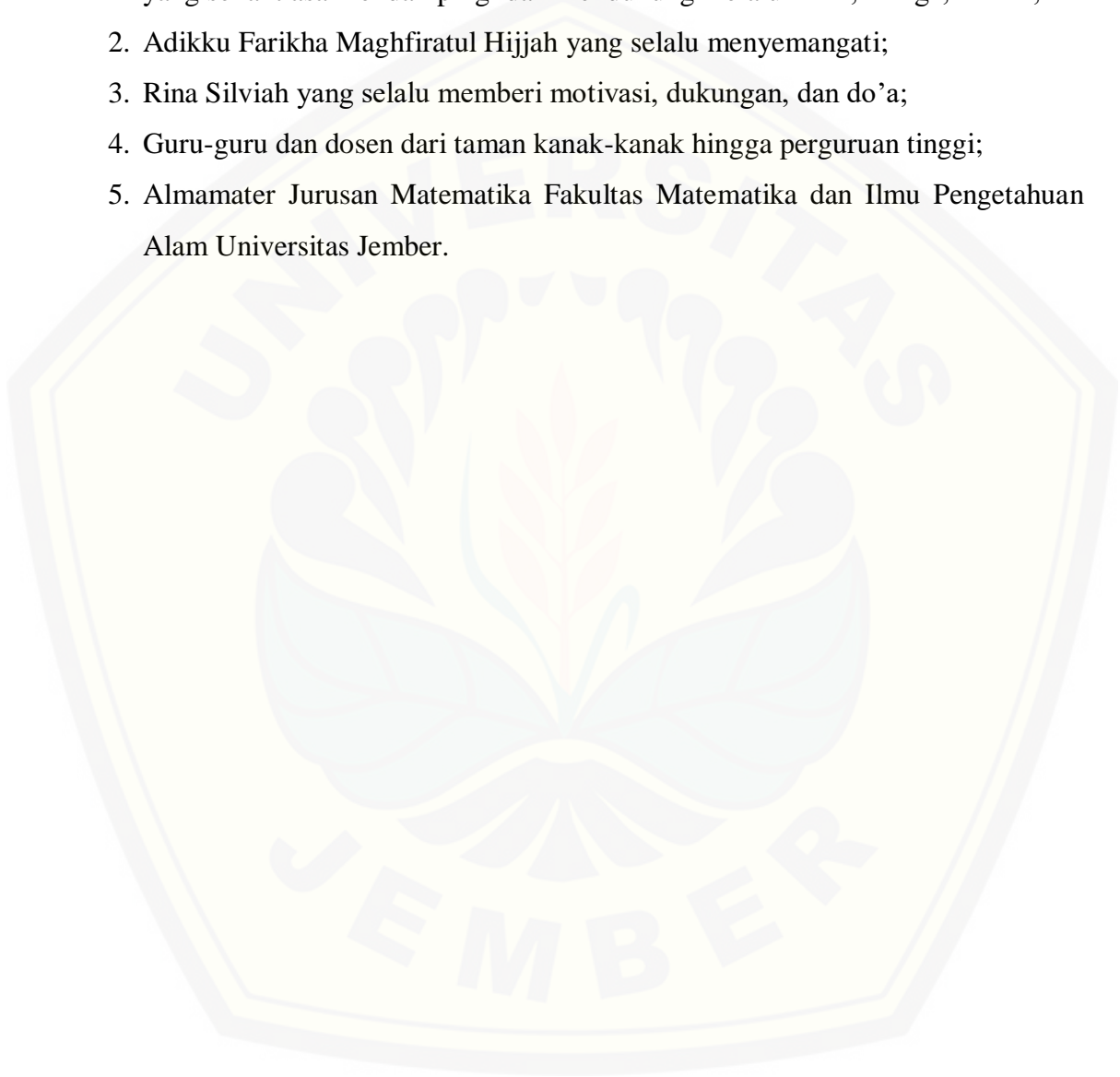
**Farokhi Abdiansyah
NIM 161810101057**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2020**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Kedua Orang tua tersayang, Ayahanda Anton Sulistiyono dan Ibunda Hartini yang senantiasa mendampingi dan mendukung melalui do'a, tenaga, materi;
2. Adikku Farikha Maghfiratul Hijjah yang selalu menyemangati;
3. Rina Silviah yang selalu memberi motivasi, dukungan, dan do'a;
4. Guru-guru dan dosen dari taman kanak-kanak hingga perguruan tinggi;
5. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.



MOTTO

“Belajarliah mengucap syukur dari hal-hal baik di hidupmu dan belajarliah menjadi pribadi yang kuat dengan hal-hal buruk dihidupmu”.¹



¹Bacharuddin Jusuf Habibie

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Farokhi Abdiansyah

NIM : 161810101057

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Perbandingan Citra RGB dan *Grayscale* pada Pengkodean *Image* dengan Algoritma *3D Playfair*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Agustus 2020
Yang menyatakan,

Farokhi Abdiansyah
NIM 161810101057

SKRIPSI

**PERBANDINGAN CITRA RGB DAN *GRAYSCALE* PADA
PENGKODEAN *IMAGE* DENGAN ALGORITMA 3D
*PLAYFAIR***

Oleh

**Farokhi Abdiansyah
NIM 161810101057**

Pembimbing

Dosen Pembimbing Utama : Dr. Kiswara Agung Santoso, S.Si., M.Kom.

Dosen Pembimbing Anggota : Ahmad Kamsyakawuni, S.Si., M.Kom.

PENGESAHAN

Skripsi berjudul “Perbandingan Citra RGB dan *Grayscale* pada Pengkodean *Image* dengan Algoritma *3D Playfair*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Ketua, Tim Penguji: Anggota I,
Dr. Kiswara Agung Santoso, S.Si., M.Kom. Ahmad Kamsyakawuni, S.Si., M.Kom.
NIP. 197209071998031003 NIP. 197211291998021001
Anggota II, Anggota III,
Kusbudiono, S.Si., M.Si. Dr. Yuliani Setia Dewi, S.Si., M.Si.
NIP. 197704302005011001 NIP. 197407162000032001

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Drs. Achmad Sjaifullah, M.Sc., Ph.D.
NIP. 195910091986021001

RINGKASAN

Perbandingan Citra RGB dan Grayscale pada Pengkodean Image dengan Algoritma 3D Playfair; Farokhi Abdiansyah, 161810101057; 2020; 123 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan informasi memudahkan manusia dalam proses pengiriman data. Bentuk data dapat berupa citra yang berformat RGB dan *grayscale*. Jika ada rekayasa citra atau penyebaran citra secara tidak sah atau ilegal yang dapat merugikan pemilik atau suatu lembaga, maka perlu mengamankan citra pengirim dari pengguna yang tidak sah atau ilegal agar citra yang dikirim dapat tersampaikan dengan aman. Cara untuk mengamankan data informasi dapat menggunakan teknik kriptografi. Pengertian kriptografi adalah salah satu ilmu yang digunakan untuk meningkatkan aspek keamanan suatu pesan atau data informasi.

Penelitian ini bertujuan untuk meningkatkan keamanan pada pengamanan citra menggunakan Algoritma 3D Playfair. Kunci yang digunakan pada penelitian ini menggunakan empat tabel berukuran 8×8 yang mendukung nilai *pixel* (0-255) kemudian dijadikan acuan untuk proses enkripsi dan deskripsi. Proses enkripsi citra RGB dan citra *grayscale* menggunakan Algoritma 3D Playfair menghasilkan citra yang terlihat acak atau sulit diinterpretasikan. Proses Dekripsi berhasil mengembalikan *cipherimage* seperti citra aslinya.

Berdasarkan data penelitian yang terdiri dari 10 yang telah diuji. Hasil nilai rata-rata NPCR Proses enkripsi Algoritma 3D Playfair menggunakan kunci citra sungai untuk citra RGB sebesar 98% dan citra *grayscale* sebesar 93%, kunci citra laut untuk citra RGB sebesar 98% dan citra *grayscale* sebesar 91%. Hasil nilai NPCR pada citra RGB lebih baik daripada citra *grayscale*. Hasil nilai rata-rata UACI proses enkripsi Algoritma 3D Playfair menggunakan kunci citra sungai untuk citra RGB sebesar 21% dan citra *grayscale* sebesar 21%, kunci citra laut untuk citra RGB sebesar 23% dan citra *grayscale* sebesar 22%. Hasil nilai UACI

pada citra RGB lebih baik daripada citra *grayscale*. Hasil nilai rata-rata NPCR dan UACI Proses dekripsi Algoritma *3D Playfair* untuk citra RGB sebesar 0% dan citra *grayscale* sebesar 0% mengindikasikan citra hasil dekripsi akan semakin mirip dengan citra aslinya atau dapat kembali ke citra awal.



PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perbandingan Citra RGB dan *Grayscale* pada Pengkodean *Image* dengan Algoritma *3D Playfair*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dr. Kiswara Agung Santoso, S.Si., M.Kom. dan Ahmad Kamsyakawuni, S.Si., M.Kom., selaku dosen pembimbing yang telah memberikan bimbingan dan bantuan dalam penyempurnaan skripsi ini;
2. Kusbudiono, S.Si., M.Si. dan Dr. Yuliani Setia Dewi, S.Si., M.Si., selaku dosen penguji yang telah memberikan kritik dan saran yang membangun dalam penyempurnaan skripsi ini;
3. Abduh Riski, S.Si., M.Si., selaku Dosen Pembimbing Akademik (DPA) yang telah membimbing dalam pemilihan matakuliah;
4. Ayahanda Anton Sulistiyono dan Ibunda Hartini yang senantiasa mendampingi dan mendukung melalui tenaga, materi, dan do’a;
5. Adikku Farikha Maghfiratul Hijjah yang senantiasa menyemangati
6. Rina Silviah “Adek Manis” yang senantiasa memberi motivasi, dukungan, semangat dan do’a;
7. Seluruh teman-teman “Misdirection” 2016, Sahabat kontraan (Gilang, Andi, Dika, Lasim), Rekan penelitian kriptografi (Tika Khairani, Giki Krisnawanti, Rahmawati Awaliyah Putri, Rif’atul Makhomah) , Seluruh teman-teman KKN 221 Bodang, dan semua teman-teman Jurusan Matematika Universitas Jember yang telah memberikan kenangan, dukungan, bantuan, dan motivasi selama masa perkuliahan ini.
8. Semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menerima segala kritik dan saran yang bersifat membangun dari semua pihak demi kesempurnaan penulisan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Agustus 2020

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBING	v
HALAMAN PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xx
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
BAB 2 TINJAUAN PUSTAKA	3
2.1 Citra	3
2.2 Kriptografi	3
2.3 3D Playfair	5
2.4 Analisis Keamanan	10
2.4.1 <i>Number of Pixels Change Rate (NPCR)</i>	10
2.4.2 <i>Unified Averaged Changed Intensity (UACI)</i>	11

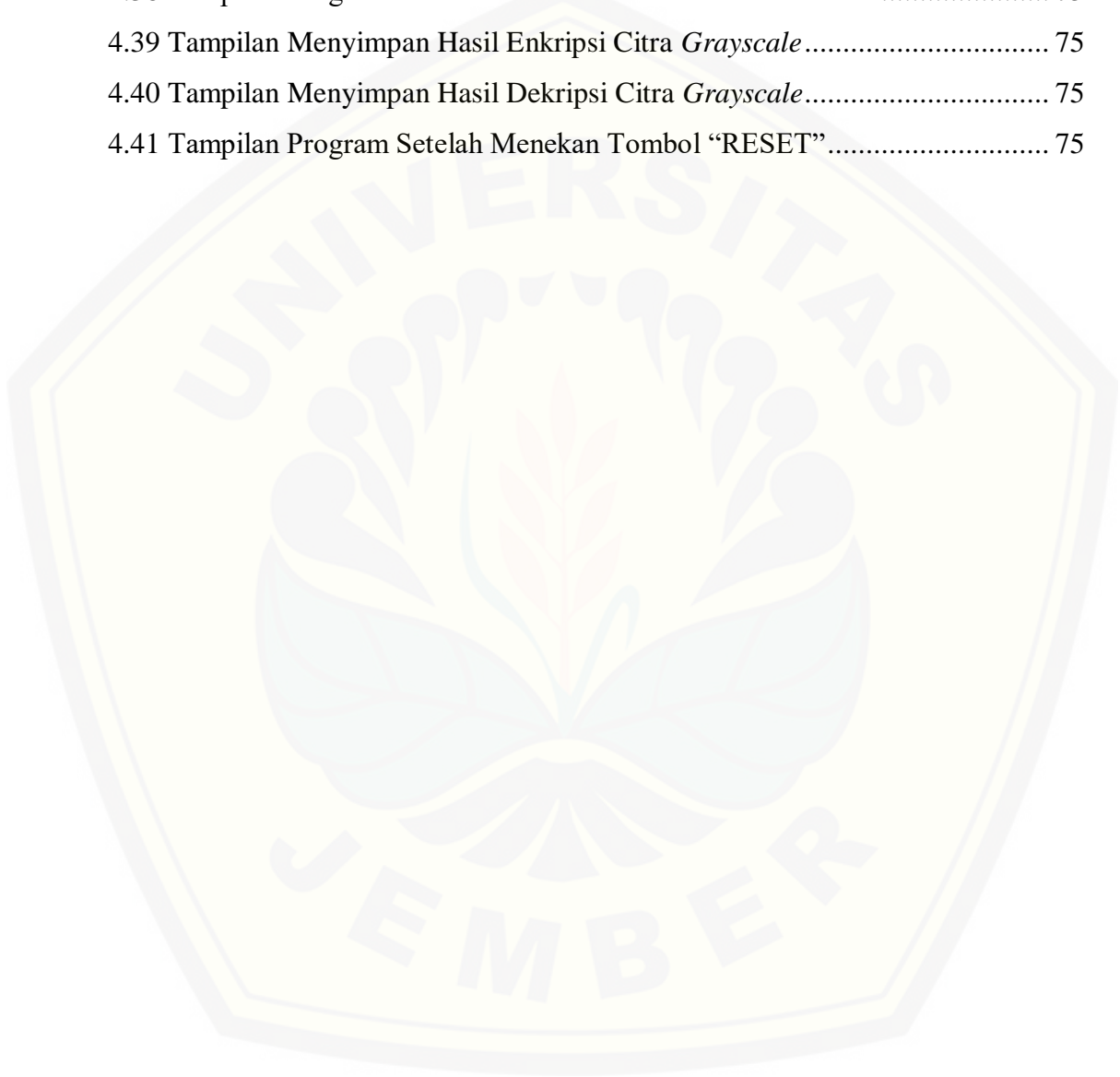
BAB 3 METODE PENELITIAN	12
3.1 Data Penelitian.....	12
3.2 Langkah-langkah Penelitian	13
BAB 4 HASIL DAN PEMBAHASAN.....	21
4.1 Hasil Penelitian	21
4.1.1 Proses Pengkodean Citra <i>Grayscale</i> Menggunakan Algoritma <i>3D Playfair</i>	21
4.1.2 Proses Pengkodean Citra RGB Menggunakan Algoritma <i>3D</i> <i>Playfair</i>	33
4.1.3 Analisis Hasil.....	56
4.1.4 Aplikasi Program	58
4.1.5 Hasil Pengkodean Citra RGB dan <i>Grayscale</i> Menggunakan Aplikasi Program	76
4.2 Pembahasan	81
4.2.1 Proses Enkripsi	81
4.2.2 Proses Dekripsi	82
4.2.3 Aplikasi Program	82
4.2.4 Hasil Nilai NPCR.....	83
4.2.5 Hasil Nilai UACI	84
BAB 5 PENUTUP	85
5.1 Kesimpulan	85
5.2 Saran	85
DAFTAR PUSTAKA	86
LAMPIRAN	87

DAFTAR GAMBAR

	Halaman
2.1 Empat Tabel Kunci <i>3D Playfair</i> Ukuran 8×8 (0-255).....	5
2.2 Empat Tabel Kunci <i>3D Playfair</i> Ukuran 8×8	8
2.3 Citra Apel (236×257) dengan Sampel Nilai <i>Pixel Grayscale</i>	8
3.1 <i>Plainimage</i>	12
3.2 Kunci Citra	13
3.3 Proses Enkripsi <i>Plainimage</i>	13
3.4 <i>Plainimage</i> Citra <i>Grayscale</i> Matriks $m \times n$	14
3.5 <i>Plainimage</i> Citra RGB Matriks $m \times n$	15
3.6 Empat Tabel Kunci Citra <i>Grayscale 3D Playfair</i> 8×8	16
3.7 Empat Tabel Kunci Citra RGB (<i>Red</i>) <i>3D Playfair</i> 8×8	17
3.8 Empat Tabel Kunci Citra RGB (<i>Green</i>) <i>3D Playfair</i> 8×8	17
3.9 Empat Tabel Kunci Citra RGB (<i>Blue</i>) <i>3D Playfair</i> 8×8	18
3.10 Proses Dekripsi <i>Cipherimage</i>	19
3.11 Diagram Alir Penelitian.....	20
4.1 Nilai <i>Pixel</i> Citra Laut <i>Grayscale</i>	21
4.2 Empat Tabel Kunci Citra <i>Grayscale</i> Sebelum Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	23
4.3 Empat Tabel Kunci Citra <i>Grayscale</i> Setelah Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	24
4.4 Nilai <i>Pixel</i> Citra Mawar <i>Grayscale</i>	24
4.5 <i>Cipherimage</i> Citra Mawar <i>Grayscale</i>	28
4.6 Nilai <i>Pixel Cipherimage</i> Citra Mawar <i>Grayscale</i>	29
4.7 <i>Plainimage</i> Citra Mawar <i>Grayscale</i>	33
4.8 Nilai <i>Pixel</i> Citra Laut RGB	34
4.9 Empat Tabel Kunci Citra RGB (<i>red</i>) Sebelum Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	38

4.10 Empat Tabel Kunci Citra RGB (<i>green</i>) Sebelum Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	39
4.11 Empat Tabel Kunci Citra RGB (<i>blue</i>) Sebelum Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	40
4.12 Empat Tabel Kunci Citra RGB (<i>red</i>) Setelah Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	40
4.13 Empat Tabel Kunci Citra RGB (<i>green</i>) Setelah Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	41
4.14 Empat Tabel Kunci Citra RGB (<i>blue</i>) Setelah Diurutkan Berdasarkan Frekuensi Secara <i>Ascending</i>	42
4.15 Nilai <i>Pixel</i> Citra Mawar RGB.....	42
4.16 <i>Cipherimage</i> Citra Mawar RGB	49
4.17 Nilai <i>Pixel Cipherimage</i> Citra Mawar RGB.....	49
4.18 <i>Plainimage</i> Citra Mawar RGB	56
4.19 Pembuatan Program Proses Enkripsi	59
4.20 Pembuatan Program Proses Dekripsi	63
4.21 Tampilan Program Enkripsi dan Dekripsi Citra RGB dan <i>Grayscale</i>	67
4.22 Tampilan Program Setelah Menekan Tombol “BUKA CITRA”	69
4.23 Tampilan Program Setelah Memilih <i>File</i> Citra RGB	69
4.24 Tampilan Program Setelah Menekan Tombol “KUNCI CITRA”	69
4.25 Tampilan Program Setelah Memilih <i>File</i> Citra RGB yang Akan dijadikan sebagai Kunci Citra RGB.....	70
4.26 Tampilan Program Setelah Menekan Tombol “ENKRIPSI”	70
4.27 Tampilan Program Setelah Menekan Tombol “DEKRIPSI”	70
4.28 Tampilan Program Setelah Menekan Tombol “ANALISIS”	71
4.29 Tampilan Menyimpan Hasil Enkripsi Citra RGB.....	71
4.30 Tampilan Menyimpan Hasil Dekripsi Citra RGB	72
4.31 Tampilan Program Setelah Menekan Tombol “RESET”	72
4.32 Tampilan Program Setelah Menekan Tombol “BUKA CITRA”	72
4.33 Tampilan Program Setelah Memilih <i>File</i> Citra <i>Grayscale</i>	73
4.34 Tampilan Program Setelah Menekan Tombol “KUNCI CITRA”	73

4.35 Tampilan Program Setelah Memilih <i>File Citra Grayscale</i> yang Akan Dijadikan Sebagai Kunci Citra <i>Grayscale</i>	73
4.36 Tampilan Program Setelah Menekan Tombol “ENKRIPSI”	74
4.37 Tampilan Program Setelah Menekan Tombol “DEKRIPSI”	74
4.38 Tampilan Program Setelah Menekan Tombol “ANALISIS”	75
4.39 Tampilan Menyimpan Hasil Enkripsi Citra <i>Grayscale</i>	75
4.40 Tampilan Menyimpan Hasil Dekripsi Citra <i>Grayscale</i>	75
4.41 Tampilan Program Setelah Menekan Tombol “RESET”	75



DAFTAR TABEL

	Halaman
2.1 Proses Enkripsi Citra <i>Grayscale</i> Menggunakan <i>3D Playfair</i>	6
2.2 Proses Dekripsi Citra <i>Grayscale</i> Menggunakan <i>3D Playfair</i>	7
2.3 Proses Enkripsi Citra <i>Grayscale (Trigraphs-1)</i>	8
2.4 Proses Enkripsi Citra <i>Grayscale (Trigraphs-2)</i>	9
2.5 Proses Dekripsi Citra <i>Grayscale (Trigraphs-1)</i>	9
2.6 Proses Dekripsi Citra <i>Grayscale (Trigraphs-2)</i>	9
3.1 Proses Enkripsi Citra RGB Menggunakan <i>3D Playfair</i>	15
3.2 Proses Dekripsi Citra RGB Menggunakan <i>3D Playfair</i>	19
4.1 Kelompok Nilai <i>Pixel</i> Berdasarkan Frekuensi	22
4.2 Proses Enkripsi Citra <i>Grayscale</i> Menggunakan Algoritma <i>3D Playfair</i>	25
4.3 Pengisian Baris, Kolom, Tabel <i>Pixel</i> Pertama <i>Trigraphs</i> pada Tabel Enkripsi <i>3D Playfair</i>	26
4.4 Pengisian Baris, Kolom, Tabel <i>Pixel</i> Kedua <i>Trigraphs</i> pada Tabel Enkripsi <i>3D Playfair</i>	26
4.5 Pengisian Baris, Kolom, Tabel <i>Pixel</i> Ketiga <i>Trigraphs</i> pada Tabel Enkripsi <i>3D Playfair</i>	26
4.6 <i>Pixel</i> Pertama <i>Trigraphs Plainimage</i> Menjadi <i>Pixel</i> Pertama <i>Trigraphs</i> <i>Cipherimage</i> pada Tabel Enkripsi <i>3D Playfair</i>	27
4.7 <i>Pixel</i> Kedua <i>Trigraphs Plainimage</i> Menjadi <i>Pixel</i> Kedua <i>Trigraphs</i> <i>Cipherimage</i> pada Tabel Enkripsi <i>3D Playfair</i>	27
4.8 <i>Pixel</i> Ketiga <i>Trigraphs Plainimage</i> Menjadi <i>Pixel</i> Ketiga <i>Trigraphs</i> <i>Cipherimage</i> pada Tabel Enkripsi <i>3D Playfair</i>	27
4.9 Proses Enkripsi Citra <i>Grayscale (Trigraphs-1)</i>	28
4.10 Proses Enkripsi Citra <i>Grayscale (Trigraphs-2)</i>	28
4.11 Proses Enkripsi Citra <i>Grayscale (Trigraphs-3)</i>	28
4.12 Proses Dekripsi Citra <i>Grayscale</i> Menggunakan <i>3D Playfair</i>	30

4.13 Pengisian Baris, Kolom, Tabel <i>Pixel A Trigraphs</i> pada Tabel Dekripsi 3D <i>Playfair</i>	30
4.14 Pengisian Baris, Kolom, Tabel <i>Pixel B Trigraphs</i> pada Tabel Dekripsi 3D <i>Playfair</i>	30
4.15 Pengisian Baris, Kolom, Tabel <i>Pixel C Trigraphs</i> pada Tabel Dekripsi 3D <i>Playfair</i>	31
4.16 <i>Pixel</i> Pertama <i>Trigraphs Cipherimage</i> Menjadi <i>Pixel</i> Pertama <i>Trigraphs Plainimage</i> pada Tabel Dekripsi 3D <i>Playfair</i>	31
4.17 <i>Pixel</i> Kedua <i>Trigraphs Cipherimage</i> Menjadi <i>Pixel</i> Kedua <i>Trigraphs Plainimage</i> pada Tabel Dekripsi 3D <i>Playfair</i>	32
4.18 <i>Pixel</i> Ketiga <i>Trigraphs Cipherimage</i> Menjadi <i>Pixel</i> Ketiga <i>Trigraphs Plainimage</i> pada Tabel Dekripsi 3D <i>Playfair</i>	32
4.19 Proses Dekripsi Citra <i>Grayscale (Trigraphs-1)</i>	32
4.20 Proses Dekripsi Citra <i>Grayscale (Trigraphs-2)</i>	32
4.21 Proses Dekripsi Citra <i>Grayscale (Trigraphs-3)</i>	33
4.22 Kelompok Nilai <i>Pixel</i> Pada <i>Layer Red</i> Berdasarkan Frekuensi	35
4.23 Kelompok Nilai <i>Pixel</i> Pada <i>Layer Green</i> Berdasarkan Frekuensi.....	36
4.24 Kelompok Nilai <i>Pixel</i> Pada <i>Layer Blue</i> Berdasarkan Frekuensi	37
4.25 Proses Enkripsi Citra RGB Menggunakan 3D <i>Playfair</i>	44
4.26 Pengisian Baris, Kolom, Tabel <i>Pixel R Trigraphs</i> pada Tabel Enkripsi 3D <i>Playfair</i>	44
4.27 Pengisian Baris, Kolom, Tabel <i>Pixel G Trigraphs</i> pada Tabel Enkripsi 3D <i>Playfair</i>	45
4.28 Pengisian Baris, Kolom, Tabel <i>Pixel B Trigraphs</i> pada Tabel Enkripsi 3D <i>Playfair</i>	45
4.29 <i>Pixel R Trigraphs Plainimage</i> Menjadi <i>Pixel B Trigraphs Cipherimage</i> pada Tabel Enkripsi 3D <i>Playfair</i>	46
4.30 <i>Pixel G Trigraphs Plainimage</i> Menjadi <i>Pixel R Trigraphs Cipherimage</i> pada Tabel Enkripsi 3D <i>Playfair</i>	46
4.31 <i>Pixel B Trigraphs Plainimage</i> Menjadi <i>Pixel G Trigraphs Cipherimage</i> pada Tabel Enkripsi 3D <i>Playfair</i>	46

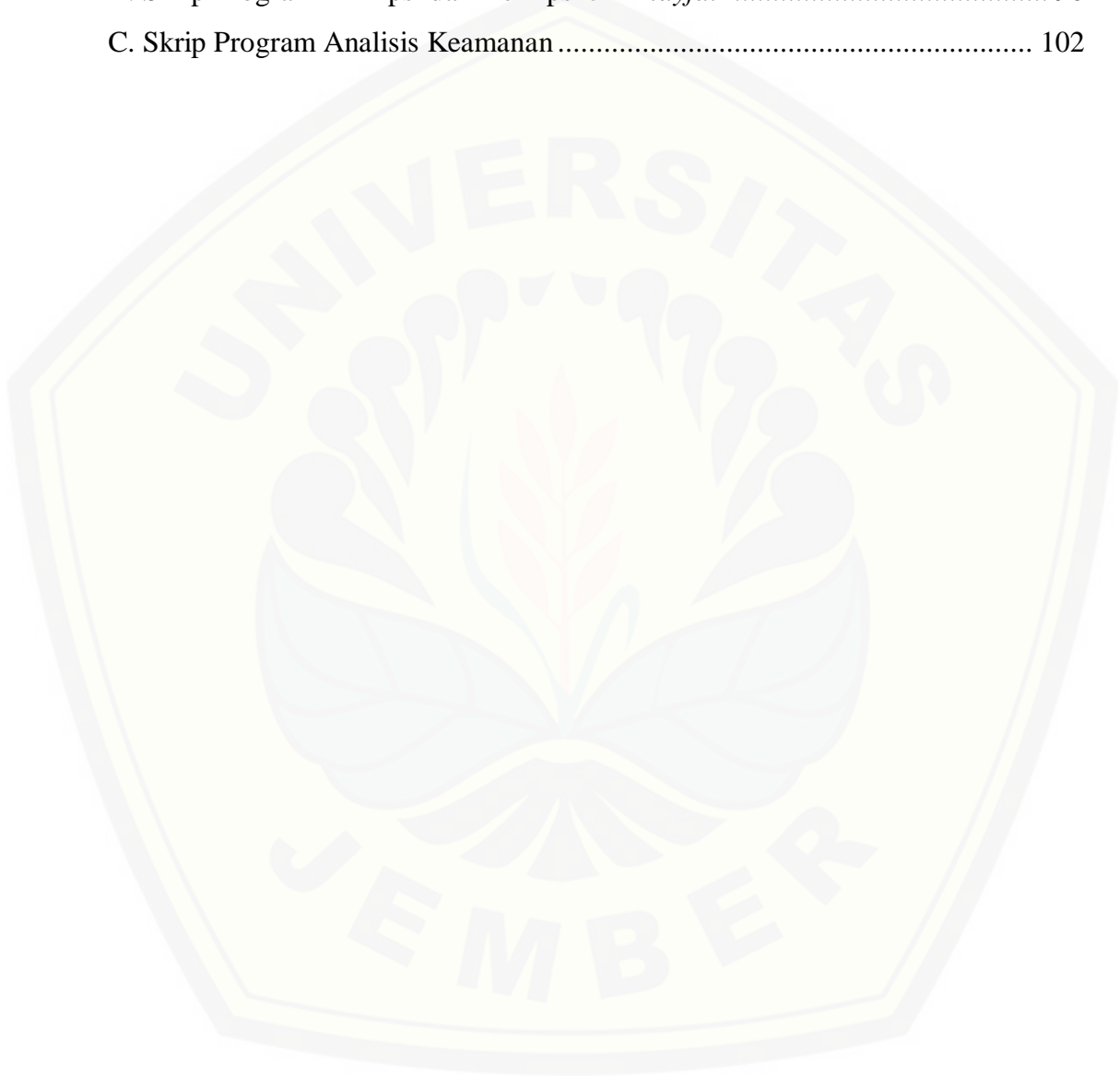
4.32 Proses Enkripsi Citra RGB (<i>Trigraphs-1</i>).....	46
4.33 Proses Enkripsi Citra RGB (<i>Trigraphs-2</i>).....	47
4.34 Proses Enkripsi Citra RGB (<i>Trigraphs-3</i>).....	47
4.35 Proses Enkripsi Citra RGB (<i>Trigraphs-4</i>).....	47
4.36 Proses Enkripsi Citra RGB (<i>Trigraphs-5</i>).....	47
4.37 Proses Enkripsi Citra RGB (<i>Trigraphs-6</i>).....	48
4.38 Proses Enkripsi Citra RGB (<i>Trigraphs-7</i>).....	48
4.39 Proses Enkripsi Citra RGB (<i>Trigraphs-8</i>).....	48
4.40 Proses Enkripsi Citra RGB (<i>Trigraphs-9</i>).....	48
4.41 Proses Dekripsi Citra RGB Menggunakan Algoritma <i>3D Playfair</i>	51
4.42 Pengisian Baris, Kolom, Tabel <i>Pixel B Trigraphs</i> pada Tabel Dekripsi <i>3D Playfair</i>	52
4.43 Pengisian Baris, Kolom, Tabel <i>Pixel R Trigraphs</i> pada Tabel Dekripsi <i>3D Playfair</i>	52
4.44 Pengisian Baris, Kolom, Tabel <i>Pixel G Trigraphs</i> pada Tabel Dekripsi <i>3D Playfair</i>	52
4.45 <i>Pixel B Trigraphs Cipherimage</i> Menjadi <i>Pixel R Trigraphs Plainimage</i> pada Tabel Dekripsi <i>3D Playfair</i>	53
4.46 <i>Pixel R Trigraphs Cipherimage</i> Menjadi <i>Pixel G Trigraphs Plainimage</i> pada Tabel Dekripsi <i>3D Playfair</i>	53
4.47 <i>Pixel G Trigraphs Cipherimage</i> Menjadi <i>Pixel B Trigraphs Plainimage</i> pada Tabel Dekripsi <i>3D Playfair</i>	53
4.48 Proses Dekripsi Citra RGB (<i>Trigraphs-1</i>).....	54
4.49 Proses Dekripsi Citra RGB (<i>Trigraphs-2</i>).....	54
4.50 Proses Dekripsi Citra RGB (<i>Trigraphs-3</i>).....	54
4.51 Proses Dekripsi Citra RGB (<i>Trigraphs-4</i>).....	54
4.52 Proses Dekripsi Citra RGB (<i>Trigraphs-5</i>).....	55
4.53 Proses Dekripsi Citra RGB (<i>Trigraphs-6</i>).....	55
4.54 Proses Dekripsi Citra RGB (<i>Trigraphs-7</i>).....	55
4.55 Proses Dekripsi Citra RGB (<i>Trigraphs-8</i>).....	55
4.56 Proses Dekripsi Citra RGB (<i>Trigraphs-9</i>).....	56

4.57 Hasil Enkripsi pada Program.....	76
4.58 Hasil Dekripsi pada Program.....	78
4.59 Hasil Nilai NPCR dan UACI Proses Enkripsi.....	80
4.60 Hasil Nilai NPCR dan UACI Proses Dekripsi.....	80



DAFTAR LAMPIRAN

	Halaman
A. Skrip Program Pembentukan Empat Tabel Kunci Citra.....	87
B. Skrip Program Enkripsi dan Dekripsi <i>3D Playfair</i>	90
C. Skrip Program Analisis Keamanan	102



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan informasi memudahkan manusia dalam proses pengiriman data. Bentuk data dapat berupa citra yang berformat RGB dan *grayscale*. Citra RGB adalah citra yang terdiri dari warna utama yaitu merah (*red*), hijau (*green*), dan biru (*blue*) disetiap *pixel*. Citra *grayscale* adalah citra yang menampilkan warna abu-abu hasil dari rata-rata citra RGB (Fatta, 2007). Jika ada rekayasa citra atau penyebaran citra secara tidak sah atau ilegal yang dapat merugikan pemilik atau suatu lembaga, maka perlu mengamankan citra pengirim dari pengguna yang tidak sah atau ilegal agar citra yang dikirim dapat tersampaikan dengan aman.

Cara untuk mengamankan data informasi dapat menggunakan teknik kriptografi. Pengertian kriptografi adalah salah satu ilmu yang digunakan untuk meningkatkan aspek keamanan suatu pesan atau data informasi. Kriptografi mendukung kebutuhan dari dua aspek keamanan informasi, yaitu perlindungan terhadap kerahasiaan data informasi (*secrecy*) dan perlindungan terhadap pemalsuan serta perubahan informasi yang tidak diinginkan (*authenticity*) (Ibrahim, 2012). Salah satu algoritma kriptografi yang digunakan untuk merahasiakan data informasi yaitu *3D Playfair*. Algoritma *3D Playfair* telah dikembangkan oleh Sukmawati (2019) dari keamanan data teks menjadi keamanan data citra. Kunci yang digunakan pada penelitian tersebut menggunakan empat tabel berukuran 8×8 yang mendukung nilai *pixel* (0-255) kemudian dijadikan acuan untuk proses enkripsi dan deskripsi. Cara kerja *3D Playfair* dalam bentuk *trigraphs* digunakan untuk proses enkripsi dan deskripsi.

Penelitian yang berkaitan dengan *3D Playfair* dan perbandingan citra RGB dengan citra *grayscale* telah dilakukan oleh beberapa peneliti. Sukmawati (2019) membahas tentang perbandingan *Playfair* dengan *3D playfair* pada pengamanan citra yang menghasilkan nilai histogram dan UACI terbaik pada Algoritma *3D playfair*. Setiadi, dkk (2017) dalam penelitiannya menunjukkan bahwa *one time pad* kriptografi dapat diimplementasikan pada gambar *grayscale* dan gambar RGB.

Berdasarkan uraian sebelumnya, penelitian ini akan melakukan perbandingan citra berformat RGB dan *grayscale* pada pengkodean *image* dengan Algoritma *3D Playfair*. Kunci yang digunakan pada penelitian ini berupa citra berformat RGB dan *grayscale*.

1.2 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana proses mengkodekan citra RGB menggunakan Algoritma *3D Playfair*?
- b. Bagaimana proses mengkodekan citra *grayscale* menggunakan Algoritma *3D Playfair*?
- c. Bagaimana hasil perbandingan tingkat keamanan antara citra RGB dan *grayscale* jika dikodekan dengan Algoritma *3D Playfair*?

1.3 Tujuan Penelitian

Adapun tujuan dalam penelitian ini adalah sebagai berikut:

- a. Mengkodekan citra RGB menggunakan Algoritma *3D Playfair*.
- b. Mengkodekan citra *grayscale* menggunakan Algoritma *3D Playfair*.
- c. Membandingkan tingkat keamanan citra RGB dan *grayscale* jika dikodekan dengan Algoritma *3D Playfair*.

1.4 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

- a. Mengetahui proses enkripsi dan dekripsi menggunakan Algoritma *3D Playfair* untuk citra RGB dan *grayscale*.
- b. Menambah pengetahuan tentang pengkodean *image* menggunakan Algoritma *3D Playfair*.

BAB 2 TINJAUAN PUSTAKA

2.1 Citra

Citra biasanya mengacu ke citra RGB. Pengertian citra RGB adalah citra yang terdiri dari warna utama yaitu merah, hijau, dan biru disetiap *pixel*. Citra RGB terdiri dari triplet dimana setiap triplet memiliki *brightness* dari elemen *red*, *green*, dan *blue*. Setiap triplet merepresentasikan 1 *pixel* (*picture element*), contoh nilai 58, 221, dan 165 berarti mengeset nilai R ke nilai 58, G ke nilai 221, B ke nilai 165. Nilai RGB ini biasanya disebut *color values*. Masing–masing warna utama memiliki nilai 0 – 255, maka totalnya adalah $255^3 = 16.581.375$ (Fatta, 2007). Citra RGB dapat diubah menjadi citra *grayscale*.

Citra *grayscale* adalah citra yang memiliki satu nilai dari setiap *pixel* (*single sample*), artinya *Red = Green = Blue*. Nilai-nilai tersebut menunjukkan intensitas warna. Citra yang terdiri atas warna abu-abu, bervariasi pada warna putih sebagai intensitas terkuat dan warna hitam sebagai intensitas terlemah (Fatta, 2007). Citra RGB dapat diubah menjadi citra *grayscale* menggunakan formula seperti pada persamaan 2.1.

$$\text{Grayscale} = (R + G + B) / 3 \quad (2.1)$$

Keterangan:

R : Unsur warna merah

G : Unsur warna hijau

B : Unsur warna biru

Nilai yang dihasilkan dari persamaan satu akan dimasukkan ke masing-masing unsur warna dasar citra *grayscale* (Noor, 2011).

2.2 Kriptografi

Kriptografi dikenal sekitar tahun 2000 sebelum masehi oleh bangsa Mesir. Berbentuk tulisan *hieroglyphic*, salah satunya terdapat pada dinding Piramida Giza. Bangsa Mesopotamia telah menggunakan kriptografi pada tahun 1500 sebelum masehi, selanjutnya dikenal juga oleh bangsa Yahudi dan bangsa Yunani (Stallings, 1999). Kriptografi (*cryptography*) berasal dari bahasa Yunani, yaitu

terdiri dari dua kata *cryptos* artinya rahasia (*secret*) dan *graphien* artinya tulisan (*writing*). Kriptografi adalah ilmu dan seni untuk menjaga keamanan dan kerahasiaan pesan (Munir, 2006).

Kekuatan algoritma kriptografi terletak pada kerahasiaan kunci yang digunakan untuk proses enkripsi dan dekripsi. Pengirim pesan harus menjaga kerahasiaan kunci tersebut disimpan dan didistribusikan kepada pihak yang berhak menerima pesan (Alfred dkk, 1997). Kunci yang dibuat harus terjaga kerahasiaannya sehingga data kunci semakin aman.

Kriptografi mempunyai beberapa istilah yang sering digunakan (Onno, 2000).

- a. *Plaintext* adalah data dalam bentuk teks sebelum proses enkripsi atau data dalam bentuk teks setelah proses dekripsi. *Plainimage* adalah data dalam bentuk citra sebelum proses enkripsi atau data dalam bentuk citra setelah proses dekripsi.
- b. *Ciphertext* adalah data dalam bentuk teks setelah proses enkripsi atau data dalam bentuk teks sebelum proses dekripsi. *Cipherimage* adalah data dalam bentuk citra setelah proses enkripsi atau data dalam bentuk citra sebelum proses dekripsi.
- c. Kunci (*key*) adalah data yang diketahui oleh pengirim dan penerima yang berhak. Digunakan bersama dengan algoritma kriptografi untuk melakukan proses enkripsi dan dekripsi.
- d. Enkripsi (*encryption*) adalah proses yang digunakan untuk menyamakan *plaintext* menjadi *ciphertext* atau *plainimage* menjadi *cipherimage*.
- e. Dekripsi (*decryption*) kebalikan dari proses enkripsi yaitu mengembalikan *ciphertext* menjadi *plaintext* atau *cipherimage* menjadi *plainimage*.
- f. Kriptosistem (*cryptosystem*) adalah sistem kriptografi yang didalamnya terdiri dari algoritma kriptografi, *plaintext*, *plainimage*, *ciphertext*, *cipherimage*, *key*, dan unsur lain yang berpengaruh dalam sistem kriptografi.
- g. *Cryptanalysis / code breaking* adalah kegiatan untuk mengubah *ciphertext* atau *cipherimage* menjadi pesan aslinya tanpa mengetahui kunci yang sesuai, dengan coba-coba (*trial and error*) secara sistematis.

2.3 3D Playfair

3D Playfair merupakan salah satu algoritma kriptografi menggunakan metode *trigraphs* (tiga *pixel* yang berdekatan) untuk melakukan proses enkripsi dan dekripsi. Kunci *3D Playfair* yang diperoleh dari teks dikonversi kedalam bentuk *decimal* dibuat menjadi empat tabel berukuran 8×8 yang berisi 256 *pixel* (0-255). *Pixel* yang diisi ke dalam empat tabel kunci *3D Playfair* tidak boleh sama (Sukmawati, 2019). Pengisian empat tabel dimulai dari Tabel 1, Tabel 2, Tabel 3, Tabel 4. Empat tabel kunci *3D Playfair* ukuran 8×8 (0-255) dapat dilihat pada Gambar 2.1.

Tabel 1								Tabel 2							
0	1	2	3	4	5	6	7	64	65	66	67	68	69	70	71
8	9	10	11	12	13	14	15	72	73	74	75	76	77	78	79
16	17	18	19	20	21	22	23	80	81	82	83	84	85	86	87
24	25	26	27	28	29	30	31	88	89	90	91	92	93	94	95
32	33	34	35	36	37	38	39	96	97	98	99	100	101	102	103
40	41	42	43	44	45	46	47	104	105	106	107	108	109	110	111
48	49	50	51	52	53	54	55	112	113	114	115	116	117	118	119
56	57	58	59	60	61	62	63	120	121	122	123	124	125	126	127
Tabel 3								Tabel 4							
128	129	130	131	132	133	134	135	192	193	194	195	196	197	198	199
136	137	138	139	140	141	142	143	200	201	202	203	204	205	206	207
144	145	146	147	148	149	150	151	208	209	210	211	212	213	214	215
152	153	154	155	156	157	158	159	216	217	218	219	220	221	222	223
160	161	162	163	164	165	166	167	224	225	226	227	228	229	230	231
168	169	170	171	172	173	174	175	232	233	234	235	236	237	238	239
176	177	178	179	180	181	182	183	240	241	242	243	244	245	246	247
184	185	186	187	188	189	190	191	248	249	250	251	252	253	254	255

Gambar 2.1 Empat Tabel Kunci *3D Playfair* Ukuran 8×8 (0-255)

3D Playfair untuk proses enkripsi adalah memecah *plainimage* (citra asli) menjadi satu kelompok yang terdiri dari tiga *pixel* (*trigraphs*). Ada angka tambahan 254 dan 255 untuk kelompok yang tidak terdiri dari tiga *pixel*, kelompok yang memiliki dua *pixel* ditambahkan 254 dibelakang, kelompok yang

memiliki satu *pixel* ditambahkan 254 dan 255 dibelakang. Proses enkripsi dan dekripsi citra *grayscale* menggunakan *trigraphs* (baris, kolom, tabel). Proses enkripsi citra *grayscale* menggunakan *3D Playfair* ditunjukkan oleh Tabel 2.1.

Tabel 2.1 Proses Enkripsi Citra *Grayscale* Menggunakan *3D Playfair*

<i>Trigraphs Plainimage</i>	<i>Trigraphs Plainimage</i>			<i>Trigraphs Cipherimage</i>
	<i>Pixel Pertama</i>	<i>Pixel Kedua</i>	<i>Pixel Ketiga</i>	
<i>Pixel Pertama</i>	Baris	Kolom	Tabel	<i>Pixel A</i>
<i>Pixel Kedua</i>	Tabel	Baris	Kolom	<i>Pixel B</i>
<i>Pixel Ketiga</i>	Kolom	Tabel	Baris	<i>Pixel C</i>

Trigraphs plainimage terdiri tiga *pixel* diperoleh dari *plainimage*. Cara memperoleh tiga *pixel* dimulai dari *pixel* yang terletak pada baris 1 kolom 1 dinamakan *pixel* pertama, *pixel* yang terletak pada baris 1 kolom 2 dinamakan *pixel* kedua, *pixel* yang terletak pada baris 1 kolom 3 dinamakan *pixel* ketiga. *Pixel* pertama, kedua, dan ketiga disebut *trigraphs plainimage* diletakkan pada baris dan kolom *trigraphs plainimage* seperti Tabel 2.1. Cara mengisi baris, kolom, dan tabel untuk *pixel* pertama, kedua, dan ketiga pada Tabel 2.1 menggunakan kunci citra Gambar 2.1 sebagai acuan pengisian. Urutan pengisian *pixel* pertama (baris, tabel, kolom), *pixel* kedua (kolom, baris, tabel), *pixel* ketiga (tabel, kolom, baris). Kunci citra Gambar 2.1 selain sebagai acuan mengisi baris, kolom, tabel pada *trigraphs plainimage* digunakan juga untuk memperoleh *trigraphs cipherimage*. Urutan *trigraphs cipherimage* pada Tabel 2.1 yaitu baris, kolom, tabel akan memperoleh *pixel A* *trigraphs cipherimage* dari Gambar 2.1, tabel, baris, kolom akan memperoleh *pixel B* *trigraphs cipherimage* dari Gambar 2.1, kolom, tabel, baris akan memperoleh *pixel C* *trigraphs cipherimage* dari Gambar 2.1.

3D Playfair untuk proses dekripsi sama seperti proses enkripsi yang membedakan posisi baris, tabel, dan kolom dalam *trigraphs*. Kunci citra untuk proses dekripsi menggunakan kunci citra Gambar 2.1 yang sama pada proses

enkripsi. Proses dekripsi citra *grayscale* menggunakan *3D Playfair* ditunjukkan oleh Tabel 2.2.

Tabel 2.2 Proses Dekripsi Citra *Grayscale* Menggunakan *3D Playfair*

<i>Trigraphs Cipherimage</i>	<i>Trigraphs Cipherimage</i>			<i>Trigraphs Plainimage</i>
	<i>Pixel A</i>	<i>Pixel B</i>	<i>Pixel C</i>	
<i>Pixel A</i>	Baris	Tabel	Kolom	<i>Pixel Pertama</i>
<i>Pixel B</i>	Kolom	Baris	Tabel	<i>Pixel Kedua</i>
<i>Pixel C</i>	Tabel	Kolom	Baris	<i>Pixel Ketiga</i>

Proses dekripsi akan mengembalikan *cipherimage* menjadi *plainimage*. Angka tambahan 254 dan 255 jika ada tidak dipakai (dihilangkan) sehingga akan mendapatkan *plainimage* yang asli. Berikut ini contoh enkripsi dan dekripsi menggunakan Algoritma *3D Playfair* pada citra *grayscale*:

Kunci : ÖËÉ¾

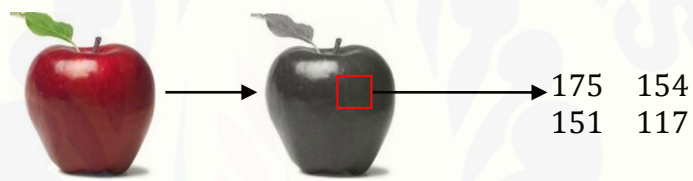
ÖËÉ¾ dikonversi kedalam bentuk *decimal* yaitu Ö menjadi 210, Ë menjadi 203, É menjadi 201, ¾ menjadi 190. Kunci teks yang sudah dikonversi kedalam bentuk *decimal* disusun menjadi empat tabel berukuran 8×8 ditunjukkan oleh Gambar 2.2. Empat tabel berukuran 8×8 dijadikan sebagai acuan untuk proses enkripsi dan dekripsi.

Tabel 1								Tabel 2							
190	201	203	210	0	1	2	3	60	61	62	63	64	65	66	67
4	5	6	7	8	9	10	11	68	69	70	71	72	73	74	75
12	13	14	15	16	17	18	19	76	77	78	79	80	81	82	83
20	21	22	23	24	25	26	27	84	85	86	87	88	89	90	91
28	29	30	31	32	33	34	35	92	93	94	95	96	97	98	99
36	37	38	39	40	41	42	43	100	101	102	103	104	105	106	107
44	45	46	47	48	49	50	51	108	109	110	111	112	113	114	115
52	53	54	55	56	57	58	59	116	117	118	119	120	121	122	123

Tabel 3								Tabel 4							
124	125	126	127	128	129	130	131	188	189	191	192	193	194	195	196
132	133	134	135	136	137	138	139	197	198	199	200	202	204	205	206
140	141	142	143	144	145	146	147	207	208	209	211	212	213	214	215
148	149	150	151	152	153	154	155	216	217	218	219	220	221	222	223
156	157	158	159	160	161	162	163	224	225	226	227	228	229	230	231
164	165	166	167	168	169	170	171	232	233	234	235	236	237	238	239
172	173	174	175	176	177	178	179	240	241	242	243	244	245	246	247
180	181	182	183	184	185	186	187	248	249	250	251	252	253	254	255

Gambar 2.2 Empat Tabel Kunci 3D Playfair Ukuran 8 × 8

Plainimage diambil sampel *pixel* secara random pada citra apel RGB yang dikonversi menjadi *grayscale* ditunjukkan oleh Gambar 2.3.



Gambar 2.3 Citra Apel (236 × 257) dengan Sampel Nilai *Pixel Grayscale*

Proses enkripsi sampel *pixel* citra apel dari Gambar 2.3.

Plainimage: 175 154 151 117

Plainimage dipecah menjadi *trigraphs* : (175 154 151), (117 254 255)

Proses enkripsi citra *grayscale* menggunakan 3D Playfair ditunjukkan oleh Tabel 2.3 sampai 2.4.

Tabel 2.3 Proses Enkripsi Citra *Grayscale (Trigraphs-1)*

<i>Trigraphs Plainimage</i>	<i>Trigraphs Plainimage</i>			<i>Trigraphs Cipherimage</i>
	175	154	151	
175	Baris 7	Kolom 7	Tabel 3	178
154	Tabel 3	Baris 4	Kolom 4	151
151	Kolom 4	Tabel 3	Baris 4	151

Tabel 2.4 Proses Enkripsi Citra *Grayscale* (*Trigraphs-2*)

<i>Trigraphs Plainimage</i>	<i>Trigraphs Plainimage</i>			<i>Trigraphs Cipherimage</i>
	117	254	255	
117	Baris 8	Kolom 7	Tabel 4	254
254	Tabel 2	Baris 8	Kolom 8	123
255	Kolom 2	Tabel 4	Baris 8	249

Sehingga diperoleh :

Plainimage : 175 154 151 117

Cipherimage : 178 151 151 254 123 249

Proses dekripsi sampel *pixel* citra apel Gambar 2.3.

Cipherimage : 178 151 151 254 123 249

Plainimage dipecah menjadi *trigraphs* : (178 151 151), (254 123 249)

Proses dekripsi citra *grayscale* menggunakan *3D Playfair* ditunjukkan oleh Tabel 2.5 sampai 2.6.

Tabel 2.5 Proses Dekripsi Citra *Grayscale* (*Trigraphs-1*)

<i>Trigraphs Cipherimage</i>	<i>Trigraphs Cipherimage</i>			<i>Trigraphs Plainimage</i>
	178	151	151	
178	Baris 7	Tabel 3	Kolom 4	175
151	Kolom 7	Baris 4	Tabel 3	154
151	Tabel 3	Kolom 4	Baris 4	151

Tabel 2.6 Proses Dekripsi Citra *Grayscale* (*Trigraphs-2*)

<i>Trigraphs Cipherimage</i>	<i>Trigraphs Cipherimage</i>			<i>Trigraphs Plainimage</i>
	254	123	249	
254	Baris 8	Tabel 2	Kolom 2	117
123	Kolom 7	Baris 8	Tabel 4	254
249	Tabel 4	Kolom 8	Baris 8	255

Sehingga diperoleh :

Cipherimage : 178 151 151 254 123 249

Plainimage : 175 154 151 117 254 255 dihapus 254 dan 255 menjadi
175 154 151 117

2.4 Analisis Keamanan

2.4.1 Number of Pixels Change Rate (NPCR)

Number of Pixels Change Rate (NPCR) adalah persentase perbandingan nilai total perubahan *pixel* antara *plainimage* dengan *cipherimage*. Menurut Wu (2011), Semakin besar nilai NPCR pada proses enkripsi citra RGB dan *grayscale*, maka enkripsi semakin baik. Jika semakin kecil nilai NPCR pada proses dekripsi citra RGB dan *grayscale*, maka citra hasil dekripsi akan semakin mirip dengan citra aslinya atau dapat kembali ke citra awal. Nilai NPCR untuk enkripsi citra menggunakan formula seperti pada persamaan 2.2.

$$NPCR = \left(\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^o \frac{d_{i,j,k}}{T} \right) \times 100\% \quad (2.2)$$

Dengan syarat sebagai berikut :

$$d_{i,j,k} = \begin{cases} 0, & \text{Jika } C_1(i, j, k) = C_2(i, j, k) \\ 1, & \text{Jika } C_1(i, j, k) \neq C_2(i, j, k) \end{cases}$$

Keterangan :

$d_{i,j,k}$: Banyaknya perbedaan *pixel*

$C_1(i, j, k)$: *Plainimage*

$C_2(i, j, k)$: *Cipherimage*

i : Baris

j : Kolom

k : *Layer*

m, n, o : Banyaknya dari baris, kolom, dan *layer*

T : Jumlah total *pixel*

2.4.2 *Unified Averaged Changed Intensity* (UACI)

Unified Averaged Changed Intensity (UACI) adalah persentase nilai *pixel* yang diperoleh melalui selisih nilai *pixel plainimage* dengan *cipherimage*. Menurut Wu (2011), Semakin besar nilai UACI pada proses enkripsi citra RGB dan *grayscale*, maka enkripsi semakin baik. Jika semakin kecil nilai UACI pada proses dekripsi citra RGB dan *grayscale*, maka citra hasil dekripsi akan semakin mirip dengan citra aslinya atau dapat kembali ke citra awal. Nilai UACI untuk enkripsi citra menggunakan formula seperti pada persamaan 2.3.

$$UACI = \left(\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^o \frac{|C_1(i,j,k) - C_2(i,j,k)|}{F.T} \right) \times 100\% \quad (2.3)$$

Keterangan :

$C_1(i,j,k)$: *Plainimage*

$C_2(i,j,k)$: *Cipherimage*

i : Baris

j : Kolom

k : *Layer*

m, n, o : Banyaknya dari baris, kolom, dan *layer*

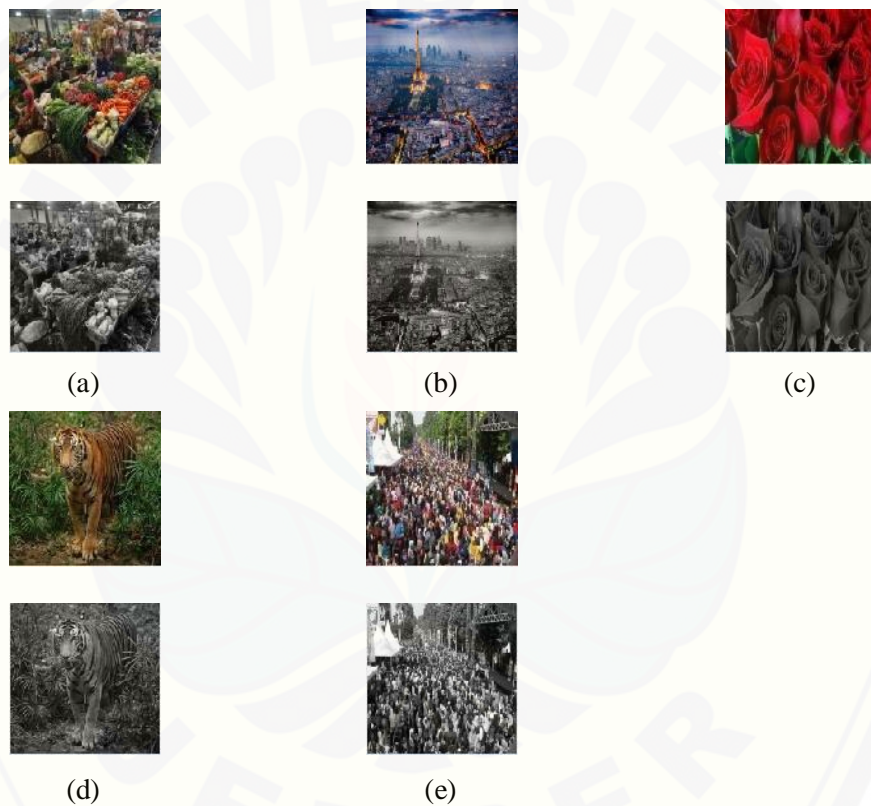
T : Jumlah total *pixel*

F : Nilai *pixel* terbesar pada citra sebesar 255

BAB 3 METODE PENELITIAN

3.1 Data Penelitian

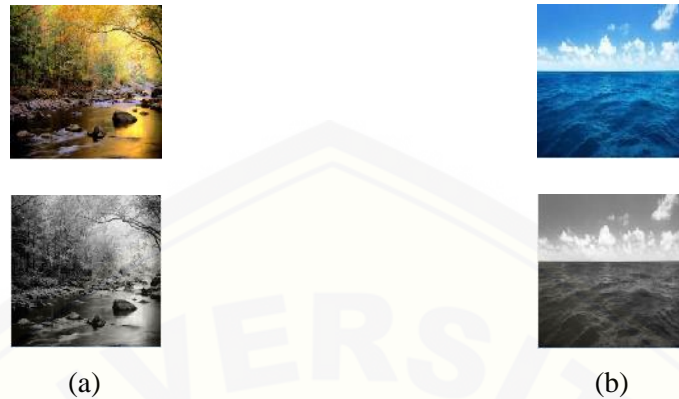
Data yang digunakan dalam penelitian ini adalah *plainimage* menggunakan gambar (citra) dan kunci (*key*) menggunakan gambar (citra) yang berbeda dengan *plainimage*. Citra yang digunakan sebagai *plainimage* ditunjukkan oleh Gambar 3.1.



- (a) Citra Pasar RGB (atas) Dan *Grayscale* (bawah) (259×195);
- (b) Citra Kota RGB (atas) Dan *Grayscale* (bawah) (2500×1600);
- (c) Citra Mawar RGB (atas) Dan *Grayscale* (bawah) (1000×500);
- (d) Citra Harimau RGB (atas) Dan *Grayscale* (bawah) (400×300);
- (e) Citra Orang RGB (atas) Dan *Grayscale* (bawah) (830×466);

Gambar 3.1 *Plainimage*

Citra yang digunakan sebagai kunci ditunjukkan oleh Gambar 3.2.



(a) Citra Sungai RGB (atas) Dan *Grayscale* (bawah) (1024×768);

(b) Citra Laut RGB (atas) Dan *Grayscale* (bawah) (318×159);

Gambar 3.2 Kunci Citra

3.2 Langkah-langkah Penelitian

Langkah-langkah dalam penelitian ini adalah sebagai berikut:

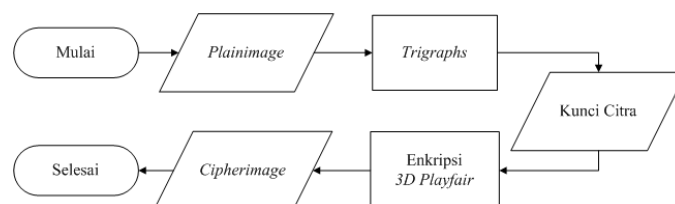
a. Studi Literatur

Peneliti mengumpulkan literatur yang berkaitan dengan Algoritma *3D Playfair* dan perbandingan citra RGB dengan *grayscale*.

b. Perancangan Pengkodean *Image* untuk Citra RGB dan *Grayscale* Menggunakan Algoritma *3D Playfair*

1) Proses enkripsi *plainimage* menggunakan Algoritma *3D Playfair*

Data yang dienkripsi berupa *plainimage* dengan kunci citra. Peneliti melakukan proses enkripsi *plainimage* menggunakan *software* MATLAB R2016b. Langkah-langkah proses enkripsi *plainimage* menggunakan Algoritma *3D Playfair* ditunjukkan oleh Gambar 3.3.



Gambar 3.3 Proses Enkripsi *Plainimage*

Adapun langkah–langkah proses enkripsi *plainimage* menggunakan Algoritma *3D Playfair* adalah sebagai berikut:

- Masukkan *plainimage* berupa citra RGB atau *grayscale*.
- Pecah *plainimage* menjadi *trigraphs*. *Trigraphs* dimasukkan kedalam tabel enkripsi *3D Playfair*. Tabel enkripsi untuk citra *grayscale* seperti pada Subbab 2.4 Tabel 2.3. Sedangkan, Tabel enkripsi untuk citra RGB seperti pada Tabel 3.1. Adapun proses *plainimage* menjadi *trigraphs* adalah sebagai berikut.

Plainimage citra *grayscale* berupa matriks $m \times n$ ditunjukkan oleh Gambar 3.4.

$$Gr = \begin{bmatrix} Gr_{11} & Gr_{12} & \dots & \dots & Gr_{1n} \\ Gr_{21} & Gr_{22} & \dots & \dots & Gr_{2n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ Gr_{m1} & Gr_{m2} & \dots & \dots & Gr_{mn} \end{bmatrix}$$

Gambar 3.4 *Plainimage* Citra *Grayscale* Matriks $m \times n$

Trigraphs citra *grayscale* :

$$(Gr_{11} \ Gr_{12} \ Gr_{13}), (Gr_{14} \ Gr_{15} \ Gr_{16}), (Gr_{17} \ Gr_{1n} \ Gr_{21}), \dots, \\ (Gr_{m1} \ Gr_{m2} \ Gr_{m3}), (Gr_{m4} \ Gr_{m5} \ Gr_{mn})$$

Keterangan :

Gr : Citra *grayscale* berupa matriks $m \times n$

Gr_{11} : Nilai *pixel* citra *grayscale* baris 1 kolom 1

Plainimage citra RGB berupa matriks $m \times n$ ditunjukkan oleh Gambar 3.5.

$$R = \begin{bmatrix} R_{11} & R_{12} & \dots & \dots & R_{1n} \\ R_{21} & R_{22} & \dots & \dots & R_{2n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ R_{m1} & R_{m2} & \dots & \dots & R_{mn} \end{bmatrix}$$

$$G = \begin{bmatrix} G_{11} & G_{12} & \dots & \dots & G_{1n} \\ G_{21} & G_{22} & \dots & \dots & G_{2n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ G_{m1} & G_{m2} & \dots & \dots & G_{mn} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & B_{12} & \dots & \dots & B_{1n} \\ B_{21} & B_{22} & \dots & \dots & B_{2n} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ B_{m1} & B_{m2} & \dots & \dots & B_{mn} \end{bmatrix}$$

Gambar 3.5 Plainimage Citra RGB Matriks $m \times n$

Trigraphs citra RGB :

$$(R_{11} \ G_{m1} \ B_{1n}), (R_{12} \ G_{51} \ B_{15}), (R_{13} \ G_{41} \ B_{14}), \dots,$$

$$(R_{1n} \ G_{11} \ B_{11}), (R_{21} \ G_{m2} \ B_{2n})$$

Tabel 3.1 Proses Enkripsi Citra RGB Menggunakan 3D Playfair

Trigraphs Plainimage	Trigraphs Plainimage			Trigraphs Cipherimage
	R_{11}	G_{m1}	B_{1n}	
R_{11}	Baris	Kolom	Tabel B	B_{1n}
G_{m1}	Tabel R	Baris	Kolom	R_{11}
B_{1n}	Kolom	Tabel G	Baris	G_{m1}

Keterangan :

- R : Citra RGB pada layer red berupa matriks $m \times n$
- G : Citra RGB pada layer green berupa matriks $m \times n$
- B : Citra RGB pada layer blue berupa matriks $m \times n$
- R_{11} : Nilai pixel citra RGB pada layer red baris 1 kolom 1
- G_{m1} : Nilai pixel citra RGB pada layer green baris m kolom 1
- B_{1n} : Nilai pixel citra RGB pada layer blue baris 1 kolom n

c) Kunci citra disesuaikan dengan *plainimage*. Jika *plainimage* menggunakan citra *grayscale* maka kunci citra menggunakan citra *grayscale*. Jika *plainimage* menggunakan citra RGB maka kunci citra menggunakan citra RGB. Kunci citra dibentuk menjadi empat tabel berukuran 8×8 yang berisi 256 *pixel* (0-255), *pixel* yang diisikan ke dalam empat tabel kunci tidak berulang. Pengisian *pixel* ke dalam empat tabel kunci diurutkan berdasarkan frekuensi secara *ascending* dimulai dari Tabel 1, Tabel 2, Tabel 3, Tabel 4. Empat tabel kunci citra 8×8 ditunjukkan oleh Gambar 3.6 sampai 3.9.

Tabel 1				Tabel 2			
FGr_{11}	FGr_{12}	...	FGr_{18}	FGr_{11}	FGr_{12}	...	FGr_{18}
FGr_{21}	FGr_{22}	...	FGr_{28}	FGr_{21}	FGr_{22}	...	FGr_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FGr_{81}	FGr_{81}	...	FGr_{88}	FGr_{81}	FGr_{81}	...	FGr_{88}

Tabel 3				Tabel 4			
FGr_{11}	FGr_{12}	...	FGr_{18}	FGr_{11}	FGr_{12}	...	FGr_{18}
FGr_{21}	FGr_{22}	...	FGr_{28}	FGr_{21}	FGr_{22}	...	FGr_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FGr_{81}	FGr_{81}	...	FGr_{88}	FGr_{81}	FGr_{81}	...	FGr_{88}

Gambar 3.6 Empat Tabel Kunci Citra *Grayscale 3D Playfair 8 x 8*

Keterangan :

$FGr_{11} \dots FGr_{88}$: Nilai *pixel* pada kunci citra *grayscale* yang diurutkan berdasarkan frekuensi secara *ascending*

Tabel 1				Tabel 2			
FR_{11}	FR_{12}	...	FR_{18}	FR_{11}	FR_{12}	...	FR_{18}
FR_{21}	FR_{22}	...	FR_{28}	FR_{21}	FR_{22}	...	FR_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FR_{81}	FR_{81}	...	FR_{88}	FR_{81}	FR_{81}	...	FR_{88}

Tabel 3				Tabel 4			
FR_{11}	FR_{12}	...	FR_{18}	FR_{11}	FR_{12}	...	FR_{18}
FR_{21}	FR_{22}	...	FR_{28}	FR_{21}	FR_{22}	...	FR_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FR_{81}	FR_{81}	...	FR_{88}	FR_{81}	FR_{81}	...	FR_{88}

Gambar 3.7 Empat Tabel Kunci Citra RGB (Red) 3D Playfair 8 x 8

Tabel 1				Tabel 2			
FG_{11}	FG_{12}	...	FG_{18}	FG_{11}	FG_{12}	...	FG_{18}
FG_{21}	FG_{22}	...	FG_{28}	FG_{21}	FG_{22}	...	FG_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FG_{81}	FG_{81}	...	FG_{88}	FG_{81}	FG_{81}	...	FG_{88}

Tabel 3				Tabel 4			
FG_{11}	FG_{12}	...	FG_{18}	FG_{11}	FG_{12}	...	FG_{18}
FG_{21}	FG_{22}	...	FG_{28}	FG_{21}	FG_{22}	...	FG_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FG_{81}	FG_{81}	...	FG_{88}	FG_{81}	FG_{81}	...	FG_{88}

Gambar 3.8 Empat Tabel Kunci Citra RGB (Green) 3D Playfair 8 x 8

Tabel 1				Tabel 2			
FB_{11}	FB_{12}	...	FB_{18}	FB_{11}	FB_{12}	...	FB_{18}
FB_{21}	FB_{22}	...	FB_{28}	FB_{21}	FB_{22}	...	FB_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FB_{81}	FB_{81}	...	FB_{88}	FB_{81}	FB_{81}	...	FB_{88}

Tabel 3				Tabel 4			
FB_{11}	FB_{12}	...	FB_{18}	FB_{11}	FB_{12}	...	FB_{18}
FB_{21}	FB_{22}	...	FB_{28}	FB_{21}	FB_{22}	...	FB_{28}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
FB_{81}	FB_{81}	...	FB_{88}	FB_{81}	FB_{81}	...	FB_{88}

Gambar 3.9 Empat Tabel Kunci Citra RGB (*Blue*) *3D Playfair* 8 x 8

Keterangan :

$FR_{11} \dots FR_{88}$: Nilai *pixel* pada kunci citra RGB (*Red*) yang diurutkan berdasarkan frekuensi secara *ascending*

$FG_{11} \dots FG_{88}$: Nilai *pixel* pada kunci citra RGB (*Green*) yang diurutkan berdasarkan frekuensi secara *ascending*

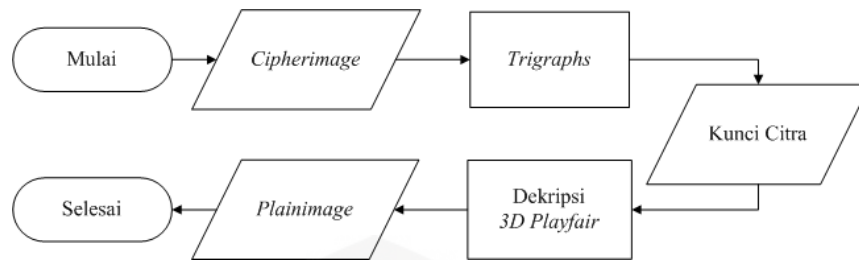
$FB_{11} \dots FB_{88}$: Nilai *pixel* pada kunci citra RGB (*Blue*) yang diurutkan berdasarkan frekuensi secara *ascending*

d) Cara melakukan enkripsi menggunakan *3D Playfair* seperti pada contoh Subbab 2.4 Tabel 2.3 sampai 2.4.

e) *Output* pada proses enkripsi *plainimage* menghasilkan *cipherimage*.

2) Proses dekripsi *cipherimage* menggunakan *3D Playfair*

Data yang didekripsi berupa *cipherimage* dengan kunci citra yang sama pada proses enkripsi. Peneliti melakukan proses dekripsi *cipherimage* menggunakan *software* MATLAB R2016b. Langkah-langkah proses dekripsi *cipherimage* menggunakan Algoritma *3D Playfair* ditunjukkan oleh Gambar 3.10.



Gambar 3.10 Proses Dekripsi *Cipherimage*

Langkah-langkah yang digunakan pada proses dekripsi *cipherimage* dengan Algoritma *3D Playfair* menggunakan langkah-langkah yang sama pada saat proses enkripsi tetapi tabel *3D Playfair* proses dekripsi berbeda dengan tabel *3D Playfair* proses enkripsi. Tabel dekripsi untuk citra *grayscale* seperti pada Subbab 2.4 Tabel 2.5. Sedangkan, Tabel dekripsi untuk citra RGB seperti pada Tabel 3.2. Proses dekripsi menggunakan *3D Playfair* seperti pada contoh Subbab 2.4 Tabel 2.5 sampai 2.6.

Tabel 3.2 Proses Dekripsi Citra RGB Menggunakan *3D Playfair*

<i>Trigraphs Cipherimage</i>	<i>Trigraphs Cipherimage</i>			<i>Trigraphs Plainimage</i>
	B_{1n}	R_{11}	G_{m1}	
B_{1n}	Baris	Tabel R	Kolom	R_{11}
R_{11}	Kolom	Baris	Tabel G	G_{m1}
G_{m1}	Tabel B	Kolom	Baris	B_{1n}

Keterangan :

R : Citra RGB pada *layer red* berupa matriks $m \times n$

G : Citra RGB pada *layer green* berupa matriks $m \times n$

B : Citra RGB pada *layer blue* berupa matriks $m \times n$

R_{11} : Nilai *pixel* citra RGB pada *layer red* baris 1 kolom 1

G_{m1} : Nilai *pixel* citra RGB pada *layer green* baris m kolom 1

B_{1n} : Nilai *pixel* citra RGB pada *layer blue* baris 1 kolom n

c. Pembuatan Program Pengkodean *Image* untuk Citra RGB dan *Grayscale* Menggunakan Algoritma *3D Playfair*

Tahap pembuatan program proses enkripsi dan dekripsi pada Algoritma *3D Playfair* pengkodean *image* untuk citra RGB dan *grayscale* menggunakan *software MatLab 2016b*.

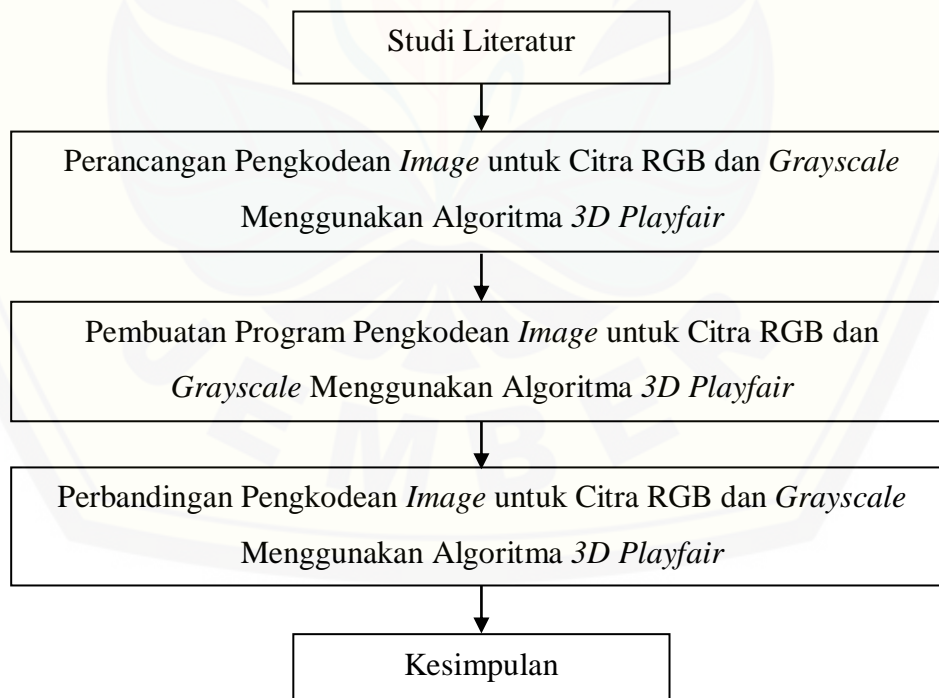
d. Perbandingan Pengkodean *Image* untuk Citra RGB dan *Grayscale* Menggunakan Algoritma *3D Playfair*

Tahap perbandingan pengkodean *image* untuk citra RGB dan *grayscale* menggunakan Algoritma *3D Playfair* menggunakan NPCR dan UACI.

e. Kesimpulan

Tahap kesimpulan merupakan tahap akhir dalam penelitian ini dimana menyimpulkan hasil keamanan citra RGB dan *grayscale* menggunakan Algoritma *3D Playfair* pada citra yang telah dienkripsi dan didekripsi.

Berikut diagram alir penelitian pada Gambar 3.11.



Gambar 3.11 Diagram Alir Penelitian

BAB 5 PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, didapat beberapa kesimpulan sebagai berikut:

- a. Empat tabel kunci 8×8 pada citra RGB terdapat disetiap *layer*. Jika Inputan *plainimage* citra RGB maka inputan kunci menggunakan citra RGB. Pembentukan *trigraphs* citra RGB diperoleh dari 3 *layer* yaitu *layer red*, *layer green*, *layer blue*. *Trigraphs* citra RGB untuk *layer red* diambil *pixel* dari kiri ke kanan, *layer green* diambil *pixel* dari bawah ke atas, *layer blue* diambil *pixel* dari kanan ke kiri.
- b. Empat tabel kunci 8×8 pada citra *grayscale* hanya satu *layer*. Jika inputan *plainimage* citra *grayscale* maka inputan kunci menggunakan citra *grayscale*, tetapi bisa menggunakan citra RGB dengan mengkonversi menjadi citra *grayscale*. Pembentukan *trigraphs* citra *grayscale* diperoleh dari satu *layer*. *Trigraphs* citra *grayscale* diambil *pixel* dari kiri ke kanan.
- c. Secara analitik (NPCR dan UACI) citra RGB lebih baik daripada citra *grayscale*, karena citra RGB mempunyai nilai NPCR dan UACI yang lebih besar daripada citra *grayscale*. Secara visual proses enkripsi citra RGB dan citra *grayscale* menggunakan Algoritma *3D Playfair* sulit diinterpretasikan artinya menghasilkan *chiperimage* yang terlihat acak. Proses dekripsi citra RGB dan citra *grayscale* menggunakan Algoritma *3D Playfair* dapat mengembalikan *cipherimage* seperti citra aslinya.

5.2 Saran

Saran untuk penelitian selanjutnya dapat membandingkan citra RGB dan citra *grayscale* menggunakan algoritma kriptografi lainnya. Mengkodekan *image* untuk citra *grayscale* menggunakan algoritma kriptografi. Mengkodekan pesan kedalam *image* untuk citra RGB menggunakan algoritma steganografi. Pembentukan *trigraphs* dalam Algoritma *3D Playfair* dengan pengambilan yang bervariasi.

DAFTAR PUSTAKA

- Alfred, J. M., C. O. Paul, and A. V. Scott. 1997. *Handbook of Applied Cryptography*. Florida, USA: CRC Press LLC.
- Fatta, H. A. 2007. Konversi Format Citra RGB Ke Format *Grayscale* Menggunakan Visual Basic. *Seminar Nasional Teknologi 2007 (SNT 2007)*. Yogyakarta: STMIK AMIKOM.
- Ibrahim, R. N. 2012. Kriptografi Algoritma *DES, AES/Rijndael, Blowfish* Untuk Keamanan Citra Digital Dengan Menggunakan Metode *Discrete Wavelet Transformation (DWT)*. *Jurnal Computech dan Bisnis*. 6(2): 82-95.
- Munir, R. 2006. *Kriptografi*. Bandung: Informatika.
- Noor, S. C. 2011. Mengubah Citra Berwarna Menjadi *Grayscale* Dan Citra Biner. *Jurnal Teknologi Informasi Dinamik*. 16(1): 14-19.
- Onno, W. P., A. A. Wahyudi. 2000. *Mengenal eCommerce*. Jakarta: Elex Media Komputindo.
- Setiadi, D. R. I. M., E. H. Rachmawanto, dan C. A. Sari. 2017. Implementasi One Time Pad Kriptografi Pada Gambar *Grayscale* Dan Gambar Berwarna. *Seminar Nasional Multi Disiplin Ilmu Dan Call For Papers Unisbank Ke 3 (SENDI_U 3)*. Semarang: Universitas Dian Nuswantoro Semarang.
- Stallings, W. 1999. *Cryptography and Network Security Principles and Practice second edition*. New Jersey, USA: Prentice Hall.
- Sukmawati, R. A. 2019. Perbandingan Playfair Cipher Dengan 3D Playfair Cipher Pada Pengaman Citra. *Skripsi*. Jember: Program Matematika Fakultas Matematika Dan Ilmu Pengetahuan Alam Universitas Jember.
- Wu, Y., J. P. Noonan, dan S. Agaian. 2011. NPCR and UACI Randomness For Image Encryption. *Journal of Selected Areas in Telecommunications (JSAT)*, pp. 31-38.

LAMPIRAN

LAMPIRAN A. Skrip Program Pembentukan Empat Tabel Kunci Citra

```
clc;clear all;
%KUNCI GRAYSCALE-----
%1.input citra
a=input('Masukkan Kunci Citra: ','s');

%2.Menampilkan pixel
aa1=imread(a);
[mk nk ok]=size(aa1);
if ok==3
    aak=rgb2gray(aa1);
else
    aak=aa1;
end

%3.Menjadi 1xn
kun11=uint8(aak);
[m n]=size(kun11);
a = kun11(1,:);
for i=1:m-1
a(1,i*n+1:(i+1)*n) = kun11(i+1,:)
end
kun11=a;
pixel_length=length(kun11);

%4.Mengelompokkan pixel
for j=1:pixel_length
a=1;
for i=1:pixel_length-j
if kun11(1,j)==kun11(1,i+j)
a=a+1;
kun11(2,j)=a;
kun11(1,i+j)=0;
end
end
end

for p=1:length(kun11)
if kun11(2,p)== 0
kun11(2,p)=1;
end
end

for p=1:length(kun11)
if kun11(1,p)~= 0
kun11_new(1,p)=kun11(1,p);
kun11_new(2,p)=kun11(2,p);
end
end

matbaru=0;
for p=1:length(kun11)
```

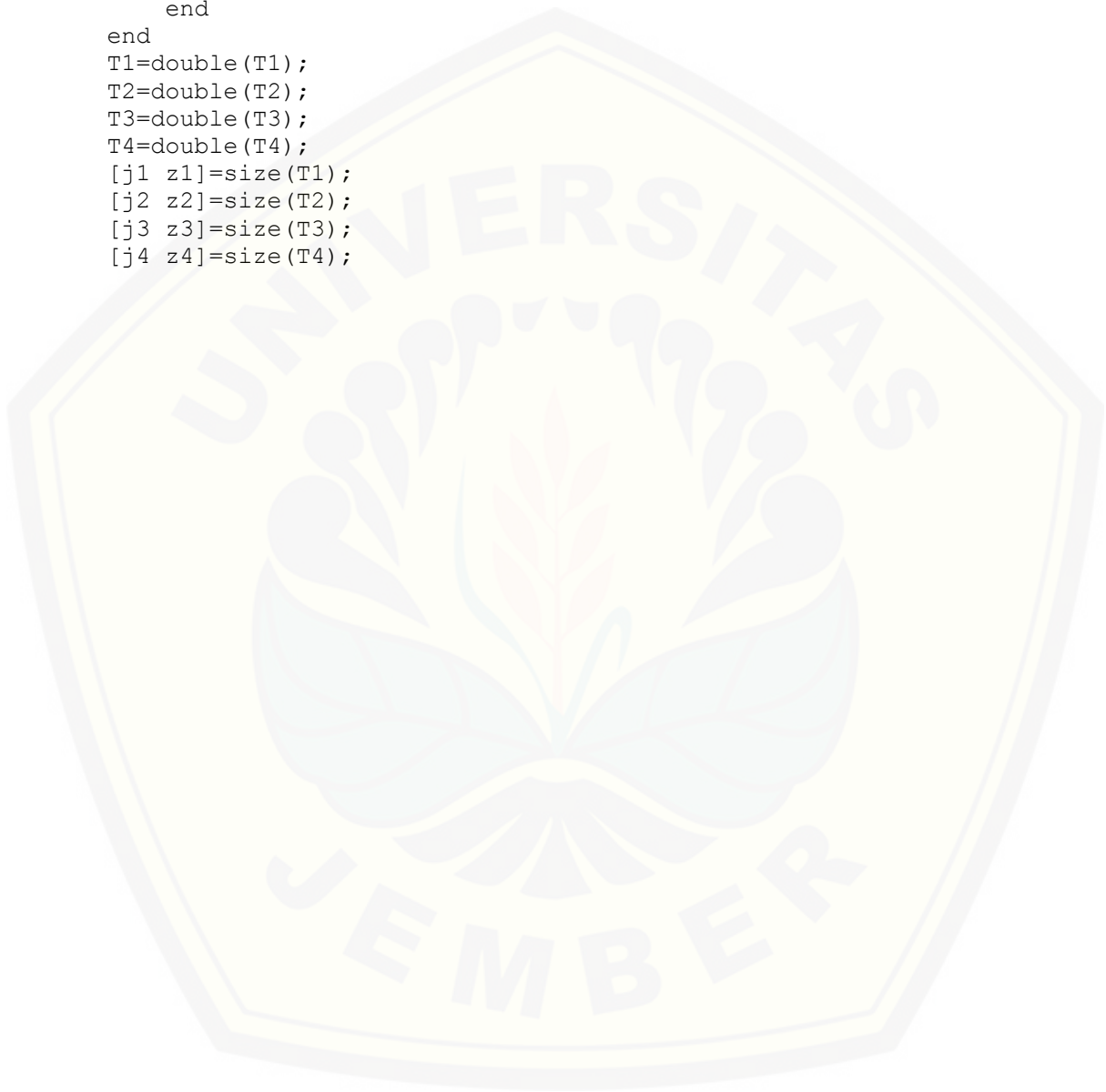
```
if kun11(1,p)~= 0
matbaru=matbaru+1;
end
end
matbaru_tem=zeros(2,matbaru);
c=1;
for p=1:length(kun11_new)
if kun11_new(1,p)~= 0
matbaru_tem(:,c)=kun11_new(:,p);
c=c+1;
end
end
matbaru_tem;

%5.pixel diurutkan secara ascending
temp = [1;1];
[m n] = size(matbaru_tem);
for i=1:n-1
for j=i+1:n
if matbaru_tem(2,i)< matbaru_tem(2,j)
temp = matbaru_tem(:,i);
matbaru_tem(:,i) = matbaru_tem(:,j);
matbaru_tem(:,j) = temp;
end
if matbaru_tem(2,i) == matbaru_tem(2,j)
if matbaru_tem(1,i) > matbaru_tem(1,j)
temp = matbaru_tem(:,i);
matbaru_tem(:,i) = matbaru_tem(:,j);
matbaru_tem(:,j) = temp;
end
end
end
end
matbaru_tem;

%6. Membuat pixel 0-255
tabel1 = matbaru_tem(1,:);
[m n] = size(tabel1);
for i = 0:255
a = ismember(i,tabel1);
if a == 0
tabel1(n+1)= i;
[m n] = size(tabel1);
end
end
end
tabel1;

%7.Membuat 4 Tabel
k= 0;
for i=1:4
for j=1:8
for z=1:8
k=k+1;
if i==1
T1(j,z)=tabel1(k);
elseif i==2
```

```
        T2(j,z)=tabel1(k);
    elseif i==3
        T3(j,z)=tabel1(k);
    elseif i==4
        T4(j,z)=tabel1(k);
    end
    end
end
end
T1=double(T1);
T2=double(T2);
T3=double(T3);
T4=double(T4);
[j1 z1]=size(T1);
[j2 z2]=size(T2);
[j3 z3]=size(T3);
[j4 z4]=size(T4);
```



Lampiran B. Skrip Program Enkripsi dan Dekripsi 3D Playfair

a. Skrip Program Enkripsi 3D Playfair

```

%Enkripsi Plainimage-----
%1. Inputan
aa=input('Masukkan Citra: ','s');

%2. Menampilkan pixel
aa2=imread(aa);
[ma na oa]=size(aa2);
if oa==3
    aa3=rgb2gray(aa2);
else
    aa3=aa2;
end

%3.mengubah ukuran mxn menjadi 1xn
plain2=uint8(aa3);
[map nap oap]=size(plain2);
a = plain2(1,:);
for i=1:map-1
a(1,i*nap+1:(i+1)*nap) = plain2(i+1,:);
end
plain2=a;

%4.Membuat trigraph (3 pixel)
[m1 n1]=size(plain2);
pixel_length1=[m1*n1];
sisa=mod(m1*n1,3);
if sisa==0
    nn=plain2(1:m1*n1);
elseif sisa==1
    nn=[plain2(1:m1*n1) 254 255];
else
    nn=[plain2(1:m1*n1) 255];
end
nn;
[m2 n2]=size(nn);
panjang=length(nn)/3;
awl=1;
ahr=3;
nn4=4;
for j11=1:panjang
    simpan=zeros(m2,n2);
    B=nn(awl:ahr);

%5. Menampilkan posisi pixel
    B1=B(1,1);
    a1=ismember(B1,T1);
    b1=ismember(B1,T2);
    c1=ismember(B1,T3);
    if a1==1
        TB1=1;
        a11=B1;
    elseif b1==1
        TB1=2;

```



```
        a12=B2;
    else
        TB2=4;
        a12=B2;
    end
    for k1=1:j1
        for l1=1:z1
            if T1(k1,l1)==a12
                baris2=k1;
                kolom2=l1;
                ambil=[baris2 kolom2];
            end
        end
    end
    for k2=1:j2
        for l2=1:z2
            if T2(k2,l2)==a12
                baris2=k2;
                kolom2=l2;
                ambil=[baris2 kolom2];
            end
        end
    end
    for k3=1:j3
        for l3=1:z3
            if T3(k3,l3)==a12
                baris2=k3;
                kolom2=l3;
                ambil=[baris2 kolom2];
            end
        end
    end
    for k4=1:j4
        for l4=1:z4
            if T4(k4,l4)==a12
                baris2=k4;
                kolom2=l4;
                ambil=[baris2 kolom2];
            end
        end
    end
    B3=B(1,3);
    a3=ismember(B3,T1);
    b3=ismember(B3,T2);
    c3=ismember(B3,T3);
    if a3==1
        TB3=1;
        a13=B3;
    elseif b3==1
        TB3=2;
        a13=B3;
    elseif c3==1
        TB3=3;
        a13=B3;
    else
        TB3=4;
```

```

        a13=B3;
    end
    for k1=1:j1
        for l1=1:z1
            if T1(k1,l1)==a13
                baris3=k1;
                kolom3=l1;
                ambil=[baris3 kolom3];
            end
        end
    end
    for k2=1:j2
        for l2=1:z2
            if T2(k2,l2)==a13
                baris3=k2;
                kolom3=l2;
                ambil=[baris3 kolom3];
            end
        end
    end
    for k3=1:j3
        for l3=1:z3
            if T3(k3,l3)==a13
                baris3=k3;
                kolom3=l3;
                ambil=[baris3 kolom3];
            end
        end
    end
    for k4=1:j4
        for l4=1:z4
            if T4(k4,l4)==a13
                baris3=k4;
                kolom3=l4;
                ambil=[baris3 kolom3];
            end
        end
    end
end

```

%6. Sesuaikan urutan baris kolom tabel sesuai tabel enkripsi agar %menghasilkan chiperimage

```

pix1=[baris1 kolom2 TB3];
pix11=[baris1 kolom2];
pix2=[baris2 kolom3 TB1];
pix12=[baris2 kolom3];
pix3=[baris3 kolom1 TB2];
pix13=[baris3 kolom1];
if TB3==1;
    T1 (baris1, kolom2);
    ci3=T1 (baris1, kolom2);
elseif TB3==2;
    T2 (baris1, kolom2);
    ci3=T2 (baris1, kolom2);
elseif TB3==3;
    T3 (baris1, kolom2);
    ci3=T3 (baris1, kolom2);

```

```

else
    T4 (baris1, kolom2);
    ci3=T4 (baris1, kolom2);
end
if    TB1==1;
    T1 (baris2, kolom3);
    ci1= T1 (baris2, kolom3);
elseif TB1==2;
    T2 (baris2, kolom3);
    ci1= T2 (baris2, kolom3);
elseif TB1==3;
    T3 (baris2, kolom3);
    ci1= T3 (baris2, kolom3);
else
    T4 (baris2, kolom3);
    ci1= T4 (baris2, kolom3);
end
if    TB2==1;
    T1 (baris3, kolom1);
    ci2=T1 (baris3, kolom1);
elseif TB2==2;
    T2 (baris3, kolom1);
    ci2=T2 (baris3, kolom1);
elseif TB2==3;
    T3 (baris3, kolom1);
    ci2=T3 (baris3, kolom1);
else
    T4 (baris3, kolom1);
    ci2=T4 (baris3, kolom1);
end
ci=[ci3 ci1 ci2];

%8. trigraphs chiperimage dibentuk menjadi matrik lxn
if j11==1
    simpan(1,1)=ci3;
    simpan(1,2)=ci1;
    simpan(1,3)=ci2;
    simpan1=simpan;
end
simpan1;
[m4 n4]=size(simpan1);
if j11==2
    simpan1(1,4)=ci3;
    simpan1(1,5)=ci1;
    simpan1(1,6)=ci2;
    simpan1;
end
simpan1;
if j11==3
    nn5=nn4+3;
    simpan1(1,nn5)=ci3;
    nn6=nn5+1;
    simpan1(1,nn6)=ci1;
    nn7=nn6+1;
    simpan1(1,nn7)=ci2;
    simpan1;

```

```

        nn4=nn7+1;
    end
    if jj1>3
        nn5=nn4;
        simpan1(1,nn5)=ci3;
        nn6=nn5+1;
        simpan1(1,nn6)=ci1;
        nn7=nn6+1;
        simpan1(1,nn7)=ci2;
        simpan1;
        nn4=nn7+1;
    end
    awl=ahr+1;
    ahr=awl+2;
end

%9.chiperimage 1xn dibentuk menjadi matrik dengan ukuran seperti
plainimage
simpan1;
[m5 n5]=size(simpan1);
m44=map;
n44=nap;
BB1=zeros(m44,n44);
[m44 n44]=size(BB1);
n4=1;
for jj=1:m44
    for ii=1:n44
        n4;
        if n4>n5
            break
        else
            n55=n4;
            BB1(jj,ii)=simpan1(1,n55);
            n4=n55+1;
        end
    end
end
end
BB1;
[m45 n45]=size(BB1);
BB111=uint8(BB1);
imshow(BB111);

```

b. Skrip Program Dekripsi 3D Playfair

```

%Dekripsi CIPHERimage-----
%1. Inputan
BB111;

%2. Menampilkan pixel
plain11=uint8(BB111);
[map nap oap]=size(plain11);

%3.mengubah ukuran mxn menjadi 1xn
a = plain11(1,:);
for i=1:map-1
    a(1,i*nap+1:(i+1)*nap) = plain11(i+1,:); end

```

```

%4.Membuat trigraph (3 pixel)
plain11=a;
[m1 n1]=size(plain11);
pixel_length1=[m1*n1];
sisamod=mod(m1*n1,3);
if sisamod==0
    nn=plain11(1:m1*n1);
elseif sisamod==1
    nn=[plain11(1:m1*n1) 254 255];
else
    nn=[plain11(1:m1*n1) 255];
end
nn;
[m2 n2]=size(nn);
panjang=length(nn)/3;
awl=1;
ahr=3;
nn4=4;
for j1=1:panjang
    simpan=zeros(m2,n2);
    B=nn(awl:ahr);

%5. Menampilkan posisi pixel
    B1=B(1,1);
    a1=ismember(B1,T1);
    b1=ismember(B1,T2);
    c1=ismember(B1,T3);
    if a1==1
        TB1=1;
        a11=B1;
    elseif b1==1
        TB1=2;
        a11=B1;
    elseif c1==1
        TB1=3;
        a11=B1;
    else
        TB1=4;
        a11=B1;
    end
    for k1=1:j1
        for l1=1:z1
            if T1(k1,l1)==a11
                baris1=k1;
                kolom1=l1;
                ambil=[baris1 kolom1];
            end
        end
    end
    for k2=1:j2
        for l2=1:z2
            if T2(k2,l2)==a11
                baris1=k2;
                kolom1=l2;
                ambil=[baris1 kolom1];
            end
        end
    end
end

```

```
        end
    end
    for k3=1:j3
        for l3=1:z3
            if T3(k3,l3)==a11
                baris1=k3;
                kolom1=l3;
                ambil=[baris1 kolom1];
            end
        end
    end
    for k4=1:j4
        for l4=1:z4
            if T4(k4,l4)==a11
                baris1=k4;
                kolom1=l4;
                ambil=[baris1 kolom1];
            end
        end
    end
    B2=B(1,2);
    a2=ismember(B2,T1);
    b2=ismember(B2,T2);
    c2=ismember(B2,T3);
    if a2==1
        TB2=1;
        a12=B2;
    elseif b2==1
        TB2=2;
        a12=B2;
    elseif c2==1
        TB2=3;
        a12=B2;
    else
        TB2=4;
        a12=B2;
    end
    for k1=1:j1
        for l1=1:z1
            if T1(k1,l1)==a12
                baris2=k1;
                kolom2=l1;
                ambil=[baris2 kolom2];
            end
        end
    end
    for k2=1:j2
        for l2=1:z2
            if T2(k2,l2)==a12
                baris2=k2;
                kolom2=l2;
                ambil=[baris2 kolom2];
            end
        end
    end
    for k3=1:j3
```

```
        for l3=1:z3
            if T3(k3,l3)==a12
                baris2=k3;
                kolom2=l3;
                ambil=[baris2 kolom2];
            end
        end
    end
end
for k4=1:j4
    for l4=1:z4
        if T4(k4,l4)==a12
            baris2=k4;
            kolom2=l4;
            ambil=[baris2 kolom2];
        end
    end
end
B3=B(1,3);
a3=ismember(B3,T1);
b3=ismember(B3,T2);
c3=ismember(B3,T3);
if a3==1
    TB3=1;
    a13=B3;
elseif b3==1
    TB3=2;
    a13=B3;
elseif c3==1
    TB3=3;
    a13=B3;
else
    TB3=4;
    a13=B3;
end
for k1=1:j1
    for l1=1:z1
        if T1(k1,l1)==a13
            baris3=k1;
            kolom3=l1;
            ambil=[baris3 kolom3];
        end
    end
end
for k2=1:j2
    for l2=1:z2
        if T2(k2,l2)==a13
            baris3=k2;
            kolom3=l2;
            ambil=[baris3 kolom3];
        end
    end
end
for k3=1:j3
    for l3=1:z3
        if T3(k3,l3)==a13
            baris3=k3;
```

```

        kolom3=13;
        ambil=[baris3 kolom3];
    end
end
end
for k4=1:j4
    for l4=1:z4
        if T4(k4,l4)==a13
            baris3=k4;
            kolom3=l4;
            ambil=[baris3 kolom3];
        end
    end
end
end
end

```

%6. Sesuaikan urutan baris kolom tabel sesuai tabel dekripsi agar menghasilkan plainimage

```

pix1=[baris1 kolom3 TB2];
pix11=[baris1 kolom3];
pix2=[baris2 kolom1 TB3];
pix12=[baris2 kolom1];
pix3=[baris3 kolom2 TB1];
pix13=[baris3 kolom2];
if TB2==1;
    T1 (baris1, kolom3);
    ci2=T1 (baris1, kolom3);
elseif TB2==2;
    T2 (baris1, kolom3);
    ci2=T2 (baris1, kolom3);
elseif TB2==3;
    T3 (baris1, kolom3);
    ci2=T3 (baris1, kolom3);
else
    T4 (baris1, kolom3);
    ci2=T4 (baris1, kolom3);
end
if TB3==1;
    T1 (baris2, kolom1);
    ci3= T1 (baris2, kolom1);
elseif TB3==2;
    T2 (baris2, kolom1);
    ci3= T2 (baris2, kolom1);
elseif TB3==3;
    T3 (baris2, kolom1);
    ci3= T3 (baris2, kolom1);
else
    T4 (baris2, kolom1);
    ci3= T4 (baris2, kolom1);
end
if TB1==1;
    T1 (baris3, kolom2);
    ci1=T1 (baris3, kolom2);
elseif TB1==2;
    T2 (baris3, kolom1);
    ci1=T2 (baris3, kolom2);
elseif TB1==3;

```



```

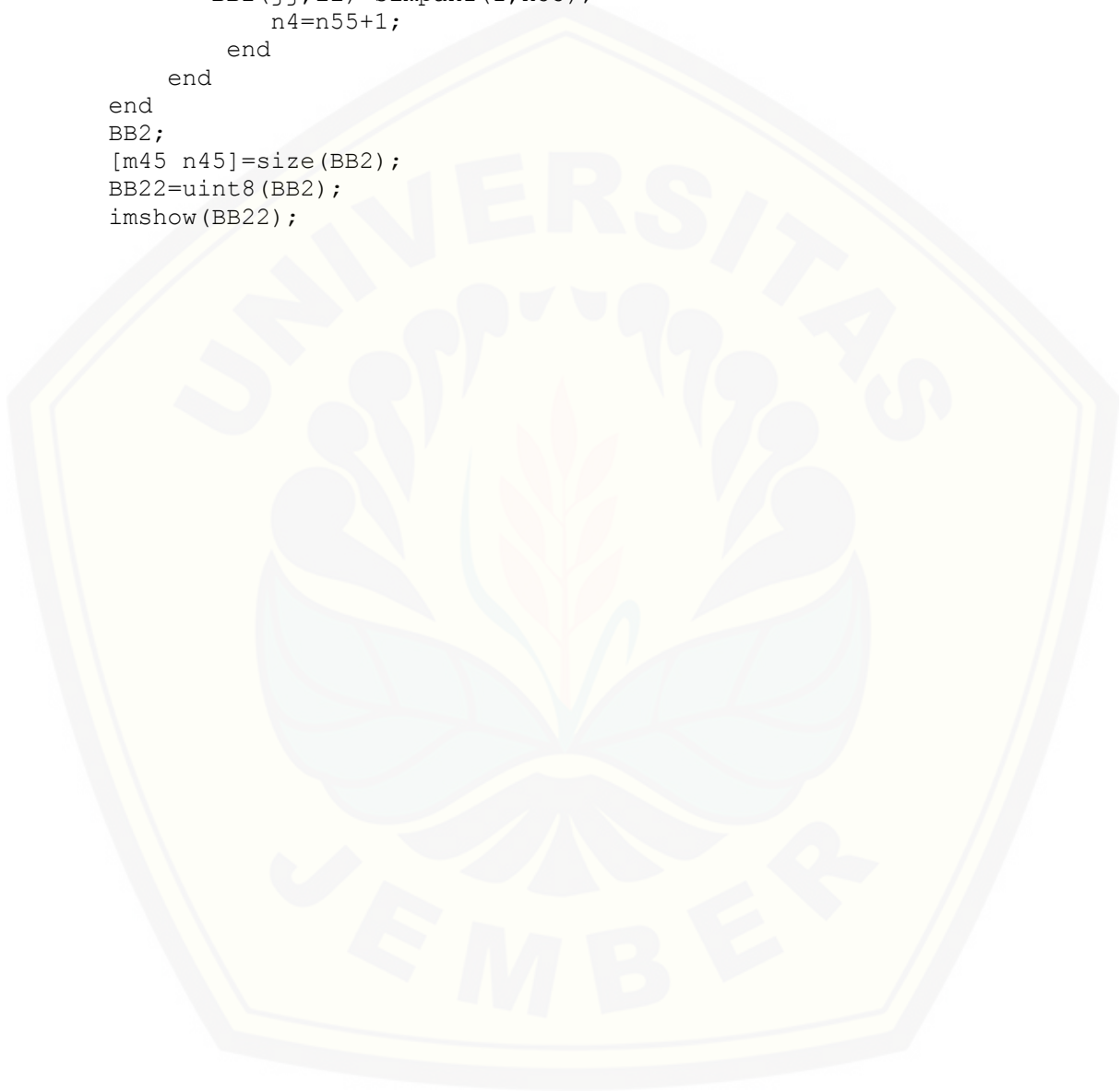
        T3 (baris3, kolom1);
        ci1=T3 (baris3, kolom2);
    else
        T4 (baris3, kolom1);
        ci1=T4 (baris3, kolom2);
    end

%8. trigraphs plainimage dibentuk menjadi matrik 1xn
ci=[ci2 ci3 ci1];
if j11==1
    simpan(1,1)=ci2;
    simpan(1,2)=ci3;
    simpan(1,3)=ci1;
    simpan1=simpan;
end
simpan1;
[m4 n4]=size(simpan1);
if j11==2
    simpan1(1,4)=ci2;
    simpan1(1,5)=ci3;
    simpan1(1,6)=ci1;
    simpan1;
end
simpan1;
if j11==3
    nn5=nn4+3;
    simpan1(1,nn5)=ci2;
    nn6=nn5+1;
    simpan1(1,nn6)=ci3;
    nn7=nn6+1;
    simpan1(1,nn7)=ci1;
    simpan1;
    nn4=nn7+1;
end
if j11>3
    nn5=nn4;
    simpan1(1,nn5)=ci2;
    nn6=nn5+1;
    simpan1(1,nn6)=ci3;
    nn7=nn6+1;
    simpan1(1,nn7)=ci1;
    simpan1;
    nn4=nn7+1;
end
    awl=ahr+1;
    ahr=awl+2;
end

%9.plainimage 1xn dibentuk menjadi matrik dengan ukuran citra asli
simpan1;
[m5 n5]=size(simpan1);
m44=map;
n44=nap;
BB2=zeros (m44,n44);
[m44 n44]=size(BB2);
n4=1;

```

```
for jj=1:m44
    for ii=1:n44
        n4;
        if n4>n5
            break
        else
            n55=n4;
            BB2(jj,ii)=simpan1(1,n55);
            n4=n55+1;
        end
    end
end
end
BB2;
[m45 n45]=size(BB2);
BB22=uint8(BB2);
imshow(BB22);
```



Lampiran C. Skrip Program Analisis Keamanan

a. NPCR Enkripsi

```
plain=double(aa3);
cipher=double(BB111);
[mp np op]=size(plain);
[mc nc oc]=size(cipher);
dij=plain-cipher;
dij(dij~=0)=1;
NPCR=(sum(sum(sum(dij)))/(mp*np*op))*100;
set(handles.text32,'string',num2str(NPCR));
```

b. NPCR Dekripsi

```
plain=double(aa3);
dekrip=double(BB22);
[mp np op]=size(plain);
[mc nc oc]=size(dekrip);
dij=plain-dekrip;
dij(dij~=0)=1;
NPCR=(sum(sum(sum(dij)))/(mp*np*op))*100;
set(handles.text77,'string',num2str(NPCR));
```

c. UACI Enkripsi

```
plain1=double(aa3);
cipher1=double(BB111);
[mp1 np1 op1]=size(plain1);
[mc1 nc1 oc1]=size(cipher1);
UACI=(sum(sum(sum(abs(plain1-cipher1))/255)))/(mp1*np1*op1))*100;
set(handles.text33,'string',num2str(UACI));
```

d. UACI Dekripsi

```
plain1=double(aa3);
dekrip1=double(BB22);
[mp1 np1 op1]=size(plain1);
[mc1 nc1 oc1]=size(dekrip1);
UACI=(sum(sum(sum(abs(plain1-dekrip1))/255)))/(mp1*np1*op1))*100;
set(handles.text79,'string',num2str(UACI));
```